Large-scale Pollen Recognition with Deep Learning

André R. de Geus, Celia A. Z. Barcelos
Department of Computer Science
Federal University of Uberlândia
Uberlândia, Brazil
geus.andre@ufu.br, celiazb@ufu.br

Marcos A. Batista, Sérgio F. da Silva
Intitute of Biotecnology
Federal University of Goiás
Catalão, Brazil
marcos.batista@pq.cnpq.br, sergio@ufg.br

***Abstract*—Pollen recognition has been shown to be important for a number of areas ranging from criminal investigations to paleoclimate studies. However, these palynology studies rely on highly qualified professionals to analyze pollen grains, which have become scarce and costly. Therefore, the automation of this task using computational methods is promising. Deep learning has proven to be the ultimate technique in computer vision tasks, but is very difficult to build a pollen data set with size enough to train such networks from scratch. This study investigated the use of transfer learning from pre-trained deep neural networks for pollen classification and compared their results with training from scratch and with promising pre-designed features. Additionally, we introduced the biggest data set of pollen to the date. Experimental results achieved up to 96.24% of classification accuracy, suggesting that the fine-tuned deep learning architectures can be successfully applied to pollen classification.**

***Index Terms*—Pollen recognition, convolutional neural networks, deep learning, transfer learning.**

I. Introduction

Palynology, the scientific discipline concerned with the study of plant pollen, spores and others microscopic planktonic organisms, arises from the morphological formation of such particles or grains. The main features of pollen grains are related to size, shape (polarity, symmetry), openings and ornamentation. Pollen grains are not subject to decay or physical alteration due to variations in environmental conditions, making their genetically established features, generally quite stable and of great diagnostic value. Identification of pollen has greatly aided in delineating the geographical distribution of many plant groups from millions of years ago to the present [1]. Palynological studies have also been helpful to emit alerts for people with allergies to same pollen [2] and to establish the location or seasonal time frame for crime scenes [3], to certificates honey production [4], to determine agricultural practices occurring at archaeological sites [1], and trace relations between different groups of plants and their evolutionary lines [5].

Computer-based pollen recognition relies on two main approaches for image feature extraction: pre-designed feature extraction and automatic feature learning. Pre-designed feature extraction summarizes some specific visual information considering algorithms that describe attributes based on color, texture or shape properties. The automatic feature learning is focused on neural networks that learn some internal representation able to extract features, such as in convolutional neural networks (CNNs).

The researches [6]–[8] focused on the use of texture features. Marcos et al. [7], for example, combines features of Haralick's gray-level cooccurrence matrices (GLCM), log-Gabor filters (LGF), local binary patterns (LBP) and discrete Tchebichef moments (DTM). Results on a 15-class pollen data set were 95% accurate. There are also several researches that employ combinations of shape and texture features [9]–[17]. Tello-Mijares et al. [11] used geometric descriptors, first order texture statistics and second order GLCM-based texture statistics obtained from the L*a*b* color model. The best result that uses linear discriminant analyses (LDA) for dimensionality reduction was 95.6% accurate on a 12-class polen data set. Pozo-Banos et al. [13] also combined geometric and texture features and achieved 94.92% of accuracy on a 17-class pollen data set. Redondo et al. [17] proposed a segmentation-based feature extraction approach, and using LDA for dimensionality reduction achieved 99% of accuracy on a 15-class pollen data set.

Focusing on the relatively new approach called automatic feature learning, Daood et al. [18] explored transfer learning and experimentally fine-tuned the whole CNN. The proposed method was 94% accurate on a 30-class pollen data set. Sevillano and Aznarte [19] achieved 97% of accuracy in a 27-class pollen data set using transfer learning from a pre-trained AlexNet network and a linear discriminant classifier. Wu et al. [20] developed a automatic system for bioaerosol sensing and classification using a Raspberry Pi board and deep learning, achieving 94% of accuracy on six type bioaerosols of plant, including pollen and spores.

This article focuses on the automatic feature learning through CNNs, transfer learning and data augmentation. The main contributions are: 1) the introduction, for our best knowledge, of the largest pollen image data set; 2) a comprehensive experimentation involving state-of-the-art pre-designed features, training of deep-learning architectures from scratch, and two approaches of transfer learning named feature extraction and fine-tuning. Experiments on the introduced 134-class pollen data set suggest that fine-tuning a pre-trained deep CNN on a large data set are applicable to pollen classification achieving up to 96.24% of accuracy.

The remainder of this article is organized as follows: Section II describes the materials and methods used to classify

978-9-0827-9703-9/19/$31.00

pollen types. Section III shows the results and finally Section IV concludes the paper.

II. Material and Methods

A. The 134-class pollen data set

We introduce a new image data set containing 134 types of pollen. The data set was built with different reagents that produces different colors of background (see Fig. 1) and different microscopes. As on each microscope slide there are multiple pollen grains that often overlaps, squared size regions of interest were extracted to isolate as much as possible the pollen grains in each image produced. Table I shows the species of plants corresponding to the pollen data set and the number of samples of each pollen. Fig. 1 illustrates one sample of each pollen given in the same order of Table I when traversing the Fig. 1 in a row by row fashion and the Table I in a column by column fashion. Observing Table I can be seen that the data set contains multiple species of the same genera, which often present similar visual characteristics, as can be seen in Fig. 1.

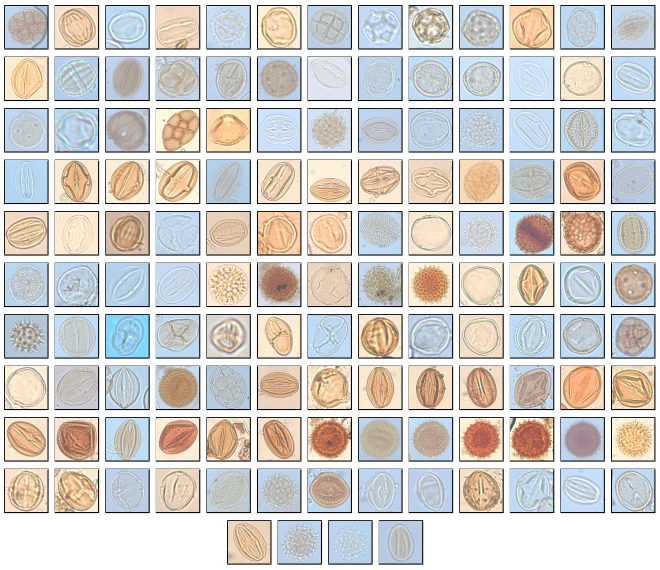

Fig. 1. Samples of each pollen type of our data set.

B. Methodology

The methodology proposed aims to investigate the use of transfer learning from CNNs pre-trained on the 2012 ImageNet data set for pollen classification and compare their results with training from scratch and with promising pre-designed features.

Convolutional Neural Networks (CNNs) and Architectures

Deep CNNs are the most used feature learning model for image classification and recognition [21]–[24]. Its basic composition consists of three types of layers: convolutional layers, subsampling (pooling) layers and fully connected layers. The convolutional layers, known as the heart of CNNs, use the convolution operation to achieve weight-sharing. Subsampling layers are used to reduce the dimensionality which usually uses the average or the maximum on a pool of values. A few fully connected layers, followed by a softmax layer [23], usually constitutes the final layers used for classification, producing a normalized probability for each class.

Given that CNNs are difficult to train with limited computing resources and small data sets, large computing companies such as Google and Microsoft have developed CNNs architectures and trained them in large image data sets. The main idea is that the ability of these networks to extract features can be contribute to new problems.

As known from the CNN theory, the first convolutional layers learn generic features (i.e., edges and color blobs) that are useful for many applications. The network progressively evolves this generic features to more specific features in subsequent layers. Therefore, a set of generic features learned in a large image data set can benefit other applications. In our experiments we used InceptionV3, ResNet and DenseNet architectures trained on the 2012 ImageNet image data set, which contains 1000 classes.

InceptionV3: It was proposed by Google's research team, focusing mainly on reducing the computational load of CNNs, seeking to maintain the same level of performance [24]. The innovation came from the new module called inception which, for most part, can be described as four parallel paths of convolution filters 1×1, 3×3, 5×5. Because of its parallel implementation, in addition to the subsampling layers in each block, the model execution time is fast. InceptionV3 is a large CNN containing 23.2 millions of parameters.

ResNet: It stands for Residual Network [22]. As the name indicates, it introduces what is called residual learning. In general in deep CNNs several layers are stacked and trained to learn features. In residual learning, instead of learning features the network learns residues. Residues can be imagined as a subtraction of features learned from the input layer to the current layer. ResNet learns residues using shortcut connections, that links the nth layer to some $(n + k)$th layer. Literature results have shown that it is easier to train ResNet than tradicional CNNs. ResNet can be configured ranging from 18 to 152 layers having up to 100.11 millions of parameters.

DenseNet: It is a logical extension of ResNet having as fundamental building block the concept of residue connections. In contrast with ResNet, DenseNet proposes to concatenate the previous layers instead of using a summation. DenseNet [21] connects each layer to every other layer in a feed-forward fashion. While traditional CNNs with L layers have L groups of connections – a group between each layer and its subsequent layer, DenseNet has $L(L + 1)/2$ groups of connections. In summary, for each layer, feature maps of all predecessor layers are used as input, and its own output map is used as input to all subsequent layers. The DenseNet used can be considered a small CNN having 7 millions of parameters in our domain.

Networking training

We considered three types of CNN training: training from scratch, fine-tuning of the whole pre-trained network and

TABLE I
POLLEN SPECIES, CLASS ID AND THE NUMBER OF SAMPLES (#) IN EACH CLASS OF THE 134-CLASS POLLEN DATA SET.

ID	Specie	#	ID	Specie	#	ID	Specie	#
0	Adenanthera pavonina L.	25	45	Chamaecrista ramosa	25	90	Piptadenia stipulacea	22
1	Aeschynomene americana	25	46	Chamaecrista repens var. multijuga	34	91	Piptochaetium montevidense	25
2	Aeschynomene benthamii	28	47	Chamaecrista serpens	23	92	Poincianella bracteosa	26
3	Aeschynomene viscidula	22	48	Chloroleucon foliosum	24	93	Prosopis juliflora	31
4	Ageratina asclepiadea	26	49	Citrus x limonia	34	94	Pseudomalachra ciliaris	22
5	Agonandra brasiliensis	29	50	Coccoloba latifolia	26	95	Senegalia riparia	25
6	Albizia lebbeck	11	51	Cocos nucifera	29	96	Senna acuruensis	28
7	Alternanthera tenella	22	52	Combretum lanceolatum	34	97	Senna alata	33
8	Alternanthera brasiliana var. villosa	25	53	Commelina benghalensis	31	98	Senna cana var. hypoleuca	25
9	Amaranthus viridis	26	54	Conocarpus erectus	26	99	Senna chrysocarpa	25
10	Amburana cearensis	27	55	Cordia oncocalyx	29	100	Senna gardneri	28
11	Anadenanthera colubrina	25	56	Curatella americana	29	101	Senna macranthera	23
12	Arcytophyllum nitidum	26	57	Crateva tapia	35	102	Senna occidentalis	32
13	Azadirachta indica	25	58	Crotalaria spectabilis Roth	25	103	Senna pendula	30
14	Asemeia violacea	27	59	Croton hirtus	25	104	Senna pilifera	28
15	Bauhinia monandra	25	60	Cynodon dactylon	26	105	Senna quinquangulata	28
16	Bejaria resinosa	23	61	Dahlia imperialis	23	106	Senna spectabilis	27
17	Bixa orellana	23	62	Dalechampia scandens	26	107	Senna trachypus	35
18	Boerhavia coccinea	25	63	Delonix regia	28	108	Senna uniflora	31
19	Bomarea hirsuta	27	64	Dicliptera mucronifolia	24	109	Senna velutina	27
20	Borago officinalis	27	65	Drimys granadensis	26	110	Sida angustissima	24
21	Borreria spinosa	37	66	Echinodorus lanceolatus	25	111	Sida ciliaris	16
22	Borreria verticillata	34	67	Euphorbia hyssopifolia	28	112	Sida cordifolia	24
23	Brachyotum strigosum	31	68	Euploca Polyphyllum	22	113	Sida galheirensis	26
24	Bromus catharticus	26	69	Froelichia humboldtiana	31	114	Sida linifolia	26
25	Bucquetia glutinosa	47	70	Gossypium hirsutum	21	115	Sida santaremensis	25
26	Bunchosia aff. acuminata	25	71	Guadua trinii	26	116	Sida spinosa	25
27	Byrsonima sericea	29	72	Ipomoea asarifolia	25	117	Simarouba amara	38
28	Caesalpinia pulcherrima	28	73	Ipomoea bahiensis	25	118	Simarouba versicolor	32
29	Calliandra sessilis	24	74	Ischaemum minus	27	119	Solanum paniculatum	31
30	Canavalia brasiliensis	25	75	Laguncularia racemosa	30	120	Spartina ciliata	35
31	Capsicum annuum	28	76	Macroptilium lathyroides	27	121	Spathodea campanulata	28
32	Carduus acanthoides	31	77	Malpighia emarginata	27	122	Sphagneticola trilobata	36
33	Cassia fistula	31	78	Melanthera latifolia	25	123	Spondias macrocarpa	26
34	Cedrela odorata	28	79	Mesosphaerum suaveolens	26	124	Swartzia simplex var. grandiflora	35
35	Centratherum punctatum	30	80	Mimosa arenosa	34	125	Syzygium malaccense	23
36	Centrolobium tomentosum	23	81	Mimosa candollei	24	126	Tanaecium selloi	26
37	Centrosema virginianum	24	82	Mimosa hirsutissima var. hirsutissima	37	127	Tarenaya spinosa	25
38	Chaetocalyx scandes var. pubescens	25	83	Mimosa pigra	30	128	Terminalia catappa	29
39	Chamaecrista calycioides	23	84	Mimosa tenuiflora	26	129	Tephrosia purpurea	21
40	Chamaecrista diphylla	25	85	Mouriri guianensis	25	130	Trischidium molle	32
41	Chamaecrista ensiformis	26	86	Myracrodrum urundeuva	28	131	Tridax procumbens	25
42	Chamaecrista flexuosa	26	87	Olyra latifolia	29	132	Wedelia paludosa	30
43	Chamaecrista hispidula	20	88	Parkinsonia aculeata	29	133	Zornia latifolia	24
44	Chamaecrista nictitans	26	89	Petiveria alliacea	27			

feature extraction (training only the softmax layer). In all cases was employed the minimization algorithm that consider the cross-entropy loss function L:

$$L = -\sum_i p_i \log(f_i) \tag{1}$$

where p is the ground-truth distribution and f is the estimated distribution by the network.

The ground-truth distribution $\mathbf{p}$ is given for each image of the data set. Considering a data set of n samples, containing c number of classes, the distributions are given by:

$$\mathbf{p}^n = (p_1^n, ..., p_c^n) \tag{2}$$

$$p_i = \begin{cases} 1, & \text{if i = class of the image} \\ 0, & \text{otherwise} \end{cases}$$

To update the CNN weights minimizing the cross-entropy we used the stochastic gradient descent method. For that, 10% of the data set were separated for test, 10% for validation and 80% for training. When training from scratch and feature extraction (training only the softmax layer) we chose an initial learning rate of 1e-2 and it was divided by 10 when three consecutively epochs does not improve the validation accuracy. For fine-tuning we initiated the learning rate at 1e-4 since we want to change minimally the pre-learned weights.

Pre-designed features

We experimented HOG, GABOR and LBP methods that have presented promising results in several studies. We also experimented feature combinations and dimensionality reduction by LDA. In all experiments with pre-designed features we used k-nearest neighbor (kNN), with $k = 1$, for classification. The TableII summarizes the experimented pre-designed features, where #feats refers to the number of features.

Preprocessing

As the pre-trained CNNs used for transfer learning have a fixed input layer structure, the pollen samples were resized accordingly. For the pre-designed features the images were resized to 299×299 pixels. The resize process does not deform the samples once they were extracted in square dimension.

TABLE II
DESCRIPTION OF THE PRE-DESIGNED FEATURES. WE DID NOT USE GABOR INDIVIDUALY BECAUSE IT PRODUCES A VERY HIGH NUMBER OF FEATURES INCURRING IN A HIGH COMPUTATIONAL COST FOR CLASSIFICATION.

Method	Description/Setting	#feats
HOG	uses cell size of 32 × 32 pixels and block size of 4 × 4 pixels	1296
LBP	uses cell size of 32 × 32 pixels	4779
Gabor/LDA	applies 40 filters, performs downsampling of 4 × 4 and dimensionality reduction by LDA	134
HOG+LBP	concatenates HOG and LBP features	6075
HOG+(Gabor/LDA)	concatenates HOG e Gabor/LDA features	1430
LBP+(Gabor/LDA)	concatenates LBP e Gabor/LDA features	4913
HOG+LBP+(Gabor/LDA)	concatenates HOG, LBP e Gabor/LDA features	6209
(HOG+LBP+Gabor)/LDA	applies LDA on the concatenation of the features produced by each method	134
HOG/LDA+LBP/LDA+Gabor/LDA	applies LDA on the features produced by each method and concatenates the reduced feature sets	402

Given that the data set contain a small number of samples per class, to apply CNNs it is essential the use of data augmentation. Data augmentation is a ubiquitous technique for increasing the size of labeled training sets by apply transformations that preserve class labels [23], [25], [26]. Data augmentation can be seen as a way to insert knowledge about variances in a task or domain. The most common image transformation are rotation, translation, mirroring, scaling, noise addition and lightning change. The process of data augmentation is also justified by the need of a large volume of data for training the network without causing *overfitting* [26]. In our experiments we augmented the training data set by performing rotations of one degree between -15° and 15° in each image.

III. RESULTS

Our experiments focused on two main analysis: 1) the comparison between training deep CNN architectures from scratch and the use of two modalities of transfer learning (feature extraction and fine-tuning); and 2) comparison between pre-designed features and feature learning.

Fig. 2 shows the accuracy results obtained by the deep CCN architectures InceptionV3, DenseNet-121 and ResNet-50 considering the three type of learning: training from scratch (random initialization), feature extraction and fine-tuning. In both feature extraction and fine-tuning we used the architectures pre-trained on the 2012 ImageNet data set and rebuilt the softmax layer accordingly to the number of class of the data set. The difference is that in feature extraction we only train the softmax layer, while in fine-tuning the weights of the entire network are adjusted. Instead of fixing the number of epochs of training we adopted an early stopping strategy that monitor the validation loss for the last 10 epochs; if no progress is noted on the validation set during the last 10 epochs the training stops. Can be seen on Fig. 2 that feature extraction did not produce a satisfactory results, suggesting that generic features learned in a large data set does not apply to pollen recognition. On the other hand, fine-tuning a pre-trained network on a big data set produced a gain in accuracy of 6.65% in average.

Table III shows the accuracy results of pre-designed features versus feature learning with fine-tuning. Can be seen that fine-tuned CNNs produce far superior results than the defined pre-designed features. We also experimented different sizes of ResNet varying the number of layers from 18 to 152 and noted that the number of layers of ResNet did not increase the accuracy rate. Regarding the CNN architectures can be noted that the best result was obtained by DenseNet, which is the smallest architecture in terms of the number of parameters. When comparing the traditional Inception CNN with residue-based CNNs (ResNet and DenseNet) can be noted that residue-based CNNs achieved better results. This result can be justified due to the densely shortcut connections of DenseNet that minimize the well know problem of vanishing/exploding gradients, which occurs when a large number of layers are stacked.

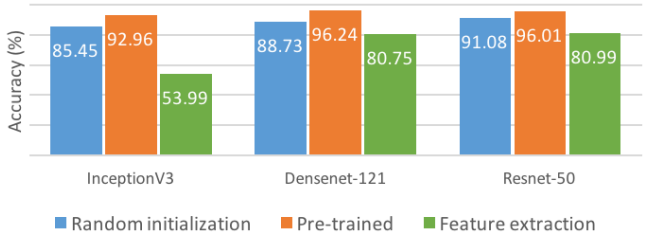

Fig. 2. Accuracy rate comparing randomly initialized weights, feature extraction and fine-tuning.

TABLE III
PRE-DESIGNED FEATURE METHODS COMPARED TO FINE-TUNE FEATURE LEARNING ON DIFFERENT CNN ARCHITECTURES.

	Method/Architecture	Accuracy
Pre-designed features	HOG	35.39%
	LBP	58.18%
	Gabor/LDA	49.09%
	HOG+LBP	56.26%
	HOG+(Gabor/LDA)	42.84%
	LBP+(Gabor/LDA)	59.96%
	HOG+LBP+(Gabor/LDA)	57.98%
	(HOG+LBP+Gabor)/LDA	48.68%
	HOG/LDA+LBP/LDA+Gabor/LDA	57.08%
Feature learning	InceptionV3	92.96%
	DenseNet	96.24%
	Resnet-18	95.54%
	Resnet-34	93.66%
	Resnet-50	96.01%
	Resnet-101	94.84%
	Resnet-152	95.07%

IV. Conclusion

The identification of pollen has great importance in several applications, ranging from scientific to industrial ones. In this study we introduced the largest data set of pollen and compared CNN feature learning with pre-designed features. We also used three promising architectures of CNNs and investigated three ways of training such CNNs: from scratch and two modalities of transfer learning named feature extraction and fine-tuning. Our results show that feature learning produced far more accurate results than pre-designed features, indicating that is quite difficult to design features for a large-scale pollen data set. Regarding CNNs training we found that feature extraction is not enough to produce satisfactory results for our pollen recognition problem. Also we found that training from scratch, even augmenting the training data set at 30 times, produced inferior results (around 6.65% of accuracy) than fine-tuning a pre-trained CNN in a large data set. Considering the CNN architectures we found the better result with DenseNet, achieving up to 96.24% of accuracy. As DenseNet is the smallest CNN regarding the number of parameters, in future work we intend to design a lightweight residual CNN architecture.

Acknowledgment

The authors thank the RCPol - Online Pollen Catalogs Network (FDTE: process number 001505) for the pollen images used in this investigation. We also thank the Brazilian agencies CNPq, CAPES and FAPEMIG for the financial support.

References

[1] A. Augustyn, P. Bauer, B. Duignan, A. Eldridge, E. Gregersen, J. Luebering, A. McKenna, M. Petruzzello, J. P. Rafferty, M. Ray, K. Rogers, A. Tikkanen, J. Wallenfeldt, A. Zeidan, and A. Zelazko., "Palynology," *Encyclopdia Britannica, inc.*, p. online, 2017.

[2] G. DAmato, L. Cecchi, S. Bonini, C. Nunes, I. Annesi-Maesano, H. Behrendt, G. Liccardi, T. Popov, and P. Van Cauwenberge, "Allergenic pollen and pollen allergy in europe," *Allergy*, vol. 62, no. 9, pp. 976–990, 2007.

[3] R. S. D. K. . U. W. R. Szibor, C. Schubert, "Pollen analysis reveals murder season," *Nature*, vol. 395, pp. 449–450, 1998.

[4] H. I. Aljohar, H. M. Maher, J. Albaqami, M. Al-Mehaizie, R. Orfali, R. Orfali, and S. Alrubia, "Physical and chemical screening of honey samples available in the saudi market: An important aspect in the authentication process and quality assessment," *Saudi Pharmaceutical Journal*, vol. 26, no. 7, pp. 932 – 942, 2018.

[5] A. K. M. G. Sarwar, Y. Hoshino, and H. Araki, "Pollen morphology and its taxonomic significance in the genus Bomarea Mirb. (Alstroemeriaceae) - I. Subgenera Baccata, Sphaerine, and Wichuraea," *Acta Botanica Brasilica*, vol. 29, pp. 425 – 432, 09 2015.

[6] P. Li and J. R. Flenley, "Pollen texture identification using neural networks," *Grana*, vol. 38, no. 1, pp. 59–64, 1999.

[7] J. V. Marcos, R. Nava, G. Cristobal, R. Redondo, B. Escalante-Ramirez, G. Bueno, O. Deniz, A. Gonzalez-Porto, C. Pardo, F. Chung, and T. Rodriguez, "Automated pollen identification using microscopic imaging and texture analysis," *Micron*, vol. 68, pp. 36–46, 2015.

[8] O. Ronneberger, H. Burkhardt, and E. Schultz, "General-purpose object recognition in 3d volume data sets using gray-scale invariants - classification of airborne pollen-grains recorded with a confocal laser scanning microscope," in *Object recognition supported by user interaction for service robots*, vol. 2, 2002, pp. 290–295.

[9] G. P. Allen, R. M. Hodgson, S. R. Marsland, and J. R. Flenley, "Machine vision for automated optical recognition and classification of pollen grains or other singulated microscopic objects," in *2008 15th International Conference on Mechatronics and Machine Vision in Practice*, Dec 2008, pp. 221–226.

[10] T. W. J., T. G. E., and F. J. R., "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*, vol. 19, no. 8, pp. 745–754, 2004.

[11] S. Tello-Mijares and F. Flores, "A novel method for the separation of overlapping pollen species for automated detection and classification," *Computational and Mathematical Methods in Medicine*, 2016.

[12] P. Li, W. J. Treloar, J. R. Flenley, and L. Empson, "Towards automation of palynology 2: the use of texture measures and neural network analysis for automated identification of optical images of pollen grains," *Journal of Quaternary Science*, vol. 19, no. 8, pp. 755–762, 2004.

[13] M. del Pozo-Baos, J. R. Ticay-Rivas, J. B. Alonso, and C. M. Travieso, "Features extraction techniques for pollen grain classification," *Neurocomputing*, vol. 150, pp. 377 – 391, 2015.

[14] R. M. Hodgson, C. A. Holdaway, Y. Zhang, D. W. Fountain, and J. R. Flenley, "Progress towards a system for the automatic recognition of pollen using light microscope images," in *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, 2005, pp. 76–81.

[15] Z. Y., F. D. W., H. R. M., F. J. R., and G. S., "Towards automation of palynology 3: pollen pattern recognition using gabor transforms and digital moments," *Journal of Quaternary Science*, vol. 19, no. 8, pp. 763–768, 2004.

[16] P. S. W., T. D. K., W. Cassandra, and M. P. G., "Classifying black and white spruce pollen using layered machine learning," *New Phytologist*, vol. 196, no. 3, pp. 937–944, 2012.

[17] R. Redondo, G. Bueno, F. Chung, R. Nava, J. V. Marcos, G. Cristbal, T. Rodrguez, A. Gonzalez-Porto, C. Pardo, O. Dniz, and B. Escalante-Ramrez, "Pollen segmentation and feature evaluation for automatic classification in bright-field microscopy," *Computers and Electronics in Agriculture*, vol. 110, pp. 56 – 69, 2015.

[18] A. Daood, E. Ribeiro, and M. Bush, "Pollen grain recognition using deep learning," in *Advances in Visual Computing. ISVC 2016*, B. G. et al. (eds), Ed. Springer, Cham, 2016, vol. 10072, ch. 10, pp. 321–330.

[19] V. Sevillano and J. L. Aznarte, "Improving classification of pollen grain images of the polen23e dataset through three different applications of deep learning convolutional neural networks," *PLOS ONE*, vol. 13, no. 9, pp. 1–18, 09 2018.

[20] Y. Wu, A. Calis, Y. Luo, C. Chen, M. Lutton, Y. Rivenson, X. Lin, H. C. Koydemir, Y. Zhang, H. Wang, Z. Grcs, and A. Ozcan, "Label-free bioaerosol sensing using mobile microscopy and deep learning," *ACS Photonics*, vol. 5, no. 11, pp. 4617–4627, 2018.

[21] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 2012, pp. 1097–1105.

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

[25] D. C. Cirean, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[26] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 1171–1179.