# Flower Image Classification Using Convolutional Neural Networks (CNN)

## 1. Introduction

Flower image classification is a common computer vision task where the goal is to automatically identify the type of flower from an image.
In this project, a Convolutional Neural Network (CNN) is built using TensorFlow/Keras to classify images into three classes:

- Rose
- Dandelion
- Daisy

A simple GUI is also built using Gradio to allow users to upload images and view predictions interactively.

## 2. Dataset Description

The dataset contains three folders, each representing a flower category:

- Images/
    - daisy
    - dandelion
    - rose

Additionally, a separate Samples/ folder contains test images for manual evaluation.

### Dataset Statistics

A script is used to count how many images exist in each class.The total dataset contains approximately:

- *Daisy*: 764 images
- *Dandelion*: 1052 images
- *Rose*: 784 images
- Total Images: 2600

These images vary in size, color intensity, and background environment, making the classification task realistic.

## 3. Data Visualization

A few sample images from each class are displayed to visually understand the dataset.
This helps confirm that:

- Images are correctly labeled
- Flower categories are visually distinct
- No corrupted images exist

**4. Data Preprocessing**

To prepare the dataset for training:

4.1. Image Resizing

All images are resized to 150 × 150 pixels.

4.2. Normalization

Pixel values are scaled to a range of 0 to 1.

4.3. Train–Validation Split

A 20% validation split is applied using `ImageDataGenerator`.

4.4. Data Loading

`flow_from_directory()` is used to automatically:

- Load images
- Generate batches
- Apply preprocessing
- Assign labels based on folder names

**5. Model Architecture**

A Convolutional Neural Network (CNN) is designed with the following components:

**5.1. Convolution & Pooling Layers**

Multiple convolutional blocks extract spatial features such as:

- Edges
- Petal shapes
- Textures

Each block includes:

- Conv2D layer with ReLU activation
- MaxPooling2D for dimensionality reduction

### 5.2. Dropout Layers

Dropout is used to reduce overfitting and improve generalization.

### 5.3. Fully Connected Layers

Flattening is performed to convert feature maps into a vector.
Then:

- A dense layer with 128 neurons (ReLU)
- Output layer with 3 neurons and softmax activation

### 5.4. Compilation

The model is compiled using:

- Loss Function: Categorical Crossentropy
- Optimizer: Adam
- Metrics: Accuracy

## 6. Model Training

The model is trained for 10 epochs using:

- Training data
- Validation data to check generalization

During training, both accuracy and loss curves are plotted to observe model performance.

## 7. Model Evaluation

### 7.1. Confusion Matrix

A confusion matrix is generated to identify:

- Which classes are predicted correctly
- Which classes are being confused

This visual analysis helps understand model strengths and weaknesses.

### 7.2. Classification Report

A detailed report is produced containing:

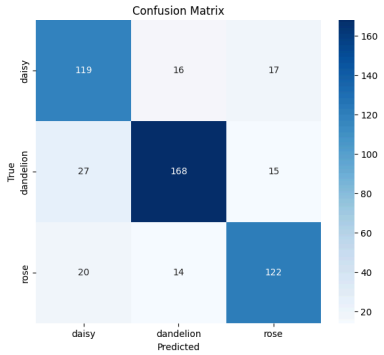
- Precision

- Recall
- F1-score
- Support (samples per class)

The model achieves approximately:

- Overall Accuracy: ~79% on validation data

This indicates strong performance for a basic CNN on a small dataset.




Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| daisy | 0.72 | 0.78 | 0.75 | 152 |
| dandelion | 0.85 | 0.80 | 0.82 | 210 |
| rose | 0.79 | 0.78 | 0.79 | 156 |

```
    accuracy                           0.79       518
   macro avg       0.79      0.79      0.79       518
weighted avg       0.79      0.79      0.79       518
```

**8. Model Saving**

The trained model is saved as:

`flower_model.h5`

This allows future use without retraining.

**9. Gradio Web Application**

A simple interactive user interface is created using Gradio.

Features:

- Upload any flower image
- Model predicts probabilities for all 3 classes
- Displays the top predictions with confidence scores
- Easy to use for non-technical users

This makes the project practical and accessible.

**10. Conclusion**

The project successfully builds a functional flower image classification system using CNNs.
Key outcomes:

- Accurate classification of rose, daisy, and dandelion images
- Strong baseline performance with validation accuracy around 79%
- Interactive prediction system using Gradio
- Detailed evaluation using confusion matrix and classification reports

The model can be further improved using techniques such as:

- Transfer Learning (MobileNet, VGG16, ResNet)
- Advanced data augmentation
- Hyperparameter tuning

**11. Future Enhancements**

Potential improvements include:

- Achieving >90% accuracy using pre-trained models
- Deploying the model as a web app or mobile app
- Adding more flower categories
- Creating a real-time camera-based flower detection system