Flower Image Classification Project Report

BY: Ori Yair Yaakov 207723198, Shlomi Asraf 207970252

April 27, 2025

Abstract

This report presents our work on classifying flower images into five categories: Daisy, Dandelion, Rose, Sunflower, and Tulip. The project explores several classical machine learning models: K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machine (SVM), and Naive Bayes. The process includes feature extraction from grayscale images, dimensionality reduction (PCA), and feature selection based on mutual information scores. Each model is evaluated based on classification performance, with results demonstrated through statistical metrics and visual analysis.

1 Introduction

1.1 Project Goal

The goal of this project is to classify flower images into one of five categories: Daisy, Dandelion, Rose, Sunflower, or Tulip. The classification is performed using classical machine learning methods. Several models are developed and evaluated: K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machine (SVM), and Naive Bayes.

1.2 Dataset

The dataset for this project consists of grayscale images resized to a uniform resolution of 32×32 pixels. Each image is labeled according to its corresponding flower class: daisy, dandelion, rose, sunflower, or tulip. For feature extraction, the raw pixel intensities were used along with two additional engineered features: the mean pixel intensity and the standard deviation of pixel values for each image. Furthermore, to ensure fair model training and evaluation, the dataset was balanced so that each flower class contains an equal number of samples.

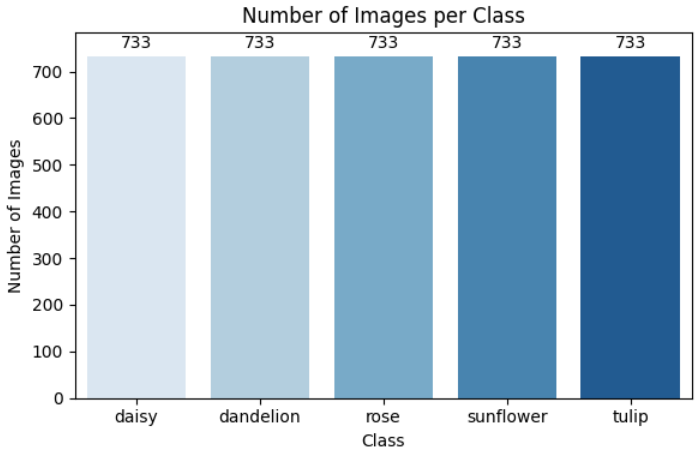

Figure 1: Graph of the Data Count per Class.

1.3 Data Preprocessing

The preprocessing phase involved preparing the raw images for machine learning. Each image was resized, converted to grayscale, and flattened into a feature vector. After appending the engineered mean and standard deviation features, all pixel values were normalized to the range [0,1]. Subsequently, feature standardization was applied to ensure zero mean and unit variance. Finally, the dataset was split into training and testing subsets using an 80%-20% split.

1.4 Feature Engineering

Following basic preprocessing, additional feature transformations were conducted to further enhance model performance. Principal Component Analysis (PCA) was applied to reduce the dimensionality of the feature space to 150 principal components, preserving the most significant variance in the data. Next, polynomial feature expansion of degree 2 was performed to capture potential interactions between features. Lastly, feature selection was applied using the SelectKBest method based on mutual information scores, selecting the top 100 most informative features for model training.

1.5 About The Models

This project explores different classical machine learning models for classifying flower images into five categories. We start with the K-Nearest Neighbors (KNN) algorithm, which classifies images based on the majority class of the nearest samples using the Manhattan distance metric. Logistic Regression is then applied, building on a linear decision boundary optimized via regularization techniques. Support Vector Machine (SVM) with an RBF kernel is used next, enabling the model to handle non-linear relationships in the data. Finally, a Naive Bayes classifier is employed, offering a simple yet efficient probabilistic approach for classification based on feature independence assumptions. In addition, a weighted soft-voting ensemble model is constructed to combine the strengths of all individual models and improve overall classification performance.

2 Exploratory Data Analysis (EDA)

In this section, we explore the structure and basic statistical properties of the dataset to gain insights before applying machine learning models.

2.1 Pixel Intensity Statistics

We begin by analyzing the distribution of pixel intensity statistics (mean and standard deviation) across different flower classes.

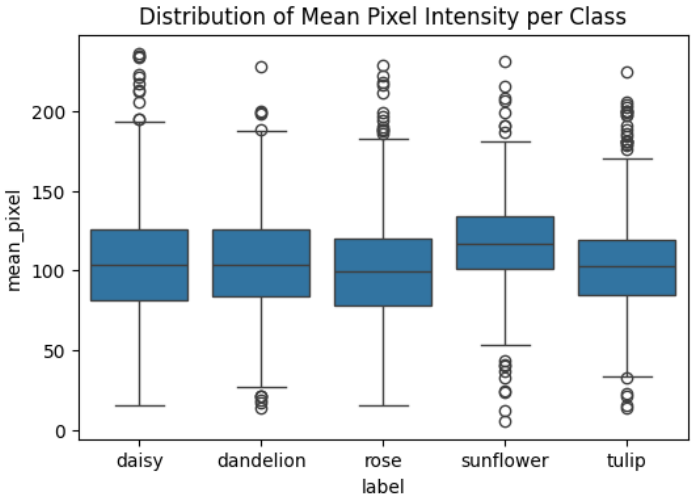

Figure 2: Distribution of Mean Pixel Intensity per Class.

As seen in the figure, sunflower images generally have slightly higher mean pixel intensity compared to other classes, suggesting that they tend to be brighter.

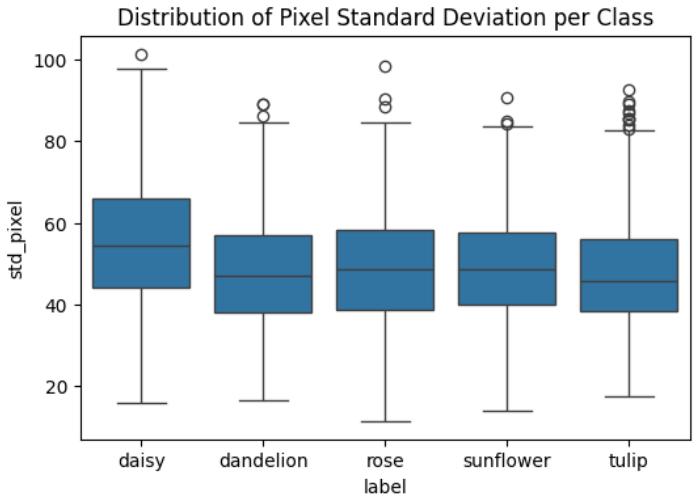

Figure 3: Distribution of Pixel Standard Deviation per Class.

The distribution of pixel standard deviations shows that daisy images exhibit more variation in pixel intensities, while rose and tulip classes have somewhat lower variability, potentially reflecting differences in texture complexity.

2.2 PCA Visualization

To gain a better understanding of the dataset's structure, Principal Component Analysis (PCA) was performed to project the data onto two dimensions.

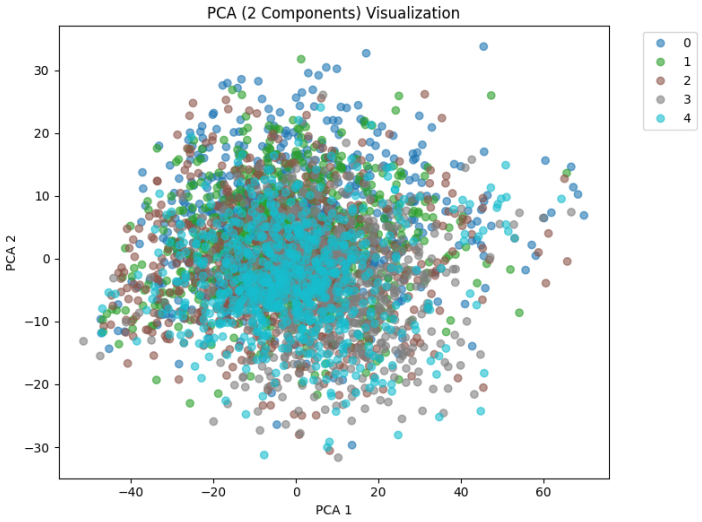

Figure 4: PCA 2D Visualization of the Dataset.

As can be seen, the samples overlap significantly across the two principal components, indicating the need for nonlinear feature engineering to better separate the classes.

2.3 Handling Nonlinear Complexity

The PCA visualization demonstrates that the flower classes are not linearly separable in the feature space. To address this nonlinear complexity, multiple advanced techniques were applied. Polynomial feature expansion of degree 2 was introduced to capture interactions between the original features, enabling models to learn nonlinear decision boundaries. Feature selection was then performed using the SelectKBest method based on mutual information scores, ensuring that only the most informative features were retained for training. Furthermore, Support Vector Machine (SVM) with an RBF kernel was utilized, allowing the model to project the data into a higher-dimensional space where a linear separation becomes possible. K-Nearest Neighbors (KNN) was also used with the Manhattan distance metric to better capture local neighborhood structures. Finally, a weighted ensemble voting classifier was built by combining the outputs of all models, leveraging the strengths of each to improve robustness and overall classification performance.

3 Feature Insights and Contribution Analysis

Beyond standard preprocessing, feature importance analysis provides deeper insights into the classification challenges.

To further enhance the classification performance, feature selection was performed using the SelectKBest method based on mutual information scores. Mutual information measures the dependency between each feature and the class label, allowing us to identify the most informative features for the classification task.

The top 10 features ranked by their mutual information scores are shown below:

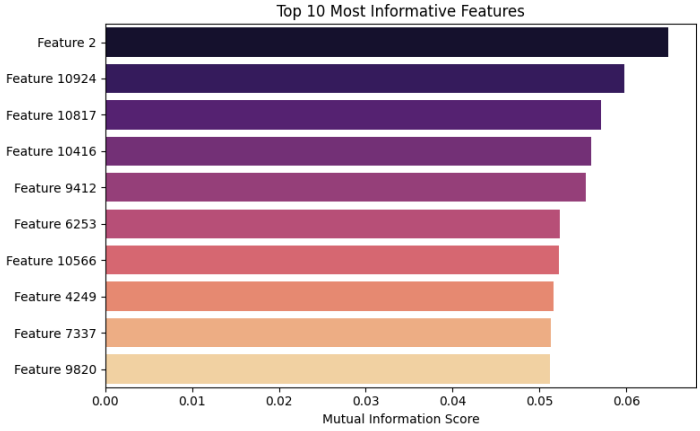

Figure 5: Top 10 Most Informative Features Based on Mutual Information.

3.1 Feature Distribution Analysis

To better understand the relevance of the selected features, we analyzed the distribution of several top-ranked features across the different flower classes.

Feature 2 shows class-dependent variations: *daisy* and *dandelion* have slightly higher median values compared to other classes, while *sunflower* and *tulip* show lower distributions. Although overlaps exist, this feature captures useful separation trends that contribute to classification.

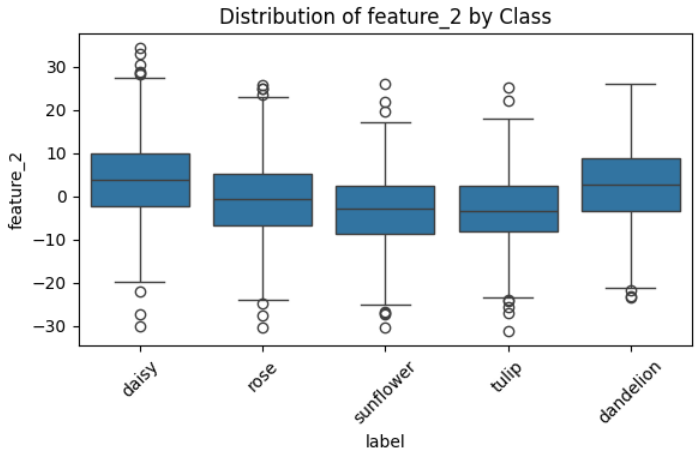

Figure 6: Distribution of Feature 2 by Flower Class.

Feature 297 exhibits a more compressed distribution across all classes. Most classes have feature values clustered near zero, with only minor variations. This suggests that while this feature was selected by the model, its individual discriminative power may be limited and it might be useful mainly through interactions with other features.

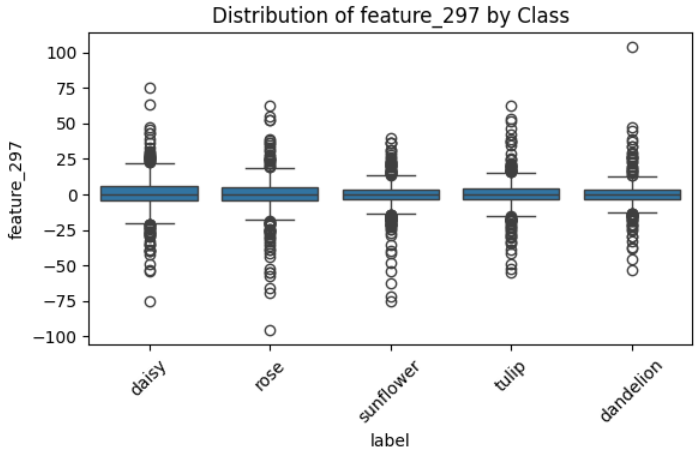

Figure 7: Distribution of Feature 297 by Flower Class.

Feature 347 behaves similarly to Feature 297, showing tightly centered distributions around zero for all classes, with a few outliers. Its selection highlights the model's ability to capture subtle, potentially nonlinear relationships between features that are not easily visible through direct analysis.

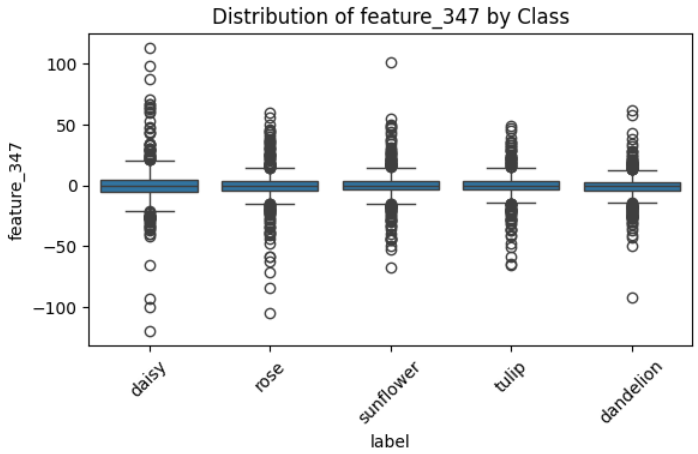

Figure 8: Distribution of Feature 347 by Flower Class.

This analysis emphasizes the importance of combining multiple features to achieve effective class separation, as some features individually offer limited discriminative power but contribute significantly when used together in nonlinear models.