Convolutional neural networks as an aid to biostratigraphy: A test on Late Paleozoic microfossils

Keywords: convolutional neural networks; deep learning; biostratigraphy; fusulinids.
Rafael Pires de Lima\textsuperscript{1,2}, rlima@ou.edu

Kurt J. Marfurt\textsuperscript{1}, kmarfurt@ou.edu

Roger Burkhalter\textsuperscript{3}, rjb@ou.edu

Murphy Cassel\textsuperscript{4}, murphyc@ou.edu

James E. Barrick\textsuperscript{5}, jim.Barrick@ttu.edu

Gerilyn S. Soreghan\textsuperscript{1}, lsoreg@ou.edu

\textsuperscript{1}School of Geology and Geophysics, The University of Oklahoma, 100 East Boyd Street, RM 710, Norman, Oklahoma, 73019, USA

\textsuperscript{2}The Geological Survey of Brazil – CPRM, 55 Rua Costa, São Paulo, São Paulo, Brazil

\textsuperscript{3}Sam Nobel Museum, The University of Oklahoma, 2401 Chautauqua Ave., Norman, Oklahoma, 73072, USA

\textsuperscript{4}Halliburton, 14206 Maranta Estates court, Cypress, Texas, 77429, USA

\textsuperscript{5}Department of Geosciences, Mail Stop 1053, Texas Tech University, Lubbock, Texas, 79409, USA
ABSTRACT

Accurate taxonomic classification of microfossils in thin-sections is an important biostratigraphic procedure. As paleontological expertise is often restricted to specific taxonomic groups and experts are not present in all institutions, geoscience researchers suffer from lack of access to critical taxonomic knowledge for biostratigraphic analyses. Moreover, diminishing emphasis on education and training in systematics poses a major challenge for the future of biostratigraphy and on associated endeavors reliant on systematics. Here we present a machine-learning approach to classify and organize fusulinids—microscopic index fossils for the Late Paleozoic. The technique we employ has the potential to use and embed this important taxonomic knowledge in mathematical models that can be applied to recognize and categorize fossil specimens. Our results demonstrate that, given adequate images and training, convolutional neural network models can correctly identify fusulinids with high levels of accuracy (> 90%). Deep-learning predictions are highly accurate, but the internal discrimination criteria and attributes remain concealed abstractly in the layers of the convolutional neural network. Regardless of elevated levels of abstraction, continued efforts in digitization of biological and paleontological collections at numerous museums and expansion of machine learning can enable highly accurate and easy-to-use classification and, thus, facilitate biostratigraphic analyses through the Phanerozoic.

PLAIN LANGUAGE SUMMARY

This research focuses on the applications of deep convolutional neural networks (CNN) as a means to facilitate fossil classification. CNN is an artificial neural network architecture commonly used for object recognition tasks. By using test-case thin-sections of fusulinids we suggest that CNN can identify and categorize images of fossilized specimens with a high level of
accuracy. We present what we believe to be one of the first applications of CNN using bisected fossil specimens identified specifically through 2D thin-sections to perform genus and species identification.
INTRODUCTION

Biostratigraphy is a critical approach for dating and correlating sedimentary successions, particularly given the common absence of material appropriate for radioisotopic dating of sedimentary strata. Biostratigraphy hinges on detailed analysis of extracted fossils, or thin-sections of fossils to identify specimens to the genus and species levels. In addition to their utility for dating, fossil assemblages shed light on paleoenvironmental conditions; foraminiferal assemblages, for example, can yield information critical for reconstructing histories of paleoclimatic and paleoceanographic conditions (e.g., Gooday, 1994; Culver, 2003; Kucera, 2007; Roozpeykar and Moghaddam, 2016). Commonly, biostratigraphy relies on species-level identification of well-established microfossil groups such as foraminifera, conodonts, and palynomorphs. The complex morphology of fossil organisms has required the use of specialists for reliable and correct systematics, especially for effecting detailed and accurate biostratigraphic correlation. Unfortunately, education and training in the identification of fossil taxa is diminishing, greatly crippling future capacity in this area (Farley and Armentrout 2000, 2002).

Although the largest investment of resources (both time and financial) for biostratigraphic studies may be for data acquisition and sample preparation, not every institution has the necessary expert to make accurate species-level identification for biostratigraphic analyses. Both old and new paleontological collections with significant biostratigraphic value may be ignored because no one is available to perform the necessary taxonomic identifications. Because of this paucity of experts, a procedure is needed that can help facilitate the access of existing taxonomic knowledge to a broader audience. If a tool with a reliable accuracy can be used to identify different taxonomic groups, a wider range of geoscientists (or scientists in general) can rely on years of accumulated, but difficult-to-access paleontological knowledge.
The ongoing revolution in big data and computational modeling is enabling the possibility of augmenting human capacity in fossil characterization and identification with machine learning techniques. In deep learning, computational models consist of numerous processing layers that learn representations of data with different levels of abstraction (LeCun et al. 2015). Although the number of layers in most models are in the order of tens, sometimes reaching hundreds, some researchers are experimenting models with up to 1200 layers (e.g. Huang et al. 2016). Recent advances in the architecture of deep-learning convolutional neural networks (CNN) have greatly improved the fields of image classification and computer vision. LeCun et al. (2015) provided details on deep learning and show some of the breakthroughs achieved by this technology. Dumoulin and Visin (2016) showed details on convolutions and arithmetic for deep learning procedures. We provide in the Appendix 1 the basics of deep learning and CNN. Our material is built on images and examples and provides the reader with the necessary intuition about the mechanics of deep learning and CNNs without delving into intricate details and computations.

Very deep CNN emerged in 2014 and increased the levels of accuracy in numerous artificial intelligence classification problems (Szegedy et al. 2014). For example, current CNN models can differentiate not only the image of a leopard from that of a mite or a container ship (objects with significantly different characteristics), but can differentiate images of leopards from their biological cousins – jaguars, cheetahs and snow leopards (objects with very similar characteristics; e.g. Krizhevsky et al., 2012). Szegedy et al.’s (2015) CNN architecture reached a 3.5% top-5 error (frequency in which the model cannot predict the correct class as one of the top five most probable guesses) and 17.3% top-1 error in the classification of the ImageNet Large-Scale Visual Recognition Challenge (a benchmark in visual recognition and classification). The
ImageNet comprises hundreds of mixed-object categories and millions of images (Russakovsky et al. 2015).

A currently underutilized application of machine learning is fossil identification - a key component of biostratigraphy. Ranaweera et al. (2009) used a computer-aided approach in which they applied clustering techniques followed by expert labeling for identification of foraminifers. Recently work has been done with the objective to generate a foraminiferal identification pipeline that uses CNN and other machine learning methods and compares such results to classifications performed by human experts (Zhong et al., 2017). Kong et al. (2016) show a novel technique that can be used for fossil pollen identification, by our understanding, the first application of CNN image classification techniques applied to fossil specimens. Kong et al. (2016) select patches of pollen grains in microscopy images and use a pretrained CNN model to extract features for pollen species identification. Pollen researchers have been working on automated identification for a long time and are currently experimenting with CNN models as well (e.g. Sevillano and Aznarte 2018). Given the current proliferation of efforts to digitize biological specimens (both modern and fossil) (Blagoderov et al. 2012; Ellwood et al. 2015), successful application of CNN methods could greatly facilitate research that relies upon fossil identification and biostratigraphy.

We provide, to our knowledge, a novel attempt to conduct automated fossil classification using CNN models, and the first attempt on a fossil group (late Paleozoic fusulinids) identified specifically through 2D thin-section analysis. This methodology does not depend on specialized bench work and can be applied to existing photomicrographs in legacy collections –indirectly capturing the knowledge of the researchers that performed the classification or labeling of these collections. Our test case analyses provide proof-of-concept verification as highly accurate
results were obtained with significantly smaller domain-specific training data when compared to traditional deep CNN applications. Although researchers are working with CNN models to perform image recognition using few examples, sometimes a single example per class is used for training (e. g. Koch 2015; Lake et al. 2015; Santoro et al. 2016), most of the CNN applications use hundreds to hundreds of thousands of examples per class. With the additional imaging of the numerous specimens in the large legacy collections of fusulinids in North America, automated classification can potentially organize a large volume of taxonomic and biostratigraphic information into a reliable and coherent system easily accessible to a variety of users, including both specialists and non-specialists.

SHORT GLOSSARY

Because machine learning, and CNN in particular, may be unfamiliar to many paleontologists, we provide this simple glossary to define some of the technical terms used in the manuscript. More detailed computer science definitions can be found in the list of references as well as online under “Machine Learning Glossary | Google Developers” n.d.

Accuracy – The ratio between the number of correct classifications and the total number of classifications performed. Accuracy can be computed for the training, test or any other dataset.

Class – The name, value, or category assigned to each data sample. In this paper we use Class in the machine learning sense rather than in the biological sense.

Classification – The process of assigning uninterpreted data to a particular class.

CNN – see Convolutional Neural Networks. Some authors also use ConvNets for shortening.
**Convolution** – The mathematical process of estimating a value at a given pixel from itself and those of its neighbors. Common convolutions include smoothing, sharpening, and color filtering.

**Convolutional Neural Network** – A neural network architecture in which at least one layer is a convolutional layer.

**Deep Learning** – An artificial neural network that uses multiple hidden layers.

**Labels** – Names applied to an instance, sample, or example (for image classification, an image) associating it with a given class. In this paper the labels are the names of the target species analyzed.

**Natural Images** – A term commonly used in computer vision literature and without a strict definition. In a broad sense, the resulting color photograph taken with an ordinary camera.

**Pooling** – A nonlinear filter that represents reduces the size of the input data, for example, replacing the value of four adjacent pixels with its maximum or minimum.

**Test Data** – Labeled images not used in training but held aside to test the accuracy of the classification.

**Training** – An iterative process that determines the ideal parameters of a machine learning model.

**Training Data** – The subset of the data used for training.

**Transfer Learning** – A process that uses rules constructed in one machine learning task (typically a large one) using similar data (such as 2D color images) as aid to solve the problem at hand.

METHODS
In the realm of machine learning techniques, the problem at hand is commonly divided into unsupervised and supervised learning. In unsupervised learning, the user provides data to the algorithm and the algorithm tries to identify patterns present in the data. In supervised learning techniques, the user provides data and corresponding labels and the algorithm tries to learn a function or a relationship between the data and labels. Therefore, the task we investigate in this paper is a supervised learning problem, as we provide data (thin-section images) and labels for training, we expect the CNN to provide a relationship between the data and the labels (the expert defined fusulinid genus or species).

The reliability of CNN results is typically associated directly with the amount of labeled data used during training. With more examples provided to the CNN, the rules used by the model are improved, generating higher-accuracy results. The CNN needs examples to recognize the features of each class it tries to differentiate. The work here focuses on assembling fusulinid thin-section data, conducting image pre-processing, and using transfer learning (Pan and Yang 2010) to generate a CNN model to classify fusulinids. Figure 1 shows a flowchart of the methodology we used. Figure 2 shows a visual representation of the transfer learning process.

Accurate identification of a fusulinid relies on attributes observable from an oriented section exposed along the long axis of the (prolate spheroid-shaped) fusulinid, bisecting the center. A transverse section is useful, but the longitudinal section is essential.

Figure 1: Flowchart summarizing the methodologies we used in this paper. We collected original images from different collections, applied simple image processing and data augmentation to generate our fusulinid database. Coming from the right side, the CNN we use as starting point (Szegedy et al.’s. (2015) Inception V-3) was trained with millions of images. We use transfer learning (Figure 2) to create our fusulinid-tailored CNN model.

Figure 2: Transfer learning process (modified from Oquab et al., 2014). A complex CNN is trained on the source task (Feature Learning, top row) with millions of images. The weights
learned by the CNN model are then transferred to the target tasks (Classifier Learning, bottom row). The greater number of images and the computational time used during training of Inception V-3 (and many other CNN models) make it a powerful tool to be used in image classification tasks. On the right (green box for top row, yellow box for bottom row) some of the classes the CNN can choose. On the bottom row, the inception V-3 weights are not updated during fusulinid training (i.e. we maintain the weights learned during ImageNet training) and we train a new layer responsible for fusulinid classification.

). Both sections reduce the complex internal morphology of fusulinids to two-dimensional views that can be easily imaged. Because fusulinid workers have used these oriented sections for years, an extremely large number of specimens oriented in the same manner exist in legacy collections in museums. Thin-section collections, however, commonly consist not only of individual specimens of a well-oriented longitudinal sections, but also thin-sections of fossil-bearing rocks in which cuts through specimens are randomly oriented and thus yield apparently different sizes and shapes. Therefore, the training set contained only those thin-sections with well-oriented longitudinal cuts and initial classification was restricted to similar longitudinal sections.

Our training dataset comprises original fusulinid thin-sections from Waddell (1966) housed at the Sam Noble Museum at the University of Oklahoma (OU) imaged through modern digital photography. The Waddell collection comprises four different Pennsylvanian fusulinid genera: Beedeina (Fusulina), Wedekindellina, Triticites, and Fusulinella. Samples from the American Museum of Natural History (AMNH) acquired through the iDigBio portal, an important initiative in digital access to biological collections, provided three additional Permian genera: Parafusulina, Pseudofusulina, and Schwagerina. Differences in thin-section image
properties (e.g., background color), and number of samples available for each genus (ranging from 3 original images to 25, Table 1 in Appendix 2) increased the difficulties encountered for training.

All images were resized to 299 by 299 pixels; this is the input size needed for the CNN model we used and also in the scale of current state-of-the-art resolution for image recognition CNN applications. We also removed the annotations present in the images acquired through the iDigBio portal.

**Transfer learning and bootstrapping training data with image augmentation**

Transfer learning can be used to address the shortage of sufficient domain-specific training data (Carranza-Rojas et al. 2017). In transfer learning, the learned characteristics of a base model trained on a base dataset are reused in a different task (Yosinski et al. 2014). In this manner, layers previously trained with a substantial volume of labeled data can be used as feature extractors and reused to address different objectives, greatly reducing the necessary training computation time. Therefore, a CNN model trained to identify the images of the ImageNet challenge can be used to classify fusulinid thin-sections with the help of transfer learning (Figure 2). In a study analyzing medical image data, Qayyum et al. (2017) found that using transfer learning achieved results comparable to or better than results from training a CNN model from scratch. Examples of transfer learning include Carranza-Rojas et al. (2017) for herbarium specimens, Esteva et al. (2017) for skin cancer classification, and Gomez Villa et al. (2017) for camera-trap images. In all of these studies involving the application of transfer learning, the authors achieved high levels of accuracy from transfer learning using GoogleNet Inception CNN (Szegedy et al. 2015) or other modern CNN architectures as feature extractors.
Here we also use GoogleNet (Inception V-3) model as a feature extractor to classify fusulinid thin-sections.

Deep neural networks have a cascading pattern in which the output of one processing layer is used as input to the next processing layer of the model. When trained on datasets of natural images, the first layer of the deep neural network learns features that resemble either color blobs or some variation of textures. This behavior is so common in CNN models that the analysis is reevaluated every time the initial layers learn any other image characteristics (other than colors or texture). When using CNN as a tool for supervised image classification, there is a transition from general to specific features learned by the model (Yosinski et al. 2014). This is why CNN with good performance on the ImageNet challenge (e.g. Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2014, 2015) can be successfully retrained for new, field-specific classification problems (e.g. Carranza-Rojas et al., 2017; Esteva et al., 2017; Gomez Villa et al., 2017; Norouzzadeh et al., 2018). As the layers become more specific as deeper they are in the model (i.e. as closer to the output of the CNN than the input), sometimes the user might want to extract only general image features. Kong et al. (2016) studies where the features of an ImageNet pretrained model can be used without modifications to extract features from pollen data.

Even though transfer learning provides a powerful approach to address the problem of an insufficient amount of training data, and has been successfully implemented in different fields, the small number of digitized thin-sections available for this work created challenges in assembling the training set. Recent examples using transfer learning for image classification employed training datasets of $10^5$ images (Esteva et al. 2017; Gomez Villa et al. 2017; Carranza-Rojas et al. 2017). In contrast, we relied upon 102 original images of fusulinid specimens, three
orders of magnitude smaller than other studies using transfer learning, with some genera containing as few as three sample images, after discarding images not suitable for this classification task—such as those with inappropriate orientations.

Owing to this limited dataset for training the CNN, we used a bootstrap process to generate pseudo-samples using the available images. As a means to address thin-section color characteristic tendencies, we used simple color-shifting techniques to generate new samples with the objective to make the CNN classification less sensitive to general image color (Figure 4). This color-changing technique is valuable when using the trained CNN model to identify organisms from a collection not used in training. The population was then augmented by simple data rotation. Each longitudinally aligned original image was rotated through a range of angles ±5° about the horizontal axis, as well as flipped about the horizontal and vertical axes to expand the training data set. These approaches increased the number of images that could be used for transfer learning to from the original 102 to 1850 (Table 1 in Appendix 2).

RESULTS

To test the accuracy of our CNN model, we generated three tests sets, two of them at the genus level and one at the species level. At the genus level, the first (smaller) test dataset consists of 13 thin-section images containing at least one of the seven fusulinid genera. The samples from this set were never used in training (the original sample was removed from the training set before bootstrapping). The second test used in the genus level set contains 58 images; the additional 45 image samples were randomly extracted from the bootstrapped training set. This means that the CNN used some variation of these 45 additional images during training. The third test set is used for a more specific CNN model; we used the *Beedeina (Fusulina)* images to classify thin-sections at the species level. We have images from 11 different *Beedeina* species, the test set is
composed of 16 images, five never used in training, and 11 bootstrapped variations. We only used *Beedeina* classification in the species level because that is the genus with the best quality examples and variations in our dataset.

The results we obtained show the potential of the methodology. At the genus level, the top-1 error for the smaller test set (13 images) is 8% and the top-3 error is 0%. For the larger test set (58 total images), the top-1 error is 2% and the top-3 error is 0%. Analysis of the results shows that the CNN is assigning a higher probability for the incorrect genus for only one of the images on the test set (Figure 5) – the correct genus (*Schwagerina*) is the one with the second-highest probability assigned (*Parafusulina* was assigned with higher probability). Given that *Schwagerina* and *Parafusulina* are two genera with major differences in the shapes of their tests, we believe this thin-section was misclassified due to a combination of insufficient examples as well as substandard specimen integrity of *Schwagerina* in the training set. This result reinforces the necessity of having an adequate number of images for the CNN training set or the necessity to apply different data preparation. Figures 6 and 7 show examples in which the CNN correctly classified the thin-section analyzed. Nine out of the 13 examples in the smaller test set had the correct genus assigned with probabilities exceeding 90%. The classification results of the 13 samples from the smaller dataset are presented in Appendix 3.

The classification at the species level also presented high levels of accuracy. The top-1 error for the species level test set is 0%. Figure 8 shows one example of the *Beedeina* species level classification. The complete results of the 16 thin-sections of the *Beedeina* species test set are presented in Appendix 4.

DISCUSSION
To our knowledge this is the first study conducted using thin-sections, commonly used in fusulinid biostratigraphy, as input for a CNN model that can be used to identify microfossils. In this approach, the user simply provides an adequate image and the CNN model outputs the probability of assignment of the specimen to a fusulinid genus. This study differs from Kong et al. (2016) because we use 2D thin-section images (not 3D stacked), and our process uses the complete image during training and testing (not selected patches). Biological complications aside, we also achieved higher accuracy (>90%) in our results, differentiating more classes (seven genera for the genus case and 11 species for the Beedeina species case) than Kong et al. (2016) (3 species). Even lacking an extensive database of images, the methodology we applied achieved high levels of accuracy-- above 90% top-1 accuracy and 100% top-3 accuracy in the genus level, 100% top-1 accuracy for the Beedeina species. Groves and Reisdorph (2009) used multivariate morphometry to show that the Beedeina species separation is statistically significant; our CNN methodology then is able to achieve high levels of accuracy because these are clearly distinct species.

Unlike a human interpreter who relies upon a defined set of morphological measurements to perform taxonomic classifications, the CNN operates from no knowledge of specific attribute analysis and performs the classification based on image characteristics. This also implies that a CNN model, at least with the current implementations, cannot be used to define a new taxonomic division (e.g., a new species), although it may separate out specimens that do fit into existing species. The set of rules created by the CNN are abstract and do not rely on specific phylogenetic systematics measurements; rather, the rules are akin to a cascading set of filters. These filters perhaps are generating rules that approximate their behavior to the measurements during taxonomy studies. But because the CNN models have many such filters, it is often difficult to
discuss the interpretability of CNN models. CNN interpretability by itself is a topic in research (e.g. Olah et al. 2017, 2018). When analyzing a new image, the CNN model (as implemented in this study) will always generate a set of probabilities that such image belongs to the CNN’s learned classes, never declaring the image is none of the pre-defined classes. Nonetheless, the methodology we implemented in this project can easily be generalized and will improve as new images are digitized and made available to the scientific community. Considering that different taxonomic divisions request different attribute analysis – e.g. during the interpretation of conodonts, specimen surface texture is not as important as caudal point and rostral point (e.g. Hogancamp and Manship 2016) – we envision that CNN techniques will go through more significant modifications as they are applied to other taxonomic groups.

Although our approach is similar to recent studies employing transfer learning in image classification (e.g. Carranza-Rojas et al., 2017; Esteva et al., 2017; Gomez Villa et al., 2017), the work we presented achieves highly reliable fossil classification using a limited domain-specific dataset -3 orders-of-magnitude smaller than used in these referenced studies. Kong et al. (2016) used 880 images to differentiate between three fossilized pollen species, whereas in this study we achieved high levels of accuracy using 41 original Beedeina images to differentiate between 11 species. As more image data are digitized, the technique we use (digitized thin-sections, image processing, input to CNN) can be applied without the need for laboratory-specific tools and knowledge, which represents a significant improvement over previous approaches requiring specialty image acquisition for CNN (e.g., Zhong et al. (2017)).

As digitization of legacy data accelerates, the approach presented here will improve with more detailed image processing. Image segmentation techniques can be used to clip the thin-sections containing significant presence of biotic or abiotic components (noise) besides the
organism being analyzed; this will help both in the CNN training and in sequential sample classification. With more data available, object detection, the computer vision task to detect occurrences of objects of different classes (Szegedy et al. 2013; Zhao et al. 2018; Agarwal et al. 2018), can be applied, increasing the potential of paleo-tailored CNN’s in the identification of varying taxa captured in the same sample. The technique we demonstrated in this paper is very general and can easily be modified to suit the identification of different fossil groups, such as conodonts, ostracods, ammonites, and others, as long as the specimen can be classified with a 2-dimensional representation (thin-section or comparable digital image).

As also suggested by Norouzzadeh et al. (2018), the use of active learning (Settles 2012; Sener and Savarese 2018), a process in which humans label images for which the CNN cannot achieve significant confidence, augments the training set to help retrain the CNN model, thus improving the quality and quantity of labeled images used.

As the CNN models are trained with expert labeled data, such expertise is captured in the mathematical functions encapsulated in the deep neural network. Therefore, a CNN model, trained on different collections and having input from different paleontologist experts, provides a means of sharing collections and interpretations across great distances. A fusulinid expert working in the US can help train a CNN fusulinid classifier and such a model (and an abstract form of the expert knowledge) can be used in Asia with no significant cost; in the meantime, researchers working with Saccorhytus fossils in China (Han et al. 2017) can train another CNN to classify their data. Such ease in the exchange of knowledge can help validate interpretations of data spread around the world.

If we are able to capture and mix different paleontological expertise (training CNNs to identify a wide range of taxa), such models can be helpful to identify specimens that might have
previously been misclassified. The combination of CNN as an easy-to-use but highly accurate tool and the digitization of stored paleontological samples can provide a rapid method to bring collections residing in museum drawers for decades to light (or to the cloud). With easy access to this valuable data, the world community can then apply modern statistics to better analyze spatial and temporal distributions, construct more precise assemblages, or simply to better track evolutionary trends. We should not discount this “discovery” component. Museums commonly have a special exhibition of a fossil or bone that was collected decades ago and was only now identified as a new genus/species.

CONCLUSION

The CNN classification test presented here was created with very few original images, yet correctly identified fusulinid specimens to genus level with a significantly high probability when compared to the other taxonomic classification options. With access to more labeled data, training can be improved, and enable the generation of a model sufficiently robust to overcome complications such as the presence of more than one specimen, geologic noise, etc. The move towards digitization of biological and paleontological collections at numerous museums will provide the big-data enhancement to enable assessment of the CNN methodology for examples of fossils from around the world, and ultimate identification to species level.

The September 2018 fire at the Brazilian Science Museum that destroyed some of the most prized collections in South America, and millions of specimens affecting two centuries of curated information (Greshko 2018; Lopes 2018; Escobar 2018), serves as a cautionary tale. Even though museums are extremely valuable to the scientific community, information stored in a single building is susceptible to tragedies that can greatly harm research in various fields. How many other specimens (small, but perhaps therefore unique) residing in collections in countries
with less access to traditional western research publications have already been lost or are at risk? The development of the work we publish here will enable a curious fossil collector in even a remote region to scan the specimen on a cellphone-based app, and document a potentially unique lifeform, preserving the finding in perpetuity.

Efforts in data digitization are important initiatives to protect scientific knowledge and the approach documented here contributes to such endeavors, and aids the use of biostratigraphic data in the scientific community. Biological variation, differences in specimen size, thin-section color tendencies, different imaging techniques, and other considerations will complicate the picture, but ultimately lead to deeper learning, and significant enhancement for all work that relies upon fossil identification.
ACKNOWLEDGMENTS

We thank the iDigBio initiative for providing access to the community for biodiversity collections data. Rafael acknowledges CNPq (grant 203589/2014-9) for the financial support and CPRM for granting the leave of absence allowing the pursuit of his Ph.D. studies. We thank the anonymous reviewers that helped us improve the paper clarity and quality.
REFERENCES


Hogancamp, N.J. and Manship, L.L. 2016. Comparison of morphometric techniques and the ability to accurately reconstruct the form and distinguish between species of the Palmatolepis winchelli group-Conodonta, Upper Devonian. *MICROPALAEONTOLOGY*

Koch, G.R. 2015. Siamese Neural Networks for One-Shot Image Recognition, p. .


Roozpeykar, A. and Moghaddam, I.M. 2016. Benthic foraminifera as biostratigraphical and


FIGURE CAPTIONS

Figure 1: Flowchart summarizing the methodologies we used in this paper. We collected original images from different collections, applied simple image processing and data augmentation to generate our fusulinid database. Coming from the right side, the CNN we use as starting point (Szegedy et al.’s. (2015) Inception V-3) was trained with millions of images. We use transfer learning (Figure 2) to create our fusulinid-tailored CNN model.

Figure 2: Transfer learning process (modified from Oquab et al., 2014). A complex CNN is trained on the source task (Feature Learning, top row) with millions of images. The weights learned by the CNN model are then transferred to the target tasks (Classifier Learning, bottom row). The greater number of images and the computational time used during training of Inception V-3 (and many other CNN models) make it a powerful tool to be used in image classification tasks. On the right (green box for top row, yellow box for bottom row) some of the classes the CNN can choose. On the bottom row, the inception V-3 weights are not updated during fusulinid training (i.e. we maintain the weights learned during ImageNet training) and we train a new layer responsible for fusulinid classification.

Figure 3: Thin-sections with different orientations from the analyzed collection: a. *Beedeina mutabilis* with a longitudinal cut, and b. *B. mutabilis* with a transverse cut.

Figure 4: Figure showing the effect of different collections and our approach to address color variations. We use simple color modification filters – inverted color and sepia filter – to generate
images with different hues. This process can help the CNN not define the class based only on colors – generally defined on the first-layers for deep CNN architectures.

Figure 5: *Schwagerina gruperaensis* thin-section classification. The CNN incorrectly assigned a higher probability for the genus *Parafusulina* for this image (85%); however, the genus *Schwagerina* is the second highest on the list (15%).

Figure 6: *Beedeina (Fusulina) sp* thin-section classification. Note that the CNN was able to correctly assign the genus for the sample analyzed with a very high probability (99%) even though the thin-section image is composed of other biotic (including a transverse cut of another organism) and abiotic components.

Figure 7: *Triticites newelli* thin-section classification. Note that the CNN was able to correctly assign the genus for the sample analyzed (*Triticites* with 98% probability) even though the thin-section image is composed of other biotic (including a transverse cut of another organism) and abiotic components.

Figure 8: *Beedeina acme* thin-section species classification. CNN correctly assigned the species for the sample analyzed (*acme* with 85% probability).
FIGURES

Figure 1

Figure 2
Figure 3
Figure 4
Figure 5

Figure 6
APPENDIX 1. BASICS OF DEEP CONVOLUTIONAL NEURAL NETWORKS

In this section we provide the reader with an overview of the essentials of image convolution, deep learning, and convolutional neural networks (CNNs). We avoid detailed mathematical explanation but rather build the fundamental intuition necessary for CNNs comprehension and usage through figures and examples. LeCun et al. (2015) provide a comprehensive description of deep learning, with information about the building blocks and techniques common to several artificial intelligence applications. Krizhevsky et al.‘s (2012) work is considered a breakthrough that used CNNs to reduce the error rate for object recognition tasks by almost 50%, leading to the subsequent adoption of deep learning in multiple fields of study.

Images and convolution

Images can be digitally stored in several different ways. Raster images (as opposed to vector graphics) are built upon the characteristics of single points (pixels - short for Picture Element). Each pixel has attributes that comprise the data necessary to represent that part of the picture. An image is a grid or matrix of such pixels.

Although there are different ways for the pixels to be archived, the most common storage is to use different levels of red, green, and blue (RGB) to represent a wide range of colors, such that an image file then can be split into its three building bands or channels (Figure A. 1). Therefore, images are digitally stored as a 3D volume, defined by a height, a width, and a “depth” of the three channels. This matrix structure is amenable to diverse image processing techniques, such as convolutions, that are essential for the image recognition tasks performed by CNNs.

Convolutions are oftentimes considered to be the most important operation in signal processing. Convolutions are a linear operation in which an input is filtered by a function (or
kernel) generating an output; such operations can be generalized to $n$-dimensional spaces. Convolutions are related to cross-variance and are used in different fields with slightly different definitions, all essentially depicting the same operation. Here, to continue developing the essential intuition of how CNNs operate for image recognition tasks, we use 2D convolutions applied to each color image, one at a time. In our images, the kernel slides from left-to-right then top-to-bottom.

Figure A. 2 shows how convolutions are performed using 2D matrices and a 3 by 3 horizontal differencing kernel. The kernel slides through an input and generates an output. The computation performed at each step is a simple convolution, or sum of the values of the input aligned and weighted by the values of the kernel themselves. Special care needs to be taken for image borders. The stride of the computation can be every pixel, every second pixel or some other multiple (Dumoulin and Visin 2016). Note the output in Figure A. 2 does not have the same size as the input. Kernel sizes other than 3 x 3 can be used with the same essential technique.

In Figure A. 3 we show results of image convolutions. The convolution of a three channel image with a three channel kernel results in a three channel output (CNNs layers actually stack or sum the results of such convolutions collapsing them into a single channel output, this would generate a single channel “grayscale” image). In the example we provide, the red kernel is set in a way in which the resulting output will highlight horizontal changes. The green and blue kernels are set as the identity, so there are no changes in the output when compared with the original bands. Several results can be obtained using these convolutions with different size kernels and different weights (the values for each one of the elements in the kernels). In CNN applications, the values for each one of the kernels are randomly initialized and, during the training process,
they are modified iteratively in a way that their outputs will help differentiate the analyzed classes. Figure A. 4 shows familiar examples of processed images that can be obtained with filters based on convolutions, most of which are available when you buy a digital camera.

Figure A. 1: Decomposition of an image in its three RGB channels.

Figure A. 2: Visual representation of a 2D convolution. The input (left) is filtered by the kernel (center) generating the output (right). This operation can be visualized as a sliding window of operation; the kernel slides over the input computing the output as a dot product between the values of the input (seen by the kernel window) with the values of the kernel themselves. In our example, the borders of the input are all zeroes and the kernel slides with stride one; after the first value (22) is computed, the kernel slides one pixel to the right and performs the operation again. Different strides and padding strategies can be used to generate different outputs.
Figure A. 3: Convolution of images. Top panel (a.) represents the convolution of the input with RGB kernels and the resulting image. Values of RGB kernels are showed in b. Red channel in input is convolved with red kernel, green channel with green kernel, blue channel with blue kernel. The resulting image shows the horizontal edges in the red channel, while the green and red components remain unchanged.
Figure A. 4: Original image and result of different image processing techniques. All the processed images are results of various types of convolutions.

**Single neuron and an overview of artificial neural networks**

A single artificial neuron is a linear transformation like those shown in Figure A. 4, followed by a nonlinear transformation (traditionally called the “activation function”). The end goal of artificial neural networks is to convert input data into probabilities of an object (image)
belonging to a set of classes. An artificial neural network then is composed of multiple neurons, each one with (in general) different weights.

The very first steps in artificial neural networks modeling are the same as linear fitting statistics frequently used in many fields. The refinement and improved power of artificial neural networks comes from the addition of the activation functions that modify the linear (convolutional) transformation to a nonlinear transformation. There are several activation functions traditionally used when building artificial neural networks. Functions like the sigmoid, hyperbolic tangent (tanh), rectified linear units (ReLu), and leaky ReLu are very common choices as activation functions (Figure A. 5). In CNNs, ReLu are very frequently used in the hidden layers whereas the sigmoid and softmax are used in the classification step (the last layer).

Figure A. 6 shows a visual representation of the computations performed in neural networks. A single neuron receives data from different inputs and processes such data generating an output. Each neuron has its own set of weights (these are the values that are modified during the training/fitting step) and an activation function that has its behavior independent of the input data. The combination of neurons receiving the same input is commonly called a layer. When more than one layer exists in a neural network, such layers are often called hidden layers. The layers of Figure A. 6b are “fully connected” having connections to all elements in the previous layers.
Figure A. 5: Examples of commonly used activation functions. Note the sum of softmax is 1, so its appearance varies according to the horizontal axis (range and sample rate).

Figure A. 6: Visual representation of neuron (a.) and an artificial neural network (b). The number of neurons and layers in b. are arbitrary (as well as number of inputs). Consider, for example, that each one of the circles in the input represents RGB colors for an image plus a bias term. Next, assume that three RGB images and a bias are sufficient to differentiate objects and assign them
to classes 0 and 1. The neuron then iteratively modifies the values of the weights that when applied to the input and fed to the activation function successfully assigns the input to class 0 or 1.

(Deep) Convolutional neural networks

Given these simple principles of digital images, convolution, and neural networks, we now continue our intuitive analysis of how CNNs work. The prefix “deep” is somewhat freely used when referring to CNNs. Until recently, most neural networks have been built on attributes (such as the attributes of an ideal match used in an online dating service). Such attribute-based neural networks have only one or two hidden layers and are sometimes called “shallow learning” (where the author of the software may assume that hair color, height, age, and education are important factors in finding a good match). In contrast CNNs architecture that have more than one or two hidden layers are classified as deep learning operation. In this section, we use the terms described before to show how an image changes when going through the layers of CNNs.

Figure A. 7 shows the flowchart of the operations described in this section. These layers (convolutions, pooling) are very common in CNNs. In the example we give, the convolution layer weights (the values for each one of the elements in the kernel) were set according to our choice. In a CNN model training, these weights are initialized with random values and are iteratively updated. Interestingly, as observed by Yosinski et al. (2014), the first layers of CNN will have weights that act as edge or color detection kernels – much like the ones we chose in this section. As the kernel values change according to the necessity of the CNN to reduce the error (difference between predicted class and true class), different transformations are applied to the images (such as in Figure A. 4).

Figure A. 8 shows a visual representation of the 3 x 3 convolution layer operations as well as the resulting image for the same fusulinid image we have been using in this section. The
resulting image is then input to a pooling layer. Pooling layers reduce the height and width of images with simple statistics and are commonly used in CNNs. Pooling layers can have different sizes (height, width) and different strides. Figure A. 9 shows an example of max pooling, which captures the maximum value of a subset of the input for a small submatrix. In Figure A. 10 we show the resulting max pooling of the output of the 3 x 3 convolution (output of Figure A. 8) continuing the flowchart depicted in Figure A. 7.

We provided the elements necessary for an overall awareness of how CNNs operate and the results of a very small (and “incomplete”) CNN. We believe this material can be used as a quick reference for some of the common terms used in deep learning techniques. Understanding how an image is transformed through convolutions and poolings can help demystify the results obtained by deep learning and CNNs applications. With the information given in this section, we believe the reader can quickly grasp the differences in architecture of robust CNN models, such as the one retrained for the main text of this paper - GoogleNet Inception V-3 (Szegedy et al. 2015) with 48 layers - as well as imagine how the data are transformed as they go through the models.

Input

![Input Image]

3 x 3 convolution

Max pooling

Output

![Output Image]

*Figure A. 7: A simple flowchart representation of the operations performed with the input image. Note that the 3 x 3 convolution block also accounts for the activation function. In the max pooling process, the maximum value of a subset of the input is stored and the image’s size (height and width) is reduced. In here we represent the resulting image as a composition of RGB channels, during deep learning applications such result is actually stacked as a single channel. Figure A. 9 shows how a max pooling layer works.*
Figure A. 8: Visual representation of the image transformation occurring when an image goes through one convolution neuron. In this example, the image is convolved with the same kernel presented in Figure A. 3. The result of the convolution then goes through the activation function in which every pixel has its values scaled. Note how the resulting image after the activation function is similar to the one in Figure A. 3; this is a simple (but nonlinear) rescaling process. In general the resulting process is a single channel (grayscale) image as the three bands are stacked, we choose to represent the fusulinid in RGB for better visualization.
Figure A. 9: A representation of the “max pooling” operation, where the output is simply the maximum value of a subset of the input. Colors are used to facilitate visualization. Different strides and padding techniques can be used with pooling layers, as well as different statistics (minimum, average, median, or other statistical measures).

Figure A. 10: Max pooling of the image result shown in Error! Reference source not found.. Note that the image is essentially the same, but that the colors are slightly different and that the image size (height and width) has been reduced.
APPENDIX 2. DATA TABLES

Table 1: Number of samples for each genus used in this project and source. Collection named Waddell refers to Waddell (1966) and iDigBio refers to samples acquired through iDigBio data portal on August, 2018. Details of downloaded samples from iDigBio on Table 2.

<table>
<thead>
<tr>
<th>Genus</th>
<th>Collection</th>
<th>Number of original samples</th>
<th>Number of samples used in training (bootstrapped samples)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beedeina</td>
<td>Waddell</td>
<td>41</td>
<td>540</td>
</tr>
<tr>
<td>Fusulinella</td>
<td>Waddell</td>
<td>6</td>
<td>74</td>
</tr>
<tr>
<td>Parafusulina</td>
<td>iDigBio</td>
<td>14</td>
<td>177</td>
</tr>
<tr>
<td>Pseudofusulina</td>
<td>iDigBio</td>
<td>3</td>
<td>60</td>
</tr>
<tr>
<td>Schwagerina</td>
<td>iDigBio</td>
<td>5</td>
<td>117</td>
</tr>
<tr>
<td>Triticites</td>
<td>Waddell + iDigBio</td>
<td>17+2</td>
<td>501</td>
</tr>
<tr>
<td>Wedekindellina</td>
<td>Waddell</td>
<td>14</td>
<td>381</td>
</tr>
</tbody>
</table>

Table 2: Details of samples acquired through iDigBio data portal on August, 2018.

<table>
<thead>
<tr>
<th>Genus</th>
<th>Species</th>
<th>Specimen designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parafusulina</td>
<td>australis</td>
<td>AMNH-FI-28263</td>
</tr>
<tr>
<td>Parafusulina</td>
<td>australis</td>
<td>AMNH-FI-28264</td>
</tr>
<tr>
<td>Parafusulina</td>
<td>biturbinata</td>
<td>AMNH-FI-28276</td>
</tr>
<tr>
<td>Parafusulina</td>
<td>biturbinata</td>
<td>AMNH-FI-28277</td>
</tr>
<tr>
<td>Parafusulina</td>
<td>erratoseptata</td>
<td>AMNH-FI-28267</td>
</tr>
<tr>
<td>Parafusulina</td>
<td>erratoseptata</td>
<td>AMNH-FI-28268</td>
</tr>
<tr>
<td>Parafusulina</td>
<td>erratoseptata</td>
<td>AMNH-FI-28278</td>
</tr>
<tr>
<td>Species</td>
<td>Variety</td>
<td>Catalogue Number</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------</td>
<td>--------------------</td>
</tr>
<tr>
<td><em>Parafusulina</em></td>
<td>guatemalaensis</td>
<td>AMNH-FI-28269</td>
</tr>
<tr>
<td><em>Parafusulina</em></td>
<td>guatemalaensis</td>
<td>AMNH-FI-28270</td>
</tr>
<tr>
<td><em>Parafusulina</em></td>
<td>kummeli</td>
<td>AMNH-FI-27088</td>
</tr>
<tr>
<td><em>Parafusulina</em></td>
<td>peruana</td>
<td>AMNH-FI-27087</td>
</tr>
<tr>
<td><em>Parafusulina</em></td>
<td>subrectangularis</td>
<td>AMNH-FI-28279</td>
</tr>
<tr>
<td><em>Parafusulina</em></td>
<td>subrectangularis</td>
<td>AMNH-FI-28280</td>
</tr>
<tr>
<td><em>Parafusulina</em></td>
<td>subrectangularis</td>
<td>AMNH-FI-28281</td>
</tr>
<tr>
<td><em>Pseudofusulina</em></td>
<td>andina</td>
<td>AMNH-FI-27084</td>
</tr>
<tr>
<td><em>Pseudofusulina</em></td>
<td>meloformata</td>
<td>AMNH-FI-27085</td>
</tr>
<tr>
<td><em>Pseudofusulina</em></td>
<td>rasuvilcensis</td>
<td>AMNH-FI-27083</td>
</tr>
<tr>
<td><em>Schwagerina</em></td>
<td>bowmani</td>
<td>AMNH-FI-27076</td>
</tr>
<tr>
<td><em>Schwagerina</em></td>
<td>gruperaensis</td>
<td>AMNH-FI-28249</td>
</tr>
<tr>
<td><em>Schwagerina</em></td>
<td>gruperaensis</td>
<td>AMNH-FI-28250</td>
</tr>
<tr>
<td><em>Schwagerina</em></td>
<td>oxia</td>
<td>AMNH-FI-27074</td>
</tr>
<tr>
<td><em>Schwagerina</em></td>
<td>tintensis</td>
<td>AMNH-FI-27075</td>
</tr>
<tr>
<td><em>Triticites</em></td>
<td>pajerensis</td>
<td>AMNH-FI-28250</td>
</tr>
<tr>
<td><em>Triticites</em></td>
<td>pillahuarensis</td>
<td>AMNH-FI-27061</td>
</tr>
</tbody>
</table>
APPENDIX 3. GENUS CLASSIFICATION - SMALL TEST SET RESULTS

This section shows the CNN classification results for the 13 thin-section images (Table 3) as well as the images used (Figures A. 11 to 23). All images are represented with 299 by 299 pixels. Specimens designated AMNH-FI-XXXXX have real size of approximately 2 by 2 cm, and specimens designated OU-9XXX have real size of approximately 5 by 5 cm. The CNN model does not require extremely precise scales; however, CNN benefits from consistency on image size and proportion within classes.
### Table 3: CNN classification results.

<table>
<thead>
<tr>
<th>Identified Genus →</th>
<th>Beedeina</th>
<th>Beedeina</th>
<th>Beedeina</th>
<th>Beedeina</th>
<th>Beedeina</th>
<th>Fusulinella</th>
<th>Parafusulina</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Figure</strong></td>
<td>A. 11</td>
<td>A. 12</td>
<td>A. 13</td>
<td>A. 14</td>
<td>A. 15</td>
<td>A. 16</td>
<td>A. 17</td>
</tr>
<tr>
<td><strong>Specimen designation</strong></td>
<td>OU-9290</td>
<td>OU-9291</td>
<td>OU-9307</td>
<td>OU-9320</td>
<td>OU-9367</td>
<td>OU-9275</td>
<td>AMNH-FI-28268</td>
</tr>
<tr>
<td><strong>Beedeina</strong></td>
<td>0.94775</td>
<td>0.54324</td>
<td>0.94200</td>
<td>0.81431</td>
<td>0.99346</td>
<td>0.00032</td>
<td>0.00000</td>
</tr>
<tr>
<td><strong>Fusulinella</strong></td>
<td>0.00004</td>
<td>0.00083</td>
<td>0.01456</td>
<td>0.00045</td>
<td>0.00002</td>
<td><strong>0.93890</strong></td>
<td>0.00000</td>
</tr>
<tr>
<td><strong>Parafusulina</strong></td>
<td>0.00001</td>
<td>0.00006</td>
<td>0.00001</td>
<td>0.00085</td>
<td>0.00001</td>
<td>0.00010</td>
<td><strong>0.99516</strong></td>
</tr>
<tr>
<td><strong>Pseudofusulina</strong></td>
<td>0.00000</td>
<td>0.00004</td>
<td>0.00018</td>
<td>0.00019</td>
<td>0.00011</td>
<td>0.00008</td>
<td>0.00001</td>
</tr>
<tr>
<td><strong>Schwagerina</strong></td>
<td>0.00000</td>
<td>0.00004</td>
<td>0.00128</td>
<td>0.00195</td>
<td>0.00001</td>
<td>0.00003</td>
<td>0.00477</td>
</tr>
<tr>
<td><strong>Triticites</strong></td>
<td>0.00035</td>
<td>0.00092</td>
<td>0.04180</td>
<td>0.16692</td>
<td>0.00384</td>
<td>0.03759</td>
<td>0.00002</td>
</tr>
<tr>
<td><strong>Wedekindellina</strong></td>
<td>0.05184</td>
<td>0.45487</td>
<td>0.00018</td>
<td>0.01533</td>
<td>0.00255</td>
<td>0.02298</td>
<td>0.00002</td>
</tr>
<tr>
<td>Identified Genus →</td>
<td>Parafusulina</td>
<td>Pseudofusulina</td>
<td>Schwagerina</td>
<td>Triticites</td>
<td>Triticites</td>
<td>Wedekindellina</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>----------------</td>
<td>-------------</td>
<td>------------</td>
<td>------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td><strong>Figure</strong></td>
<td>A. 18</td>
<td>A. 19</td>
<td>A. 20</td>
<td>A. 21</td>
<td>A. 22</td>
<td>A. 23</td>
<td></td>
</tr>
<tr>
<td><strong>Specimen designation</strong></td>
<td>AMNH-FI-28280</td>
<td>AMNH-FI-27084</td>
<td>AMNH-FI-28249</td>
<td>OU-9354</td>
<td>OU-9359</td>
<td>OU-9289</td>
<td></td>
</tr>
<tr>
<td>Beedeina</td>
<td>0.00002</td>
<td>0.26930</td>
<td>0.00002</td>
<td>0.01483</td>
<td>0.00908</td>
<td>0.00131</td>
<td></td>
</tr>
<tr>
<td>Fusulinella</td>
<td>0.00000</td>
<td>0.00103</td>
<td>0.00000</td>
<td>0.00005</td>
<td>0.00721</td>
<td>0.00033</td>
<td></td>
</tr>
<tr>
<td>Parafusulina</td>
<td><strong>0.97851</strong></td>
<td>0.00861</td>
<td>0.84910</td>
<td>0.00020</td>
<td>0.00018</td>
<td>0.00005</td>
<td></td>
</tr>
<tr>
<td>Pseudofusulina</td>
<td>0.00083</td>
<td><strong>0.47971</strong></td>
<td>0.00029</td>
<td>0.00009</td>
<td>0.00016</td>
<td>0.00001</td>
<td></td>
</tr>
<tr>
<td>Schwagerina</td>
<td>0.02062</td>
<td>0.04249</td>
<td><strong>0.15016</strong></td>
<td>0.00106</td>
<td>0.00117</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>Triticites</td>
<td>0.00001</td>
<td>0.14670</td>
<td>0.00001</td>
<td><strong>0.98366</strong></td>
<td><strong>0.97206</strong></td>
<td>0.00023</td>
<td></td>
</tr>
<tr>
<td>Wedekindellina</td>
<td>0.00001</td>
<td>0.05216</td>
<td>0.00043</td>
<td>0.00012</td>
<td>0.01014</td>
<td><strong>0.99807</strong></td>
<td></td>
</tr>
</tbody>
</table>
Figure A. 11: Beedeina pulmila, specimen designation OU-9290.

Figure A. 12: Beedeina pulmila, specimen designation OU-9291.

Figure A. 13: Beedeina euryteines, specimen designation OU-9307.

Figure A. 14: Beedeina erugata, specimen designation OU-9320.
Figure A. 15: Beedeina sp., specimen designation OU-9367.

Figure A. 16: Fusulinella dakotensis, specimen designation OU-9275.

Figure A. 17: Parafusulina erratoseptata, specimen designation AMNH-FI-28268.

Figure A. 18: Parafusulina subrectangularis, specimen designation AMNH-FI-28280.
Figure A. 19: *Pseudofusulina andina*, specimen designation AMNH-FI-27084.

Figure A. 20: *Schwagerina gruperaensis*, specimen designation AMNH-FI-28249.

Figure A. 21: *Triticites newelli*, specimen designation OU-9354.

Figure A. 22: *Triticites primarius*, specimen designation OU-9359.
Figure A. 23: Wedekindellina sp. A, specimen designation

OU-9289.
APPENDIX 4. SPECIES CLASSIFICATION - BEEDEINA TEST SET RESULTS

This section shows the CNN classification results for the 16 thin-section images (Tables 4 and 5). The test set is composed of five samples that were never used during training and 11 thin-section images that were extracted from the bootstrapped training set. The thin-sections from this test set are presented in Figures A. 24 to A.39. The top-1 error for this test was 0% (indicating the CNN model correctly classified all the thin-section images analyzed).
Table 4: CNN classification results: Beedeina species.

<table>
<thead>
<tr>
<th>Identified Species</th>
<th>acme</th>
<th>acme</th>
<th>affFwhitakeri</th>
<th>cfFnovamexicana</th>
<th>erugata</th>
<th>euryteines</th>
<th>euryteines</th>
<th>haworthi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variation used in training?</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Specimen designation</td>
<td>OU-9336</td>
<td>OU-9339</td>
<td>OU-9332</td>
<td>OU-9301</td>
<td>OU-9324</td>
<td>OU-9306</td>
<td>OU-9307</td>
<td>OU-9314</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identified Species</th>
<th>acme</th>
<th>acme</th>
<th>affFwhitakeri</th>
<th>cfFnovamexicana</th>
<th>erugata</th>
<th>euryteines</th>
<th>euryteines</th>
<th>haworthi</th>
</tr>
</thead>
<tbody>
<tr>
<td>acme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>affFwhitakeri</td>
<td>0.00432</td>
<td>0.13620</td>
<td>0.95368</td>
<td>0.00161</td>
<td>0.00046</td>
<td>0.00126</td>
<td>0.00084</td>
<td>0.00234</td>
</tr>
<tr>
<td>cfFnovamexicana</td>
<td>0.00025</td>
<td>0.00016</td>
<td>0.01708</td>
<td>0.99474</td>
<td>0.00028</td>
<td>0.00304</td>
<td>0.00161</td>
<td>0.00084</td>
</tr>
<tr>
<td>erugata</td>
<td>0.00102</td>
<td>0.00102</td>
<td>0.02491</td>
<td>0.00063</td>
<td>0.99662</td>
<td>0.00003</td>
<td>0.00004</td>
<td>0.00016</td>
</tr>
<tr>
<td>euryteines</td>
<td>0.00225</td>
<td>0.00445</td>
<td>0.00012</td>
<td>0.00044</td>
<td>0.00009</td>
<td>0.96508</td>
<td>0.85859</td>
<td>0.42230</td>
</tr>
<tr>
<td>haworthi</td>
<td>0.01110</td>
<td>0.00610</td>
<td>0.00002</td>
<td>0.00016</td>
<td>0.00004</td>
<td>0.02240</td>
<td>0.13092</td>
<td>0.56743</td>
</tr>
<tr>
<td>insolita</td>
<td>0.00001</td>
<td>0.00008</td>
<td>0.00000</td>
<td>0.00006</td>
<td>0.00001</td>
<td>0.00002</td>
<td>0.00003</td>
<td>0.00048</td>
</tr>
<tr>
<td>mutabilis</td>
<td>0.00039</td>
<td>0.00214</td>
<td>0.00274</td>
<td>0.00025</td>
<td>0.00189</td>
<td>0.00183</td>
<td>0.00197</td>
<td>0.00135</td>
</tr>
<tr>
<td>plattensis</td>
<td>0.00008</td>
<td>0.00001</td>
<td>0.00000</td>
<td>0.00081</td>
<td>0.00001</td>
<td>0.00032</td>
<td>0.00087</td>
<td>0.00048</td>
</tr>
<tr>
<td>pulmila</td>
<td>0.00062</td>
<td>0.00005</td>
<td>0.00002</td>
<td>0.00053</td>
<td>0.00011</td>
<td>0.00112</td>
<td>0.00252</td>
<td>0.00116</td>
</tr>
<tr>
<td>sp</td>
<td>0.00059</td>
<td>0.00183</td>
<td>0.00109</td>
<td>0.00069</td>
<td>0.00005</td>
<td>0.00028</td>
<td>0.00014</td>
<td>0.00135</td>
</tr>
<tr>
<td>Identified Species</td>
<td>haworthi</td>
<td>insolita</td>
<td>mutabilis</td>
<td>plattensis</td>
<td>plattensis</td>
<td>pulmila</td>
<td>sp</td>
<td>sp</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
<td>----------</td>
<td>-----------</td>
<td>------------</td>
<td>------------</td>
<td>--------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Figure</td>
<td>A. 32</td>
<td>A. 33</td>
<td>A. 34</td>
<td>A. 35</td>
<td>A. 36</td>
<td>A. 37</td>
<td>A. 38</td>
<td>A. 39</td>
</tr>
<tr>
<td>Variation used in training?</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Specimen designation</td>
<td>OU-9315</td>
<td>OU-9279</td>
<td>OU-9286</td>
<td>OU-9295</td>
<td>OU-9297</td>
<td>OU-9292</td>
<td>OU-9365</td>
<td>OU-9367</td>
</tr>
<tr>
<td>acme</td>
<td>0.00088</td>
<td>0.00041</td>
<td>0.00074</td>
<td>0.00241</td>
<td>0.00003</td>
<td>0.00019</td>
<td>0.00190</td>
<td>0.00040</td>
</tr>
<tr>
<td>affwhitakeri</td>
<td>0.00060</td>
<td>0.00000</td>
<td>0.00004</td>
<td>0.00116</td>
<td>0.00008</td>
<td>0.00004</td>
<td>0.00008</td>
<td>0.02025</td>
</tr>
<tr>
<td>cffnovamexicana</td>
<td>0.00101</td>
<td>0.00021</td>
<td>0.00062</td>
<td>0.04012</td>
<td>0.00235</td>
<td>0.00279</td>
<td>0.00054</td>
<td>0.00062</td>
</tr>
<tr>
<td>erugata</td>
<td>0.00028</td>
<td>0.00001</td>
<td>0.00009</td>
<td>0.00134</td>
<td>0.00004</td>
<td>0.00006</td>
<td>0.00004</td>
<td>0.00122</td>
</tr>
<tr>
<td>euryteines</td>
<td>0.00178</td>
<td>0.00036</td>
<td>0.00153</td>
<td>0.00061</td>
<td>0.00001</td>
<td>0.00938</td>
<td>0.00024</td>
<td>0.00048</td>
</tr>
<tr>
<td>haworthi</td>
<td>0.99497</td>
<td>0.00004</td>
<td>0.00060</td>
<td>0.00271</td>
<td>0.00053</td>
<td>0.00008</td>
<td>0.00001</td>
<td>0.01149</td>
</tr>
<tr>
<td>insolita</td>
<td>0.00002</td>
<td>0.99089</td>
<td>0.00710</td>
<td>0.10624</td>
<td>0.00011</td>
<td>0.00007</td>
<td>0.22150</td>
<td>0.00033</td>
</tr>
<tr>
<td>mutabilis</td>
<td>0.00010</td>
<td>0.00256</td>
<td>0.96911</td>
<td>0.00578</td>
<td>0.00114</td>
<td>0.00131</td>
<td>0.00743</td>
<td>0.00154</td>
</tr>
<tr>
<td>plattensis</td>
<td>0.00007</td>
<td>0.00141</td>
<td>0.00218</td>
<td>0.75072</td>
<td>0.99359</td>
<td>0.00138</td>
<td>0.00076</td>
<td>0.00106</td>
</tr>
<tr>
<td>pulmila</td>
<td>0.00003</td>
<td>0.00041</td>
<td>0.01770</td>
<td>0.08637</td>
<td>0.00211</td>
<td>0.98415</td>
<td>0.00155</td>
<td>0.00046</td>
</tr>
<tr>
<td>sp</td>
<td>0.00027</td>
<td>0.00370</td>
<td>0.00029</td>
<td>0.00255</td>
<td>0.00001</td>
<td>0.00056</td>
<td>0.76596</td>
<td>0.96217</td>
</tr>
</tbody>
</table>
Figure A. 24: Beedeina acme, specimen designation OU-9336.

Figure A. 25: Beedeina acme, specimen designation OU-9339.

Figure A. 26: Beedeina aff whitakeri, specimen designation OU-9332.

Figure A. 27: Beedeina cf novamexicana, specimen designation OU-9301.
Figure A. 28: Beedeina erugata, specimen designation OU-9324.

Figure A. 29: Beedeina euryteines, specimen designation OU-9306.

Figure A. 30: Beedeina euryteines, specimen designation OU-9307.

Figure A. 31: Beedeina haworthi, specimen designation OU-9314.
**Figure A. 32**: Beedeina haworthi, specimen designation OU-9315.

**Figure A. 33**: Beedeina insolita, specimen designation OU-9279.

**Figure A. 34**: Beedeina mutabilis, specimen designation OU-9286.

**Figure A. 35**: Beedeina plattensis, specimen designation OU-9295.
Figure A. 36: *Beedeina plattensis*, specimen designation OU-9297.

Figure A. 37: *Beedeina pulmila*, specimen designation OU-9292.

Figure A. 38: *Beedeina sp*, specimen designation OU-9365.

Figure A. 39: *Beedeina sp*, specimen designation OU-9367.