Flower Recognition

Name	ID
Joud Ahmad Al-huthaly	444002970
Nehal Hamed Al-zahrani	444001073
Hanin Mesfer Al-Malki	444001268
Reem Ghazi Al-osaimi	444001268

DEPARTMENT OF (INFORMATION SCIENCE-DATA SCIENCE)
COLLEGE OF COMPUTER AND INFORMATION SYSTEMS UMM AL-QURA UNIVERSITY

Table of Contents

Links

- **Google Colab Notebook:** Flowers Recognition Model
- **Kaggle Dataset:** Flowers Recognition Dataset
- **Feedforward Neural Network (FNN) Model :** FNN Model

Abstract

This project focuses on the image classification of five distinct flower types: Daisy, Sunflower, Tulip, Dandelion, and Rose, using a dataset comprising 4,297 images. The primary goal was to develop effective models that accurately classify these flower types despite the inherent variability in the dataset, influenced by factors like lighting conditions, background distractions, and natural differences in flower characteristics. The dataset was preprocessed by resizing all images to 64 × 64 pixels to ensure uniformity, and data augmentation techniques, including rotation and flipping, were employed to enhance the training set and mitigate overfitting.

We developed a custom Convolutional Neural Network (CNN) and explored the use of Feedforward Neural Networks (FNNs) as baseline models, alongside Random Forest and Support Vector Machine (SVM) classifiers. The CNN architecture included convolutional, pooling, and dropout layers to effectively extract features and reduce overfitting. FNNs provided a simpler comparison to gauge the performance benefits of more complex models.

Project Objectives

1. **Image Classification**: Develop models to classify images of five flower types: Daisy, Sunflower, Tulip, Dandelion, and Rose, aiming for high accuracy.

2. **Model Development**: Build a custom Convolutional Neural Network (CNN) and explore Feedforward Neural Networks (FNNs) as a baseline, along with Random Forest and SVM for comparison.

3. **Performance Evaluation**: Measure model performance using metrics such as accuracy, precision, recall, and F1-score.

Introuduction

Image classification is a core task in computer vision, aiming to assign a label to an image based on its content. In this project, we focus on classifying images of five distinct flower types: Daisy, Sunflower, Tulip, Dandelion, and Rose. The challenge in this classification task lies in the inherent variability among the images, influenced by factors such as lighting conditions, background distractions, and the flowers' natural differences in shape and color. This variability can lead to overlapping features, making it difficult for models to distinguish between similar classes.

Dataset Overview The dataset used for this classification task is well-known in the computer vision community and contains a balanced set of images for each flower type. Each image has been resized to dimensions of 64×64 pixels, ensuring a uniform input size for the convolutional neural networks (CNNs) and feedforward models employed in the study. The balanced representation of each class helps prevent biases during training, thereby enhancing the model's ability to generalize.

Methodology

Data Preprocessing To prepare the dataset for effective model training, several preprocessing steps were undertaken:

Image Resizing: All images were resized to 128x128 pixels to standardize pixel values. This step is crucial for improving model convergence and ensuring consistent input dimensions across the models.

Data Augmentation: Various augmentation techniques, including rotation and flipping, were applied to increase the diversity of the training dataset. This helps the model learn more robust features and reduces the risk of overfitting.

Dataset Optimization: After augmentation, a total of 12,000 images were removed from the dataset. This optimization aimed to enhance the model's performance and speed, ensuring more efficient operation during training and evaluation.

Data Exploration and Description

Overview The dataset used for this flower classification project comprises images of five distinct flower types, totaling 4,297 images. Each flower type is represented by a varying number of images, ensuring a balanced dataset for effective model training.

Class Details

1. **Total Classes**: 5
    - Daisy
    - Rose
    - Tulip
    - Sunflower
    - Dandelion
2. **Total Images**: 4,297
3. **Image Distribution by Class**:
    - **Daisy**: 764 images
    - **Rose**: 784 images
    - **Tulip**: 984 images
    - **Sunflower**: 733 images
    - **Dandelion**: 1,052 images

Image Characteristics

- **Image Format**: All images are in JPG format, which is widely used for photographs and ensures good quality with manageable file sizes.
- **Image Dimensions**: Each image has been resized to 128 × 128 pixels to standardize input for the models, enabling efficient training and faster processing times.

Variability Challenges

1. **Lighting Conditions**: Images can vary in brightness and contrast, making it difficult for models to generalize.
2. **Background Distractions**: Different backgrounds can obscure flower features and create noise in the dataset.
3. **Natural Differences**: Variations in flower size, shape, and color can lead to overlapping features among different classes, complicating classification.

Preprocessing Challenges

1. **Noise**: Images may contain irrelevant information or artifacts that can interfere with model learning.
2. **Missing Data**: Incomplete datasets can lead to poor model performance, particularly if certain classes have fewer images

Methodology

Data Preprocessing Techniques

Data preprocessing is crucial for preparing the flower dataset for effective model training. The following techniques were employed:

1. **Image Resizing**: All images were resized to **128 × 128 pixels** to standardize input dimensions across models.

2. **Data Augmentation**: To increase dataset variability and reduce overfitting, several augmentation techniques were applied, including:

   - **Rotation**: Randomly rotating images within a specified degree range.
   - **Flipping**: Horizontally flipping images to introduce variability.

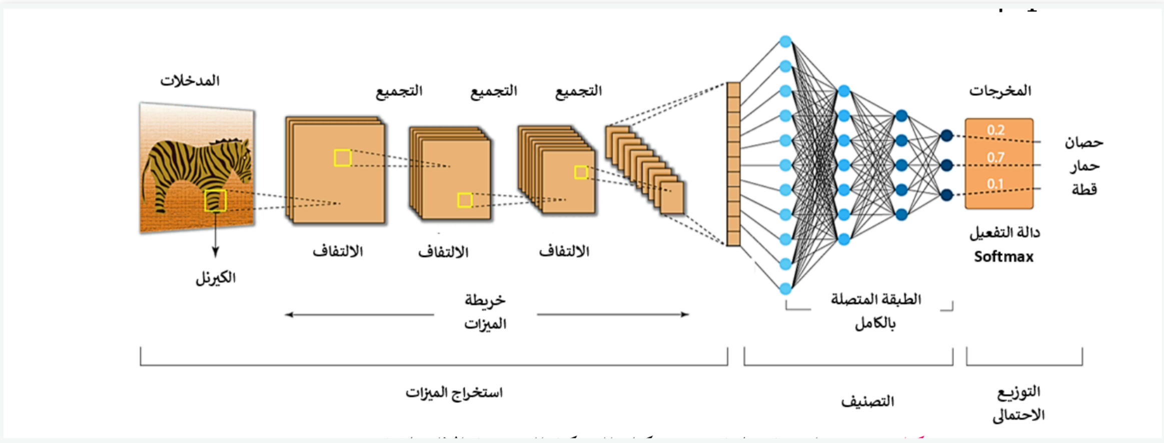
Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of deep learning model widely used in image processing and computer vision. They apply weights and biases to discern important features within images, enabling effective recognition and classification of different objects. CNNs fall under supervised learning, requiring labeled data for optimal training.

While CNNs excel in image handling, their applications extend to other data types, such as numerical data, where one-dimensional convolutions are preferred for sequential or time-series data. This versatility underscores CNNs' capability to extract features and enhance performance across various tasks.

Layers in CNNs

- **Input Layer**
- **Convolutional Layer**
- **Activation Layer**
- **Pooling Layer**
- **Dropout Layer:**
- **Fully Connected Layer**
- **Output Layer**

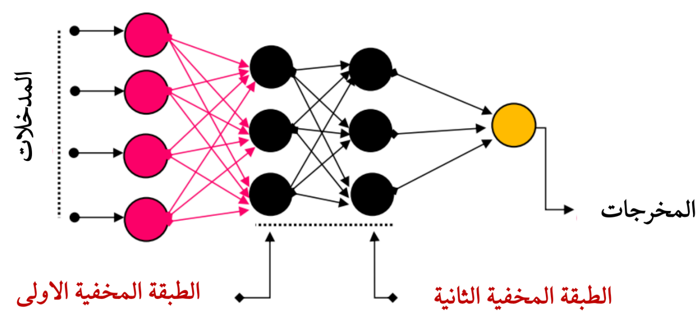


Feedforward Neural Network (FNN)

Feedforward neural networks are a basic type of artificial intelligence used in machine learning. In these networks, information flows in one direction—from the input layer through hidden layers to the output layer—without any feedback.

Each layer consists of neurons that process data by calculating a weighted sum and applying a function to create outputs.Layers in FNN

- **Input Layer**
- **Dense Layers (Hidden Layers)**
- **Dropout Layers**
- **Output Layer**



الشكل 3_2. شبكة عصبية ذات طبقتين مخفيتين

Random Forest and Its Use in Image Data

Random forest is a machine learning algorithm commonly used in various domains, including image processing. While not as typical for direct image classification as Convolutional Neural Networks (CNNs), random forests can still be effective in certain scenarios involving image data.

1. **Feature Extraction**: Before applying random forests, relevant features are extracted from images, such as color distributions, texture patterns, and shapes.

2. **Data Representation**: Each image is represented as a set of features, allowing the random forest to treat it like any other dataset.

3. **Classification**: The random forest model uses multiple decision trees to classify images based on the extracted features.

While random forests may not directly process raw pixel data like CNNs, they can be valuable in tasks like image segmentation, where features can help distinguish between different regions or objects in an image.

SVM (Support Vector Machine)

Support Vector Machine (SVM) is a powerful tool in machine learning, effective for various tasks such as classifying images, finding patterns, and identifying unusual data points.

How SVM Works in Image Processing

- **Finding the Best Separator**:
  SVM excels at identifying the optimal hyperplane that separates different classes of images. The goal is to maximize the margin, ensuring a wide gap between these classes for better classification.

- **Versatility**:
  SVM is highly flexible, capable of handling both simple (binary) and complex (multiclass) classification tasks. This adaptability makes it suitable for a wide range of applications.

- **Effective with Small Datasets**:
  SVM performs particularly well with small datasets, making it a strong choice compared to other methods like deep learning, which typically require larger datasets to achieve good performance.

Evaluation Metrics

Model performance was evaluated using the following metrics:

- **Accuracy**: The proportion of correctly classified images.
- **Confusion Matrix**: To visualize the performance across different flower classes.
- **Classification Report**: This included precision, recall, and F1-score for a comprehensive assessment of model performance.

Hyperparameters tuning CNN

- **Learning Rate**: Set to **0.001** for optimal convergence.

- **Image Resizing**: The target size of all images was changed from 128 × 128 pixels to 64 × 64 pixels to improve training speed and model efficiency.
- **Batch Size**: **64** images per batch to balance training speed and memory usage.
- **Epochs**: Models were trained for **10 epochs**, with **early stopping** criteria based on validation loss to prevent overfitting.
- **Training and Validation Split**: The dataset was divided into **80% training** and **20% validation** sets to monitor performance during training.

Hyperparameters tuning FNN

- **Learning Rate**: Set to **0.001** for optimal convergence.
- **Image Resizing**: The target size of all images was changed from **128 × 128** pixels to **120 × 120** pixels to improve training speed and model efficiency.
- **Batch Size**: **64** images per batch to balance training speed and memory usage.
- **Epochs**: Models were trained for **15 epochs**, with **early stopping** criteria based on validation loss to prevent overfitting.
- **Training and Validation Split**: The dataset was divided into **80% training** and **20% validation** sets to monitor performance during training.

Justification of Model Choices

The selection of models for our flower classification project was primarily driven by the high variability within the dataset.

Convolutional Neural Networks (CNNs) are particularly effective for image classification tasks due to their unique architecture. They excel at learning spatial hierarchies and extracting intricate features from images. This capability is essential for distinguishing between the subtle differences in flower types, especially when images may vary significantly in lighting, angle, and background.

In addition, the inclusion of a **feedforward neural network (FNN)** served as a baseline model, providing valuable insights into the performance enhancements offered by more advanced architectures. While the FNN is simpler and less capable of handling the complexities of image data, it helped establish a reference point for evaluating the effectiveness of CNNs and pre-trained models.

This comparison underscored the advantages of using CNNs for our flower classification task, highlighting their superior ability to manage the intricacies inherent in image data.

Patterns Discovery and Results

Convolutional Neural Networks (CNNs)

Model Performance :

Overall Accuracy

- **Test Accuracy**: **27%**
- **Validation Accuracy**: **65.2%**
- The model's accuracy on validation data was **65.2%**, while it dropped to **27% on test** data.
- This significant decline suggests a potential **overfitting** issue, where the model struggles to generalize to new data.

Class-Level Performance Analysis

- **Daisy:** 14% accuracy (Weak performance)
- **Dandelion:** 32% accuracy (Decent performance)
- **Rose:** 14% accuracy and 5% recall (Weak performance)
- **Sunflower:** 13% accuracy (Complete failure to recognize)
- **Tulip:** 30% accuracy and 40% recall (Best performance)

The model performed relatively better with the "Dandelion" and "Tulip" classes. However, performance was very weak with classes like "Daisy," "Rose," and “Sunflower."

Loss and Accuracy Insights

- **Loss:** Training and validation loss decrease rapidly initially, then validation loss plateaus, indicating potential overfitting.
- **Accuracy:** Training and validation accuracy increase, but a gap develops, suggesting overfitting.

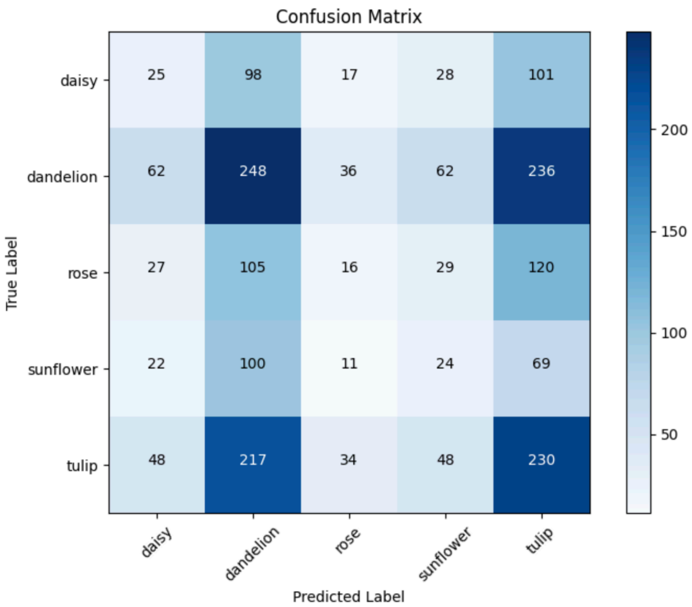
Additional Insights

- **The varying performance indicates a potential class imbalance or similarity in features among some classes, which may hinder the model's ability to differentiate between them.**


```
32/32 ━━━━━━━━━━━━━━━━━━━━ 13s 409ms/step
              precision    recall  f1-score   support

       daisy       0.14      0.09      0.11       269
   dandelion       0.32      0.39      0.35       644
        rose       0.14      0.05      0.08       297
   sunflower       0.13      0.11      0.12       226
       tulip       0.30      0.40      0.35       577

    accuracy                           0.27      2013
   macro avg       0.21      0.21      0.20      2013
weighted avg       0.24      0.27      0.25      2013
```



Feedforward Neural Network (FNN)

Model Performance :

1. Overall Accuracy

- **Test Accuracy**: 30%
- **Validation Accuracy**: 44%

2. Class-Level Performance Analysis

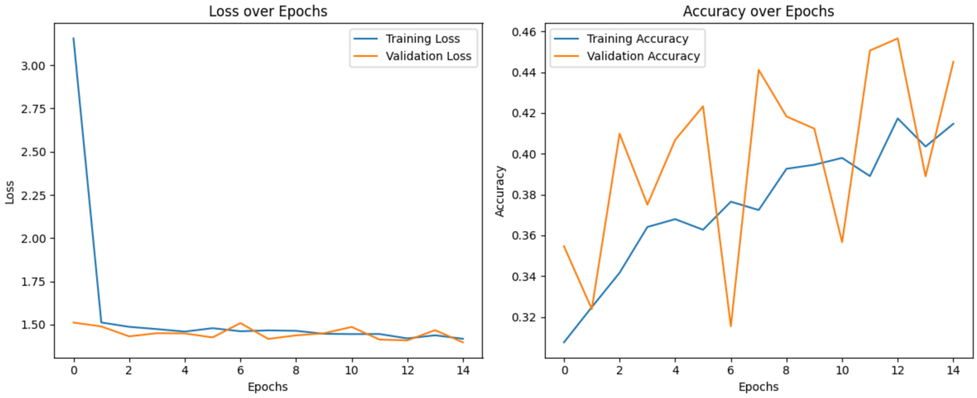
- **Daisy & Sunflower**:
  - **Metrics**: Accuracy, Recall, and F1-score: **0.0**
  - **Insight**: The model fails entirely to predict these classes, indicating a severe issue, possibly due to a lack of representative samples or ineffective feature extraction for these classes.
- **Dandelion & Tulip**:
  - **Performance**: Relative **performance is better**, but still low accuracy.
  - **Insight**: While the model shows some ability to classify these classes, the low accuracy suggests it may struggle with overlapping features or insufficient distinguishing characteristics.
- **Rose**:
  - **Recall**: Extremely low at **0.02**
  - **Insight**: This indicates that the model almost never correctly identifies the rose class. It suggests significant difficulties in classification, possibly due to a lack of distinct features or severe class imbalance.

- **Loss: Training loss decreases steadily; validation loss fluctuates, indicating potential overfitting.**
- **Accuracy: Training accuracy improves to 43%; validation accuracy is slightly lower and unstable.**

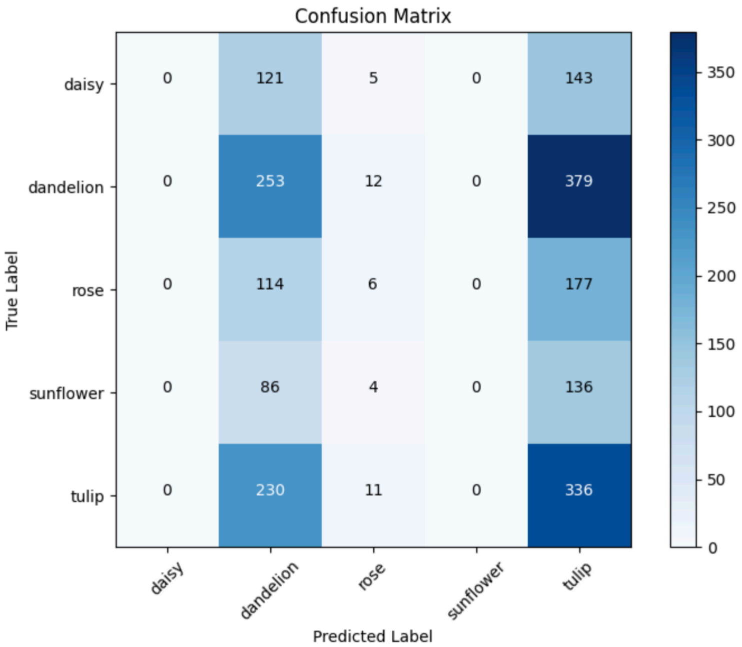
Additional InsightsInsight:

- **The model shows low accuracy across all datasets, indicating significant challenges in learning meaningful patterns from the data. This suggests potential issues such as insufficient features, inadequate training, or overfitting.**

```
Validation Loss: 1.397617220878601
Validation Accuracy: 0.4451068043708801
```



	precision	recall	f1-score	support
daisy	0.00	0.00	0.00	269
dandelion	0.31	0.39	0.35	644
rose	0.16	0.02	0.04	297
sunflower	0.00	0.00	0.00	226
tulip	0.29	0.58	0.38	577
accuracy			0.30	2013
macro avg	0.15	0.20	0.15	2013
eighted avg	0.21	0.30	0.23	2013



Random Forest

Overall Accuracy: The model achieved an overall accuracy of 61%

Performance Variance Across Classes:

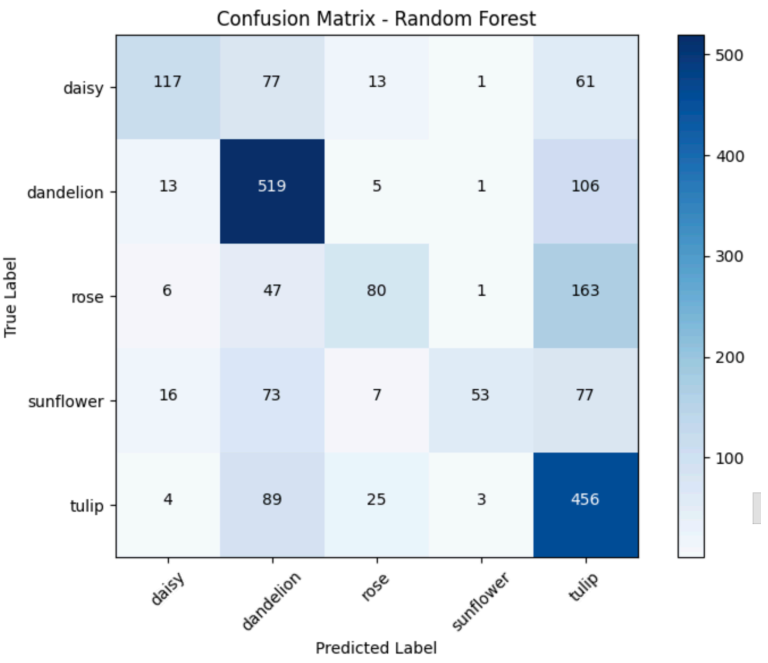
Model Performance by Class

- **Daisy:**
  - Precision: 0.75
  - Recall: 0.43
  - **Note:** The model struggles to recognize this class.
- **Dandelion:**
  - Precision: 0.64
  - Recall: 0.81
  - **Note:** The most balanced performance, indicating effective recognition.
- **Rose:**
  - Precision: 0.62
  - Recall: 0.27
  - **Note:** Overall poor performance, suggesting difficulty in distinguishing it from other classes.
- **Sunflower:**
  - Precision: 0.90
  - Recall: 0.23
  - **Note:** Good precision, but misses many instances.
- **Tulip:**
  - Precision: 0.53
  - Recall: 0.79
  - **Note:** Balanced performance, recognizing most instances but with less than ideal precision.

```
32/32 ━━━━━━━━━━━━━━━━━━━━ 117s 4s/step
              precision    recall  f1-score   support

           0       0.75      0.43      0.55       269
           1       0.64      0.81      0.72       644
           2       0.62      0.27      0.37       297
           3       0.90      0.23      0.37       226
           4       0.53      0.79      0.63       577

    accuracy                           0.61      2013
   macro avg       0.69      0.51      0.53      2013
weighted avg       0.65      0.61      0.58      2013
```



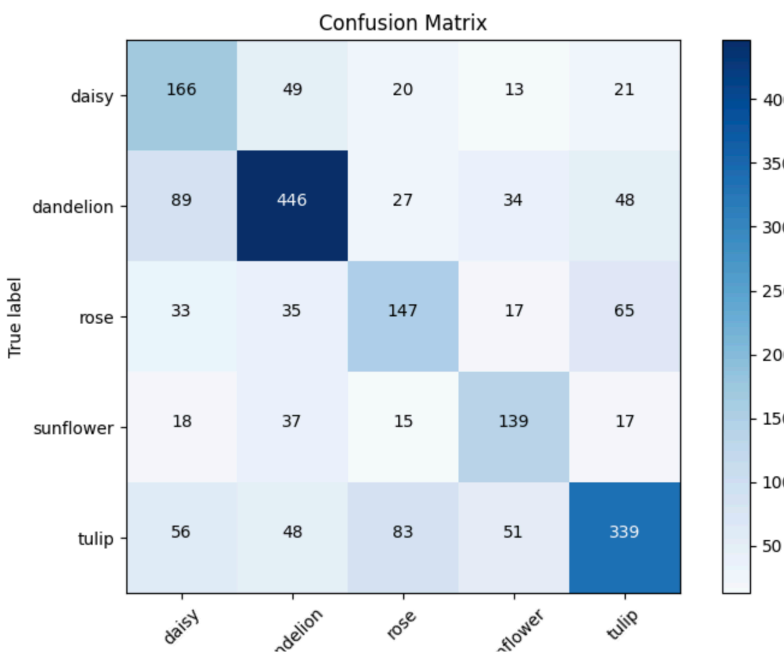
SVM (Support Vector Machine)

Based on the classification report, the model for classifying images into five flower categories achieved an overall accuracy of 61%, indicating challenges in distinguishing between some classes.

Model Performance Summary

- **Daisy:**
  - Precision: 0.46
  - Recall: 0.62
  - **Note:** Struggles to identify daisies but recognizes some instances.
- **Dandelion:**
  - Precision: 0.73
  - Recall: 0.69
  - **Note:** Performs well, accurately identifying most dandelions.
- **Rose:**
  - Precision: 0.50
  - Recall: 0.49
  - **Note:** Average performance, indicating difficulty in distinguishing roses.
- **Sunflower:**
  - Precision: 0.55
  - Recall: 0.62
  - **Note:** Decent performance, identifies many sunflowers but misclassifies some.
- **Tulip:**
  - Precision: 0.69
  - Recall: 0.59
  - **Note:** Strong precision but lower recall, missing some actual tulips.

	precision	recall	f1-score	support
daisy	0.46	0.62	0.53	269
dandelion	0.73	0.69	0.71	644
rose	0.50	0.49	0.50	297
sunflower	0.55	0.62	0.58	226
tulip	0.69	0.59	0.64	577
accuracy			0.61	2013
macro avg	0.59	0.60	0.59	2013
weighted avg	0.63	0.61	0.62	2013



Model Comparison for Image Classification

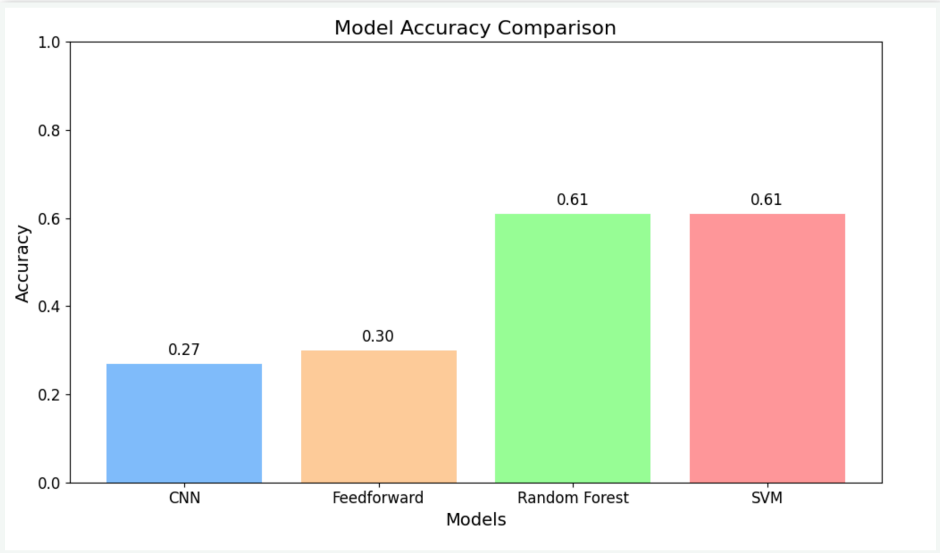
1. **CNN (Convolutional Neural Network)**
   Advantages:
   - **Good Feature Learning:** Automatically learns important features from images, making it effective for complex image classification.
   - **Robust Performance:** Higher validation accuracy compared to simpler models, indicating it can handle intricate patterns.
2. **Disadvantages:**
   - **Requires More Data:** Needs a larger dataset for training to perform optimally.
   - **Long Training Time:** Can take longer to train compared to simpler models; may require more epochs.
3. **Feedforward Neural Network (FFNN)**
   Advantages:
   - **Simplicity:** Easier to set up and understand.
   - **Less Computationally Intensive:** Requires fewer resources compared to CNNs, making it suitable for environments with limited computational power.
4. **Disadvantages:**
   - **Poor Performance:** Lower accuracy and higher loss, indicating it struggles with complex images.
   - **Limited Feature Learning:** Doesn't automatically learn features as effectively as CNNs.
5. **Random Forest**
   Advantages:
   - **Robustness to Overfitting:** Generally performs well without overfitting.
   - **Less Hyperparameter Tuning:** Requires less extensive hyperparameter tuning compared to neural networks, speeding up model development.
6. **Disadvantages:**
   - **Slower with Large Datasets:** Can be slower to predict with large datasets compared to deep learning methods.
7. **SVM (Support Vector Machine)**
   Advantages:
   - **Effective in High Dimensions:** Performs well with high-dimensional data, such as images.

- **Margin Maximization:** Finds the best boundary between classes, leading to robust classifications.

8. **Disadvantages:**
   - **Long Training Time:** Can be slow to train, especially with large datasets.

Overall Results

- **Best Performer:** Both Random Forest and SVM achieve similar accuracy (61%), but Random Forest offers better interpretability.
- **Simplicity vs. Performance:** The Feedforward Neural Network (FNN) shows lower performance (30%), making it less suitable for complex image tasks.
- **CNN Performance:** CNN shows poor performance at 27%.
- **Best Machine:** The machine learning model performed the best overall



Summary

For complex image classification tasks, CNN is the worst choice, while Random Forest and SVM provides a good balance of performance and interpretability. FNN is useful for simpler tasks or as a baseline.

Conclusion

In this project, we aimed to classify images of five flower types—Daisy, Sunflower, Tulip, Dandelion, and Rose—using a dataset of 4,297 images. Here are the key insights and challenges identified:

Summary of Findings

- **Model Performance:**

- **CNN:** Achieved a validation accuracy of 65.2% but dropped to 27% on test data, indicating potential overfitting.
- **FNN:** Showed low accuracy, around 30%.
- **Random Forest and SVM:** Both classifiers reached 61%, making them the best performers, although they still have limitations with complex image data

- **Class-Specific Performance:**
    - Dandelion and Tulip were easier to classify.
    - Daisy, Rose, and Sunflower had significantly lower recall rates, indicating difficulties due to overlapping features or class imbalance.

Challenges of High Variability

- **Lighting and Background Variability:** Changes in brightness and complexity hindered model generalization.
- **Feature Overlap:** Similarities among flower types complicated classification, resulting in lower accuracy for certain classes.

Potential Improvements and Future Work

- **Data Augmentation:** Use diverse techniques (e.g., scaling, color adjustments) to improve robustness and generalization.
- **Fine-Tuning Models:** Implement transfer learning with pre-trained models (e.g., VGG16, ResNet) to enhance accuracy.
- **Hyperparameter Optimization:** Conduct exhaustive searches for optimal hyperparameters (learning rates, batch sizes).
- **Class Weight Adjustment:** Modify class weights during training to address imbalances and improve focus on underrepresented classes.

In summary, while CNNs proved effective for flower classification, addressing challenges from high variability is crucial for improving accuracy and generalization. Future work should implement these enhancements for better classification outcomes.

Reference

[1]A. Shafi, “Sklearn Random Forest Classifiers in Python Tutorial,” www.datacamp.com, Feb. 2023. https://www.datacamp.com/tutorial/random-forests-classifier-python

[2]A. Sasidharan, “Support Vector Machine Algorithm,” GeeksforGeeks, Jan. 20, 2021. https://www.geeksforgeeks.org/support-vector-machine-algorithm/

[3]د. علاء طعيمة, “كتاب التعلم العميق : من الأساسيات حتى بناء شبكة عصبية عميقة بلغة بالعربي || الدكتور علاء طعيمة - موقع التعلم الآلي DL || البايثون,” التعلم العميق بالعربي والتعلم العميق وعلم البيانات باللغة العربية || د. علاء طعيمة, Dec. 08, 2022. https://dlarabic.com/%d9%83%d8%aa%d8%a7%d8%a8-%d8%a7%d9%84%d8%aa%d8%b9%d9%84%d9%85-%d8%a7%d9%84%d8%b9%d9%85%d9%8a%d9%82-%d9%85%d9%86-%d8%a7%d9%84%d8%a3%d8%b3%d8%a7%d8%b3%d9%8a%d8%a7%d8%aa-%d8%ad%d8%aa%d9%89-%d8%a8%d9%86/ (accessed Oct. 26, 2024).