PHYS5020 SimpleITK and MRI Assessment

In this assignment, you will write Python code to process MRI images provided to you in DICOM format. These images were acquired as part of regular MRI scanner QA on a Large ACR phantom. You will first write code in Python to read DICOM files, extract relevant DICOM tags and perform four QA tests. The ACR MRI QA manual is provided to you and contains details of the image series and slices as well as the formulae that are to be used to perform the tests. The code framework is already provided to you with pseudo code describing the required steps. Complete missing code as needed. You can also write your own code from scratch if you wish. Complete the questions below and upload your answer sheet on Canvas. **Please do not submit your code**.

Requirements:

Python 3.x installation

Libraries - numpy, scipy, SimpleITK, os, glob, tqdm and matplotlib

Python code:

You are provided with the code framework for this assignment. Some of the function definitions and some lines in the code have been taken out and are for you to complete. The code and data are provided in a folder structure as –

/Data/Localiser

/ACR\_T1

/ACR\_T2\_multiecho

/Site\_T1

/Site\_T2

/T2map

Code/Question1.py

/Question2.py

/Question3.py

/Question4.py

/Question5.py

/common\_functions.py

/main.py

Imaging data:

1. Localiser – This is a single, sagittal image slice acquired through the center of the phantom.
2. ACR\_T1 – This is an axial series of 11 T1-weighted images acquired using ACR scan protocol.
3. ACR\_T2 – This is an axial series of 44 T2-weighted images acquired using the ACR scan protocol. Each set of 11 T2-weighted images are acquired with a different TE. A total of 4 TEs are used.
4. Site\_T1 – This is an axial series of 11 T1-weighted images acquired using a site-specific protocol.
5. Site\_T2 – This is an axial series of 11 T2-weighted images acquired using a site-specific protocol.
6. T2map – This is a T2 map calculated by the scanner using data from #3.

Questions:

1. **(2 marks) Read DICOM images and metadata** - Read the images and extract information about each dataset. Get the header/metadata from the first image in each dataset. Much of the code for this part is already given to you. The main function for you to complete is the ‘get\_dicom\_series’ function in common\_functions.py. You will need this function for subsequent questions, so make sure it works correctly.

The code to print the required DICOM tag values has already been completed for you. Fill in the table below. Note: Use an appropriate number of significant figures for each entry.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Series name (folder name) | Sequence name | Slice thickness (mm) | Matrix size  (rows x columns) | Field of view (mm x mm) | Spatial resolution (mm x mm) |
| e.g. ACR\_T1 (add rows as needed) |  |  |  |  |  |

|  |
| --- |
| Marking criteria:   * 1 mark for completing the table correctly * 1 mark for appropriate significant figures on all entries |

1. **(6 marks) GEOMETRIC ACCURACY** –
   1. Measure the dimensions of the phantom (length and diameter). Following instructions in the ACR manual, complete the given Python code and enter the measurements in the following table. Note that the given code does not use the ACR approach for measurement of the phantom dimensions, which requires manual drawing of lines on the phantom. Note: Use appropriate number of significant figures to report measurements.

Length (known = 148 mm):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Series name | Slice number | Measured length (mm) | Error (mm) | Test pass/fail |
| (Add rows as needed) |  |  |  |  |

Diameter (known = 190 mm):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Series name | Slice number | Measured length (mm) | | Error (mm) | | Test pass/fail |
| vertical | horizontal | vertical | horizontal |  |
| e.g. ACR T1 | 1 |  |  |  |  |  |
| (add rows as needed) |  |  |  |  |  |  |

* 1. Name one potential source of the geometric accuracy error. Explain why this occurs.

|  |
| --- |
| Marking criteria:   * 3 marks for completing the tables correctly (1 mark for length, 2 marks for diameter). * 1 mark for appropriate significant figures on all entries * 2 marks for answering Question 2.ii correctly |

1. **(3 marks) IMAGE INTENSITY UNIFORMITY** –

Complete the given code to measure the percentage integral uniformity (PIU) in the recommended image series/slice (check the ACR manual). Note that the given code does not use the ACR approach for calculation of the PIU, which requires manual selection of ROIs. Use the test criteria given in the manual to determine if the given dataset has passed/failed the test. Fill in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Series name | Slice number | ROI area | PIU (%) | Test pass/fail |
| (Add rows as needed) |  |  |  |  |

|  |
| --- |
| Marking criteria:   * 2 marks for completing the tables correctly * 1 mark for appropriate significant figures and units |

1. **(5 marks) T2 MAPPING** – This is not an ACR specified test!!!!
   1. Calculate a T2 map using the given Python code. Use the signal equation,

Image intensities at each of the echo times are fitted to the above exponential decay curve whose time constant is estimated as the T2. This is performed voxel by voxel to create a map of T2 values. The dataset also contains a T2 map that was calculated by the scanner. Compare your calculated T2 map with the provided T2 map.

Display the two T2 maps as subplots in a single figure, with appropriate labeling and a colorbar. Choose an appropriate range of values to display the maps.

* 1. Choose a single image slice. Extract the ROI statistics within a circular region in the image slice in both T2 maps and fill the following table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Slice number | ROI diameter (specify units) | Mean | Standard deviation | Median | Inter-quartile range (IQR) |
| Calculated T2 map |  |  |  |  |  |  |
| Given T2 map |  |  |  |  |  |  |

|  |
| --- |
| Marking criteria:   * 2 marks for display of the subplots * 0.5 mark for appropriate labelling and colorbar * 0.5 mark for appropriate display range * 2 marks for completing the ROI statistics table |

1. **(4 marks) PERCENT SIGNAL GHOSTING** –
   1. Complete the given code to measure the percentage ghosting ratio in the recommended image series/slice (check the ACR manual). Use the test criteria given in the manual to determine if the given dataset has passed/failed the test. Fill in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Series name | Slice number | Ghosting ratio (%) | Test pass/fail |
| (Add rows as needed) |  |  |  |

* 1. Display all the masks/ROIs used in this test overlaid on the image slice. HINT: Use matplotlib’s imshow function and two different colormaps for the image and masks.
  2. For the sequences used for ACR phantom imaging, ghosting always occurs in the phase encoding direction. On the image displayed in 5.ii., which is the phase encoding direction (horizontal or vertical)? How did you know?

|  |
| --- |
| Marking criteria:   * 1 mark for completing the table with appropriate significant figures * 1.5 marks for displaying the image and overlaid masks * 0.5 mark for identifying the phase encoding direction * 1 mark for explaining reasoning on PE direction |