Machine Perception
COMP3007

ASSIGNMENT

Due Date: Week 13 - Friday 30 October 2019 at 5pm.
Weight: **30%** of the unit mark.

Note: *This document is subject to corrections and updates. Announcements will be made promptly on Blackboard and during lectures. Always check for the latest version of the assignment. Failure to do so may result in you not completing the tasks according to the specifications.*

Your total score for this assignment and Assignment 1 will need to be at least **15 marks** out of the total **50** marks for these two assignments. You will fail this unit if you cannot meet this basic pass mark, regardless of your scores in the mid-semester test and final exam.

1 Overview

This assignment provides an opportunity for you to demonstrate how you can use what you have learned from lectures and practicals to develop computer vision algorithms for a real world application problem. For a successful completion of this assignment, you need to understand the fundamental algorithms covered in the lectures, conduct some research into the machine perception problem to design suitable features and classification methods, and use the skills that you have developed through completing practical exercises to build essential components of a computer vision algorithm. External codes, except for the software in the provided VM, are NOT allowed to use in your assignment. Feel free to use the work you have done in your practical exercises.

2 Background

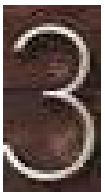
In this assignment, you will develop a computer vision program to detect and recognise the house numbers from images.

A typical machine perception approach to solving this problem would consist of at least the following

- Read an input image;
- Perform necessary image-processing operations;
- Detect and localise the house numbers;
- Segment the numbers into individual digits;
- Recognize individual digits and output the house number.

(a) Input Image

(b) Detected Area



(c) Extracted Digits

Figure 1: House number detection and recognition.

You will implement a suitable machine perception pipeline to perform the above steps by writing Python programs.

It is expected that you will make use of the computer vision algorithms discussed in the lectures and the skills acquired through doing practical exercises in order to complete the required task. You will also need to conduct your own research in order to understand different approaches to solving each task and decide your own choice of implementation according to the specific settings of this assignment.

3 The Task

Develop a program that reads in colour images from a specified directory. For each image, detect the digits, extract and classify the digits, and finally output the house number, which may include different number of digits. Your program is considered working if

- The detected area meets the following criteria
  - It must be a rectangle shape not exceeding the maximum allowable size, which is specified as follows
    - The overlap of the detected area with the ground truth must contain at least 50% of the detected area and contain at least 50% of the ground truth area.
  - It must contain all the digits of the house number.
- It must extract all the digits;
- It must recognize the digits correctly.

An illustrative example for house number detection and recognition is provided in Figure 1.

4 Specifications and Marking Guide

4.1 Report: 50%

A written report must be submitted, in PDF format, to Blackboard by the due date. This submission must contain

- A completed assignment cover sheet
- Printout of your source code
- A document that includes:
  - Statements on how much you have attempted the assignment.
  - The detail of your implementation for the task: this must clearly indicate your approach, the features you extract, the methods you use for detection, segmentation, and digit recognition, etc. It must allow the marker to understand how you approach the machine perception tasks.
  - The performance of your program on the validation set for individual tasks.
  - Supporting diagrams, figures, images that help describe your programs clearly.
  - References that your implementation is based on, or inspired from.

Your report will be marked based on: 1) the clarity and presentation(20%); 2) the proposed methods and the judgements of your design(50%); and 3) experimental results on the validation images and discussions(30%). In your report, you can also report multiple different pipelines you have implemented, even they may not work well, and compare their performances.

4.2 Implementation: 50%

Your implementation will be marked based on the quality of your code (30%) and the testing performance (70%). Your codes are expected to be well written with comments and good structures. If you have implemented multiple pipelines, please specify the best pipeline. Only the specified pipeline will be used to test the performance which will be used to give marks.

Important: *For the evaluation of your codes, the marks will be given based on the outputs within 1 minute of running your program for the task. If your program cannot complete the task after a period of **1 minute**, the program will be terminated. Note that the one minute limit is on the testing of the six images. Don't include unnecessary training codes in the Bash script task.sh.*

4.2.1 Evaluation Environment

- *Your implementation will be tested using a virtual machine, which has been provided for your practicals.*
- *Programming language: Python (only the versions found in the provided virtual machine are accepted).*
- *Data set for training: the training dataset for you to build your model and test your programs will be uploaded in Blackboard.*
- *Data set for testing: six images will be used to test the performances of your programs.*
- *Validation data sets: a separate validation set for the task that simulate the test images - so that you can estimate how your program will perform - will be provided in Blackboard. You will need to test your programs on the validation data sets, and include the results in your reports.*

4.2.2 Your electronic submission

- *Your electronic submission to Blackboard must be a compressed file (zip) with the following naming convention*

[surname]_[given names]_[student ID].zip

For example, if your name is Mike Jordan and your student ID is 123456 then your compressed filename is

jordan_mike_123456.zip

- *Your files will be uncompressed to the following directory (suppose that you are Mike Jordan as shown above)*

  /home/student/jordan_mike_123456

  I will refer to this top directory as `$SUBMISSION_DIR` *hereinafter.*

- *In this directory, I will expect to find the following Bash script*

  – `task.sh`

  This script must set up suitable environment variables and invoke your programs to perform the corresponding tasks. Do not expect the virtual machine for testing your assignment to have the same `bashrc` *file and/or other environment variables as found on your own copy of the VM! Make sure that these scripts can be readily executed by the user* `student`*.*

- *Your electronic submission should contain only*

  – *Source code*
  – *Bash script (described above)*
  – *Compiled programs (possibly more relevant for people using Java)*
  – *Necessary files for your models*

  To save space, your submission must not contain

  – *Temporary files that are not necessary for running your programs*
  – *The original training images.*

4.2.3 Input

Important: *Input images are placed under specific directories outlined below. Use absolute path to retrieve them.*

Testing Images: *The images used for testing your implementation will be located in the following directory:*

/home/student/test

The images used for testing will not be available until all the students have submitted their assignments. To test whether your implementation works as expected, you can copy the images used for validation into the test directories and change the file names accordingly.

The performance of your implementation will be tested only on the images for test. However, the performances of your programs on the images for training and validation may be tested to verify the accuracy of your reports.

In case that your programs need the images for training and validation to test their performances on the testing images, please refer to the following directories for these images

Training Images: *The training images provided in Black Board will be stored in the VM under the directory*

/home/student/train

Validation Images: *The validation images provided in Black Board will be stored in the VM under the directory*

/home/student/val

4.2.4 Output

Important: *The output files produced by your programs should be placed under the* `output` *in* ***your*** *submission directory* `$SUBMISSION_DIR/output/`*, e.g.*

/home/student/jordan_mike_123456/output

Specifically, the output files should be produced as follows:

- *All output files should be written to* `$SUBMISSION_DIR/output`*. For every input image* `testX.jpg` *your program must produce*
  - *An image file* `DetectedAreaX.jpg` *that shows only the detected area containing the building number.*
  - *A text file BoundingBoxX.txt which reports the bounding box, with four items* (x, y, w, h)*, where* (x, y) *is the top left corner, and* w, h *are the width and height respectively, for the detected area containing the building number.*
  - *A text file* `HouseX.txt` *which reports the building number. For the example in Figure 1, the correct txt file should read as "Building 803".*

4.2.5 Your demonstration

A demonstration session will be conducted during a practical to verify your submission. You will be asked questions about your programs. The purpose of this demonstration is to make sure that your submission is your own work and you know exactly what you are doing.

5 Submission

You are required to submit your assignment, including your written report and source code, by Friday 30-Oct-2020, 5:00pm Perth time (Week 13).

Upload your submission electronically via Blackboard, under the Assessments section.

You are responsible for ensuring that your submission is correct and not corrupted. You may make multiple submissions, but only your newest submission will be marked.

You will need to make yourself available for the demonstration session. Exact date and time will be announced on Blackboard.

The late submission policy (see the Unit Outline) will be strictly enforced. A submission 1 second late, according to Blackboard, will be considered 1 day late. A submission 24 hours and 1 second late will be considered 2 days late, and so on.

You must also submit a completed, signed ***"Declaration of Originality"*** *form.*

6 Academic Misconduct – Plagiarism and Collusion

Please note the following, which is standard across all units in the department:

Copying material (from other students, websites or other sources) and presenting it as your own work is plagiarism. Even with your own (possibly extensive) modifications,it is still plagiarism.

Exchanging assignment solutions, or parts thereof, with other students is collusion. Engaging in such activities may lead to a grade of ANN (Result Annulled Due to Academic Misconduct) being awarded for the unit, or other penalties. Serious or repeated offences may result in termination or expulsion.

You are expected to understand this at all times, across all your university studies, with or without warnings like this.

END OF ASSIGNMENT.