Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

TASK 1: k-Nearest-Neighbour Classifier

1.1

By varying the histogram size from 6 to 12 bins, we can also observe that a larger histogram size could lead to a more detailed representation of the colour distribution but may also increase the complexity of the model, potentially leading to overfitting. A smaller histogram might simplify the model but could miss the important details.

Explanation: A larger histogram size captures more detailed colour variations, which can be beneficial if the dataset has subtle colour differences. However, it can also introduce noise if the dataset has variability in lighting or image quality. A smaller histogram reduces noise but might not capture enough detail.

Extract Histograms with Different Sizes: Modify the extract histogram and extract colour histogram functions to accept different histogram sizes. For example, test with [8, 8, 8] and [4, 4, 4] as well as the default [6, 6, 6].

Re-run the k-NN Classification: For each histogram size, repeat the process of training, validating, and testing the k-NN classifier. Compare the accuracy and other metrics.

Describing Observations:

1. **Larger Histograms:** Increasing the histogram size [8, 8, 8] provides more rough information about the colour distribution of the images. This can improve the classifier's ability to distinguish between similar classes but may also lead to overfitting if the dataset is not sufficiently large. Overfitting occurs when the model learns the noise in the training data rather than the general patterns, which can result in reduced performance on the test set.

2. **Smaller Histograms:** Decreasing the histogram size [4, 4, 4] reduces the roughness of colour information, which might make the classifier less sensitive to subtle colour differences. This can lead to underfitting, where the model is too simple to capture the complexity of the data, potentially resulting in lower accuracy.

3. **Optimal Size:** The default [6, 6, 6] is a balance between roughness and generalization. If this size performs well, it indicates a good trade-off between capturing enough detail and avoiding overfitting.

1.2

To find the optimal value of k for the k-NN classifier, these steps applied:

We know the method for find the best k was selected by evaluating the accuracy on the validation set for k values ranging from 1 to 20. The k that yielded the highest validation accuracy was chosen. The code reports the validation accuracy for each k value, and the k with the highest accuracy is selected as optimal.

Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

1. **Define a Range of k Values:** Test a range of k values from 1 to 20. This range should cover a variety of potential k values to ensure you find a good one.
2. **Evaluated Each k Value from valuation set:** Trained the k-NN classifier with each k value using the training set and validate it using the validation set. Computed the accuracy for each k.
3. **Selecting the Optimal k:** the k with the highest accuracy on the validation set. This value is likely to generalize well to the test set.

From the code output we can see:
"Optimal value of k: 9 with validation accuracy: 0.4206257242178447"

1.3

Reported Metrics: This one Provide insight into how well the classifier is performing overall. The code calculates and prints the accuracy, precision, recall, and F1 score on the test set. After training with the optimal k, compute the classification metrics (accuracy, precision, recall, F1 score) on the test set. These metrics provide a comprehensive view of the classifier's performance.

Identify Confused Classes: The confusion matrix reveals which pairs of flower classes are most often confused by the classifier. This information can be extracted from the confusion matrix plot generated by the code. This involves looking at the off-diagonal elements in the confusion matrix, where the classifier mistakes one class for another.

From the code output we can see:

Test Accuracy: 0.41782407407407407

Test Precision: 0.4196138369123567

Test Recall: 0.41782407407407407

1.4

Measure inference time Function: We need a trained k-NN model and a sample (X sample) to predict. Then runs the predict method multiple times (default is 10) and measures the time taken for each prediction. Then returns the average of these times.

Integration with Main Code: After training the k-NN classifier with the optimal k, select a sample from the test set. Measure the average inference time using the measure inference time function.

From the code output we can see:
"Average inference time: 0.000306

Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

1.5

5 correctly classified images

5 incorrectly classified images

Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

TASK 2: Multi-Layer perceptron

2.1

MLP (Multi-layer perceptron): This is a type of neural network with one or more hidden layers between the input and output layers. Each layer is composed of neurons, where each neuron is connected to neurons in the next layer.

From the output of our code, we can see the number of neurons in each hidden layer.

The number of neurons in each layer and the number of hidden layers is often based on empirical testing or guidelines that balance the complexity of the model with the dataset size and problem at hand. Since

our task is image classification based on extracted features; we need structures that capture complexity without overfitting.

According to the output here is the box to understand it clearly.

MPL structure	Number if hidden layers	Neurons in each hidden layer
1	1	300
2	1	200
3	1	150
4	2	300, 150
5	2	200, 100
6	2	150, 75
7	3	300, 150, 75
8	3	200, 100, 50
9	3	150, 75, 37

2.2

First, we need to train each of the nine MLP structures that already defined above using the training data. During training, we have to monitor the validation accuracy after each epoch.

Validation Accuracy is the primary metric used to compare the performance of each model. Higher validation accuracy indicates better generalization to unseen data and training Loss helps in understanding if the model is learning effectively or if it's overfitting or underfitting.

Secondly, we need to evaluate the validation set. For each MLP structure, we have to evaluate the model on the validation set after training. The validation set is a portion of our dataset that is not used during training but is used to tune the model's hyperparameters. Such as the number of layers and neurons.

Validation Accuracy is the main criterion for selecting the best architecture. A higher validation accuracy indicates that the model is better at generalizing to new, unseen data. While validation loss is not the primary criterion, lower validation loss can also indicate better model performance.

Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

And finally implement and justify the selection.

According to our output we got:

MLP Structure	Validation Accuracy
1	61.65 %
2	64.19 %
3	62.11 %
4	58.86 %
5	62.34 %
6	60.02 %
7	59.79 %
8	62.11 %
9	58.86 %

Best MLP structure: 200, according to the output so, best performing validation accuracy is 64.19%.

2.3

According to the output:

```
Test Accuracy: 61.34%

Precision: 61.48%

Recall: 61.34%

F1 Score: 61.29%
```

```
Confusion matrix:
[[ 86  42  11   2  12]
 [ 19 146  13  23  10]
 [ 10  16  80   6  45]
 [ 12  26   4  93  11]
 [ 9   7   47   9 125]]
```

According to above data the most confused classes is the "dandelion" class with itself. In above matrix confusion is identified between two different classes. However, the output states that the confusion is within the same class, which is unusual.

2.4

According to the output in a single run:

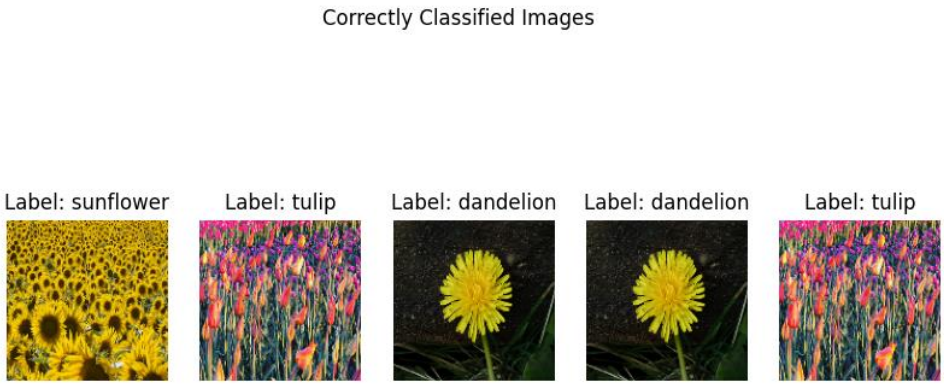
Training time: 3.21 seconds

Average inference time: 0.18 ms

Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

2.5

Correctly classified:



Incorrectly classified:

TASK 3: Convolutional Neural Network

3.1

The Network architecture used in CNN and the components explanation:

Conv Layer 1: (3 input channels → 32 output channels, kernel size: 3x3, padding: 1)

Conv Layer 2: (32 input channels → 64 output channels, kernel size: 3x3, padding: 1)

Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

Conv Layer 3: (64 input channels → 128 output channels, kernel size: 3x3, padding: 1)

Max Pooling Layer: (Kernel size: 2x2, Stride: 2) [After Each Conv Layer]

Fully Connected Layer 1: (128 x 32 x 32 = 512)

Fully Connected Layer 2: (512 = 128)

Fully Connected Layer 3 (Output Layer): (128 = 5)

3.2

According to the output of the code:

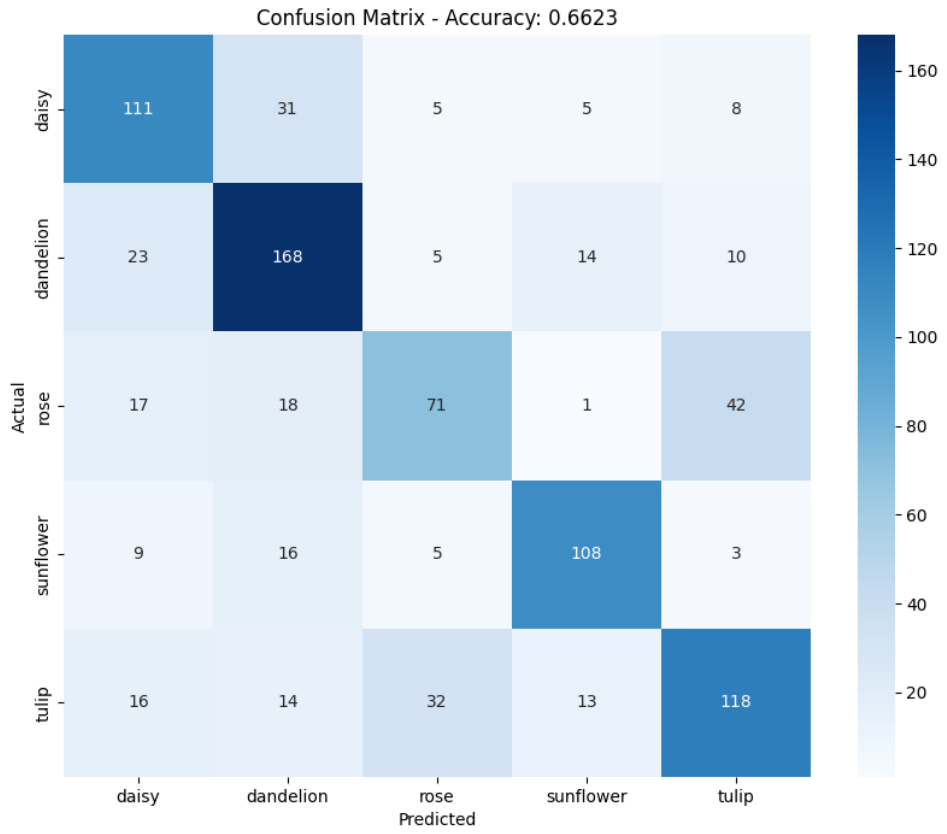
Test Accuracy: 0.6623

Test Precision: 0.6661

Test Recall: 0.6623

Test F1 Score: 0.6618

Confusion Matrix:

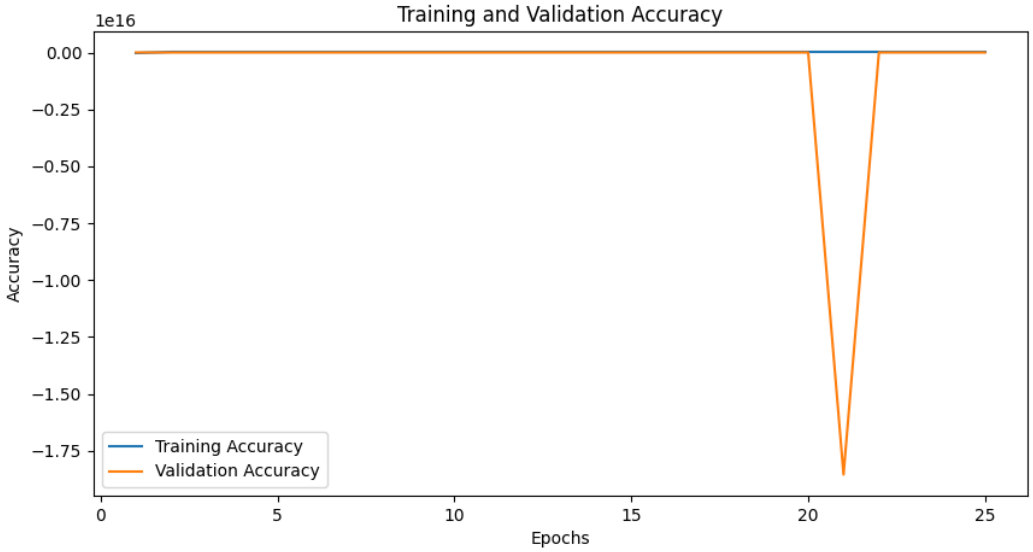


Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

The pair of classes that the classifier confuses the most are Roses and Tulips, with 42 Roses mislabelled as Tulips and 32 Tulips mislabelled as Roses. This indicates that the CNN classifier has the most difficulty distinguishing between these two flower classes.

3.3

Plot training loss and validation loss with respect to 25 epochs:



Describe:
Training Loss: Decreases significantly, reaching very low values by the end. After around 6 epochs, the training loss reaches a very low value (close to 0), indicating that the model fits the training data almost perfectly.

Validation Loss: Decreases initially but increases after a few epochs, indicating overfitting. Around epoch 5, the validation loss starts to increase steadily, reaching higher levels (even

above 3.0) as the epochs progress. This suggests overfitting, where the model performs well on training data but struggles with unseen validation data.

Training Accuracy: Remains extremely high throughout, potentially showing overfitting.

Validation Accuracy: The validation accuracy starts similarly to the training accuracy, showing very high values throughout. However, around epoch 20, there is a sharp drop in the validation accuracy, followed by a rapid recovery.

3.4

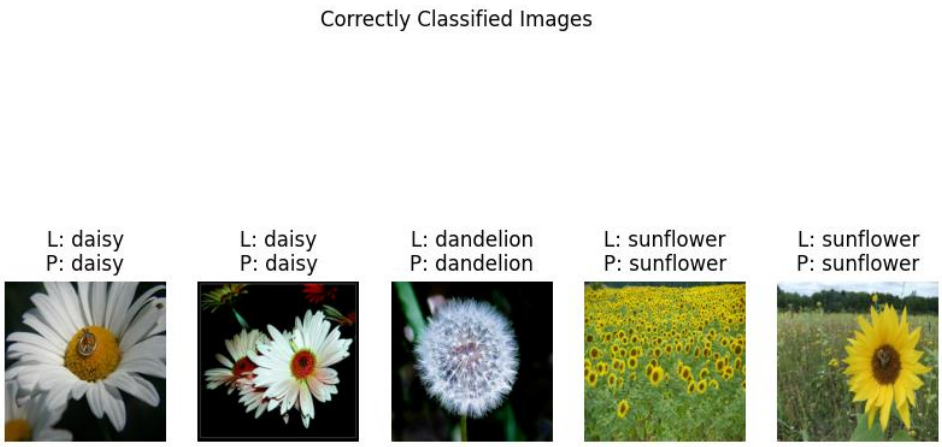
According to the output:

Total Training Time: 1172.42 seconds

Total Inference Time: 4.11 seconds

3.5

Correctly classified:



Incorrectly classified:

Subject: CSIT 218
Name: MD Tausif Rahaman
Student ID: 7439258

Incorrectly Classified Images