Pollen's Profiling: Automated Classification of Pollen Grains

Welcome to Pollen's Profiling! Our project focuses on revolutionizing pollen analysis through automated classification using deep learning. This presentation outlines our methodology, findings, and the exciting future of this technology.

Made with GAMMA

Introduction to Automated Pollen Classification

Addressing Manual Limitations

Manual pollen classification is time-consuming and prone to human error. Our project automates this critical process using advanced deep learning techniques.

Diverse Applications

Automated pollen classification has wide-ranging applications, from allergy research and agricultural studies to forensic analysis and honey purity testing.

Dataset Overview

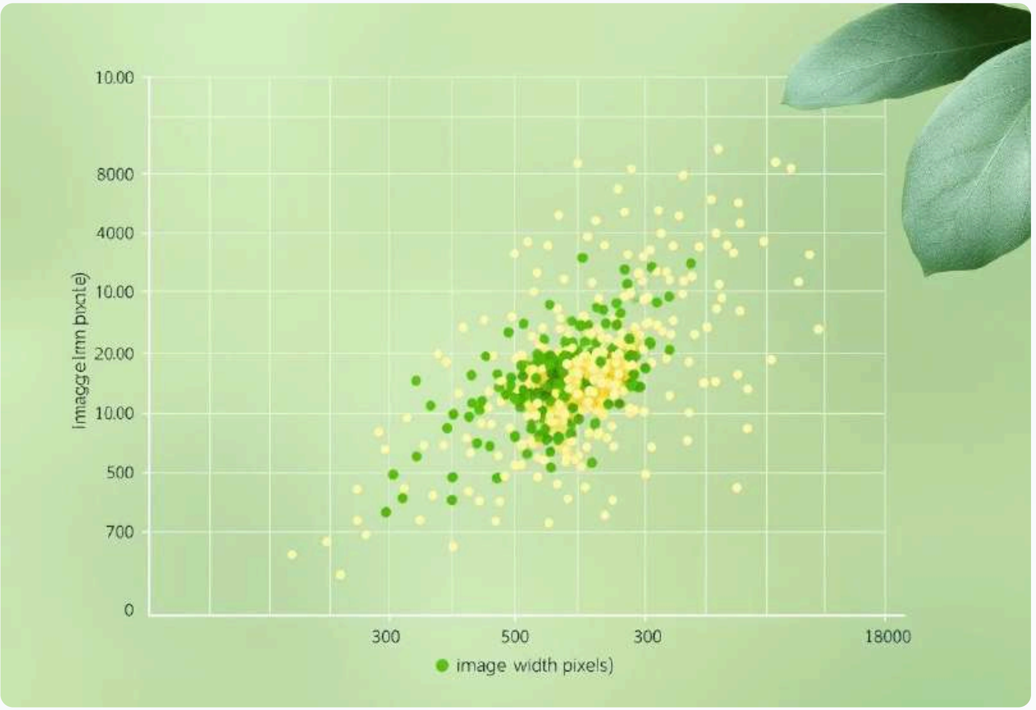
- Total Images: **790** high-resolution images.
- Classes derived directly from folder names, ensuring clear categorization.
- Visual class distribution analyzed via bar plots for insight.
- Sample images from each class showcased for diverse representation.

Made with GAMMA

Image Analysis and Preprocessing

Image Size Distribution

We analyzed image dimensions using OpenCV. A scatterplot revealed that the majority of images comfortably fit within an 800x800 pixel window, optimizing processing efficiency.



Data Preprocessing Steps

- Images were normalized using OpenCV to standardize pixel values across the dataset.
- All images were consistently resized to 128x128 pixels, preparing them for model input.
- Labels were efficiently encoded using LabelEncoder.
- One-hot encoding was applied to the labels for robust multi-classification.

Made with GAMMA

Model Architecture and Data Splitting

Data Splitting Strategy

- Data was split into training and testing sets with approximately 71.5% for training and 28.5% for testing.
- Stratified sampling was employed to ensure proportional representation of all pollen classes in both sets.
- Class weights were utilized to balance the dataset, addressing any potential class imbalances.

CNN Model Architecture

1

3 Convolutional Layers

Featuring increasing filter sizes (32, 64, 128) to extract hierarchical features.

2

Normalization & Pooling

Batch Normalization for stable training and MaxPooling for dimensionality reduction.

3

Dense & Output Layers

A 256-unit dense layer feeds into a 23-neuron softmax output layer for classification.

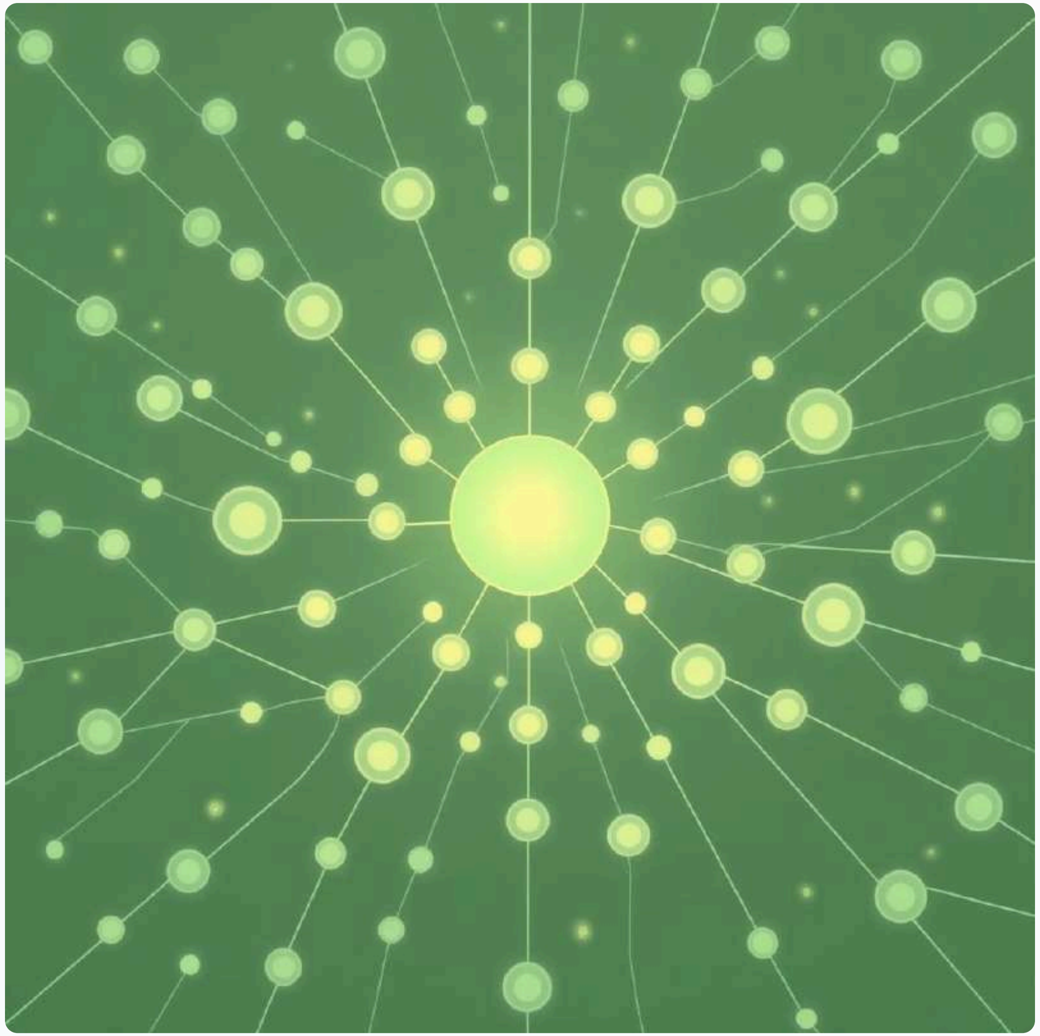
4

Regularization

Dropout layers are strategically placed to prevent overfitting.

Made with GAMMA

Model Summary and Optimization



Key Model Statistics

- The model boasts a substantial **8.48 Million** total parameters, all trainable.
- A minimal 448 non-trainable parameters, indicating an efficient architecture.

Optimization Details

- We utilized the **Adam Optimizer** for its adaptive learning rate capabilities.
- The **Categorical Crossentropy** loss function was chosen, perfectly suited for multi-class classification tasks.

Made with GAMMA

Training Setup and Results

Training Configuration

- **ImageDataGenerator** was instrumental for real-time data augmentation.
- The model was trained over **500 epochs** with a batch size of **32**.
- Augmentation included rotation, zoom, shifts, and horizontal/vertical flips for enhanced generalization.

Training Results Overview

200+

Epochs Trained

The model was rigorously trained for over 200 epochs.

85-90%

Final Accuracy

Achieved a high final accuracy of approximately 85-90%.

100

Loss Stabilization

Loss functions stabilized consistently after 100 epochs.

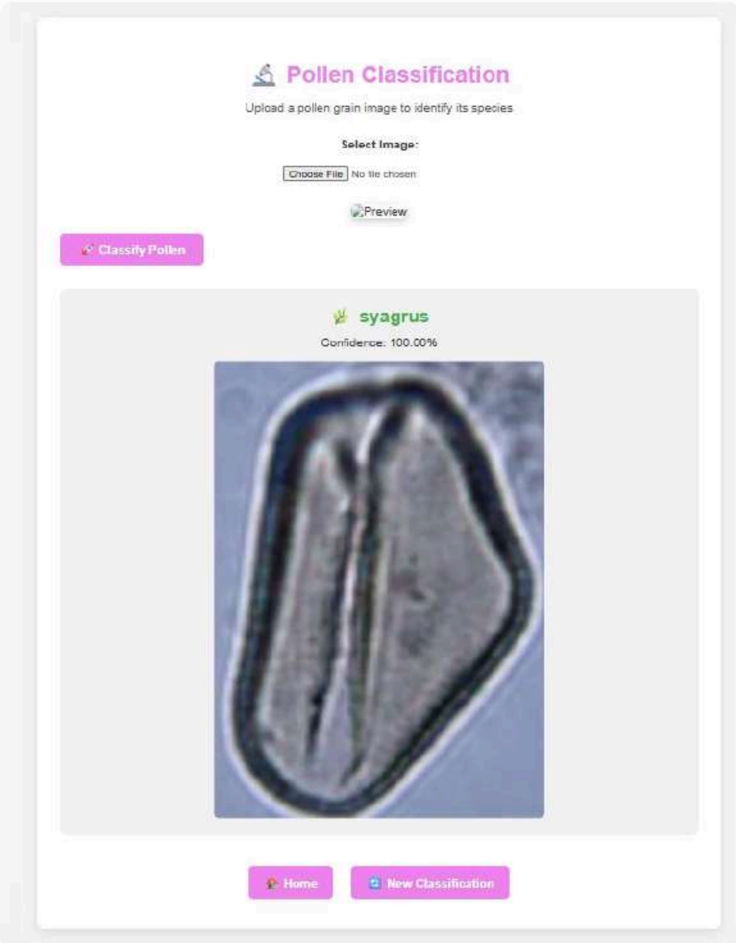
89.82%

Peak Validation Accuracy

Validation accuracy reached an impressive peak of 89.82%.

Made with GAMMA

sample outputs

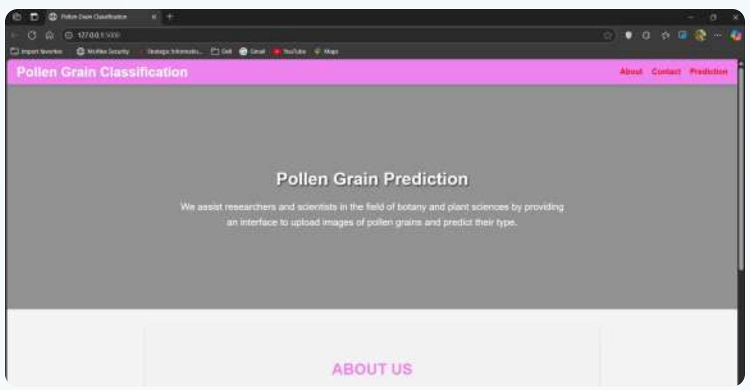


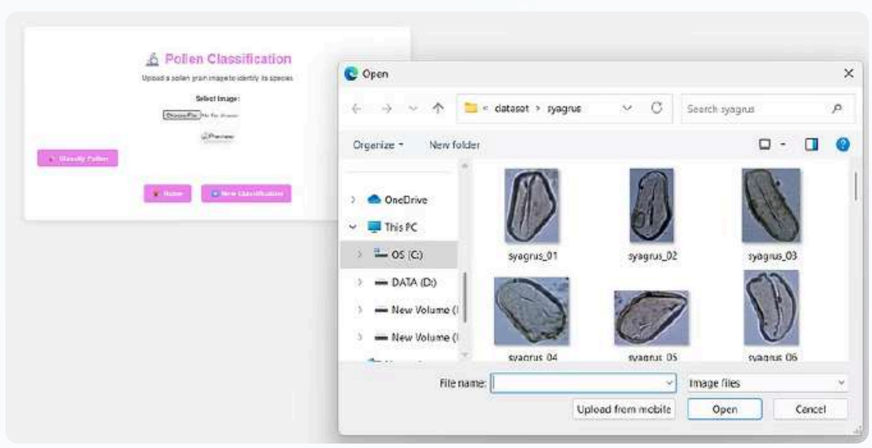
← classified output

← home page

prediction page

↓





Made with GAMMA

Key Observations and Conclusion

Key Observations

- **Balanced Training:** Implementing balanced training techniques significantly improved model stability and performance.
- **Overfitting Mitigation:** Data augmentation and dropout layers were highly effective in preventing overfitting, ensuring robust generalization.
- **Class Imbalance Handling:** The use of class weights successfully addressed and mitigated challenges posed by class imbalance in the dataset.

Project Conclusion

Automated Pollen Classifier

We successfully developed and implemented an automated system for pollen grain classification.

High Accuracy with CNN

Our Convolutional Neural Network (CNN) demonstrated exceptional accuracy on the unseen test set.

Real-time Application Potential

The model's performance suggests strong potential for extension into real-time applications using TFLite.

Made with GAMMA

Future Work and Deployment

Our successful classification model lays the groundwork for exciting future developments, focusing on deployment and enhanced functionality.

Flask Web App Deployment

Our immediate next step is to deploy the model as a user-friendly web application using Flask, enabling easy access and interaction.

Model Quantization for Edge Devices

We plan to quantize the model for efficient deployment on edge devices, enabling real-time, on-site pollen analysis.

Dataset Expansion

Expanding the dataset to include more pollen types will further enhance the model's robustness and applicability across diverse flora.

Microscopy Preprocessing Filters

Incorporating advanced microscopy preprocessing filters will improve image quality and, consequently, classification accuracy.

Made with GAMMA