Introduction to 3D slicer:

Chart, treemap chart

Description automatically generated

Layout icon: change the layout of four panels of images; “four up” is usually used.

Conventional: three axis at the bottom

Chart

Description automatically generated

A = anterior, P = posterior, R = right, L = left, S = superior and I = inferior.

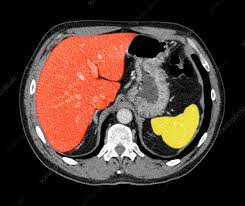
* Loading data: (drag and drop)
* Cropping (crop volume module and volume rendering module):
* Creating Label Map (Editor module: threshold effect and change island effect):
* Building Model:
* Saving Model:

Case study (**04/2022**) – CT of spleen:

<https://radiopaedia.org/articles/spleen-1>

<https://insightsimaging.springeropen.com/articles/10.1007/s13244-012-0202-z>

* Data from MSD; venous phase from patients undergoing chemotherapy treatment for liver metastases. ROI 🡪 spleen; selected due to large variations in the FOV
* Wedge-shaped, left upper quadrant (superior + left)



Liver (red) + spleen (yellow)

1. Resampling volume:

Module = “Resample scalar volume”

spacing = 1,1,1; interpolation = linear, NearestNeighbor; bspline …

input volume = its original file name

output volume = create your own name (better include new spacing info)

1. Cropping volume:

Go to “vendering volume” 🡪 use the ROI 3d box to select regions

Go to crop volume 🡪 apply

note: go to “Data” 🡪 turn on ROI to select regions

1. Segment editor: (at least **2 segmented** regions are required)

Add: edit name + double click to edit color (spleen == dark purple)

Create seeds: manually paint ROI in different views

Grow seed 🡪 initialize 🡪 apply

A picture containing dark, dessert

Description automatically generated

A close-up of a light bulb

Description automatically generated with low confidenceA picture containing indoor, white, close

Description automatically generatedA picture containing text

Description automatically generated

Case study (04/12) – mp-MRI of Brain tumor (4D volumes)

* Chosen due to complex and heterogeneously located targets
* 750 images

Resampling 🡪 registration 🡪 bias correction 🡪 skull stripping 🡪 normalization

mp-MRI: multiparametric MRI (native T1-weighted, post-Gadolinium contrast T1-weighted (T1-Gd), native T2-weighted, T2 Fluid Attenuated Inversion Recovery (FLAIR))

<https://case.edu/med/neurology/NR/MRI%20Basics.htm>

dim = (240, 240, 155, 4); sequence: T1 🡪 T1-Gd 🡪 T2 🡪 FLAIR

target ROI: edema (build-up fluid in tissue), enhancing, non-enhancing

* For peripherally enhancing masses, the main differential diagnosis lies between **high-grade and secondary brain tumors, inflammatory or demyelinating lesions and abscesses**. HGG (fast-growing, hard to treat)
* Non-enhancing lesions may represent **low-grade gliomas (LGGs), viral encephalitis and developmental anomalies**, such as focal cortical dysplasia. LGG (longer-term survival in younger patients)

Question: seemed already skull stripped ?????

A picture containing calendar

Description automatically generated

A picture containing graphical user interface

Description automatically generated

DICOM: digital imaging and communications in medicine

NIfTI: neuroimaging informatics technology initiative

**CT windowing:** <https://www.kaggle.com/code/redwankarimsony/ct-scans-dicom-files-windowing-explained/notebook>

Windowing, also known as **grey-level mapping**, **contrast stretching**, **histogram modification** or **contrast enhancement** is the process in which the CT image greyscale component of an image is manipulated via the CT numbers; doing this will change the appearance of the picture to highlight particular structures. The brightness of the image is adjusted via the window level. The contrast is adjusted via the window width.

Tissue density is measured in [Hounsfield units (HU)](https://en.wikipedia.org/wiki/Hounsfield_scale)

* This is defined as **Air = −1000 HU**; **Water = 0 HU.**

Density of tissues in CT-Scans:

𝐴𝑖𝑟<𝐹𝑎𝑡<𝐹𝑙𝑢𝑖𝑑<𝑆𝑜𝑓𝑡𝑡𝑖𝑠𝑠𝑢𝑒<𝐵𝑜𝑛𝑒<𝑀𝑒𝑡𝑎𝑙Air<Fat<Fluid<Softtissue<Bone<Metal

The easier way to remember this is (Fat floats on water, so is less dense than fluid; Soft tissue is mostly intracellular fluid with some connective tissue)

* Air = −1000 HU
* Lung ≈ −500 HU (partially air, partially soft tissue)
* Fat ≈ −50 HU (slightly less dense than simple fluid)
* Water = 0 HU
* Soft tissue (& blood) ≈ +50 HU (slightly more dense than simple fluid)
* Bone ≈ +1000 HU (much more dense)

Text

Description automatically generated with medium confidence

To ascertain a window, a ‘level’ and a ‘width’ is defined. For example, a window with a level of 0 HU and a width of 400 HU will have a range of −200 HU to +200 HU. Any tissue with a density of −200 HU or less will be black, and any tissue with a density of +200 HU or more will be white. And values between -200 HU to +200 HU will be spread between the whole grayscale range. A **window** can be set to look at certain tissues of interest. **A small range of tissue density is represented by a full greyscale spectrum from black to white, thus making subtle density differences within the specified range easier to see.**

### Typical window width and level values:

Although this varies somewhat from institution to institution and vendor to vendor, window width and centers are generally fairly similar. **The values below are written as width and level (W:x L:y) in Hounsfield units (HU).**

* head and neck
* brain W:80 L:40
* subdural W:130-300 L:50-100
* stroke W:8 L:32 or W:40 L:40 3
* temporal bones W:2800 L:600
* soft tissues: W:350–400 L:20–60 4
* chest
* lungs W:1500 L:-600
* mediastinum W:350 L:50
* abdomen
* soft tissues W:400 L:50
* liver W:150 L:30
* spine
* soft tissues W:250 L:50
* bone W:1800 L:400

w: window width

L: window center

img = (img\*slope +intercept) *#for translation adjustments given in the dicom file.*

img\_min = window\_center - window\_width//2 *#minimum HU level*

img\_max = window\_center + window\_width//2 *#maximum HU level*

img[img<img\_min] = img\_min *#set img\_min for all HU levels less than minimum HU level*

img[img>img\_max] = img\_max *#set img\_max for all HU levels higher than maximum HU level*

if rescale:

img = (img - img\_min) / (img\_max - img\_min)\*255.0

return img

Apr 27, 2022, Wed:

Full preprocessing tutorial: <https://www.kaggle.com/code/gzuidhof/full-preprocessing-tutorial/notebook>

Image View Position: <http://www.grahamwideman.com/gw/brain/orientation/orientterms.htm>

Orientation: <https://www.aliza-dicom-viewer.com/manual/orientation>

Code reference: <https://vincentblog.xyz/posts/medical-images-in-python-computed-tomography>

A picture containing text, tool

Description automatically generated

Original vs. Windowing:

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

Cropped & Resampled & Windowed Images through ROI reference:

A picture containing graphical user interface

Description automatically generated

May 2, 2021, Mon: meeting

Onboarding (estimated at the beginning of June):

* receive email from HireRight 🡪 finish background check
* background check takes up to 3 weeks 🡪 receive confirmation number
* send confirmation # back to HR 🡪 receive MGH email from HR

Parameters to tune:

* model hyperparameters
* preprocessing parameters (rule-based)???

U-Net: try segmentation project with 2D CT images (Kaggle)

* Q: **should we consider taking just 2D slices since lack of training samples**

Anything to do without using lab computer:

* Try dataset one by one? 2D or 3D

May 5, 2022, Thu: MRI reorientation

Align ROI and images by rotation

A picture containing text, monitor, screen, screenshot

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

N4BiasCorrection

Graphical user interface, website

Description automatically generatedGraphical user interface, website

Description automatically generated

**May 13, 2022, Fri: meeting with Jay and Ben (lab computer)**

* Log in terminal: ssh [yy908@glacier.nmr.mgh.harvard.edu](mailto:yy908@glacier.nmr.mgh.harvard.edu)
* Put data and code in “**project” and “data”** folder: local\_mount/space/glacier/1
* **dockerhub** 🡪 **register an account (use ubuntu containers; tensorflow containers):** docker run --gpus '"device=0:0"' --rm -it -v /local\_mount/space/glacier/1/data/:/april -v /local\_mount/space/glacier/1/projects/april/:/app tensorflow/tensorflow:latest-gpu bash
* docker run –rm … (good habit: if done with something, go delete the container)

**GPU and deep learning packages:**

* search tensorflow image with cuda (find them on docker hub when setting up an image/container)
* Ben already pulled it so I can directly use it

docker run -it --rm --gpus '"device=0:0"' projectmonai/monai nvidia-smi

(real gpu : virtual gpu)

* vscode: remote ssh (extension needs to be installed; already installed!)
* ctrl + shift + p
* ctrl + tilt (open terminal)

Text

Description automatically generated

* if using Jupyter notebook, remember changing kernel (already there: python 3.8.10)

Change container/glacier:

* Command + shift + p: type **remote ssh** or **remote container**

Sign up availability before and after using GPU:

* <https://docs.google.com/spreadsheets/d/1oJ3h7CumO_UEXJsSxRt_lcvRJBNaqOZ5mJlWKT9aBlY/edit#gid=0>

vim: save all your user names

**Next step:**

* Create data + code folder
* Move brain images?? Or CT
* Use a few images to build the U-Net model
* Brain preprocessing

May 23, Mon, 2022:

Document 🡪 open folder A screenshot of a computer

Description automatically generated with medium confidence

Steps:

1. Command + shift + p: open ssh remote 🡪 connect to glacier
2. Build the container:

docker run --gpus '"device=1:1"' --name ayan\_temp --rm -it -v /local\_mount/space/glacier/1/data/:/april\_data -v /local\_mount/space/glacier/1/projects/april\_project/:/app tensorflow/tensorflow:latest-gpu bash

1. Ctrl + shift + p: open remote container (select the one built just now). Everything will be synced in the container (i.e., virtual machine)
2. When entering code, change kernel to python 3.8.10

Attach + detach:

* In terminal: ctrl + p + q (detach)

FileZilla: move large data files/folders to glacier

A screenshot of a computer

Description automatically generated with medium confidence

docker run --gpus '"device=0:0"' --name april\_project\_gpu0 --rm -it -v /local\_mount/space/glacier/1/data/april\_data/:/data -v /local\_mount/space/glacier/1/projects/april\_project/:/app jaybpatel/jay\_tensorflow2.2:latest bash

--detach-keys=”ctrl-[”

May 29, 2022, Sun: preprocessing coding

* Resampling: change voxel spacing to 1,1,1

Img.header[‘pixdim’]:

The codes below can be used in xyzt\_units to indicate the units of pixdim.

As noted earlier, dimensions 1,2,3 are for x,y,z; dimension 4 is for

time (t).

- If dim[4]=1 or dim[0] < 4, there is no time axis.

- A single time series (no space) would be specified with

- dim[0] = 4 (for scalar data) or dim[0] = 5 (for vector data)

- dim[1] = dim[2] = dim[3] = 1

- dim[4] = number of time points

- pixdim[4] = time step

- xyzt\_units indicates units of pixdim[4]

- dim[5] = number of values stored at each time point

Original Images:

A picture containing black

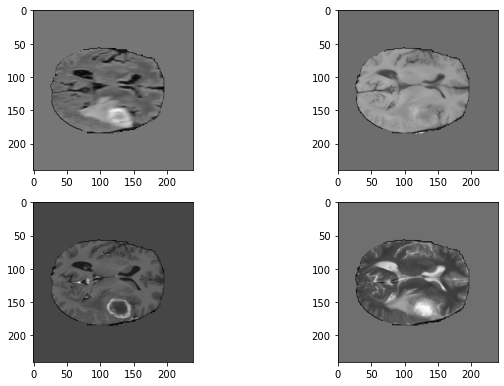
Description automatically generated

Bias corrected:

A picture containing echinoderm

Description automatically generated

Normalized:



June 15, Wed, 2022

docker run --gpus '"device=0:1"' --name april\_brain\_segment\_1 --rm -it -v /local\_mount/space/glacier/1/data/april\_data/Patient\_4chn/:/home/neural\_network\_code/Data/Patients -v /local\_mount/space/glacier/1/projects/april\_project/Code/:/home/neural\_network\_code/Code jaybpatel/jay\_tensorflow2.2:latest bash

* Logout: type exit
* Detach from container: ctrl p + ctrl q (delete default in VS code)
* Open terminal in VS: ctrl + ` (tilt)
* Open search bar: command + shift + p

Epoch1: 364/364 - 1886s - hard\_dice\_coef\_metric: 0.3725 - dice\_coef\_metric: 0.3600 - loss: 0.6400 - val\_hard\_dice\_coef\_metric: 0.3864 - val\_dice\_coef\_metric: 0.3830 - val\_loss: 0.6170

Epoch2: 364/364 - 1147s - hard\_dice\_coef\_metric: 0.5329 - dice\_coef\_metric: 0.5295 - loss: 0.4705 - val\_hard\_dice\_coef\_metric: 0.5700 - val\_dice\_coef\_metric: 0.5654 - val\_loss: 0.4346

Epoch3: 364/364 - 883s - hard\_dice\_coef\_metric: 0.5726 - dice\_coef\_metric: 0.5701 - loss: 0.4299 - val\_hard\_dice\_coef\_metric: 0.5756 - val\_dice\_coef\_metric: 0.5736 - val\_loss: 0.4264

Epoch4 364/364 - 883s - hard\_dice\_coef\_metric: 0.5866 - dice\_coef\_metric: 0.5846 - loss: 0.4154 - val\_hard\_dice\_coef\_metric: 0.5660 - val\_dice\_coef\_metric: 0.5648 - val\_loss: 0.4352

Epoch5: 364/364 - 884s - hard\_dice\_coef\_metric: 0.6027 - dice\_coef\_metric: 0.6011 - loss: 0.3989 - val\_hard\_dice\_coef\_metric: 0.6286 - val\_dice\_coef\_metric: 0.6272 - val\_loss: 0.3728

Epoch6:

Epoch7:

Epoch8:

Epoch9:

Epoch10: