The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Sequential Recognition of Pollen Grain Z-Stacks by Combining CNN and RNN

Amar Daood, Eraldo Ribeiro, Mark Bush
Florida Institute of Technology
Melbourne, U.S.A.

Abstract

Pollen recognition has a wide range of industrial and scientific applications. It guides the energy industry to potential oil and gas deposits, it is proxy data for climate-change scientists, and it increases agricultural production. However, pollen recognition is time consuming because it is usually done by visual inspection. Current automated solutions rely on predesigned measurements of texture and contours, which require tuning for optimal features of a dataset. Also, most methods classify pollen using single-focus images, which require pollen grains to be captured at specific focal planes. We take a difference approach. Instead of using single-focus images, we use stacks of multifocal images (i.e., z-stack) to account for both visual characteristics and 3-D information. We automatically learn from the data the best visual characteristics for classifying pollen using deep-learning methods. Here, we train convolutional and recurrent neural networks (CNN and RNN) to learn the optimal features and recognize a pollen grain as a sequence of multifocal images acquired by an optical microscope. Additionally, we transfer the knowledge pre-trained network to ours to improve its classification and convergence speed. We evaluated our method using 392 stack sequences of 10 types of pollen grains with 10 images for each sequence. Our method achieved a remarkable classification rate of 100%.

1 Introduction

Palynology or the study of pollen grains yields essential data for scientific and industrial applications. For instance, by analyzing fossil pollen found in soil extracted from the bottom of ancient lakes, ecologists map thousands of years of past climate (Treloar, Taylor, and Flenley 2004). Archeologists analyze pollen to find clues about plants, ground cover, and climate (Holt and Bennett 2014; Hodgson et al. 2005). Allergy control scientists analyze allergen levels of pollen collected from aerial traps (Vega et al. 2012; Boucher et al. 2002). Interestingly, pollen grains also help in oil discovery (Hopping 1967).

Most applications identify pollen by visual inspection, a lengthy task that can take days to complete. Pollen recognition can be done in just a few hours by an automated system. Regardless of being done manually or automatically, recognizing pollen grains requires placing the grain sample on a slide under a microscope. When settled on the slide, the grains' position may hide distinctive characteristics from view. Even for spherical grains, for which settling position is not a problem, palynologists must adjust the scope's focal plane to view specific features (e.g., surface texture, pours, spikes). Thus, palynologists examine grains at various focal planes to see all distinguishing visual characteristics.

To automate the pollen recognition, we can use sequences of multifocal images, which are called *z-stacks*. While z-stacks have been used for pollen recognition (Chica 2012; Punyasena et al. 2012; Lagerstrom et al. 2013; Riley et al. 2015), they are still under-exploited. Specifically, previous methods concatenated z-stack features without considering the images as an actual sequence. The sequence of focal planes convey implicit 3-D information that can be useful for recognition. The use of sequence information in z-stacks of pollen grains has been recently used for classification (Riley et al. 2015; Daood, Ribeiro, and Bush 2016a). However, these methods still rely on pre-designed features that encode texture and contours, which might not be distinctive.

In this paper, we propose a method to recognize pollen species using multifocal image sequences. Instead of using concatenated pre-designed features, our method uses deep learning to both learn optimal features and classify the pollen types from multifocal z-stacks as a sequential data. We begin by training a convolutional neural network (CNN) to find descriptive visual characteristics of pollen types. Then, we combine the CNN with a Recurrent Neural Network (RNN) to recognize the pollen type as a sequence of multifocal images. CNN extracts discriminative features and RNN classifies sequential data. We trained our networks on a dataset of 10 pollen types. The dataset has 392 z-stacks with 10 focal planes each. Figure 1 shows one z-stack sample for each pollen type in our dataset.

2 Our Method

We combine two different deep-learning networks to perform the recognition of pollen grains from optical-microscope images. Inspired by (Daood, Ribeiro, and Bush 2016b), we use a CNN to learn optimal visual characteristics of pollen grains. Then, our method uses a RNN to classify the multi-focal z-stack of the pollen as sequential data. RNN is a deep-learning network that classifies sequential data (Graves and Schmidhuber 2009). Its memory units store

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

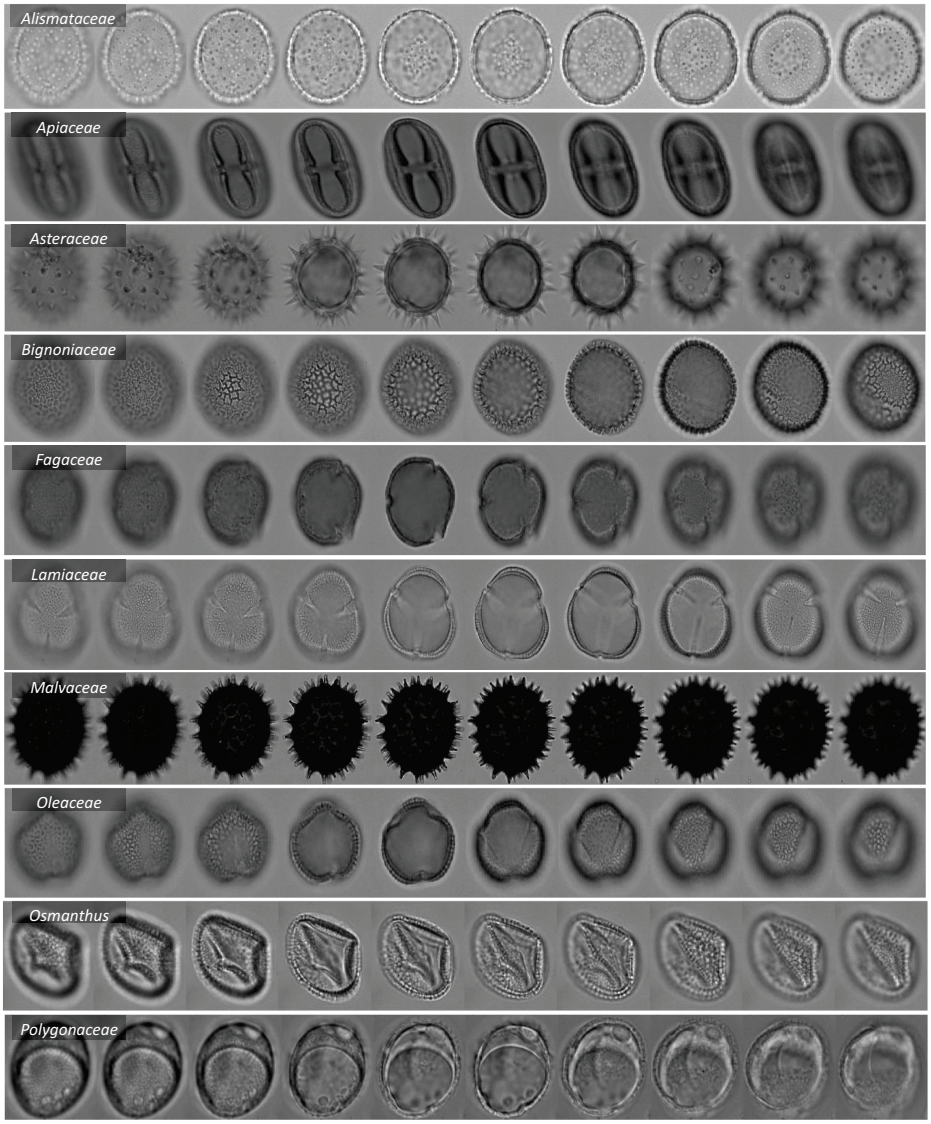

Figure 1: Samples of multifocal z-stack of pollen grains in our dataset.

the history of previous inputs and allow for cyclical connections (Graves and others 2012). These connections map the input sequences to any network output, making RNN ideal for sequence learning. RNN has shown promising results on speech recognition and natural-language processing. It has solved challenging problems such as generating sequences and synthesizing handwriting (Graves 2013).

Our method is divided into two parts. First, we train a customized CNN to extract the features. Then, we improve the recognition accuracy by tuning a pre-trained model VGG16 (Simonyan and Zisserman 2014), which is an approach called transfer learning. Figure 2 shows the architecture of our networks. To train and tune any CNN, we need to re-structure our dataset. Here, we transform each multifocal sequence from 3D space to 2D space. We partition our data into 75% of training and 25% for testing. Our training dataset has 294 stack sequences and the testing dataset has 98 stack sequences. After re-structuring, the training dataset has 2,940 images and the testing dataset has 980 images.

Training a customized CNN

Our CNN has nine learned layers (Figure 2.a). The first two layers are convolutional ones that are followed by two blocks of max pooling and convolutional layers. The convolutional layers share configurations, where each layer includes a filters unit, a rectified unit (ReLUs), and a local normalization unit. The last two layers are a fully connected layer and a soft-max layer of 10 types.

Network configuration (i.e., network depth and filters' size) affects training speed. Increasing the CNN's depth and filters size increases recognition rate but it also increase CPU and memory consumption. In our design, network's configu-

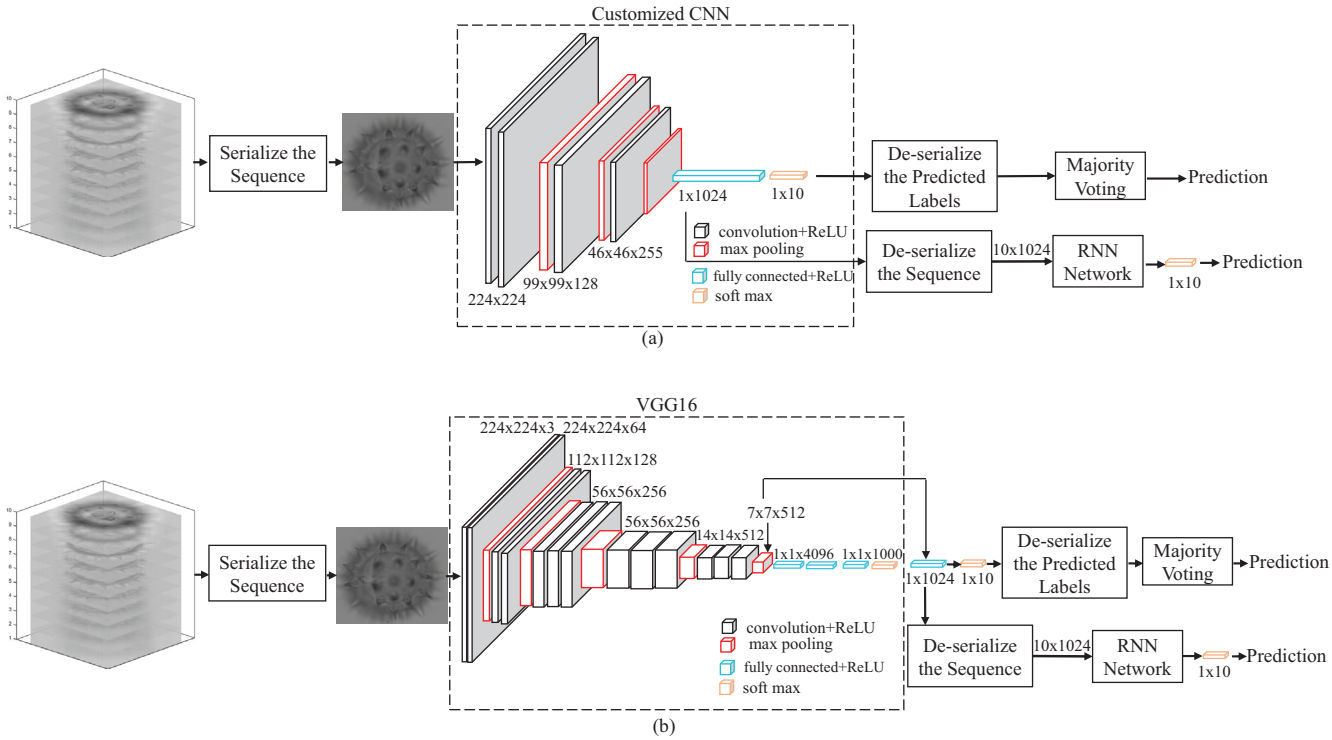

Figure 2: Network architectures. (a) Training a customized CNN. (b) Tuning transferred learning from VGG16.

rations were determined experimentally by maximizing the classification rate. Specific configurations are image resolution, network depth (i.e., number of layers), filters' size for each individual layer, and the training window size (i.e., number of images used in the training process of each step to update network's parameters). For parameter initialization and learning rate, we followed (Krizhevsky, Sutskever, and Hinton 2012). The input of the first layer is 224×224 (i.e., the input image) with 32 filters of size 17×17. The second layer takes the output of the first layer and convolves it with 64 filters of size 11×11. The number of filters and their size of the rest of layers are: 128, 9×9, 250, 5×5, 1024, 1×1, 10, 1×1. Networks were trained using stochastic gradient descent with window size of 32 images.

Our network has 108,644,625 parameters, which is rather large in comparison to our small-size training dataset. To limit over fitting, we artificially augmented the data technique to increase our training data from 2,940 to 14,700 images by applying 5 different rotations to each image. Moreover, we added two drop-out layers to the network by a 0.5 factor. Dropping out some network units during training helps prevent excessive parameter updating. We initialized the weights in each layer using a zero-mean Gaussian distribution. Biases were initialized with constant values of 1, and the learning rate was set to 0.001. We trained our network for 30 epochs using 14,700 samples of pollen grains on a single machine with core 7 cpu and 24GB of memory.

We classify multifocal sequences of pollen grains using a CNN followed by a recurrent neural network (RNN). First, we de-serialized the predicted labels from CNN, and we applied majority voting to estimate the final prediction. However, taking the majority voting of 10 labels ignores sequential nature of multifocal images. Thus, we *serialized* the extracted features from the CNN to create sequences of features (i.e., 10×1024) describing characteristics of the multifocal images. Then, the sequence of features was then used to train a recurrent neural network (RNN). The trained RNN has of 512 units of long-short-term memory (LSTM) followed by a soft-max layer of 10 types. By using RNN to classify the pollen z-stack as sequences of features, we improved the classification accuracy.

Transfer learning

We improved the classification accuracy by adopting *transfer learning* to leverage the learned knowledge from pre-trained models. Tuning pre-trained models has shown promising results when compared to using random features (Yosinski et al. 2014). Additionally, transfer learning improves convergence time during training.

Available deep-learning models that were pre-trained on large-scale data include VGG16, VGG19 (Simonyan and Zisserman 2014), Xception (Chollet 2016), ResNet (He et al. 2015), and Inception (Szegedy et al. 2016). We use VGG16 (Figure 2.b) because of its small size. VGG16 has five blocks. The first block has two convolutional layers. The remaining blocks have a max-pooling layer followed by convolutional layers. Finally, the top part of VGG16 has two fully connected layers and a soft-max layer of 1,000

types. We removed the final part of VGG16 and connected to it a fully connected layer of 1×1024 followed by a softmax layer of 10 types. We re-trained the new architecture to fine tune the network parameters using our training data of 14,700 images. Then, we performed the identification process as a similar way as described in Section 2.

3 Results

Using RNN with transfer learning by tuning VGG16 achieved a remarkable recognition rate of 100%. Figures 3 and 4 show the accuracy and the loss of our models during the training of the CNN and the tuning of VGG16, respectively. We computed the accuracy and the loss at each epoch, and used them to track convergence. The convergence during tuning of VGG16 is much faster than training the CNN from the scratch. For example, the accuracy shown in Figure 4 became nearly 98% in the first two epochs. Figure 5 shows the learned filters of the CNN's first layer.

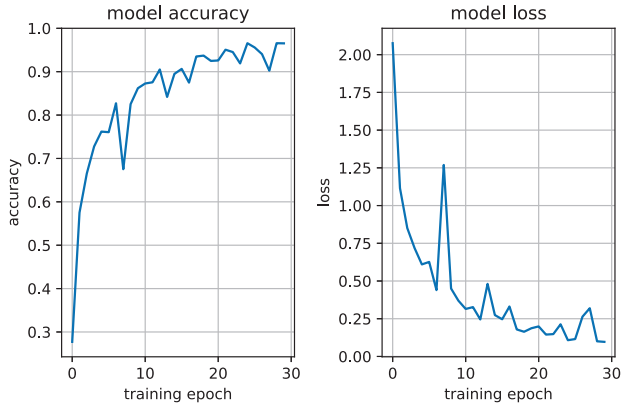

Figure 3: Accuracy and the loss of the training. At each iteration, feed forward is used to compute the accuracy of the network, and the loss of the training.

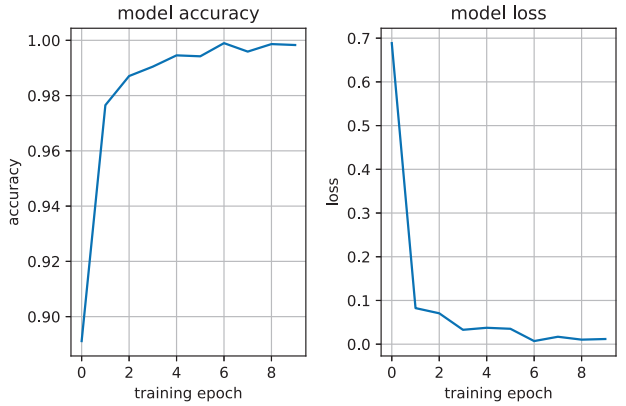

Figure 4: Accuracy and the loss of tuning process. At each iteration, feed forward is used to compute the accuracy of the network, and the loss of the training.

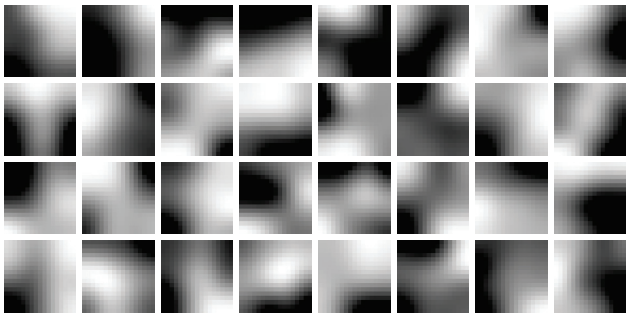

Figure 5: Learned filter($17\times17\times32$) of the first layer of the customized CNNs. Primitive features such as corners, edges, and blobs were learned.

Table 1: Classification rates

Method	Classification Rate
Histogram features, Gray level statistics	81.92%
Geometrical features, fractal dimension	80.12%
Gray level co-occurrence matrix	73.44%
Moments invariants	70.35%
Gabor features	76.04%
HOG	75.63%
LBP	84.73%
Chica's Method	86.18%
Lagerstrom's Method	83.96%
Histogram, gray-level statistics, fractal dimension, LBP	88.88%
Customized CNN+Majority voting (Ours)	**91.83%**
Customized CNN+RNN (Ours)	**95.91%**
Transfer learning+Majority voting (Ours)	**97.95%**
Transfer learning+RNN (Ours)	**100.00%**

To compare our networks' performance with other approaches that use pre-designed features, we extracted some of the most commonly used features to perform the recognition process. These approaches are based on pre-processing the pollen grain images (i.e., enhancement and segmentation), feature extraction, and classification. We used the following features: histogram features (i.e., the mean and variance of histogram), gray-level statistics (i.e., the mean, variance, and entropy), geometrical features (i.e., area, perimeter, compactness, roundness, and aspect ratio), fractal dimension, gray-level co-occurrence matrix (GLCM), Hu's invariant moments, Gabor features, histograms of oriented gradient (HOG), and local binary pattern histogram (LBP). After we performed features extraction, we trained a support vector machine classifier. We also reproduced the results of two works in the literature that used concatenated features from multifocal planes, i.e., Chica's Method (Chica 2012) and Lagerstrom's Method (Lagerstrom et al. 2013). Chica extracted shape and texture features from three focal images of a pollen grain. Lagerstrom extracted histogram statistics, moments, grey-level co-occurrence matrix, and Gabor features from nine focal planes. The results of these comparisons are shown in Table 1.

We compared our method with the best method in Table 1, which achieved a 88.88% classification rate. This method combines histogram, gray-level statistics, fractal dimension, and LBP. The P-value was 1.78×10^{-7}, which rejected the null hypothesis. We computed the average of precision, re-

call, specificity, and F-score (Sokolova and Lapalme 2009).

Table 2: Evaluation Measurements

Method	Precision	Recall	Specificity	F-score
Features combination	89.77%	89.07%	98.75%	88.98%
Customized CNN+Majority voting	**92.07%**	**91.69%**	**99.09%**	**91.58%**
Customized CNN+RNN	**96.42%**	**95.83%**	**99.54%**	**95.64%**
Transfer learning+Majority voting	**98.18%**	**97.14%**	**99.78%**	**97.33%**
Transfer learning+RNN	**100.00%**	**100.00%**	**100.00%**	**100.00%**

4 Conclusion and Future Work

In this paper, we presented a method to identify pollen grains using sequences of multi-focal images. Our method combines two deep-learning networks: a convolutional neural network (CNN) and a recurrent neural network (RNN). The CNN learned discriminating visual characteristics such as corners, blobs, and edges. Then, the learned features were aggregated to create a sequence of features to describe the stack of multi-focal images. We used these extracted features to train a RNN network to classify the pollen as a sequence. We used data augmentation and drop-out layers to reduce the effect of over fitting during training.

Additionally, we used the pre-trained model VGG16 to leverage learned features to improve the classification rates (i.e., transfer learning). By adopting transfer learning, we achieved a 100% classification rate. We compared our results with previous techniques that use pre-designed features. These techniques were largely outperformed by our method. Even though our approach achieves a 100% classification rate, the slow training time of CNNs is an issue when using standard PCs. Faster training speed can be achieved using parallel processing and GPU architectures. As future work, we plan on using high-performance computing for network training and also on increasing the number of pollen types in our dataset.

References

Boucher, A.; Hidalgo, P.; Thonnnt, M.; Belmonte, J.; Galan, C.; Bonton, P.; and Tomczak, R. 2002. Development of a semi automatic system for pollen recognition. *Aerobiologia* 18(3-4):195–201.

Chica, M. 2012. Authentication of bee pollen grains in bright-field microscopy by combining one-class classification techniques and image processing. *Microscopy research and technique* 75(11):1475–1485.

Chollet, F. 2016. Xception: Deep learning with depthwise separable convolutions. *CoRR* abs/1610.02357.

Daood, A.; Ribeiro, E.; and Bush, M. 2016a. Classifying pollen using robust sequence alignment of sparse z-stack volumes. In *International Symposium on Visual Computing*, 331–340. Springer.

Daood, A.; Ribeiro, E.; and Bush, M. 2016b. Pollen grain recognition using deep learning. In *International Symposium on Visual Computing*, 321–330. Springer.

Graves, A., et al. 2012. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

Graves, A., and Schmidhuber, J. 2009. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, 545–552.

Graves, A. 2013. Generating sequences with recurrent neural networks. *CoRR* abs/1308.0850.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. *CoRR* abs/1512.03385.

Hodgson, R. M.; Holdaway, C.; Zhang, Y.; Fountain, D.; and Flenley, J. 2005. Progress towards a system for the automatic recognition of pollen using light microscope images. In *Intl. Symposium on Image and Signal Processing and Analysis (ISPA)*, 76–81.

Holt, K. A., and Bennett, K. D. 2014. Principles and methods for automated palynology. *New Phytologist* 203(3):735–742.

Hopping, C. 1967. Palynology and the oil industry. *Review of Palaeobotany and Palynology* 2:23–48.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1097–1105.

Lagerstrom, R.; Arzhaeva, Y.; Bischof, L.; Haberle, S.; Hopf, F.; and Lovell, D. 2013. A comparison of classification algorithms within the classifynder pollen imaging system. In *Intl. Symposium On Computational Models For Life Sciences*, volume 1559, 250–259. AIP Publishing.

Punyasena, S. W.; Tcheng, D. K.; Wesseln, C.; and Mueller, P. G. 2012. Classifying black and white spruce pollen using layered machine learning. *New Phytologist* 196(3):937–944.

Riley, K. C.; Woodard, J. P.; Hwang, G. M.; and Punyasena, S. W. 2015. Progress towards establishing collection standards for semi-automated pollen classification in forensic geo-historical location applications. *Review of Palaeobotany and Palynology* 221:117 – 127.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Sokolova, M., and Lapalme, G. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management* 45(4):427–437.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, Las Vegas, USA*, 2818–2826.

Treloar, W. J.; Taylor, G. E.; and Flenley, J. R. 2004. Towards automation of palynology 1: analysis of pollen shape and ornamentation using simple geometric measures, derived from scanning electron microscope images. *Journal of Quaternary Science* 19(8):745–754.

Vega, G. L.; Benezeth, Y.; Uhler, M.; Boochs, F.; and Marzani, F. 2012. Sketch of an automatic image based pollen detection system. *32. Wissenschaftlich-Technische Jahrestagung der DGPF* 21:202–209.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, 3320–3328. Cambridge, MA, USA: MIT Press.