Digital Images

Formats and Datatypes

header
magic number
lossy/lossless
dicom
BRIK/HEAD
img/hdr
nifti
p-file

grecons
signed/unsigned
char/uchar
short/ushort
float
double
label images
header size

Why do you care?

If you work with images, you will convert between formats.

You must make the right choices, or the images can be mysteriously messed up.

Understanding image formats,

and how and where the information for an image is stored,

will help you make the right choices and keep your images safe.

What are Image Formats?

Each format is a standard for organizing and storing images.

e.g., tiff, jpg, dicom, BRIK

Programs must recognize and understand the image format in order to display it.

Much of the information necessary to display an image is stored in the header.

Image & Header

An **image** is a set of numbers representing pixel values:
1111000011110000

The **header** explains how to display the numbers. e.g., how many rows and columns? 1x16? 2X8? 4x4

Below is the same raw image data displayed in 3 different ways depending on the header information.

1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0

1	1	1	1
0	0	0	0
1	1	1	1
0	0	0	0

Headers

Other information may be stored in the header:

For MRI images: the date, scanner coordinates, patient name, FOV, Big Endian/Little Endian…

For digital camera images: GPS coordinates, date and time, camera settings…

More Headers

Exactly what the header contains depends on the image format.

Headers often occupy a predetermined number of bytes at the beginning
of an image file (e.g., jpg, gif, nii),

But, the header may also be a separate file **paired** with the raw image (img/hdr, BRIK/HEAD)…in which case, they **have to stay together and have matching names** (e.g., brain1.hdr goes with brain1.img NOT brain2.img)

Magic Numbers (File Signatures)

- Many files contain a signature (magic number) that identifies them to a program.
- In an image file, the signature is in the header.
- e.g., JPG image files begin with 0xffd8ffe0

Lossy and Lossless Compression

Images use a lot of file storage space compared to text.

There are 2 types of image file compression:

Lossless compression reduces file size without quality loss.

Lossy compression discards information that is mostly invisible to the eye.

Graphics Interchange Format

- A lossless bitmap format introduced by Compuserve in 1986.
- 8 bit image (256 distinct colors)
- Supports animation
- Supports transparency
- Widely used on web pages.
- Great for icons, not so good for photographs.

Joint Photographic Experts Group

A bitmap image format, standardized in 1992; excellent for photographs (smooth gradients); not so good for line drawings or text (sharp contrasts).

Highly compressible, to save storage space; but "lossy" (compression reduces quality).

Not appropriate for scientific imaging where information must not be lost.

Tagged Image File Format

A flexible lossless (does not degrade when compressed) format that supports extremely high color depth.

Allows layers, allows some vector graphics.

Good for scientific imaging.

Not supported by web browsers.

Images tend to require lots of storage space.

Portable Network Graphics or "PNG's Not GIF"

Lossless, created to replace GIF, 1996

Supports 24 or 48 bits (not just 8 bit)

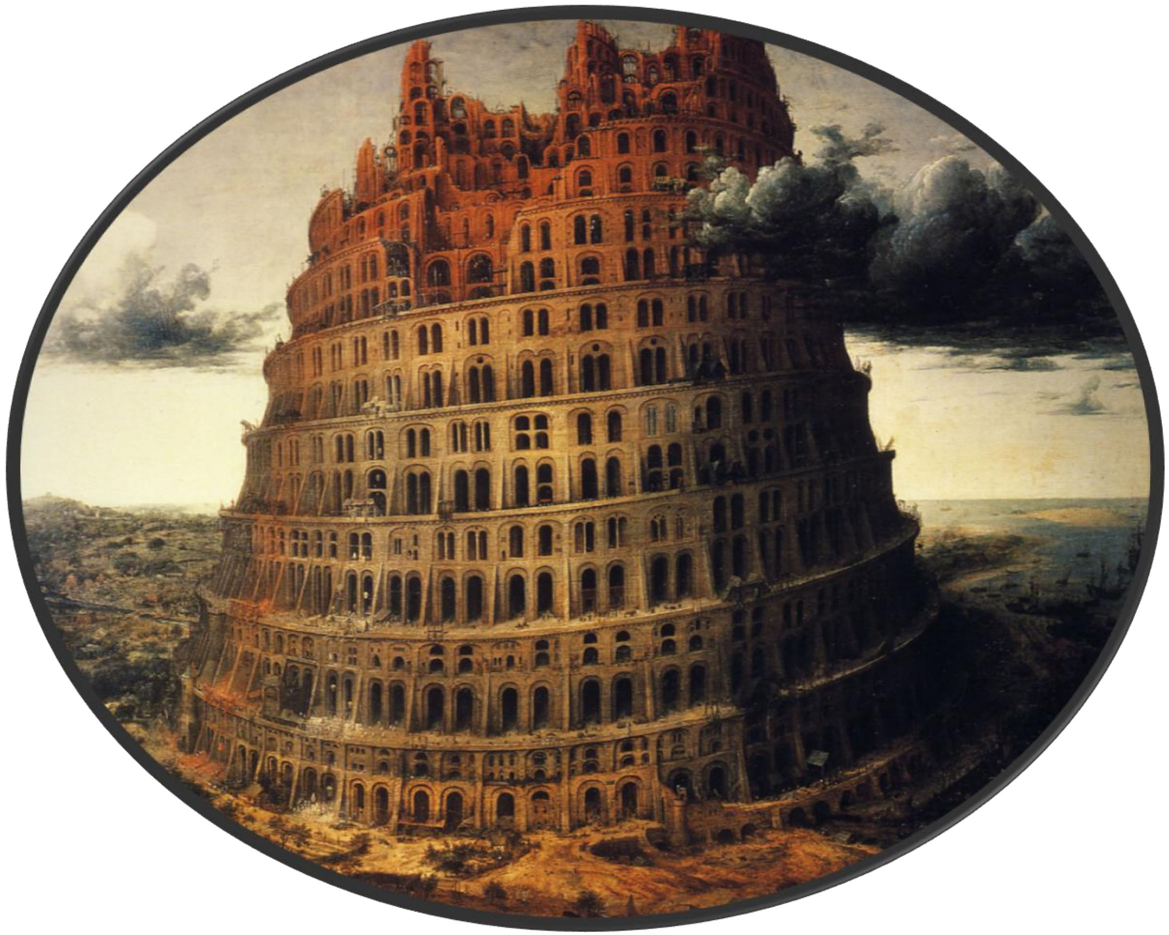
Has transparency and animation

Supported by web browsers

Relatively new, so not as well supported as GIF.

Medical Image Formats

The Tower of Babel problem



- Dozens of neuroimaging analysis programs developed over the last decade, often each with its own image format.

Bruker

GiFTI

HRRT

BRIK/HEAD

nrrd

img/hdr

COR

MINC

Interfile

ECAT

PFF

MedX

NetCDF

GIPL

- Converting between formats is a major headache that makes it difficult to use multiple tools or share data.

BRIK/HEAD

Bruker

GiFTI

nrrd

HRRT

PFF

img/hdr

COR

Interfile

MINC

MedX

NetCDF

ECAT

Digital **I**maging and **Co**mmunications in **M**edicine

Scanners and other medical equipment use DICOM

DICOM includes an image format and networking protocol for image exchange.

DICOM enables integration of scanners, workstations printers etc. from multiple manufacturers into a **P**icture **A**rchiving and **C**ommunication **S**ystem (PACS).

The dicom standard covers a broad range of images, has a long history of changes and is interpreted differently by different manufacturers.

Because of its variability, DICOM is more like a family of standards than a single standard.

P-files

P-files are generated by a spiral scanner sequence that helps correct the distortion in EPI fMRI images.

The spiral sequence was written by Gary Glover at Stanford for GE scanners.

The UofA is lucky enough to be allowed to use the spiral sequence.

Grecons

Each time the scanner software is upgraded, the spiral sequence changes with it.

When the spiral sequence changes, a new grecons is needed.

To reconstruct the image data in a Pfile, you need the correct version of grecons.

fMRI data collected before 2003 requires a very old grecons that only runs on Suns and SGIs.

The CNL website has a table of dates and grecons versions. Except for the oldest one (pre 2003), they are available on the linux machines.

BRIK and HEAD

The native Afni* image format. Scanner images are converted to this format to use them in afni.

*Automated Functional Neuroimaging Program

The BRIK file is the raw image data.

The HEAD file contains the header information, and is paired with the BRIK.

The HEAD file can be displayed by any text editor.

img and hdr

The img and hdr are native to Analyze, and used by SPM (with header modifications).

The img file is the raw image data.

The hdr file is paired with it and contains header information.

Some img/hdr pairs are actually nifti* format images. The hdr contains a magic number indicating this.

More on nifti in a minute…

The Neuroimaging Informatics Technology Initiative

is a collaborative effort that set out to change the "Tower of Babel" problem:

In 2004 they created an image format optimized for neuroimaging, and got promises from all major neuroimaging analysis packages to support that format.

NIFTI

- 5 years later, NIFTI is widely supported: AFNI, FSL, SPM, Mricron, ImageJ, Itk-snap, Slicer3D, Bioimage Suite, Brain Voyager, Mipav, freesurfer, Caret…
- It is a simple, compact, versatile format
- It retains Left/right information (a godsend).
- And it retains positional information from the scanner. This positional information can improve image registration.

NIFTI is a good first step toward alleviating compatibility & conversion problems.

NIFTI continues to evolve.

Image Formats and Supported Data types

Each image format has a range of data types it supports.

- gifs are 8 bit integers.
- pngs support several data types.
- scanner images are 16 or 32 bit.

Image Quantization

Converting an image from one data type to another may cause clipping of the intensity values.

Clipping results in a reduced range.

Image Clipping

Range: 0-255

Subtracted 100:
Range 0-155

Added 100:
Range 100-255

Image Formats and Supported Data types

Results can be severe.

e.g., a floating point image with fractional values from 0 to 1 may lose nearly all values if converted to an integer type, because the integer type can't represent the fractional values.

FSL: Converting the FA (Fractional Anisotropy) image on the left to an integer type causes it to make 0-0.499 black, and 0.5-1 white. Considerable data is lost.

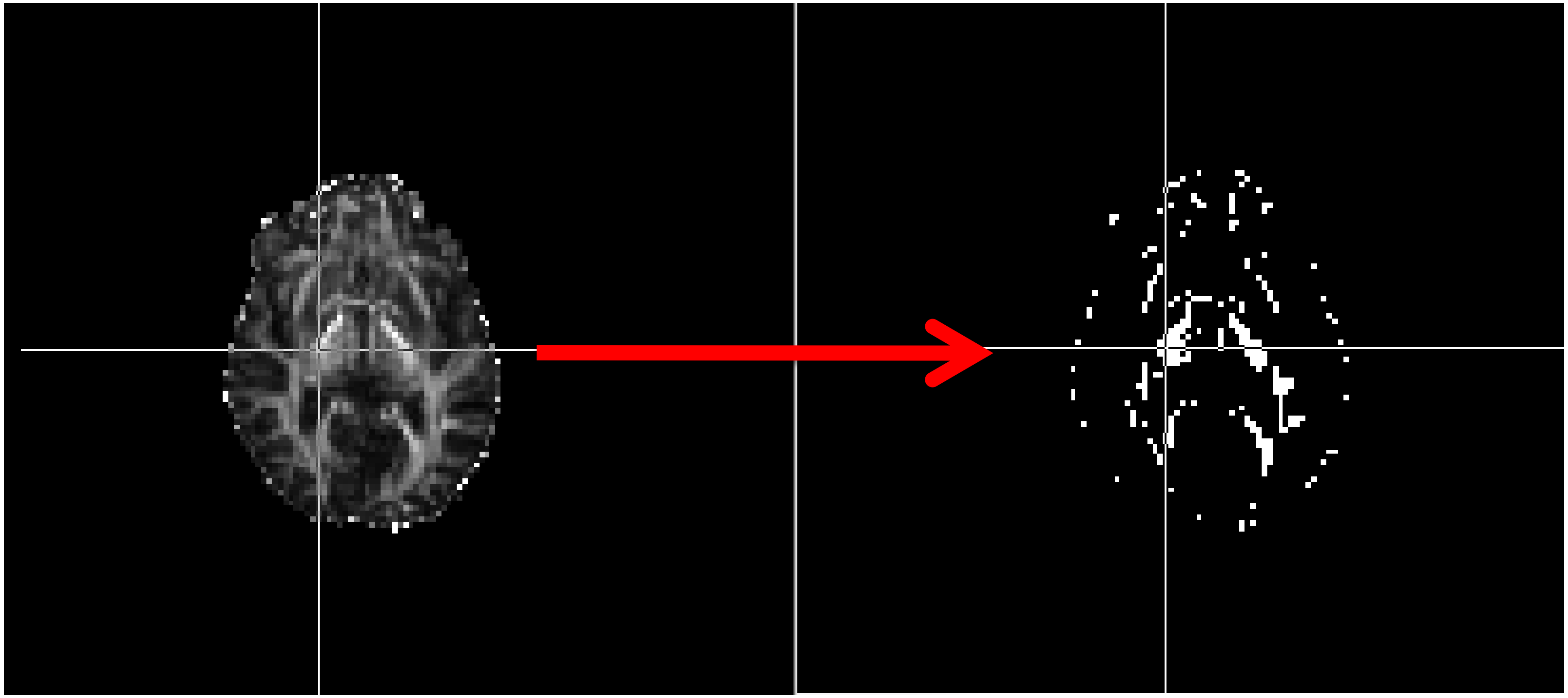
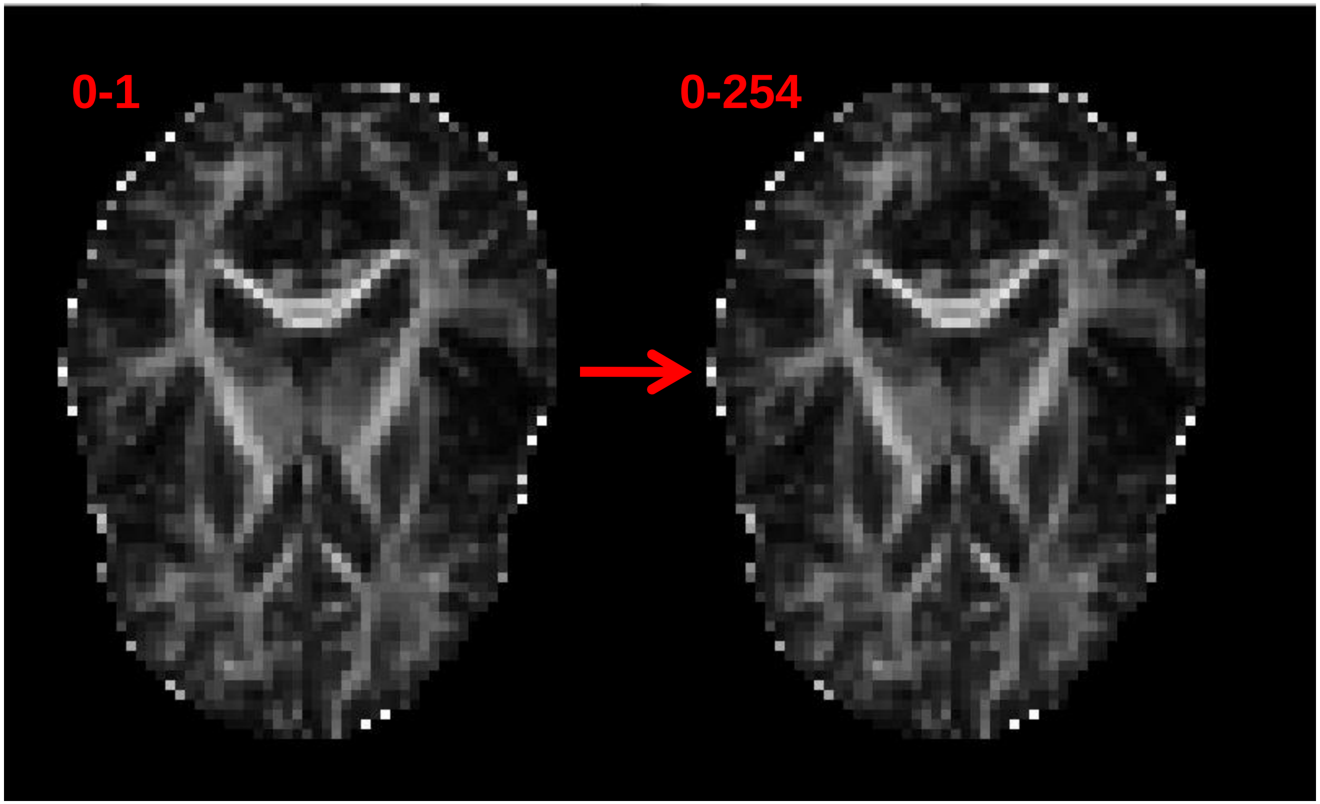


Image Formats and Supported Data types

To make it worse, different programs may define the data types differently, and handle conversion differently.

ImageJ: Converting the FA (Fractional Anisotropy) image on the left to an integer type (8 bit in this case) causes it to try to maintain the look of the original, though the actual value range is changed.

Note that ImageJ displays the images upside down.



So, it pays to be aware of data type of your image and of the kinds of problems that can result from conversion between types.

Signed vs Unsigned

If a 16 bit integer data type is **unsigned**, the values range from 0 to 65,535.

If it is **signed**, the values range from -32,768 to 32,767.

Common Data types

Char: 1 byte per pixel, integers only:
Unsigned Char or **Uchar**: 0 to 255;
Signed Char or **Char**: -128 to 127.

Short: 2 bytes per pixel, integers only:
Unsigned Short (Ushort): 0 to 65535.
Signed Short (Short): -32768 to 32767.

Float – 4 bytes per pixel unlimited range, can store decimal points with a finite precision.

Double – 8 bytes per pixel, same as float but with "double" precision.

Binary

Images that consist of 1's and 0's are very useful in image math (more on this later)

If the 1-0 images are truly binary (1 bit/pixel), they are extremely efficient to store.

Most Medical image processing programs readily create these 1-0 binary images, however, ImageJ creates 0-255 images…they LOOK the same (black and white)…but are different when you do math on them.

8 Bit Unsigned Int

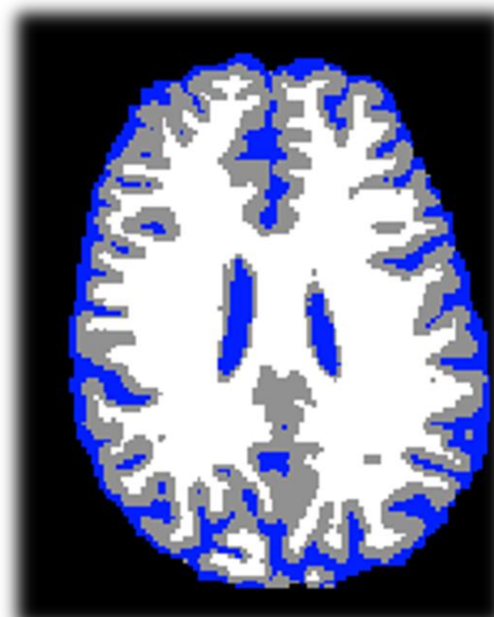
The human eye can't distinguish ~65,000 levels of grey in an MRI image, so 8 bit greyscale images are preferred for viewing/publication.

It is also common to label regions with numbers to indicate atlas structures, segmentation, clusters etc. The resulting images are mask-like…but contain more than 2 values. They are usually 8 bit unsigned.

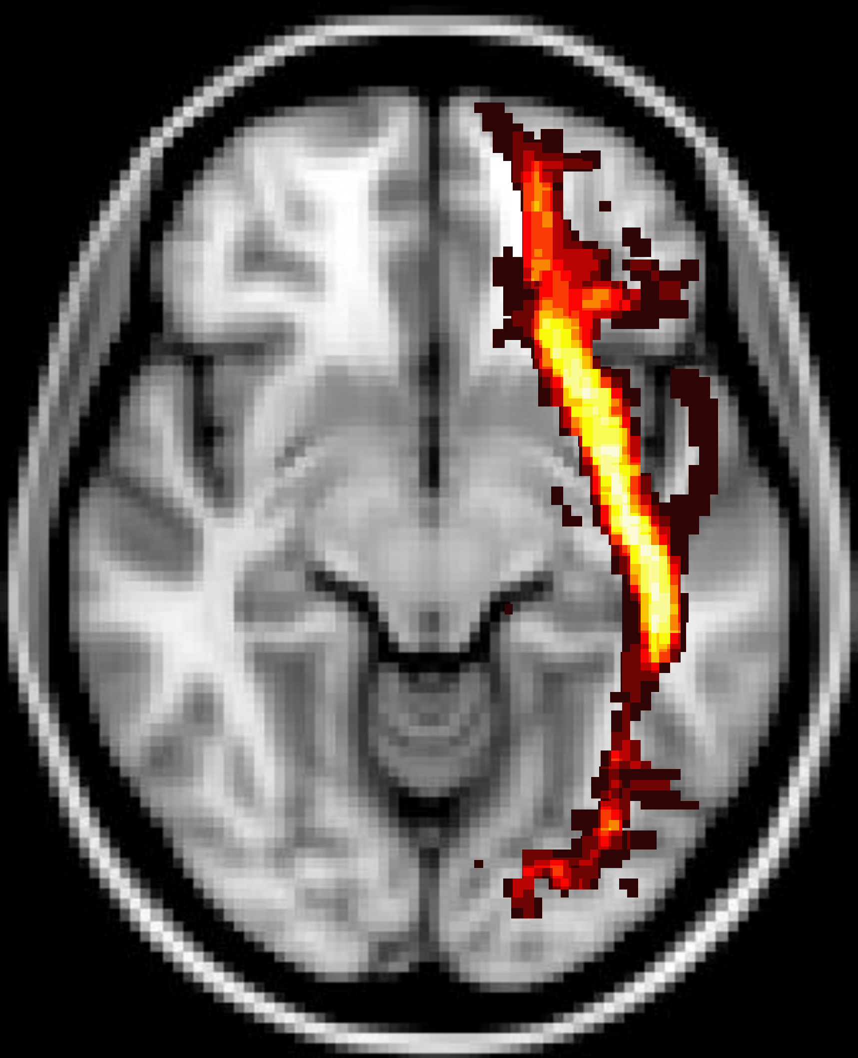
8 Bit Label Images

An atlas provides a dozen or more labels (numbered regions) corresponding to brain structures:

A segmentation labels Grey Matter, White Matter and CSF with 3 different numbers:

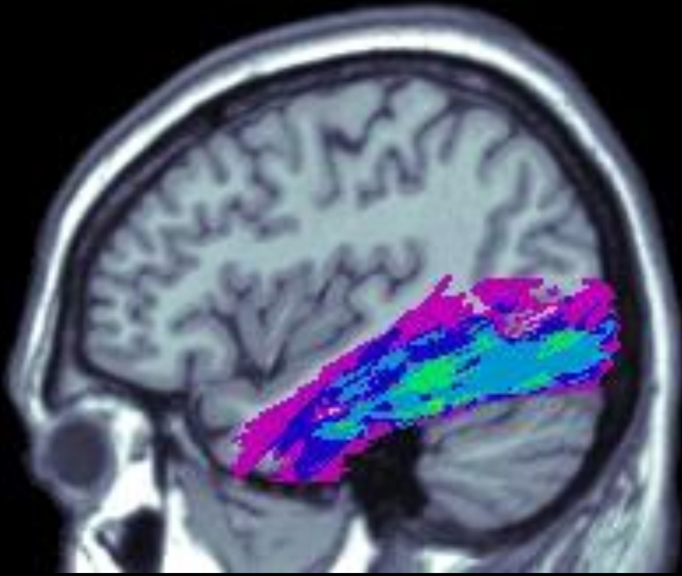


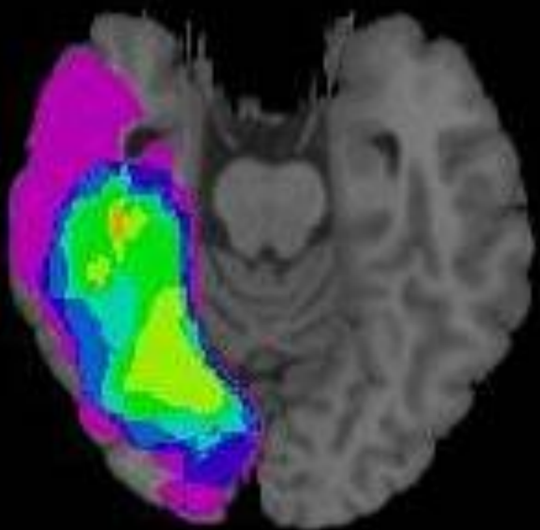
Overlays: 8 bit images



Binary inferior occipital fasciculus masks from 12 subjects.

More 8 Bit Images



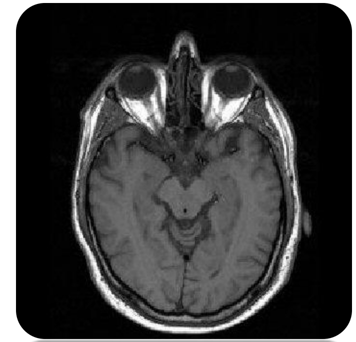


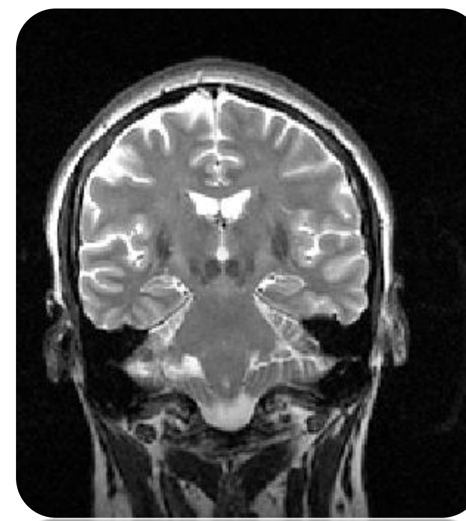
This is a color overlay of binary lesion masks from different subjects. The lesions all cause agraphia. Rapcsak & Beeson (2004). (Thanks to Hyesuk Cho)

16 Bit Ints (Shorts)

Most medical images are signed or unsigned 16 bit ints:

- 16 bit **signed integers** (-32,768 to +32,767), or
- 16 bit **unsigned integers** (0 to 65,535)





32 Bit (Floats)

In a Fractional Anisotropy image used for DTI, the values range from 0 to 1 (roughly)…so we can't use an integer to represent it.

We need a **floating point number** (32 bits) to represent decimal point precision.

Calculate Image & header size

- (Image size on disk-Storage space necessary for raw pixels)=Header Size.

- Flair 16 bit image.nii 7,471,456 bytes.

- 256x256x57=3,735,552 x 2 bytes each=7,471,104 (raw image bytes)

- 7441456-747104=352 bytes header

- header
- magic number
- lossy/lossless
- dicom
- BRIK/HEAD
- img/hdr
- nifti
- p-file

- grecons
- signed/unsigned
- char/uchar
- short/ushort
- float
- double
- label images
- header size