

Practical 1: Processing Fundamentals

ID5220: Biomedical Imaging and Sensing

Due date: Thursday 22nd February 2024 (22/02/2024) at 12:00 (midday) UK time
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 have experience programming and explaining the fundamentals of processing of medical signals. You will load some signals and images and perform some signal processing 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

The scenario is you are working for a medical technology start up. Their mission statement is to improve patient well-being through image and signal processing. As phone cameras are cheap and readily available in many communities these have been identified to be a good target acquisition device to collect medical information. The company wants you to create some code and a report around how to achieve these tasks (along with the advantages and limitations of the approach used and outputs of prototype code).

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.

The company has identified two main targets:

- (a) Measuring heart rates using data from an optical sensor.
- (b) Measuring heart rates using the video feed of a camera.

Optical sensor: Create a method to automatically calculate the heart rate from some time series data from the output of an optical sensor. Provided is a table (approximately thirty seconds long) of the intensity of light reflected from a finger over time, the file for you to analyse is a comma separated values (csv) file with two columns: time (seconds), followed by amplitude of signal for each time point (data in the file `heart_beat_signal.csv`). Explain the process you take (justifying your decisions) and show and explain your results. The way you do this is up to you, to load the data you may wish to use the inbuilt csv library or the pandas csv reader (`pandas.read_csv`) or the numpy csv reader (`numpy.genfromtxt()` where you set the delimiter to a comma).

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

1. Display a plot of the signal over time (in seconds).
2. Calculate the heart rate derived from the video using a script you write, the method you use is up to you (you may wish to use a Fast Fourier Transform).

Heart rate from video: Create a method to automatically calculate the heart rate from a video of a finger placed on the camera lens. Provided is a video (approximately a minute long) of a persons finger (video_of_finger.MOV). Explain the process you take (justifying your decisions) and show and explain your results. The way you do this is up to you, you may wish to use a python version of OpenCV, perhaps with pyvideoreader <https://pypi.org/project/pyvideoreader/>.

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

1. Display a plot of the mean grayscale value from within each frame over time (in seconds).
2. Calculate the heart rate derived from the video using a script you write, the method you use is up to you (you may wish to use a Fast Fourier Transform). You can use the method you used for the optical sensor task.

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. But do not include the original datasets or linked software.
- 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. Your report should include a brief introduction followed by the methods results and discussion for both main tasks detailing the subtasks for each along with a final conclusion.

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 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. Perhaps only one of the main components is attempted however it's not done well.
- 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 you have achieved only one of the two main sections.
- 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 are expected to try all tasks and have these written up but perhaps the final task is not fully complete.
- 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 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-penalties/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.