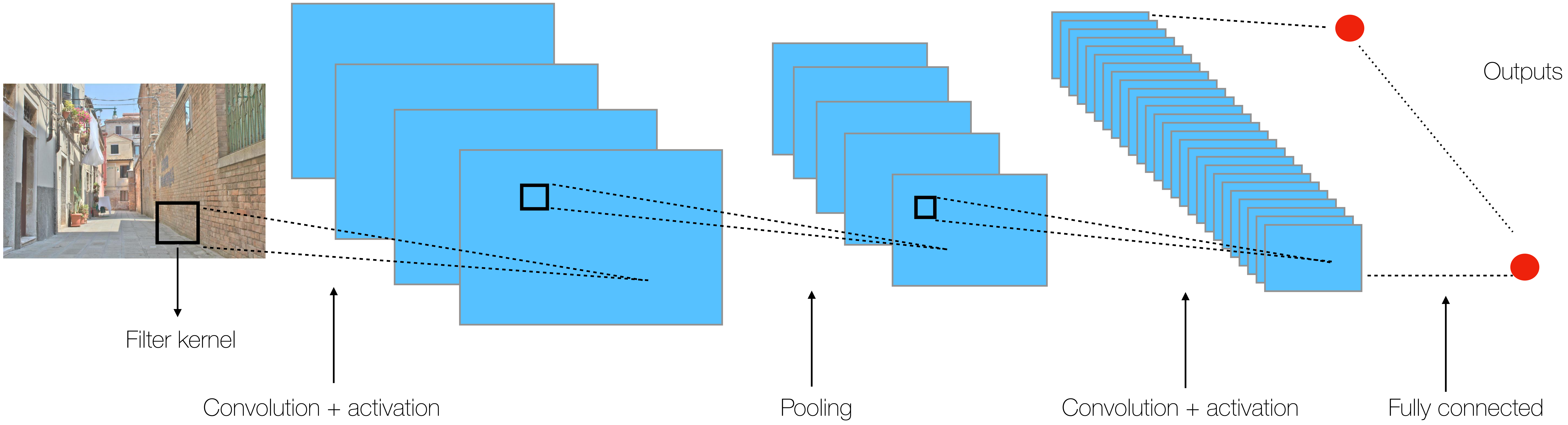
Modern High Dynamic Range Imaging at the Time of Deep Learning

Main Deep Learning Architectures

Francesco Banterle and Alessandro Artusi

Convolutional Neural Networks



Convolutional layer - Principle

2D input image

10	35	145	45	255
4	110	2	3	120
90	9	25	15	115
75	125	6	1	2
38	3	65	15	210

$f(x, y)$

$*$

3x3 kernel filter

1	0	1
-1	0	-1
0	1	1

$g(i, j)$

$=$

2D convolved image

		96		

$\sum f(x-i, y-j)g(i,j)$

Stride - shift of the kernel on the input image

Convolutional layer - Padding

2D input image

0	0	0	0	0	0	0
0	10	35	145	45	255	0
0	4	110	2	3	120	0
0	90	9	25	15	115	0
0	75	125	6	1	2	0
0	38	3	65	15	210	0
0	0	0	0	0	0	0

$f(x, y)$

$*$

3x3 kernel filter

1	0	1
-1	0	-1
0	1	1

$g(i, j)$

$=$

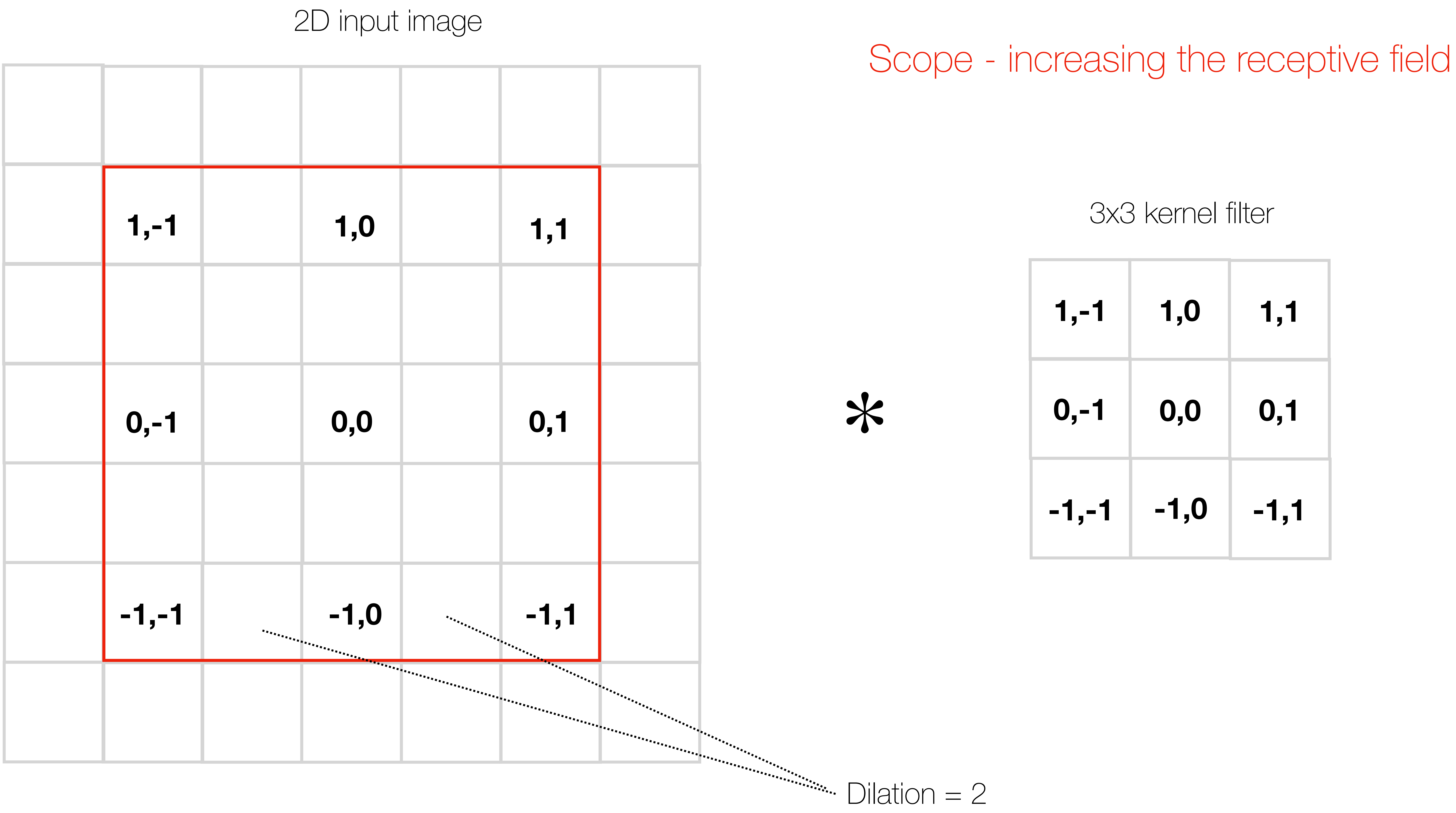
2D convolved image

		96		

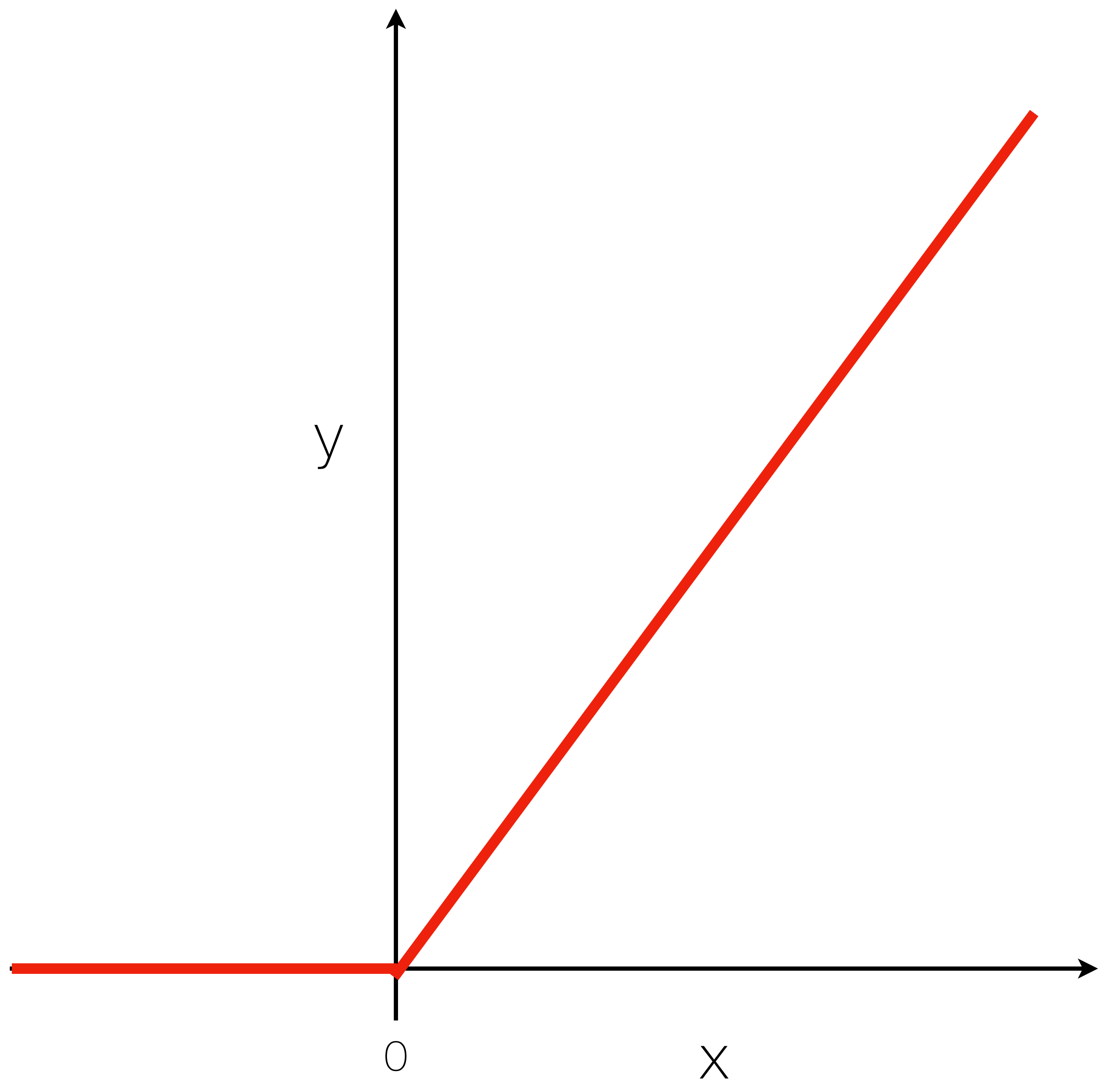
$$\sum f(x - i, y - j) g(i, j)$$

Stride - shift of the kernel on the input image

Convolutional layer - Dilatation (x,y)



Activation Functions (layers) categories - most used

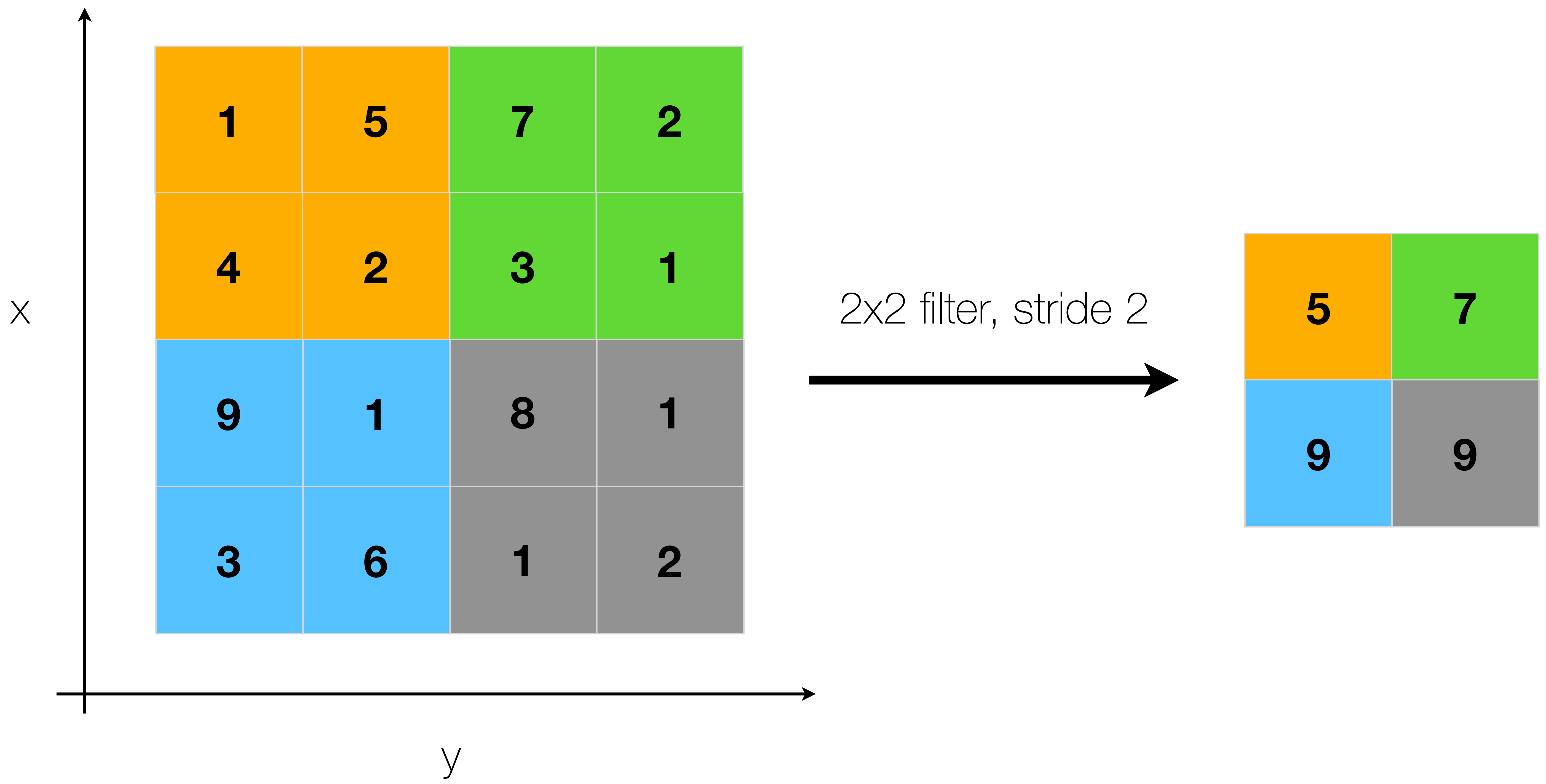


1. Ridge activation functions
   1.1 linear
   1.2 ReLU
   1.3 logistic

2. Radial activation functions
   2.1 gaussian
   2.2 multi quadratics
   2.3 polynomials

$$ReLU(x) = \max(0, a + x'b)$$

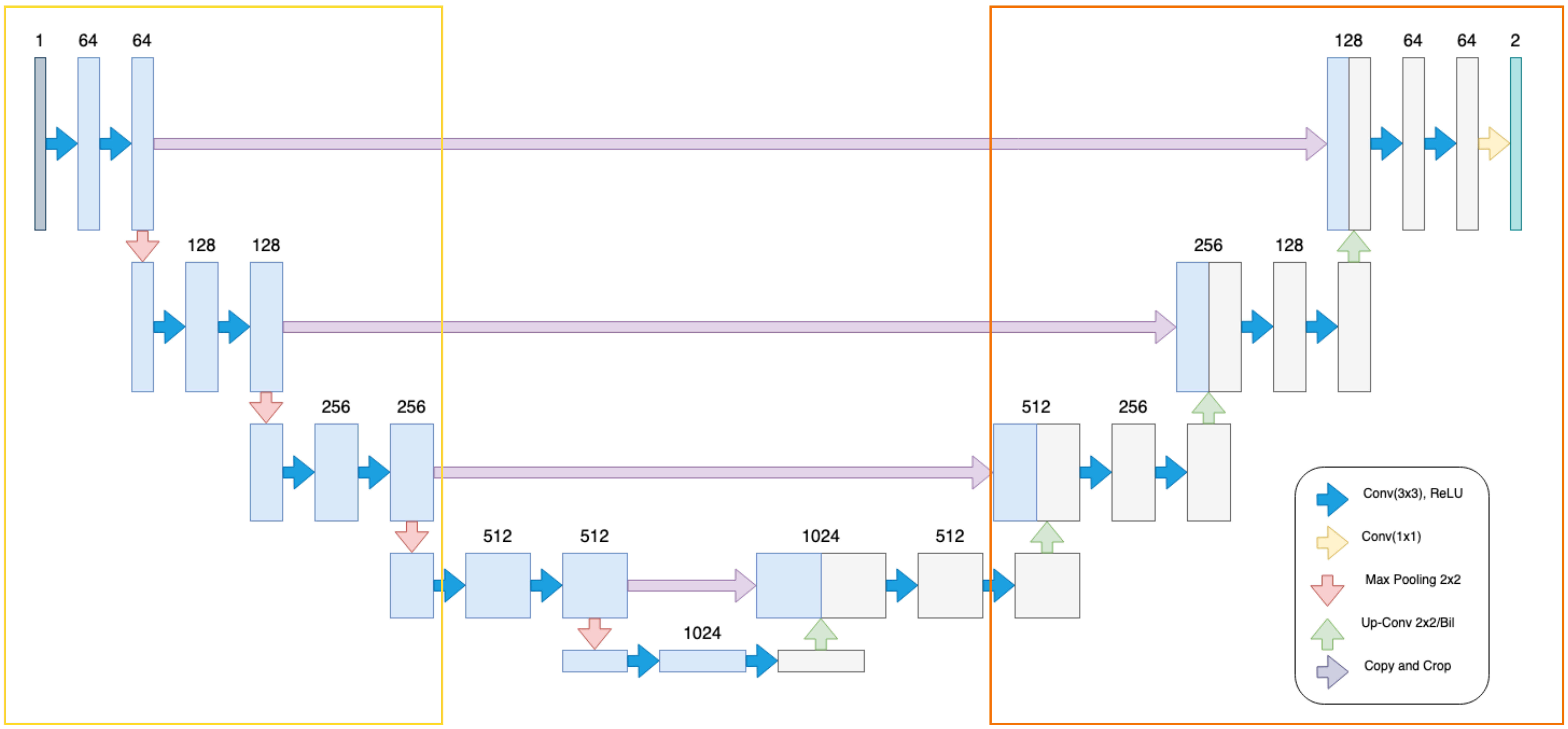
Pooling - downsampling (e.g., max function)



- controlling overfitting
- reducing the number of parameters

- memory footprint
- reducing the number of computations

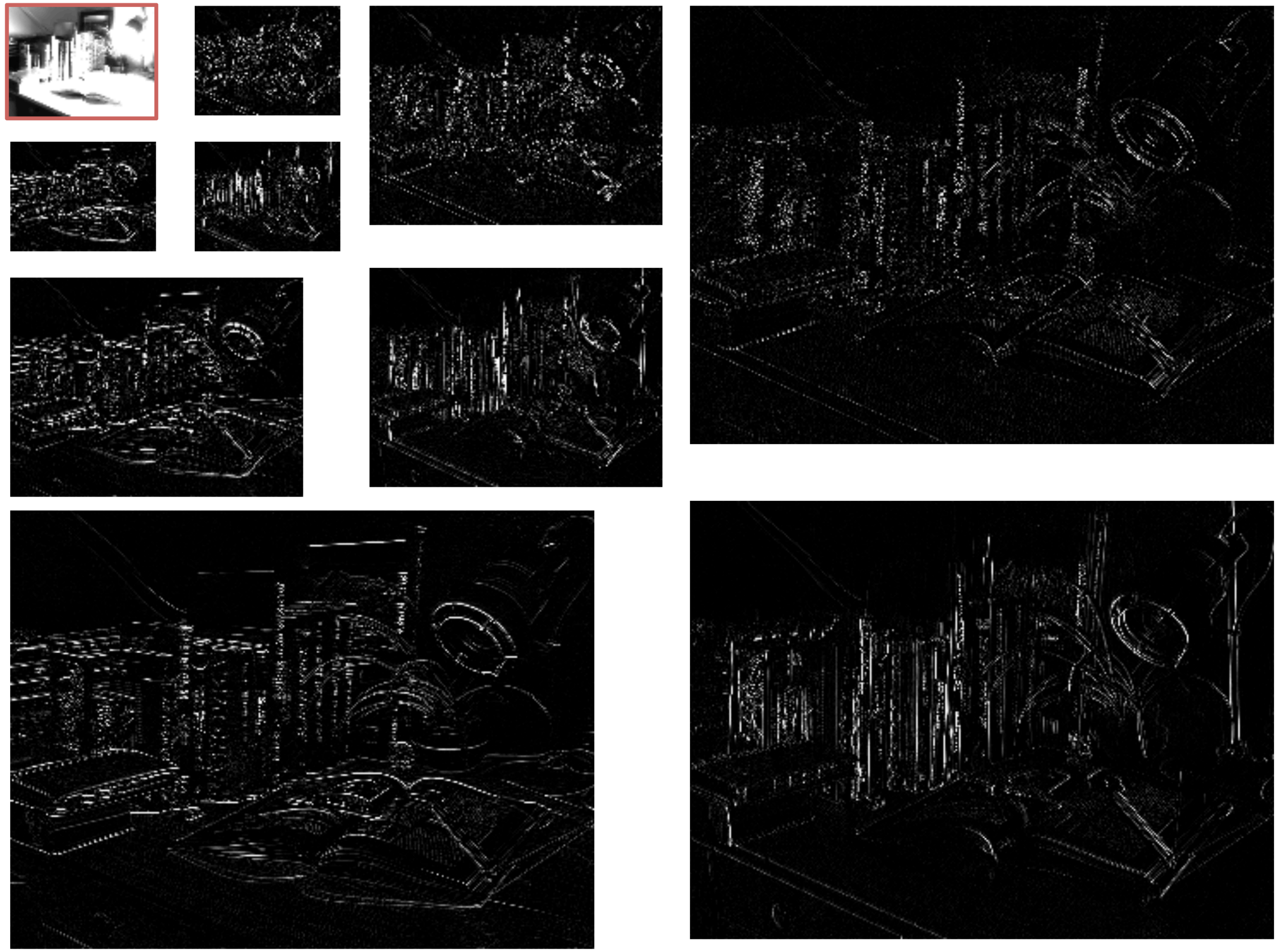
The U-Net



Encoder/Contraction

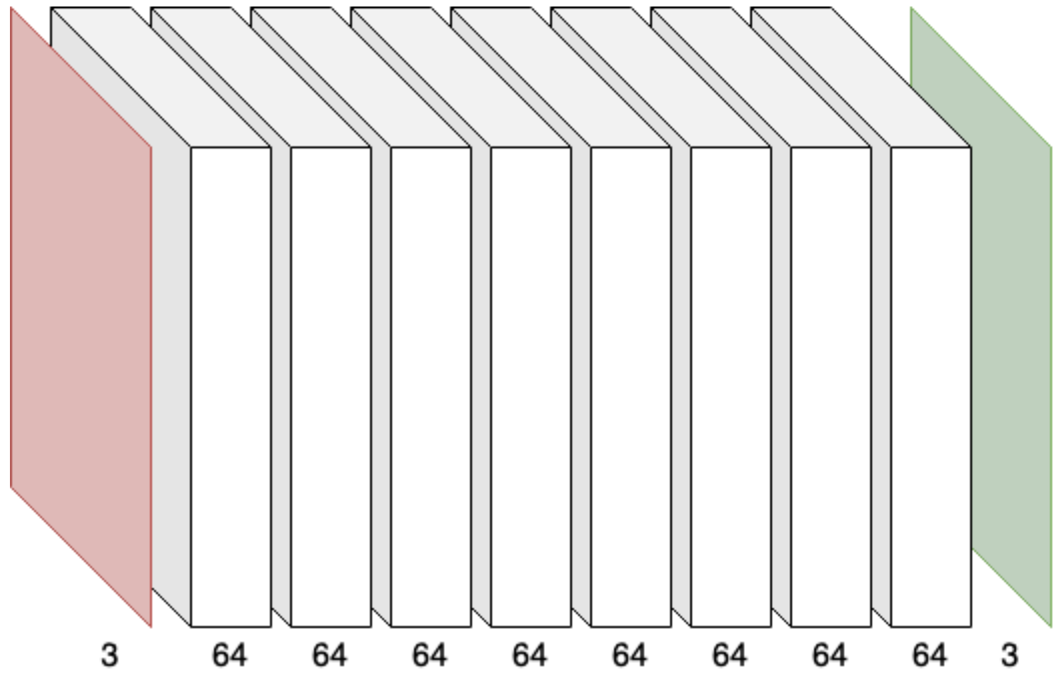
Decoder/Expansion

The U-Net - Multi-scale concept in Image Processing

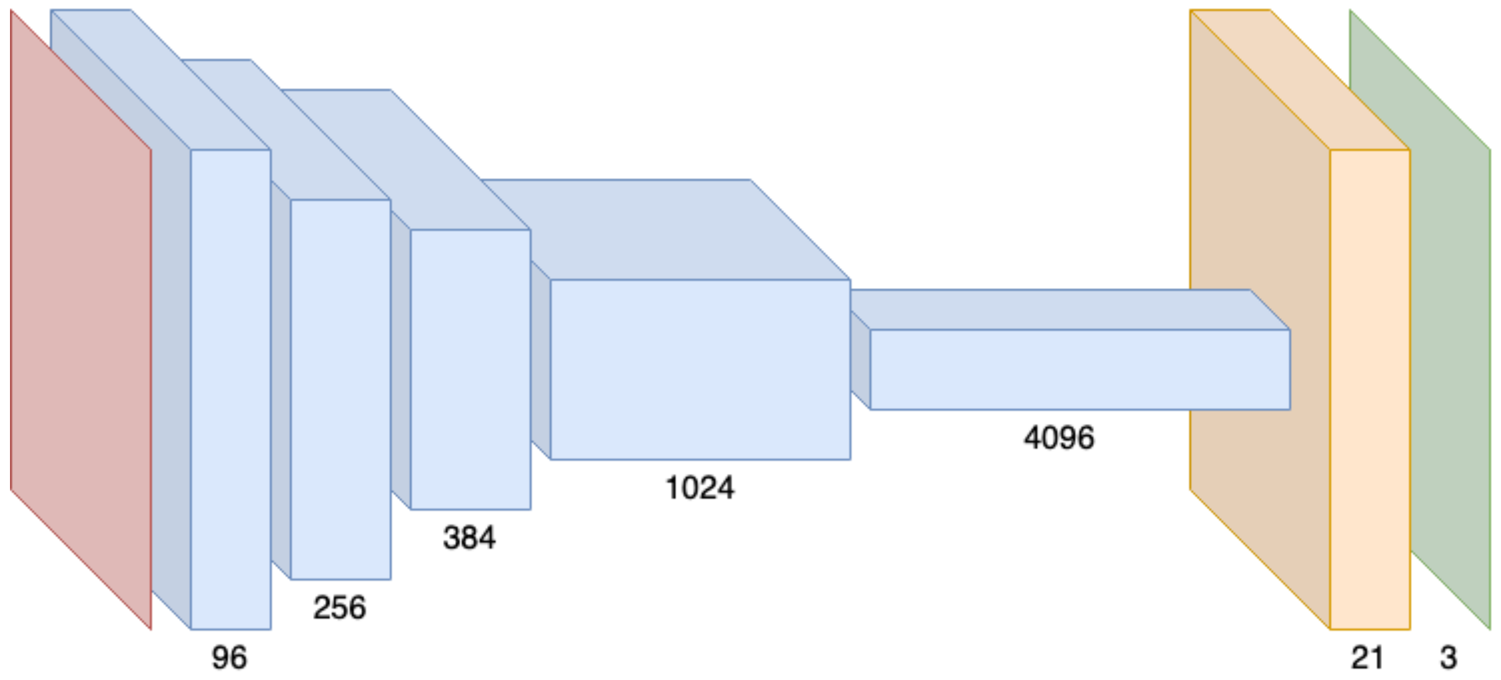


- How Human Visual system works:
  - Distance places a role in the perceived details of the image
    - Far away fine details are not visible
    - Closer we are, we are able to perceive fine details in objects.

Fully Convolutional Neural Networks



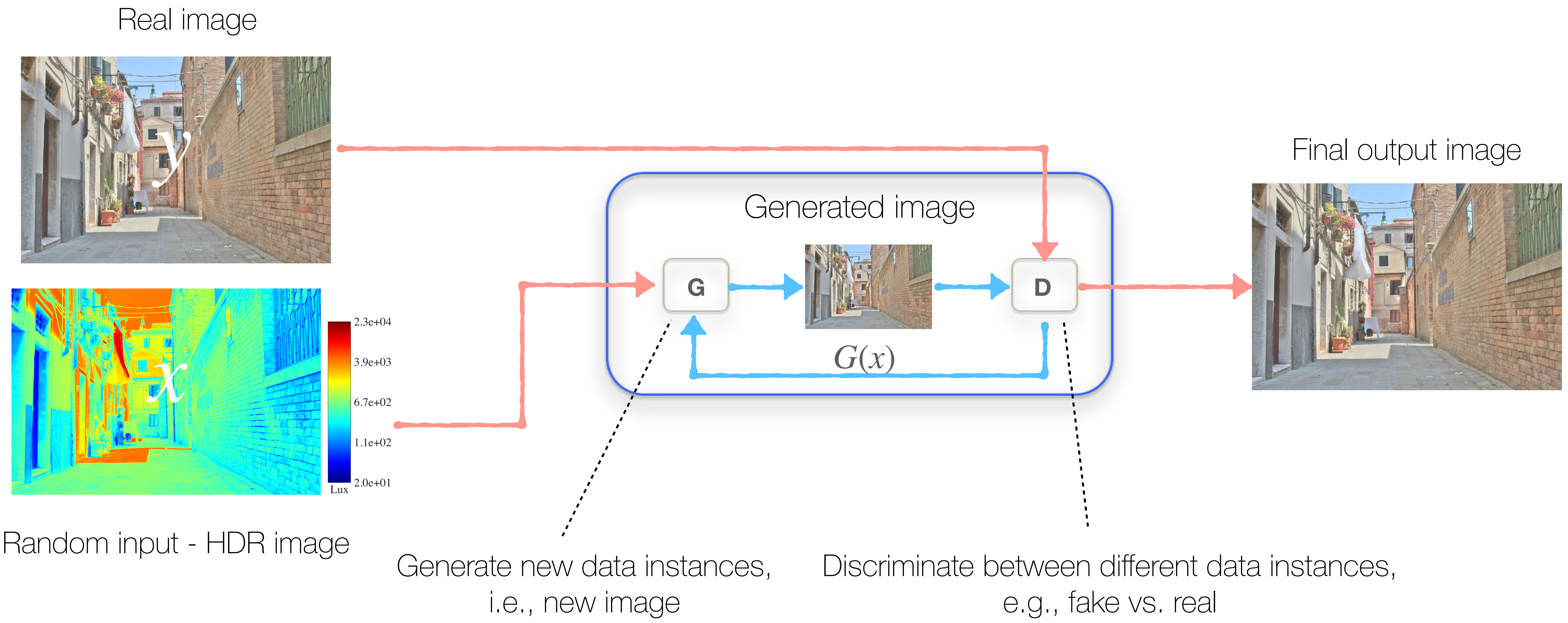
no-Contraction and no-Expansion



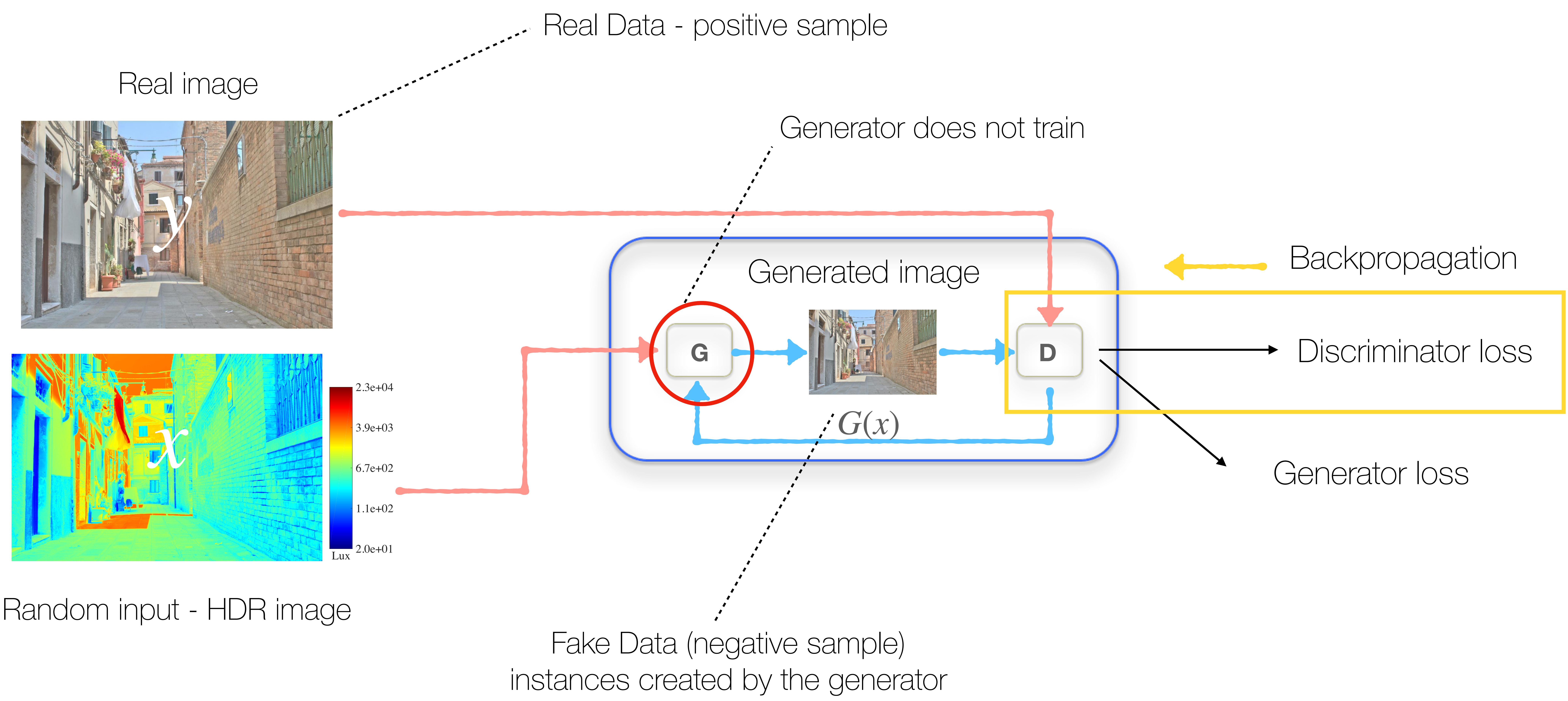
Contraction and Expansion
but different from U-net

- FCN what is it?
  - Architecture with only convolutional layers
  - No dense layers
  - U-net is an example
  - But others architectures are possible

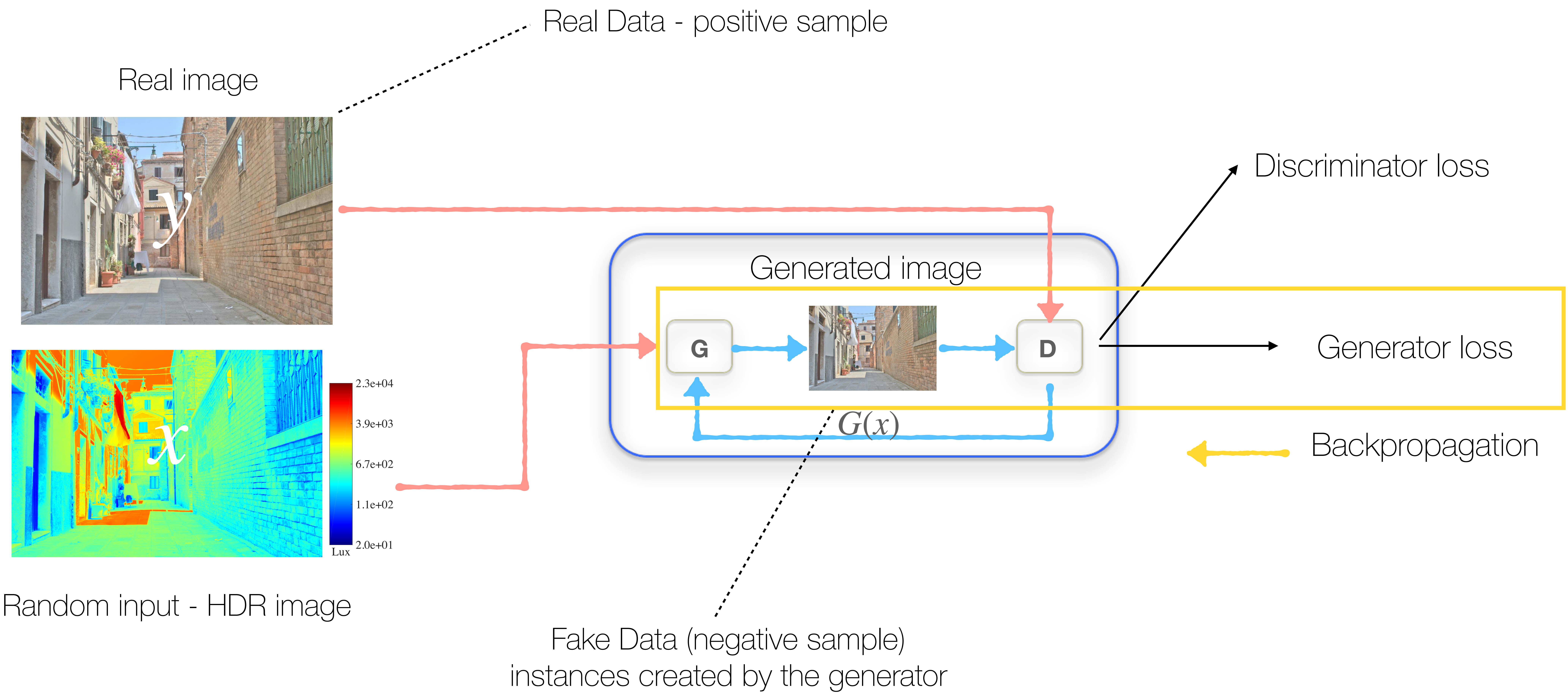
Generative Adversarial Networks (GANs)



GANs: Backpropagation in Discriminator



GANs: Backpropagation in Generator



GANs: Loss Function - e.g., Minimax loss

$$L_{GAN}(G, D) = \mathbb{E}_y[logD(y)] + \mathbb{E}_x[1 - logD(G(x))]$$

Discriminator loss Generator loss

$D(y)$ = discriminator estimated probability that the real data instance y is real

$\mathbb{E}_y$ = expected value over all the real y instances

$G(x)$ = generator instance output value when given random input/input image x

$D(G(x))$ = discriminator estimated probability that a fake instance is real

$\mathbb{E}_x$ = expected value over all fake generated instances