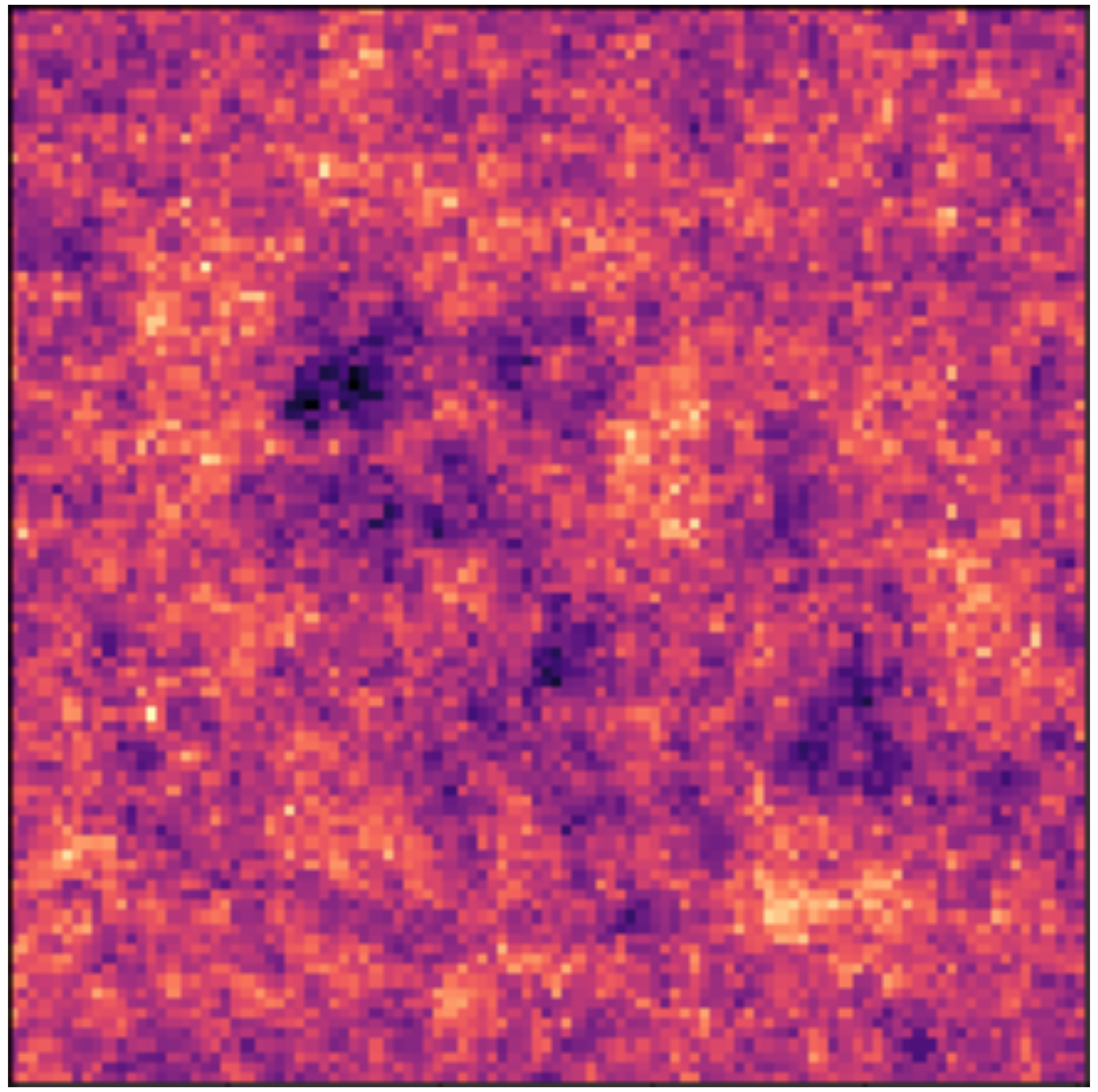
Noise2Self: Blind Denoising by Self-Supervision

Joshua Batson
Loïc Royer

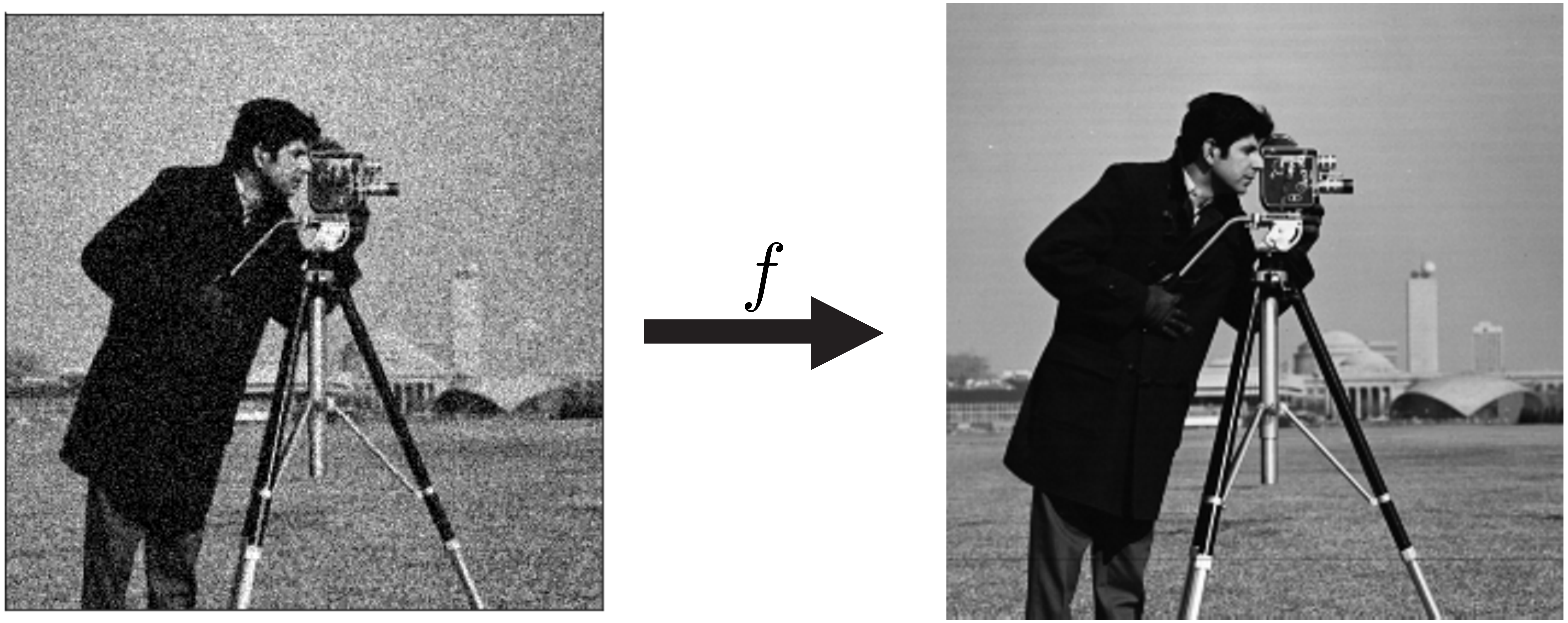
CHAN ZUCKERBERG BIOHUB

Noisy Data

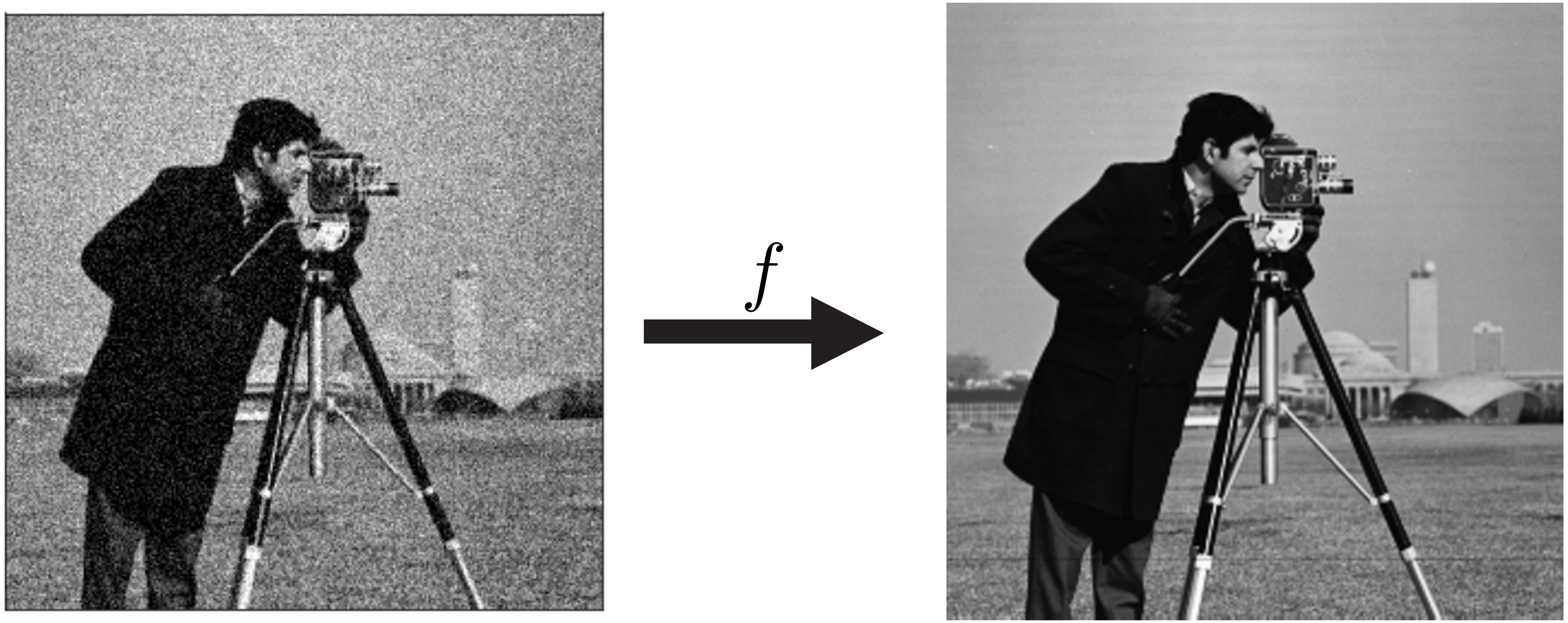


Supervision

Supervision



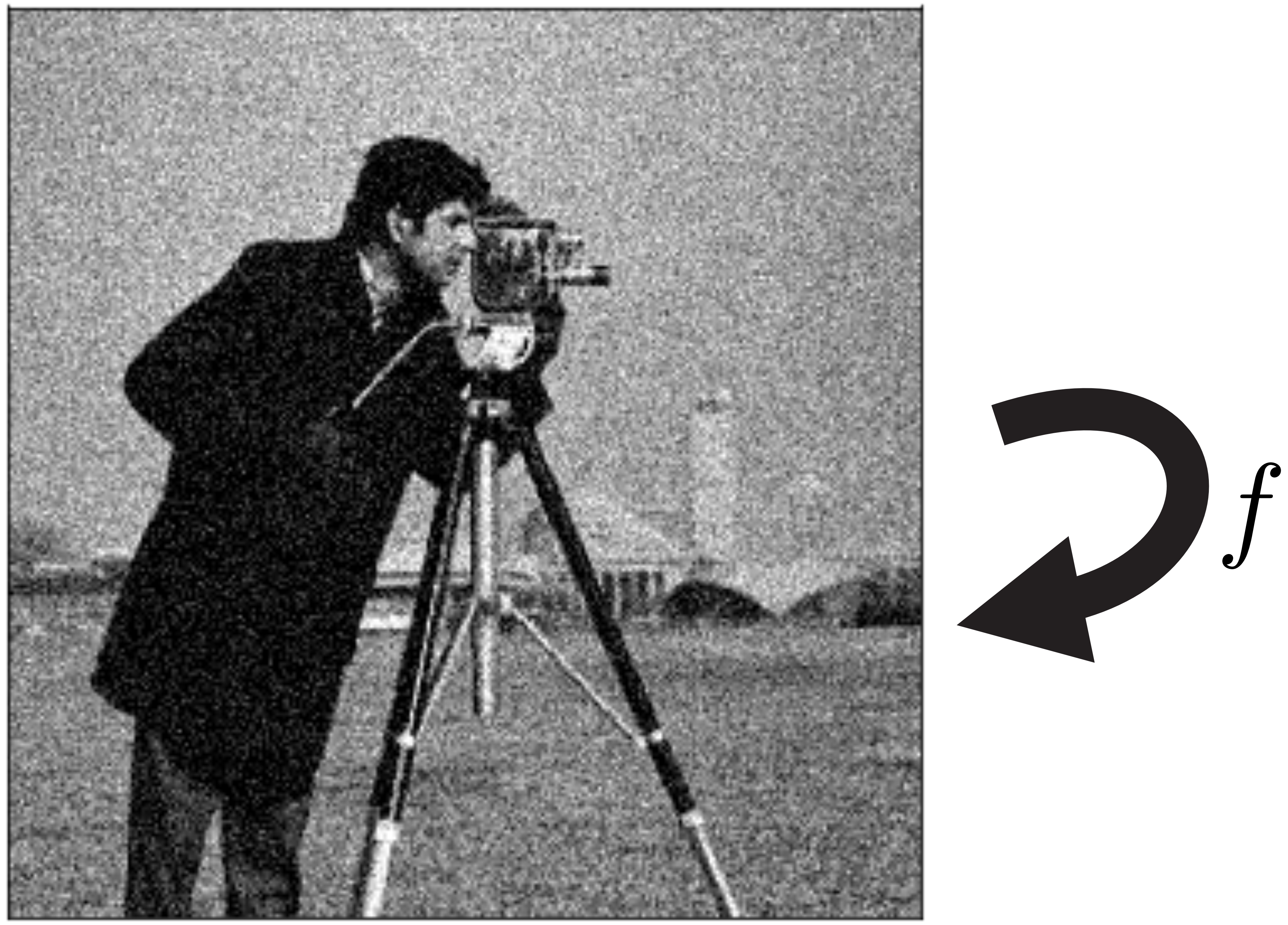
Supervision



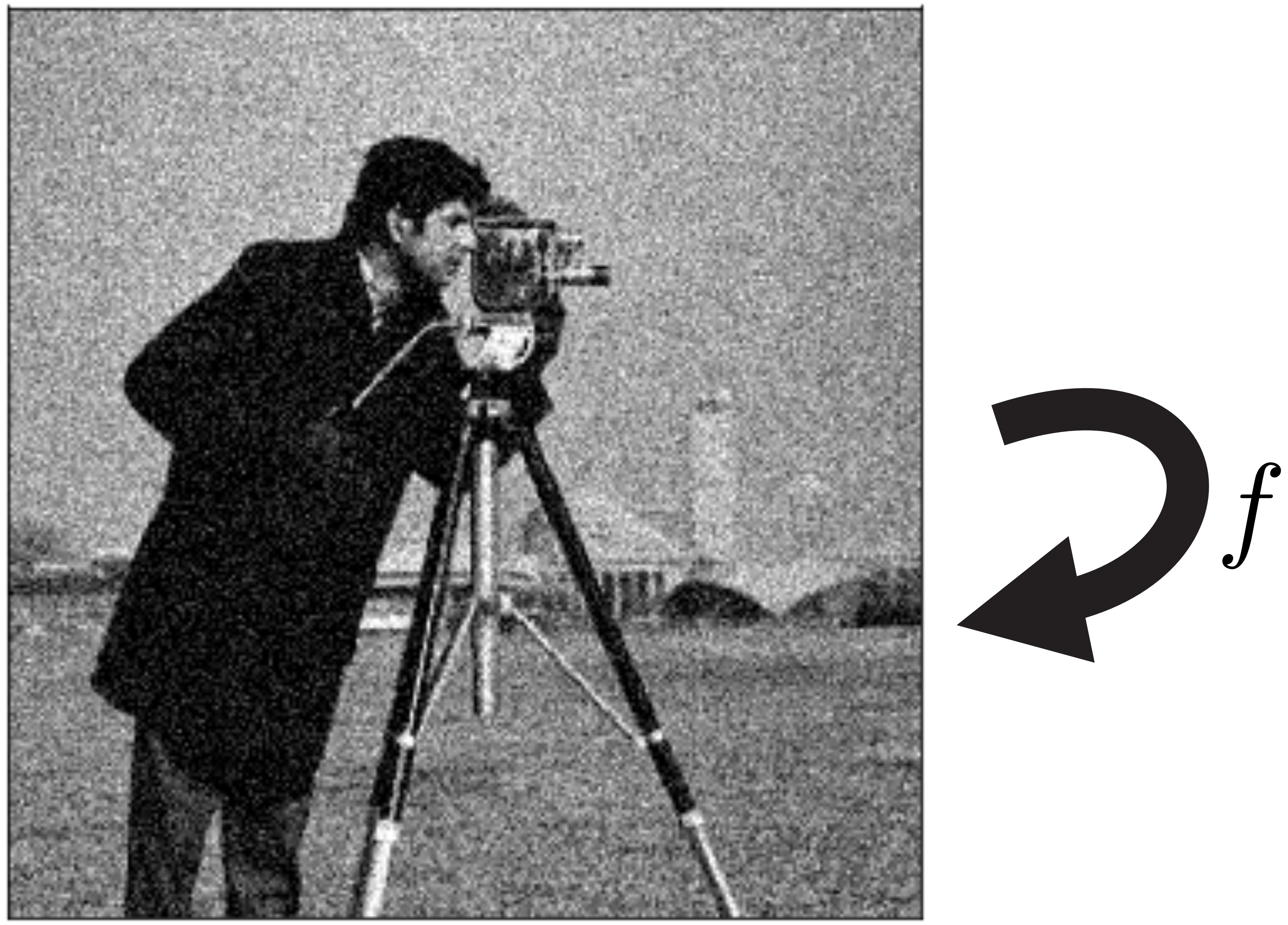
$$\|f(x) - y\|^2$$

Self-Supervision?

Self-Supervision?

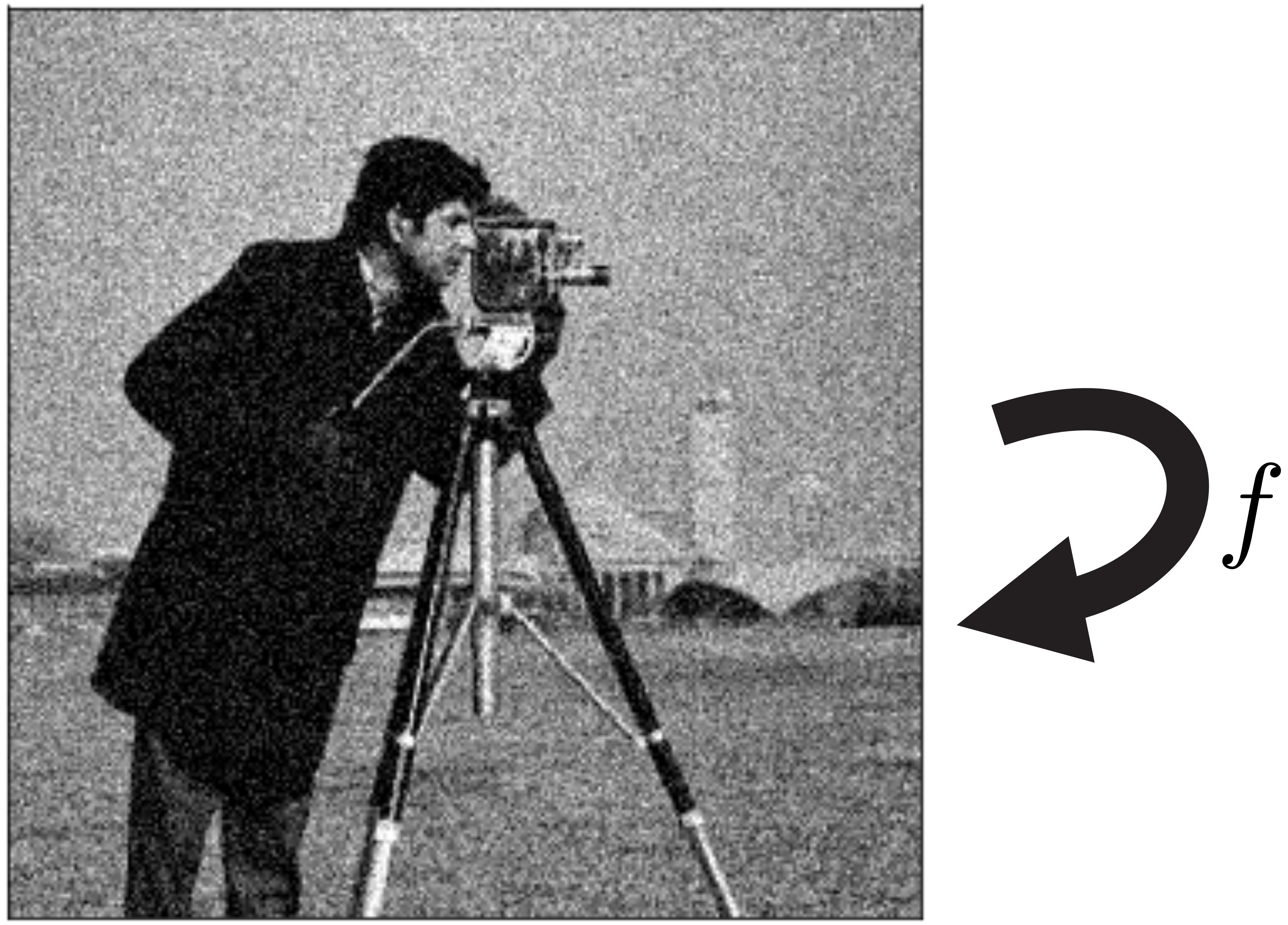


Self-Supervision?



$$\|f(x) - x\|^2$$

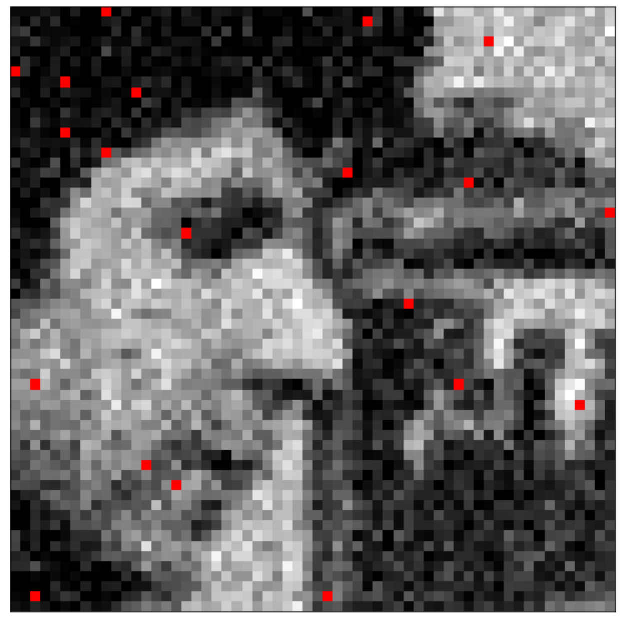
Self-Supervision?



$$\|f(x) - x\|^2$$

$$f^* = \text{Identity}$$

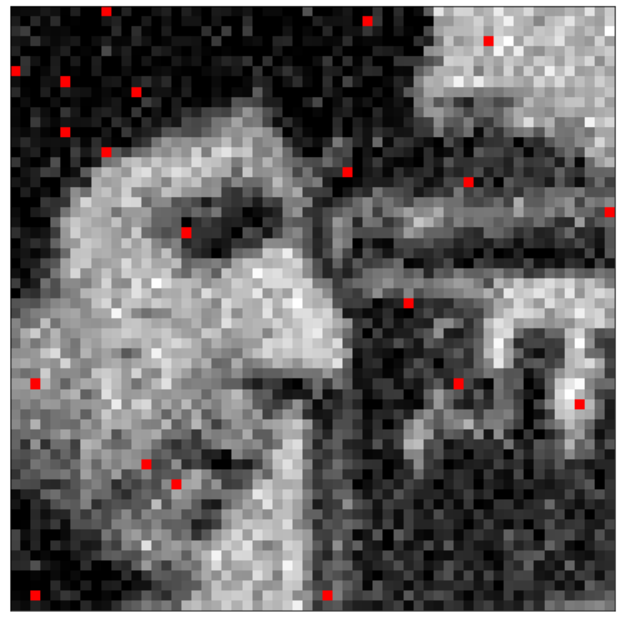
Self-Supervision?

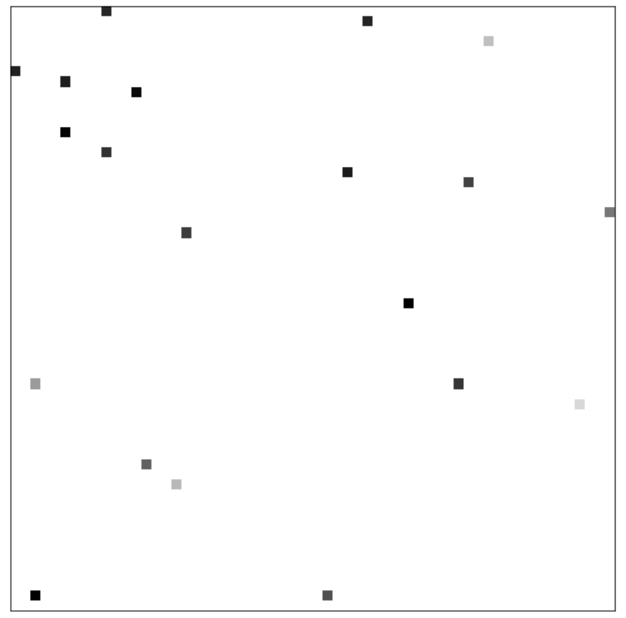


$$\|f(x) - x\|^2$$

$$f^* = \text{Identity}$$

Self-Supervision?

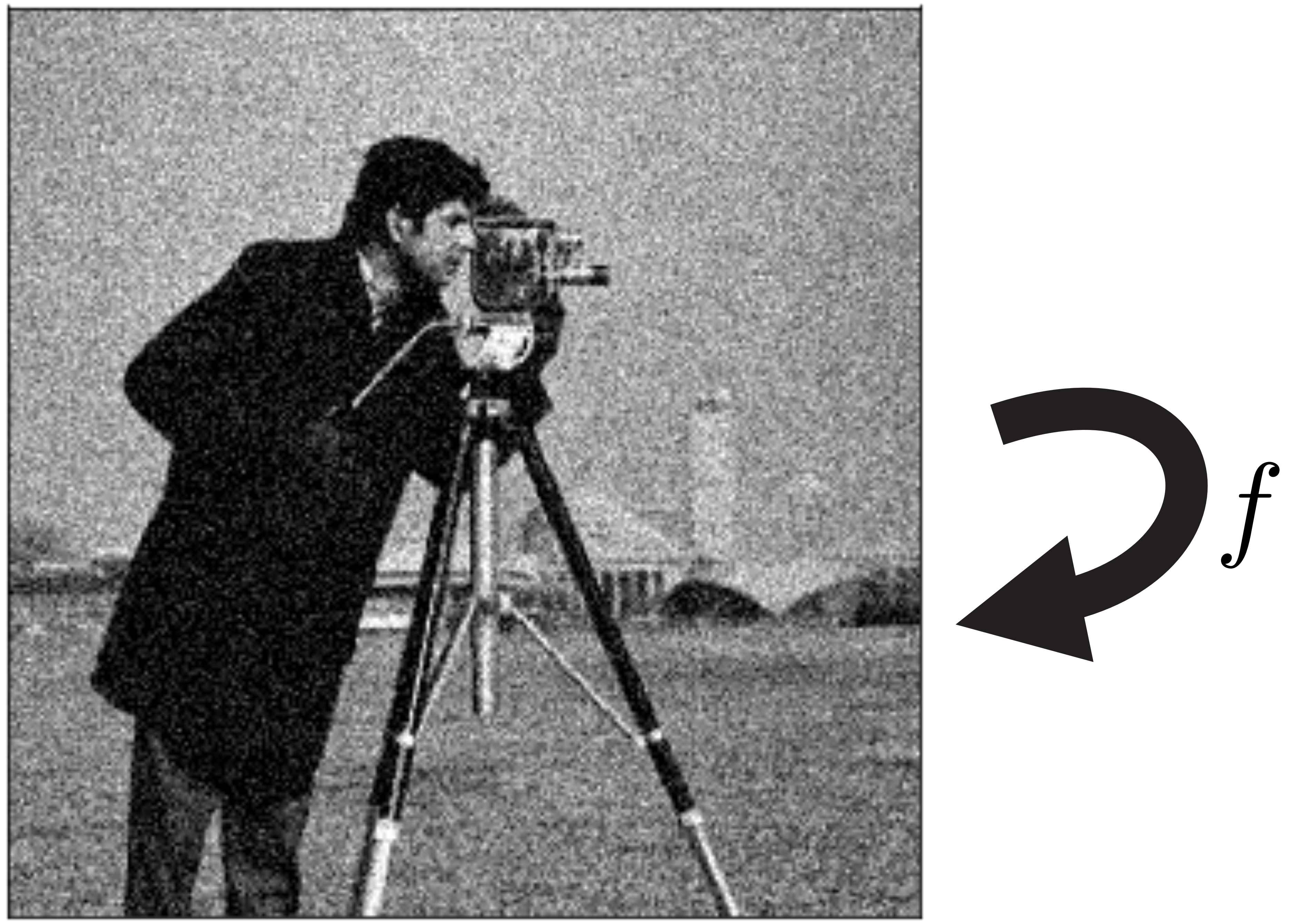




$$\|f(x) - x\|^2$$

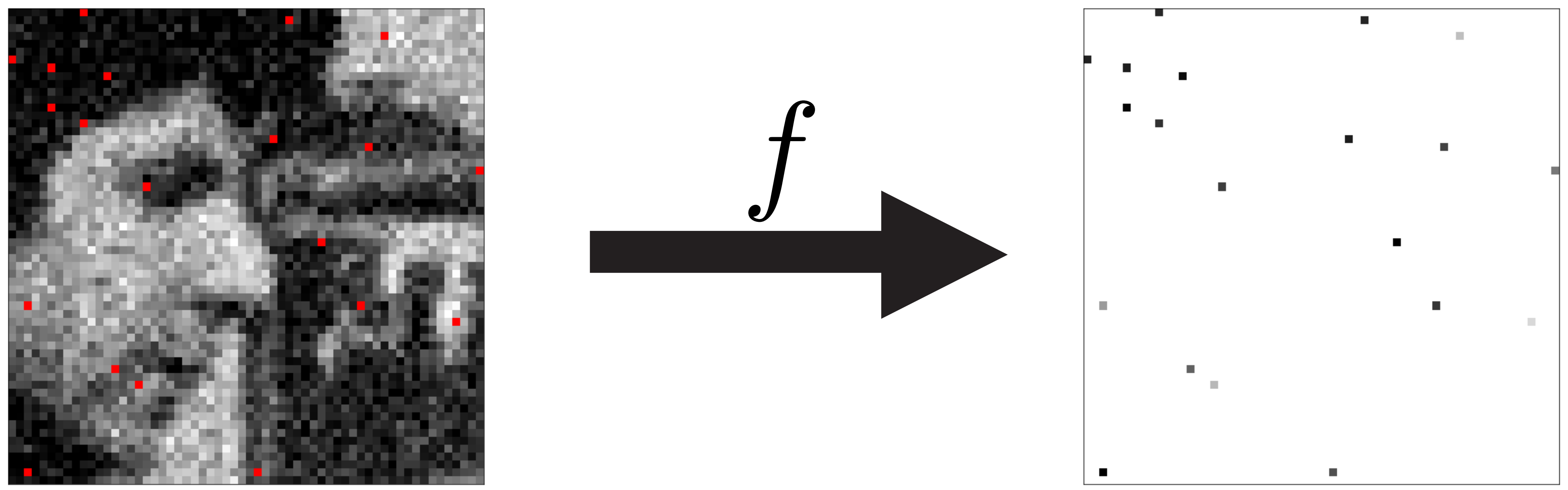
$$f^* = \text{Identity}$$

Self-Supervision?

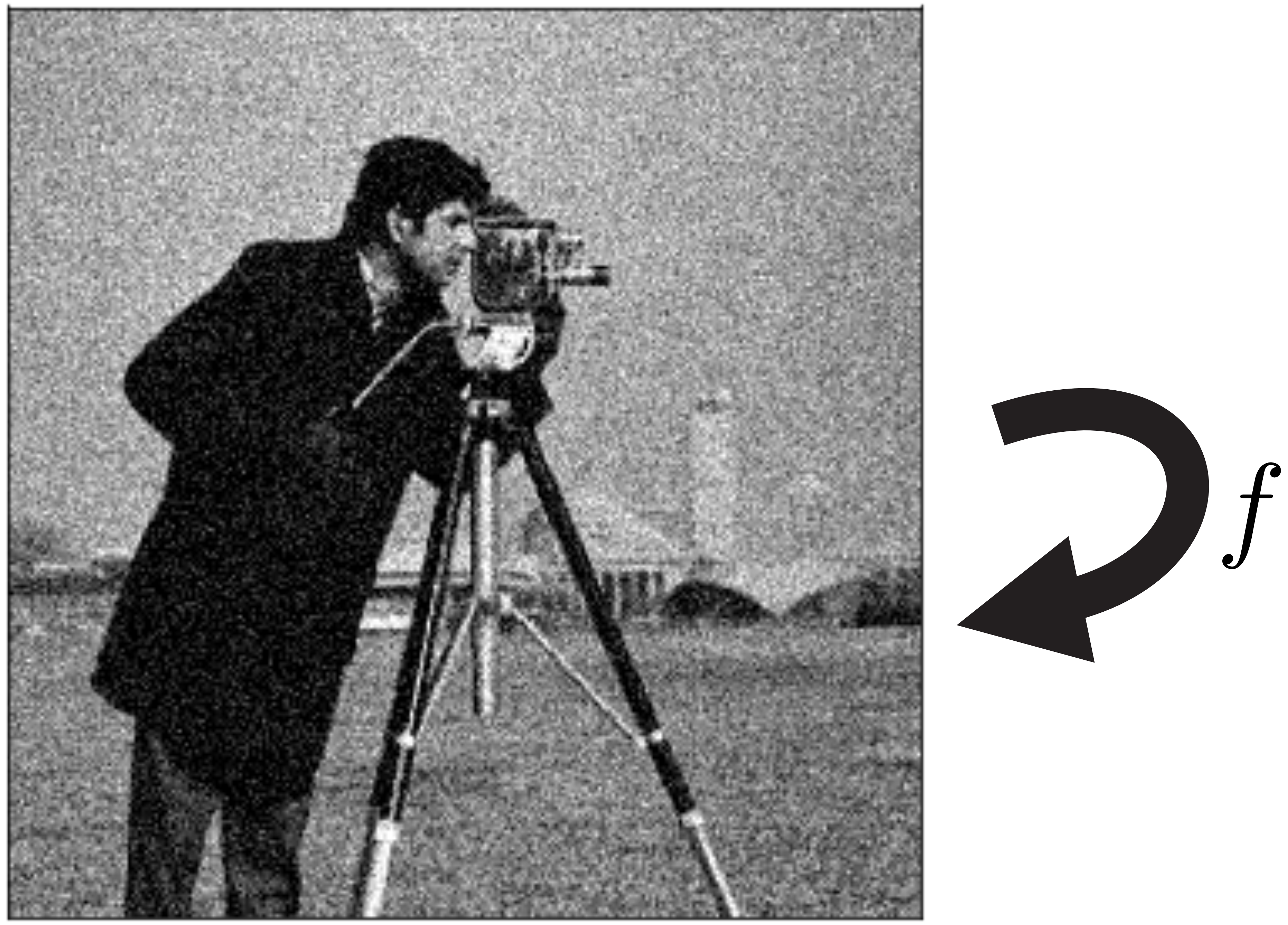


$$\|f(x) - x\|^2$$

$$f^* = \text{Identity}$$

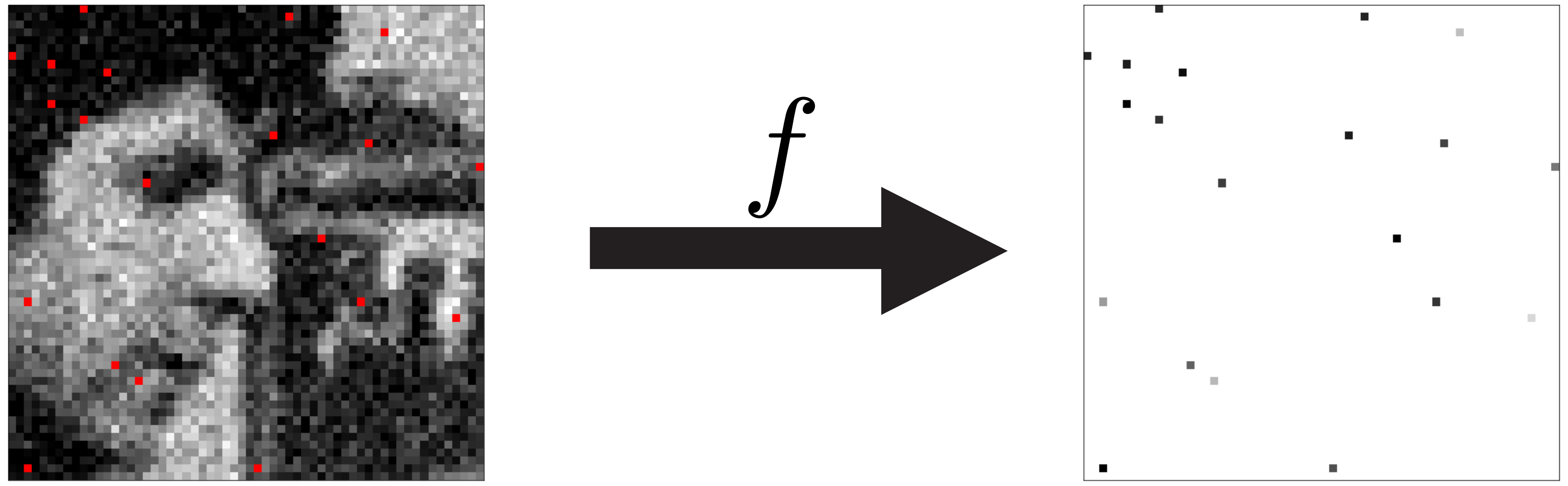


Self-Supervision?

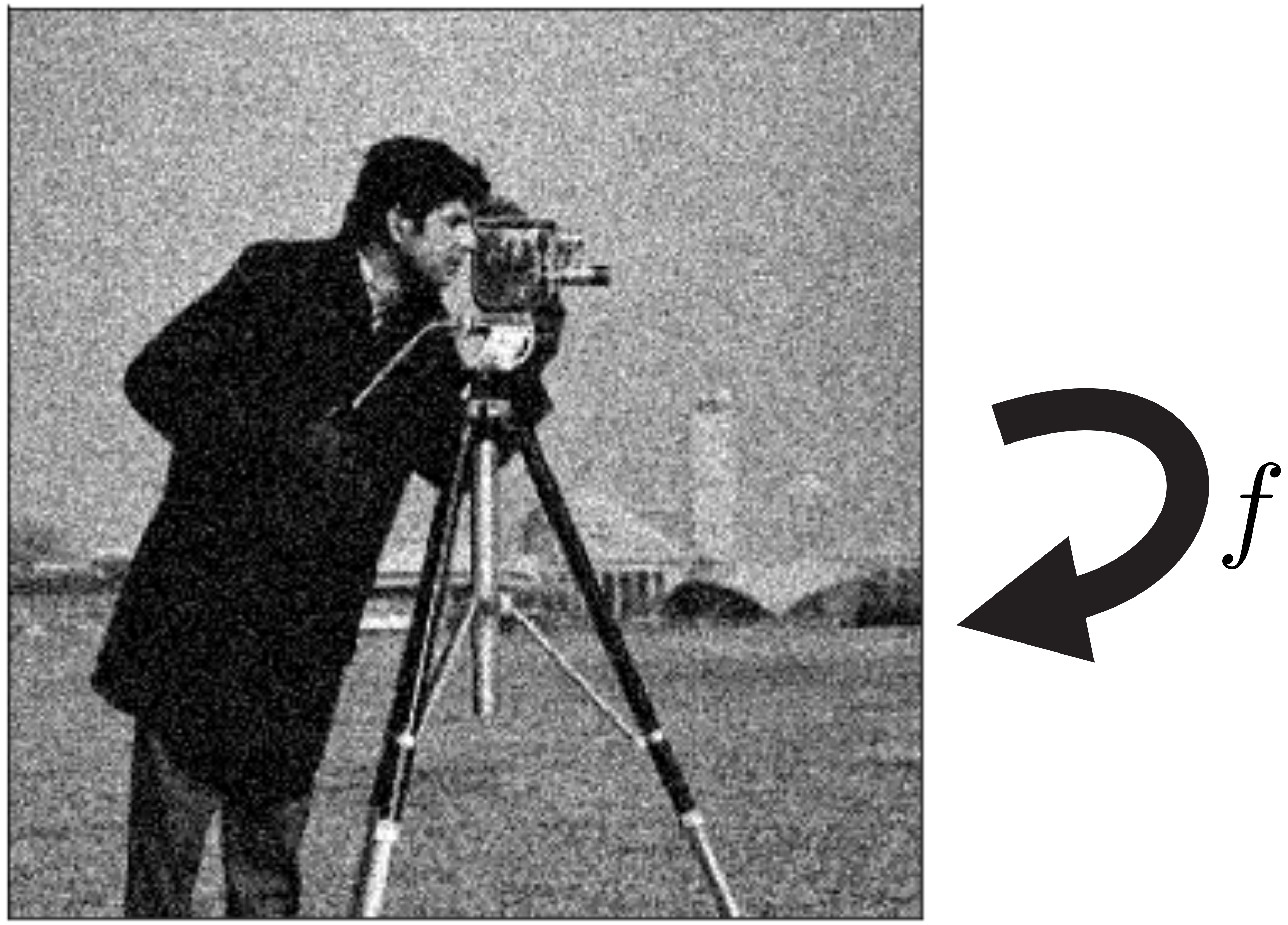


$$\|f(x) - x\|^2$$

~~$f^* =$ Identity~~

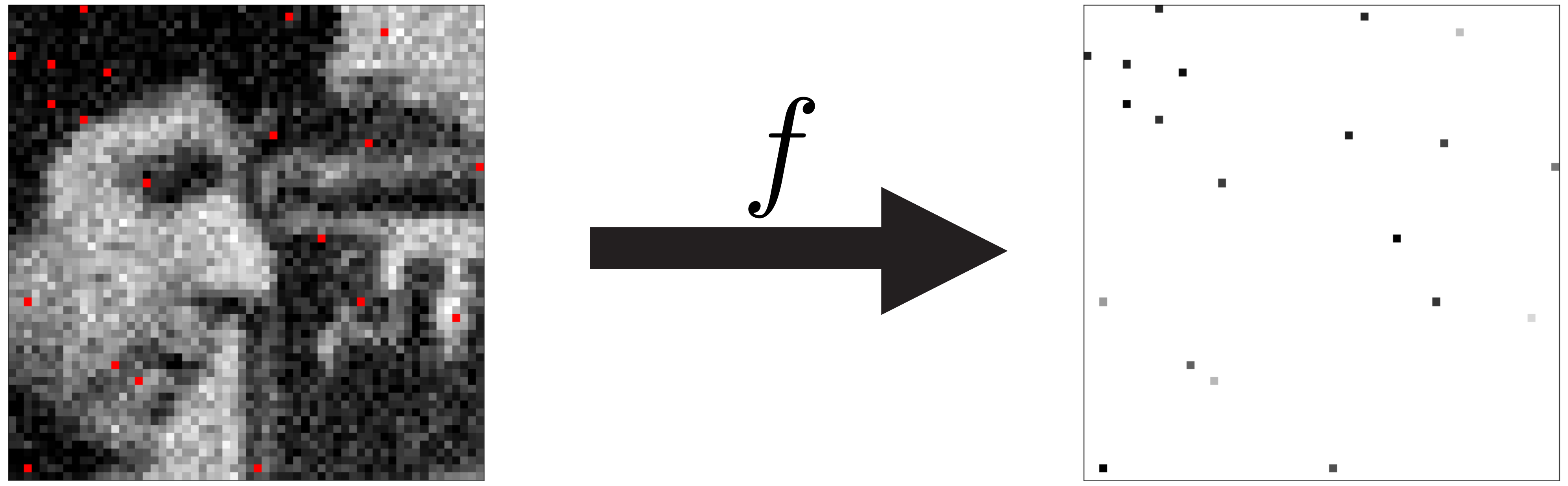


Self-Supervision?



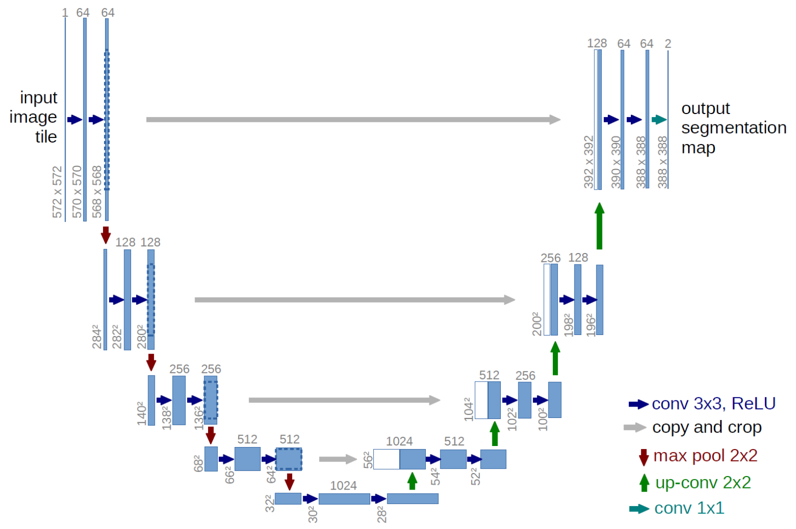
$$\|f(x) - x\|^2$$

~~$f^* = \text{Identity}$~~

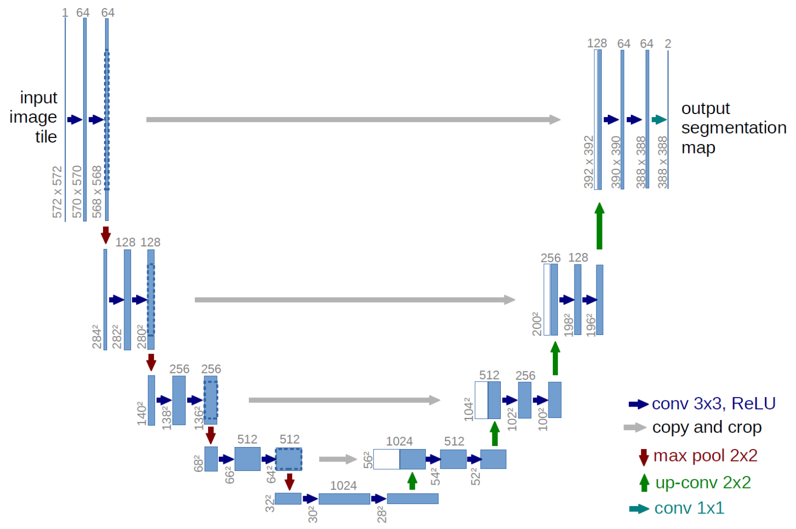


$$f^* = \mathbb{E}[x_{-J} | x_J]$$

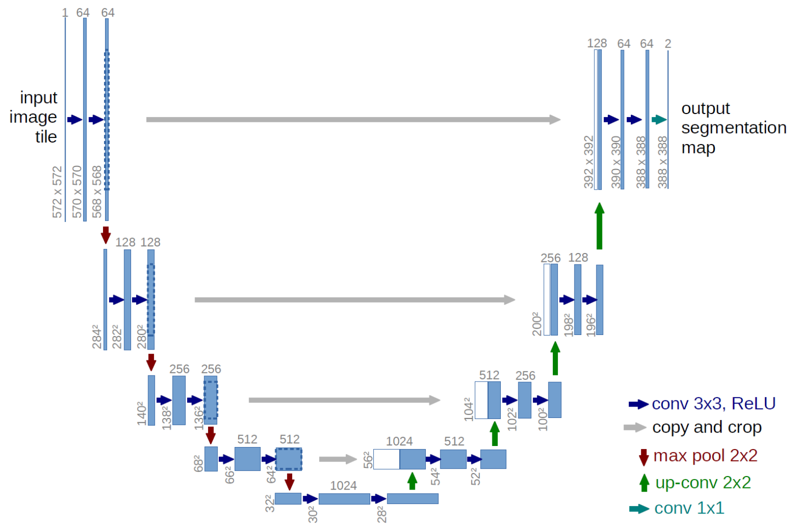
Single-Image Self-Supervised CNN Training



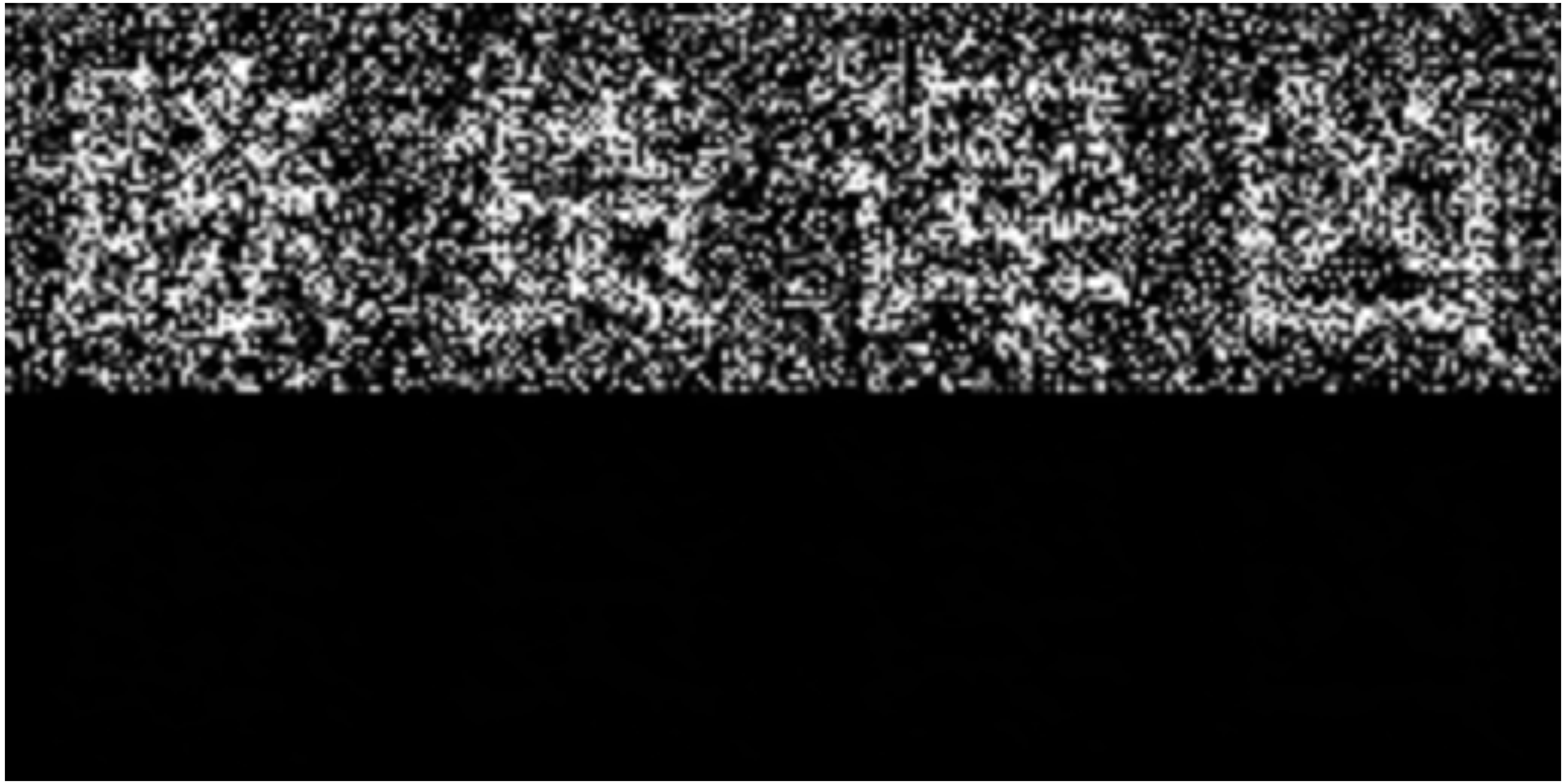
Single-Image Self-Supervised CNN Training



Single-Image Self-Supervised CNN Training



J-invariant Deep CNN



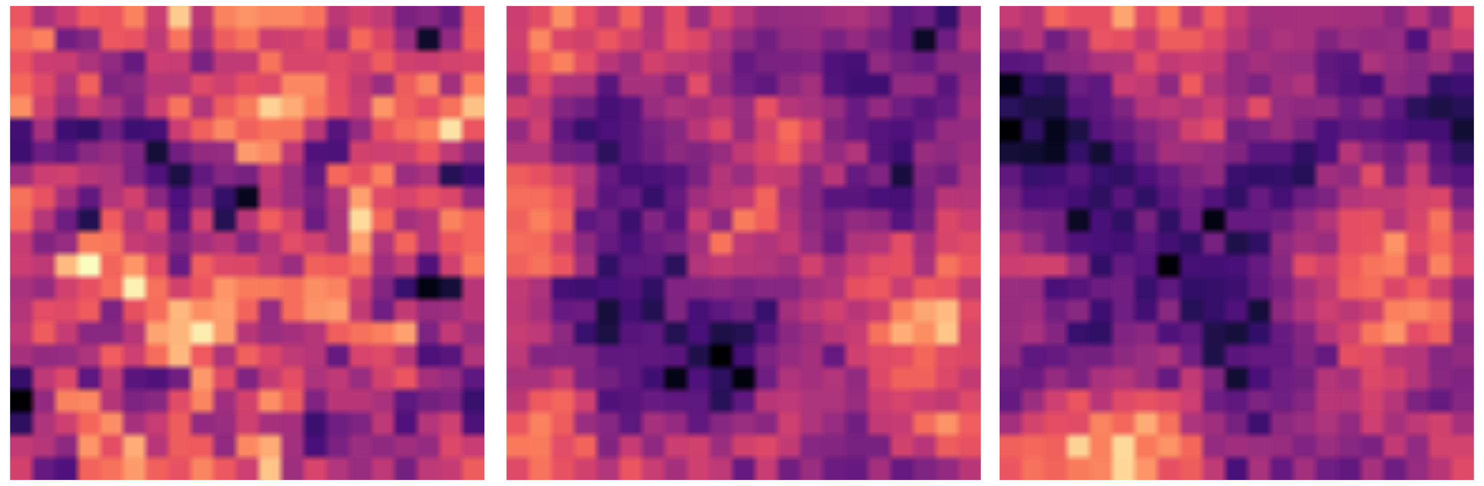
J-invariant Deep CNN

Plus...

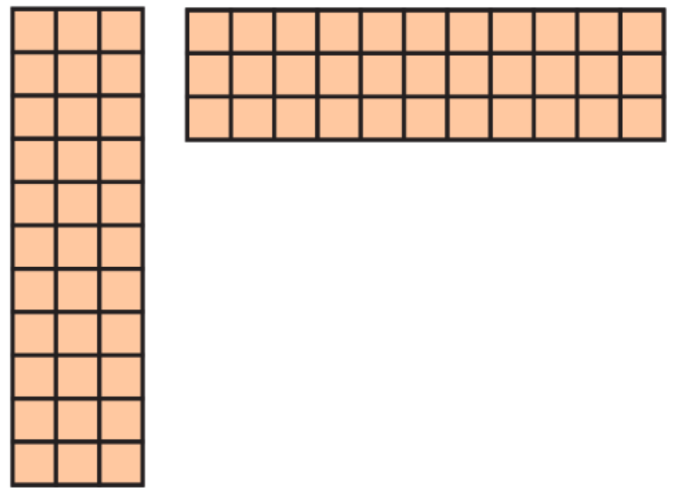
Definitions

Definition. Let $\mathcal{J}$ be a partition of the dimensions $\{1, \ldots, m\}$ and let $J \in \mathcal{J}$. A function $f : \mathbb{R}^m \to \mathbb{R}^m$ is *J-invariant* if $f(x)_J$ does not depend on the value of x_J. It is *$\mathcal{J}$-invariant* if it is *J-invariant* for each $J \in \mathcal{J}$.

Gaussian Processes



Matrix Factorization

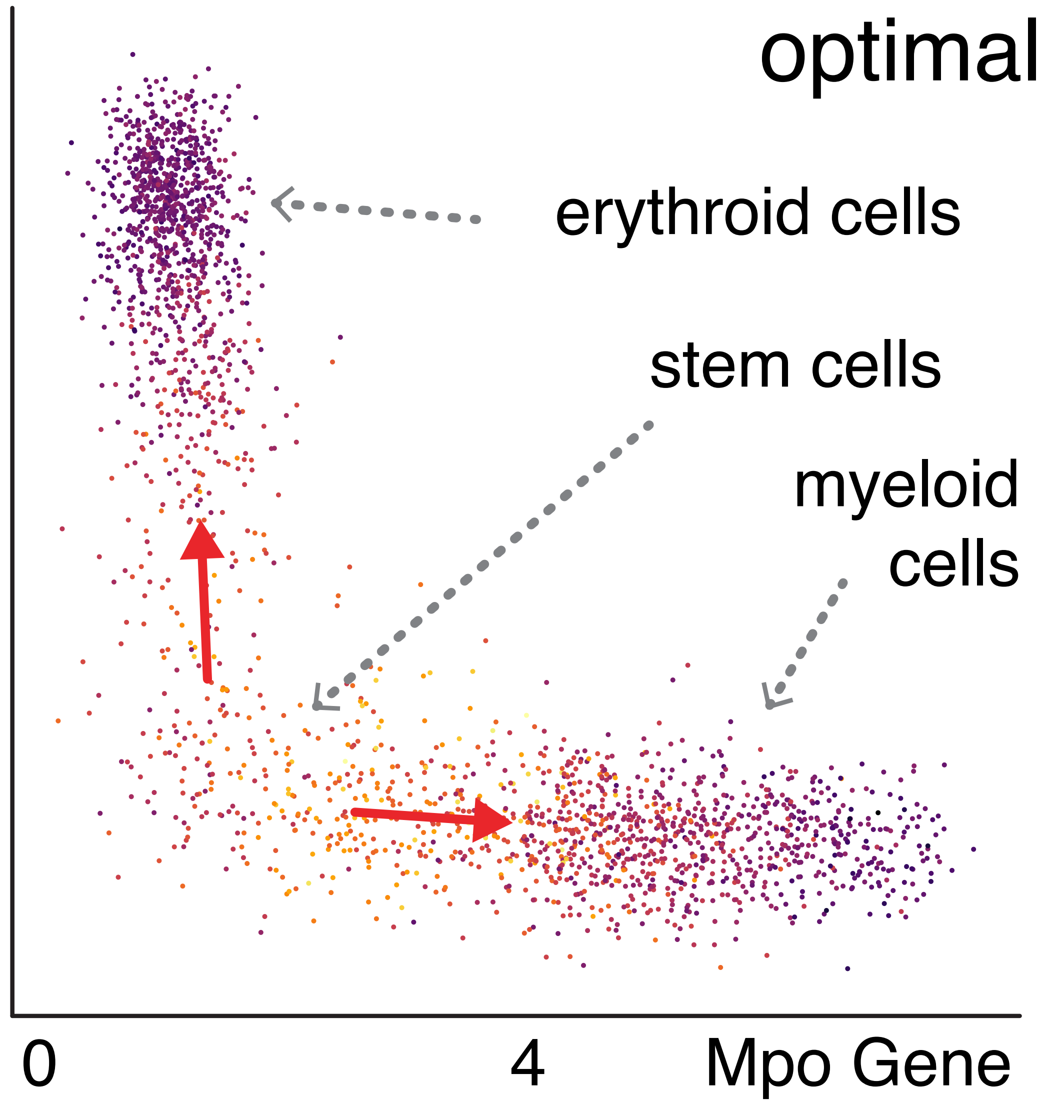


Theorems

Proposition 3. *Let x, y be random variables and let x^G and y^G be Gaussian random variables with the same covariance matrix. Let $f^*_{\mathcal{J}}$ and $f^{*,G}_{\mathcal{J}}$ be the corresponding optimal $\mathcal{J}$-invariant predictors. Then*

$$\mathbb{E}\|y - f^*_{\mathcal{J}}(x)\|^2 \leq \mathbb{E}\|y - f^{*,G}_{\mathcal{J}}(x)\|^2.$$

Single-Cell Sequencing



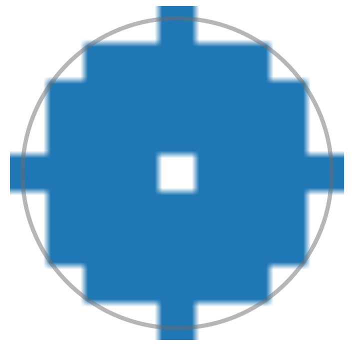
Code

```
for i, batch in enumerate(data_loader):
    noisy_images = batch
    input, mask = masker.mask(noisy_images, i)
    output = model(input)
    loss = loss_function(output*mask,
                         noisy_images*mask)
```

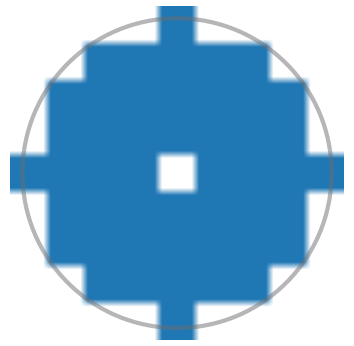
poster #118

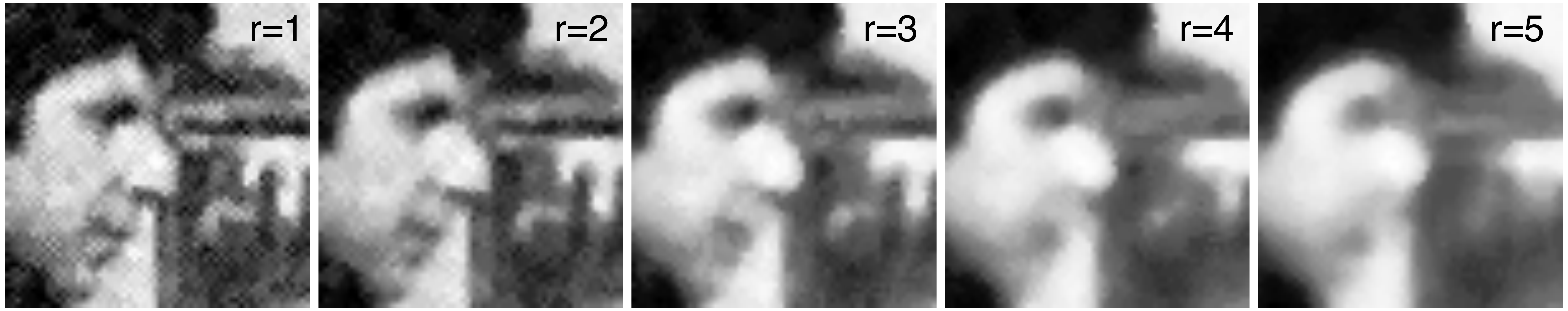
github.com/czbiohub/noise2self

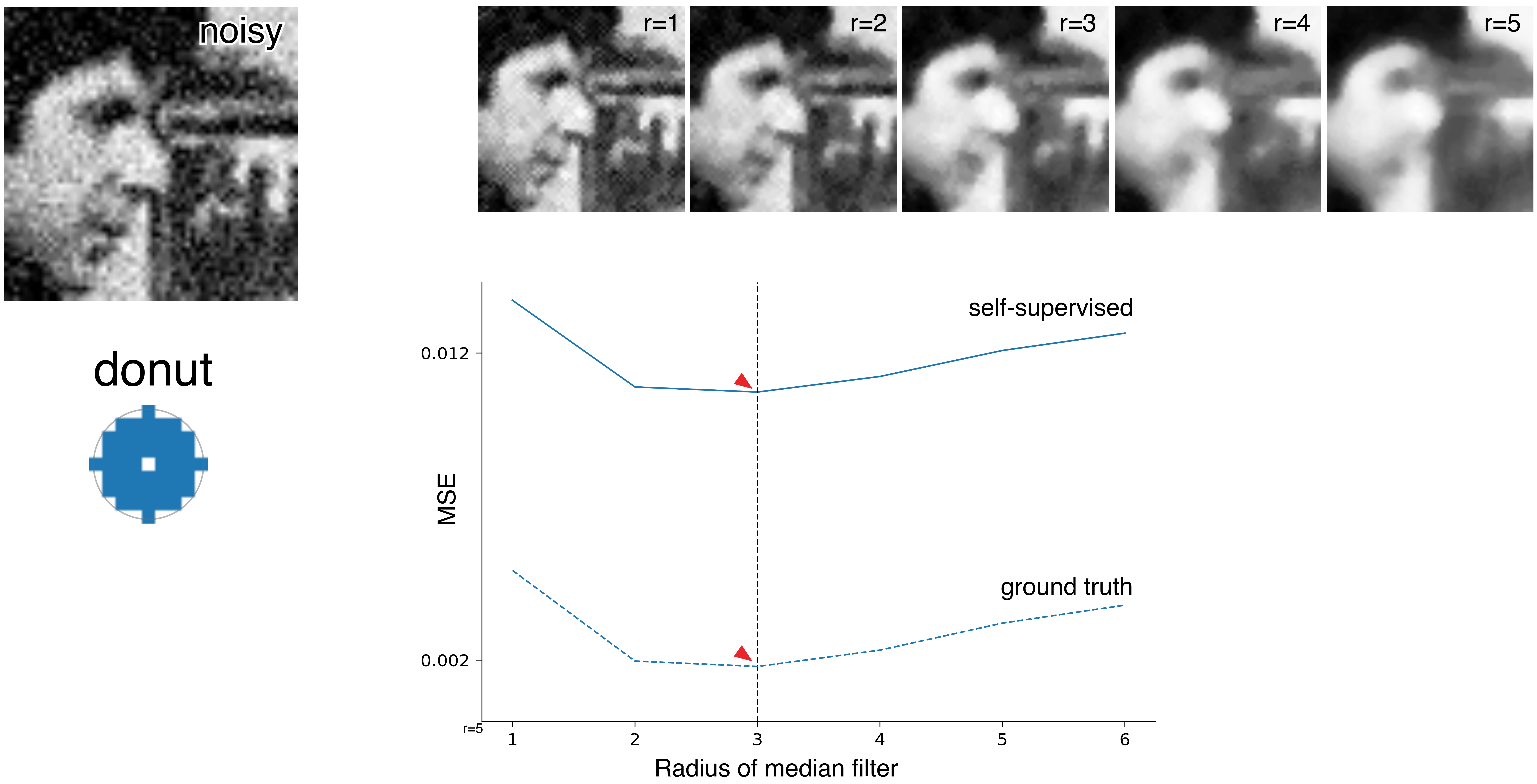
donut



donut






noisy
donut
r=1
r=2
r=3
r=4
r=5
self-supervised
ground truth
MSE
0.012
0.002
r=5
1
2
3
4
5
6
Radius of median filter