# U-Net: Convolutional Networks for Biomedical Image Segmentation

Ing. Milan Němý

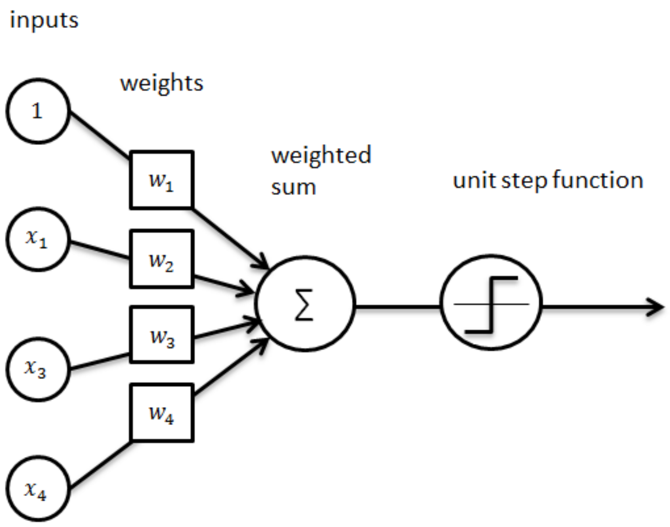
PhD candidate

FEL ČVUT, 7. 12. 2018

# Outline

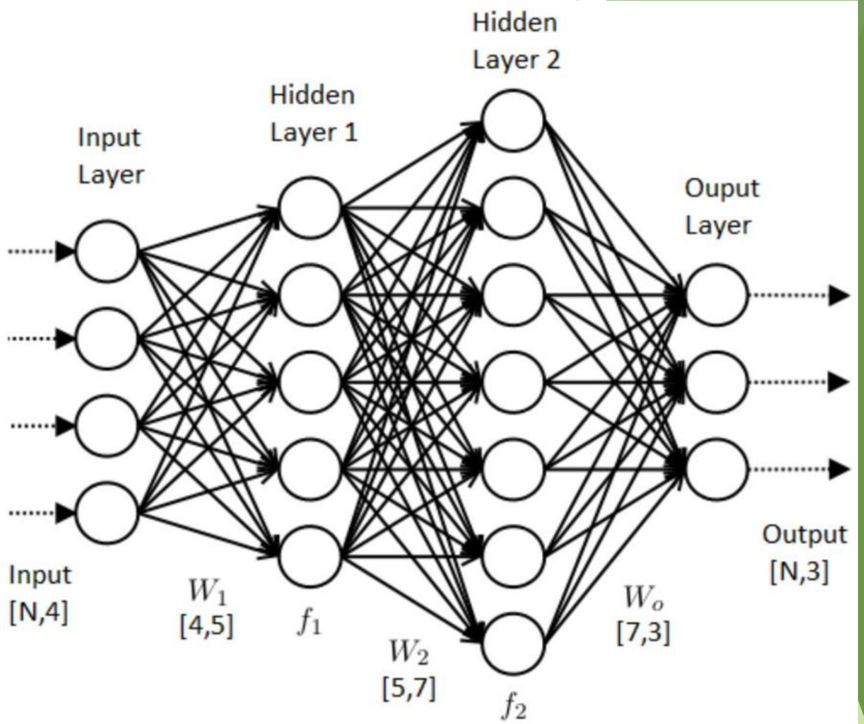
1. Artificial neural networks
2. Convolutional neural networks (classification)
3. Convolutional neural networks (segmentation)
    1. Sliding-window setup
    2. U-Net

# Artificial neural networks




**Perceptron model**

$$y(x) = sign(w.x + b)$$




**Artificial Neural Network**

- Hidden layers
- Non-linear activation function (tanh, sigmoid)
- Cost function
- Back-propagation algorithm

# Artificial neural networks

- **Multi-class classification**
- Activation function – softmax

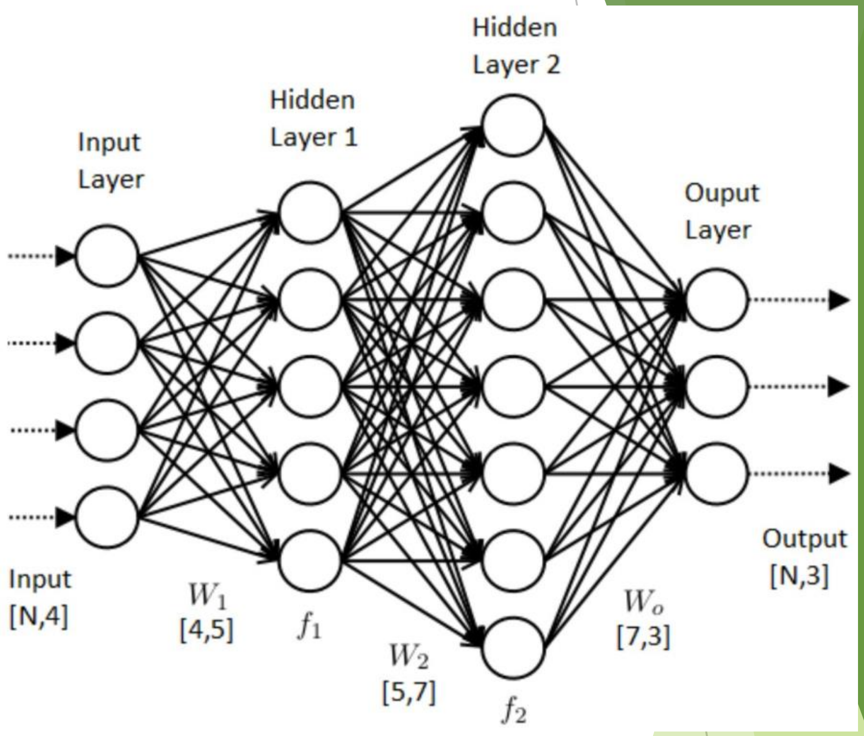
$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)}$$

- Cost function – cross entropy

$$C = -\sum_j d_j \log(p_j)$$

$d_j$ - target probability for output unit $j$

$p_j$ - probability output for $j$

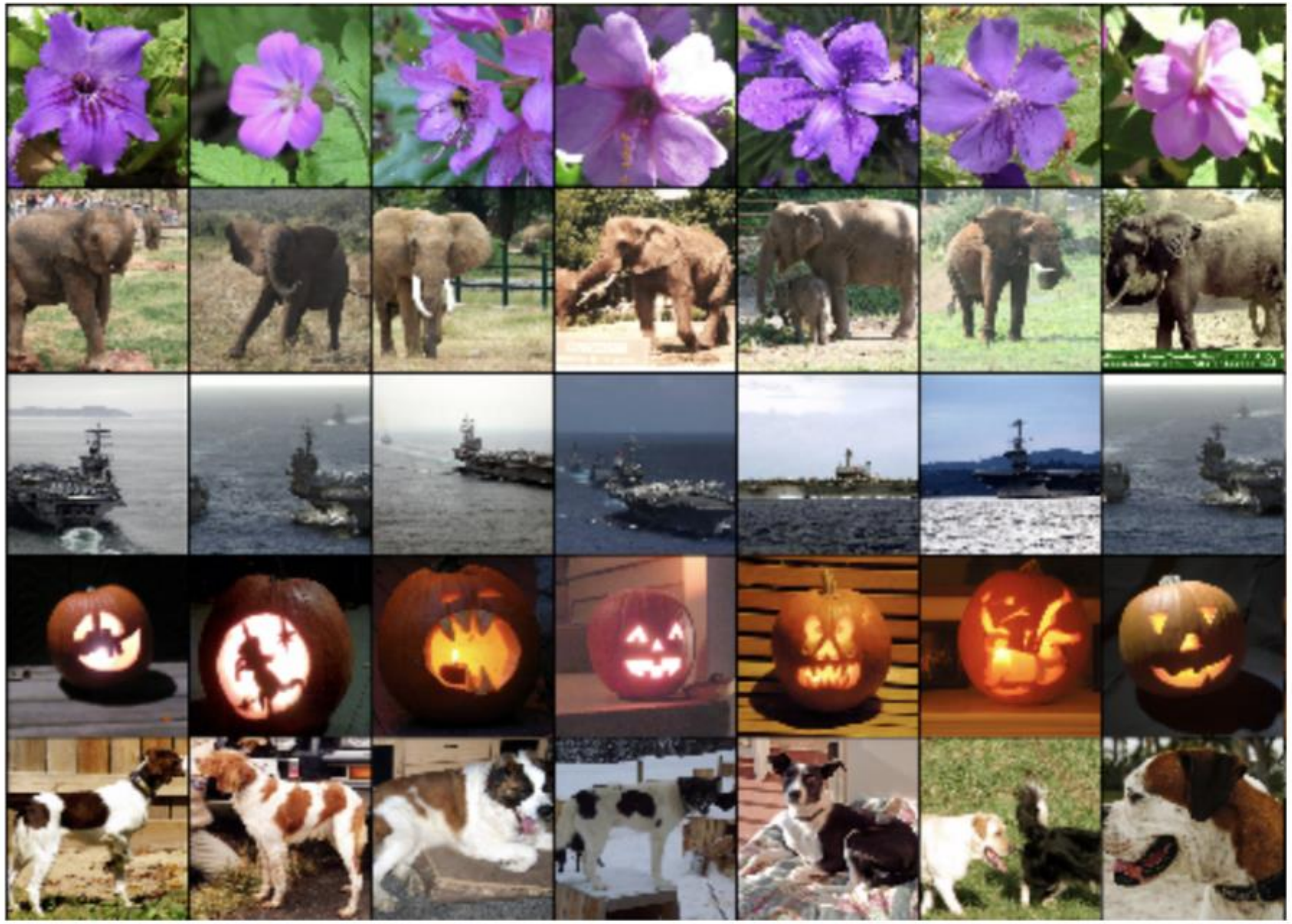



**Artificial Neural Network**

- Hidden layers
- Non-linear activation function (tanh, sigmoid)
- Cost function
- Back-propagation algorithm

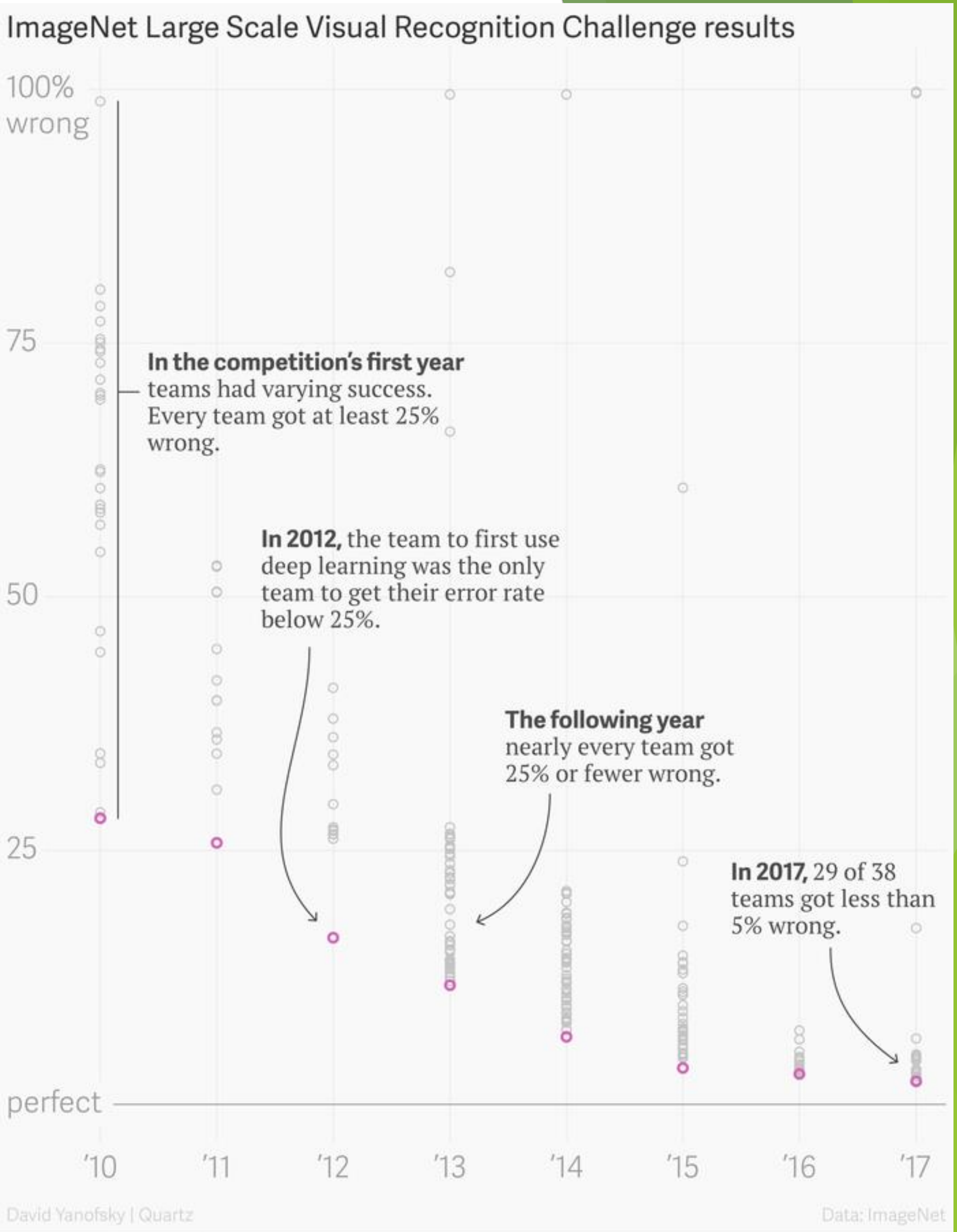
# Convolutional Neural Networks (CNN)

- (Krizhevsky et al., 2012)
- Winner of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012
  - 1.2 M high-resolution training images
  - 50k validation images
  - 150k testing images
  - 1000 classes



# Convolutional Neural Netwo

- (Krizhevsky et al., 2012) - AlexNet
- Winner of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012
  - 1.2 M high-resolution training images
  - 50k validation images
  - 150k testing images
  - 1000 classes



# Convolutional Neural Networks (CNN)

- (Krizhevsky et al., 2012)

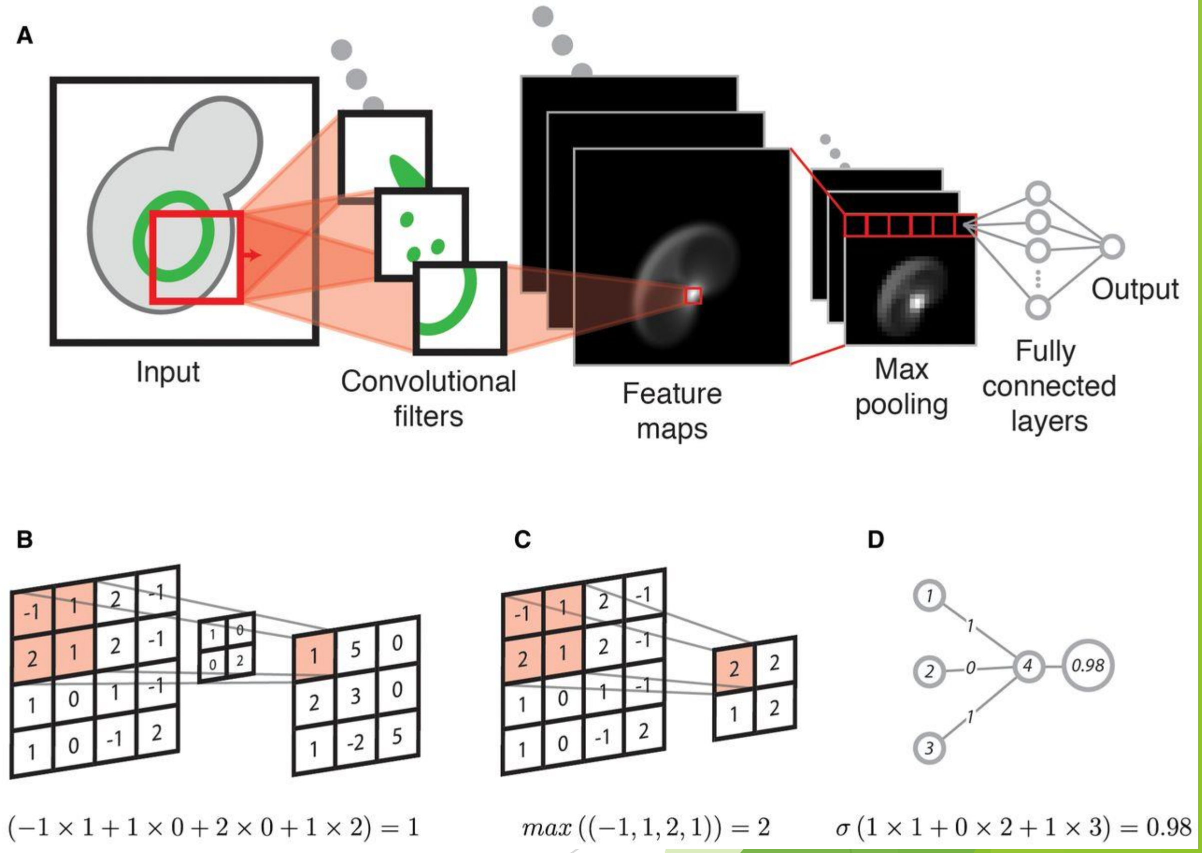
- Convolution
- = application of a filter

Input image

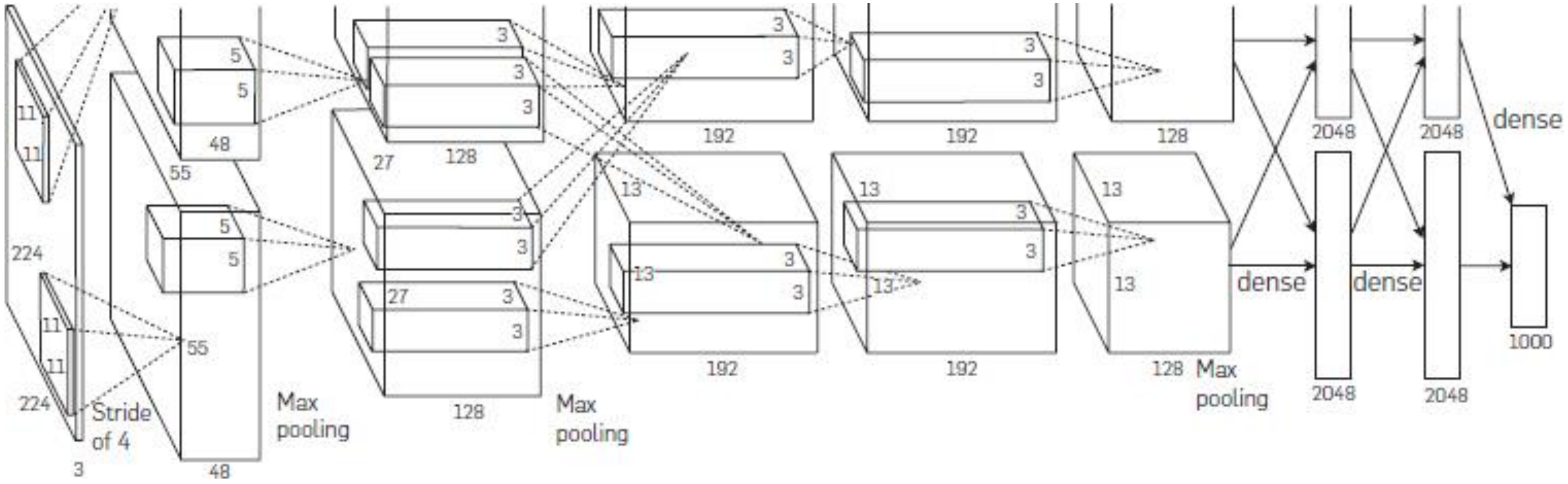
Convolution Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map

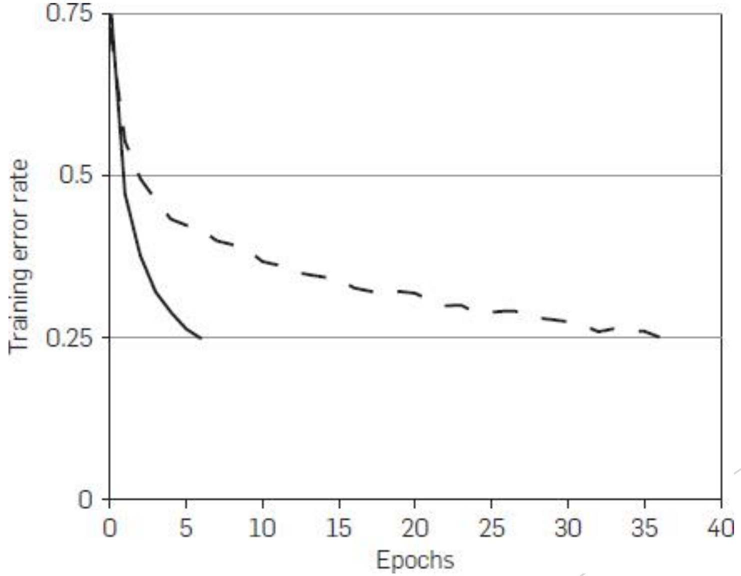


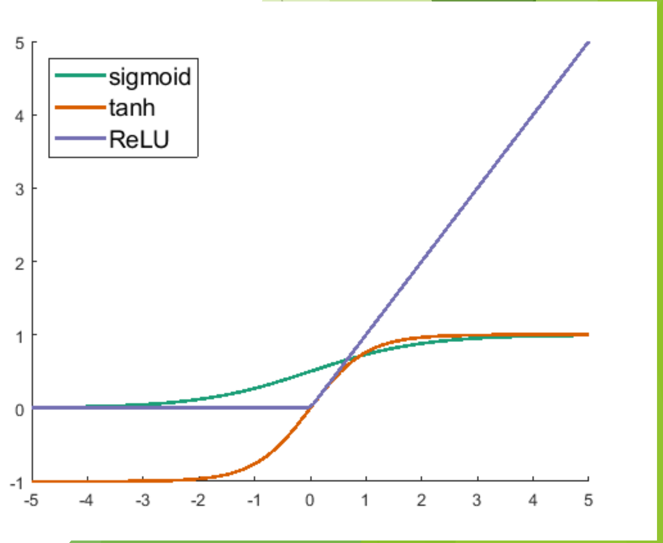
# Convolutional Neural Networks (CNN)




- Rectified Linear Units (ReLUs)
- Non-saturating nonlinearity
- $f(x) = \max(0, x)$

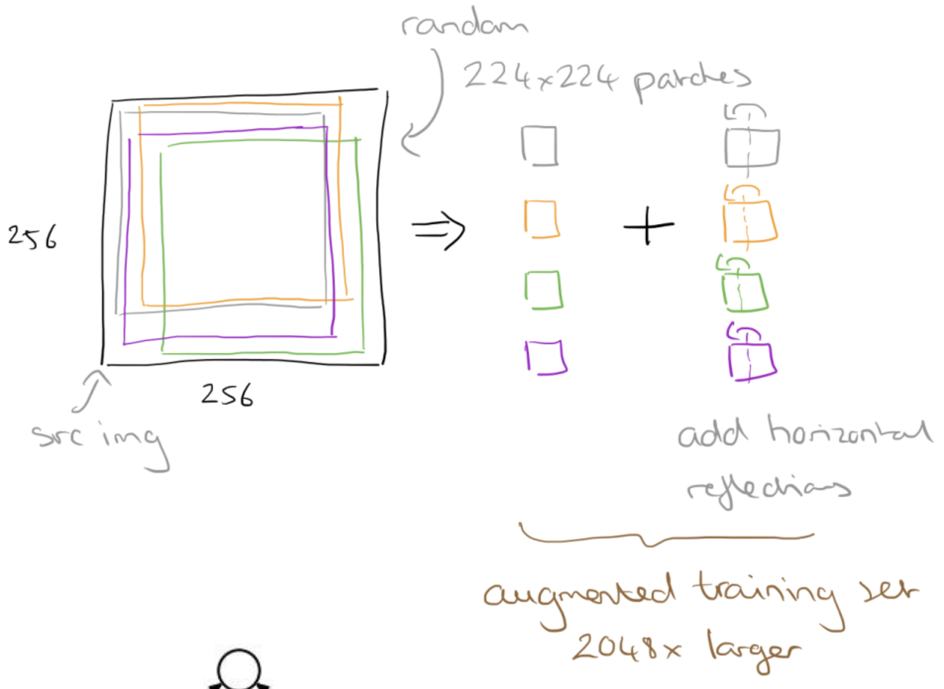
- Traininig on multiple GPUs
- 5 conv layers
- 3 fully-connected
- output – 1000-way softmax

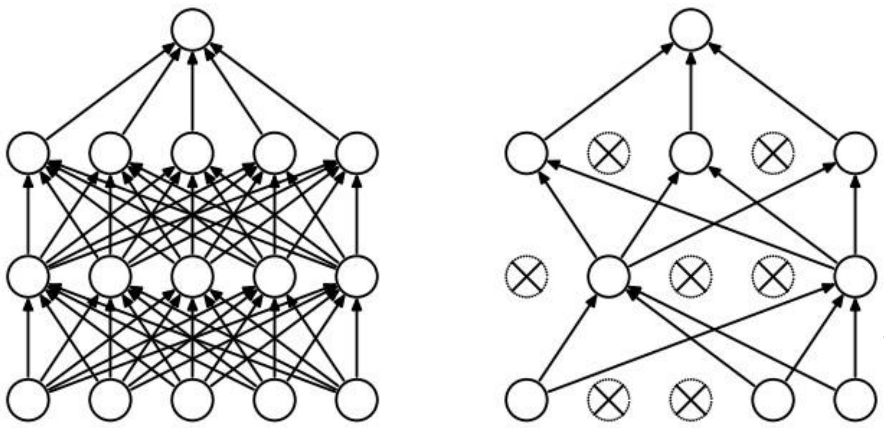


# Convolutional Neural Networks (CNN)

- (Krizhevsky et al., 2012)
- 60 M parameters
- But overfitting
  - Data augmentation
  - Dropout – Learning Less to Learn Better



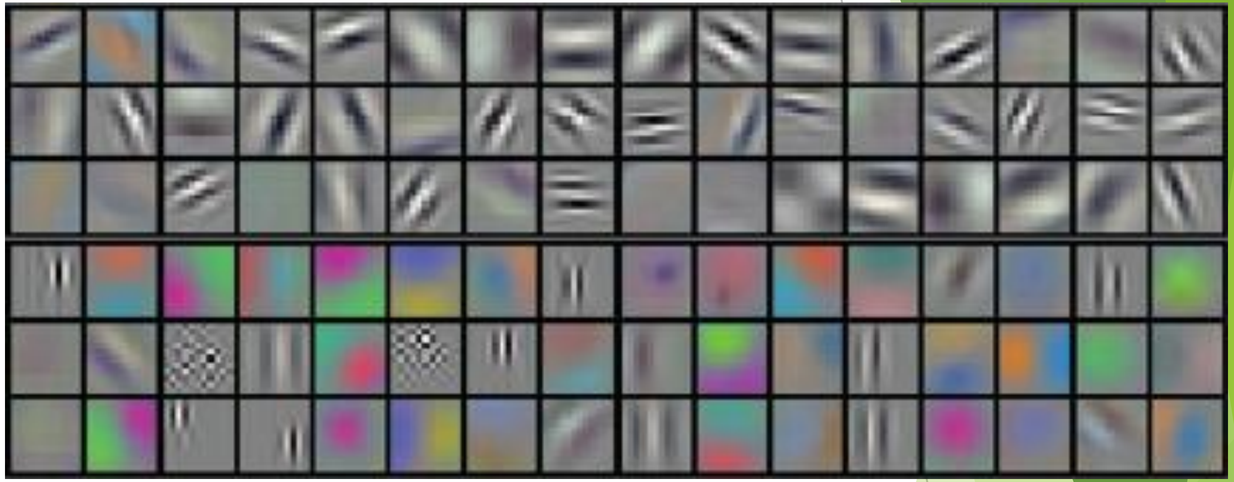



(a) Standard Neural Net

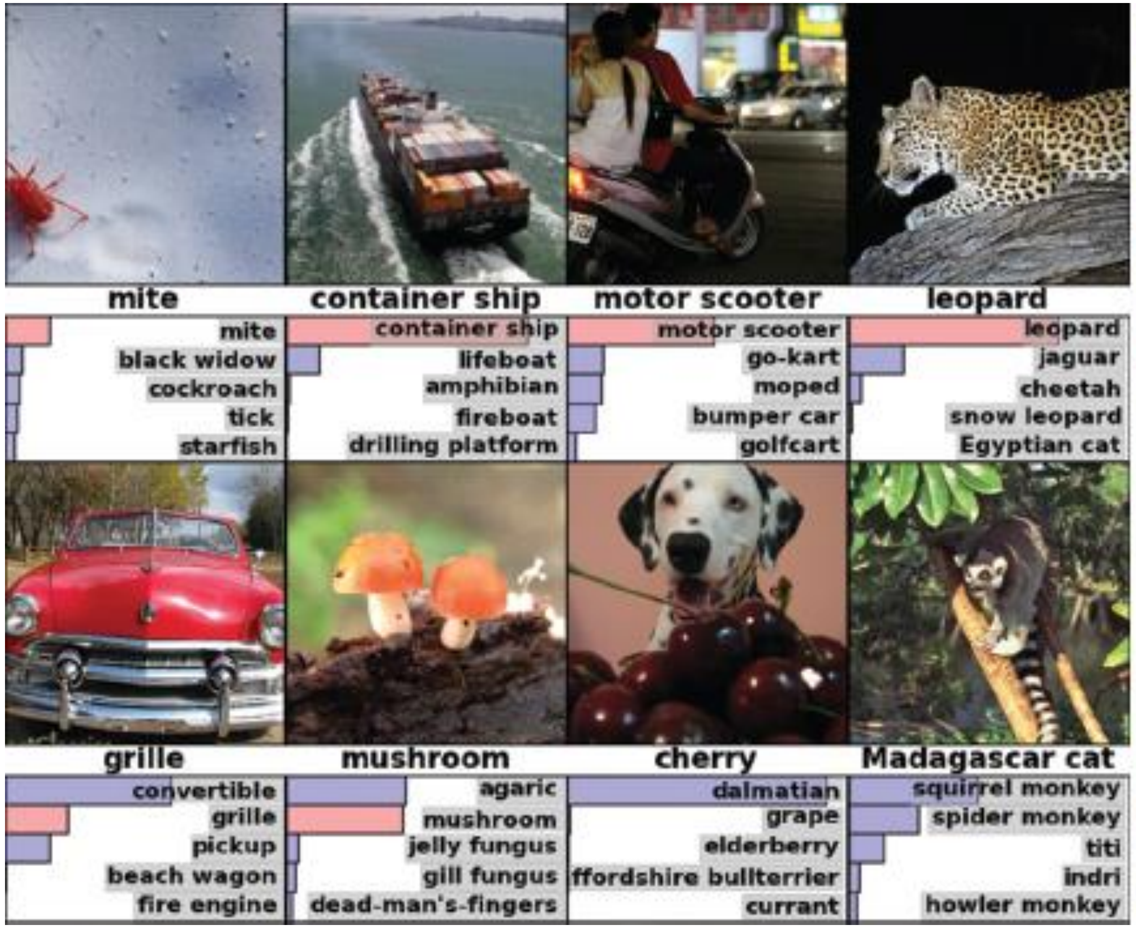
(b) After applying dropout.

# Convolutional Neural Networks (CNN)

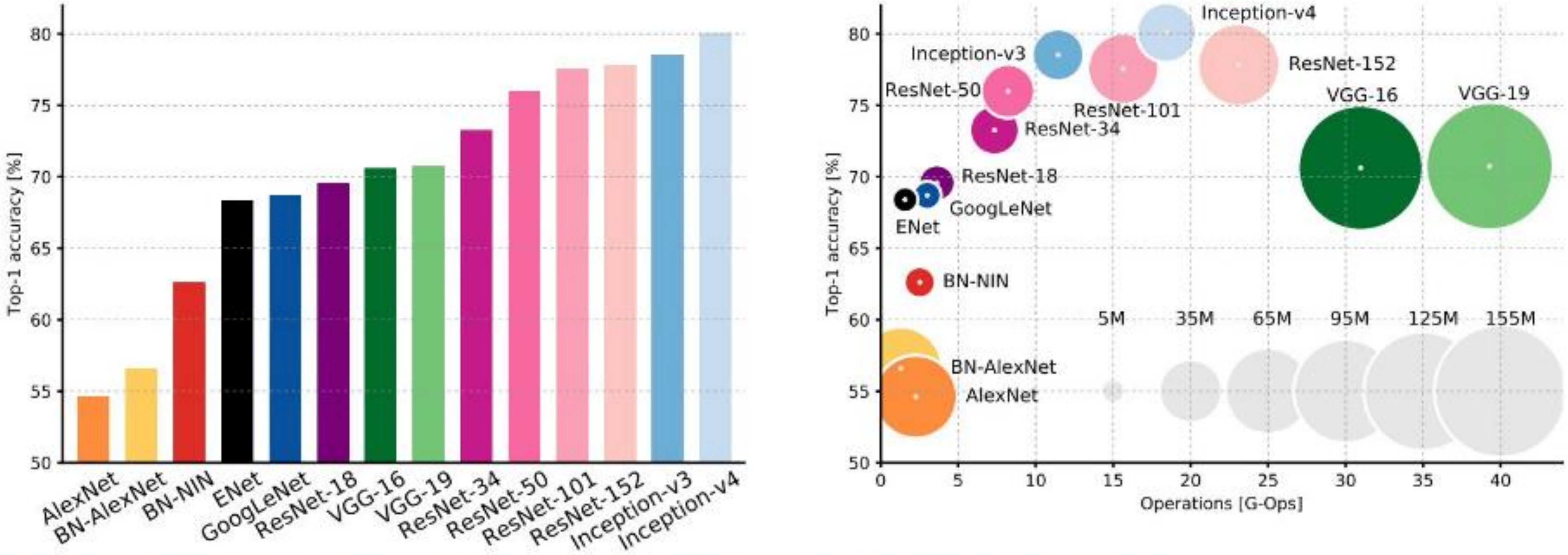
- (Krizhevsky et al., 2012)
- 60 M parameters
- But overfitting
  - Data augmentation
  - Dropout – Learning Less to Learn Better



| Model | Top-1 (val, %) | Top-5 (val, %) | Top-5 (test, %) |
|---|---|---|---|
| *SIFT + FVs*[6] | – | – | *26.2* |
| 1 CNN | 40.7 | 18.2 | – |
| 5 CNNs | 38.1 | 16.4 | **16.4** |
| 1 CNN* | 39.0 | 16.6 | – |
| 7 CNNs* | 36.7 | 15.4 | **15.3** |


mite
container ship
motor scooter
leopard
mite
black widow
cockroach
tick
starfish
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
go-kart
moped
bumper car
golfcart
leopard
jaguar
cheetah
snow leopard
Egyptian cat
grille
mushroom
cherry
Madagascar cat
convertible
grille
pickup
beach wagon
fire engine
agaric
mushroom
jelly fungus
gill fungus
dead-man's-fingers
dalmatian
grape
elderberry
ffordshire bullterrier
currant
squirrel monkey
spider monkey
titi
indri
howler monkey

# Convolutional Neural Networks (CNN) – AlexNet performance

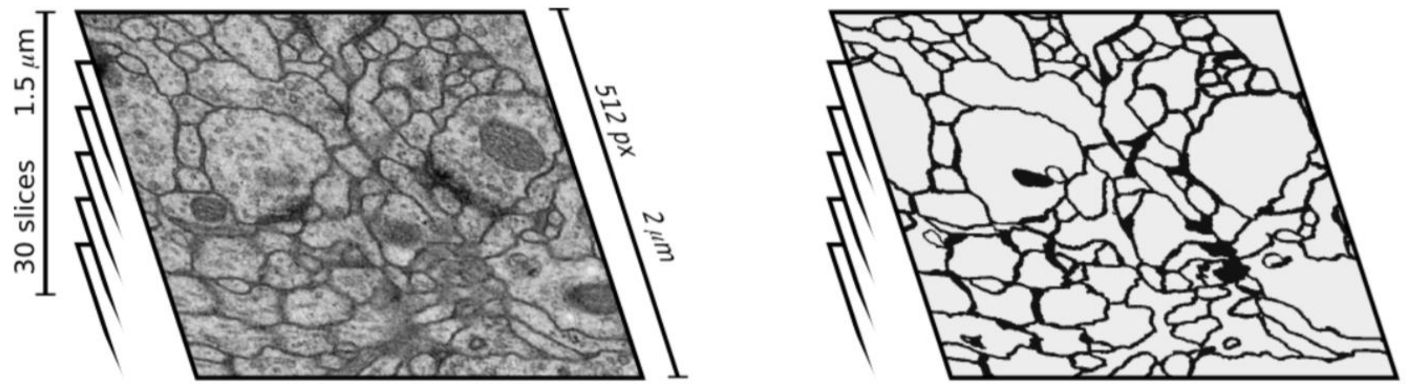



An Analysis of Deep Neural Network Models for Practical Applications, 2017.

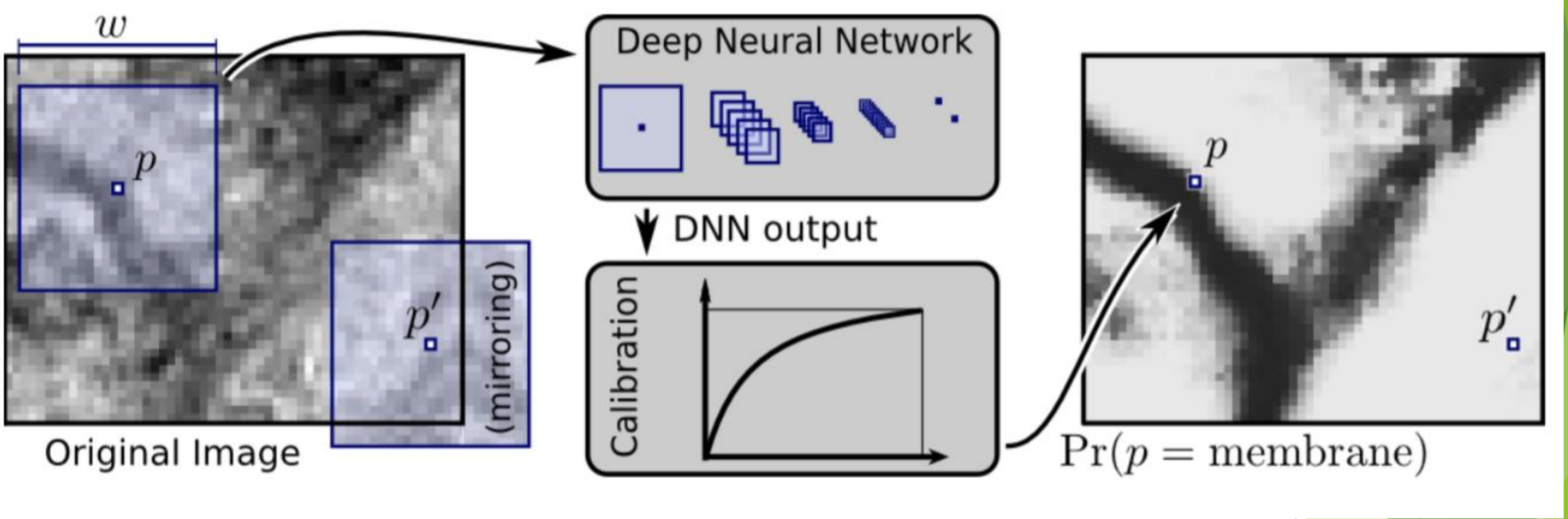
Canziani, A., Paszke, A., & Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

# Convolutional Neural Networks - Segmentation

- (Ciresan et al., 2012)
- Neuronal structures
- Electron microscopy (EM) images
- Segment neuron membranes
- CNN as pixel classifier
- ISBI 2012 EM Segmentation Challenge



# Convolutional Neural Networks - Segmentation

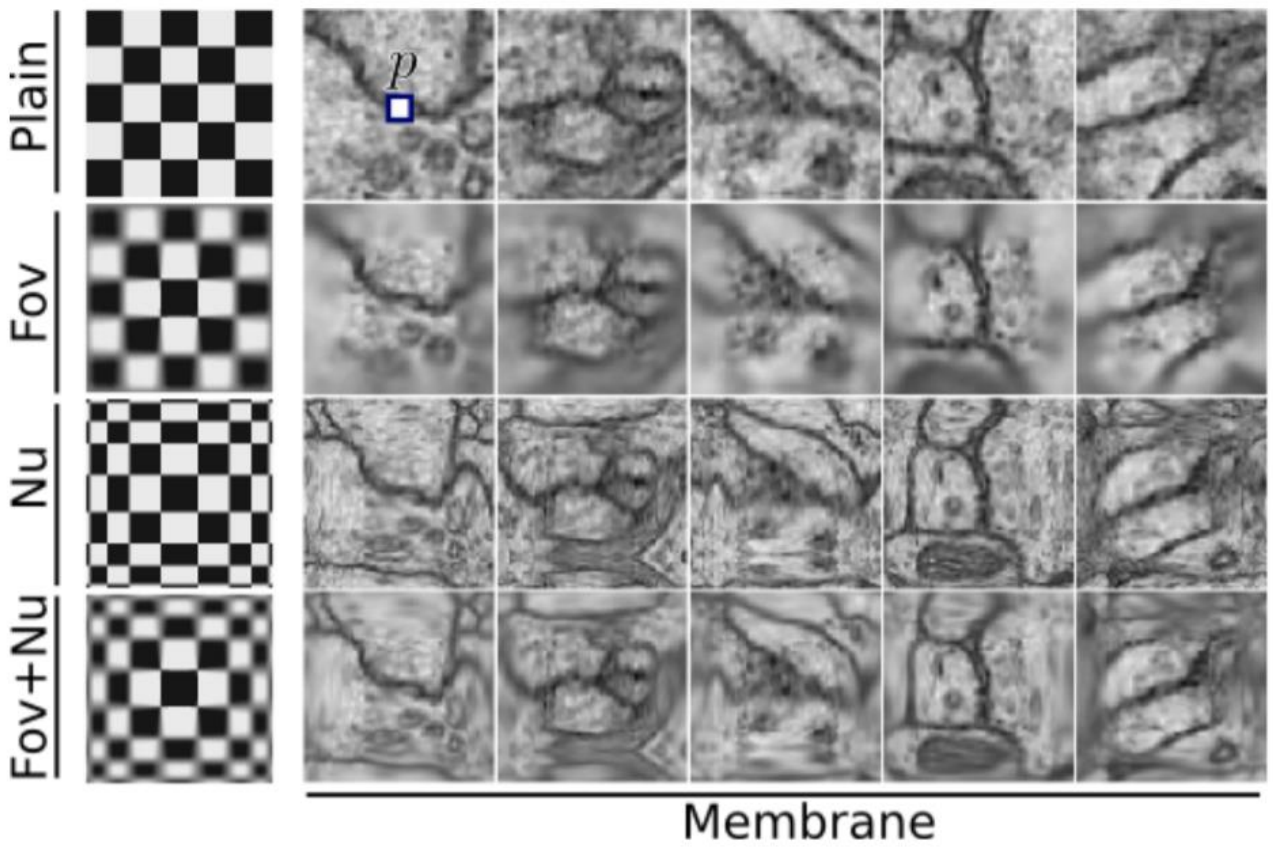



- 1-4 stages of conv and max-pooling layes
- Several fully connected layers
- Softmax activation function

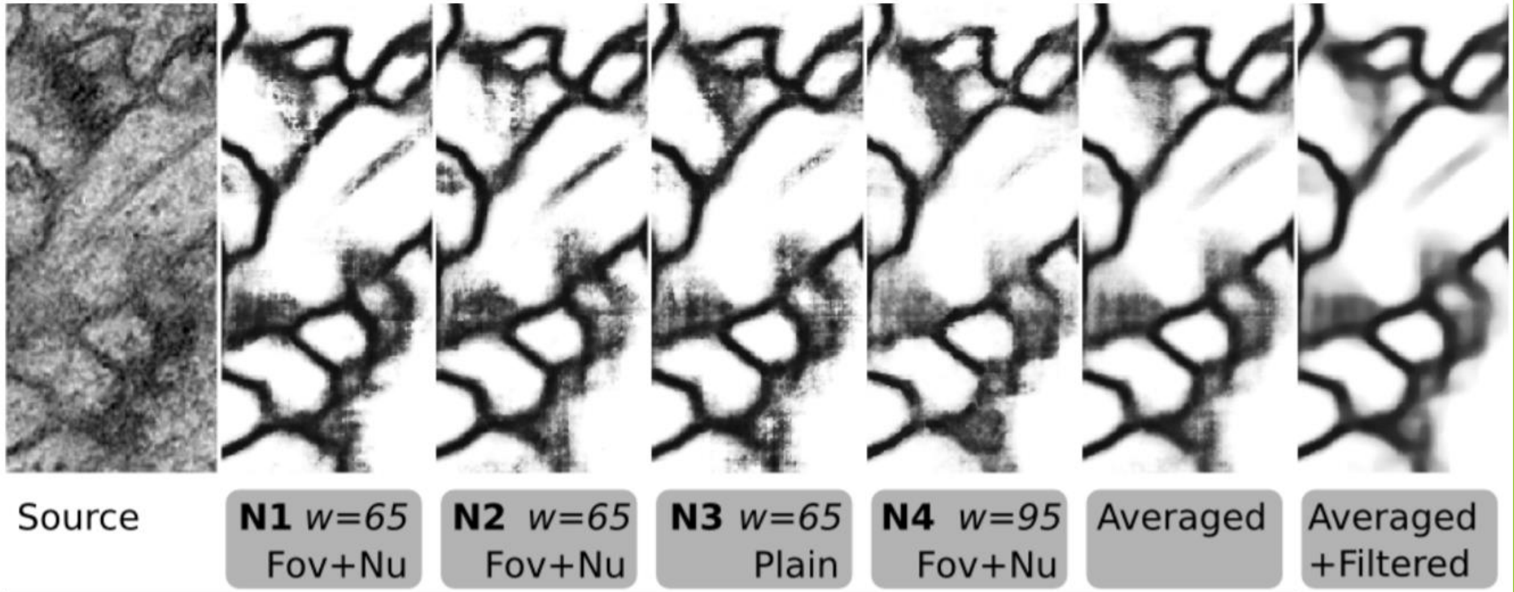
# Convolutional Neural Networks - Segmentation

- 30 images at 512x512
- ~50k membrane pixels per image
- ~50k non-memberane pixels per images
- => 3 M training examples
- + data augmentation (mirroring and rotating)

- Foveation
- Nonuniform sampling



- Core i7 3.06 GHz, 24 GB RAM and four GTX 580 graphic cards
- GPU acceleration by a factor of 50
- One epoch
  - N1 – 170 min
  - N4 – 340 min
- 30 epochs => several days




| | | N1 | N2 | N3 | N4 | Averaged + Filtered (after filtering) |
|---|---|---|---|---|---|---|
| Rand err. | $[\cdot 10^{-3}]$ | 64 | 68 | 57 | 61 | **48** |
| Warping | $[\cdot 10^{-6}]$ | 457 | 485 | 618 | 524 | **434** |
| Pixel err. | $[\cdot 10^{-3}]$ | 65 | 66 | 66 | 68 | **60** |

Table 1: 11-layer architecture for network N4, $w = 95$.

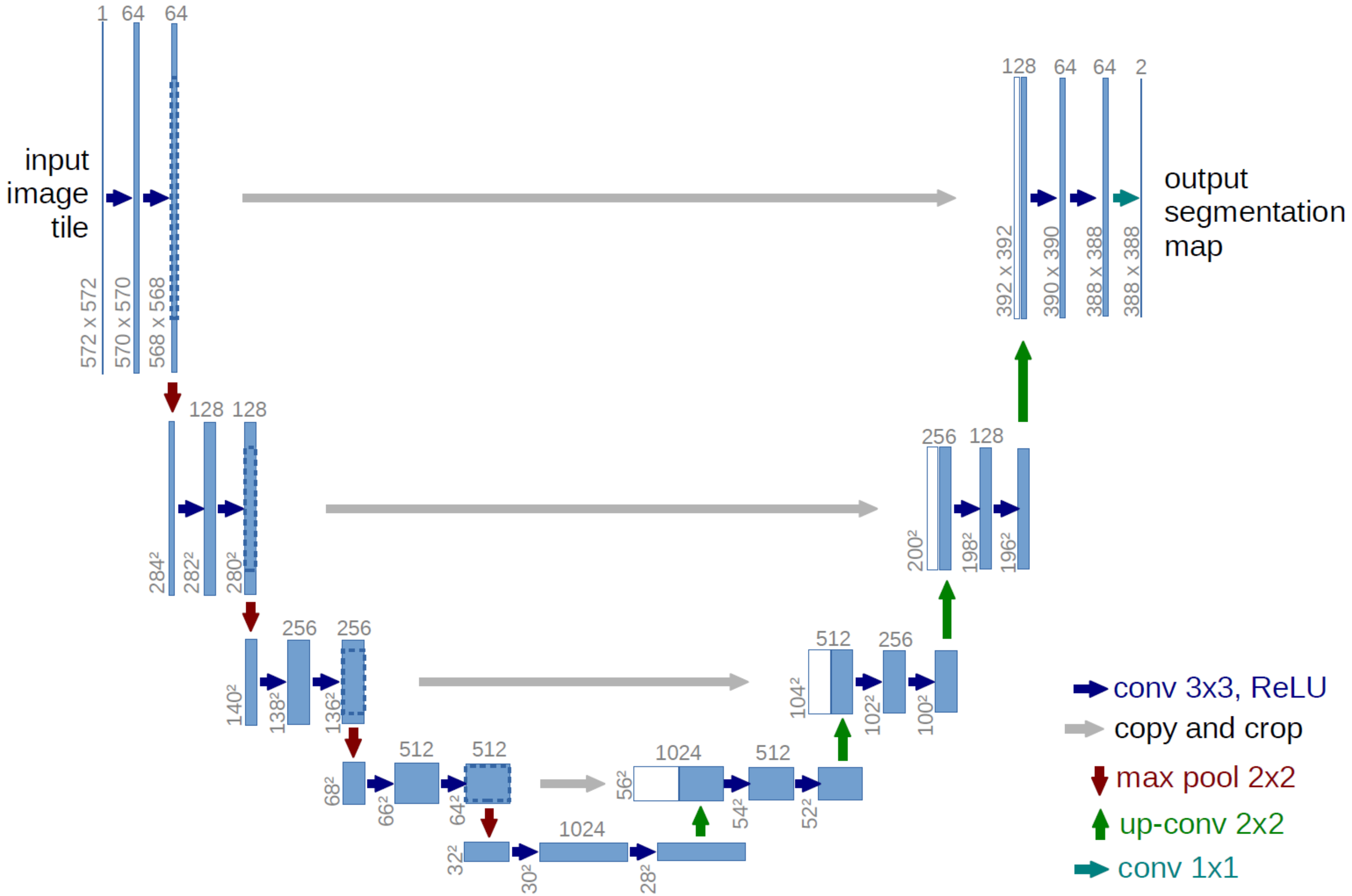
| Layer | Type | Maps and neurons | Kernel size |
|---|---|---|---|
| 0 | input | 1 map of 95x95 neurons | |
| 1 | convolutional | 48 maps of 92x92 neurons | 4x4 |
| 2 | max pooling | 48 maps of 46x46 neurons | 2x2 |
| 3 | convolutional | 48 maps of 42x42 neurons | 5x5 |
| 4 | max pooling | 48 maps of 21x21 neurons | 2x2 |
| 5 | convolutional | 48 maps of 18x18 neurons | 4x4 |
| 6 | max pooling | 48 maps of 9x9 neurons | 2x2 |
| 7 | convolutional | 48 maps of 6x6 neurons | 4x4 |
| 8 | max pooling | 48 maps of 3x3 neurons | 2x2 |
| 9 | fully connected | 200 neurons | 1x1 |
| 10 | fully connected | 2 neurons | 1x1 |

# Convolutional Neural Networks – Segmentation (Sliding-window setup)

- Drawbacks

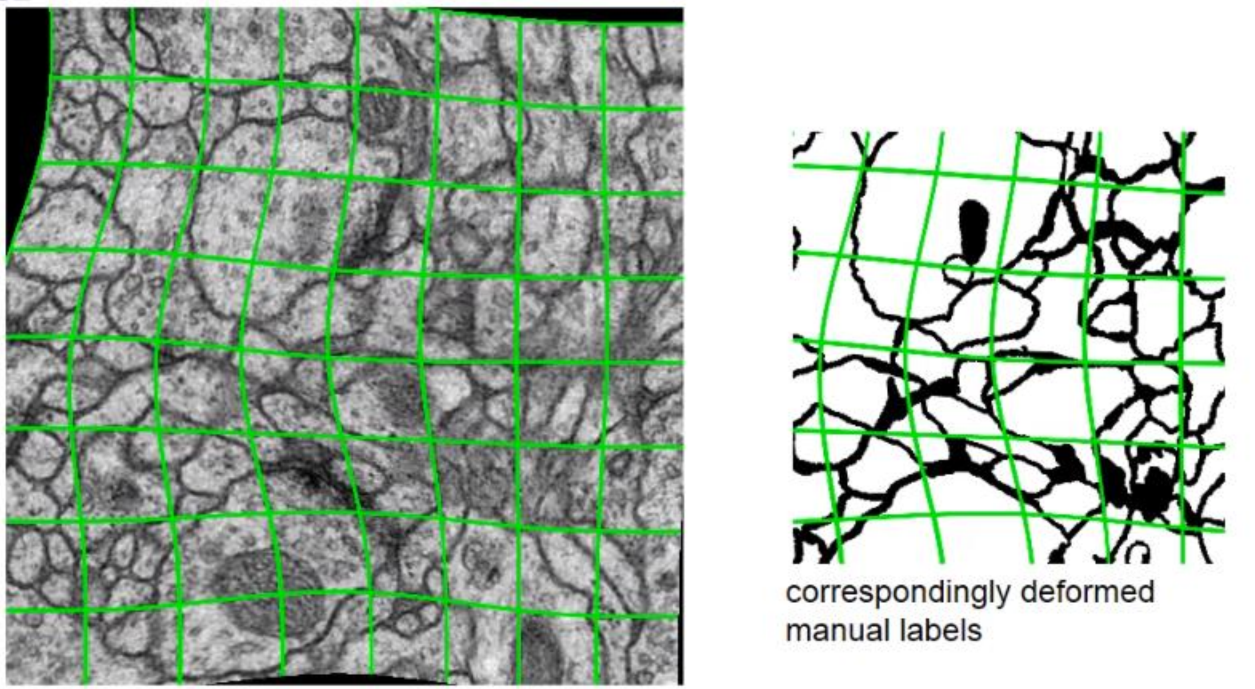
1. Separate runs for each patch + overlapping patches => slow
2. Localization accuracy vs. the use of context

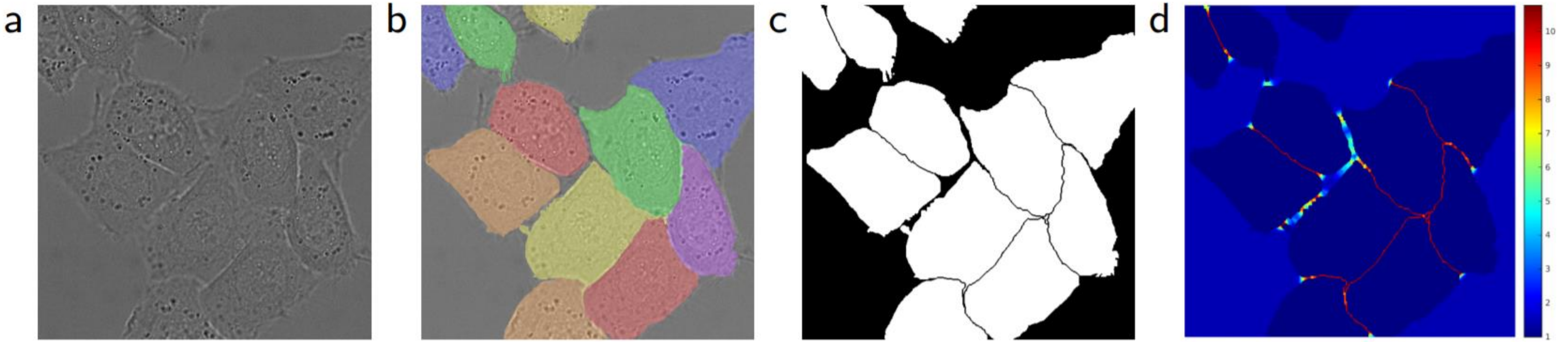
# U-net architecture



# U-net architecture

- Data augmentation
- Separation of touching objects



# U-net architecture

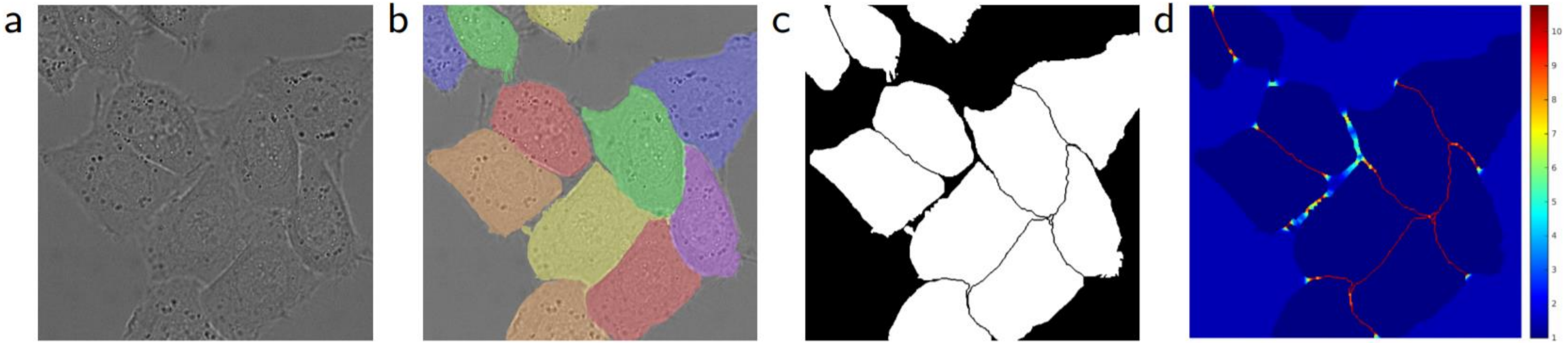
- Data augmentation
- Separation of touching objects

Balances class frequencies

Distance to the border of the nearest cell

Distance to the border of the second nearest cell

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



# U-net architecture

- Data augmentation
- Separation of touching objects

Balances class frequencies

Distance to the border of the nearest cell

Distance to the border of the second nearest cell

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

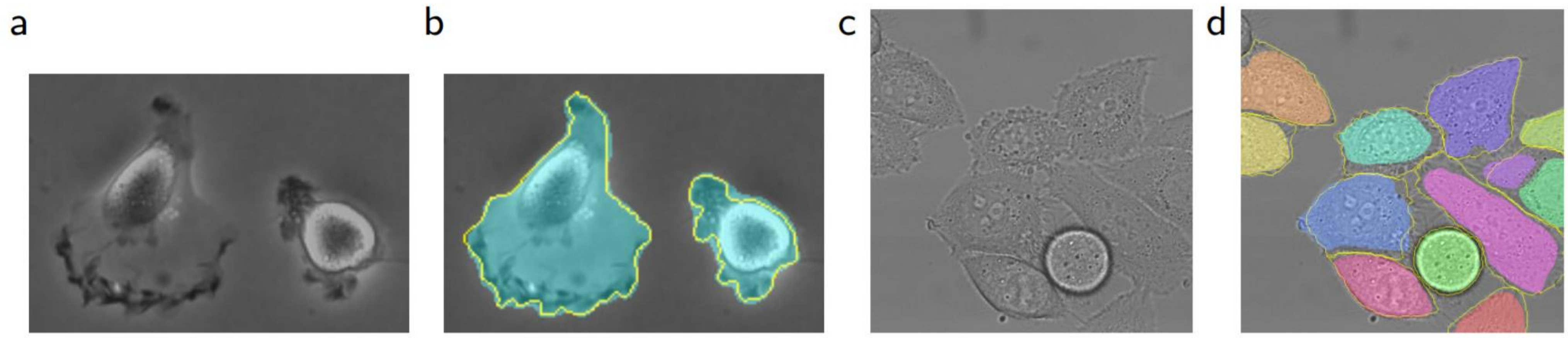
Cross entropy loss function

$$E = \sum_{x \in \Omega} w(x) \log(p_{l(x)}(x))$$

Soft-max activation function

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

# U-net



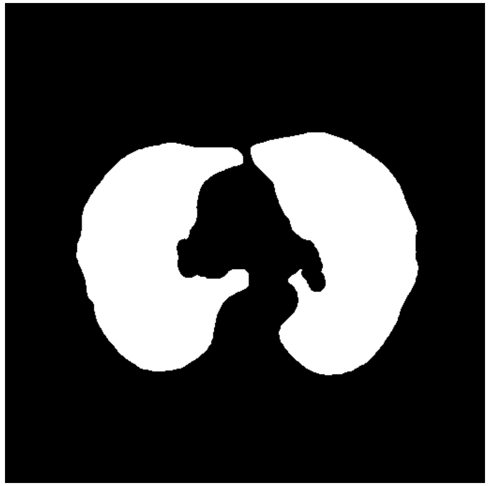

- Training time: 10 h

| Rank | Group name | Warping Error | Rand Error | Pixel Error |
|---|---|---|---|---|
| | ** human values ** | 0.000005 | 0.0021 | 0.0010 |
| 1. | u-net | **0.000353** | 0.0382 | 0.0611 |
| 2. | DIVE-SCI | 0.000355 | 0.0305 | 0.0584 |
| 3. | IDSIA [1] | 0.000420 | 0.0504 | 0.0613 |
| 4. | DIVE | 0.000430 | 0.0545 | **0.0582** |

Sliding-window technique →

# Demonstration - Finding and Measuring Lungs in CT Data

kaggle



?

$$Dice = \frac{2 \cdot |mask \cap prediction|}{|mask| + |prediction|}$$

Notebook: https://www.kaggle.com/toregil/a-lung-u-net-in-keras/notebook

# References


- [1] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
- [2] Ciresan, D., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In Advances in neural information processing systems (pp. 2843-2851).
- [3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).