Registration option for final project:

I suggest doing a segmentation task for your final project. If you would much rather do registration, then start by examining 2D brain MRI registration from this tutorial:
http://tutorial.voxelmorph.net/
This tutorial assumes a 3D-registration deep-net is available for your use. Instead, you will implement your own (manual or deep net) 3D registration system. The tutorial demonstrates 2D brain registration, after which it shows you how to evaluate 3D registration. *You are NOT required to do any of the MNIST or 2D brain registration work.* Your only tasks are to develop your own 3D brain registration and to evaluate it using some of the evaluation methodology in the last sections of the tutorial.

Choices for segmentation projects:

I suggest choosing one of the following segmentation datasets for your project.

Dataset 1—Liver Tumor Segmentation:

https://www.kaggle.com/datasets/andrewmvd/liver-tumor-segmentation
This dataset was originally from the Liver Tumor Segmentation Challenge (LiTS17) of MICCAI 2017 + ISBI 2017. The dataset includes ~50 CT image volumes plus segmentations of the liver as well as tumor lesions. You could either take a ML approach using the entire dataset, or manually develop a classical approach using a few of the images. The choice is yours. *You must register with kaggle if you want access to this data.*

Dataset 2—NAMIC Lupus Brain-Lesion MRI for 5 patients:

https://slicer-packages.kitware.com/#collection/6260e399e8408647b387c9a2/folder/6260e3eee8408647b387c9d1
For each patient this dataset contains pre-registered T1+T2+FLAIR MRI images, with manual segmentation masks for both the brain and for the lesions inside. I suggest focusing on the lesions (e.g. lupus001_lesion_manual_reg.nii.gz). Examine the different types of MRI images to see which type(s) make the lesions easiest to see with the clearest boundaries around the lesions. Be aware that the masks may use higher pixel values (e.g. 100) to represent the foreground. I suggest you not go crazy, and start by either detecting OR segmenting a relatively easy set of lesions that have more clear boundaries.

Dataset 3—Large spine dataset

This dataset is very large, suitable for deep-learning. Here are the links for the paper and the data:
https://pubs.rsna.org/doi/10.1148/ryai.2020190138
https://osf.io/nqjyw/

- Note: Safari users may need to choose “View” → “Reload and Show IP Address”

I suggest choosing just a few images and expert annotations to use for your project (unless you’re training a ML/CNN system). You should provide a seed point for each vertebra to be segmented, for a few vertebrae of your choosing (at least 4 per patient for at least 3 patients). It's up to you whether or not you want the extra challenge of including vertebrae with screws, cracks, etc. in your test set (I recommend starting with healthy vertebrae, and only expand to challenges if you have time). Don't worry about trying to automatically label which vertebra is which.

Tasks for a segmentation project:

Task 1: How to evaluate?

Decide how to compare your algorithm's segmentation to each of the experts. Do you assign each pixel with a majority vote from the radiologists? Do you instead compare to each radiologist, knowing you can't perfectly match all 3 because they're different? Do you try to somehow take into account where the radiologists did-vs-didn't agree, such as with statistics, multiple error metrics, etc.?

Task 2: How to segment?

Implement a segmentation algorithm/system that can "accurately" segment the target of your dataset (e.g., brain anatomy, liver tumor., etc.) Your code should take as input the coordinates of a single seed pixel somewhere inside the target of interest. Use your evaluation from task 1 to optimize your algorithm, manually and/or automatically adjusting parameters, trying different filters, etc.

Task 3: Document results

Describe what you decided to segment, how your algorithms work, and why. Document the different things you tried. Clearly describe how your final system works. Create one or more graphs, tables, etc. of how your evaluation metric improved or varied with various parameters/choices in task 2. Provide pretty pictures comparing your algorithm's output to the human-expert annotations.

Stuff to keep in mind

To perform "substantial validation" you should carefully compare validation data with the results of one or more algorithms, using one or more parameter sets. As an example, if you have a favorite segmentation algorithm that has 3 parameters, each of which you want to test with 4 different values, and you have 4 test images, then you would test your segmentation algorithm 256 different ways (3 different parameters, with 4 possible values for each, and 4 images = 4^3*4 = 256 possibilities). For each of the 256 tests, you would then compare the segmentation result against the "correct" segmentation for the specific image that you used. The comparison needs to produce a numeric score, and the comparison would almost certainly have to be automated. A naive and simple comparison would be to automatically count the number of pixels that overlap between your segmentation and the "correct" one, and then divide by the total number of pixels in the "correct" segmentation. Instead, I recommend using something more intelligent. A good starting place is the DICE comparison/similarity metric.