Project Report: Flower Species

Introduction:

The objective of this project is to identify species of flowers from images using both traditional machine learning techniques and deep learning approaches. The project involves preprocessing the images, feature extraction using SIFT for traditional methods, and implementing a Convolutional Neural Network (CNN) for deep learning. The dataset consists of flower images categorized into five classes: daisy, dandelion, rose, sunflower, and tulip.

Data Collection and Preprocessing:

1. **Dataset Source:**
   The dataset was sourced from Kaggle, specifically the "flowers-recognition" dataset, which contains over 7000 images categorized into five classes.

2. **Preprocessing Techniques:**
   - Resizing: Images were resized to 128x128 pixels to ensure uniformity.
   - Normalization: Pixel values were scaled to the range [0, 1] to standardize the data.
   - Data Augmentation: Techniques such as rotation, width and height shifts, shear, zoom, and horizontal flips were applied to increase the diversity of the training data and improve model generalization.

3. **Ethical Considerations:**
   The dataset is publicly available and does not contain personally identifiable information or sensitive data, mitigating privacy concerns.

4. **Feature Extraction and Traditional Machine Learning**

   Feature Extraction with SIFT:

   SIFT (Scale-Invariant Feature Transform) was chosen for its robustness to changes in scale, rotation, and illumination. Keypoints and descriptors were extracted from each image and clustered using k-means to create a visual vocabulary.

5. Bag-of-Words Representation:

A visual vocabulary size of 100 clusters was determined through iterative testing. Histograms of visual words were constructed for each image to ensure consistent feature representation.

6. Machine Learning Classifiers:

A Support Vector Machine (SVM) with an RBF kernel was selected due to its effectiveness in handling high-dimensional feature spaces and non-linear classification tasks.

Deep Learning with CNN

CNN Architecture Design:

The CNN model was designed with the following architecture:

- Convolutional Layers: Four convolutional layers with filter sizes increasing from 32 to 256, using ReLU activation functions.

- Pooling Layers: Max-pooling layers with a pool size of 2x2 and a stride of 2.

- Fully Connected Layers: Two fully connected layers with 256 and 128 neurons, followed by a softmax layer for classification.

- Regularization: Dropout layers with a rate of 0.5 were used to prevent overfitting.

Training Process:

- Learning Rate: A learning rate of 0.001 was used with the Adam optimizer.

- Batch Size: A batch size of 32 was chosen based on experimentation.

- Epochs: The model was trained for 50 epochs, with early stopping monitored using validation loss.

Data Handling:

- Data Augmentation: Applied as mentioned in preprocessing to improve model generalization.

- Normalization: Scaling pixel values to the range [0, 1].

Model Evaluation

Evaluation Metrics:

The models were evaluated using accuracy, precision, recall, and F1-score. Confusion matrices were also generated to visualize the performance across different classes.

SVM Model Performance:

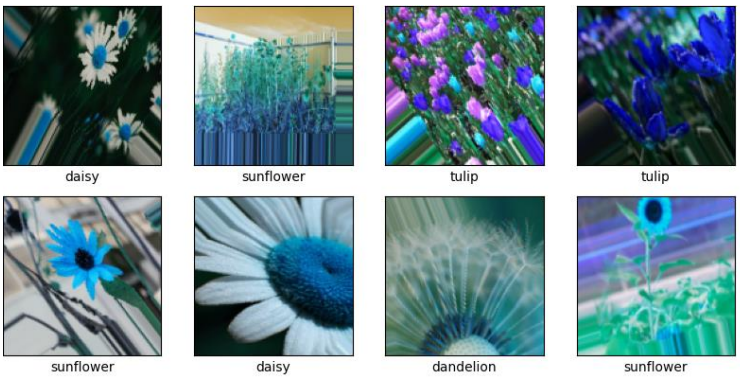
- Accuracy: 54%
- Confusion Matrix:
  - Daisy: Precision 0.76, Recall 0.49, F1-score 0.60
  - Dandelion: Precision 0.55, Recall 0.68, F1-score 0.60
  - Rose: Precision 0.52, Recall 0.49, F1-score 0.50
  - Sunflower: Precision 0.54, Recall 0.51, F1-score 0.52
  - Tulip: Precision 0.45, Recall 0.51, F1-score 0.48

CNN Model Performance:

- Accuracy: 43%
- Confusion Matrix:
  - Daisy: Precision 0.28, Recall 0.81, F1-score 0.42
  - Dandelion: Precision 0.62, Recall 0.44, F1-score 0.51
  - Rose: Precision 0.54, Recall 0.21, F1-score 0.30
  - Sunflower: Precision 0.55, Recall 0.56, F1-score 0.56
  - Tulip: Precision 0.57, Recall 0.20, F1-score 0.30

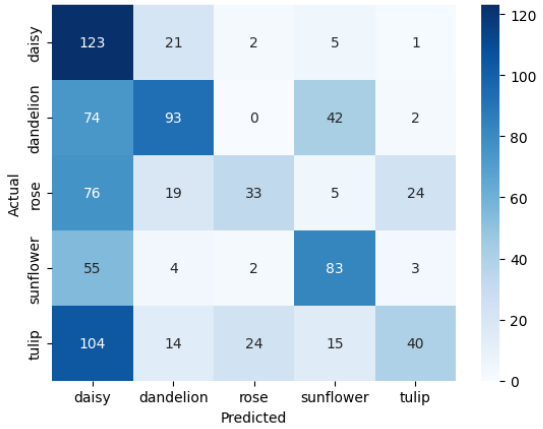
Sample Images:

Training Images:



Prediction Results:

	precision	recall	f1-score	support
daisy	0.28	0.81	0.42	152
dandelion	0.62	0.44	0.51	211
rose	0.54	0.21	0.30	157
sunflower	0.55	0.56	0.56	147
tulip	0.57	0.20	0.30	197
accuracy			0.43	864
macro avg	0.51	0.45	0.42	864
weighted avg	0.52	0.43	0.42	864



Recommendations for Future Work:

1. Enhance Data Augmentation: To further improve model generalization.

2. Hyperparameter Tuning: More extensive tuning for both SVM and CNN models.

3. Hybrid Model: Combine SVM and CNN approaches to balance performance and efficiency.

4. Transfer Learning: Incorporate pre-trained models for potentially higher accuracy.