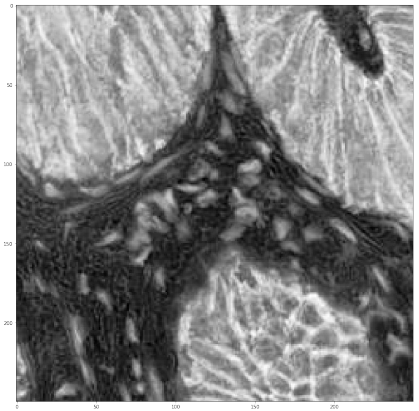
Learning to denoise without clean data

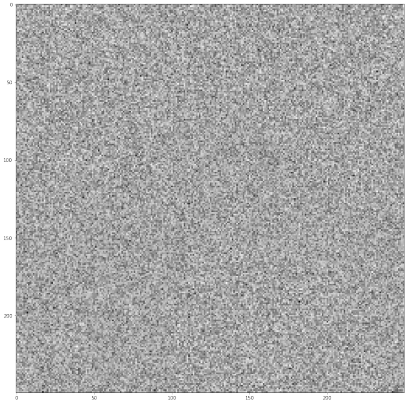
Joshua Batson

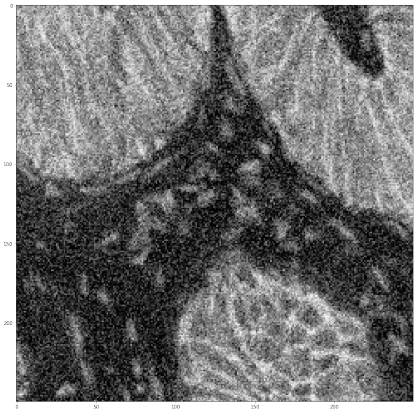
hep-ai seminar 10/18/18

Noisy data is clean data + noise

$$y = x + n \qquad n \sim \mathcal{N}(0, \sigma^2)$$



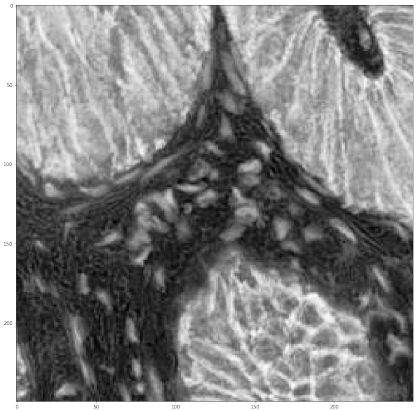


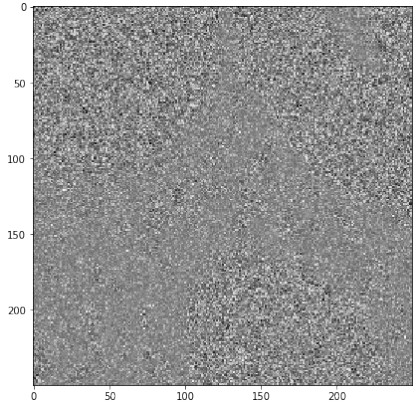


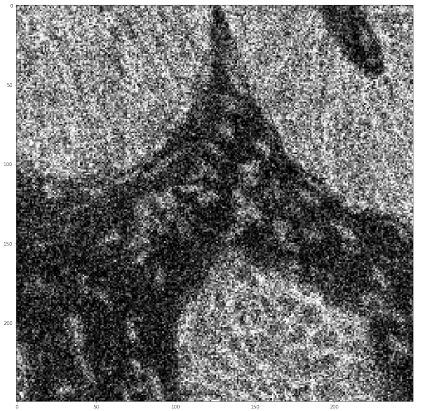
Noisy data is clean data + noise

$$y \sim Poisson(Nx)/N$$

$$N = \text{mean photons at peak intensity}$$







$$y = x + n$$

We want to predict

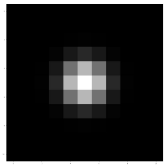
$$y \mapsto x$$

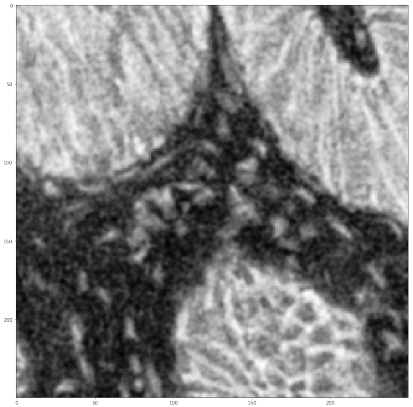
You need a prior

Prior: nearby pixels are similar

Denoising strategy: local averaging

$$f(y) = G * y$$



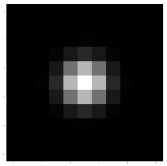


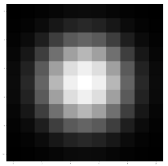
You need a prior

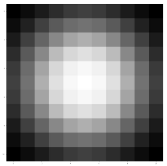
Prior: nearby pixels are similar

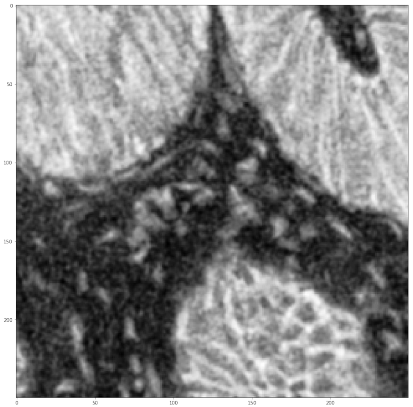
Denoising strategy: local averaging

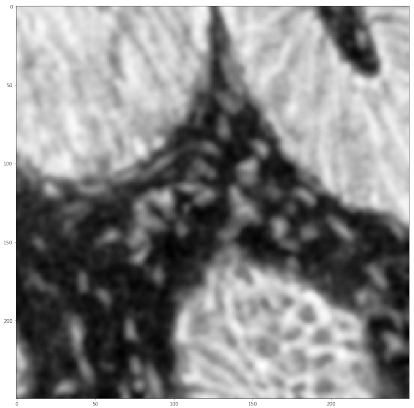
$$f(y) = G * y$$

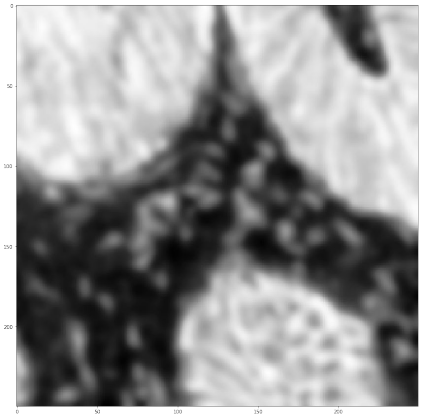










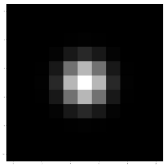


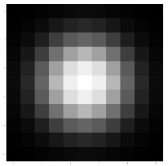
You need a prior

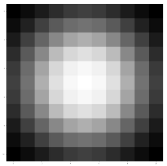
Prior: nearby pixels are similar

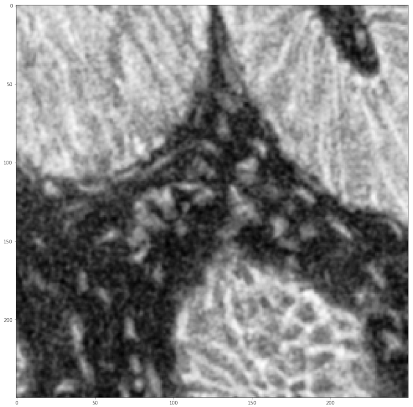
Denoising strategy: local averaging

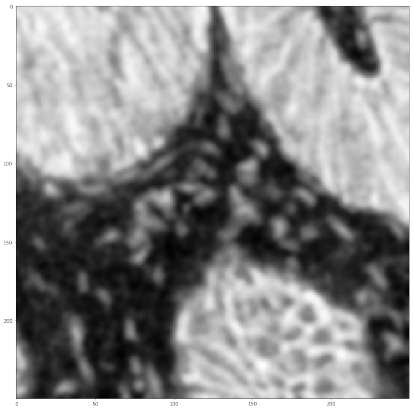
$$f(y) = G * y$$

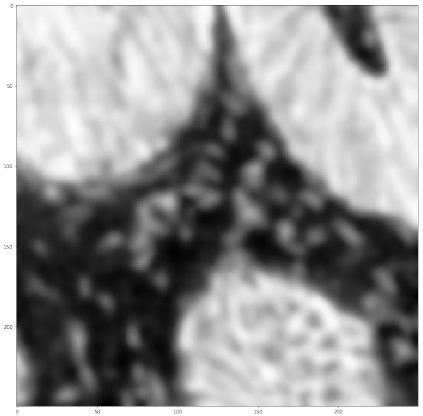








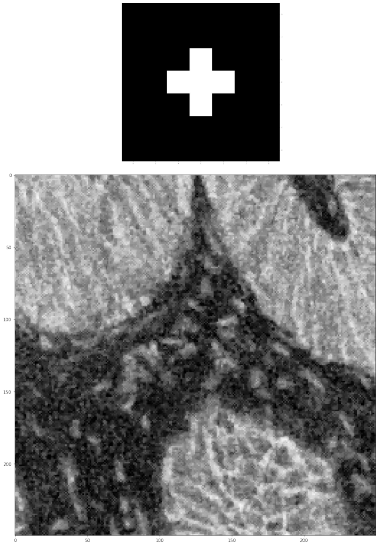


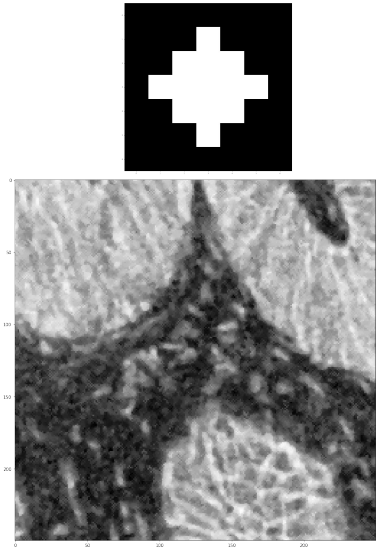


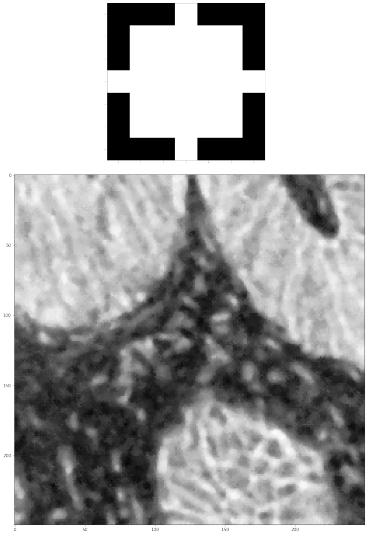
You need a prior

Prior: nearby pixels are similar, edges exist

Denoising strategy: local medians



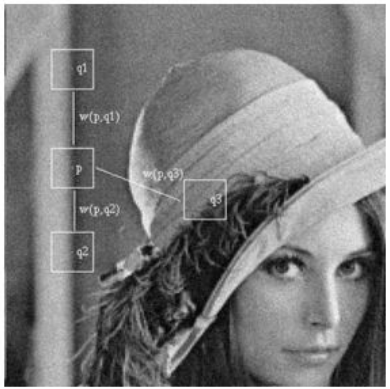




You need a prior

Prior: nearby patches may be similar, corners exist

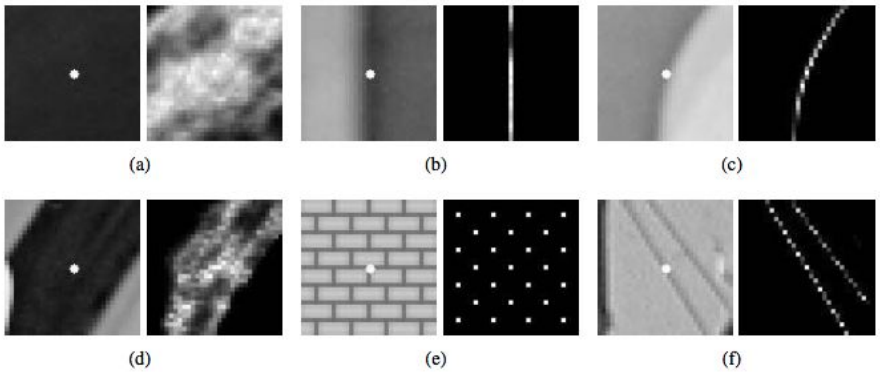
Denoising strategy: NL-means



You need a prior

Prior: nearby patches may be similar, corners exist

Denoising strategy: NL-means



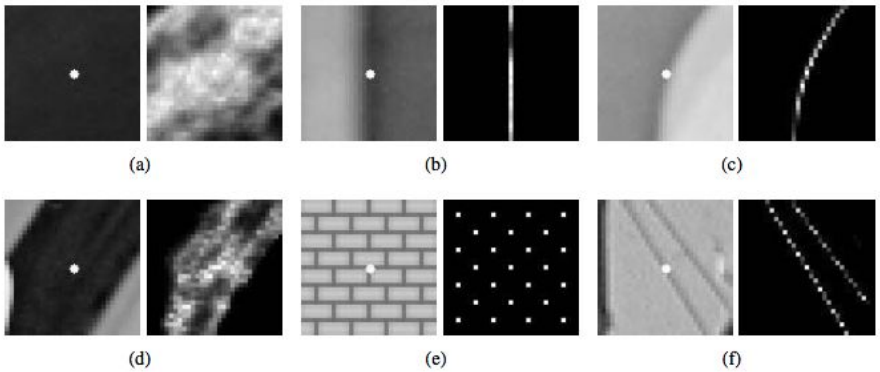
$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||v(\mathcal{N}_i) - v(\mathcal{N}_j)||^2_{2,a}}{h^2}},$$

$$NL[v](i) = \sum_{j \in I} w(i,j) v(j),$$

You need a prior

Prior: nearby patches may be similar, corners exist

Denoising strategy: NL-means



$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||v(\mathcal{N}_i) - v(\mathcal{N}_j)||^2_{2,a}}{h^2}},$$

$$NL[v](i) = \sum_{j \in I} w(i,j) v(j),$$

Aside: astronauts and models

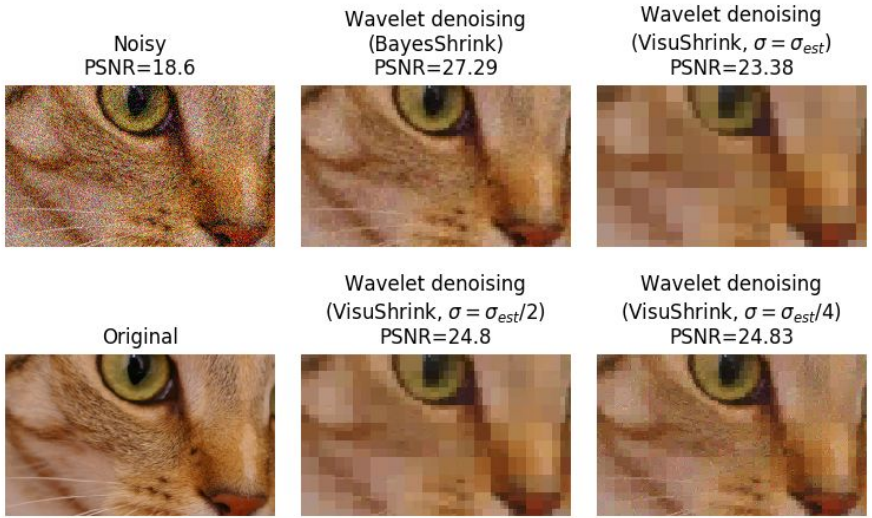
Data

Self tuned restoration

You need a prior

Prior: x is sparse in some basis (wavelet, fourier)

Denoising strategy: shrinkage in that basis



You need a prior

Prior: x is in the output of a neural net, G

Denoising strategy:

$$\min_z L(G(z), y)$$

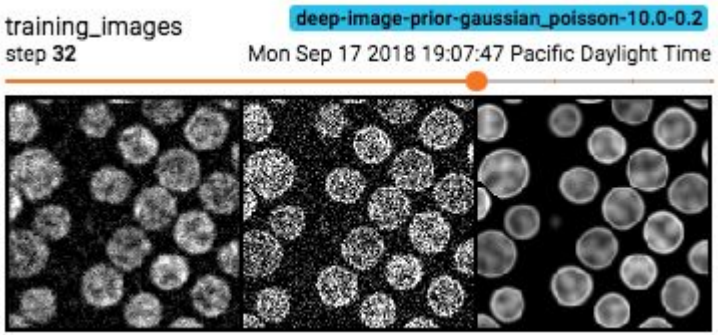
You need a prior

Prior: neural nets learn structured before noise

Denoising strategy: Deep Image Prior.

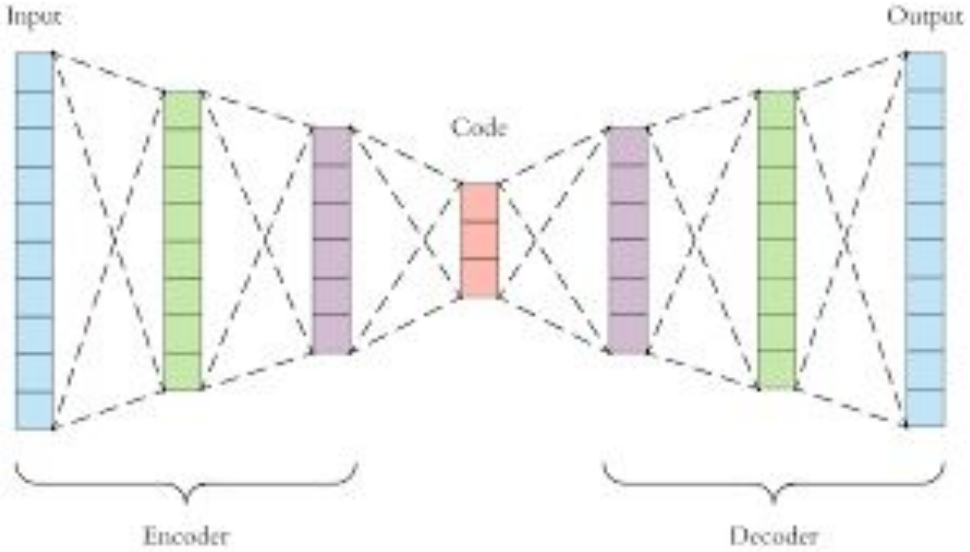
$$\hat{\theta} = \min_{\theta} ||f_{\theta}(z) - y||^2$$

$$\hat{x} = f_{\hat{\theta}}(z)$$

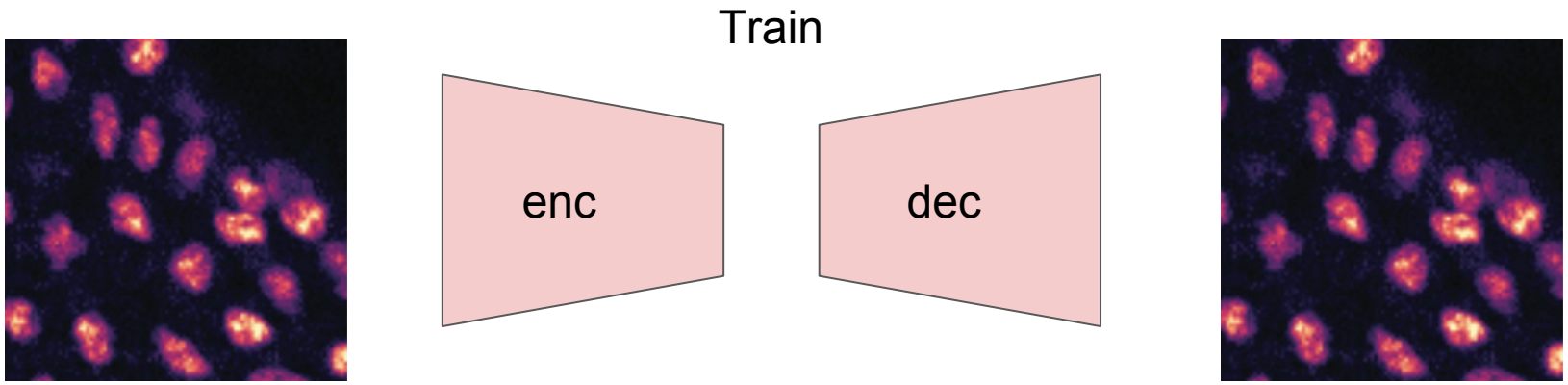


Autoencoders

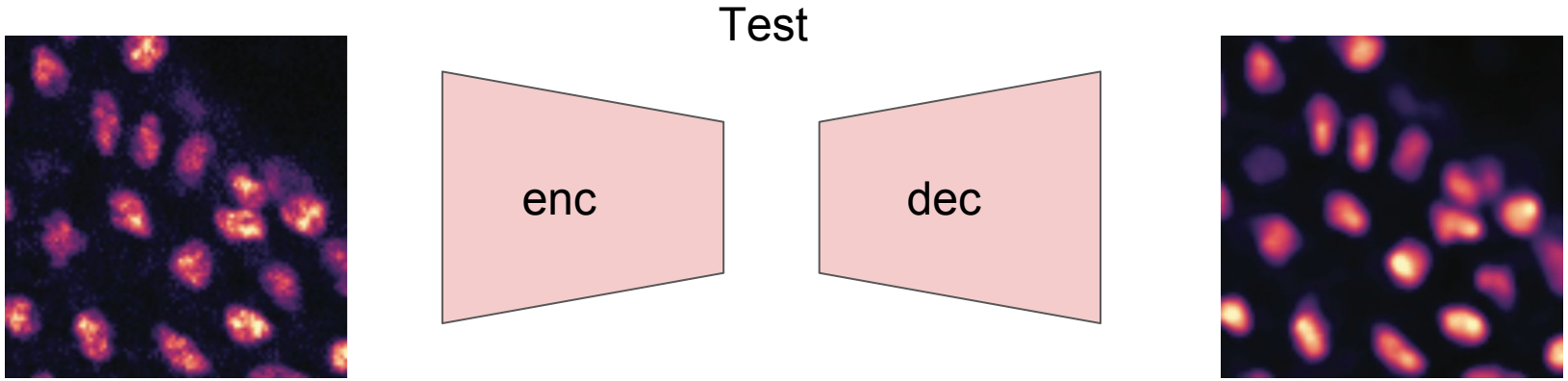
Prior: signal is the “low-complexity” part



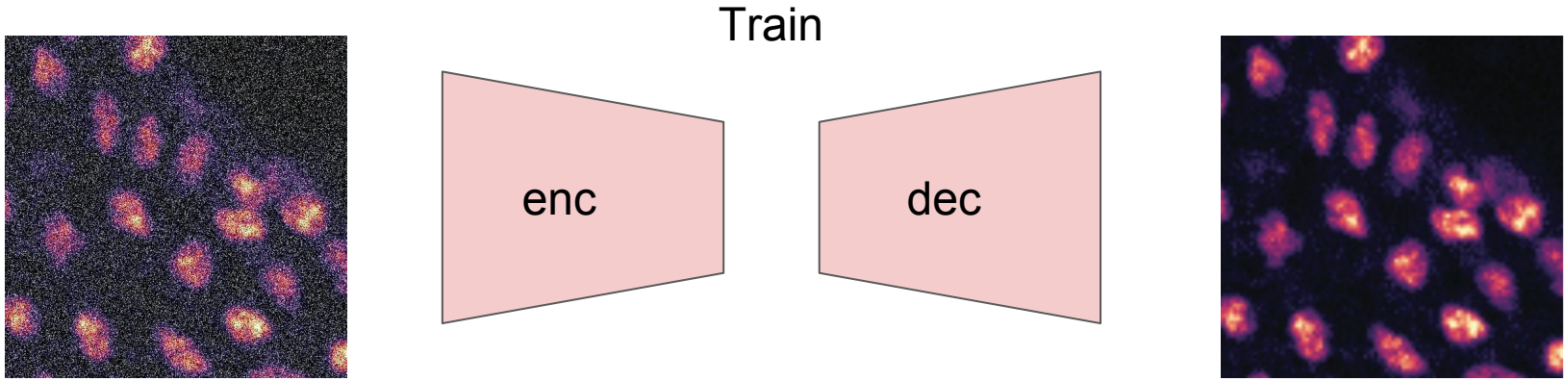
(Variational) Autoencoder



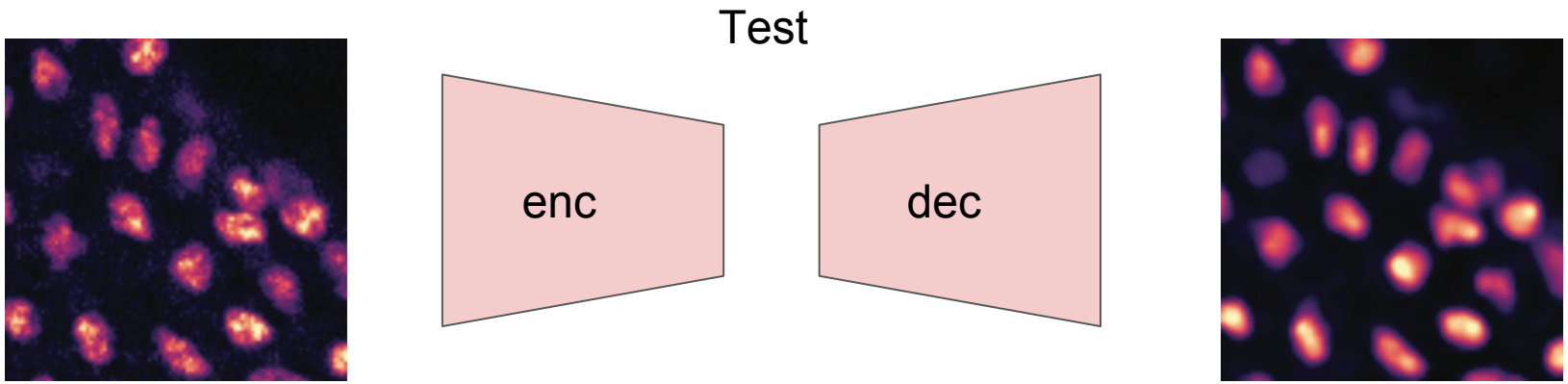
(Variational) Autoencoder



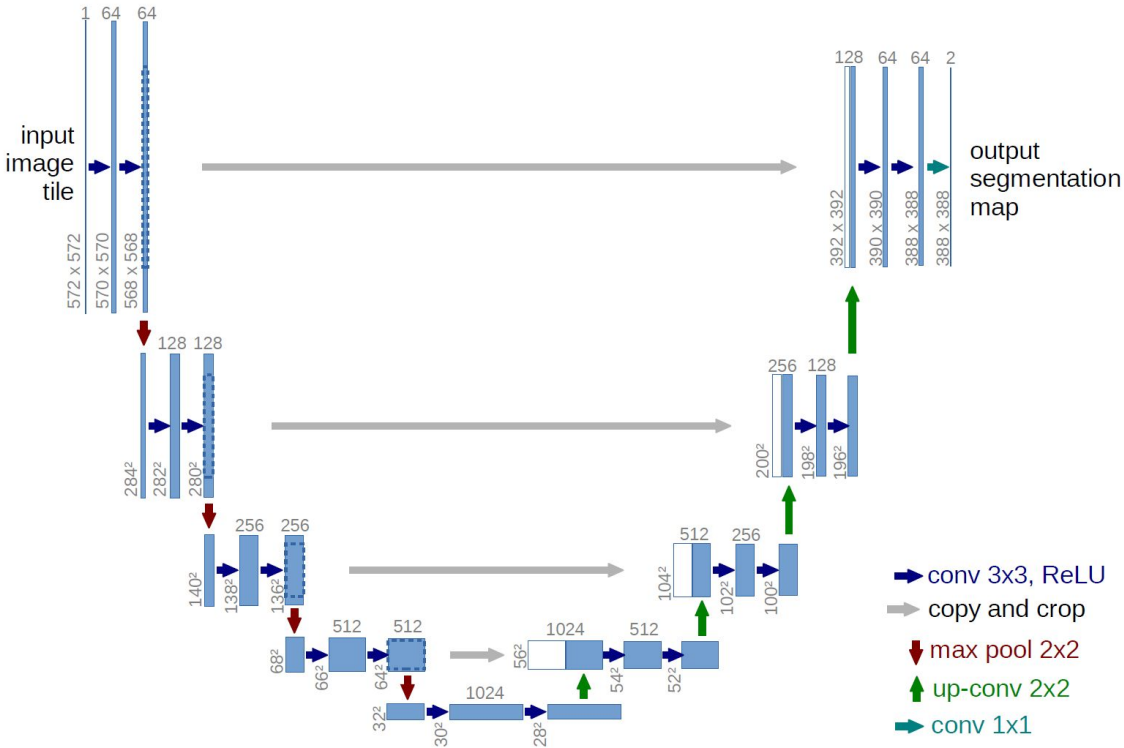
Denoising Autoencoder



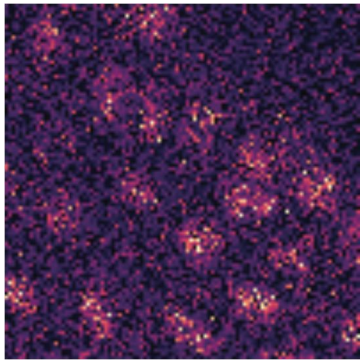
Denoising Autoencoder



UNet


input image tile
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Reconstruction from downsampling (CARE)

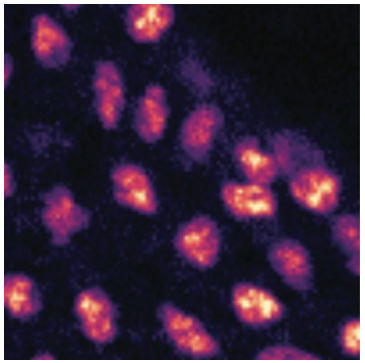


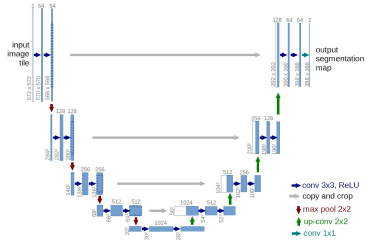
Train

enc

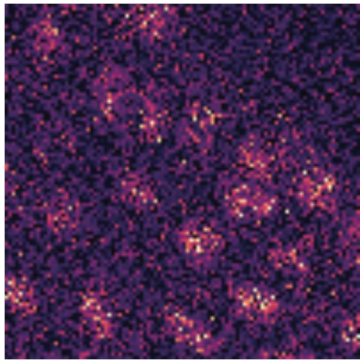
dec

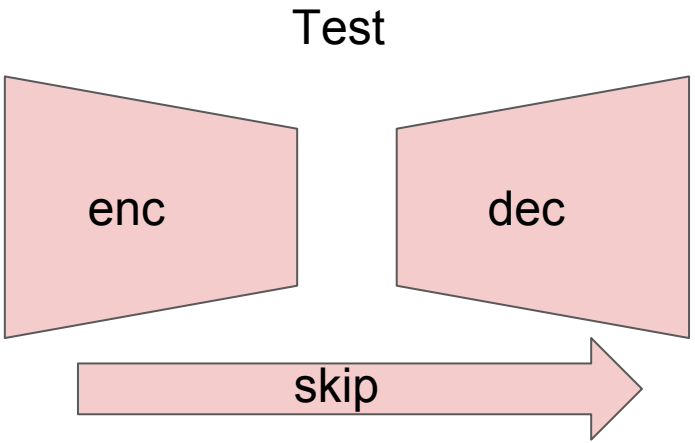
skip

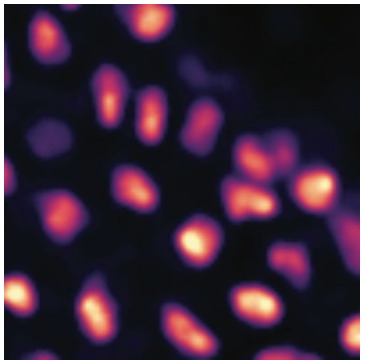




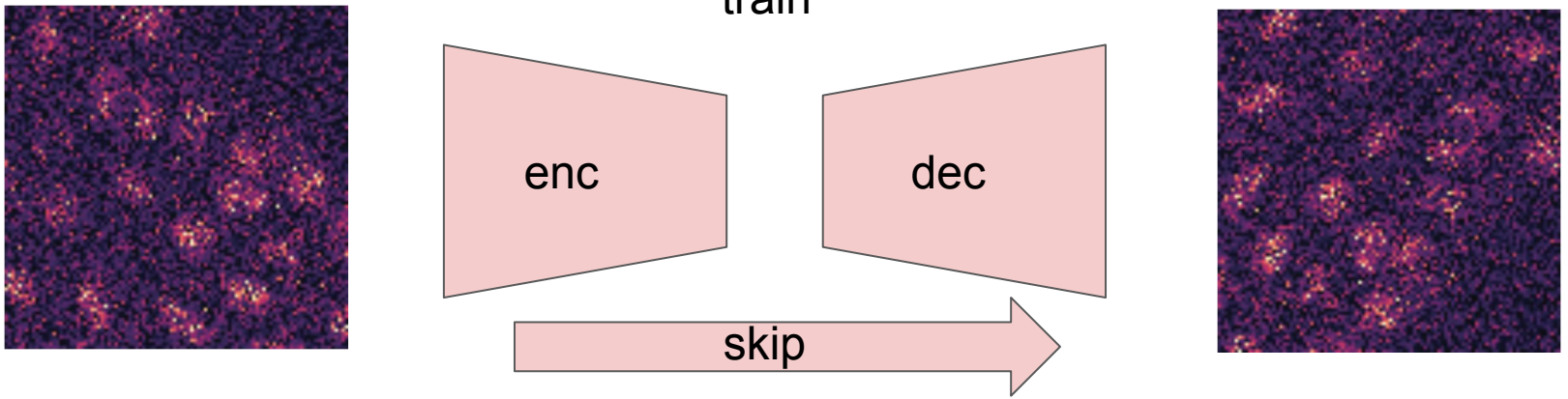
Reconstruction from downsampling (CARE)







noise2noise



Independent noise in two measurements of each sample.

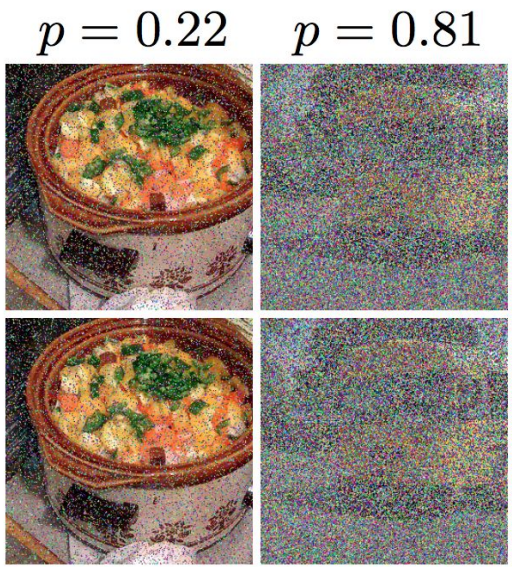
Noise2Noise: Learning Image Restoration without Clean Data

Jaakko Lehtinen [1 2] Jacob Munkberg [1] Jon Hasselgren [1] Samuli Laine [1] Tero Karras [1] Miika Aittala [3]
Timo Aila [1]

$$y = x + n$$

$$y = x + n'$$

$$\min_{\theta} \sum_{i} L(f_\theta(y), y')$$



Example training pairs