Fully Convolutional Networks for Semantic Segmentation

Marek Leibl

Center for Machine Perception, Department of Cybernetics, FEL CTU

March 30, 2017

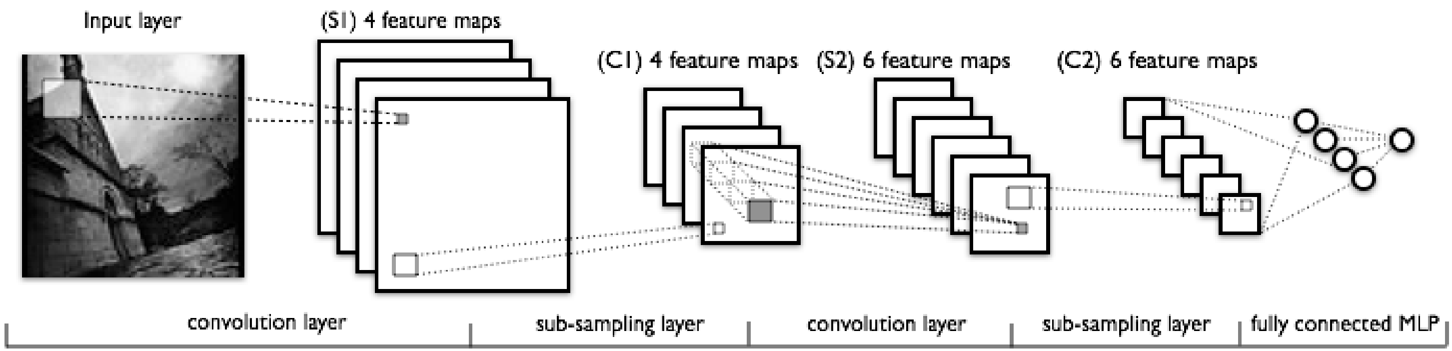
Presentation Outline

- Introduction ... convolutional neural networks
- Semantic Segmentation ... fully convolutional neural networks
- Conditional Random Fields ... and recurrent neural networks
- Applications ... drosophila eggs segmentation

Convolutional Neural Networks

Convolutional Neural Networks - Introduction

- Convolutional Neural Networks (CNNs)
  - layered artificial neural network
  - inspired by organization of the animal visual cortex
  - local connectivity
- Deep Convolutional Neural Networks
  - many convolutional layers often combined with max-pooling and relu



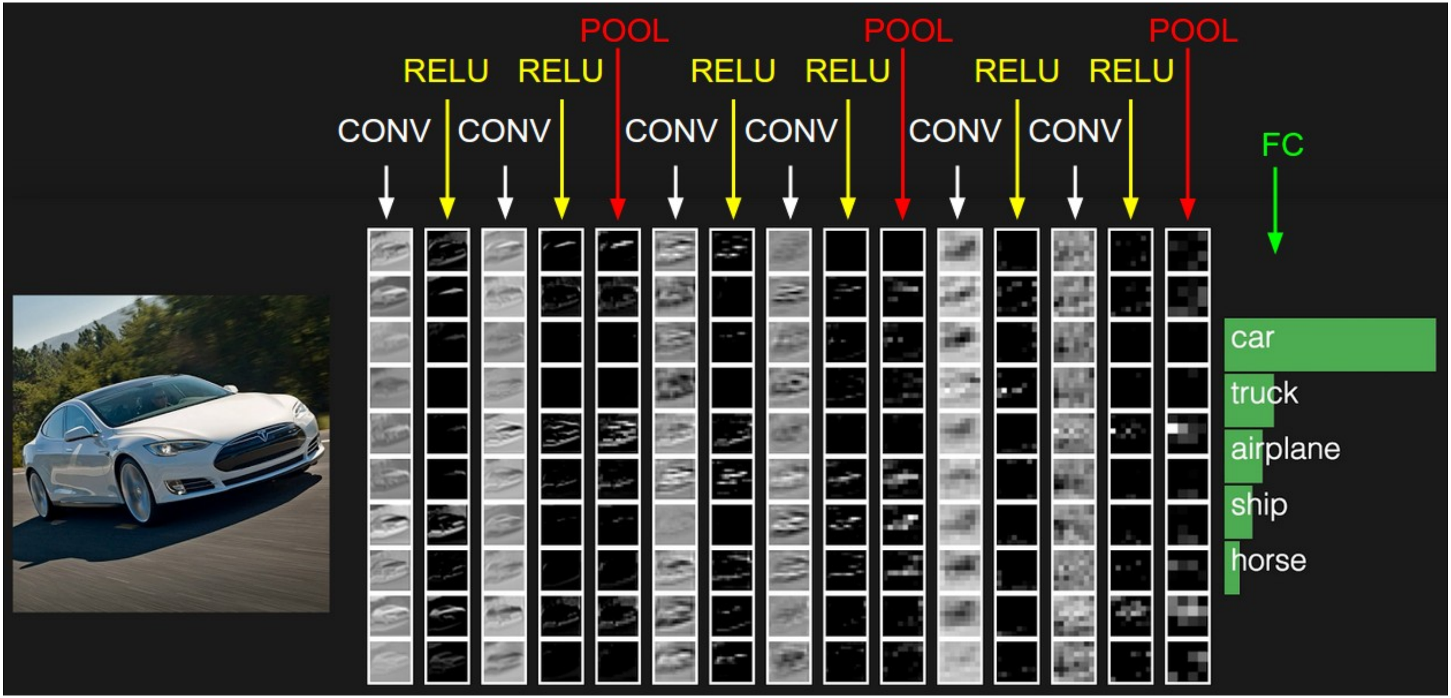
CNNs - Convolution

- Convolution = building block of CNNs

$$* \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

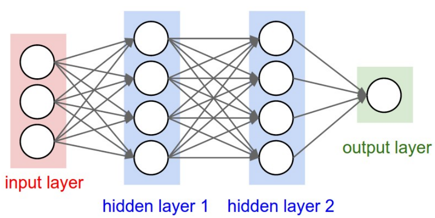
$$G[i,j] = \sum_{u} \sum_{v} H[u,v] \cdot F[i-u, j-v]$$

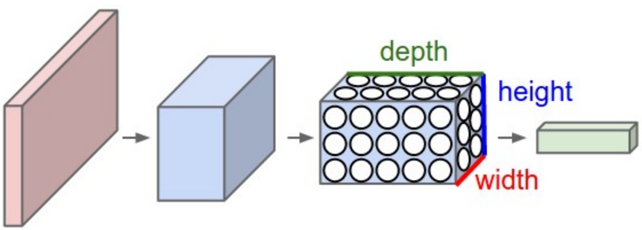
CNNs - Example



CNNs - Structure

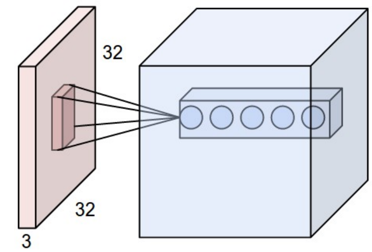
- Neurons in each layer organized into 3D matrixes
  - 2 spatial dimensions
  - 1 dimension for channels (features)
- Local connectivity
  - each neuron relatively small number of input connections ... receptive field
- Shared weights
  - weights are invariant to translations in spatial dimensions
  - reduced search space and better robustness

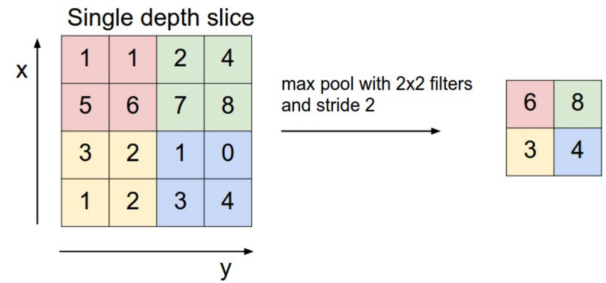


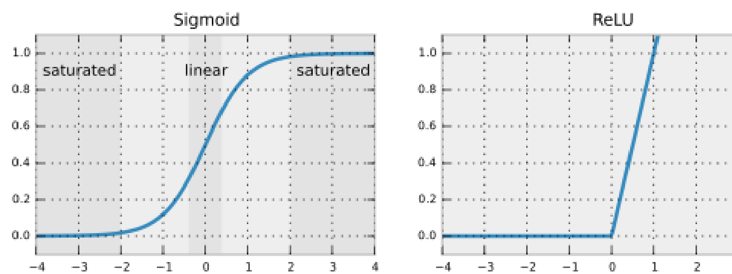


CNNs - Building blocks

- Convolutional layer
  - trainable weights
  - typically followed by Relu
- Relu (rectified linear unit)
  - $f(x) = \max(0, x)$
  - introduces non-linearity to CNN
  - easy to compute, no saturation
- Max-pooling
  - reducing spatial size
- Softmax
  - normalized exponential function
  - $f(x)_j = \frac{e^{x_j}}{\sum_i e_i^x}$

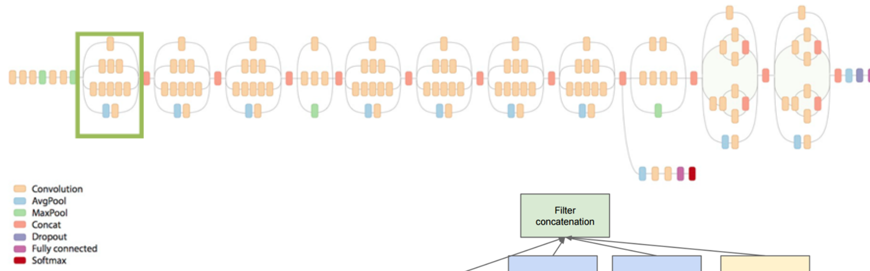




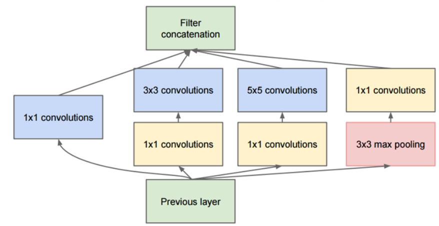


CNNs - Deep CNN architectures for image classification

- GoogLeNet (2015)
  - 9 Inception modules
  - over 100 layers
  - trained on a few GPUs within a week

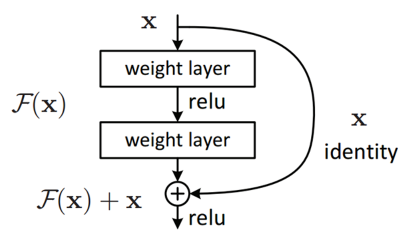


Green box shows parallel region of GoogLeNet

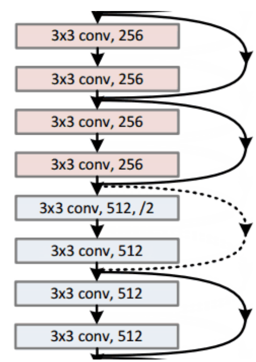


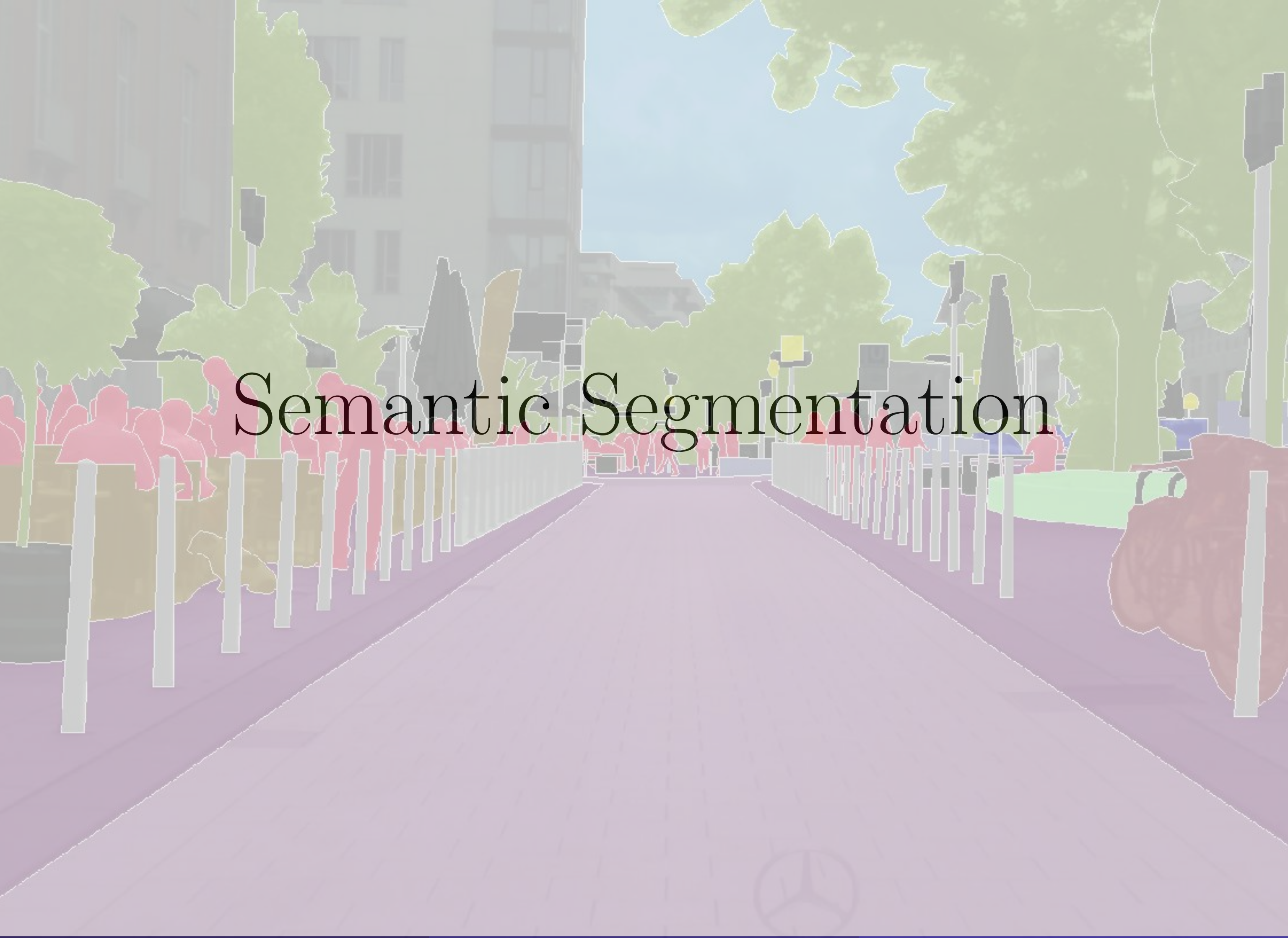
Full Inception module

- Microsoft ResNet (2015)
  - 152 layers
  - residual blocks
    $\rightarrow$ easier to optimize
  - trained on an 8 GPU machine for two to three weeks



A residual block



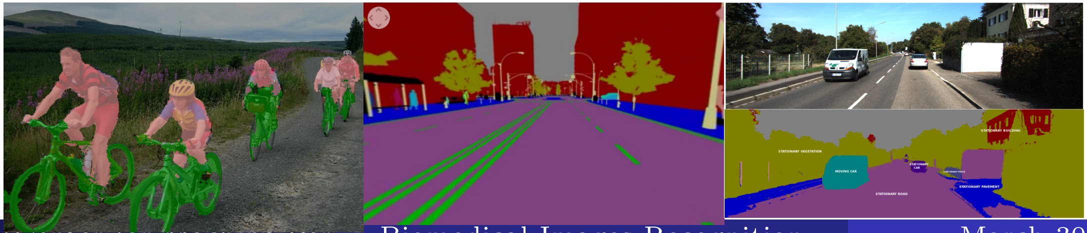


Semantic Segmentation

Semantic Segmentation vs Image Classification

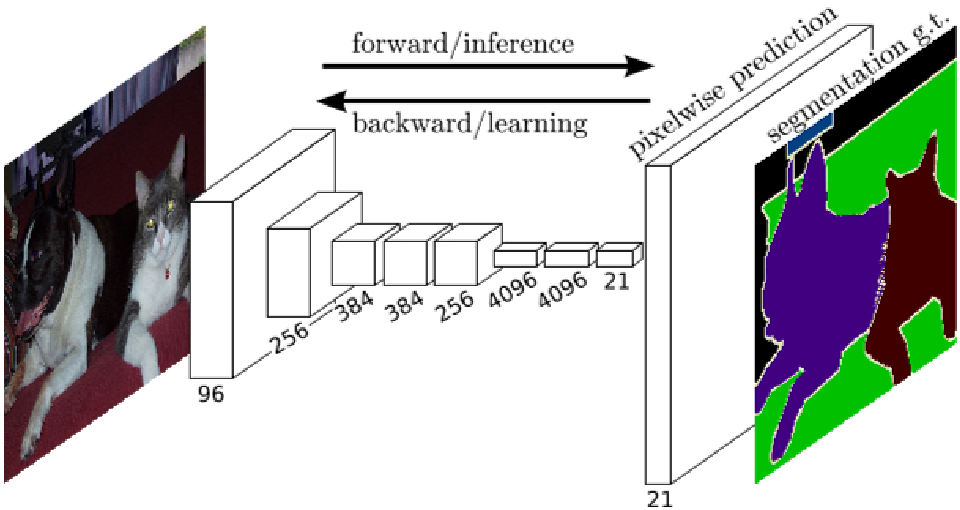
- Image classification: determine class for a given image
  - super human performance on some tasks

- Semantic segmentation: determine class for each pixel
  - harder problem



Fully Convolutional Neural Netwoks

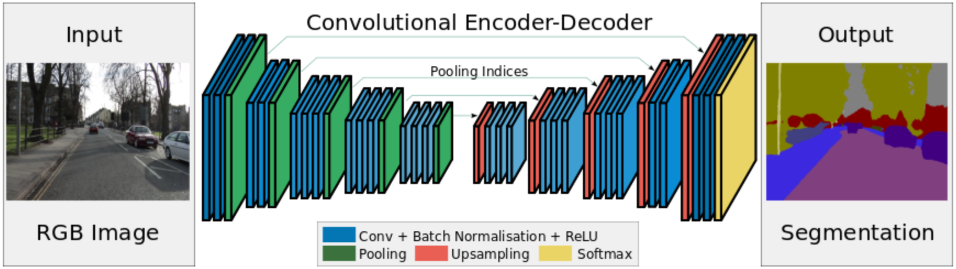
- Fully Convolutional Neural Netwok
  - no fully connected layer
  - output: pixel-wise prediction



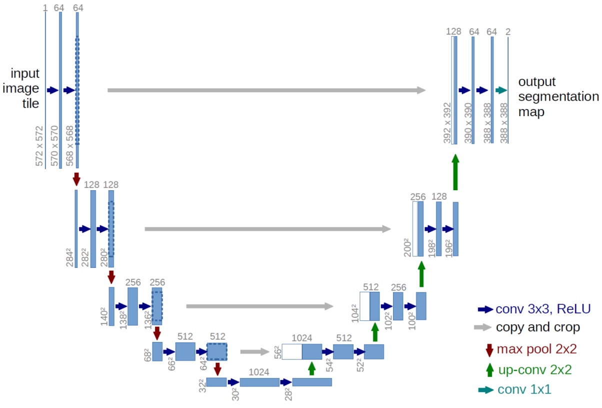
- Training Fully Convolutional NNs
  - needed label for each pixel (ground truth)
  - need to be manually segmented by human
    $\rightarrow$ training set is usually expensive

Fully Convolutional Neural Netwoks: Architectures

- SegNet (2015)
  - deep encoder-decoder architecture
  - University of Cambridge

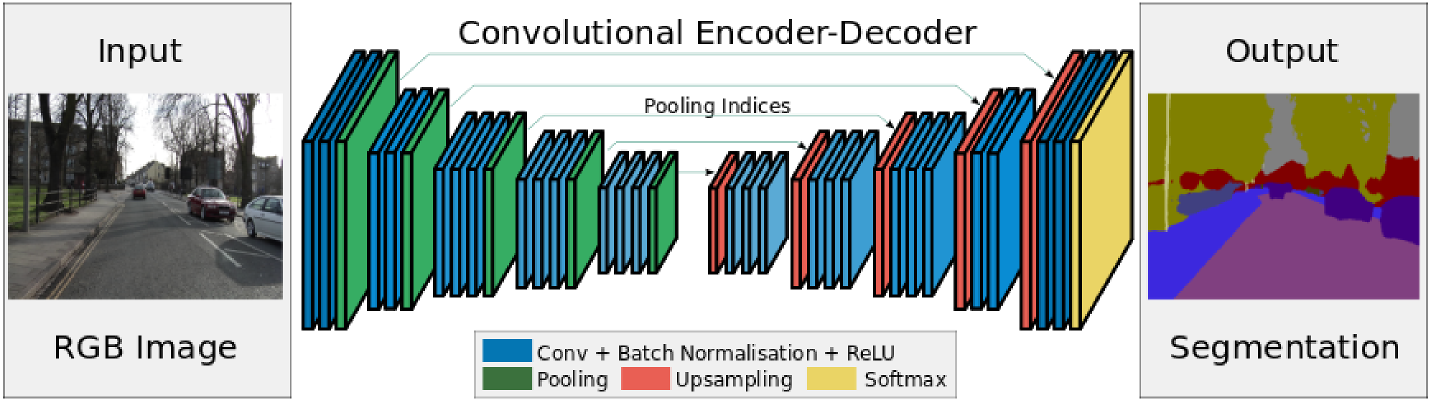


- U-Net (2015)
  - U-shaped architecture
  - biomedical image segmentation



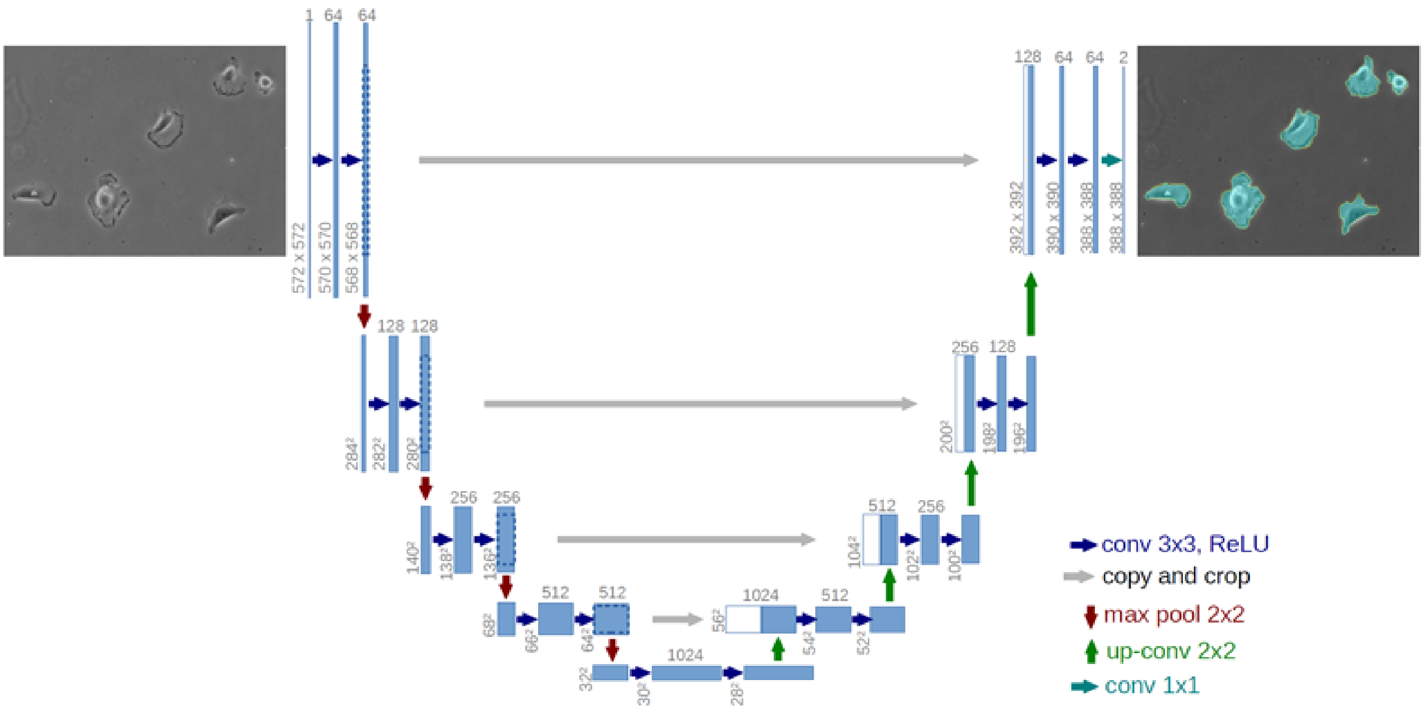
Fully Convolutional Neural Netwoks: SegNet

- SegNet (2015)
  - deep encoder-decoder architecture
  - University of Cambridge
  - http://mi.eng.cam.ac.uk/projects/segnet/

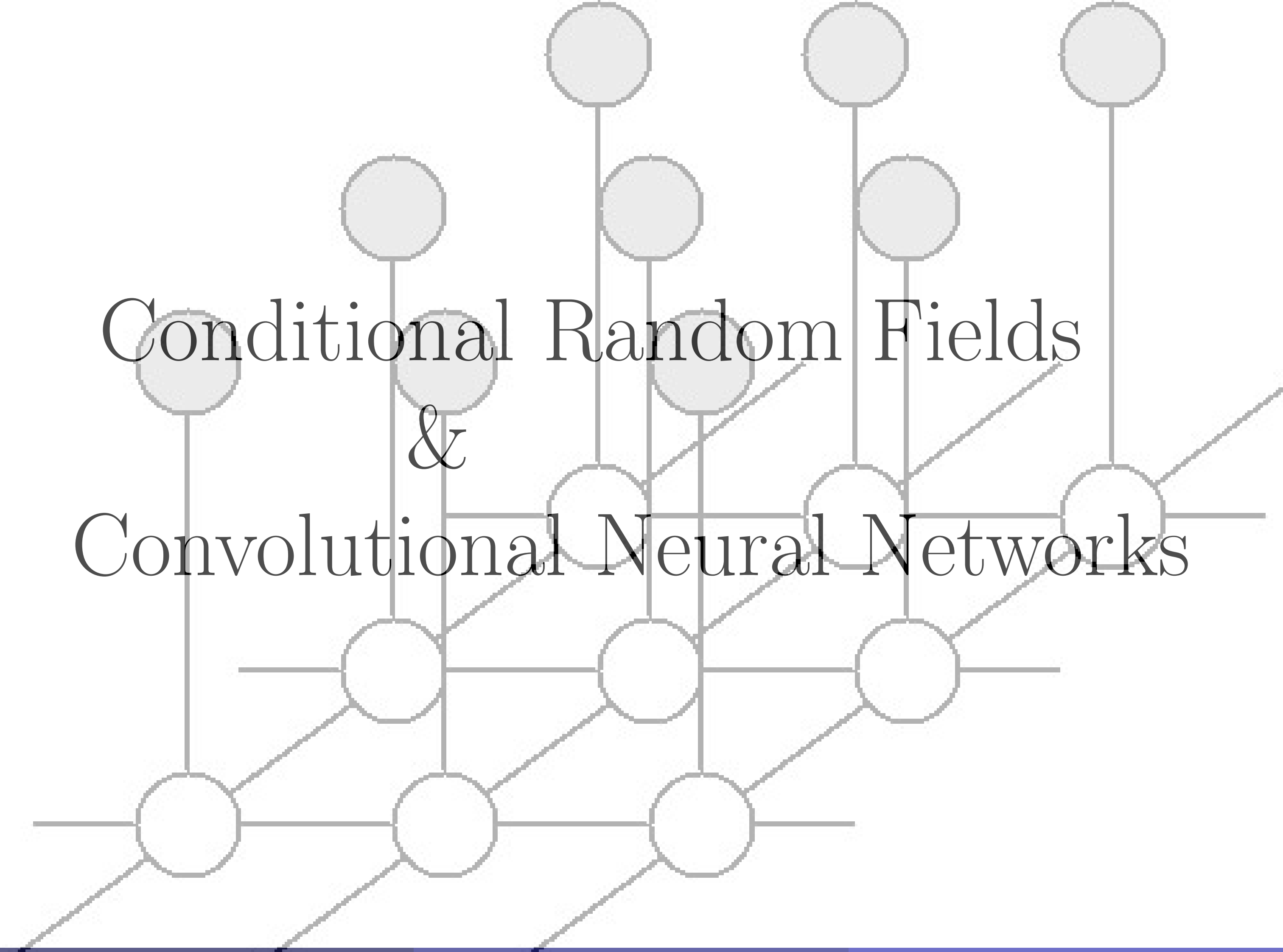


Fully Convolutional Neural Netwoks: U-Net

- U-Net (2015)
  - U-shaped architecture
  - biomedical image segmentation

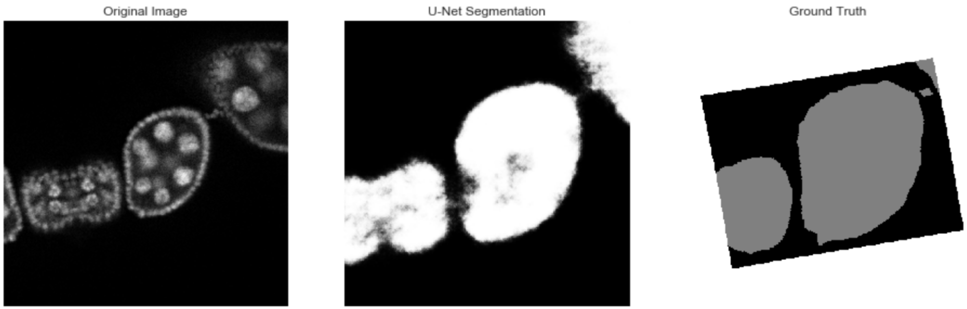


Conditional Random Fields
&
Convolutional Neural Networks

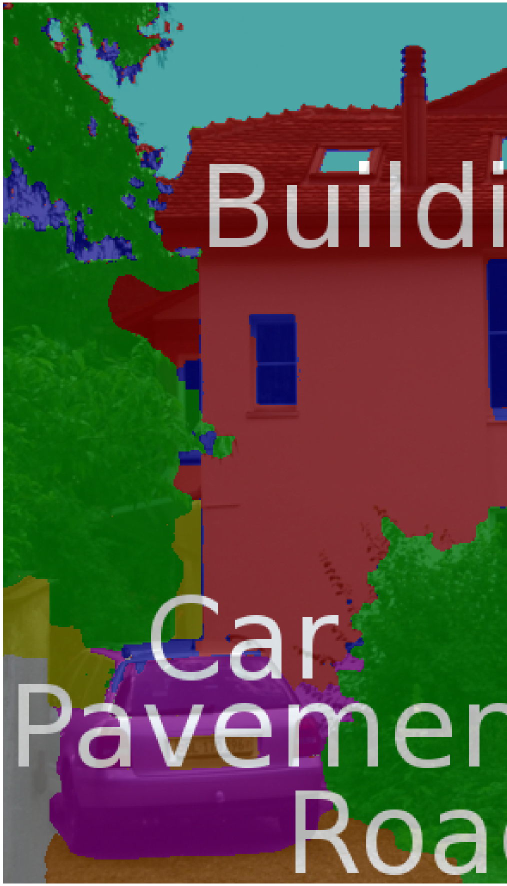


Drawbacks of fully connected CNNs

- CNNs ouperforms "classical" methods on majority of semantic segmentations tasks
- they still fail in some cases

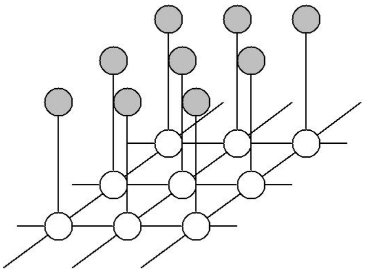


- Cross entropy loss $E = -\sum_x \sum_c p_{xc} \log(y_{xc})$
    - does not consider global consistency
    - can only use information within its receptive field
        - $\rightarrow$ "holes" in the segmentation map
        - $\rightarrow$ inconsistent segmentation

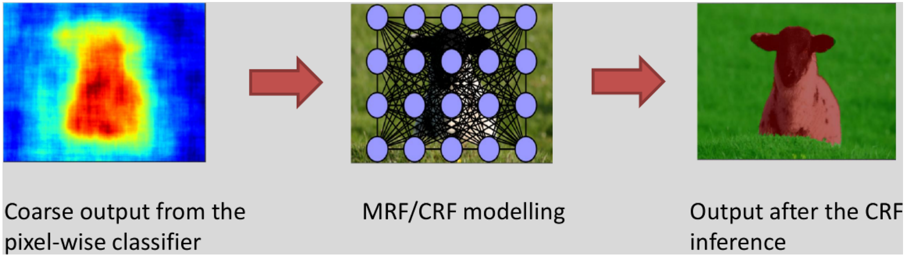


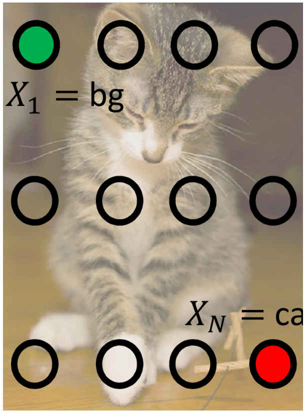
Conditional Random Fields: Introduction

- Conditional Random Field (CRF)
  - probabilistic graphical model
  - two types of variables:
    - observations (e.g. original image)
    - random variables (e.g. pixel-wise segmentation)
  - used for structured prediction
    e.g. semantic segmentation
  - these days outperformed by CNNs



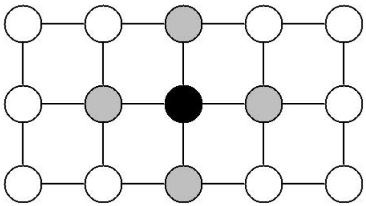
- CRF as a post-processing method
  to improve CNN output



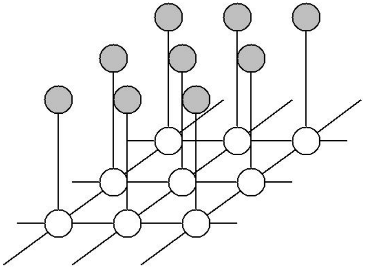


Conditional Random Fields: Formal Definition

- Markov Random Field (MRF)
  - $G = (V, E)$ undirected graph
  - $Y = \{Y_v\}_{v \in V}$
  - $Y_v | Y_{N(v)} \perp\!\!\!\perp Y_{V \setminus N[v]} | Y_{N(v)}$

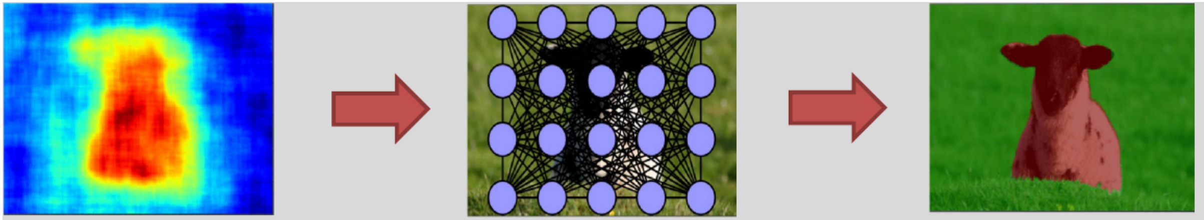


- Conditional Random Field (CRF)
  - $G = (V, E)$ undirected graph
    - $V = \{1, ..., N\}$ ... pixels
    - E ... define neighbors
  - $I = \{I_i\}_1^N$ ... observation for each pixel
  - $X = \{X_i\}_1^N$ ... random variable for each pixel
  - $X|I$ is a Markov Random Field (MRF)
    $\rightarrow P(X = x|I) = \frac{1}{Z(I)} \exp\{-E(x|I)\}$



Conditional Random Fields Postprocessing

- Goal: maximize $P(X = x|I) = \frac{1}{Z(I)} \exp\{-E_I(x)\}$
  - I ... CNN segmentation
  - X ... final segmentation
- $\rightarrow$ minimize energy function:
  - $E_I(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j)$
  - NP-hard for general graphs
- $\rightarrow$ Mean Field Approximation



Coarse output from the pixel-wise classifier

MRF/CRF modelling

Output after the CRF inference

CRF: Mean Field Iteration as RNN

$$Q_i(x_i = l) = \frac{1}{Z_i} \exp\left\{ -\psi_u(x_i) - \sum_{l' \in \mathcal{L}} \mu(l, l') \sum_{m=1}^{K} w^{(m)} \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l') \right\}.$$

Algorithm 1 Mean field in fully connected CRFs

Initialize Q ▷ $Q_i(x_i) \leftarrow \frac{1}{Z_i} \exp\{-\phi_u(x_i)\}$

while not converged **do** ▷ See Section 6 for convergence analysis

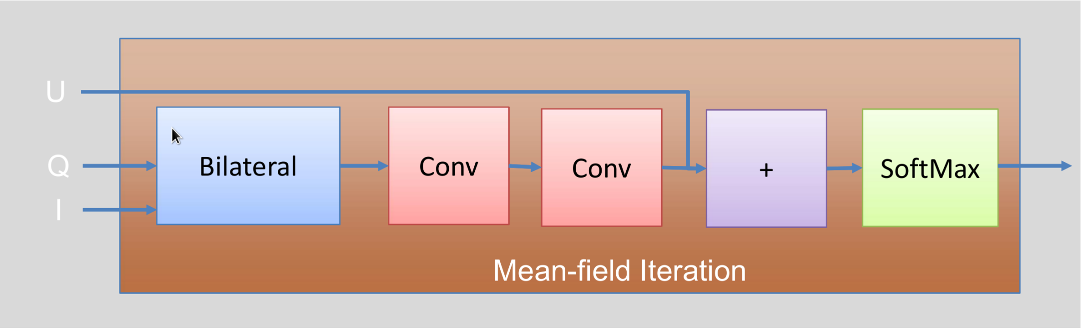
  $\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m ▷ **Message passing** from all X_j to all X_i

  $\hat{Q}_i(x_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$ ▷ **Compatibility transform**

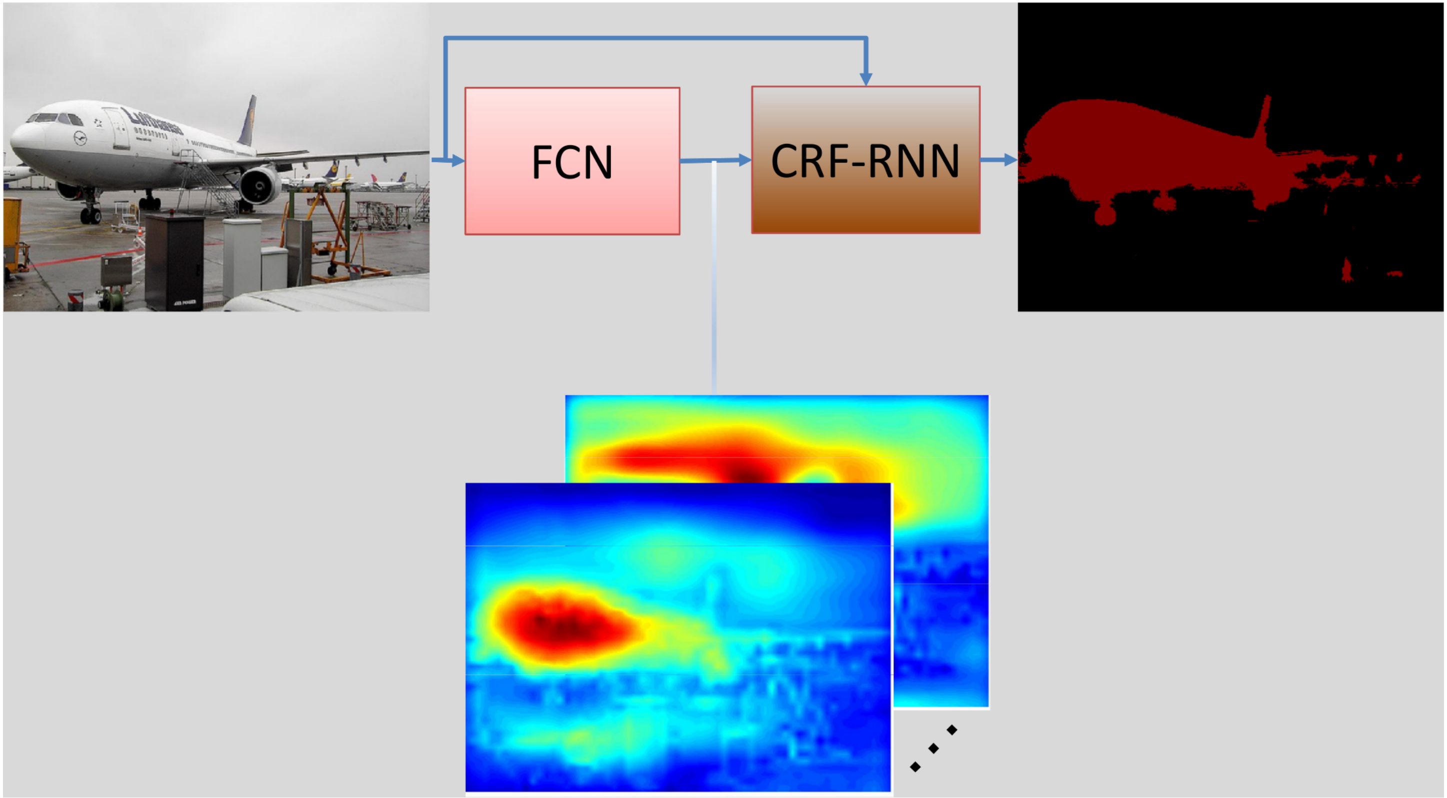
  $Q_i(x_i) \leftarrow \exp\{-\psi_u(x_i) - \hat{Q}_i(x_i)\}$ ▷ **Local update**

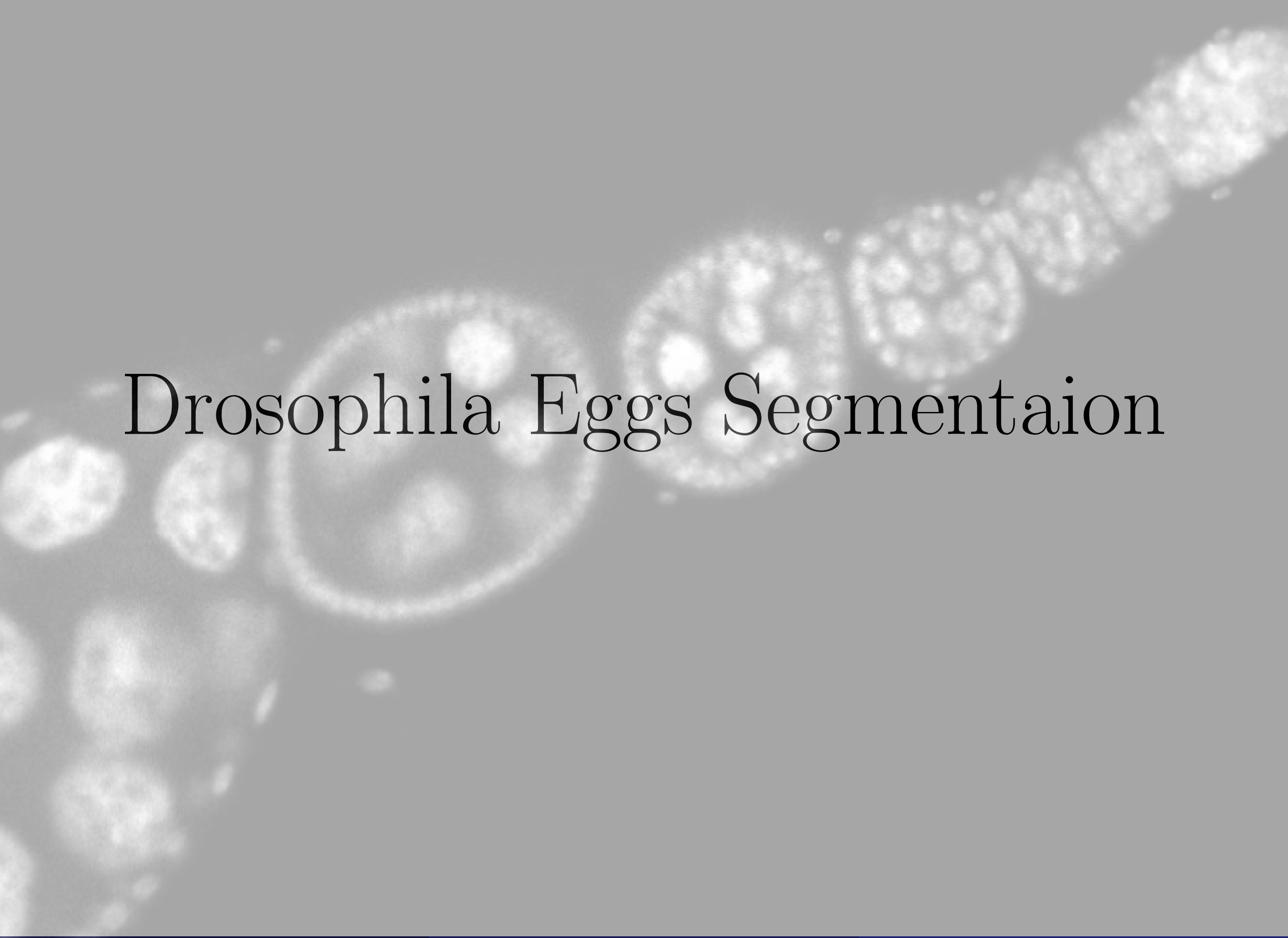
  normalize $Q_i(x_i)$

end while



CRF as RNN

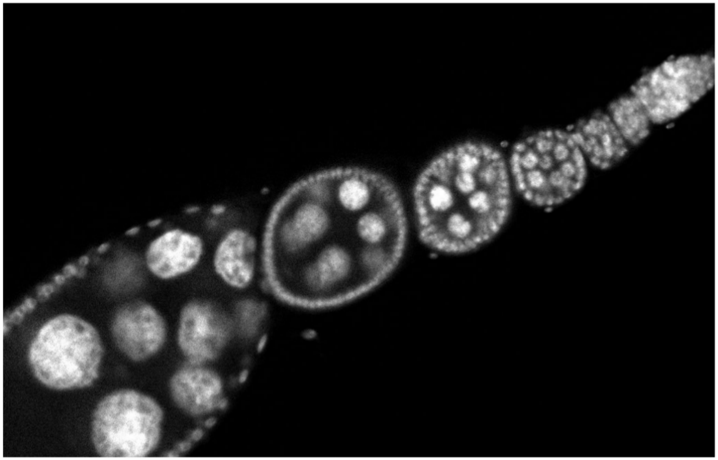




Drosophila Eggs Segmentaion

Problem Definition

- Binary segmentation of microscopy scans
  - classes: the eggs and the background

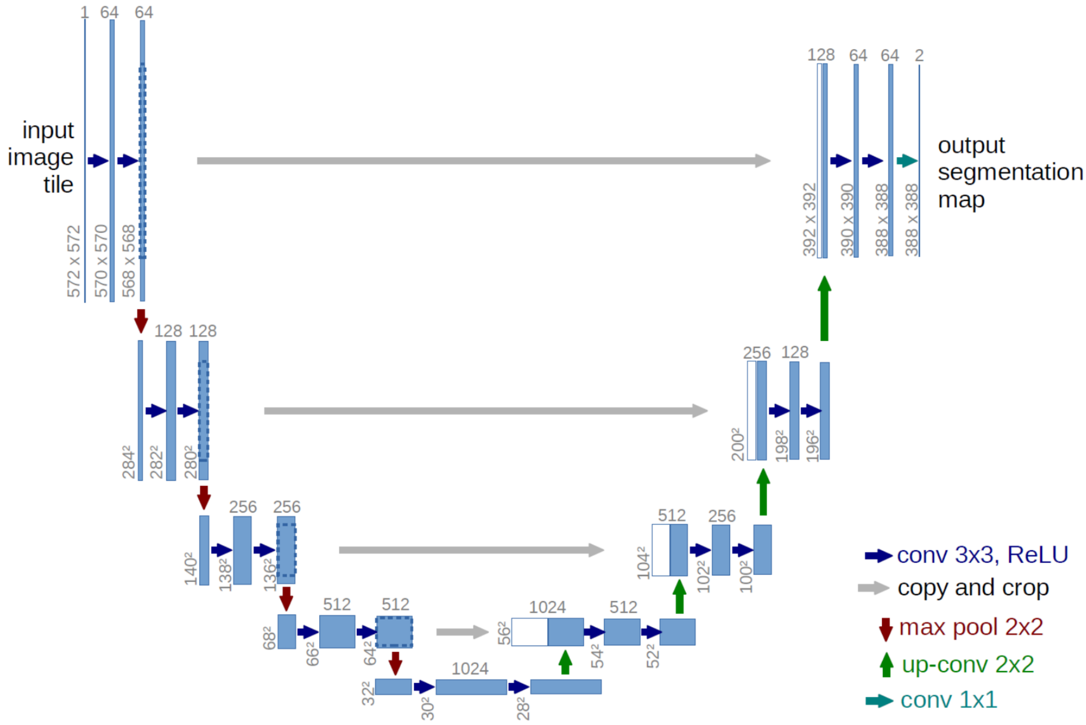


(a) Input slice

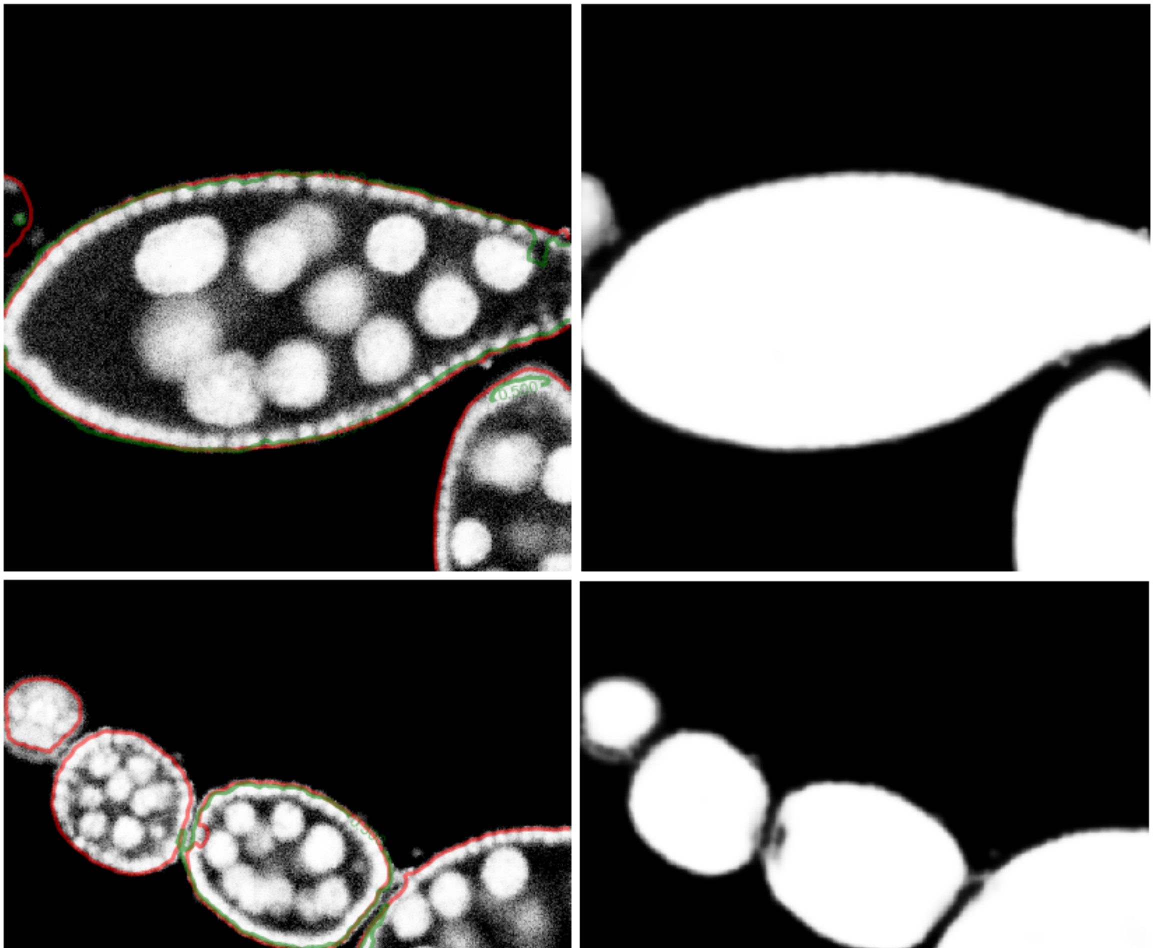
(b) Desired output segmentation

- 75 manually annotated images of drosophila eggs
- 4000 unlabeled images

- U-shaped Convolutional Neural Network
  - the usual convolution-pooling path followed by the upsampling path
  - output: segmentation map of the input image



Results



Thank You!

Links

- http://cs231n.stanford.edu/
- https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html
- http://www.robots.ox.ac.uk/ szheng/CRFasRNN.html
- https://www.tensorflow.org