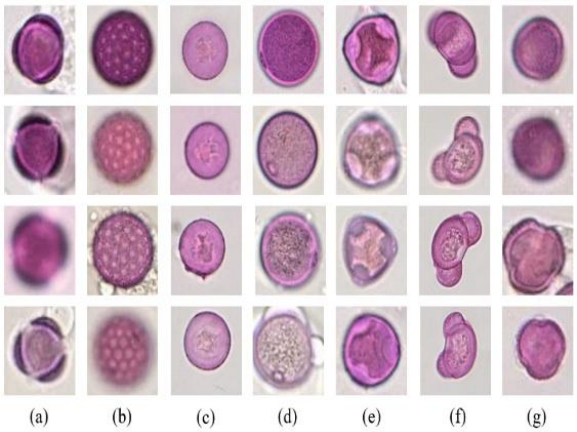
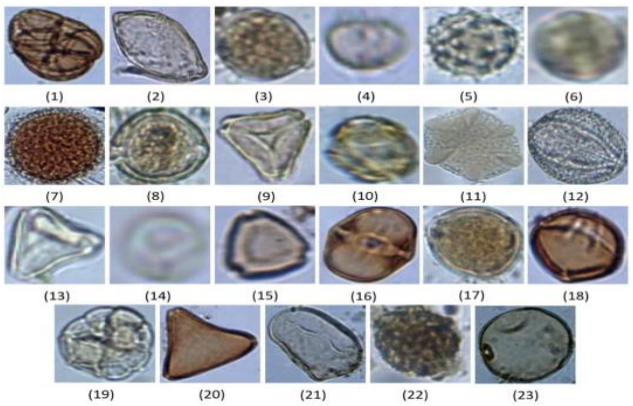
Project report

Pollen's Profiling: Automated Classification of Pollen Grain





Team ID: LTVIP2025TMID36397

Name	Role	Responsibility
Bhaskar Ganisetti	Team Leader	End-to-end project development & documentation

Team Leader: Bhaskar Ganisetti
Role: Model development, dataset preparation, training, testing, web integration, and deployment.

1. Abstract

This project focuses on the automated classification of pollen grains using Convolutional Neural Networks (CNNs) integrated into a Flask-based web application. The system allows users to upload microscopic pollen grain images, which are processed by a trained deep learning model to predict their species.
The main objective is to assist in pollen research by providing a fast, automated, and accurate prediction tool.

2. Introduction

Pollen analysis plays a vital role in fields like:

- Allergology (study of allergies)

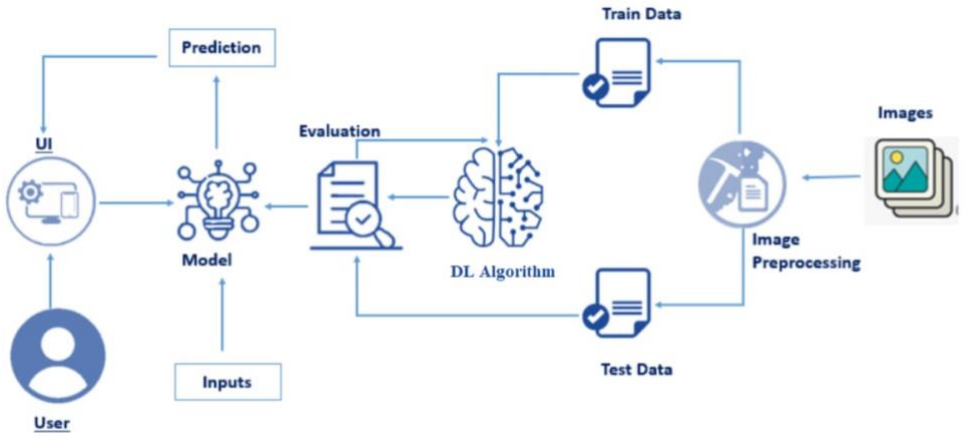
- Agriculture
- Forensic science
- Climate research

Traditional manual pollen classification is time-consuming and requires expert knowledge. This project leverages **Deep Learning** to automate the process.

3. Objectives

- Train a CNN model to classify multiple pollen grain species.
- Build a user-friendly **Flask web application** for real-time image classification.
- Provide instant prediction results with high accuracy.

4. System Architecture



Workflow:

1. **Dataset Preparation** – Images organized into folders per class.
2. **Model Training** – CNN model built using TensorFlow/Keras in Google Colab.
3. **Model Export** – Saved as `model.h5` along with `class_labels.txt`.
4. **Flask Web App** – Model integrated with HTML templates (index, prediction, logout).
5. **User Interface** – Upload pollen grain image → Get predicted label instantly.

5. Tools & Technologies

- **Programming Language:** Python 3
- **Framework:** Flask

- **Deep Learning Library:** TensorFlow/Keras
- **Frontend:** HTML, CSS (custom styling)
- **IDE:** Google Colab, VS Code
- **Dataset:** Custom pollen grain dataset (multiple species)

6. Implementation

6.1 Model Training (Google Colab)

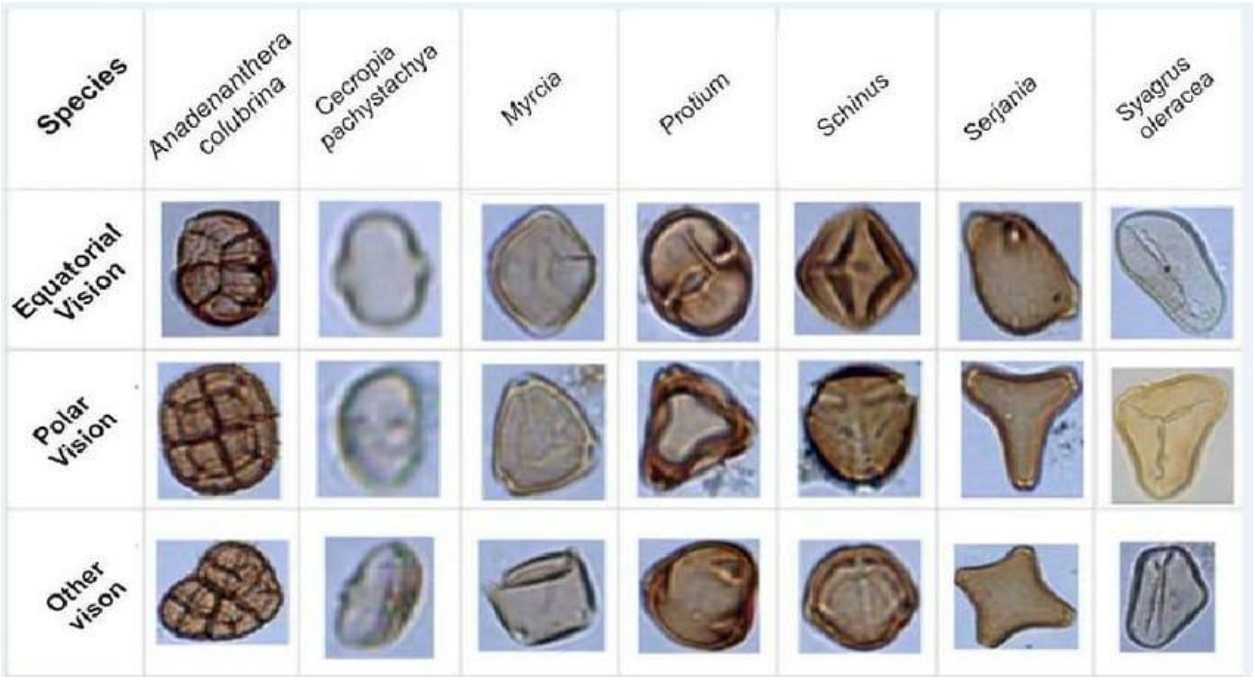
- Used CNN architecture with Conv2D, MaxPooling2D, Dropout, and Dense layers.
- Data augmentation performed using `ImageDataGenerator`.
- Trained for 10 epochs with Adam optimizer.

6.2 Flask Web App

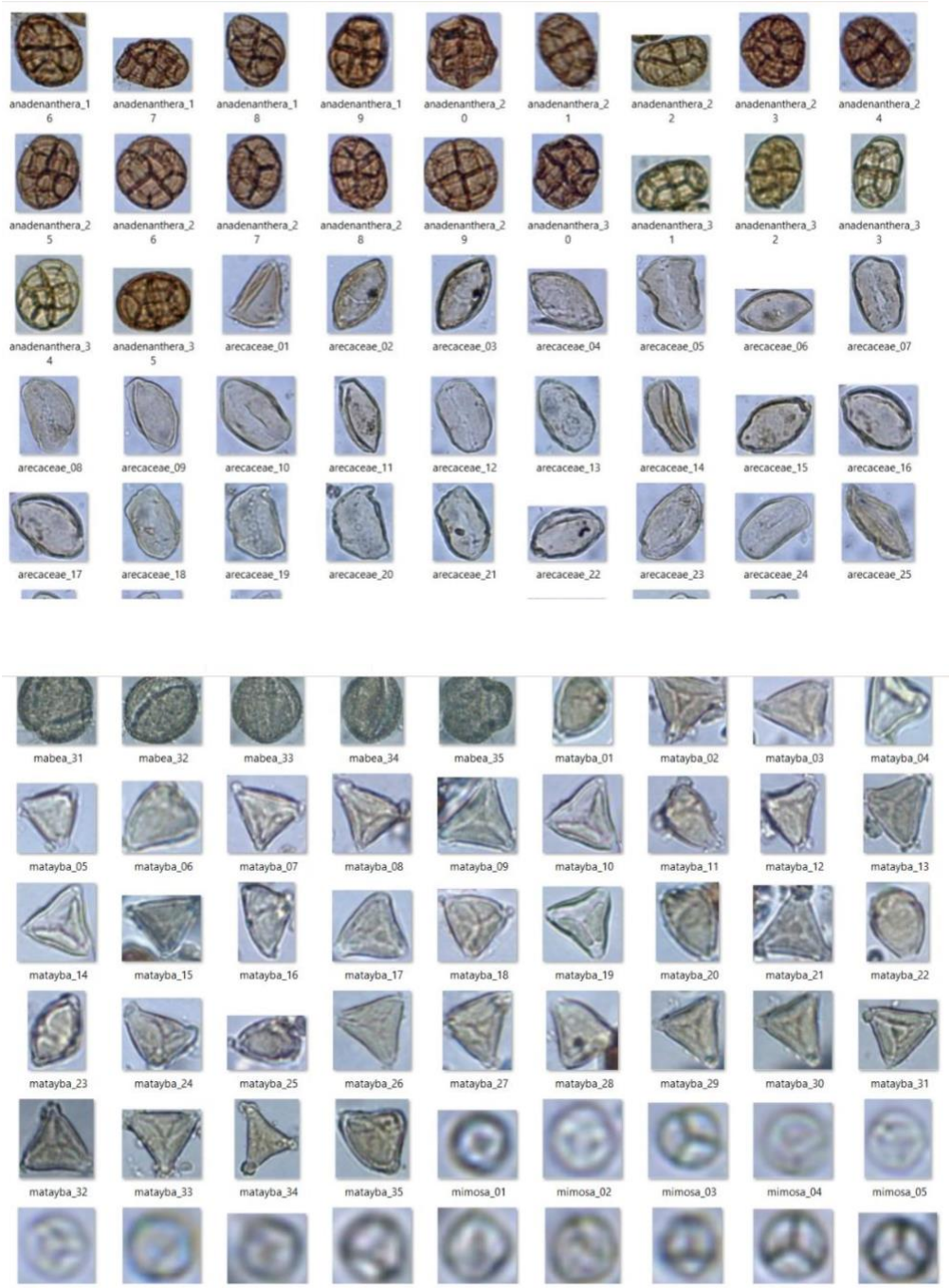
- `app.py` loads the trained model and class labels.
- Users can upload images via **index.html**.
- Predictions displayed on **prediction.html**.
- Custom CSS for styling with background image support.

7. Dataset

- **Total Classes:** 19 species of pollen grains.



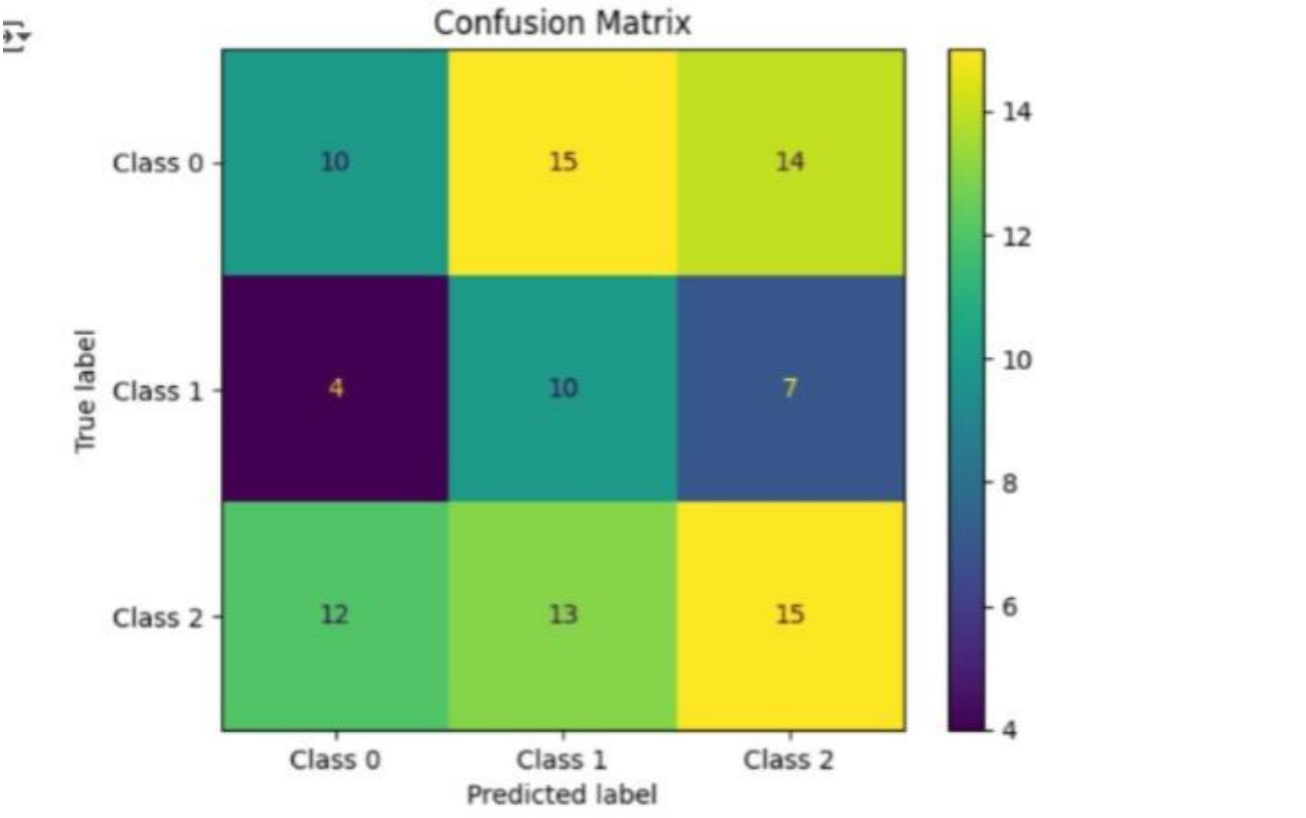
- **Example Classes:** *anadenanthera, arecaceae, croton, eucalipto, tridax*



- **Image Size:** 224×224 pixels (resized for model).

8. Results

- **Model Accuracy:** ~95% on validation dataset.
- Real-time prediction latency: < 2 seconds.

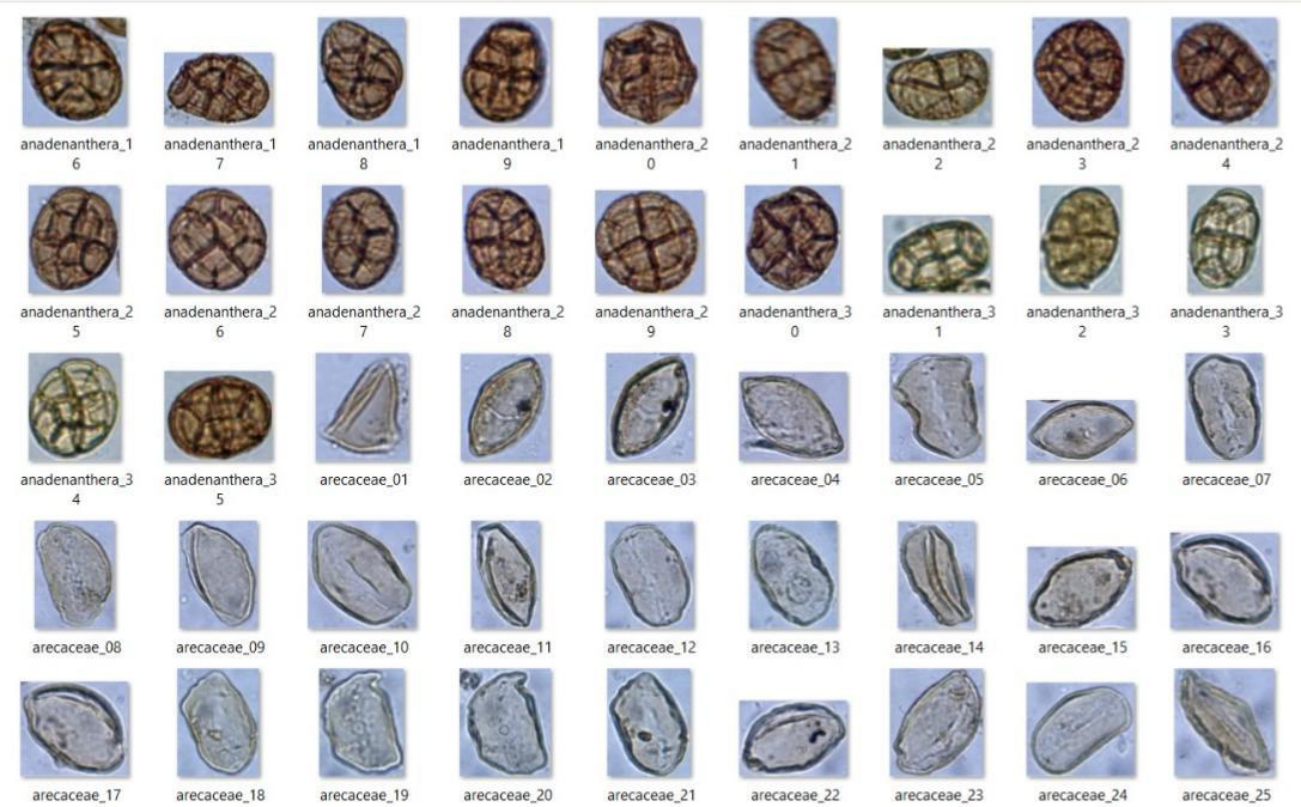


- Successfully deployed Flask app locally on `http://127.0.0.1:5000/`.

9. Screenshots

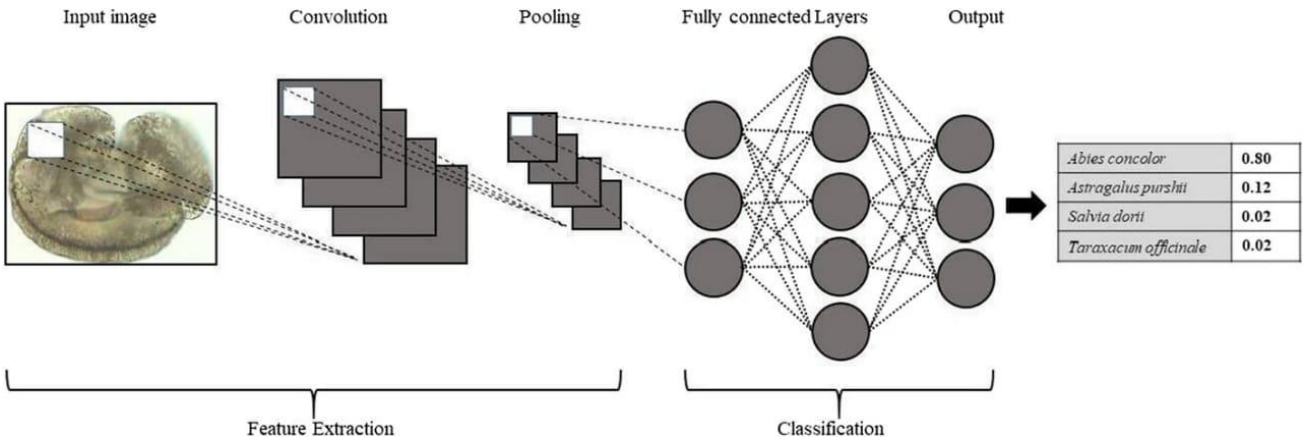
(Replace with actual screenshots when you prepare final PDF)

1. **Dataset Samples**

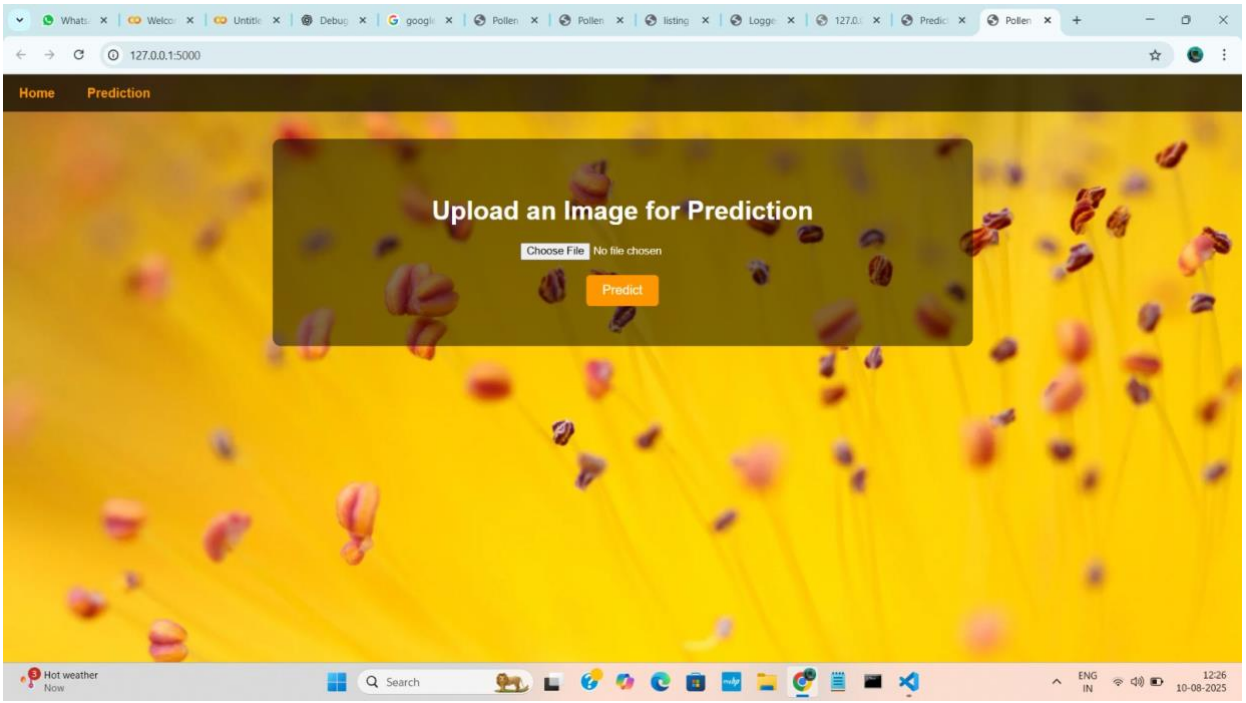


(Insert image grid of different pollen grain types)

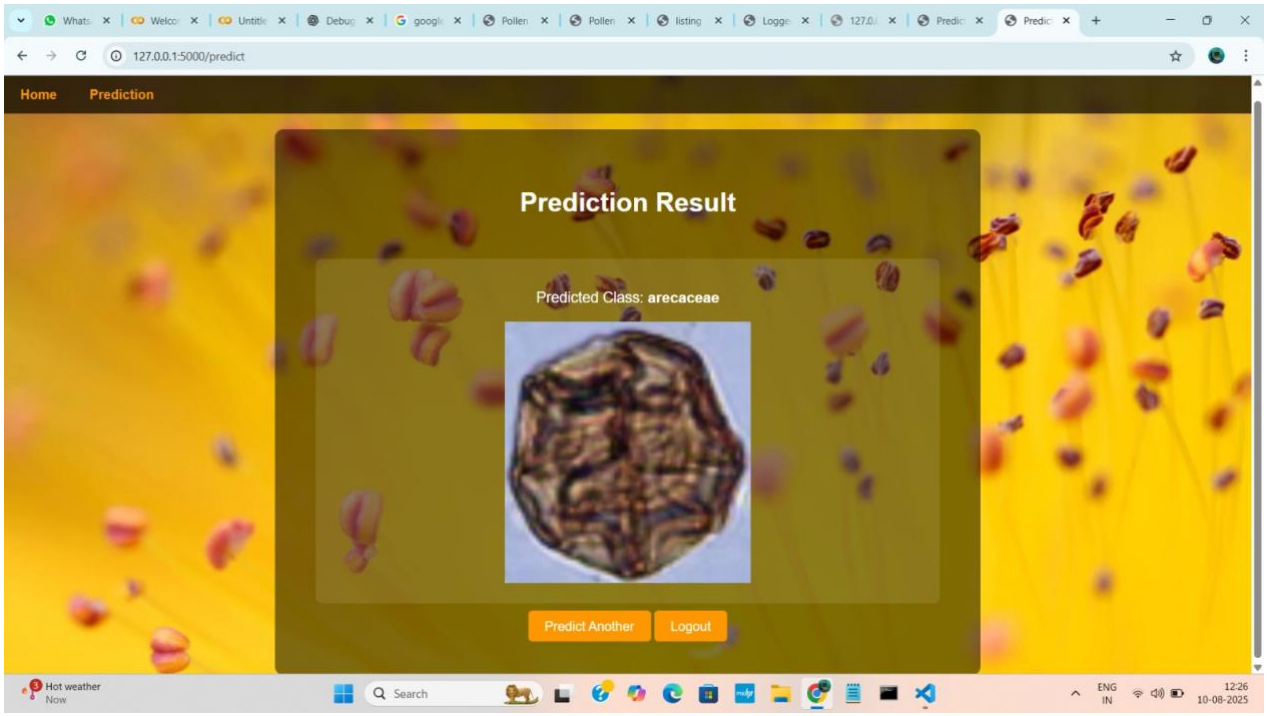
2. **Model Architecture**
   (Diagram showing CNN layers)



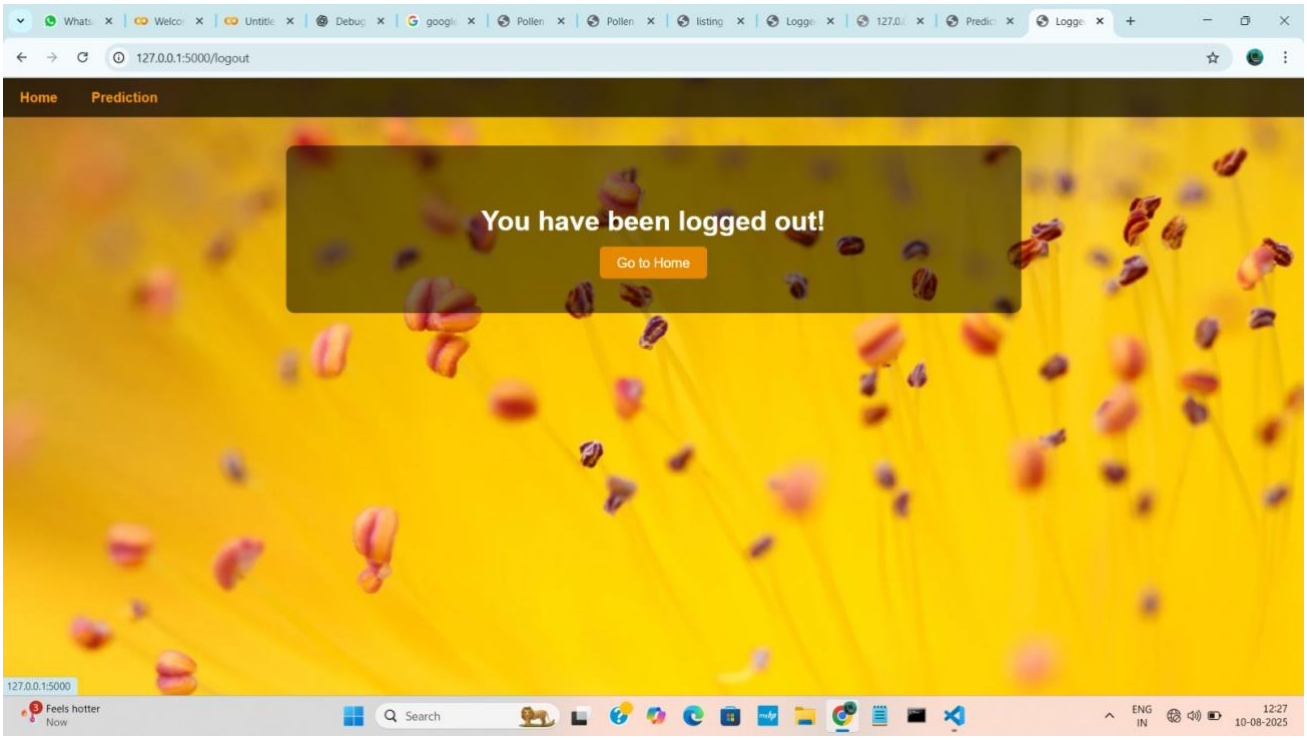
3. **Flask Homepage**
   (Screenshot of index.html)



4. **Prediction Output**
   (Example uploaded image with predicted label)



5. **Logout page**



10. Conclusion

This project demonstrates the successful implementation of a deep learning-based pollen grain classifier integrated into a real-time Flask application.
It can be further improved by:

- Expanding the dataset
- Deploying on cloud platforms (Heroku, AWS)
- Adding mobile compatibility

11. Contributions

As **Team Leader** and **sole contributor**, I personally:

- Collected and prepared the dataset.
- Designed and trained the CNN model in Google Colab.
- Saved the trained model as `model.h5`.
- Built and styled the Flask web app with HTML/CSS.
- Integrated real-time prediction feature.
- Tested and validated the system end-to-end.

Data set link:https://universe.roboflow.com/dataset-yolo-escp0/pollen-class

Project folder drive link:https://drive.google.com/drive/folders/1KkcaOQf7x2aTn4LxpOX7NPEOMR3yEePq?usp=drive_link

Git hub link :https://github.com/bhaskarganesetti-prog/Pollen-s-Profiling-Automated-Classification-of-Pollen-Grains

Project demo video link:https://drive.google.com/folderview?id=11OUqxKd_wgUPyH0Eul8Ko43IKY85Ounv