**Flower Classification Using CNN - Final Evaluation Report**

**1. Objective** The goal of this project is to classify images of flowers into five categories: **Daisy**, **Dandelion**, **Rose**, **Sunflower**, and **Tulip** using a **Convolutional Neural Network (CNN)**.

**2. Dataset Summary**

* Source: Image dataset containing **4317 labeled images** from 5 flower classes.
* Image Size: Resized to **128x128** for uniformity.
* Split: **80% training**, **20% testing**.
* Normalization: Pixel values scaled to [0, 1].

**3. CNN Architecture Used**

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(128,128,3)),

MaxPooling2D(pool\_size=(2, 2)),

Conv2D(64, (3, 3), activation='relu'),

MaxPooling2D(pool\_size=(2, 2)),

Conv2D(128, (3, 3), activation='relu'),

MaxPooling2D(pool\_size=(2, 2)),

GlobalAveragePooling2D(),

Dense(128, activation='relu'),

Dropout(0.5),

Dense(5, activation='softmax')

])

* Optimizer: Adam(learning\_rate=0.001)
* Loss: categorical\_crossentropy
* Metrics: accuracy

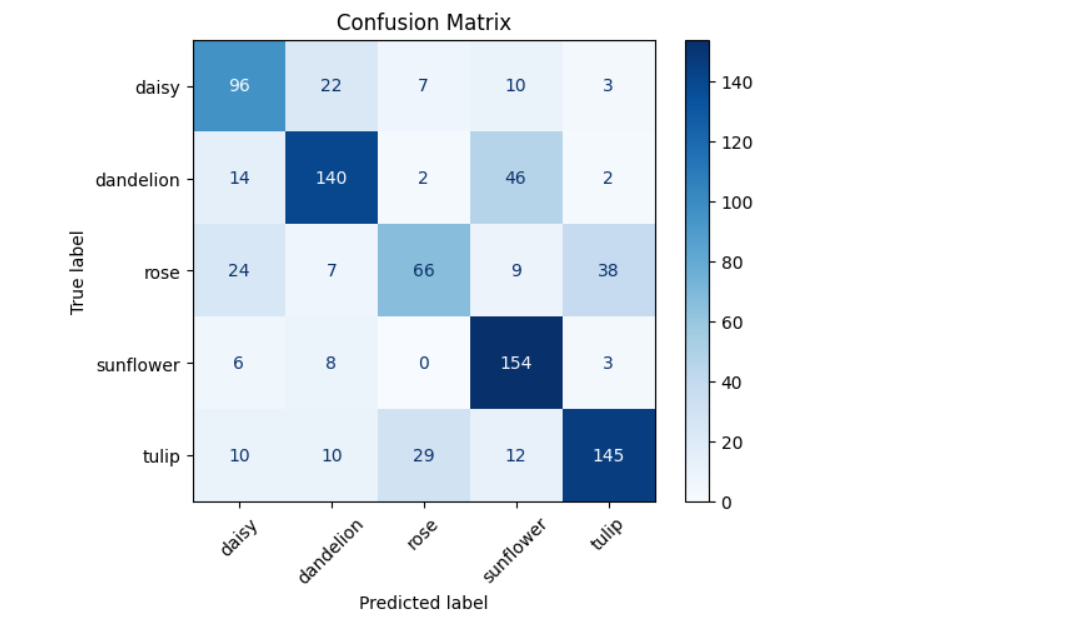
**4. Evaluation Metrics (after 20 epochs)**

* **Training Accuracy**: 69.70%
* **Validation Accuracy**: **69.64%**
* **Training Loss**: 0.7972
* **Validation Loss**: 0.7864

**5. Classification Report (Test Set)**

| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| Daisy | 0.64 | 0.70 | 0.67 | 138 |
| Dandelion | 0.75 | 0.69 | 0.72 | 204 |
| Rose | 0.63 | 0.46 | 0.53 | 144 |
| Sunflower | 0.67 | 0.90 | 0.77 | 171 |
| Tulip | 0.76 | 0.70 | 0.73 | 206 |
| **Accuracy** |  |  | **0.70** | **863** |
| **Macro Avg** | 0.69 | 0.69 | 0.68 | 863 |
| **Weighted Avg** | 0.70 | 0.70 | 0.69 | 863 |

**6. Confusion Matrix Insights**

****

* **Rose** is often confused with **Tulip**, reducing its recall.
* **Sunflower** is classified very accurately with high recall.
* **Tulip** and **Dandelion** have solid precision and recall.

**7. Challenges Faced**

* **Visual Similarity**: High misclassification between Rose and Tulip due to overlapping visual features.
* **Overfitting Risk**: Managed using dropout and image augmentation.

**8. Why CNN is Better Than Standard Neural Networks**

| **Feature** | **Standard Neural Net** | **CNN** |
| --- | --- | --- |
| Input Handling | Requires flattening | Preserves 2D image structure |
| Spatial Awareness | None | High (detects local patterns) |
| Parameters | Very high | Efficient (shared filters) |
| Translation Invariance | No | Yes (via pooling layers) |
| Feature Learning | No hierarchy | Learns features hierarchically |

**End of Report**