Evaluation of Transfer Learning Scenarios in Plankton Image Classification

Francisco Caio Maia Rodrigues[1], Nina S. T. Hirata[1], Antonio A. Abello[1], Leandro T. De La Cruz[1], Rubens M. Lopes[2] and R. Hirata Jr.[1]

[1]*Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão 1010, São Paulo - SP, Brazil*

[2]*Institute of Oceanography, University of São Paulo, Praça do Oceanográfico 191, São Paulo - SP, Brazil*

Keywords: Transfer Learning, Plankton Classification, CNN, SVM, Alexnet, ImageNet.

Abstract: Automated in situ plankton image classification is a challenging task. To take advantage of recent progress in machine learning techniques, a large amount of labeled data is necessary. However, beyond being time consuming, labeling is a task that may require frequent redoing due to variations in plankton population as well as image characteristics. Transfer learning, which is a machine learning technique concerned with transferring knowledge obtained in some data domain to a second distinct data domain, appears as a potential approach to be employed in this scenario. We use convolutional neural networks, trained on publicly available distinct datasets, to extract features from our plankton image data and then train SVM classifiers to perform the classification. Results show evidences that indicate the effectiveness of transfer learning in real plankton image classification situations.

1 INTRODUCTION

Plankton communities form the basis of aquatic food webs and exert a major influence on material cycles relevant to global climate change, such as carbon dioxide and methane. Therefore, it is essential to understand the spatial distribution and temporal variability of planktonic organisms in the ocean. Plankton collection and analysis has been traditionally carried out by net tows and subsequent microscopic inspection of preserved samples. Such approach has led to a significant increase in the knowledge about taxonomic composition and distribution of several plankton groups, but fine-scale sampling is usually not feasible with nets and many fragile organisms are destroyed by collision with the net mesh or disintegrate in fixatives.

Recent advances in digital image acquisition and Machine Learning (ML) techniques have stimulated the application of in situ imaging to generate highly resolved vertical profiles of plankton composition and abundance (counts per volume). While high-quality image acquisition technologies represent the first step in such task, new approaches in ML techniques are in the core of our increasing capability to deal with the complex and highly variable geometry of plankton organisms.

Convolutional Neural Networks (CNN) have emerged as a powerful technique for image classification and its variants are being successfully employed on a variety of classification tasks. The characteristic of being data-driven, not requiring specifically designed features, make them a suitable model to cope with the high variability of plankton species distribution in space and time. However, the training success of such models depends not only on experimentation and adjustment of parameter values but on the availability of large amount of training data. This is a critical point in supervised learning tasks such as classification.

There have been some efforts to make available labeled plankton image datasets. The International Council for the Exploration of the Sea (ICES) initiative, http://www.ices.dk/marine-data/dataset-collections/Pages/Plankton.aspx and the Kaggle's National DataScience Bowl (NDSB) competition, via the In Situ Icthyoplankton Imaging System (ISIIS), https://www.kaggle.com/c/datasciencebowl, are a few of the examples. These datasets may differ largely with respect to plankton composition and image quality. Diversity may originate from differences in locations (geographical and along the water column), in imaging technologies which may target plankton of different size ranges, or even in the goals of the research project. Due to those differences, available

Rodrigues, F., Hirata, N., Abello, A., Cruz, L., Lopes, R. and Jr., R.
Evaluation of Transfer Learning Scenarios in Plankton Image Classification.
DOI: 10.5220/0006626703590366
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 5: VISAPP*, pages 359-366
ISBN: 978-989-758-290-5

datasets may not be directly useful to the application context of a given research program. At the same time, labeling a large amount of data each time a set of observations with new characteristics is available is unfeasible.

One approach to deal with these types of situations is transfer learning (TL) (Pan and Yang, 2010). In ML[1], TL expresses the concept of using or adapting a model induced in a specific context to another context. For instance, using or adapting a model learned using a plankton dataset from Atlantic ocean to classify another one from the Pacific ocean.

In this work we investigate TL applied to plankton classification. Our goal is to develop an algorithm to classify samples in our in-house dataset. Since we foresee different deployment scenarios in the future, we would like to have a data-driven classification approach. Therefore, CNNs appear as an interesting option except by the fact that the small size of our dataset makes training such a model from scratch an unfeasible task and thus we resort to adapt pre-trained models as feature extractors. Taking advantage of the fact that there exists a public available massive dataset of plankton images used in previously mentioned Kaggle's NDSB competition, our approach is to train a CNN, specifically the one proposed by the winning team, using this dataset to extract meaningful features from our smaller in-house built dataset. Although there are differences in the datasets with respect to the classes of plankton species they include, either because a particular species or class in one of the datasets is not in the other , it is reasonable to expect that they could be efficiently classified by the same set of features. To further investigate the quality of the features obtained from this process, we also employ a different CNN trained on this same dataset and on ImageNet (Russakovsky et al., 2015), which contains images from a completely distinct domain. By using CNNs and external domain source datasets, we would like to understand how transfer learning performs and whether an external dataset will help or not the classification of our data.

Plankton image classification using CNNs started to be considered only recently (Al-Barazanchi et al., 2015; Dai et al., 2016; Py et al., 2016) and, in particular, transfer learning of features computed by CNNs (Orenstein and Beijbom, 2017) has not been explored much yet in this context. The present contribution aims to deepen our understanding of transfer learning in planktonic data.

The remaining of the text is organized as follows. In Section 2 we briefly recall the transfer learning formulation and outline the methods to be used in our experiments. In Section 3 we describe the datasets and CNN models to be used. Then, in Section 4 we detail the experiments and discuss the results. We present the conclusions of this work in Section 5.

[1]The concept is used in Psychology and Education Research, as well.

2 METHOD OVERVIEW

Given an input space of observations, denoted as $\mathcal{X}$, and a set of class labels, denoted as Y, classification can be modeled as the problem of predicting a class label y for each instance $\mathbf{x}$ in $\mathcal{X}$. Assuming there is a joint probability distribution p on $\mathcal{X} \times Y$, the minimum error classification can be determined from the conditional probabilities $p(y|\mathbf{x})$. Discriminative approaches in supervised machine learning often tries, for each input $\mathbf{x}$, to approximate their outputs to the probabilities $p(y|\mathbf{x})$.

In classification, $\mathcal{X}$ defines a domain and Y defines a task. Elements in $\mathcal{X}$ usually consist of convenient encodings of objects to be classified and set Y consists of the corresponding class labels for each element in $\mathcal{X}$. For instance, $\mathcal{X}$ could represent the feature vectors extracted from plankton images and set Y could be a set of numbers representing the taxonomy of the different species of plankton.

In many situations, there is no sufficient amount of labeled data to train a classifier in a given target domain. Among the approaches used to handle this type of situation, there is for instance data augmentation (Simard et al., 2003), transfer learning (Pan and Yang, 2010), and bootstrap methods (Hastie et al., 2009). Transfer learning refers to using knowledge obtained from a distinct domain data, and possibly distinct task, to learn the conditional probability distribution of the target domain.

Representations learned by CNNs are reported to be very useful for the classification of data, even in distinct domains (Bengio, 2012; Yosinski et al., 2014). The usual approach to exploit this is to select an intermediate layer as a target layer, freeze it and its preceding layers and adjust the subsequent layers. The earlier the layer chosen, the more general and therefore, more transferable the representation is (Yosinski et al., 2014), but also the more data is necessary to adjust it, since it has a higher dimension. The adjustment of subsequent layers may be done via fine-tuning, continuing training with new samples, or by training an entirely new classifier from scratch using the output of the intermediate target layer as features, which is called (deep) feature extraction. In this work we chose the latter option, using pre-trained CNNs as feature extractors.

The experiments have been designed to answer the following questions:

- how DeepSea trained on ISIIS (NDSB competition) – DeepSea(ISIIS) – will perform on our in-house dataset (LAPSDS)?
- how classifiers using features extracted from DeepSea(ISIIS) will perform on LAPSDS?
- how classifiers using features extracted from AlexNet trained on ISIIS – AlexNet(ISIIS) – will perform on LAPSDS?
- how classifiers using features extracted from AlexNet trained on ImageNet – AlexNet(ImageNet) – will perform on LAPSDS?

In addition to these TL scenarios, we also consider the traditional feature extraction approach that will serve as a baseline. Diagram in Fig. 1 summarizes the scenarios to be evaluated. Four sets of features are extracted from LAPSDS and they are used to train SVM classifiers as detailed ahead in Section 4.

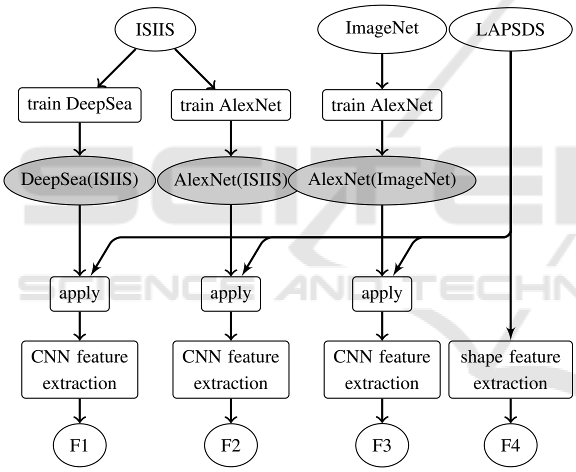


Figure 1: Deep feature extraction scenarios considered here. ISIIS ImageNet and LAPSDS denote image datasets, gray shaded nodes indicate the pre-trained CNNs, and CNN feature extraction consists of extracting the values from a specific layer of a CNN, after a forward pass of samples in LAPSDS.

3 DATASETS AND CNN MODELS

In this section, we describe the datasets and the two chosen CNN models used in the experiments.

3.1 Datasets

3.1.1 LAPS Dataset (LAPSDS)

In situ plankton images have been acquired with a submersible instrument developed at our lab LAPS-IOUSP[2]. The instrument has been vertically deployed between surface and 30m depth off the lab base[3] and gray-scale images were acquired at approximately 15 frames per second, with dimensions of 2448×2050 pixels and resolution of $\sim 5\mu m$. Image stacks belonging to the same vertical profile were converted into video files to mitigate data storage and management. A total of 230,000 Regions of Interest (ROI) were extracted from 16 selected videos and 5175 ROIs were used in the creation of in-house dataset. A labeling process was carried by plankton experts belonging to the same lab.

LAPSDS is composed of 20 classes containing at least 100 samples each, and as expected, the number of images varies from class to class. Table 1 shows the class distribution of the dataset, as well as the name and the identifier number of each class. Instances of some of the classes are shown in Fig. 2.

Table 1: Histogram of classes of the LAPSDS.

ID	H classes	Size	ID	H classes	Size
0	appendicularia_shape_s	216	10	detritus_uf_stick_bw	286
1	appendicularia_curve	114	11	dinoflagellates_tripus_2	242
2	cladocera	435	12	dinoflagellates_tripus	316
3	copepod_calanoid	315	13	nauplii	465
4	copepod_cyclopoida	106	14	phytoplankton_0	259
5	copepod_poecilostomatoida	163	15	phytoplankton_1	127
6	detritus_df_bk	288	16	phytoplankton_5	159
7	detritus_uf_dot_bk	344	17	chaetocero	546
8	detritus_uf_dot_bw	274	18	diatoms_coscinodiscus	120
9	detritus_uf_stick_bk	152	19	shadow	249

In-situ images are prone to natural variability in illumination, turbulent flow and turbidity, among other factors, which may compromise image quality because ROIs from different videos may have different background intensities (see Fig.2). Thus, for convenience, the background of the ROIs have been removed using a technique of background subtraction adapted to deal with illumination changes (Jacques et al., 2006). An example is shown in Fig. 3.

[2]Laboratory of Plankton Systems, Oceanographic Institute, University of São Paulo, Brazil

[3](lat:-23.499913, long:-45.119381)

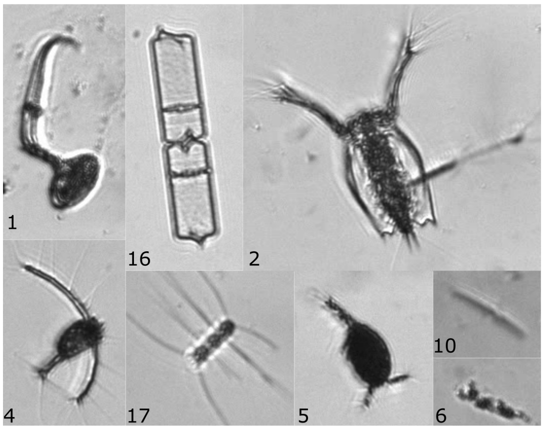
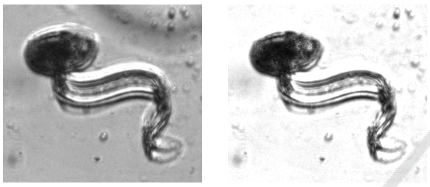


Figure 2: Image sample from LAPSDS. Number on the ROI indicates the class that they belong to.



(a) (b)

Figure 3: Background removal example: (a) Original image, labeled as "appendicularia_shape_s" and (b) result of the background removal of image in (a).

3.1.2 Kaggle's National Data Science Bowl

The National Data Science Bowl (NDSB) was a competition hosted by Kaggle in a collaboration with Oregon State University's Hatfield Marine Science Center. Several research teams competed to develop and train supervised classifiers, given a dataset provided by the Hatfield Marine Science Center(Cowen et al., 2015).

According to the competition organizers, the images were collected in the Straits of Florida using an underwater imaging system called ISIIS (In Situ Ichthyoplankton Imaging System). It captured high-resolution continuous images that were parsed in 2048x2048 pixel frames. The resulting frames were thresholded and segmented. Finally, regions of interest were extracted and became the images that comprise the dataset after being annotated by the Marine Science Center's personnel.

The dataset was divided by taxonomy, behavior and shape into 121 classes. Each class contained between 9 and 1979 individual examples, totalizing 30,336 images.

3.1.3 ImageNet

ImageNet is a dataset that became one of the benchmarks for object classification and detection. It is comprised of over 14 million images divided into 1000 classes hierarchically subdivided (Russakovsky et al., 2015). The classes subjects range from human persons to animals and fungi to everyday objects, constituting a very general dataset. Since 2010 a competition including diverse tasks such as classification and detection on pictures or video on this dataset is held each year.

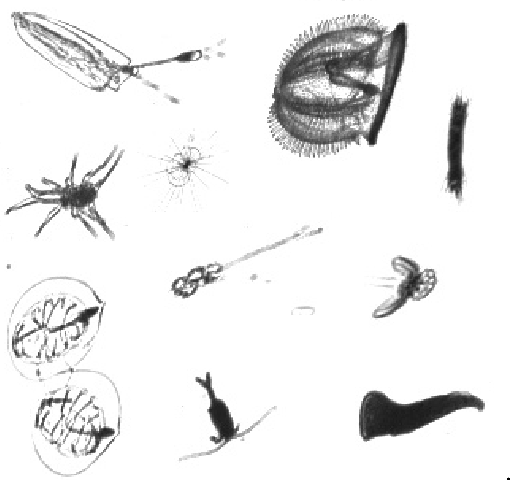


Figure 4: Assorted plankton from the ISIIS dataset. Each sample is from a different class. Note the absence of background.

3.2 CNN MODELS

The two network architectures used in this work are from winning teams in computer vision competitions. They are the AlexNet (Krizhevsky et al.,), from the 2012 ImageNet Large Scale Visual Recognition Competition (ILSVRC), and a model from the "Deep Sea" team, that won Kaggle's ISIIS in 2014.

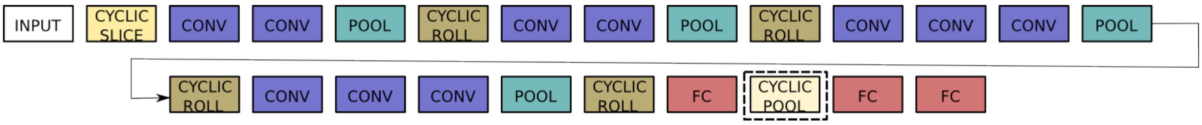
3.2.1 AlexNet

AlexNet is a Convolutional Neural Network model that was introduced in the ILSVRC held in 2012. Under the team name of "SuperVision", it won both the classification and localization tasks by a large margin[4], being the first case of success in applying this kind of model in the competition and establishing a strong trend of its use in the next years.

This model introduced and popularized a lot of novelty features for improving training time, performance and reducing overfitting including, but not limited to: ReLU , Dropout and Local Response Normalization. We refer to the original paper for a more

[4] http://image-net.org/challenges/LSVRC/2012/results

(a) AlexNet



(b) DeepSea's convroll4 network

Figure 5: Neural Networks architectures used in the experiments. Although DeepSea's model is much deeper than AlexNet, it has less parameters (i.e. filters in Convolutional layers and units in Fully Connected layers) to fit during the training. The dashed boxes indicate which layer was used in the transfer learning experiments.

detailed explanation of these innovations and their impact (Krizhevsky et al.,) (see Fig. 5(a) for a representing diagram of the CNN).

We did not explicitly train AlexNet model in the ImageNet dataset, but used instead a pre-trained model with available weights online [5]. In order to feed our images to this model, a couple minor modifications were required, such as converting our one-channel grayscale images to three-channels RGB and resizing, via a wrap padding tactic, to match the expected input.

AlexNet implementation that was trained on ISIIS dataset was heavily based on DeepSea's model, following exactly the same training procedure for both networks (i.e. data preprocessing and data augmentation). Thus, this network's input expects grayscale images with size 95x95 and its final layer contains 121 units.

3.2.2 Deep Sea's Model

Deep Sea was the winning team of the Kaggle ISIIS. They used an approach of ensembling multiple deep learning models with minor differences to improve generalization. We used the most simple model available, consisting solely of a CNN, which here we call DeepSea.

The main innovation brought by the team was a couple of layers designed to increase the network robustness to cyclic variation (Dieleman et al., 2016). In the "cyclic slice" layer the input is rotated four times and processed separately by the network from that point onward. Then, in the "cyclic roll" layer, the feature maps from the four paths are permuted and interchanged. Eventually, in the "cyclic pooling" layer the four network paths are merged again into a single one. We again refer to the paper on this architecture for a more detailed explanation (Dieleman et al., 2016) (see Fig. 5(b) for a representing diagram of the CNN).

[5] https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet

4 EXPERIMENTS AND DISCUSSION

Experiments followed the outline presented in Section 2. Given a pre-trained CNN, the steps to be executed consist of feature extraction, classifier training, and classifier performance evaluation. We describe these steps in the subsequent sections and at the end we present some discussions.

4.1 Feature Extraction

4.1.1 Deep Features

From DeepSea(ISIIS). The features were extracted from the output of the last Cyclic Pooling Layer, as shown in Figure 5(b) highlighted by enclosing dashed lines, resulting in 256 features per images. These features correspond to F1 in the diagram of Fig. 1. In a Cyclic Pooling Layer the effect of rotations introduced by previous Cyclic Slice and Cyclic Roll layers are undone, hence capturing the output from this layer is the most appropriate choice since we can leverage on the learned invariances.

From AlexNet(ISIIS) and AlexNet(ImageNet). From the two pre-trained AlexNet, AlexNet(ISIIS) and AlexNet(ImageNet), features were extracted from the first Fully Connected layer, as shown in Figure 5(a) highlighted by enclosing dashed lines, resulting in 4096 features per image. These features correspond to F2 and F3, respectively, in the diagram of Fig. 1.

4.1.2 Shape Features

We extracted 74 features commonly used in traditional shape recognition procedures. They are divided into the following three categories:

- 54 shape features (area, perimeter, solidity, convexity, etc). Most of the feature descriptors are implemented in the OpenCV library and they are usually presented in automatic plankton classification works that use shape features (Blaschko et al., 2005).
- 10 from Local Binary Patterns (LBP) histograms (Ojala et al., 2000) extracted using a 3×3 window.
- 10 from Haralick descriptors, extracted from the co-occurrence matrix (Haralick et al., 1973).

Shape and LBP features are extracted from the images segmented using Otsu's threshold (Otsu, 1979). Haralick's descriptors are extracted from gray-level images. These features correspond to F4 in the diagram of Fig. 1.

4.2 Classifier Training and Evaluation

To train and evaluate the SVM classifiers with respect to each of the four feature sets, we performed a 9:1 train-test split that preserved class proportions. This split resulted in a training set of 4658 and a test set of 517 samples.

Before training, a data normalization to convert all feature values to the $[0.0, 1.0]$ range was applied to each individual feature of the four feature sets. The normalization parameters were inferred using the training samples in order to not add bias to the classifier. Test samples were then transformed by those same parameters.

Sklearn's (Pedregosa et al., 2011) grid search with cross-validation was employed to explore the space of possible parameters for SVM, namely the kernel type, value of C and, if a RBF kernel was used, γ values. In this work, we considered *linear* and *RBF* kernels, $C \in \{1, 10, 100, 200\}$ and $\gamma \in \{0.01, 0.001, 0.0001, \frac{1}{nf}\}$, where nf is the number of features, this is a common well-known heuristics. The best parameters found for each feature set are displayed on Table 2. The same table also shows the overall accuracies computed on test set.

Table 2: Table summarizing the results obtained from different transfer learning scenarios. The value of 0.0002 for γ was selected because of the $\frac{1}{nf}$ option. Accuracy refers to the test set.

Feature extractor	SVM parameters			Acc.
	kernel	C	γ	
DeepSea(ISIIS)	rbf	100	0.01	84%
AlexNet(NSDB)	rbf	10	0.01	81%
AlexNet(ImageNet)	rbf	100	0.0002	80%
Shape Features	linear	100	-	72%

4.3 Discussions

Global accuracy alone, specially in cases as ours, where the methods present similar performance, is not much informative. To better understand the results, we plotted a *confusion matrix* (Fig. 6) for each feature set.

As it can be seen, the first plot corresponding to DeepSea(ISIIS), the one that achieved the best performance, has a darker diagonal compared to the other plots. Confusion is larger in the last plot, the one that is based on shape features. In general, there is confusion between class 3 (copepod_calanoid) and classes 4 (copepod_cyclopoida) and 5 (copepod_poecilostomatoida), between classes 7 (detritus_uf_dot_bk) and 8 (detritus_uf_dot_bw), and beween classes 9 (detritus_uf_skick_bk) and 10 (detritus_uf_stick_bw). Figure 7(a) presents some examples of copepods subtypes that can confuse the classifiers. The figure is organized in three columns, one for each copepod subtype: column 1 shows four examples of calanoids, column 2 shows three examples of cyclopoida, column 3 shows four examples of poecilostomatoids. Each image is labeled with zero to four colored squares that represent the success of the corresponding classifier in classifying correctly that image. As one can see, the plankton belonging to these classes are similar in several aspects and it is not difficult to understand why these classes cause confounding errors. A similar scenario has been found for detrital particles. Figure 7(b) presents a similar set of images of examples of detritus subtypes (detritus_uf_dot_bk and detritus_uf_dot_bw) that can confuse the classifiers.

Another view of the results is shown in Fig. 8. We present a bar chart displaying the accuracy of each classifier per class. Classes 2, 11, 12, 13 and 18 were well classified by all the four classifiers and therefore they could be considered as the easy classes. On the other hand, classes 4 and 9 are those where most classifiers did poorly, and thus they are the hardest ones. Classes 0 and 1 are those with the largest variation between the best and worst performing classifiers.

Hand designed features performed clearly worse than any of the CNN extracted ones. Although no careful feature selection was performed, it is also true that no careful deep feature extraction was performed. Thus, in a situation where a quick solution is required, making use of a pre-trained CNN could be more effective than using a large set of hand designed feature extractors.

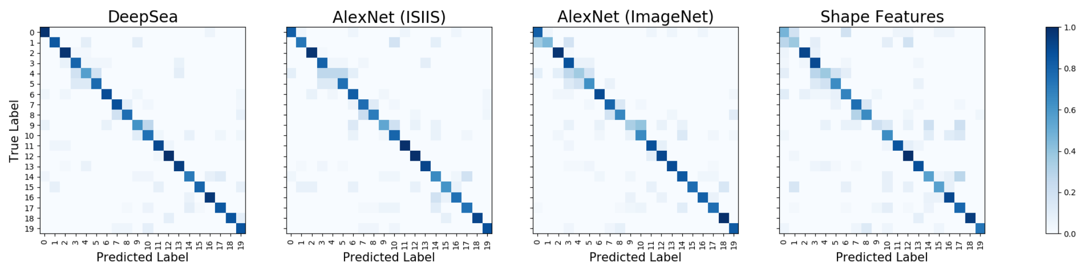
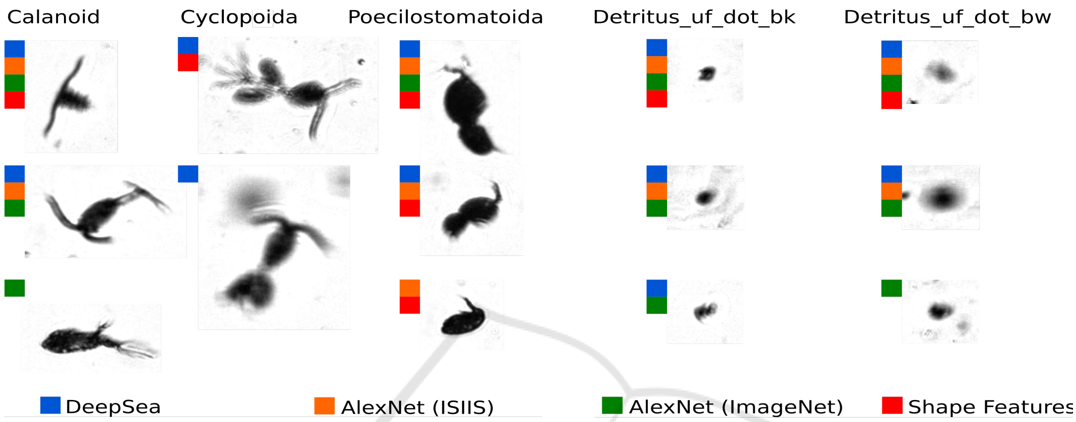


Figure 6: Confusion Matrices.



(a) Cyclopoida, Calanoid, Poecilostomatoida

(b) uf_dot_bk, uf_dot_bw

Figure 7: Two sets of plankton images from confounding classes.

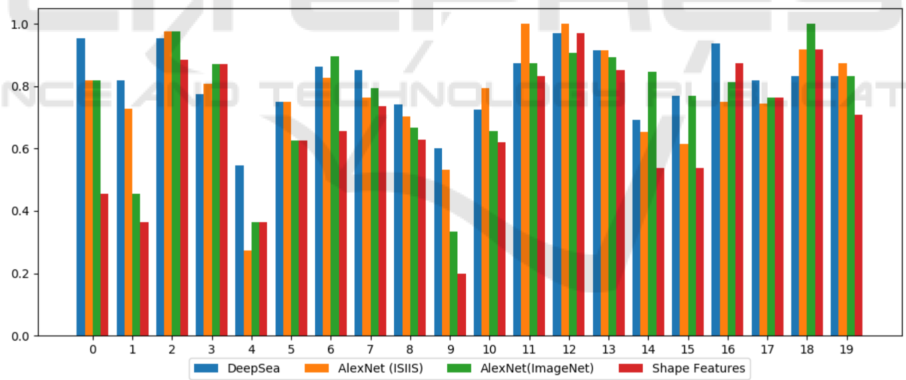


Figure 8: Class accuracy histogram.

5 CONCLUSIONS

We have presented an evaluation of transfer learning scenarios in the context of plankton image classification. We have used CNNs pre-trained on external datasets as feature extractors from our in-house dataset images. In particular, we have considered two very distinct external datasets, one of plankton images (and thus similar to our data) and another of natural images (ImageNet), and the corresponding "winning" CNN architectures. Transfer learning experiments showed that the architecture developed for plankton images (DeepSea) performed better than the architecture developed for natural image classification (AlexNet), even when both were trained with the same plankton image dataset. We also observed that AlexNet trained on natural images performed almost as well as the same network trained on plankton images. These two observations indicate that in transfer learning using CNNs, the architecture may play an important role, even larger than the dataset per se. To complement these observations, it would be interesting to train

DeepSea with ImageNet and evaluate how well it will perform on our data.

Overall, our conclusion is that transfer learning using CNNs as feature extractors might be an effective approach to cope with large scale and high variability of plankton images. However the optimal choice of external datasets and network architectures are still not well understood and should be further investigated in order to push up the accuracy. For future works we plan to experiment with ensemble of classifiers, as already done by DeepSea team and also do data augmentation by blurring the well focused images in a way that resembles the bad focused ones (classifiers usually do not perform well when classifying images with this kind of problem).

ACKNOWLEDGEMENTS

Funding was provided by CAPES (CIMAR 2001/2014), CNPq (565062/2010-7, 311936/2013-0, 400221/2014-4 and 446709/2014-0) and FAPESP (13/17633-6, 2015/01587-0).

REFERENCES

Al-Barazanchi, H. A., Verma, A., and Wang, S. (2015). Performance Evaluation of Hybrid CNN for SIPPER Plankton Image Class. In *2015 Third Int. Conf. on Image Inf. Processing (ICIIP)*, pages 551–556.

Bengio, Y. (2012). Deep Learning of Repr. for Unsupervised and Transfer Learning. In *Proc. of ICML Work. on Unsuperv. and Transf. Learning*, pages 17–36.

Blaschko, M. B., Holness, G., Mattar, M. A., Lisin, D., Utgoff, P. E., Hanson, A. R., Schultz, H., and Riseman, E. M. (2005). Automatic in situ identification of plankton. In *App. of Computer Vision. Seventh IEEE Workshops on WACV/MOTIONS'05*, volume 1, pages 79–86. IEEE.

Cowen, K., R., Sponaugle, S., Robinson, K., and Luo, J. (2015). 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.

Dai, J., Wang, R., Zheng, H., Ji, G., and Qiao, X. (2016). ZooplanktoNet: Deep Conv. Network for Zooplankton Classification. In *OCEANS 2016 - Shanghai*, pages 1–6.

Dieleman, S., Fauw, J. D., and Kavukcuoglu, K. (2016). Exploiting Cyclic Symmetry in Conv. Neural Networks. *CoRR*, abs/1602.02660.

Haralick, R. M., Shanmugam, K., et al. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, 3(6):610–621.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition.

Jacques, J. C. S., Jung, C. R., and Musse, S. R. (2006). A background subtraction model adapted to illumination changes. *Proceedings - International Conference on Image Processing, ICIP*, pages 1817–1820.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Conv. Neural Networks. In *Adv. in Neural Information Processing Systems 25*.

Ojala, T., Pietikäinen, M., and Mäenpää, T. (2000). Gray Scale and Rotation Invariant Texture Class. with Local Binary Patterns. *In Practice*, 1842:404–420.

Orenstein, E. C. and Beijbom, O. (2017). Transfer Learning and Deep Feature Extraction for Planktonic Image Data Sets. In *2017 IEEE Winter Conf. on App. of Computer Vision (WACV)*, pages 1082–1088.

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

Pan, S. J. and Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Py, O., Hong, H., and Zhongzhi, S. (2016). Plankton Classification with Deep Conv. Neural Networks. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 132–136.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best Practices for Conv. Neural Networks Applied to Visual Document Analysis. In *ICDAR*, volume 3, pages 958–962.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Adv. in neural information processing systems*, pages 3320–3328.