CSCI218 Foundations of Artificial Intelligence

Assignment (15%)

Due Date:

Flower Image Recognition using k-NN, MLP and CNN

Overview

Automatic flower image classification has applications in floriculture, botany, home interior design, and online shopping, to name just a few. In this assignment, you need to implement flower image classification by using three kinds of classifiers, i.e., k-nearest-neighbour classifier (k-NN), multi-layer perceptron (MLP), and convolutional neural networks (CNN).

You are provided with a labelled flower image data set collected from Kaggle. It contains 4000+ photos of flowers including daisy, tulip, rose, sunflower, and dandelion. About 800~1000+ photos are available for each class with various pixels resolution and sizes.

What you need to complete this assignment:

- The **flower image data set** which can be downloaded from “Week 3 Lab (Flower Recognition)” in the Moodle site of this subject.
- The lecture & lab content and recordings in Weeks 3, 4 and 5.
- Sample code available on the Moodle site.
- Python 3 programming environment + required libraries, packages, and modules.

Note that:

- The sample code provides the essential functions for extracting features and loading dataset. You **must modify and complete the code to solve the tasks**.
- You should maximally utilise the predefined functions in existing libraries, packages, and modules (e.g., scikit-learn) to complete this assignment.

Task 1: k-Nearest-Neighbour Classifier

Task objectives

- Learn to implement the basic procedure of classification.
- Understand k-nearest-neighbour (k-NN) classifier and handcrafted features.
- Learn to evaluate the performance of a classifier.
- Learn to use the k-NN and other functions provided by Scikit Learn.
- Learn to use libraries, packages and modules related to image classification.

Instructions

Your program must include the following steps:

- Import the required modules.

- Extract colour histogram features.
- Randomly split the dataset into training (60%), validation (20%) and test (20%) sets.
- Find a good k for the k-NN classifier with the validation set.
- Build a k-NN classifier with the optimal value of k.
- Evaluate the performance of the k-NN classifier on the test set.

Questions (5 marks)

1.1. The default size of the colour histogram is [6, 6, 6]. Extract colour histograms with larger or smaller sizes and observe the change of the performance of the k-NN classifier on the test set. Describe your observation and explain it **(1 mark)**.

1.2. Describe how you find a good k for the k-NN classifier. Use quantitative results to support your selection **(1 mark)**.

1.3. 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 k-NN classifier most **(1 mark)**.

1.4. Report the average inference time of the k-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)**.

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

Task 2: Multi-layer Perceptron

Tak objectives

- Understand neural networks and multilayer perceptron (MLP) classifier.
- Learn to implement an MLP classifier with different structures.

Instructions

This task is based on your completion of Task 1. Your program must include the following steps:

- Import the required modules.
- Extract colour histogram features.
- Randomly split the dataset into training (60%), validation (20%) and test (20%) sets.
- Select a good structure (see below) for the MLP classifier with the validation set.
- Build the MLP classifier with the identified architecture.
- Evaluate the performance of the MLP classifier on the test set.
- Show examples of correctly and wrongly classified flower images.

Questions (5 marks)

2.1. 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 m 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	{?}
2	1	{?}
3	1	{?}
4	2	{?,?}
5	2	{?,?}
6	2	{?,?}
7	3	{?,?,?}
8	3	{?,?,?}
9	3	{?,?,?}

2.2. Describe how to identify the optimal network architecture from the nine options. Report the quantitative results to support your selection **(1 mark)**.

2.3. 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 MLP classifier most **(1 mark)**.

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

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

Task 3: Convolutional Neural Network

Task objectives

- Understand convolutional neural networks (CNN) classifier.
- Learn to implement a CNN classifier with different architectures.

Instructions

This task is based on your completion of Task 1. Your program must include the following steps:

- Import the required modules.
- Randomly split the dataset into training (60%), validation (20%) and test (20%) sets.
- Build a CNN classifier by following the architecture in the example code.
- Monitor the training process of CNN on training and validation sets.
- Evaluate the performance of the CNN classifier on the test set.

Questions (5 marks)

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

3.2. 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 most **(1 mark)**.

3.3. Plot training loss and validation loss with respect to the number of epochs. Describe the changes of these two losses. Plot and answer the same question for CNN model accuracy **(1 mark)**.

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

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

Submission

- Submit **a single PDF file** which contains your answers to the questions of all tasks. All questions are to be answered. A clear and complete explanation needs to be provided with each answer.
- The PDF must contain typed text of your answers (**do not submit a scan of a handwritten document**. Any handwritten document will be ignored). The document can include computer generated graphics and illustrations (hand-drawn graphics and illustrations will be ignored).
- The PDF document of your answers should be no more than 12 pages including all graphs and illustrations. Appendix is allowed and will not be counted for the 12-page limit.
- You must show your name and student number on the first page of the PDF report.
- The size limit for this PDF report is 20MB.
- Your PDF report must begin with a short introduction on the assignment and the dataset.
- Submit **your Python code for each task** separately. Your code must be well organised and commented.
- ZIP all the files into **a single zip file and submit it via the submission link on Moodle**.
- **Late submission will not be accepted** without academic consideration being granted.

**** END ****