Francois Rheault, PhD
Medical-image Analysis and Statistical Interpretation (MASI)

Electrical engineering department
Vanderbilt University, TN, USA
https://my.vanderbilt.edu/masi/

Crash course in Imaging

Intro to image processing

Plan

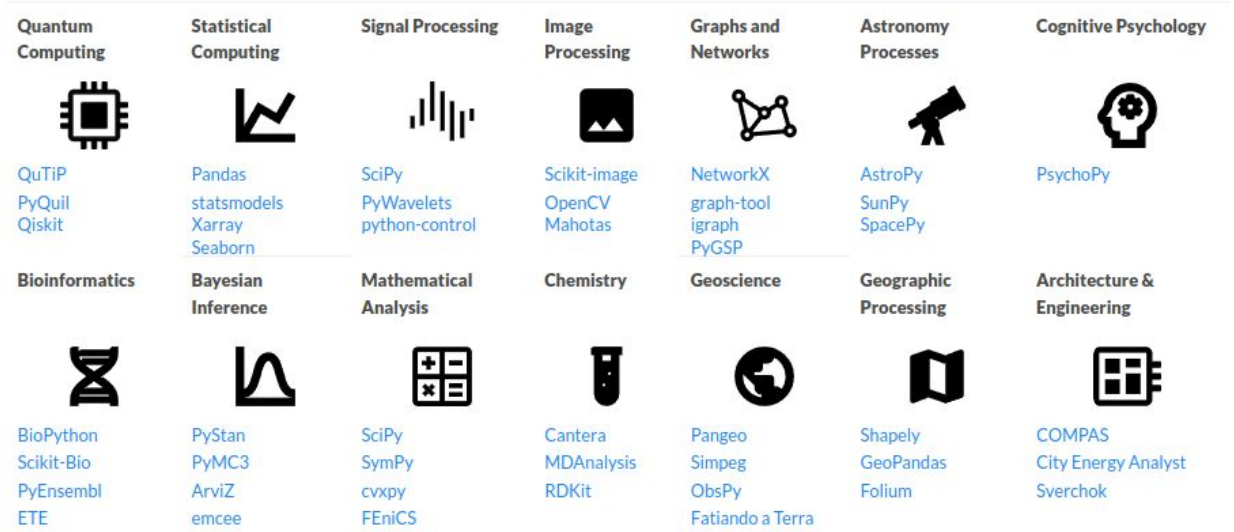
Understanding Numpy

- Array attribute
- Array indexing
- Array slicing
- Array operations
- Inf vs NaN, overflow
- Array broadcasting
- Concept of masking (binary)
- Morphological operations
- Concept of labelling/segmentation

Understanding Numpy

What is numpy?

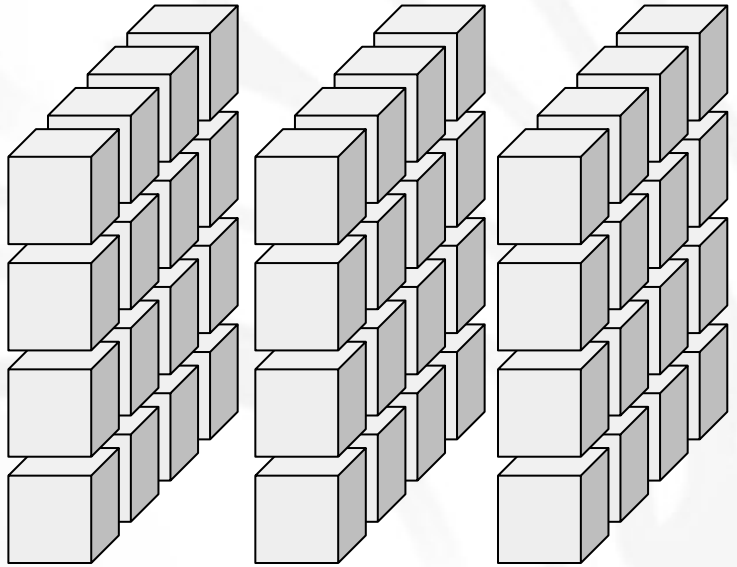
NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.



Understanding Numpy

Attributes:

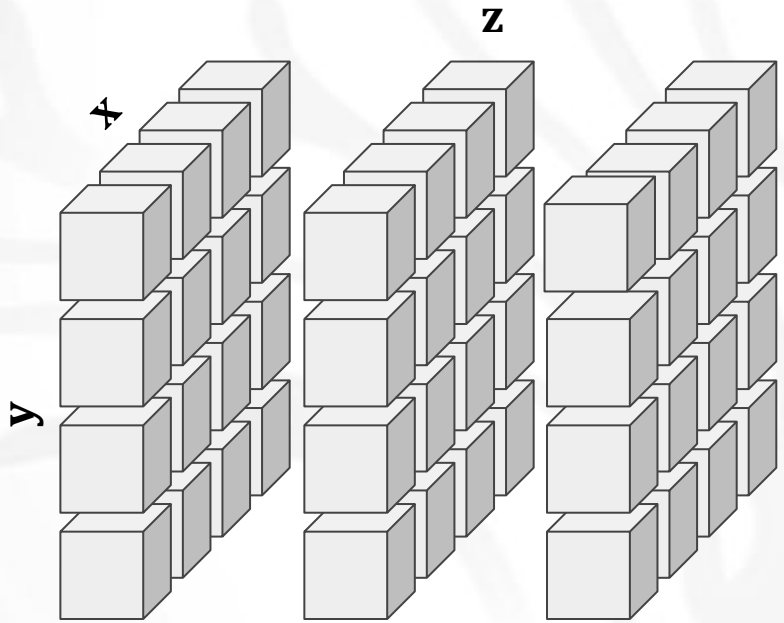
- **ndim**: number of dimension
- **shape**: number of element in each dimension (4x4x3)
- **size**: total number of element
- **dtype**: datatype (bool, int, float, double)



Understanding Numpy

Indexing:

- Act of accessing an element
- Starts at 0 (not 1, *warning matlab user*)
- Knowing where things start & end



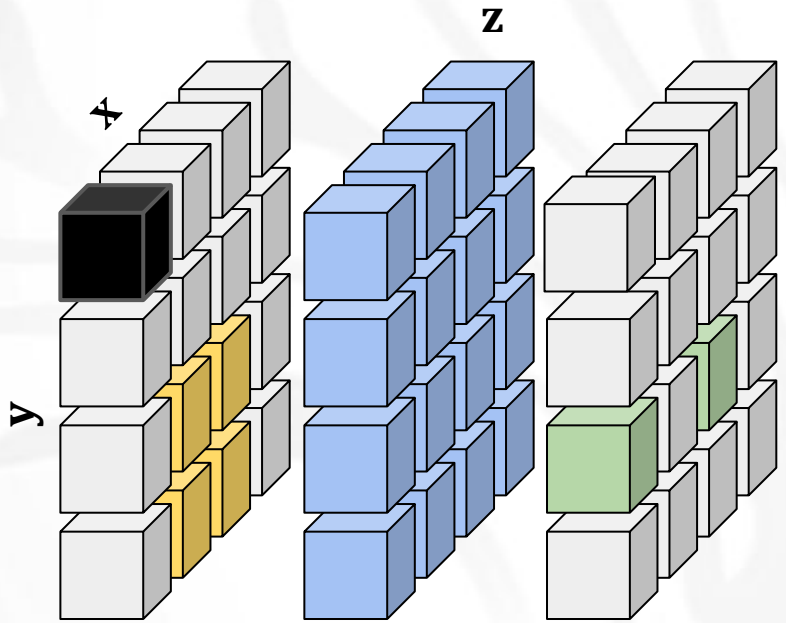
Understanding Numpy

Indexing:

- Act of accessing an element
- Starts at 0 (not 1, *warning matlab user*)
- Knowing where things start & end
  - array[0, 0, 0] **?**
  - Array[1,1,3]-

Slicing:

  - array[:, :, 1]
  - array[1:3, 1:3, 0]
  - array[::2, 2, 2]



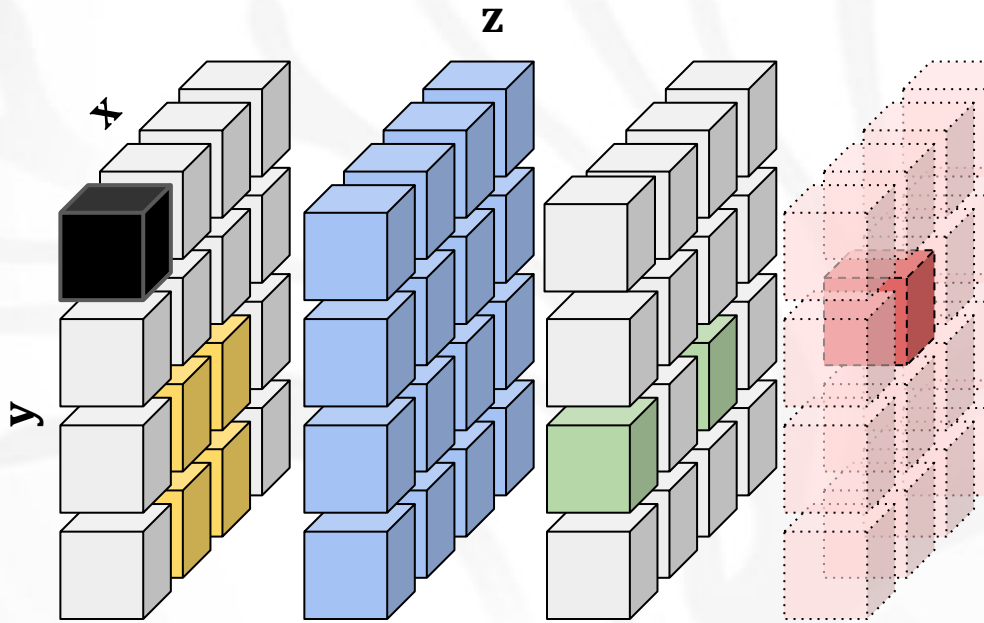
Understanding Numpy

Indexing:

- Act of accessing an element
- Starts at 0 (not 1, *warning matlab user*)
- Knowing where things start & end
  - array[0, 0, 0] **?**
  - Array[1,1,3]-

Slicing:

  - array[:, :, 1]
  - array[1:3, 1:3, 0]
  - array[::2, 2, 2]



Understanding Numpy

Operators

- array = array_1 + array_2
  - array_1 += array_2
- array = array_1 - array_2
  - array_1 -= array_2
- array = array_1 * array_2
  - array_1 *= array_2
- array = array_1 / array_2
  - array_1 /= array_2
- array = array_1 ** array_2
  - array_1 **= array_2

Functions

- array = np.add(array_1, array_2)
- array = np.subtract(array_1, array_2)
- array = np.multiply(array_1, array_2)
- array = np.divide(array_1, array_2)
- array = np.power(array_1, array_2)

np.arange, np.linspace, np.zeros, np.ones, etc.
Np.round, np.ceil, np.floor, np.max, np.min, etc.
np.sum, np.average, np.std, np.sin, np.cos, etc.

And a ton more!

Understanding Numpy

Limits of the representation

- [1,2,3] /= [0,0,0] -> [inf, inf, inf]
- [0,0,0] /= [0,0,0] -> [nan, nan, nan]
- [255, 1, 1] += 1 -> [256, 2, 2] **OR** [0, 2, 2] (**int**16/32/64 and **float**16/32/64 vs int8)
- [0, 10, 10] -= 1 -> [-1, 9, 9] **OR** [255, 9, 9] (any **int** vs **uint**)

Understanding Numpy

Limits of the representation

- [1,2,3] /= [0,0,0] -> [inf, inf, inf]
- [0,0,0] /= [0,0,0] -> [nan, nan, nan]
- [255, 1, 1] += 1 -> [256, 2, 2] **OR** [0, 2, 2] (**int**16/32/64 and **float**16/32/64 vs int8)
- [0, 10, 10] -= 1 -> [-1, 9, 9] **OR** [255, 9, 9] (any **int** vs **uint**)

uint8: (0, 255), int8: (-128, 127)
uint16: (0, 65535), int16: (-32768, -32767)
uint32: (0, 2^32), int32: (--2^32 / 2, -2^32 / 2)
uint64: (0, 2^64), int64: (--2^64 / 2, -2^64 / 2)

float16: from -6.550e+04 to 6.550e+04, smallest representable value: 6.103e-05, precision: 1.0e-03
float32: from -3.402e+38 to 3.402e+38, smallest representable value: 1.175e-38, precision: 1.0e-06
float16: from -1.797e+308 to 1.797e+308, smallest representable value: 2.225e-308, precision: 1.0e-015

Understanding Numpy

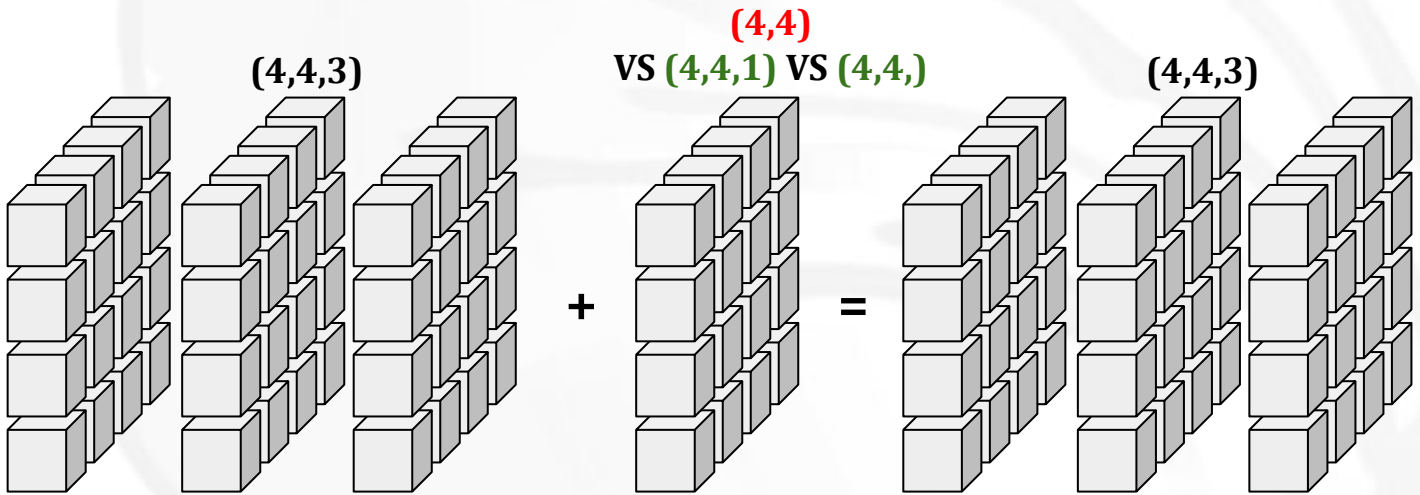
Broadcasting

- Numpy tries to help you by matching dimensions (if **possible**, sometimes by **accident**)

  array_1 = [1, 2, 3, 4, 5]

  array_1 += array_1 **VS** array_1 *= 2

- Matching in the joint dimension OR empty extra dimension



Understanding Numpy

Masking

- Hidden values of an array using another array (or threshold)
- array_1[array_1 > 0] = 1
  - Binarize the array, using a lower threshold of 1
- array_2 *= array_1
  - Set values to 0 using a mask

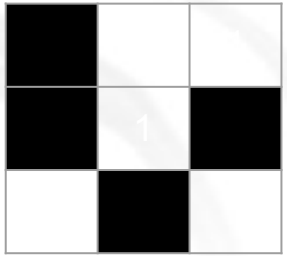
0	2	3
0	5	0
11	0	13

->

0	1	1
0	1	0
1	0	1

1	2	3
4	5	6
7	8	9

*



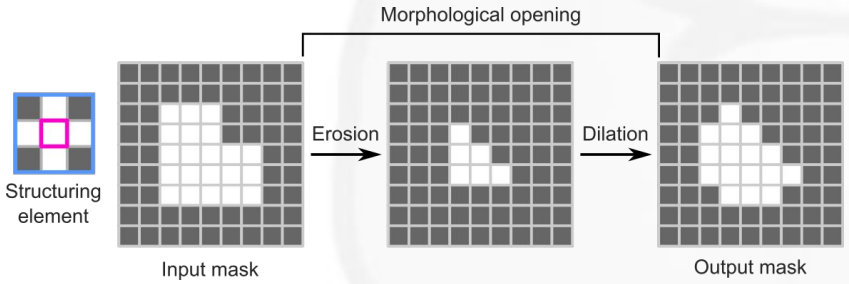
*

0	2	3
0	5	0
7	0	9

Understanding Numpy

Modify the morphology of a binary array using a binary structure

Erosion vs Dilation

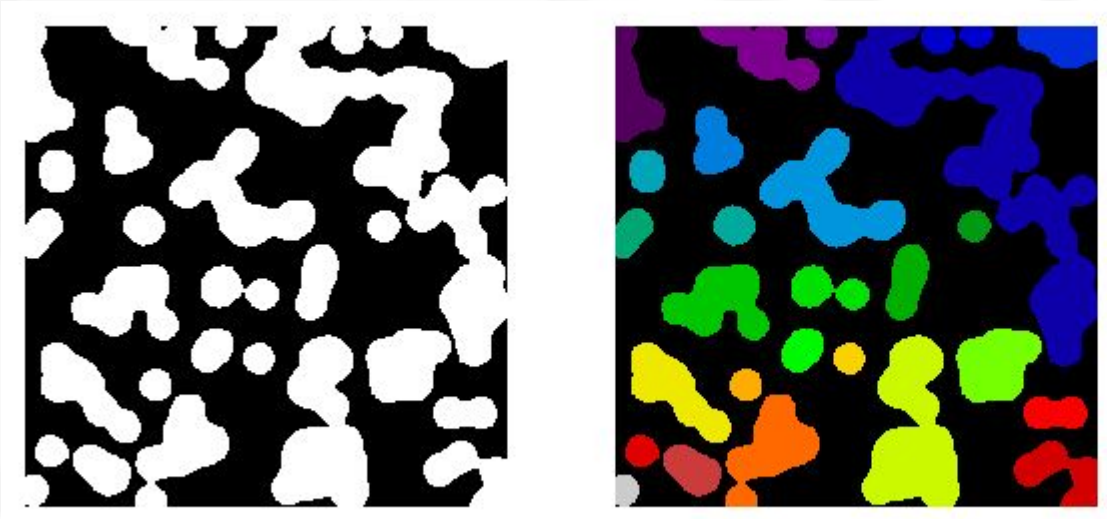


Useful to increase of decrease the size of a mask (measure of certainty) or to improve a mask when using threshold (minimum size of elements)

Understanding Numpy

Segmentation

- Segmentation, labelling, classification, etc.
- Numpy array with integers, all elements with the same value have the same label/class



Libraries

Pillow (PIL)

This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

ImageIO

Imageio is a Python library that provides an easy interface to read and write a wide range of image data, including animated images, volumetric data, and scientific formats.

Nibabel

This package provides read +/- write access to some common medical and neuroimaging file formats, including: ANALYZE (plain, SPM99, SPM2 and later), GIFTI, NIfTI1, NIfTI2, CIFTI-2, MINC1, MINC2, AFNI BRIK/HEAD, MGH. We can read and write FreeSurfer geometry, annotation and morphometry files (limited support for DICOM)

The various image format classes give full or selective access to header (meta) information and access to the image data is made available via NumPy arrays.

Libraries

Scipy

SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics and many other classes of problems.

Scikit-Image

scikit-image is an open-source image processing library for the Python programming language. It includes algorithms for segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature detection, and more.

Scikit-Learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms

Francois Rheault, PhD
Medical-image Analysis and Statistical Interpretation (MASI)

Electrical engineering department
Vanderbilt University, TN, USA
https://my.vanderbilt.edu/masi/

Crash course in Imaging

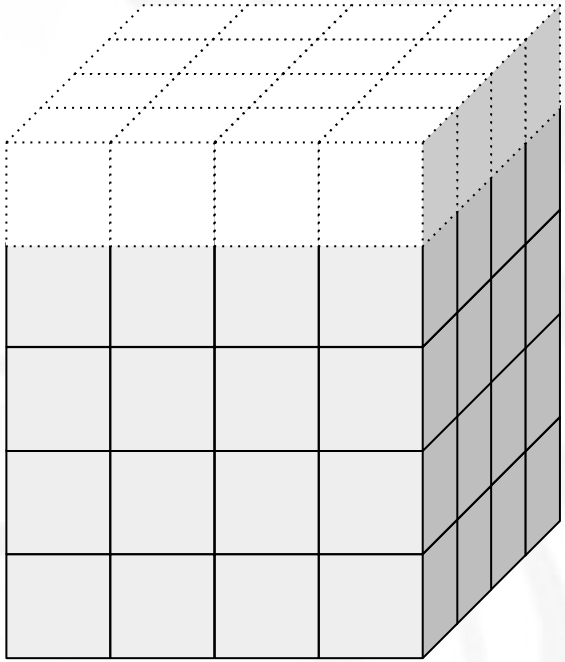
Intro to medical image processing

Understanding NIFTI

NIFTI: Neuroimaging Informatics Technology Initiative

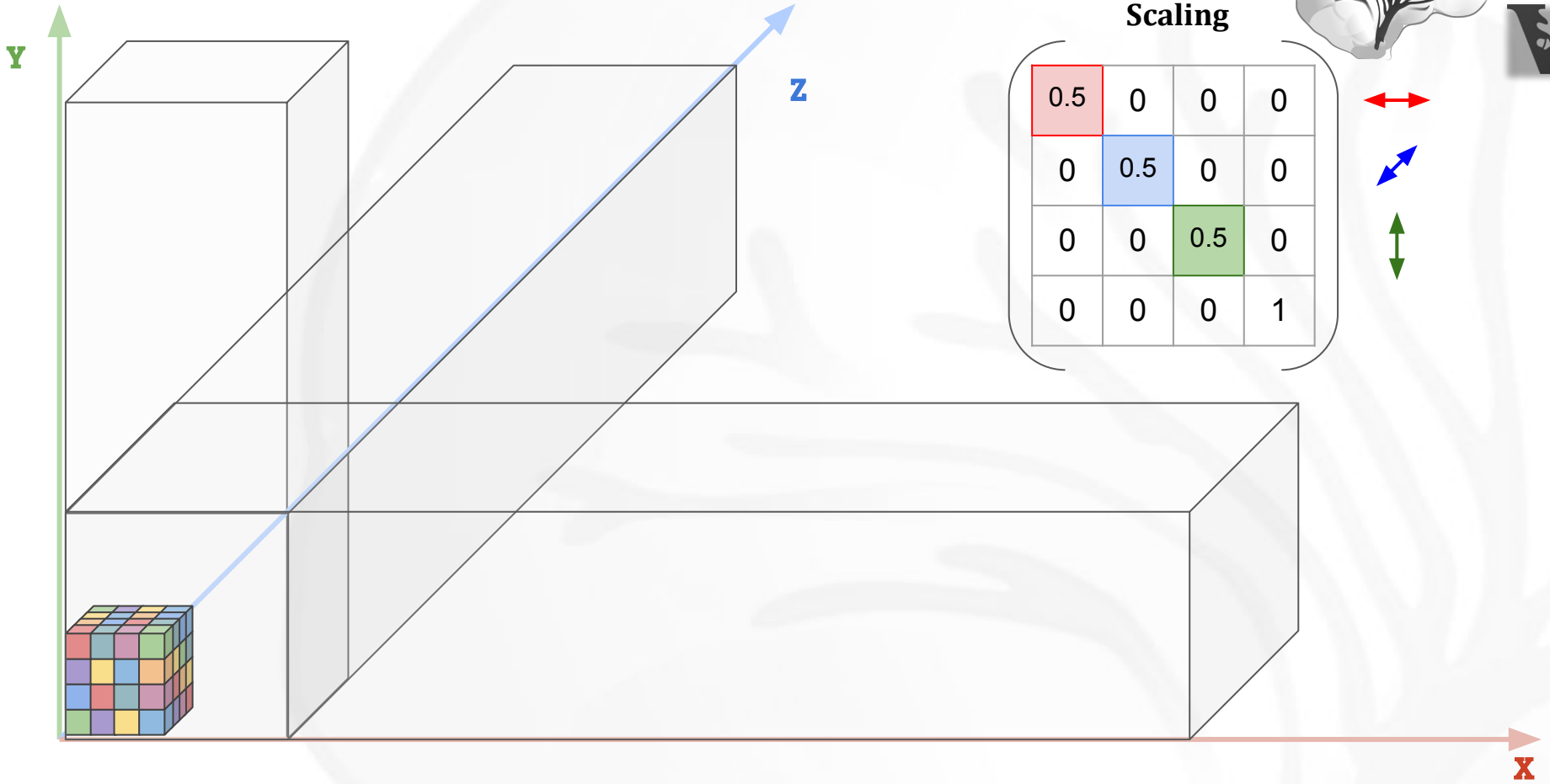
Understanding the attributes of NIFTI files, starting with the grid (or data/volume/array)

1) How many elements are present in my grid?
Dimensions or **Shape**



4x5x4 **vs** 4x4x4

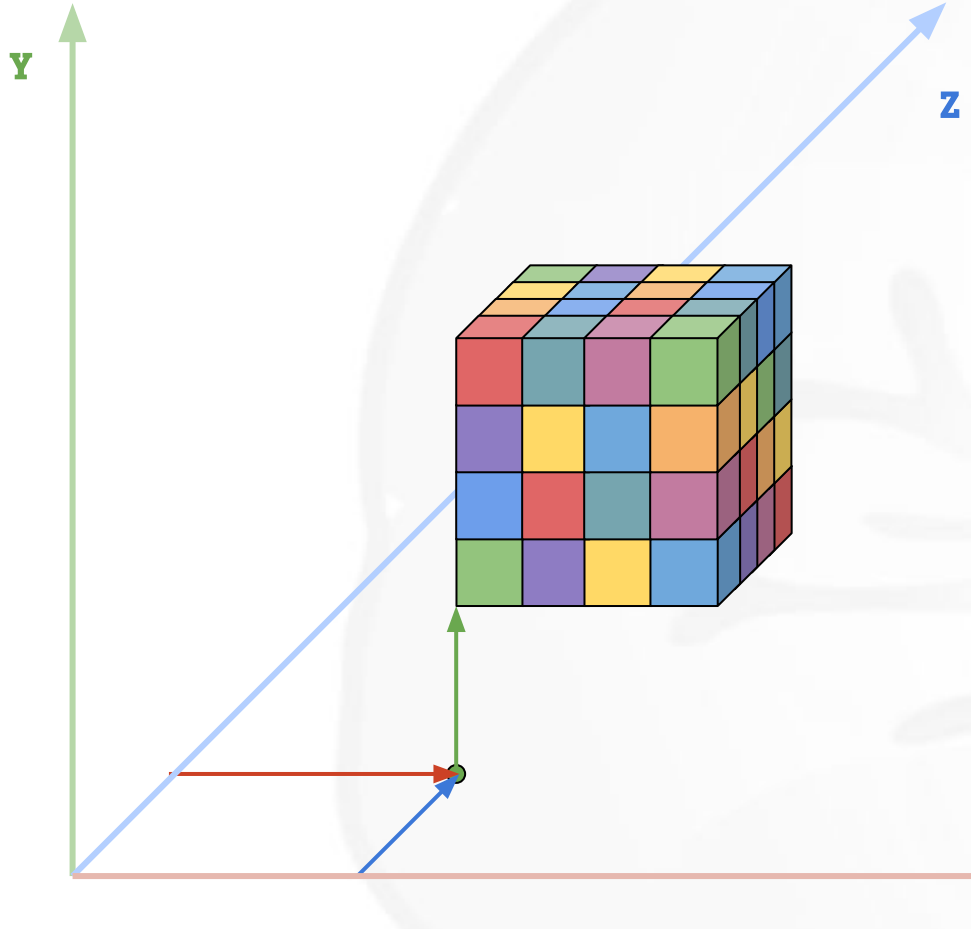
Understanding NIFTI



Scaling

0.5	0	0	0
0	0.5	0	0
0	0	0.5	0
0	0	0	1

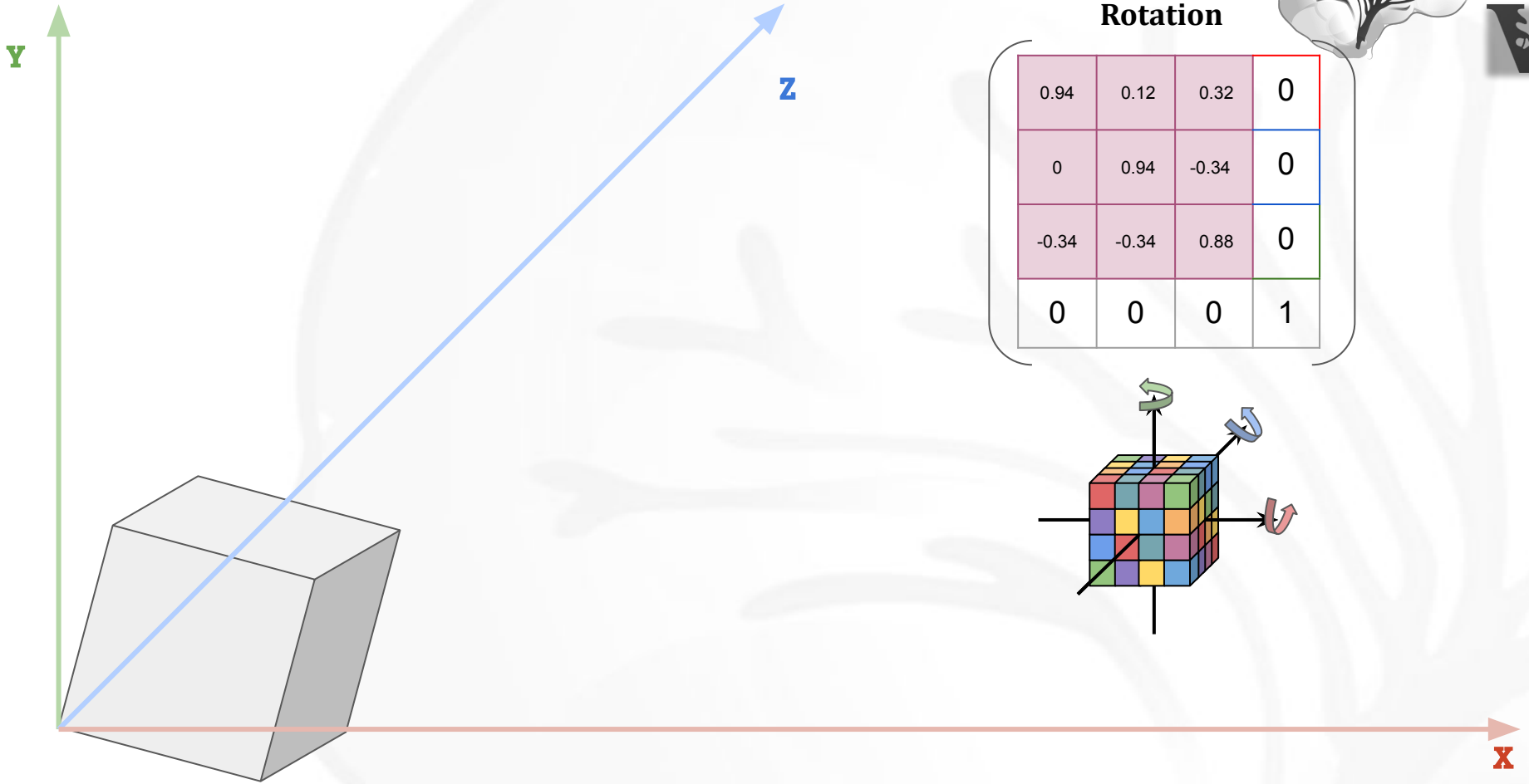
Understanding NIFTI



Translation

1	0	0	5
0	1	0	2
0	0	1	3
0	0	0	1

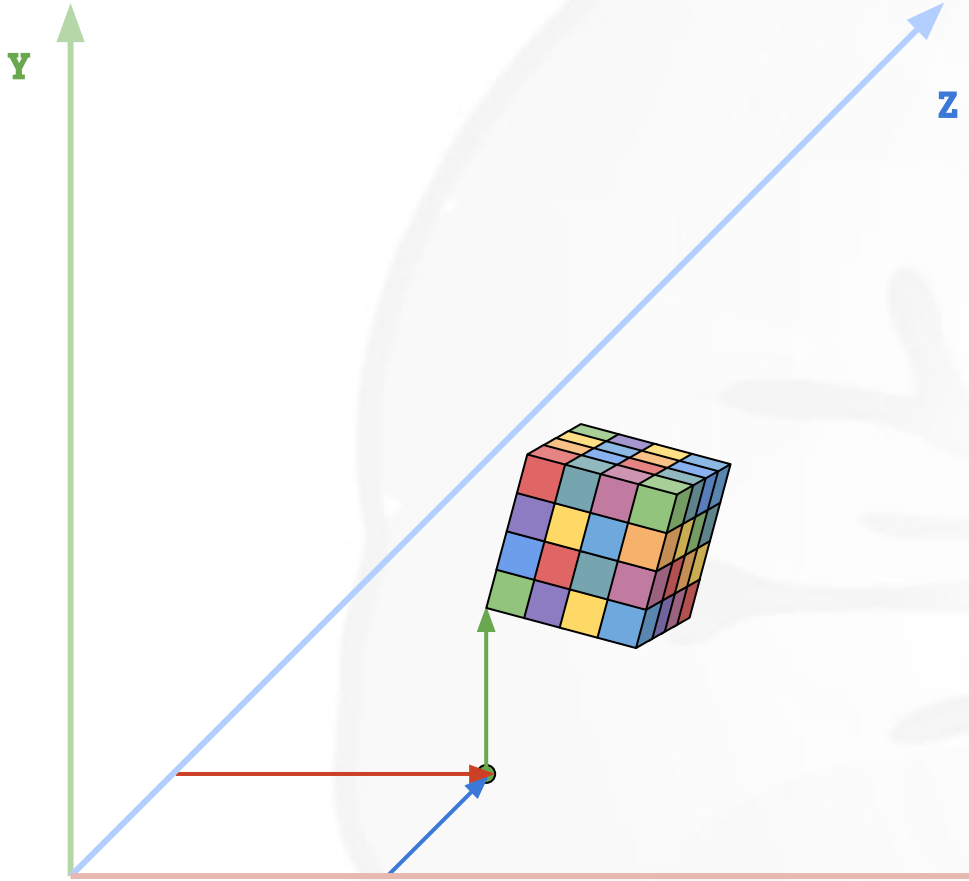
Understanding NIFTI



Rotation

0.94	0.12	0.32	0
0	0.94	-0.34	0
-0.34	-0.34	0.88	0
0	0	0	1

Understanding NIFTI



Affine Transform

$$\begin{pmatrix} 0.47 & 0.06 & 0.16 & 5 \\ 0 & 0.47 & -0.17 & 2 \\ -0.17 & -0.17 & 0.44 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

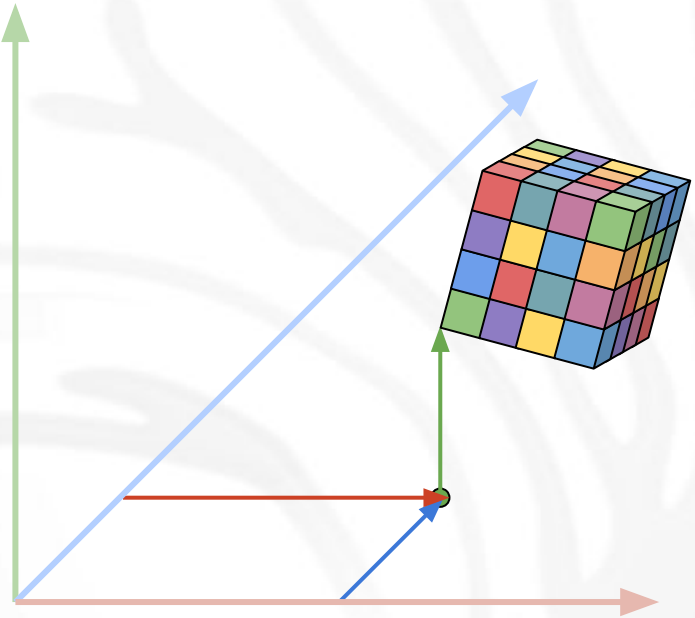
Understanding NIFTI

NIFTI: Neuroimaging Informatics Technology Initiative

Spatial Transformations will move the 'simple' grid *somewhere*

By adding a **scaling**, **translation** and **rotation**, now our grid has a real size (mm) and has a position in space (relative to the scanner).

If the grid is sometime called '*voxel space*', this new concept is sometime called '*world space*' (or '*scanner space*', or '*rasmm*').

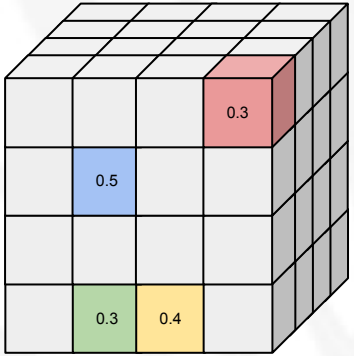


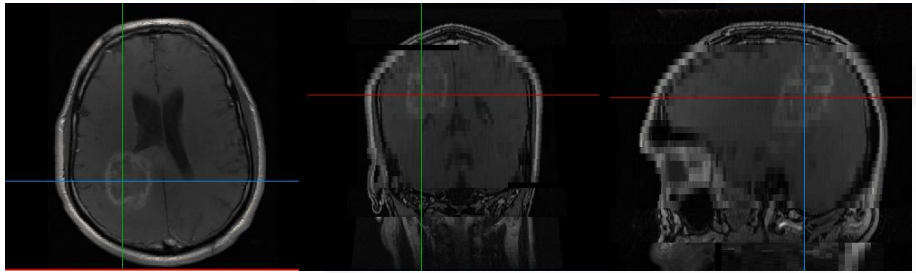
Understanding NIFTI

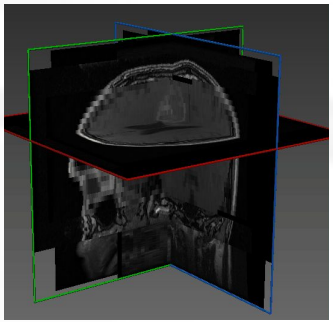
Why switching from world space to voxel space (and vice versa) so important?

- Voxel Space
  - Interpolation of metrics at specific position (see on the right)
  - Counting voxels to estimate volume
- World Space
  - Convex hull to compute precise volume
  - Registration between datasets (see below)

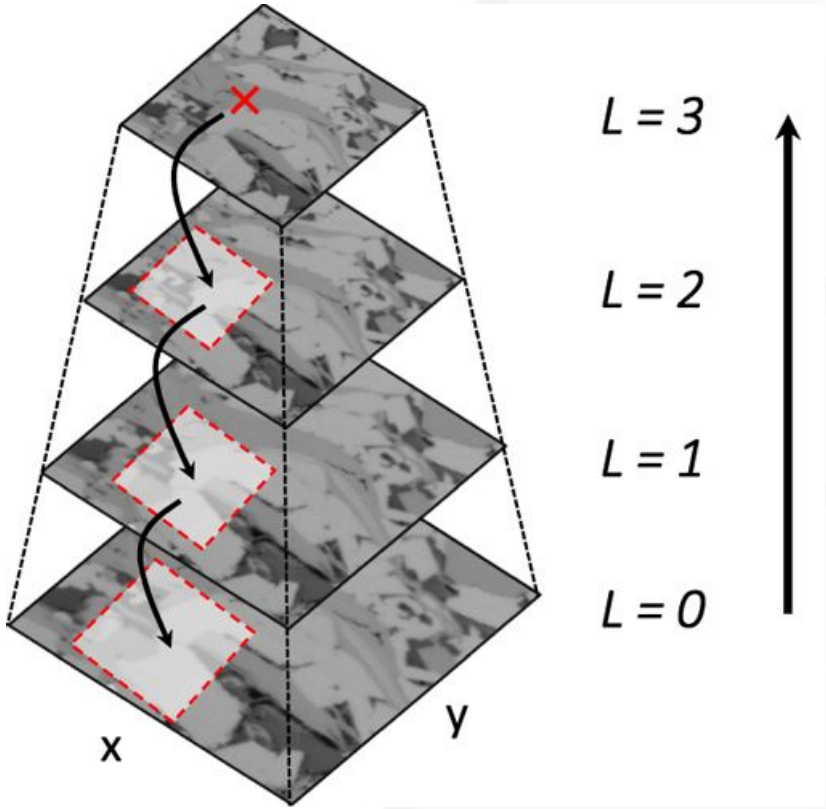
    (across session, multi-modalities, inter-subjects, etc.)

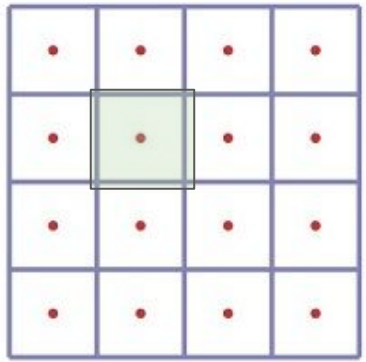




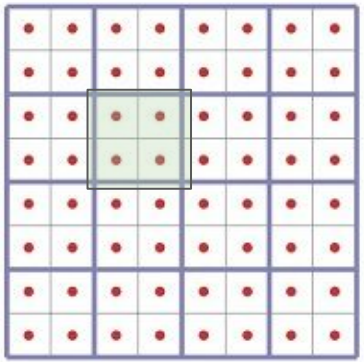


Understanding NIFTI





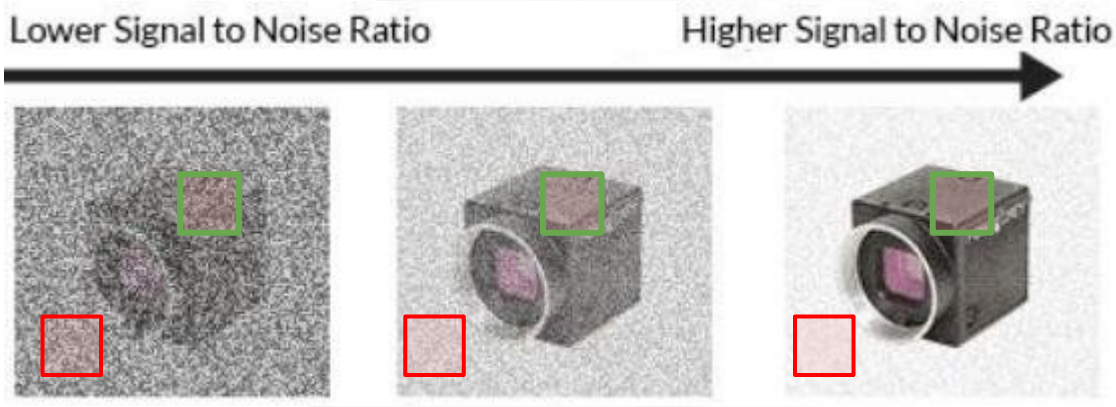
Level 0

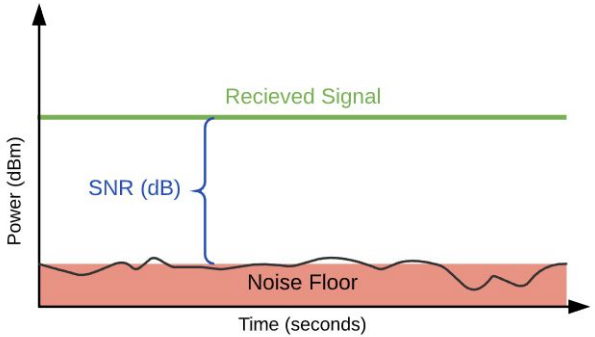


Level 1

SNR

A good signal-to-noise ratio makes everything simple

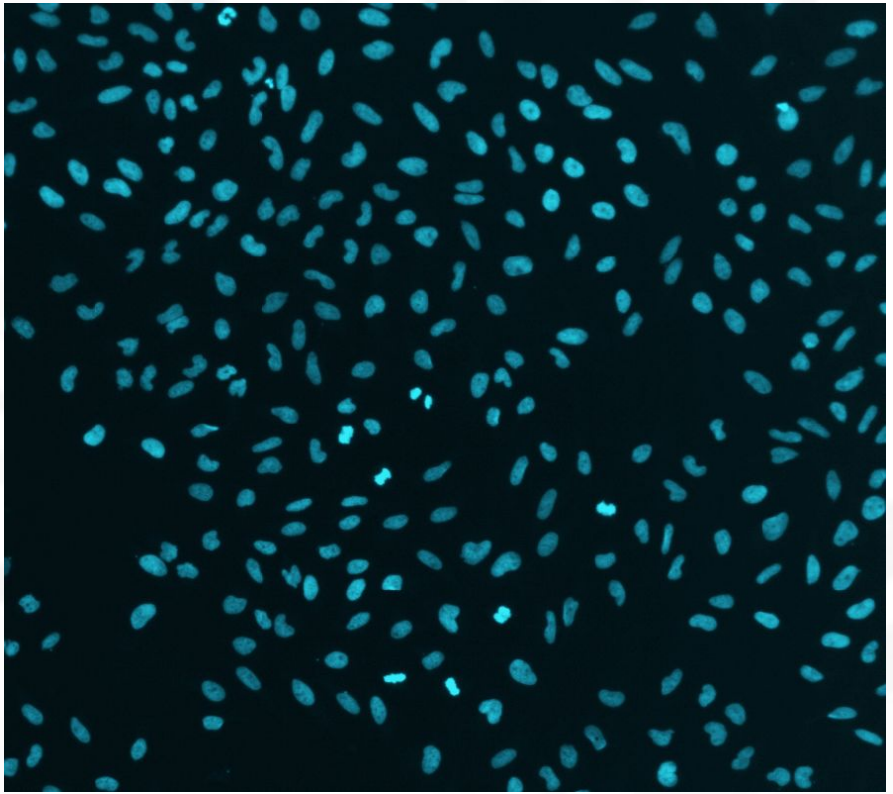




Cell Counting

Classic image processing task

1. Background vs Cell
    a. Good contrast?
2. Average size of cell (in pixel)
3. Cell body = dense
    a. Local maximal
4. Average shape?
    a. Line vs circle, convex vs concave



Measuring similarity

Compare agreement (intra/inter rater)

1. Are we measuring the same thing: What is the surface covered by windows?