Computational Neuroscience Laboratory

ML-based MRI Preprocessing

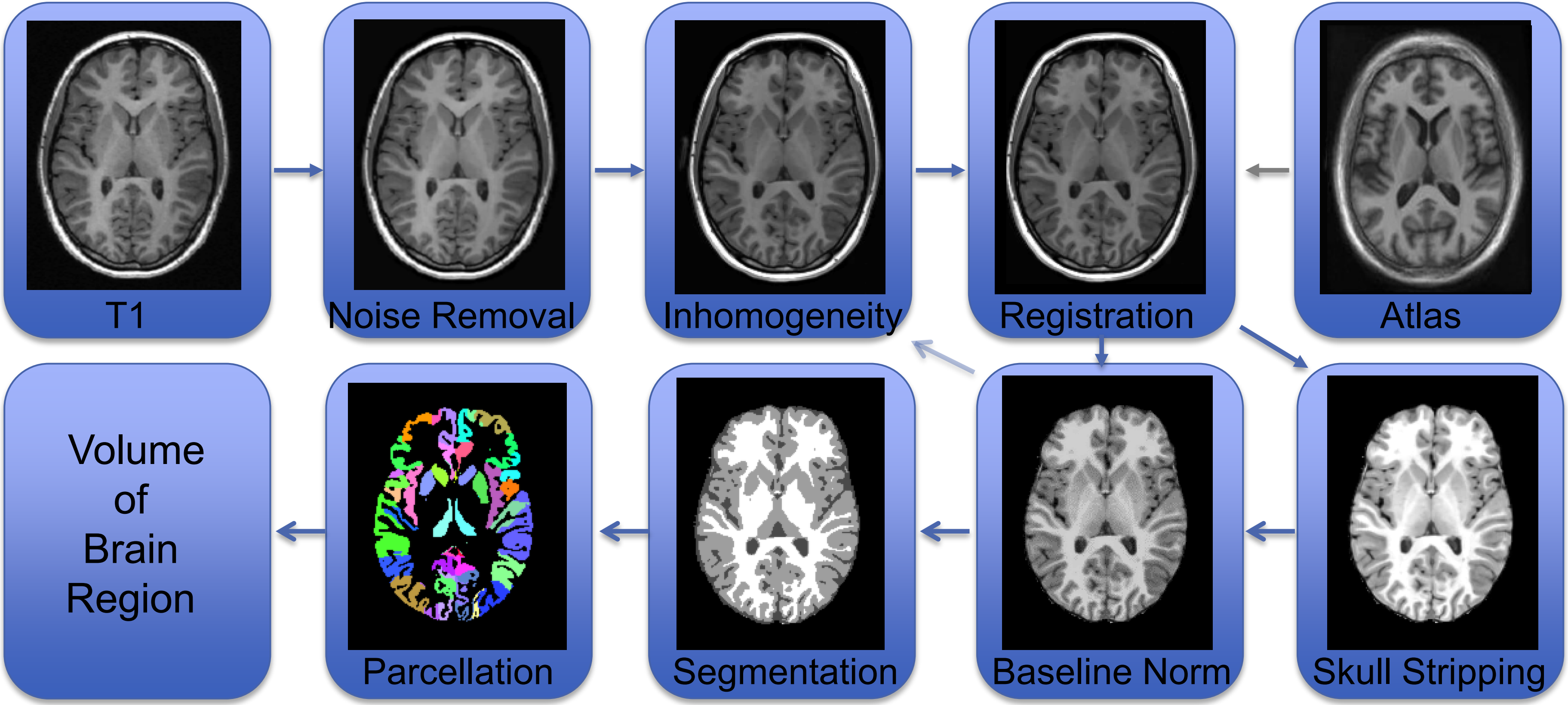
Autumn 2023

Session 7 – 10/17/2023

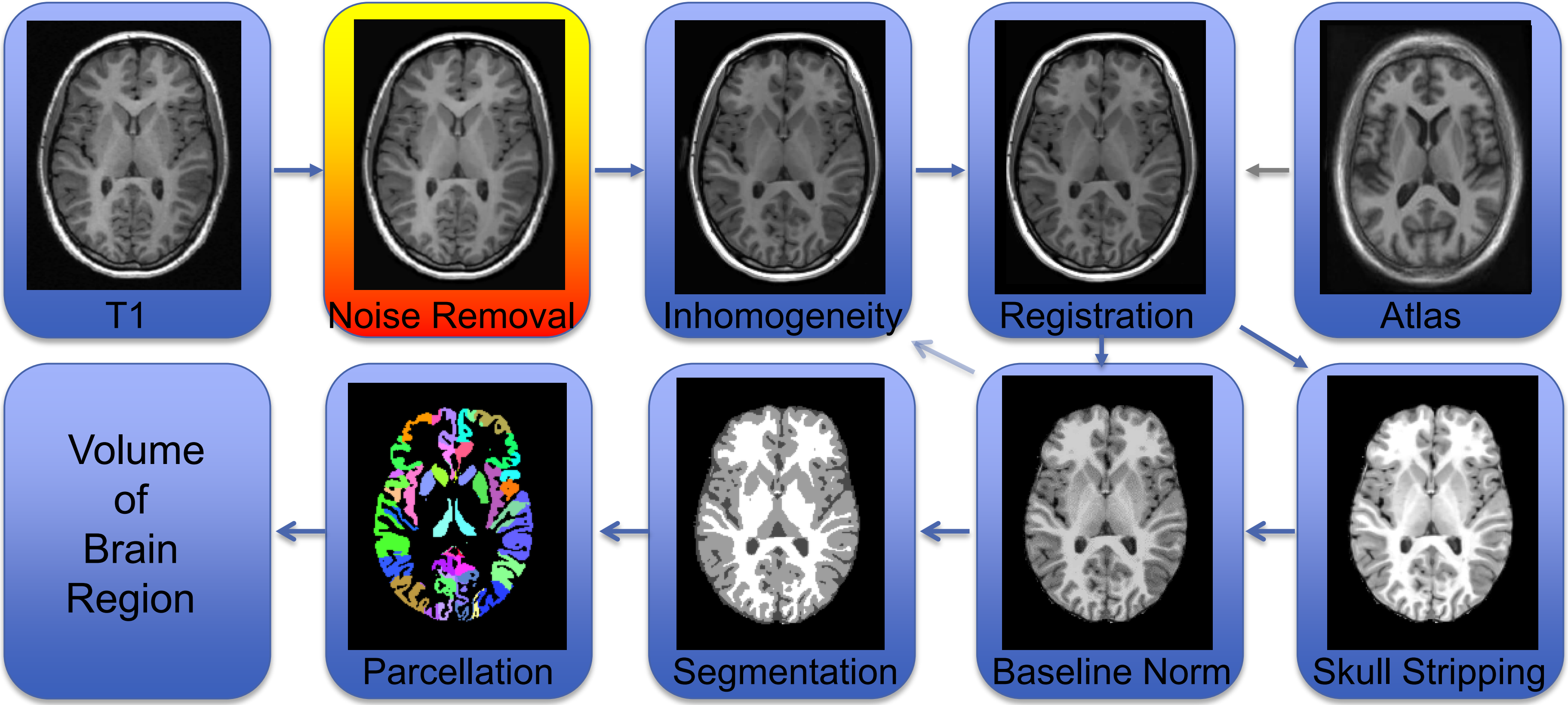
Today…

- Principle Component Analysis (PCA) for Noise Removal
- Expectation Maximization (EM) for Image Inhomogeneity Correction
- U-Net for Segmentation

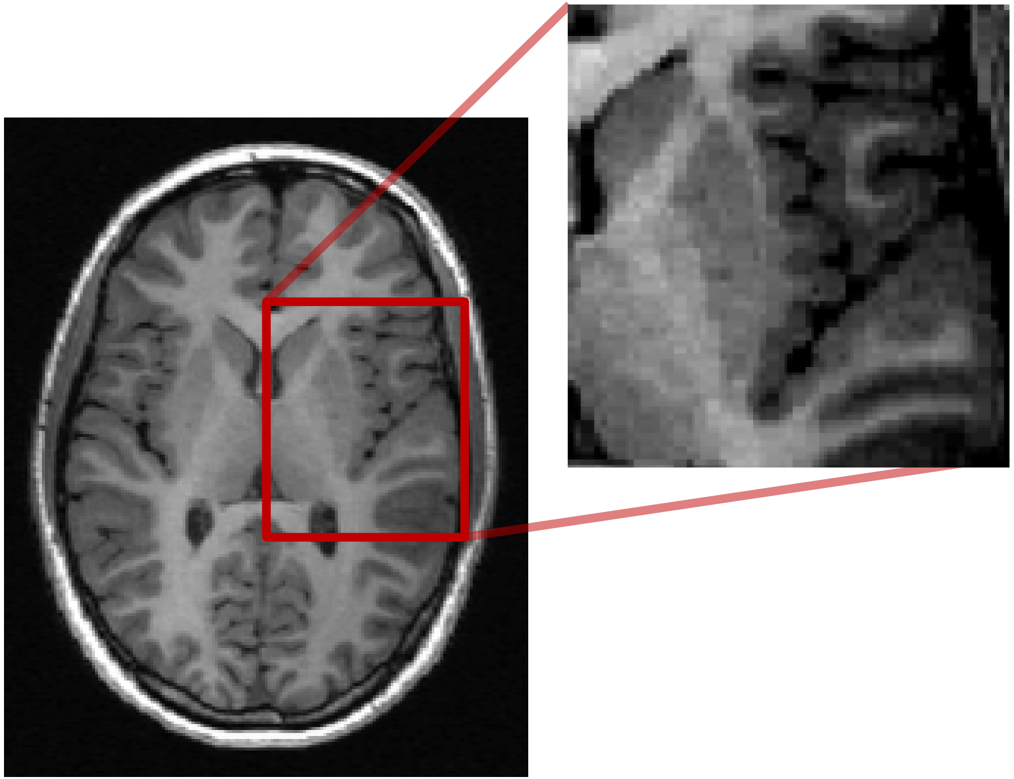
Processing of Structural MRI



Processing of Structural MRI

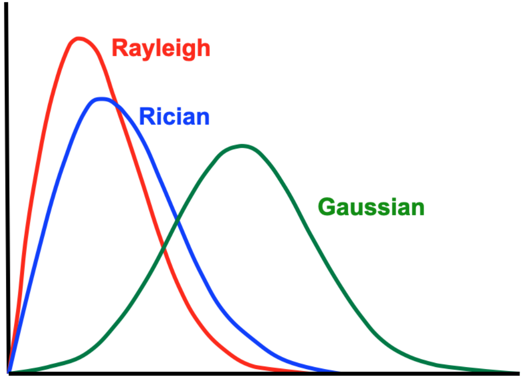


Noise



(magnitude) T1 MRI

By converting the "raw" MRI (defined in complex space with Gaussian noise) into a magnitude image, the noise becomes ***Rician*** distributed.



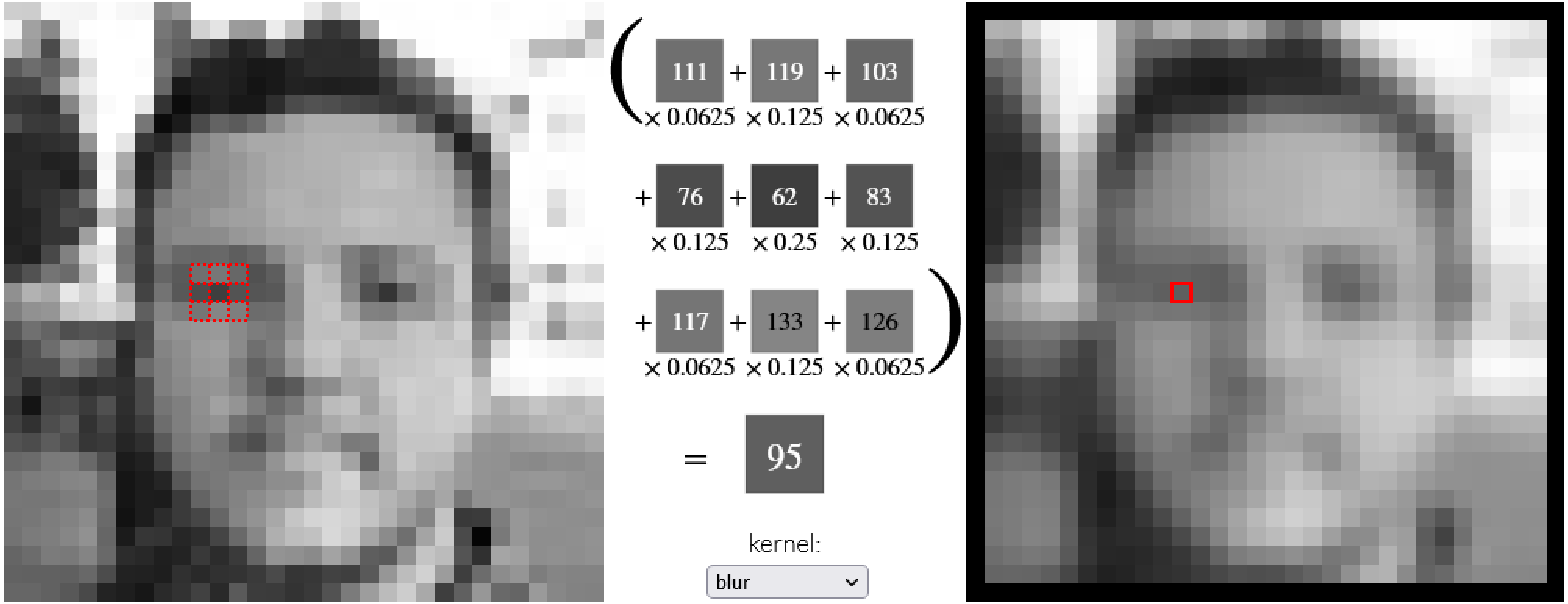
https://mriquestions.com/signal-to-noise.html

Interpret Image as a Matrix

206	206	247	245	244	253	247	245	136	151	255	255	255	255	255	255	234	207	231	255	254	254	255	255	254	255	252	255	255	254	255	247
244	161	137	244	254	255	254	255	118	103	209	228	155	153	236	193	74	52	68	173	255	254	254	255	255	255	254	255	254	253	244	184
192	154	75	200	249	255	255	255	110	98	84	61	35	44	89	53	44	45	43	54	140	213	253	255	255	255	255	245	187	186	176	223
90	109	98	143	223	255	255	252	117	75	41	35	31	24	25	36	45	44	44	46	81	118	148	234	252	254	255	248	231	248	255	254
67	69	107	196	236	255	255	255	104	25	34	35	29	20	25	34	32	30	32	34	53	85	100	142	231	242	247	249	255	255	255	255
55	51	45	134	218	251	255	232	51	12	26	33	24	24	46	75	82	78	71	66	58	53	67	90	136	226	206	158	253	246	249	255
79	58	56	75	224	255	255	118	11	27	74	99	91	106	140	162	173	173	173	172	158	137	92	46	78	187	217	206	254	222	233	255
38	43	47	52	147	255	229	56	41	81	129	145	160	169	169	172	178	179	178	179	177	177	172	110	31	82	209	238	255	244	249	255
40	40	33	36	90	245	171	32	65	110	139	145	151	162	171	174	178	179	182	184	187	183	173	162	71	45	187	255	254	255	254	255
37	44	44	31	69	250	158	36	70	129	143	142	153	162	171	175	177	178	182	191	194	188	180	170	120	51	137	255	254	250	254	255
34	45	51	64	116	237	181	53	116	138	140	143	154	164	176	178	174	177	183	186	186	186	183	178	140	66	141	254	252	225	249	255
34	36	52	74	71	188	156	63	131	134	144	155	160	161	173	179	178	179	189	193	190	185	187	182	156	93	146	250	254	214	247	255
32	38	52	54	159	250	126	57	129	136	138	140	151	156	166	168	171	178	180	187	186	185	185	183	180	102	136	242	255	255	254	254
36	32	72	129	212	228	115	66	121	104	102	104	94	103	134	158	170	162	126	106	121	143	155	190	191	104	134	230	253	253	255	251
61	82	116	107	179	247	124	60	101	90	111	119	103	81	94	147	191	178	126	98	123	153	147	161	200	92	100	222	207	167	227	215
144	176	167	231	210	232	170	67	115	88	76	62	83	85	88	139	192	190	135	80	53	99	141	165	201	97	79	192	245	235	248	249
127	145	149	195	204	213	197	95	133	122	117	133	126	108	110	139	191	197	167	129	127	148	147	171	188	110	121	228	233	180	215	212
87	112	100	79	85	82	65	75	142	148	151	153	138	125	120	149	191	190	193	175	174	193	198	190	208	127	163	239	219	149	198	195
63	83	109	134	129	106	39	78	132	142	155	159	139	111	124	164	195	200	188	192	191	195	200	202	200	143	217	253	249	242	238	254
69	78	78	113	97	74	43	106	127	140	152	155	125	97	112	150	185	194	174	183	196	198	202	208	209	186	247	254	255	254	254	254
72	44	63	59	46	52	49	74	127	137	146	149	132	103	78	90	134	141	168	165	199	207	204	203	216	193	236	244	251	242	238	243
55	20	69	73	59	80	46	74	117	127	144	161	146	124	105	120	156	187	193	162	189	206	201	205	214	194	174	185	197	188	183	193
65	49	77	89	50	68	43	61	109	127	141	147	113	100	121	145	$\boxed{148}$	169	181	176	181	201	201	205	202	174	166	169	176	183	188	184
82	76	92	79	54	58	37	47	90	121	132	116	89	78	111	146	163	149	122	124	180	197	197	198	178	149	146	152	155	157	159	168
104	107	122	123	105	79	27	33	68	111	122	120	114	114	147	175	190	196	163	101	170	200	187	185	156	146	145	139	137	141	140	145
117	124	127	133	135	105	21	28	37	88	115	121	128	128	141	142	168	202	212	153	164	186	180	168	154	146	144	149	151	151	147	144
119	118	118	125	128	111	21	29	28	58	100	118	131	140	151	159	186	201	206	192	180	168	149	166	119	144	147	143	140	141	144	148
117	119	125	130	139	106	18	29	44	58	70	102	133	147	168	197	212	215	210	195	177	152	133	126	57	59	126	151	145	143	142	141
115	123	126	134	145	102	27	54	52	38	45	69	105	135	175	189	193	216	206	166	139	111	164	203	74	5	121	151	142	142	143	146
101	108	123	121	132	105	44	40	31	35	57	44	58	101	147	144	138	163	145	94	90	145	196	187	84	48	166	160	142	144	142	145
98	97	97	98	104	78	34	33	30	48	41	49	51	58	74	53	55	66	63	89	150	188	209	158	62	108	140	149	125	133	131	131
102	102	97	88	73	35	30	23	42	50	65	41	90	60	59	51	57	82	123	157	187	205	169	62	98	151	105	101	154	135	130	129

demo

Gaussian Smoothing



input image

output image

Single Value Decomposition (SVD)

The SVD of a *m-by-n* matrix **A** is given by the formula :

$$\underset{n \times d}{\mathbf{A}} = \underset{n \times n}{\mathbf{U}} \; \underset{n \times d}{\mathbf{W}} \; \underset{d \times d}{\mathbf{V}^T}$$

Where :

- **U** is orthonormal eigenvectors of $\mathbf{AA}^T$, i.e., $U^TU = I$
- $\mathbf{V}^T$ is the transpose of the orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$, i.e., $V^TV = I$
- **W** is a diagonal matrix of the *singular values, i.e.,* the square roots of the eigenvalues of $\mathbf{V}^T$ (or U)

for proof see https://web.mit.edu/be.400/www/SVD/Singular_Value_Decomposition.htm

Single Value Decomposition (SVD)

The SVD of a *m-by-n* matrix **A** is given by the formula :

$$\underset{n \times d}{\mathbf{A}} = \underset{n \times n}{\mathbf{U}} \; \underset{n \times d}{\mathbf{W}} \; \underset{d \times d}{\mathbf{V}^T}$$

Where :

- **U** is orthonormal eigenvectors of $\mathbf{AA}^T$, i.e., $U^TU = I$
- $\mathbf{V}^T$ is the transpose of the orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$, i.e., $V^TV = I$
- **W** is a diagonal matrix of the *singular values, i.e.,* the square roots of the eigenvalues of $\mathbf{V}^T$ (or U)

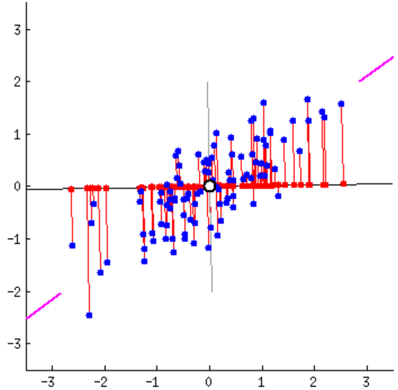
$$\begin{bmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ 0 & \ddots & \vdots \\ \vdots & & \sqrt{\lambda_n} \\ 0 & & 0 \\ 0 & \cdots & 0 \end{bmatrix}$$

Principle Component Analysis (PCA)

- Eigenvalues and their corresponding eigenvectors are ranked according to importance

$$\lambda_1 > \lambda_2 > \dots \lambda_n \geq 0$$

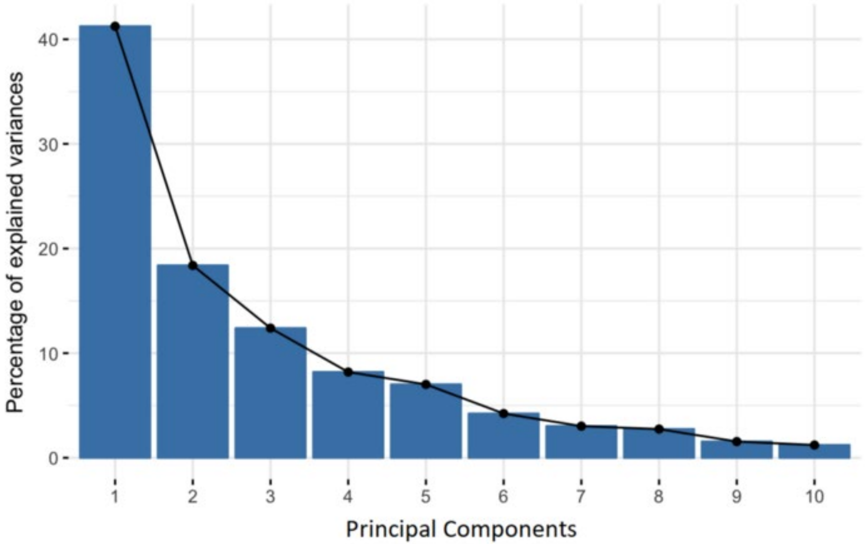
- 1st Eigenvector (EV) is the direction of maximum variance (i.e., λ_1)



https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

- 2nd EV is the direction orthogonal to 1st one with maximum variance

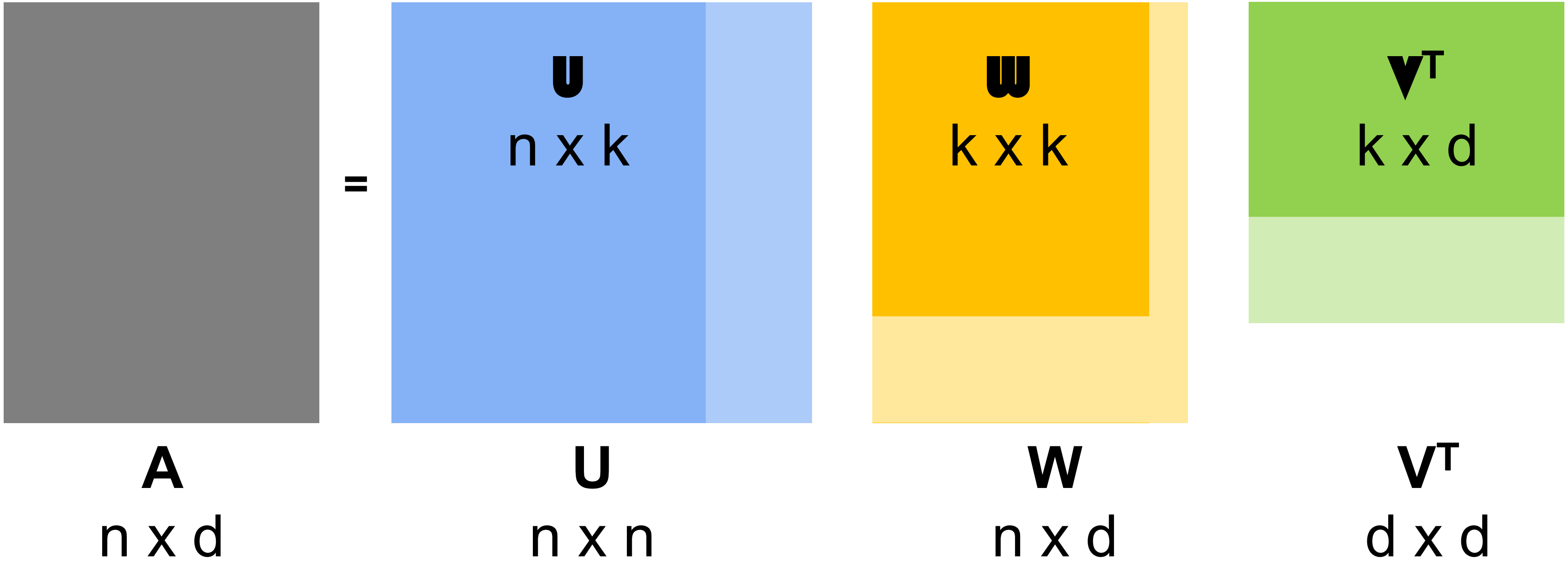
Selecting the Number of Eigenvectors



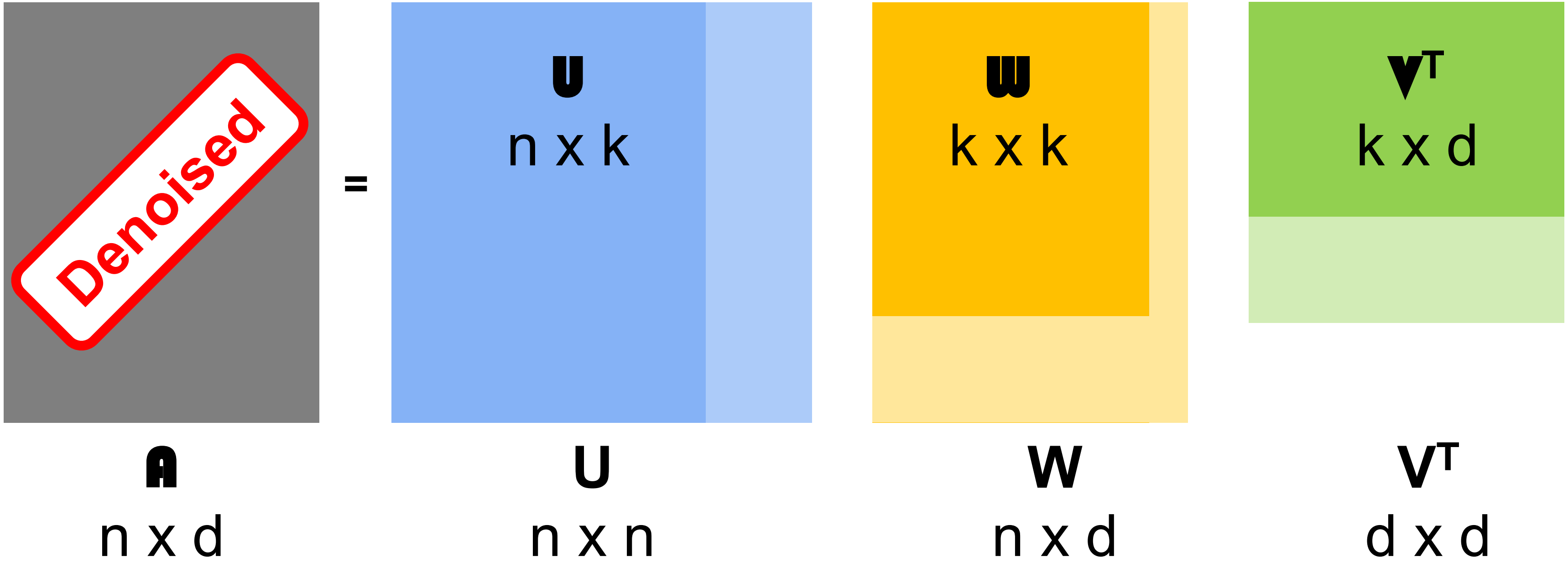
Number of Eigenvectors should explain 95% for variance, i.e., k =7

Vectors 8-10 are viewed as “noise”

Reconstruct Image



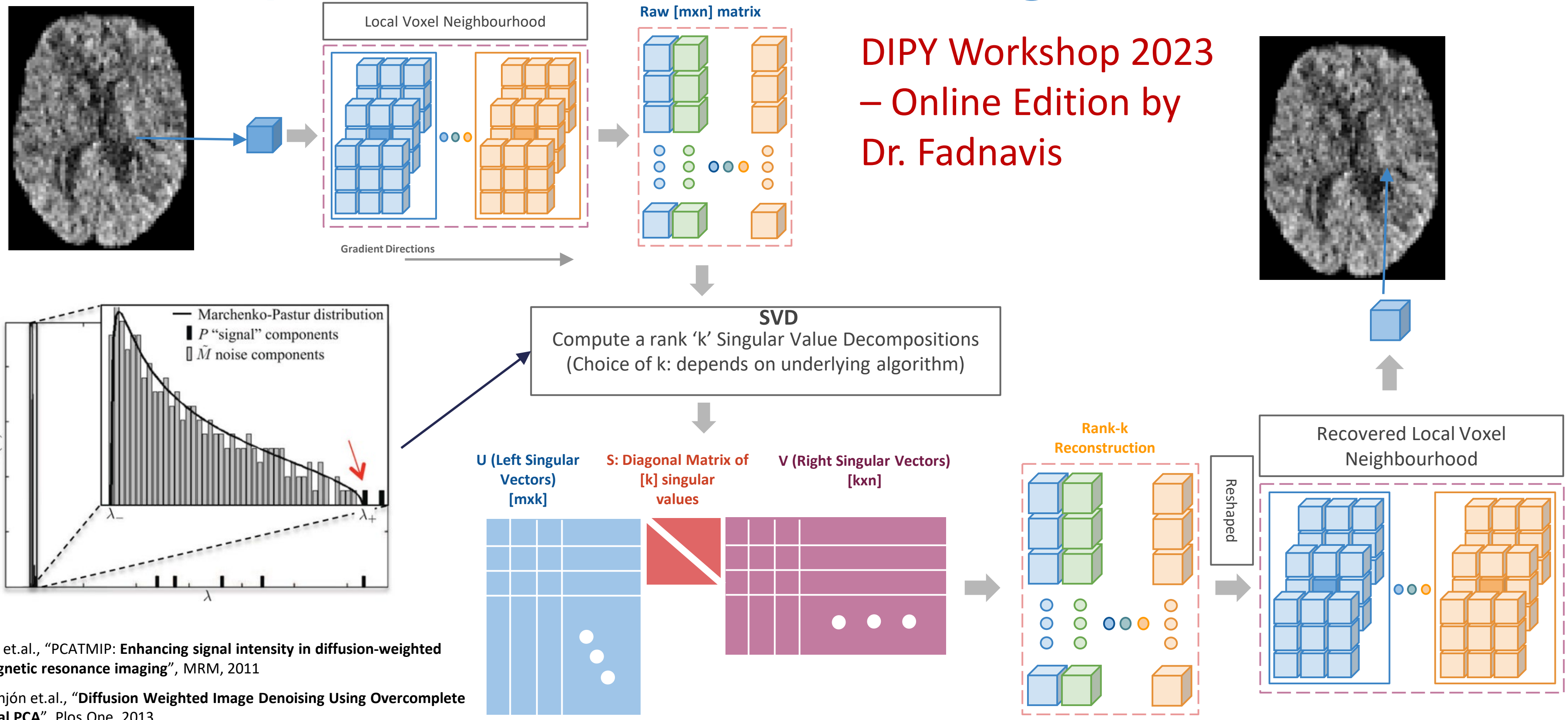
Reconstruct Image



Problem:

capturing anatomical details with global PCA is difficult as constrained to linear relationship => to much smoothing

How to perform local smoothing



DIPY Workshop 2023 – Online Edition by Dr. Fadnavis

Pai, et.al., "PCATMIP: **Enhancing signal intensity in diffusion-weighted magnetic resonance imaging**", MRM, 2011

Manjón et.al., "**Diffusion Weighted Image Denoising Using Overcomplete Local PCA**", Plos One, 2013

Implementation in DIPY

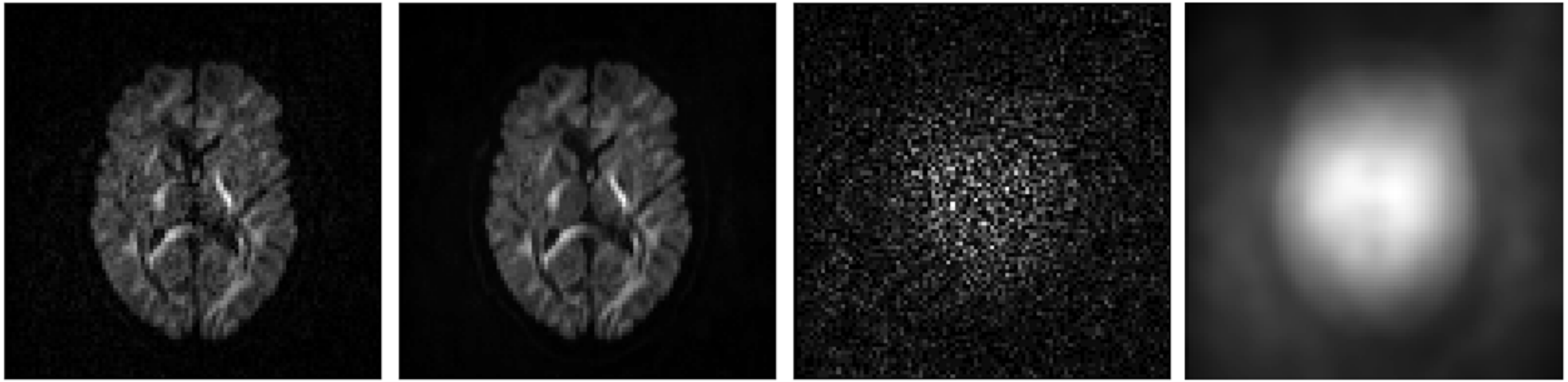
Modules you need:

```python
from dipy.denoise.localpca import mppca
```

Perform the denoising:

```python
denoised_arr, sigma = mppca(data, patch_radius=2, return_sigma=True)
```

Noisy Data | Denoised Data | Residuals | Standard Deviation



See also [tutorial]

Other ML Based Denoising Tools

Low Rank Approximations:
Approximate the data to a smaller basis, typically via a low-rank approximation and then reconstruct from the low rank approximations.
Rank *'r'* is the parameter that needs to be found for doing such an approximation.
Typically done via PCA for Diffusion MRI.

Local PCA (Manjon et. al., 2013)
Marchenko Pastur PCA (Veraart et. al., 2016)

Sparsity:
Images are often sparse and this property can be leveraged for denoising the data. These methods require a predetermined basis set to do the denoising such as DCT, DWT, etc. or by learning the basis from the data, known as dictionary learning.

K-SVD Overcomplete Dictionary Learning - Aarhon and Elad 2006
Denoising and fast diffusion imaging with physically constrained sparse dictionary learning, Gramfort, et. al., 2014

Self-Similarity:
Images are often self-similar, in that each patch in an image is similar to many other patches from the same image.

Image denoising with **block-matching and 3D filtering** (Dabov, 2006)

NL-Means (Buades et. al. 2005) and its variants ONLM, AONLM, etc.
BM3D (Dabov et al., 2007) thresholds in frequency space.

Statistical Independence:
Since noise originates from random fluctuations, noise in one subset of an image can be assumed to be statistically independent of the noise in another subset. Using one subset to predict the other gives denoising performance.

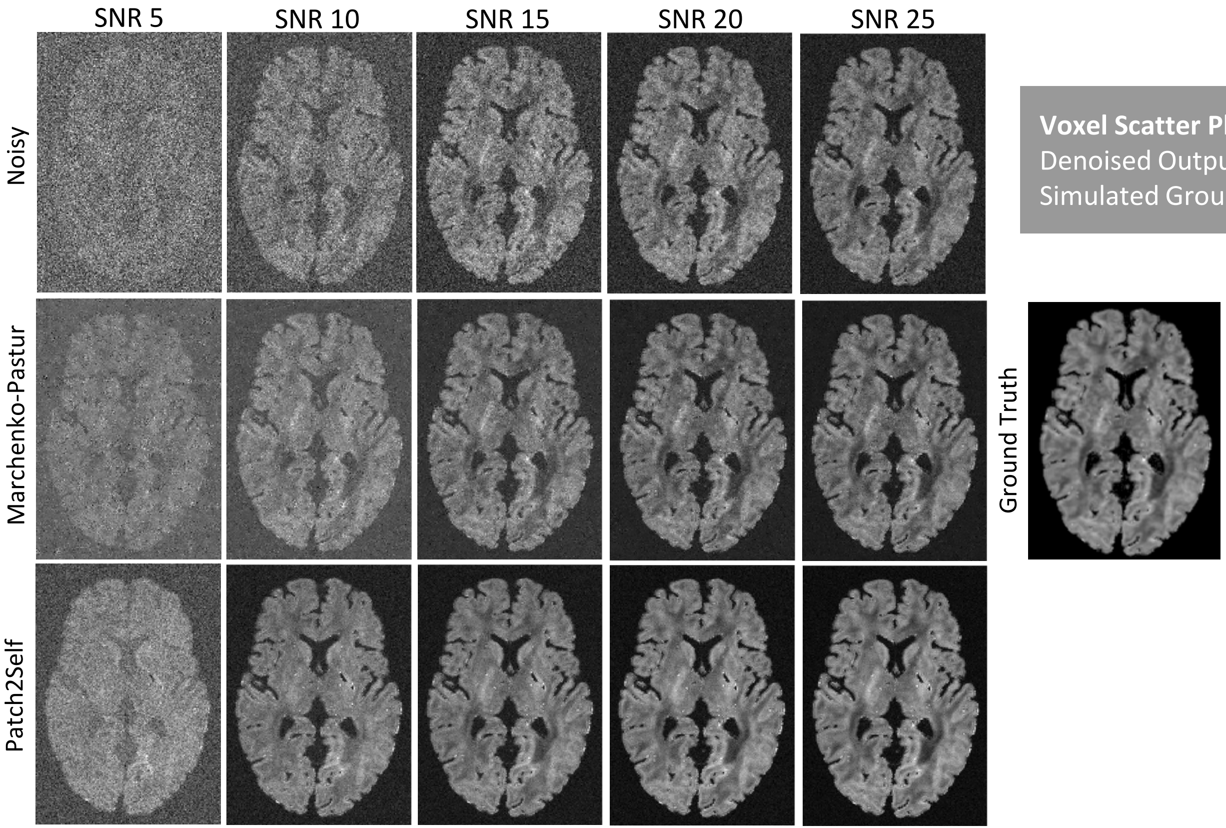
Patch2Self, Fadnavis, et. al. 2020
Noise2Self, Batson, et. al. 2019
Noise2Noise, Lehtinen, et. al. 2018

Slide by Shreyas Fadnavis

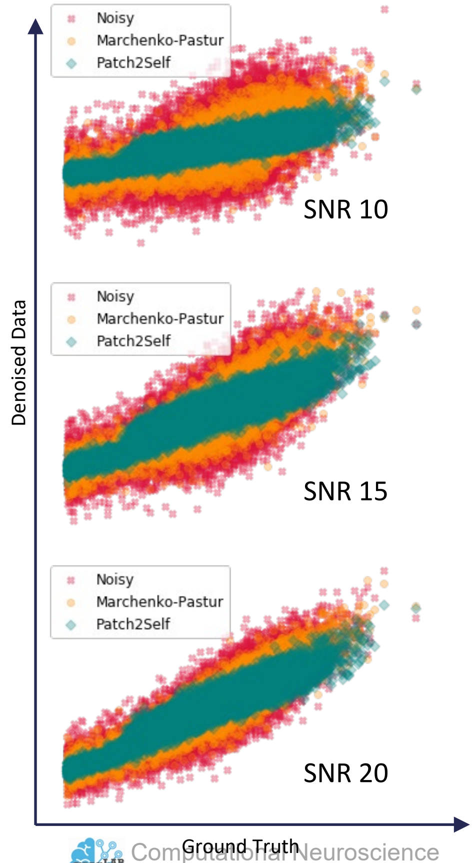
Examples: https://dipy.org/documentation/1.7.0/examples_built/#preprocessing

Outcome

DIPY Workshop 2023



Voxel Scatter Plots:
Denoised Outputs *vs*
Simulated Ground Truth



Notes on PCA

Tutorials

- Fadnavis' tutorial on youtube
- PCA explained step-by-step

Other applications

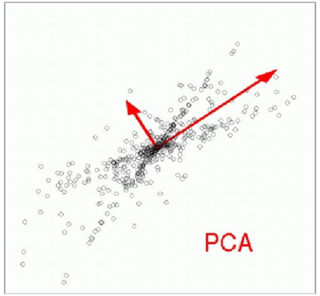
- Identifying group differences (e.g., via Voxel Based Morphometry)
- Data compression
- Clustering
- …

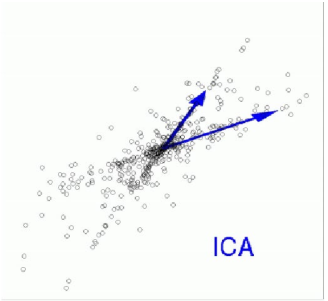
Related but Different

Independent Component Analysis (ICA)

transforms data into independent components so that they are at least correlated to each other, which are

- not orthogonal

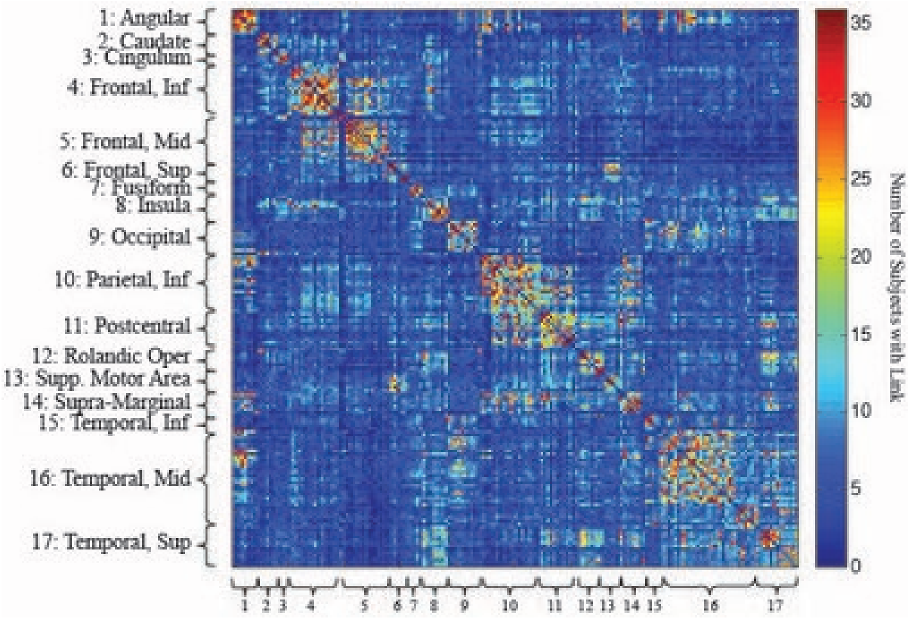


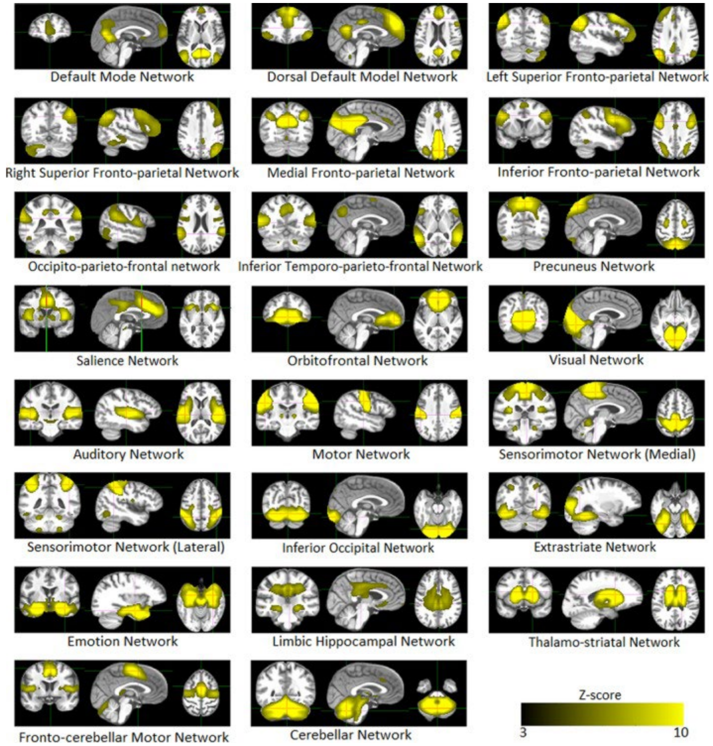


http://compneurosci.com/wiki/images/4/42/Intro_to_PCA_and_ICA.pdf

- equally important
- removes higher order dependencies (in addition to correlations)

Application: Functional Brain Networks



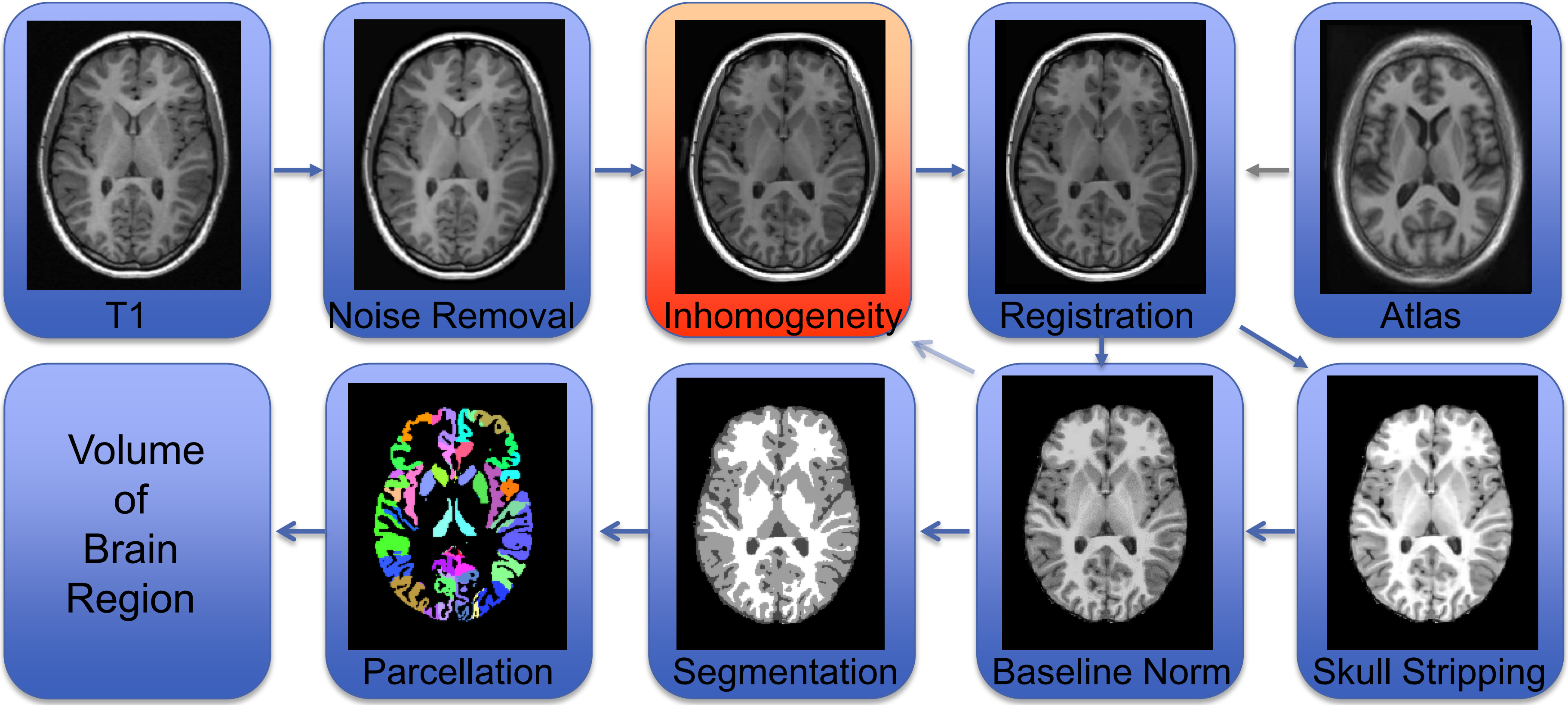


Zhao et al., Human Brain Mapping , 2019

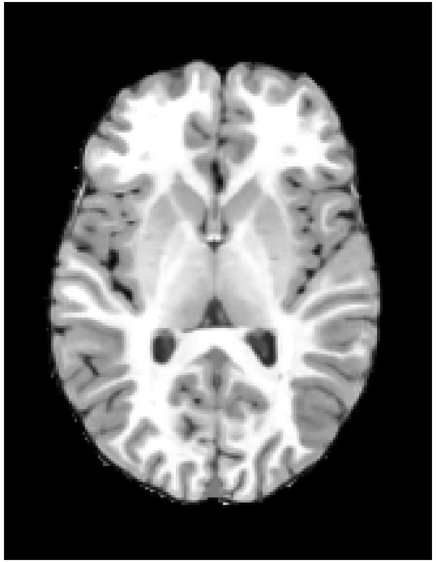
Today…

- Principle Component Analysis (PCA) for Noise Removal
- Expectation Maximization (EM) for Image Inhomogeneity Correction
- U-Net for Segmentation

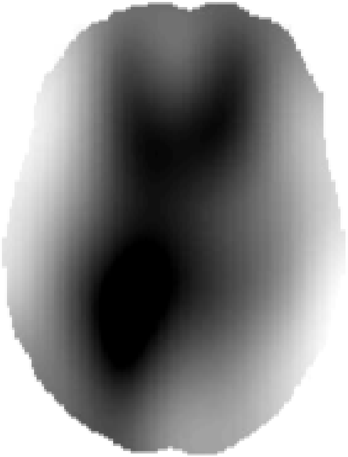
Processing of Structural MRI



Problem: Estimating Image Inhomogeneity



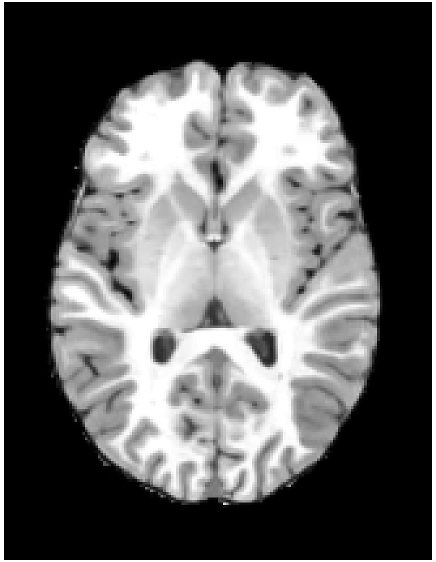
Observed
X



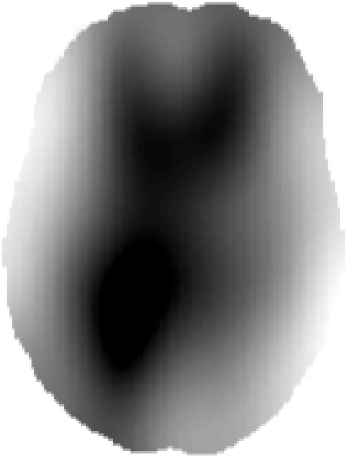
Parameter
θ

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta \in \Theta} \ln \mathcal{P}(\mathbf{X}|\theta)$$

Problem: Estimating Image Inhomogeneity



$$\mathcal{P}(\mathbf{X}|\theta)\ ?$$



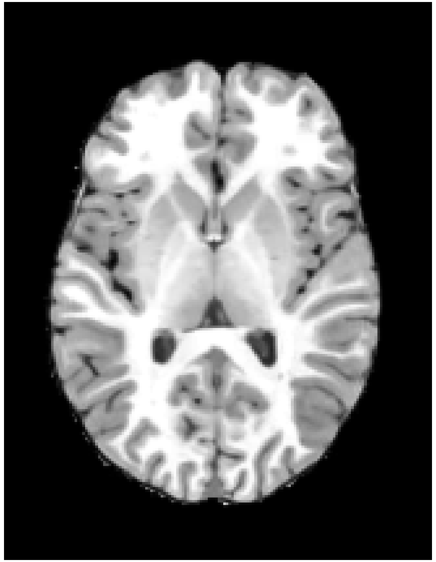
Observed
X

Parameter
θ

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta \in \Theta} \ln \mathcal{P}(\mathbf{X}|\theta)$$

Multiplicative Gain Field

Note: using log transform then becomes additive

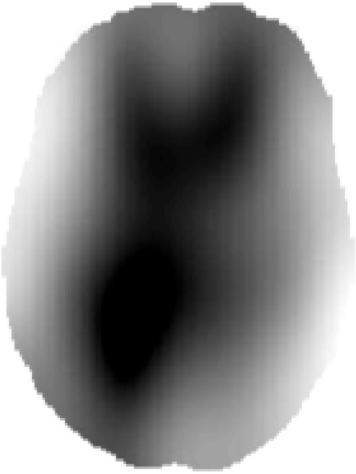


Log (MRI)
Observed X

=

Log (True Intensities)

+

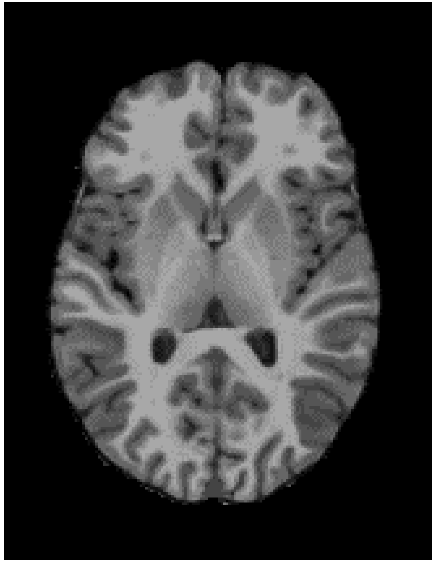


Log (Inhomogeneity)
Parameter θ

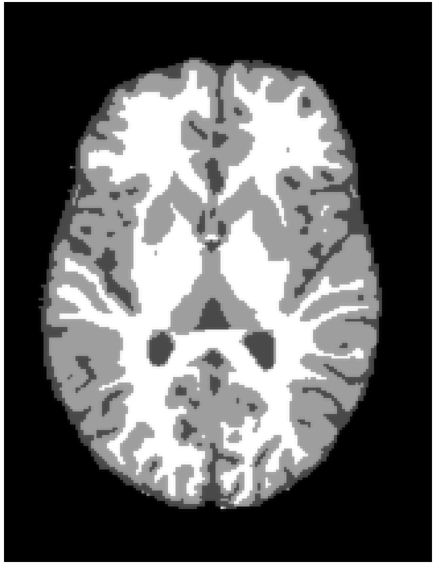
+

White Gaussian
Noise N

How to determine true intensities

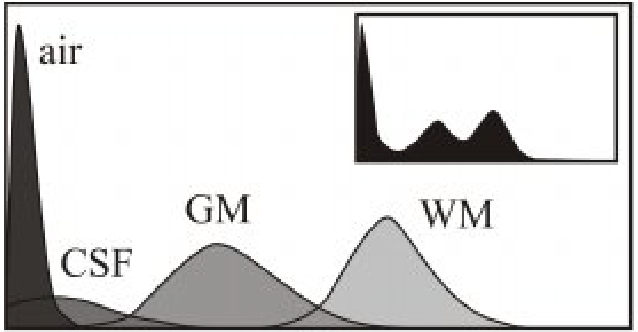


Log (True Intensities)



Label Map

Hidden Variable z



Intensity Distributions
(Additional Parameters)

Deriving Expectation Maximization Algorithm

X = MRI
z = Label Map
θ = Inhomogeneity

$$\ln \mathcal{P}(\mathbf{X}|\theta) = \ln \sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)$$

Given estimate θ_n and Jensen Inequality

$$\geq \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)} \right)$$

Maximizing lower bound for θ results in

$$\begin{aligned}\theta_{n+1} &= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \right\} \\ &= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}) \right\}\end{aligned}$$

Expectation Maximization

E-Step: At each voxel location compute for each structure z the weights

W(z):= $\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)$

X = MRI
z = Label Map
θ = Inhomogeneity

M-Step: Determine $\theta_{n+1} = \arg\max_{\theta} \sum_{\mathbf{z}}$ W(z) $\ln \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z})$,

which is according to *

= R · H

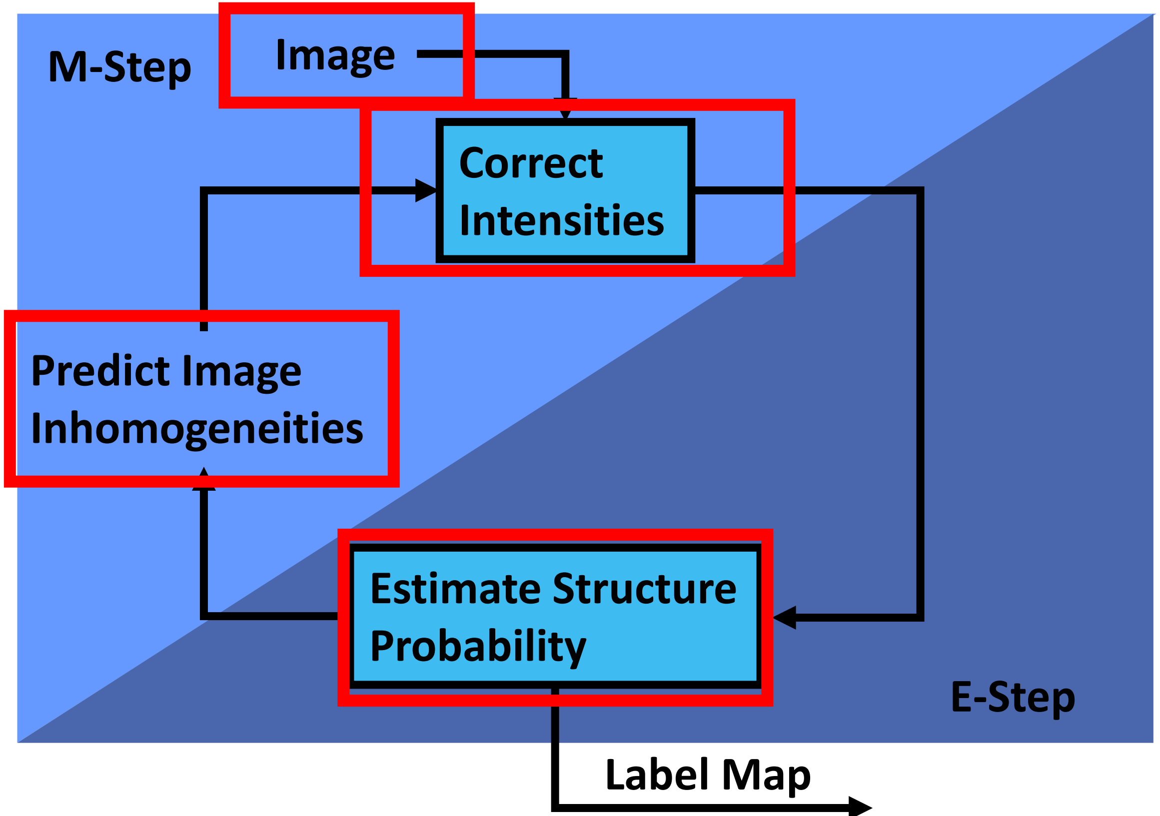
where $\bar{R}_x = \sum_z \mathcal{W}(z)\sigma_z^{-1}(x - \mu_z)$ is residual

and H is a low pass filter

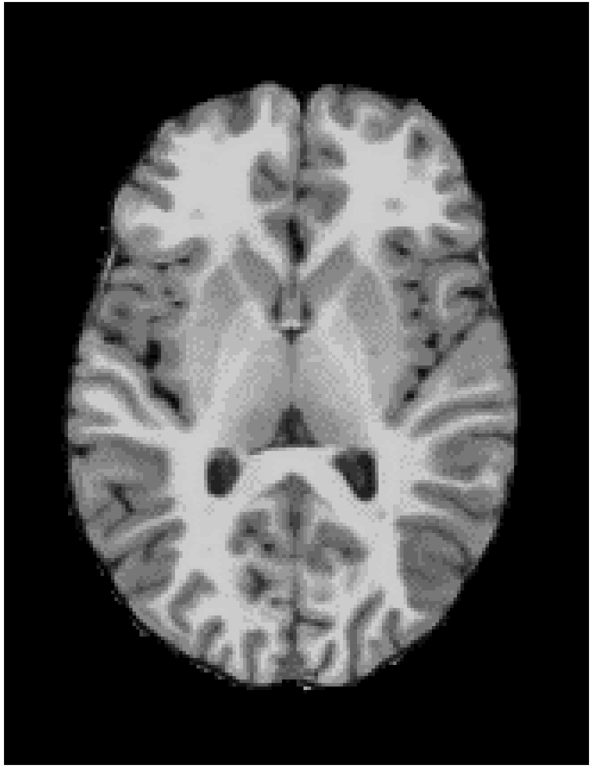
* Wells et al: IEEE Transactions of Medical Imaging, 1996

Tutorial: https://www.lri.fr/~sebag/COURS/EM_algorithm.pdf

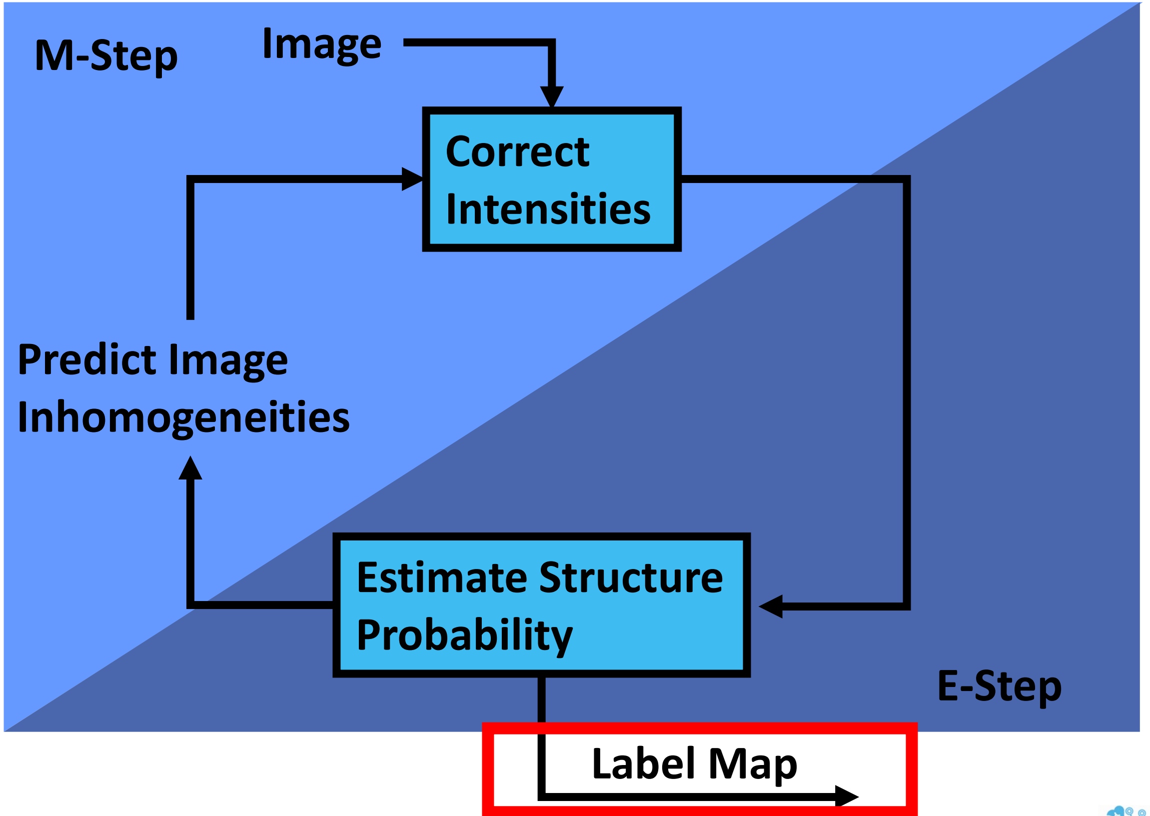
Expectation Maximization



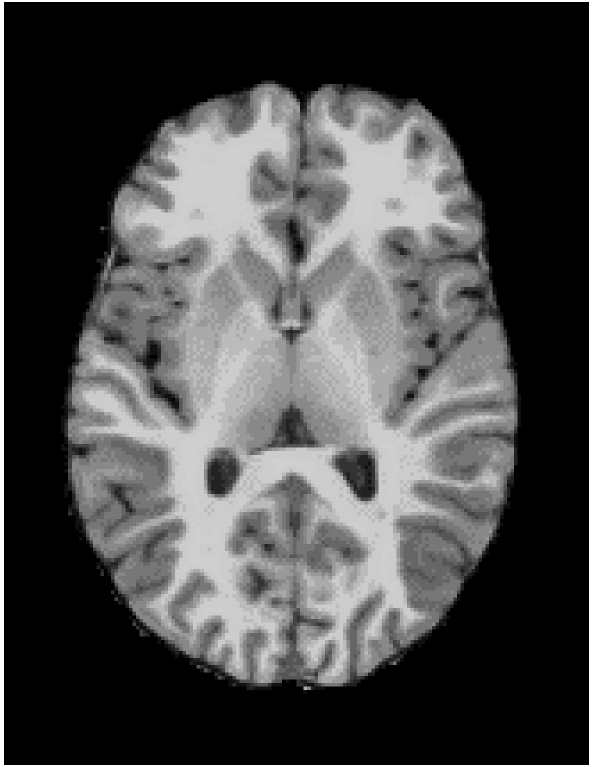
Output of EM

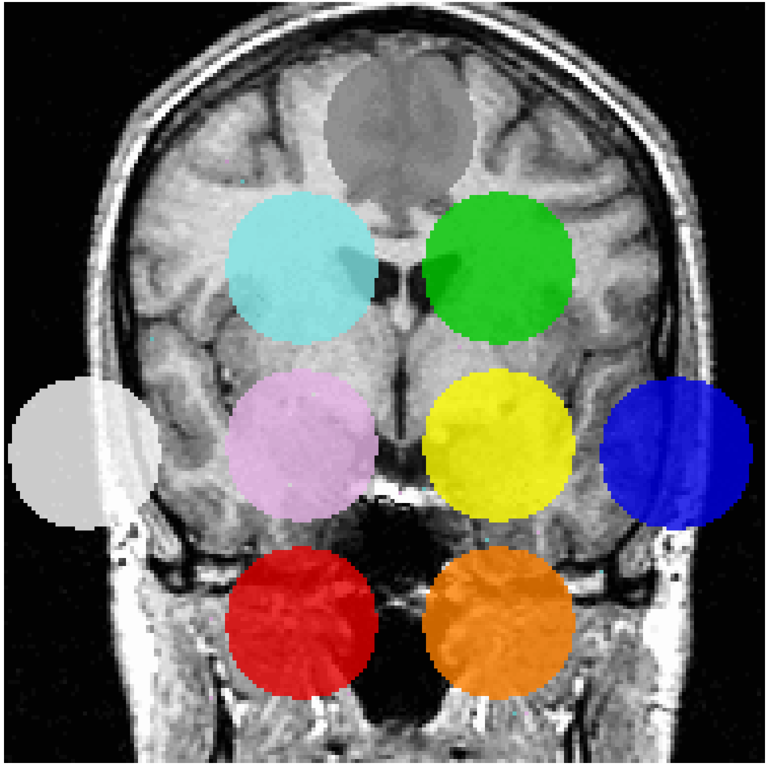


Expectation Maximization



Output of EM

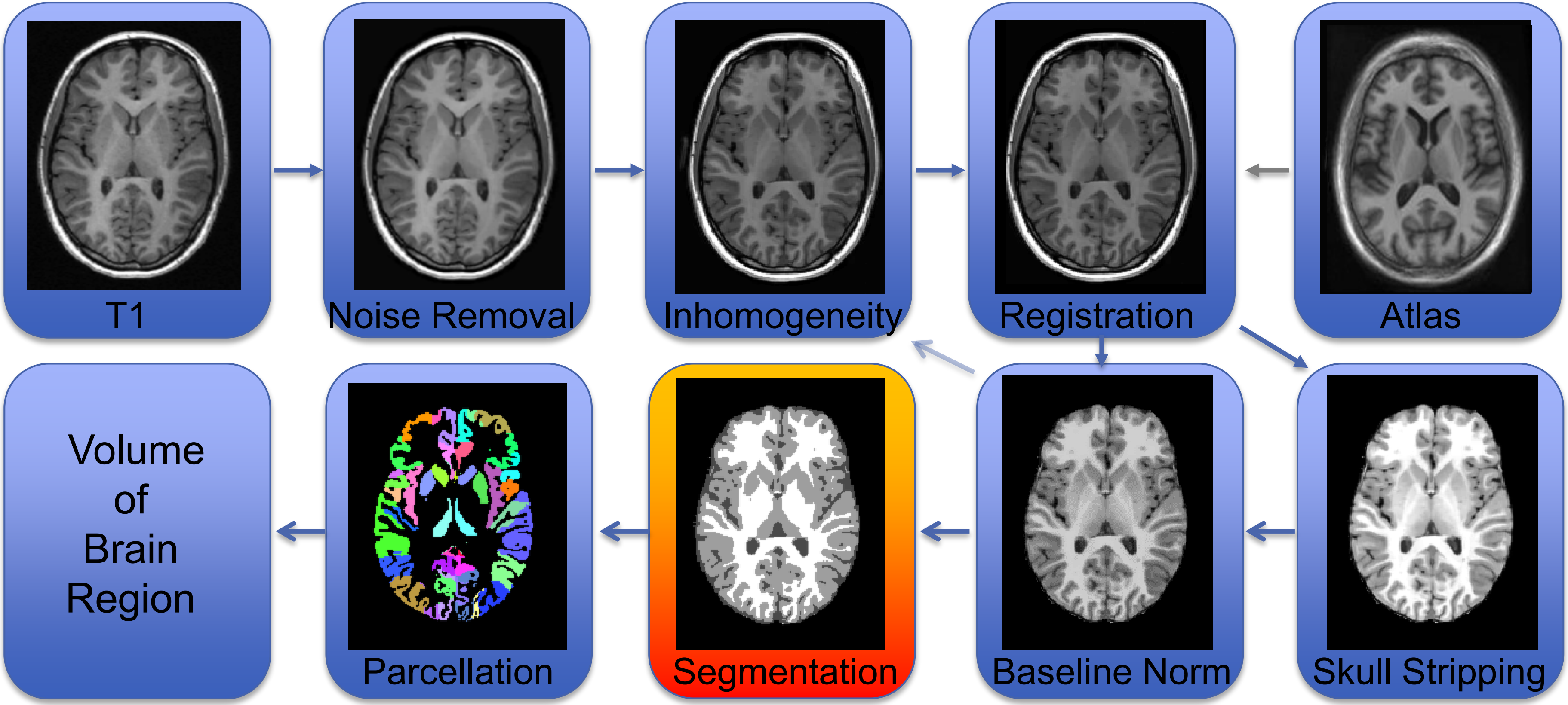




Today…

- Principle Component Analysis (PCA) for Noise Removal
- Expectation Maximization (EM) for Image Inhomogeneity Correction
- U-Net for Segmentation

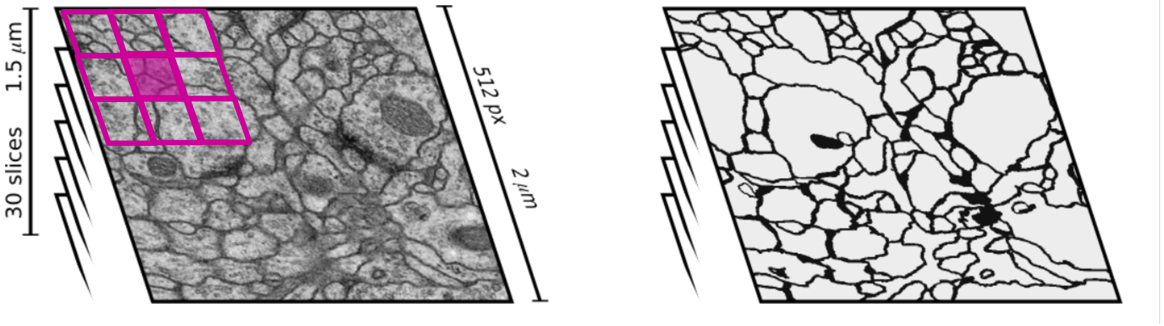
Processing of Structural MRI

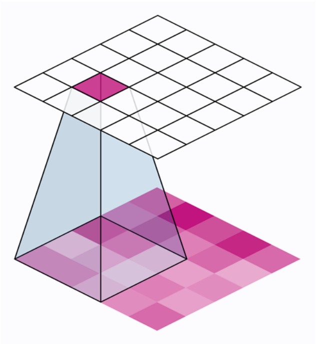


Sliding Window

First influential deep learning approach for the segmentation of medical images

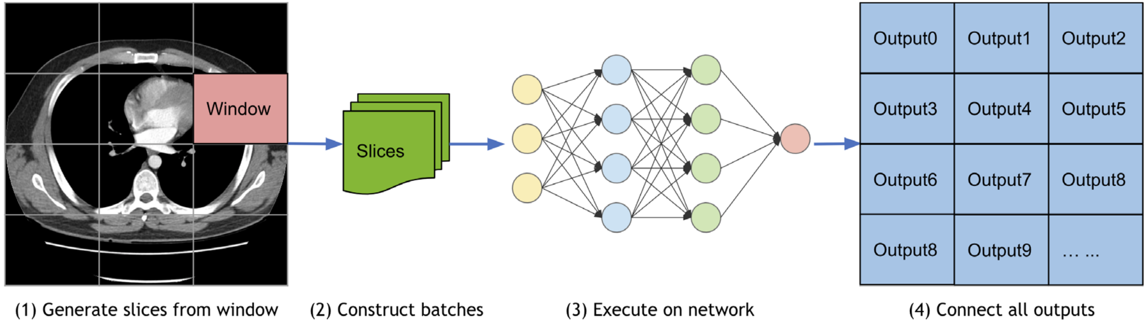
Ciresan et al. NIPS 2012





https://www.kaggle.com/code/ryanholbrook/the-sliding-window

Sliding Window



https://docs.monai.io/en/0.5.1/highlights.html#sliding-window-inference

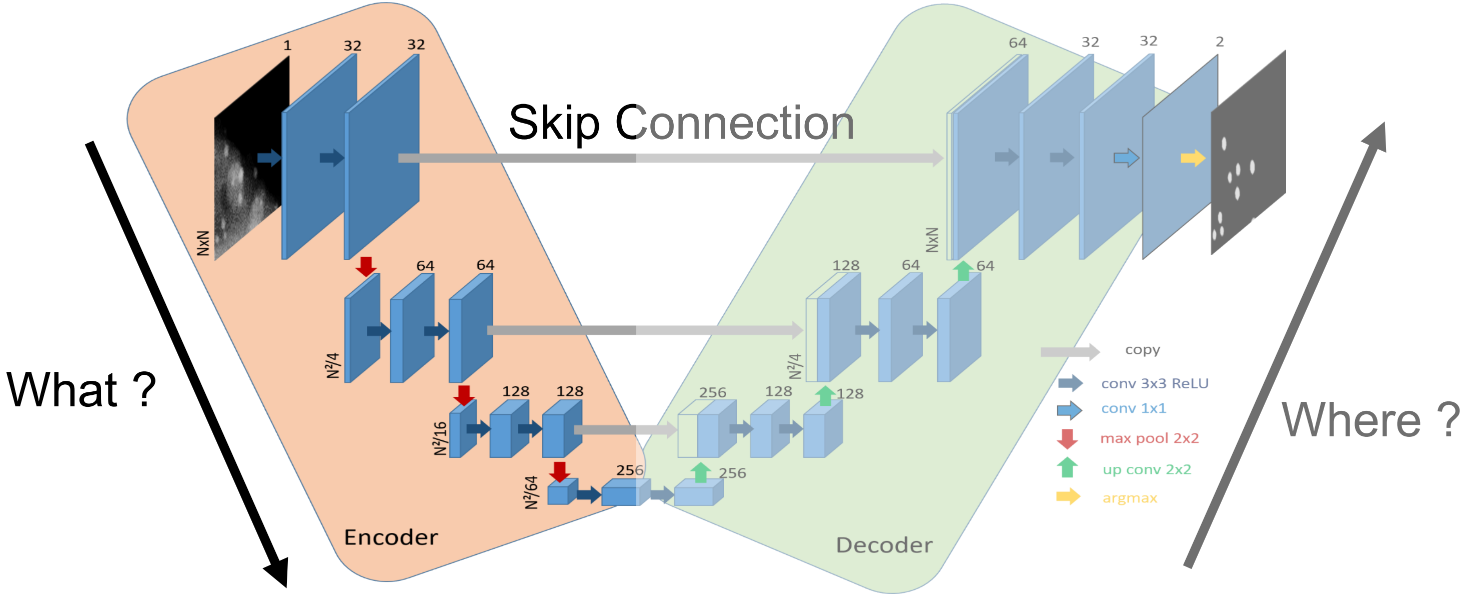
Cons:

- quite slow because the network must be run separately for each patch
- trade-off between localization accuracy and the use of context:
    - larger patches require more max-pooling layers reducing the localization accuracy
    - small patches allow the network to see only little context.

U-Net

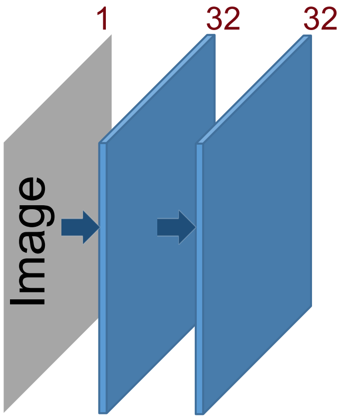
State-Of-The-Art: Ronneberger et al. MICCAI 2015

https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net



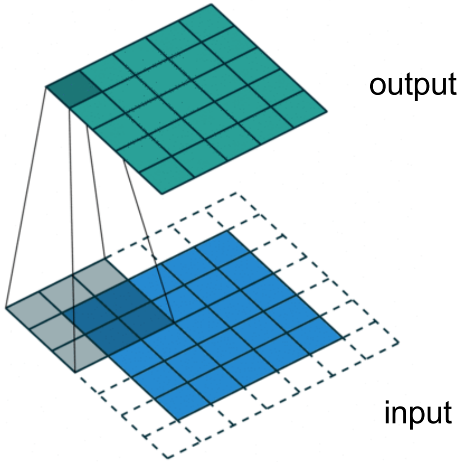
https://www.creatis.insa-lyon.fr/~grenier/wp-content/uploads/teaching/UNet/UNet_InBrief.pdf

Encoder – Conv Net



conv 3x3 ReLU

2D convolution, kernel size of 3, stride of 1 and padding

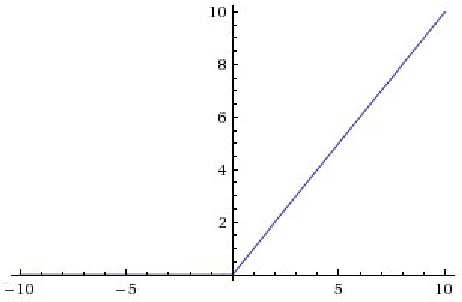


https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d

N = Number of Features

Example of image kernels

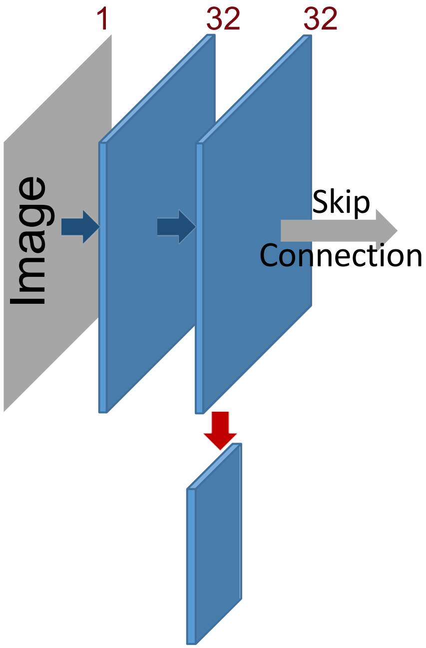
Rectified Linear Units (ReLU)



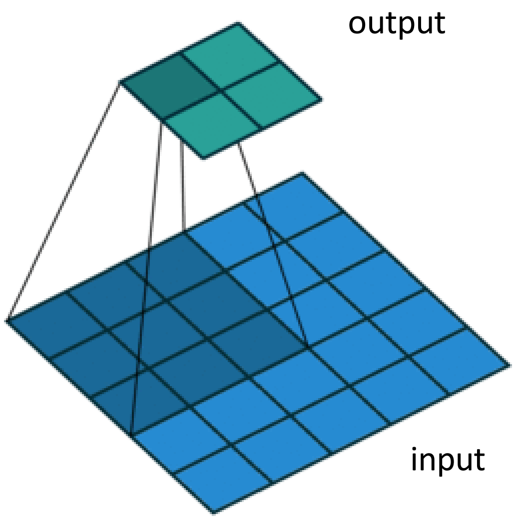
$$f(x) = max(0, x).$$

Largely avoiding vanishing gradient problem (unlike tanh)

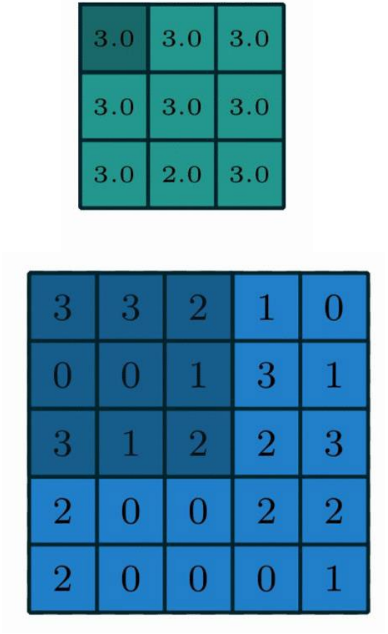
Encoder – Down Sampling



kernel size of 3, stride of 2, no padding



Max pooling, 3x3 kernel, stride of 1, no padding



conv 3x3 ReLU

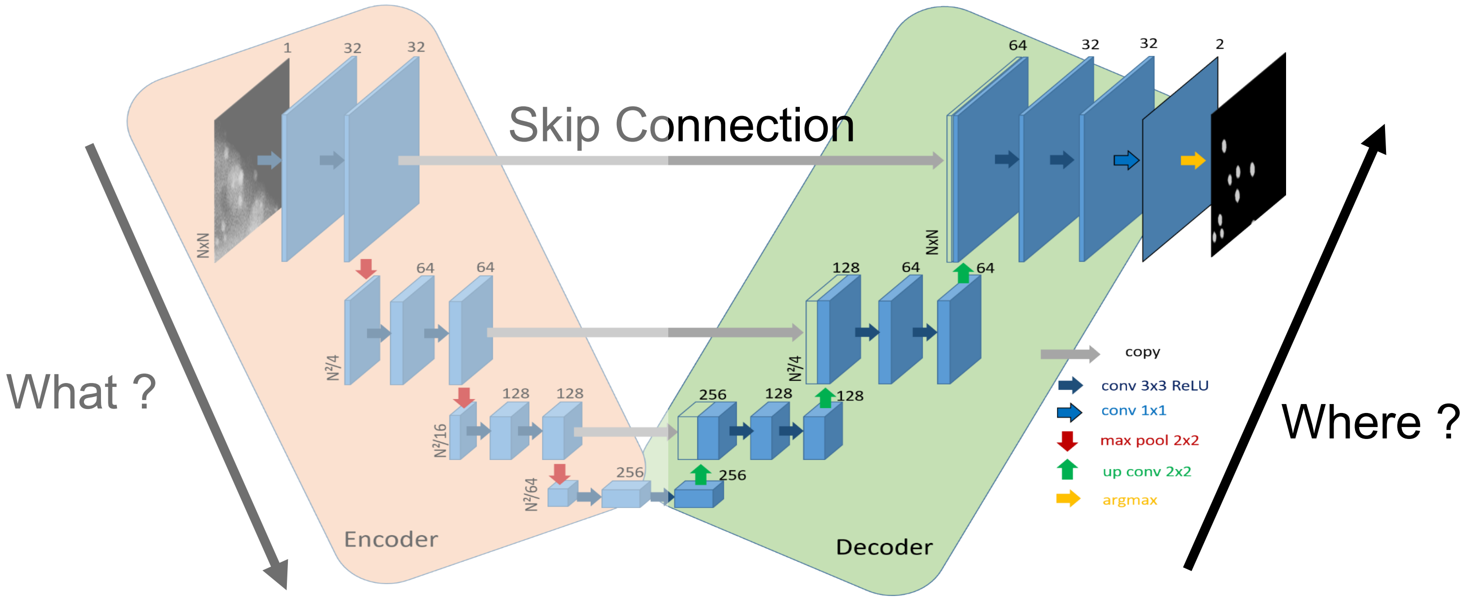
copy

max pool 2x2

U Net

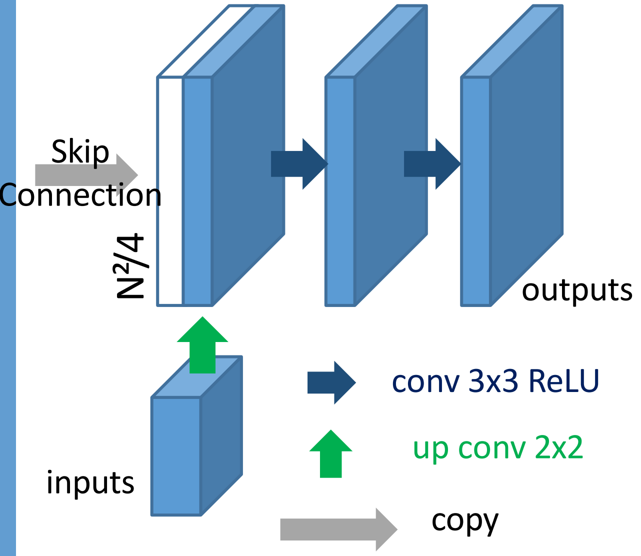
Current state of the art - Ronneberger et al. MICCAI 2015

https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net

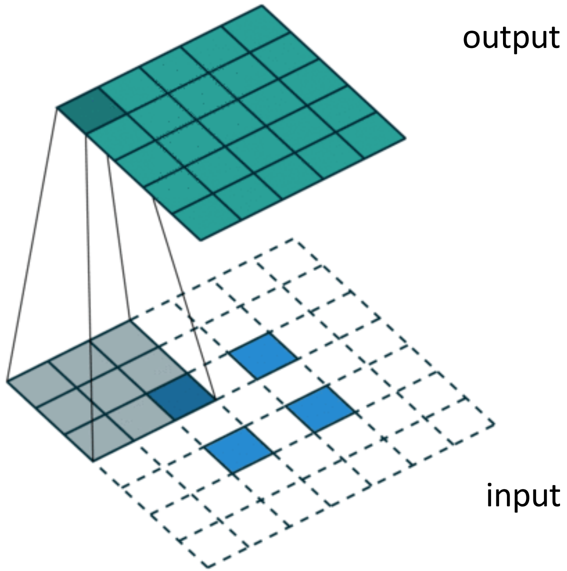


https://www.creatis.insa-lyon.fr/~grenier/wp-content/uploads/teaching/UNet/UNet_InBrief.pdf

Upsampling

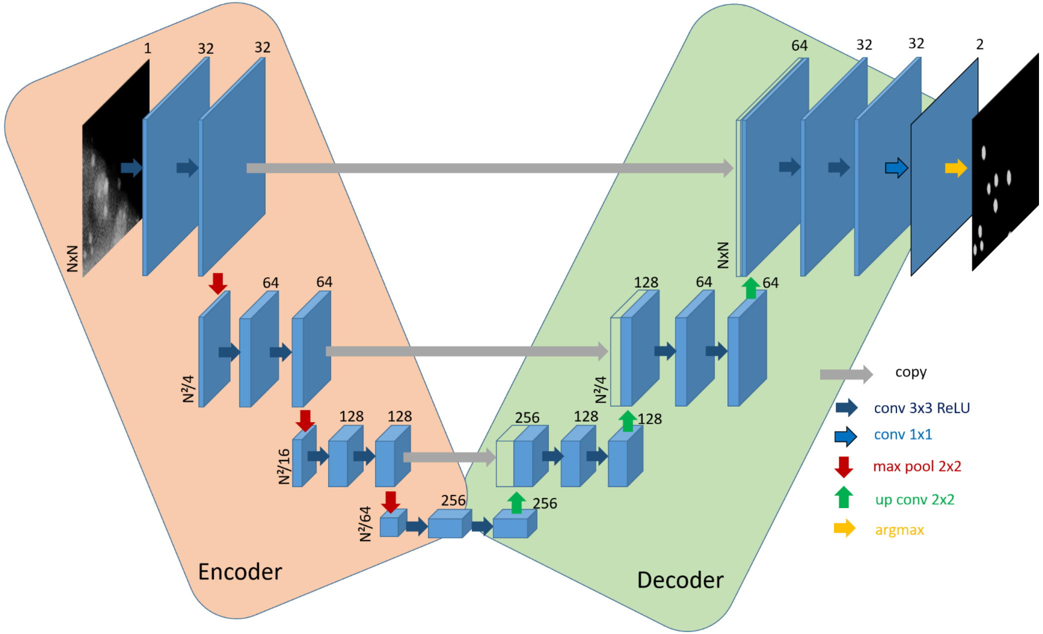


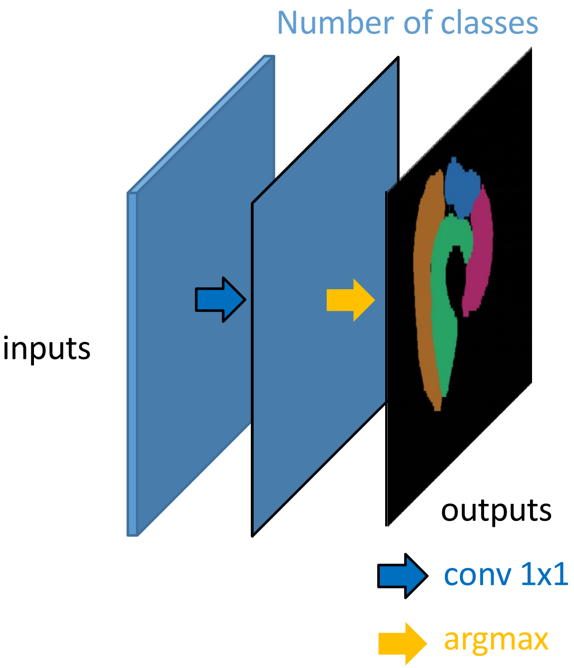
Transposed 2D convolution with padding, stride of 1 and kernel of 3



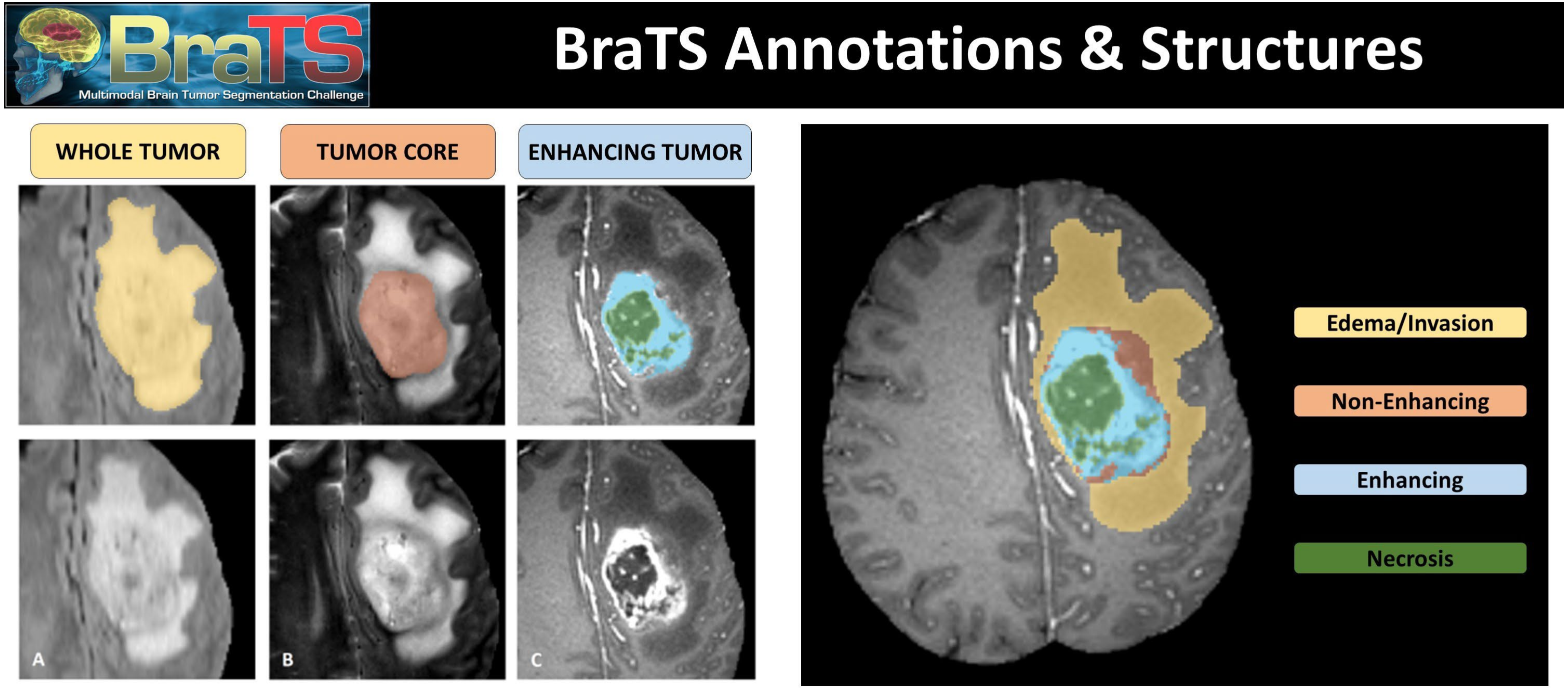
https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d

Output


1
32
32
NxN
64
64
N²/4
128
128
N²/16
256
N²/64
256
256
128
128
128
64
64
N²/4
64
32
32
2
NxN
Encoder
Decoder
copy
conv 3x3 ReLU
conv 1x1
max pool 2x2
up conv 2x2
argmax


Number of classes
inputs
outputs
conv 1x1
argmax

U-Net not only changed …

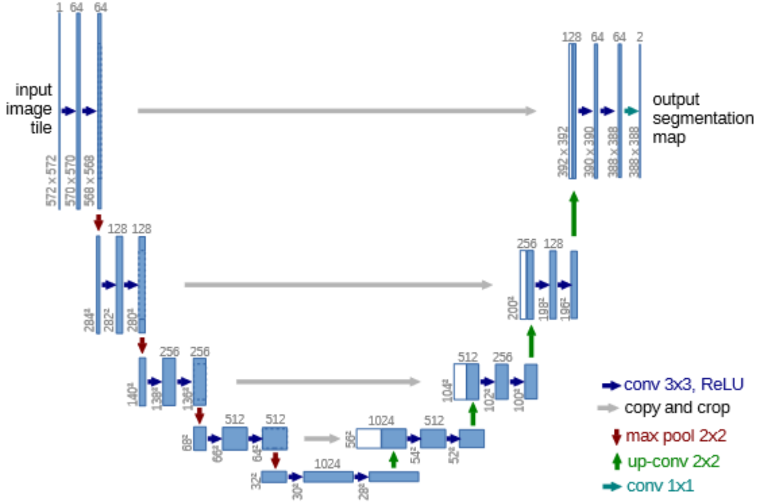


… but also other multi-modal segmentation tasks, such as satellite imaging and self-driving cars.

Things to be aware of

- Many design choices:
  Justify design via ablation study, i.e., omit part of the network and report on the accuracy
- Take a long time to train:
  train on large GPU cluster and freeze layers that were trained already on a different task
- Be aware of overfitting

Actual first published U-Net



Extra Class Session (4 Unit Students)

ML Q&A session with Camila Gonzalez

Thursdays 5 - 6:20 pm

Thank you!

- https://ml4n.Stanford.edu/
- psyc221-aut2324-staff@staff.stanford.edu