

# Tutorial 13: Deep losses and priors

Digital Image Processing (236860)

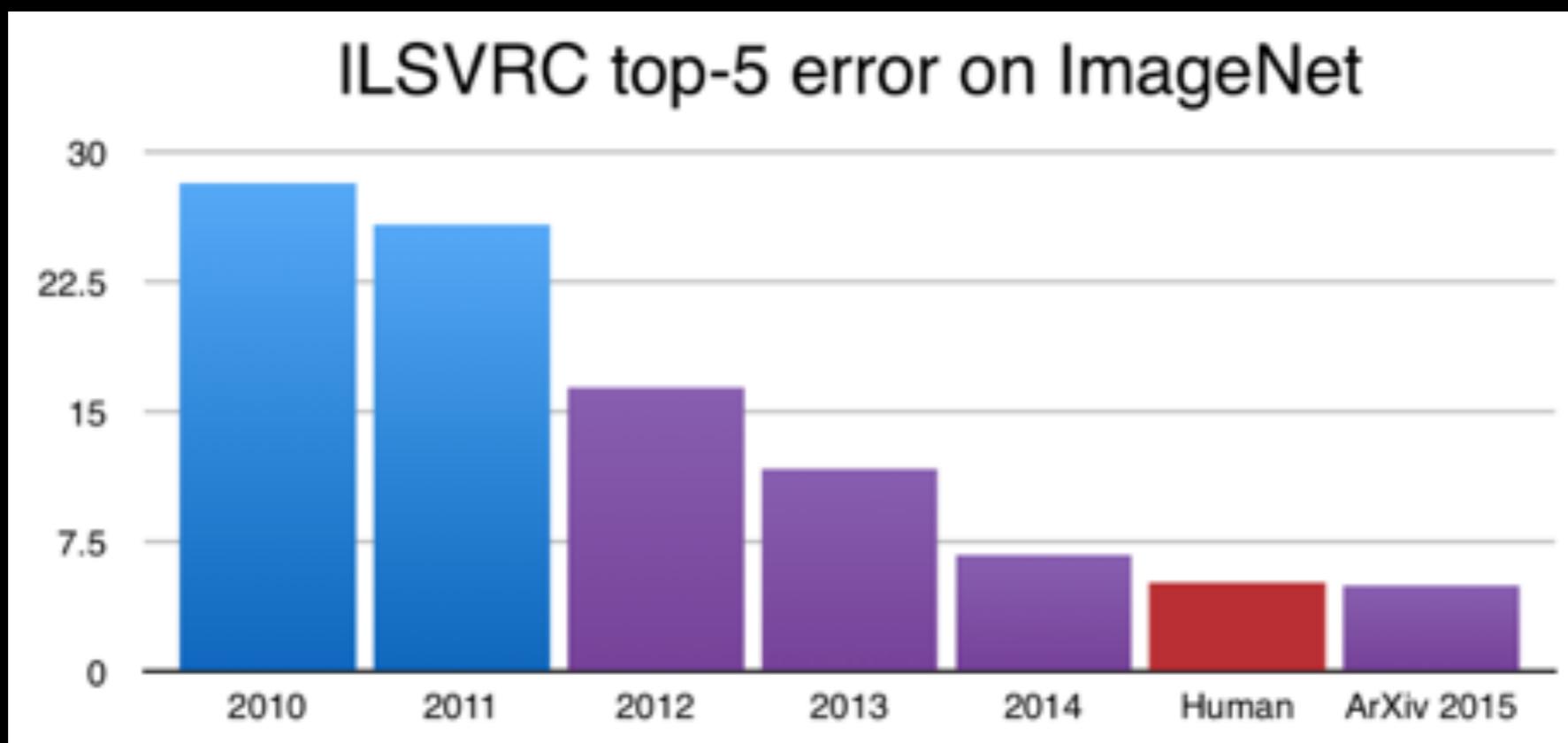


# Agenda

- Common architectures: AlexNet, VGGNet
- Perceptual losses
- Deep image prior

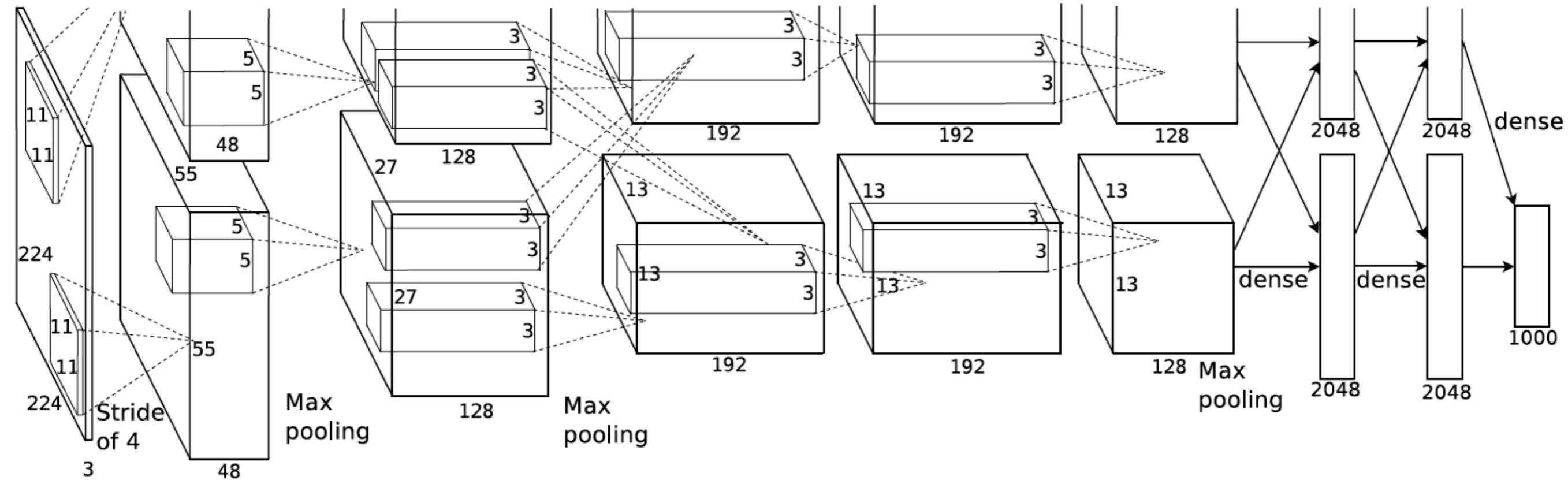
# Advances in image classification

- ImageNet: a dataset of 1M images with 1000 categories.



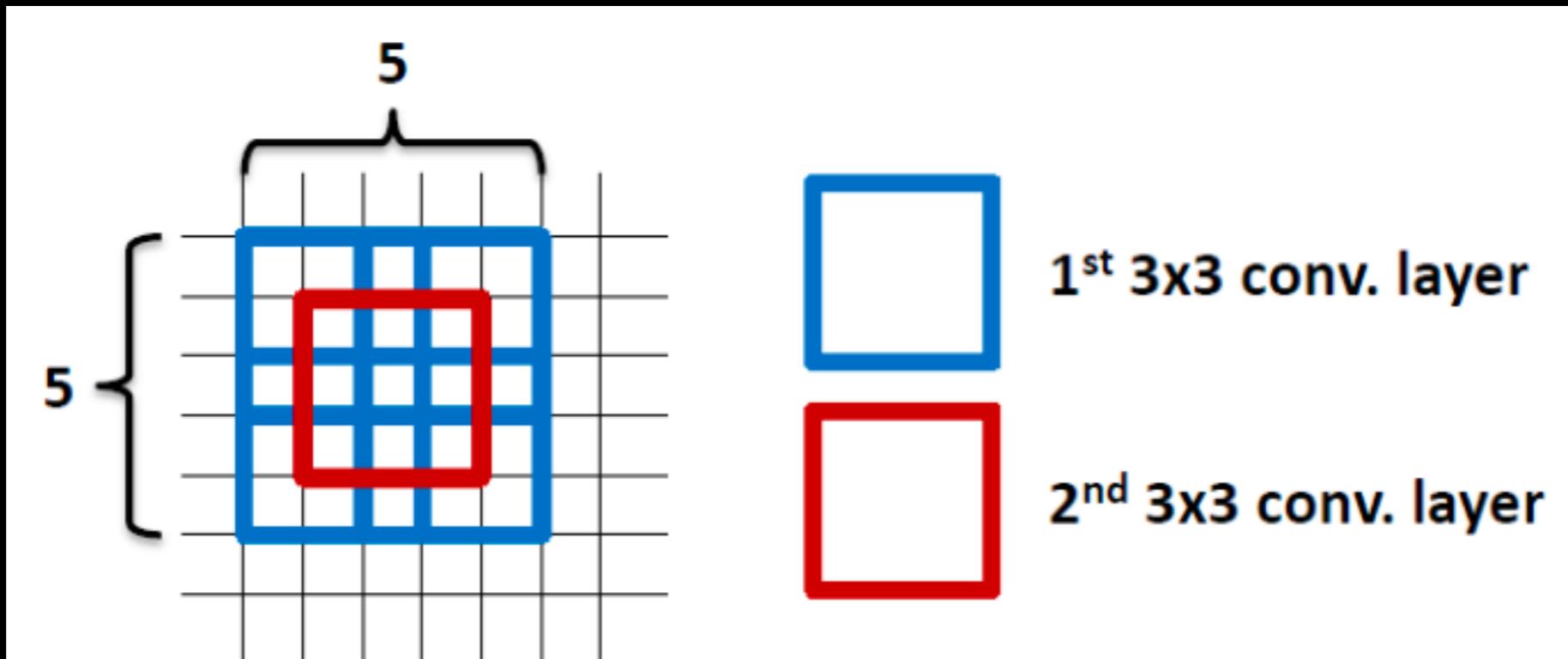
# AlexNet

- Primary result: the depth of the model is essential to its high performance.
- The utilisation of GPUs made this model feasible.

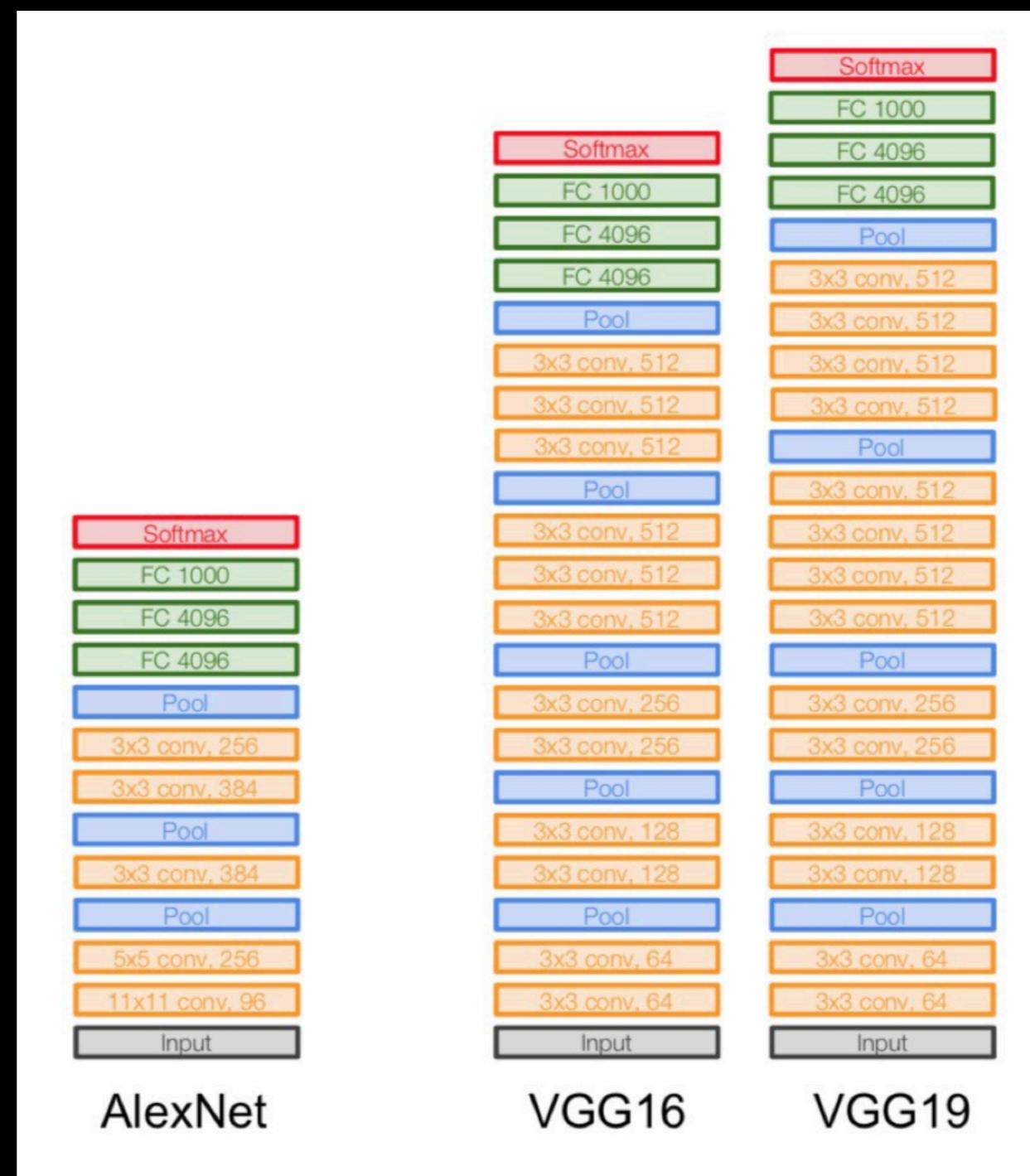


# VGGNet

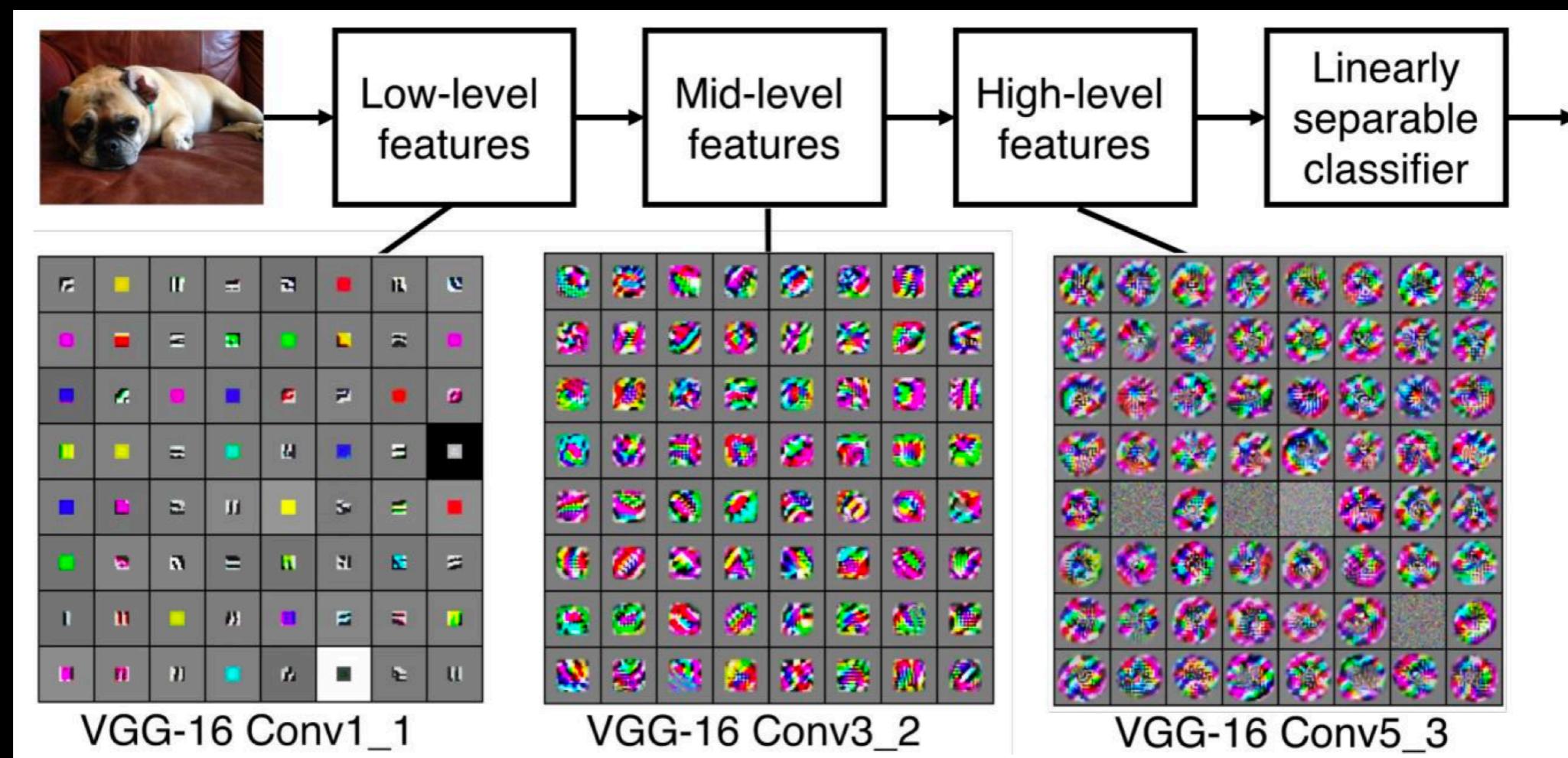
- Only uses 3x3 filters.
- 1 layer of 5x5 filter =>  $5 \times 5 = 25$  parameters.
- 2 layers of 3x3 filter =>  $3 \times 3 \times 2 = 18$  parameters.
- Number of parameters is reduced by 28%!



# VGGNet

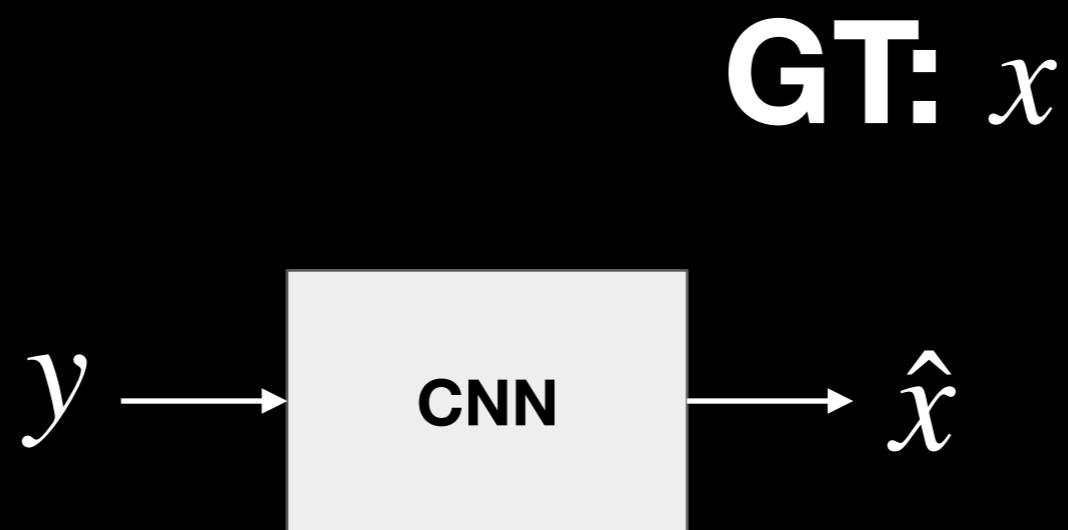


# VGGNet



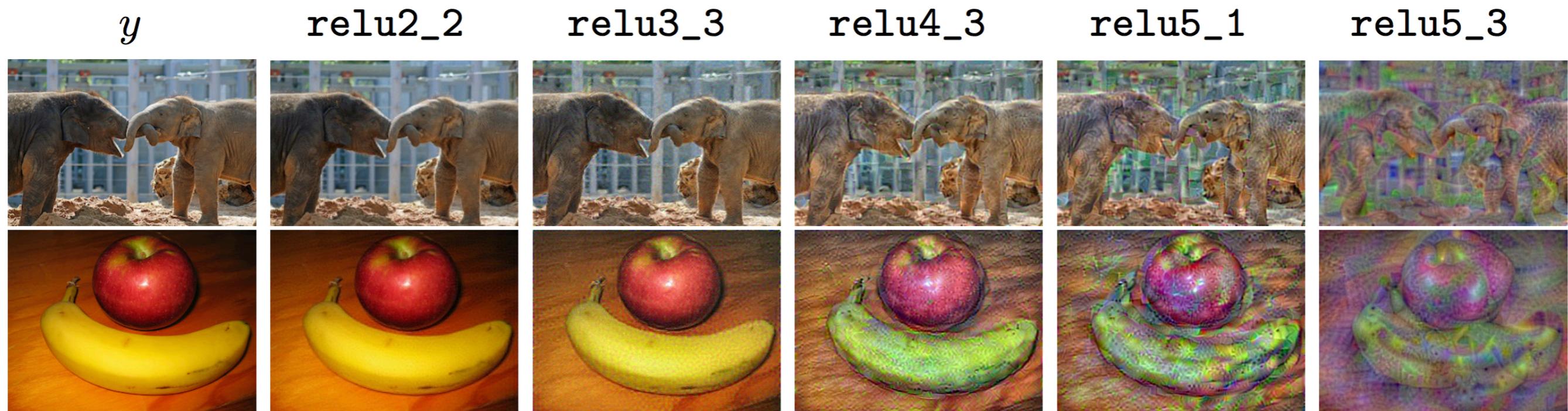
# Perceptual losses

- Instead of minimising per pixel discrepancy  $\| \hat{x} - x \|$ , minimise a loss that measures perception.
- Assume we have a pre-trained VGG classification network  $\phi$ .



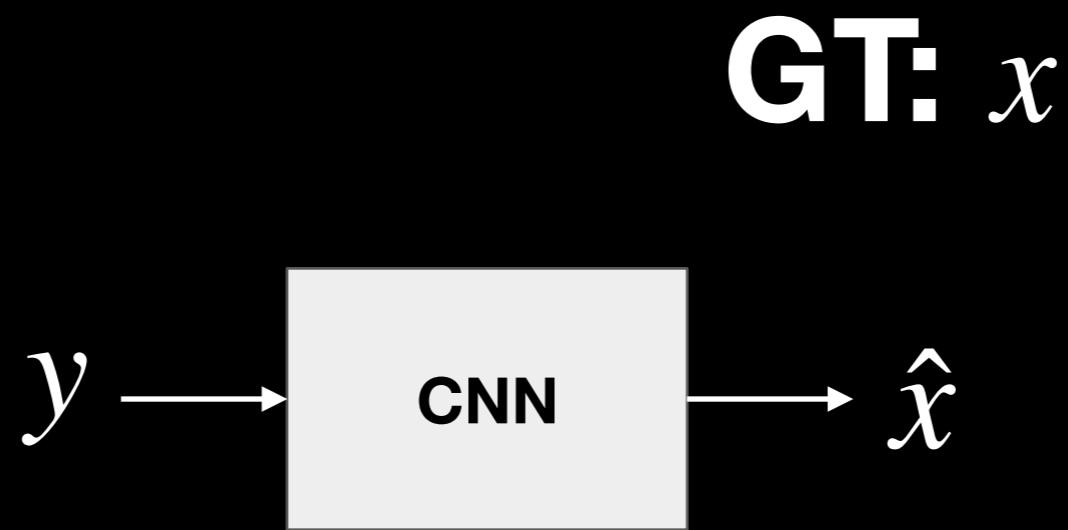
# Feature construction loss

- VGG features contain semantic information.



# Feature construction loss

- Minimise  $\parallel \phi_j(\hat{x}) - \phi_j(x) \parallel$ , where  $\phi_j$  are the activations of the  $j$ -th layer of the network.



# Feature reconstruction loss for super resolution



Ground Truth	Bicubic	Ours ( $\ell_{pixel}$ )	SRCNN [11]	Ours ( $\ell_{feat}$ )
This image	31.78 / 0.8577	31.47 / 0.8573	32.99 / 0.8784	29.24 / 0.7841
Set5 mean	28.43 / 0.8114	28.40 / 0.8205	30.48 / 0.8628	27.09 / 0.7680

**PSNR/SSIM are not good perceptual measures!**

# Neural style transfer

Content Image



Style Image



Style Transfer!



+

=

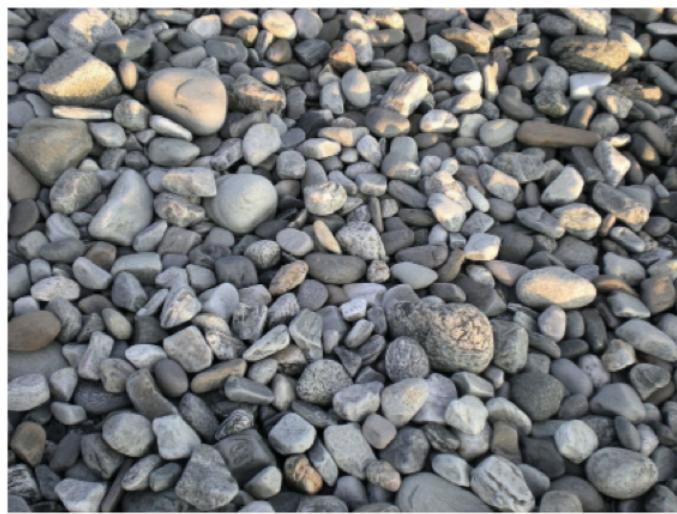
feature reconstruction loss



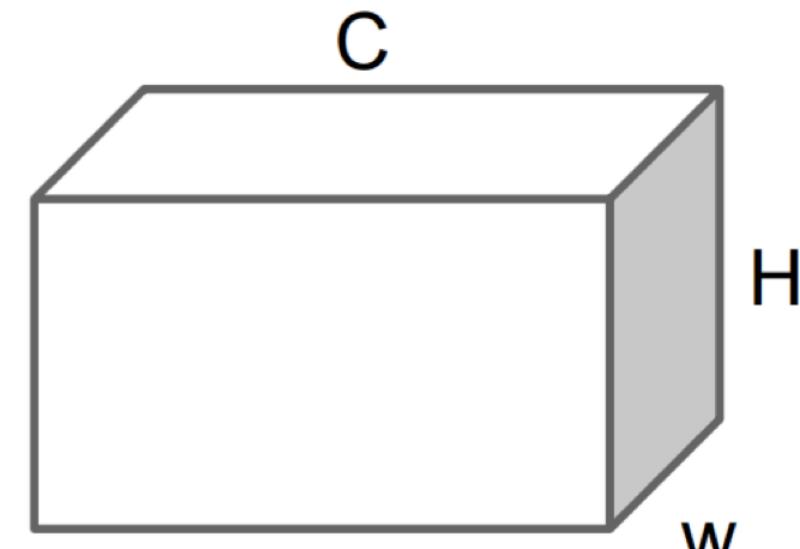
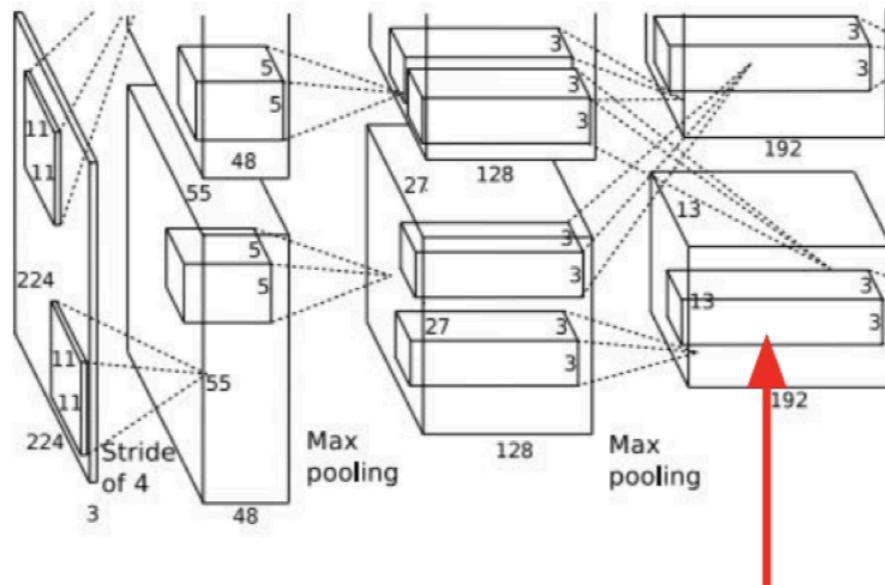
How can we measure “style”?



# Style reconstruction loss



This image is in the public domain.

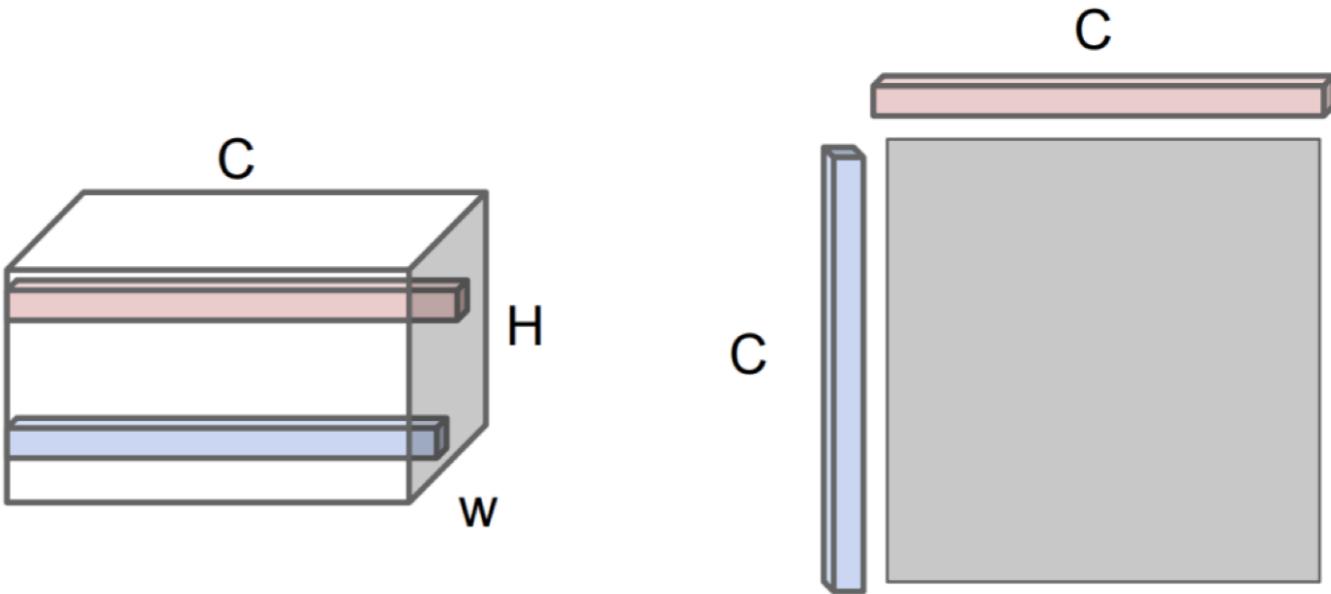
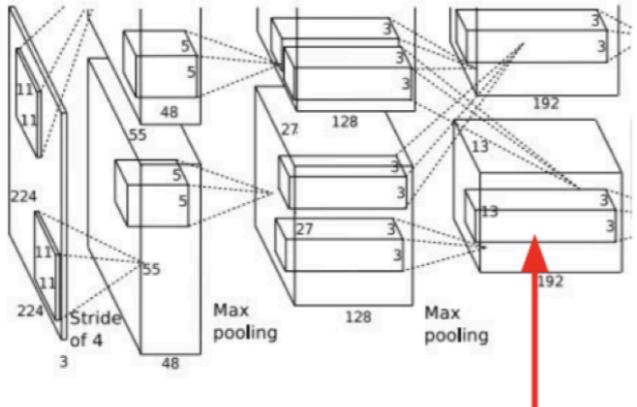


Each layer of CNN gives  $C \times H \times W$  tensor of features;  $H \times W$  grid of  $C$ -dimensional vectors

# Style reconstruction loss



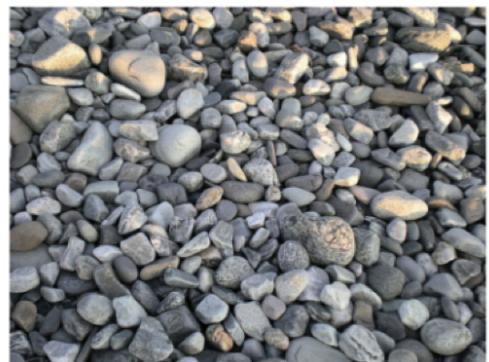
[This image is in the public domain.](#)



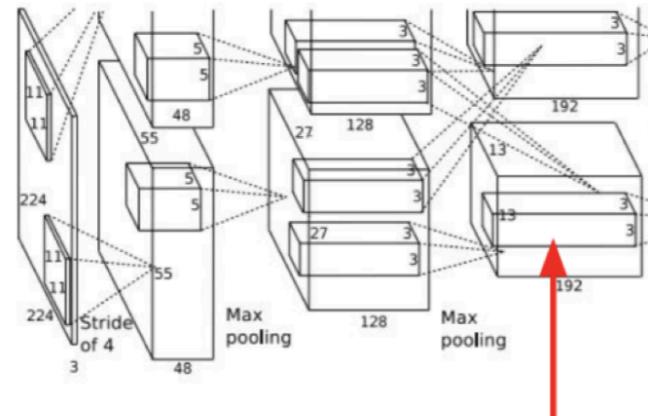
Each layer of CNN gives  $C \times H \times W$  tensor of features;  $H \times W$  grid of  $C$ -dimensional vectors

Outer product of two  $C$ -dimensional vectors gives  $C \times C$  matrix measuring co-occurrence

# Style reconstruction loss



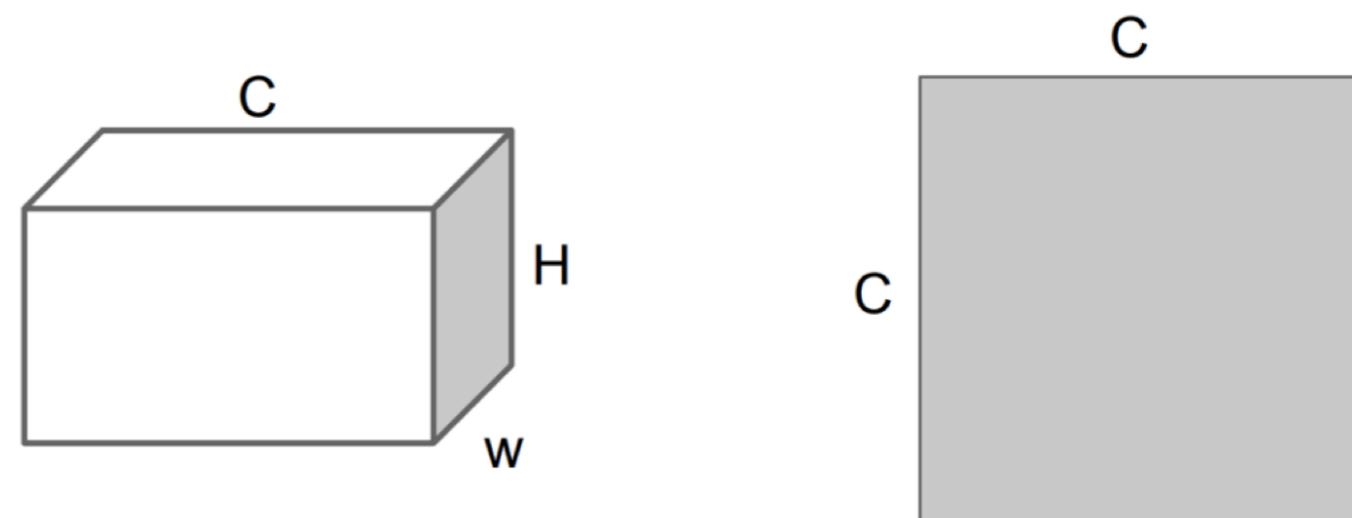
This image is in the public domain.



Each layer of CNN gives  $C \times H \times W$  tensor of features;  $H \times W$  grid of  $C$ -dimensional vectors

Outer product of two  $C$ -dimensional vectors gives  $C \times C$  matrix measuring co-occurrence

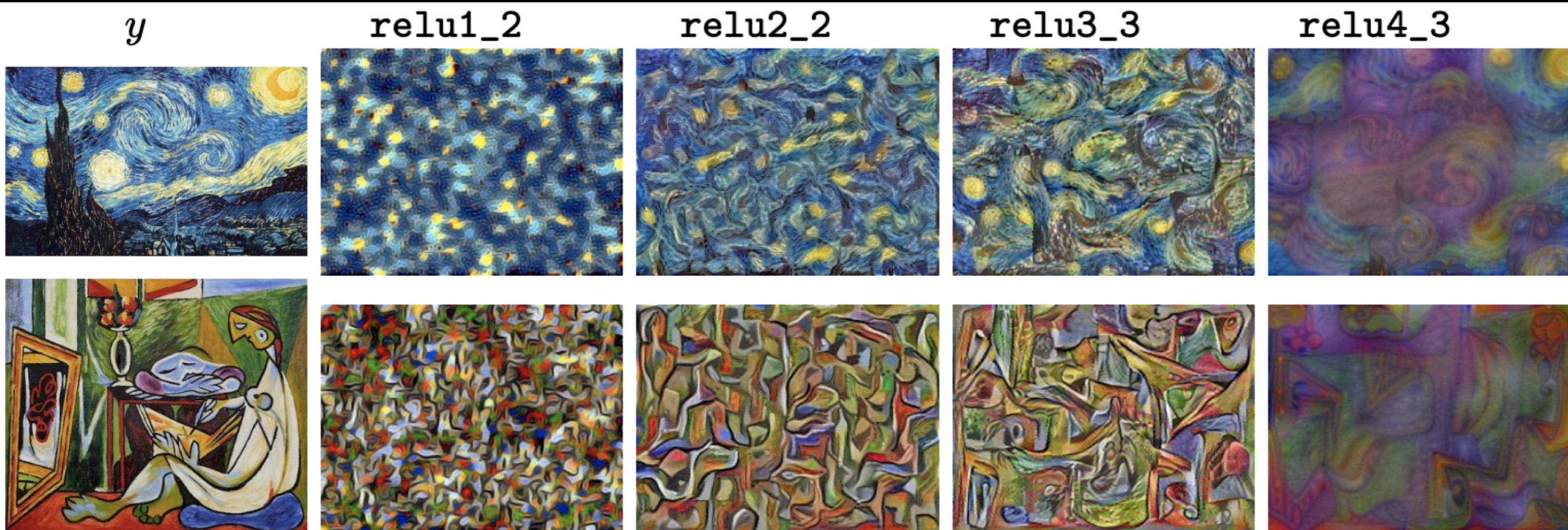
Average over all  $HW$  pairs of vectors, giving **Gram matrix** of shape  $C \times C$



The gram matrix doesn't carry any spatial information!

$$G_j^\phi(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}$$

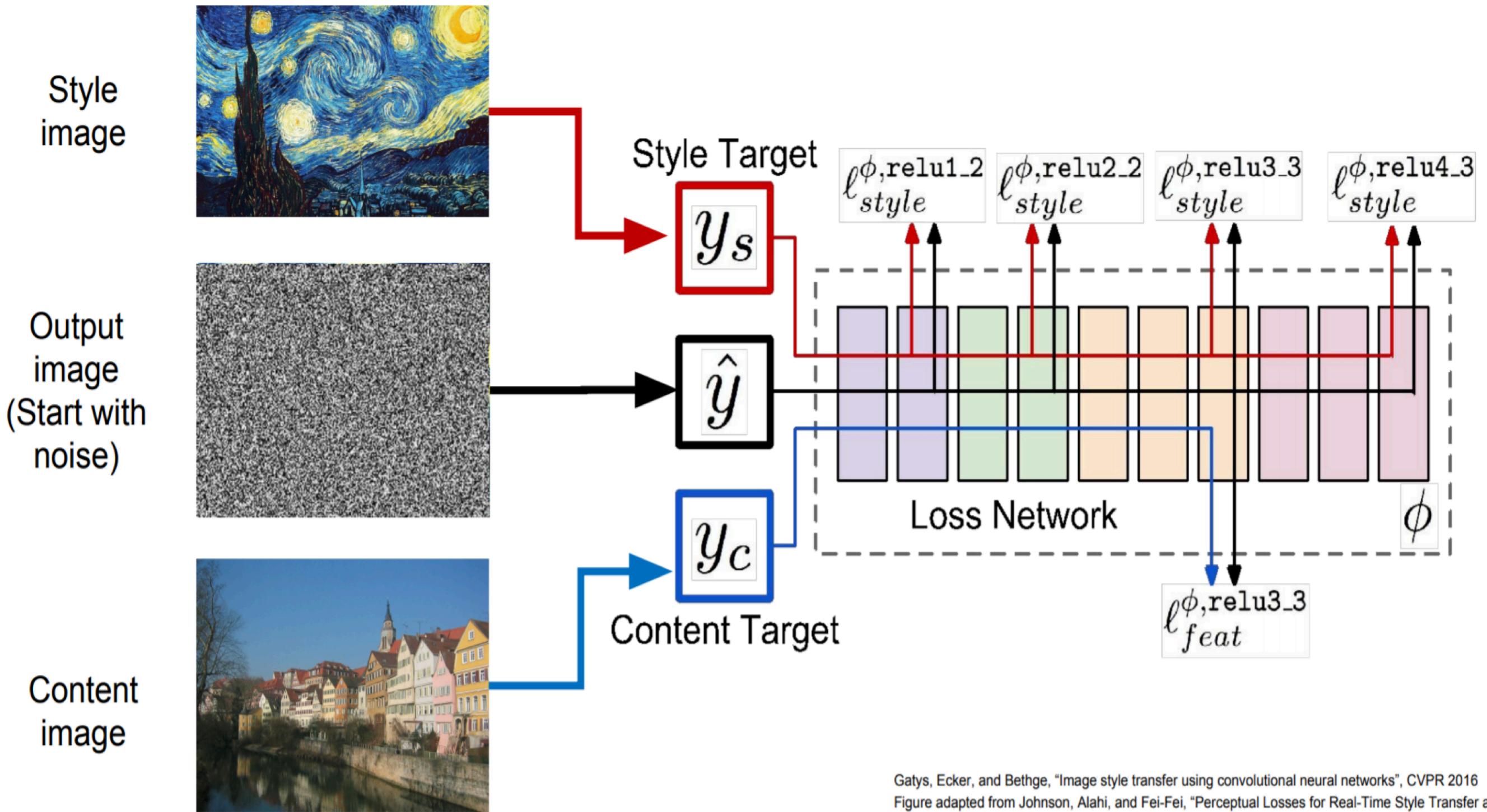
# Style reconstruction loss



# Summary: conceptual losses

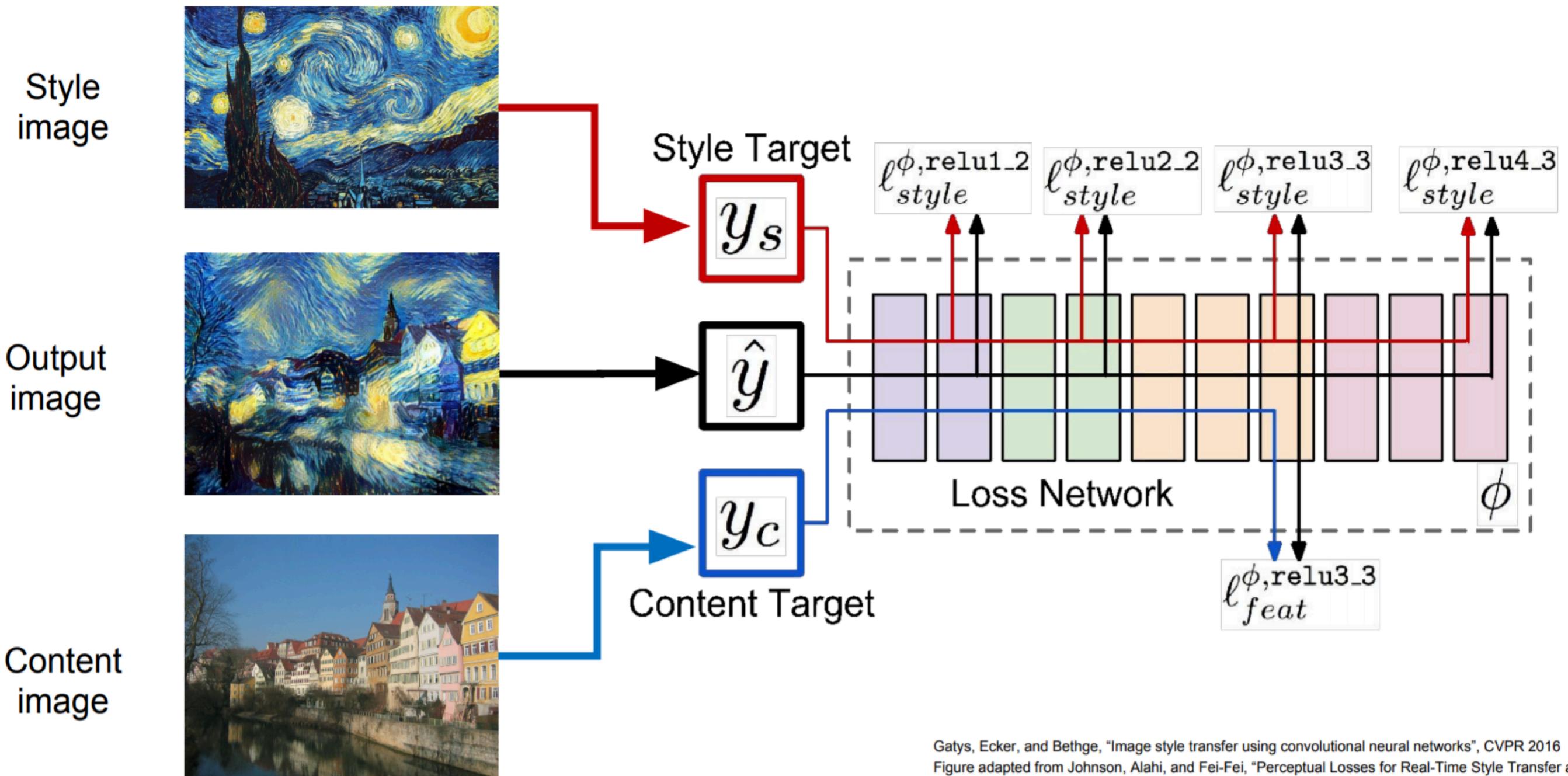
- Feature reconstruction loss:  $\left\| \phi_j(\hat{x}) - \phi_j(x) \right\|$
- Style reconstruction loss:  $\left\| G_j^\phi(\hat{x}) - G_j^\phi(x) \right\|$

# Neural style transfer



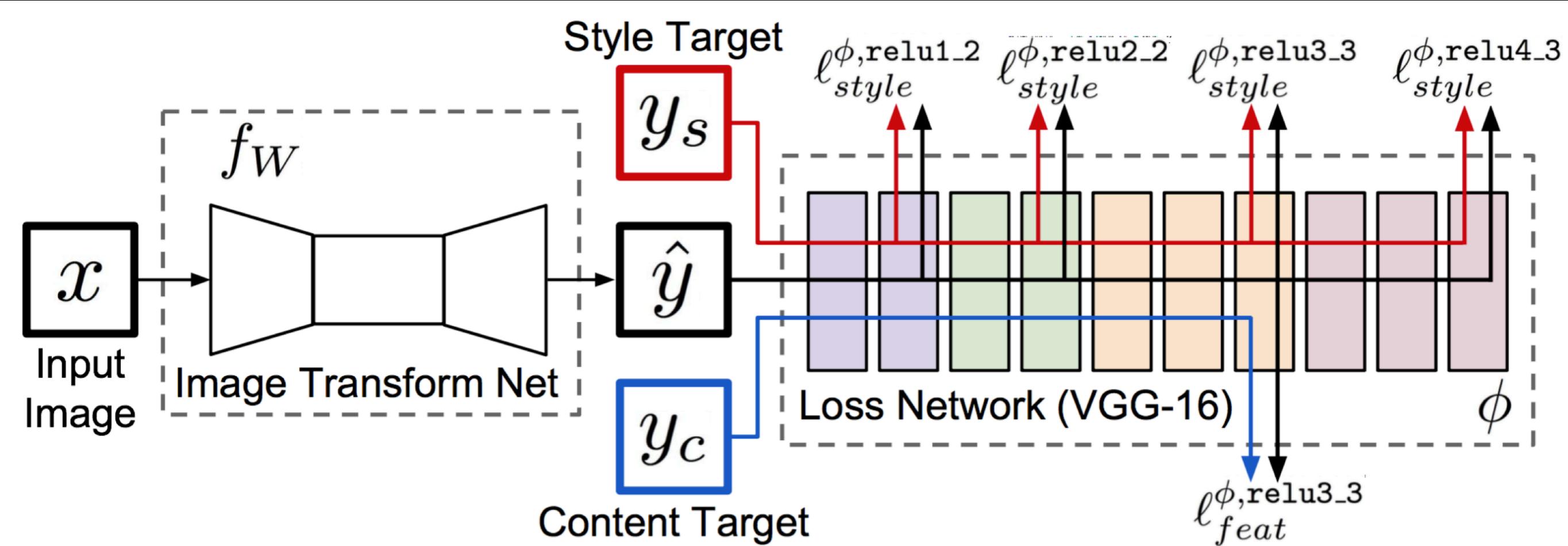
Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016  
Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

# Neural style transfer



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016  
Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

# Neural style transfer



$$\hat{y} = \arg \min_y \lambda_c \ell_{feat}^{\phi, j}(y, y_c) + \lambda_s \ell_{style}^{\phi, J}(y, y_s) + \lambda_{TV} \ell_{TV}(y)$$

# Neural style transfer

Style  
*Sketch*



Style  
*The Simpsons*



Content

[10]

Ours

Content

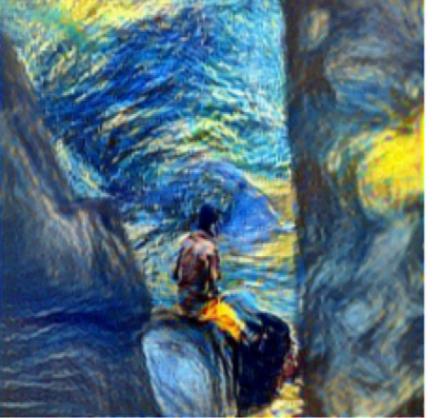
[10]

Ours

# Neural style transfer

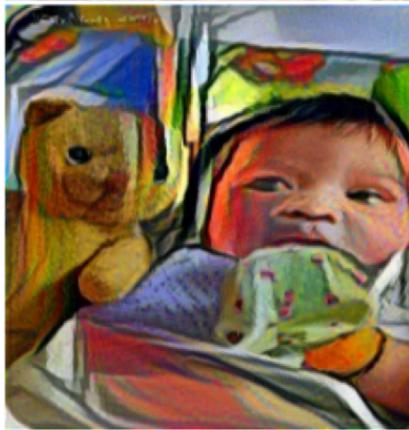
Style

*The Starry Night*,  
Vincent van Gogh,  
1889

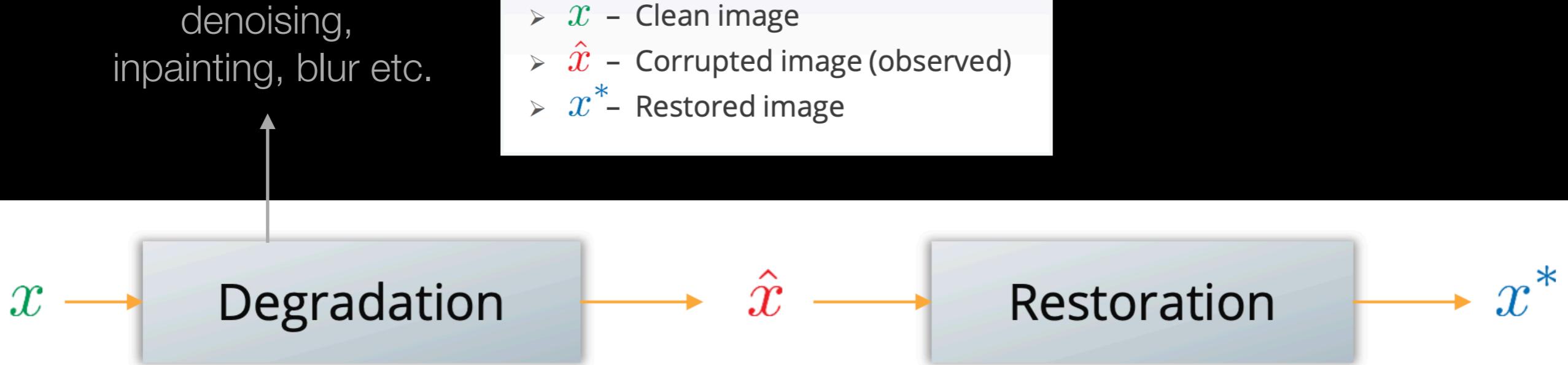


Style

*The Muse*,  
Pablo Picasso,  
1935



# Image priors



**MAP:**  $x^* = \arg \max_x p(\textcolor{teal}{x}|\hat{x})$

$$p(\textcolor{teal}{x}|\hat{x}) = \frac{p(\hat{x}|\textcolor{teal}{x})p(\textcolor{teal}{x})}{p(\hat{x})} \propto p(\hat{x}|\textcolor{teal}{x})p(\textcolor{teal}{x})$$

Likelihood                      Prior

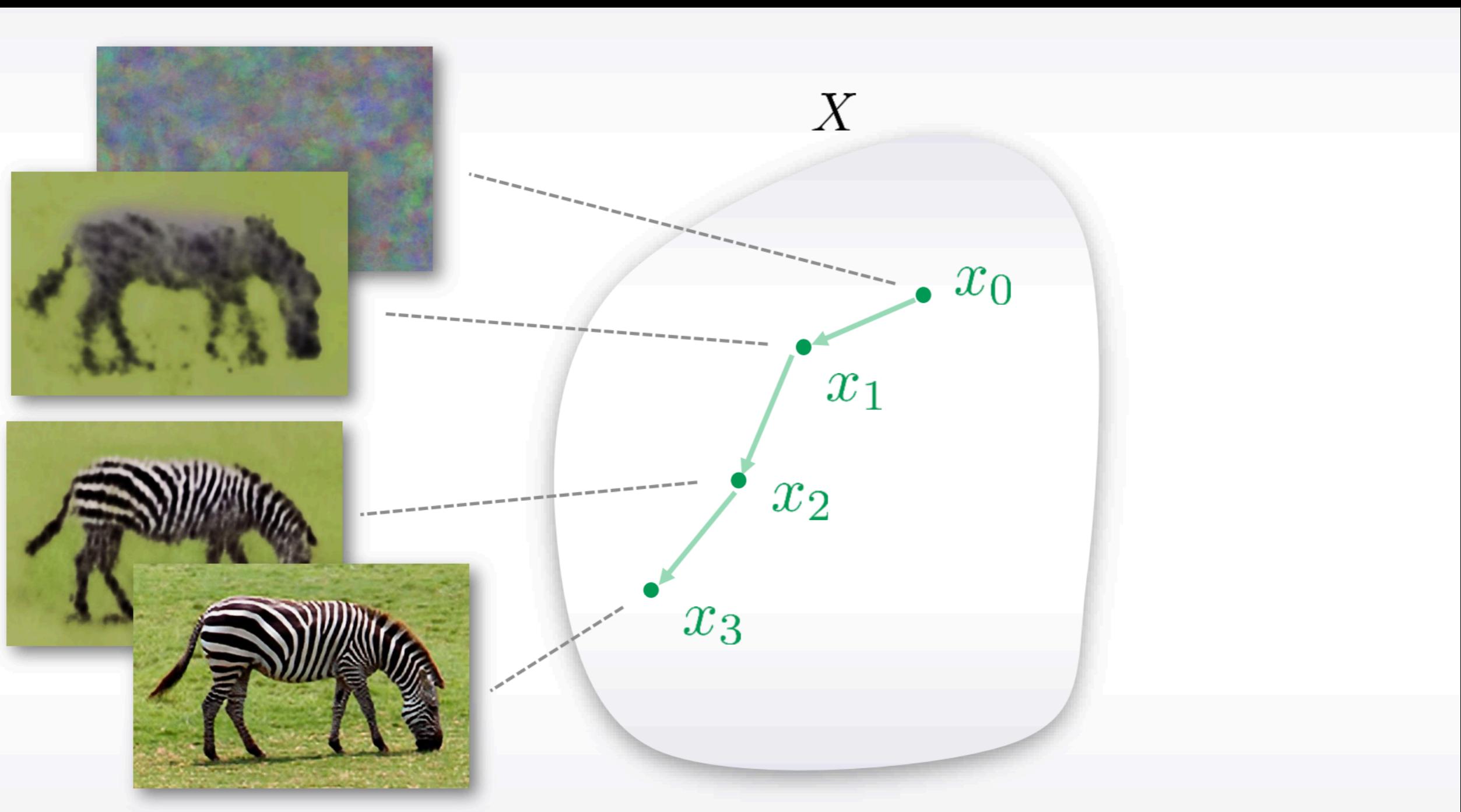
# Image priors

- $\mathbf{x}$  - Clean image
- $\hat{\mathbf{x}}$  - Corrupted image (observed)
- $\mathbf{x}^*$  - Restored image

$$\begin{aligned}\mathbf{x}^* &= \arg \max_{\mathbf{x}} p(\hat{\mathbf{x}}|\mathbf{x})p(\mathbf{x}) \\ &= \arg \min_{\mathbf{x}} -\log p(\hat{\mathbf{x}}|\mathbf{x}) - \log p(\mathbf{x}) \\ &= \arg \min_{\mathbf{x}} E(\mathbf{x}; \hat{\mathbf{x}}) + R(\mathbf{x})\end{aligned}$$

We've seen several instances of how to model priors.

# Image priors



**Regular:**  $\arg \min_x E(\textcolor{teal}{x}; \hat{x}) + R(\textcolor{teal}{x})$

# Parametrised image priors

- $\mathbf{x}$  - Clean image
- $\hat{\mathbf{x}}$  - Corrupted image (observed)
- $\mathbf{x}^*$  - Restored image

**Regular:**

$$\arg \min_{\mathbf{x}} E(\mathbf{x}; \hat{\mathbf{x}}) + R(\mathbf{x})$$

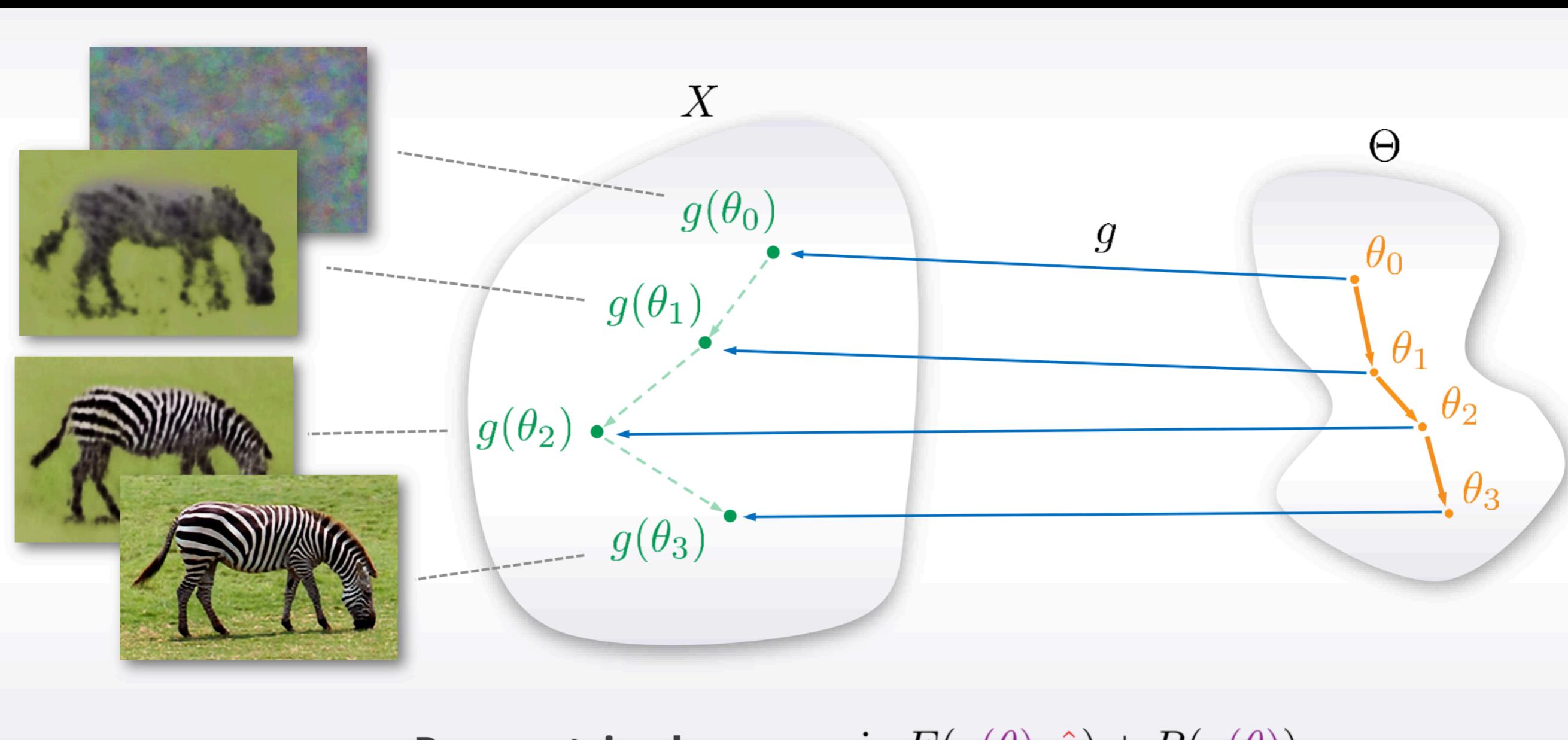
Search in the image space

**Parametrized:**

$$\arg \min_{\theta} E(g(\theta); \hat{\mathbf{x}}) + R(g(\theta))$$

Search in some other space

# Parameterised image priors

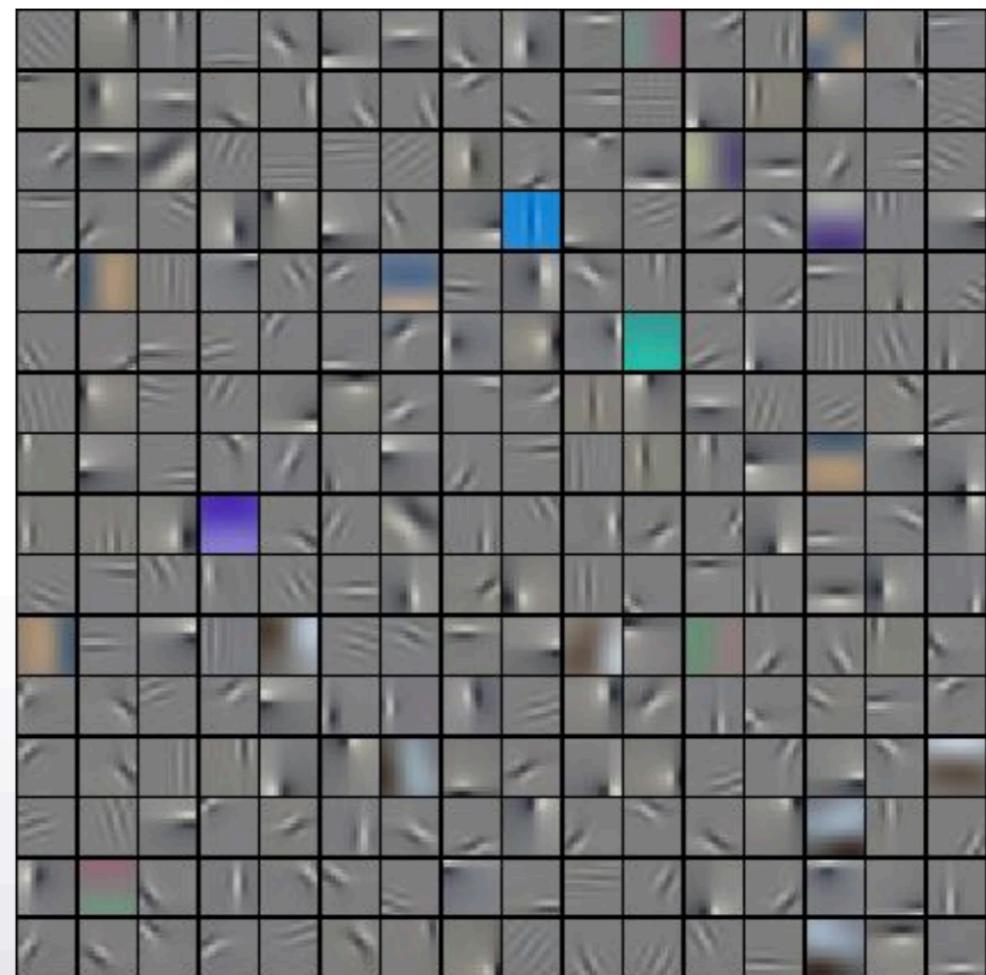


# Parameterised image priors

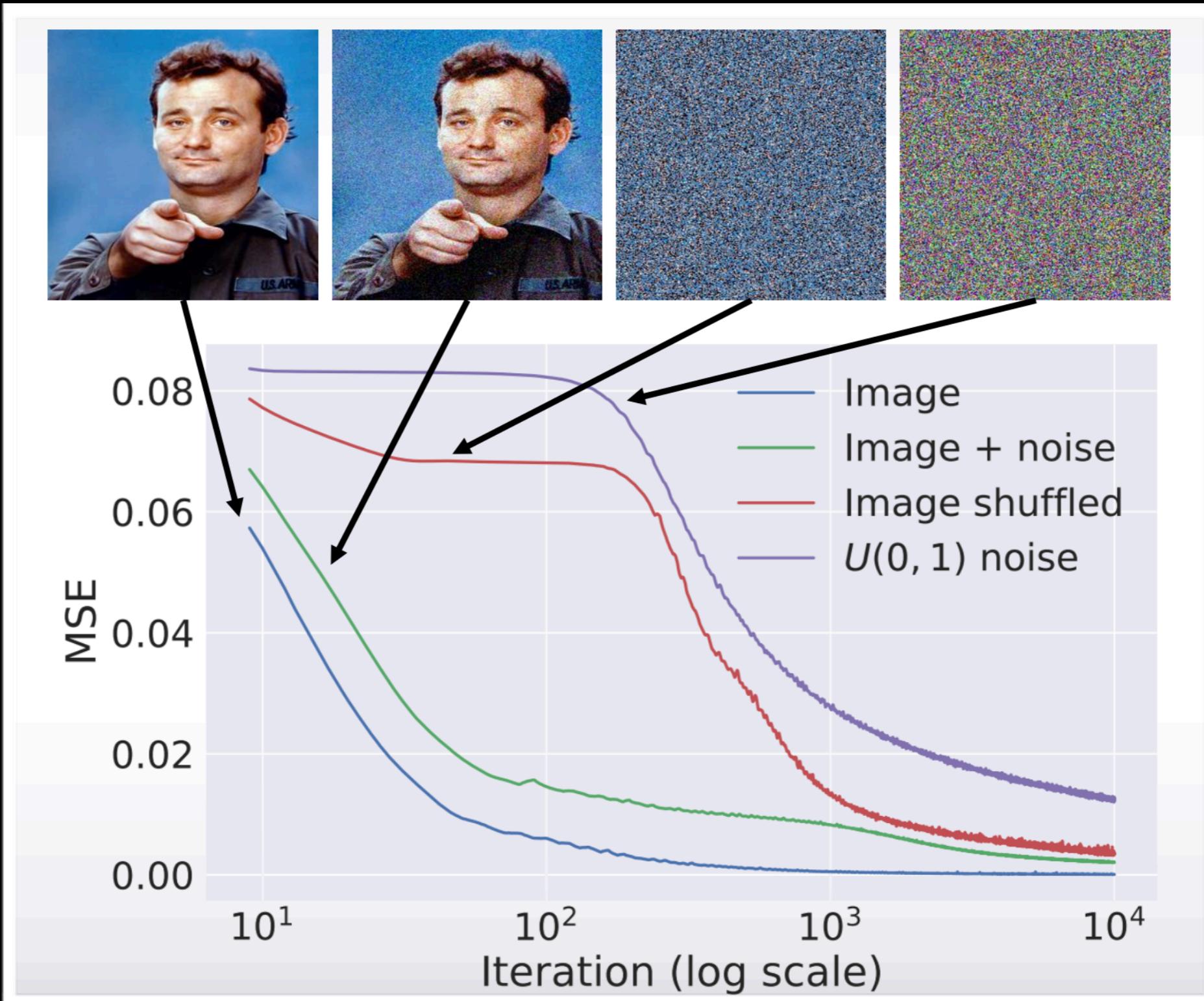
**Parametrized:**  $\arg \min_{\theta} E(g(\theta); \hat{x}) + R(g(\theta))$

$$g(\theta) = D\theta$$

**Coefficients**   
**Dictionary** 



# Deep image prior



# Deep image prior

**Parametrized:**  $\arg \min_{\theta} E(g(\theta); \hat{x}) + R(g(\theta))$

$$g(\theta) \equiv f_{\theta}(z)$$

**Fixed input**

**Convolutional network with parameters  $\theta$**

**Optimize only data term:**  $\arg \min_{\theta} E(f_{\theta}(z); \hat{x})$

# Deep image prior

➤  $\hat{x}$  – Corrupted image (observed)

## 1. Initialize $z$

- For example fill it with uniform noise  $U(-1, 1)$

## 2. Solve

$$\arg \min_{\theta} E(f_{\theta}(z); \hat{x}))$$

- With your favorite gradient-based method

$$\theta^{k+1} = \theta^k - \alpha \frac{\partial E(f_{\theta}(z); \hat{x})}{\partial \theta}$$

## 3. Get the solution

$$x^* = f_{\theta^*}(z)$$

# Deep image prior

- $\mathbf{x}$  – Clean image
- $\hat{\mathbf{x}}$  – Corrupted image (observed)
- $m$  – Binary mask

**Objective:**  $\arg \min_{\theta} E(f_{\theta}(z); \hat{\mathbf{x}}))$

- **Denoising:**  $E(\mathbf{x}; \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$  Needs early stopping!
- **Inpainting:**  $E(\mathbf{x}; \hat{\mathbf{x}}) = \|(\mathbf{x} - \hat{\mathbf{x}}) \cdot m\|^2$
- **Super-Res.:**  $E(\mathbf{x}; \hat{\mathbf{x}}) = \|d(\mathbf{x}) - \hat{\mathbf{x}}\|^2$

# Deep image prior

