FINAL PROJECT

In this practice you have to work with several images stored in NIFTI files. You will need the Matlab functions that are in NIFTI_reader.zip. Unzip this in a folder and add that folder to your Matlab path. This way you will be able to read the images. Some tips for reading these files:

The load_nii function returns an structure with the pixel data (3D matrix with intensities) and other information. An example use would be:

```
myImage = load_nii('filename.nii');

my3DVolume=myImage.img;

mypixelsize= myImage.hdr.dime.pixdim(2:4)
```

If you want to display your image in 3D, you can use the 3D Viewer included with this library:

```
view_nii(myImage);
```

This viewer can be useful to explore the images, but it cannot be used to display the results in your final project code.

Data provided:

- CT.nii: a thorax CT image from a lung cancer patient
- PET.nii: the corresponding PET image of the same patient, with the same dimensions and pixel size as the CT
- CT_lungs.nii: a binary mask of the lungs of the patient
- CT_lungs_noise.nii: a binary mask of the lungs of the patient

Notice that all images are 3D!

Exercise 1

Implement a function named `otsu_segmentation` that segments the original CT and PET images, separating patient data from the background (in the CT) and tumor from the rest of the image (in the PET). The function will perform this segmentation using Otsu method. You will need to program your own Otsu function that will obtain the optimal Threshold calculating the between class variance for all possible thresholds and selecting the one with maximum value. The only input for your function will be the 3D image, and the outputs will be the optimal threshold and the segmented mask (the image with value 1 for the object and 0 for the background). Take into account the following considerations:

- Write your own version of the histogram function inside your main code or as a separate function. *If your version of the histogram does not work, you can use Matlab histogram functions, but using your own function will increase your score +0.5/10.0.*

- You cannot use `graythresh`, `imbinarize` or other similar functions from Matlab in your final result (although you can test them against your results).
- You can use other software to check the Otsu threshold that can be obtained, but take into account that the specific value will slightly change depending on each implementation.
- Your code should work for both the CT and the PET (obtaining a different threshold for each image). Show the threshold that you obtain for each image (CT or PET) in your presentation, with a value in the same range as the original CT or PET image. Don't forget that the dynamic range of each image is very different, so your code should consider that to work correctly.

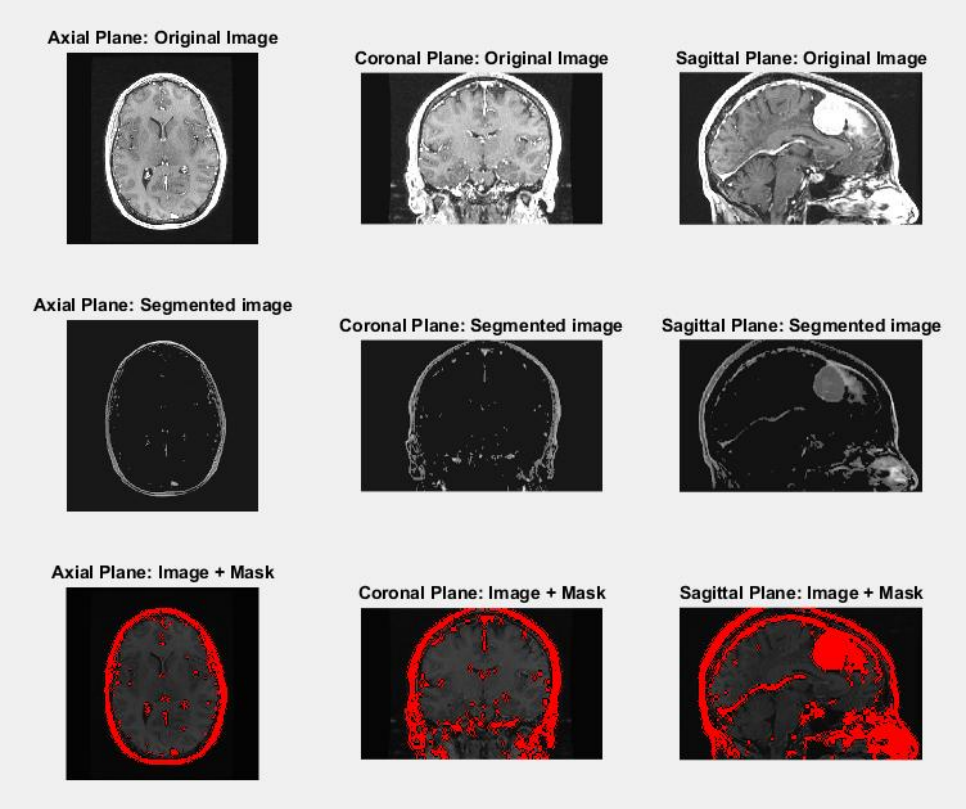
There are several ways to calculate the histogram and the Otsu threshold. Some of them are faster than others. If you have tested several implementations, present the computing time for each of them and give a brief explanation of your results.

Exercise 2

Write a script that calculates both segmentation masks with the function developed in Exercise 1, and displays the following results in a figure for each image:

- A first row with the original axial, sagittal, and coronal views of the 3D image. You can select the slice in the middle of the volume for each view. You should use the pixel size to display the images with the correct aspect ratio.
- A second row displaying the same axial, sagittal, and coronal views of the segmented image, with the same considerations as in the first row.
- A third row where you display axial, sagittal, and coronal views combining the information from the original image with the mask. The pixels belonging to the mask should be displayed in color (red or any other color), while the pixels not in the mask should be displayed with their original values.

The following image is an example of the result on an image that is not part of your project:



You may need to select specific Window/Level settings to improve your visualization.

Exercise 3

In this part, we will work with a segmentation mask provided for the CT image (CT_lungs_noise.nii). This mask includes the lungs of the patient, but as you can see it has some small holes inside the lungs. Implement some morphological operation (use the available Matlab functions `imerode`, `imdilate`, `imopen` and `imclose` for that) to remove as many wholes as possible from the segmentation mask (large areas such as the vessels don't have to be removed). Take into account the following considerations:

- Test several options for the parameters of your morphological filter and discuss the different results.
- You can use the mask image named CT_lungs.nii as a reference of the result you should obtain (it does not have to be exactly the same).

- Display some slices or views (axial, sagittal, coronal) of the original mask and the processed one to describe your results. Display also the mask on top of the original CT using the display functions from exercise 2.

Exercise 4

Now you will write the code to calculate statistics of the images in the different segmentation masks:

- CTmask: patient mask resulting from Otsu thresholding of the CT in Exercise 1.
- PETmask: tumor mask resulting from Otsu thresholding of the PET in Exercise 1.
- CTlungs_noise: original mask provided for the lungs.
- CTlungs_filtered: resulting from exercise 3 (with the morphological filter parameter that you consider the best for removing the noise inside the mask)

Write a Matlab function called `quantify_ct_pet` that takes a CT volume, the PET volume and the four segmentation masks as inputs shows the following results on the screen for every region (CTmask, PETmask, CTlungs_noise and CTlungs_filtered) both in the CT and the PET (include these results in your presentation):

- Total region volume (in cubic mm).
- Average gray value per pixel.

You need to calculate those two values for each image (PET and CT) and each mask. Discuss the values that you obtain during your presentation.

Exercise 5 (up to 2.5 extra points)

This extra exercise requires some exploration of the capabilities of 3DSlicer for medical image processing. The main purpose of this exercise is that you repeat the steps from the previous exercises 1…4 but using 3DSlicer software. The results that you can obtain are flexible, depending on how far you want to experience with 3DSlicer, but some ideas could be:

- Loading the data provided (from .nii files). Those 3D images that correspond to segmentation masks should be loaded as ‘Segmentation’.
- Create your Otsu segmentation from the CT and PET volumes.
- Display the different segmentation masks with different colors in 2D and 3D views.
- Remove the holes in the lungs as you did in exercise 2.
- Quantify the masks on the volume as you did in exercise 4.

Show screenshots of your different steps and discuss your experiences.

You may need to read some 3DSlicer documentation: https://slicer.readthedocs.io/en/latest/

If you want to export your results from Matlab and display them in 3DSlicer, you will need the functions provided in NIFTI_reader.zip. You will need to use first a function `make_nii` to create a structure from the image data and the pixelsize, and then save it using `save_nii` function.

RESULTS

You have to upload all the code that you wrote for the different exercises and a Powerpoint presentation to discuss in class, including all your results. Do not forget to include all your figures and values for the results in the presentation, since the professors will only run your code if something is not clear enough during the presentation. Provide detailed explanations of your tests, a good description of your results and any other ideas you want to present in your PPT. You have to upload the following to AG (all in a single ZIP file):

- All code produced. Take into account that the code should read the images from the same folder from which it is run; that will be the working directory in Matlab. This means that if the teacher puts all your code in a folder containing the working images and sets that folder as the Matlab working directory, every function should work.
- The PPT that you will use to present the results in class.

The evaluation of your group practice will be based mainly on the PPT and how you present your results in class.