COMP9517: Computer Vision

2024 T3 Lab 3 Specification

Maximum Marks Achievable: 2.5

This lab is **worth 2.5% of the total course marks**.

The lab files should be submitted online. Instructions for submission will be posted closer to the deadline. **Deadline for submission is Week 5, Friday 11 October 2024, 18:00:00 AET.**

Objective: This lab revisits important concepts covered in the lectures of Week 4 and aims to make you familiar with implementing specific algorithms.

Software: You are required to use OpenCV 3+ with Python 3+ and submit your code as a Jupyter notebook (see coding and submission requirements below). For this lab you will also need to use SciKit-Learn (see links below). In the tutor consultation session this week, you can ask any questions you may have about this lab.

Materials: The dataset to be used in this lab is the Chinese MNIST dataset. It consists of images of 15 numbers, handwritten 10 times by 100 nationals, resulting in a dataset of 15,000 images and corresponding labels. Each image is 64 x 64 pixels. The dataset is available on WebCMS3. The original website with more information can be found here:
https://www.kaggle.com/datasets/gpreda/chinese-mnist

Submission: All code and requested results are assessable after the lab. Submit your source code as a Jupyter notebook (.ipynb file) that includes all output and answers to all questions (see coding requirements at the end of this document) by the above deadline. The submission link will be announced in due time.

Task (2.5 marks)

The goal of this lab is to implement and compare the performance of a K-Nearest Neighbours (KNN) classifier, a Decision Tree (DT) classifier, and a Stochastic Gradient Descent (SGD) classifier on the Chinese MINST dataset.

Step 1: Import relevant packages

We will mainly be using SciKit-Learn for this lab, so make sure you have correctly installed this library before moving to the next steps. Check the following link to learn more about the

library and ways to install it: https://scikit-learn.org/stable/

Check the following links on how to import KNN, DT, and SGD classifiers:
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

Step 2: Download the dataset
Download the dataset from WebCMS3 or from the original Kaggle webpage (linked above) and familiarize yourself with it. Specifically, you can check how many images and labels there are in the entire dataset, what the size of each image is, and how many classes there are. Also, have a look at some images of each label.

Step 3: Split and sample the dataset
There are 15,000 images in the dataset. To reduce computational cost, we can work with a subset of the complete dataset. Initially, take 5,000 randomly sampled images for training and 1,000 randomly sampled images for testing.

Make sure that random sampling is done in a stratified way. That is, it must be done such that the number of images in each class ends up being approximately the same. With 15 classes, you should have about 333 images of each class in the training set, and about 66 images of each class in the testing set. This is to avoid class imbalance in training and testing.

Also make sure that the test images are not included in the training set. The training and testing sets must be completely independent. This is to avoid leakage from the training to the testing results (it is generally easier for a classifier to correctly classify a sample from the training set than a new sample it has never seen before).

After completing all the steps below, double the number of training images from 5,000 to 10,000, and repeat the experiment to see if it makes a difference in classification performance (you can keep the number of test images fixed at 1,000).

Step 4: Perform necessary data reshaping
Once you get a subset of the dataset to work with, you need to reshape the training and testing data in order to apply machine learning classifiers. Each image in the Chinese MNIST dataset is 64 x 64 pixels, so you need to reshape it.

Step 5: Initialise the classifier
For each classifier (KNN, DT, SGD), initialise the classifier object. It is important to read the documentation to learn about various parameters that can be configured when initialising classifiers. For KNN, set k to 3. For SGD, set the maximum number of iterations (epochs) to

250. Use default settings for all other parameters.

Step 6: Fit the classifier to the training data

The SciKit-Learn library has a fitting method to learn from data. Use the fit() method to train each classifier by passing the training data and the training labels as parameters. The correct label (class) of each image follows from the file name (or the separate CSV file).

Step 7: Evaluate the trained model on the testing data

After you have trained a classifier (also called a model), you can use it to make predictions on the testing data. Use the predict() method provided by the SciKit-Learn library.

Step 8: Report the performance of each classifier

To quantify how each of the trained classifiers performs, use standard classification metrics such as accuracy, precision, recall, and the F1-score. To summarise the errors for each class, use the confusion matrix.

The SciKit-Learn library provides built-in methods to automatically calculate these measures by comparing the predicted labels with the provided ground-truth labels. Click on the following link to find these methods and import them:
https://scikit-learn.org/stable/modules/model_evaluation.html

For each classifier and training sample size, show the values of all of the abovementioned standard classification metrics and the confusion matrix in your Jupyter notebook.

Coding Requirements

Make sure that in your Jupyter notebook all outputs and other requested results are displayed in the notebook environment. All cells in your notebook must have been executed so that the tutor/marker does not need to execute the notebook again to see the results.

Released: 30 September 2024