A Comparative Study of Image Classification Algorithms for Foraminifera Identification

Boxuan Zhong*, Qian Ge*, Bhargav Kanakiya*, Ritayan Mitra†, Thomas Marchitto† and Edgar Lobaton*

*Department of Electrical and Computer Engineering
North Carolina State University,
Raleigh, North Carolina 27695–7911
Email: {bzhong2, qge2, bkanaki, edgar.lobaton}@ncsu.edu

†Institute of Arctic and Alpine Research
University of Colorado Boulder, Boulder, Colorado 80309–0552
Email: {Ritayan.mitra, thomas.marchitto}@colorado.edu

***Abstract*—Identifying Foraminifera (or forams for short) is essential for oceanographic and geoscience research as well as petroleum exploration. Currently, this is mostly accomplished using trained human pickers, routinely taking weeks or even months to accomplish the task. In this paper, a foram identification pipeline is proposed to automatic identify forams based on computer vision and machine learning techniques. A microscope based image capturing system is used to collect a labelled image data set. Various popular image classification algorithms are adapted to this specific task and evaluated under various conditions. Finally, the potential of a weighted cross-entropy loss function in adjusting the trade-off between precision and recall is tested. The classification algorithms provide competitive results when compared to human experts labeling of the data set.**

I. Introduction

Foraminifera (or forams, for short), a kind of ubiquitous ocean dwelling organisms, are widely used in oceanographic and geoscience research. They are the most widely used fossil organisms for biostratigraphy, age-dating and correlation of sediments, and paleoenvironmental interpretation [1]. While only a few hundred microns in size, foraminifera have become invaluable tools for various academic and industrial purposes. For example, the relative abundance of different species indicates unique paleoenvironmental conditions, while chemical measurements of their calcium carbonate shells are important information to interpret paleoclimatological parameters such as temperature, salinity, ocean chemistry, and global ice volume [2]. On the other hand, for the oil industry, analyzing forams is one of the most important ways to find potential hydrocarbon deposits [3] and forams are routinely used as an indicator of the ages and paleoenvironments of sedimentary strata in oil wells during petroleum exploration [4]. Classifying and collecting forams according to their species are necessary processes for most studies. Unfortunately, for most laboratories, identification of different species have to be done manually by employed personnel. Generally speaking, a typical study require searching through many thousands of similar sized forams, taking weeks or even months to accomplish.

Recently, the computer vision community has experienced a rapid development thanks to deep convolutional neural networks and the availability of large scale images data sets [5], [6]. Besides deep learning based methods, other standard frameworks such as bag-of-features [7], [8] and dictionary learning [9] methods have also gained significant popularity for image classification in the past years. There exist few studies related to computer-aided forams identification [10], [11], [12]. However, these studies mainly focused on either semi-automatic approaches or simply leaving the classification task to experts. Although these methods significantly accelerate the identification of forams, we claim that by taking advantages of the development of image classification algorithms in the past decade, accurate full-automatic classification becomes possible.

In this paper, we propose a pipeline to automatically identify foraminifera samples based on computer vision and pattern recognition techniques. First, images of foraminifera samples through a microscope under different controlled lighting conditions are captured. Second, images taken under different illumination conditions are fused and corresponding visual features are extracted. Finally, the probabilities of which classes the foram samples possibly belong to are predicted.

Considering the huge number of different species of forams (50 planktonic and 10,000 benthic extant), in this study we aim to provide a "proof of concept" by focusing on several taxa of widely used planktonic foraminifera that are most important for geochemical proxy measurements and certain census counts. Accordingly, the following six species are chosen because of their widespread use within the paleoceanography community [13], [14], [15] : Globigerinoides ruber (G.ruber), Globigerinoides sacculifer (G.sacculilfer), Globigerinoides bulloides (G.bulloides), Neogloboquadrina pachyderma (N.pachyderma) and Neogloboquadrina incompta (N.incompta) and Neogloboquadrina dutertrei (N.dutertrei). Figure 1 includes some examples of the 6 species targeted in this study.

The main contributions of the paper are summarized below:

1) A pipeline is proposed to automatically classify different classes of forams includes lighting-controlled image capturing, information fusion, feature extraction and classification;
2) An image data set of 6 species of forams is collected, which can be used for not only computer vision but also

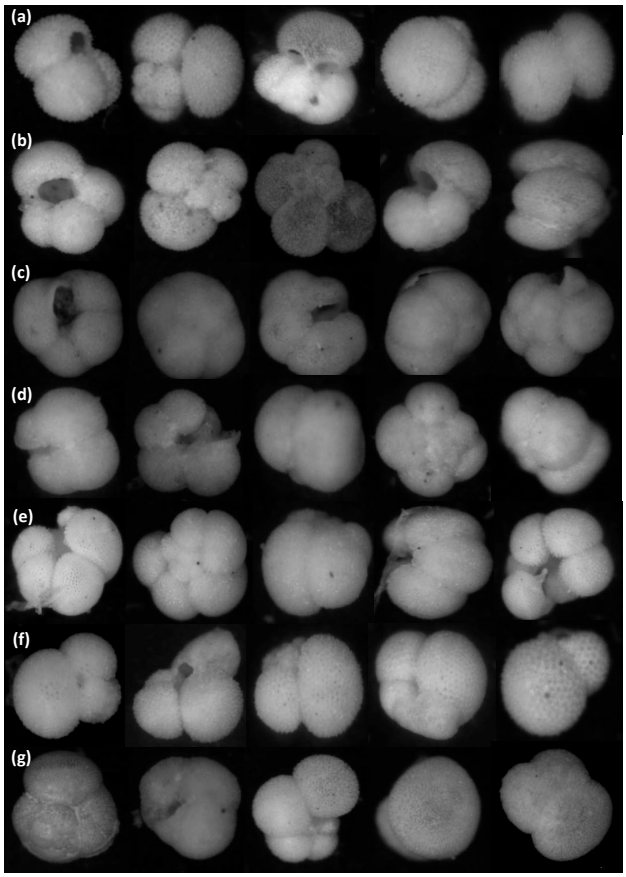

Fig. 1. Species of forams targeted in this paper (a) G. ruber, (b) G. bulloides, (c) N. pachyderma, (d) N. incompta, (e) N. dutertrei, (f) G. sacculilfer, (g) Other species.

oceanographic and geoscience purposes;

3) Different state-of-the-art image classification algorithms are modified and adapted to the forams classification task and their performances are compared in detail;
4) A weighted cross-entropy loss function is proposed to adjust the trade-off between precision and recall according to different applications;
5) The effects of different data sizes and classes distributions on different methods are analyzed in detail.

The rest of the paper is organized as follows: In section II, we show the entire process of data collection; section III, most relevant existing works are briefly discussed; section IV, the processes of adapting different state-of-the-art algorithms to our problem is described; finally, the performances of different algorithms under different conditions are compared and discussed in Section V.

II. Forams Images Data Set

To develop a system which can identify forams using a common microscope and camera, we propose the following pipeline for data collection. The data acquisition system uses an AmScope SE305R-PZ microscope with a 30X zoom) and an AmScope MD500 camera, which is a 5MP USB camera attached to the microscope. Assuming that each sample is found at the center of the field of view, the height of the microscope is adjusted manually to get the samples on focus, providing an approximate resolution of 450×450 pixels per sample (the region of one foram in the image). To capture more information, a ring with 16 LED light is used, which is controlled by an Arduino UNO Microcontroller communicating with MATLAB via Bluetooth. Thus, for each sample, 16 images are captured with 16 different lighting directions.

The current data set has 1437 forams, including 178 G. bulloides, 182 G. ruber, 150 G. sacculifer, 151 N. dutertrei, 174 N. incompta, 152 N. pachyderma and 450 samples of "Others". The dataset is publicly available online [16]. We collected a larger number of samples of other species to better capture the larger variability of this group and mimic the real identification tasks. All these samples were identified and separated manually by experts.

Due to the variations within species and similarity between different species, identifying forams is not an easy task; rather it is one requiring special training and background knowledge. Moreover, in reality, experts can rotate the samples to identify the species while the visual system only has a fixed view of the samples from the microscope. Figure 1 shows some examples of images in our data set. Thus, to better evaluate the performance of classification algorithms, we also collected 2 experts' labelling of 540 randomly chosen samples in our data set.

III. Related Work

Recently, deep Convolutional Neural Networks (CNNs) has triggered a revolutionary change in the image classification community, especially for very large scale data sets. Although CNNs normally require a large amount of training data, various pre-trained models have been published and the different levels of features learned with the original data sets have promising potential to be transferred to new data sets [17], [18], [19]. Among these models, Vgg16 [20], Inception V3 [21] and ResNet50 [22] are most representative and show competitive performance on various computer vision tasks. VggNet, proposed in 2014, emphasizes the importance of deeper network for hierarchical representation of features. Although VggNet has good performance and a simple architecture, its computational cost is also high. Compared to VggNet, Inception V3, on the other hand, is deeper and has better performances but lower computational complexity. Finally, ResNet, the winner of ILSVRC 2015, is even deeper than Inception V3 and has similar performance as the Inception V3 network on ILSVRC data set. In ResNet, residual connections are claimed to be inherently necessary for training "very deep" network [22] and has been shown to accelerate the training process greatly [23]. To guide the training process into the right direction, a feasible loss function is required to evaluate the current model appropriately. Among various loss

functions, cross-entropy is the most popular loss function for classification tasks with softmax layer. The general formula of cross-entropy loss function is

$$H(p,q) = -\sum_{x} w(x)\ p(x)\ log(q(x)) \tag{1}$$

where p, q are respectively the "true" and predicted probability distributions, $w(x)$ is the corresponding weight for sample x with $w = 1$ for all x if it is an uniformly weighted cross-entropy loss function. From the equation we can see that the cross-entropy loss function measures the difference between the "true" and predicted probability distributions, the minimization of which trains the model to approach the "true" distributions [24].

Although deep learning attracts most of the attention in the past few years, traditional classification algorithms, usually involving various manual feature extraction processes, are still showing promising performance on specific or relatively small data sets. Within this group, bag-of-features (BoF) [7] is one of the most popular frameworks, in which an image is represented by a large amount of local features extracted from different points with various scales. PHOW [25], as an extension of BoF, is a dense SIFT descriptor extracted at multiple scales and its implementation is commonly available in various toolboxes such as VLFeat [26]. By dividing the images into pyramids, the PHOW features are good at capturing not only the texture but also spatial features [27]. Recently, a Dirichlet-derived GMM Fisher (DGMMF) kernel was proposed [8] for BoF framework to achieve a compact and dense representation with better discriminative power. Within this branch of image classification algorithms, the pipeline with PHOW and DGMMF kernel is one of the most competitive algorithms on not only standard benchmarks but also our data set. However, for forams in particular, their outer contours contain important shape information which usually get lost when extracting PHOW features. Shape representation is a popular problem in the computer vision community. Particularly, a shape representation called Bag of Contour Fragments is proposed in [28], in which the shape is decomposed into contour fragments and each of them is individually described using a shape descriptor.

IV. Methodology

Two groups of methods are studied in this paper: (1) The BoF pipeline with manually extracted features; (2) Pre-trained CNNs based algorithms. Specifically, Vgg16, ResNet50 and Inception V3 are chosen.

A. Image Fusion

The models chosen in this paper were not originally designed for our task; the BoF method is designed to classify color or grey images while the CNNs were pre-trained with color images. In our case, the color of different species is the same but only the textures and structures are different. Further, pictures using a single lighting condition are usually insufficient to expose the structure of the sample. For example, due to the small size and similar appearance of samples, the edges between chambers will be blurred if the lighting is straight or too bright while the textures can not be seen under shadows or if the light is too dim. To capture more texture and structure information robustly, 16 grey images under different lighting conditions were captured. We found that training a new convolutional layer at the beginning of the pre-trained CNN models would result in serious overfitting problems due to our limited data. Thus, as a compromise, we chose to first fuse the 16 grey images into a 3-dimensional image, used directly as the input for the pre-trained CNNs. Also, for the BoF method, we tested and compared the output of using grayscale and the fused images. The results turned out to be very similar. Thus, to better compare the BoF method with the CNN methods, we chose to use the fused images directly as the input for all the methods involved in this paper.

To fuse the 16 grey images into a 3-dimensional image, we have implemented and tested Principal Components Analysis (PCA) in a similar manner to computing Eigenfaces [29]. However, during the process, important texture information is lost, leading to low classification accuracy. As a result, we chose to take the element-wise max, min, and mean values for the 16 images (matrices) and construct a new image with these three values as the 3 dimensions; however, by taking the max and min values of the pixels, the approach became susceptible to image-specific noise (e.g. specularities), present in only a few of the 16 images. To resolve this, we chose to replace the max, min and mean with 90%, 50% and 10% percentiles. This allowed the algorithm to capture enough variance in the data while reducing its susceptibility to the aforementioned sources of noise. The examples of fused images are shown in Figure 2.

B. Bag-of-Features Framework

As discussed in Section III, the BoF pipeline uses the fused images generated above to extract PHOW features and form a codebook which is then fed into the DGMMF kernel. Finally a linear SVM classifier is used for classification. The code of this pipeline is provided in [8]. We refer to this method as "PHOW" in our analysis.

To extract the shape information of the outer contours of forams, we first calculate the boundary of the sample's silhouette by image binarization of the average of the 16

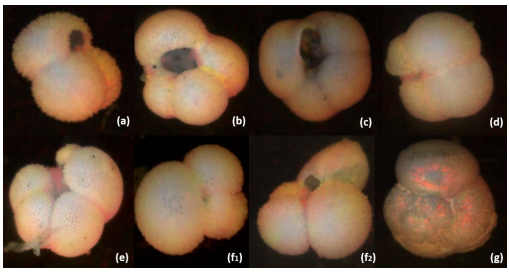

Fig. 2. Fused images of targeted forams (a) G.ruber, (b) G.bulloides, (c) N.pachyderma, (d) N.incompta, (e) N.dutertrei, (f) G.sacculilfer without (f_1) and with (f_2) sac-like final chamber, (g) Other species.

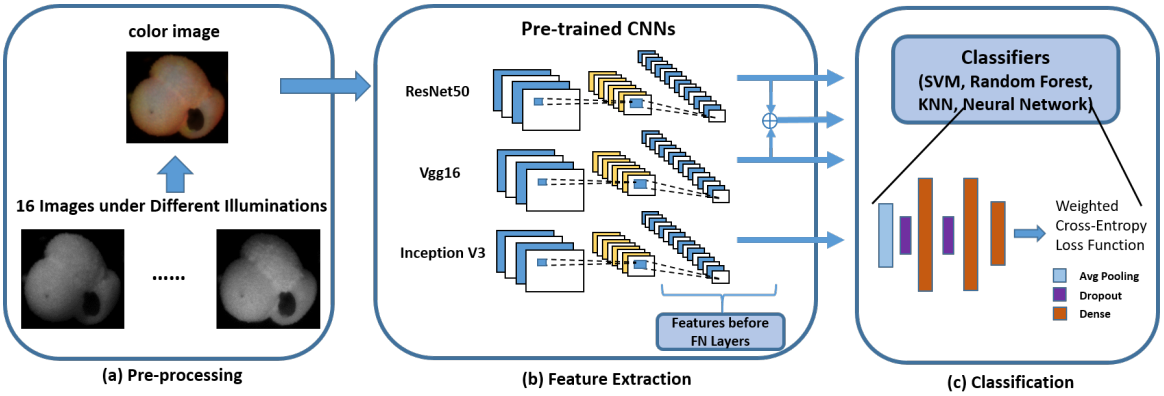

Fig. 3. Pipeline of transfer learning from pre-trained CNNs.

images and then use the Bag of Contour Fragments (BCF) method proposed in [28], with code available online. The descriptor used to represent the contour fragments is the shape context descriptor proposed in [30]. This method represents contour also in a Bag-of-Words (BoW) framework using a spatial pyramid matching (SPM) technique [31]. Finally, local-constrained linear coding (LLC) [32] is used to transform the representation into a space that is suitable to be classified with a linear SVM. We merge the features from "PHOW" (encoded by DGMMF) and the features from "BCF" (encoded by LLC), and then use a linear SVM for classification. We refer to this method as "PHOW+Contour" in our analysis.

C. Transfer Learning for Convolutional Neural Networks

Due to the limited size of the data set, we did not train the deep neural networks from scratch; instead, we performed transfer learning with CNNs pre-trained with ILSVRC, a large scale image classification data set [33]. In our pipeline, we use the output of the feature maps from the pre-trained models. Afterwards, we attach three new fully connected layers with dropout regularization in between for classification on the new data set. The entire pipeline is shown in Figure 3. If we count from the left to right. The first layer is a global average pooling layer. The first two fully connected (FC) layers both have 512 nodes and Relu as the activation function while the last FC layer has 7 (the same number as the classes including "Others") nodes with softmax activation function. The first and the second dropout rates are 0.05 and 0.15, respectively.

Additionally, although the models are pre-trained with the same data set (ILSVRC), different networks learn to classify based on different information, leading to features complementary to each other. Thus, by concatenating the output features of Vgg16 and ResNet50, the combined model achieves better performance.

In actual forams identification tasks, the requirements for precision and recall for each species vary according to different applications. For example, for chemical measuring of certain species, we would like to use samples that contain less unexpected species, which is to say, high precision of the target species is more favorable than high recall but lower precision. This trade-off can be achieved by choosing an appropriate weighting loss function. Thus, we propose a cross-entropy based weighted loss function to enable the application oriented trade-off potential. The loss for one sample can be expressed as

$$\begin{aligned} L(Y,\hat{Y}) &= f(Y,\hat{Y})\sum_i H(y_i,\hat{y}_i) \\ &= -f(Y,\hat{Y})\sum_i y_i log(\hat{y}_i) \end{aligned} \tag{2}$$

where $y_i, \hat{y}_i$ are respectively the "true" and predicted probability of class i. $Y = \{y_i\}$ is the "true" distribution of target sample. In classification tasks, y_i generally is set equal to 1 if the sample belongs to class i, otherwise 0. $\hat{Y} = \{\hat{y}_i\}$ is the output vector of the last softmax layer of the model which has the same length as the classes need to classify (in our case, it equals to 7). The function $f(Y,\hat{Y})$ is defined as

$$f(Y,\hat{Y}) = W_{l(Y),k(\hat{Y})} \tag{3}$$

where $[W_{l,k}]$ is a matrix of weights, $l(Y) = \text{argmax}_{l'}\ y_{l'}$ and $k(\hat{Y}) = \text{argmax}_{k'}\ \hat{y}_{k'}$. The entry $W_{l,k}$ represents the weight associated with classifying the target sample (with true class l) as class k in the loss function. In our case, we will want to maintain high precision for the 6 species mentioned previously. Thus, we set $W_{7,k} > W_{l,k}$ for all $l < 7$ where class 7 is the class of "Others". That is, we put more weight on misclassifying samples in the "Others" class which would lead to a drop on precision for the species that we are aiming to identify. For simplicity, we set $W_{7,k} = w$ and $W_{l,k} = 1$ for all k and $l < 7$. To test how much the weights can influence the performance, we change the value of w between 10^{-4} and 10^4 for our analysis.

V. Data Analysis

Our experiments are conducted to explore the following:

- How various algorithms perform under different amount of training data;
- How the proportion of "Other" in the training set influences performance;

- How the weighted cross-entropy loss function influences the identification results.

To evaluate different algorithms and the experts' labeling, we adopt three metrics: precision, recall and F1 score. In actual forams identification tasks, the correct identification of targeted species is relevant; therefore, we calculate the precision, recall and F1 score of the 6 target species and report the average weighted according to their proportions in the testing or human labeling data set.

A. Human Experts Results

As mentioned in Section II, we evaluate the performance of the classification algorithms by comparing the results to the annotations made by two experts for 540 randomly chosen samples from our dataset. Expert A has more than 5 years of experience in forams identification while expert B has 6 months of experience. The experts had eight choices when labeling. The first seven choices correspond to the classes considered for this study, and the final option was "Not Identifiable." The last label was used if the expert believed that the sample cannot be identified with the provided images.

We compute the weighted average precision, recall and F1 score of the 6 target species under two different scenarios: First, we regard the samples with label "Not Identifiable" as ones that the experts fail to correctly recognize, which will be counted as false negatives. We refer to this scenario as "With NI." For the second scenario, we discard the "Not Identifiable" samples from our computation of precision and recall. We refer to this scenario as "Without NI." The results of the two experts under these two scenarios are shown in Table I.

Finally, as we will see in this section, the classification algorithms show competitive performance both in precision and recall to human experts. The results show that human experts perform better in precision but worse in recall, indicating that human experts are good at preventing mixing erroneous samples as the target species but will miss samples from the target species as a result.

B. Performances Comparison

During the training process, classes are weighted inversely proportional to class frequencies in the training data. This is done to mitigate the effect of classes that have fewer samples than others, and is achieved by adjusting the corresponding parameters of the model fitting function provided by *Keras* [34] and the parameters of the classifiers provided by *scikit-learn* [35]. This weighting procedure is an extra step besides weighted cross entropy loss function defined in equation (2) and remains the same for all the experiments.

TABLE I
PERFORMANCE OF HUMAN EXPERTS

	Scenarios	Expert A	Expert B	Average
Precision	With NI	80.44	81.14	80.79
	Without NI	84.39	80.62	82.51
Recall	With NI	58.98	53.61	56.25
	Without NI	87.60	67.01	77.31
F1 Score	With NI	66.71	59.92	63.31
	Without NI	85.84	69.33	77.59

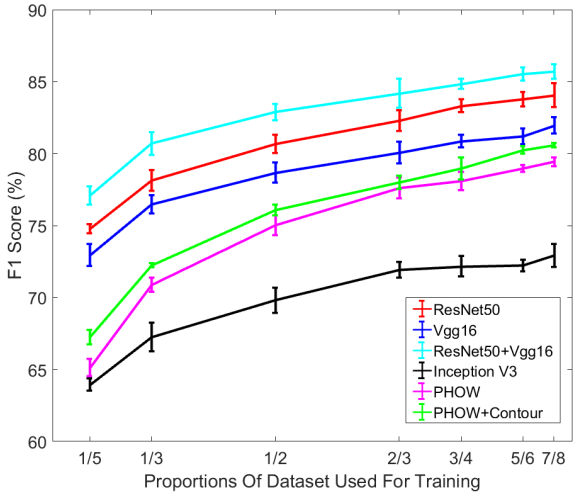

Fig. 4. F1 scores of different algorithms with different sizes of training data.

To appropriately split the training and testing data sets, we adopt a stratified K-Folds cross validation approach, in which the proportions of different classes are preserved in the training and testing set. The data set will be randomly divided into K groups, with one group for testing and the rest for training. The process will be repeated until all the groups have been used for testing once. To evaluate the effect of different sizes of training and testing data. A series of numbers of splitting groups are tested: 8, 6, 5, 4, 3, 2. Furthermore, to test even smaller training data sets, we also split the data into 3, 5 groups and use 1 group for training while the rest for testing. Similarly, the process will be repeated until all the groups have been used for training once. Finally, to get smoother and more valid results, the whole process is repeated 10 times and the mean and standard deviation of the results are reported. For simplicity, we only perform 4 repetitions for the BoF methods, while all the other testing procedure is the same.

1) Comparison of Methods: In this paper, 6 representative algorithms are tested and compared. As shown in Figure 4, the ensemble model of ResNet50 and Vgg16 shows the best performance, followed by ResNet50. The BoF method with PHOW and contour shape features performs worse than Vgg16 but better than the method with PHOW features alone. Finally, Inception V3 has the worst performance compared to the other five methods. Because the relative ranking and trend are similar for different proportions of "Others" during training, only the results using 100% of the "Others" samples is reported. Later we will show results of different methodologies as the proportion of "Others" is changed for training.

For the CNNs, although our forams data set is small and

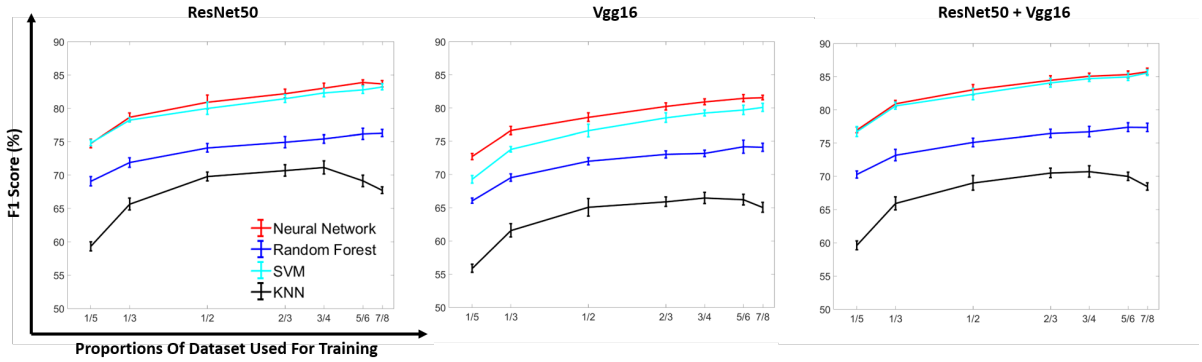

Fig. 5. F1 scores as a function of the proportion of training data for the different classifiers used in the CNN pipeline.

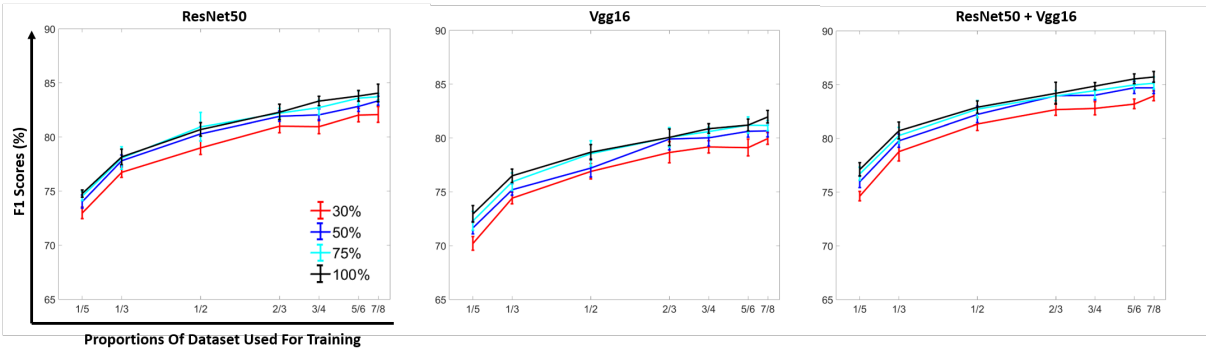

Fig. 6. F1 scores as a function of the proportion of training data. Each curve correspond to a different proportion of "Others" samples (i.e., 30%, 50%, 75% or 100%). A proportion of 100% means that we used all the samples of "Others" available for training.

different from the natural image data set (ILSVRC), the pre-trained models still show promising potential in extracting informative features. We note that their performance in our data set is different from the ranking in the ILSVRC data set. Furthermore, we conjecture that the extracted features from each method are complementary since merging the feature from ResNet50 and Vgg16 leads to better performance.

2) CNN Pipeline Comparison: After extracting features with the pre-trained CNNs, we make use of different classifiers with these features. These classifiers are : Random Forest (RF) where 250 trees are used; SVM where a Radial Basis Function (RBF) kernel is used; kNN where points are weighted by the inverse of their distance; and a Neural Network (NN). The classes are weighted according to their proportions in the testing data as discussed previously and all the other parameters are set to the default values in *scikit-learn* [35]. The NN classifier has the same architecture and parameters as discussed in Section IV.

As shown in Figure 5, NN in average shows the best performance followed by SVM. The gap between NN and SVM is very small but larger for Vgg16. For all the different CNNs, Random Forest (RF) always ranks the third while kNN has the worst performance. The gap between RF and kNN, and RF and SVM are the largest. The results indicate that neural network suits the forams classification task better than the other three classifiers when features extracted by pre-trained CNNs are used. We only report the results of "ResNet50", "Vgg16" and "ResNet50 + Vgg16" with 100% of the "Others" samples used for training. The trends for the other proportions are similar.

In the following sections, we will analyze the performance of the CNNs with the Neural Network classifier at the end.

3) Effect of the Proportion of Training Data: As shown in Figure 4, with the increase of the training data size the overall performance increases as well. BoF related methods seem to be more sensitive to very small training data sets than pre-trained CNN based methods. The reason might be that in the BoF framework, the codebook (for mid-level features) needs to be established by clustering the low-level features extracted from the training data, while the entire features extracting process of pre-trained CNNs does not depend on the training data. Although all the methods favor larger training data size, the BoF framework in particular depends heavier on the data sizes. Additionally, even when only $\frac{1}{5}$ of the data is used for training, the accuracy of CNNs methods is still acceptable, with the gap of scores within 10% between each other.

4) Effect of the Proportion of "Others": The class labelled as "Others" is a mixture of different species excluding the 6 target species. The number of this class is around 3 times the number of each target class in our data set. To study the influence of the proportion of "Other" in the training data, we tested the models by training respectively with 30%, 50%,

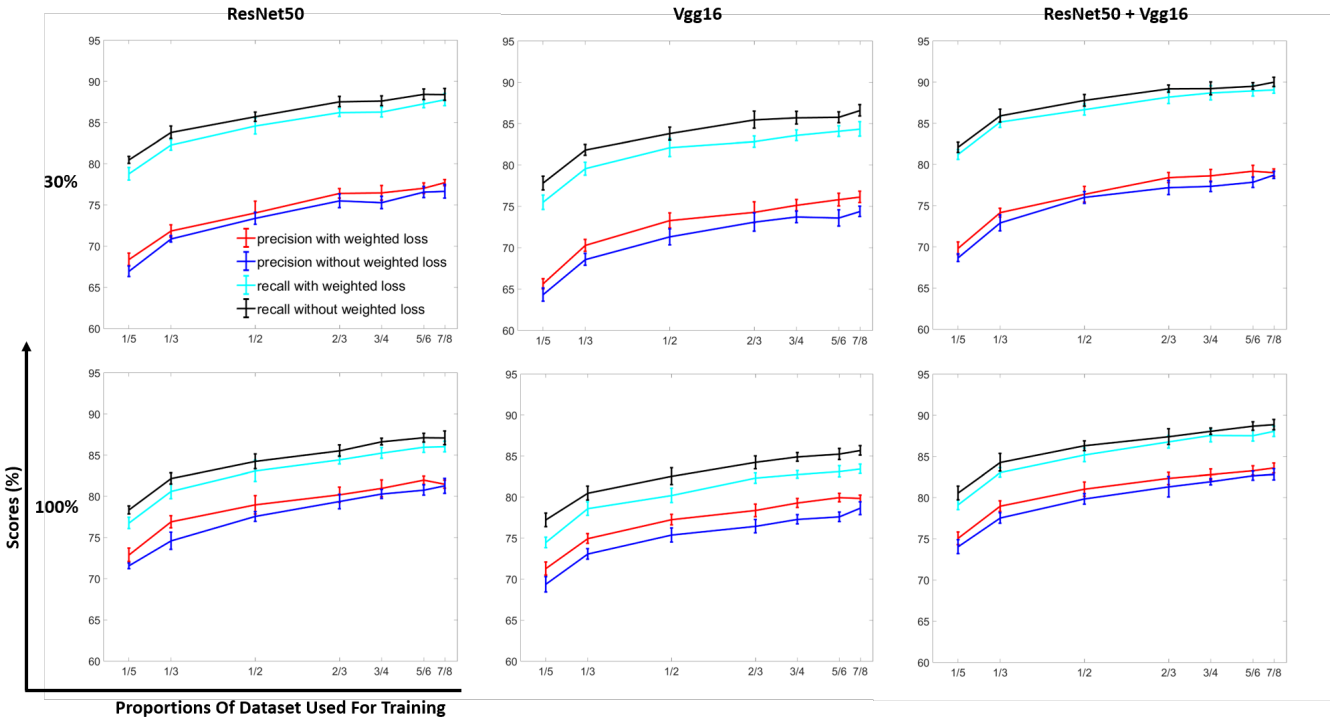

Fig. 7. Precision and Recall curves as a function of the proportion of data used for training. These plots correspond to the results without a weighted loss function, and with a weighted ($w = 4$) loss function. The first row of the data shows the results when only 30% of the "Others" data points are used for training, and the second row shows the results for 100%.

75% and 100% of all the "Others" samples originally chosen for training via the Stratified K-Folds (SKF) cross validation algorithms. The proportion of "Others" remains at 100% for testing (i.e., this proportion is not changed for testing). For all other sections in this paper, the proportion of "Others" remain the same for both training and testing data.

As shown in Figure 6, larger proportions of "Others" in the training data set tend to generate better performance. The performance is very similar for proportions above 50%. These observations indicate that, at least for the forams identification application, the algorithms are not very sensitive to the class distribution (here, it mainly means the proportion of "Others") of the training data. The trends are similar for all methods. Figure 6 shows the results of "ResNet50", "Vgg16" and "ResNet50 + Vgg16."

C. Weighted Cross-Entropy

In Figure 7, the performance between the neural networks without a weighted loss function and with a weighted loss function with $w = 4$ (as defined in equation 2) are compared. The weighted average precision of the 6 target species is higher if a weighted cross-entropy loss function is used. As a trade-off, the recall decreases by a similar amount, resulting in a similar F1 score. The results demonstrate that the weighted cross-entropy loss function can be used to adjust the trade-off between precision and recall. The trends are similar for different proportions of training data, CNN models, and proportions of "Others".

To further test how much trade-off the weighted loss function can achieve, we varied w between 10^{-4} and 10^{4}. Here, "ResNet50+Vgg16" is tested with a five-folds cross validation scheme and 100% of "Others" data points are used. The results are shown in Figure 8. As expected for larger values of weight w, the precision improves but the value seems to saturate around 86%. As w decreases, the recall improves while still saturating around 90%.

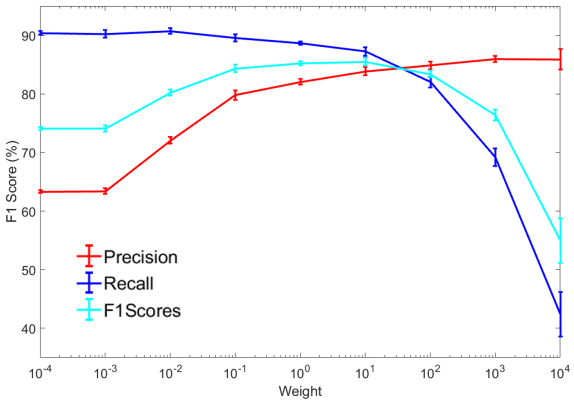

Fig. 8. Precision, Recall and F1 curves as a function of the weight w on the loss function. Results correspond to the "ResNet50+Vgg16" model.

VI. Conclusion

We proposed a pipeline for an automatic foraminifera identification system based on image classification techniques. Several of the most representative and state-of-the-art algorithms are modified to suit our problem and evaluated under different training data sizes and distributions of classes. Finally, a weighted cross-entropy loss function was proposed and its potential in adjusting the trade-off between precision and recall was discussed.

The results indicate that current image classification techniques have promising performances on this task, which is comparable to the performance of human experts. Moreover, the performance is still acceptable when a small amount of data is available for training. However, the current computer vision based identification system still has its own limitations. The biggest limitation is that human experts can rotate the samples and observe the whole sample to get nearly perfect identification results, but for our current system, only pictures of one fixed view are available.

Acknowledgment

We thank Svapnil Ankolkar, Mehdi Garcia Cornell, Bryant Delgado and Sadaf Iqbal for help on image capturing and Jeremy Cole for his help on proofreading of the manuscript. This work is supported by the National Science Foundation under award OCE-1637039.

References

[1] A. R. Loeblich Jr and H. Tappan, *Foraminiferal genera and their classification*. Springer, 2015.

[2] J. Zachos, M. Pagani, L. Sloan, E. Thomas, and K. Billups, "Trends, rhythms, and aberrations in global climate 65 ma to present," *Science*, vol. 292, no. 5517, pp. 686–693, 2001.

[3] A. H. Cheetham, A. J. Rowell, and R. Boardman, *Fossil invertebrates*. JSTOR, 1987.

[4] A. Singh, "Micropaleontology in petroleum exploration," in *7th International Conference and Exposition of Petroleum Geophysics*, 2008, pp. 14–16.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[6] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2285–2294.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[8] T. Kobayashi, "Dirichlet-based histogram feature transform for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3278–3285.

[9] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent k-svd: Learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[10] K. Ranaweera, A. P. Harrison, S. Bains, and D. Joseph, "Feasibility of computer-aided identification of foraminiferal tests," *Marine Micropaleontology*, vol. 72, no. 1, pp. 66–75, 2009.

[11] K. Ranaweera, S. Bains, and D. Joseph, "Analysis of image-based classification of foraminiferal tests," *Marine Micropaleontology*, vol. 72, no. 1, pp. 60–65, 2009.

[12] C. M. Wong and D. Joseph, "Dynamic hierarchical algorithm for accelerated microfossil identification," pp. 940 503–940 503–15, 2015. [Online]. Available: http://dx.doi.org/10.1117/12.2082596

[13] H. J. Spero, K. M. Mielke, E. M. Kalve, D. W. Lea, and D. K. Pak, "Multispecies approach to reconstructing eastern equatorial pacific thermocline hydrography during the past 360 kyr," *Paleoceanography*, vol. 18, no. 1, 2003.

[14] A. K. Gupta, D. M. Anderson, and J. T. Overpeck, "Abrupt changes in the asian southwest monsoon during the holocene and their links to the north atlantic ocean," *Nature*, vol. 421, no. 6921, pp. 354–357, 2003.

[15] H. J. Spero and D. W. Lea, "The cause of carbon isotope minimum events on glacial terminations," *Science*, vol. 296, no. 5567, pp. 522–525, 2002.

[16] [Online]. Available: https://research.ece.ncsu.edu/aros/foram-identification/

[17] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[18] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

[24] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *arXiv preprint arXiv:1702.05659*, 2017.

[25] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[26] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[27] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[28] X. Wang, B. Feng, X. Bai, W. Liu, and L. J. Latecki, "Bag of contour fragments for robust shape classification," *Pattern Recognition*, vol. 47, no. 6, pp. 2116–2125, 2014.

[29] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. IEEE, 1991, pp. 586–591.

[30] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[31] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[32] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3360–3367.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[34] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

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