Hierarchical Classification of Diatom Images with Transfer Learning

Marija Chaushevska[1] , Ivica Dimitrovski[2], Sašo Džeroski[3] and Hristijan Gjoreski[1]

[1] Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University in Skopje
Rugjer Boskovik bb,1000 Skopje, North Macedonia
[2]Faculty of Computer Science and Engineering, Ss Cyril and Methodius University in Skopje
Rugjer Boskovik bb,1000 Skopje, North Macedonia
[3] Department of Knowledge Technologies, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia

Abstract. In this paper, we address the task of taxonomic classification of diatoms from images taken under a light microscope. The corresponding machine learning task is the task of hierarchical multi-label classification, where the taxonomy plays the role of the label hierarchy. More specifically, an image is assigned several labels, including a single lowest-level taxonomic unit (species), as well as the ancestor ones (family). Since Convolutional Neural Networks are state-of the art in image classification, we apply them to this problem. Since we have a relatively small set of diatom images, we apply the paradigm of transfer learning and use an ImageNet pre-trained InceptionV3 model. We explore two avenues of transfer, one of which is commonly applied, namely to freeze some layers of the pre-trained network and allow for fine-tuning of the unfrozen layers with diatom images. We use one output neuron for each of the leaf nodes in the taxonomy. The second avenue we explore is to use the features extracted by the ImageNet pre-trained InceptionV3 model and train a tree-ensemble classifier. In particular, we use ensembles of predictive clustering trees [6] for hierarchical multi-label classification (PCTs for HMC). We compare our results with earlier work on the task at hand. This includes the use of ensembles of PCTs for HMC on hand-crafted features extracted from the diatom images, as well as features extracted by scale-invariant feature transforms. The transfer learning approach of fine-tuning the ImageNet pre-trained CNN achieves excellent predictive performance.

Keywords: Hierarchical multi-label classification · Diatoms · Transfer learning · Convolutional neural networks · Feature extraction · Predictive clustering trees · Tree ensembles

1 Introduction

Diatoms are specific, large and ecologically important group of algae organisms [1]. The species are found in a water reserve constitute a bio indicator of its quality and whether some kind of activities are more suitable or not [4]. The cell wall can be divided into two halves. Each half of the cell consists of a valve and a number of girdle bands. One half is slightly then the other and overlaps it. In the variety of uses of diatoms, such as water quality monitoring, paleoecology and forensics [1], microscope slides must be first scanned for diatoms: if diatoms are present, they need to be classified. Most classifications are done using classification keys and/or comparing specimens using slides, photographs or drawings of diatoms in books and atlases (Stoermer

and Smol, 2004). This is not a trivial task, taking into consideration that taxonomists estimate that there may be 200,000 different diatom species, half of them still undiscovered, and many of these extremely hard to distinguish on the basis of morphology (du Buf and Bayer, 2002). Furthermore, this is very tedious and repetitive work, thus any degree of automation can greatly help. Therefore, in this paper we propose a method for hierarchical diatom classification with Transfer Learning using Convolutional Neural Networks (CNNs). There are several important properties of the diatoms that can be used to distinguish them: the valve's outline, the contour (symmetry, global and local shape characteristics, length and width of the diatoms), and the ornamentation of the valve face. Some of these characteristics can be noted in Fig.1 and Fig.2. Our approach consists of two parts: feature extraction using transfer learning, and hierarchical image classification. The feature extraction uses the pre-trained InceptionV3 model [5] in order to extract image features. The idea is that the feature extraction phase would extract numerous relevant features so that a hierarchical model can be learned to distinguish the diatoms. The second part of the approach classifies the image into hierarchy of classes, (note that an image can be labeled with more than one label). The classes can be organized into different levels in the hierarchy of taxonomic ranks: genus, species, variety and form. We use predictive clustering trees (PCTs) that can exploit the hierarchical taxonomy and simultaneously predict all taxonomic ranks.

In our experiments, we used a subset of 1100 microscopic images that are classified using the taxonomic rank of diatoms. The diatoms from the images belong to 55 different classes (taxa). For each class there are at least 10 images up to a maximum 29 images. Images in the dataset vary in shape and ornamentation. Images (Fig.1 and Fig.2) below, show two different taxa that belong to different genus.

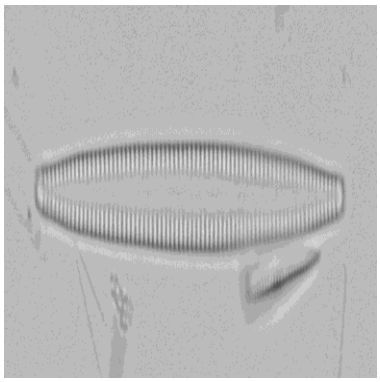

Fig. 1. Tabularia_sp.1__

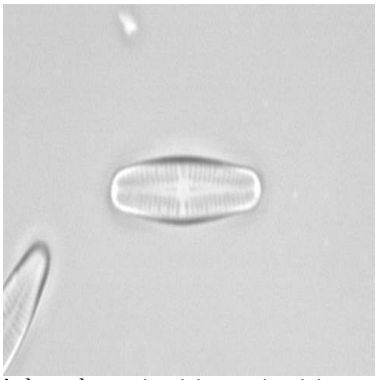

Fig.2. Achnanthes_minutisima_minutisima_

2 Related Work

There are several studies that directly address automatic diatom classification. One of that studies uses random forests of PCTs for HMC, bagging of PCTs for HMC and SVMs of diatom images [1]. In the PCT framework, a tree is viewed as a hierarchy of clusters: the top node corresponds to one cluster containing all data, which is recur-

sively partitioned into smaller clusters while moving down the tree. This system for automatic diatom classification also has two parts: image processing (feature extraction) and image classification [1]. The main difference between this model and our hierarchical classification is image processing part. First model has implemented two feature extraction techniques. The first technique produces descriptors (Fourier descriptors), that contain information concerning the properties of the valve outline. While descriptors from the second technique, called Scale Invariant Feature Transform (SIFT histograms) contain information about the ornamentation of the valve face. But, before extracting features from the images, they performed image segmentation. The problem of image segmentation, i.e., contour extraction, of gray-scale diatom images can be solved mainly by applying four methods: threshold-based, boundary-based, region-based and hybrid methods [1]. In their system for automatic image classification they used marker-controlled watershed segmentation which has already been successfully applied for diatom image classification. For image classification part, they used ensembles of PCTs, in particular bagging and random forest of PCTs. They compared the results and predictive performance of the ensembles of PCTs [6] for HMC on the three variants of the image database (55 taxa, 48 taxa, 38 taxa). Table 1 in Dimitrovski's paper [1] shows the predictive performance of the feature extraction algorithms and their combination evaluated using recognition rate. We noticed that random forest classifier is better than bagging and SVMs over the variants of database and the three types of descriptors. Most recent researches have shown the lack of ability in solving this problem using deep learning. Due to these research results, we decided on proceeding in solving hierarchical classification on diatom images using transfer learning.

3 CNNs for image classification

Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning [3]. Learning can be supervised, semi-supervised or unsupervised. The adjective "deep" in deep learning comes from the use of multiple layers in the network. Early work showed that a linear perceptron cannot be a universal classifier, and then that a network with a no polynomial activation function with one hidden layer of unbounded width can on the other hand so be. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability, whence the "structured" part. Deep learning techniques overcome the problem of feature selection by not requiring pre-selected features but extracting the significant features from raw input automatically for a problem in hand.

3.1 Convolutional Neural Networks (CNNs)

CNNs have deep feed-forward architecture and ability to generalize in a better way as compared to networks with fully connected layers [3]. Figure 3 describes CNN as the concept of hierarchical feature extractors. It learns highly abstract features from the diatom image, and identifies its characteristics efficiently.

The reasons why CNN generalizes better than classical models are as follows. First, the key interest for applying CNN lies in the idea of using concept of weight sharing, due to which the number of parameters that needs training is substantially reduced, resulting in improved generalization. Due to lesser parameters, CNN can be trained smoothly and does not suffer overfitting. Secondly, the classification stage is incorporated with feature extraction stage, both uses learning process. Thirdly, it is much difficult to implement large networks using general models of artificial neural network (ANN) than implementing in CNN. CNNs are widely being used in various domains due to their remarkable performance such as image classification, object detection, face detection, speech recognition, vehicle recognition, diabetic retinopathy, facial expression recognition and many more.

Typical CNN has two parts: convolutional base and classifier - shown in Fig.3. The first part, convolutional base, is composed by stack of convolutional and pooling layers. The main goal of the convolutional base is to generate features from the images. The first layer of each CNN used is ‘input layer’ which takes images, resize them for passing onto further layers for feature extraction. The next few layers of the convolutional base are convolution layers which act as filters for images, hence finding out features from images and also used for calculating the match feature points during testing. The extracted feature sets are then passed to ‘pooling layer’. This layer takes large images and shrink them down while preserving the most important information in them. It keeps the maximum value from each window, it preserves the best fits of each feature within the window. The ReLU (Rectified Linear Unit) layer is a non-linear activation function that swaps every negative number of the pooling layer with 0. This helps the CNN stay mathematically stable by keeping learned values from getting stuck near 0 or blowing up toward infinity. The second part in each CNN is classifier, which is usually composed by fully connected layers. The main goal of the classifier is to take the high-level filtered images and translate them into categories with labels. A fully connected layer is a layer whose neurons have full connections to all activation in the previous layer.

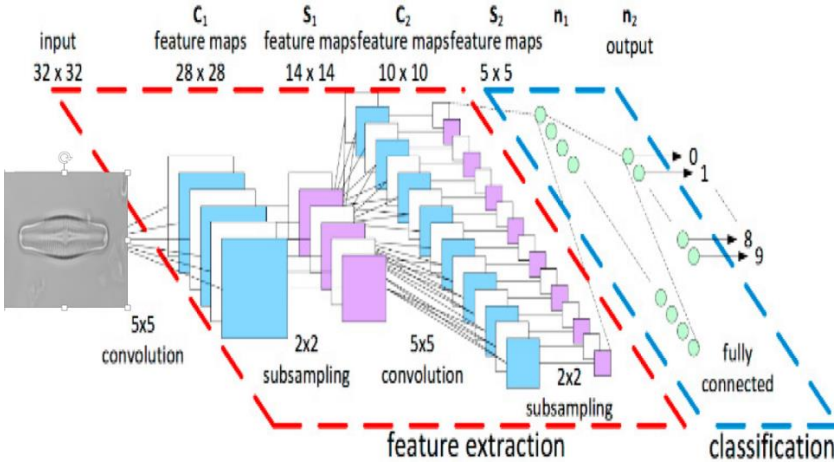

Fig.3. CNN architecture

3.2 Transfer Learning

Transfer learning is a popular method in computer vision because it allows to build accurate models in a timesaving way [3] [5]. With transfer learning, instead of starting the learning process from scratch, you start from patterns that have been learned when solving a different problem. This way you leverage previous learnings and avoid starting from scratch. Transfer learning allows to train deep networks using significantly less data then we would need if we had to train from scratch. With transfer learning, we are in effect transferring the "knowledge" that a model has learned from a previous task, to our current one. The idea is that the two tasks are not totally disjoint, and as such we can leverage whatever network parameters that model has learned through its extensive training, without having to do that training ourselves. Transfer learning has been consistently proven to boost model accuracy and reduce require training time.

In our approach, the goal of transfer learning is to transfer the knowledge learned by InceptionV3 on millions of images by learning to classify thousands of classes. In particular, we use the final output of the convolutional base (the last layer in the red section in Fig.3) as input to another hierarchical classifier.

3.3 Pre-Training

In computer vision, transfer learning is usually expressed through the use of pre-trained models [8]. A pre-trained model is a saved network that was previously trained on a large benchmark dataset, typically on a large-scale image-classification task, to solve a problem similar to the one that we want to solve. Deep Neural Network is trained on that large dataset, usually on ImageNet dataset [5]. ImageNet is an image database organized according to the WordNet hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images. Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature (e.g. VGG, InceptionV3, MobileNet). Several pre-trained models used in transfer learning are based on large convolutional neural networks (CNN). Its

high performance and its easiness in training are two of the main factors driving the popularity of CNN over the last years. The intuition behind transfer learning for image classification is that if a model is trained on a large and general enough dataset, this model will effectively serve as a generic model of the visual world. You can then take advantage of these learned feature maps without having to start from scratch by training a large model on a large dataset. The first step in this approach is to create the base model from the pre-trained convnets. We created the base model from the InceptionV3 model developed at Google. Inception V3 by Google is the 3rd version in a series of Deep Learning Convolutional Architectures. Inception V3 was trained using a dataset of 1,000 classes from the original ImageNet dataset which was trained with over 1 million training images, the Tensorflow version has 1,001 classes which is due to an additional "background' class not used in the original ImageNet. Inception V3 was trained for the ImageNet Large Visual Recognition Challenge where it was a first runner up. We are using this model to extract features from diatoms images. The input shape of the images in InceptionV3 is 299x299x3. It should have exactly 3 inputs channels, and width and height should be no smaller than 75. E.g. we used (150, 150, 3) input shape values for width and height. The output of the last Dense layer is array with 1000 neurons, which is the number of classes that InceptionV3 model was trained. Detailed architecture of InceptionV3 pre- trained model is shown on Fig.4.

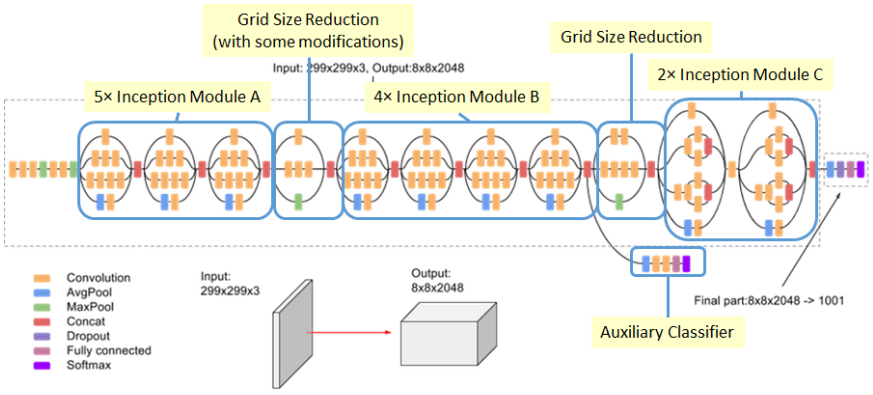

Fig.4. InceptionV3 architecture [12]

3.4 Fine Tuning

We are giving a diatom dataset to fine tune the pre-trained CNN. Consider that the diatom dataset is almost similar to the original dataset used for pre-training. Since the new dataset is similar, the same weights can be used for extracting the features from the new dataset. If the dataset is very small, as the diatoms dataset, it’s better to train only the final layers of the network to avoid overfitting, keeping all other layers fixed. So the aim is to remove the final layers of the pre-trained network. Add new layers and retrain only the new layers.

4 Transfer learning for hierarchical classification of images

From the wide range of pre-trained models that are available, we picked one that is suitable for our problem. The model that we used to solve the problem is InceptionV3 pre-trained model. The goal of the Inception module is to act as a "multi-level feature extractor" by computing *1×1, 3×3,* and *5×5* convolutions within the same module of the network — the output of these filters are then stacked along the channel dimension and before being fed into the next layer in the network. The weights for Inception V3 are smaller than both VGG and ResNet, coming in at 96MB. Number of trainable parameters in this model is 23 817 352. It has a lot of convolutional, pooling and activation layers. The input shape of the diatom images in InceptionV3 that we used is 150x150x3 (shown in Figure 5), and the output of the last Dense layer is array with 3* 3* 2048 extracted features from each image (Figure 5). We simply added a new classifier, which was trained from scratch, on top of the pre-trained model so that it can repurpose the feature maps learned previously for the dataset. We don't have to (re)train the entire model. The base convolutional network (InceptionV3) already contains features that are generically useful for classifying pictures. When we initialized our base model we set include_top=False. This setting is important, as it means that we won't be keeping the Fully-Connected (FC) layers at the end of the model. This is exactly what we want since we're going to train our own brand new fully connected layers for transfer learning. However, the final, classification part of the pre-trained model is specific to the original classification task, and subsequently specific to the set of classes on which the model was trained. The classification part of our model consists of two Dense layers and one Dropout layer, which reduces chances of overfitting. The first Dense layer has 256 neurons and as an input takes the features extracted from convolutional base, and the second Dense layer is using "softmax" activation function and has 55 units, one for each leaf in the hierarchy (taxon). Fine-tuning part of the classification of diatom images consists of unfreezing last 14 layers, which means that all layers up should be frozen. Earlier layers in the convolutional base encode more generic, reusable features, while layers higher up encode more specialized features.

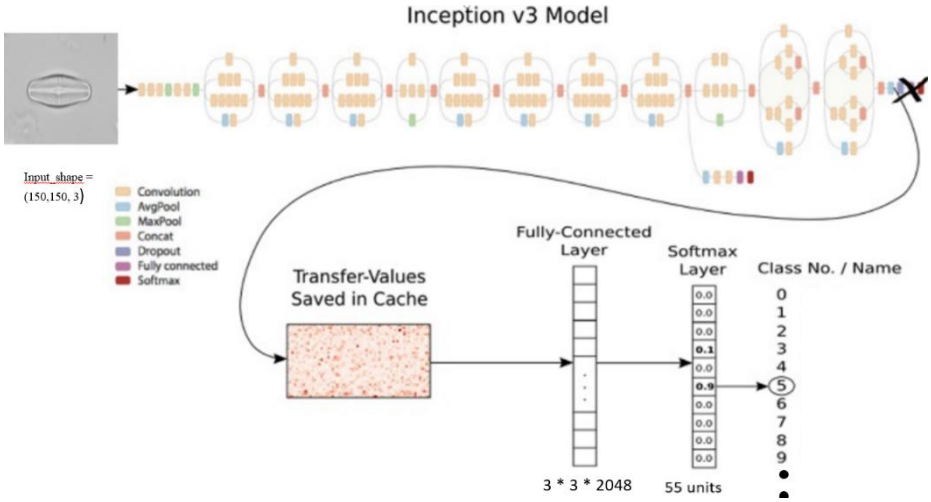

Fig.5. Transfer Learning model for HMC of diatom images [11]

It is more useful to fine-tune the more specialized features, as these are the ones that need to be repurposed on our new problem. There would be fast-decreasing returns in fine-tuning lower layers. Unfreeze a few of the top layers of a frozen model base and jointly train both the newly-added classifier layers and the last layers of the base model. This allows us to "fine-tune" the higher-order feature representations in the base model in order to make them more relevant for the specific task. The final step is to train our model and to evaluate the performances.

5 Ensembles of PCTs

Predictive Clustering Trees (PCTs) generalize decision trees and can be used for a variety of learning tasks including different types of prediction and clustering. A tree is viewed as a hierarchy of clusters [2]: the top-node corresponds to one cluster containing all data, which is recursively partitioned into smaller clusters while moving down the tree. The leaves represent the clusters at the lowest level of the hierarchy and each leaf is labeled with its cluster's prototype (prediction). The features that we extracted using InceptionV3 pre-trained model, combined together with the annotations of the images, are used to train a classifier. An ensemble classifier is a set of (base) classifiers [1]. Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote. We use PCTs for HMC as base classifiers. We consider three ensemble learning techniques that have primarily been used in the context of decision trees: bagging, random forest and support vector machines. Random forests are a combination of tree predictors [7] such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. More precisely, at each node in the decision tree, a random subset of the input attributes is taken, and the best feature is selected from this subset

(instead of the set of all attributes). For training the SVM we used linear kernel and for random forest classifier we used 100 trees in the forest. Bagging classifier is an ensemble meta-estimator [9] that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting) to form a final prediction. The base estimator that fits on random subsets of the dataset is decision tree with 100 base estimators (trees).

6 Experiments and Results

6.1 Estimating performance

Once any image analysis method has been applied, it is important to quantify the performance to know how accurate it is or compare it with other methods. In this section, we present the experimental setup to evaluate the proposal system and compare the results from hierarchical image classification using Transfer Learning and ensembles of PCTs. As a model validation technique, we used k-cross-validation. Cross-validation [10] is a statistical method used to estimate the skill of machine learning models. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (called the validation dataset or testing set). The goal of k-cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem). The procedure has a single parameter called k that refers to the number of groups that a given data is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation (Fig.6). It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split. The general procedure is as follows:

- Shuffle the dataset randomly
- Split the dataset into k folds (groups)
- For each unique group:
    - Take the group as a hold out or test data set
    - Take the remaining groups as a training data set
    - Fit a model on the training set – using 100 epochs and 10 batch_size
    - Evaluate the model on the test set
    - Print precision and recall measures for each fold
    - Retain the evaluation score and discard the model

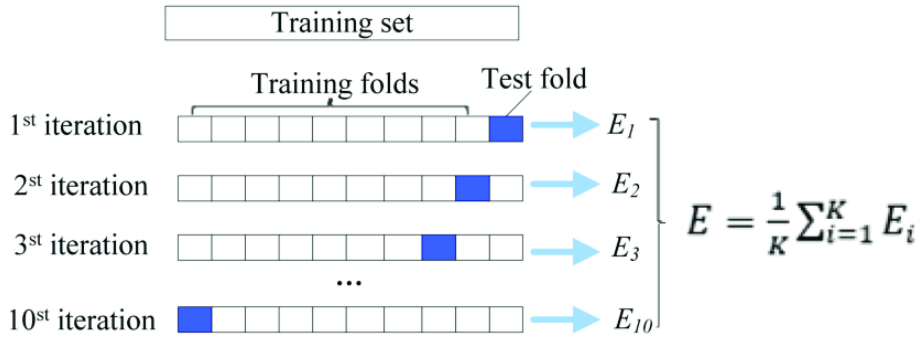

Fig.6. 10-fold Cross-validation

6.2 Performance measures

As a performance measure, we use primarily the overall recognition rate. This measure is also known as classification accuracy. It is calculated as the number of correctly classified images divided by the number of all classified images.

For a more detailed evaluation of performance, we calculate the precision and recall measures for all classes. Precision (1) measures the proportion of diatom images belonging to a given class that are correctly labeled by the classifier. Recall (2), on the other hand, measures the proportion of diatom images labeled by the classifier with a given class that truly belong to that class.

$$\text{Precision} = \frac{\text{TruePositives}}{\text{True Positives} + \text{FalsePositives}} \quad (1)$$

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (2)$$

6.3 Performance of PCT ensembles and SVMs with CNN extracted features

Table 1 summarizes the performance of three machine learning algorithms (bagging and random forests of PCTs for HMC and SVMs). The predictive performance is compared with the results from Dimitrovski's paper [1], where the features used by the machine learning algorithms (bagging, random forest and SVMs) were Fourier descriptors and SIFT histograms. In their work, random forests of PCTs for HMC performed best, followed by bagging of PCTs for HMC and then SVMs.

When we apply SVMs to the features extracted from the InceptionV3 pre-trained model, we get worse results as compared to using Fourier descriptors and SIFT histograms as features. The SVM results are only slightly worse. However, the PCT ensemble results are much worse. To some extent, this is probably due to the very high dimensionality of the feature vectors extracted from the CNN as compared to the

manually extracted features (the difference is two orders of magnitude) and the fact that SVMs are designed to learn in very high dimensional feature spaces.

Table 1. Comparison between the predictive performances of the features extracted from InceptionV3 model and the Fourier descriptors + SIFT histograms (Dimitrovski [1]), evaluated using overall recognition rate

Classifier	Overall recognition rate, InceptionV3	Overall recognition rate, Fourier desc. + SIFT hist.
	55 diatom taxa	55 diatom taxa
Bagging	80.27	95.45
Random Forest	86.36	96.17
SVM	91.09	92.35
Fine-tuned CNN	98.72	N/A

6.4 Performance of the fine-tuned CNN

The overall recognition rate of the fine-tuned CNN, where transfer learning was performed by unfreezing the final layers, is 98.72%. This is better than the best results on the Fourier descriptors+SIFT features. These are the best results reported so far on the dataset at hand.

We now discuss these results in terms of precision and recall values (shown in Table 2) for each class/ diatom species. Results are compared with those of Dimitrovski et al. [1], obtained with the combined feature set (Fourier descriptors + SIFT histograms) and the approach of learning random forest of PCTs for HMC. For most taxa, the CNN results are clearly better.

We further discuss the taxa for which we achieved poor annotation results. The worst precision for the CNN is 0.80 and is obtained for the taxon Navicula/gregaria, where Dimitrovski et al. obtain a precision of 0.88. We believe that this is because of the similarity with other species from the same genus, the small number of images and the fact that the images are not clean and contain other artifacts. On the other hand, the worst precision of Dimitrovski et al. is 0.56 for Nitzschia/hantzschiana, where the CNN has a precision of 1.

Table 2. Precision and recall per taxon obtained with the fine-tuned InceptionV3CNN, compared with those with a combined feature sets (Fourier descriptors + SIFT histograms) and the approach of random forest of PCTs for HMC

		Transfer Learning CNN		Random Forest of PCTs for HMC (Dimitrovski et al. [1])	
		55 diatom taxa			
Taxon	#images	Precision	Recall	Precision	Recall

Achnanthes/minutissima	10	1	0.9	0.83	0.71
Achnanthes/oblongella	12	1	1	0.67	1
Caloneis/amphisbaena	18	1	1	1	1
Cocconeis/placentula	19	1	1	1	1
Cocconeis/neodiminuta	20	1	1	1	1
Cocconeis/stauroneiformis	23	1	1	1	1
Cymbella/helvetica	26	1	1	1	1
Cymbella/hybrida	20	1	1	1	1
Cymbella/subequalis	21	1	1	0.9	1
Denticula/tenuis	22	1	1	1	1
Diatoma/mesodon	26	1	1	1	1
Diatoma/moniliformis	20	1	1	1	1
Encyonema/neogracile	10	1	1	1	1
Encyonema/silesiacum	25	1	1	1	1
Epithemia/sorex	19	0.9	0.9	1	1
Eunotia/bilunaris	12	0.87	0.87	0.8	1
Eunotia/denticulata	22	0.87	0.87	1	1
Eunotia/incisa	20	1	0.96	1	1
Eunotia/tenella	21	1	1	1	0.88
Fallacia/forcipata	26	1	1	1	1
Fallacia/sp.5	17	1	1	1	1
Fragilariforma/bicapitata	20	1	1	1	1
Gomphonema/augur	20	0.96	1	0.91	0.88
Gomphonema/minutum	24	1	1	1	1
Gomphonema/sp.1	20	1	0.96	0.94	0.84
Gyrosigma/acuminatum	20	1	1	1	1
Meridion/circulare	20	1	1	1	1
Navicula__	24	0.9	0.9	N/A	N/A
Navicula/capitata	20	1	1	1	0.86
Navicula/constans	22	1	0.96	1	1
Navicula/gregaria	11	0.8	0.75	0.88	0.78
Navicula/lanceolata	27	0.98	1	1	1
Navicula/menisculus	18	1	1	1	1
Navicula/radiosa	21	1	1	1	1
Navicula/reinhardtii	29	0.96	1	1	1
Navicula/rhynchocephala	19	0.94	1	1	1
Navicula/viridula	19	1	0.97	0.94	1
Nitzschia/dissipata	20	1	0.96	1	0.9
Nitzschia/hantzschiana	20	1	1	0.56	0.83
Nitzschia/sinuata	20	1	0.98	1	0.89
Nitzschia/sp.2	27	0.98	1	0.93	0.79
Opephora/olsenii	20	0.84	1	0.84	0.8

					9
Parlibellus/delognei	20	0.97	1	1	0.95
Petroneis/humerosa	20	1	1	1	1
Pinnularia/kuetzingii	21	1	1	1	1
Pinnularia/silvatica	10	1	1	0.78	1
Pinnularia/subcapitata	15	1	1	N/A	N/A
Sellaphora/bacillum	18	1	0.94	1	1
Stauroneis/smithii	19	1	1	0.86	0.9
Staurosirella/pinnata	17	0.95	1	N/A	N/A
Surirella/Surirella_brebissonii	26	1	1	1	0.9 1
Tabellaria_flocculosa__	20	1	1	1	1
Tabellaria_quadriseptata__	23	1	1	1	1
Tabularia_investiens__	21	0.88	0.88	1	1
Tabularia_sp.1__	20	1	0.94	1	1

7 Conclusion

To summarize, we propose a system for hierarchical classification of diatom images using Transfer Learning from an ImageNet pre-trained InceptionV3 model. We explore two avenues of transfer, one of which is the typical approach of freezing most layers of the network and fine-tuning the final layers. This approach turns out to work very well and achieves the best results so far on the dataset at hand.

The other transfer approach we explore is to use the features extracted from the final convolutional layers of the pre-trained network. These are then used by ensembles of predictive clustering trees for hierarchical multi-label classification. Unfortunately, this approach does not perform very well, probably due to the ineffective use of the large number of features by the tree-based approaches. This hypothesis is supported by the fact that kernel-based approaches using the same features perform much better.

Many avenues of further work remain to be explored. One of these is certainly the use of more recent and larger datasets of diatom images. Another is the use of both hand-crafted and more classical features (such as the Fourier descriptors and SIFT histograms), on one hand, and features extracted from CNNs, on the other hand. Using data augmentations is a further possibility to explore.

CNNs have already been used for multi-label classification. However, our current approach was considering each of the species as a class value in a multi-class classification problem, ignoring the hierarchy and the multi-label aspect. We will also explore the possibility to adapt CNNs for hierarchical multi-label classification.

References

1. Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, Sašo Džeroski: Hierarchical classification of diatom images using ensembles of predictive clustering trees.
2. Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel :Decision Trees for Hierarchical Multi-label Classification
3. Keiron O'Shea, Ryan Nash : An Introduction to Convolutional Neural Networks
4. Anibal Pedraza, Gloria Bueno, Oscar Deniz, Gabriel Cristóbal, Saúl Blanco, María Borrego-Ramos 3Automated Diatom Classification (Part B): A Deep Learning Approach
5. Mahbub Hussain, Jordan J. Bird, and Diego R. Faria: A Study on CNN Transfer Learning for Image Classification
6. Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and, Sašo Džeroski: Image reprsentation, annotation and retrieval with predictive clustering trees
7. LEO BREIMAN: Random Forests. Machine Learning, **45**, 5–32, 2001_c 2001 Kluwer Academic Publishers. Manufactured in The Netherlands.
8. Karl Weiss, Taghi M. Khoshgoftaar and DingDing Wang: A survey of transfer learning
9. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html
10. Sylvain Arlot and Alain Celisse: A survey of cross-validation procedures for model selection
11. https://software.intel.com/content/www/us/en/develop/articles/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloidlymphoblastic.html
12. https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c