CS463/516 assignment 1

Due: Thursday, May 27th at 11:59 PM (Eastern time).
Submit: a pdf report showing your source code, displayed images, and any explanation/notes.

Topic: Basic numpy, medical imaging modalities, SNR, and denoising
In this assignment we will familiarize ourselves with basic numpy array manipulation, different imaging modalities, and gain understanding of some basic image features (contrast and SNR).

Setup

First, install spyder (or any python IDE). If you are on windows, you will probably want to install anaconda first: https://docs.anaconda.com/anaconda/install/

- Note – you don't *need* to use spyder (any python IDE is fine) but it will help a lot in this course because the ipython console is very useful for debugging and visualization of intermediate results.

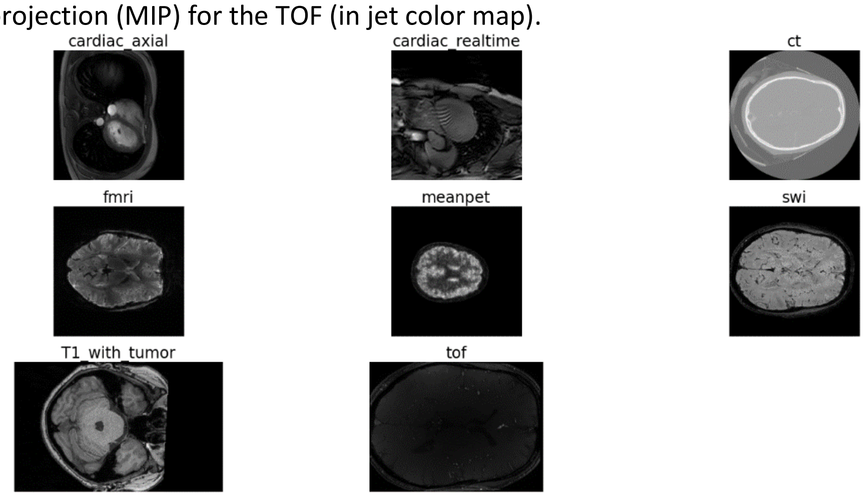
Second, get the images (link below). This directory contains 8 images:

cardiac_axial.nii.gz	4/27/2020 8:52 PM	WinRAR archive	10,204 KB
cardiac_realtime.nii.gz	4/18/2019 6:58 PM	WinRAR archive	14,366 KB
ct.nii.gz	5/3/2021 12:36 PM	WinRAR archive	53,569 KB
fmri.nii.gz	5/10/2021 12:40 PM	WinRAR archive	165,597 KB
meanpet.nii.gz	5/3/2021 10:38 AM	WinRAR archive	10,331 KB
swi.nii.gz	5/3/2021 10:57 AM	WinRAR archive	53,716 KB
T1_with_tumor.nii.gz	5/3/2021 10:26 AM	WinRAR archive	7,749 KB
tof.nii.gz	5/3/2021 10:57 AM	WinRAR archive	42,719 KB

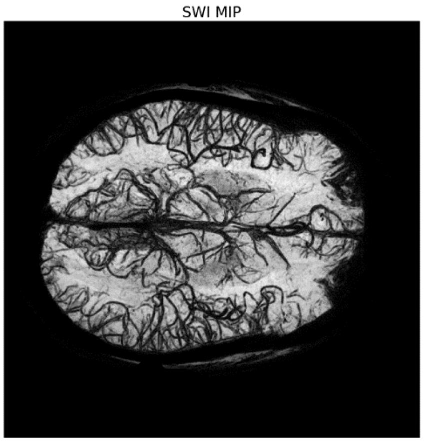
The images were all acquired using MRI scanner, with the exception of *ct.nii.gz* and *meanpet.nii.gz*, which are from CT and PET scanner, respectively. Most of the images are 3D, with the exception of *cardiac_axial.nii.gz*, *cardiac_realtime.nii.gz*, and *fmri.nii.gz*, which are 4D.
https://drive.google.com/file/d/10_bOHQmfe9WkTlK9Td5IhjVHhBp-s2ni/view?usp=sharing

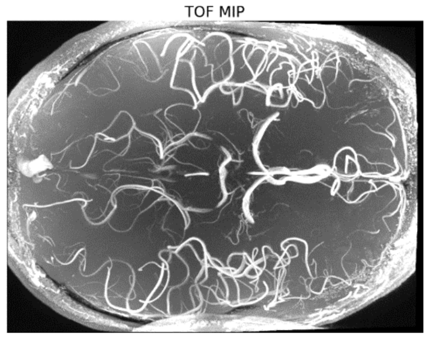
Part 1: (25%): simple plotting with matplotlib

a) Display all images middle z-slice (3rd dimension, axis=2) (as seen below). Use the 'jet' color scale (instead of the gray color scale that I use below in my example). Above each image, show the title of the modality. Remove the x/y axis labels (as I did below). Use `plt.subplot`
b) display the *minimum* intensity projection (MIP) for the swi.nii.gz, and the *maximum* intensity projection (MIP) for the TOF (in jet color map).



Left: part (a) – replicate this but use jet color map instead of gray.





Left: part (b). note how the blood vessels are displayed prominently due to the projection. You will need to restrict the z-slices from the SWI to achieve a good MIP. Use `np.min` and `np.max`.

Part 2 (25%): contrast estimation

Using numpy, get 3 different contrast measures for each image (root mean square, Michelson, and entropy, see **lecture 3 slide 4, 5**). Report the contrast (all 3 versions) in the title of the plots in figure 1a. base your contrast estimation on the entire 3D or 4D image (not just the slice shown in the figures).

Part 3 (25%): SNR estimation, quantifying noise

Using the method outlined in the **lecture 3 slide 7**, report the SNR for each of the modalities. Which modality has the highest SNR and which has the lowest?
Plot histograms of the noise in each image. What type of distribution does the noise follow?

To display the solution to part 3, create a new figure (as in part 1) and display the noise histogram of each image (instead of the image itself) in each sub plot. Show the SNR as the title above each histogram (along with the image name).

caution – when selecting your noise patch, be sure the patch isn't all zeros, otherwise your noise will be estimated as 0 and the SNR will be infinite

Part 4 (25%): linear filtering

Using the Fourier transform method shown towards the end of lecture 2 video, apply linear filtering to each image for $\sigma = 2, \sigma = 4$, and $\sigma = 15$. Create 3 new versions of the figure in part 1a, one figure for each sigma. Show the middle slice of the filtered image in all subplots.

Submission: Put all your figures in your pdf, along with the code used to generate them and any comments you have about the work.

<u>Bonus +5%:</u>
Make a python class that can display 3d and 4d images (scroll through the slices and time points), similar to AFNI's method for time series display (see lecture 1 at time 58:12).