BIOMATH 208 Week 1

Aidan Jan

January 6, 2025

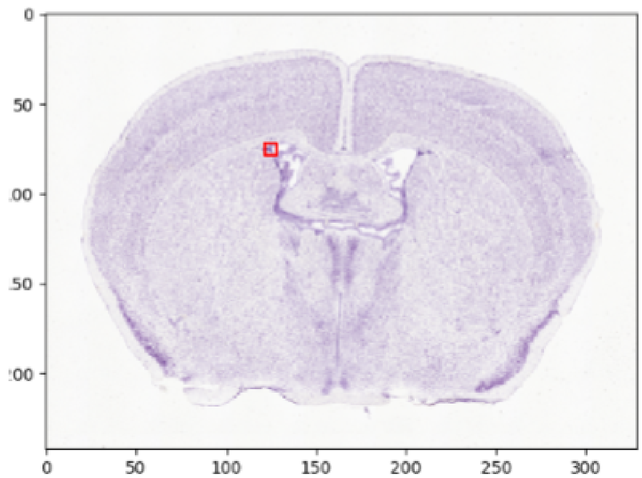
Overview

1. Representing and visualizing imaging data
   - Pixels, vectors, etc.
2. Multilinear Algebra
   - Linear algebra with stuff added on top
   - Two types of square matrices
3. Curves and surfaces
   - Comparing curves and surfaces using multilinear algebra
4. Manifolds
5. Transformation Groups
   - Understanding rotation, linear transformations
6. Tangent spaces
7. Optimization and image registration
   - Aligning multiple images of the same object
8. Metric manifolds
   - Finding distance between two rotation matrices, ellipsoids, probabilities, etc.
   - Using distances to compute averages, make predictions, etc.
9. Averaging filtering and regression

Representing images

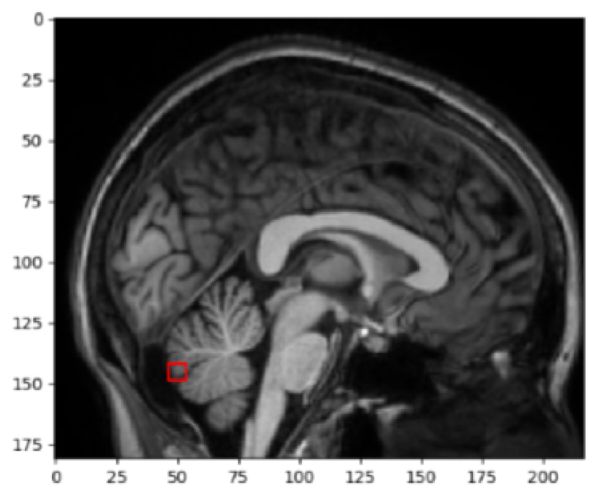
- Definition (images): We generally consider images as functions from some set $X \subseteq \mathbb{R}^d$, to some other set, S.
  - X describes space; usually, $d = 2$ or $d = 3$, depending on the dimension of the image.
- What this means:
  - Let X represent a picture, some 2D set of pixels in this example. Let $x \subset X$, where x is a pixel in the picture.
  - We can define the image, I, as a function, such that $X : X \to S$, where S is a pixel value.
    - $S \in \mathbb{R}$ if the image is grayscale, and $S \in R^3$ if the image is colored.

Example 1



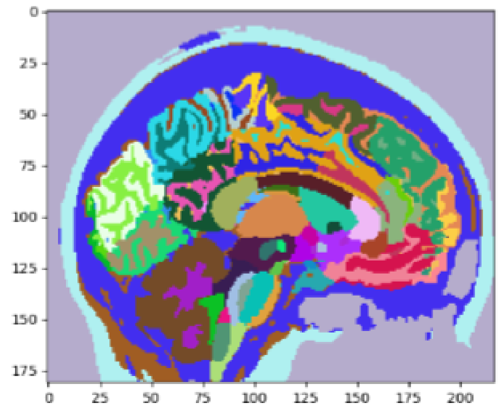
- Above is a microscopy image of a Nissl stained mouse brain.
- Here, $d = 2$ with X some rectangle, and $S = [0,1]^3$ containing 3 RGB values, each value between 0 and 1.

Example 2



- Above is a part of a human brain MRI. The set X is shown on the axes.
- Here, $d = 3$, since this is a 3D image. $S = \mathbb{R}$ since it is grayscale, therefore only one value.
  - The S value is a value between $[0, 1]$ representing the brightness of the pixel.

Example 3 - Label Images

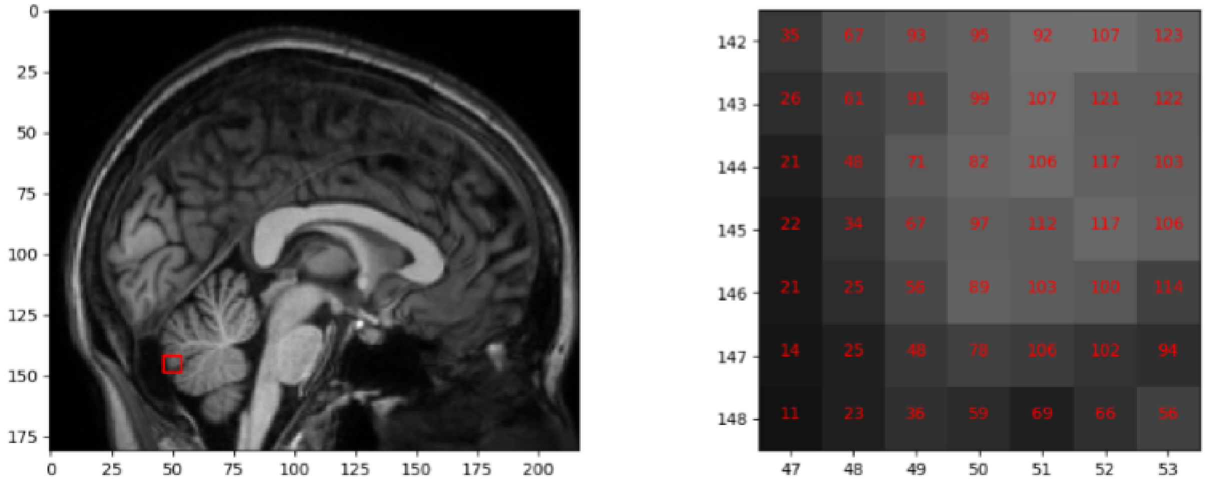


- Here, $d = 3$ with X some rectangular prism, and $S = \mathbb{N}$ containing 1 integer, `id`. Each `id` represents some particular type of structure.
- Part of a labeled brain image is shown. Integers are shown as colors, and the set X is shown on the axes for one slice of the 3D volume.

Discrete Images

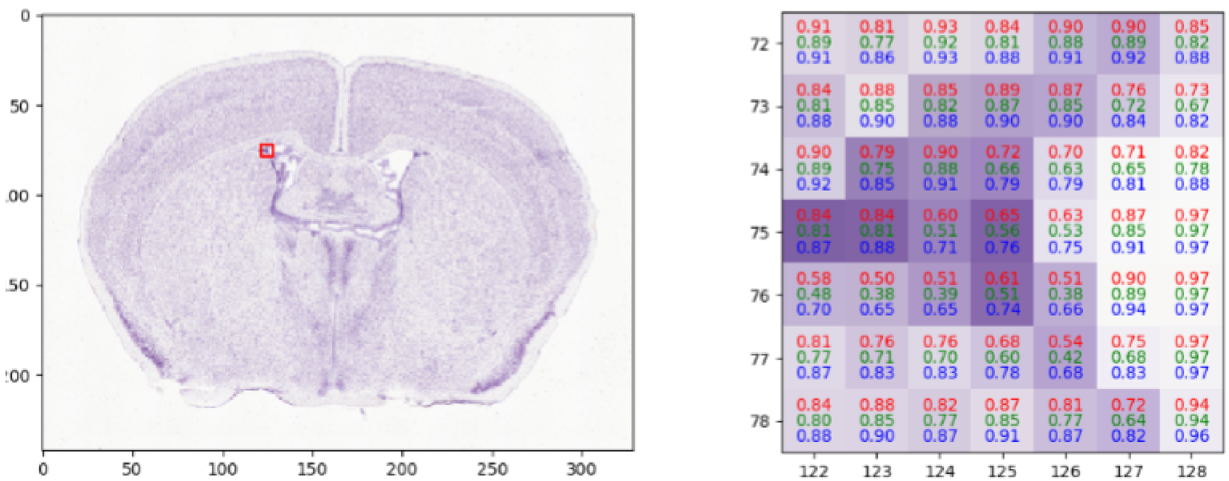
- Treating images as functions is nice, because they have basically infinite resolution. (You can plug in any coordinate value and receive the corresponding values.)
- However, we can't store the functions in a computer. As a result, we usually work with **discrete images**.
- In this case, images are d-dimensional arrays, storing values in S.
  - Geometry is stored with pixel size $\Delta \in \mathbb{R}^d$, and origin $O \in \mathbb{R}^d$.
  - Square brackets are used to index starting from 0. `img[0, 0]` would represent the top left corner.
  - In 3D, we would use rows, columns, and slices.
  - We may also specify the location of the origin, O, to scale in real life. Origin is `img[0, 0]`, but may be assigned some real world location, for example, [10ft, 20ft]. Δ is the pixel size, and therefore the physical location for any pixel can be calculated. (Real location of pixel `img[i,j]` would be $O + \Delta \cdot \binom{i}{j}$).

Grayscale images



- Here we see a sagittal view of a brain MR image, and a zoom in showing pixel values.
- Here, the origin is (0, 0), and the spacing is (1, 1) (in mm).

Color Histology



- Here we see an image, and a zoom in showing pixel values as RGB triples.
- Here the origin is (0, 0), and the spacing is (58.8, 58.8) (in microns).

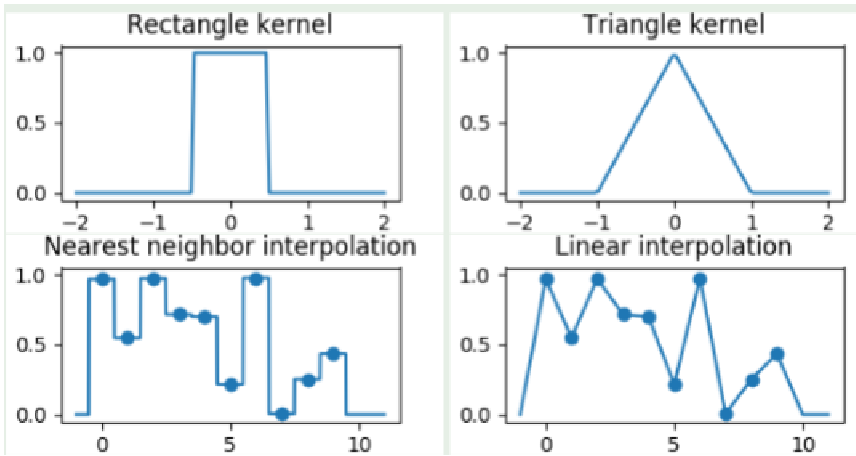
Interpolation

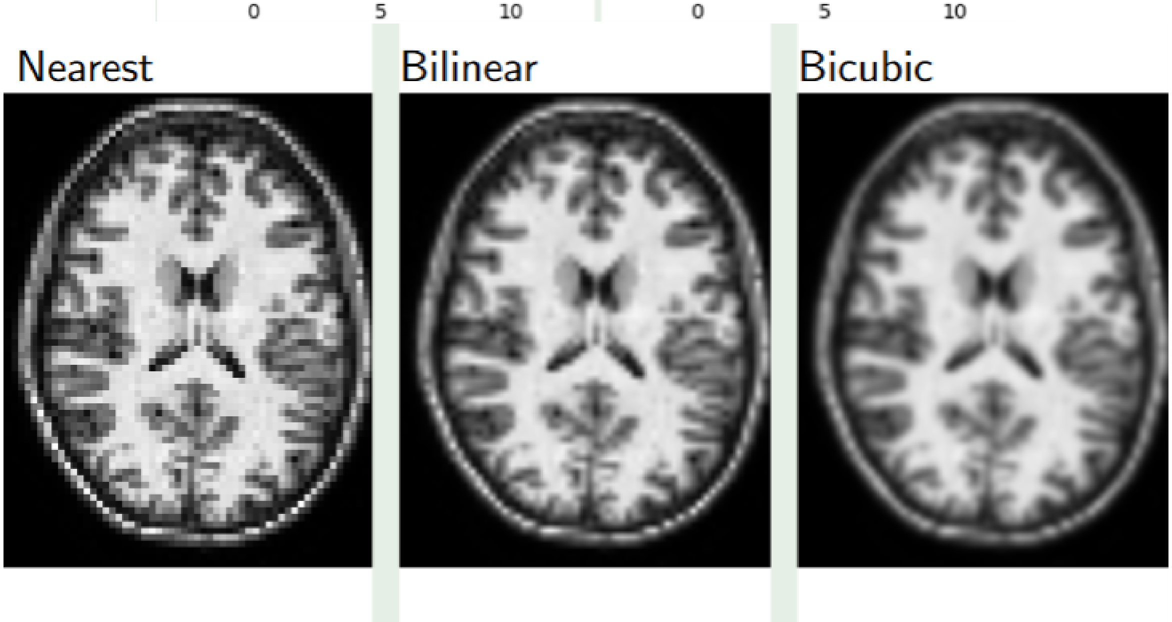
Interpolation links images as functions to images as discrete arrays. If S is a vector space, then we can specify an interpolation kernel h and write

$$I(x) = \sum_{i,j,k} I[i,j,k]h(x - x[i,j,k])$$

where $x[i,j,k] = O + \mathrm{diag}(\Delta)\begin{pmatrix} i \\ j \\ k \end{pmatrix}$, the coordinate of the i, j, k-th pixel.

- $I(x)$ is the image function, and $I[i, j, k]$ is the array of pixels. h is often a function that is close to 1 near the origin and decays down to zero as you get far away.
  - $h()$ is also known as a point-spread function (psf), or an impulse response function.
- What the function basically does is that it multiplies the pixel value by the impulse response function at that real-world location.
- We are convolving the kernels with the discrete pixel dataset. Triangle kernel gives linear interpolation, and rectangle kernel gives nearest neighbor interpolation.



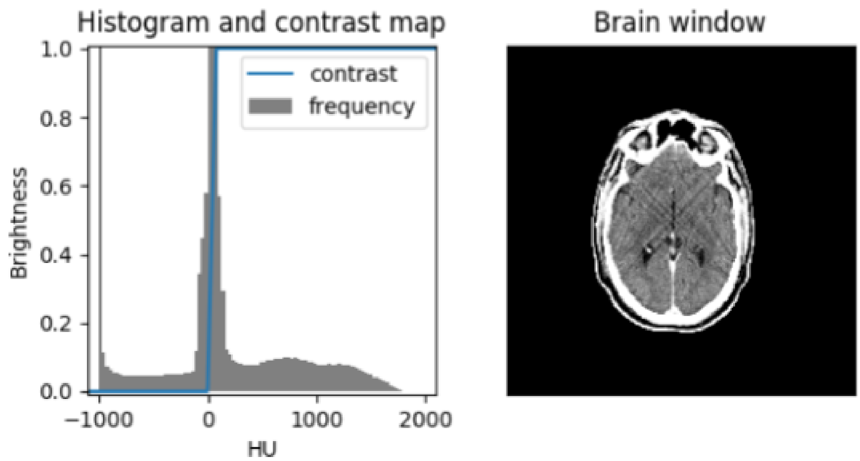


- The image on the left is a brain interpolated (convolved) using the rectangle kernel
- The middle image is one interpolated with the triangle kernel
- The right image is a brain interpolated with a piecewise cubic function (aka. third degree spline).

Importantly, the range of values the function takes is the same as the values. Since the kernel function is always between values 0-1, the minimum value of the interpolated image is 0, and the maximum is the highest original pixel value.

2D grayscale images

- To display images, intensity values at each pixel need to be mapped to a brightness value on the screen (a number between 0 and 1)
- A piecewise linear function is used to do this, by selecting a lower value for black, and an upper value for white.
- A histogram of pixel values can be used to choose these parameters.



- The histogram is one that shows pixel values. The center should be set at the highest peak of the contrast map, e.g., setting brightness and contrast.
- The brightness value is essentially "which pixel value should be mapped to black", and contrast is the "difference between white pixel and black pixel" or, "what pixel value is white relative to black".
- The above image sets black to 0 HU, and white to 80 HU. Anything below the range is black, anything above is white.