COMP 448/548: Medical Image Analysis

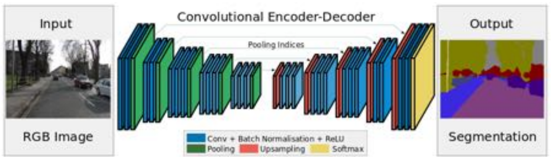
Dense prediction networks

Çiğdem Gündüz Demir
cgunduz@ku.edu.tr

1

Dense prediction networks

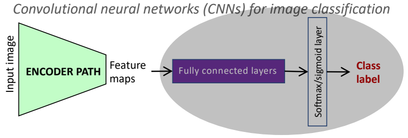
- They recover a larger-size segmentation map from the compressed image
  - Downsampling path captures semantic/contextual information
  - Upsampling path recovers spatial information
  - No fully connected layer is used on the top
  - Skip connections (concatenations) from downsampling to upsampling layers are often used to recover the fine-grained spatial information lost in the downsampling path



2

Fully convolutional networks (FCNs)

Convolutional neural networks (CNNs) for image classification



- Converts all fully connected layers to convolutions
- Adds a layer that predicts probabilities of classes at each coarse output location
- Upsamples the coarse outputs to pixel-dense outputs via *deconvolution* (backward strided convolution)

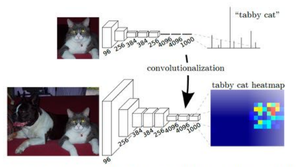


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Long et al, 2015. Fully convolutional networks for semantic segmentation. CVPR 2015.
https://arxiv.org/pdf/1411.4038.pdf

3

Fully convolutional networks (FCNs)

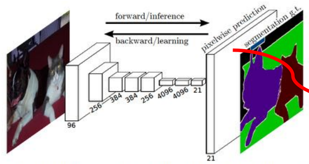
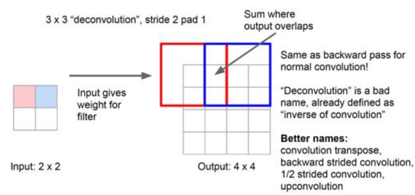


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

Learnable upsampling: "deconvolution"



Long et al, 2015. Fully convolutional networks for semantic segmentation. CVPR 2015.
https://arxiv.org/pdf/1411.4038.pdf

Slide credit: F.-F. Li, A. Karpathy, J. Johnson

4

Fully convolutional networks (FCNs)

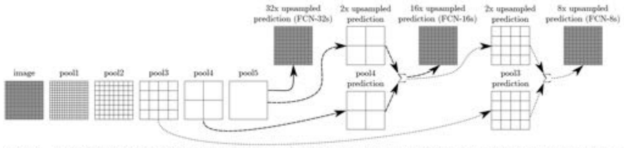


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the pool4 layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from pool3, at stride 8, provide further precision.

- Combines layers of the feature hierarchy and refines the spatial precision of the output

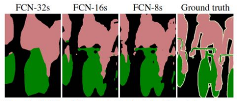
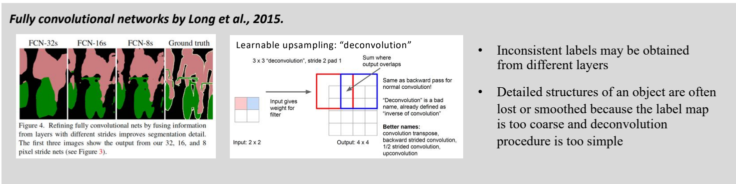


Figure 4. Refining fully convolutional nets by fusing information from layers with different strides improves segmentation detail. The first three images show the output from our 32, 16, and 8 pixel stride nets (see Figure 3).

Long et al, 2015. Fully convolutional networks for semantic segmentation. CVPR 2015.
https://arxiv.org/pdf/1411.4038.pdf

5

Learning deconvolution networks



➢ **Also learn a multi-layer deconvolution network, which is composed of successive deconvolution, unpooling, and rectified linear unit (ReLU) layers**

Noh et al, 2015. Learning deconvolution network for semantic segmentation. ICCV 2015.
https://arxiv.org/pdf/1505.04366.pdf

6

Learning deconvolution networks

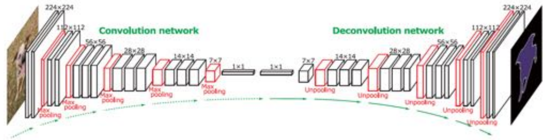


Figure 2. Overall architecture of the proposed network. On top of the convolution network based on VGG 16-layer net, we put a multi-layer deconvolution network to generate the accurate segmentation map of an input proposal. Given a feature representation obtained from the convolution network, dense pixel-wise class prediction map is constructed through multiple series of unpooling, deconvolution and rectification operations.

- Convolution network corresponds to a feature extractor that transforms the input image to multidimensional feature representation
- Deconvolution network is a shape generator that produces object segmentation from the features extracted from the convolution network
- Final output is a map of class probabilities

Noh et al, 2015. Learning deconvolution network for semantic segmentation. ICCV 2015.
https://arxiv.org/pdf/1505.04366.pdf

7

Learning deconvolution networks

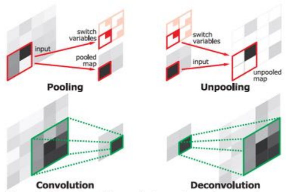


Figure 3. Illustration of deconvolution and unpooling operations.

Unpooling
- Records the locations of maximum activations selected during pooling in switch variables, which are employed to place each activation back to its original pooled location
- Its output is an enlarged, yet sparse activation map

Deconvolution
- Densifies sparse activations obtained by unpooling through convolution-like operations with multiple learned filters
- Convolutional layers connect multiple inputs to a single activation whereas deconvolutional layers associate a single input with multiple outputs

Noh et al, 2015. Learning deconvolution network for semantic segmentation. ICCV 2015.
https://arxiv.org/pdf/1505.04366.pdf

8

Long skip connections (U-Net architecture)

- Long skip connections are defined between an encoder layer and the corresponding decoder layer
  - Deconvolution is applied on the concatenation of high-resolution features of an encoder layer and the output of the previous upsampling layer
  - Helps better recover the fine-grained spatial information lost in downsampling
  - U-shaped architecture with an equal number of downsampling and upsampling layers

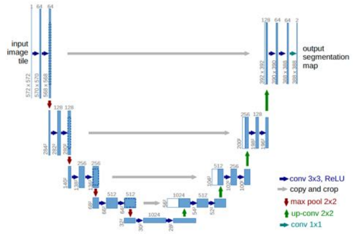


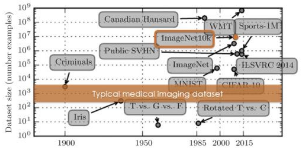
Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Ronneberger et al, 2015. U-net: Convolutional networks for biomedical image segmentation. MICCAI 2015.
https://arxiv.org/pdf/1505.04597.pdf

9

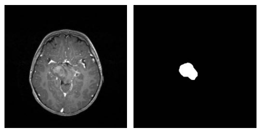
Challenges for medical image segmentation

1. Limited training data

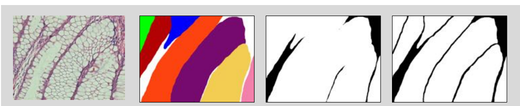


www.quantib.com/blog/deep-learning-radiology-and-challenges-radiology-ai

2. Imbalanced classes



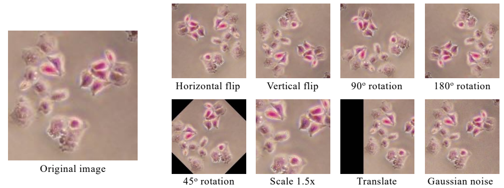
3. Separation of touching objects



10

Data augmentation

- Apply some transformations to existing images
  - Flipping, rotation, scaling, translation, adding Gaussian noise, …
  - Interpolation techniques may be necessary to preserve original image sizes
  - Some methods may not be applicable to particular applications



- Generate synthetic data using generative adversarial networks (GANs)

11

Loss functions for foreground/background classification

$$loss = \sum_{I \in D_{tr}} \sum_{p \in I} loss(y_p, \hat{y}_p)$$

$$\underbrace{loss(y_p, \hat{y}_p) = -y_p \cdot \log(\hat{y}_p) - (1 - y_p) \cdot \log(1 - \hat{y}_p)}_{\textit{Binary cross entropy}}$$

D_{tr} is the training set

$I \in D_{tr}$ is an image in the training set

$p \in I$ is a pixel in the image

$$y_p = \begin{cases} 1 & p \in \text{ foreground} \\ 0 & p \in \text{ background} \end{cases}$$

$\hat{y}_p$ is the estimated probability for p being a foreground pixel

12

Weighted loss functions

$$loss = \sum_{I \in D_{tr}} \sum_{p \in I} c_p \cdot loss(y_p, \hat{y}_p)$$

Pixel weight $c_p = cw_p$ can be selected according to the frequency of pixels in the classes

- All pixels belonging to the same class have the same weight (contribution) in the loss function
- These weights are typically selected inversely proportional to the pixel frequencies

To address the issue of separating touching objects, it is possible to give more importance to correctly classifying pixels close to object boundaries

$$c_p = cw_p + w_0 \cdot \exp\left(-\frac{[d_1(p) + d_1(p)]^2}{2\,\sigma^2}\right)$$

$d_1(p)$ is the distance from pixel p to the border of the nearest foreground object

$d_2(p)$ is the distance from pixel p to the border of the second nearest foreground object

w_0 and σ are the parameters

Ronneberger et al, 2015. U-net: Convolutional networks for biomedical image segmentation. MICCAI 2015.
https://arxiv.org/pdf/1505.04597.pdf

13

Focal loss for foreground/background classification

$$\underbrace{loss(y_p, \hat{y}_p) = -y_p \cdot \log(\hat{y}_p) - (1 - y_p) \cdot \log(1 - \hat{y}_p)}_{\textit{Binary cross entropy}}$$

$$\underbrace{loss(y_p, \hat{y}_p) = -y_p \cdot (1 - \hat{y}_p)^{\gamma} \cdot \log(\hat{y}_p) - (1 - y_p) \cdot \hat{y}_p^{\gamma} \cdot \log(1 - \hat{y}_p)}_{\textit{Focal loss for binary classification}}$$

Larger values of the focusing parameter γ reduces the weights of easy-to-learn pixels more, resulting in relatively more focusing on learning hard-to-learn pixels

- When a foreground pixel p is
  - Misclassified and $\hat{y}_p$ is small, the modulating factor $(1 - \hat{y}_p)^{\gamma}$ is close to 1 and the loss is almost unaffected
  - Correctly classified and $\hat{y}_p \to 1$, the factor goes to 0 and the loss of this well-classified pixel is downweighed
- When a background pixel p is
  - Misclassified and $\hat{y}_p$ is large, the modulating factor $\hat{y}_p^{\gamma}$ is close to 1 and the loss is almost unaffected
  - Correctly classified and $\hat{y}_p \to 0$, the factor goes to 0 and the loss of this well-classified pixel is downweighed

Lin et al, 2018. Focal loss for dense object detection. *https://arxiv.org/pdf/1708.02002.pdf*

14

Dice loss for foreground/background classification

- Having a smaller number of foreground pixels causes problem
  - Network training is typically biased towards classifying pixels as background (negative)
  - This leads to high precision TP / (TP + FP) but low recall TP / (TP + FN) values
  - This is undesirable especially in medical applications where FNs are much less tolerable than FPs

$$\text{Dice loss} = 1 - \frac{2 \sum_p y_p \cdot \hat{y}_p}{\sum_p y_p + \sum_p \hat{y}_p}$$

$$\text{Dice coefficient} = \frac{2\,TP}{2\,TP + FP + FN} = \frac{2\,TP}{(TP + FN) + (TP + FP)} = \frac{2\,TP}{actual\ P + estimated\ as\ P}$$

This definition gives equal importance to false positive and false negative pixels

Sudre et al, 2017. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations.
https://arxiv.org/pdf/1707.03237.pdf

15

Tversky loss for foreground/background classification

- Having a smaller number of foreground pixels causes problem
  - Network training is typically biased towards classifying pixels as background (negative)
  - This leads to high precision TP / (TP + FP) but low recall TP / (TP + FN) values
  - This is undesirable especially in medical applications where FNs are much less tolerable than FPs

$$\text{Tversky loss} = 1 - \frac{\sum_p y_p \cdot \hat{y}_p}{\sum_p y_p \cdot \hat{y}_p + \underset{\uparrow}{\alpha} \sum_p \hat{y}_p \cdot (1 - y_p) + \underset{\uparrow}{\beta} \sum_p y_p \cdot (1 - \hat{y}_p)}$$

penalty for FPs *penalty for FNs*

Allows giving different levels of importance to false positive and false negative pixels

Salehi et al, 2017. Tversky loss function for image segmentation using 3D fully convolutional deep networks.
https://arxiv.org/pdf/1706.05721.pdf

16

Adaptive loss adjustment

- Multi-stage network with a loss adjustment mechanism based on adaptive boosting
- Modulates the attention of each stage to correct the mistakes of previous stages by adjusting the loss weight of each pixel prediction separately with respect to how accurate the previous stages are on this pixel

$$\beta_n(p) = \begin{cases} 1 - |\hat{y}_n(p) - 0.5| & \text{if } \hat{y}_n(p) \text{ is correct} \\ 1 + |\hat{y}_n(p) - 0.5| & \text{if } \hat{y}_n(p) \text{ is incorrect} \end{cases}$$

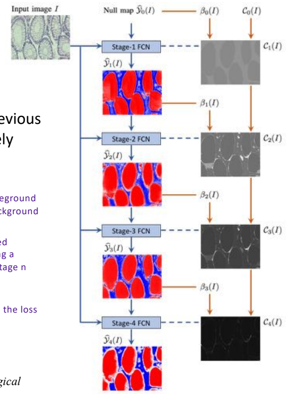
$$c_{n+1}(p) = \beta_n(p) \cdot c_n(p)$$

$$loss_n = \sum_{I \in D_{tr}} \sum_{p \in I} c_n(p) \cdot loss\big(y(p), \hat{y}_n(p)\big)$$

$$y_p = \begin{cases} 1 & p \in \text{foreground} \\ 0 & p \in \text{background} \end{cases}$$

$\hat{y}_n(p)$ is the estimated probability for p being a foreground pixel at stage n

$c_n(p)$ is the weight (contribution) of p in the loss function at stage n

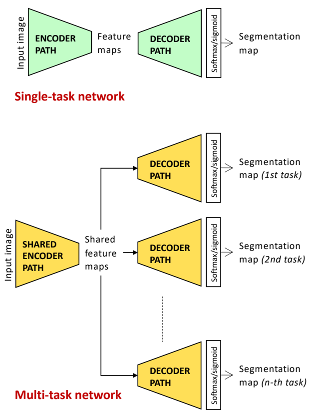


Gunesli et al., 2020. AttentionBoost: Learning what to attend for gland segmentation in histopathological images by boosting fully convolutional networks. IEEE Transactions on Medical Imaging.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9164886

17

Multi-task networks

- Dense prediction networks that learn related but different tasks from shared feature representations
- They consist of a shared encoder path and multiple decoder paths, one defined for each task
- Joint loss is defined usually as a linear combination of losses defined on all tasks
- All tasks are concurrently learned in parallel by training the network to minimize this joint loss
- This approach helps better avoid local optimal solutions as it is less likely to finetune the weights of the shared encoder for all tasks at the same time



18

Contour-aware networks

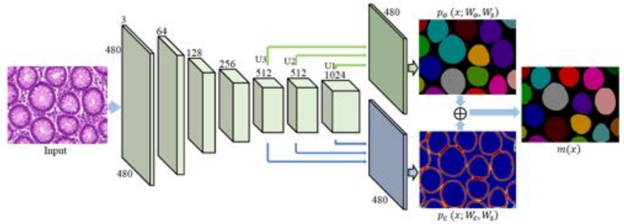
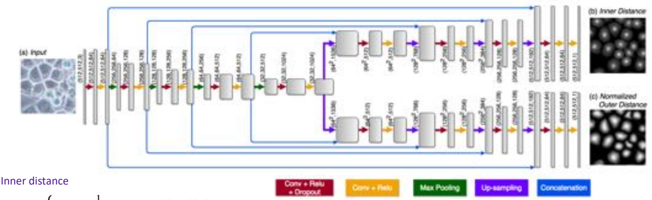


Figure 3: The overview of the proposed deep contour-aware network.

Chen et al., 2017. DCAN: Deep contour-aware networks for accurate gland segmentation. Medical Image Analysis.
https://www.sciencedirect.com/science/article/pii/S1361841516302043

19

Multi-task regression network

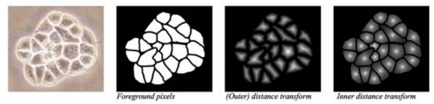


Inner distance

$$d(q) = \begin{cases} \dfrac{1}{1 + \alpha \, \|q - C(a_i)\|^2} & \text{if } q \in P(a_i) \\ 0 & \text{if } q \in \text{background} \end{cases}$$

Normalized outer distance

$$d(q) = \begin{cases} \dfrac{\min\limits_{b_{ik} \in B(a_i)} \|q - b_{ik}\|^2}{\max\limits_{r \in P(a_i)} \min\limits_{b_{ik} \in B(a_i)} \|r - b_{ik}\|^2} & \text{if } q \in P(a_i) \\ 0 & \text{if } q \in \text{background} \end{cases}$$

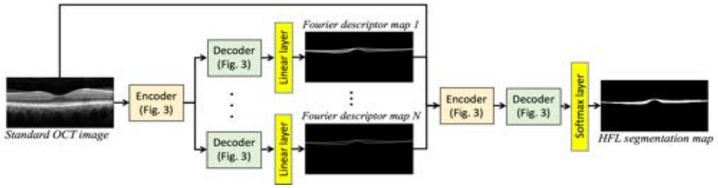


Koyuncu et al., 2020. DeepDistance: A multi-task deep regression model for cell detection in inverted microscopy images. Medical Image Analysis.
https://www.sciencedirect.com/science/article/pii/S1361841520300840?via%3Dihub

20

Shape-preserving cascaded network

- A cascaded network design:
  - Quantifies the shape of an object with a function defined on its contour
  - Expands this function in a Fourier series and use the harmonic amplitudes of its Fourier coefficients as the Fourier descriptors of the object
  - Defines a regression task of learning these descriptors



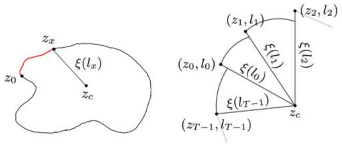
Cansiz et al., 2023. FourierNet: Shape-preserving network for Henle's fiber layer segmentation in optical coherence tomography images. IEEE Journal of Biomedical and Health Informatics. https://ieeexplore.ieee.org/document/9973287

21

Shape-preserving cascaded network

The distance-to-center function $\xi(l_x)$ outputs the distance from the object centroid z_c to the point z_x for which the arc length is l_x. This function is expanded in a Fourier series as

$$\begin{aligned}
\xi(l_x) &= a_0 + \sum_{n=1}^{\infty}\left[a_n \cos\left(\frac{2\pi n l_x}{L}\right) + b_n \sin\left(\frac{2\pi n l_x}{L}\right)\right] \\
a_n &= \frac{2}{L}\int_0^L \xi(l_x)\cos\left(\frac{2\pi n l_x}{L}\right) dl_x \\
b_n &= \frac{2}{L}\int_0^L \xi(l_x)\sin\left(\frac{2\pi n l_x}{L}\right) dl_x
\end{aligned}$$



For the curve γ_o, which is an interpolation of T discrete pixels, it can be divided into T intervals of $[l_{t-1}, l_t)$.

$$a_n = \frac{2}{L}\sum_{t=1}^{T}\int_{l_{t-1}}^{l_t} \xi(l_x)\cos\left(\frac{2\pi n l_x}{L}\right) dl_x = \frac{1}{\pi n}\sum_{t=1}^{T}\Delta\xi_t \sin\left(\frac{2\pi n l_t}{L}\right)$$

$$b_n = \frac{2}{L}\sum_{t=1}^{T}\int_{l_{t-1}}^{l_t} \xi(l_x)\sin\left(\frac{2\pi n l_x}{L}\right) dl_x = -\frac{1}{\pi n}\sum_{t=1}^{T}\Delta\xi_t \cos\left(\frac{2\pi n l_t}{L}\right)$$

Fourier descriptors of an object is defined as the first N harmonic amplitudes (in the polar coordinate) of its Fourier coefficients

Cansiz et al., 2023. FourierNet: Shape-preserving network for Henle's fiber layer segmentation in optical coherence tomography images. IEEE Journal of Biomedical and Health Informatics. https://ieeexplore.ieee.org/document/9973287

22

Image reconstruction as an auxiliary unsupervised task

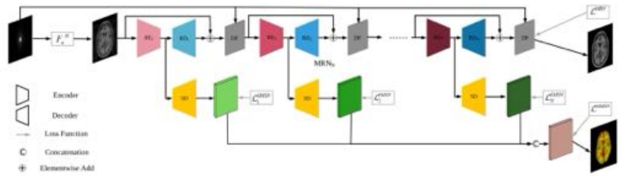


Figure 3: The SegNetMRI structure, formed by connecting the discussed MRN (top) for reconstruction, with MSN (bottom) for segmentation.

$$\mathcal{L}^{\text{SegNetMRI}} = \mathcal{L}^{\text{MRN}} + \lambda\mathcal{L}^{\text{OMSN}}$$

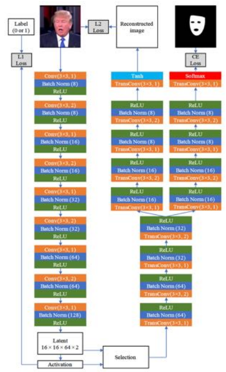
$$\mathcal{L}^{\text{MRN}} = \frac{1}{L}\sum\nolimits_{i=1}^{L}\left\|x_i^{fs} - x_i\right\|_2^2$$

$$\mathcal{L}^{\text{MSN}} = -\sum\nolimits_{i=1}^{L}\sum\nolimits_{j=1}^{N}\sum\nolimits_{c=1}^{C} t_{ijc}^{gt}\ln t_{ijc}$$

Sun et al., 2018. Joint CS-MRI reconstruction and segmentation with a unified deep network. https://arxiv.org/pdf/1805.02165.pdf

23

Image reconstruction as an auxiliary unsupervised task



$$\mathcal{L} = \gamma_{act}\mathcal{L}_{act} + \gamma_{seg}\mathcal{L}_{seg} + \gamma_{rec}\mathcal{L}_{rec}$$

$$\mathcal{L}_{seg} = \frac{1}{N}\sum_i \|m_i \log(s_i) + (1 - m_i)\log(1 - s_i)\|_1$$

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_i \|x_i - \hat{x}_i\|_2$$

$$\mathcal{L}_{act} = \frac{1}{N}\sum_i |a_{i,1} - y_i| + |a_{i,0} - (1 - y_i)|$$

Nguyen et al., 2019. Multi-task learning for detecting and segmenting manipulated facial images and video. https://arxiv.org/pdf/1906.06876.pdf

24

3D U-net for MR image segmentation

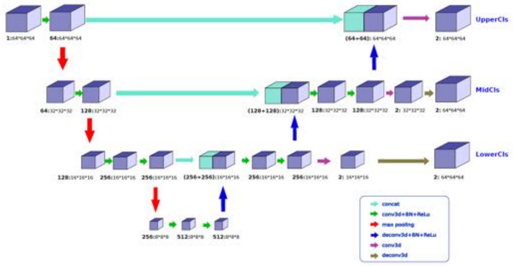


Fig. 1. Illustration of our proposed network architecture

Zeng et al., 2017. 3D U-net with multi-level deep supervision: Fully automatic segmentation of proximal femur in 3D MR images. MLMI 2017.
https://link.springer.com/chapter/10.1007/978-3-319-67389-9_32

Dice loss for 3D segmentation

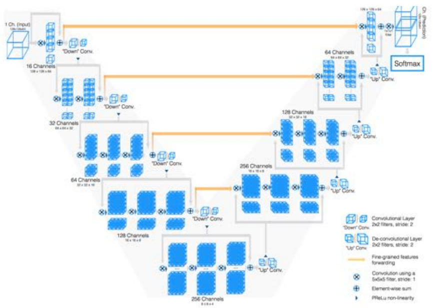


Fig. 2. Schematic representation of our network architecture. Our custom implementation of Caffe [5] processes 3D data by performing volumetric convolutions.

In order to deal with a strong imbalance between the number of foreground and background voxels, it defines an objective function based on the Dice coefficient

Milletari et al., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation.
https://arxiv.org/pdf/1606.04797.pdf

Voxelwise residual network for 3D segmentation

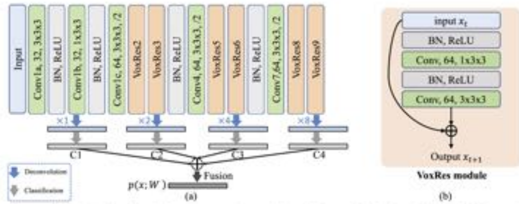
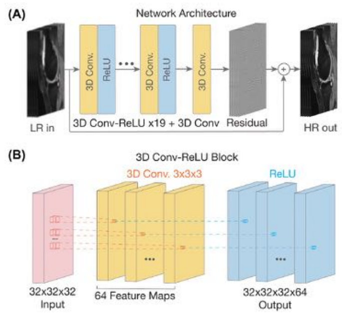


Fig. 2. (a) The architecture of proposed VoxResNet for volumetric image segmentation, consisting of batch normalization layers (BN), rectified linear units (ReLU), and convolutional layers N (ConvN) with number of channels, filter size and downsampling stride; (b) The illustration of VoxRes module.

Chen et al., 2018. VoxResNet: Deep voxelwise residual networks for brain segmentation from3D MR images. NeuroImage.
https://www.sciencedirect.com/science/article/pii/S1053811917303348

Super-resolution imaging using 3D networks



It designs a 3D network architecture to generate high-resolution thin slices from low-resolution thick slices

Chaudhari et al., 2018. Super-resolution musculoskeletal MRI using deep learning. Magnetic Resonance in Medicine.
https://onlinelibrary.wiley.com/doi/epdf/10.1002/mrm.27178

Thank you!

Next time:

Generative adversarial networks

29