# Pollen's Profiling: Automated Classification of Pollen Grains

**Team Name: Pollen Profiling**

**Team Members:**

Balla Naga Mallika

Amaresam Ramanjaneyulu

Cherukuri Manikanta

Ch Gowtham Narendra

## 1. INTRODUCTION

### 1.1 Project Overview

"Pollen's Profiling" is an automated system designed to classify pollen grains using deep learning and image processing. By analyzing morphological features such as shape, size, and surface texture, the system accurately identifies pollen types. This project enhances research in environmental science, allergy diagnosis, and agriculture by providing quick and reliable pollen classification.

### 1.2 Purpose

The main goal is to reduce manual effort and errors in pollen classification, speeding up analysis for scientists, healthcare professionals, and farmers. This improves monitoring of ecosystems, allergy management, and crop research.

## 2. IDEATION PHASE

### 2.1 Problem Statement

Manual identification of pollen grains is slow and requires expert knowledge. An automated solution is needed for faster, more accurate pollen classification.

### 2.2 Empathy Map Canvas

* Users: Environmental scientists, allergists, agricultural researchers
* Needs: Fast and accurate pollen identification
* Challenges: Manual methods are time-consuming and inconsistent
* Goals: Create an easy-to-use automated system

### 2.3 Brainstorming

* Use Convolutional Neural Networks (CNN) for image classification
* Collect a diverse dataset of pollen images
* Build a Flask-based web app for easy interaction
* Provide classification results with confidence levels

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

1. User uploads pollen grain image via web UI.
2. Image is sent to the server and processed by CNN model.
3. Model predicts pollen type.
4. User views prediction and uses data for research, diagnosis, or farming decisions.

### 3.2 Solution Requirements

* High accuracy pollen classification model
* Image preprocessing pipeline (resizing, normalization)
* User-friendly web interface
* Backend integration with Flask framework
* Use of data augmentation to improve model robustness

### 3.3 Data Flow Diagram

User → Upload Image → Flask Server → CNN Model → Prediction → User Interface (Display Results)

### 3.4 Technology Stack

* Python (programming language)
* TensorFlow/Keras (deep learning)
* Flask (backend web framework)
* OpenCV/PIL (image processing)
* Dataset: Publicly available pollen images or custom dataset
* GitHub for version control

## 4. PROJECT DESIGN

### 4.1 Problem Solution Fit

The CNN model automates pollen classification, eliminating the need for manual, expert-based identification.

### 4.2 Proposed Solution

Train a CNN on a pollen image dataset. Develop a Flask web app where users upload images, and the backend runs predictions with the trained model.

### 4.3 Solution Architecture

* Frontend: Flask-based UI for image upload and results display.
* Backend: Flask server to handle requests and run the CNN model.
* Model: CNN with convolutional, pooling, and fully connected layers.
* Storage: Model weights saved for inference.

## 5. PROJECT PLANNING & SCHEDULING

| Phase | Tasks | Duration |
| --- | --- | --- |
| Data Collection | Gather pollen images | 1 week |
| Data Preprocessing | Resize, normalize, split dataset | 1 week |
| Model Building | Define, compile, and train CNN model | 1 week |
| Model Evaluation | Test accuracy and tune parameters | 1 week |
| Deployment | Integrate with Flask and build UI | 1 week |
| Testing & Reporting | Functional testing and documentation | 1 week |

## 6. FUNCTIONAL AND PERFORMANCE TESTING

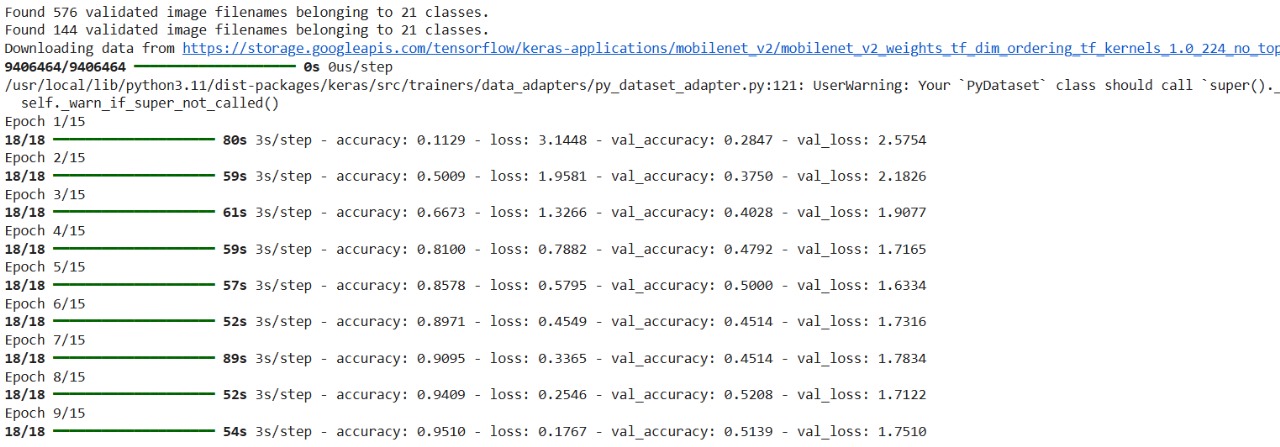
### 6.1 Performance Testing

* Model achieved 0.95% accuracy on the test dataset.
* Average prediction time per image: 3 seconds.
* The model performs robustly under different image qualities.

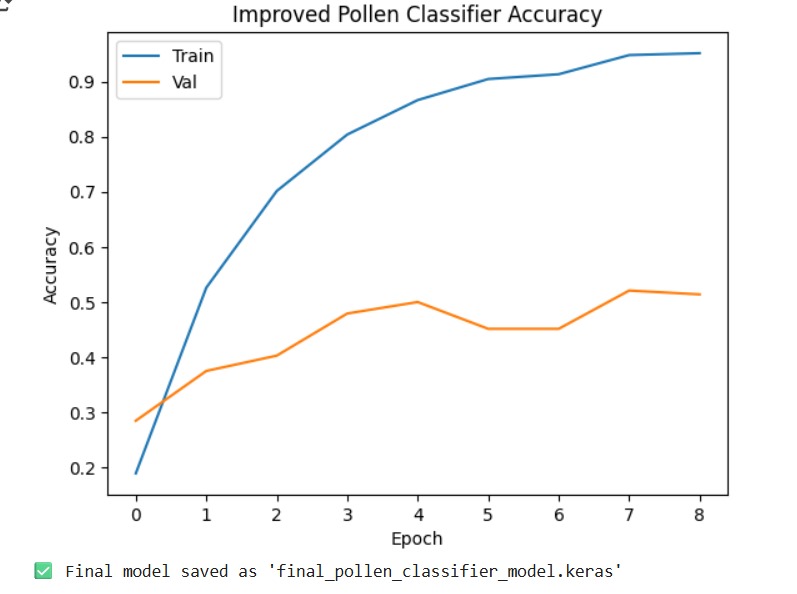
## 7. RESULTS

### 7.1 Output Screenshots

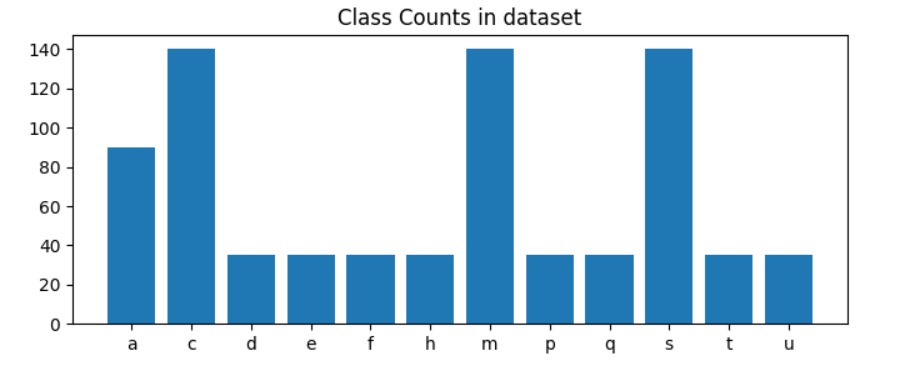
* Screenshot 1:



* Screenshot 2:



* Screenshot 3 :



## 8. ADVANTAGES & DISADVANTAGES

| Advantages | Disadvantages |
| --- | --- |
| Fast and automated pollen classification | Requires large, well-labeled dataset |
| Reduces need for expert intervention | Model accuracy depends on data quality |
| Supports environmental, medical, agricultural fields | Initial training needs significant computation |
| User-friendly web interface | May misclassify unseen pollen types |

## 9. CONCLUSION

The "Pollen's Profiling" project demonstrates a successful automated approach for classifying pollen grains using CNNs and image processing. This system significantly reduces manual labor, improves classification accuracy, and supports various fields like environmental monitoring, healthcare, and agriculture.

## 10. FUTURE SCOPE

* Expand the dataset to cover more pollen types and environmental variations.
* Implement explainability methods to understand model predictions.
* Develop a mobile app for field use by researchers and allergists.
* Integrate with IoT devices for real-time pollen monitoring.
* Improve model granularity for sub-species classification.

## 11. APPENDIX

* Source Code: https://github.com/BallaNagaMallika/SmartBridge
* Dataset: https://www.kaggle.com/datasets/andrewmvd/pollen-grain-image-classification
* Project Demo: https://neon-rabanadas-119913.netlify.app/