Pollen Profiling: Automated Classification of Pollen Grains

1. Introduction

Pollen Profiling is a deep learning project that automates the classification of pollen grains based on their morphological features such as shape, size, and texture. It uses a Convolutional Neural Network (CNN) for classification and a Flask web application for deployment.

2. Objectives

• Automate pollen grain classification using CNNs.
• Provide a user-friendly web interface for image uploads.
• Support environmental monitoring, allergy diagnosis, and agricultural research.

3. Applications

1. Environmental Monitoring: Study biodiversity and ecosystem health.
2. Allergy Diagnosis: Identify pollen allergens for treatment plans.
3. Agriculture: Improve crop breeding and pollination research.

4. Project Workflow

1. Data Collection from Kaggle datasets.
2. Data Preprocessing: resizing, normalization, and splitting.
3. Model Building with Conv2D, MaxPooling, Dense layers.
4. Model Training using ImageDataGenerator for augmentation.
5. Model Evaluation with accuracy and F1-score.
6. Flask Application for real-time predictions.

5. Technical Stack

• Python
• TensorFlow, Keras
• OpenCV, NumPy, Pillow
• Flask
• HTML, CSS

6. Project Structure

PollenProfiling/
■■■ app.py # Flask app
■■■ model.py # CNN model building & training
■■■ requirements.txt # Dependencies
■■■ README.md # Quick start guide
■■■ DOCUMENTATION.pdf # Full project documentation
■■■ static/ # CSS styles
■■■ templates/ # HTML templates
■■■ saved_model/ # Trained model (pollen_model.h5)

7. How to Run

Local Machine:
1. pip install -r requirements.txt
2. python model.py (optional training)
3. python app.py
4. Visit http://127.0.0.1:5000

Google Colab:
• Upload project ZIP
• Train using model.py
• Run Flask app with flask-ngrok for public access.

8. Results

• Predicts pollen type with confidence score.
• Flask app provides real-time results.
• Dummy model fallback ensures demo works without training.

9. Future Improvements

• Larger datasets for better accuracy.
• Deploy on cloud (Heroku, AWS).
• Add REST API for mobile integration.
• Visualization dashboard for pollen monitoring.

10. Conclusion

The project showcases how deep learning and web apps can solve real-world biological and environmental challenges. With future improvements, it can become a valuable tool for researchers, healthcare professionals, and farmers.