Phytoplankton Functional Groups Automatic Recognition using Convolutional Neural Networks

Robin Fuchs[*], Melilotus Thyssen[†], Gérald Gregori[‡], Denys Pommeret [§], Samuel Soubeyrand [¶]

June 19, 2020

Abstract

Phytoplankton diversity and distribution are complex and crucial to solve. This relies on adequate datasets to solve their heterogeneity and variability. Flow Cytometry has enabled to collect significant phytoplankton datasets in quasi-real time. It counts the number of cells belonging to each phytoplankton species or group of species sharing common optical characteristics, called phytoplankton functional groups (PFT). However, raw data have to be manually post-processed which is very time-consuming and may be a source of classification errors for which we provide here an estimation.

As a result, there were many recent propositions to make this classification automatic based on Machine Learning methods. However, most of these methods presents either a long training process, are unable to predict the class of very small cells or need to manually design some features. Our method is based on recent advances in Deep Convolutional Neural Networks and classify Mediterranean phytoplankton functional groups thanks to optical curves provided by a pulse shape recording flow cytometer. We show on a novel long and high frequency time series that our model presents a high accuracy that seems not affected by infra-day cells cycles [exaggerated for the moment]. It also exhibits a significant per class precision despite of the extreme imbalanced nature of the data. Finally, the model seems also to be able to reuse features learnt on some particles to predict the class of particles coming from another location. Thus, it might be redeployed in other Mediterranean geographic zones with minor adjustments.

[*] robin.fuchs@mio.osupytheas.fr

[†] melilotus.thyssen@mio.osupytheas.fr

[‡] gerald.gregori@mio.osupytheas.fr

[§] denys.pommeret@univ-amu.fr

[¶] samuel.soubeyrand@inra.fr

1 Introduction

The datasets in Oceanography are conspicuous for being large (with a high number of observations) and of high dimension (comprising a substantial number of variables). This makes oceanography a very suited field for statistical analysis. This is particularly the case of phytoplankton-related data. Phytoplankton, under a unique denomination, actually contains several thousand species. Some of these species share common optical characteristics such as the size, the pigment content, the habitat or biogeochemical features and can be gathered in groups of species called phytoplankton functional groups (PFTs).

The study of the PFTs is of primary importance given that the contribution of phytoplankton to marine primary production, the amount of underwater dissolved CO_2 absorbed by phytoplankton cells per unit of time, is equivalent to all of the primary terrestrial production. This is the case even if phytoplankton cells represents less than 1% of the terrestrial autotrophic biomass [9]. This means that phytoplankton has a very rapid growth capacity (it can divide several times a day) [add source] and therefore highlights the need for high frequency observations to encompass their morphological diversity and to correctly assess the classification power of statistical models.

The development of pulse shape recording flow cytometry has made possible a vast automated data acquisition on PFTs given that a flow cytometer (FCM) can process up to 10,000 cells per second. From each cell in the sample, the flow cytometer is able to generate a set of curves which represents the optical profile of their scatter and fluorescence. After reprocessing, the oceanographers manually determine the different functional groups existing in the data using a collection of two dimensional cytograms. These operations are however very time-consuming and their automation seems today necessary.

Automating classification tasks using statistical methods, *i.e.* designing methods to automatically assign a label to an observation based on its features, has received special interest in oceanography for the last twenty years. Concerning phytoplankton classification (also called phytoplankton group gating), most of the effort seems to have been spent on image processing and computer vision. One can for example cite the count of coccoliths using shallow neural networks in the seminal work by Beaufort and Dollfus (2004) [3] or more recently the works by Dunker & al. (2019) [7] or by González & al. (2019) [11] based on Residual Neural Networks and Transfer Learning [22].

Automated recognition of PFTs from flow cytometric optical curves was less explored with some notable exceptions such as Malkassian and al. (2011) [17] which classifies the species of phytoplankton by plunging these curves into a Fourier basis and calculating distances between them. This is also the case of the R package Rclustools which implements existing and new statistical methods as recent developments by Wacquet and al. (2013) [21] to deal with the optical curves in either a supervised, a semi-supervised or an unsupervised manner.
Other works using Flow Cytometry data but not on phytoplankton cells also exist as in del Barrio and al. (2019) [1] where the authors create curves templates and classify the observations curves with respect to these templates using Wasserstein distance and optimal transport.

Contrary to these works, we aim to automate the cells count of functional groups and not the count of the species. This task might be more challenging as the morphological diversity among a set of species is at least superior to the morphological diversity among each species. The raw data used for this classification are the flow cytometric curves (FCCs) of each cell going through the flow cytometer. Compared to phytoplankton images, the size on disk of the 5 curves per particle is approximately the same : 4.5kb per observation. However, classifying and then count the cells using FCCs presents a real advantage because it can deal with particles smaller than 20 microns. On the contrary, the image resolution is not sufficient to obtain proper images of such small particles. This is a real issue considering that the very vast majority of phytoplankton particles in the *in-situ* samples are smaller than this threshold. The second main advantage is the shorter training process because of the absence of transfer learning procedure [18], contrary to the images that require to fine-tune very heavy networks such as Residual Networks [12].

In this work we use recent advances in now standard Deep Convolutional Neural Networks. These networks have known a very fast development in image recognition and computer vision during the last ten years starting with the seminal works of Krizhevsky and al. (2012). [14]. This class of networks is here applied in order to automate PFTs classification from FCM data. This is a difficult task because of the morphological diversity evoked earlier, of the format of raw data and due to the dramatically imbalanced nature of datasets. These aspects also make manual classification challenging for oceanographers and make it nearly impossible to perform gating with hourly data collected during several months. Although manual gating of cells groups belonging to several PFTs is well transmitted between experts with a long time accepted consensus on groups identification from cytograms, differences between users exist. Assessments of the diversity of experts classifications are hardly performed in flow cytometric studies.

We begin this work by presenting the data used and our methodology. The data come from the FUMSECK campaign (DOI 10.17600/18001155) which occurred in off the coast Ligurian Sea waters and from the Sea Water Sensing Laboratory @ MIO Marseille (SSLAMM) in coastal Mediterranean waters. Details are then given about the manual gating process, about the how an estimation of its error can be obtained and a description of our predictive pipeline.
In section 3, the assessment of the manual gating error is presented and compared to the error obtained by our network and other Machine Learning models. Evidence are then given that our model can be used to predict long and high-frequency time series and that it seems not affected by seasons nor infra-day cellular cycles [exaggerated for the moment]. Finally, we give insights about the robustness of our model by using the features learnt on a dataset and predicting observations from a different data source. Section 4 closes this work by analysis the limits of our work and giving axes for future research.

2 Material and Methods

In this section, we first detail how the data are collected, manually gated by FCM experts and the strategy used to quantify the resulting error. Then we dwell on the data pre-treatment and the specification of our predictive pipeline.

2.1 Data presentation

All the data presented and used here have been acquired with a Cytosense © (Cytobuoy$^{\text{TM}}$, N.L.) pulse shape recording flow cytometer. The general principle of this flow cytometer is as follows: a sample of water to be analyzed is pumped in the flow cytometer from the environment studied. The cytometer aligns the cells in suspension in the sample one by one thanks to the generation of a laminar net that generates an hydrodynamic focus and makes them pass in front of a laser beam. This method allows each cell or particle to pass in front of a coherent laser beam. The interception of this beam by the particles generates a set of optical profiles of diffusion and natural fluorescence when pigments are present. These optical profiles take the form of a set of curves. The frequency of acquisition of curves of the cytometer is 4 MHz which gives it a collection capacity of barely 10,000 cells per second. The datasets generated are therefore rapidly massive.

In our case the cytometer issues five curves: the curvature curve, the red (FLR) and orange fluorescence (FLO) curves and the forward (FWS) and Sideward scatter (SWS) (other cytometers might have a different number of curves). Because of the high proportion of background and electronical noise particles, a common practice is to perform two types of data acquisitions using two Red Fluorescence (FLR) thresholds: a low one here denoted FLR6 and a high one denoted FLR25. The lower threshold enables to count the particles that have the smallest total red fluorescence (which are also the smaller particles). The FLR25 enables to clear out the small particles in order to better count the biggest ones. Then the total count by class is simply the count of low fluorescent particles in the FLR6 file and of high fluorescent particles in the FLR25 file.

Hence, each observation is made of five curves which length is closely linked to the size of the particle (the bigger the particle the longer the sequence). In order for all sequences to be comparable, the curves have been interpolated using quadratic interpolation. Using truncated and padded with zeros sequences as in Natural Language Processing (NLP) has also been implemented and led to poorer performance.

We have chosen a fixed length of 120 values for each curve of all observations that corresponded approximately to the third quartile of the distribution of particles curves lengths. The influence of this length choice on performance has not however been tested and could be in further research. Once the curves resized, one obtains for each observation five curves of length 120 or alternatively a 5 × 120 matrix which a representation which is given in Figure 1.

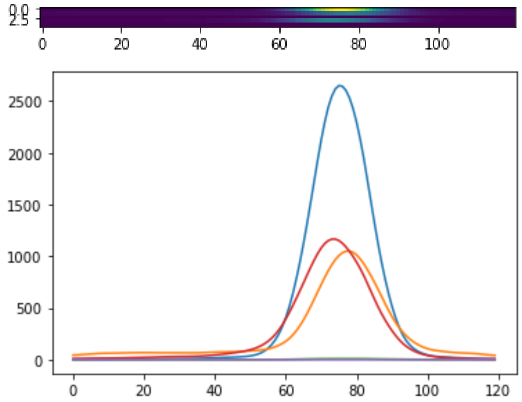

Figure 1: Matrix and curves representation of the five curves of an observation

Concerning the origin of the data, two main sources of data have been used in this work: The FUMSECK Campaign data and the Endoume Marine Station (SSLAMM) data.
The FUMSECK campaign was a cruise that took place from April, 30, 2019 to May, 05, 2019 off the Ligurian Sea. During this campaign, 610 cytometer flow samples have collected at a depth of two meters and represent barely ... particles.
The SSLAMM time series has started in September 2019 at the Endoume Marine Station in Marseille. Some sea water is continuously pumped at 10 meters from the coast and at a depth of 3 meters and is delivered unaltered into the laboratory buildings where analyses are conducted. A cytometer is making acquisitions into this water flow every two hours, hence generating 12 FLR6 and 12 FLR25 files per day. Such an infrastructure has no equivalent in Europe and to our knowledge in the World [is it true ?].

All the datasets presented here share the same functional groups nomenclature. It is composed of six phytoplankton classes: Prochlorococcus, Synechococcus, Picoeukaryotes, Nanoeukaryotes, Cryptophytes and Microphytoplankton. Two additional classes have been added: noise particles and airbbubles. Noise particles are actually very heterogeneous: detritic particles *stricto sensu*, phytoplankton predators that present a fluorescence, phytoplankton cells in decomposition... etc. The airbubbles begin to appear in the cytometer when some manual interventions have to be performed. It is then more a class helpful for machine monitoring rather than for purely scientific purposis. As no definitive standardized PFT nomenclature exists for the moment, this nomenclature might evolve in the future to meet interoperability needs.

In order to fit the models, for each application the data were split in three parts: the training the set, the validation and the test set. The models were trained on the training set, the choice of their hyper-parameters done according to the performance on the validation set and the final performance assessment on the test set.

2.2 Manual gating methodology

In this subsection, the manual gating methodology used by experts is described. Then, we present the methodology of the experiment conducted to assess manual gating ranges of errors.

The raw data extracted by the Cytosense^{TM} FCM and the Cytoclus4 © software are series of 5 curves exhibiting variable lengths. These raw pieces of information are hard to visualize and it is to difficult to use them as such for classification. Experts prefer rather to summarize this signal by computing a single value for each curve instead, typically the area under the curve. Doing so, one obtain a point of dimension five for each observation. The dataset can under this form be represented by a series of 2D projections. For example, using the area under the curve for all of them one can plot the Total Red Fluorescence against the Total Orange Fluorescence as in Figure 3, then the Total Red Fluorecence against the Total Forward Scatter and so on.

This is this collection of 2D projections that the experts use to perform the manual gating of all the cells contained in each sample. The samples treated are very imbalanced between PFTs. For example, the ratio between the most and the less represented class in our data ranges between 10^4 and 10^5. Hence, the less represented particles can be "hidden" by the most represented ones on the 2D scatter plots used by cytometrists to identify the PFTs.
This can create significant biases in the estimated count of these classes. For instance, in the FCM datasets coming from Mediterranean waters there are typically at most a few dozen of microphytoplancton particles while there are dozen of thousands of synechoccocus particles. Hence, misclassifying 10 particles of microphytoplancton could result in a 30% error rate while misclassifying 10 particles of synechoccocus would be negligible.

This issue is a type of statistical data-contamination and can have significant effects on the patterns learnt by Machine Learning algorithms. Without any estimation of this contamination, it is impossible to disentangle the errors coming from the data from the error coming from the training process. We could not find an estimation of this phenomenon and conducted a small study to give a raw assessment of the inter-individual manual classification errors.

In our experiment, we have asked five experts to classify SSLAMM data coming from three samples acquired at different seasons and time of the day. They were given a list of the PFTs groups existing in those samples. Once they have performed the classification, it is possible to measure the standard errors in the count of each functional group which gives a good idea of the potential biases.

In parallel, they have been asked to gate two additional acquisitions in which more cells from under-represented PFTs have been artificially added. The interest of this second set of gatings was twofold.
First, by giving the classes an almost equal visibility on 2D cytograms, we can test that the under-representation of those cells is responsible for higher errors, which is our base hypothesis. If not, this could mean that these classes are also more difficult to identify intrinsically than highly represented PFTs.
Secondly, it enables to create a robust training set for our network by keeping only the particles that were given the same labels by all experts. Using rebalanced datasets for this goal rather than genuine ones avoids to make the experts label hundreds of files to have enough instances of low represented PFTs.

2.3 Prediction pipeline presentation

The core of the predictive pipeline is a Deep Convolutional Neural Network initially designed for image recognition. The general idea of such a network is to learn a series of filters that detect some patterns in images and help to discriminate between the classes. More formally, these filters are tables of coefficients iteratively used to compute convolutional operations on the data going through the layers. Compared to Dense layers the convolutional ones relies on the assumption that regions in the images conveys useful information and that close pixels often carry very redundant information. As a result, the total number of parameters of the model is reduced and the training of the model is kept tractable even for big three color channels (Red, Green, Blue) images. Once the features extracted by the convolutional networks, one can use Dense layers at the end of the network to perform the classification itself, which is what is conducted here.

A $l \times L$ color image is represented by $l \times L$ coefficients for each of the three RGB color channel i.e. $l \times L \times 3$ integer coefficients ranging from 0 to 255. In the case of black and white images it is a $l \times L \times 1$ or $l \times L$ table. In our case, our network is not applied to images but to 5×120 matrices of float coefficients which therefore share the same shape as black and white images. Hence, the same networks as for images can be used with minor modifications.

An alternative option rather than using the five stacked raw curves is to generate the images of each curve and to use these 5 images per observation as input. Yet, taking the curves images actually dilutes the signal held by the curves among a very substantial number of white pixels. The signal was then too hard to perceive for the network and resulted in network training failures.

Thus, we decided to keep the matrix representation that is the raw signal itself (up to a quadratic interpolation) rather than manually designed features. We expect this very rich signal to be highly efficient for classification purposes as information used by oceanographers is much more simplified and enable the manual classification. The model architecture is presented $model_spe$. Features are first extracted by three convolutional layers. Then, local averages of the coefficients are taken by a Global Average Pooling layer relying on the same idea that a part of the signal is redundant if taken at a too fine level. These "averaged features" are then treated by a series of dense layers. The dropout layers enable not to train every neuron of the layer. It avoids the network to become too specialize over a dataset, which is known as "model overfitting" in Machine Learning.
At the end of the dense layers a softmax layer is computing the probabilities that an observation belongs to each class and the loss of the model is computed.

The loss is measuring the gap existing between the class probabilities that the model outputs and the actual class of the observation. This gap represents an error that is then back-propagated to update the parameters of the network accordingly. The most common loss used for single-label multivariate classification is the negative log-likelihood or categorical cross-entropy (negLL). Its expression is given by :

$$negLL = -\sum_{k=1}^{K}\sum_{i=1}^{n}(y_{i,k} * log(\hat{p}_{i,k}))$$

with $i \in [1, n]$ the observation index, $k \in [1, K]$ the class index, $y_{i,k}$ equals 1 if observation i is in class k and $\hat{p}_{i,k}$ the probability that observation i is in class k predicted by the model.

From the expression, it appears that this loss gives the same weights to all errors whatever the classes of the observations. This is not particularly suited for very imbalanced datasets as the network tends to focus only on accurately predicting the highly represented classes to ensure a good average accuracy. In this respect, three extensions of this loss have been used here: the weighted categorical loss entropy, the focal loss [15] and the class-balanced focal loss [5] in order to achieve a good overall accuracy but also a good per-class accuracy.
The weighted negative Log-Likelihood loss (WnegLL) is a straightforward extension of the negLL which gives more weights to the errors performed on some classes. In general the less represented classes are chosen to be overweighted. However, this approach is very sensitive to the way the weights are computed [add source]. The most common practice is to set those weights to $\frac{1}{n_k}$ or to $\frac{1}{\sqrt{n_k}}$, with n_k the number of observations in the class k. Intuitively as our data are extremely imbalanced, using $\frac{1}{n}$ might lead to too small weights for the most represented classes and we expect the second one to perform better.

Over-weighting very unrepresented data hinges on the hypothesis that rare classes are difficult to predict. This claim can be justified by the fact that bigger particles tend to present a wider range of morphological diversity and that they are also the least represented PFTs. However, in order not to rely on this hypothesis nor on the parametric form of the weights formula we have used the newly introduced focal loss (FL) its generalization, the Focal Class-Balanced loss (FCBL), which have the following expressions: [check the formulas].

$$FL = -\sum_{k=1}^{K}\alpha_k\sum_{i=1}^{n}y_{i,k}(1-\hat{p}_{i,k})^{\gamma}\log(\hat{p}_{i,k})$$

and

$$FCBL = -\sum_{k=1}^{K}\frac{1-\beta}{1-\beta^{n_k}}\sum_{i=1}^{n}y_{i,k}(1-\hat{p}_{i,k})^{\gamma}\log(\hat{p}_{i,k})$$

n_k is the number of observations of class k, α is a class-dependent weight acting in the same spirit as the WnegLL weights. γ is a focusing parameter, it defines how little the contribution of easy-to-predict observations is. β controls how the re-weighting depends on class frequency: $\beta = 0$ corresponds to negLL and $\beta = 1$ correspond to WnegLL with the weights equals to $\frac{1}{n_k}$.

From the expressions below, it appears that the focal loss decreases the contribution of easy well-classified observations *i.e.* the observations that exhibit a $\hat{p}_{i,k}$ close to one. It is done through the training thanks to the "modulating" factor $(1 - \hat{p}_{i,k})^\gamma$ combined with some weights α as in the WnegLL. The FBCL automates in some way the choice of α, but also relies on a new parameter β. We are implementing the three losses, give the hyper-parameters value chosen in Appendix B and compare the performances obtained in the results section.

The loss is a key component of the model and is the main way we have decided to treat the fact that the data were very imbalanced. In addition to the loss, undersampling methods have also been used to slightly reduce the gap between PFTs in order to have both enough instances per class and a tractable total number of observations in the dataset. Random undersampling strategy, *i.e.* picking a random subset of observations in the most represented class was used here given that it gave similar performance results as more advanced under-sampling techniques such as Edited Nearest Neighbors or Tomek's links [2]. The use of these last techniques was in addition not straightforward due to the very particular functional form of the data.

Beyond the choice of the loss, an important choice is the one of the optimizer which deals with how the parameters of the network are updated with respect to the loss. We have here benchmarked two optimizers: Adam and its extension Ranger. Ranger comes from the combination of two very recent publications: RectifiedAdam [16] and Lookahead [23].

In order not to be stuck in bad local maxima, it is a common practice to slowly update the parameters of the models at the beginning of the training, where really promising parameters regions are not for the moment identified. This adaptation rate of the parameters with respect to the loss is called the learning rate of the model and is hence often chosen to be small in the early stages of the training process [19]. Radam adapts the learning rate to avoid the learning rate variance to grow too substantially, which according to the authors is often very detrimental to the learning process. On the other hand, Lookahead enables the network to get a better understanding of the loss topology. In order to do so, two sets of weights are designed by Lookahead: a faster set of weights that are frequently updated to "explore" the loss surface and a slower set of weights (less frequently updated) to ensure the stability of the learning process. The faster set of weights is updated using not all the data but only a set of several observations batches to get a raw idea of promising regions to explore. In the ranger case those fast weights are updated thanks to the Radam optimizer.

As for the losses and most of the parts of the neural networks, the behaviour of the optimizer is also ruled by a set of hyper-parameters that need to be chosen by the user. The number of possible combinations is far too high for all the combinations to be tried and then pick the best network specification.
One popular approach relies on Bayesian Hyperoptimisation algorithms [4] which are implemented in our case in the Python library Hyperas (Hyperopt for Keras). The idea of Hyperoptimisation methods is to consider hyperparameters as statistical random variables with a prior and to identify posterior regions that presents a low loss value. Hence, some draws are taken from the prior distributions, the model is evaluated and low loss regions are identified and focused on. It avoids to spend very significant computational efforts on non-promising regions of the hyper-parameters space as it is often the case using standard line search.

Once the network has output its predictions, we perform of a post-processing to account for the special status of the noise class. The noise particles are defined by the fact that they are not phytoplankton cells rather than as a biologically consistent class. Conceptually by creating this noise class, we are here making a two stages procedure in a single step. The first stage would be to predict if the particle is a phytoplankton cell or not. If it is not, the particle would be classified as noise. If it is, then a second step would be performed to determine its class among the other classes of the nomenclature.

Our experiments show that in a vast majority of times, when the network is not sure about the class of a particle it is because it is not a phytoplankton particle but a noise one.

As a result, all the particles presenting low predicted probabilities have been assigned the noise label rather than the label of the highest probability class. Low confidence probabilities threshold have been tuned on a separate dataset. This post-processing is not theoretically very well-funded but really gave a significant performance gain. Re-designing the classes of the nomenclature to break down the noise class into several more relevant ones would be more rigorous. On the other hand, using a one-vs-all probability output layer rather than a softmax at the end of the network could enable to implement the idea of the two-stage procedure and will be more founded [give more details here].

3 Results on in-situ data

This section presents an estimation of the manual gating error range and puts it in perspective with the results obtained by our model in different contexts. Three cases of application are considered. First, the model is benchmarked again other models on the FUMSECK campaign data in order to illustrate its predictive power. Then, predictions are made upon samples acquired at the SSLAMM in order to show the invariance of our network to seasonality and infra-day shape changes of the cells. Finally, the model was trained on FUMSECK data to predict Endoume samples to illustrate its generalisation power.

3.1 Manual gating error estimation

[This experiment has not been performed yet, the text will change accordingly]

The results of the manual gating on the three SSLAMM acquisitions is given in Figure 3.1. The "mean count" line gives the per class count obtained in average by the five experts and the next line gives the standard errors of these counts. Hence the closer to zero the standard error is, the more the classification is the same for all researchers for this class.

PFT	Airbubbles	Cryptophytes	Microphytoplankton
Mean count			
Standard Error			

PFT	Nanoeukaryotes	Noise	Picoeukaryotes
Mean count			
Standard Error			

PFT	Prochlorococcus	Synechococcus
Mean count		
Standard Error		

[Add comments about the two additional gated acquisitions and how the number of cells per class seems to influence the std].

3.2 Model benchmark on FUMSECK data

[Should we have a robust dataset for FUMSECK data also ?]

In this section, we train the our model over 25.000 observations [to precise] taken from FUMSECK data. The validation is made of ... particles and the test set of ... cells. The repartition of the samples between the three sets was perfectly random.

The CNN has been benchmarked against other supervised models in order to illustrate its performances. The algorithms compared are Light GBM (LGBM) [13], Feed Forward Neural Network (FFNN) [10], the k-Nearest Neighbors (kNN) [6] and Support Vector Machines (SVM) [20]. LGBM has been chosen because it is in practice very used by Machine Learning practitioners in several fields of application and has won recent several Kaggle challenges (as it was the case of Random Forests models earlier on). The last three models have been picked as they were the ones implemented in the RclusTool package, which is a package implementing Machine methods applied to flow cytometry data.

However, these models could not process the raw signal as it is the case in our model and there is a need to manually compute some features. The presented results have then to be considered by keeping in mind that the choice of the features created from the signal highly influence the performances of the models. We rely on the 10 features per curve created by default by the

CytoClus4©software. The feature list is given in Appendix C.

The metrics reported for each class and each algorithm are the precision and the recall. The precision is the proportion of particles actually belonging to class k among all those identified as belonging to class k by the algorithm. The recall is the proportion of particles effectively belonging to class k among all the particles of class k existing in the dataset.
There is a precision / recall arbitration and obtaining precision and recall close to 1 constitutes the horizon of any supervised algorithm. For example, an algorithm that would predict "noise" for all particles would have a recall of 1 and relatively poor precision for the category "noise".

The following tables report the results obtained by the five models for each data class.

PFT	Airbubbles		Cryptophytes		Microphytoplankton	
Model/Metric	Precision	Recall	Precision	Recall	Precision	Recall
CNN						
LGBM						
FFNN						
kNN						
SVM						

PFT	Nanoeukaryotes		Noise		Picoeukaryotes	
Model/Metric	Precision	Recall	Precision	Recall	Precision	Recall
CNN						
LGBM						
FFNN						
kNN						
SVM						

PFT	Prochlorococcus		Synechococcus	
Model/Metric	Precision	Recall	Precision	Recall
CNN				
LGBM				
FFNN				
kNN				
SVM				

3.3 Prediction of Endoume Time Series data

This section illustrates the ability of our model to be deployed in production and to provide accurate estimates of the count of each class on a daily basis. The model was used to predict the count of each PFT contained in each FLR6 and FLR25 files. The time series aspect of the data is here not taken into account and all files are treated as independent points. The training set, validation set and test set were made of respectively 12, 2 and 2 acquisitions (hence 24, 4 and 4 files) chosen to be representative of different months and different time within the day. Once the best specification chosen, the model was applied to the whole series for prediction. Figure 3.3 presents the time series obtained with manual counts and automatic count for the synechococcus, the noise and the nanoeucaryote particles.

[labels too small and the end of the time series is not predicted]

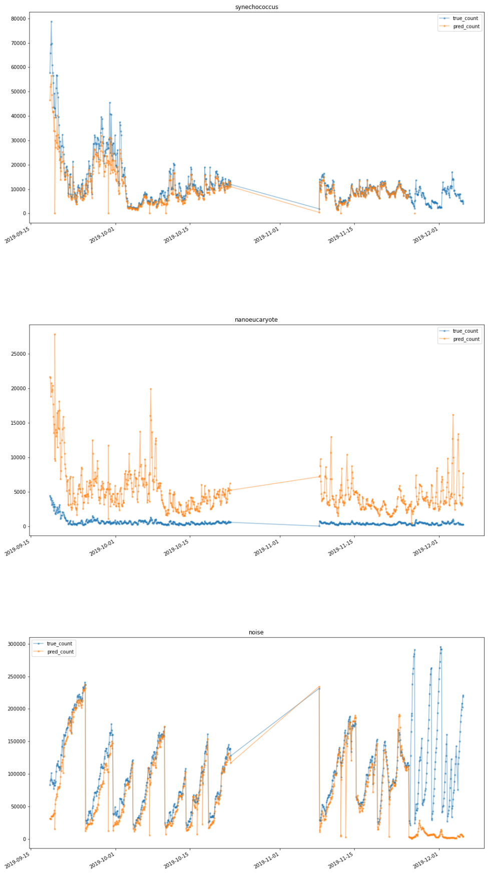

Figure 2: Manual and automated counts prediction for synechococcus, nanoeukaryotes and noise particles respectively

The noise and synechococcus particles are well counted whereas the model tends to largely overcount the nanoeukaryotes. Looking at the confusion matrix, it reveals that a part of the noise is actually taken to be nanoeukaryotes [to

check]. [It advocates once again for a nomenclature change].
What is striking is that the quality of the predictions does not seem to vary with the hour of the day [provide zoomed image to check] nor the month on which there are performed. It means that our predictions are not influenced by the cell divisions that might occur at an infra-day frequency.

From these predicted time series, it is possible to compute the total diffusion and fluorescence of all the PFTs. These quantities are particularly useful for oceanographers to assess local carbon primary production. [Plot the estimated diffusion and total fluorecence over the period].

3.4 Estimation of the generalization power of the model

Finally, we provide an illustration of how general the features learnt by the model are by choosing two different data sources for training and testing the model.

The model is here trained on SSLAMM data sampled a few meter from the coast to predict FUMSECK counts which data are sampled further away off the coast. The SSLAMM training, validation and test sets are the same as in the previous section. Figure 3 presents a 2D cytogram of the Total Red Fluorescence (area under the curve) as a function of the Total Orange Fluorescence. This representation is often used by cytometrists to separate ... from ...

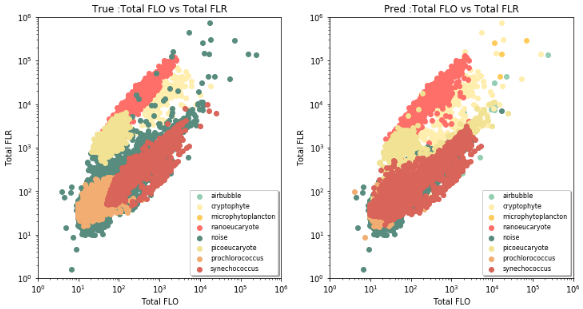

Figure 3: Manual vs. automated count

[Image to replace with the new one and then add interpretation]

A common way to perform and visualize manual gatings for cytometrists is to draw polygons, called selection sets or decision boundaries, around the identified groups. We reconstitute these selections sets, by computing the convex hull over our prediction and plot the actual versus the predicted selection sets. As by definition the convex hull is very sensitive to outliers, the predicted selection sets are computed without considering the 10% most extreme points for each class. Figure 4 present the predicted versus manual selection sets.

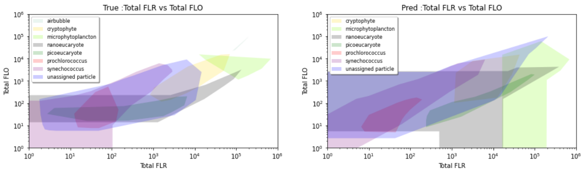

Figure 4: Comparison of the manual and predicted decision boundaries over a FLR6 file

[Change with the new specification boundaries]

4 Discussion

Our work aimed at providing a better understanding of the magnitude of manual classification biases. It confirms that less represented groups suffer more from these biases than the best represented ones [to check]. This highlights the need for data reviewed by several experts in order to obtain good ground-truth data for model training. Such multi-reviewed datasets are more and more popular in the Machine Learning community, the best example being the ImageNet repository [8]. As a result, we call for the creation of an equivalent repository for cytometric curves PFT recognition.
This is all the more so necessary that the prediction error obtained by our network lies in the same error range as manual classification [to check]. Hence, better data may be even more useful than exploring better model specifications in order to achieve better performance.

Through this work we propose a full PFT prediction pipeline able to make quasi-real time PFT identification at the cell level. The total training time of our model is of less than a minute for a training set of nearly 45,000 observations on Google Collab GPU [give more details about machine hardware] in contrast to several hours or days of Residual Networks transfer learning on images as in González & al. (2019) [11]. The data pre-processing and in particular the interpolation of the curves is actually the slowest part of our automated pipeline (between 1 and 2 minutes) whereas the prediction themselves take only a few seconds. This is explained by the fact that the curves are for the moment interpolated in a sequential manner, observation per observation. More efficient methods have to be implemented to reduce this computational bottleneck.

Thanks to new software developments, the pipeline will soon be able to feed the predictions back into the cytometric software CytoClus©and enable the oceanographers to manually modify the automated selection sets of the PFTs that seem erroneous to them. In this respect, our pipeline could also be used as a turnkey pre-gating tool made to speed up the manual classification tasks.

Concerning the network itself improvements are possible. Our methodology is based on a "classify and count" approach which is strongly criticized by González & al. (2019) [11]. Indeed, our pipeline attributes each cell to a PFT and then count the number of cells in each PFT. González & al. (2019) [11] present a reformulation of the cells count problem, called "quantification problem" in the literature, and show that training a series of one-versus-all simple predictors on features extracted from the network is better suited for this task. This will be investigated in future research.

Finally, this work is a preliminary work in order to study the behaviour of the PFT dynamics on Endoume data. Indeed, with a FLR6 and FLR25 acquisitions every two hours, the data load is hardly treatable by a single person and needs to be automated. The data collection process at the SSLAMM is relatively independent of meteorological conditions compared for instance to cruises that can take place only in case of reasonable conditions. These data will thus enable to track organisms reactions over a very wide range of environmental conditions and especially extreme ones during which little is known.

5 Aknowledgments

Cytobuoy for software development of the DAL
Funders of the SSLAMM
ENS Thesis grant ?
ENDOUME DIVERS

References

[1] Eustasio del Barrio et al. "optimalFlow: Optimal-transport approach to flow cytometry gating and population matching". In: *arXiv preprint arXiv:1907.08006* (2019).

[2] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. "A study of the behavior of several methods for balancing machine learning training data". In: *ACM SIGKDD explorations newsletter* 6.1 (2004), pp. 20–29.

[3] L Beaufort and D Dollfus. "Automatic recognition of coccoliths by dynamical neural networks". In: *Marine Micropaleontology* 51.1-2 (2004), pp. 57–73.

[4] James Bergstra, Daniel Yamins, and David Daniel Cox. "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures". In: (2013).

[5] Yin Cui et al. "Class-balanced loss based on effective number of samples". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9268–9277.

[6] Belur V Dasarathy. "Nearest neighbor (NN) norms: NN pattern classification techniques". In: *IEEE Computer Society Tutorial* (1991).

[7] Susanne Dunker. "Hidden Secrets Behind Dots: Improved Phytoplankton Taxonomic Resolution Using High-Throughput Imaging Flow Cytometry". In: *Cytometry Part A* 95.8 (2019), pp. 854–868.

[8] Li Fei-Fei. "ImageNet: crowdsourcing, benchmarking & other cool things". In: *CMU VASC Seminar*. Vol. 16. 2010, pp. 18–25.

[9] Christopher B Field et al. "Primary production of the biosphere: integrating terrestrial and oceanic components". In: *science* 281.5374 (1998), pp. 237–240.

[10] Terrence L Fine. *Feedforward neural network methodology*. Springer Science & Business Media, 2006.

[11] Pablo González et al. "Automatic plankton quantification using deep features". In: *Journal of Plankton Research* 41.4 (2019), pp. 449–463.

[12] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[13] Guolin Ke et al. "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in neural information processing systems*. 2017, pp. 3146–3154.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[15] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[16] Liyuan Liu et al. "On the variance of the adaptive learning rate and beyond". In: *arXiv preprint arXiv:1908.03265* (2019).

[17] Anthony Malkassian et al. "Functional analysis and classification of phytoplankton based on data from an automated flow cytometer". In: *Cytometry part A* 79.4 (2011), pp. 263–275.

[18] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[19] Martin Popel and Ondřej Bojar. "Training tips for the transformer model". In: *The Prague Bulletin of Mathematical Linguistics* 110.1 (2018), pp. 43–70.

[20] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[21] Guillaume Wacquet et al. “Constrained spectral embedding for K-way data clustering”. In: *Pattern Recognition Letters* 34.9 (2013), pp. 1009–1017.

[22] Jason Yosinski et al. “How transferable are features in deep neural networks?” In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.

[23] Michael Zhang et al. “Lookahead Optimizer: k steps forward, 1 step back”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 9593–9604.

A Model specification used

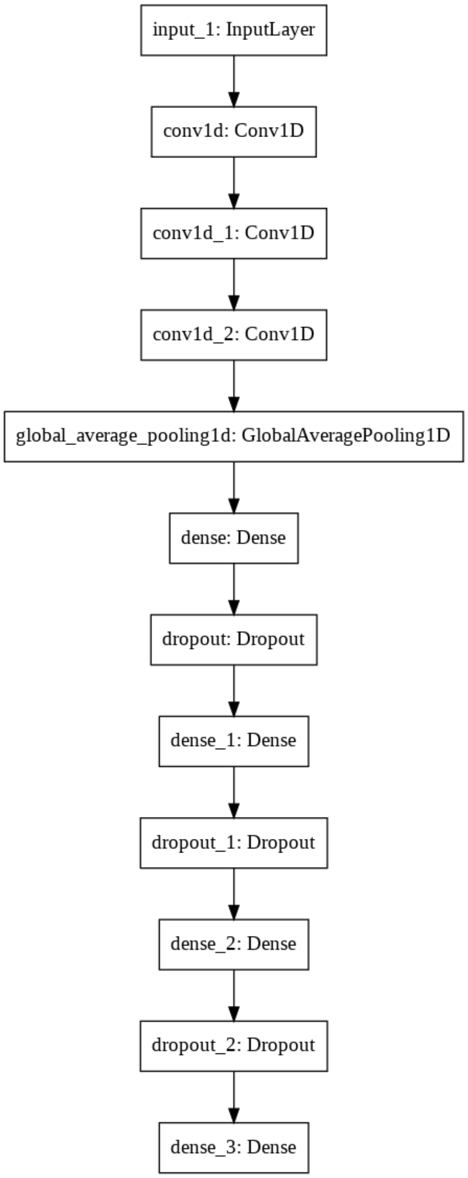

Figure 5: Model specification

B Hyperparameters chosen

This section presents the final architecture choice used for all the results presented below. The best performance was obtained for the focal loss and the Ranger optimizer. The following table presents the main hyperparameters of our model.

Hyperparameters	α	batch size	Dropout	γ	Optimizer	Radam learning rate	Lookahead slow step size	Lookahead Sync period
Value	6.980E-4	256	5.093E-3	2.046	Ranger	3.810E-3	2.074E-1	10

C Listmode features

For each optical curve the CytoClus4© software can output the following features:

- Asymmetry
- Average: The average value of the curve
- Center of gravity
- Fill factor
- Inertia:
- Length: The length of the curve
- Maximum: The maximal value of the curve
- Number of cells:
- Time Of Flight:
- Total: The area under the curve

D Noise selection threshold