ETH zürich

# Lecture “Machine Learning for Healthcare” (261-5120-00L)

## Basics of ML for Medical Image Analysis

**Julia Vogt & Valentina Boeva & Gunnar Rätsch**

Institute for Machine Learning, Computer Science Department

@gxr @gxrlab

#DataScience #PrecisionMedicine #ClinicalData

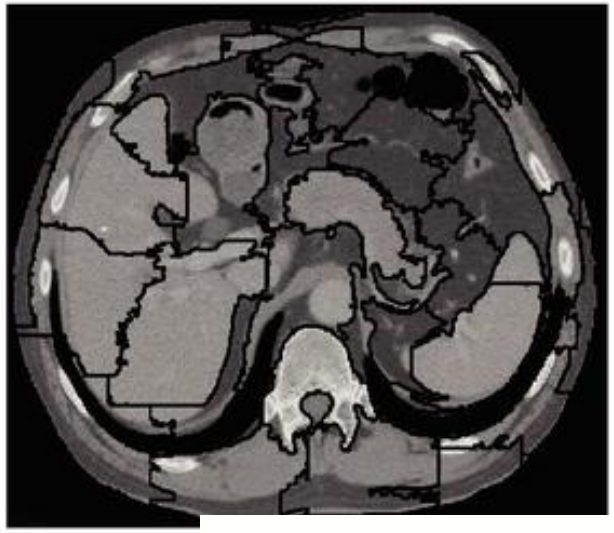
BIOMEDICAL INFORMATICS DINFK DBIOL UniversityHospital Zurich

# Topics for Today

- Medical Image Data
- Typical medical image analysis problems
- Segmentation
  - Superpixels
  - Markov Random Fields
- Image Classification
  - Convolutional Neural Networks
- Application in Digital Pathology

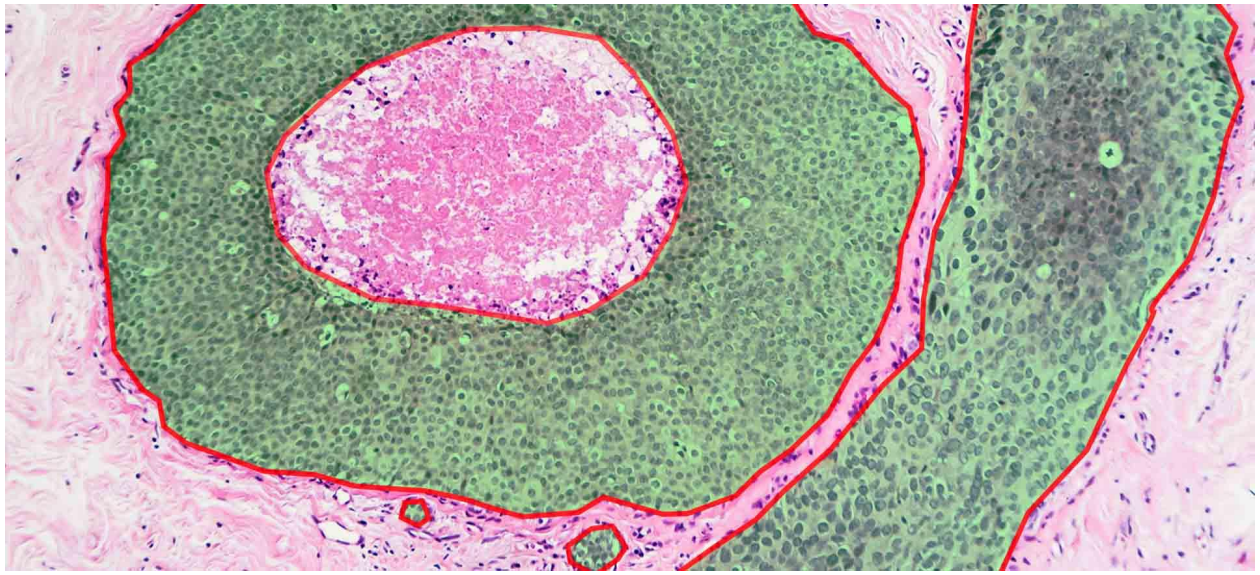
# Analysis of Medical Images

Radiology (2d, 3d, low res.)

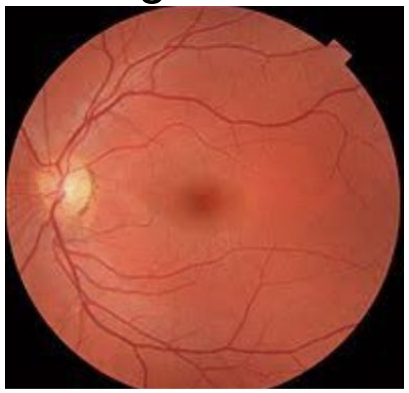


MRI
CT

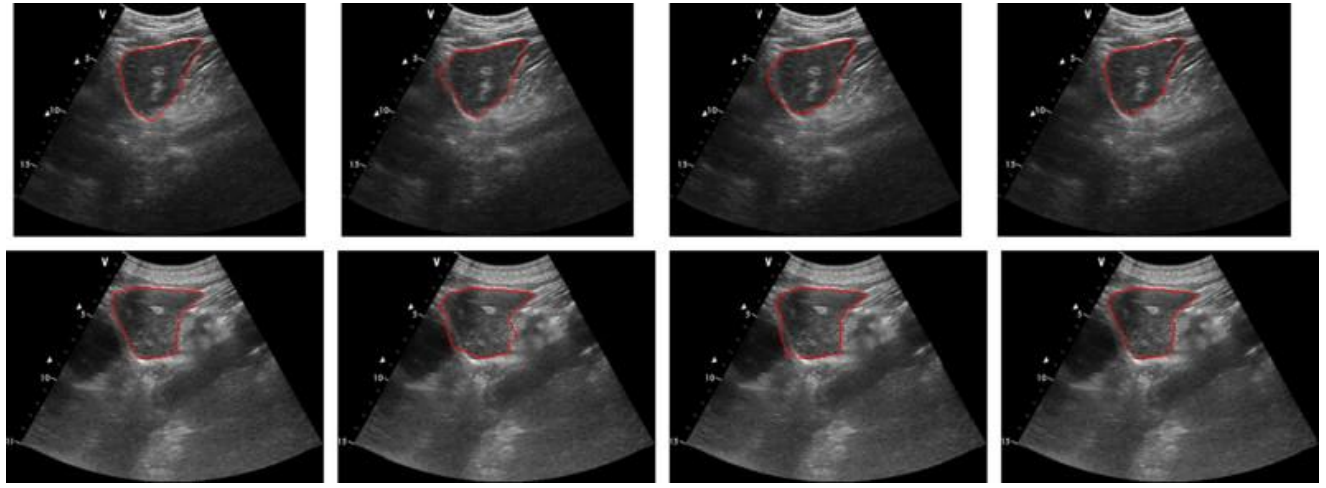
Pathology (2d, high resolution)



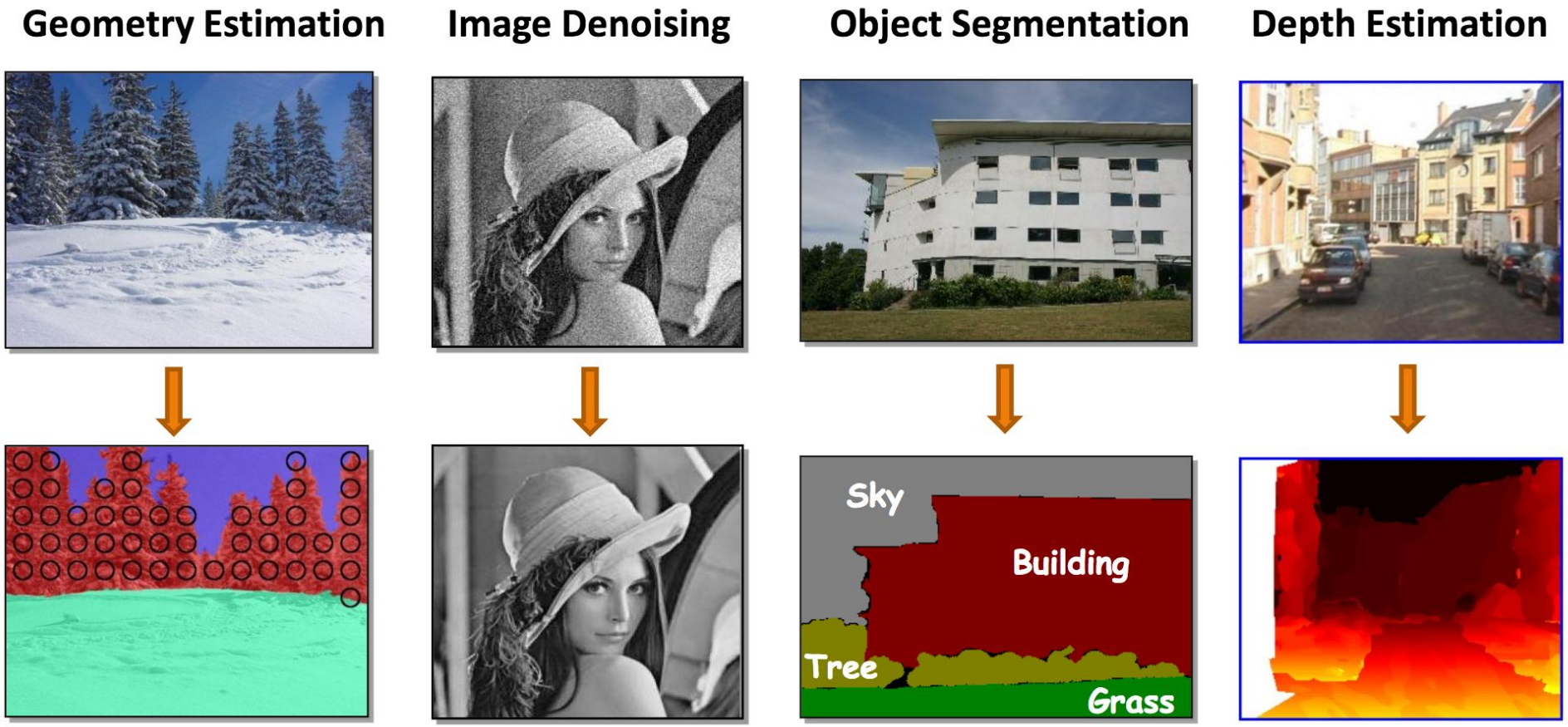
Retina Fundus
2d high resolution



Ultrasonic (low resolution, temporal)



# "Zoo" of Image Analysis/Labeling Problems



# Image Analysis Problems

Non-complete list of (medical) image analysis problems

- Image classification (“normal vs. diseased eye fundus”)
- Image registration (“register multiple images of same patient”)
- Image labeling (“find cancer cells”)
- 3d object reconstruction (“heart model”)
- Image segmentation (“identify vasculature”, more next page)
- ...

Image analysis is a very broad field with many challenges. It would need its own lecture on that topic. Actually multiple lectures. We can only cover some aspects and only some of the basics.

# Segmentation in Medical Imaging

- Determination of the volumes of abdominal solid organs and focal lesions has great potential importance:
  - Monitoring the response to therapy and the progression of tumors and preoperative examination of living liver donors are the most common clinical applications of volume determination.
- MRI volumetry of the hippocampus can help distinguish patients with Alzheimer's disease from elderly controls with a high degree of accuracy (80%-90%).
- In order to be able to detect and quantify *vascular diseases* one of the first step is the segmentation of the *vasculature.*

# Segmentation

Segmentation of an image entails the division or separation of the image into regions of similar attribute.

- Categorization of different segmentation methods:
  - **Boundary-based**: optimum boundary, active boundary, live wire, level sets
  - **Shape Model-based**: Manual tracing, live wire, active shape/appearance, M-reps, atlas-based
  - **Region-based**: clustering, kNN, CM, FCM, fuzzy connectedness, MRF, graph cut, watershed, optimum partitioning

# Superpixel

- Algorithms in computer vision use the pixel-grid as the underlying representation.
- The pixel-grid is not a natural representation of visual scenes, it is rather just an "artifact" of a digital imaging process.
- It would be more natural to work with perceptually meaningful entities obtained from a low-level grouping process.
- **Superpixels** are essentially the visually homogeneous regions of an image, that were acquired by partitioning the image into N regions, where the pixels within a region share some low-level property (color, texture etc.)

# Superpixels

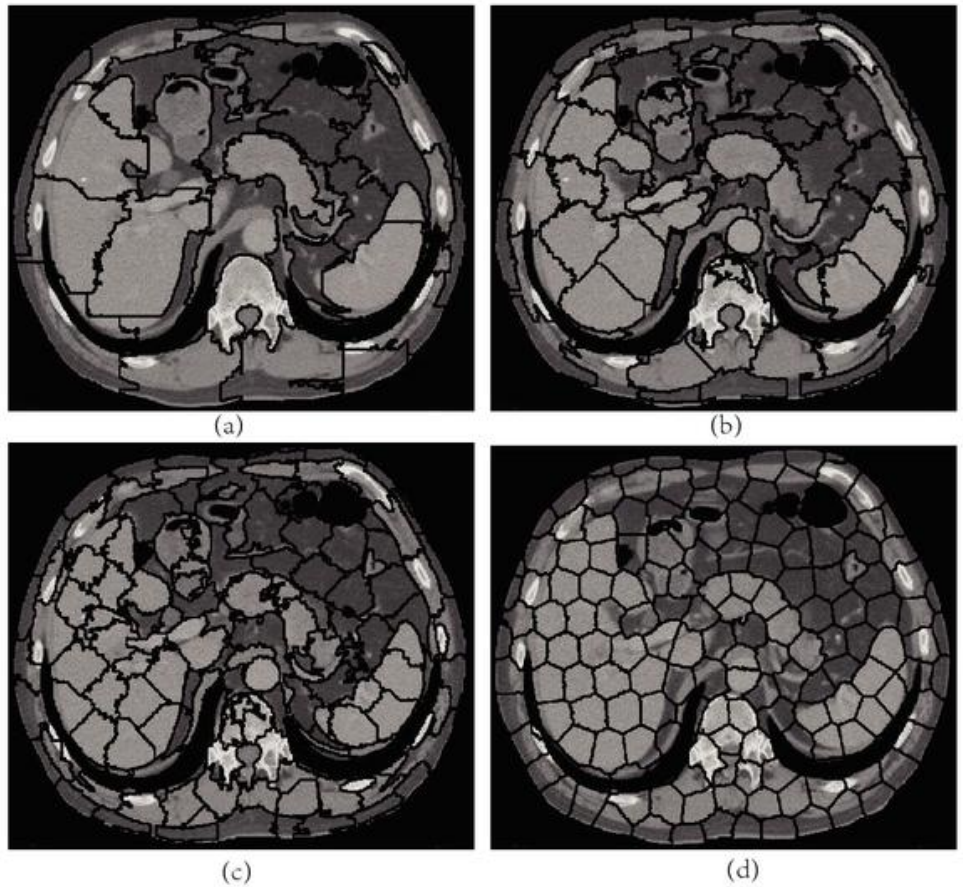
Superpixels images of different superpixels number Ks and different distance functions

(a) K = 50 (Euclidean distance),

(b) K = 100 (Euclidean distance),

(c) K = 200 (Euclidean distance),

(d) K = 200 (Mahalanobis distance).




Source: L. Zhang et. al. An improved method for pancreas segmentation using SLIC and interactive region merging

# Superpixel properties

- It is **computationally** efficient: it reduces the complexity of images from hundreds of thousands (millions) of pixels to only a few hundred (thousand) superpixels.
- It is also **representationally** efficient: pairwise constraints between units, while only for adjacent pixels on the pixel-grid, can now model much longer-range interactions between superpixels.
- The superpixels are **perceptually meaningful**: each superpixel is a consistent unit, i.e. all pixels in a superpixel are most likely uniform in, color or texture.
- It is **near-complete**: since superpixels are results of an over-segmentation, most structures in the image are conserved. There is very little loss in moving from the pixel-grid to the superpixel map.

# Simple Linear Iterative Clustering [SLIC] Outline

- SLIC is a simple and efficient method to partition an image in visually homogeneous regions.
- It is based on a spatially localized version of k-means clustering.
- Each pixel is associated to a feature vector:

$$\Psi(x, y) = [\lambda x, \lambda y, I(x, y)]$$

  where I(x,y) is the pixel value(s) of the image at the given location, λ coefficient balances the spatial and appearance components of the feature vectors, imposing a degree of spatial regularization to the extracted regions.
- Using these feature vectors **k-means clustering** is applied and pixels assigned to the same cluster will form a superpixel.

# Simple Linear Iterative Clustering [SLIC] Algorithm

- Input parameters
  - **Region Size (RS)**: the nominal size of the regions (superpixels)
  - **Regularizer (R)**: the strength of the spatial regularization
- The image is first divided into a grid with step **RS**.
- The center of each grid tile is then used to initialize a corresponding k-means.
- The acquired k-means centers and clusters are refined by using the k-means Lloyd algorithm.
- The parameter regularizer sets the trade-off between clustering appearance and spatial regularization, which is obtained by setting

$$\lambda = \frac{RS}{R}$$

  in the definition of the feature ψ(x,y).
- After the k-means step, SLIC optionally removes any segment whose area is smaller than a given threshold by merging them into larger ones.

# Resulting Superpixels

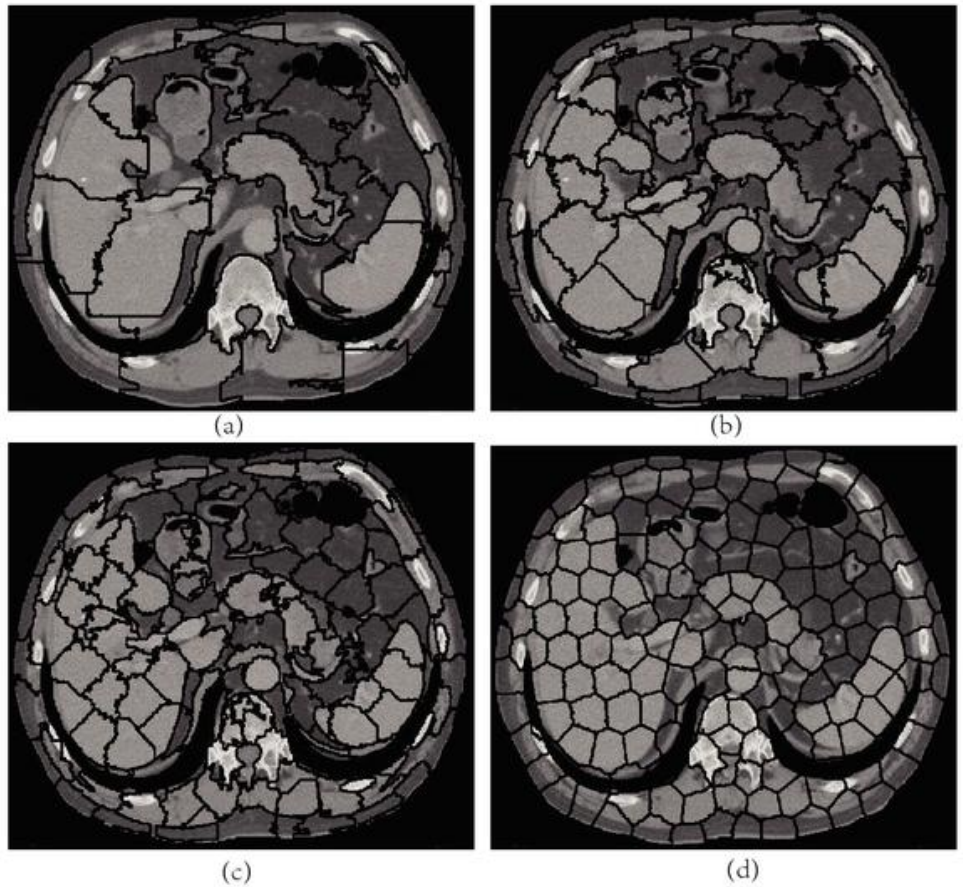
Superpixels images of different superpixels number Ks and different distance functions

(a) K = 50 (Euclidean distance),

(b) K = 100 (Euclidean distance),

(c) K = 200 (Euclidean distance),

(d) K = 200 (Mahalanobis distance).




Source: L. Zhang et. al. An improved method for pancreas segmentation using SLIC and interactive region merging

# Image Segmentation -> Labelling Pixels

- Labellings highly structured
- Labels highly correlated with very complex dependencies
- Independent label estimation too hard
- It is desired that the whole labelling should be formulated as one optimisation problem.
- High resolution images:
  - Hard to train complex dependencies
  - Optimisation problem is hard to infer

# Segmentation as an Energy Minimization Problem

- $E_{data}$ assigns non-negative penalties to a pixel location *i* when assigning a label to this location.
- $E_{smooth}$ assigns non-negative penalties by comparing the assigned labels at adjacent positions *i* and *j*

This optimization model is characterized by local interactions along edges between adjacent pixels, and often called MRF (Markov Random Field) model.

# Markov Random Field

- MRF is a graphical model over an undirected graph (G=(V,E)) positivity property (P(x) > 0) and Markov property.
  - Set of random variables linked to nodes: $\{x_i \in V\}$
  - Set of neighbored random variable: $N(x_i) = \{x_j \mid j \in N_i\}$
  - Markov property: $P(x_i \mid x_{V-\{i\}}) = P(x_i \mid x_{Ni})$

- Pairwise MRFs:

$$P(\mathbf{x}) = \frac{1}{Z} exp(-E(\mathbf{x}))$$

$$E(\mathbf{x}) = \sum_{i \in V} \psi_i(x_i) + \sum_{i \in V, j \in N_i} \psi_{i,j}(x_i, x_j)$$

# Example: Foreground / Background Estimation

- $x_i = 0 \rightarrow$ i is in background (to be determined)
- $x_i = 1 \rightarrow$ i is in foreground (to be determined)
- Data term (i=1,...,n):

$$\psi_i(0) = -\log P(x_i \in BG)$$
$$\psi_i(1) = -\log P(x_i \in FG)$$

Probabilities are estimated using FG / BG colour models (from pretrained model)

- Smoothness term (i,j=1,...,n):

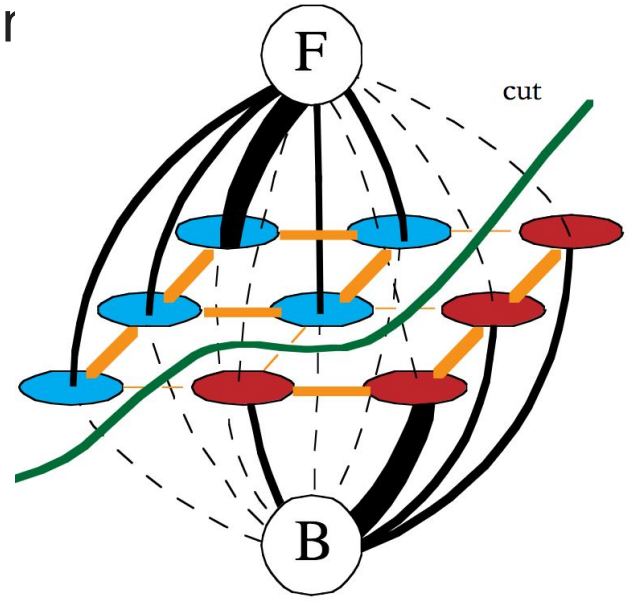
$$\psi_{ij}(x_i, x_i) = K_{ij}\delta(x_i \neq x_j)$$
$$K_{ij} = \lambda_1 + \lambda_2 \exp(-\beta(I_i - I_j)^2)$$

Intensity dependent smoothness

- Looking for $\mathbf{x}^* \in \{0,1\}^n$ that minimizes E(x), with fixed background/foreground labels

# Foreground / Background Estimation

- $x^* = \text{argmin}_x E(x)$
- This optimization problem can be solved by transforming the energy function into a min-cut/max-flow problem and solve it (S="F", T="B")
- **Max-flow min-cut theorem.**
  - The maximum value of an S-T flow is equal to the minir capacity over all S-T cuts.
- Ford–Fulkerson algorithm to compute the maximum flow




- Energy optimization equivalent to graph min-cut
- Cut: remove edges to disconnect F from B
- Minimum: minimize sum of cut edge weight

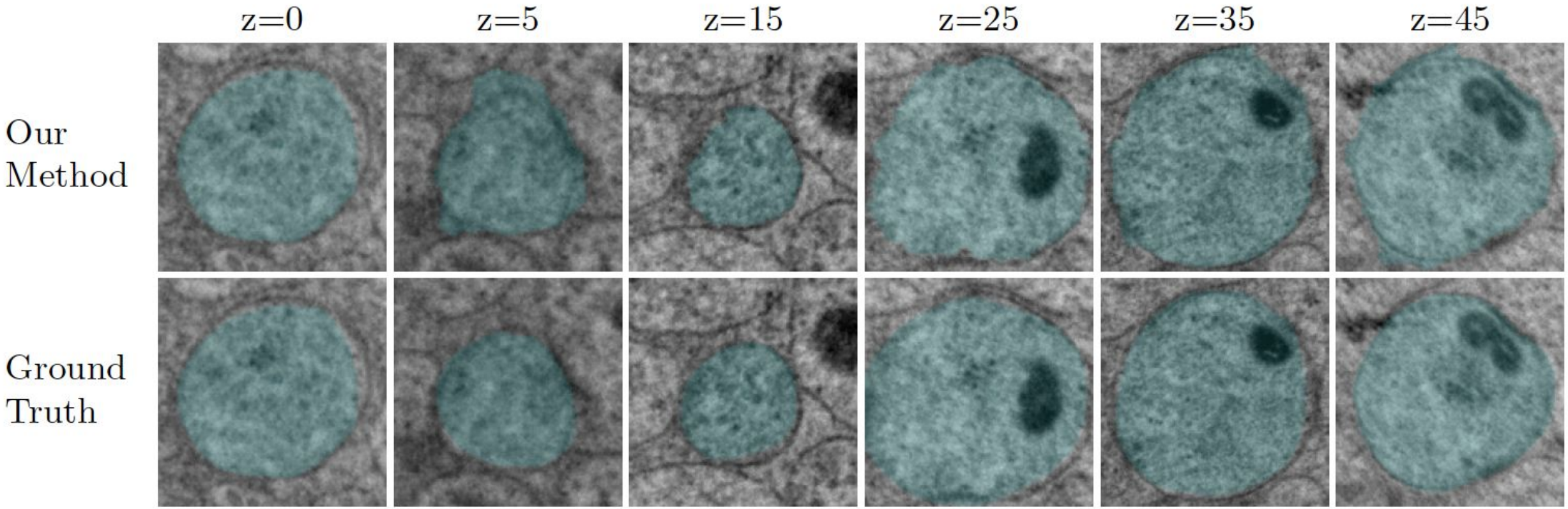
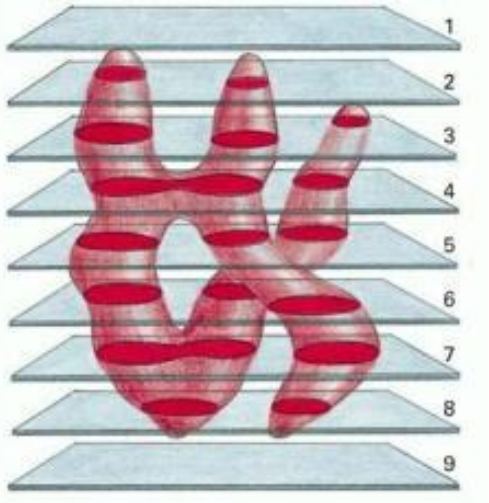
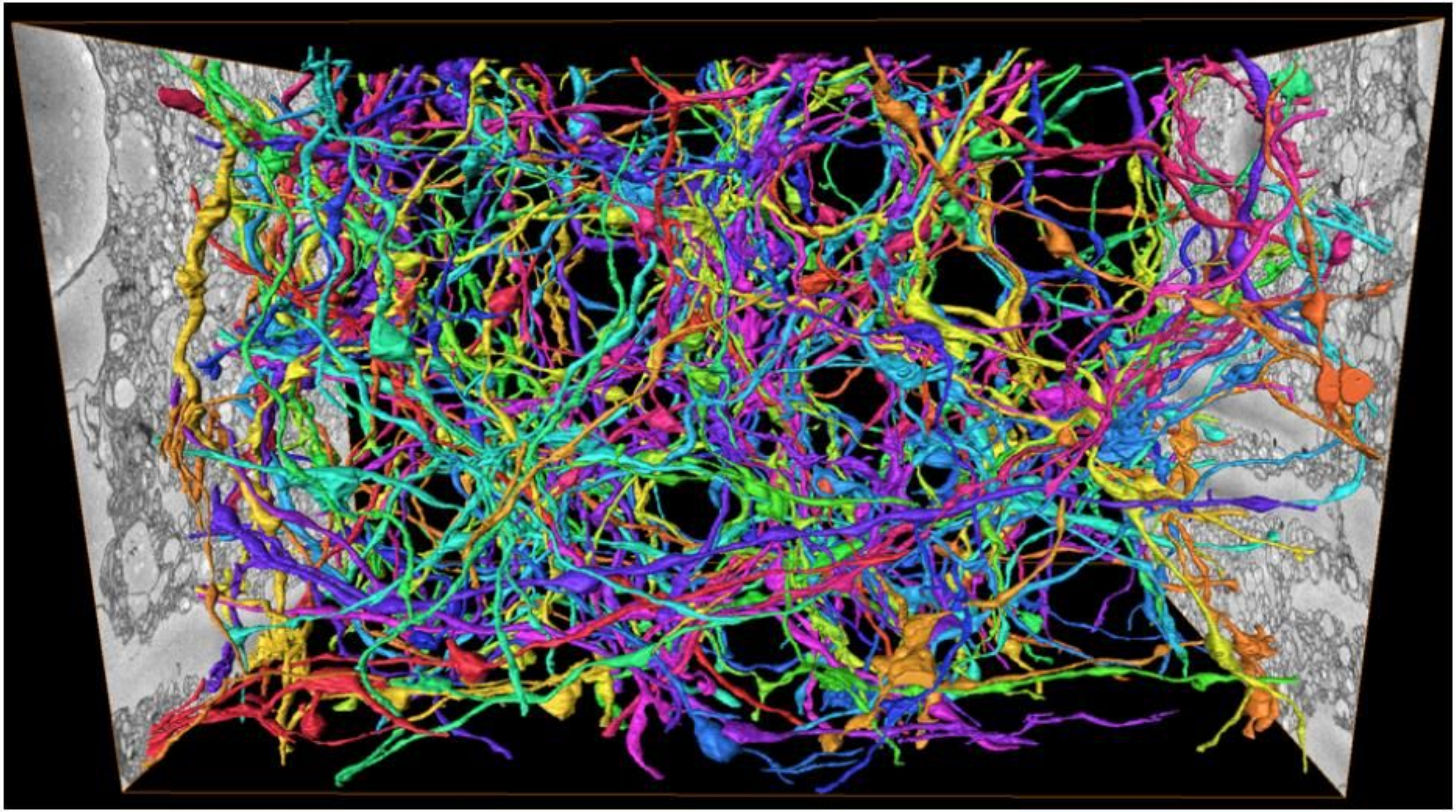



**Figure 3.11.** Segmentation result for neuron with ID #39828




s

Behrouz Tajoddin, M.Sc. thesis, 2011



# Topic 2: Image Classification


Caltech 101 dataset
Fei Fei et al., 2004

# Neural Networks for image analysis

## Neuron

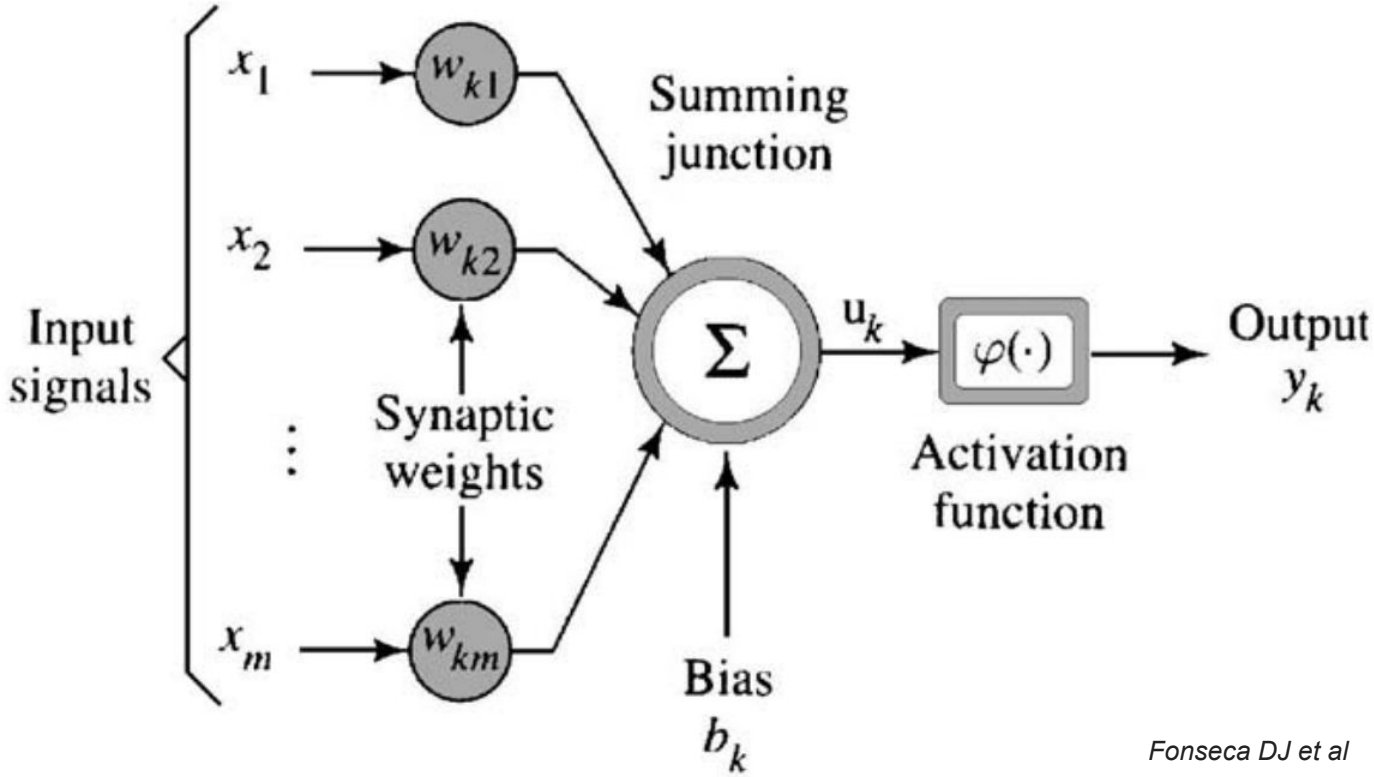
- **activation (weighted sum of inputs+bias)**

$$u_k = \sum_{i=1}^{m} w_{ki} x_i + b_k$$

- **activation function $\varphi$**
  - usually non-linear
  - e.g. tanh, sigmoid and ReLU

- **output**

$$y_k = \varphi(u_k)$$

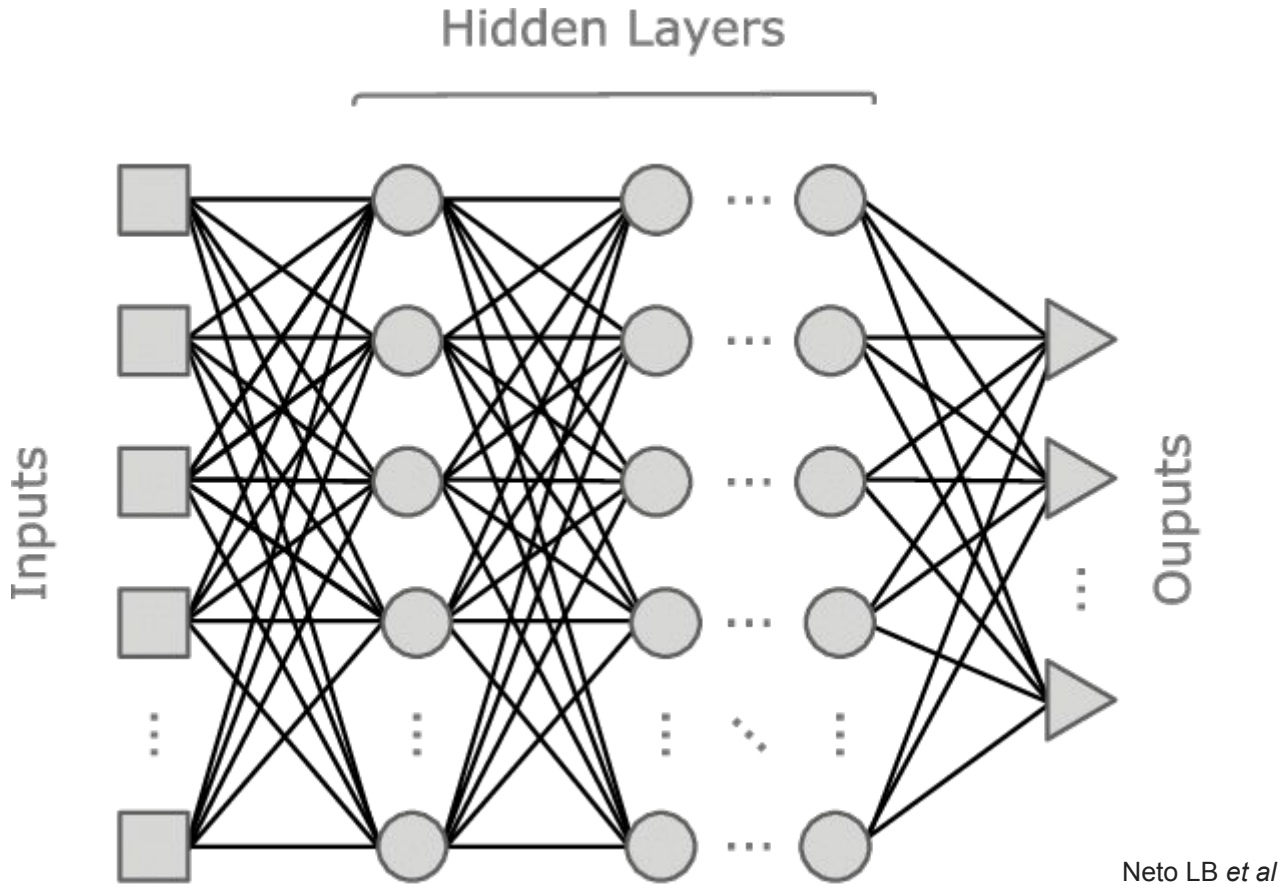



*Fonseca DJ et al*

# Neural Networks

## Regular Neural Networks

- **one input layer**
- **multiple hidden layers**
  - the more layers, the deeper the model
  - # neurons at each hidden layer can be different
- **one output layer**




Neto LB *et al*

One connection = one parameter

Fully-connected NNs have a huge number of parameters.

E.g., For input images with size 200x200x3, a fully-connected neuron in the first layer has 200x200x3=120,000 weights.

# Drawbacks of regular neural networks

- **Huge number of parameters**
  - do not scale well to large images
  - computationally heavy
  - local minima during training
  - overfitting

- **Make no assumption on the locality of pixel dependencies**
  - the nature of image: neighboring pixels have higher dependencies than pixels far away
  - regular neural networks are unable to extract local features using only global weighted sum

  **=> Convolutions to build-in "locality"**

# How do convolutions work?

## Convolutional layers

- consist of a set of small-size filters
- extract local features from the input layer, or outputs from the previous layer

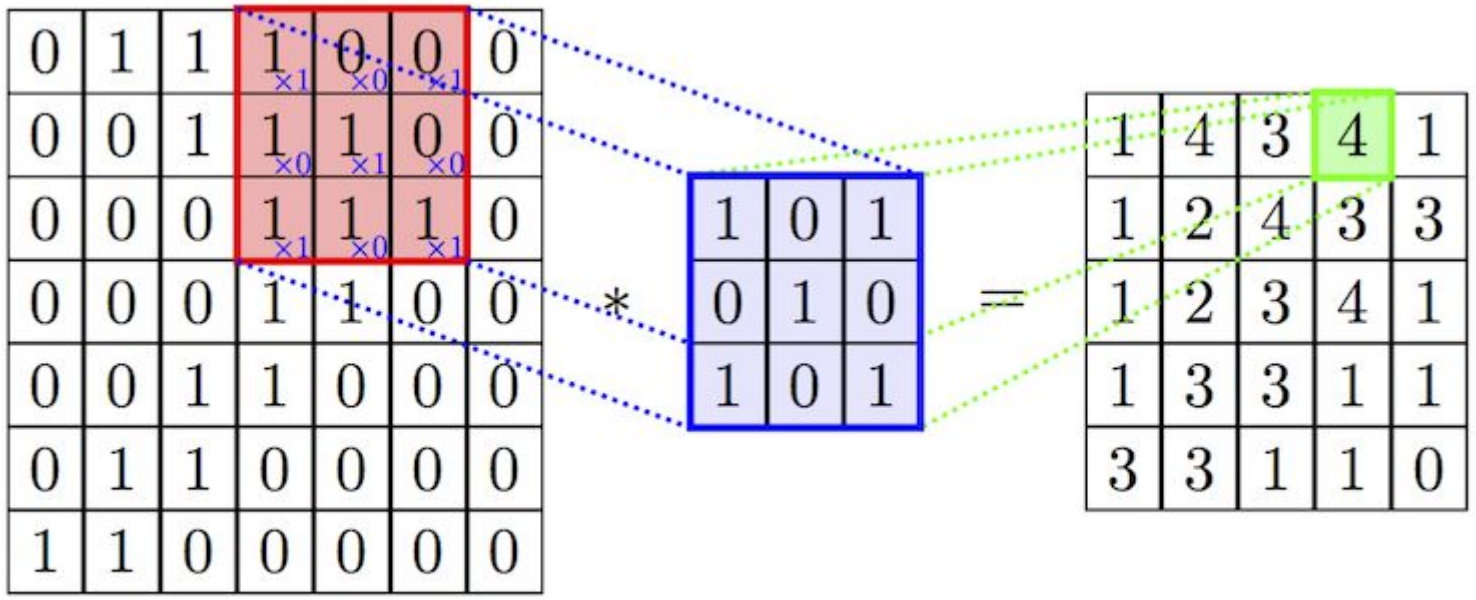
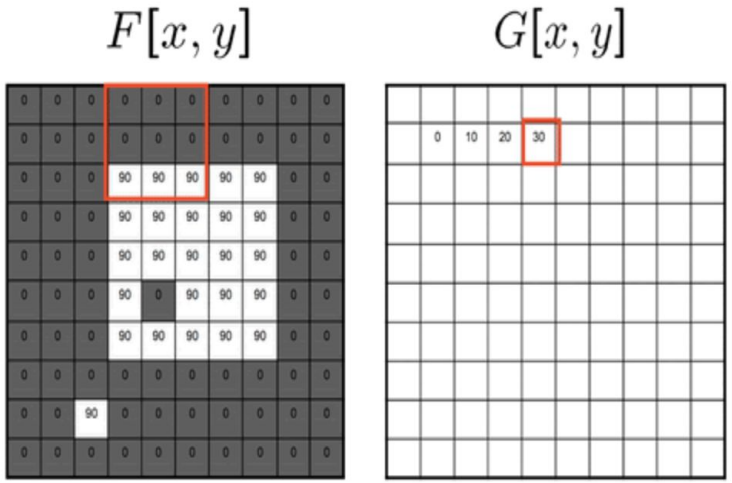



**Image** **Filter**

*Petar Veličković, Cambridge Spark*

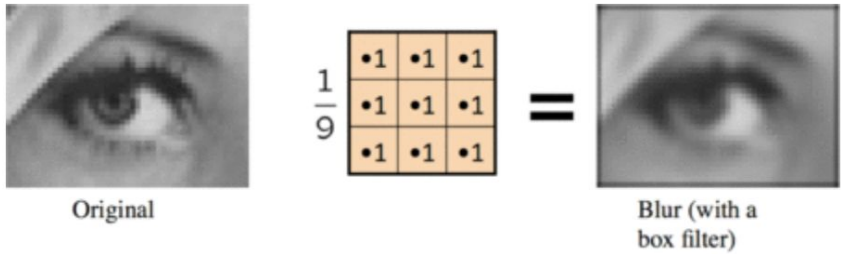
# Convolutional Filter Examples I




Note the edge artifact.*

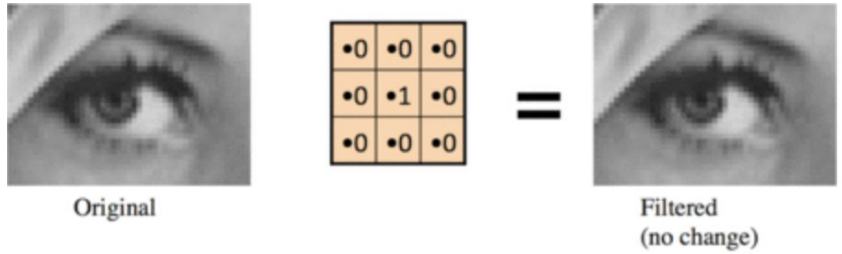
"2D Convolution"

Smoothing filter




Note the edge artifact.*

Identity filter




Hundreds of other filters that have been developed of the last decades for specific needs, including denoising, sharpening etc.

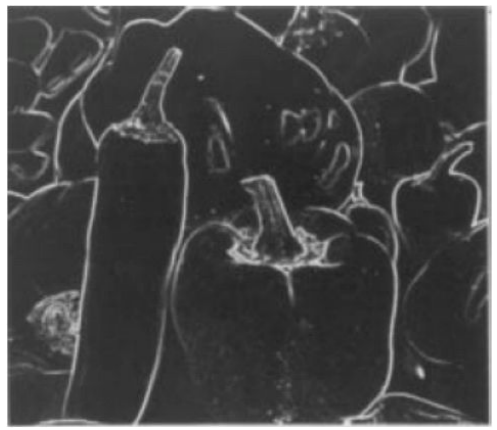
# Convolutional Filter Examples II

## Examples of commonly used filters in image processing

**Original image**

**Robert Cross**

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



**Prewitt**

$$\frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

# Convolutional Filter Examples III

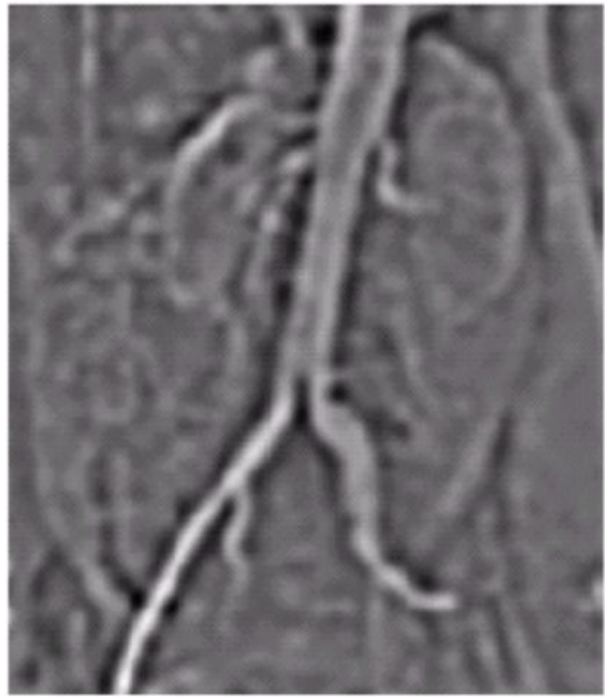
## Examples of commonly used filters in image processing

**Original image**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | −1 | 0 | 0 |
| 0 | −1 | −2 | −1 | 0 |
| −1 | −2 | 16 | −2 | −1 |
| 0 | −1 | −2 | −1 | 0 |
| 0 | 0 | −1 | 0 | 0 |

**Laplacian of Gaussian (LoG)**

**LoG filtering**



**The filter parameters in convolutional neural networks are learned not pre-defined.**

# Convolutional Layers

Images: multiple channels (e.g. $3$ color channels, RGB)

Define **window size**, e.g. $3 \times 3$, $5 \times 5$, ... = **input dimensionality**

Chose **number of channels** $k$ = **layer width**

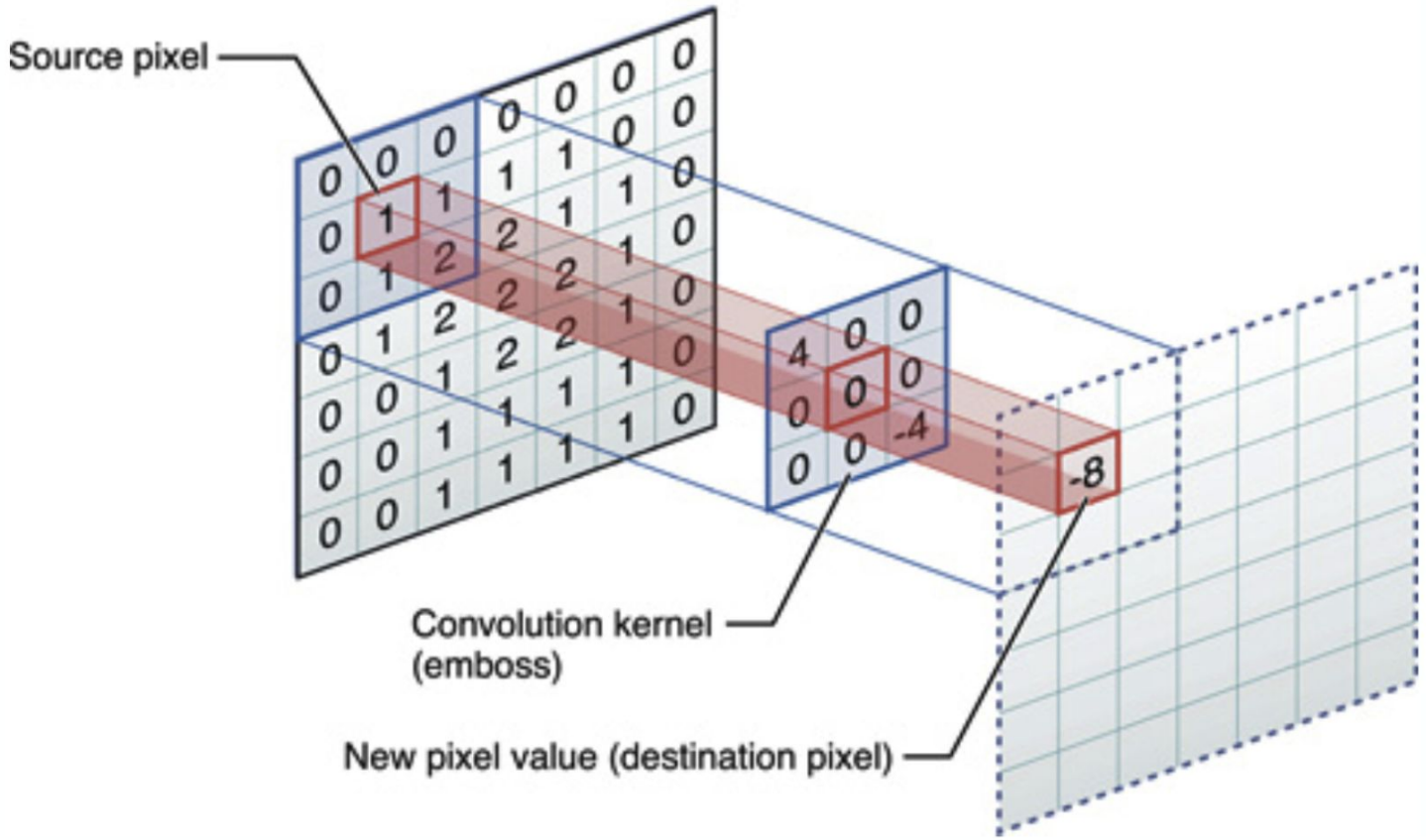
Kernel weights = **parameters**

$$\theta_{i,(\delta x,\delta y,j)}, \quad i = 1,\dots,k, \quad j = 1,2,3,$$
$$\delta x, \delta y \in \{\dots,-1,0,1,\dots\}$$

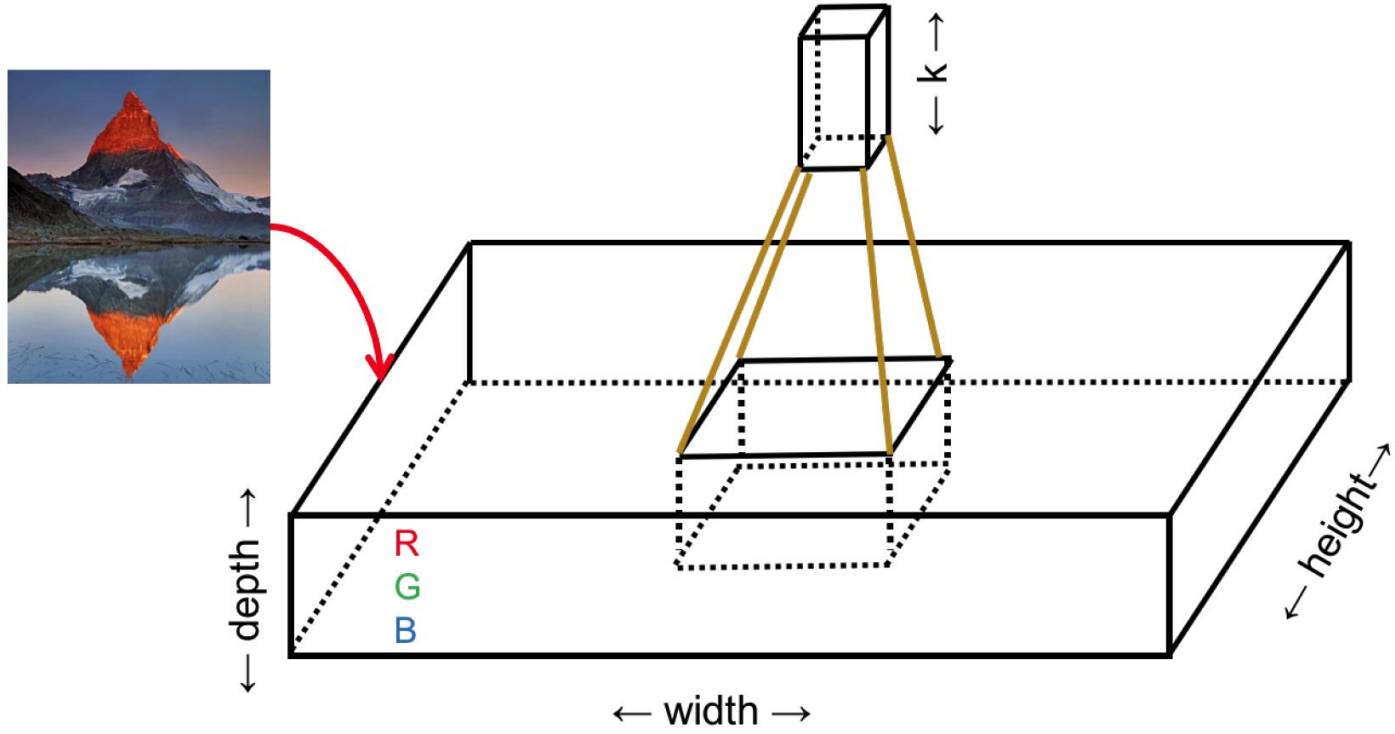
**Feature map** = feature function applied to shifted signal

$= k$ -vector associated with every grid point $(x, y)$

# CNN: Single Image Channel



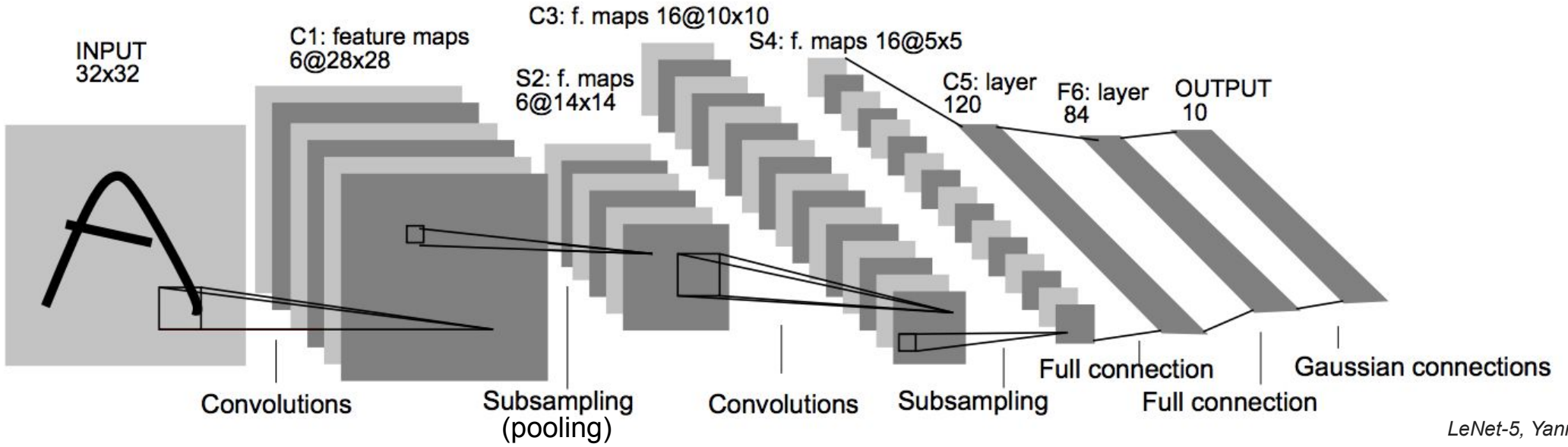
# Convolution details




ReLU CNN layer map

$$\underbrace{z_{i,(u,v)}}_{\text{3-tensor}} = \left( \sum_{j=1}^{n} \sum_{\delta u, \delta v} \underbrace{\theta_{i,j,(\delta u,\delta v)}}_{\text{4-tensor}} \underbrace{x_{j,(u+\delta u,v+\delta v)}}_{\text{3-tensor}} \right)_{+}$$

# Convolutional neural networks




*LeNet-5, Yann LeCun et al*

## Three main types of layers

- Convolutional layers (here, 1st convolutional layer has 6 filters)
- Pooling layers, also called subsampling layers
- Fully-connected layers

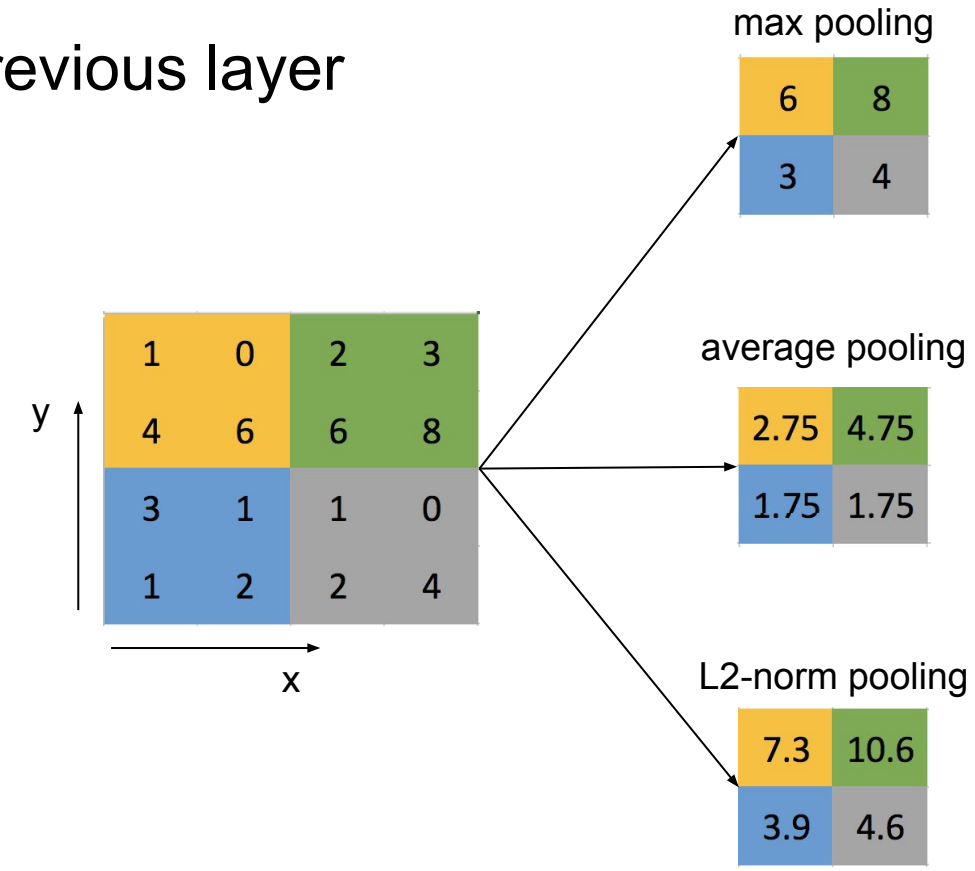
# Convolutional Neural Networks

## Pooling layers

- downsample the representation size from previous layer
- reduce the number of parameters
- control overfitting

## Types of pooling

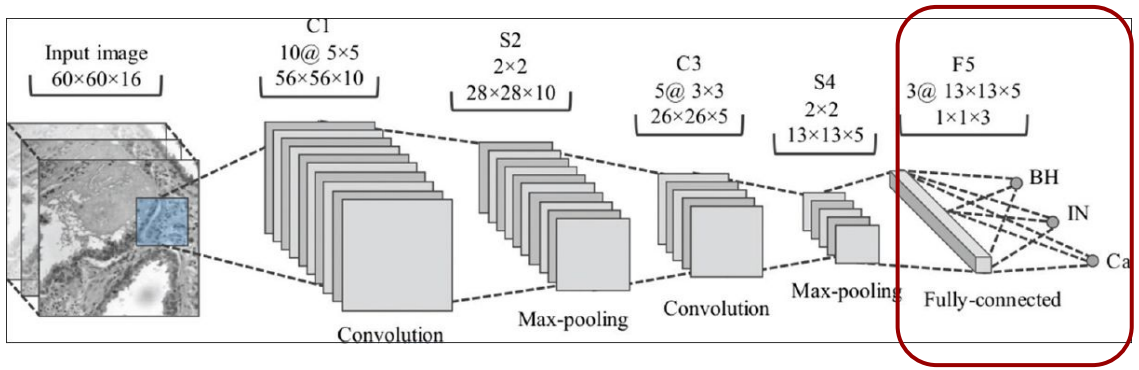
- max pooling (most commonly used)
- average pooling
- L2-norm pooling



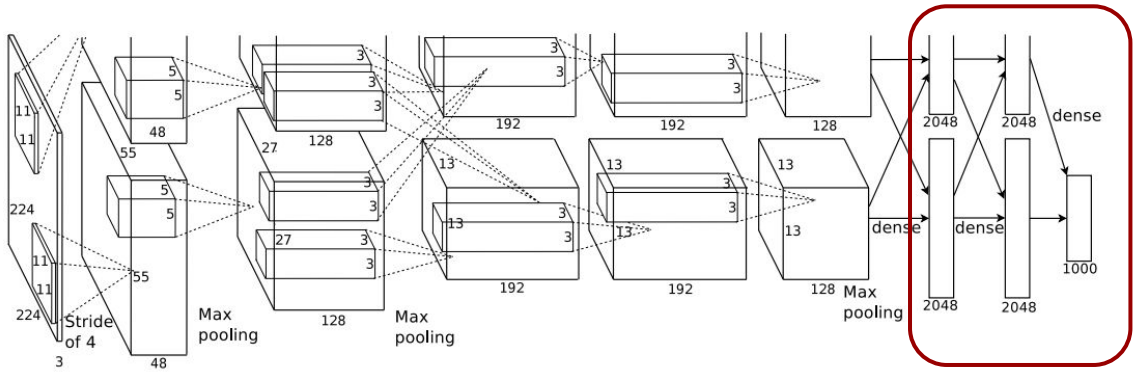
# Convolutional neural networks

## Fully-connected layers

- have full connections to all outputs from the previous layer
- are usually used as the last layer(s) of CNNs




*Haj-Hassan H et al*

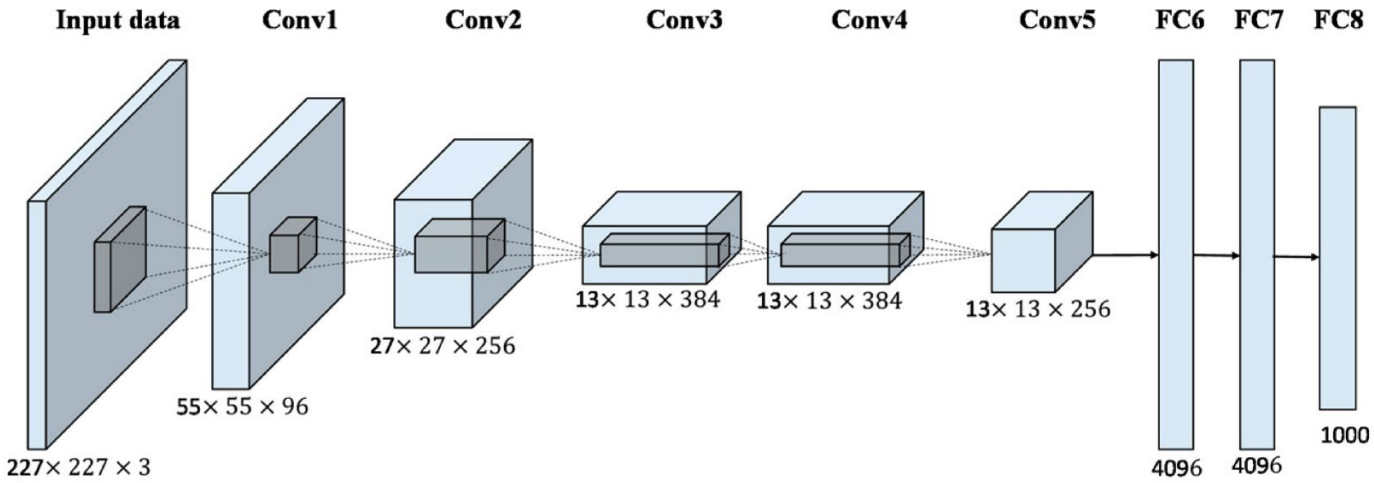



*Krizhevsky A. et al*

# CNN Vision Architecture

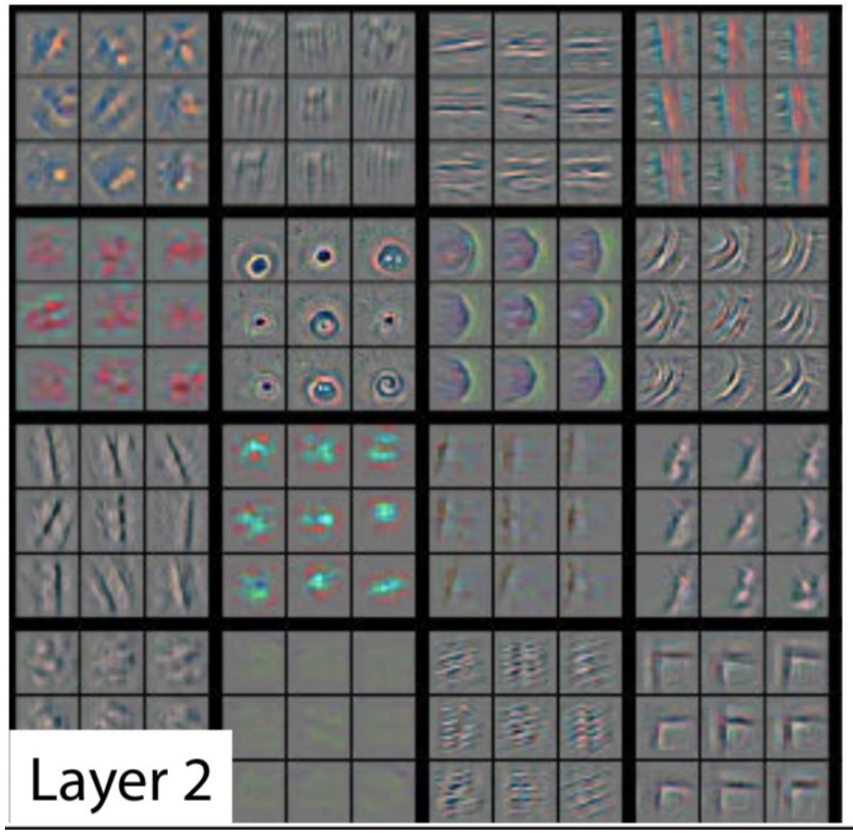
Typical CNN architecture for computer vision pyramidal structure. Depth, lower resolution, many filters and fully connected at the end.

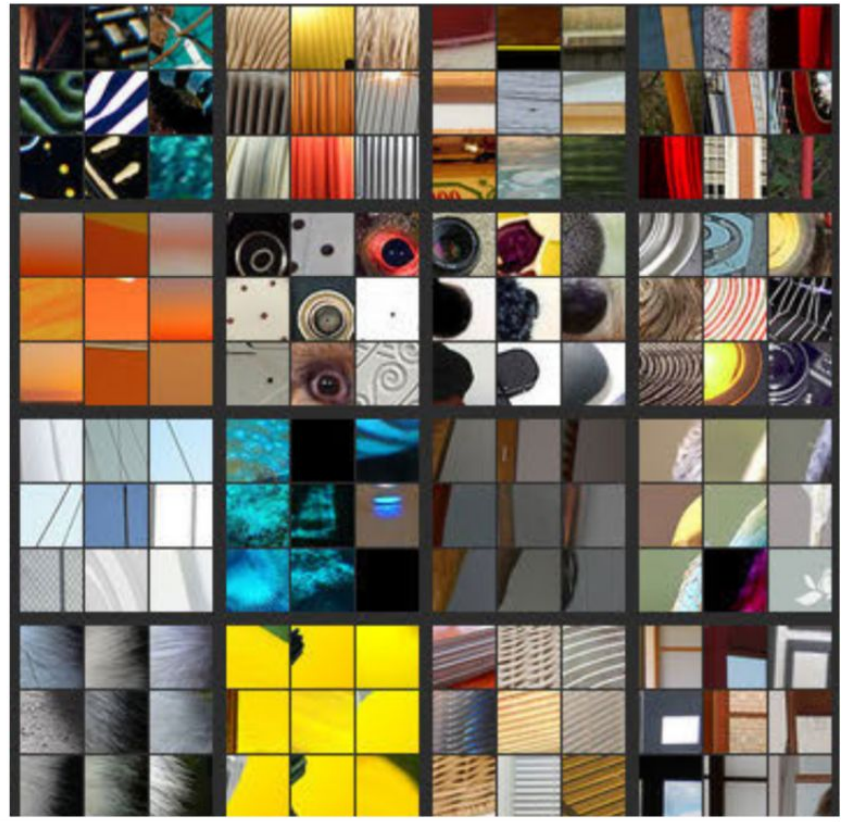
AlexNet (2012)



# Visual Feature Hierarchy

Layer 1



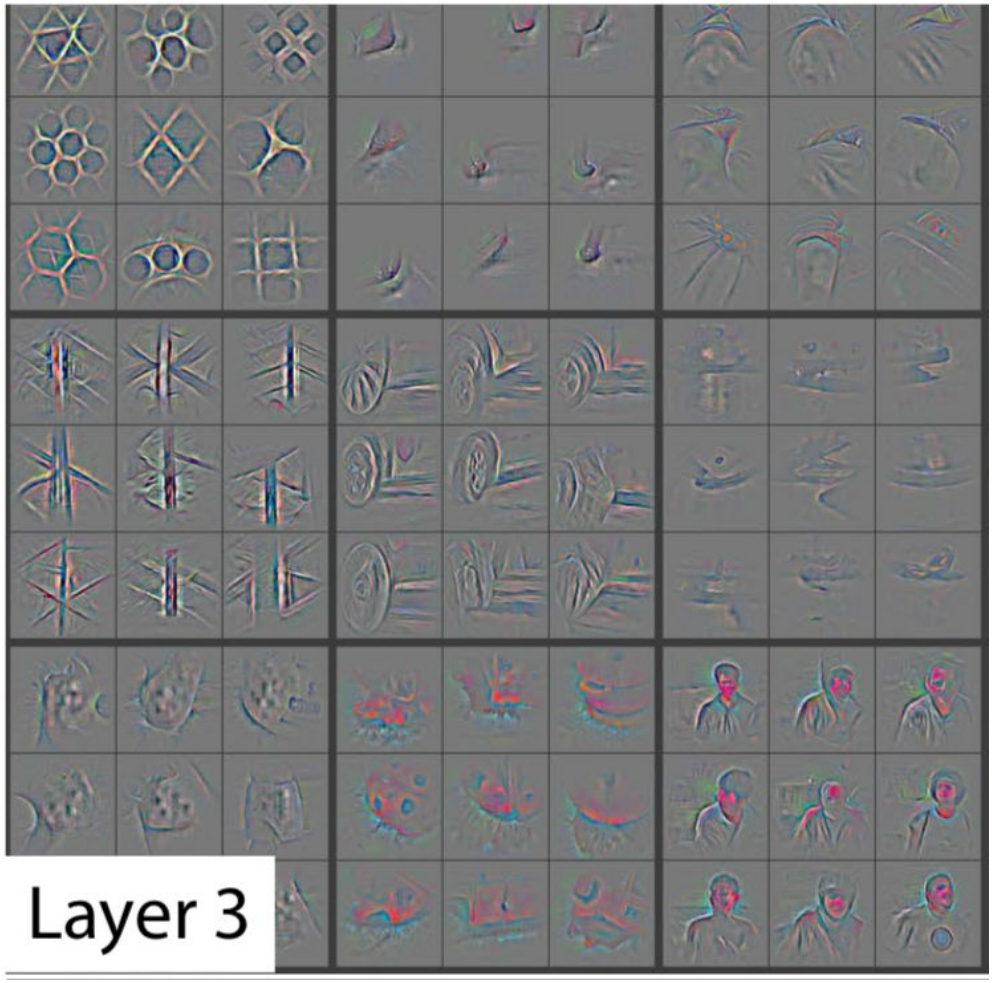



*Zeiler MD et al*

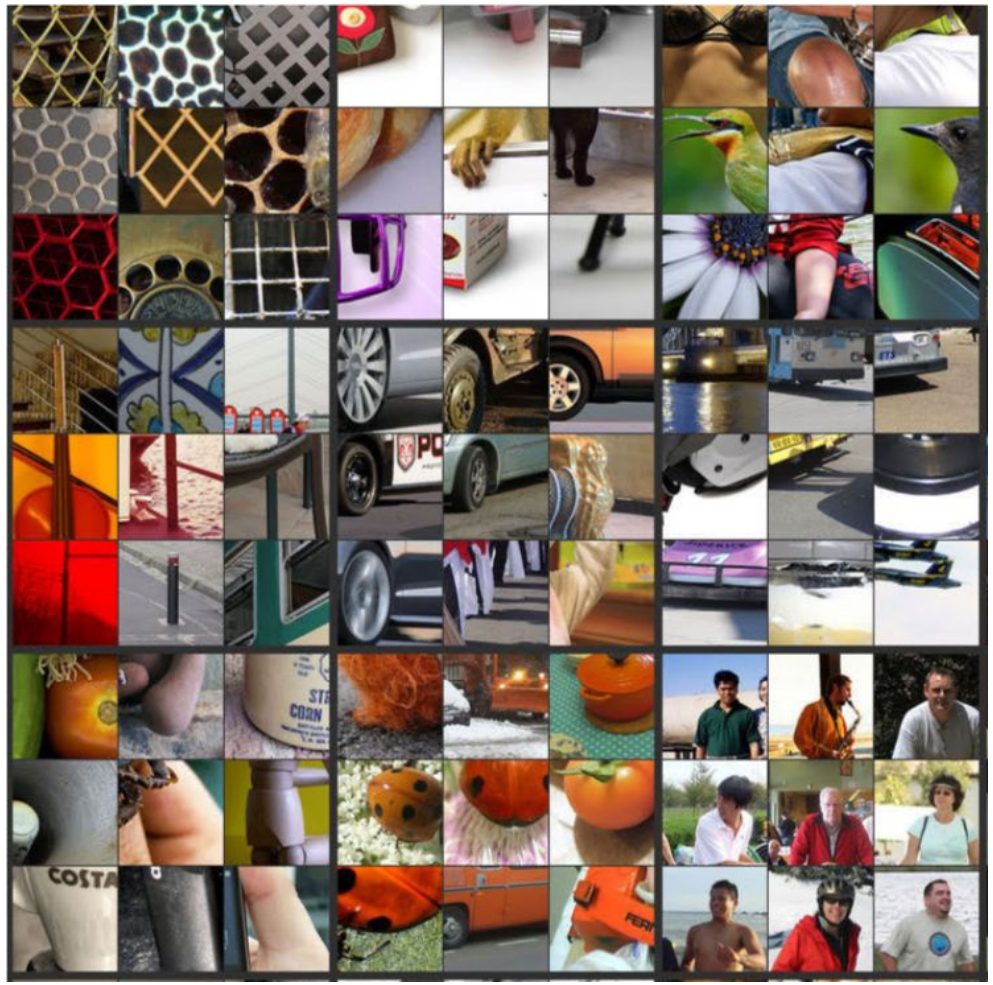
Convolutional Layer 2

original image

# Visual Feature Hierarchy




Convolutional Layer 3



original image

*Zeiler MD et al*

# Advantages of convolutional neural networks

- **Parameter sharing in the convolutional layers**
  - reduce the amount of parameter and computation
  - control overfitting
- **Encode the spatial dependencies at different levels**
  - able to extract local features from the lower layers
  - and more abstract and global features on top of the local ones
- **Excellent performance on image classification tasks**

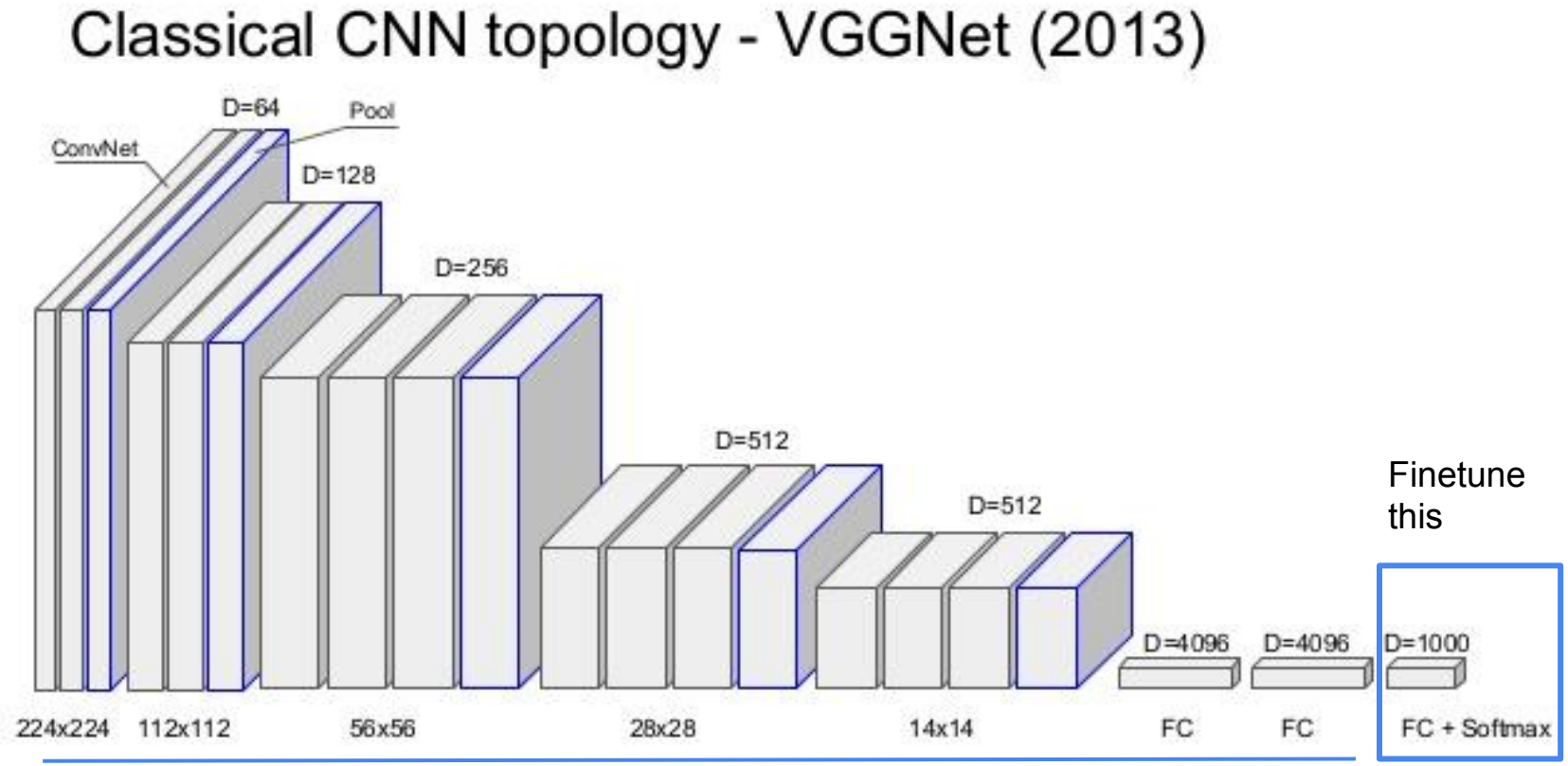
# How to train convolutional neural networks?

- **Parameters (excluding hyperparameters & architecture choices)**
  - filters in the convolutional layers
  - weights in the fully-connected layers
  - (The pooling layers are non-parametric)
- **Input**
  - as much data as possible
  - data augmentation: translation, rotation, scaling and random crop
- **Depth**
  - the more layers, the deeper the model, the better
- **Challenges**
  - long training time even with much fewer parameters than regular NNs
  - overfitting caused by the large number parameters in the fully-connected layers (a common technique used to control overfitting: dropout)
  - GPUs make convolutional models feasible

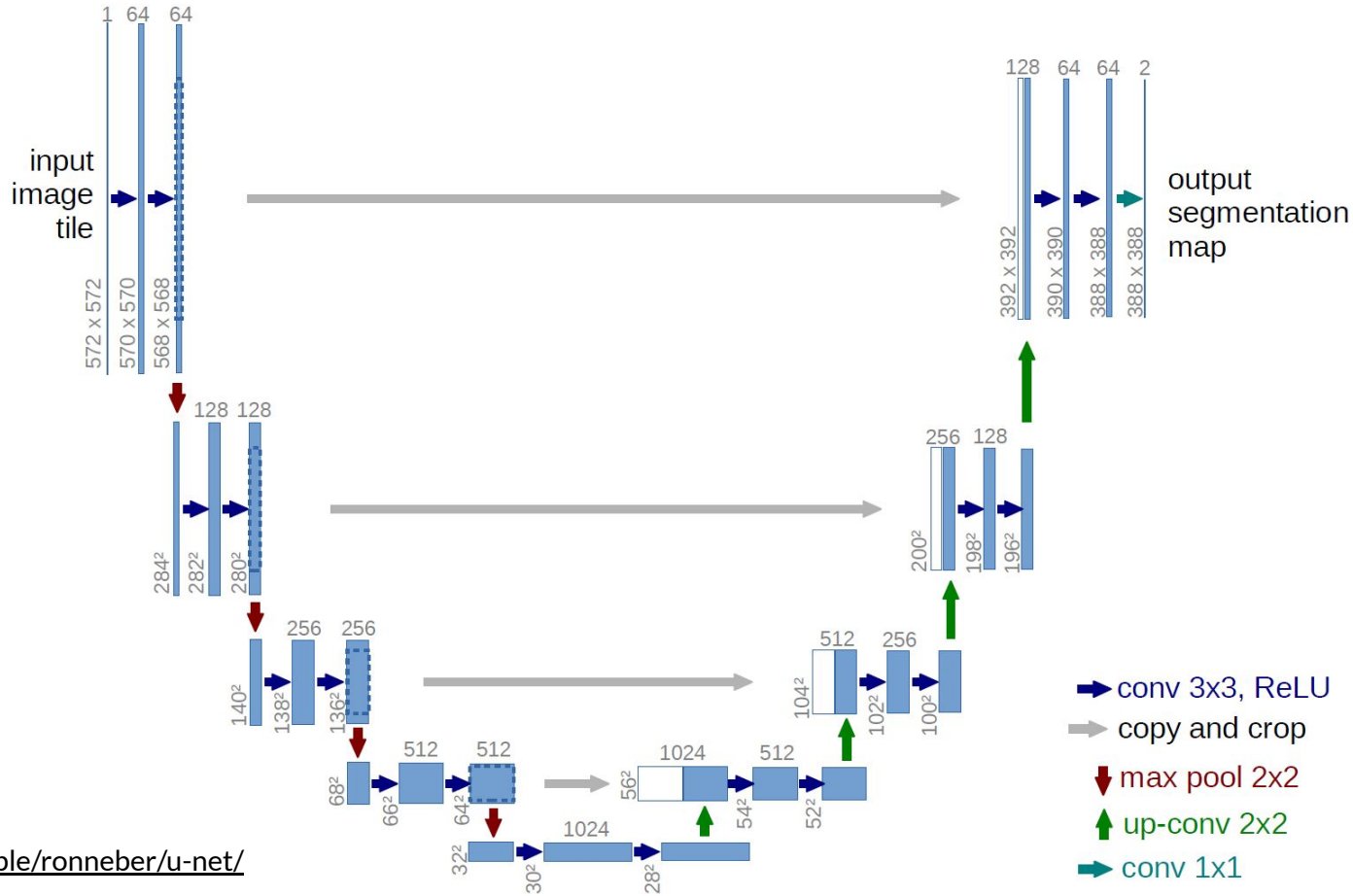
# Common pre-trained networks

- LeNet
- VGGNet
- ResNet
- YOLO

**Typical approach:**
Pretraining on very large datasets, then fine-tuning on application-specific datasets



# Image segmentation with U-nets

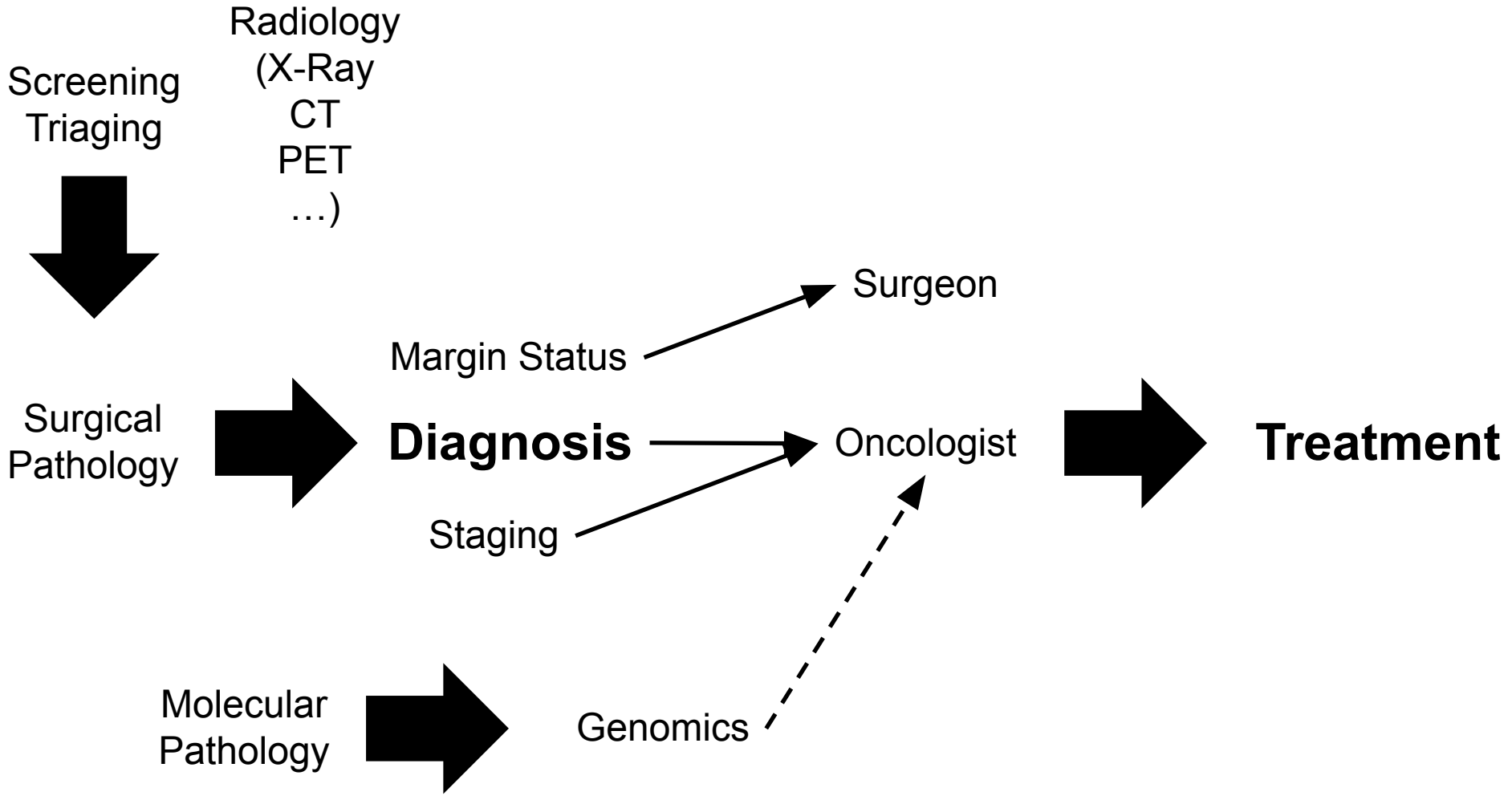



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

# Take Home Messages

- Superpixels can reduce resolution without much loss of key image elements
- Image segmentation is very useful, but non-trivial to get right
  - Markov Random Fields
- Traditional neural networks often don’t work well on images due to overfitting
- Image filters are very useful & powerful
- Convolutional Neural Networks
  - Build on filters -> we can learn them
  - deal with large-scale image inputs
- Parameter-sharing in convolutional layers helps reducing overfitting
- Pooling layers aggregate information to lower resolutions
- Convolutional and pooling layers learn feature extractors that are the input the the final fully connected layers
- Training takes long and can be parallelized (GPUs!)

# Cancer Diagnosis Workflow






Slides on Digital Pathology are courtesy of Gabriele Campanella and Thomas Fuchs.

# Pathology Workflow

**Biopsy**

**Slide Preparation**

Check-in

Grossing

Processing

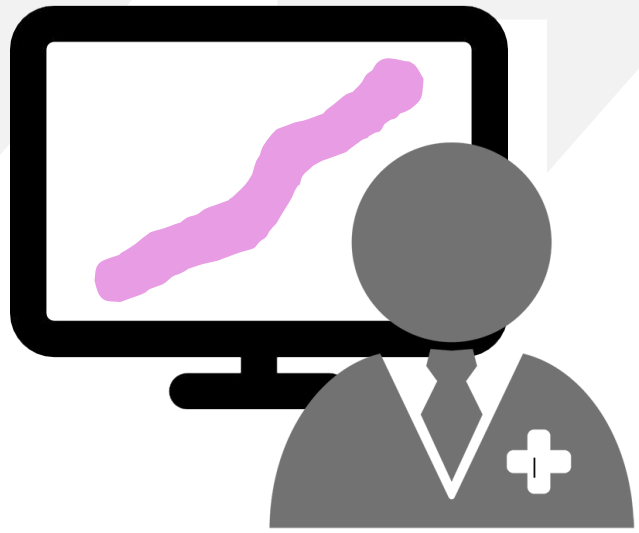
Embedding

Cutting

Staining

**Slide Analysis**

**Diagnostic Reporting**



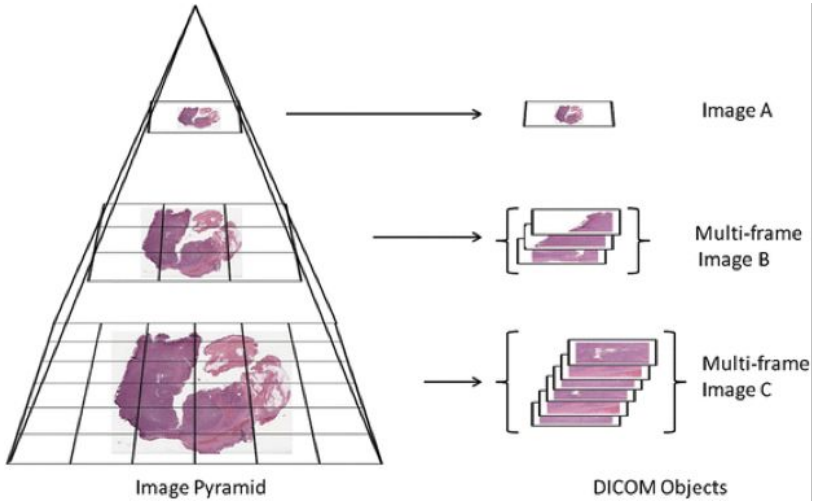
# From Pathology to Digital Pathology

- From glass slides to digital slides
  - Better retrieval and sharing
  - Opinion from other experts
  - Opened doors for machine learning researchers
    - Idea is not to replace pathologists but to make their life easier
    - Automating redundant time consuming tasks
    - Discovery of novel biomarkers

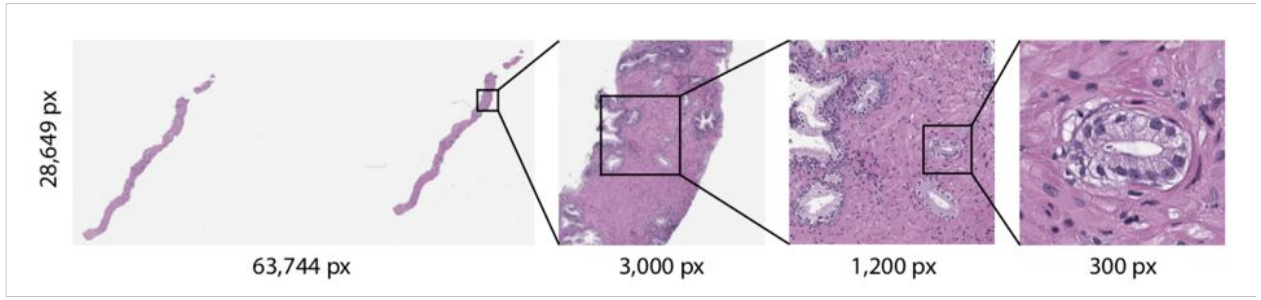

Source: https://www.leicabiosystems.com/

# Pathology to digital pathology

- H&E images: thin tissue sections (3-5 µm)
  - Hematoxylin and eosin staining
    - Purple staining of nucleus
    - Pink staining of stroma and membrane
  - Access to different resolutions -- like in a microscope
    - Image in highest resolution ~100kx100k pixels



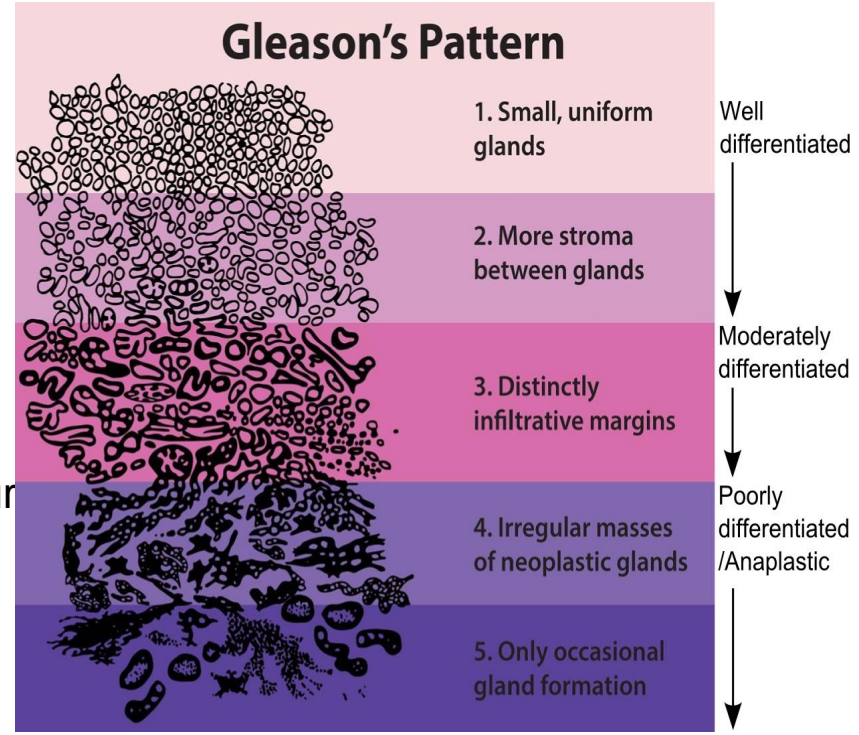

Whole slide image (WSI): Pyramidal image




Bruce A. Beckwith, Digital Pathology, 2016. pp 87-97
James Cuff, https://www.nextplatform.com/, 2018

# “Tasks at hand” for a pathologist

- Bottom up: Prostate adenocarcinoma
  - Cells: cell detection, cell typing
    - Nuclear features predictive of survival, grading of cancer
    - Example: counting mitotic cells
  - Glands: detection, segmentation
    - Shape and structure of glands important morphological featur
    - Example for prostate cancer diagnosis
  - Tissue: grading, tumor detection
    - Eg. gleason score grading of prostate cancer tissue




Source: wikipedia

# Image Data

**Deep Learning for Identifying Metastatic Breast Cancer**
Harvard, MIT (2016)

**400**
Camelyon

**Detecting Cancer Metastases on Gigapixel Pathology Images**
Google (2017)

**400**
Camelyon

**Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning**
NYU (2018)

**1,600**
TCGA

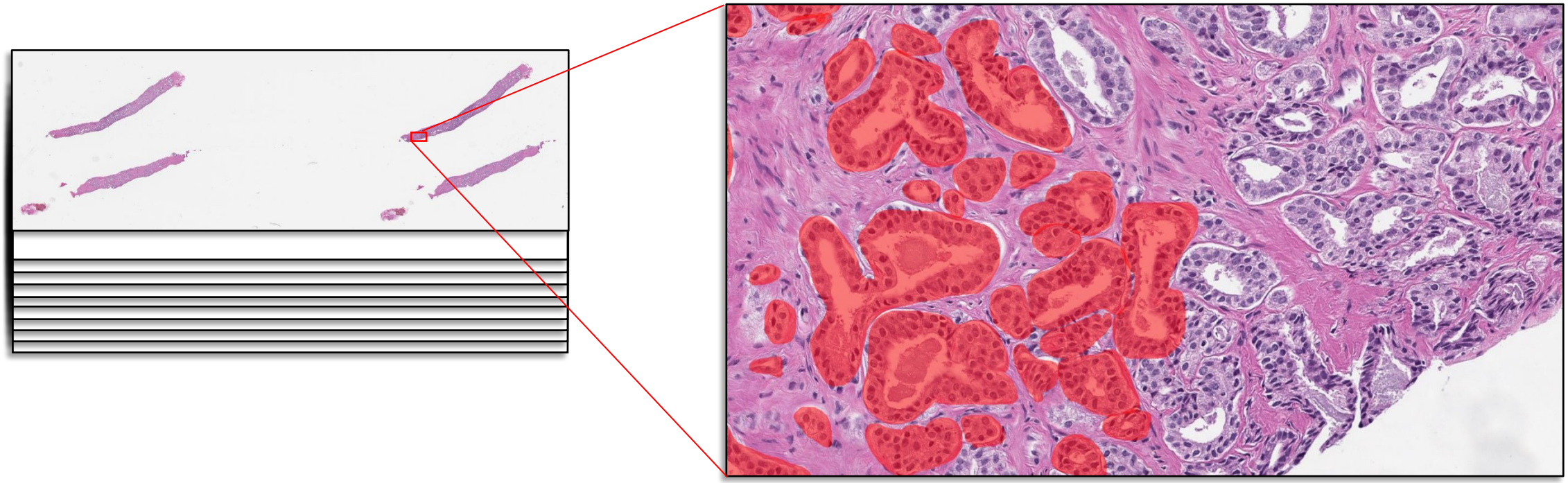
**Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis**
EMBL (2019)

**9,754**
TCGA

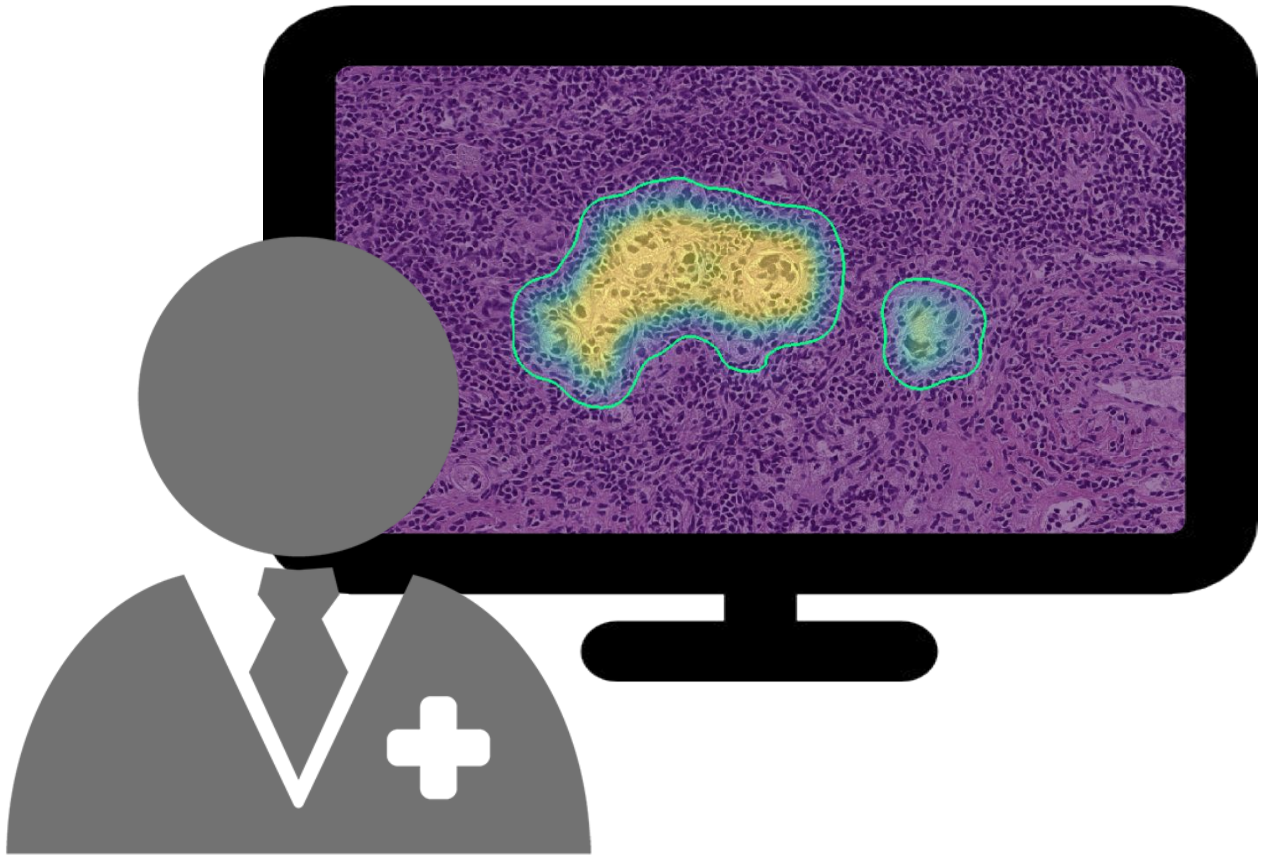
# Large Image Data

**470 Whole-Slide Images** ≈ **ImageNet** 14M images

# Expert Annotations



# Goal: Clinical-grade Decision Support



Given a WSI, return:

- Score representing tumor probability
- Highlight lesion location



Campanella et al., Nature Medicine, 2019
https://www.nature.com/articles/s41591-019-0508-1

# Clinical-grade Decision Support

1. Proposed a method that does not require manual annotations

2. Use datasets much larger than previous studies
3. Learn from the full wealth of biological and technical variability
4. No data curation is necessary

5. Better generalization to real data in pathology practice

6. Defined clinical relevance for computational pathology

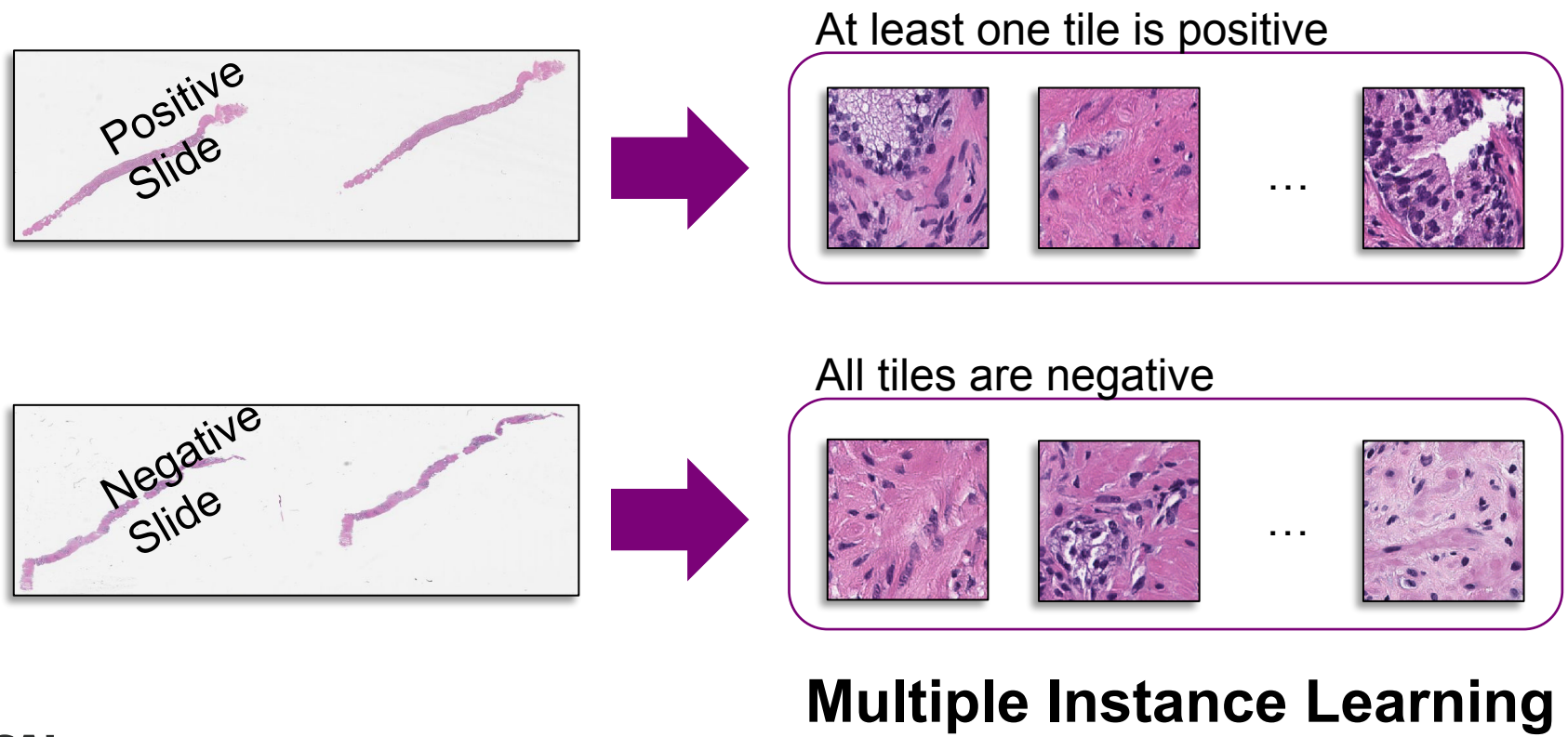
7. Proposed a strategy to integrate this system in the clinical workflow



Campanella et al., Nature Medicine, 2019

# Clinical-grade Decision Support

1. Proposed a method that does not require manual annotations

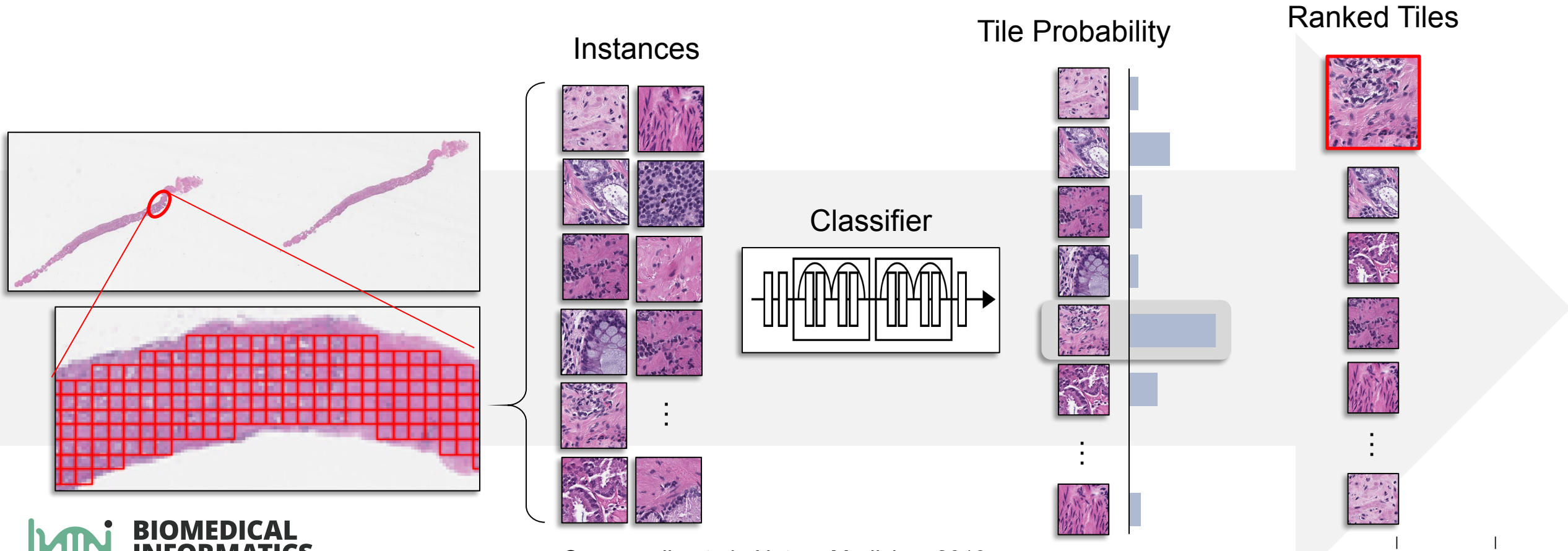



**Multiple Instance Learning**
Dietterich et al. 1997

Campanella et al., Nature Medicine, 2019

# Clinical-grade Decision Support

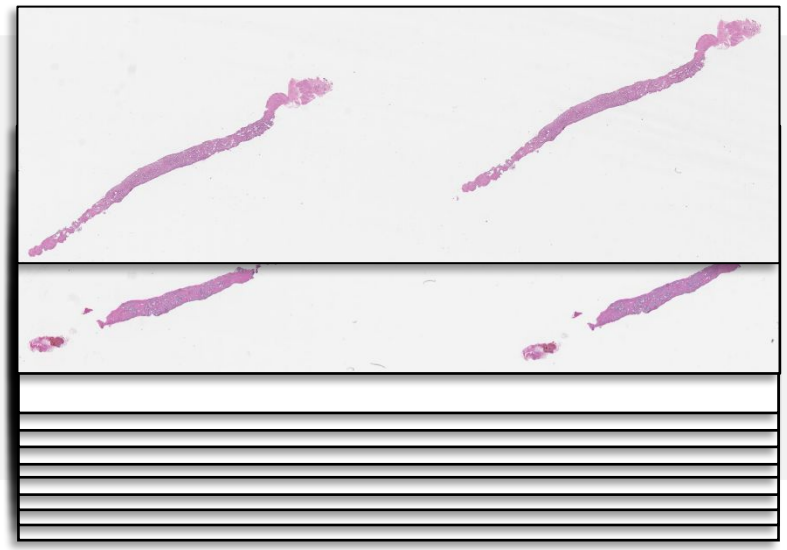
1. Proposed a method that does not require manual annotations




Campanella et al., Nature Medicine, 2019

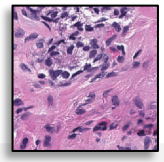
# Clinical-grade Decision Support

1. Proposed a method that does not require manual annotations

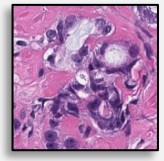


Top-1 Tiles

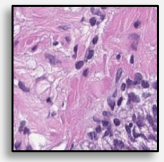
Slide Targets



1

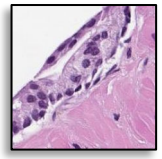


1



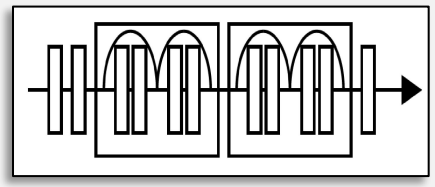
0

⋮ ⋮



0

Model Optimization

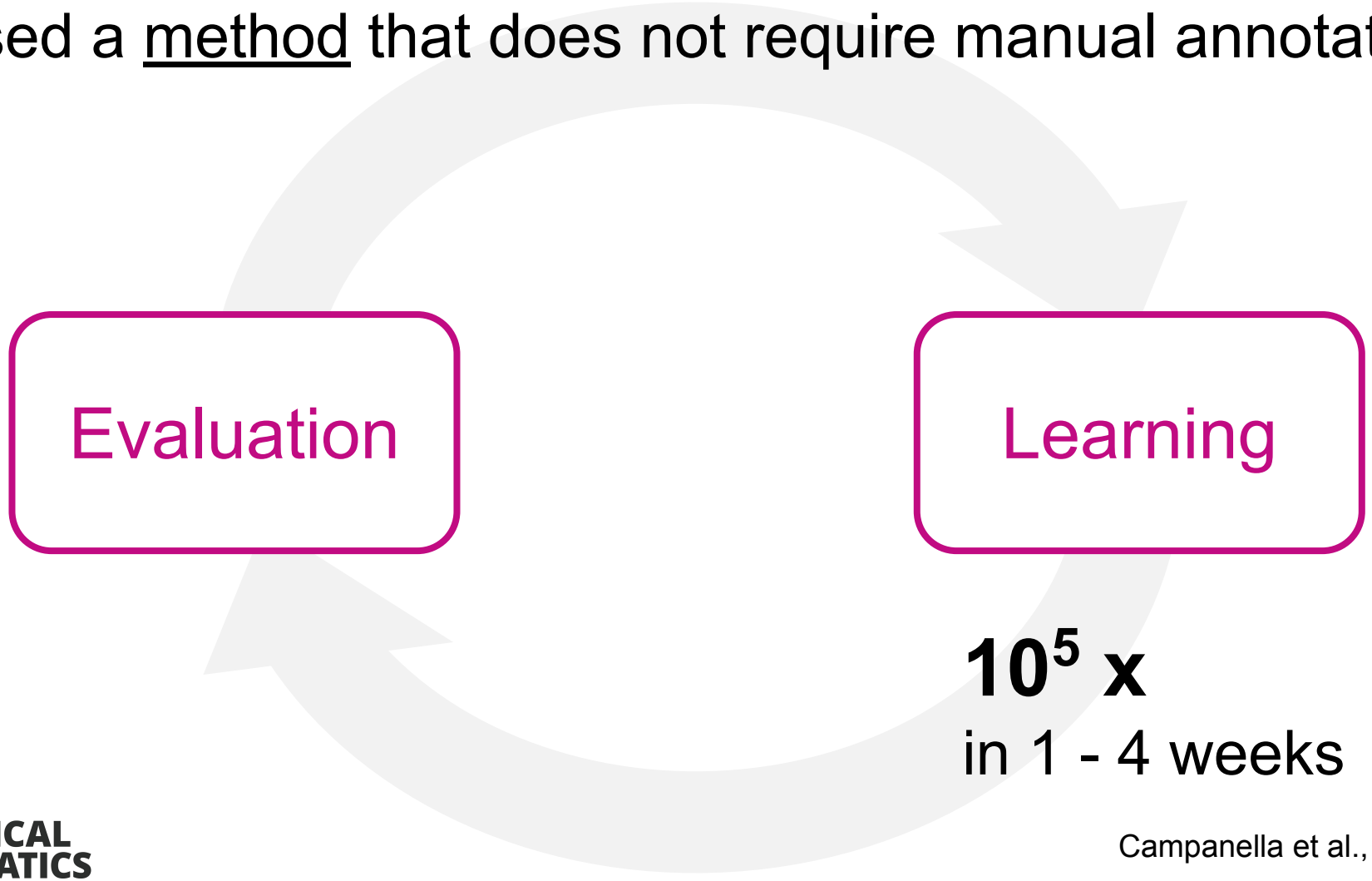




Campanella et al., Nature Medicine, 2019

# Clinical-grade Decision Support

1. Proposed a method that does not require manual annotations

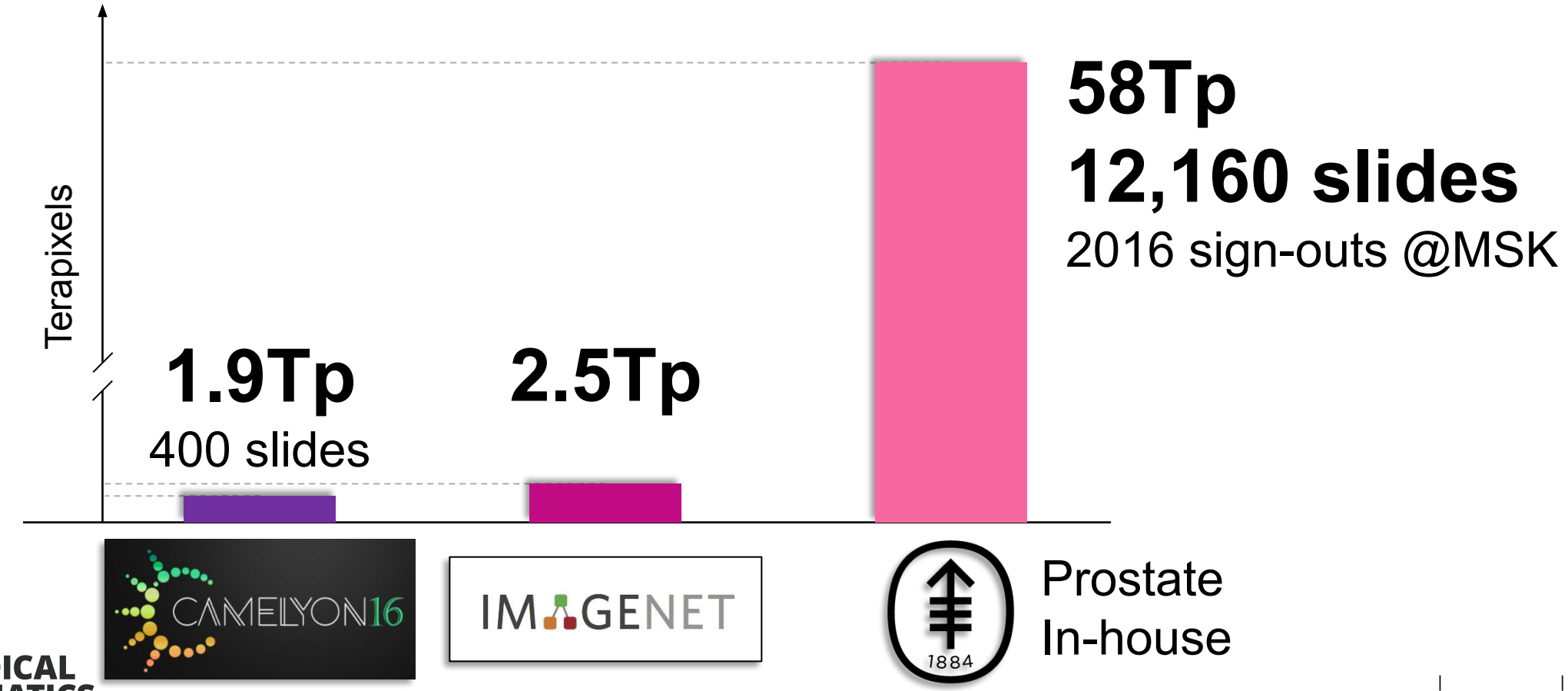





Campanella et al., Nature Medicine, 2019

# Clinical-grade Decision Support

2. Use datasets much larger than previous studies

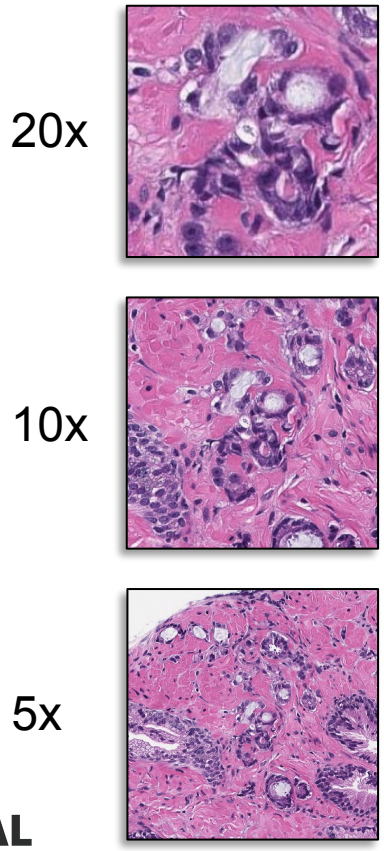


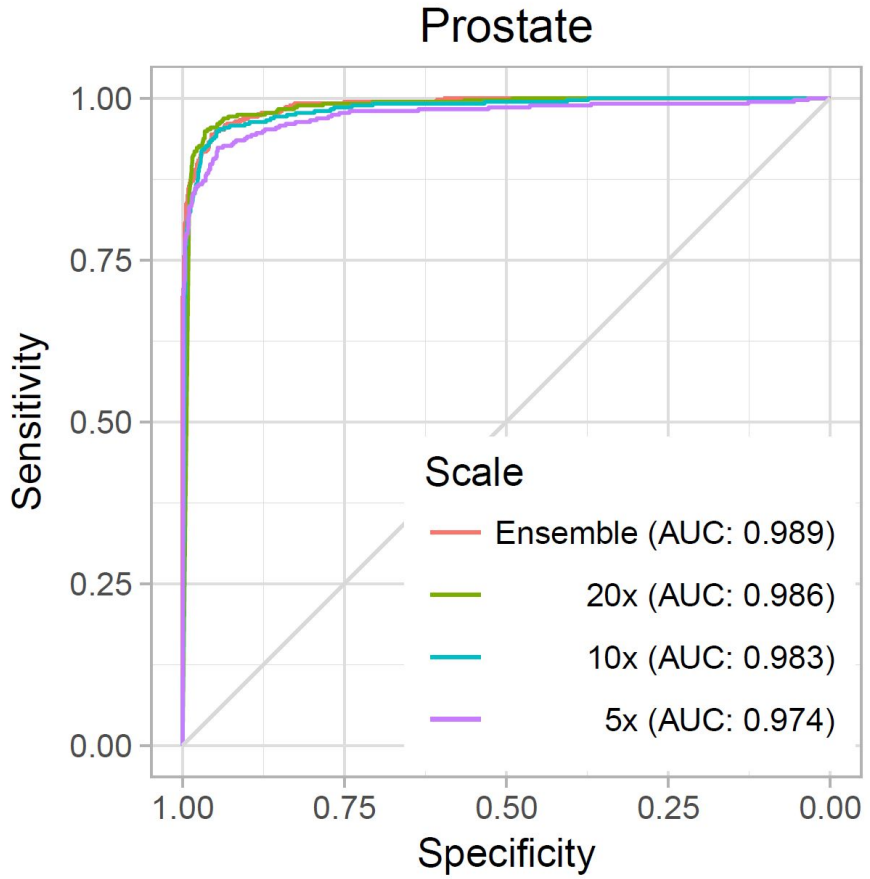



Campanella et al., Nature Medicine, 2019

# Clinical-grade Decision Support

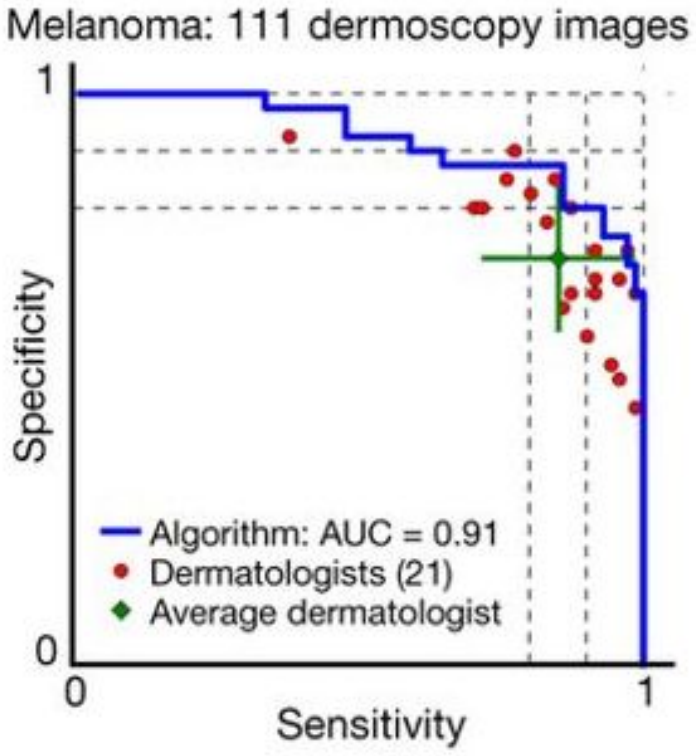
5. Better generalization to real data in pathology practice



# Clinical-grade Decision Support

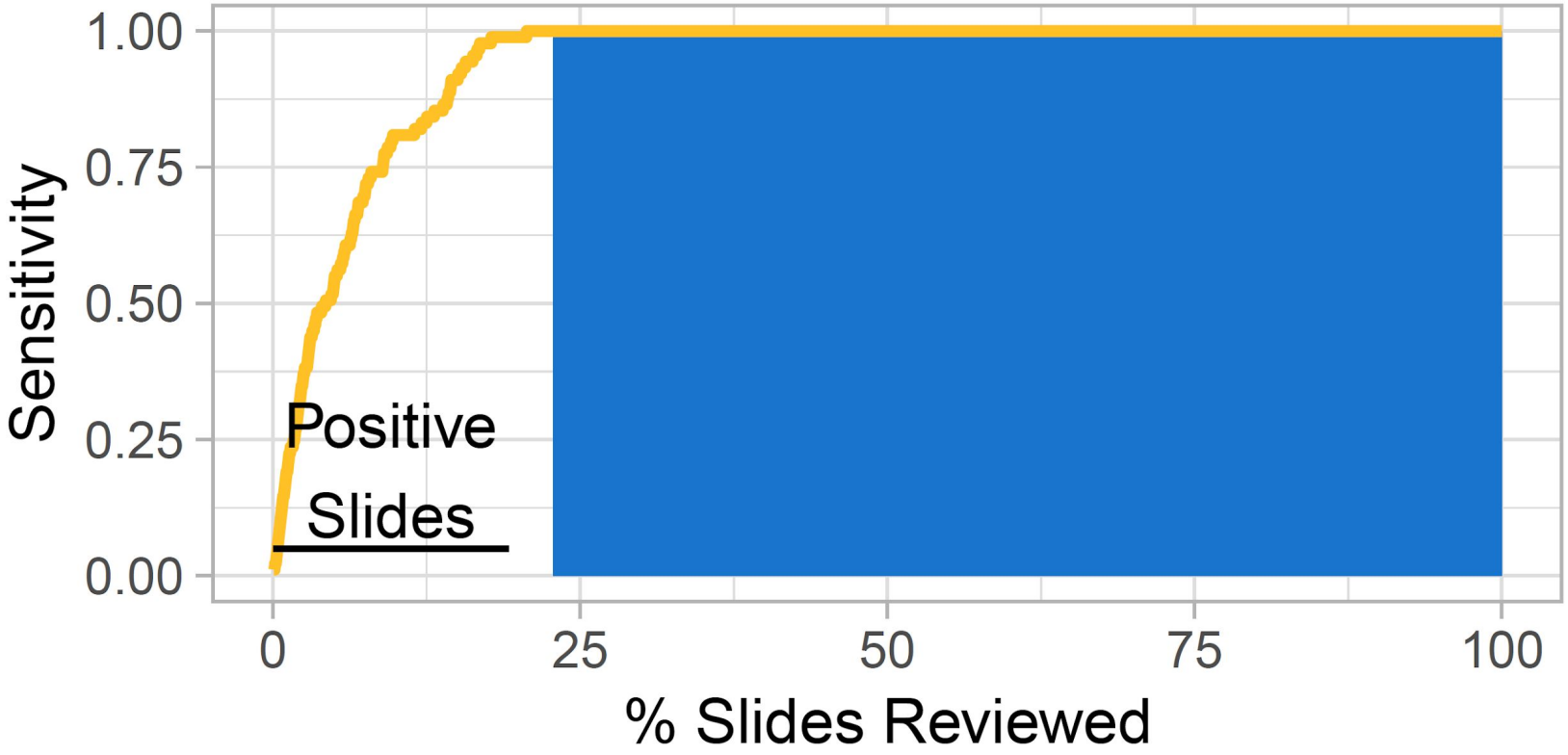
6. Defined clinical relevance for computational pathology




Esteva et al. 2017

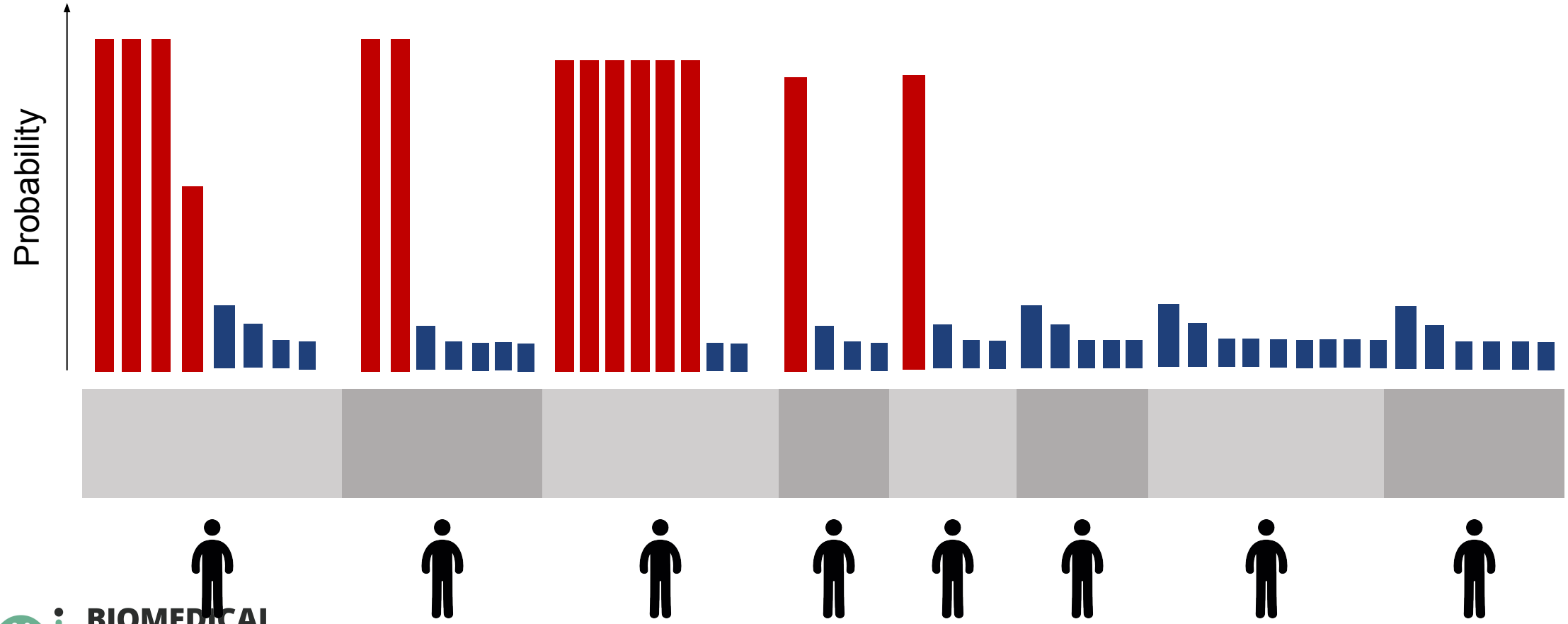
# Clinical-grade Decision Support

6. Defined clinical relevance for computational pathology



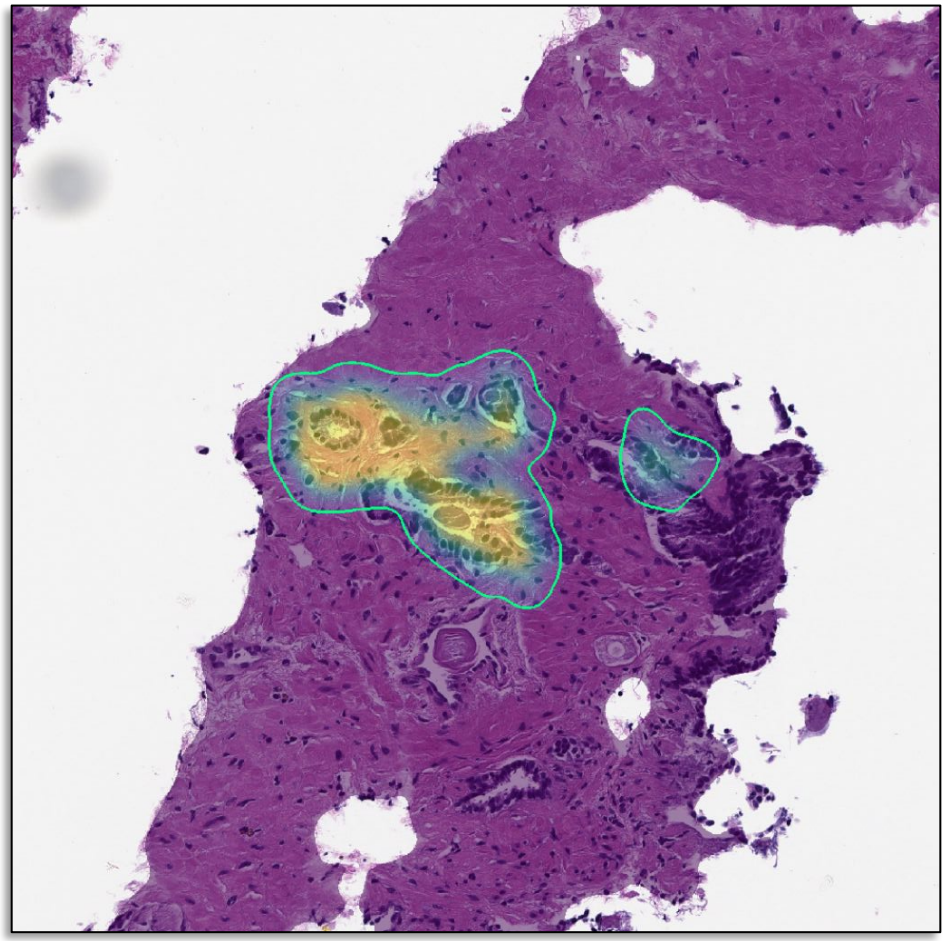
# Clinical-grade Decision Support

7. Proposed a strategy to integrate this system in the clinical workflow



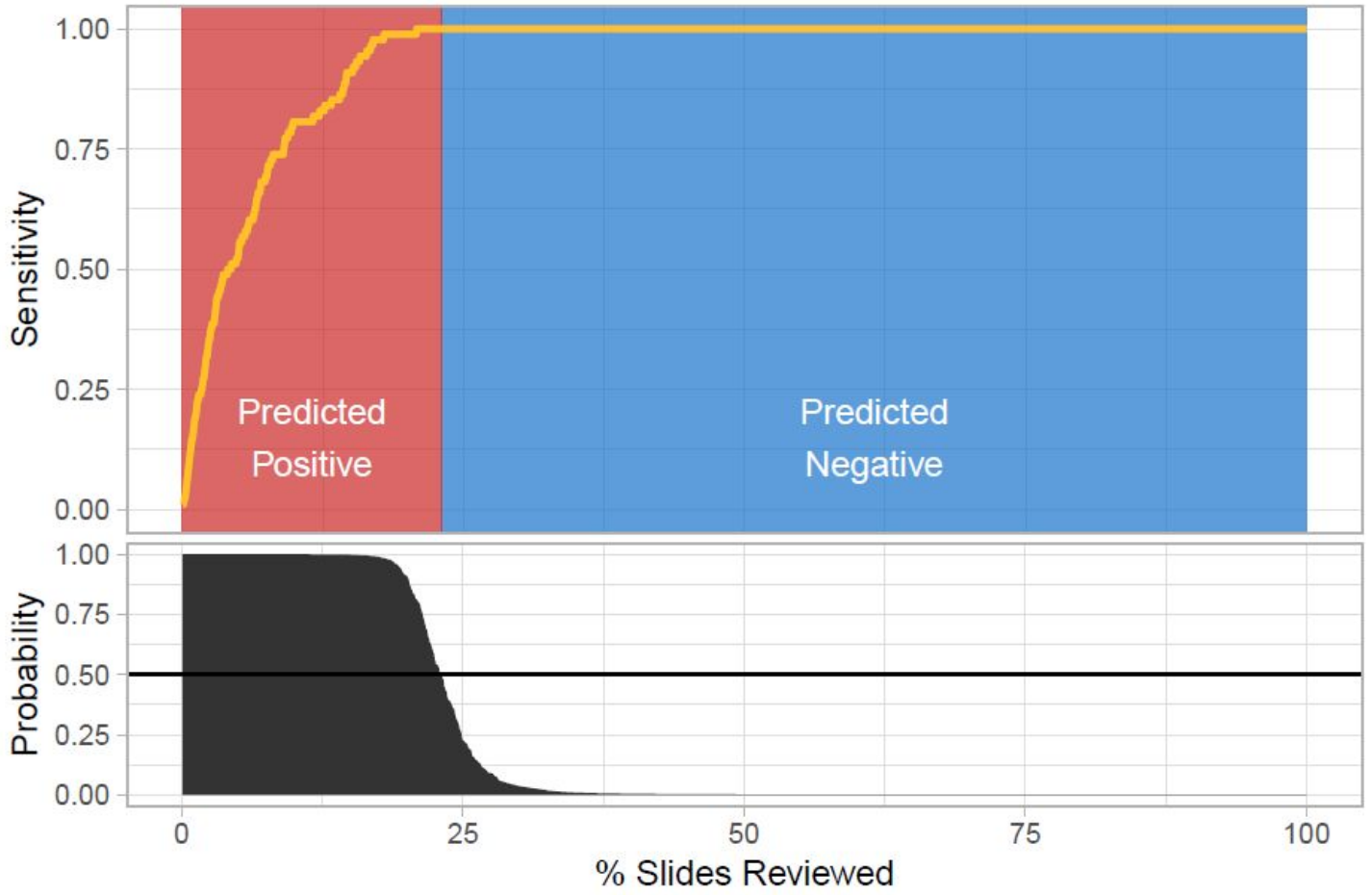
# Clinical-grade Decision Support

7. Proposed a strategy to integrate this system in the clinical workflow



# Clinical-grade Decision Support

7. Proposed a strategy to integrate this system in the clinical workflow



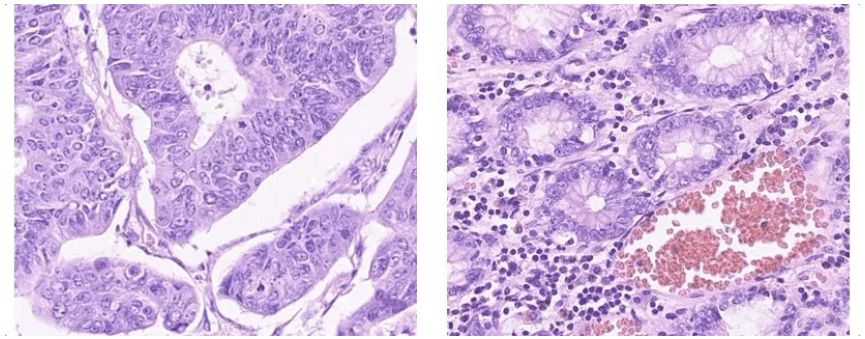

Decrease workload by 75%

# Summary Computational Pathology

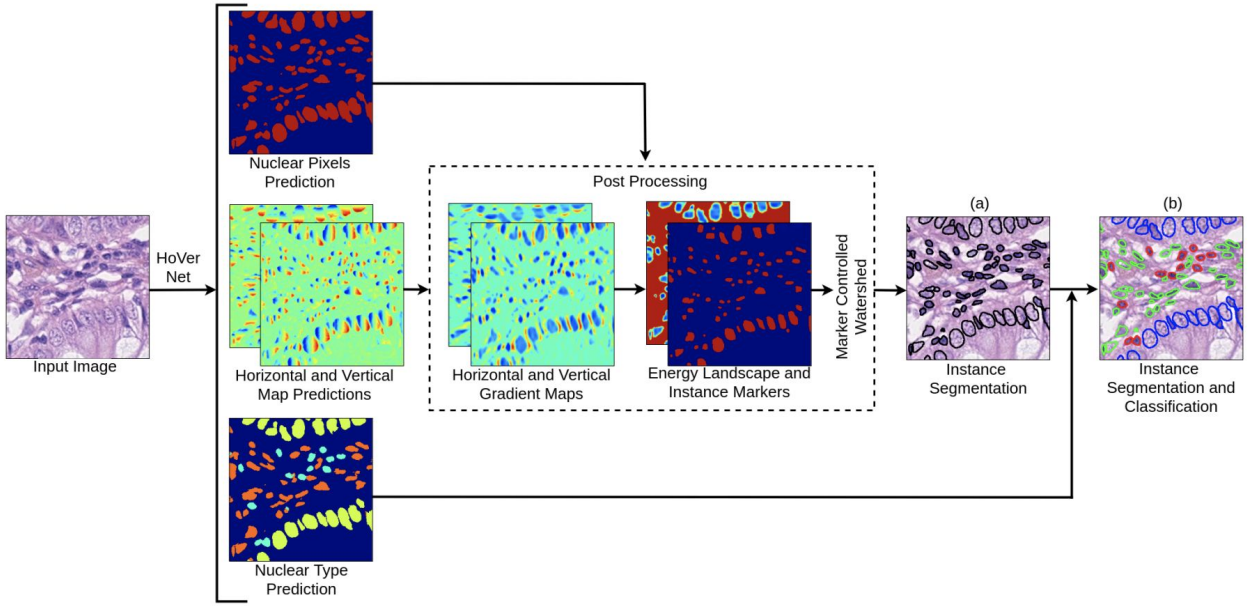
- Computational Pathology is data rich
- Prime example of using deep learning on medical images
- Proposed approach leveraged weak labeling of images
- Takes advantage of vast image archives at a large cancer hospital
- Proposed innovative ways to use methodology in clinical workflow

# Solving “Tasks at hand” using ML

- Cell detection and classification: HoVer-Net:
  - Why: nuclear features predictive of survival, grading of cancer



Blue: epithelial cells
Red: inflammatory cells
Green: spindle-shaped cells
Cyan: miscellaneous cells




Graham, Simon, et al. "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images." *Medical Image Analysis* 58 (2019): 101563.

HoVer-Net pipeline