Deep Learning for various CV tasks

Computer Vision (SJK02)

Universitat Jaume I

Part A:

Classification

Segmentation

Object detection

(Image-based) biometrics

Part B:

Sequence processing

Optical flow

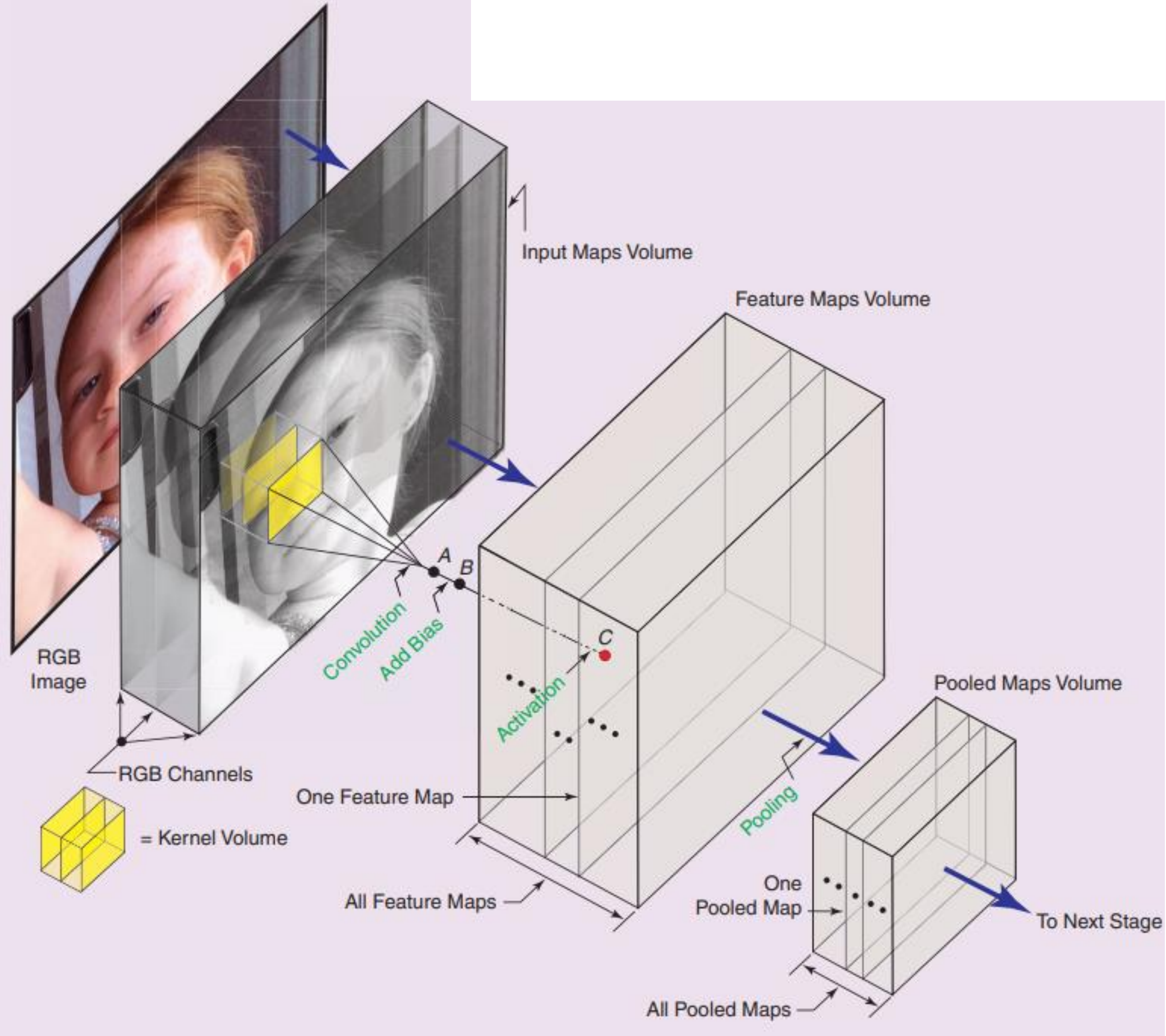
Action Recognition

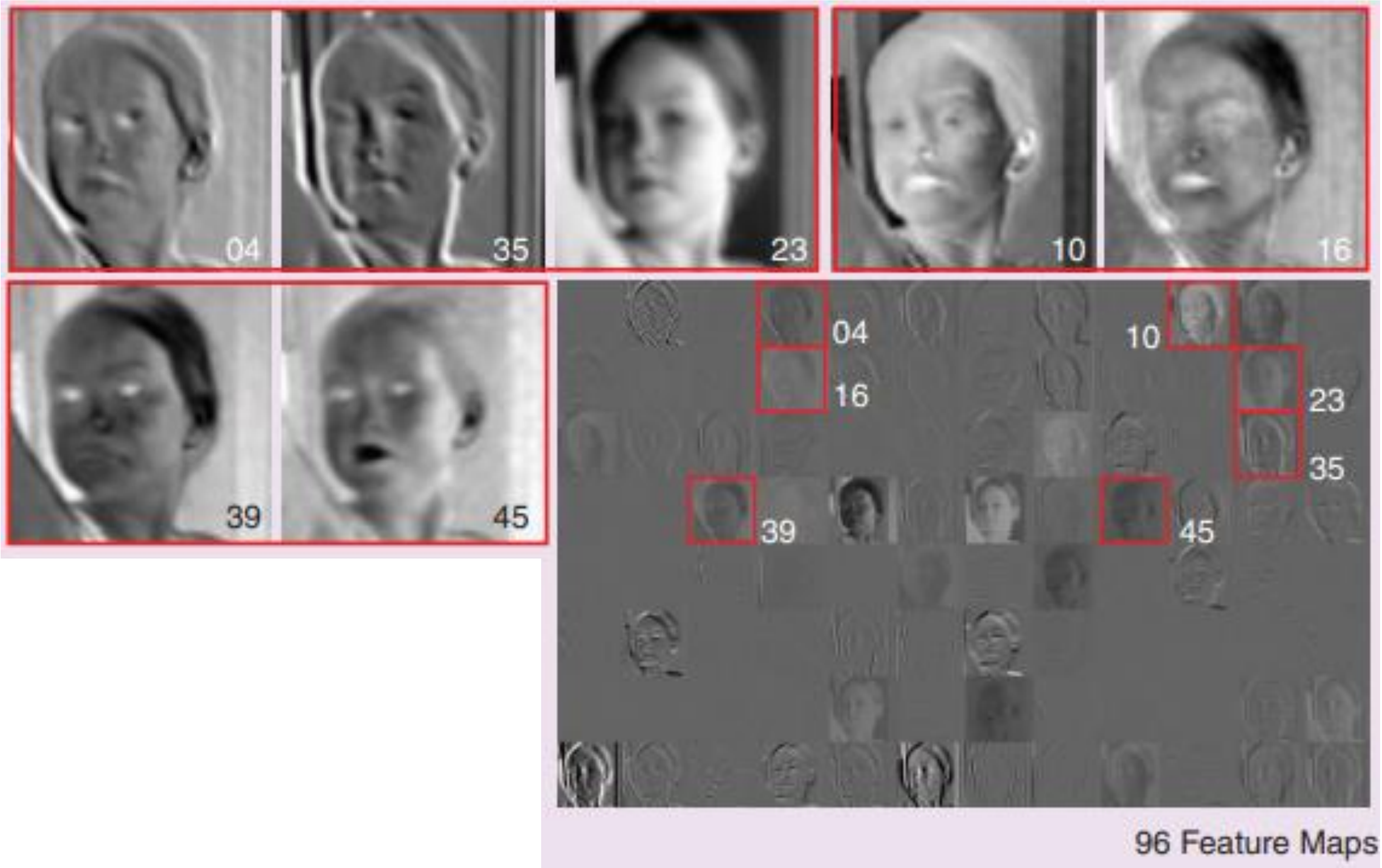
Self-supervised learning

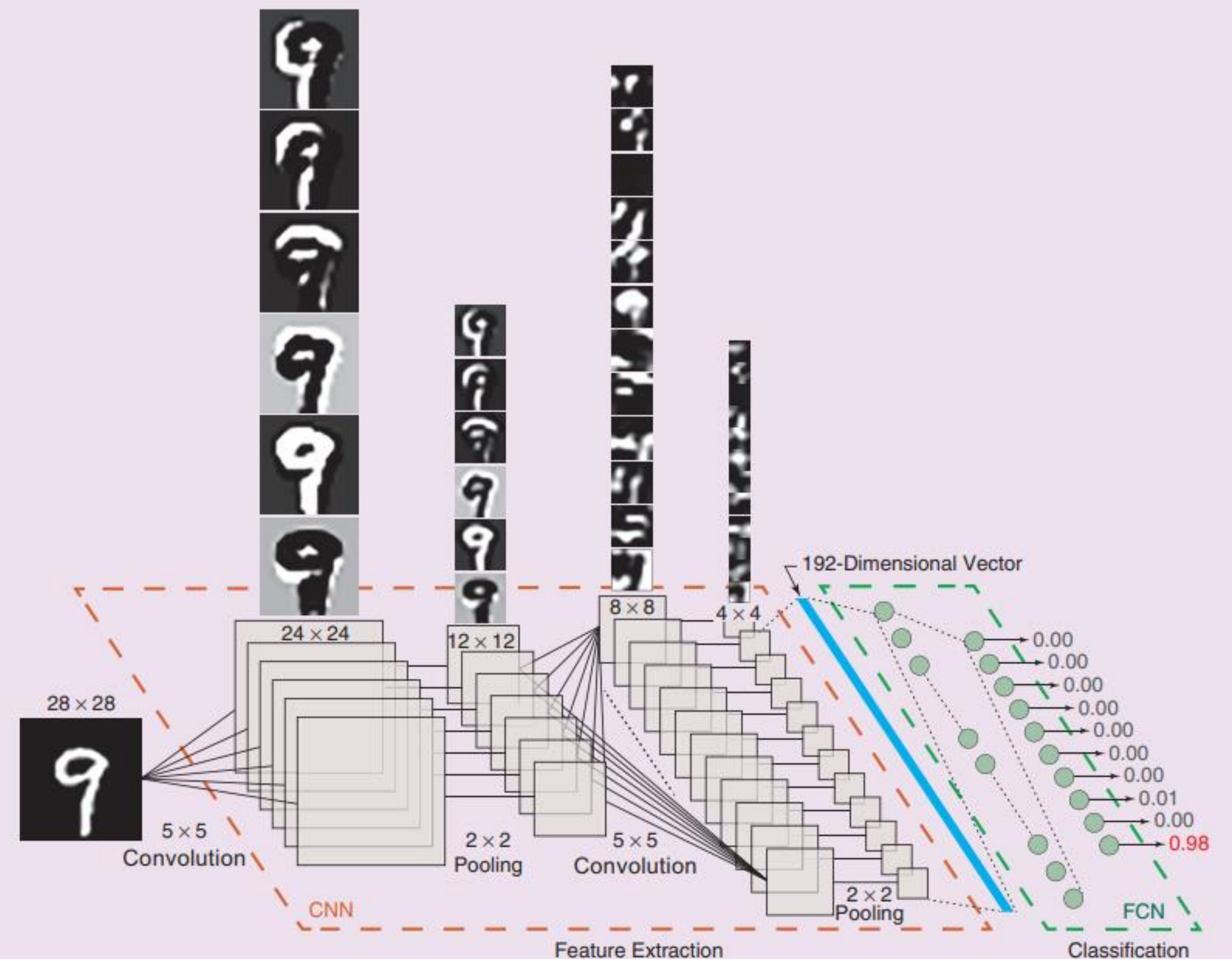
Transformers

Convolutional neural networks (CNNs)
~A quick review~

Deep Convolutional Neural Networks [Lecture Notes]

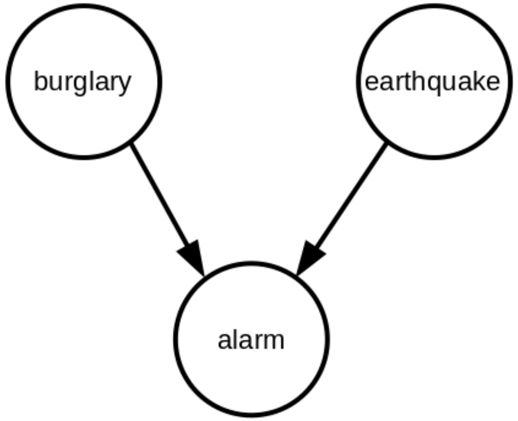

Input Maps Volume
Feature Maps Volume
A
B
C
Convolution
Add Bias
Activation
RGB
Image
RGB Channels
= Kernel Volume
One Feature Map
All Feature Maps
Pooling
Pooled Maps Volume
One
Pooled Map
To Next Stage
All Pooled Maps


04
35
23
10
16
39
45
04
10
16
23
35
39
45
96 Feature Maps

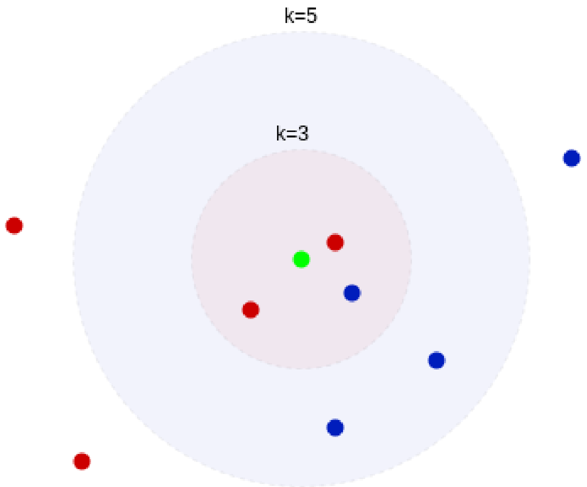

28 × 28
5 × 5
Convolution
24 × 24
12 × 12
2 × 2
Pooling
8 × 8
5 × 5
Convolution
4 × 4
2 × 2
Pooling
192-Dimensional Vector
0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.01
0.00
0.98
CNN
FCN
Feature Extraction
Classification

Inductive bias

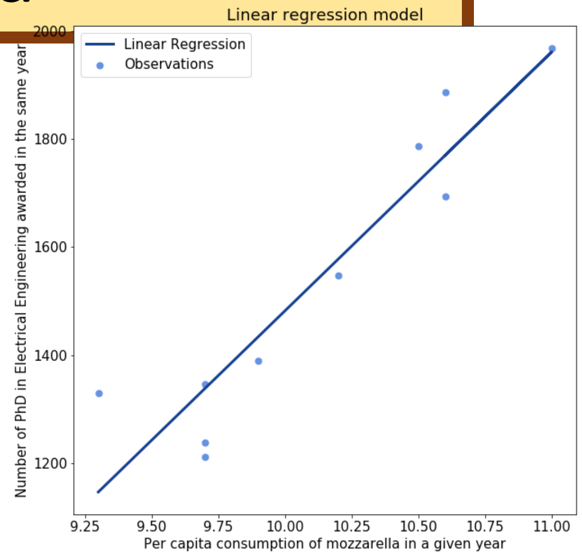
- Beliefs/assumptions made by ML models
- Constraints in hypotheses space
- May help generalise to unseen data



- Priors
- Structure



Similar data are closer

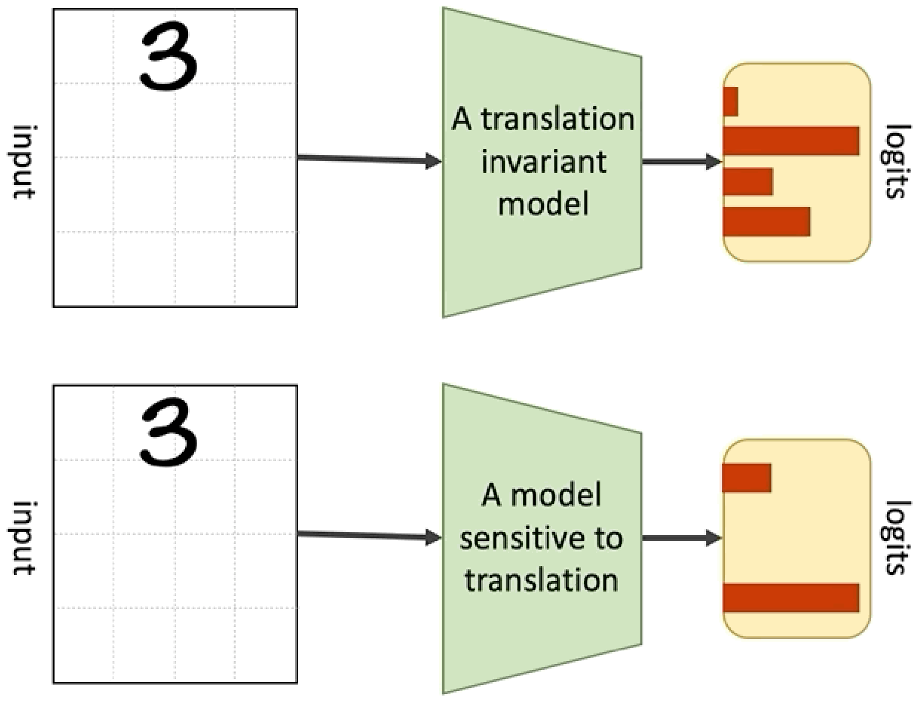


Linear dependence

https://www.baeldung.com/cs/ml-inductive-bias

Inductive bias in CNNs

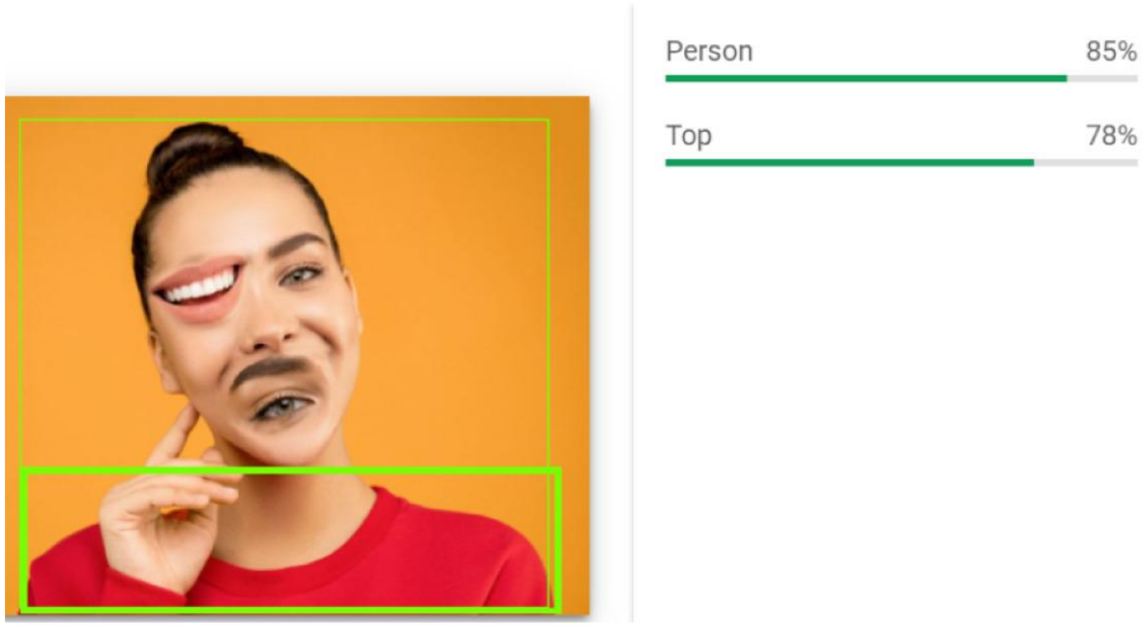
- 2D spatial relations (locality)
- Translation invariance (convolution + max pooling)



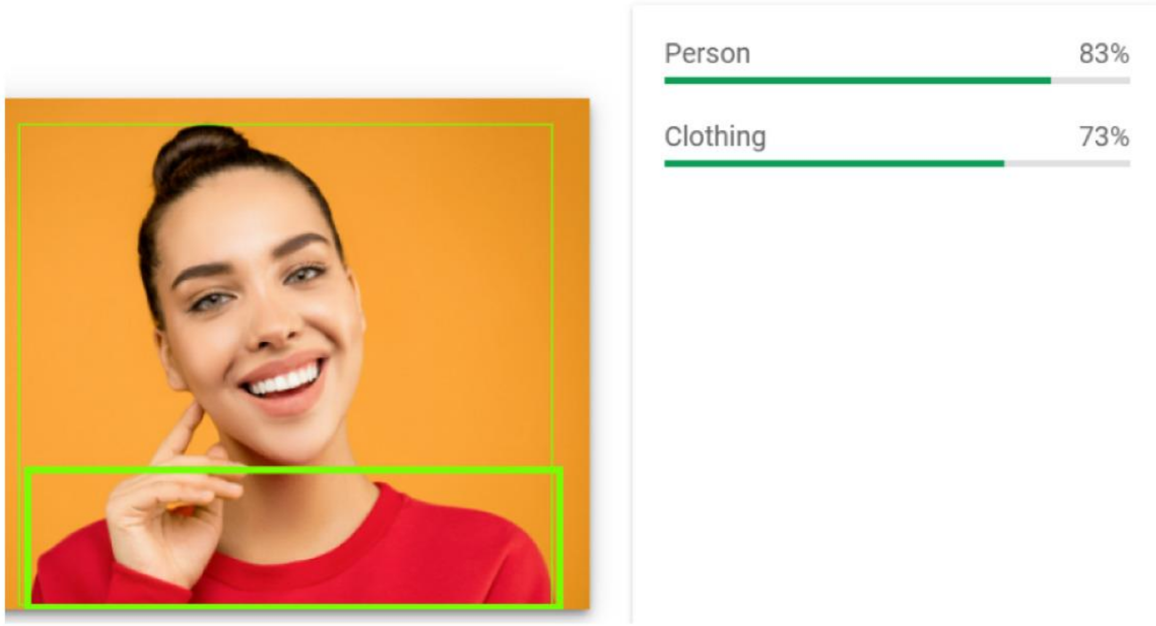
Are CNNs rotation invariant?

https://samiraabnar.github.io/articles/2020-05/indist

Picasso effect



(a) Distorted Face



(b) Real Face

[Combining neural and symbolic approaches to solve the Picasso problem: A first step]() (Displays 2022)

Classification: AlexNet, VGG, Inception, ResNet

https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

LeNet (1998)

AlexNet (2012)

- 5 conv, 3 fc, 60 M params

GoogLeNet / Inception (2014)

VGG (2014)

- 16 layers, 96 M params

ResNet (2015)

- 152 layers, 1 M params

MobileNet (2017)

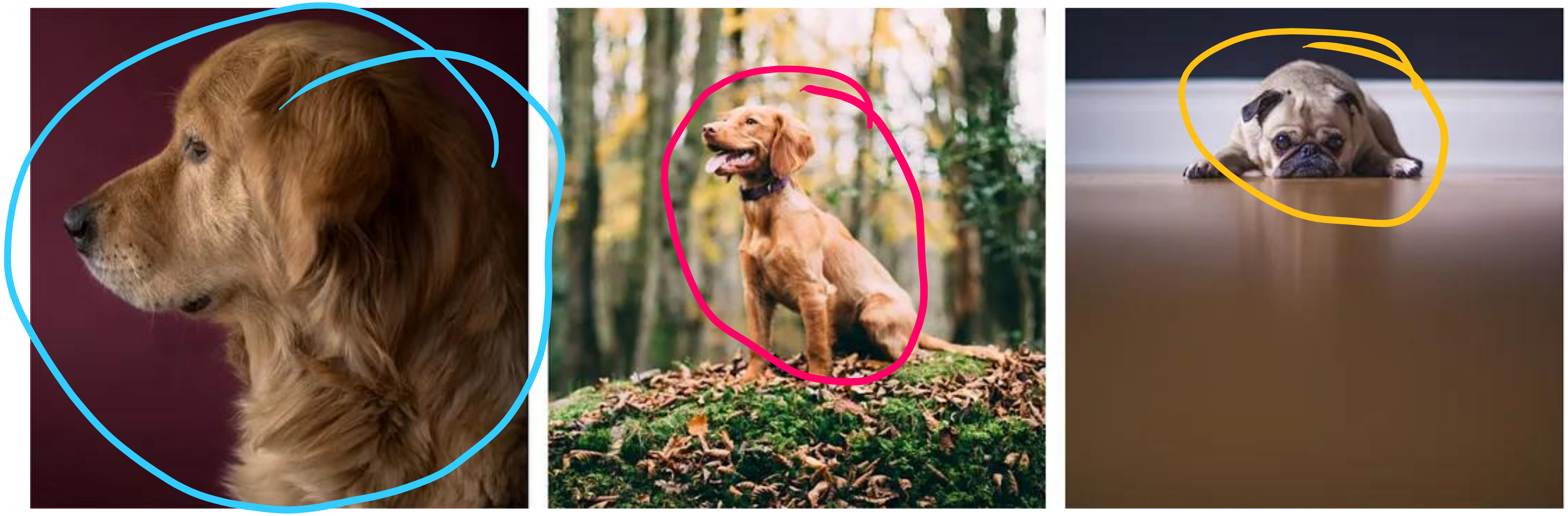
- A few hundred layers, fit on smart phones

EfficientNet (2019)

- Systematic balance between net depth, width, and resolution

GoogLeNet (Inception v1)

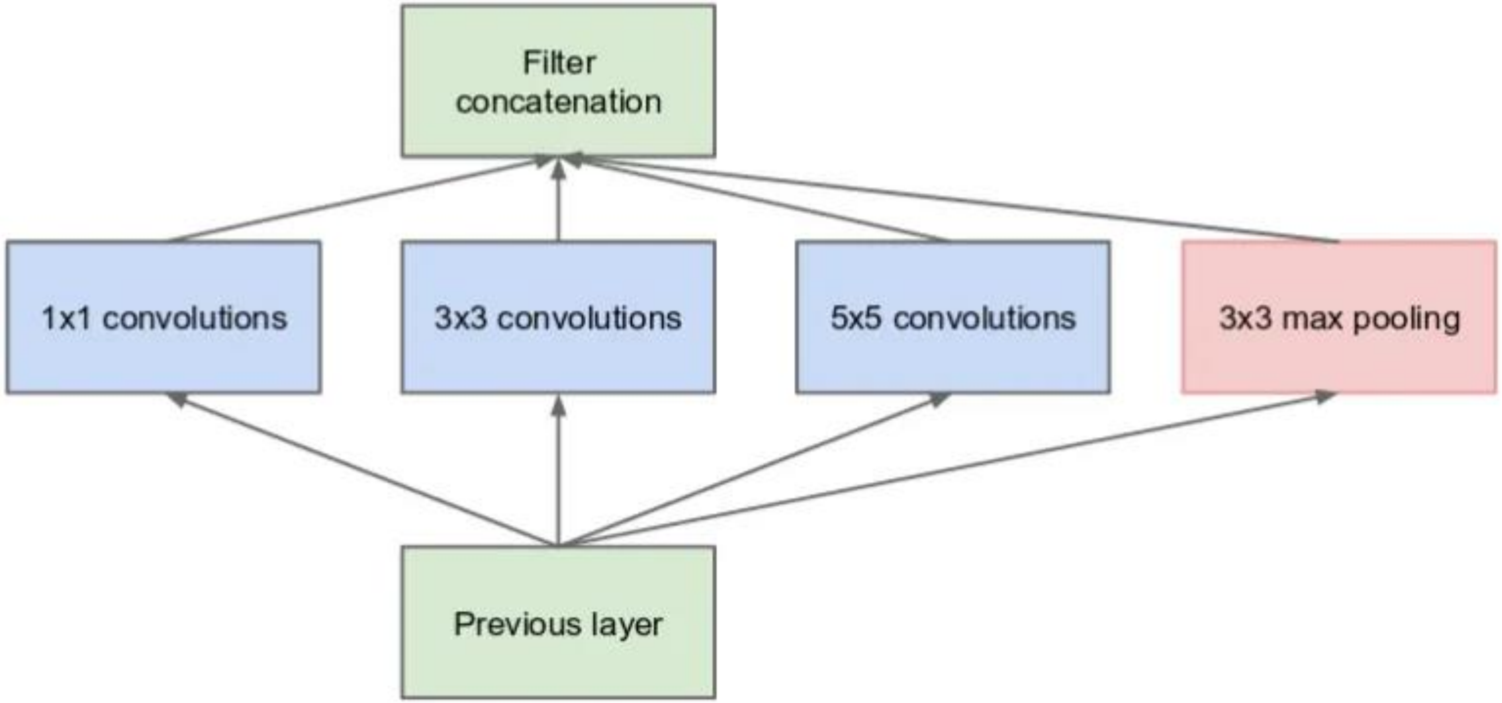
Simple stacking layers is costly and may not be so effective
What about regions of interest of very different sizes?



BIG MED SMALL

https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202

Inception module

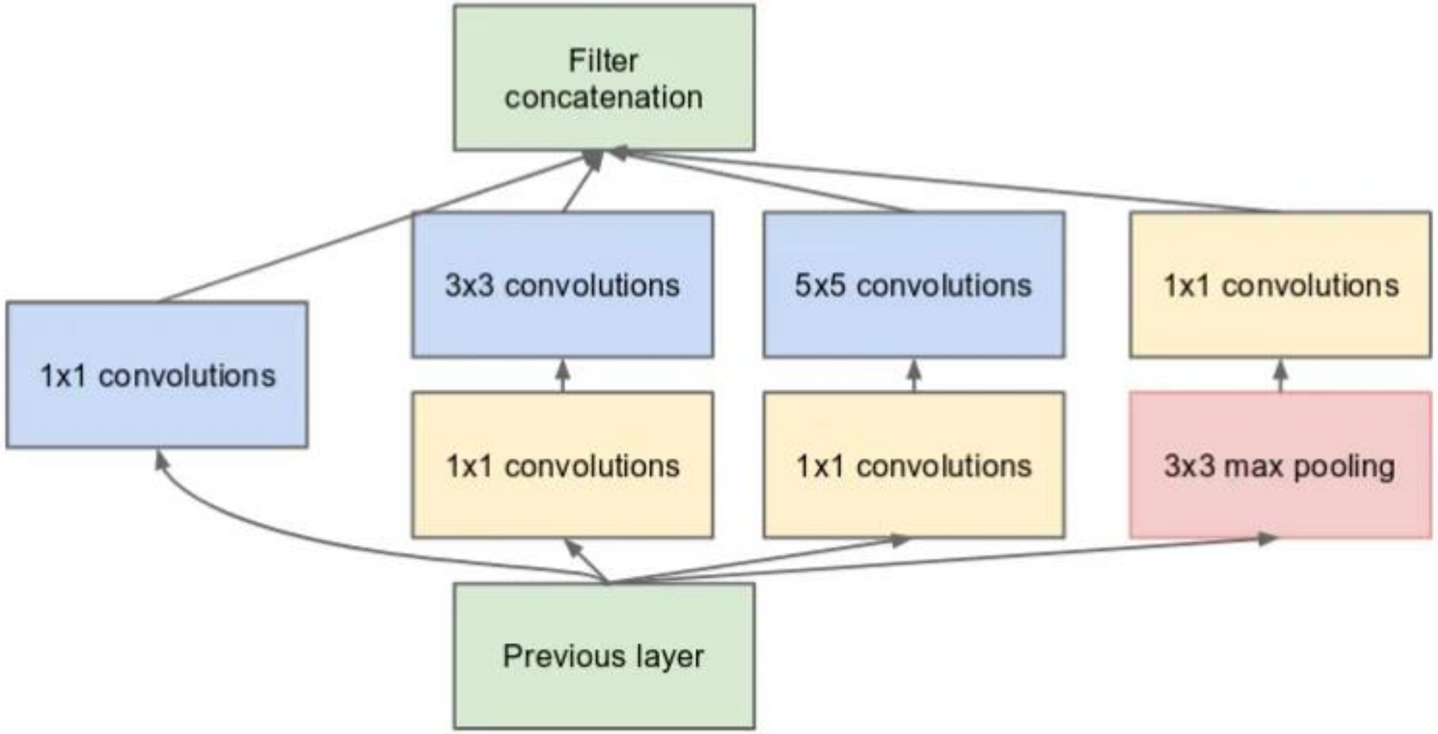


(a) Inception module, naïve version

Net can be wider rather than deeper

1x1 convolutions

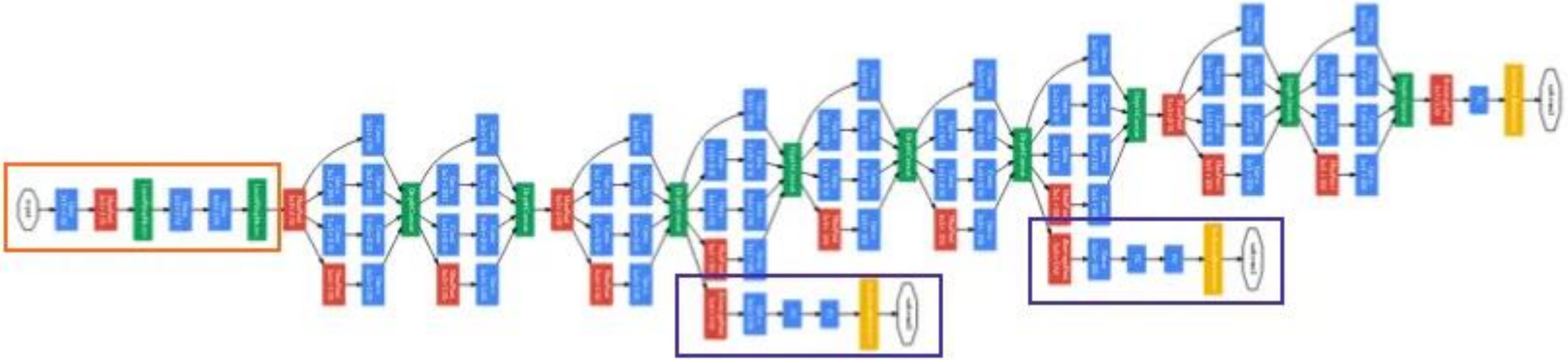
Computation grow with # channels
1x1 filters can change (reduce) # channels



(b) Inception module with dimension reductions

1x1 filter changes #channels, but...
How can width x height of activation maps be changed?

~22 layers



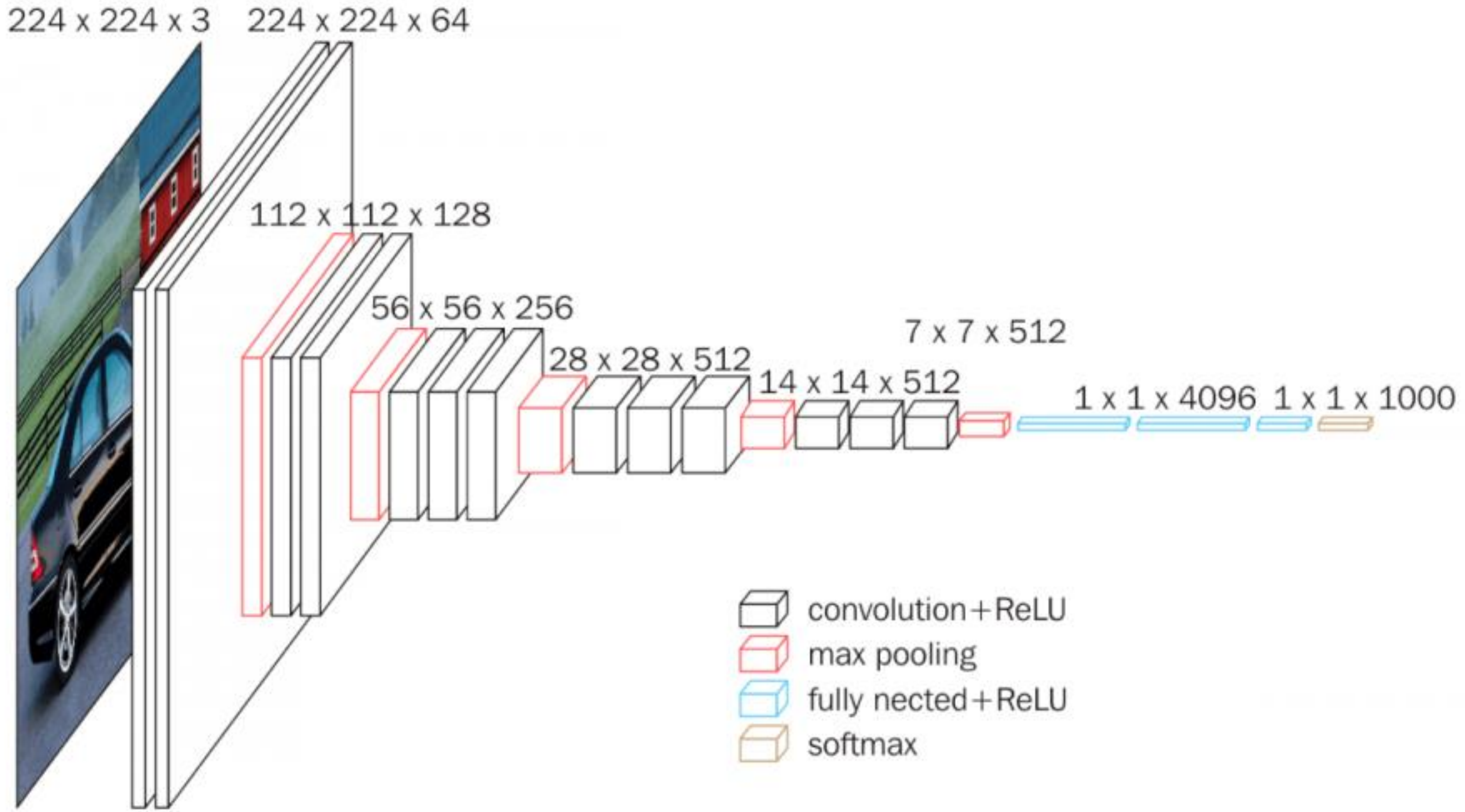
How many inception modules does it have?

www.socrative.com
Room 219986

Other Inception versions

Inception v2: factorize convolutions (3x3 conv = 1x3 conv, 3x1 conv)
Inception v3: batchnorm, label smoothing
Inception v5: change of stem part

VGG



https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96

Very Deep Convolutional Networks for Large-Scale Image Recognition (ICLR 2015)

How many layers (with trainable weights) does VGGNet have?

How many object categories can it classify?

www.socrative.com
Room 219986

VGG16 - Structural Details													
#	**Input Image**			**output**			**Layer**	**Stride**	**Kernel**		**in**	**out**	**Param**
1	224	224	3	224	224	64	conv3-64	1	3	3	3	64	1792
2	224	224	64	224	224	64	conv3064	1	3	3	64	64	36928
	224	224	64	112	112	64	maxpool	2	2	2	64	64	0
3	112	112	64	112	112	128	conv3-128	1	3	3	64	128	73856
4	112	112	128	112	112	128	conv3-128	1	3	3	128	128	147584
	112	112	128	56	56	128	maxpool	2	2	2	128	128	65664
5	56	56	128	56	56	256	conv3-256	1	3	3	128	256	295168
6	56	56	256	56	56	256	conv3-256	1	3	3	256	256	590080
7	56	56	256	56	56	256	conv3-256	1	3	3	256	256	590080
	56	56	256	28	28	256	maxpool	2	2	2	256	256	0
8	28	28	256	28	28	512	conv3-512	1	3	3	256	512	1180160
9	28	28	512	28	28	512	conv3-512	1	3	3	512	512	2359808
10	28	28	512	28	28	512	conv3-512	1	3	3	512	512	2359808
	28	28	512	14	14	512	maxpool	2	2	2	512	512	0
11	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
12	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
13	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
	14	14	512	7	7	512	maxpool	2	2	2	512	512	0
14	1	1	25088	1	1	4096	fc		1	1	25088	4096	102764544
15	1	1	4096	1	1	4096	fc		1	1	4096	4096	16781312
16	1	1	4096	1	1	1000	fc		1	1	4096	1000	4097000
Total													**138,423,208**

Kernel of size 3x3 for all conv layers

- Unlike previous nets with 5x5, 7x7, 11x11

How many trainable weights are required...?

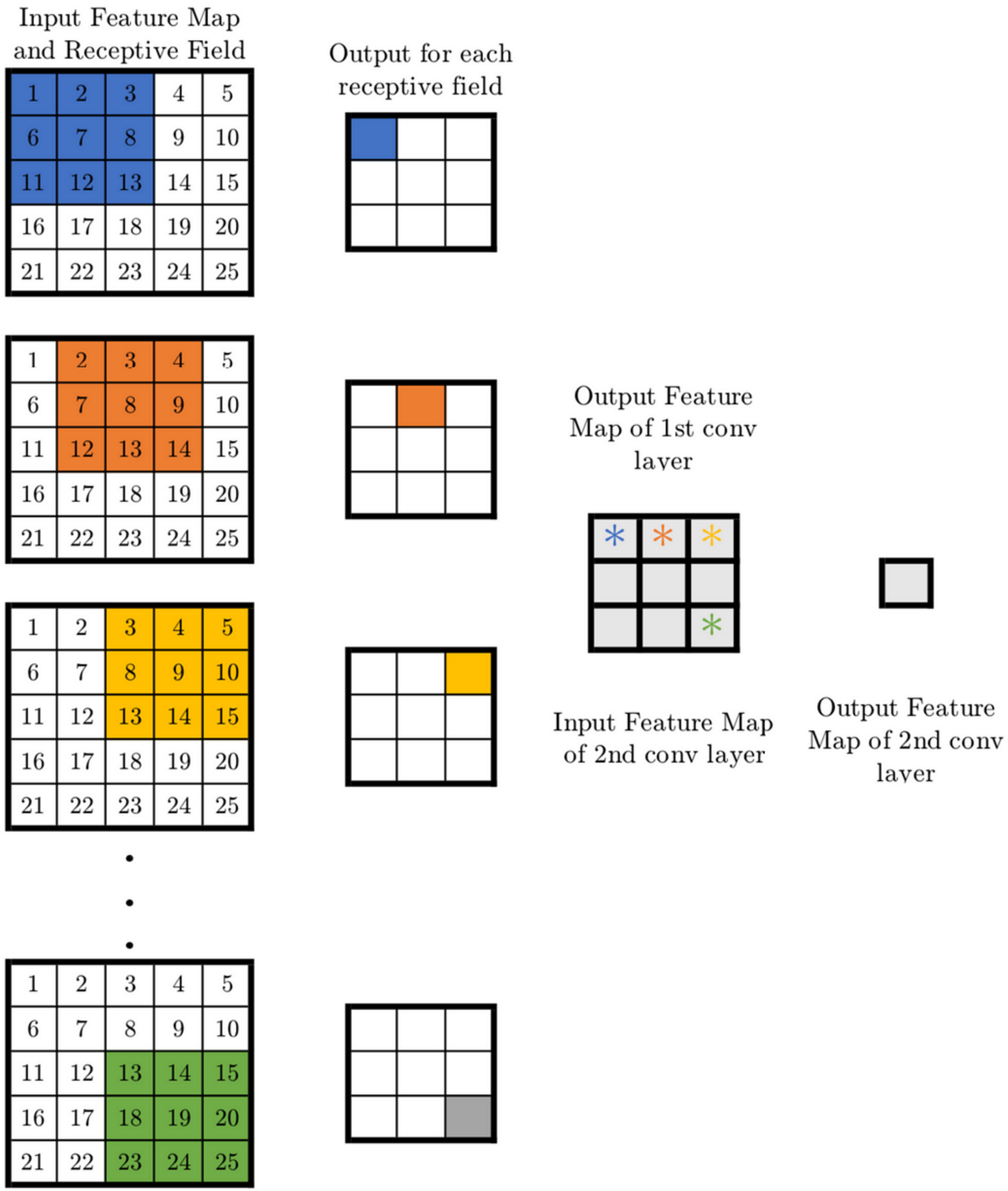
- Input feature map of 100x100x1
- conv layer with 1 filter of size 5x5

And with two conv layers, each with 3x3 filters?

The same effective "receptive field" can be achieved with **more layers of smaller filters,** resulting in less parameters

What's the benefit of less parameters?

- Faster training
- Less prone to overfitting
- Less memory footprint


Input Feature Map and Receptive Field
Output for each receptive field
Output Feature Map of 1st conv layer
Input Feature Map of 2nd conv layer
Output Feature Map of 2nd conv layer

ResNet

Many layers --> vanishing gradient in backpropagation

Innovative solution: **skip connections**

https://datagen.tech/guides/computer-vision/resnet/

Residual block

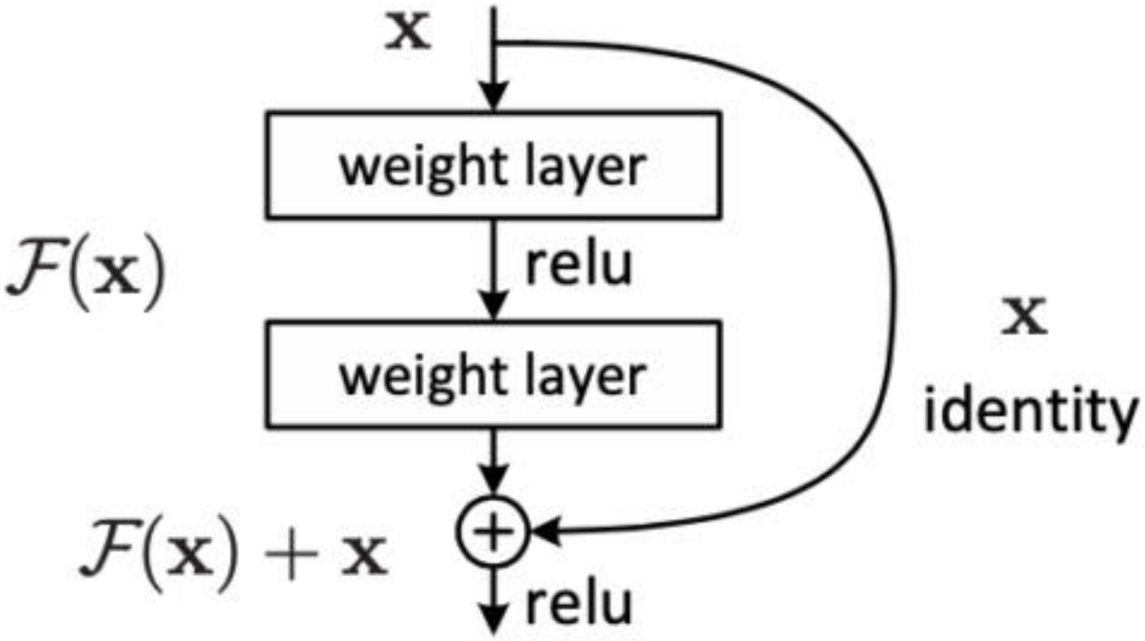
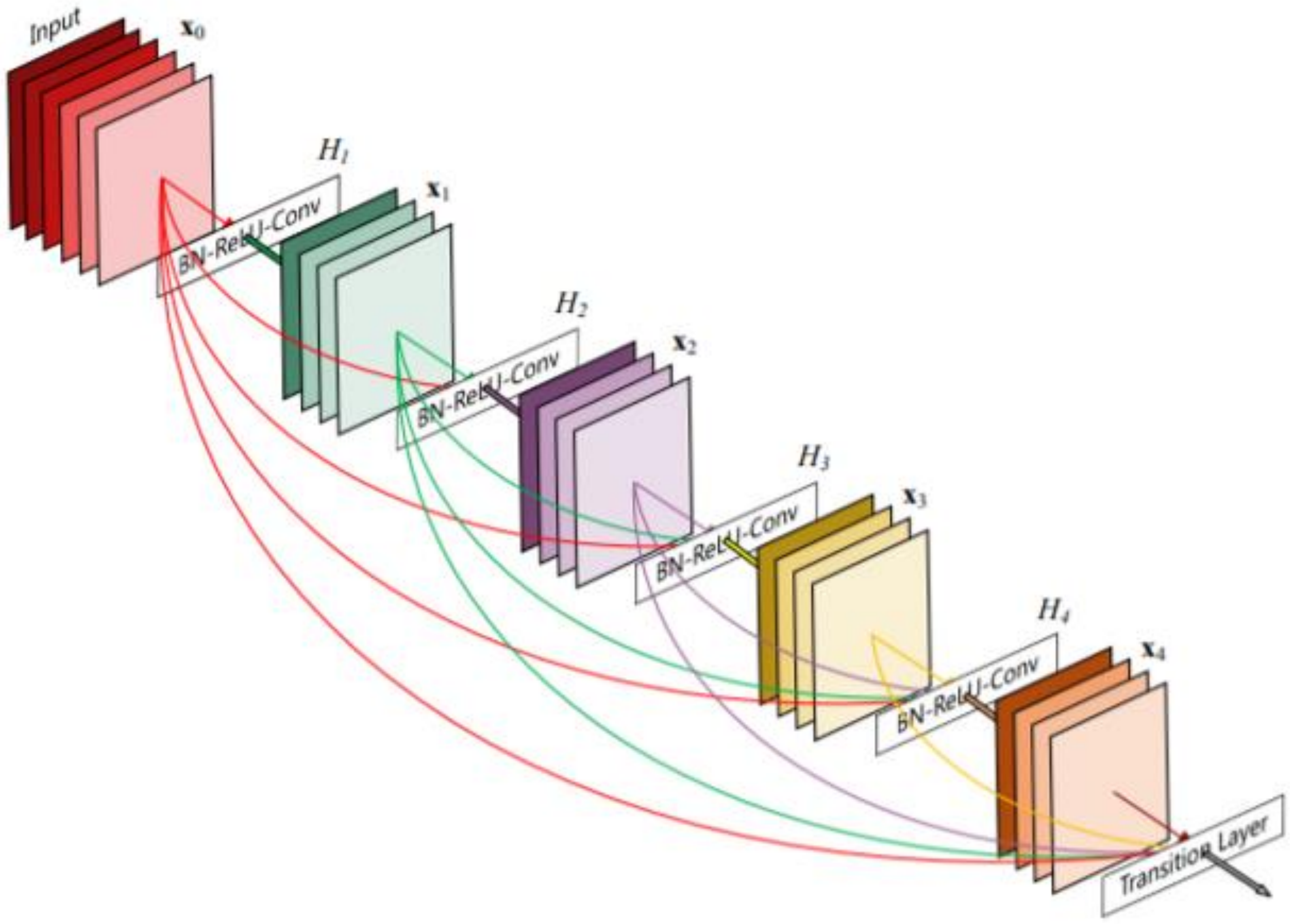


Figure 2. Residual learning: a building block.

DenseNet (variant of ResNet)



EfficientNet

How to scale up models?

- Depth-wise?
- Width-wise?
- Higher resolution?

How to decide?

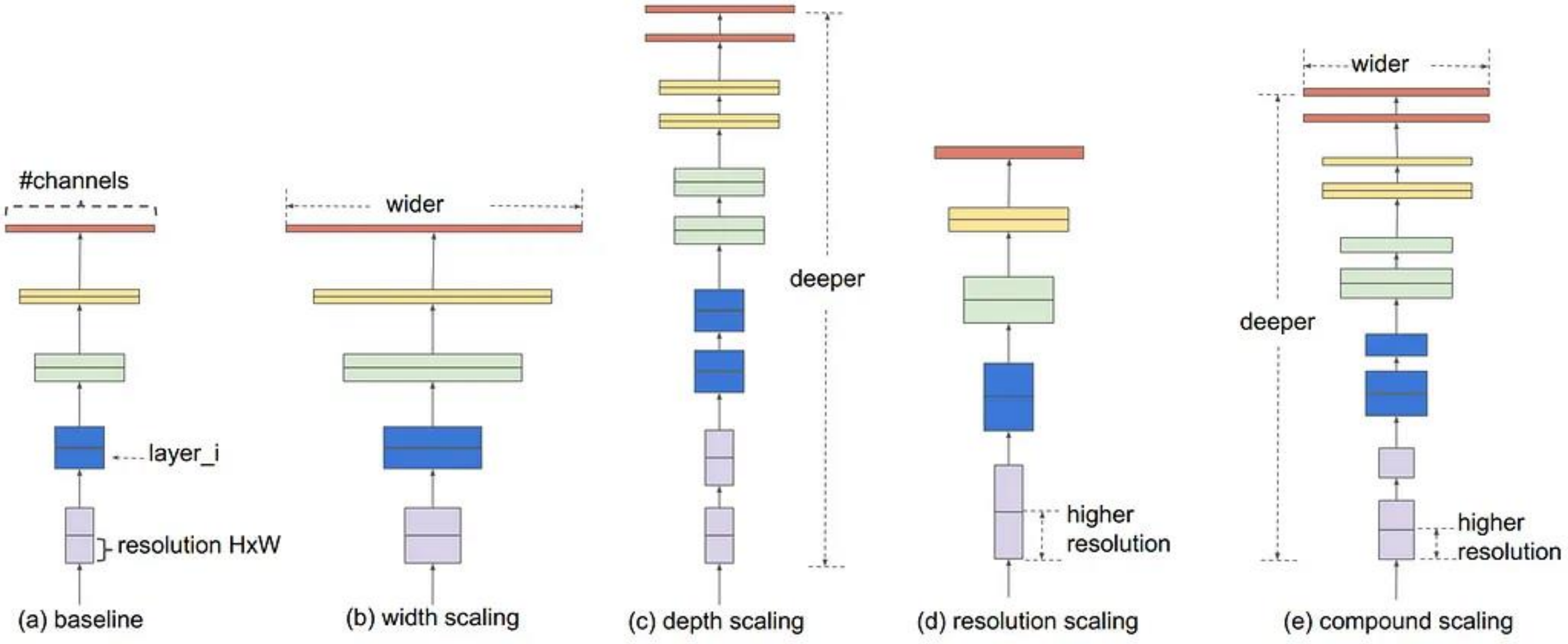
- Much human effort
- Lot of manual tunning

Can we do better?

https://medium.com/mlearning-ai/understanding-efficientnet-the-most-powerful-cnn-architecture-eaeb40386fad

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (ICML 2019)

Different scaling methods



Balancing the scale in all the three dimensions improves the overall model performance

General idea

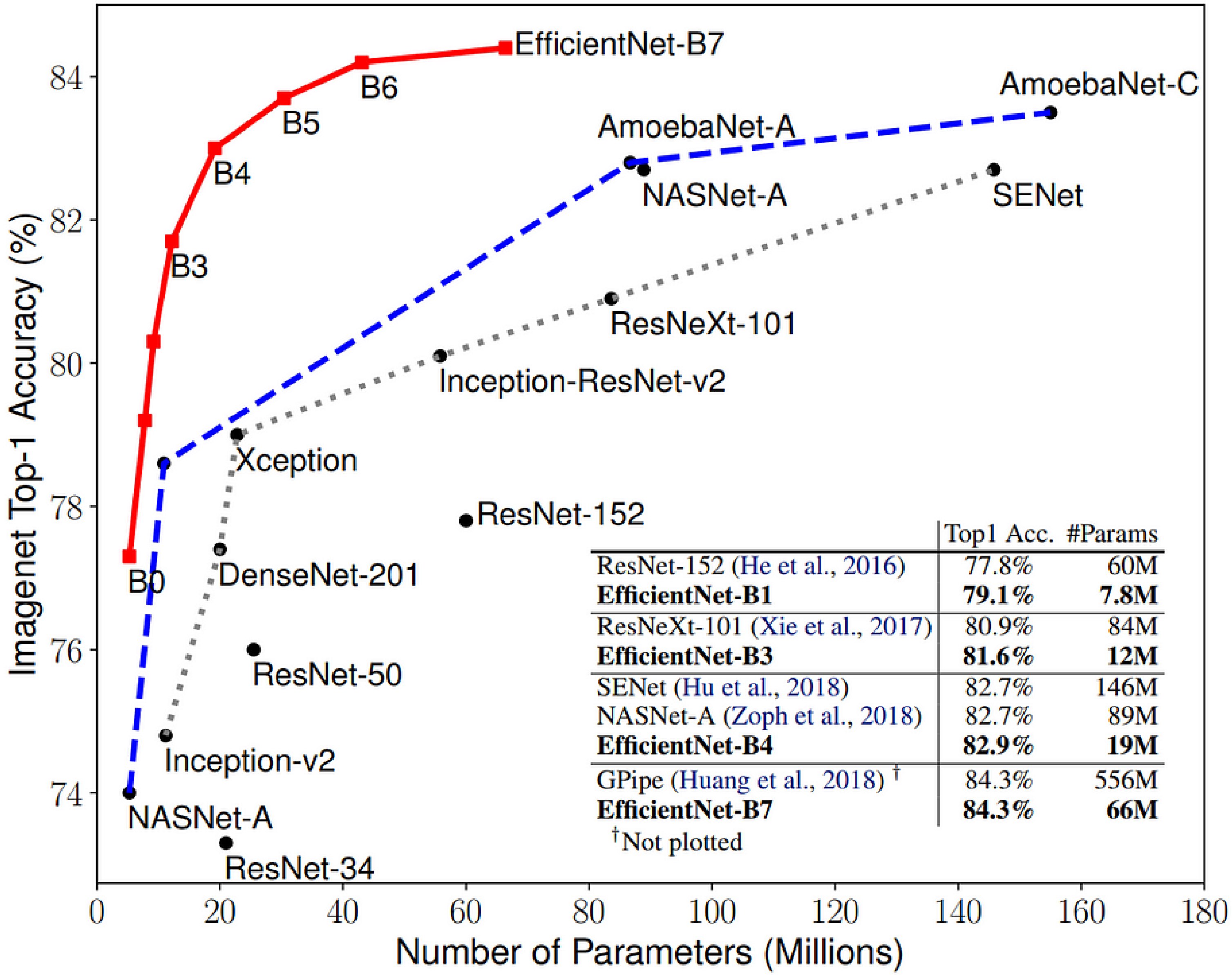
Architecture search (AutoML)

Optimize for...

- Max accuracy
- Penalize computational requirements
- Penalize slow inference

Family of EfficientNets

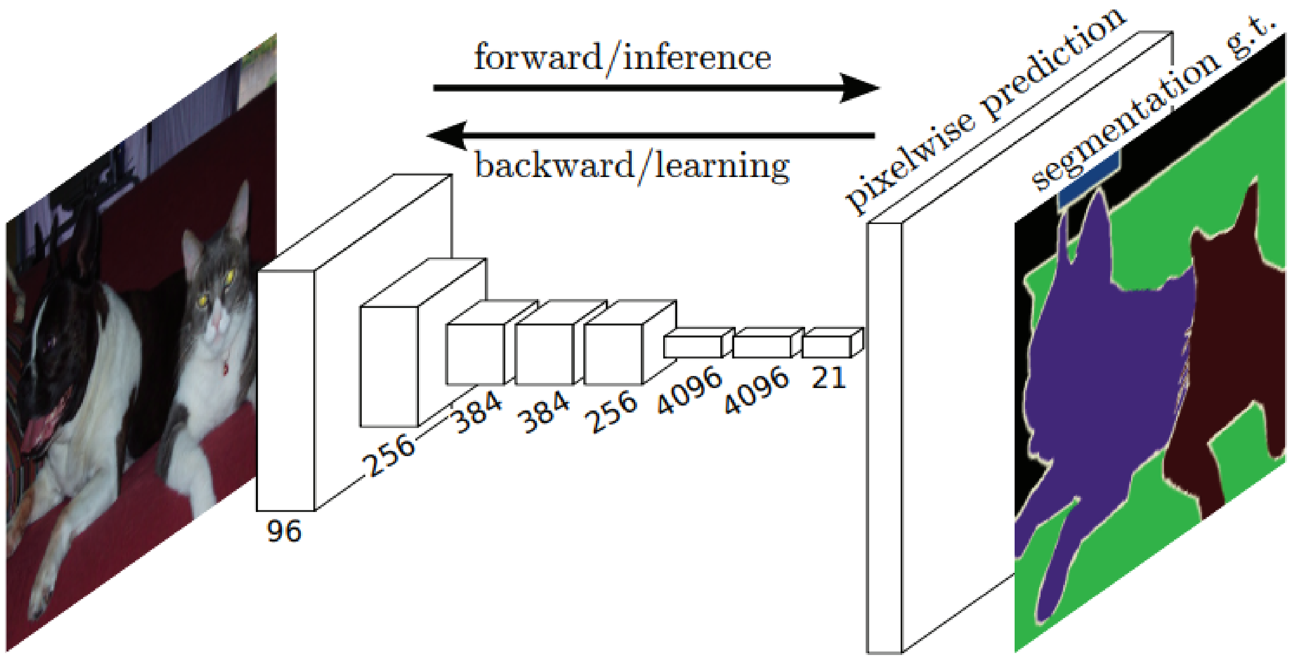
Performance comparison



Segmentation: UNet

[U-Net: Convolutional Networks for Biomedical Image Segmentation]() (MICCAI 2015)

CNN vs Fully Convolutional Network (FCN)

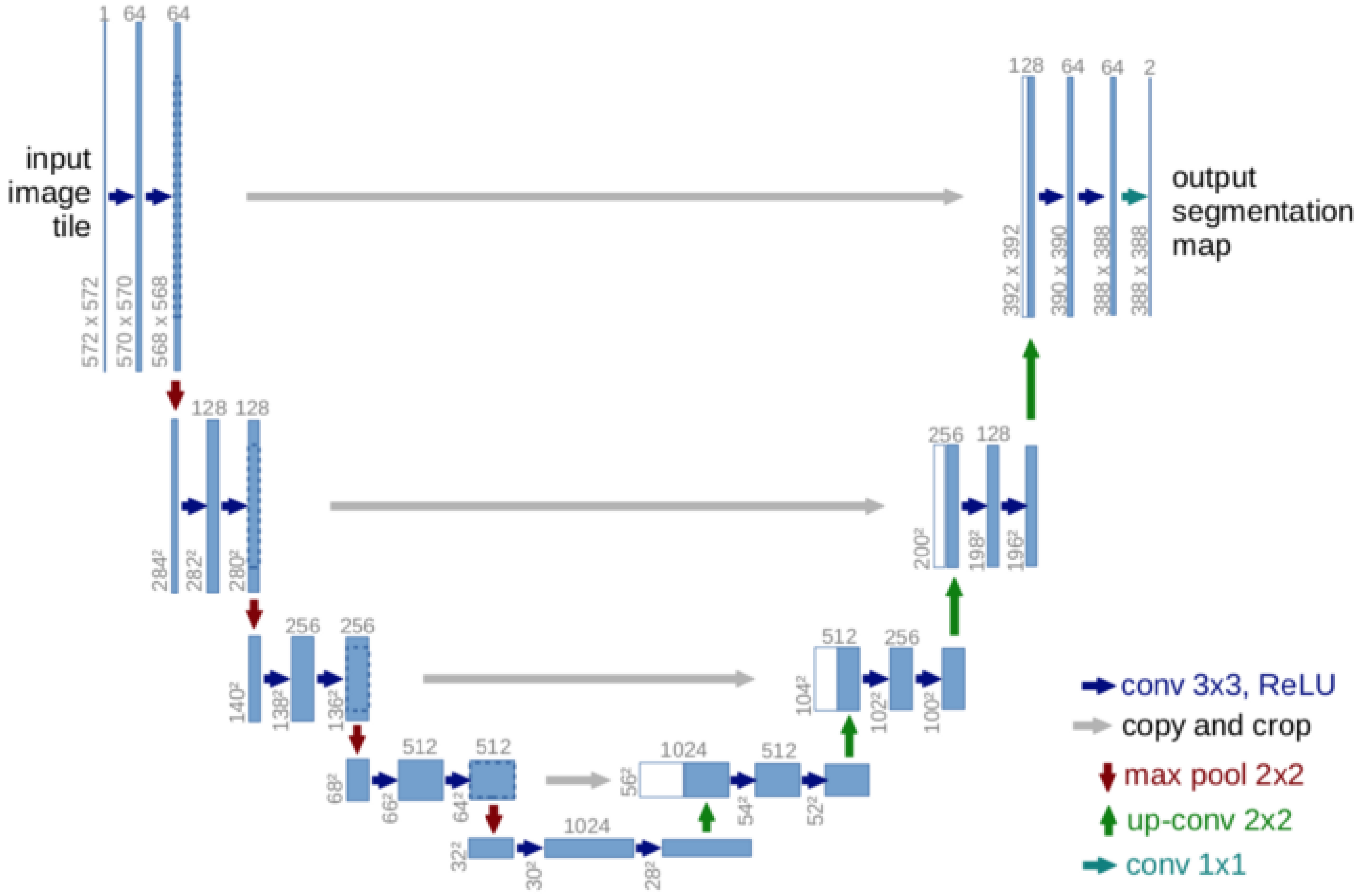


CNN for *global* classification:
one value per input image

FCN: one value per pixel
(e.g. pixel-level classification)

Fully Convolutional Networks for Semantic Segmentation (PAMI 2017)

Architecture



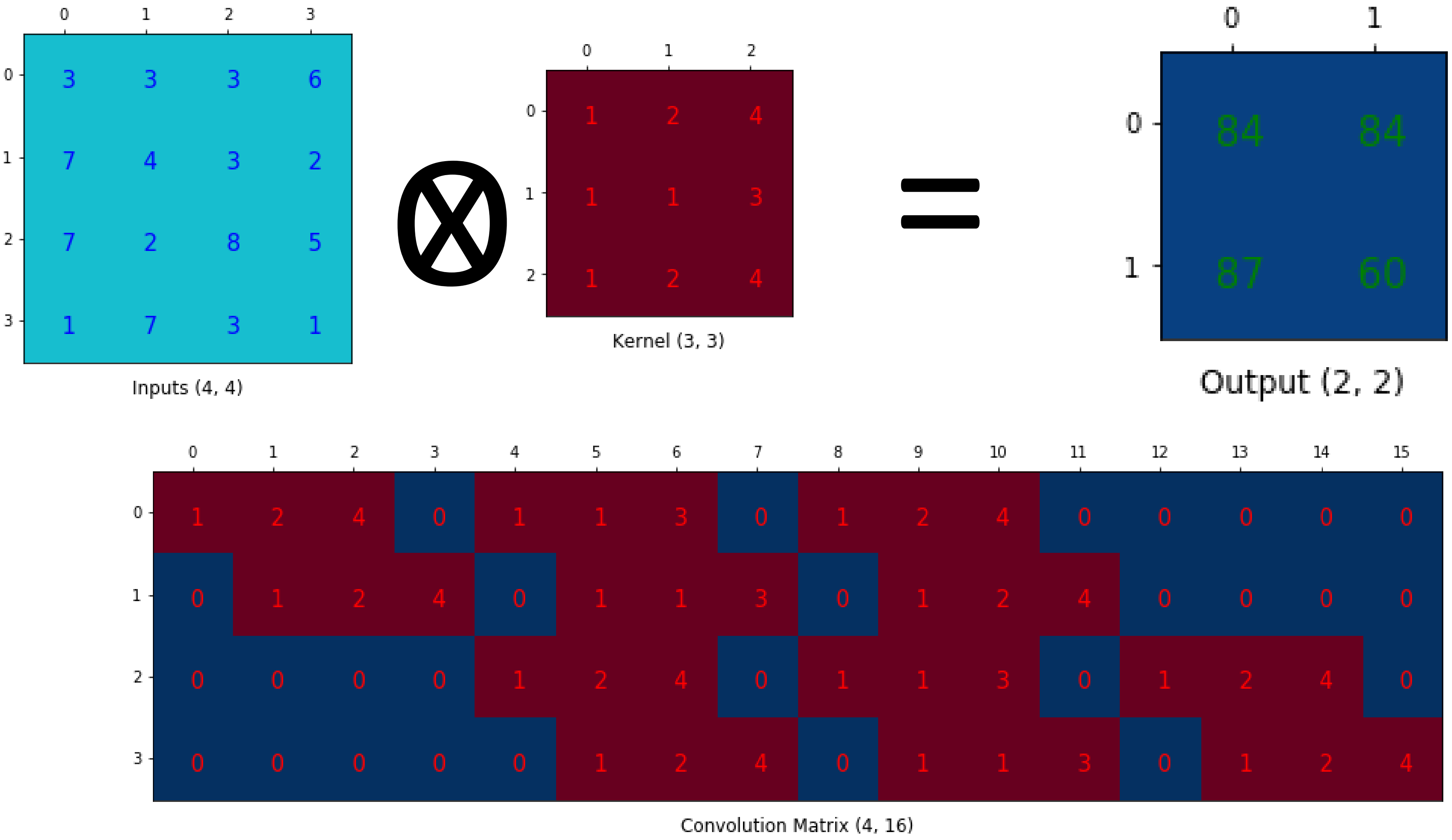
contracting part **expansive part**

How to upsample?

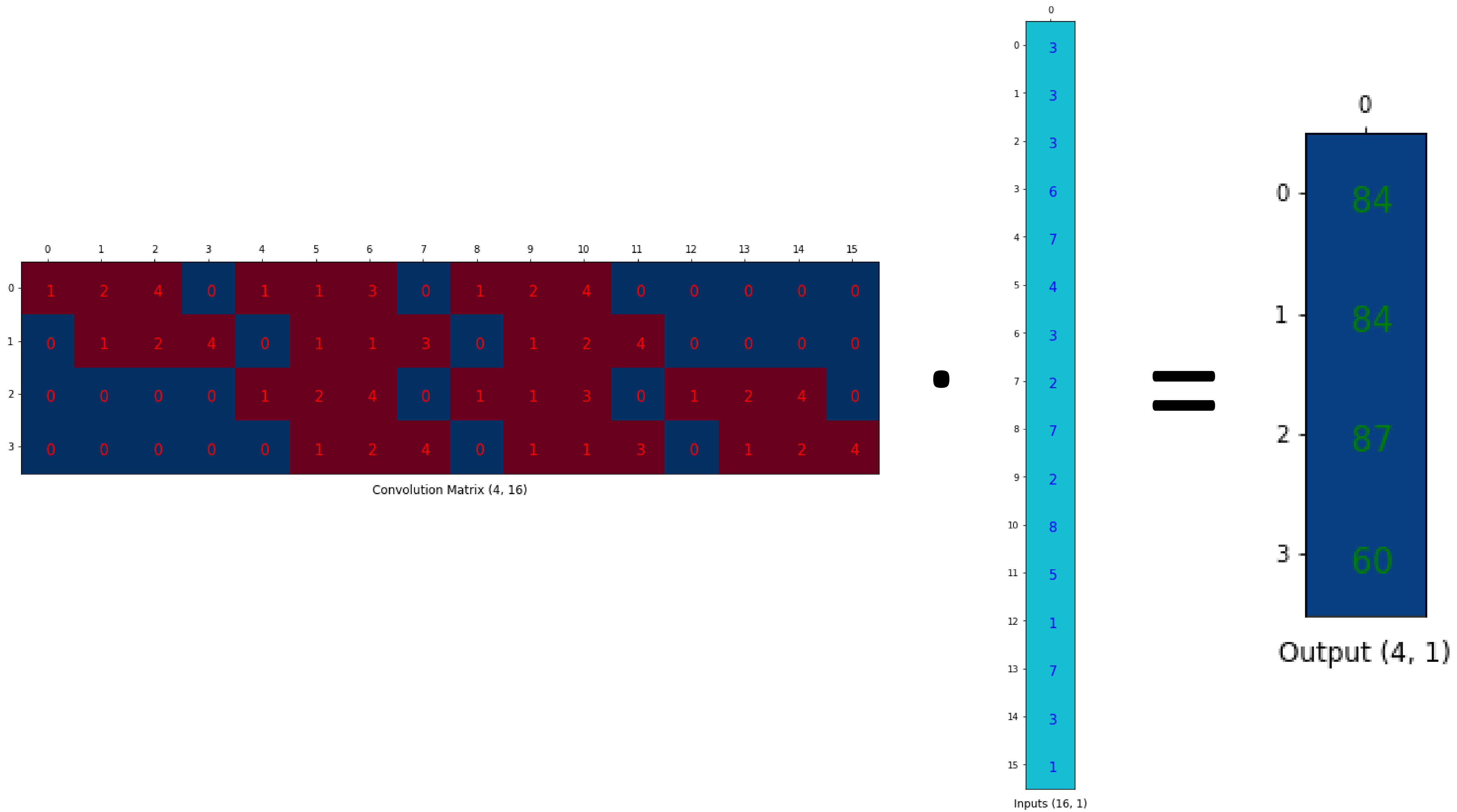
Interpolation
- Nearest neighbor
- Bi-linear
- Bi-cubic

These are fixed, not learnable!
Can we do better?

Transposed convolutions



https://github.com/naokishibuya/deep-learning/blob/master/python/transposed_convolution.ipynb


Convolution Matrix (4, 16)
Inputs (16, 1)
Output (4, 1)

$$
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 0 & 0 \\
2 & 4 & 2 & 0 & 0 \\
3 & 0 & 4 & 0 & 0 \\
4 & 1 & 0 & 1 & 0 \\
5 & 1 & 1 & 2 & 1 \\
6 & 3 & 1 & 4 & 2 \\
7 & 0 & 3 & 0 & 4 \\
8 & 1 & 0 & 1 & 0 \\
9 & 2 & 1 & 1 & 1 \\
10 & 4 & 2 & 3 & 1 \\
11 & 0 & 4 & 0 & 3 \\
12 & 0 & 0 & 1 & 0 \\
13 & 0 & 0 & 2 & 1 \\
14 & 0 & 0 & 4 & 2 \\
15 & 0 & 0 & 0 & 4
\end{array}
\cdot
\begin{array}{c|c}
 & 0 \\
\hline
0 & 2 \\
1 & 3 \\
2 & 4 \\
3 & 3
\end{array}
=
\begin{array}{c|c}
 & 0 \\
\hline
0 & 2 \\
1 & 7 \\
2 & 14 \\
3 & 12 \\
4 & 6 \\
5 & 16 \\
6 & 31 \\
7 & 21 \\
8 & 6 \\
9 & 14 \\
10 & 29 \\
11 & 21 \\
12 & 4 \\
13 & 11 \\
14 & 22 \\
15 & 12
\end{array}
\qquad
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & 2 & 7 & 14 & 12 \\
1 & 6 & 16 & 31 & 21 \\
2 & 6 & 14 & 29 & 21 \\
3 & 4 & 11 & 22 & 12
\end{array}
$$

Transposed Convolution Matrix (16, 4)

Inputs (4, 1)

Output (16, 1)

Output (4, 4)

Skip connections: why?

Combines

- Rich **semantic** features of deep layers
- Finer **localisation** of contracting layers

Scarce training data

Data **augmentation**

- Shifts and rotations
- Gray-level transformations
- Elastic deformations

Region borders are tricky to segment

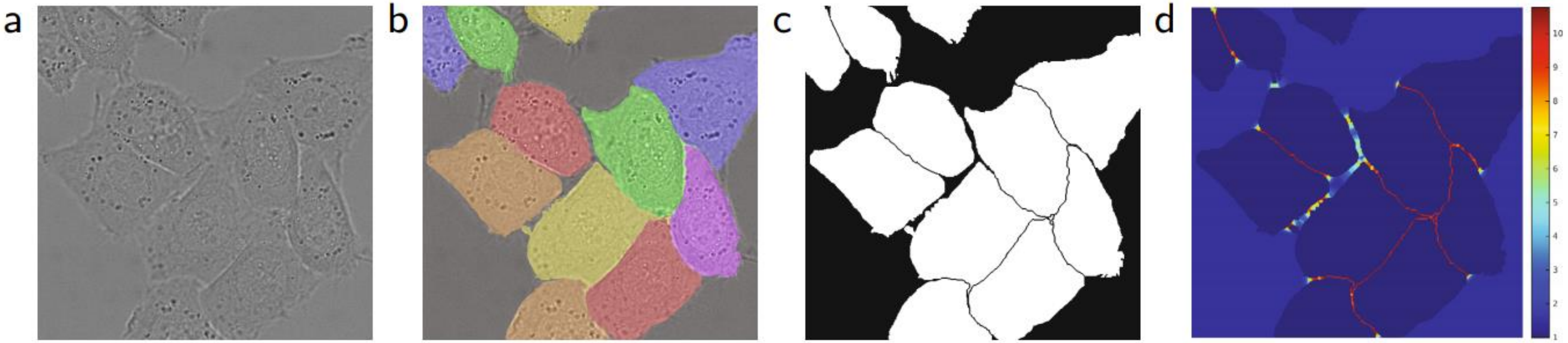


Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (**a**) raw image. (**b**) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (**c**) generated segmentation mask (white: foreground, black: background). (**d**) map with a pixel-wise loss weight to force the network to learn the border pixels.

Weighted loss

Give higher importance to borders

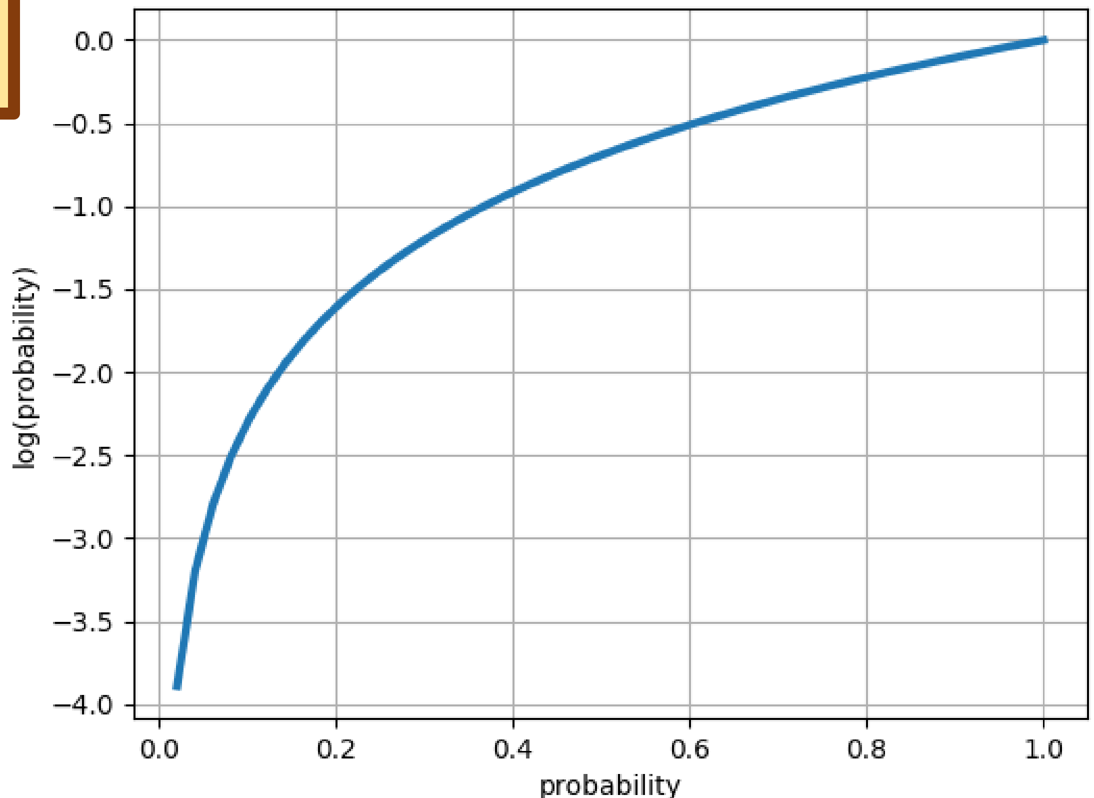
Loss function

Per channel soft-max (one channel per label)

$$p_k(\mathbf{x}) = \exp(a_k(\mathbf{x})) / \left(\sum_{k'=1}^{K} \exp(a_{k'}(\mathbf{x}))\right)$$

Cross entropy: penalize wrong labels

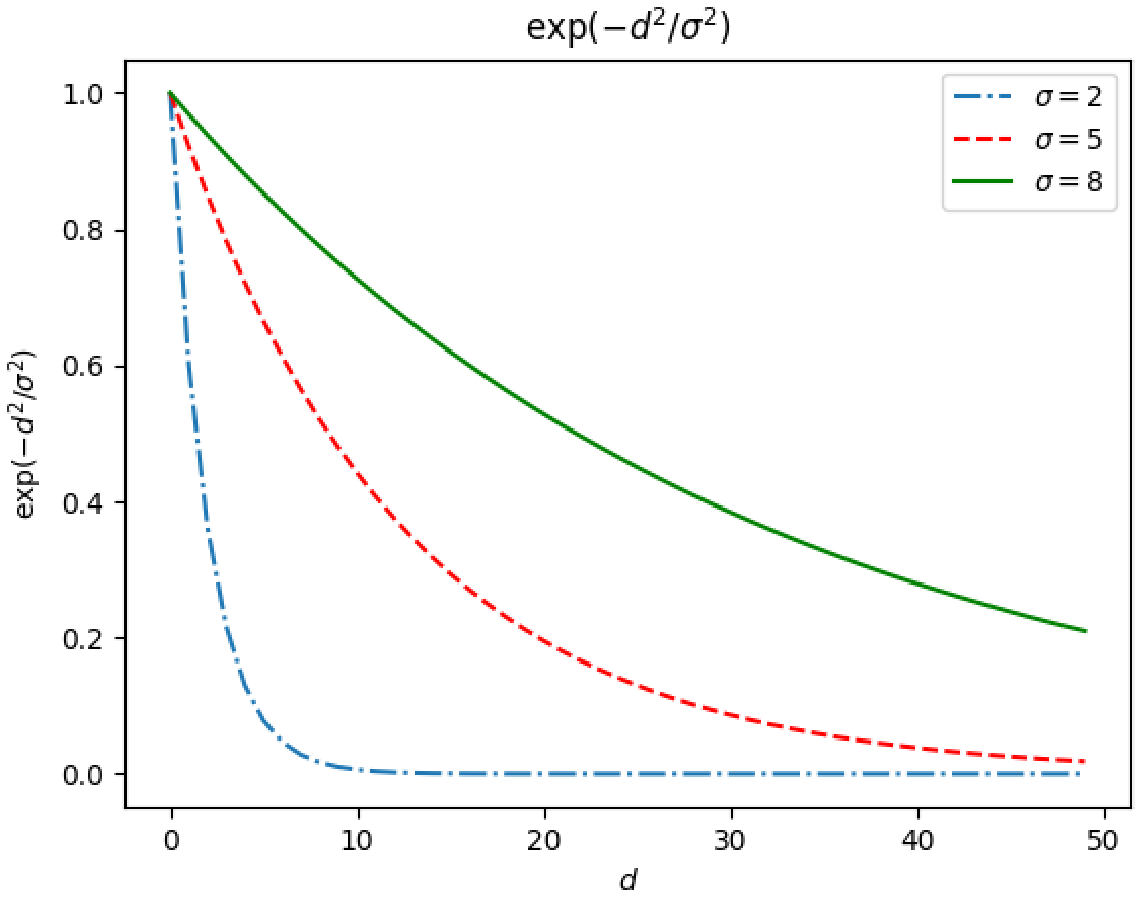
$$\sum_{\mathbf{x} \in \Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x}))$$

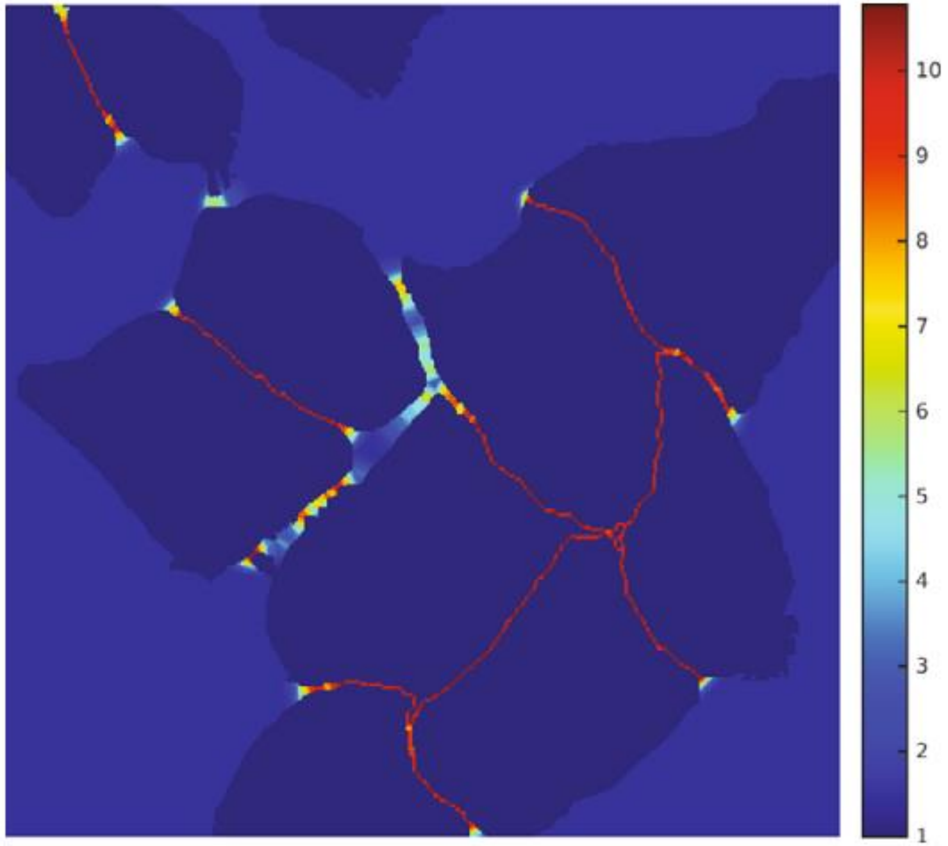


Is it a correct loss like that?

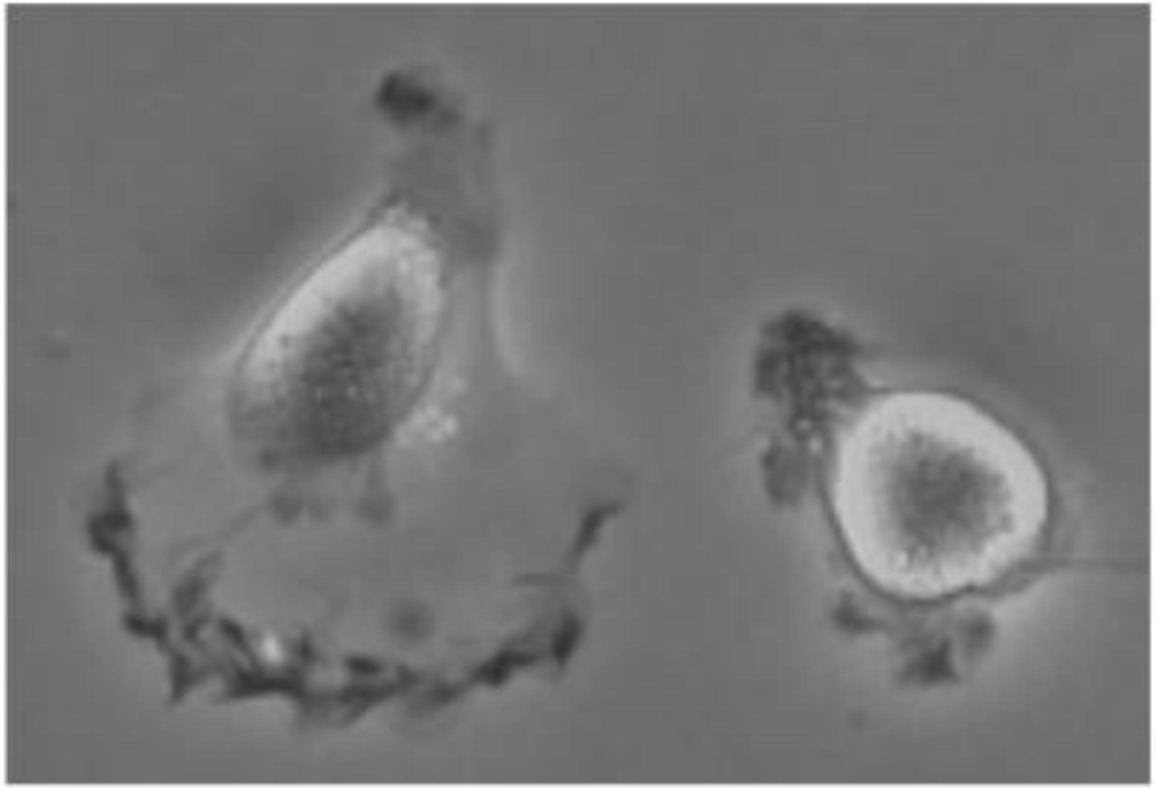
Weight map: compensate class imbalance and "highlight" borders

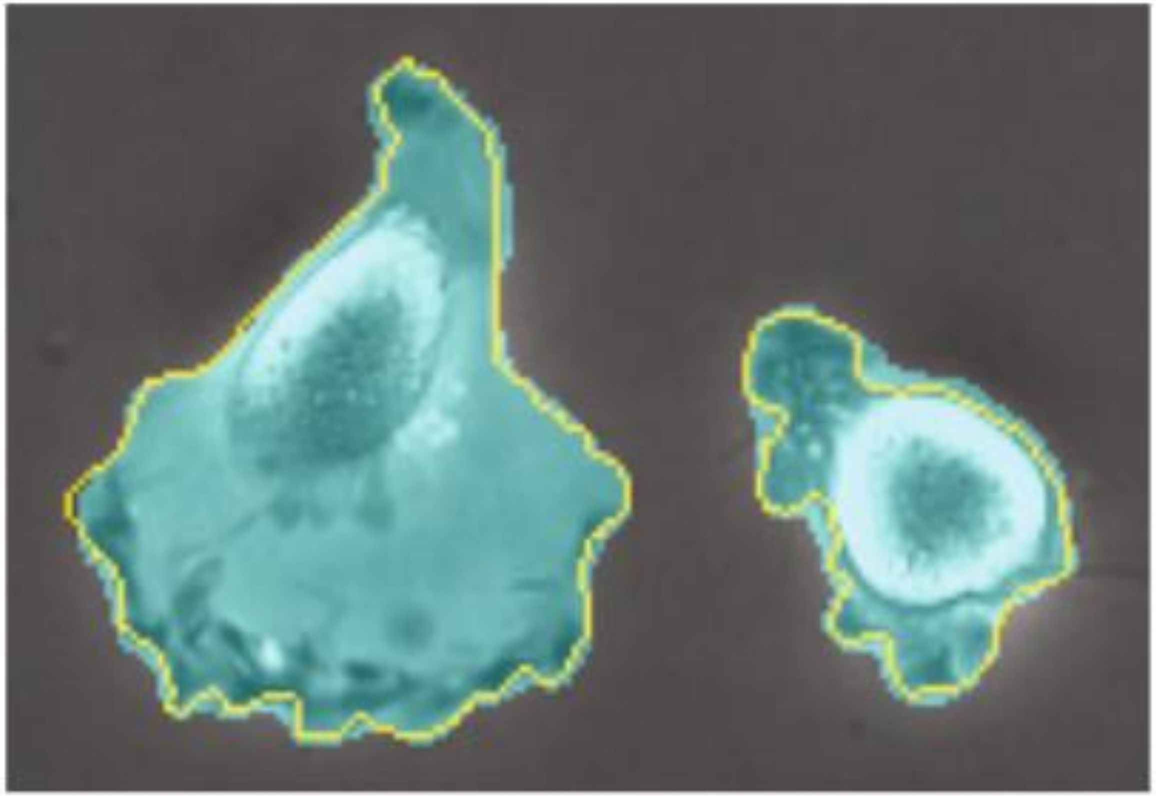
$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right)$$



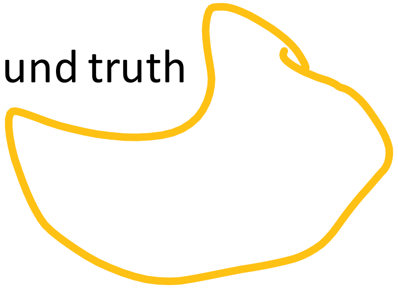


Some results

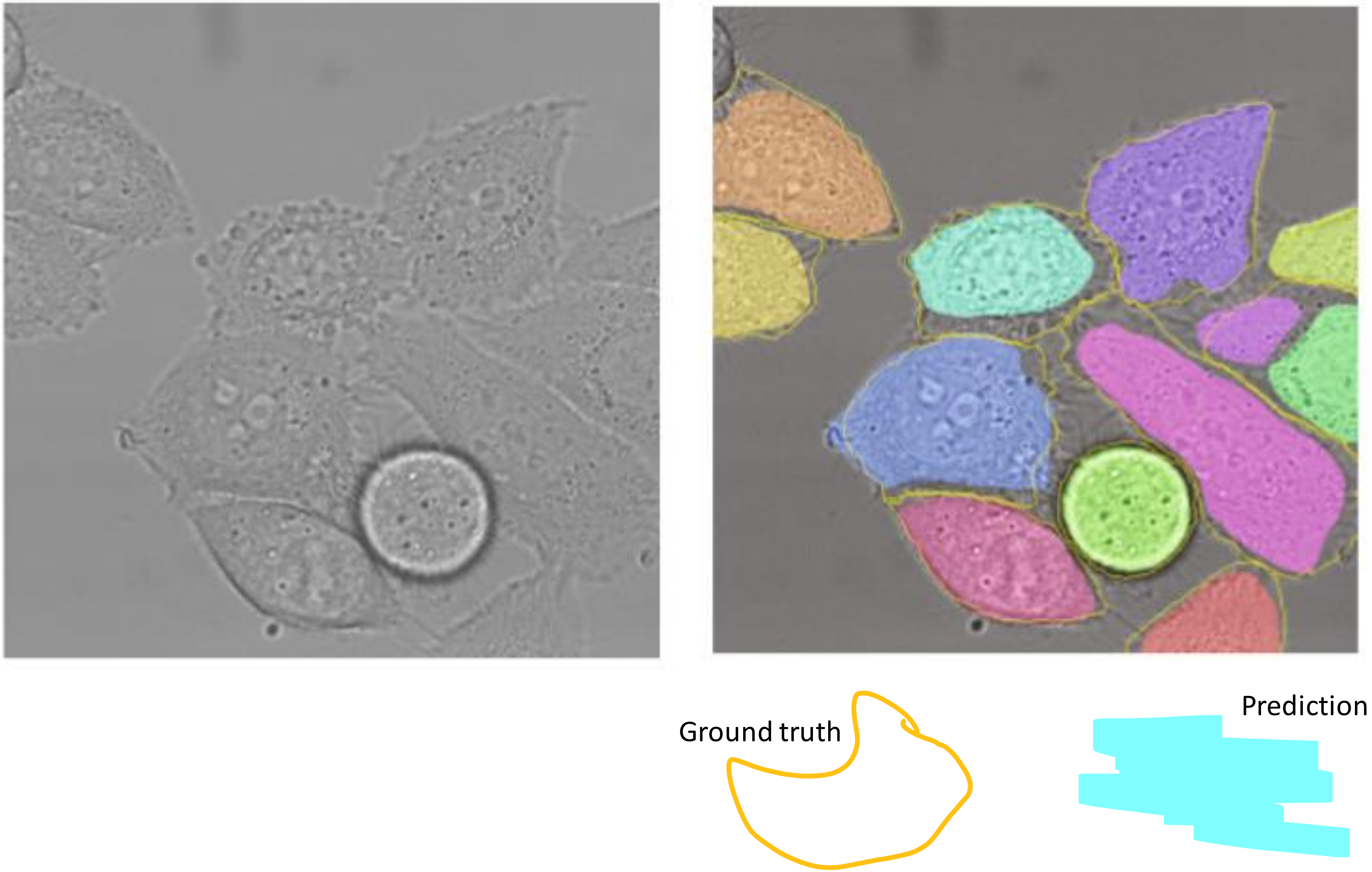




Ground truth



Prediction


Ground truth
Prediction

Intersection over Union (IoU)

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**