Practical 1: Medical Imaging Practical

GD5302: Health Data Science Practice

Due date: Tuesday 5th March 2024 (05/03/2024) at 12:00 (midday) UK time
25% of the coursework grade.
Please note that MMS is definitive for weighting and deadlines, which, occasionally have to be changed.

Aims

The main aim of this practical is to process and analyse some medical images. You will load a series of medical images and perform some image processing and analysis tasks and write up a report on what you did and why you did it that way, along with the outputs of your experiments.

Task

A clinician has passed you some X-ray Computed Tomography (CT) scans (in DICOM format) from a couple of patients (patient A and patient B) who have signs of COVID-19, after testing one of whom has COVID-19 and one does not. The data can be found in data.zip on moodle. The clinician wishes to view the images they sent to you in a few different ways to understand the content of the dataset more and so that they are able to view and interpret similar images easily in the future.

Note that some parts may require you doing some research, part of this practical is to get you to be familiar with how to figure out how to find information for yourself.

This dataset of axial CT scans comes from the Medical Imaging Data Resource Center (MIDRC) - RSNA International COVID Open Research Database (RICORD) Release 1a and 1b - Chest CT Covid- (MIDRC-RICORD-1a and MIDRC-RICORD-1b) from:
`https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=80969742` and `https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=80969771`. These files are in DICOM format. Each patient has multiple scan slices (images) taken across their body.

You will need to solve the following tasks (written by you in python):

1. Load in the medical images provided to you in python. You may wish to use the pydicom library - see `https://pydicom.github.io/pydicom/stable/tutorials/installation.html` and `https://pydicom.github.io/pydicom/stable/auto_examples/input_output/plot_read_dicom.html`).
2. Display one of the CT image slices from the middle of the scan on screen, save this figure as a png and put this example image in your report.
3. Increase the contrast for the lungs so that the clinician can view the detail of the lungs easily. Display the same image but with a dynamic range suitable for the clinician to the view the detail within the lungs more clearly, save this figure as a png and put this example image in your report.

4. Calculate some statistics on the pixel value range including the min, max, mean and standard deviation of the pixel intensity values for one example image (chosen by you), across all images for each patient, and across both patients.

5. Plot a histogram of the pixel values for a single image (chosen by you) from the dataset. Describe what this shows and why. Also plot histograms of the pixel values across all images for each patient and across both patients.

6. Reconstruct an image in the coronal plane which is approximately in the middle of the patient, change the dynamic range as you see fit (and justify the range you used). Do this for both patients. Save these figures as pngs and put these images in your report.

7. Reconstruct a maximum intensity projection (MIP) image in the coronal plane for each patient. Save these figures as pngs and put these images in your report. Explain what these show.

8. Reconstruct a maximum intensity projection (MIP) image in the axial plane for each patient. Save these figures as a pngs and put these images in your report. Explain what these show.

9. Write a program to calculate the lung area (in pixels) for each image cross section across one patient. Explain your method and show the output in your report.

10. Decide on how you will evaluate this program (given that you're not provided ground truth data). Then evaluate your system and document this in your report.

11. Write a program to calculate the lung volume (in pixels) for each patient. Produce a table of lung volume size for each patient and show this in the report.

Each of these steps should be clearly explained in the report. In all cases, you should explain what you did, why you did it that way, what the results show, what the results mean and put these into context. Evidence and justify your arguments both with relevant citations and with the output of your analysis.

Try to keep the report informative and focussed on the important details and insights – the report also demonstrates an understanding of what is important. **There is a maximum page limit of 10 pages**, including figures but excluding references, note that this is a limit not a target. All figures must be referenced in the main body of the text, and have captions and must have legible axis labels.

Deliverables

Hand in via MMS:

- The Python code that you write (either as a .py (python script) file or a .ipynb (python jupyter notebook) file.
- The files for the generated figures (include the relevant task number in the filename for each figure).
- A report in PDF format which contains details of each step of the process, justification for any decisions you take, and an evaluation of the final analysis. This should also contain evidence of functionality (via your results in the report) and any notable figures you have produced with relevant citations throughout. Clearly link the text and files with the task numbers above.

Please create a `.zip` file containing all files and submit this to MMS in the Code1 slot, please also upload your pdf report to the Report1 slot. Please note that MMS is definitive for weighting and deadlines, which, occasionally have to be changed. Ensure you give yourself time to upload and download and check that you've uploaded the correct files, do this with enough time to fix if required and re-check before the deadline.

Marking and Extensions

This practical will be marked according to the graduate school mark descriptors. All documents relating to the mark descriptors (and their conversion to the 20 point scale) can be found on the graduate school webpages: `https://www.st-andrews.ac.uk/graduate-school/students/rules/msc-and-mlitt-documents/`.

For the mark descriptors see: https://www.st-andrews.ac.uk/assets/university/graduate-school/documents/GSIS%20Mark%20Descriptors.pdf. You should also be aware of the following marking guidelines for this practical:

- To get a mark **in the 0-3 band** is a submission which shows little evidence of any attempt to complete the work.
- To get mark **in the 4-6 band** is a submission which shows little evidence of any acceptable attempt to complete the work, with no substantial relevant material submitted.
- A *partial working implementation* **in the 7-10 grade band** is a submission which shows evidence of a reasonable attempt addressing some of the requirements, or is accompanied by a very weak report which does not evidence good understanding.
- A *basic implementation* **in the 11-13 band** is a submission which achieves a solution in a straight-forward way and contains some evaluation, but is lacking in quality and detail, or is accompanied by a weaker report which does not evidence good understanding. Perhaps only completing tasks 1-8.
- An implementation **in the 14-16 range** should complete all parts of the specification, consist of clean and understandable code, and be accompanied by a good report which clearly describes the process and reasoning behind each step and contains a good discussion of the achieved results including graphs and evaluation measures. You may not have a fully working solution for tasks 9-11 but you have attempted this and suggested a sensible approach.
- To achieve a grade of **17 and higher**, should have excellent justification and experimentation into the methods used with relevant citations linking with the literature. You must have completed all tasks.

Note that the goal is *solid methodology and understanding* rather than a collection of extensions – a good scientific approach and analysis are difficult, whereas running many different algorithms on the same data is easy. Be thorough in your solution and strengthen your basic argument and methodology.
Also note that:

- We will not focus on software engineering practice and advanced Python techniques when marking, but your code should be sensibly organised, commented, and easy to follow. The result (outputs) of your code must be in the pdf report to evidence that your code worked.

- Overlength penalty: Scheme A, 1 mark for work that is 10% over-length, then a further 1 mark per additional 10% over. See https://www.st-andrews.ac.uk/policy/academic-policies-assessment-examination-and-award-coursework-penalties/coursework-penalties.pdf
- Lateness penalty: 1 mark per 24-hour period, or part thereof. See https://www.st-andrews.ac.uk/policy/academic-policies-assessment-examination-and-award-coursework-p coursework-penalties.pdf
- Details for good academic practice are outlined on the University webpages here: https://www.st-andrews.ac.uk/students/rules/academicpractice/
- For more details on Graduate School penalties, extension request etc, please refer to https://www.st-andrews.ac.uk/graduate-school/students/rules/
- Any use of AI tools, including large language models such as ChatGPT, needs to be acknowledged, referenced, and logged. If used, text generated by AI should be in quotation marks and referenced as private communication. Code and its comments need to be clearly highlighted and referenced. All AI interactions used for coding, or for report writing should be annexed to the submission as a searchable text file.