1. Knowing which machine learning methods you use, describe possible testing metrics and procedures (accuracy, precision, recall, sensitivity, F-score, etc.)

Accuracy- The ratio of correctly predicted instances to the total instances.

Precision- The ratio of correctly predicted positive instances to the total predicted positives.

Sensitivity- The ratio of correctly predicted positive instances to all actual positives.

F-score - The harmonic mean of precision and recall.

Specificity – The ratio of correctly predicted negative instances to all actual negatives.

Confusion matrix - shows how well the model works for each class. It shows how many instances were correctly classified and how many were misclassified.

2. Choose <u>few</u> (more than one) methods/metrics which are suitable to your problem. Justify why.

Accuracy - Tracking accuracy lets us monitor our model behavior over time, thus lets us choose the best model for our task.

Confusion matrix - shows how well the model works for each class. It shows how many instances were correctly classified and how many were misclassified.

Precision - the ratio of correctly predicted positive observations to the total predicted positives. It indicates how many of the predicted positive instances were actually correct.

Recall - the ratio of correctly predicted positive observations to all observations in the actual class. It measures how well the model can identify all relevant instances of a class.

$$\text{precision} = \frac{tp}{tp + fp},$$

$$\text{recall} = \frac{tp}{tp + fn},$$

$$precision = \frac{correct}{all\ returned}$$

$$recall = \frac{correct}{all\ relevant}$$

F1-score - the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall, making it useful when you need to account for both false positives and false negatives.

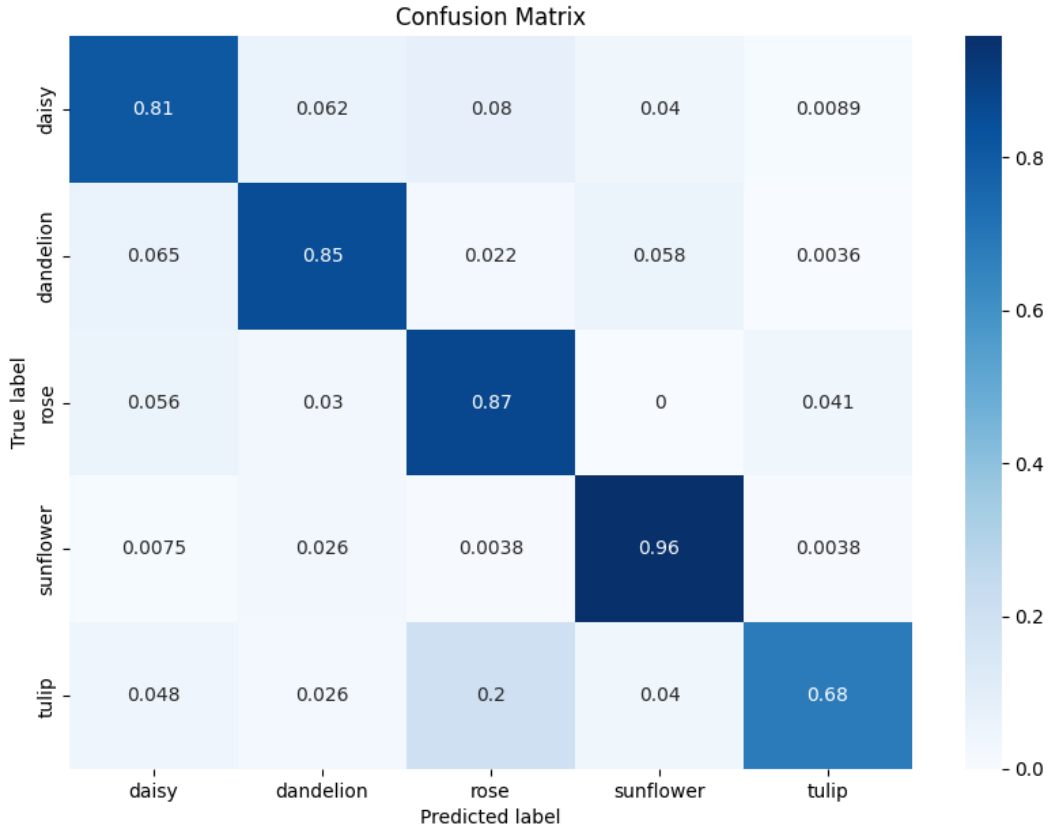
$$F_\beta = (1 + \beta^2)\frac{\text{precision} \times \text{recall}}{\beta^2\text{precision} + \text{recall}}$$

Support - refers to the number of actual occurrences of each class in the dataset. It indicates how many instances of each class are present.

3. Test your models created in Task 3 (MODEL1, MODEL2, MODEL3) using all defined measures. For each test, make some useful plots and result visualizations. [10 points]

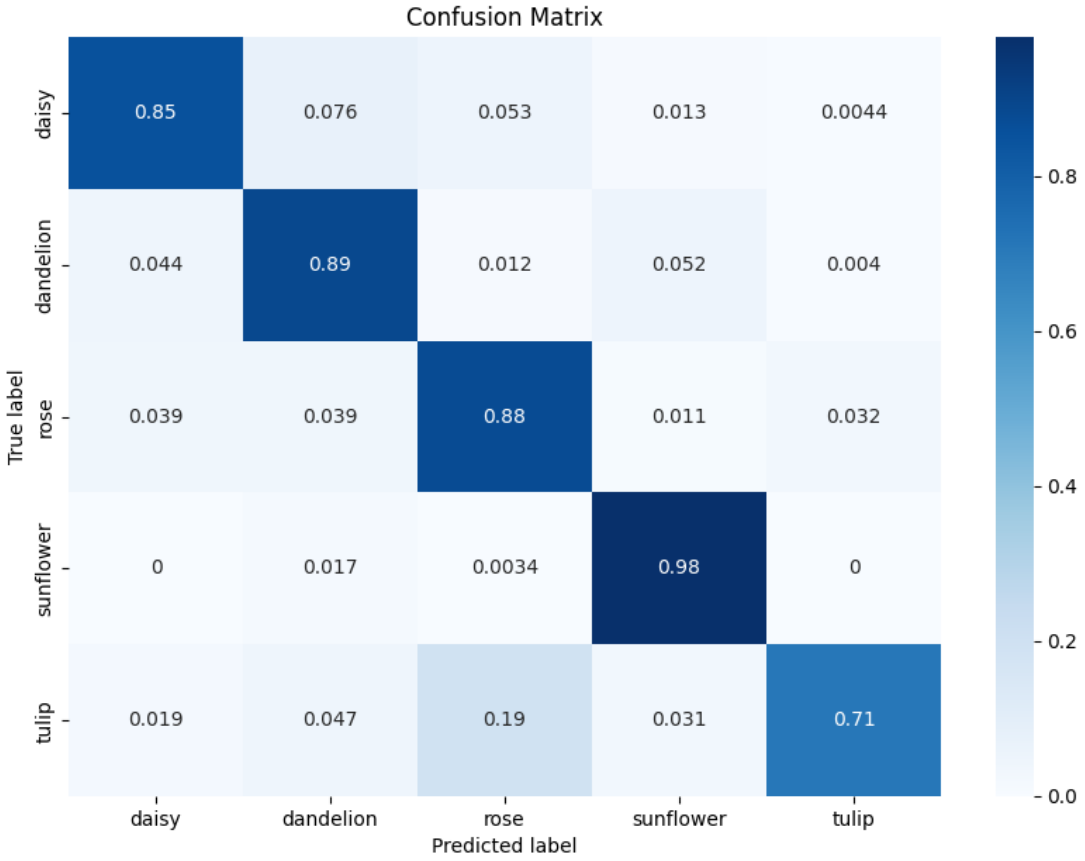
MODEL1

Test on TRAIN subsets



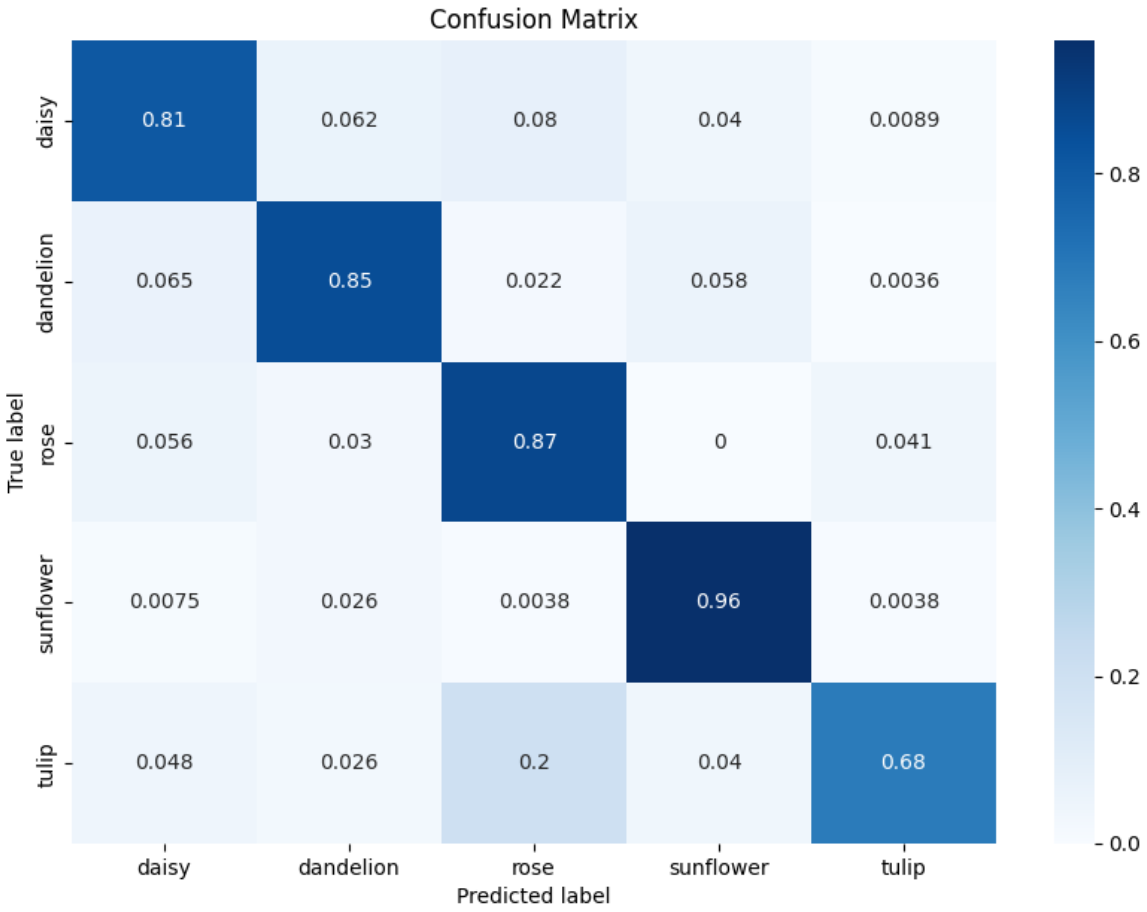
	precision	recall	f1-score	support
daisy	0.79	0.81	0.80	225
dandelion	0.87	0.85	0.86	276
rose	0.75	0.87	0.80	269
sunflower	0.88	0.96	0.92	265
tulip	0.93	0.68	0.79	273
accuracy			0.84	1308
macro avg	0.84	0.84	0.83	1308
weighted avg	0.84	0.84	0.83	1308

Test on VAL subsets



	precision	recall	f1-score	support
daisy	0.88	0.85	0.86	225
dandelion	0.83	0.89	0.86	251
rose	0.79	0.88	0.83	282
sunflower	0.91	0.98	0.95	292
tulip	0.94	0.71	0.81	258
accuracy			0.87	1308
macro avg	0.87	0.86	0.86	1308
weighted avg	0.87	0.87	0.86	1308

Test on TEST subsets



	precision	recall	f1-score	support
daisy	0.79	0.81	0.80	225
dandelion	0.87	0.85	0.86	276
rose	0.75	0.87	0.80	269
sunflower	0.88	0.96	0.92	265
tulip	0.93	0.68	0.79	273
accuracy			0.84	1308
macro avg	0.84	0.84	0.83	1308
weighted avg	0.84	0.84	0.83	1308

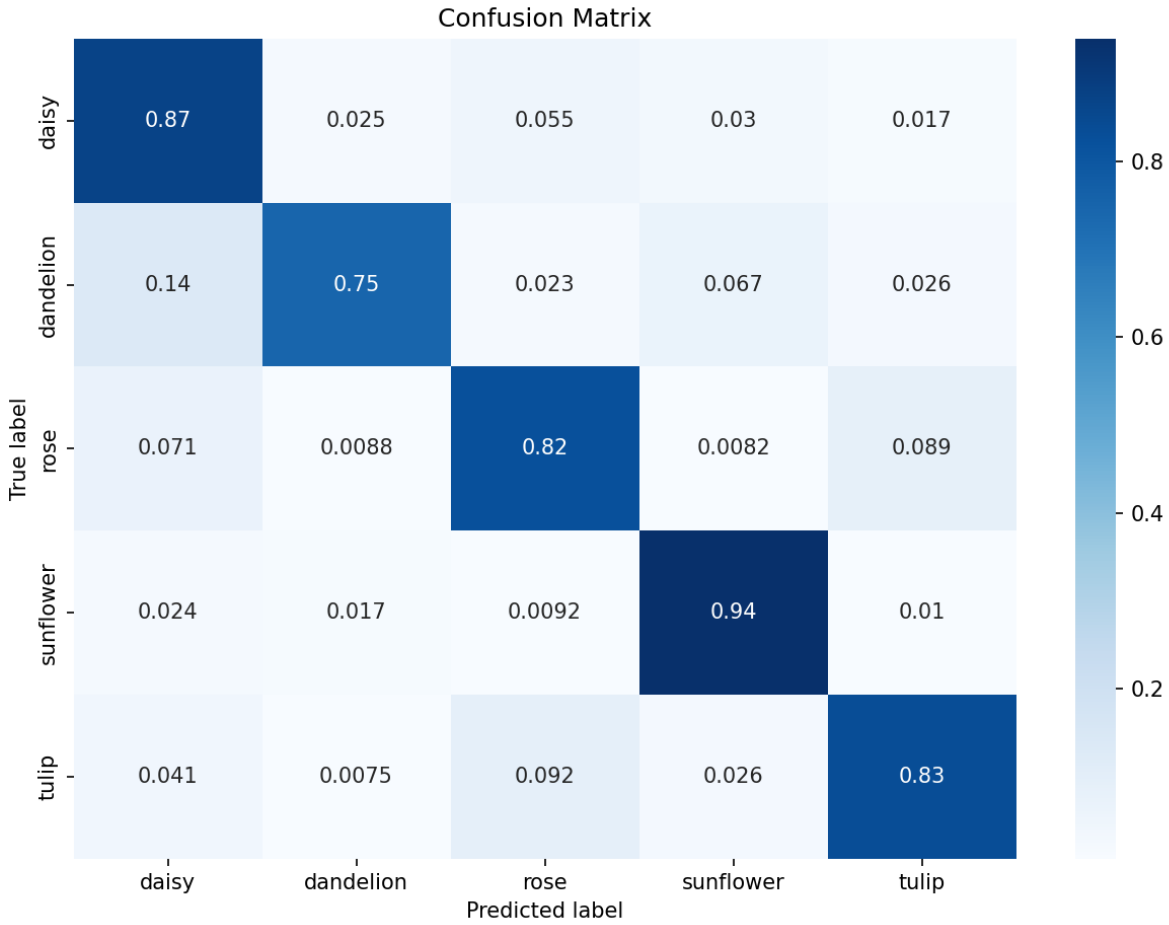
Discuss the results.

Biggest problem: Model often mistake roses for tulips.

The model performs well, with an accuracy of around 84-87%. It is good at classifying sunflowers but struggles with tulips. Precision and recall are high for most classes, indicating good generalisation. Further improvements can be made through data augmentation for tulips and hyperparameter tuning.

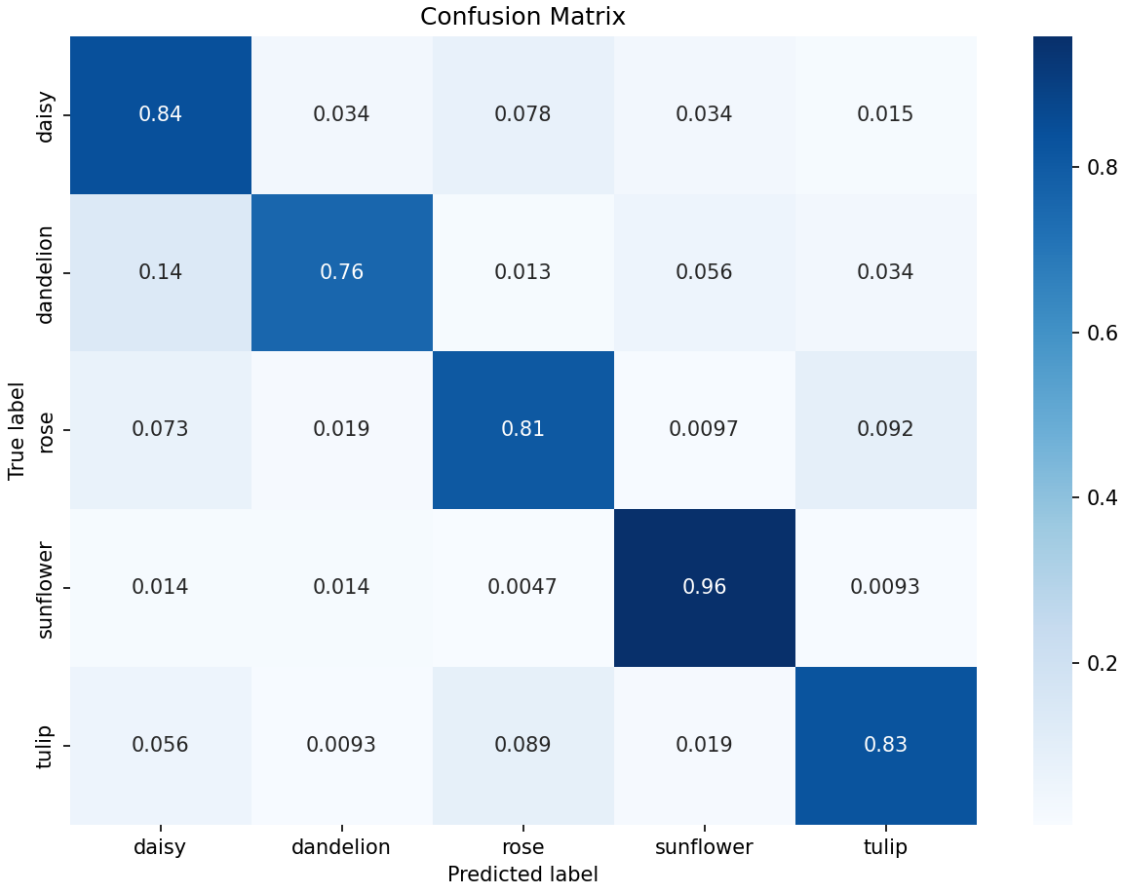
MODEL2

Test on TRAIN subsets



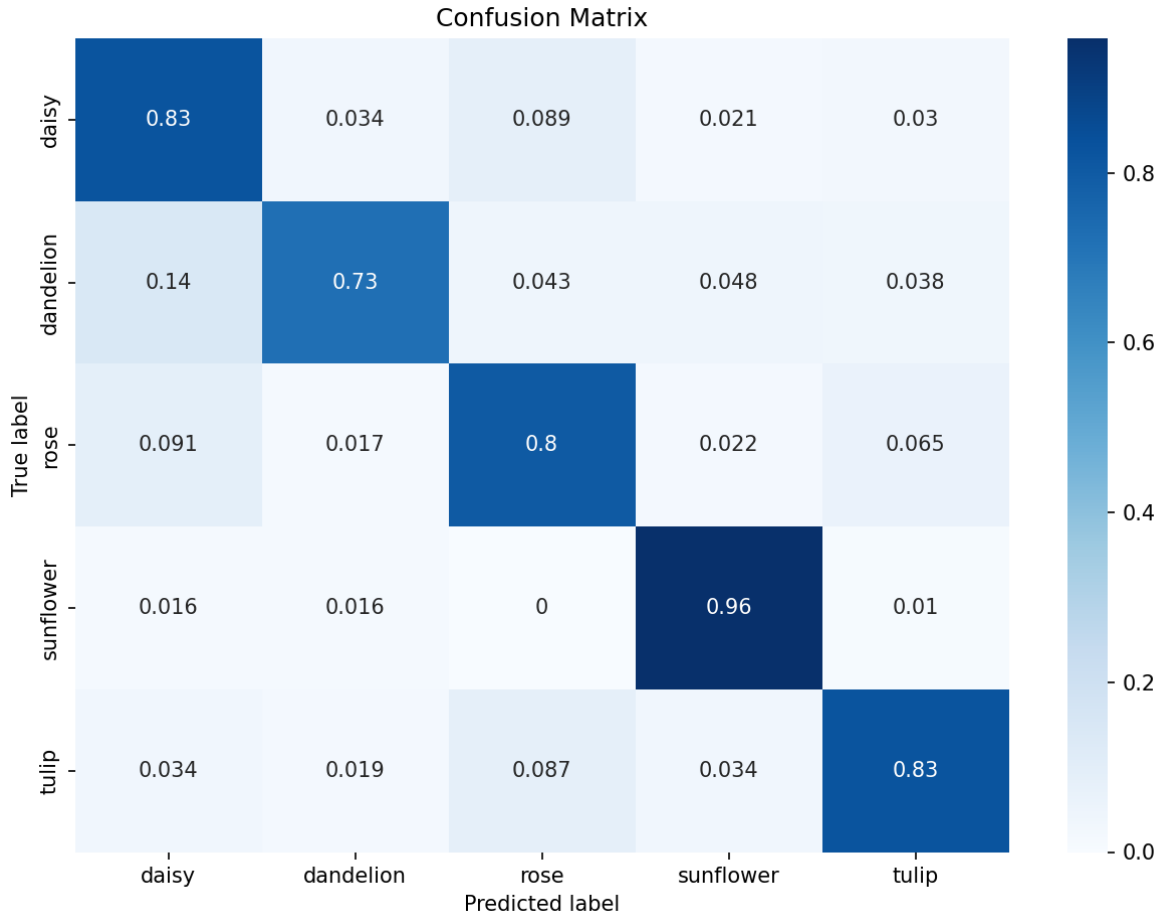
	precision	recall	f1-score	support
daisy	0.76	0.87	0.81	1705
dandelion	0.93	0.75	0.83	1705
rose	0.82	0.82	0.82	1710
sunflower	0.88	0.94	0.91	1738
tulip	0.86	0.83	0.84	1726
accuracy			0.84	8584
macro avg	0.85	0.84	0.84	8584
weighted avg	0.85	0.84	0.84	8584

Test on VAL subsets



	precision	recall	f1-score	support
daisy	0.74	0.84	0.78	205
dandelion	0.92	0.76	0.83	233
rose	0.81	0.81	0.81	206
sunflower	0.89	0.96	0.92	215
tulip	0.85	0.83	0.84	214
accuracy			0.84	1073
macro avg	0.84	0.84	0.84	1073
weighted avg	0.84	0.84	0.84	1073

Test on TEST subsets



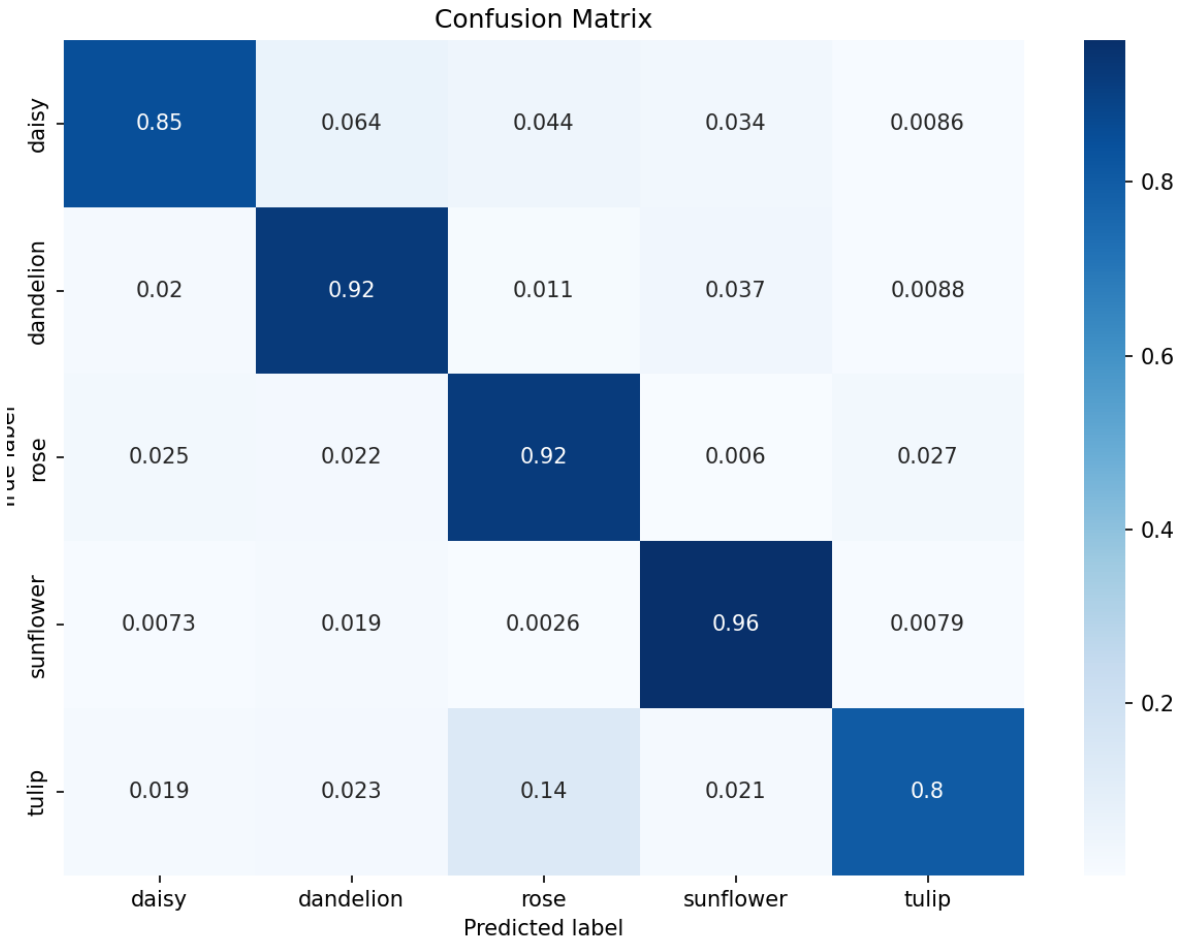
	precision	recall	f1-score	support
daisy	0.76	0.83	0.79	236
dandelion	0.89	0.73	0.80	208
rose	0.79	0.80	0.80	230
sunflower	0.87	0.96	0.91	193
tulip	0.84	0.83	0.83	206
accuracy			0.83	1073
macro avg	0.83	0.83	0.83	1073
weighted avg	0.83	0.83	0.82	1073

Conclusions

Despite applying data augmentation and normalization, the overall accuracy decreased compared to SPLIT 1. The model trained with uniformly distributed, normalized, and augmented data achieved an accuracy of 83-84% across train, validation, and test subsets. While augmentation and normalization typically enhance model performance, in this case, the overall accuracy slightly declined. The model, however, remains robust with consistent performance metrics across different data subsets.

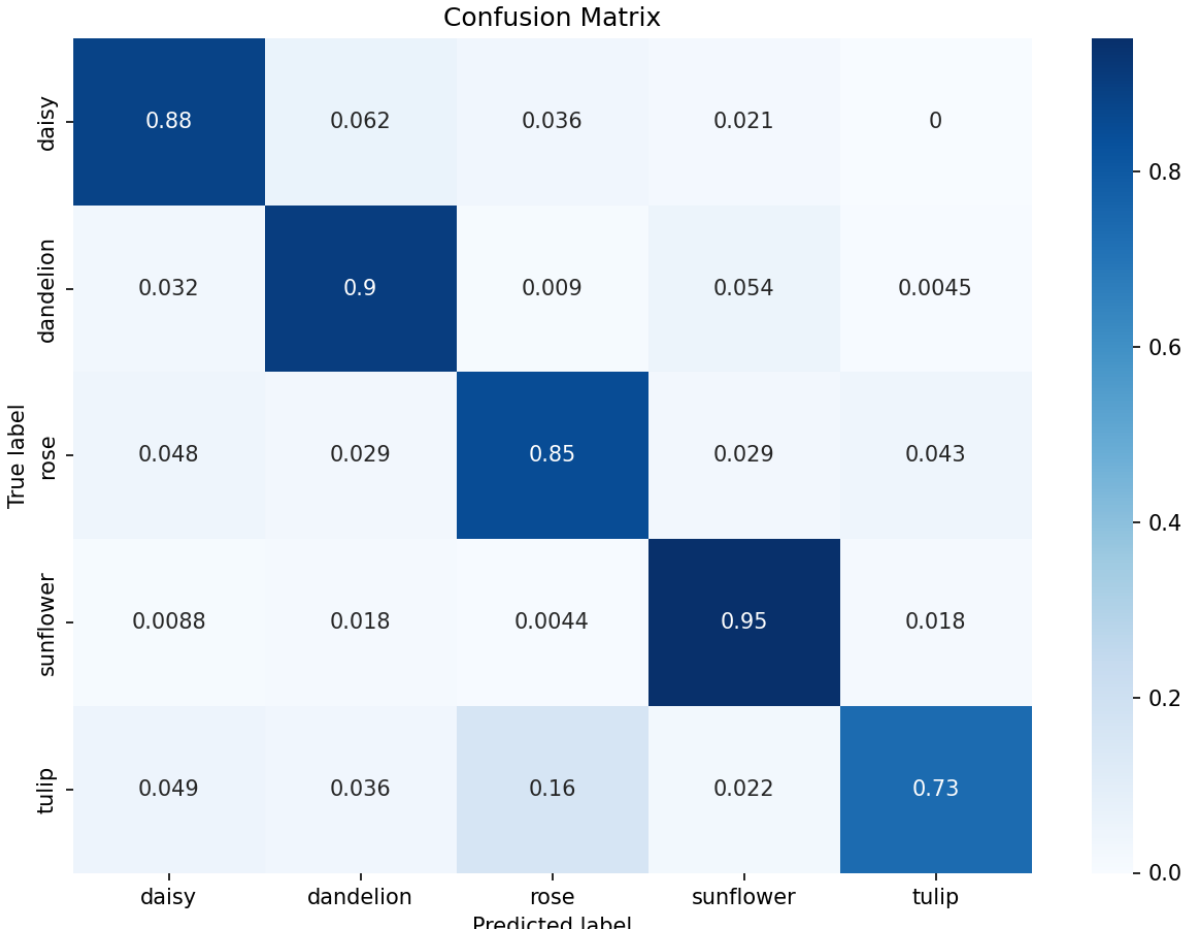
MODEL3

Test on TRAIN subsets



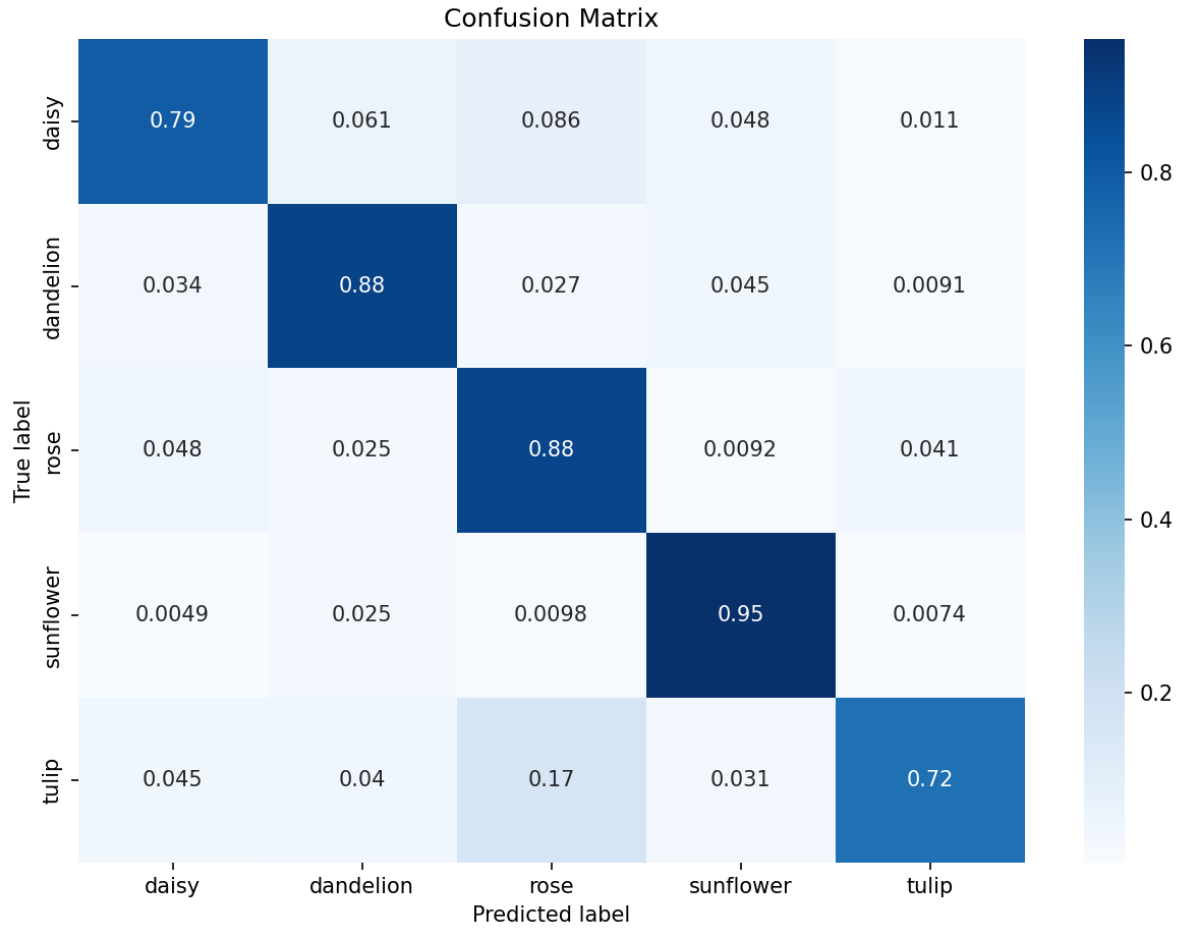
daisy	0.92	0.85	0.88	1510
dandelion	0.88	0.92	0.90	1484
rose	0.83	0.92	0.87	1503
sunflower	0.91	0.96	0.94	1512
tulip	0.94	0.80	0.87	1502
accuracy			0.89	7511
macro avg	0.89	0.89	0.89	7511
weighted avg	0.89	0.89	0.89	7511

Test on VAL subsets



	precision	recall	f1-score	support
daisy	0.85	0.88	0.87	195
dandelion	0.87	0.90	0.88	221
rose	0.79	0.85	0.82	207
sunflower	0.89	0.95	0.92	226
tulip	0.92	0.73	0.82	224
accuracy			0.86	1073
macro avg	0.86	0.86	0.86	1073
weighted avg	0.87	0.86	0.86	1073

Test on TEST subsets



	precision	recall	f1-score	support
daisy	0.86	0.79	0.83	441
dandelion	0.86	0.88	0.87	441
rose	0.75	0.88	0.81	436
sunflower	0.87	0.95	0.91	408
tulip	0.91	0.72	0.80	420
accuracy			0.84	2146
macro avg	0.85	0.84	0.84	2146
weighted avg	0.85	0.84	0.84	2146

Conclusions

Validation and test accuracies are similar across all splits. The high performance for "Sunflower" and confusion between other classes suggest areas for further refinement in the model and data preprocessing.