CS463/516 Medical Imaging

Assignment 4 – **denoising + segmentation** *Due Monday July 20th at 11:55 pm*

Assignment is to be done individually or in pairs (2 people). Groups of 3 or more are not permitted.

Submission format: one partner should submit a .zip file to moodle with:

1) pdf containing images and descriptions showing you have completed all parts of the assignment/bonuses
    a. should also include most important code segments in pdf, so I can easily evaluate your algorithms
2) full python source code **in .py format**
3) A README file describing any incomplete parts of the assignment and your group member names.

Goals: implement a few simple denoising and segmentation algorithms in python, and gain experience using more advanced techniques on real-world datasets.

Part 1 - 35%: denoising: in python implement your denoising method of choice, from the following options:

1) Bilateral filtering
2) Non-local means
3) Denoising autoencoder (may use Keras or any other deep learning library)

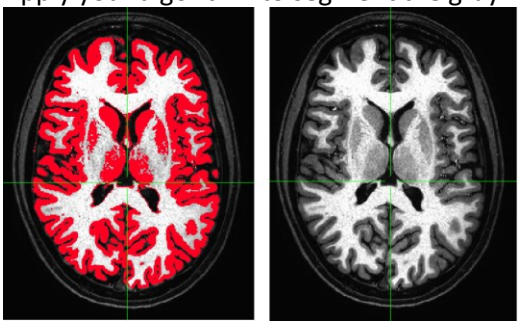
You have been provided with 5 test images. Use the method from lecture 13, slide 7 to estimate SNR in each of the images, and report these values in your pdf.

Once you have implemented your denoising technique of choice, apply it to the 5 images given in the handout. In your pdf, show the noisy, denoised, and noisy minus denoised (method noise) image for your algorithm of choice, for all test images provided (similar to lecture 13, slide 21).

Part 2 - 35%: segmentation: in python implement your segmentation method of choice from the following options:

1) Otsu's method
2) Watershed transform or region growing
3) Mean-shift clustering
4) Graph-cut
5) Neural network (any type, may use Keras or any other deep learning library)

Apply your algorithm to segment the gray matter in the 5 images provided (example below)



Gray matter segmentation shown in red (left), original image on right

In your pdf, show the original images and the binary segmentation side by side (or show the segmentation overlaid on the original). Comment on the performance of your algorithm. Where did it fail and where did it succeed, and can you explain why?

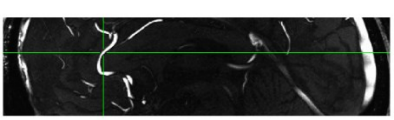
* for parts 1&2, do *not* simply call opencv or scipy, or other python library. You must implement the algorithm yourself using primitive array operations in numpy. I will check your code line by line in the .py file you submit, to verify this*

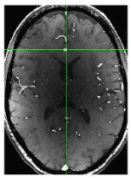
Part 3 – 30%: practical challenge (vascular segmentation)

Segmentation of the arterial and venous cerebrovasculature is a challenging task, often requiring multiple steps of denoising and enhancement, which will vary according to many factors including image acquisition method, voxel size, and SNR. Here, you are provided with two images (swi.nii and tof.nii). you must segment the arteries from the tof, and the veins from the swi. You have complete freedom in how the segmentation is performed (can use any existing library or technique). However, there is no ground truth available so you will need to use unsupervised or semi-supervised techniques. Be sure to include all code and a detailed explanation of how/why you took the steps you did to perform the segmentation (any denoising, enhancement, segmentation algorithm, etc.)

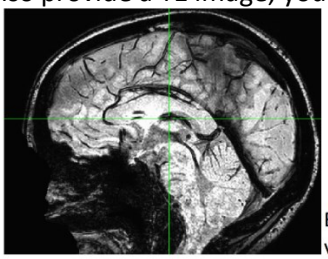
Link to data: https://drive.google.com/file/d/1igX2545uqTnasMoxvPvnNGx-JwewNIMD/view?usp=sharing

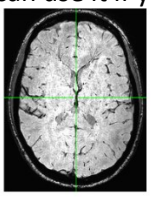
Example of the SWI and TOF images below: (I have also provide a T1 image, you can use it if you feel it will be useful)





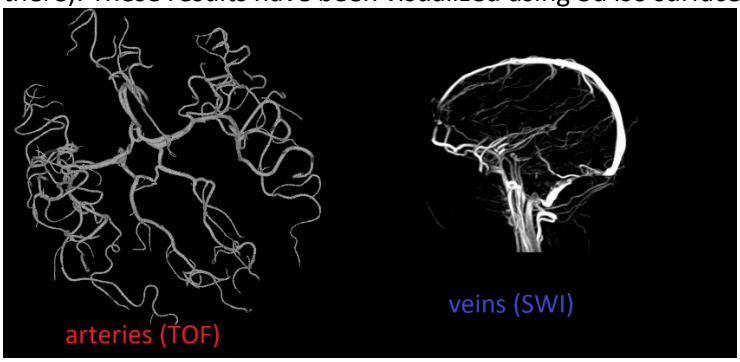
A) time-of-flight (TOF) arteries are bright



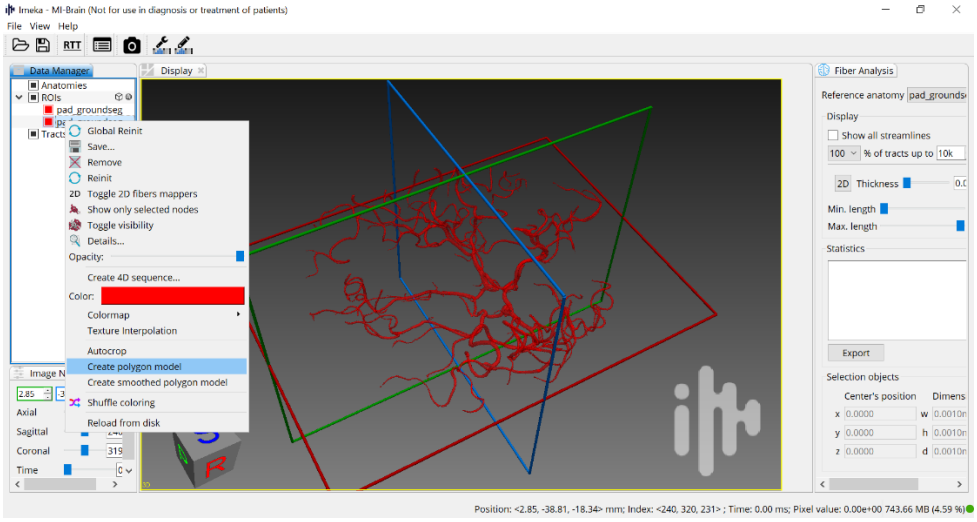


B) susceptibility-weighted (SWI) veins are dark

Here is an idea of how the segmentations should look (your results may differ, but the major structures should be there). These results have been visualized using 3d iso surface, so they can be viewed in 3d.



To visualize your results in 3d, the simplest and best platform is mibrain, from Imeka. This software was actually created in by a local company from Sherbrooke (Imeka). https://www.imeka.ca/mi-brain/ simply download and run mibrain, open your segmentation file (Saved in .nii format) and right-click, then select 'create polygon model' which will generate a 3d representation (see below).

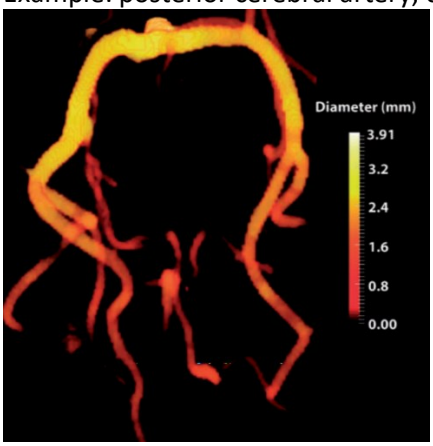


Do this for both your tof and swi segmentations, and show the final segmentation for both swi and tof in your pdf.

Bonuses:

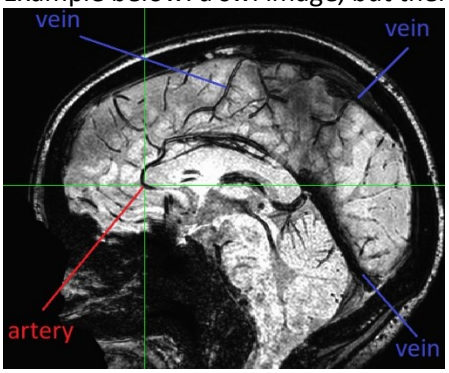
1) implement all segmentation and denoising algorithms from part 1&2 (using only numpy), show the results, code +8%

2) vessel diameters +8% – after segmentation in part 3, run a skeletonization algorithm (can use kimimaro skeletonization, or other) to get the skeleton of the segmentation. Then, for each point in the skeleton, find the diameter of the vein/artery to which this point belongs. Replace every voxel value in the original segmentation with the diameter of the vessel at that specific voxel.
Example: posterior cerebral artery, color coded by diameter. Larger vessels have larger diameter.



3) separating veins from arteries +5% – you'll notice that there are vascular structures which are present in both the swi and tof images. This is because neither swi or tof MRI acquisition techniques can completely image veins while suppressing the signal from arteries, or vice versa (image only arteries while suppressing signal from veins). Therefore, the tof segmentation will invariably contain some veins (it should be all arteries) and the swi segmentation will contain some arteries (it should be all veins). Can you think of a simple feature detection or shape recognition algorithm, or clustering approach, which will allow to recognize veins from arteries, and hence remove/keep them in the image during post-processing?

Example below: a swi image, but there is clearly an artery showing up as dark, along with the veins!



Example 2: a tof image, but there are some veins showing up as bright, along with the arteries!