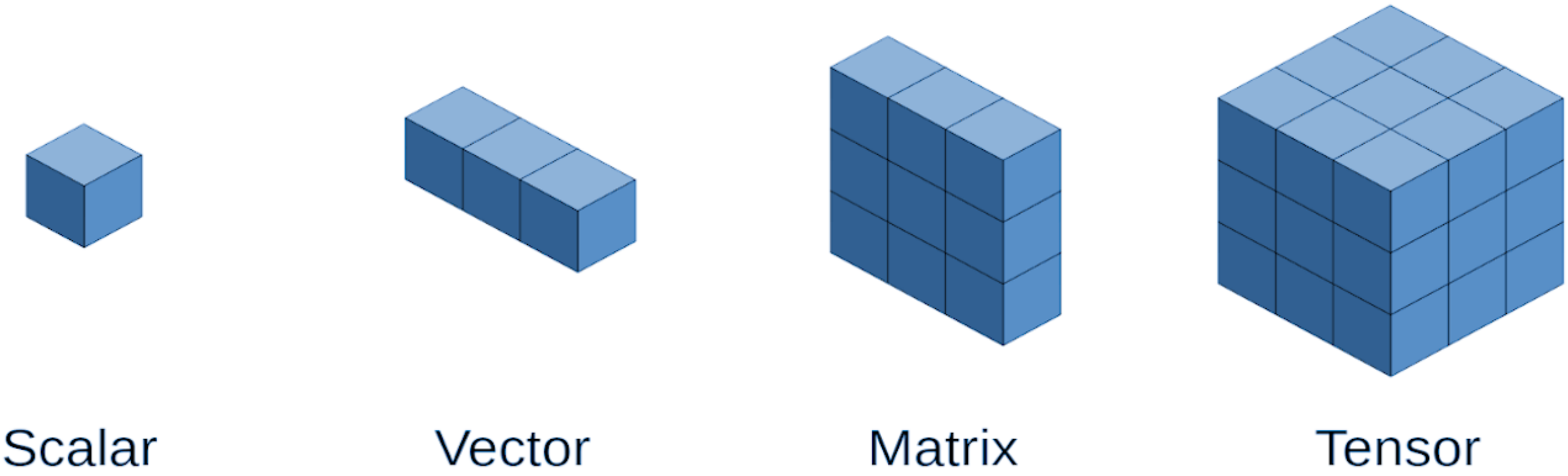
Strojové učení II

Convolutional NN

Institute of Information Theory
and Automation of the AS CR

PyTorch Tensors

- N-D array



- PyTorch convention: N x C x H x W
  - N … number of images (mini-batch size)
  - C … number of channels (or filters) <== FEATURES
  - H … height
  - W … width

2D Convolution

$$[u * h](x, y) = \int \int u(s,t) h(x-s, y-t) ds dt$$

Single Channel Image

5	0	8	7	8	1
1	9	5	0	7	7
6	0	2	4	6	6
9	7	6	6	8	4
8	3	8	5	1	3
7	2	7	0	1	0

1 x 1 x 6 x 6

*

Filter

0	0	0
0	1	0
0	0	0

1 x 1 x 3 x 3

Region

5	0	8
1	9	5
6	0	2

x

Filter

0	0	0
0	1	0
0	0	0

=

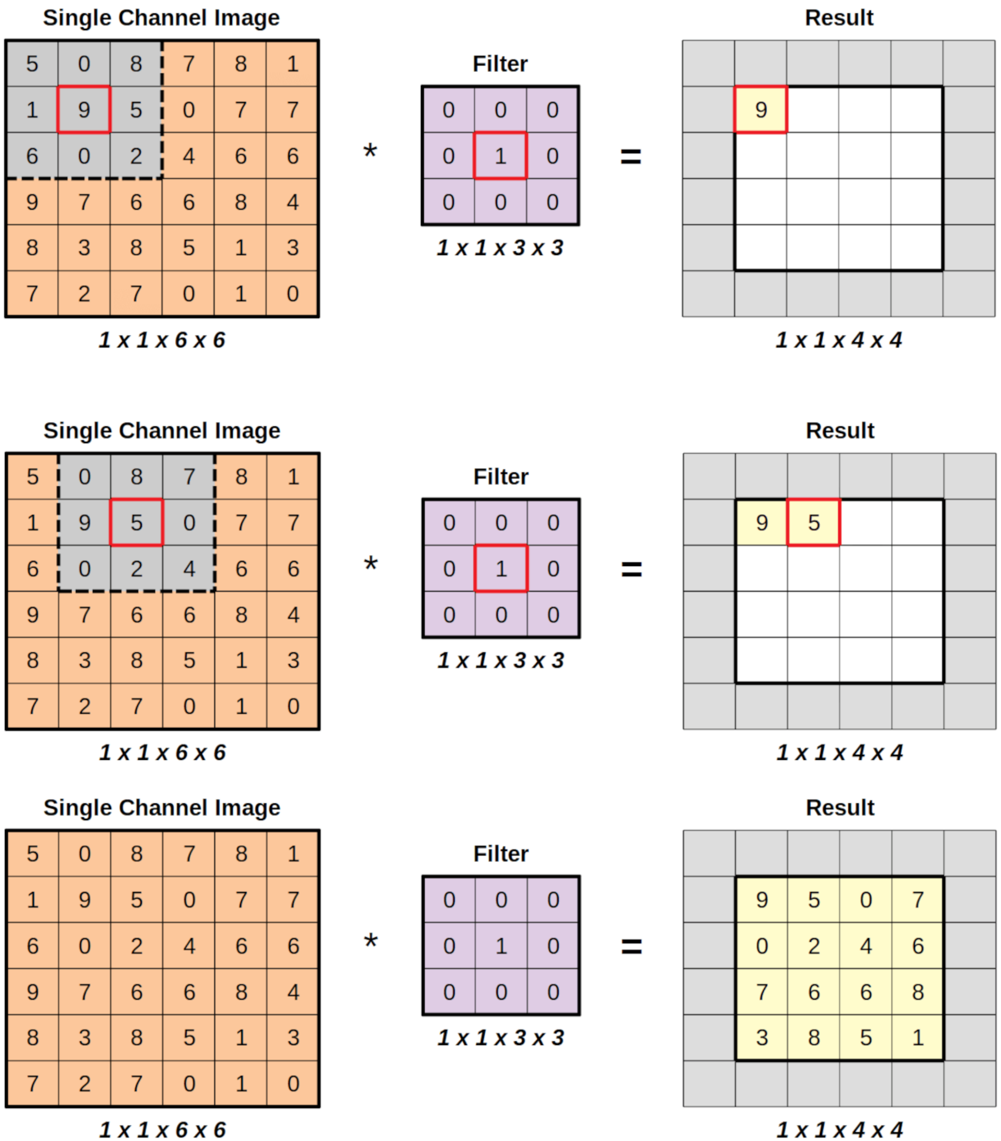
Multiplication

0	0	0
0	9	0
0	0	0

adds up to **9**

Implemented in frameworks as correlation.

2D Convolution



Padding

- Different output sizes – valid, same, full

Replication Padding

5	5	0	8	7	8	1	1
5	**5**	**0**	**8**	**7**	**8**	**1**	1
1	**1**	**9**	**5**	**0**	**7**	**7**	7
6	**6**	**0**	**2**	**4**	**6**	**6**	6
9	**9**	**7**	**6**	**6**	**8**	**4**	4
8	**8**	**3**	**8**	**5**	**1**	**3**	3
7	**7**	**2**	**7**	**0**	**1**	**0**	0
7	7	2	7	0	1	0	0

Reflection Padding

9	1	9	5	0	7	7	7
0	**5**	**0**	**8**	**7**	**8**	**1**	8
9	**1**	**9**	**5**	**0**	**7**	**7**	7
0	**6**	**0**	**2**	**4**	**6**	**6**	6
7	**9**	**7**	**6**	**6**	**8**	**4**	8
3	**8**	**3**	**8**	**5**	**1**	**3**	1
2	**7**	**2**	**7**	**0**	**1**	**0**	1
3	8	3	8	5	1	3	1

Circular Padding

0	7	2	7	0	1	0	7
1	**5**	**0**	**8**	**7**	**8**	**1**	5
7	**1**	**9**	**5**	**0**	**7**	**7**	1
6	**6**	**0**	**2**	**4**	**6**	**6**	6
4	**9**	**7**	**6**	**6**	**8**	**4**	9
3	**8**	**3**	**8**	**5**	**1**	**3**	8
0	**7**	**2**	**7**	**0**	**1**	**0**	7
1	5	0	8	7	8	1	5

Striding

Single Channel Image

5	0	8	7	8	1
1	$\boxed{9}$	5	$\boxed{0}$	7	7
6	0	2	4	6	6
9	$\boxed{7}$	6	$\boxed{6}$	8	4
8	3	8	5	1	3
7	2	7	0	1	0

1 x 1 x 6 x 6

*

Filter

0	0	0
0	1	0
0	0	0

1 x 1 x 3 x 3

=

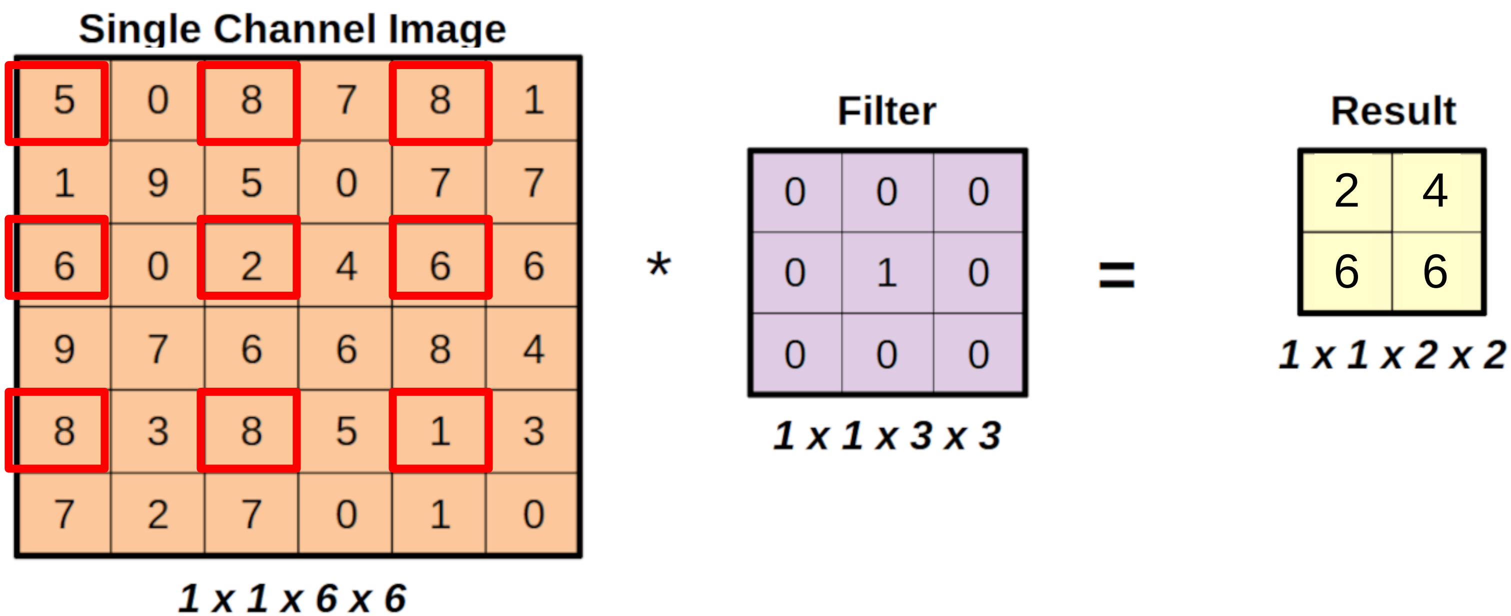
Result

9	0
7	6

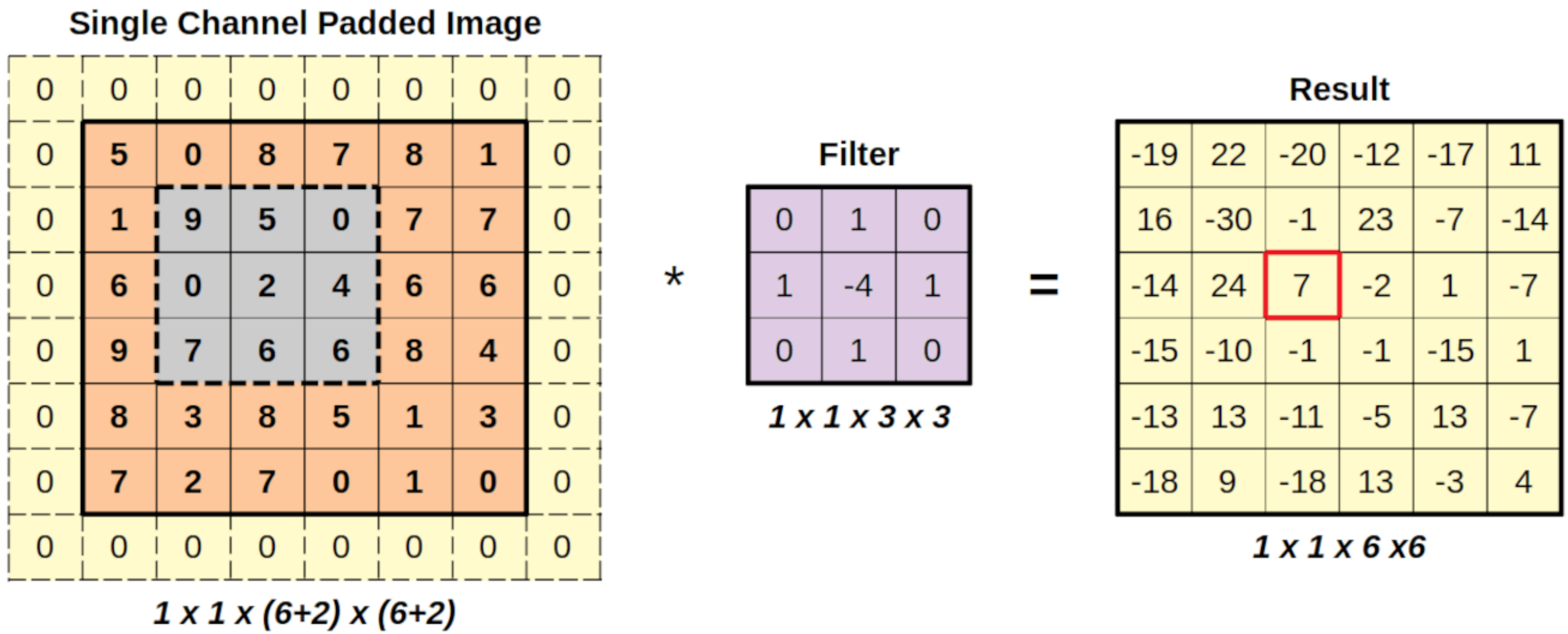
1 x 1 x 2 x 2

Dilation

- Dilation = 2



- “Same” convolution with zero padding and no striding



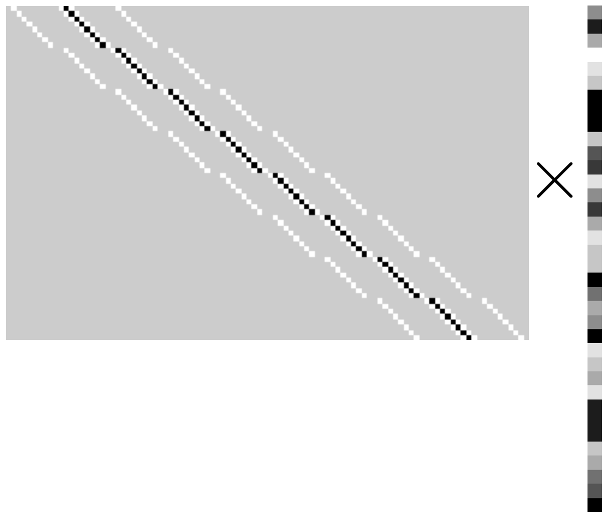
- Convolution animations

Properties of Convolution Unit

- Linear operation $\quad h * x \quad \equiv \quad Wx$

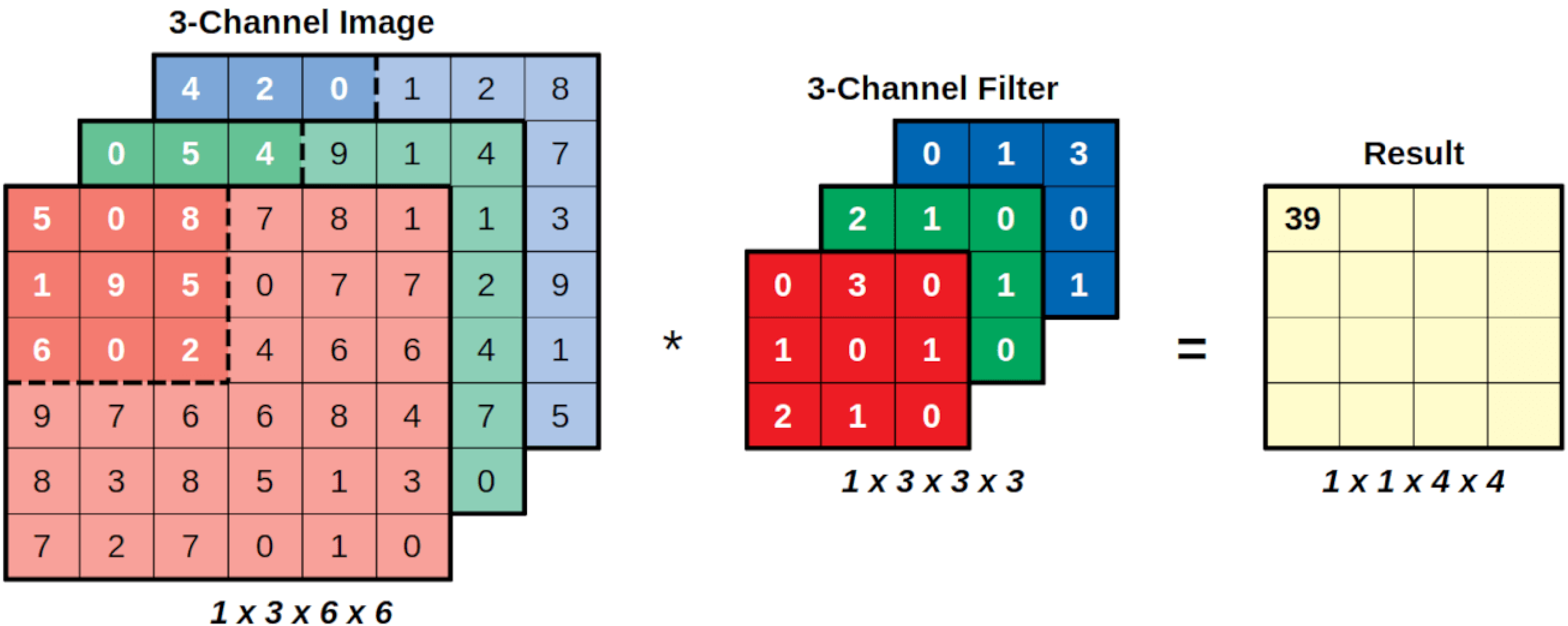
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} *$$

5	0	8	7	8	1
1	9	5	0	7	7
6	0	2	4	6	6
9	7	6	6	8	4
8	3	8	5	1	3
7	2	7	0	1	0



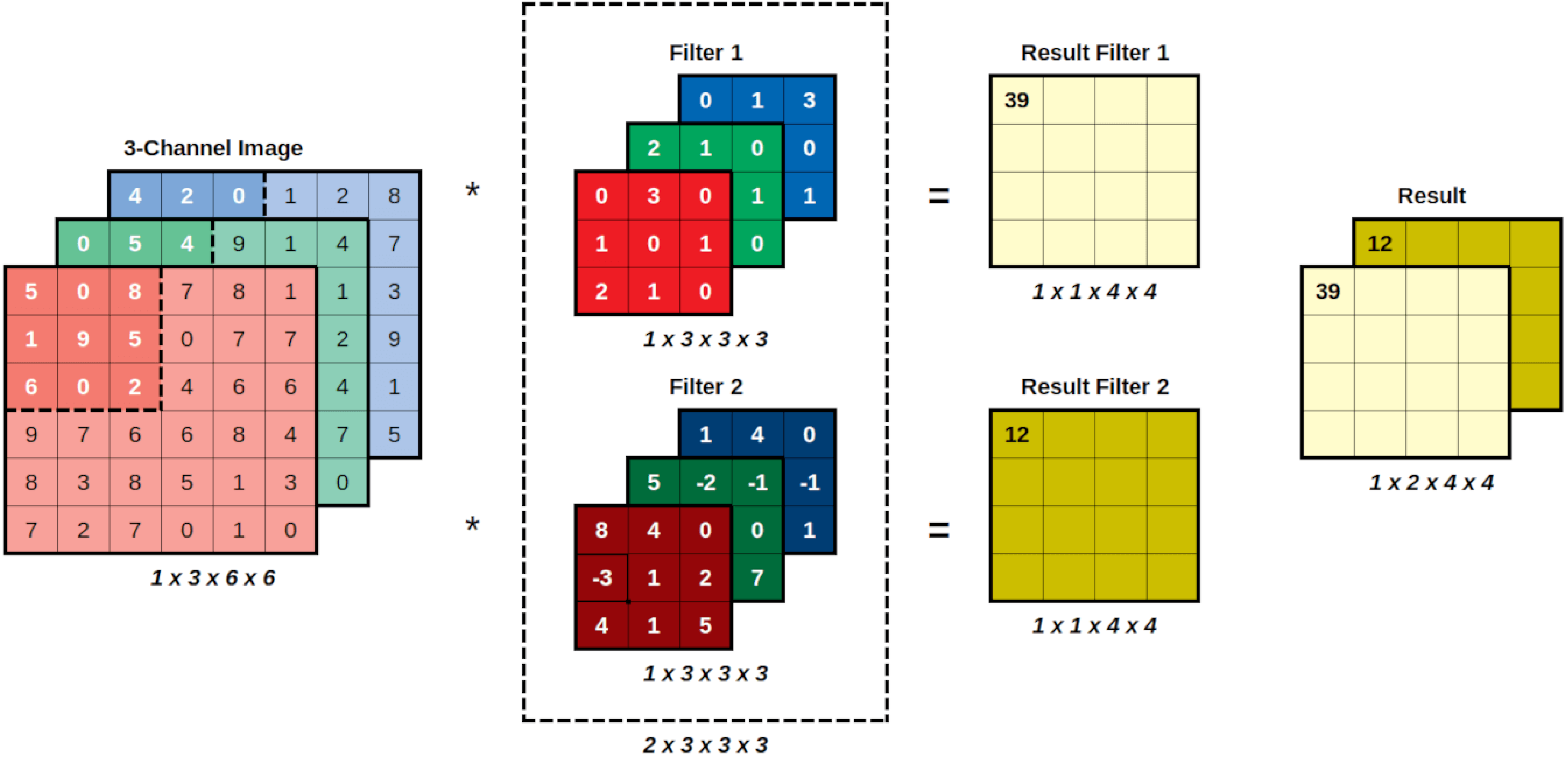
- Sparse interactions
- Parameter sharing
- Equivariance to translation

Convolution with Multiple Channels

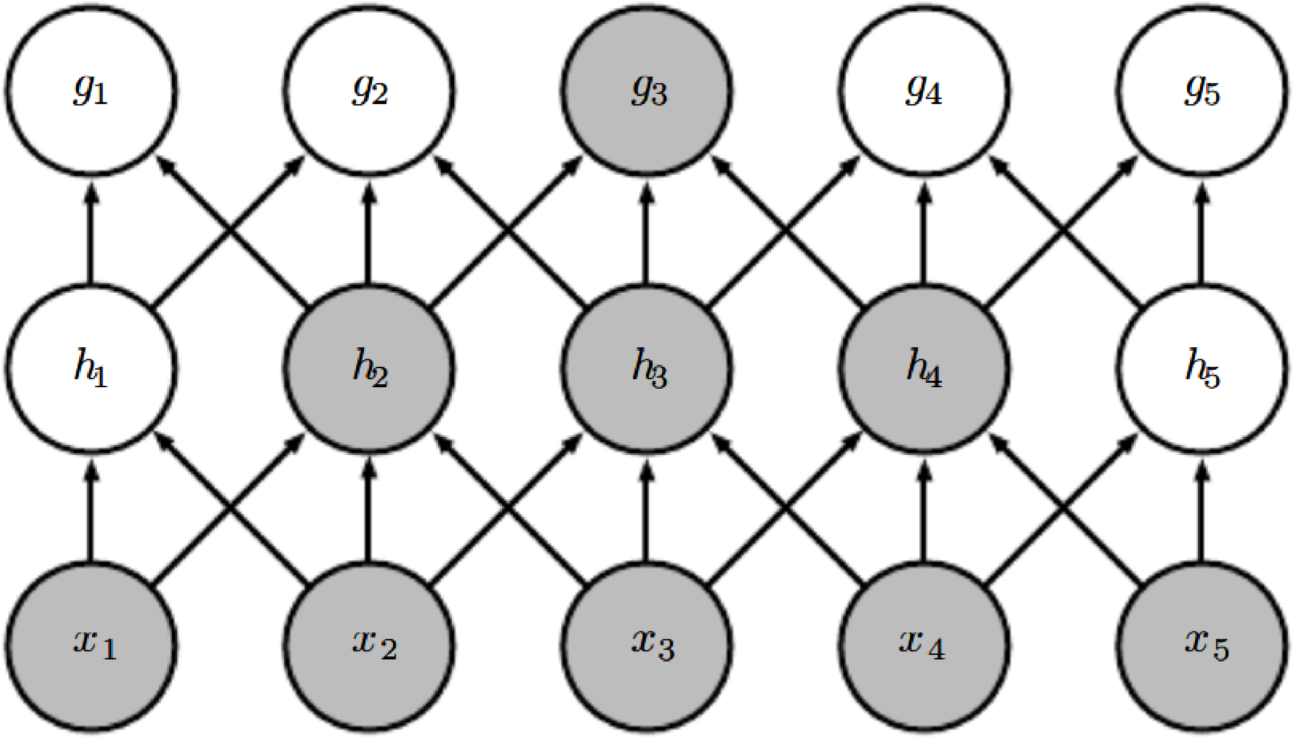


Convolution with Multiple Channels

- What if we have more filters?

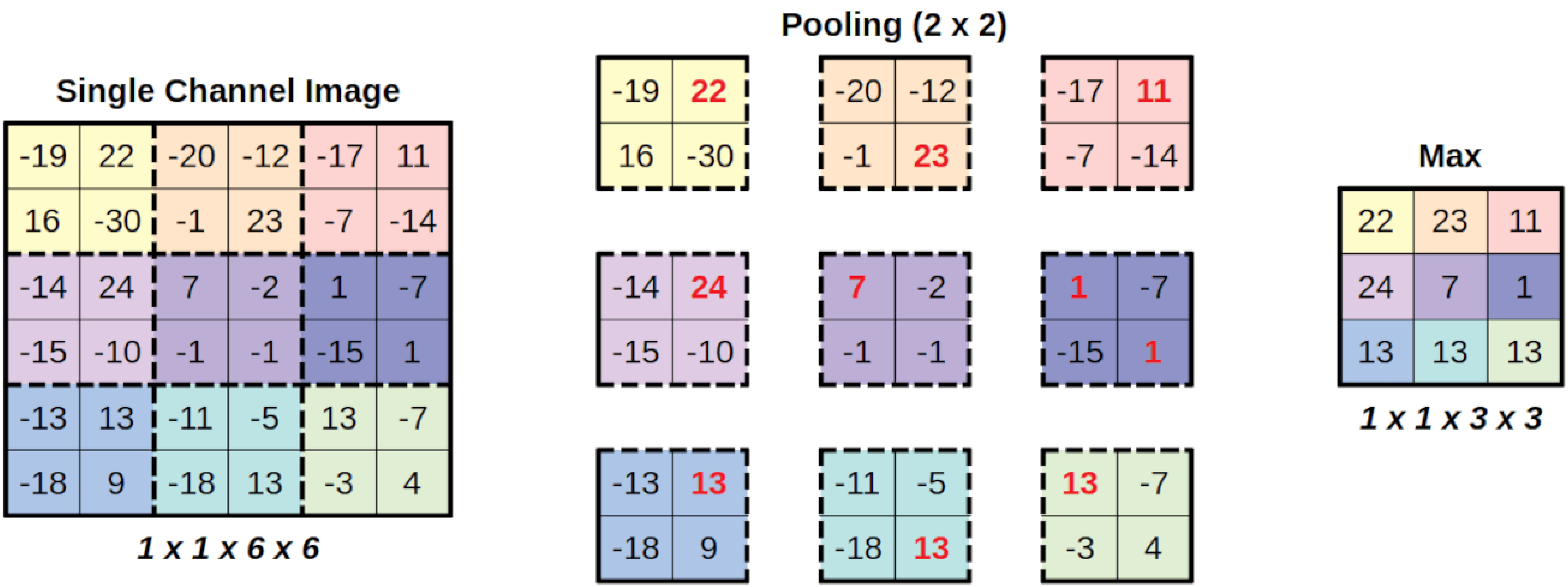


Receptive Field



Pooling

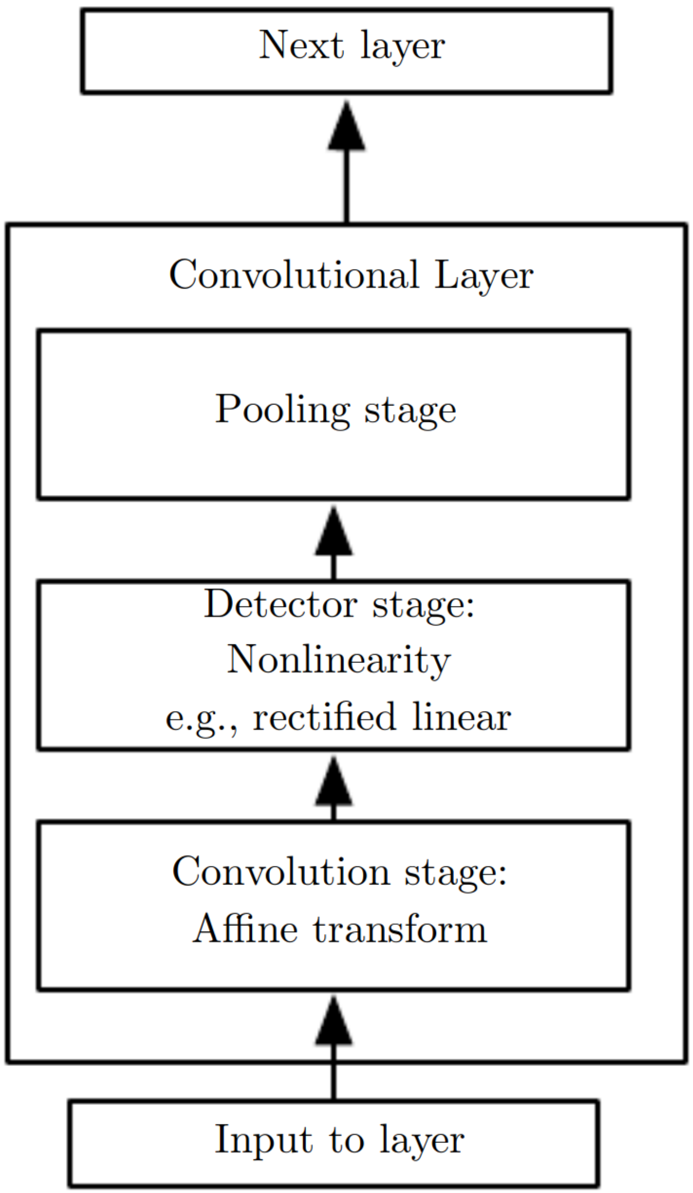
- Parameters: Kernel size, Stride, Operation (max, avg,…)
- Example with kernel size = 2x2, stride=[2,2], operation=max



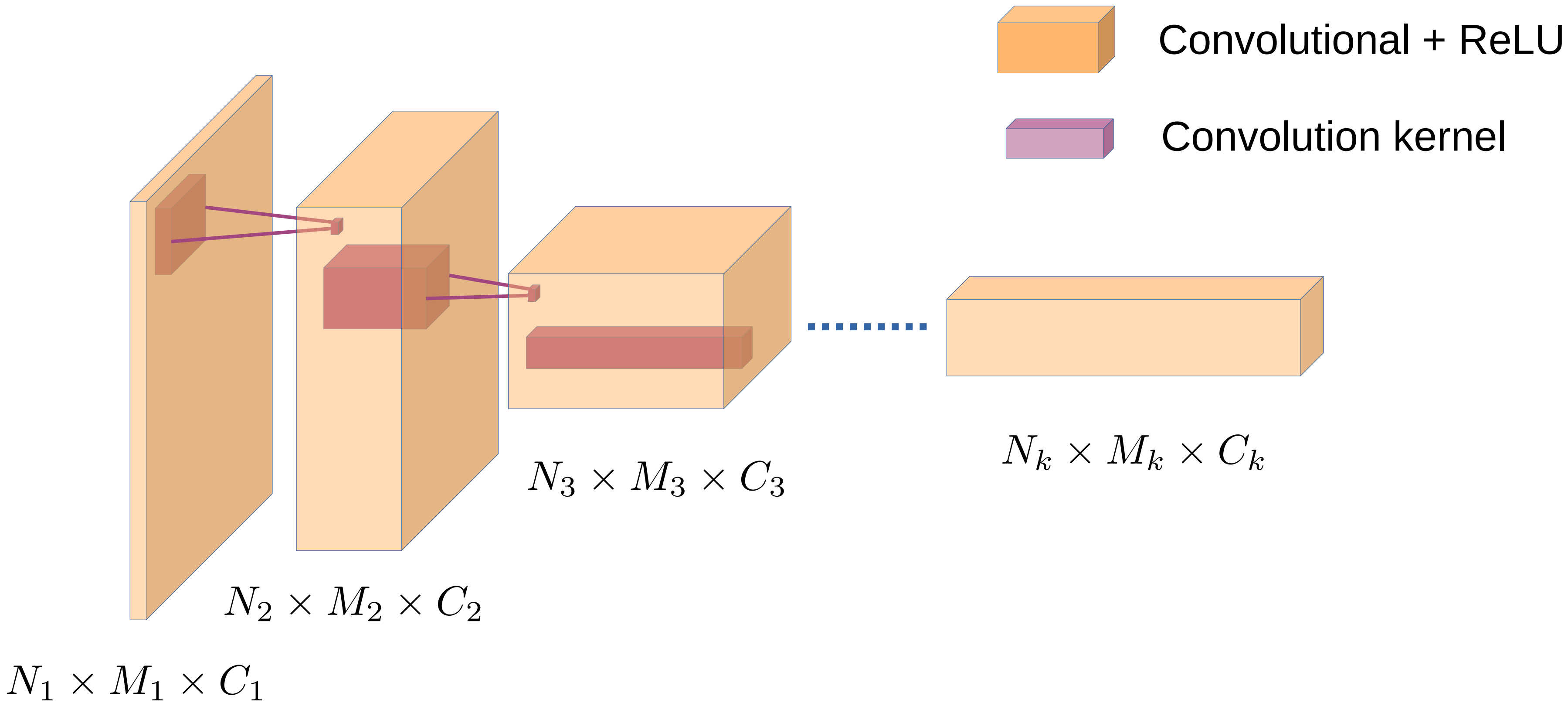
- invariant to small translation

Typical Convolutional Layer

- Infinitely strong prior:
  - Convolution: force local interactions equivariant to translation
  - Pooling: invariant to small translation



Baseline architecture

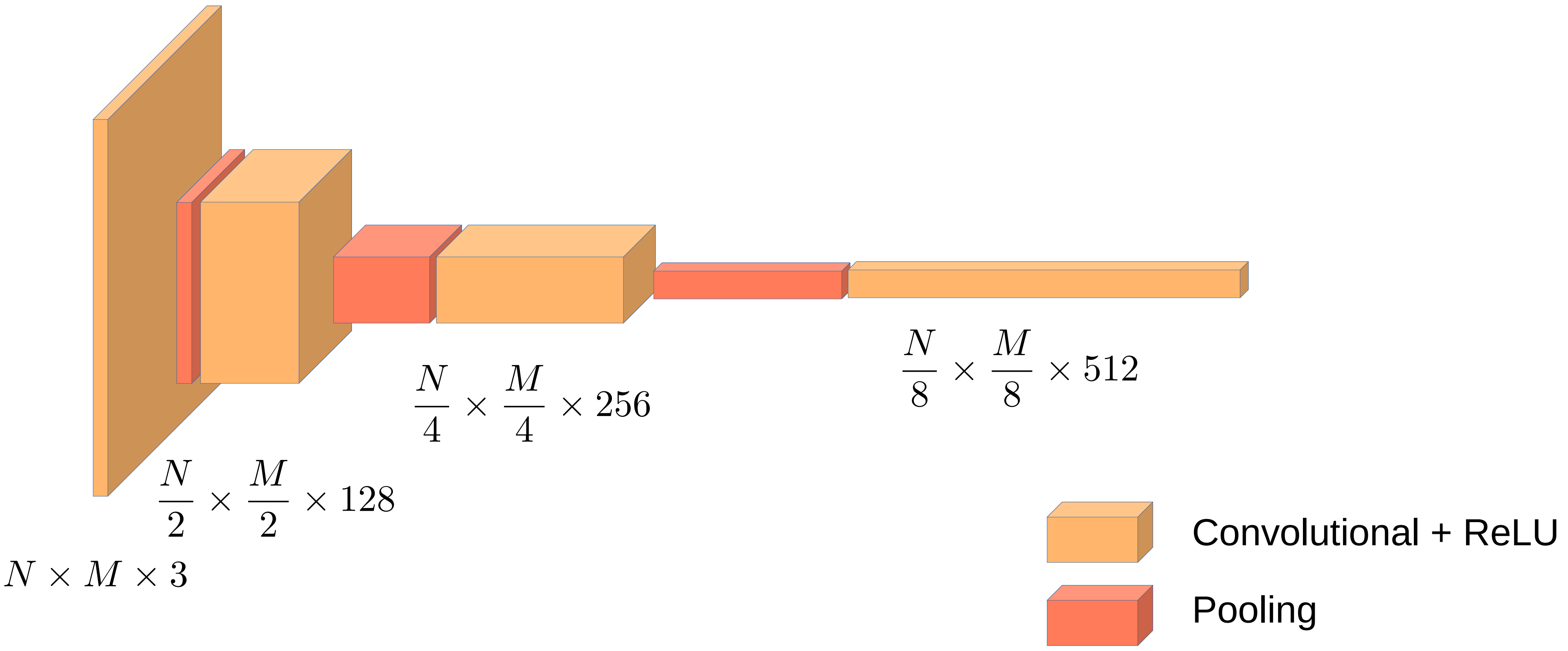


- Resolution: $N \times M$
- Width: C
- Depth: k

Scaling dimensions

Pooling

- Pooling/Stride is also very practical --> saves memory



Strojové učení II

Convolutional NN Architectures

Institute of Information Theory and Automation of the AS CR

Image Classification

airplane

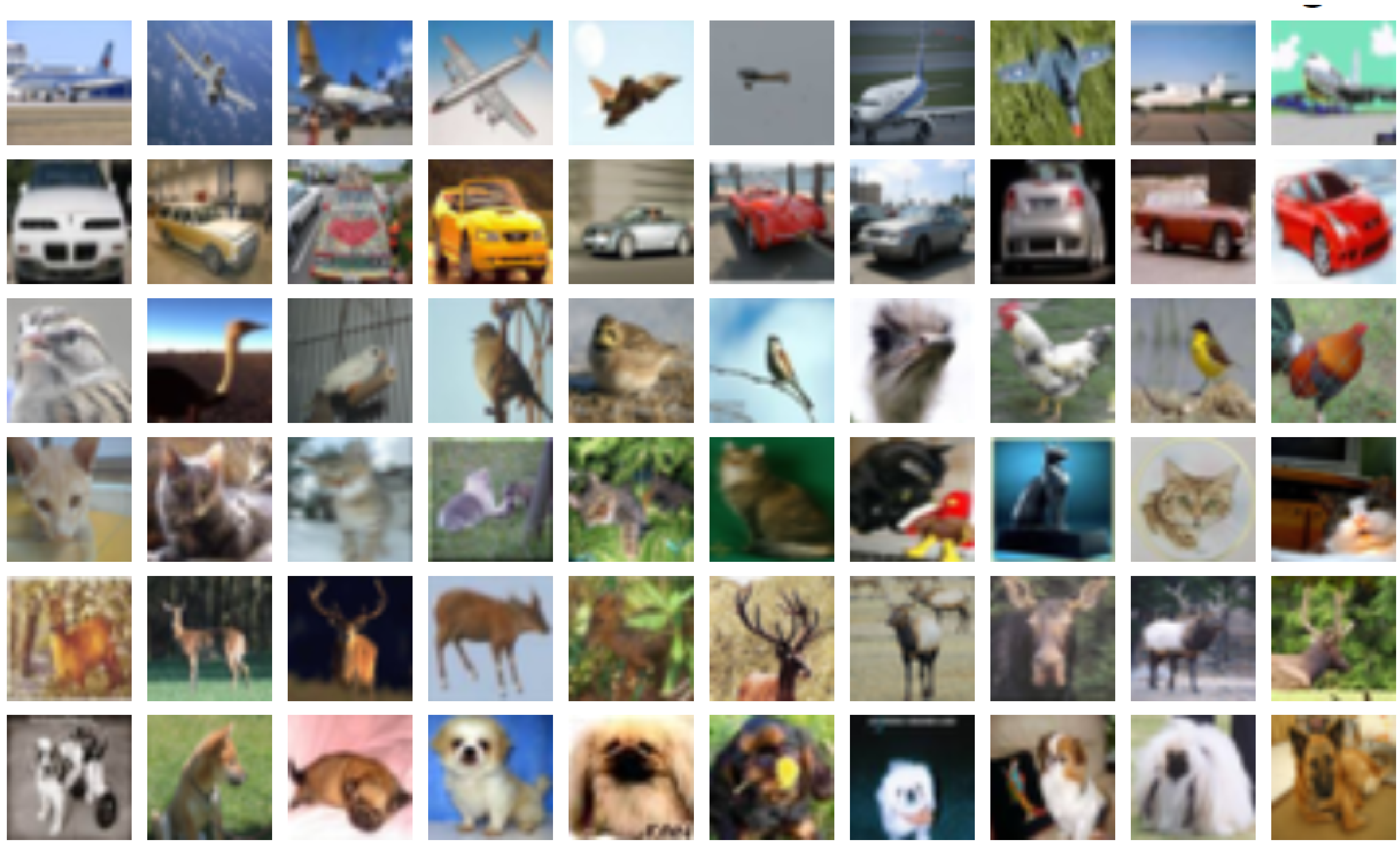
automobile

bird

cat

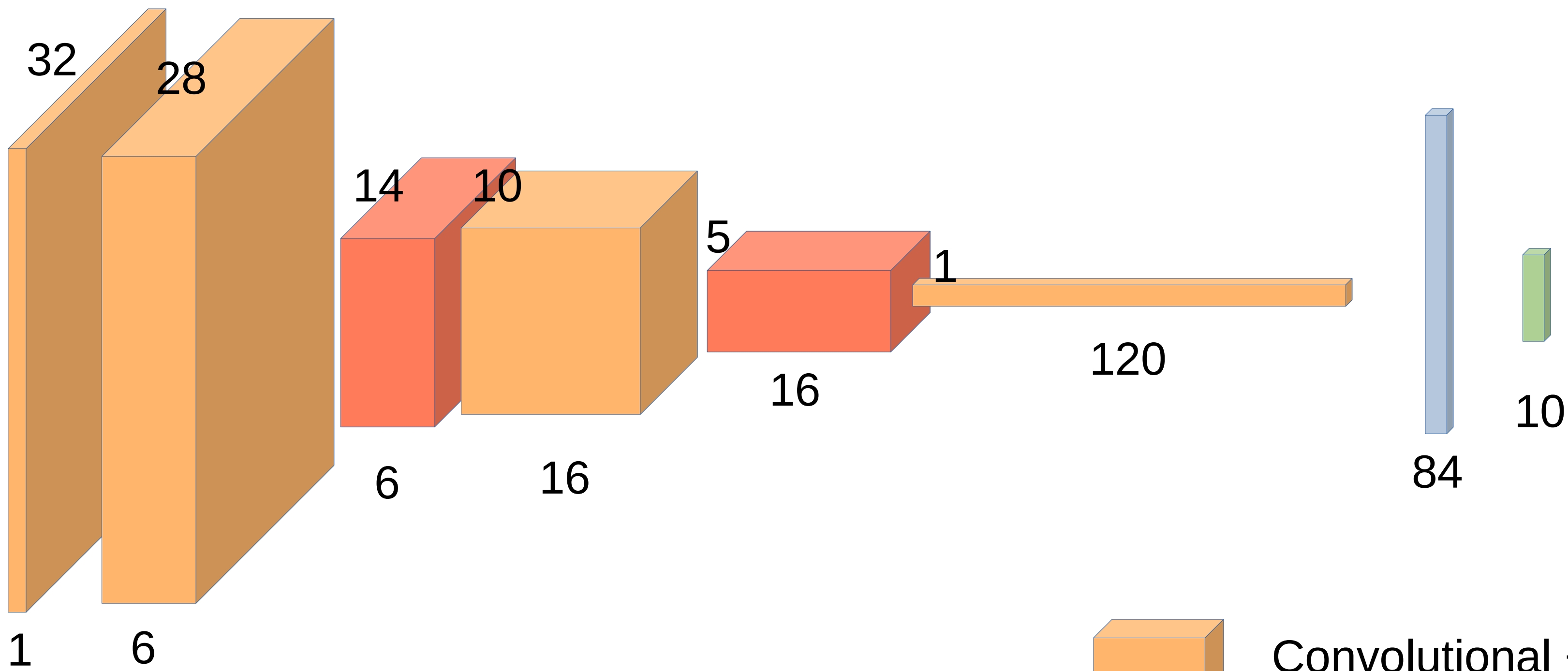
deer

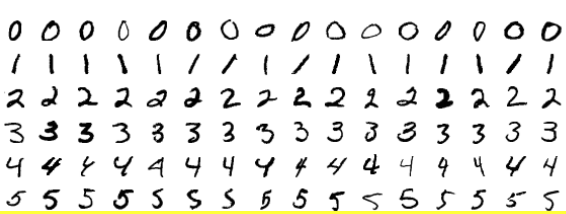
dog



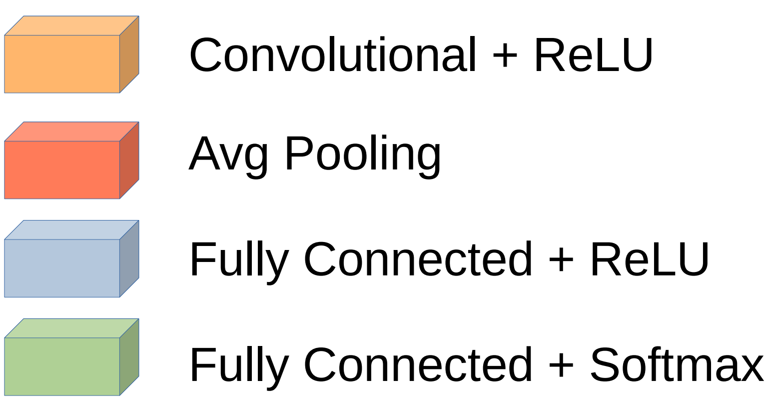
LeNet-5

- LaCun 1998
- MNIST dataset – classification of handwritten digits





What are the filter sizes?



Loss

- Multinoulli distribution
- Training set: $(x^{(m)}, y^{(m)})$

$$y^{(m)} \in \mathbb{R}^C \qquad y_i^{(m)} = \begin{cases} 1 & \text{if } x^{(m)} \text{ is from the i-th class} \\ 0 & \text{elsewhere} \end{cases}$$

- network prediction: $\hat{y}^{(m)}$
- CE Loss:

$$L = -\sum_{m \in \mathbb{B}} \sum_{i=1}^{C} y_i^{(m)} \log(\hat{y}_i^{(m)})$$

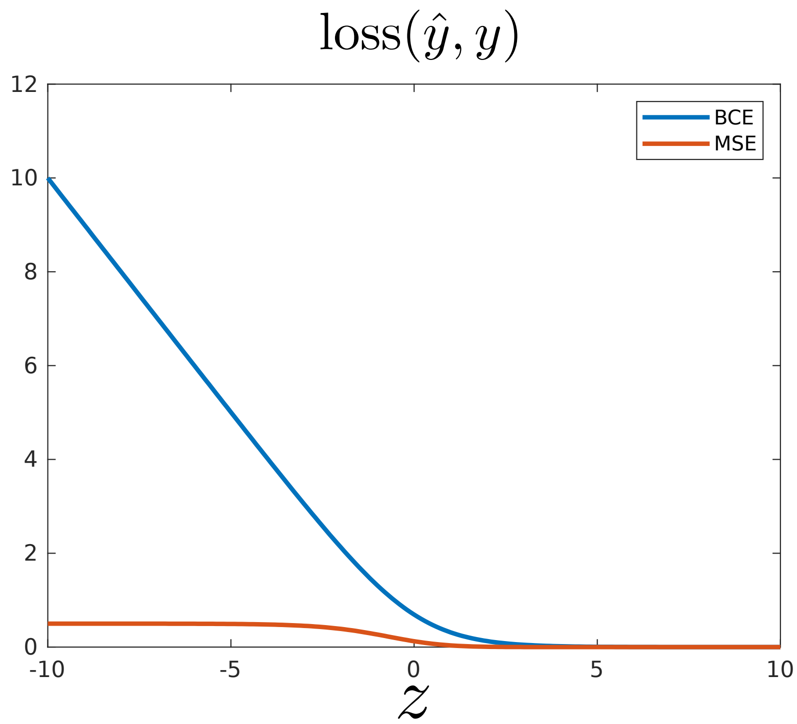
Vanishing Gradient: BCE vs MSE

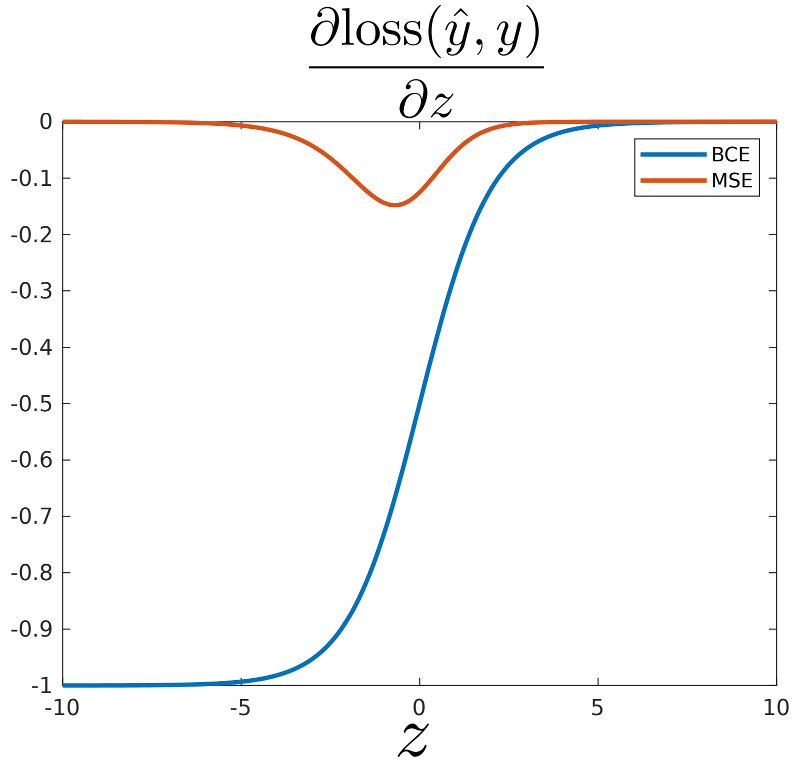
y ... GT output (label)

$\hat{y} = \sigma(z)$ ... estimated output

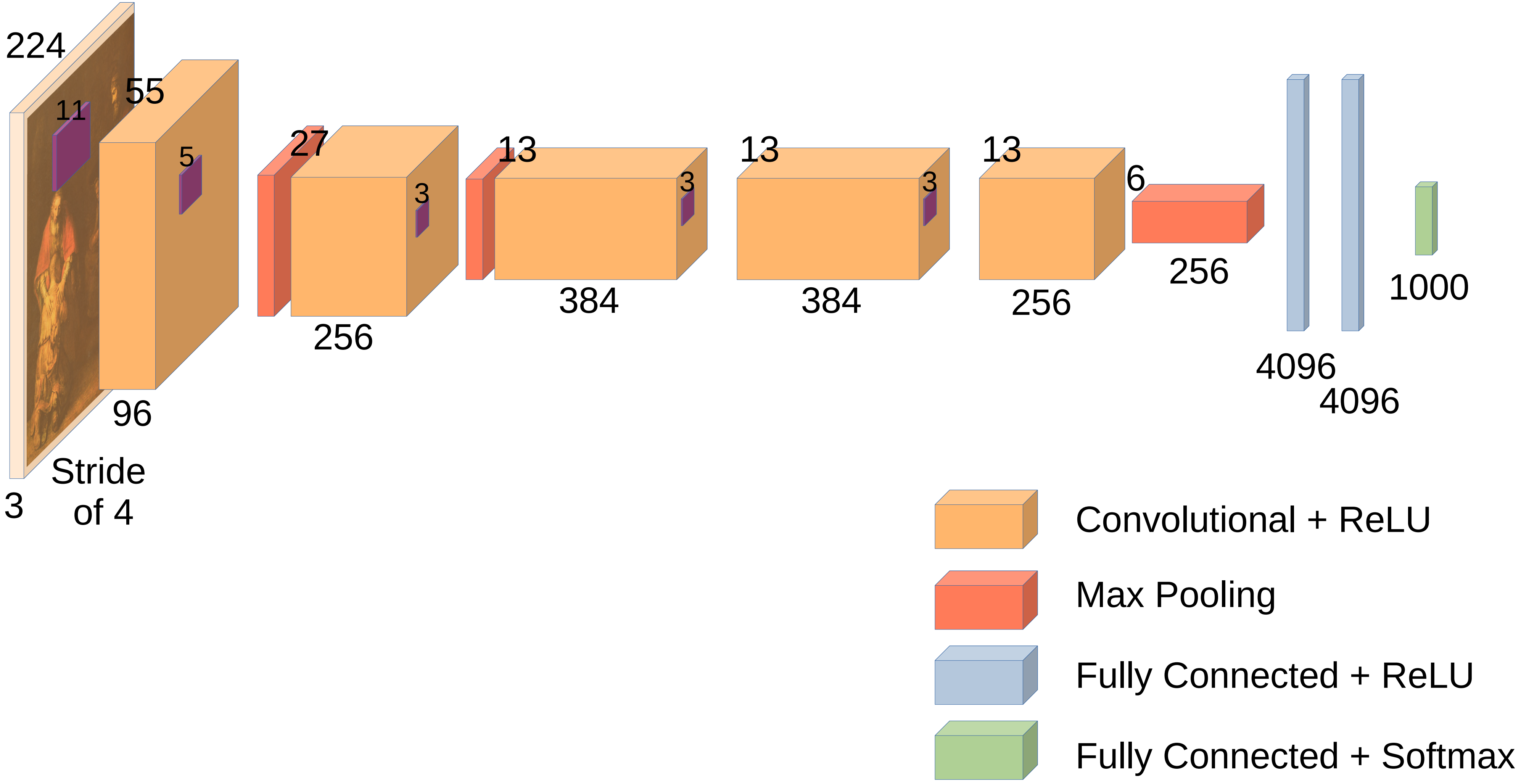
$$\mathrm{loss}(\hat{y}, y) = -y \log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$$\mathrm{loss}(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$





AlexNet

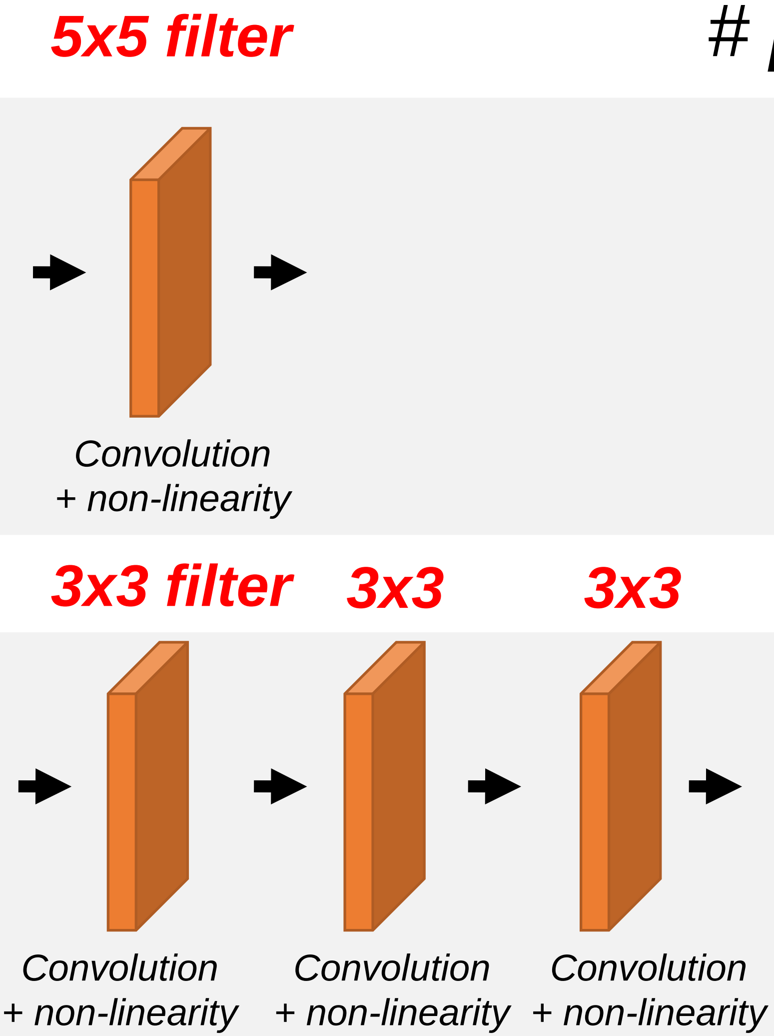


Krizhevsky et al., NIPS 2012

VGG Very Deep Net

- Improvements over AlexNet
- More layers + smaller convolutional filters (3x3)

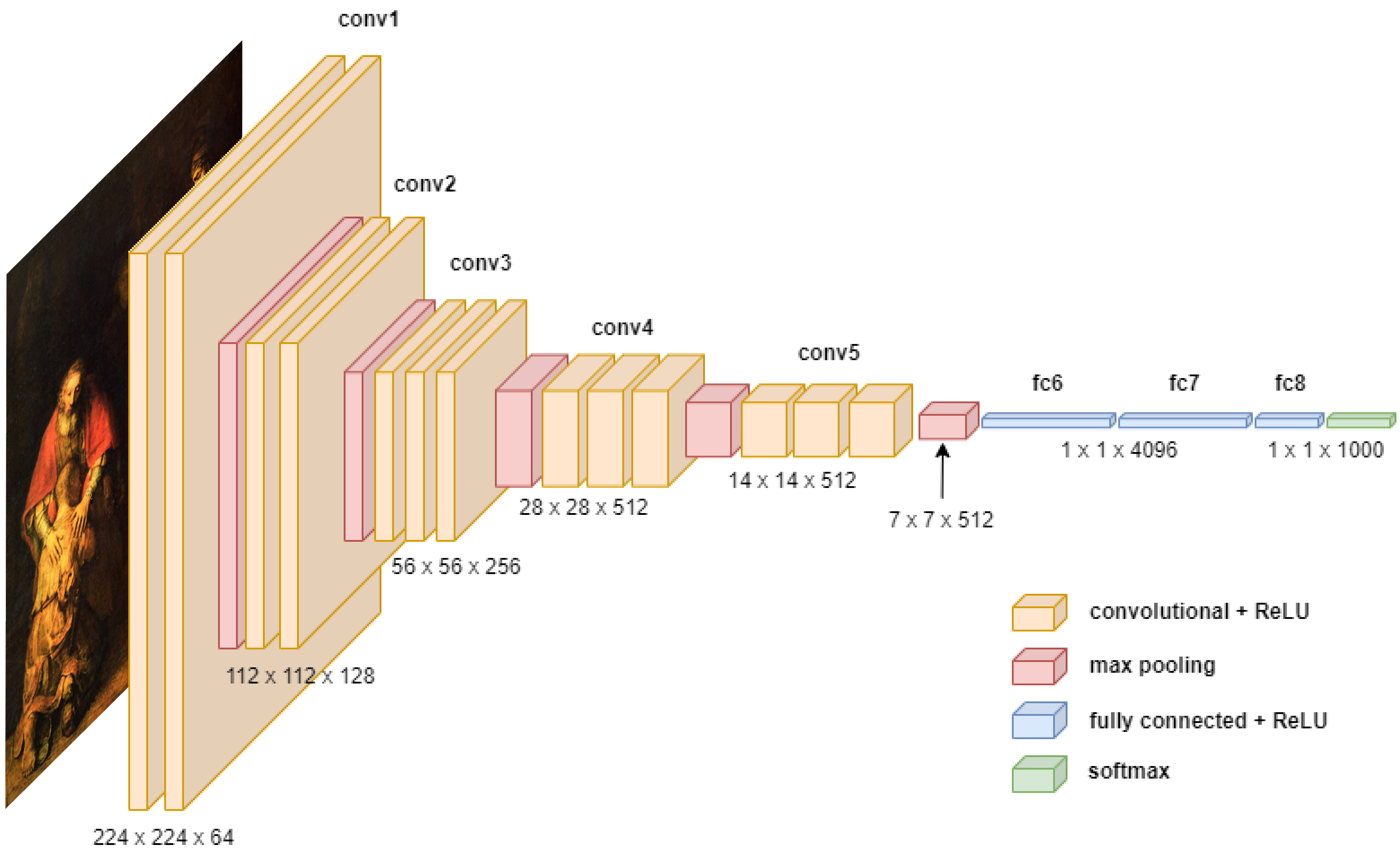
	Filter	# parameters	receptive field
AlexNet block	5x5 filter	25	5x5
VGG-16 block	3x3 filter, 3x3, 3x3	27	7x7



Larger receptive field with same # parameters

VGG

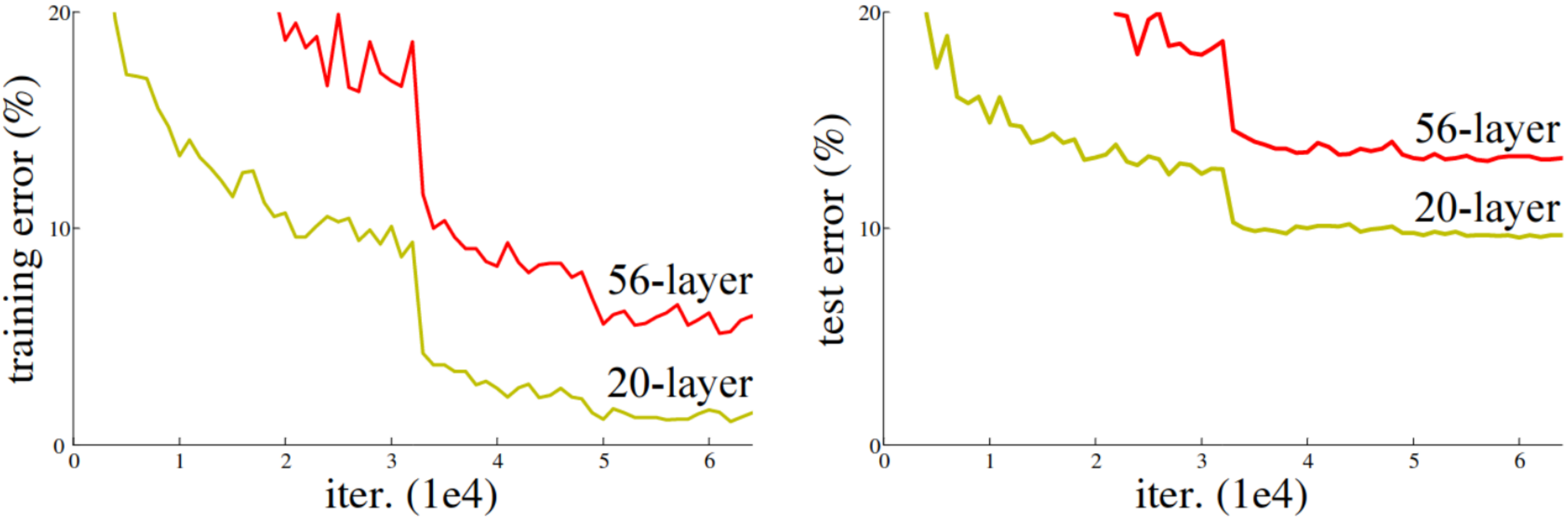
- VGG-16



Simonyan & Zisserman, ICLR 2014

How deep can we go?

- What happens if we keep increasing the depth of VGG?

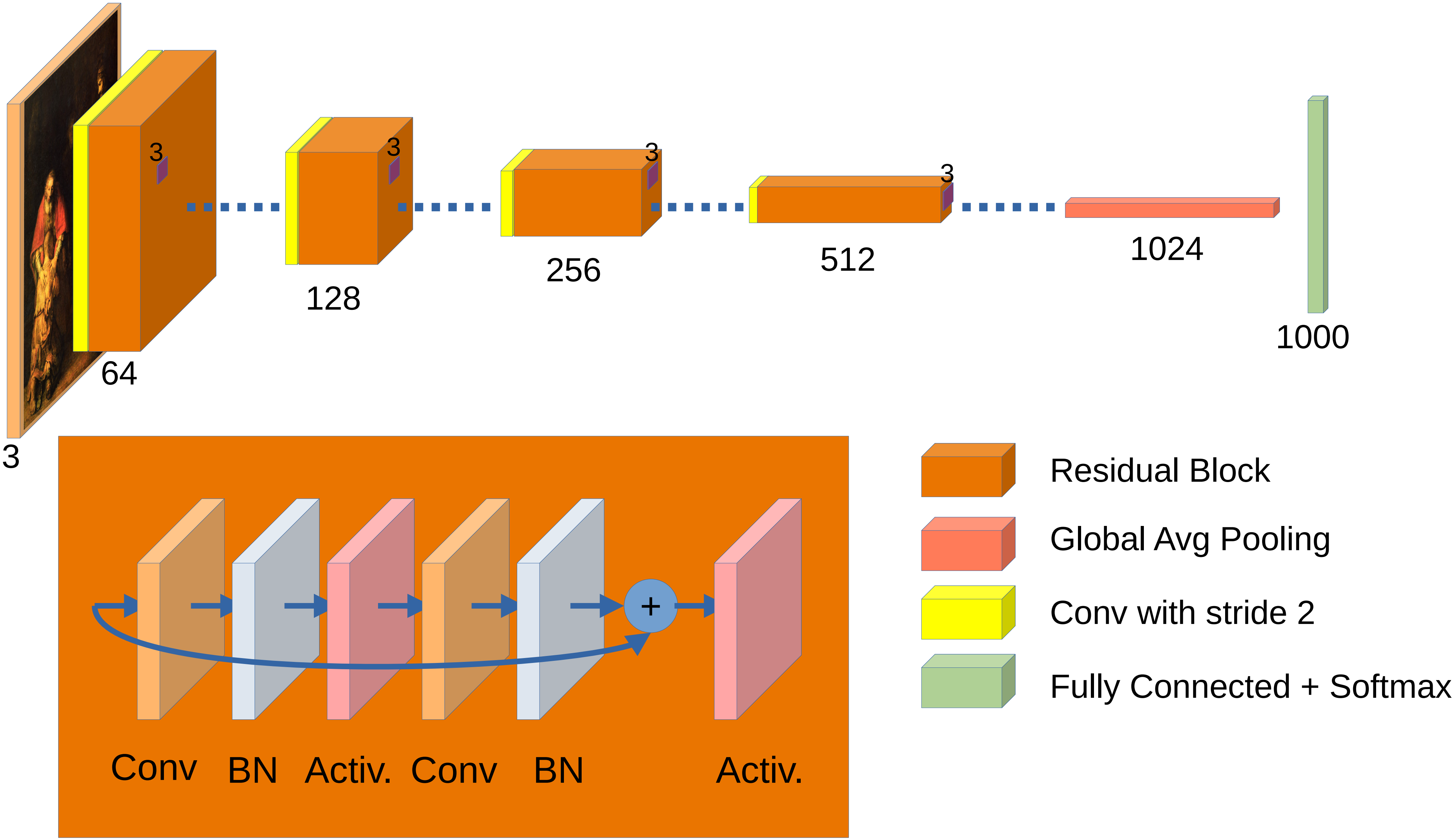


- A (too) deep network is too hard to optimize!

He et al.: *Deep Residual Learning for Image Recognition*, CVPR 2016

ResNet

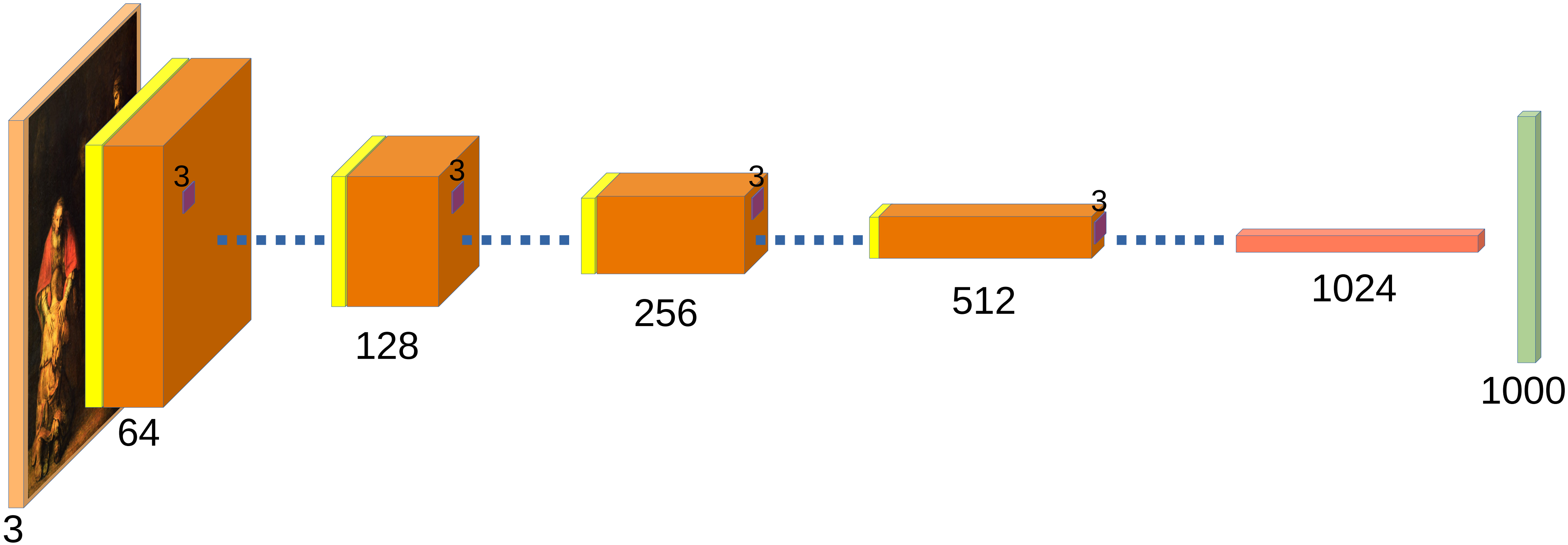
- Residual connections



He et al.: *Deep Residual Learning for Image Recognition*, CVPR 2016

ResNet

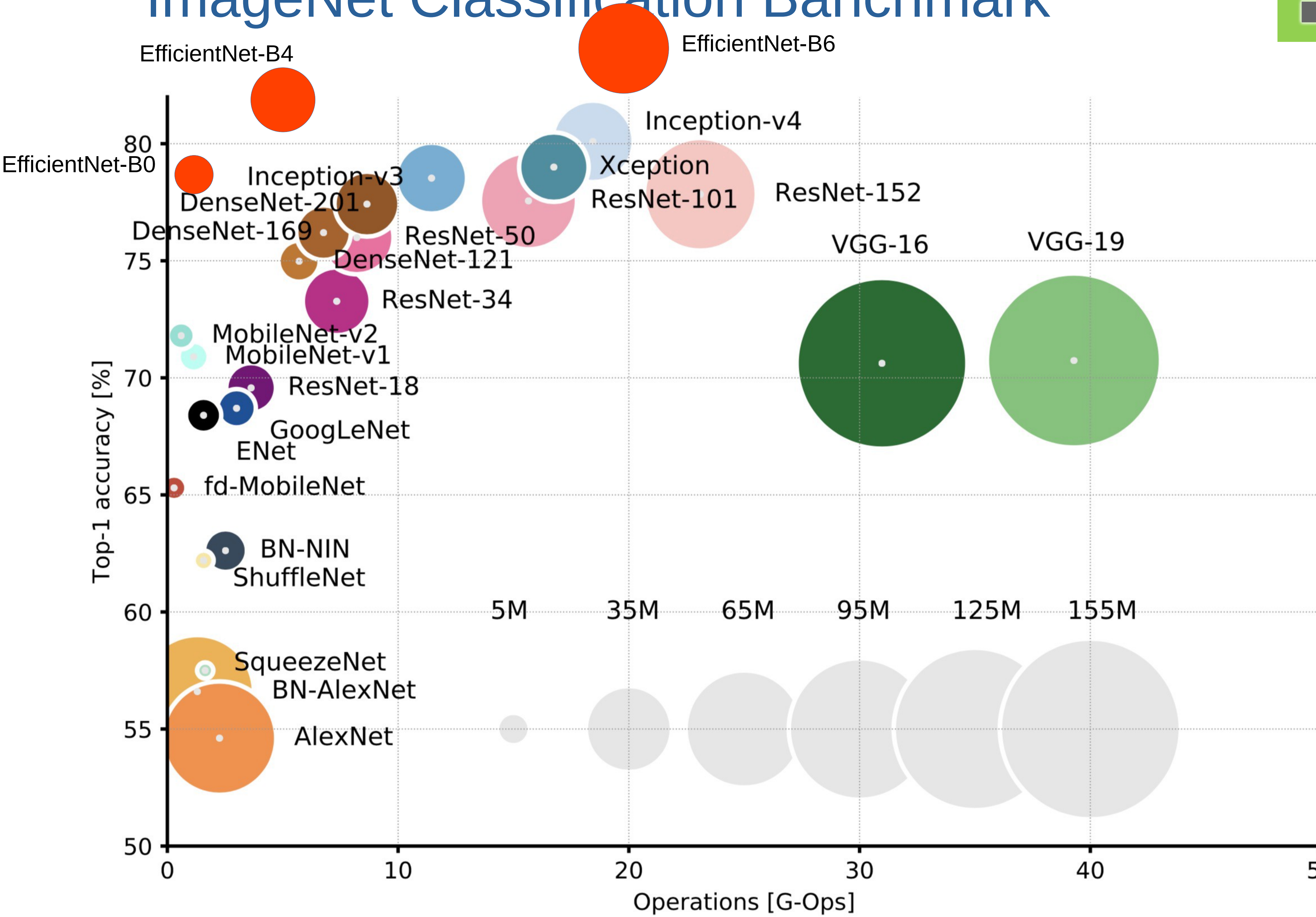
- Residual connections



- We can go now very deep!
- ResNet variants:
  - 34, 50, 101, 152 layers

What is the benefit of Global Avg Pooling?

ImageNet Classification Banchmark

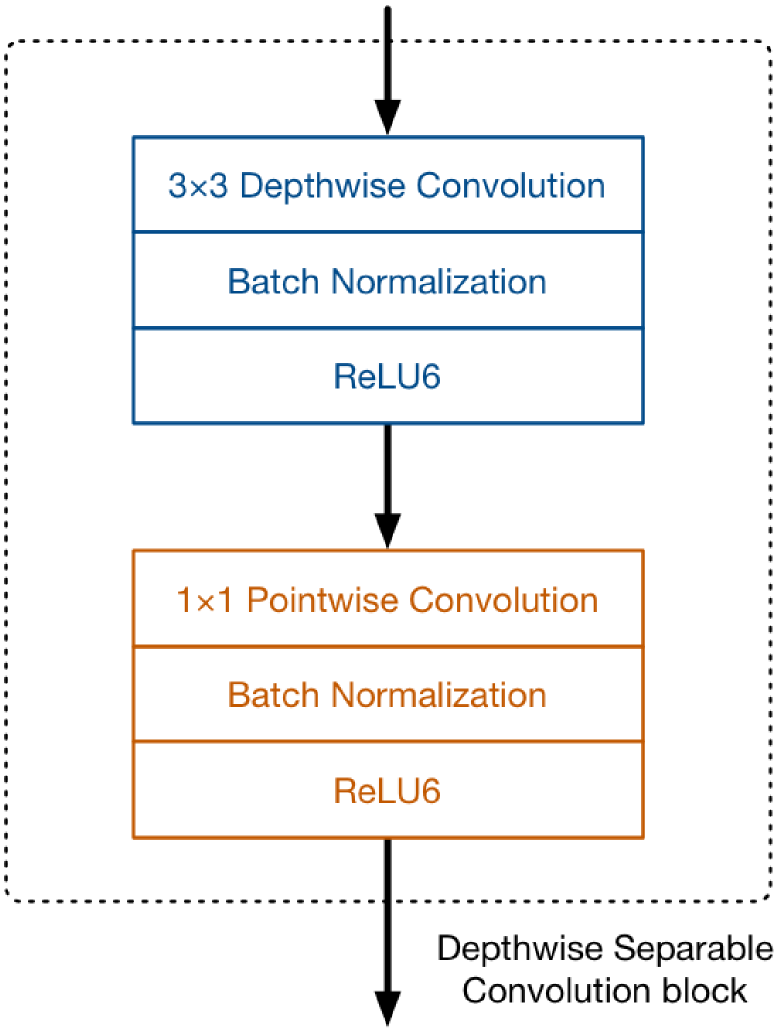


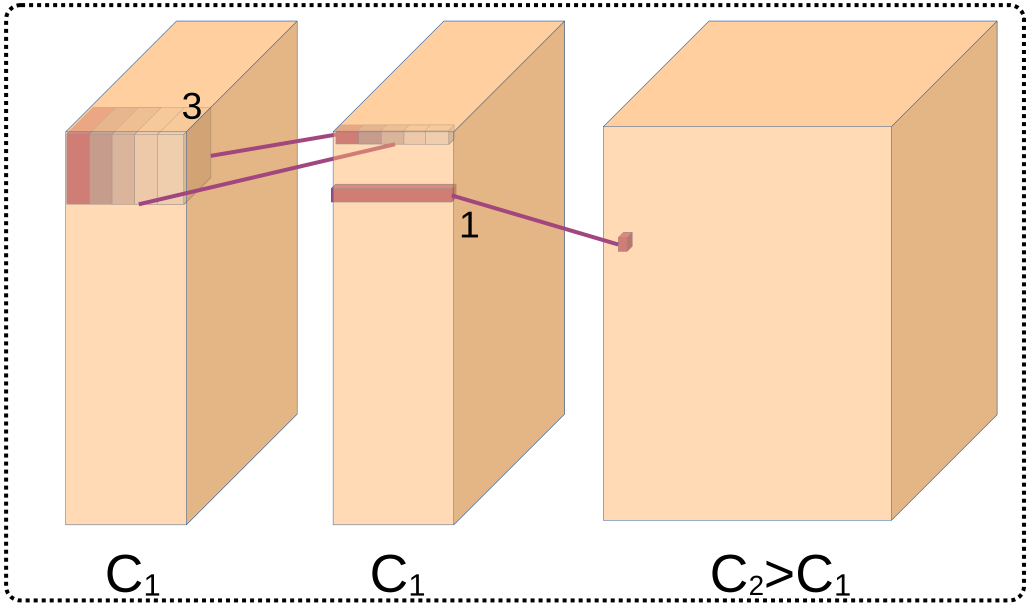
Bianco et al., IEEE Access, 2018

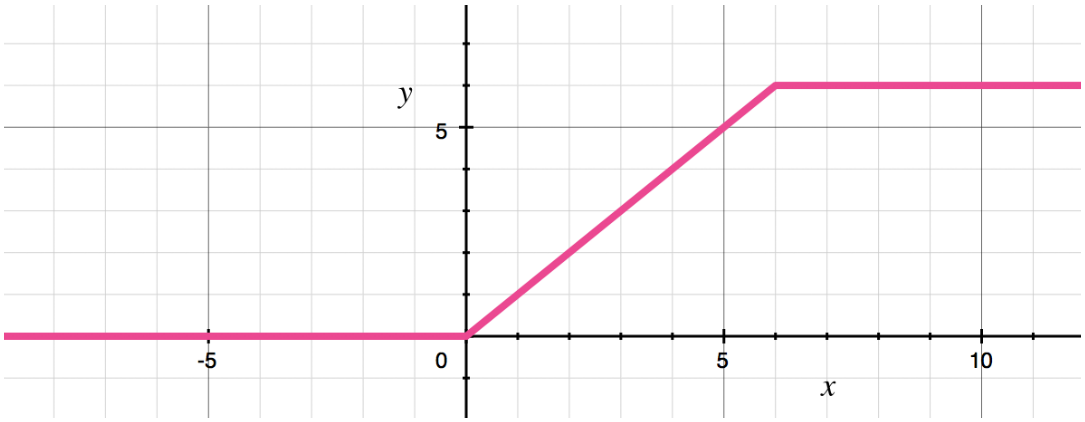
MobileNet

Compare the number of parameters when the standard convolution is used.

- Computationally less demanding, fixed-point arithmetic

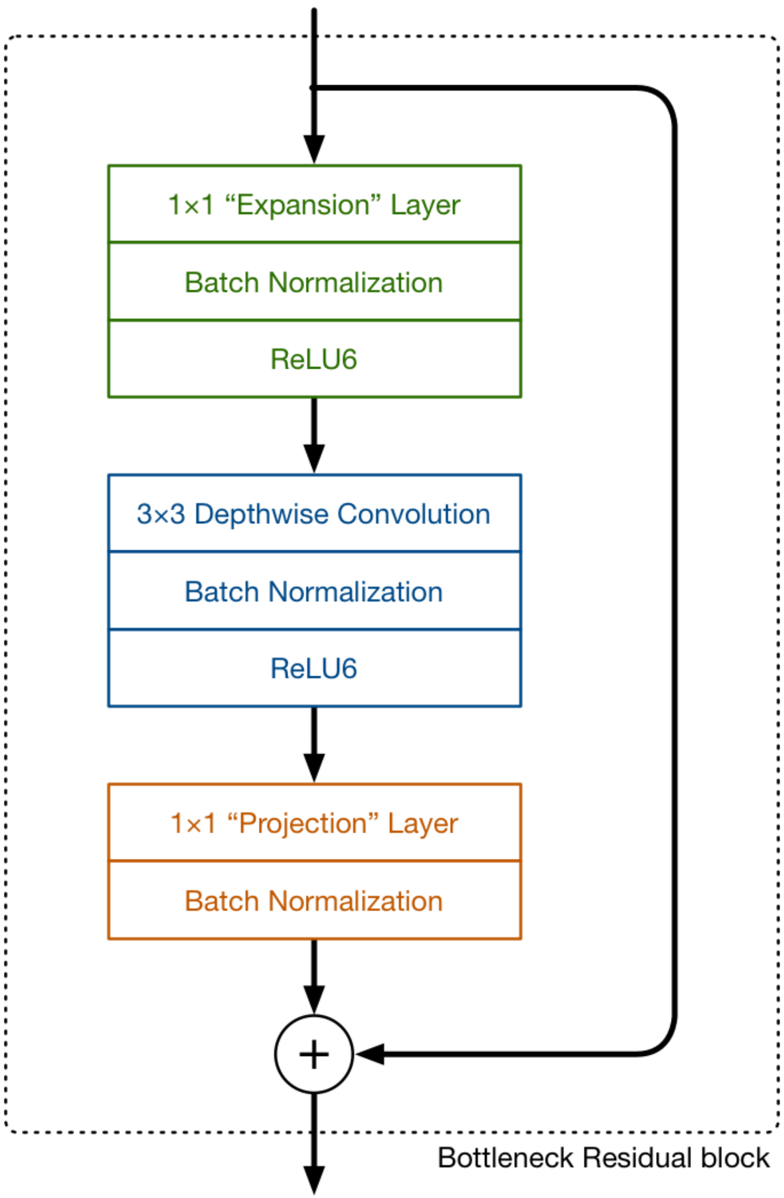


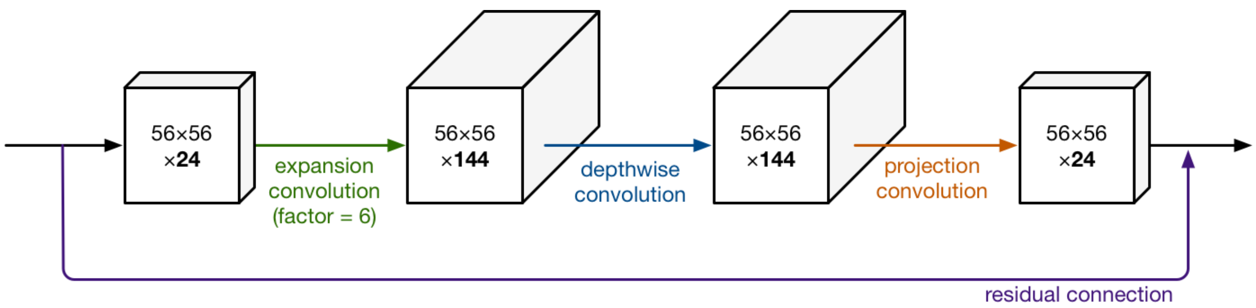




ReLU6

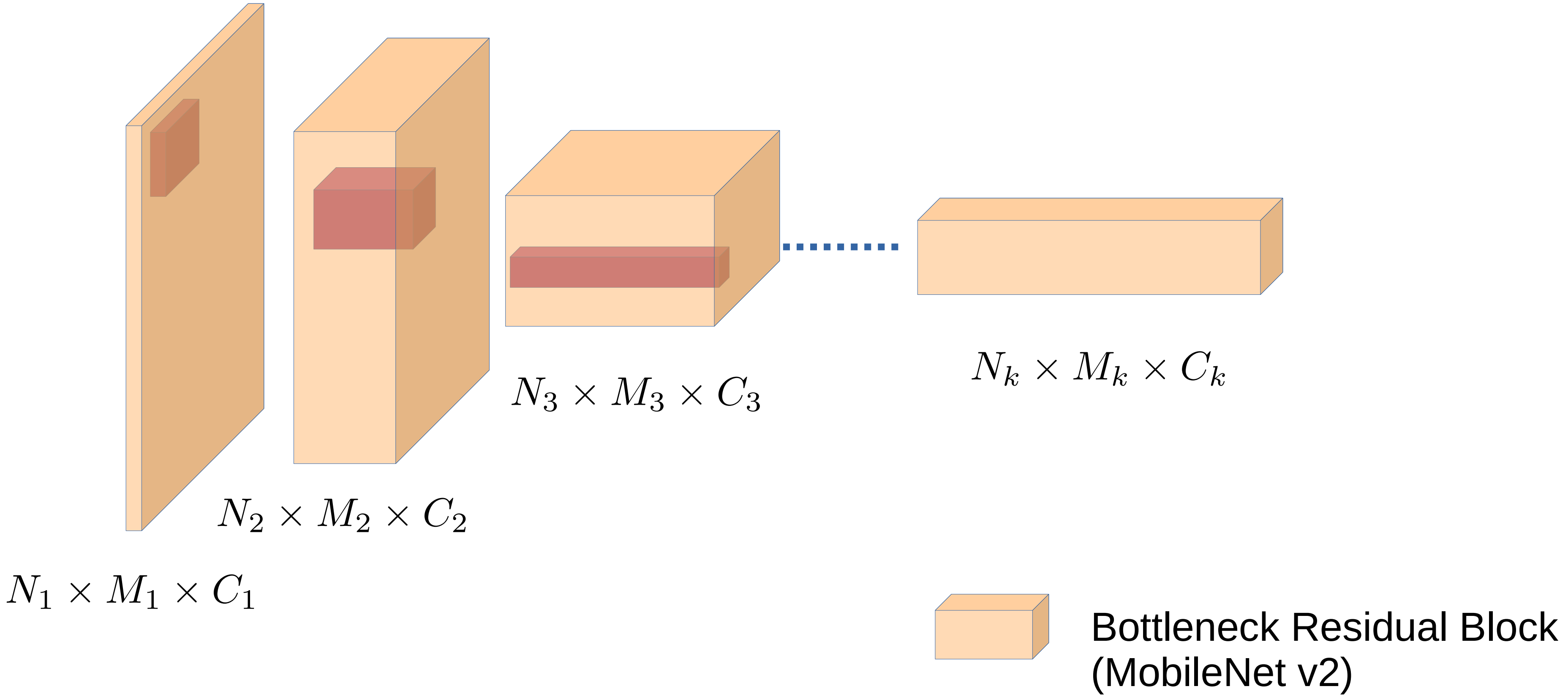
MobileNet v2





- Expand
- Filter in the higher dimensional space
- Project back
- Add

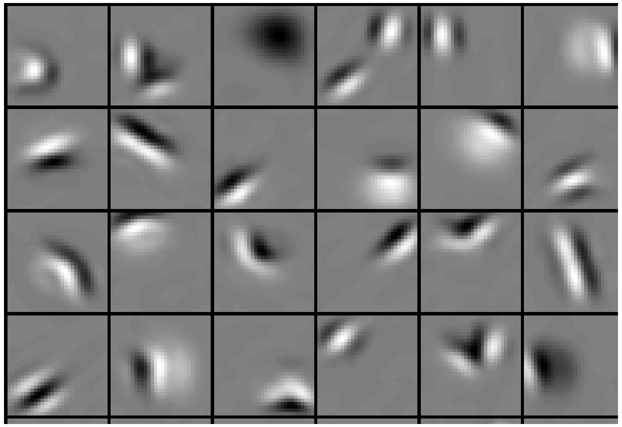
EfficientNet



- Optimal resolution, width and depth - > Compound Scaling

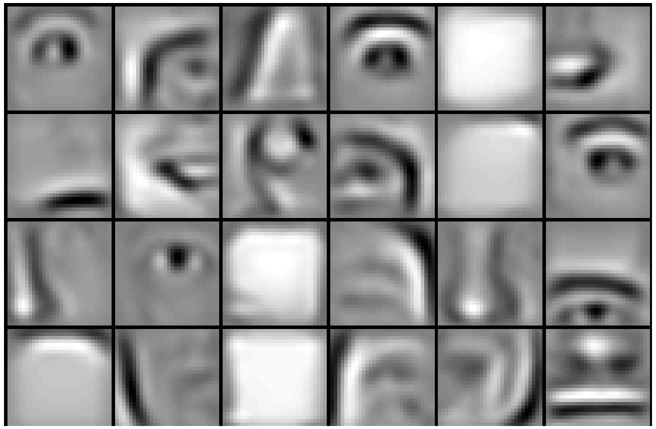
Visualization of Deep CNN Features

Low level features



1st conv layer

Mid level features



2nd conv layer

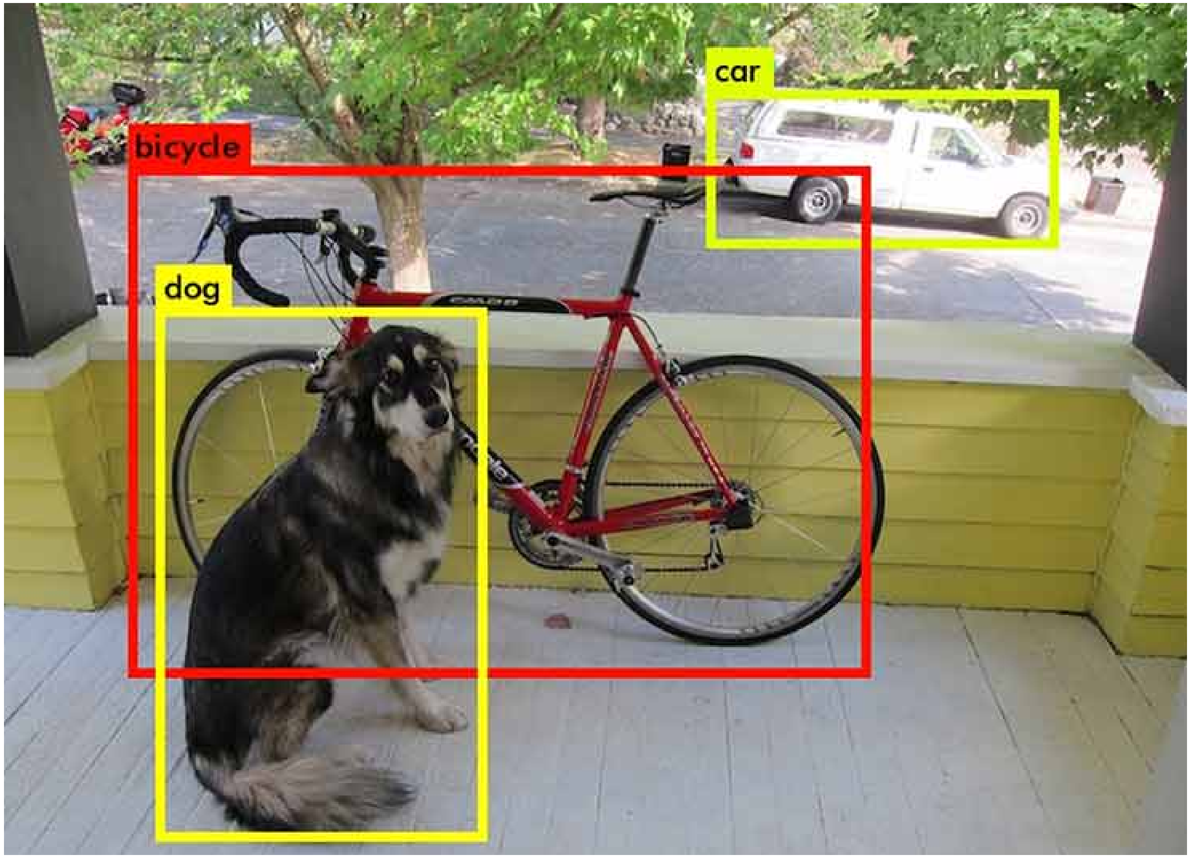
High level features

3rd conv layer

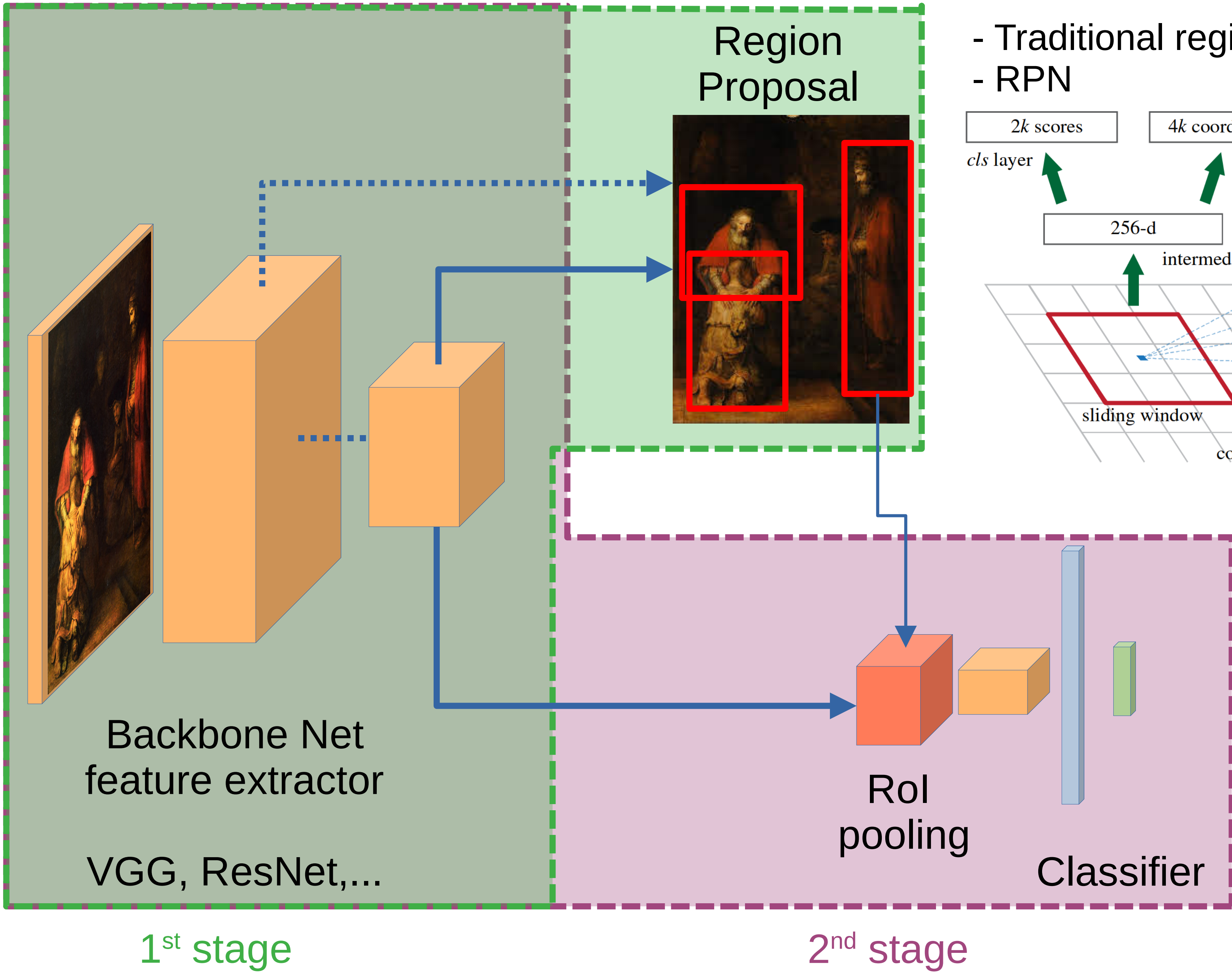
Recent Advances

- Inception module
- Convolutional Block Attention module
- Transformers – self-attention (CoAtNet)

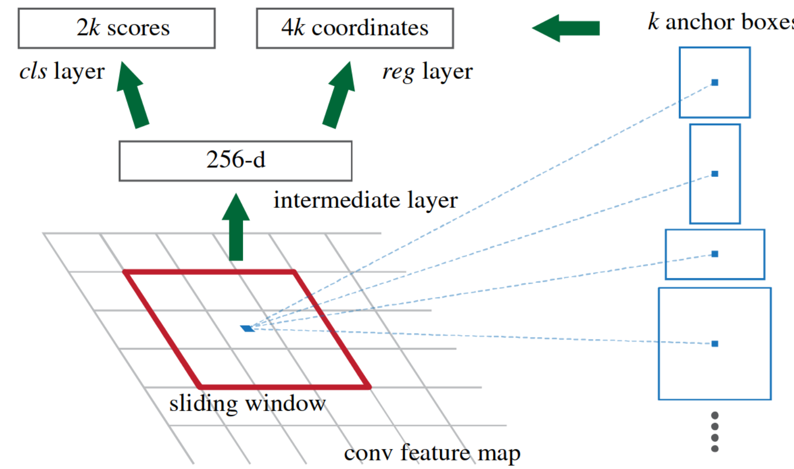
Object Detection



Two-Stage Detector

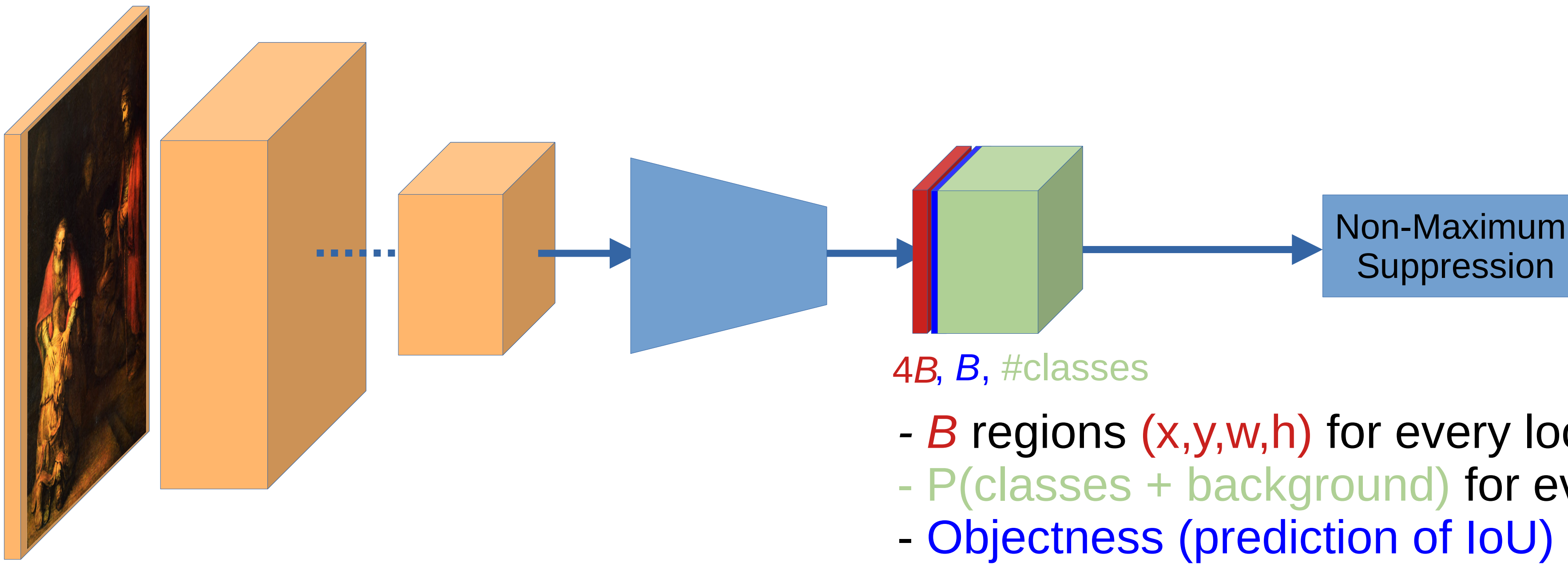


- Traditional region growing methods
- RPN



Girshick et al., R-CNN, 2015
Girshick, Fast R-CNN, 2015
Ren et al., Faster R-CNN, 2016

One-Stage Detector



- *B* regions (x,y,w,h) for every location
- P(classes + background) for every location
- Objectness (prediction of IoU)

Backbone Net
feature extractor

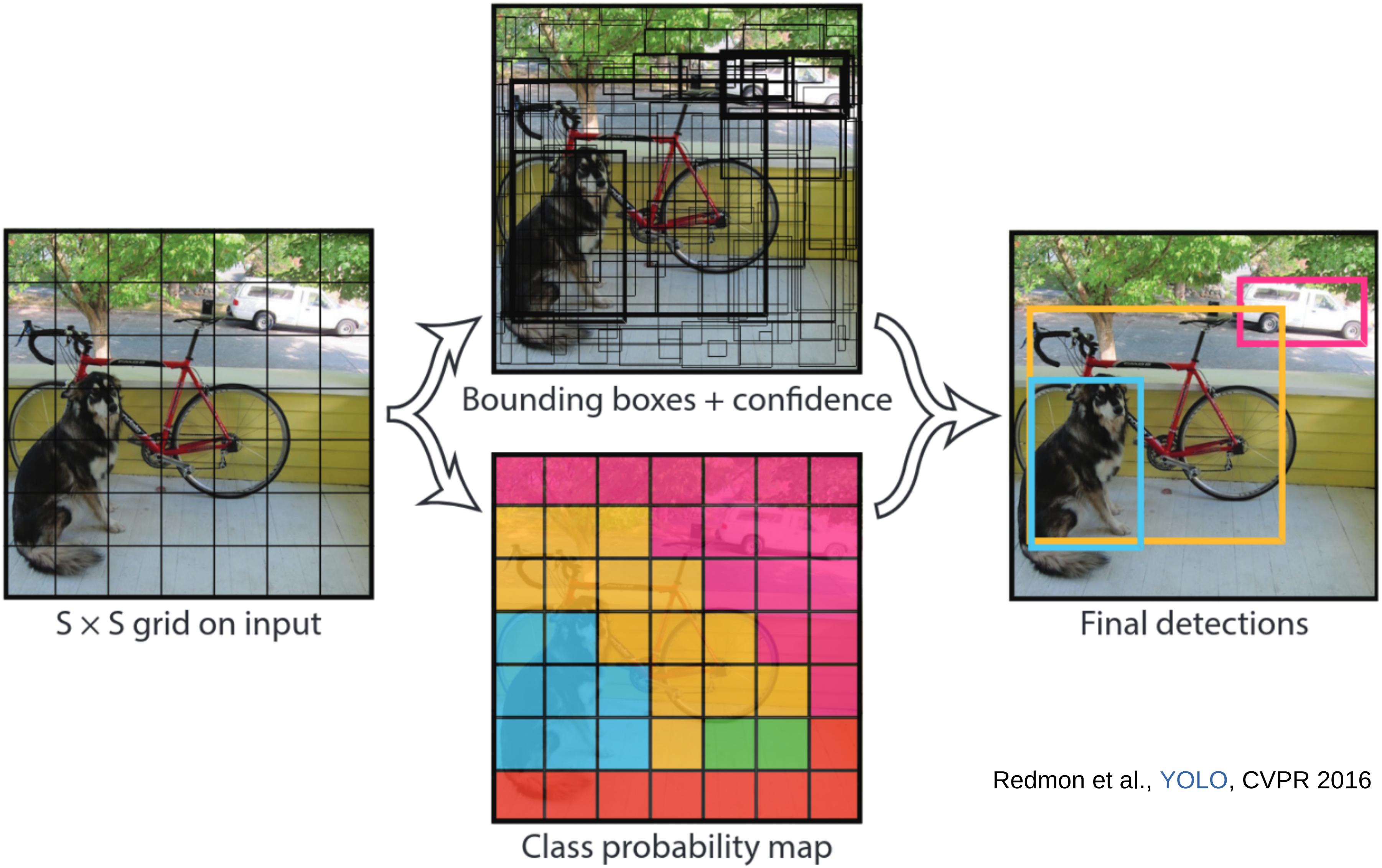
VGG, ResNet,...

- Class imbalance: background regions more common than other!

Redmon et al., YOLO, CVPR 2016
Liu et al., SSD, ECCV 2017
Lin et al., RetinaNet, 2018

YOLO

- You Only Live/Look Once



Redmon et al., YOLO, CVPR 2016

YOLO loss

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

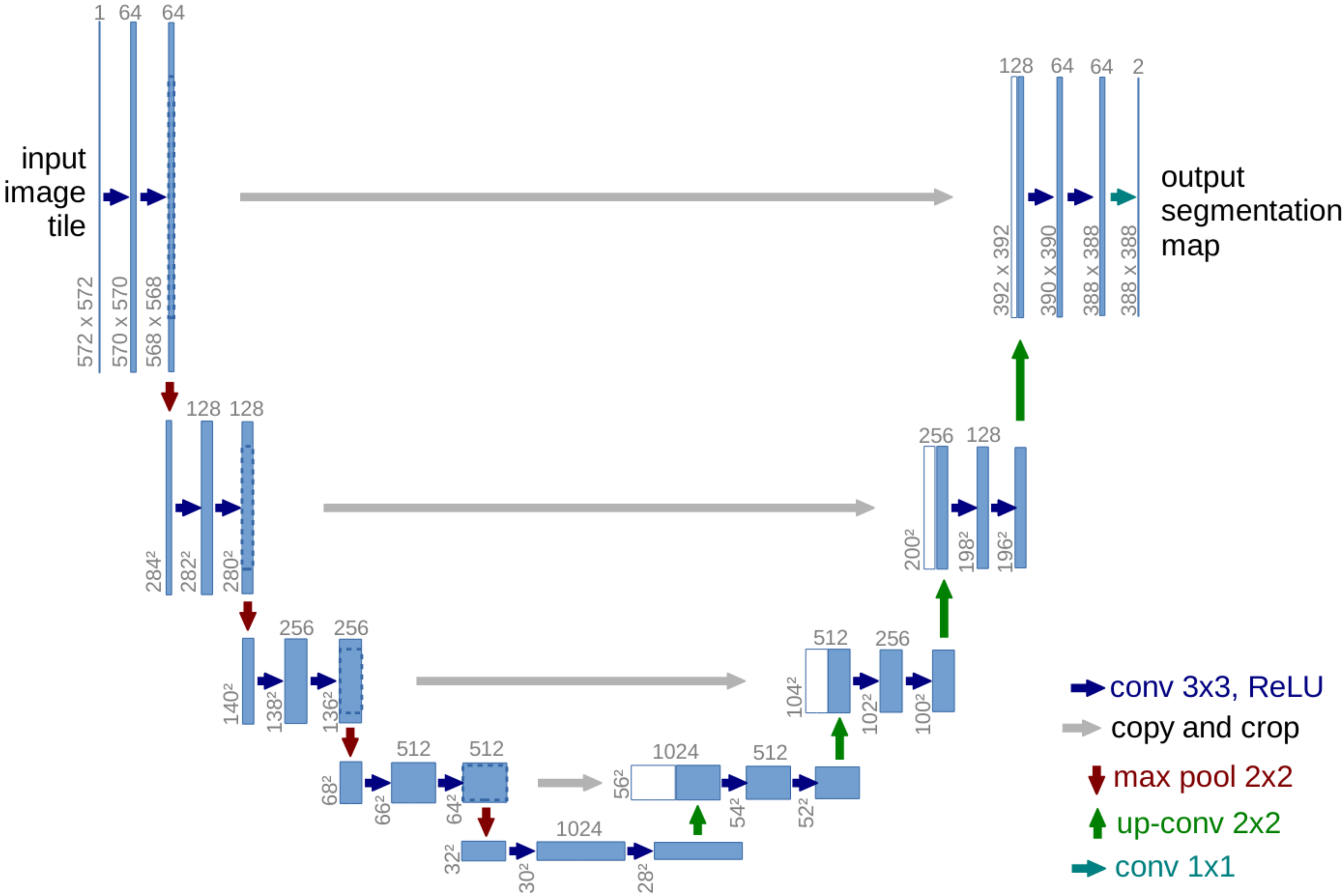
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

$\mathbb{1}_{ij}^{\text{obj}}$ *j*-th bounding box predictor in cell *i* is “responsible” for the prediction.

$\mathbb{1}_{ij}^{\text{noobj}}$ otherwise

Semantic Segmentation

U-Net



Ronneberger et al., U-Net, 2015

Autoencoder

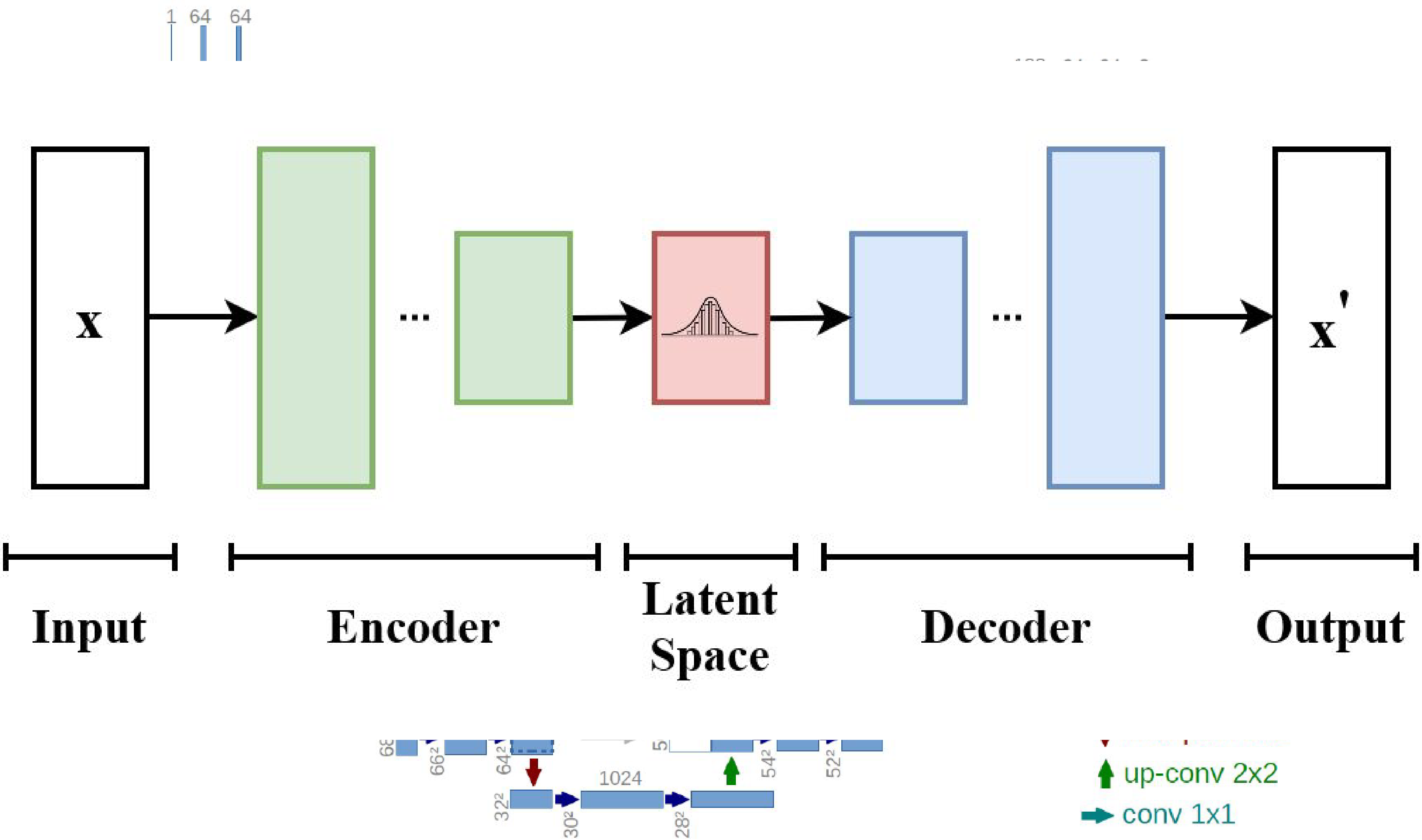
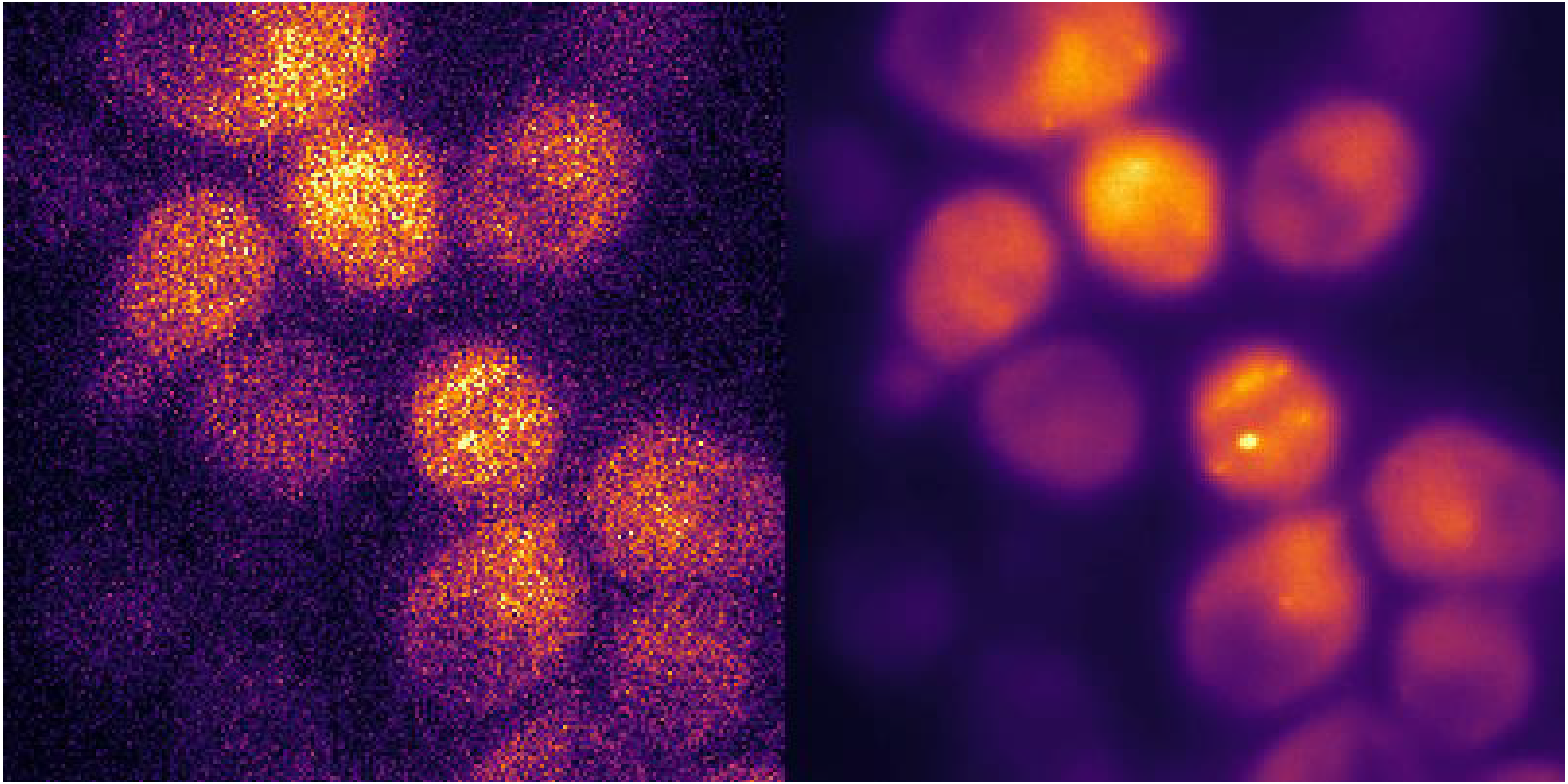
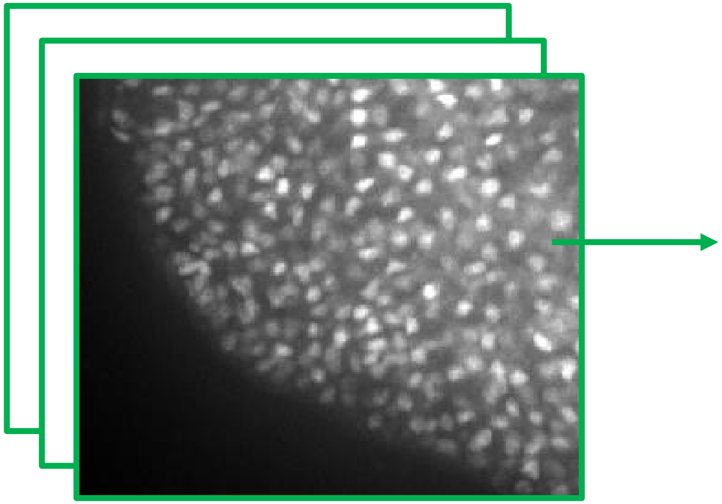


Image Reconstruction - Denoising

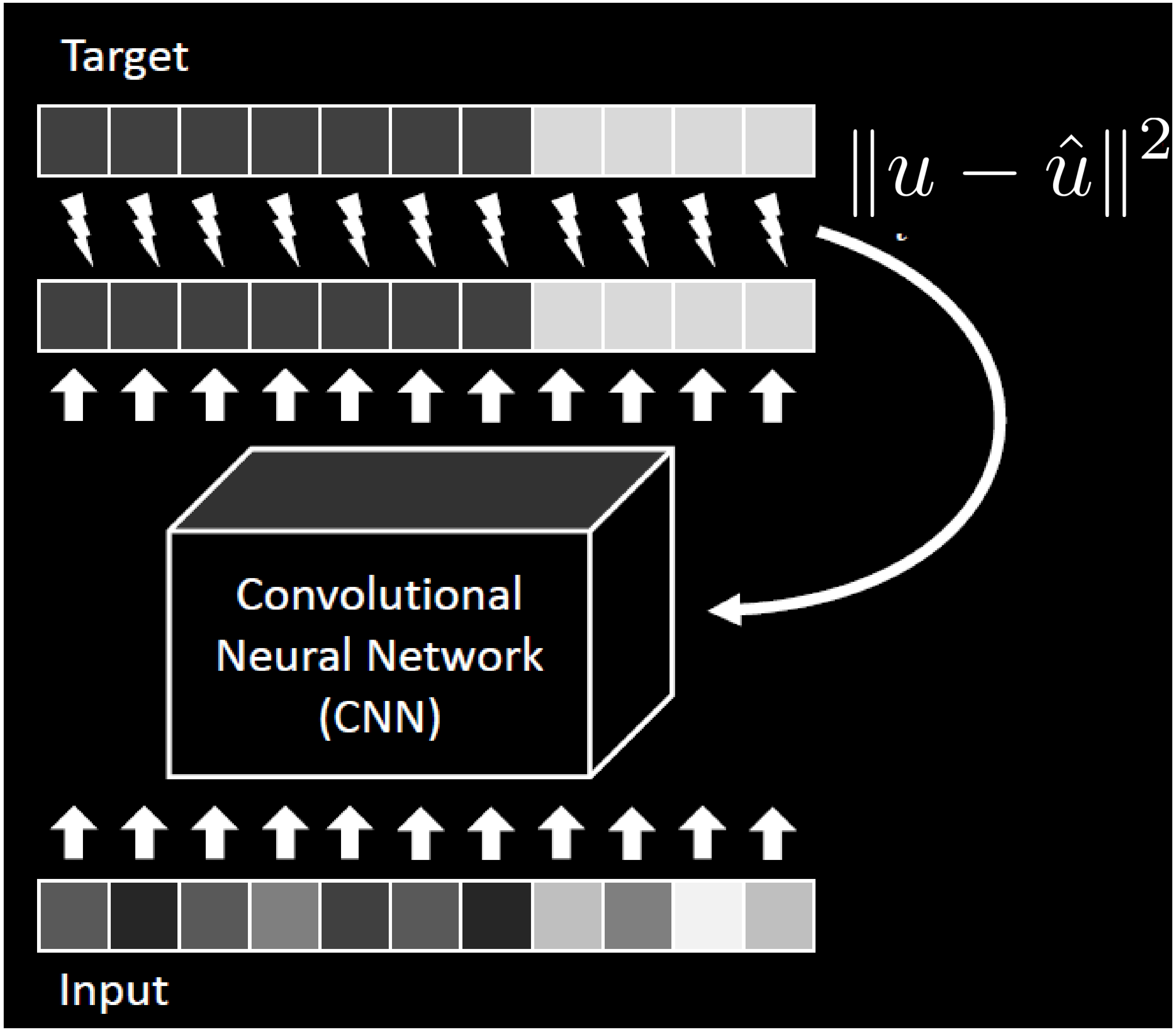


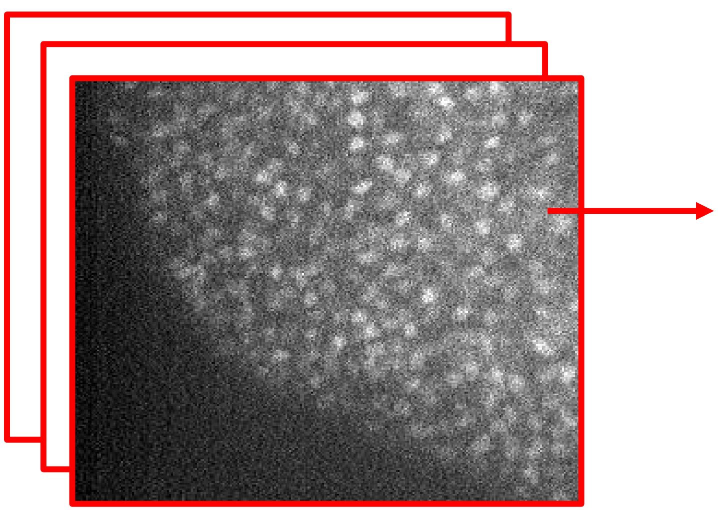
Data by Stephanie Heinrich

Supervised Learning



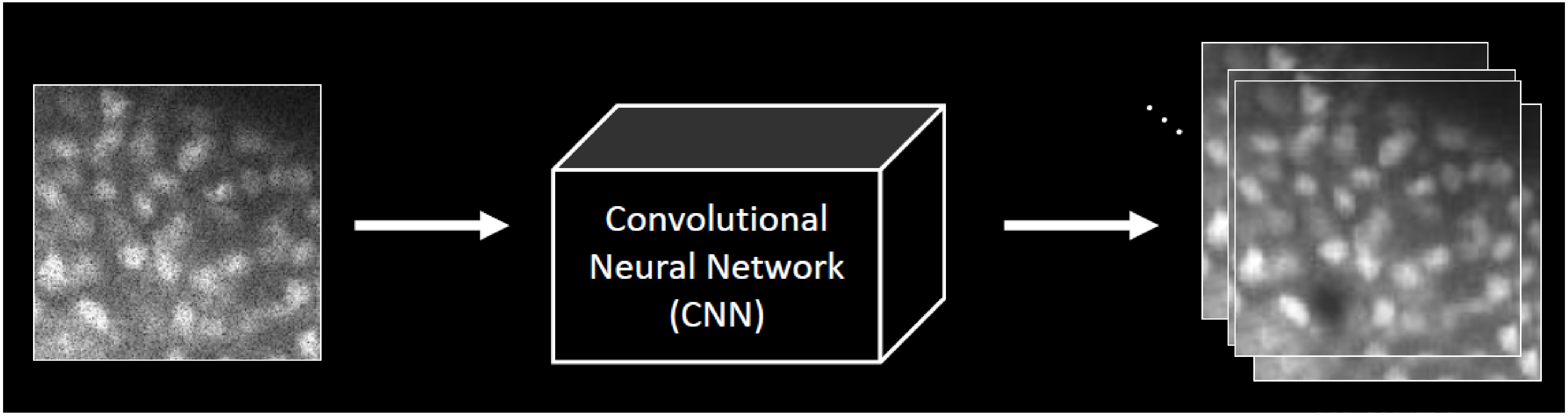
Clean image u





Degraded image z

Minimizing the Squared Error

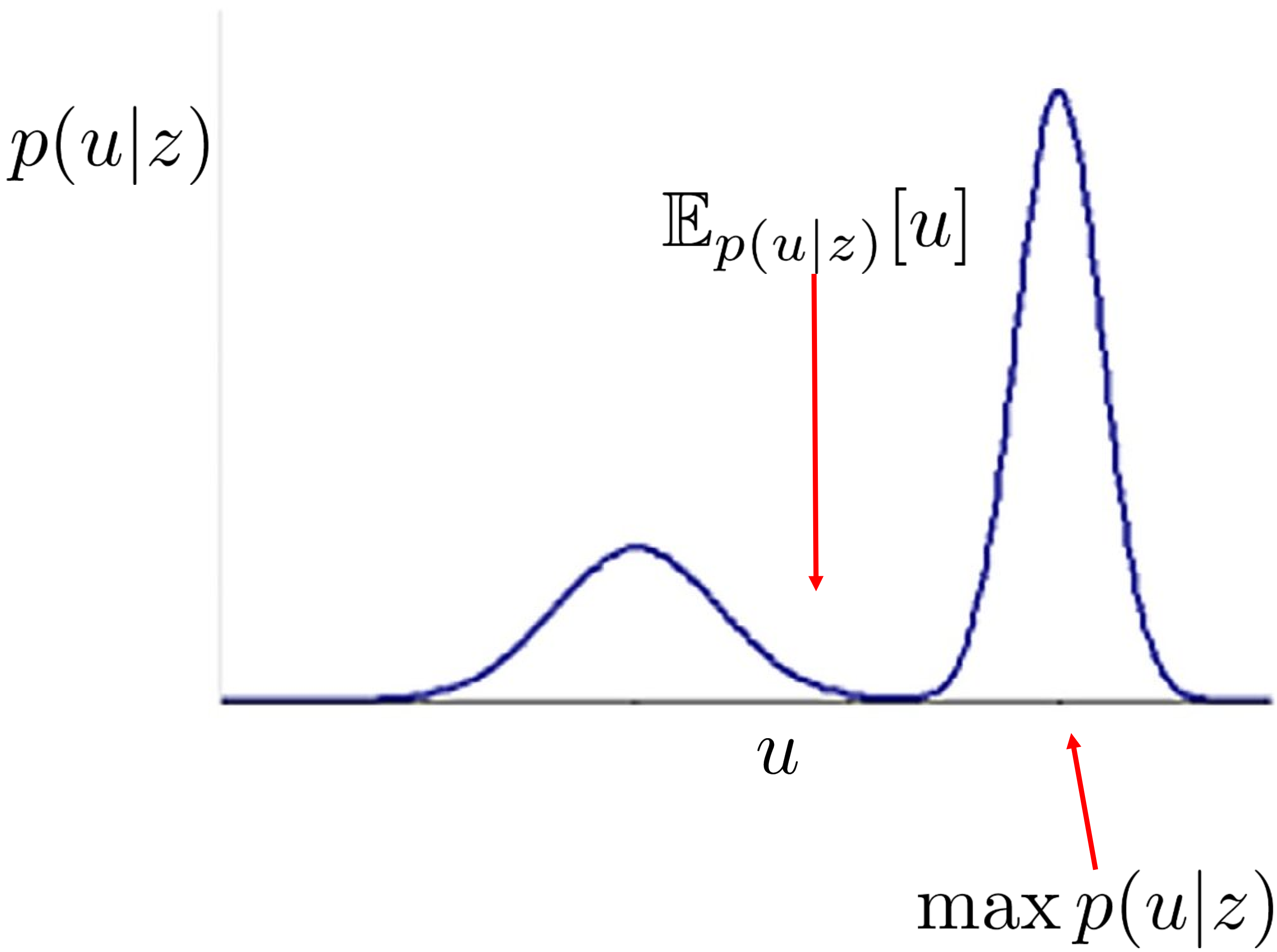


z $\hat{u}$

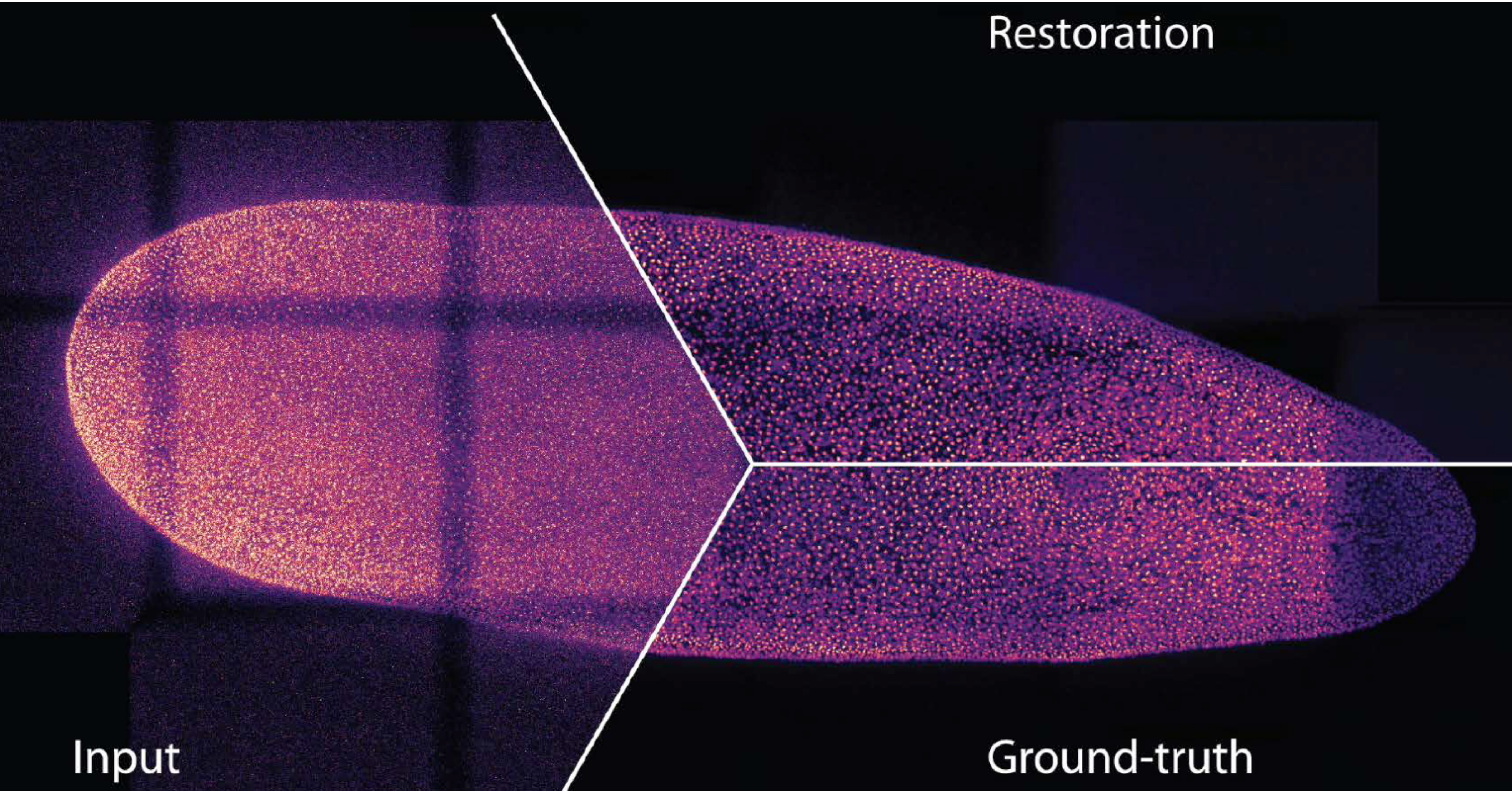
- Minimizing $||u - \hat{u}||^2$
- Remember MAP?

$$\hat{u} \approx \mathbb{E}_{p(u|z)}[u]$$

$$\hat{u} \approx \max p(u|z)$$


$p(u|z)$
$\mathbb{E}_{p(u|z)}[u]$
u
$\max p(u|z)$

Content-aware Restoration



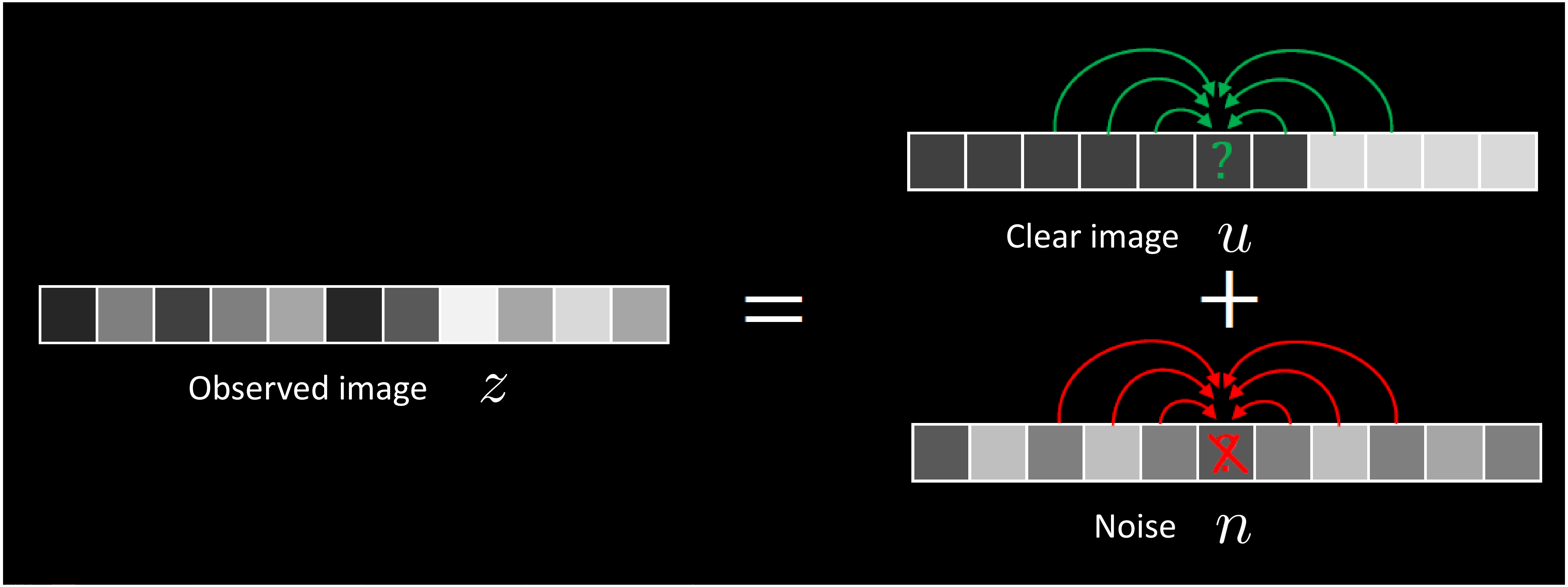
Limitations

- GT must be obtainable!
- Training data must sample all visual features of interest!

- Can we do if we have degraded data only?
  (self-supervised)
- YES! But it works only for white noise.

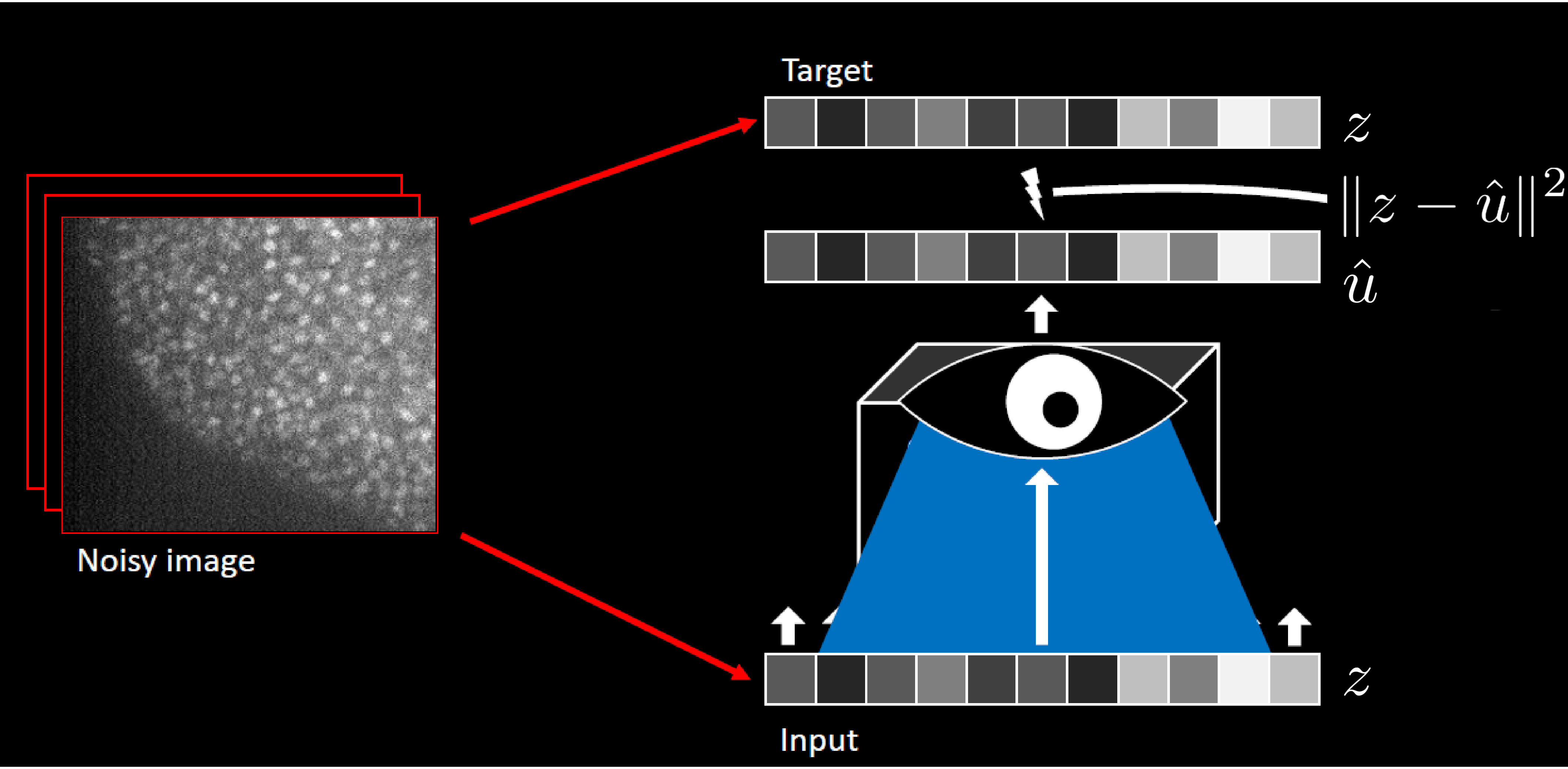
Self-Supervised Learning

- Noise2Void assumption

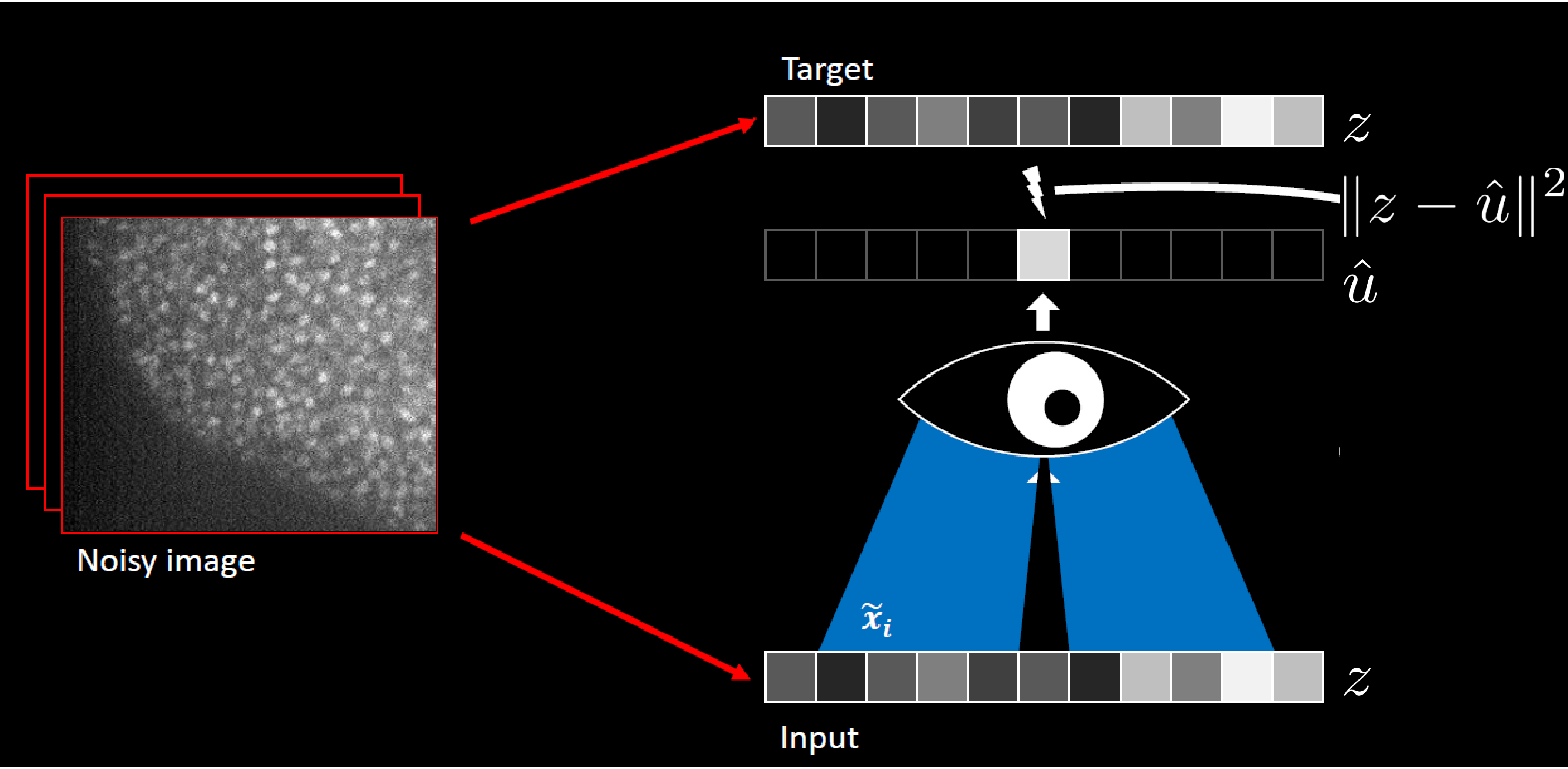


Krull, Buchholz, and Jug 2019

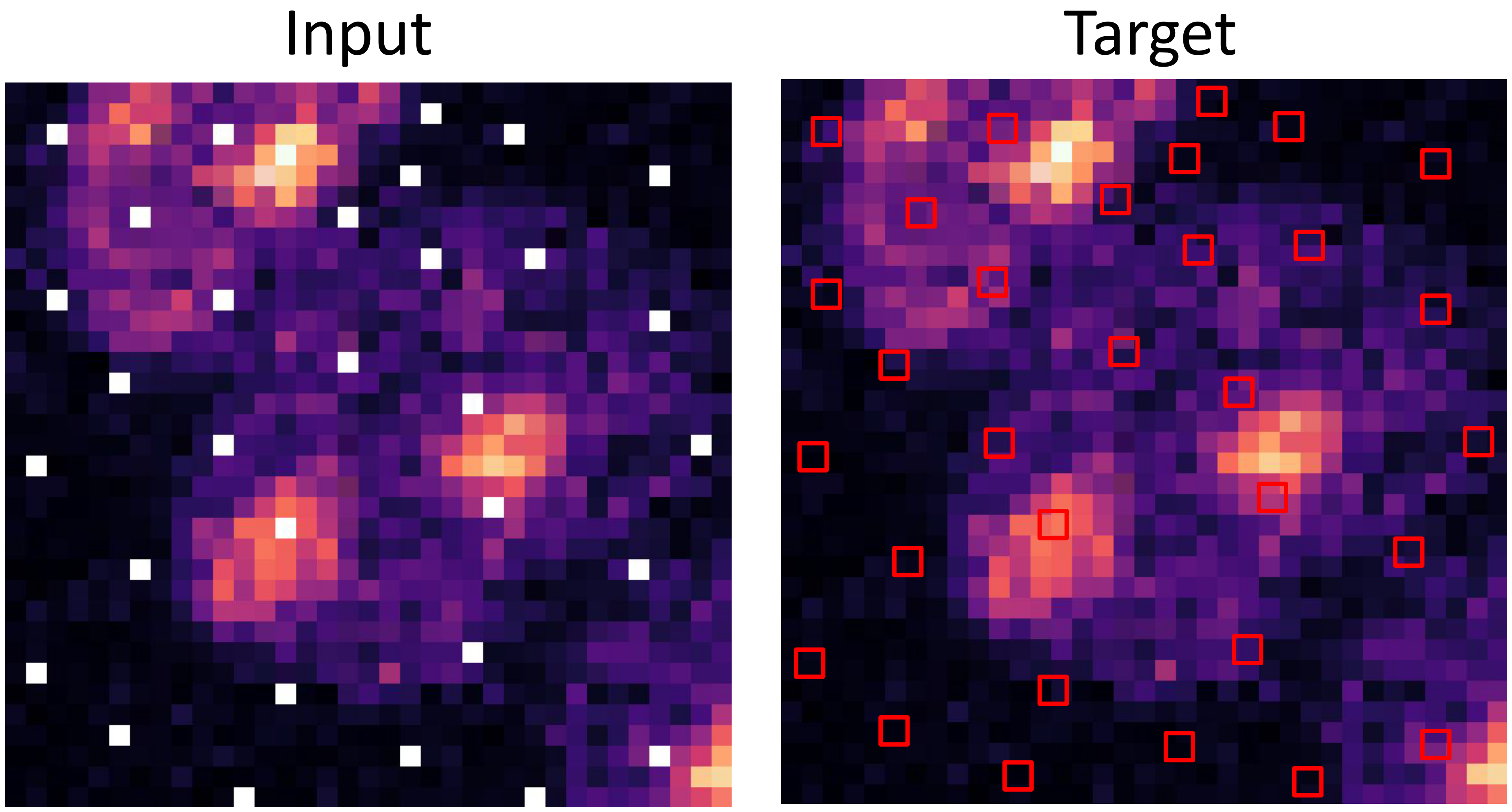
Noise2Void



Noise2Void



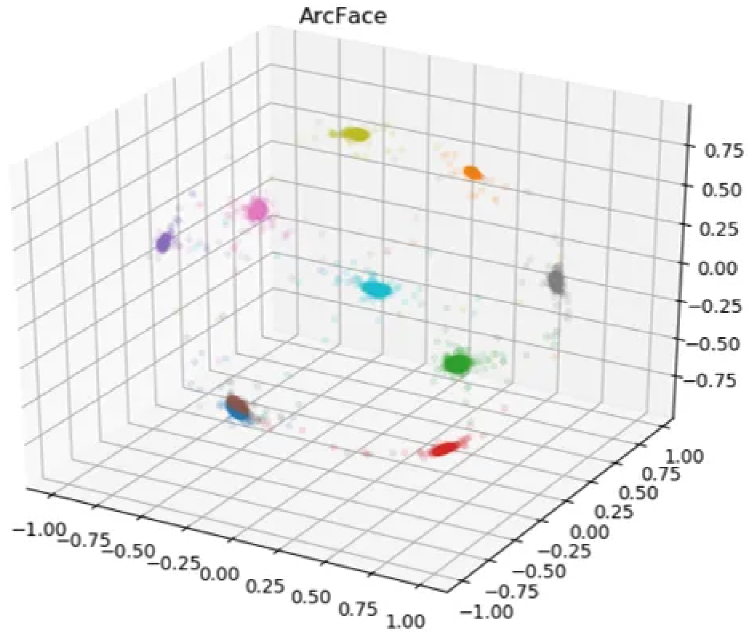
Blind Spot Implementation

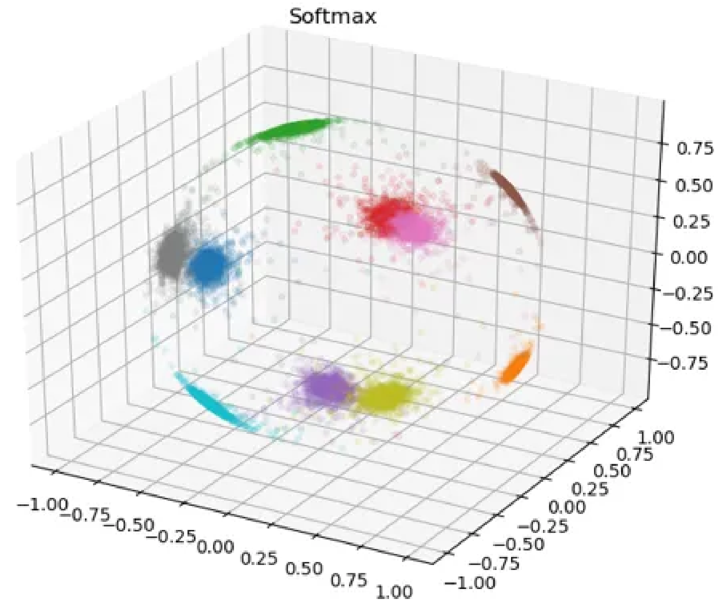


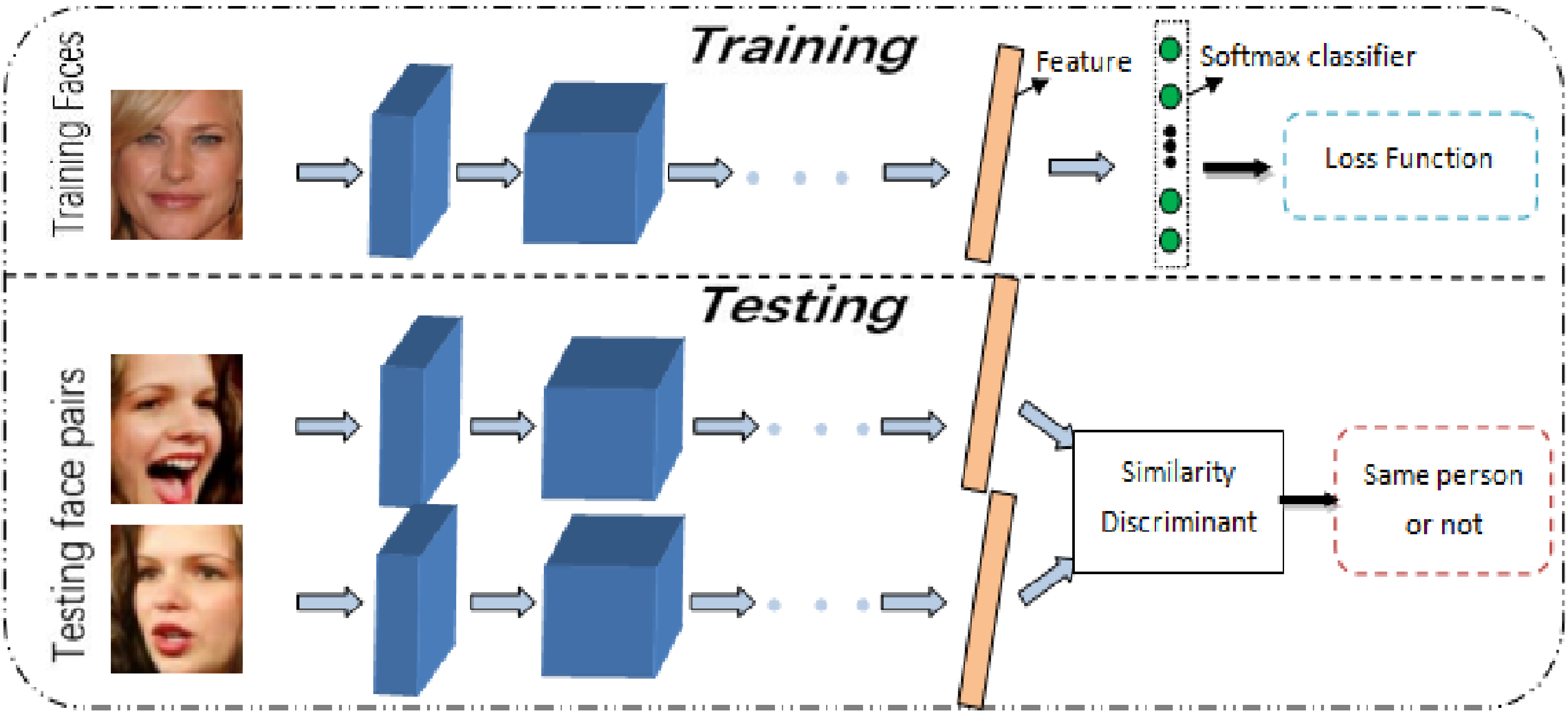
Face Recognition

- Softmax (classification) is not appropriate
- One-shot learning
  - Recognize a person even if we have a single photo.
  - No retraining if a new person enters the db.

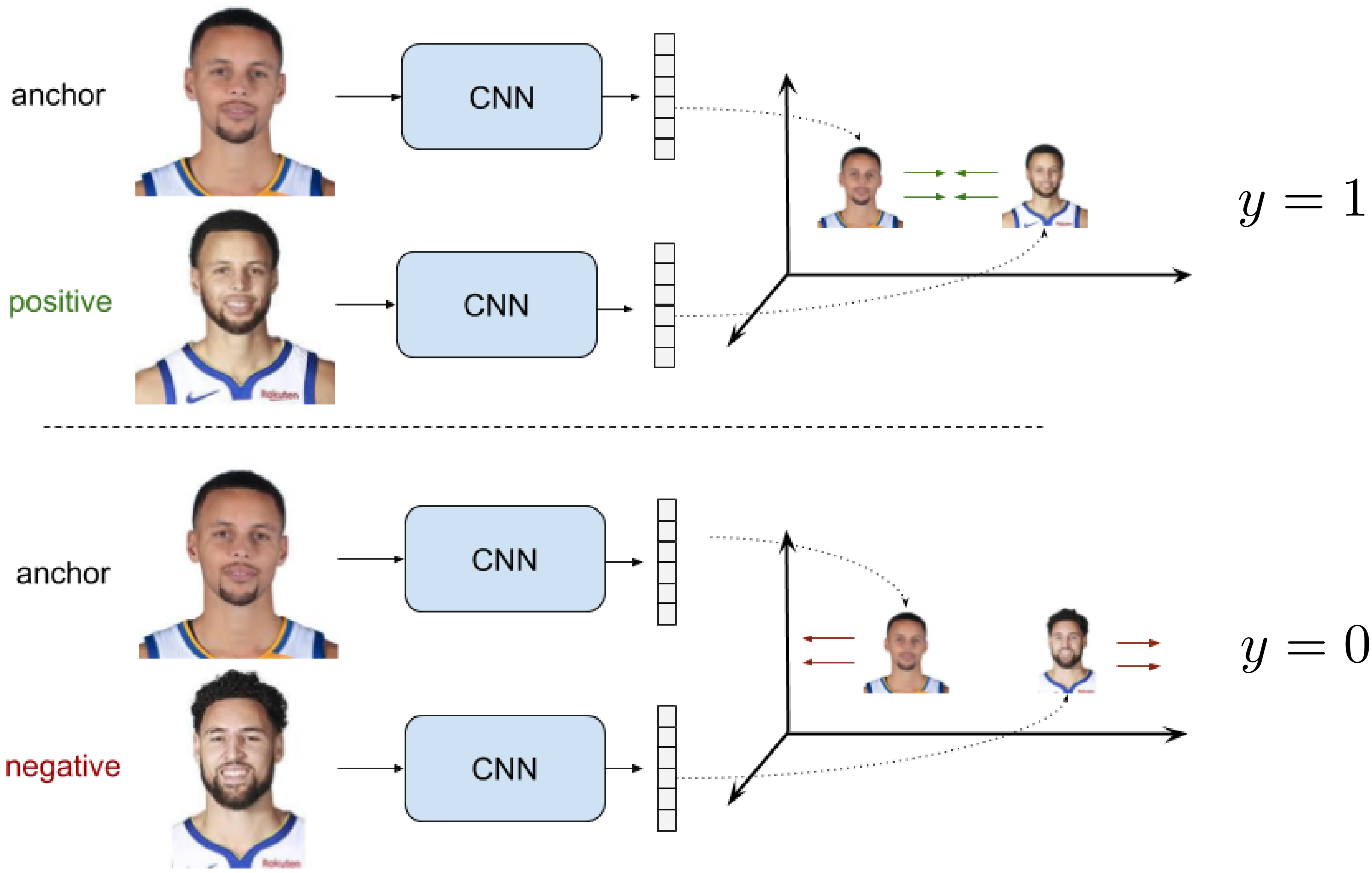
- Metric learning problem





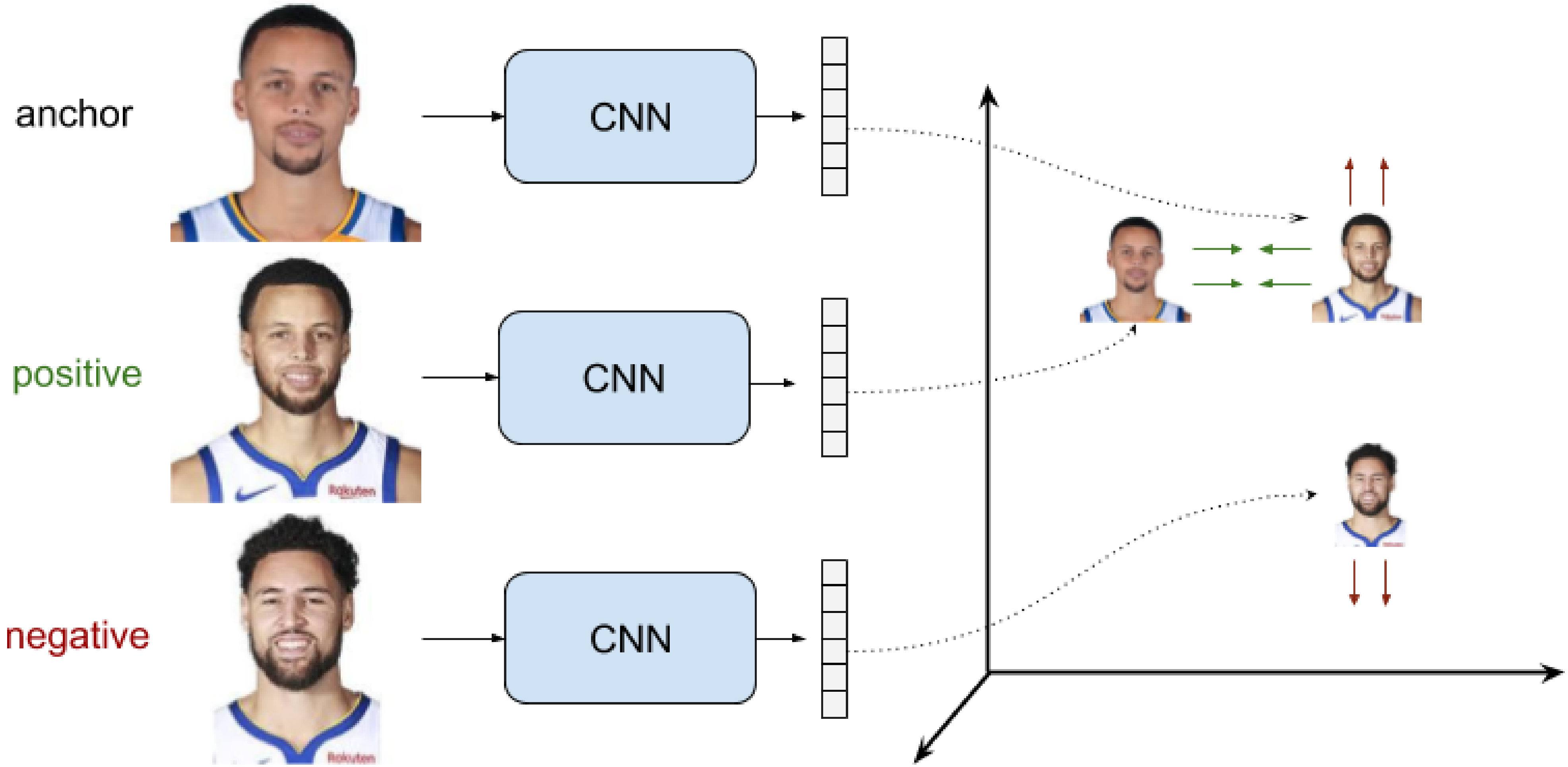

Training
Training Faces
Feature
Softmax classifier
Loss Function
Testing
Testing face pairs
Similarity
Discriminant
Same person
or not

Contrastive Loss



$$L(A, PN, y) = yd(A, PN) + (1 - y)\max(0, m - d(A, PN))$$

Triplet Loss



$$L(A, P, N) = \max(0, m + d(A, P) - d(A, N))$$

- Softmax

$$-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$$

x_i denotes the deep feature of the i-th sample, belonging to the y_i-th class.
W_j^T denotes the j-th column of the weight W and b_j is the bias term.
The batch size and the class number are N and n, respectively.

- ArcFace

$$-\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s*(\cos(\theta_{y_i}+m))}}{e^{s*(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n} e^{s*cos\theta_j}}$$

where θ_j is the angle between the weight W_j and the feature x_i
s - feature scale, the hypersphere radius
m - angular margin penalty