CS463/516 final project – group size = max of 4 persons. Due Sunday August 15th at 11:59 PM.

*The final project is worth **40%** of your overall grade in the course. There will be no extensions on final project because I must submit your final grade by August 18th and I require 3 days to mark.*

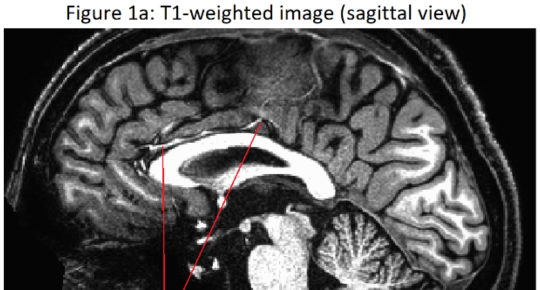
I will provide hints and live answers during breakfast clubs on July 28th, August 4th, and August 11th at 9:00 AM EST.

Final project: Deep learning for learning arterial segmentation from T1-weighted images

GOAL: apply deep learning (convnet) to segment arteries from T1-weighted image

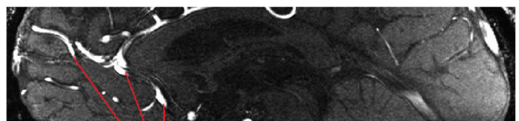
Overview: T1-weighted (T1) (Figure 1a) is a popular MRI modality used to capture contrast between gray/white matter boundaries in the brain. T1 is the most common structural MRI image and nearly every neuroimaging study will acquire a T1 as an anatomical reference. Time-of-flight (TOF) (Figure 1b) is another MRI modality, used to capture contrast between arteries and surrounding tissue. TOF is almost never acquired because most studies are interested in gray matter, and not the arteries. Nevertheless, recently there has been increased interest in arteries, because of disorders such as Alzheimer's and multiple sclerosis.

Figure 1a: T1-weighted image (sagittal view)



arteries (lightly visible on T1, not possible to segment using simple threshold)

Figure 1b: TOF image (sagittal view)



arteries (easily visible on TOF, can be segmented with simple intesity-based threshold)

If you go on openneuro and scroll through the datasets (over 550 datasets as of July 21 2021) you will see that nearly every dataset contains a T1, but only two datasets (forrest and midnight) contain TOF. To analyze arteries on *all* these datasets we need a way to segment the arteries from a T1 image *alone* (because most datasets contain T1 images, but not TOF images). Therefore, we must use one of the few datasets containing both T1 *and* TOF to a) generate a ground truth arterial segmentation (using TOF) and b) train a convnet to segment arteries based on T1w alone. we will then (hopefully) be able to apply our pre-trained model on the rest of the datasets which contain only T1.

You will apply machine learning to solve this problem of segmenting arteries from a T1. Convolutional neural network (convnet) is a specialized deep neural network architecture for image analysis.

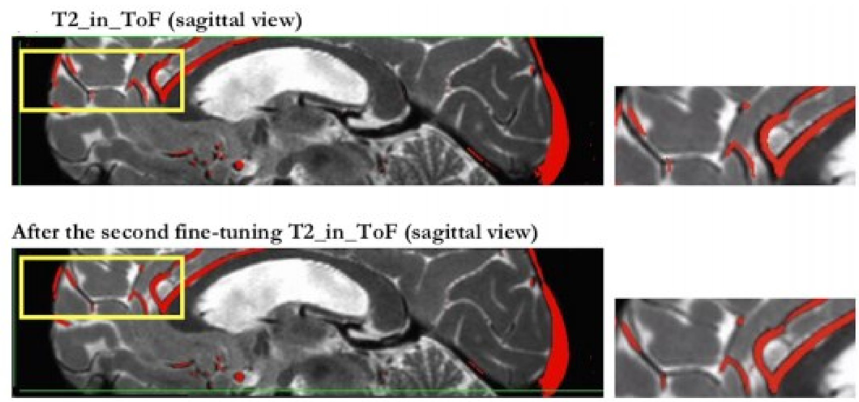
As you can see from Figure 1a, the arteries are visible on T1, but cannot be isolated through a simple threshold due to lack of contrast. *You must train the convent to segment the arteries from the T1 by using TOF as ground truth.*

Ground truth generation steps:

- Download data from the forrest link above (figure 2). Get all subjects (subjects 1-20) both TOF and T1. The TOF is called sub-N_ses-forrestgump_angio.nii.gz and the T1 is called sub-N_ses-forrestgump_T1w.nii.gz
- For each subject, register the T1 to the TOF. Be sure to visually inspect the registration result, because the registration must be PERFECT (figure 3) in order for the convnet to learn features relevant to segmentation.
  - Registration from T1 to TOF is challenging and might not work using a simple command like
    ```
    flirt -in t1.nii.gz -ref tof.nii.gz -dof 6 -omat t1_2_tof.m -out t1_in_tof.nii.gz
    ```
    (you can try the above command, but it will probably fail on most subjects, because the images are very different in both their contrast and the field of view. Instead, you will need to find some way to perform a preliminary segmentation on TOF and T1 to isolate the same tissue class in both images, apply flirt using the segmentations as input and reference, and the use the output matrix later, on the raw images.

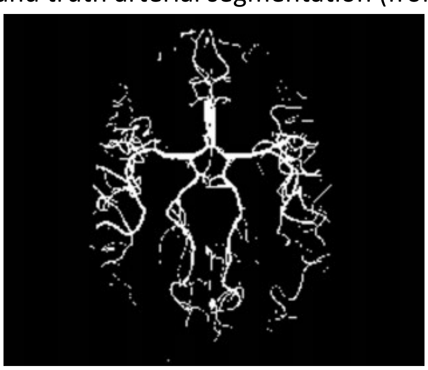
Figure 2: forrest_gump dataset from openneuro (subject 16)

Figure 3: registration must be <u>perfect</u> to create ground truth! The TOF segmentation is overlay in red, the T2 is underlay (Same principle for the T1). Clearly, the registration is not perfect in the top image, which makes the ground truth incorrect. The accurate registration is shown in the bottom image.



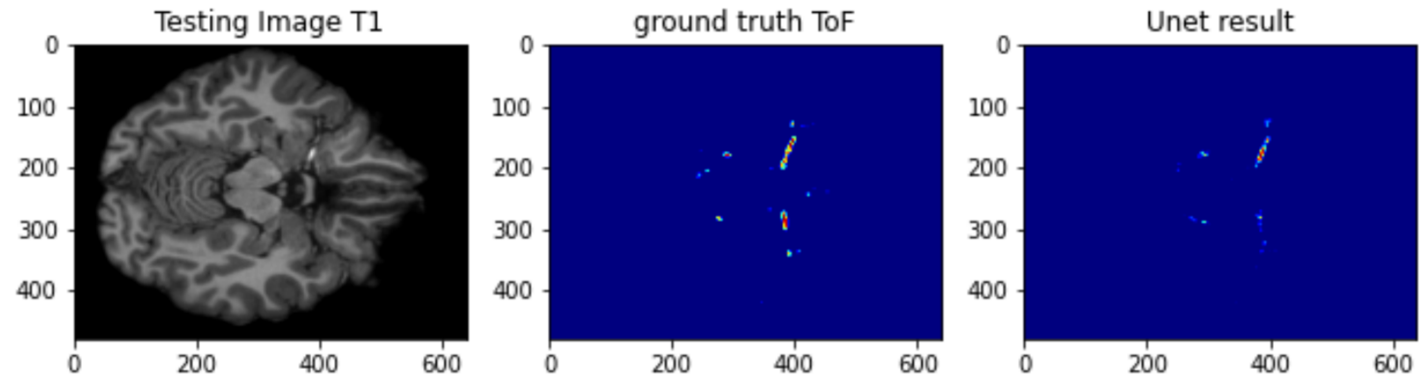
- Once you have T1 and TOF in same space, segment TOF by applying an arbitrary threshold.
- Skull strip the unregistered T1 (before registering to TOF) and bring the resulting brain mask into TOF space using the matrix from T1 to TOF registration. Alternatively, you can try computing brain mask in TOF space, but it is more challenging because the brain is not the usual shape due to the limited field of view in TOF which cuts out lower/upper parts of brain (BET will fail on this type of truncated image).
- Use brain mask to mask thresholded TOF, this gives your ground truth, which is arteries inside the brain only. It should look something like below (axial view).

Ground truth arterial segmentation (from TOF)

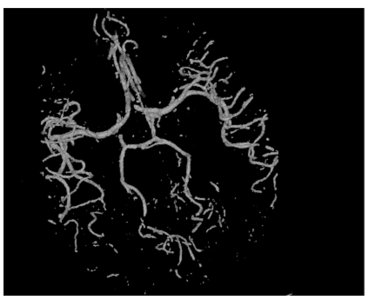


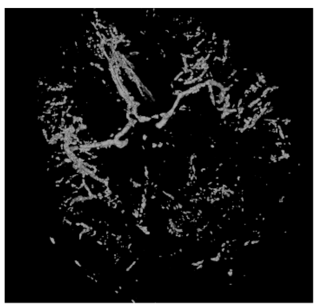
Deep learning steps (after ground truth is complete):

- Read the lectures on ML and deep learning. Choose your environment (I recommend Keras, but you can use pytorch or any other deep learning library).
- Go to google colab, open an account, and select GPU in settings.
- Create a python script to load the ground truth (TOF segmentation) and training images (T1_in_TOF).
- Set up a deep model in your script. I recommend the Unet model which is a popular convnet adapted for medical imaging domain, but there are a wide range of models to choose from. Your model can handle 2D or 3D images (recommended to handle 2D axial images) but if you go for 3D it will take more memory.
- Train the model on subjects 1-15. test it on subjects 16-20. Show the arteries from T1 as your final result, it might look like the following (if shown in 2D):



Expected output: a report showing all your relevant code and preprocessing steps, with images of each step. Show a 3d rendering of the ground truth and the T1 side by side (*it might look something like below, but hopefully you can achieve an even better result!*).


Ground truth (TOF segmentation)


Prediction from T1 on test subject

You must create the above figure for every subject in your dataset, show all 20 subjects, and indicate which subjects were used for the training, and which were used for the testing. Show also the loss curve from the training set as a separate chart in your pdf. Experiment with different cost functions and report your results.

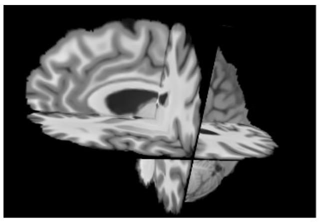
Notice how in the above image 'Prediction from T1 on test subject' much of the arteries are missing when compared to the 'Ground truth (TOF segmentation)'. This is not necessarily a bad thing because the T1 has inherently poor contrast between arteries and surrounding tissue, so even with an advanced deep learning approach such as convnets, it is very challenging to achieve accurate arterial segmentation from a T1. Nevertheless, your goal should be to improve on the above result, and achieve a prediction (from test subject) as close as possible to the ground truth.

BONUS +15%: surface-based brain visualization in Unity

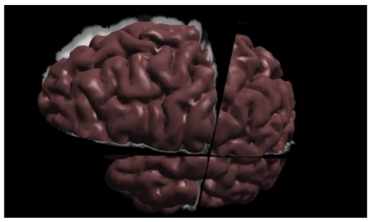
All code must be written for Unity in C#.

As a bonus, create a dynamic surface visualizer in Unity which can do the following:

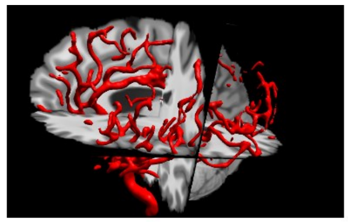
1) Load a nifti format image (code provided here) and display it as a 2D texture in 3 planes (optional, don't need to display as texture, but it's a good sanity check to make sure the image loaded correctly).
2) Using two adjustable sliders, determine the threshold for the image (two sliders for upper/lower thresh)
3) Once the threshold is set, generate a surface mesh for the image and display it in the scene
    a. The surface mesh can be generated from the volumetric data using marching cubes algorithm (or other similar algorithm). You can use any pre-existing marching cubes implementation to create your mesh.
4) Allow user to change the color of the mesh from (3) using some type of UI



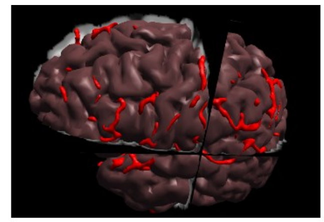
step 1: loaded nifti in Unity showing texture on 3 planes (coronal, sagittal, and axial)



step 3: displayed surface mesh (brown) showing the gray matter segmentation



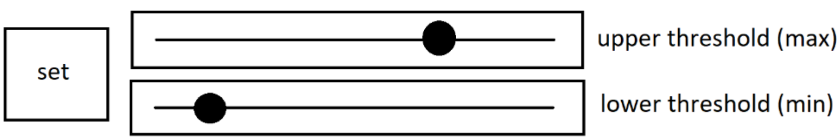
step 3: another surface mesh, this time from a TOF image, showing arteries in red



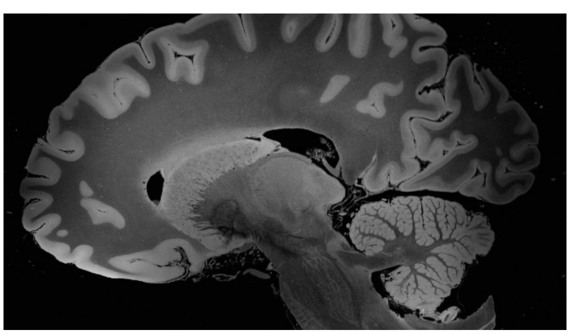
two separate surface meshes (arteries and gray matter) displayed in the same scene.

Your program should also support loading of multiple images (see above).

Basically your program will load the .nii image, apply a threshold using the slider (if necessary) and then run marching cubes to generate the mesh. Once the mesh is complete, it will be displayed in the scene. If the user changes the threshold using the slider, the mesh will be re-generated based on the segmentation derived from the new thresholds. An example of a simple UI is below, user adjusts threshold and then clicks 'set' to generate and display the mesh.



Expected output of bonus: provide a second pdf report (bonus report) containing all relevant code and screenshots. Provide a windows executable (built from Unity) that I can run to verify your program works as expected.



Bonus 2 (+2%): display the '7-Tesla MRI of ex-vivo human brain at 100 micron resolution' in your surface renderer: https://openneuro.org/datasets/ds002179/versions/1.1.0 . this is a massive image with extremely high spatial resolution (0.1 mm isotropic) so it should show some very nice details when you display it as a surface. You might want to pre-segment the image before loading it in Unity, to make it less memory-intensive.

https://www.sciencenews.org/article/mri-scan-most-detailed-look-yet-whole-human-brain