Deconvolution Networks

Johan Brynolfsson

Mathematical Statistics
Centre for Mathematical Sciences
Lund University

December 6th 2016

Deconvolution Neural Networks

Image Deconvolution

- "True Deconvolution"
- Using neural network to deconvolve blurring in an image.
- For the unblurring to be effective, large convolutional kernels must be used. This however is both hard to optimise and expensive computationally.
- L. Xu et al presents a method in **A Deep Convolutional Neural Network for Image Deconvolution**.

Image Deconvolution

- A simple model is used to model the blurring

$$y = x * k \tag{1}$$

- The image can be reconstructed using convolution

$$x = \mathcal{F}^{-1}\left(\mathcal{F}\left(y\right)/\mathcal{F}\left(k\right)\right) = \mathcal{F}^{-1}\left(1/\mathcal{F}\left(k\right)\right) * y \tag{2}$$

- A Wiener deconvolution is proposed

$$x = \mathcal{F}^{-1}\left(\frac{1}{\mathcal{F}\left(k\right)}\left\{\frac{|\mathcal{F}(k)|^2}{|\mathcal{F}(k)|^2 + \frac{1}{SNR}}\right\}\right) * y = k^{\dagger} * y \tag{3}$$

$$x = k^{\dagger} * y \tag{4}$$

- For the unblurring to be effective, large convolutional kernels must be used. This however is both hard to optimise and expensive computationally, even with the model.
- By introducing the noise term $k^{\dagger}$ becomes compact with finite support.
- The kernel separability is achieved using singular value decomposition, making the problem 1D

$$x = k^{\dagger} * y = \sum_j s_j \cdot u_j * \left(v_j^T * y\right) \tag{5}$$

Image Deconvolution

$$x = k^{\dagger} * y = \sum_j s_j \cdot u_j * \left(v_j^T * y\right) \tag{6}$$

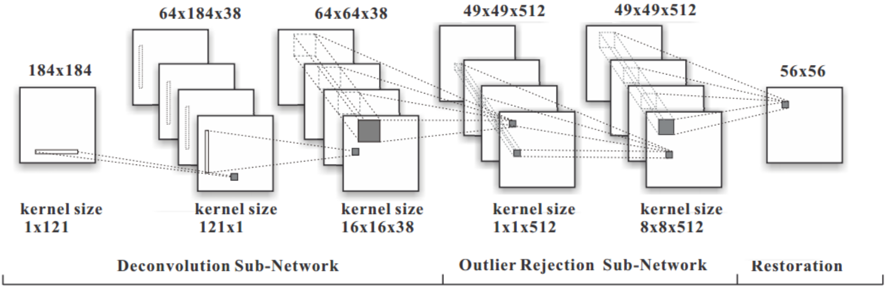


Image Deconvolution

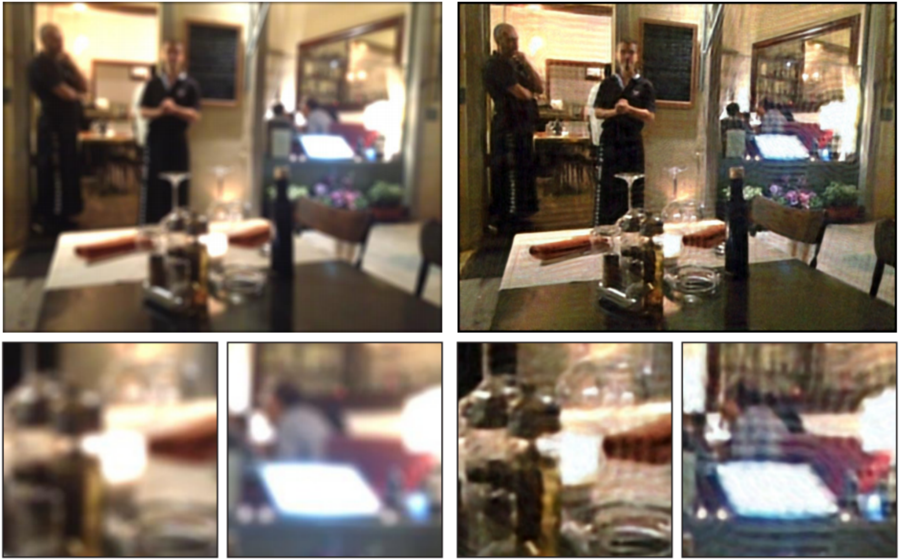
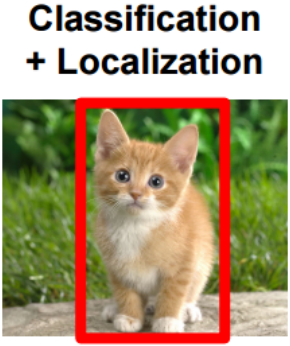


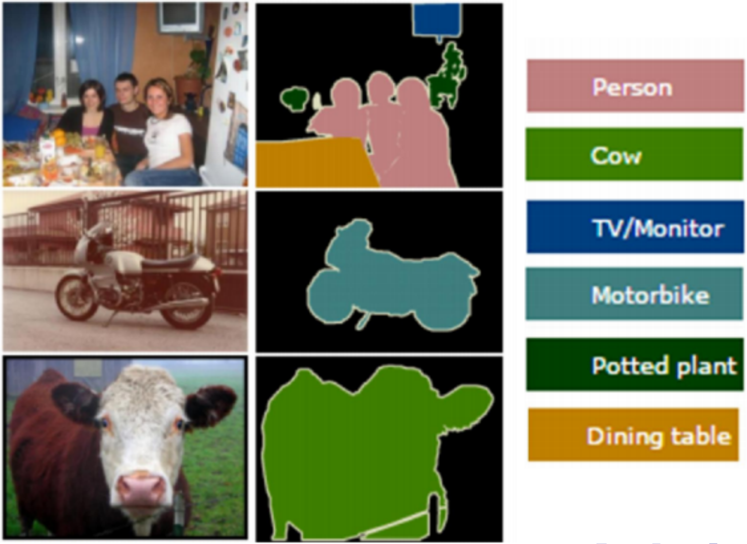
Image Deconvolution

Deconvolution Networks for Semantic Segmentation

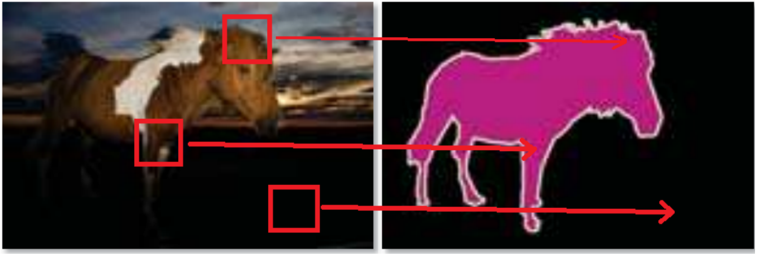
- Instead of only wanting a class prediction we also want a localisation of where in the image the object(s) is(are).
- One way to solve this is to produce 5 values - image class + two bounding box coordinates.



Semantic Segmentation

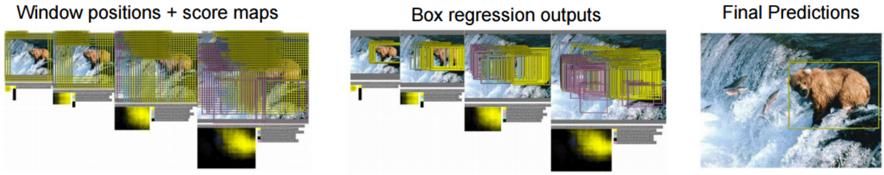


- Very naive approach: Pixel by pixel classification by sending separate patches into neural network.

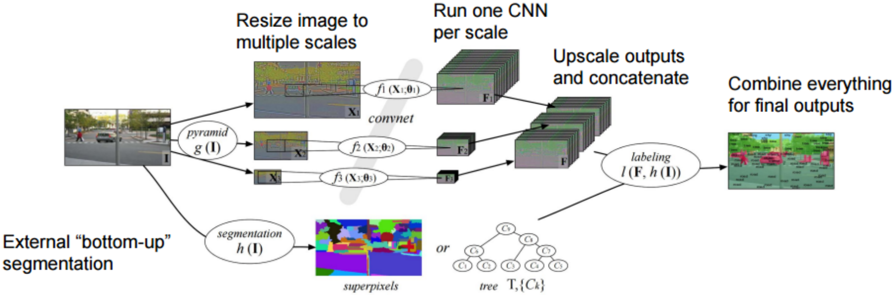


- Non-robust and noisy

- Less naive approach proposed by Sermanet et al in **Integrated Recognition, Localisation and Detection using Convolutional Networks**
- Pixel by pixel classification by sending separate patches into neural network.
- Repeat using multiple different size of boxes

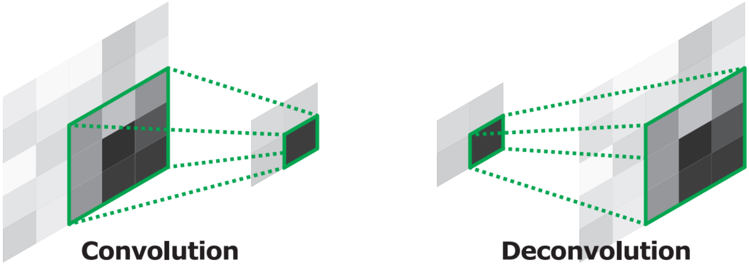


- Farabet et al proposes a multi-scale setup in **Learning Hierarchical Features for Scene Labeling**, together with an image section segmentation

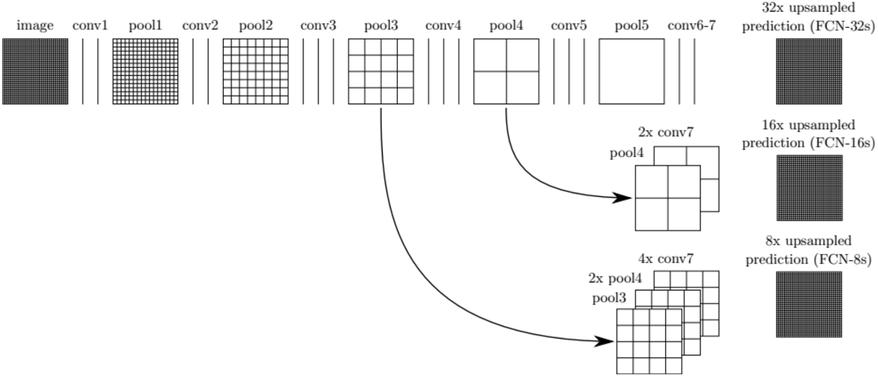


Deconvolution

- Disliked name, but seems to have have stuck
- Also called transpose convolution, fractional strided convolution, or backwards convolution.

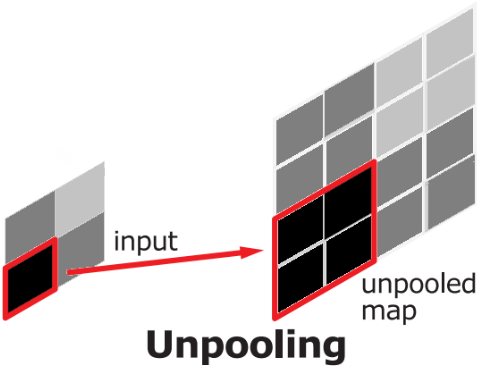


- Long et al proposes deconvolution in **Fully Convolutional Networks for Semantic Segmentation**



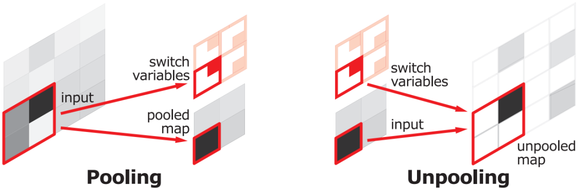
Unpooling

- Used to sample up the image between deconvolution
- First proposed to be used naively, by up-sampling each pixel to e.g. 2x2 pixels.



Unpooling

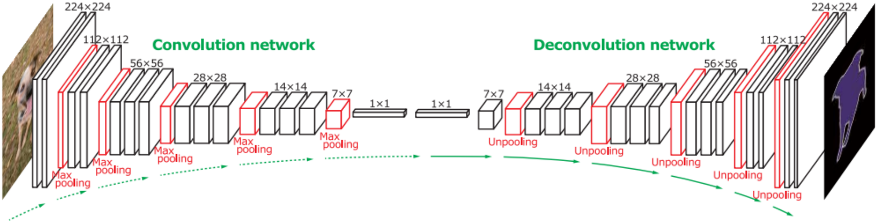
- Noh et al proposes in an improved unpooling in **Learning Deconvolution Network for Semantic Segmentation**
- In each max-pooling layer, the coordinates of max-value is stored.
- In a corresponding unpooling layer, values from a previous layer are entered into stored coordinated, setting the rest of the pixels to zero.



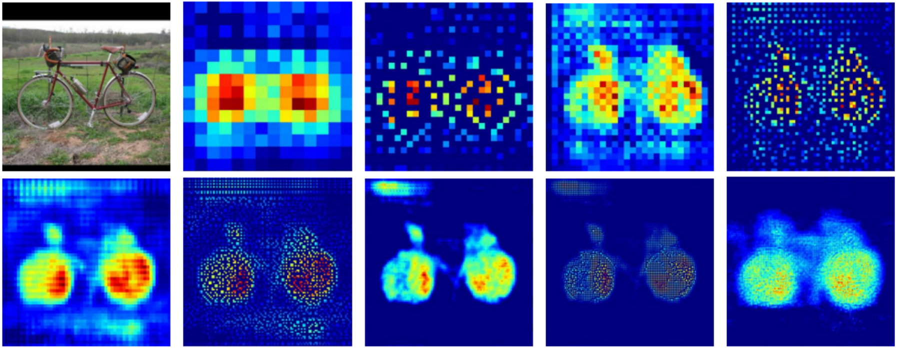
- By doing this, localisation information is included in the unpooling.

Deconvolution Networks

- Noh et al proposes a VGG 16-layer net, with classification layer removed
- The structure is reversed, with deconvolution and unpooling, to produce a semantic segmentation

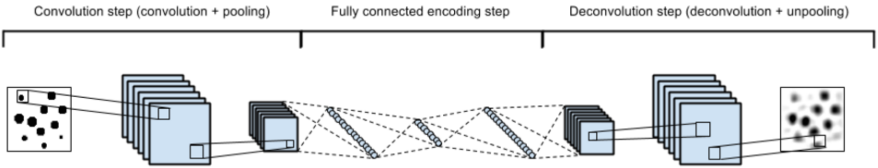


- An example of deconvolution/unpooling

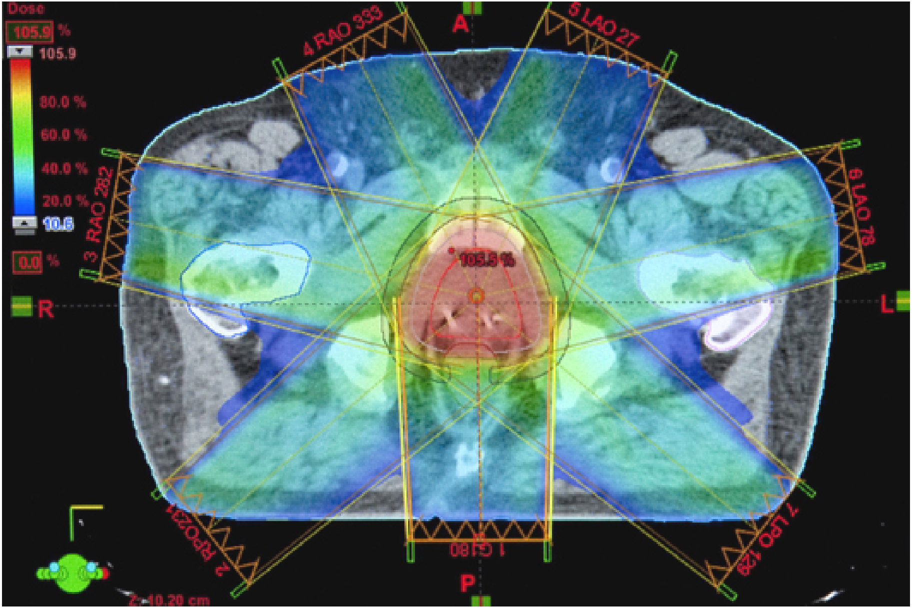


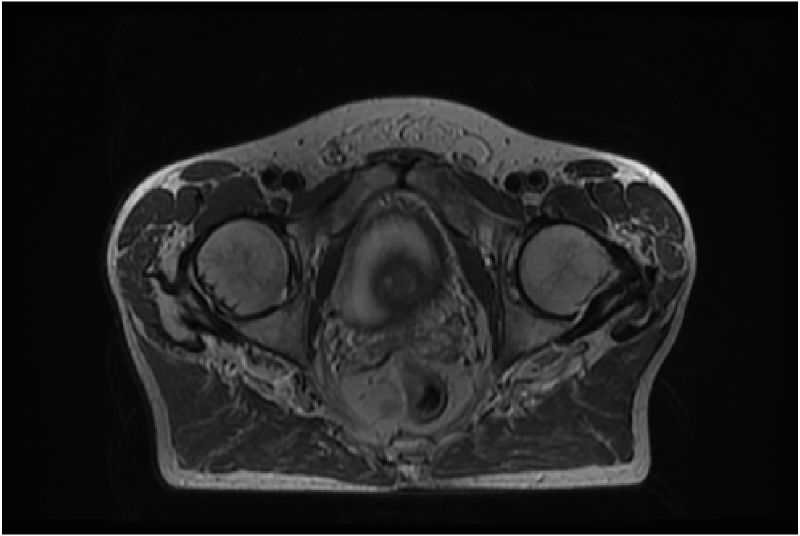
Neural network deconvolution for auto-encoding

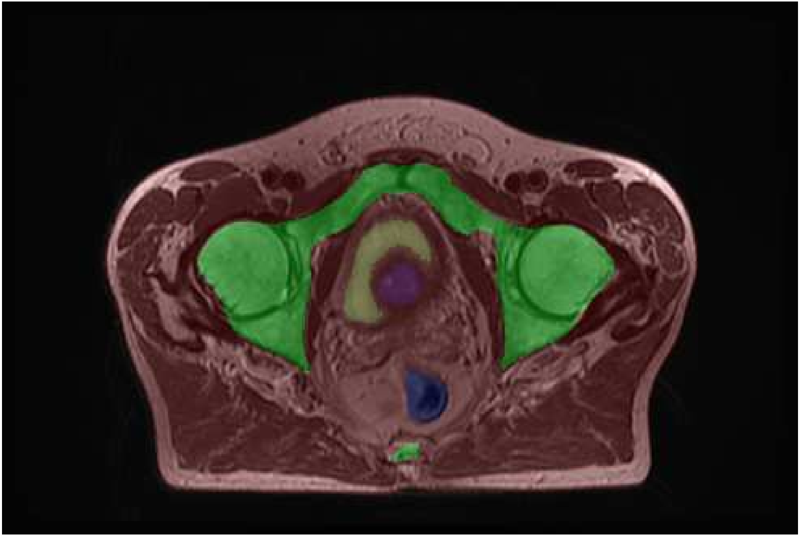
- The deconvolution network may also be used for auto encoding, where the image itself is given as ground-truth



https://github.com/mikesj-public/convolutional_autoencoder/tree/master

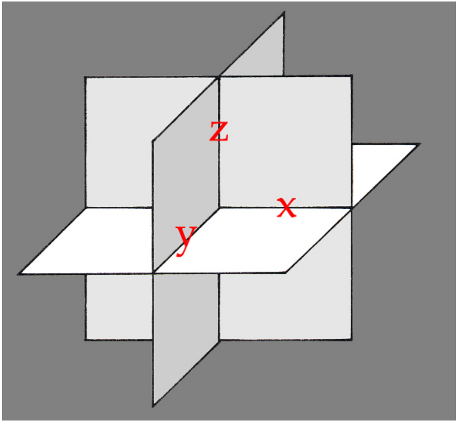

Dose
105.9 %
105.9
80.0 %
60.0 %
40.0 %
20.0 %
10.6
0.0 %
A
P
R
L
4 RAO 333
5 LAO 27
3 RAO 282
6 LAO 78
2 RPO231
7 LPO 128
105.9 %





- Physicians spend $\sim$ 1h / scan segmenting!
- Huge amount of data (500x500x250 = 62’500’000 voxels)
- Noisy data
- Poor resolution
- Arbitrary intensities
- Tumors, hernias, prosthetics, tatoos etc.

- A successful solution is to extract data around each voxel



- Use multiple networks
- Cure segmentation using shape awareness

HOMEWORK
(One of the following is sufficient)

- Install and test the deconvolution network for semantic segmentation (caffe)
  https://github.com/HyeonwooNoh/DeconvNet
- Install and test deconvolution for auto-encoding and compare with fully connected auto-encoding for the mnist dataset (theano)
  https://github.com/mikesj-public/convolutional_autoencoder/tree/master