Assignment 1

Michael McMillan – 8116775

Introduction

I have been tasked with developing a flower image recognition system using three different machine learning classifiers: k-nearest-neighbour (i.e. k-NN), multi-layer-perceptron (MLP) and convolutional neural networks (CNN). The primary goal has been to explore and compare these classifiers in terms of their performance on a dataset of flower images. The dataset, sourced from Kaggle, consists of over 4,000 labelled images representing five flower classes: daisy, tulip, rose, sunflower and dandelion. Each class contains between 800 and 1,000 photos of varying resolutions and sizes, providing a diverse data set for training and evaluation.

Task 1: k-Nearest-Neighbour Classifier

1. The default size of the colour histogram is [6, 6, 6]. Extract colour histograms with larger or smaller sizes and observe the change in the performance of the kk-NN classifier on the test set. Describe your observation and explain it **(1 mark)**.
   a. Smaller Histogram Size (e.g., [3, 3, 3]):
      - **Enhanced Performance:**
        - **Observation:** The more minor histogram size results in the best accuracy of 47.97% with `k=15` and precision and recall values of 0.48 and 0.47, respectively.
        - **Explanation:** Fewer bins simplify the colour representation, reducing noise and making it easier for the classifier to distinguish between classes. The reduced complexity improves performance metrics.
      - **Faster Inference Time**
        - **Observation:** The average time is 0.0101 seconds, indicating faster classification than larger histogram sizes.
        - **Explanation:** A more minor feature vector results in quicker classification. The reduced dimensions allow for faster processing and comparison by the k-NN classifier.
   b. Default Histogram Size (e.g., [6, 6, 6]):
      - **Balanced Performance:**
        - **Observation:** The default histogram size offers a balanced performance with an accuracy of 44.96% with `k=15` and precision and recall values of 0.48 and 0.46, respectively.
        - **Explanation:** This histogram size provides a good trade-off between detail and feature vector size, capturing sufficient colour information for effective classification without excessive detail that could lead to overfitting.
      - **Moderate Inference Time:**
        - **Observation:** The average inference time is 0.0122 seconds, reflecting a moderate time requirement per prediction.
        - **Explanation:** The default histogram size balances complexity and efficiency. The feature vector size allows for efficient computation while maintaining good classification performance.
   c. Larger Histogram Size (e.g., [12, 12, 12]):
      - **Reduced Accuracy:**

    - **Observation:** The accuracy is significantly lower with the larger histogram size. The best accuracy is 39.40%.
    - **Explanation:** Excessive granularity in colour histograms can lead to loss of useful information and noise, making it harder for the classifier to differentiate between images, resulting in reduced accuracy.
- **Reduced Precision and Recall:**
    - **Observation:** Lower precision and recall values
    - **Explanation:** Detailed histograms may not capture distinguishing features well, leading to more misclassifications.
- **Stable and Moderate Inference Time:**
    - **Observation:** The inference times are relatively stable and moderate, with an average time of 0.0492.
    - **Explanation:** Although the feature vector is large, inference time remains moderate. The issue is more about classifier performance than computation time.

2. Describe how you find an excellent k for the k-NN classifier. Use quantitative results to support your selection **(1 mark)**.
    - **Define a range of (k) values:** Create a list of possible (k) values to test.
    - **Initialise variables:** Set up variables to keep track of the best accuracy and the corresponding (k) value.
    - **Evaluate each (k) value:** For each (k) value in the k list, train the (k)-NN classifier on the training set and evaluate it on the validation set. Measure the accuracy for each (k) value.
    - **Select the best (k):** compare the accuracy scores for each (k) value and select the (k) value that gives the highest accuracy on the validation set.

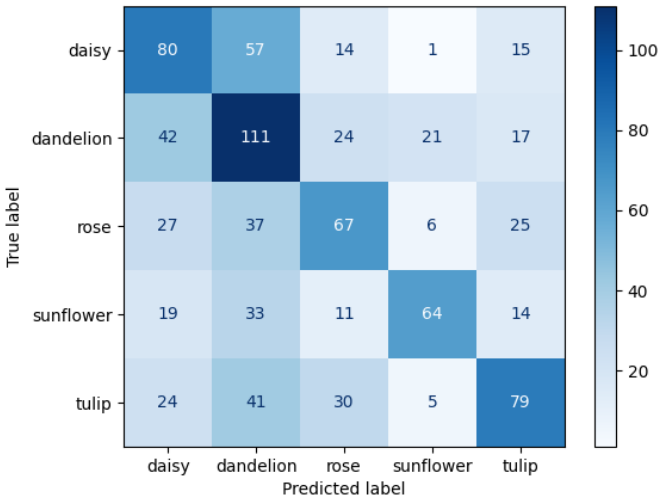
Quantitative Results

```
k=1, Accuracy=40.67
k=2, Accuracy=39.05
k=3, Accuracy=42.41
k=4, Accuracy=44.38
k=5, Accuracy=43.57
k=6, Accuracy=42.76
k=7, Accuracy=43.34
k=8, Accuracy=43.45
k=9, Accuracy=43.11
k=10, Accuracy=44.03
k=11, Accuracy=44.73
k=12, Accuracy=44.50
k=13, Accuracy=44.38
k=14, Accuracy=43.57
k=15, Accuracy=44.96
k=16, Accuracy=44.26
k=17, Accuracy=44.50
k=18, Accuracy=44.50
k=19, Accuracy=44.03
k=20, Accuracy=42.06
k=21, Accuracy=41.95
k=22, Accuracy=42.87
k=23, Accuracy=42.99
k=24, Accuracy=43.80
k=25, Accuracy=43.11
k=26, Accuracy=42.87
k=27, Accuracy=42.87
k=28, Accuracy=43.57
k=29, Accuracy=43.68
k=30, Accuracy=44.38

Best accuracy is 44.96, k=15
k = 15: Error = 53.59
```

Name: Michael McMillan
Student no. 8116775

3. Report the classification metrics (i.e., accuracy, precision, recall, and F1 score) on the test set and the pair of flower classes that confuses the kk-NN classifier most **(1 mark)**.




```
Accuracy: 0.46
Precision: 0.48
Recall: 0.46
F1: 0.47
```

Based on the Confusion Matrix, the flower classes most confusing the classifier are daisy and dandelion, with daisy being misclassified as dandelion 57 times.

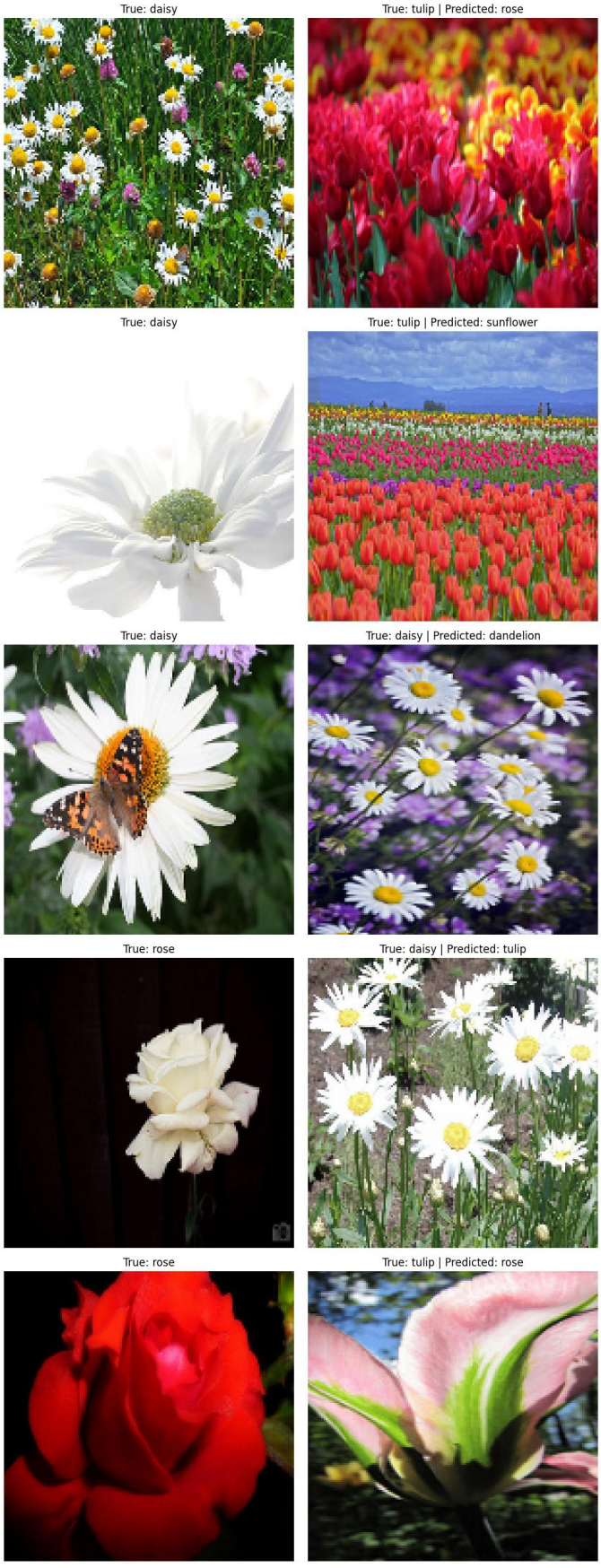
4. Report the average inference time of the kk-NN classifier, i.e., how long the classifier takes to classify a single sample. Run the code ten times and compute the average of the inference times **(1 mark)**.

```
Inference time 1 - 0.015902300001471303
Inference time 2 - 0.012081099999704747
Inference time 3 - 0.01094829999965441
Inference time 4 - 0.011564799999177922
Inference time 5 - 0.012357300000076066
Inference time 6 - 0.011990400000286172
Inference time 7 - 0.010889500001212582
Inference time 8 - 0.012224999998579733
Inference time 9 - 0.011399800001527183
Inference time 10 - 0.010755799999969895
Average inference time: 0.0120 seconds
```

The average inference time based on the results given by the script is 0.0120 seconds.

Name: Michael McMillan
Student no. 8116775

5. Show five correctly and five incorrectly classified images **(1 mark)**.



Name: Michael McMillan
Student no. 8116775

Task 2: Multi-layer Perceptron

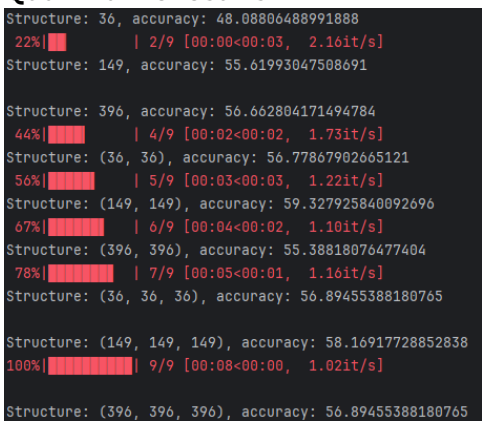
1. The structure of an MLP classifier usually consists of the number of hidden layers and the number of neurons in each hidden layer. The number of hidden layers is usually set as 1, 2, or 3. The number of neurons in a hidden layer mm can be set by using one of the following empirical rules:
    - M should be between the size of the input layer and the size of the output layer
    - M should be 2/3 the size of the input layer, plus the size of the output layer
    - M should be less than twice the size of the input layer.

    Use the above information to design *nine* different structures for your MLPs, as shown in the following table. Please replace "?" with the number of neurons in each hidden layer in your design **(1 mark)**.

MLP structure	Number of hidden layers	Number of neurons in each hidden layer
1	1	159
2	1	149
3	1	171
4	2	159, 159
5	2	149, 149
6	2	171, 171
7	3	159, 159, 159
8	3	149, 149, 149
9	3	171, 171, 171

2. Describe how to identify the optimal network architecture from the nine options. Report the quantitative results to support your selection **(1 mark)**.
    - Choosing appropriate metrics to evaluate the network architecture. Typical metrics include Loss and accuracy.
    - Prepare the data by splitting it into training, validation, and test data. Ensure that the split is representative and consistent across all structures.
    - Train and evaluate each structure/architecture through:
        - creating the MLP with specified hidden layers and neurons.
        - Train the network using the training set and apply the same procedure across all structures.
        - Evaluate the model on the validation set after training to monitor performance and avoid overfitting.
        - Test the model to evaluate generalisation.

Quantitative results

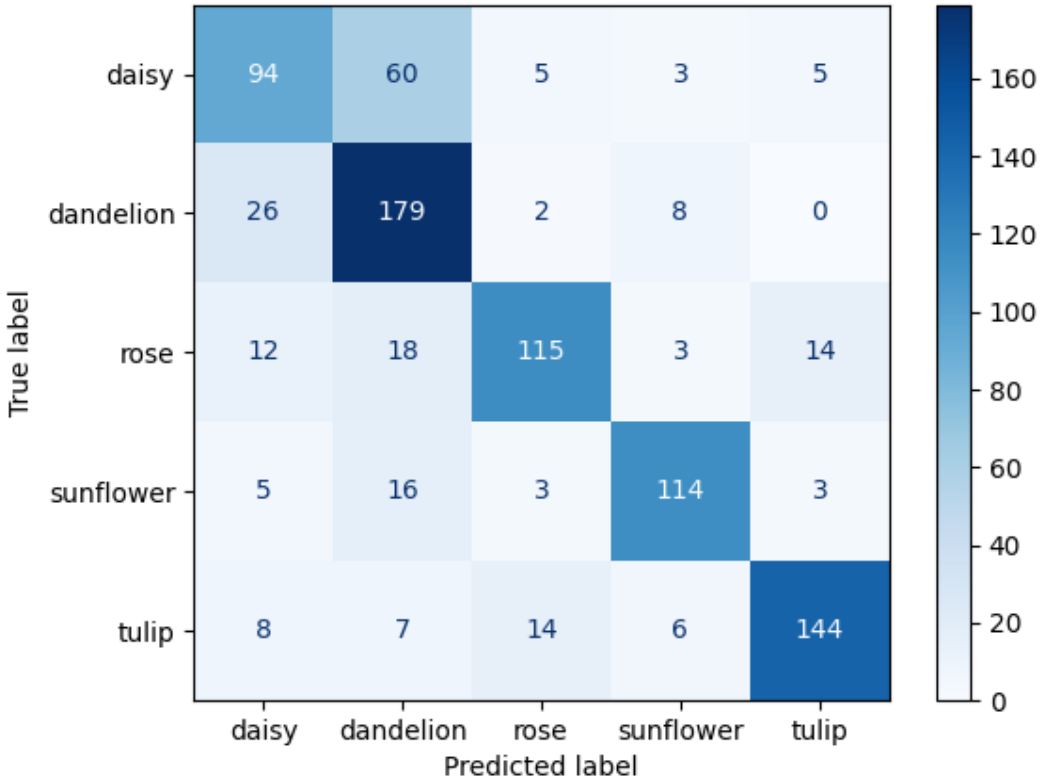


Name: Michael McMillan
Student no. 8116775

2. Report the classification metrics (i.e., accuracy, precision, recall, and F1 score) on the test set and the pair of flower classes that confuses the MLP classifier most **(1 mark)**.

```
Accuracy score: 54.5139 | Precision: 56.7620 | Recall: 53.7147 | F1: 54.0365
```

Confusion Matrix:



Based on the Confusion Matrix, the flower classes that confused the MLP classifier the most are dandelion and daisy, where dandelion is misclassified as daisy 60 times.

3. Report the training time and inference time of the MLP classifier for a single run **(1 mark)**.

```
Training time is 0.2754 seconds
Inference time is 0.0011 seconds
Accuracy score: 68.0556 | Precision: 69.6355 | Recall: 67.9239 | F1: 68.3049
Time taken: 0.5746 seconds
```

Name: Michael McMillan
Student no. 8116775

4. Show five correctly classified images and five incorrectly classified images **(1 mark)**.

True: dandelion | Predicted: dandelion

True: rose | Predicted: dandelion

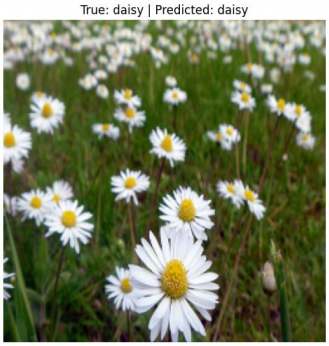
True: daisy | Predicted: daisy

True: dandelion | Predicted: daisy

True: tulip | Predicted: tulip

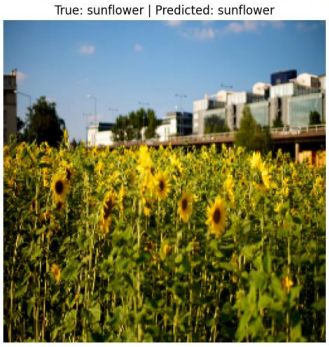
True: dandelion | Predicted: daisy

True: daisy | Predicted: daisy



True: dandelion | Predicted: sunflower

True: sunflower | Predicted: sunflower



True: dandelion | Predicted: daisy

Name: Michael McMillan
Student no. 8116775

Task 3: Convolutional Neural Network

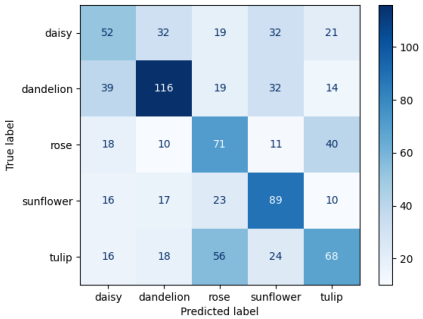
1. Show the designed network architecture and describe its components **(1 mark)**.

```
FlowerClassificationModel(
 (conv_block1): Sequential(
  (0): Conv2d(3, 30, kernel_size=(3, 3), stride=(1, 1))
  (1): ReLU()
  (2): Conv2d(30, 30, kernel_size=(3, 3), stride=(1, 1))
  (3): ReLU()
  (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
 )
 (conv_block2): Sequential(
  (0): Conv2d(30, 30, kernel_size=(3, 3), stride=(1, 1))
  (1): ReLU()
  (2): Conv2d(30, 30, kernel_size=(3, 3), stride=(1, 1))
  (3): ReLU()
  (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
 )
 (classifier): Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=111630, out_features=5, bias=True)
 )
)
```

- **Convolutional Blocks**
  - `**conv_block1**`: Applies two convolutional layers (`**Conv2d(3, 30)**` and `**Conv2d(30, 30)**`) with ReLU activation after each, followed by max pooling (`**MaxPool2d(kernel_size=2, stride=2)**`), to detect and refine features from input images.
  - `**conv_block2**`: Applies two more convolutional layers (`**Conv2d(30, 30)**` and `**Conv2d(30, 30)**`), with ReLU activation, followed by max pooling, further extracting and abstracting features.
- **Classifier**
  - `**Flatten(start_dim=1)**`: Converts the 3D feature maps into a 1D feature vector.
  - `**Linear(in_features=111630, out_features=5)**`: Maps the flattened vector into the output classes (5), providing the final classification.

2. Report the classification metrics (i.e., accuracy, precision, recall, and F1 score) on the test set and the pair of flower classes that confuses the CNN classifier the most **(1 mark)**.

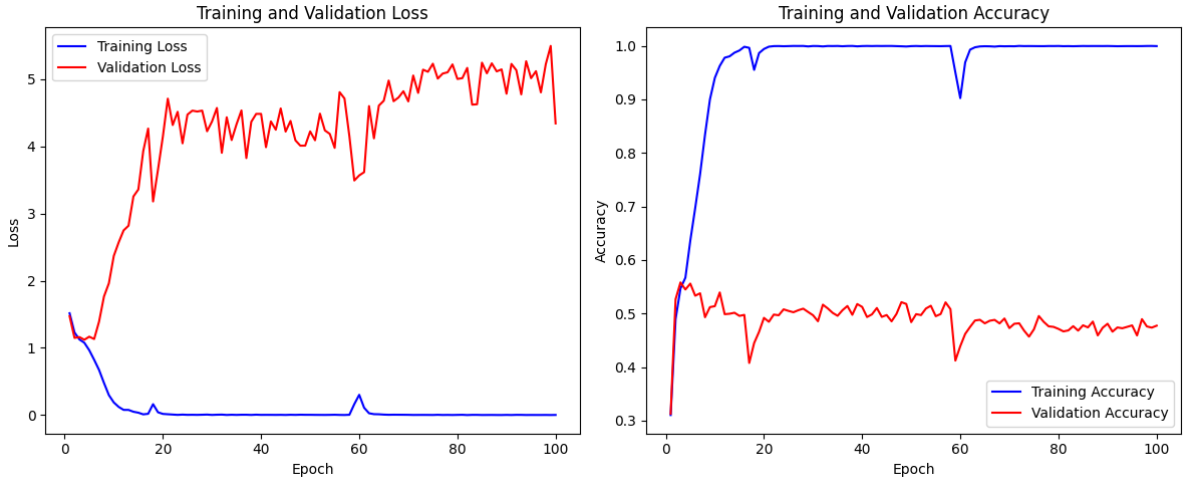
Test accuracy: 0.4586 | Test f1: 0.4324 | Test recall: 0.4435 | Test precision: 0.4586



Name: Michael McMillan
Student no. 8116775

Based on the Confusion Matrix, the two flower classes most confusing to the CNN classifier are tulip and rose, where tulip is misclassified as rose 56 times.

3. Plot training loss and validation loss concerning the number of epochs. Describe the changes in these two losses. Plot and answer the same question for CNN model accuracy



- **Training loss and validation loss changes**
  - Initial Epochs (0-10):
    - Training loss rapidly decreases, showing effective learning.
    - The validation loss initially decreases slightly but then increases.
  - Mid to Late Epochs (10-100):
    - The training loss stabilises near zero, indicating an excellent fit to the training data.
    - The validation loss increases significantly and fluctuates erratically.
  - Analysis:
    - Diverging loss trends suggest overfitting, where the model memorises training data but struggles to generalise to unseen data.
- **Training accuracy and validation accuracy changes**
  - Initial Epochs (0-10):
    - The training accuracy rapidly increases, approaching 1.0.
    - The validation accuracy increases initially but plateaus around 50%.
  - Mid to Late Epochs (10-100):
    - The training accuracy stays nearly 1.0, indicating that the model predicts the training data perfectly.
    - The validation accuracy fluctuates slightly but stays well below the training accuracy, showing that the model does not generalise well.
  - Analysis
    - The significant gap between the high training accuracy and low validation accuracy is a clear sign of overfitting, leading to poor performance on unseen data.

Name: Michael McMillan
Student no. 8116775

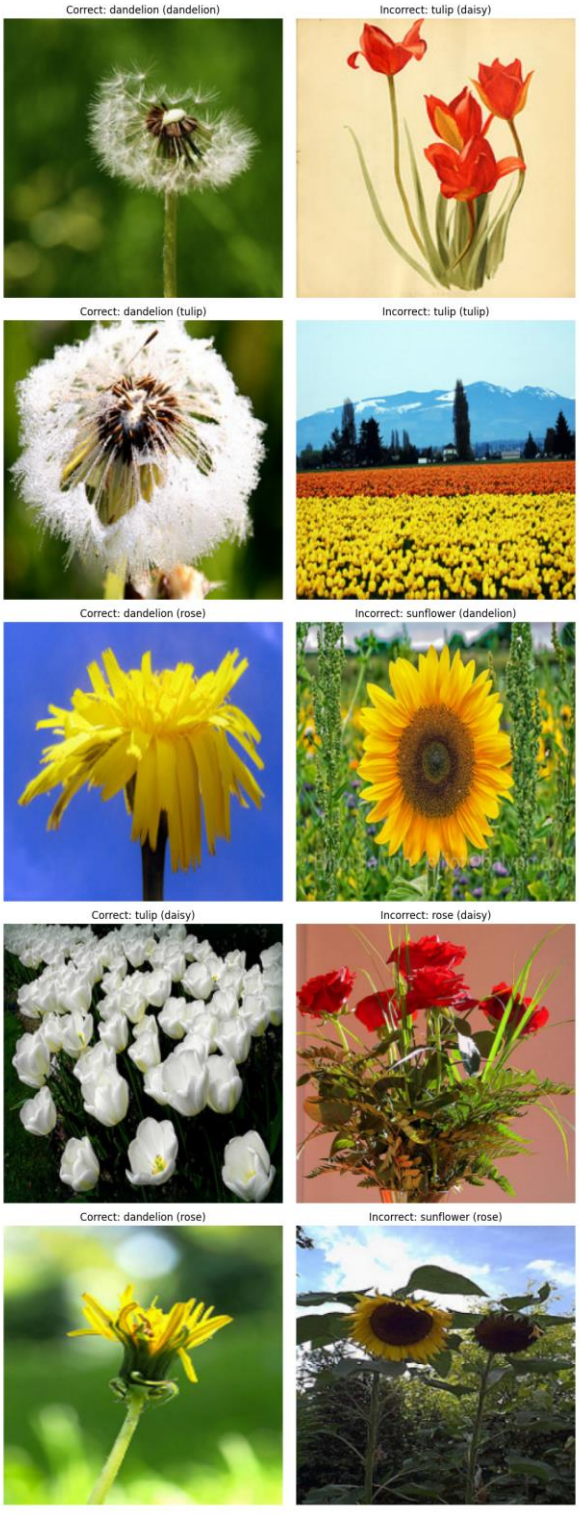
4. Report training time and inference time of the CNN classifier for a single run

Training time for a single run: 9.4131 seconds

Inference time for a single run: 2.5406

5. Show 5 correctly and 5 incorrectly classified images

I suspect the reason that the code is unable to classify the images reliably is probably because the model is not able to gain enough detail about the images to classify them correctly.



W

Name: Michael McMillan

Student no. 8116775