Biomedical Imaging Lecture 4
Image Segmentation

Jens Rittscher

Quantitative Biomedical Imaging Group
Institute of Biomedical Engineering
Big Data Institute
University of Oxford

Course overview

- Lecture 1: Introduction of biomedical imaging
- Lecture 2: Working with digital images
- Lecture 3: Preprocessing and extracting features
- Lecture 4: Image segmentation
- Lecture 5: Detecting objects of interest
- Lecture 6: Using deep neural networks for detection and segmentation
- Lecture 7: Assessing morphology and shape
- Lecture 8: Analysing phenotypes

Learning objectives

- Motivation for deep learning
- Elements of deep networks
- Convolutional Neural Networks
- Semantic segmentation
- U-Net

Universal approximation theorem

A feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of $\mathbf{R}^n$ under some mild assumptions.

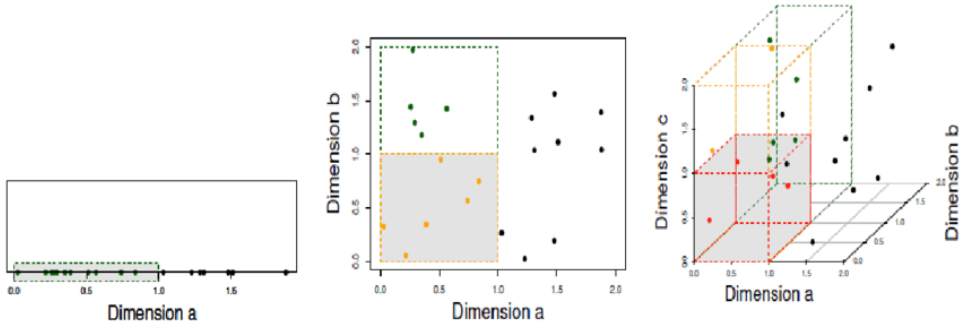
Cybenko (1989), Hornik (1991)

However:

- This theoretical results does not provide us with any guidance on how to find ‘right’ number of hidden units.
- Furthermore, it does not specify how the parameters of such a network should be learnt.

Curse of dimensionality

- As the dimensionality increases more and more samples are needed to populate the space
- Example: a regular mesh divided into 10 units of a 6-dimensional cube would have 10^ nodes
- Typically we have far less training data



Distribution of 20 objects in space:

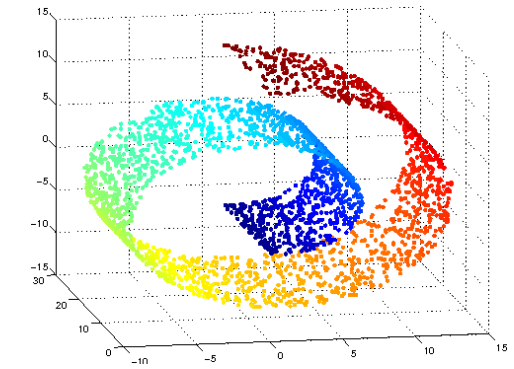
- Left – in one dimension: about 50% of all objects are in a unit bin
- Middle – two dimensions: only 25% of all objects are in a unit bin
- Right – three dimensions: only 12.5% of all objects are in a unit bin

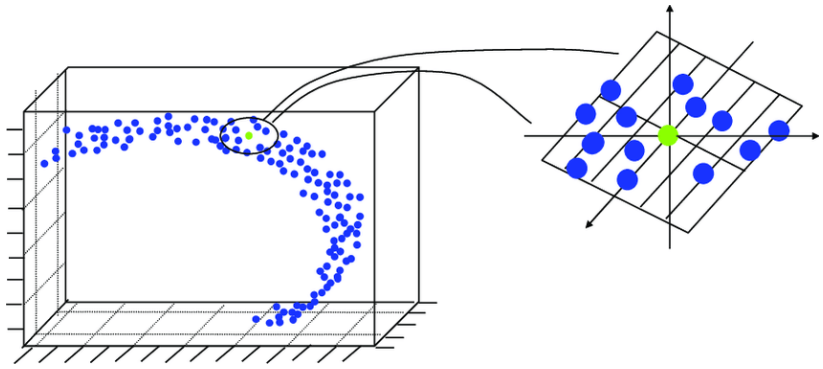
Generalizability

- How can we represent complex functions that have many more regions to be distinguished than training samples?
- Local smoothness constraints ensure a good representation around training samples but what happens elsewhere?
- It is crucial to make assumptions about the structure of the underlying data distribution

Manifold learning

- We do not assume that the data lies in a globally linear subspace
- Each data point and its immediate neighbours lie in a local linear subspace
- Extracting the underlying manifold is difficult





Deep learning

- Models that capture the underlying data representation
- Learn the distribution from the data
- Models whose parameters can be learnt efficiently through gradient decent

$$f = f(x, \theta, w) = \phi(x, \theta)^T w$$

X – features
Θ – parameters of the function ϕ
ϕ - a function that generalises the kernel (see SVMs)
w – vector that maps the parameters to the output y

Elements of deep networks

- Cost functions
- Outputs
- Hidden units
- Architectures
- Parameter estimation

Cost function

- NN are trained using maximum likelihood
- It is described as the cross entropy between the training data and the model distribution

$$J(\theta) = -\mathbb{E}_{x,y} \log p_{\text{model}}(y|x)$$

- Mean square error cost is given as

$$J(\theta) = \frac{1}{2}\mathbb{E}_{x,y}||y - f(x;\theta||^2 + \text{const}$$

- Mean square error or L1 error can lead to poor results, which is why the cross entropy is more popular.

Output units

- A NN provides a set of hidden features

$$h = f(x; \theta)$$

- The final output layer then maps these hidden features to complete the task the network performs.

- The output can be a binary variable, a class label, or a continuous variable

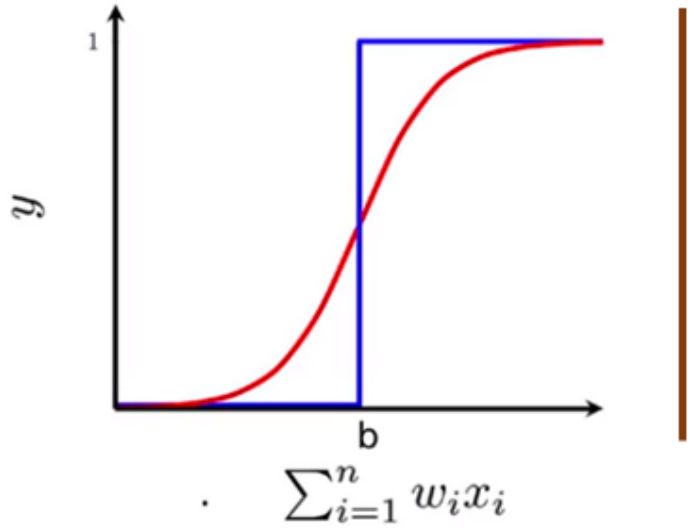
Output units: linear units

- An affine transformation with no non-linearity

$$\hat{y} = W^t h + b$$

Output unit: sigmoid unit for binary output

- Sigmoid or logistic output provides a binary output
- Smooth gradient, i.e. prevents jumps in the output values



$$y = \frac{1}{1+e^{-(w^T x+b)}}$$

Output unit: softmax

- Used to predict a discrete distribution over a discrete variable with n possible values
- A value for all output values is being calculated

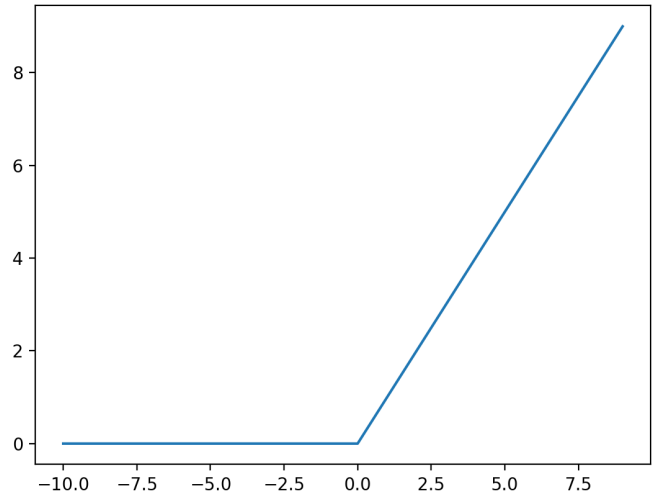
$$z = W^T h + b \qquad \text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

$$\log \text{softmax}(z_i) = z_i - \log \sum_j \exp(z_j)$$

Hidden units: rectified linear units

- Easy to optimize because they are similar to linear units
- Negative values are being set to zero
- Derivatives is non-zero when the unit is active

$$g(z) = \max\{0, z\}$$



Hidden units: maxout units

- Generalisation of the rectified linear unit
- The maxout unit can learn a piecewise linear, convex function with up to k pieces

$$g(z_i) = \max_{j \in \mathbb{G}^i} z_j$$

Hidden unit: logistic sigmoid and hyperbolic tangent

- Logistic sigmoid is one of the classical activation functions
- The two functions are related
- During learning the hyperbolic tangent typically performs better

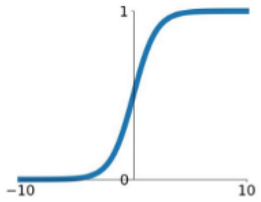
$$g(z) = \sigma(z) := \frac{\exp(z)}{1 + \exp(z)}$$

$$g(z) = \tanh(z) \qquad \tanh(z) = 2\sigma(2z) - 1$$

Hidden units: summary

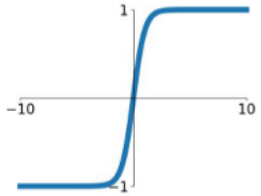
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



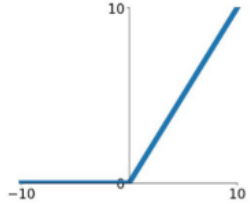
tanh

$\tanh(x)$



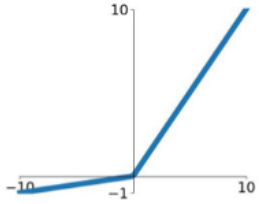
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

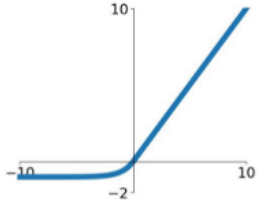


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

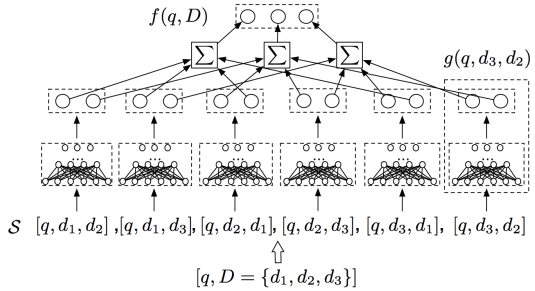


Architectures

- The architecture is now constructed as a sequence of layers
- A large number of different architectures has been developed for various different tasks, including detection and segmentation

$$h^{(1)} = g(W^{(1)T}x + b^{(1)})$$

$$h^{(2)} = g(W^{(2)T}h^{(1)} + b^{(2)})$$



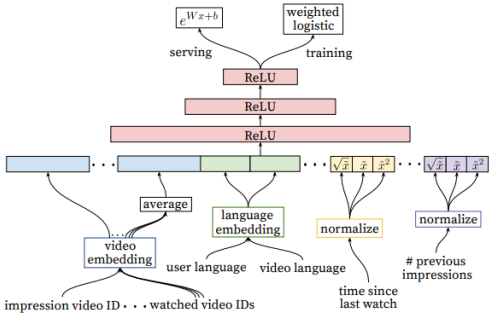


Figure 7: Deep ranking network architecture depicting embedded categorical features (both univalent and multivalent) with shared embeddings and powers of normalized continuous features. All layers are fully connected. In practice, hundreds of features are fed into the network.

Regularization

- Regularization is generally required in machine learning to build algorithms that no only perform well on the training data
- Many regularization strategies have been developed for traditional machine learning methods
- In some sense the regularization limits the capacity of the model

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta)$$

$$\Omega(\theta) = \frac{1}{2}||\theta||_2 \qquad \Omega(\theta) = ||\theta||_1$$

Optimization for training deep models

- A detailed discussion of the optimization methods is beyond the scope of this course
- In general, gradient based optimization method are being used to train deep neural networks
- In ML we try to optimize a performance metric indirectly by optimizing a cost function J

Optimization for training deep models

- In the context of supervised learning the cost function can be written as an average over the training set

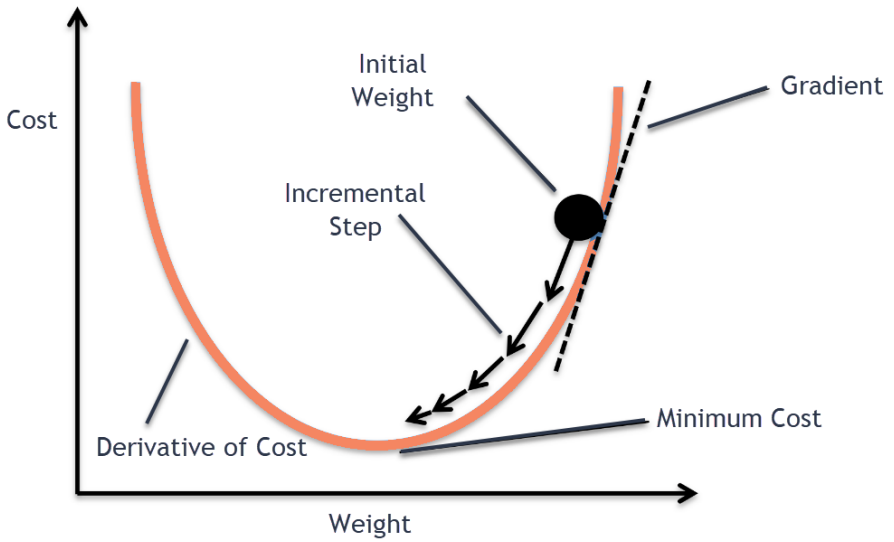
$$J(\theta) = \mathbb{E}_{(x,y)} L(f(x;\theta), y) = \frac{1}{N} \sum_i L(f(x^{(i)};\theta), y^{(i)})$$

- Minimizing this average training error is known as **empirical risk minimization.**

- This process can lead to overfitting to the training data

Stochastic gradient decent

- In practice a methods iterative methods such as stochastic gradient decent are being used. \
- In each incremental step a random subset of the training data (mini batch) is being used for estimating the gradient

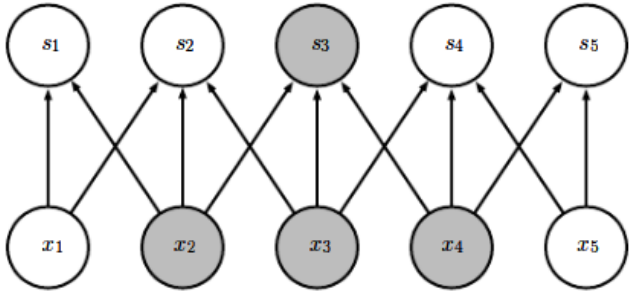


Convolutional networks

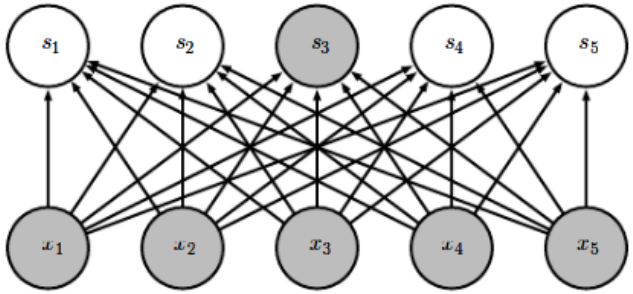
- Convolutional networks are networks that use convolution in place of general matrix multiplication in at least one of the layers
- They are particularly relevant to image analysis and signal processing problem
- Here the convolution operation you have already learnt is being used. However, the parameters of the filter or kernel are being learnt directly from the training data.

Convolutional network

Convolutional layers implement sparse interactions



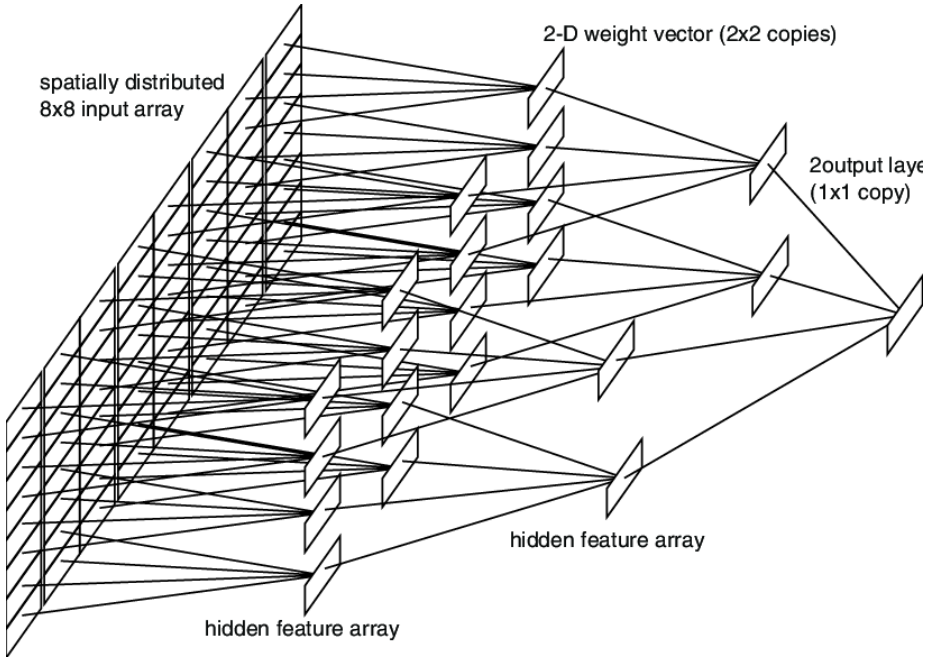
Convolution



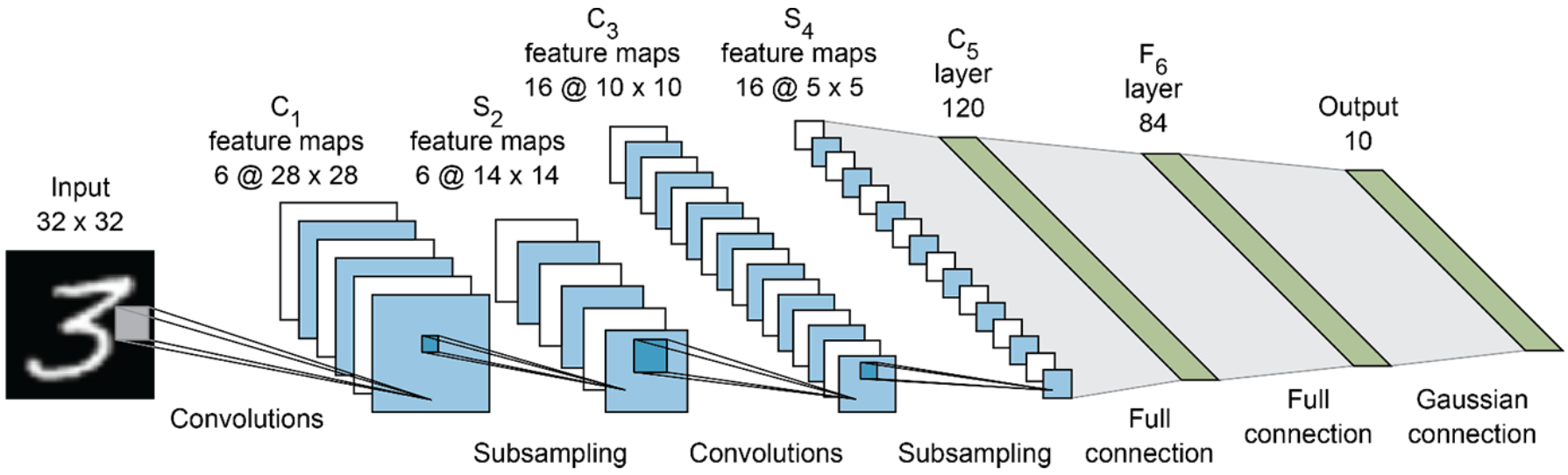
Matrix multiplication

Convolutional networks

Through the hierarchical architecture the network effectively interact with a large portion of the input



LeNet: A layered model for the classification of digits



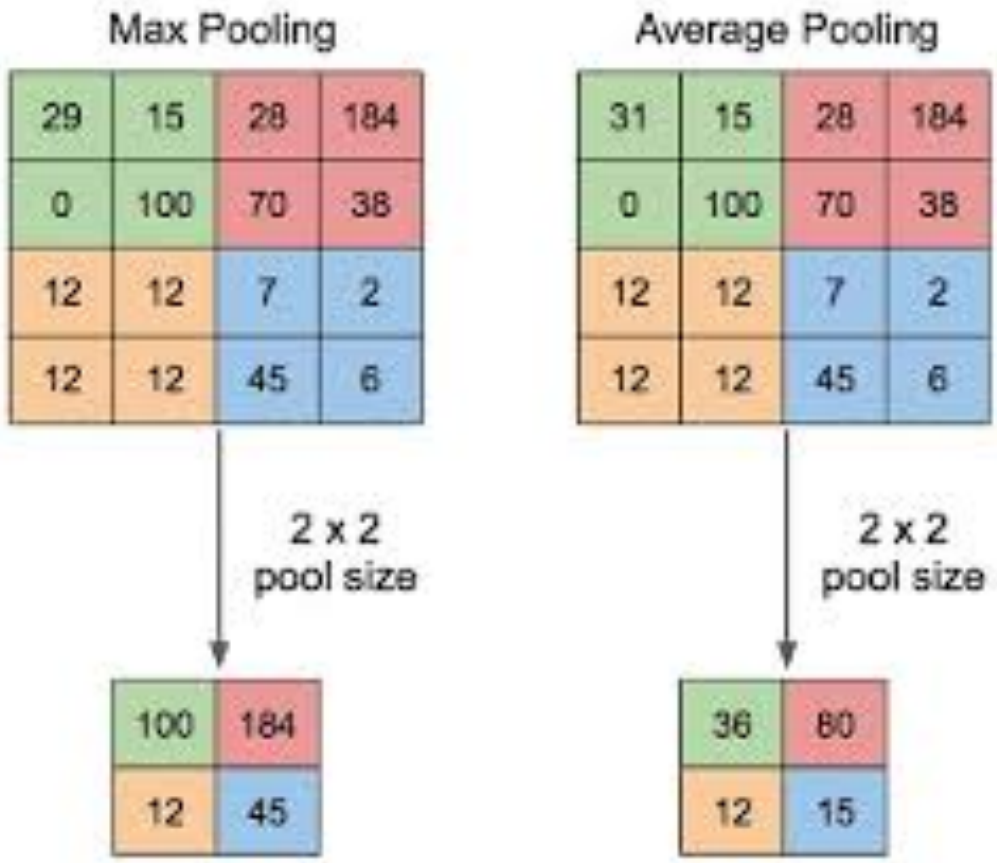
LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), pp.541-551.

Pooling operation

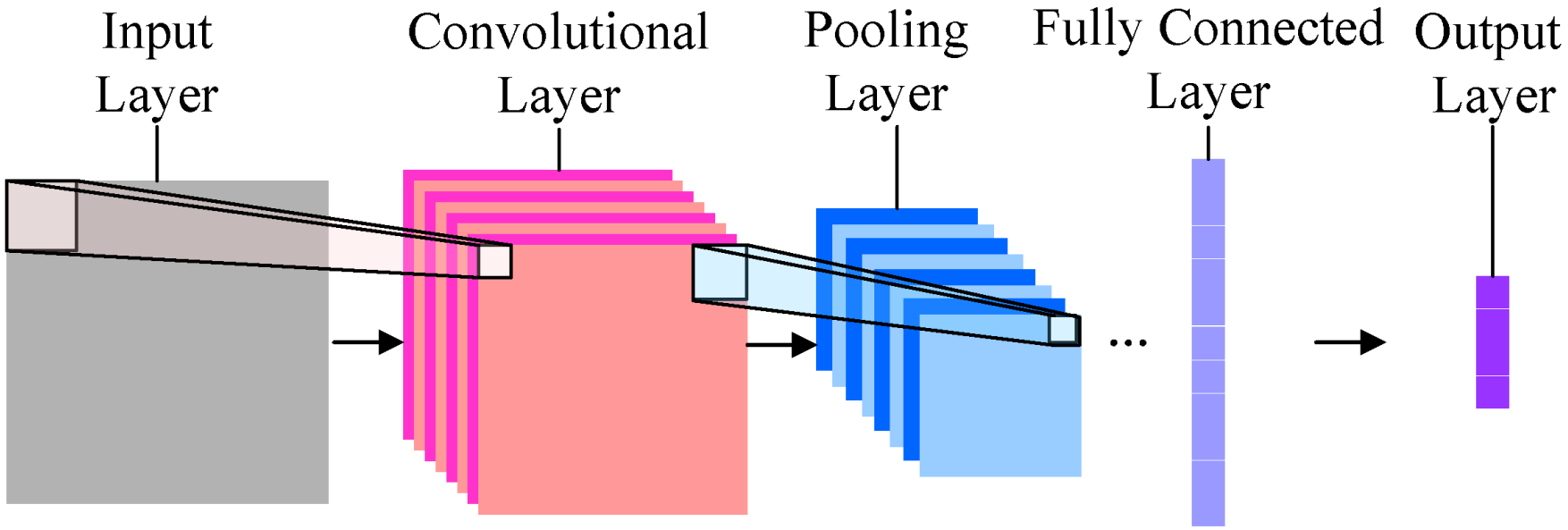
- A pooling function replaces the output of the network at a certain location with a summary statistic of the nearby outputs

- Pooling helps to make the representation approximately invariant to small translations of the input

Pooling


Max Pooling
29
15
28
184
0
100
70
38
12
12
7
2
12
12
45
6
2 x 2
pool size
100
184
12
45
Average Pooling
31
15
28
184
0
100
70
38
12
12
7
2
12
12
45
6
2 x 2
pool size
36
80
12
15

A typical convolution layer

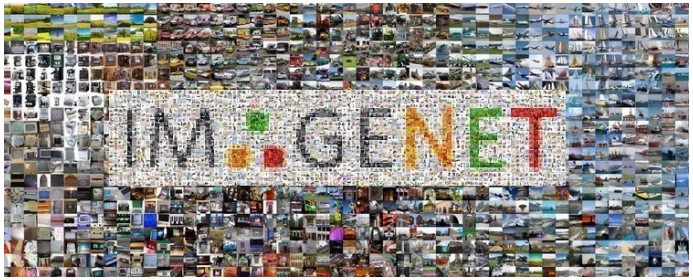


ImageNet – Large Scale Visual Recognition Challenge

The challenge evaluates algorithms for object localization and detection in videos. It includes the following tasks:

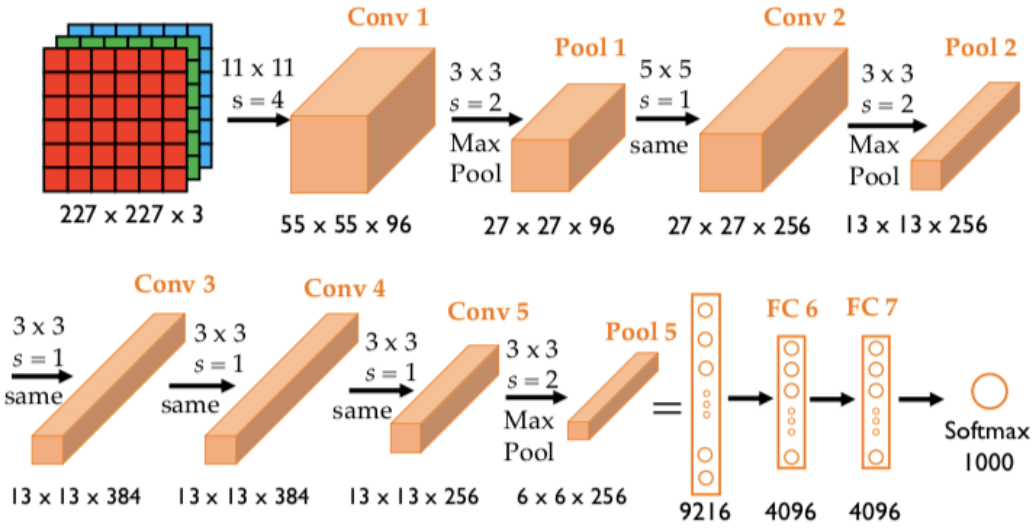
- Object localization for 1000 categories
- Object detection for 200 fully labelled categories
- Object detection from video for 30 fully labelled categories

The data consists millions for hand annotated images taken from flickr and other search engines



AlexNet

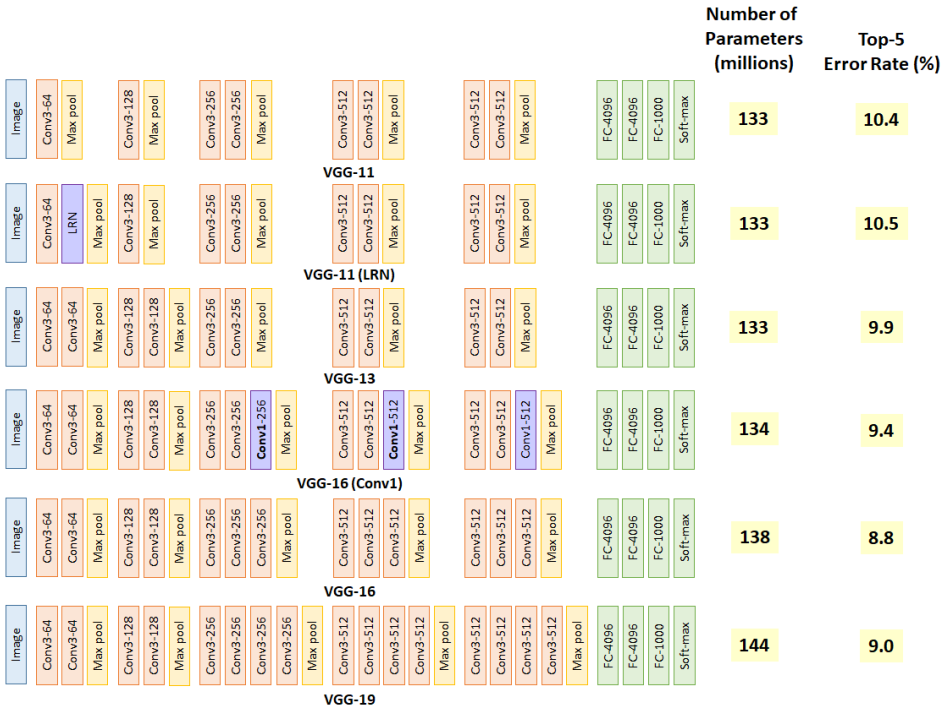
- Winner of the 2012 ImageNet Large Scale Recognition Challenge



Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

VGG Net Architectures

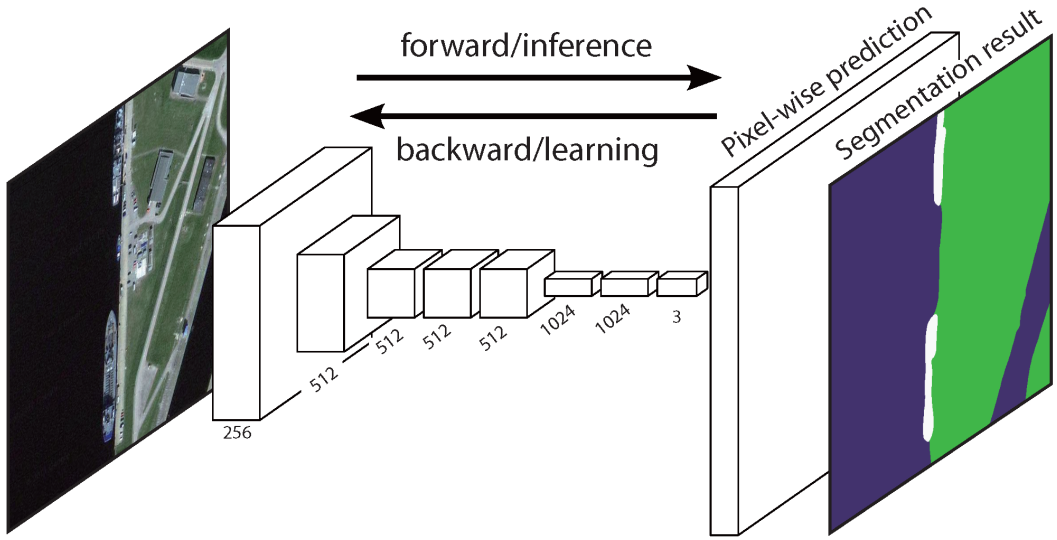
1st runner up in the ImageNet Challenge in 2014



Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

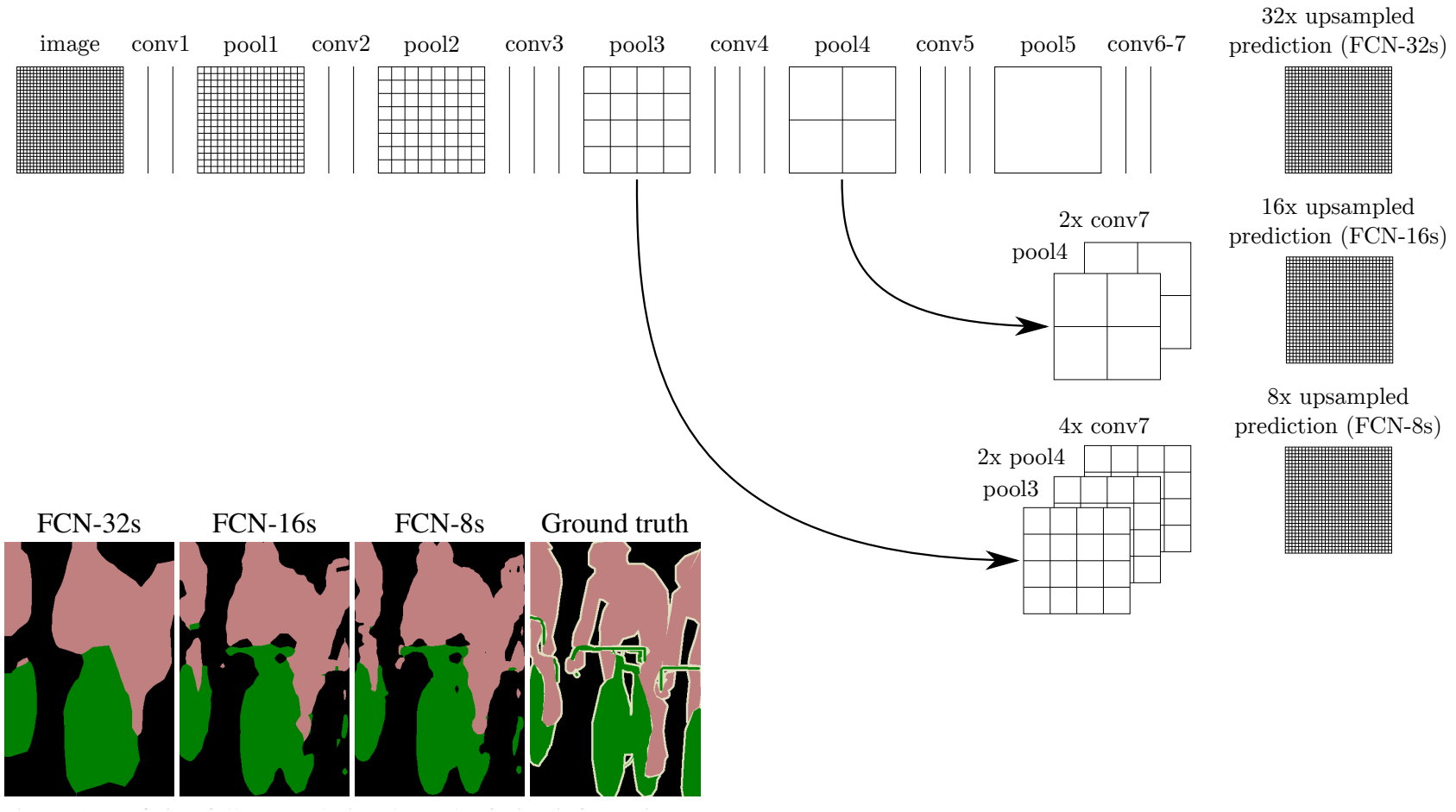
Semantic segmentation – fully convolutional networks

The original convolutional network learns a mapping from pixels to pixels via an upsampling step



Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

Fully convolutional networks



How can we utilize these innovations for biomedical imaging?

- Typically, we do not have access to a large set of annotated training images

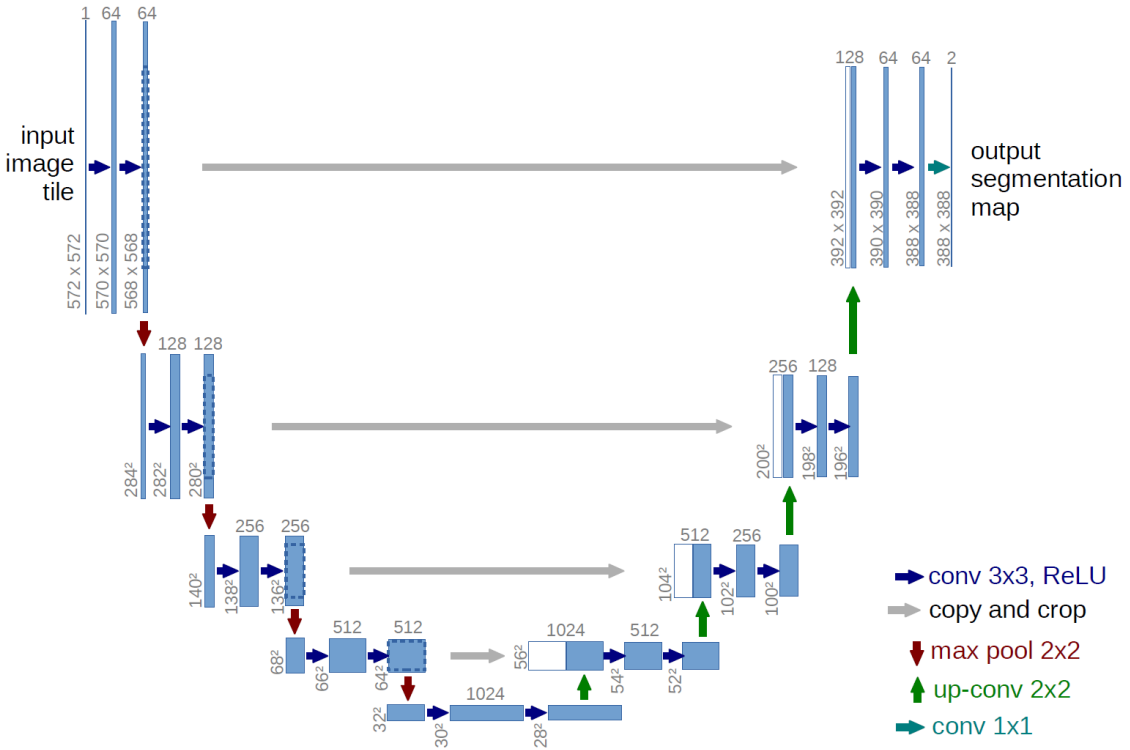
- Pixel level annotations are harder to obtain and require experts

U-Net

- An architecture that contains two paths: an encoder or contraction path and an expanding path
- Contraction path: designed to capture image context
- Expanding path: a symmetric path which uses transponded convolutions for precise localisation

Ronneberger, O., Fischer, P. and 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.

U-Net


input image tile
572 x 572
570 x 570
568 x 568
1 64 64
128 128
284²
282²
280²
256 256
140²
138²
136²
512 512
68²
66²
64²
1024
32²
30²
28²
1024 512
56²
54²
52²
512 256
104²
102²
100²
256 128
200²
198²
196²
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

U-Net examples

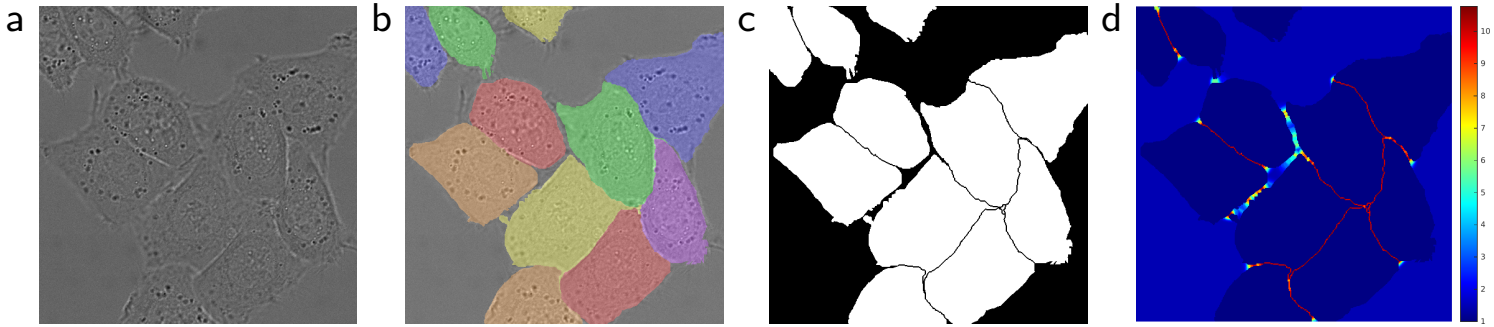


Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (**a**) raw image. (**b**) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (**c**) generated segmentation mask (white: foreground, black: background).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

U-Net examples

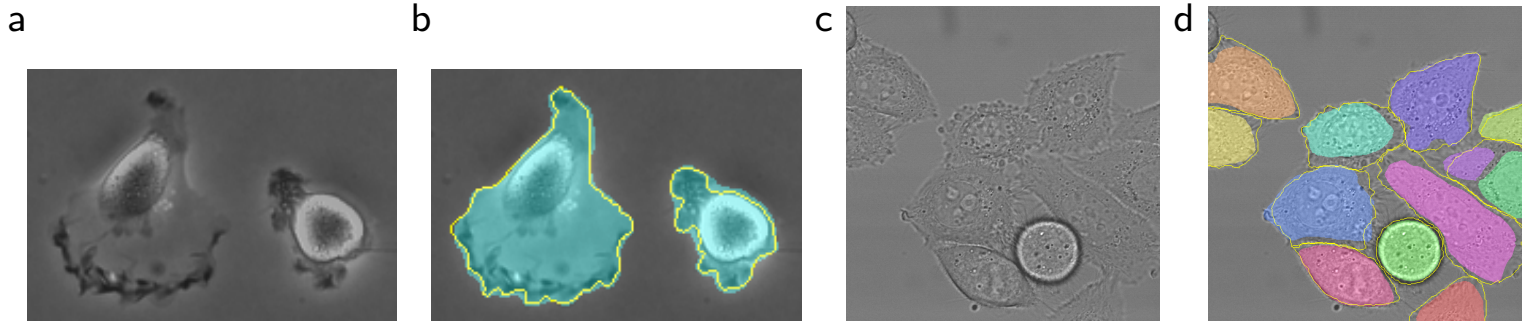
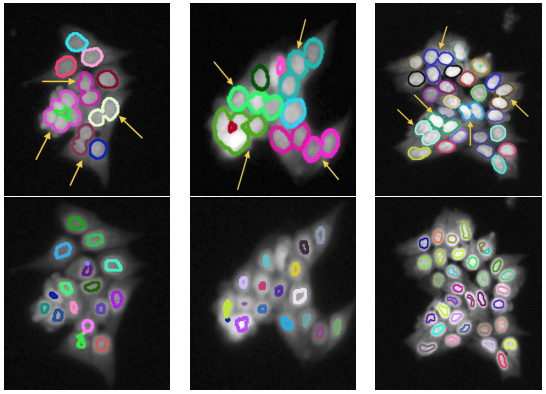


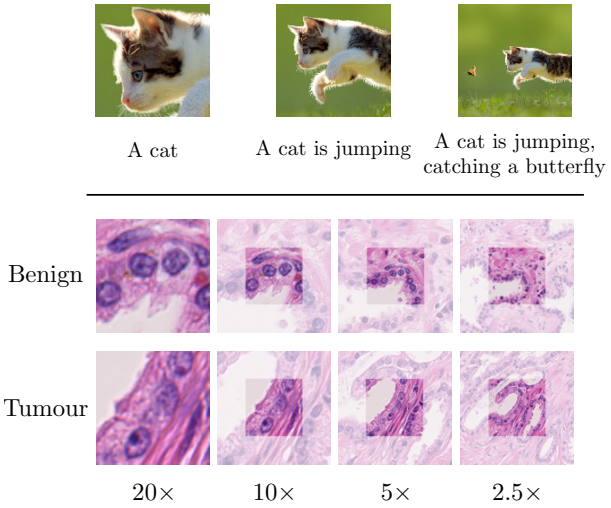
Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

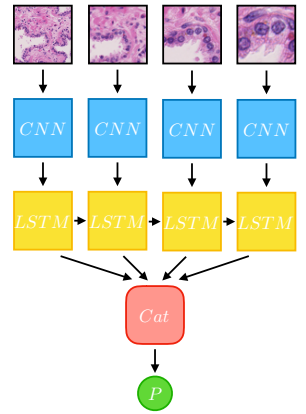
Open challenges in segmentation

- The segmentation of complex structures is still a problem
- Lack of contrast can make the segmentation task very challenging
- Segmenting touching cells is still difficult
- We still need to find ways to train models on small training datasets.

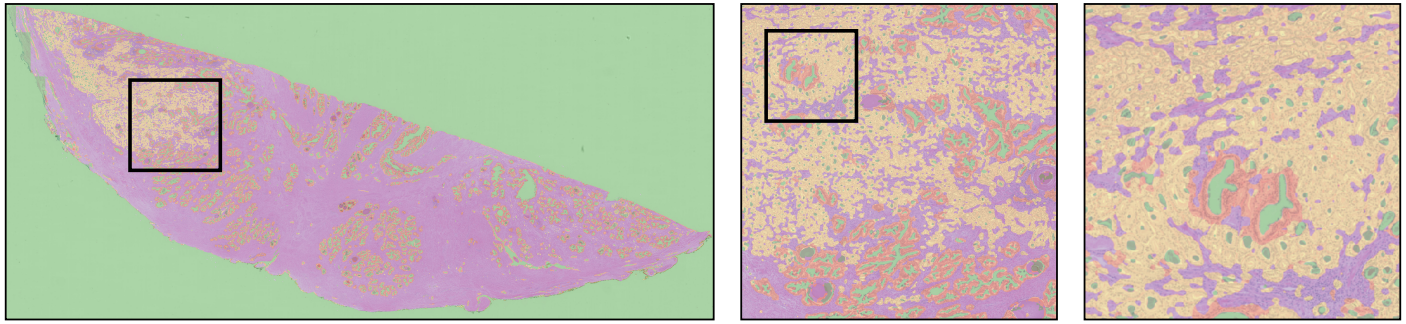


Whole slide imaging for histology imaging

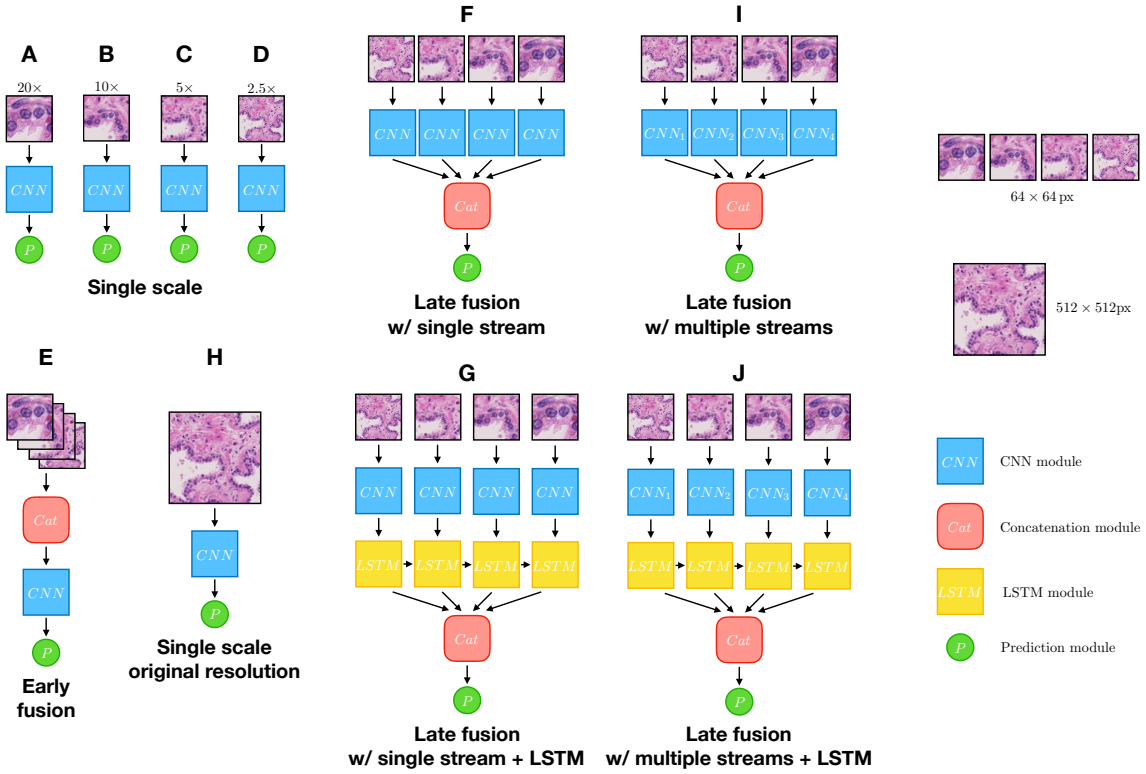




Late fusion
w/ single stream + LSTM



Comparison of achitectures



Model comparison

Dataset	Class	Method									
		A	B	C	D	E	F	G	H	I	J
Prostate	Lumen	0.728	0.663	0.705	0.716	0.739	0.738	0.748	0.713	0.722	**0.758**
	Stroma	0.797	0.855	0.849	0.790	0.875	0.869	0.884	**0.891**	0.862	0.883
	Benign	0.508	0.646	0.712	0.717	0.734	0.745	0.766	0.763	0.765	**0.782**
	Tumour	0.562	0.653	0.629	0.579	0.699	0.687	0.728	**0.746**	0.674	0.712
Breast	Normal	0.501	0.468	0.523	0.513	0.509	**0.603**	0.573	0.252	0.241	0.323
	Benign	0.453	0.468	0.482	0.444	0.410	0.369	0.423	**0.489**	0.333	0.437
	InSitu	0.468	0.476	0.486	0.533	**0.615**	0.614	0.581	0.286	0.311	0.452
	Invasive	0.401	0.477	0.430	0.540	0.557	0.548	0.576	0.520	0.446	**0.580**
Rank-sum (Prostate)		20	19	17	19	13	13	8	8	12	7
Rank-sum (Breast)		22	21	19	18	16	13	14	18	24	18
Total rank-sum		42	40	36	37	29	26	22	26	36	25
No. of parameters		7.2M	7.2M	7.2M	7.2M	7.3M	8.0M	10.1M	19.8M	28.9M	31.0M
Running time (s)		7.16	7.16	7.16	7.16	7.21	7.61	7.62	35.59	7.60	7.70

Deep learning for de-noising

- Image de-noising is another very important task in medical image processing

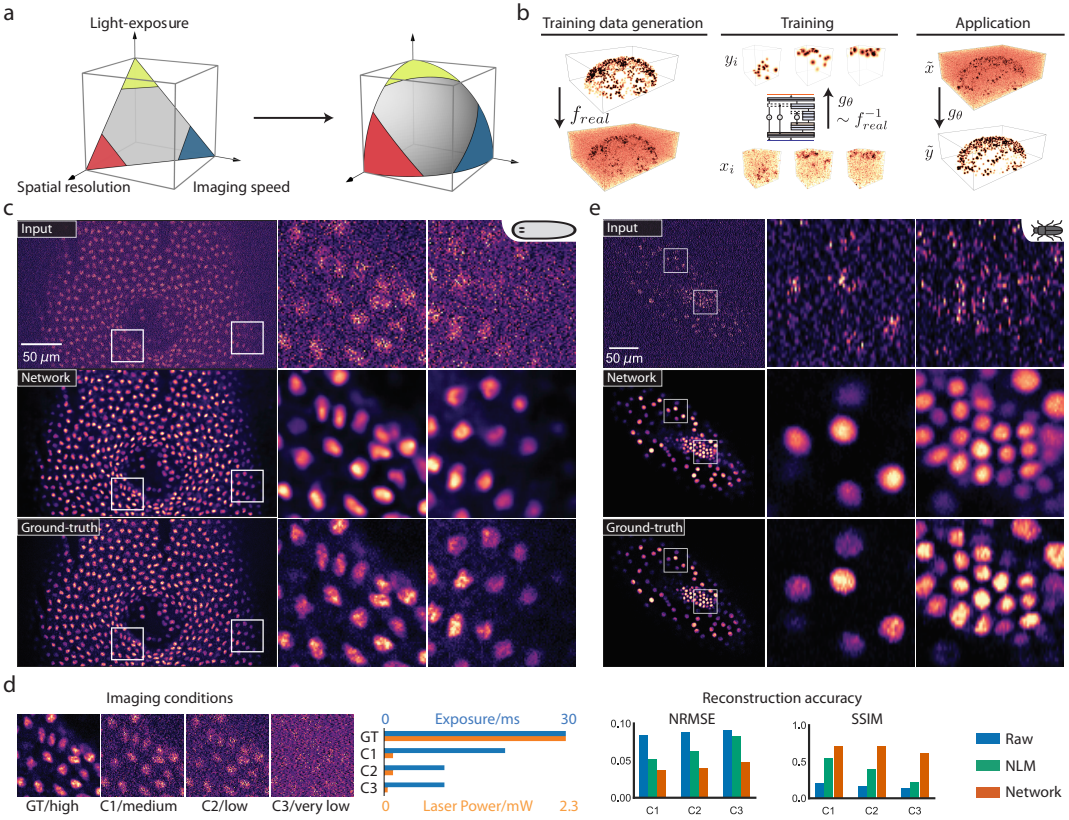
- It is often challenging to model the noise analytically, it would be an advantage if this could be learnt directly from training data

Content-aware image reconstruction

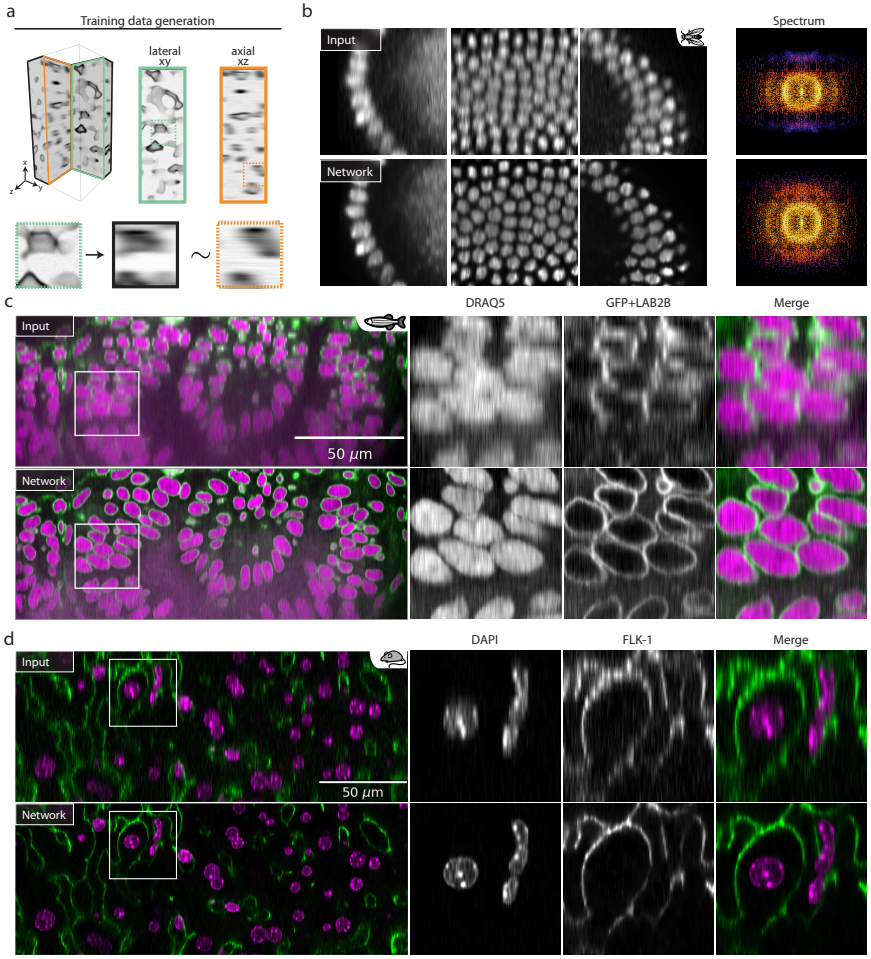
- Deep learning is used for image denoising, surface projection and recovery of isotropic resolution
- Training data is acquired through a set of specific experiments to train the model
- The proposed architecture uses two U-Net based sub-networks: first one performing the projection of voxel intensities, the second one for denoising

Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S. and Rocha-Martins, M., 2018. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature methods*, *15*(12), pp.1090-1097.

Content aware image reconstruction



Examples of 3D reconstructions



Open challenge

- Generalization to new data is still an open problem
- It is often not feasible to generate a sufficient amount of training data.
- Understanding why a model makes a certain decision is an area of active research

Refereneces

- Goodfellow, Bengio, Courville, Deep Learning [html-version]