Species Recognition from Pollen Images

Nguyen Ha Anh
Hanoi University of Science and Technology
Hanoi, Vietnam
anh.nh224298@sis.hust.edu.vn

Vu Thanh Hang
Hanoi University of Science and Technology
Hanoi, Vietnam
hang.vt224310@sis.hust.edu.vn

Abstract

Pollen classification is a crucial task in various fields such as ecology, agriculture, and medicine, where accurate identification of pollen grains is essential for environmental monitoring, crop management, and allergy prediction. In recent years, machine learning (ML) techniques have gained attention for automating and enhancing the accuracy of pollen identification. This project introduces a pollen classifier that utilizes advanced transfer learning algorithms, particularly the DenseNet121 architecture, to automatically classify pollen grains based on their morphological features extracted from microscopic images. While the method yields promising results, there is still room for further refinement and improvement.

1. Introduction

Pollen is a substance produced by flowering plants, trees, grasses, and weeds to fertilize other plants of the same species. There are various types of pollen, each with distinct morphological features that can be identified under a microscope. Accurate identification of pollen species is crucial for verifying honey quality, reconstructing past vegetation to understand historical climate changes, biodiversity, and human impacts, as well as serving as a forensic tool.

Traditionally, pollen identification is carried out manually with a microscope, which is very time-consuming. This inspection involves the recognition of differences in shapes, texture, and other visual features from the pollen exine that can, sometimes, be very subtle and lead to classification errors by novice palynologists.

With the importance and issues mentioned above, this project aims to develop a robust system for the automated recognition of pollen species through the application of advanced image processing and deep learning techniques. We will address the use of transfer learning techniques to leverage pre-trained models, enhancing the accuracy and efficiency of the system in recognizing pollen species.

1.1. Desired Outcomes

Input: The input consists of microscopic images of pollen grains. For this image-based classification, each input is a high-resolution image which contains one or several pollen grains in an image. The given dataset is provided by the lecturer.

Output: The model should output the predicted class label for each pollen grain image or feature set, identifying the specific species of the pollen grain.

1.2. Challenges

- **High variability in appearance and inter-class similarity:** Pollen grains often have similar features in the same species or genera, and the appearance can vary due to factors like the angle of view, lighting conditions, and the presence of contamination or noise.

- **Limited labeled data:** Large, high-quality labeled datasets for pollen classification may be scarce, which can make training deep learning models challenging.

- **Preprocessing and noise reduction:** Microscopic images can contain noise, contamination, or artifacts, requiring effective image processing and preprocessing techniques to enhance classification accuracy.

2. Related Work

Before the widespread use of machine learning, pollen classification was performed manually by expert botanists and palynologists. Pollen grains are typically examined under a microscope, and key features such as shape, size, surface texture (e.g., spines, grooves), and the number of pores (apertures) are used to distinguish among variations of a species. However, this technique is time-consuming, and it requires high expertise, prone to human error and struggles to handle large datasets since workloads are high [5] . The performance of the feature extraction stage elevated with the rise of SEM (Scanning Electronic Microscope). There is an experiment conducted SEM for feature extraction pollen of Prunus species [3] but still heavily rely on manual labour for the act of samples' preparation.

Further research using CST algorithm, known as color-based technique, for feature extractions [1] .They using 4 Machine Learning techniques has been implemented and fine-tuned for this field of research: 2 variations of SVM (SMO, C-SVC), J48 and KNN and given result that SVM-based classifier outperform the two others.

In recent years, deep learning techniques have shown great promise in automating both feature extraction and classification from raw image data. These models are trained on large annotated datasets and can learn highly discriminative features for classifying pollen grains. But pollen grain labeled data is scarce, transfer learning has become a new promising approach [2]. CNN models final's layer are fully connected layers with softmax activation were replaced by a flattened layer to transform data. Still, Transfer learning requires a sufficient amount of annotated data and computational resources. In this report, we would like to use the DenseNet architecture to classify pollens from 22 different classes.

3. Proposed Method

In the scope of the course's final project, with limited experience and time, and to tackle the challenge of identical pollen structure and small dataset, we proposed a method using transfer learning with DenseNet, a pre-trained Convolutional Neural Network (CNN) model, together with Data Augmentation to enhance the model robustness.

Transfer learning is a technique in machine learning in which knowledge learned from a task is re-used in order to boost performance on a related task.

DenseNet121(Dense Convolutional Network) is a convolutional neural network (CNN) architecture designed to enhance the flow of information and gradients between layers. Introduced by Gao Huang et al. [4], DenseNet has become a popular choice in image processing due to its tightly connected layers. DenseNet121 is a specific variant with 121 layers, and was chosen because of its efficiency with small dataset, and can achieve high performance even with limited computational resources.

Data Augmentation in image classification involves generating new training images by applying various transformations to existing ones. This includes techniques like geometric transformations (rotation, flipping), color adjustments, and noise injection. These methods help increase the diversity of the training set, improving the model's robustness and generalization ability, especially when limited data is available.

3.1. Data structure

Input images

DenseNet121 has some specific input requirements to ensure compatibility and efficiency: Input images must have a fixed size of 224×224 pixels, with 3 color channels (RGB). Pixel values in the image must be normalized to the range $[0, 1]$. Input images must be represented as tensors with the following shape: $(\text{batch\_size}, 224, 224, 3)$.

In order to meet these requirements, we first define a function to preprocess all the images in the dataset. After the preprocessing step, all images will have a shape of $(224, 224, 3)$. List $X = []$ was used to append processed images. Then, the list was converted into a numpy array: $X = \text{np.array}(X)$. After this step, X will have a shape of $(N, 224, 224, 3)$ with N being the number of images.

Input labels

Since this is a multi-class classification task, the labels also need to be one-hot encoded.

We use the list $Y = []$ to append all the class labels corresponding to each processed image. Then, the list was also converted into a numpy array: $Y = \text{np.array}(Y)$. After this step, Y will have a shape of $(N,)$, where N is the number of images, and each element in Y is a string.

To represent Y as a one-hot vector, the `LabelEncoder` was used to convert each string class label in Y to an integer: $Y_{\text{le}} = \text{le.fit\_transform}(Y)$. After that, the integer-encoded labels were converted into one-hot encoded vectors, where each vector has a length equal to M, which is the number of classes: $Y_{\text{cat}} = \text{to\_categorical}(Y_{\text{le}}, M)$. The final shape of Y_{cat} will be (N, M), with each row being a one-hot vector of length M.

The two arrays X and Y_{cat} were then split into training data, validation data, and test data with the ratio of $80/10/10$.

3.2. Model structure

The transfer learning technique requires a base model, in our case DenseNet121. The key feature of DenseNet is that it connects each layer to every other layer in a feed-forward manner. Each layer receives the feature maps from all previous layers, which helps improve gradient flow and reduces the vanishing gradient problem.

Basic Structure of DenseNet121

- **Initial convolutional layer:** A 7×7 kernel with a stride of 2, followed by batch normalization and ReLU activation.
- **Dense Blocks:** 4 Dense Blocks, each consisting of several Dense Layers. Each Dense Layer uses a small 3×3 convolution kernel, followed by batch normalization and ReLU activation.
- **Transition Layers:** Positioned between Dense Blocks to reduce the number of feature maps, each transition

layer consists of a 1×1 convolutional layer to reduce feature maps and a 2×2 average pooling layer for downsampling.

- **Final Layer:** Global Average Pooling (GAP) to reduce dimensionality and a Fully Connected Layer to perform classification.

In the proposed method, only the convolutional base of the model was retained to perform feature extraction. The weights of the model were initialized with the pretrained weights from DenseNet121 and were frozen during training to prevent updates by the backpropagation algorithm. The output of the convolutional layers was a multi-dimensional feature map. Global Average Pooling was applied to convert the feature map into a single 1D vector for each image, which was then fed into fully connected layers. Dropout was used to randomly set some neurons in the previous layer to zero, preventing overfitting.

After feature extraction, non-linear relationships between the features were learned using fully connected layers with ReLU activation. At the output layer, the number of neurons corresponded to the number of classes in the task. The softmax activation function ensured that the output was a probability distribution across all classes, with the sum of the output probabilities equal to 1.

3.3. Algorithm

ImageDataGenerator was used as a data augmentation tool to apply random transformations to the input images during training, improving the model's robustness and reducing overfitting. The logic was encapsulated in an object `data_generator`. The `fit()` function was used to compute any statistics required for the augmentation process, such as feature-wise normalization (mean and standard deviation).

Training Process: The model was compiled using the Adam optimizer, categorical cross-entropy loss function for multi-class classification, and evaluated using the accuracy metric, which measures the proportion of correctly classified instances during training and evaluation.

The model was trained with a batch size of 16 for a maximum of 100 epochs. To enhance training efficiency and prevent overfitting, callback was implemented with EarlyStopping where the training stop if validation loss does not improve for 60 consecutive epochs.

The training data was fed to the model using a data generator (`data_generator.flow`) that creates batches of `X_train` and `Y_train`, while the model's performance was validated on a separate validation set (`X_val`, `Y_val`). Throughout the process, detailed outputs were displayed in the console (`verbose=1`), and training metrics, along with loss values, were stored in the `history` object for further analysis.

4. Experiment

4.1. Dataset

Pollen_VNUA is a dataset created by Vietnam National University of Agriculture. This project will use only a subset of this dataset, containing 5322 images of 22 species.

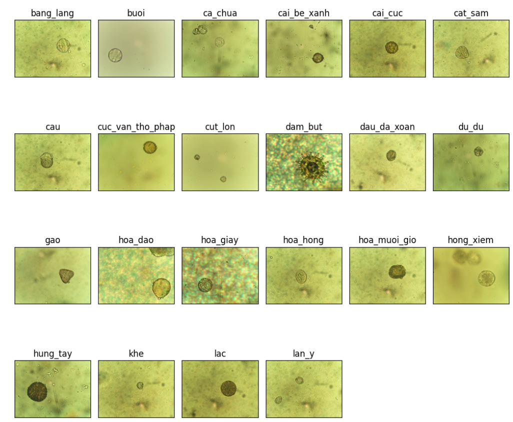

Figure 1. Example images of 22 species from the subset

After several steps of exploration, we created a plot showing the number of images for each species. We observed that the species *hoa giay* has the largest number of samples, with 593 images, while *cai be xanh* has the fewest, with only 69 images. The remaining species have sample sizes ranging from 200 to 400 images.

4.2. Result

After applying transfer learning with the DenseNet121 model, we obtained promising results. The accuracy and loss graphs for both the training and validation sets show steady progress throughout the epochs as shown in Figure 2. Specifically, the training and validation accuracy are closely aligned, suggesting good generalization and a low likelihood of overfitting. The loss for both sets decreases steadily across epochs, indicating that the model is learning relevant features from the data.

The evaluation on the test set yielded an accuracy of approximately 92%, which is considered an acceptable result for the model at this stage.

4.3. Discussion

Although the model performed well on both the training and validation sets, when tested with new input images outside the dataset, it failed to make accurate predictions. This issue suggests a problem with dataset bias. Specifically, the images in the training set share similar backgrounds, and the model may have learned to rely on these background features instead of focusing on the actual object features. As a result, the model struggled to generalize when faced

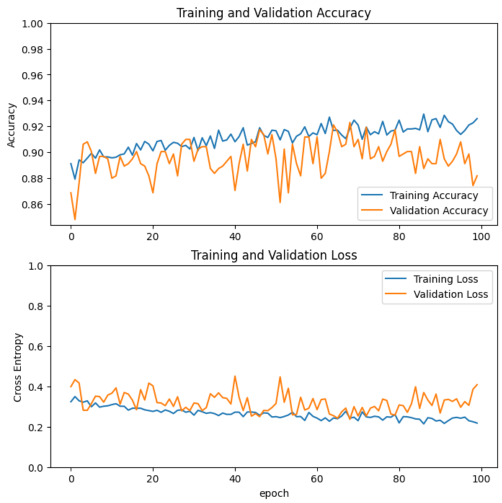

Figure 2. Example images of 22 species from the subset

with new images with different backgrounds or other varying conditions.

Compared to the results on the test set, the accuracy drops significantly on new images, indicating that the model's generalization ability is limited.

To address this issue and improve the model's performance, it is crucial to enrich the training data to include more diverse backgrounds and conditions. Additionally, applying data augmentation techniques could help the model become more robust. Fine-tuning the model to reduce its dependence on background features and focus more on the object-related features will also help improve its generalization capabilities.

5. Conclusion

5.1. Achievement

This project aimed to develop a robust, automated system for classifying pollen species from high-resolution microscopic images using deep learning techniques. To tackle challenges such as high inter-class similarity and limited labeled data, we employed transfer learning with DenseNet121, a pre-trained Convolutional Neural Network (CNN). The model achieved promising results, with a test accuracy of approximately 92%, a notable accomplishment given the constraints of a small dataset and the complexity of pollen morphology. We optimized the training process through techniques like data augmentation and dropout to mitigate overfitting. Additionally, transfer learning allowed the model to leverage pre-trained weights, making it well-suited to the limited labeled data available.

However, when tested with new images outside the training dataset, the model struggled with accurate predictions, likely due to dataset bias.

5.2. Future Work

Future work may include enhancing the preprocessing step by isolating the pollen grains from the background and handling noisy images. Experimenting with different architectures or hyperparameters will also be crucial to improve its performance and help create a more robust model capable of handling diverse real-world data.

References

[1] Gonçalves AB, Souza JS, Silva GGd, Cereda MP, Pott A, Naka MH, and Pistori H. Feature extraction and machine learning for the classification of brazilian savannah pollen grains. *PLOS ONE*, 2016. doi:https://doi.org/10.1371/journal.pone.0157044. 2

[2] Benjamin Garga, Hamadjam Abboubakar, Rodrigue Saoungoumi Sourpele, David Libouga Li Gwet, and Laurent Bitjoka. Pollen grain classification using some convolutional neural network architectures. *J. Imaging*, 2024. doi:https://doi.org/10.3390/jimaging10070158. 2

[3] Anna Geraci, Vincenza Polizzano, Pasquale Marino, and Rosario Schicchi. Investigation on the pollen morphology of traditional cultivars of prunus species in sicily. *Acta Societatis Botanicorum Poloniae*, 2012. doi:https://doi.org/10.5586/asbp.2012.017. 2

[4] Gao Huang, Zhuang Liu, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2018. 2

[5] France I., A.W.G. Duller, Duller G.A.T., and Lamb H.F. A new approach to automated pollen analysis. *Quaternary Science Reviews*, 2000. doi:https://doi.org/10.1016/s0277-3791(99)00021-9. 1