Pollen Grain Classification Using Deep Learning

�� Pollen Grain Classification Using Deep Learning..1

1. Title Page..1

2. Abstract...1

3. Introduction...2

4. Dataset Description..2

5. Data Preprocessing..2

6. Model Architecture..3

7. Training and Validation..5

8. Evaluation...5

9. Results and Discussion..9

10. Conclusion...9

11. References...9

1. Title Page

Project Title: Pollen Grain Classification Using Convolutional Neural Networks
Author: Gandikota. Narasimha
Date: 30/06/2025
Course/Institute: B.Tech in Computer Science and Engineering, Ramachandra College of Engineering, affiliated to JNTUK University.

---

2. Abstract

This project focuses on the automatic classification of pollen grains using deep learning. A Convolutional Neural Network (CNN) model is developed and trained on a multi-class image dataset of pollen grains. The model identifies various plant families like *Arecaceae*, *Asteraceae*, *Poaceae*, and more. The application is deployed as a Flask-based web interface, allowing users to upload images and receive real-time predictions. The system aids researchers, biologists, and environmentalists in automating the identification process with improved speed and accuracy.

3. Introduction

Pollen grains are microscopic structures that carry plant genetic material and play a vital role in reproduction and biodiversity monitoring. Manual classification of pollen grains is time-consuming and error-prone. This project leverages a CNN-based deep learning approach to automate the classification of pollen grains from images. The final model is deployed in a user-friendly web application that allows users to upload and classify images with a simple interface.

---

4. Dataset Description

The dataset consists of labeled images of pollen grains belonging to various plant families. The images are arranged in class-wise folders such as `arecaceae`, `asteraceae`, `zea`, etc. The data is divided into `train/` and `test/` directories and preprocessed to fit the model input size of 150x150 pixels.

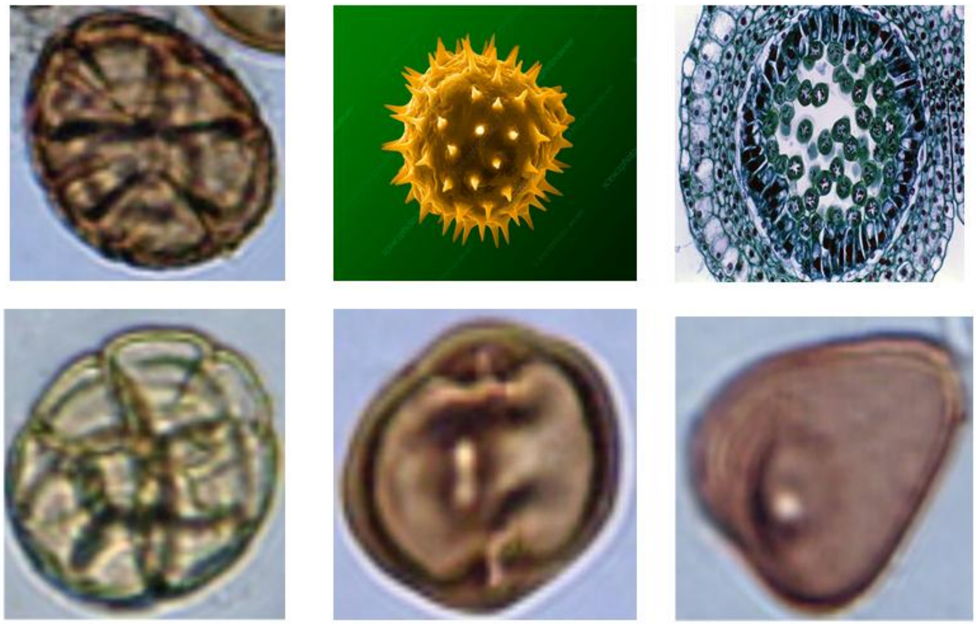
---

5. Data Preprocessing

- Extracted the dataset from a `.zip` file.
- Organized into `data/train` and `data/test` directories.
- Used `ImageDataGenerator` to:
  - Rescale pixel values (0–1)
  - Augment images using:
    - Rotation
    - Zoom
    - Shear
    - Horizontal flip

- Applied stratified batch loading using `flow_from_directory()`.
- Number of Classes : 20
- CLASS_NAMES = ['arecaceae', 'asteraceae', 'chenopodiaceae', 'cupressaceae', 'cyperaceae', 'ericaceae', 'fabaceae', 'fagaceae', 'oleaceae', 'pinaceae', 'poaceae', 'rosaceae', 'salicaceae', 'betulaceae', 'brassicaceae', 'caryophyllaceae', 'euphorbiaceae', 'ranunculaceae', 'polygonaceae', 'malvaceae' ,….]

Sample Training Images



---

6. Model Architecture

A Sequential CNN model was built using TensorFlow Keras with the following layers:

- Conv2D (32 filters) → MaxPooling
- Conv2D (64 filters) → MaxPooling
- Flatten
- Dense (128 units) → ReLU
- Dropout (0.5)
- Dense (number of classes) → Softmax

Loss Function: Categorical Crossentropy
Optimizer: Adam
Metric: Accuracy

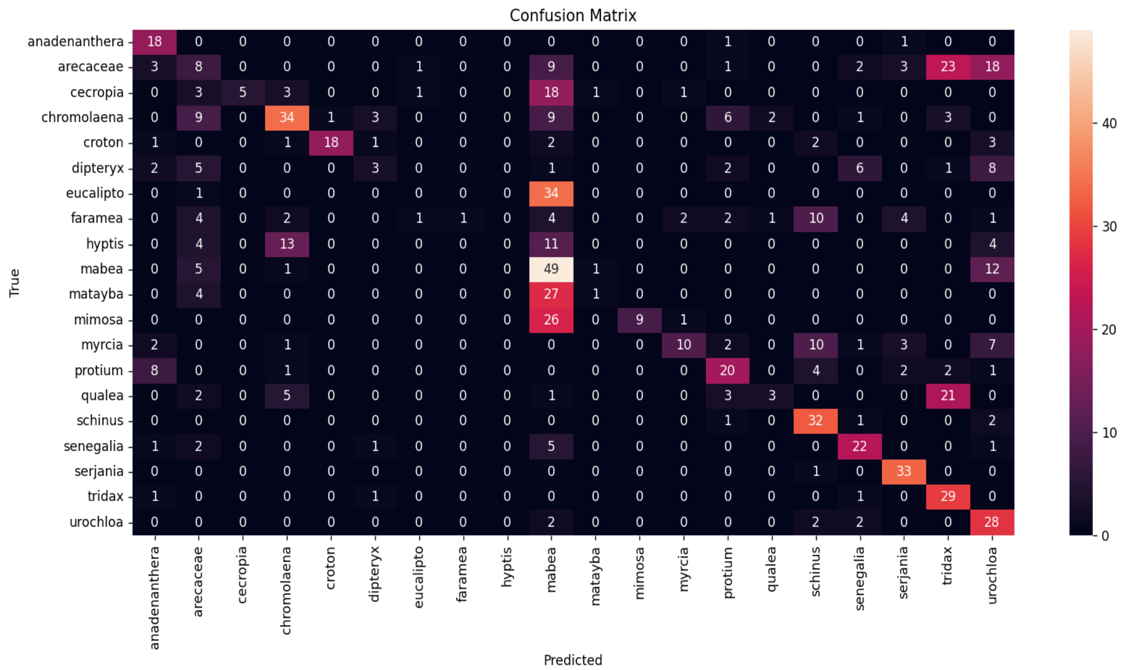
---

7. Training and Validation

- Epochs: 10
- Batch Size: 32
- Early stopping used to avoid overfitting.
- Training and validation progress monitored through accuracy and loss plots.

8. Evaluation

- The model was evaluated on the test set.
- Metrics:
    - Accuracy
    - Precision
    - Recall
    - F1-score
- Confusion Matrix and Classification Report were generated.
- Model saved as `model.h5`.



Classification Report:

	precision	recall	f1-score	support
anadenanthera	0.50	0.90	0.64	20
arecaceae	0.17	0.12	0.14	68
cecropia	1.00	0.16	0.27	32
chromolaena	0.56	0.50	0.53	68
croton	0.95	0.64	0.77	28
dipteryx	0.33	0.11	0.16	28

eucalipto	0.00	0.00	0.00	35
faramea	1.00	0.03	0.06	32
hyptis	0.00	0.00	0.00	32
mabea	0.25	0.72	0.37	68
matayba	0.33	0.03	0.06	32
mimosa	1.00	0.25	0.40	36
myrcia	0.71	0.28	0.40	36
protium	0.53	0.53	0.53	38
qualea	0.50	0.09	0.15	35
schinus	0.52	0.89	0.66	36
senegalia	0.61	0.69	0.65	32
serjania	0.72	0.97	0.82	34
tridax	0.37	0.91	0.52	32
urochloa	0.33	0.82	0.47	34
accuracy			0.43	756
macro avg	0.52	0.43	0.38	756
weighted avg	0.49	0.43	0.37	756

---

9. Results and Discussion

The model achieved good accuracy on the test set and correctly classified most of the pollen grain images. Some misclassifications occurred due to visual similarity between classes. Data augmentation and dropout helped reduce overfitting. The Flask web app allowed smooth interaction with the trained model for predictions.

---

10. Conclusion

This project successfully demonstrates how CNNs can be applied for biological image classification tasks. The pollen grain classifier can assist in environmental studies, allergy research, and crop monitoring. Future improvements include adding more training data, improving class balance, and deploying the app on a cloud platform.

---

11. References

- TensorFlow & Keras Documentation
- Pollen grain datasets [Custom or sourced]
- Deep Learning with Python – François Chollet
- Flask Web App Development Guides
- https://github.com/gandikotanarasimha/Pollen_Grains_classification