Non-Cancer Region Identification of Prostrate Cancer using Different U-Net architectures

Bharat Nagaraju – A0178258N

Vigneshram Andiappan Selvaraj – A0178215A

Abstract

There is strong consensus that successful deep network training needs several thousand annotated training samples. Here we present a Fully Convoluted Neural Network and Training Strategy which is focused on the heavy use of data augmentation to allow more effective use of the available annotated samples .

The architecture consists of a context-capture contracting path and a symmetrical expanding path that allows for precise localisation. We show that such a network can be trained end-to-end from very few images and performs the best prior method (a convolutionary sliding window network) on the ISBI challenge of segmenting neuronal structures in electron microscopic stacks. We also tried the recently renowned, Dense Connections that attracted substantial attention in computer vision because they promote gradient flow and suggest deep supervision during training. In particular, DenseNet, which connects each layer in a feed-forward fashion with each other, has shown impressive performance in the tasks of classification of natural images. We are proposing HyperDenseNet, a fully convolutionary 3D neural network that extends dense connectivity concept to multi-modal segmentation problems. -- imaging modality has a path, and dense connections occur not only within the same path between the pairs of layers but also between those across different paths. This contrasts with the current multi-modal approaches to CNN, in which modelling multiple modalities relies entirely on a single joint layer (or level of abstraction) for fusion usually either at the input or at the network output. Therefore the network proposed has absolute freedom to learn more complex combinations between the modalities, within and above all levels of abstraction, which greatly enhances the learning representation. Unlike traditional networks, these links directly propagate gradients backwards, thereby reducing the gradient-disappearing problem and allowing for deeper networks. We also tried 2 more strategies but were less successful as compared to the neural network Fully convolute.

Introduction

Fully convoluted Neural Network

For several visual recognition tasks, deep convolutionary networks have achieved the state of the art over the past few years. Yet their effectiveness has been limited because of the size of the training sets available and the size of the networks considered. Older methods included training a network in a sliding-window setup to predict each pixel's class label by having a local (patch) region around that pixel. Second, it should localize the network. Second, in terms of patches, the training data is much greater than the number of training images. There are two disadvantages to this approach implemented by Ciresan et al. Second, it's very slow because the network has to run independently for each patch because there's a lot of redundancy due to patch duplication. Second, there is a trade-off between the precision of localization and the use of context. Larger patches need more max-pooling layers to minimize the accuracy of the localisation, while small patches allow the network to see only a small context. We have developed a more elegant architecture, the so-called fully convolutional network. We change and expand this architecture to work with very few training images and create more accurate segmentations. The key concept in is to complement successive layers of a normal contracting network, where pooling operators are replaced by upsampling operators. Such layers thus improve output resolution. High resolution features from the contracting path are combined with the upsampled output to localize. As for our tasks, there are very few training data available, we are using excessive data augmentation by adding elastic deformations to the training images available.

Dense connections in deep networks

In a wide range of computer vision problems, recently the implementation of residual learning in shortcut connections from early to late layers has become very common. Like traditional networks, these Back connections directly transmit gradients, thereby reducing the issue of gradient vanishing and allowing for deeper networks. In addition, they turn an whole network into a wide collection of shallower networks, providing competitive efficiency in various applications. The definition of shortcut connections was expanded by DenseNet, with the input of each layer corresponding to the outputs from all previous layers. Such a dense network promotes gradient flow and the learning of more complex patterns, resulting in significant improvements in accuracy and efficiency for classification tasks of natural images

NAS-Unet

Several studies have recently attempted to expand NAS to image segmentation that demonstrates preliminary feasibility. Both of them, however, concentrate on looking for semantine segmentation architecture in natural scenes. We designed three types of primitive search space operation, and two DownSC and UpSC cell architecture for semantic image segmentation, especially medical image segmentation. This includes separate primitive operation sets for searching for DownSC and UpSC on U-Like backbone network, respectively. But this one could not reporoduce the same findings as those stated in the articles of reference.

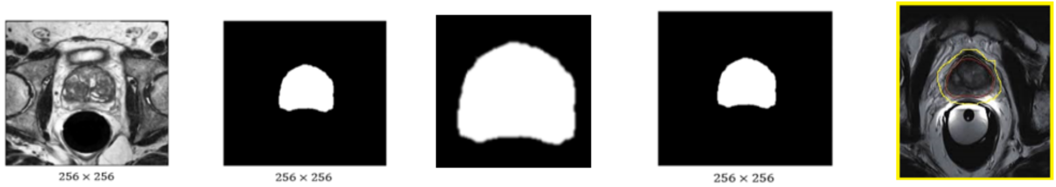
A NOVEL FOCAL TVERSKY LOSS FUNCTION WITH IMPROVED ATTENTION U-NET

We also tried a generalized focal loss function based on the Tversky index to answer the problem of data imbalance in segmentation of medical images. Our loss feature performs a better trade-off between precision and recall when working on small structures compared to the widely used loss in the Dice. One drawback of the Dice loss function is that it weights detections of false positives (FP) and false negatives (FN) equally. For action this results in high-precision, but low-recall segmentation charts. For highly imbalanced data and low ROIs, such as skin lesions, it is important to weigh FN detections higher than FPs to increase the rate of recall. The Tversky similarity index is a Dice score generalization which allows for flexibility in balancing FP and FNs. The network has the richest possible representation of features at the deepest level of encoding. However, spatial information tend to get lost in the high-level performance maps of cascaded convolutions and non-linearities. This makes it difficult for small objects to reduce false detections that display great variation in form. To address this issue, we use soft attention gates (AGs) to define and propagate relevant spatial information from low level feature maps to the decoding stage.

2. Network Architecture

Fully convolutional U Net with residual connections

One important change in our architecture is that we now have a large number of feature channels in the upsampling part which allows the network to propagate context information to higher resolution layers. As a result, the expansive path to the contracting path is more or less symmetrical, and yields a u-shaped architecture. The network has no fully connected layers and uses only the relevant part of each convolution, i.e. the segmentation map includes only the pixels, for which the entire context is available in the input image. This technique allows arbitrarily large images to be smoothly segmented by an overlap-tile strategy. The missing context is extrapolated by mirroring the input image to determine the pixels within the image's border region.



Process of Identifying Non Cancer Region

This tiling strategy is critical for applying the network to large images, as otherwise the GPU memory would limit the resolution. As for our activities, there are very few training data available, we are using excessive data augmentation by adding elastic deformations to the training images available. This helps the network to learn invariance to these deformations, without having such transformations in the annotated corpus of images. This is especially important in biomedical segmentation, since deformation used to be the most common tissue variation and realistic deformations can be effectively simulated. The isolation of touching objects of the same class is another problem in several cell segmentation tasks. To this end, we suggest the use of a weighted loss, where the separating context labels between touching cells obtain a great weight in the role of loss. The resulting network contributes to different problems around biomedical segmentation. The architecture of the network is shown in Figure 1 It consists of a contracting path (left side) and an open path (right side). Typical architecture of a convolutionary network follows the contracting direction. It consists of repeated application of two 3x3 convolutions (unpadded convolutions), each preceded by a rectified linear unit (ReLU) and a downsampling process of 2x2 max pooling with phase 2.At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is required in any convolution, due to the loss of boundary pixels. A 1x1 convolution is used at the final layer to map each of the 64 component feature vectors to the number of classes desired. The network has 23 convolutionary layers all in all. To allow the output segmentation map to be tilted smoothly, it is necessary to select the size of the input tile so that all 2x2 max-pooling operations are applied to a layer with x- and y-size equal.

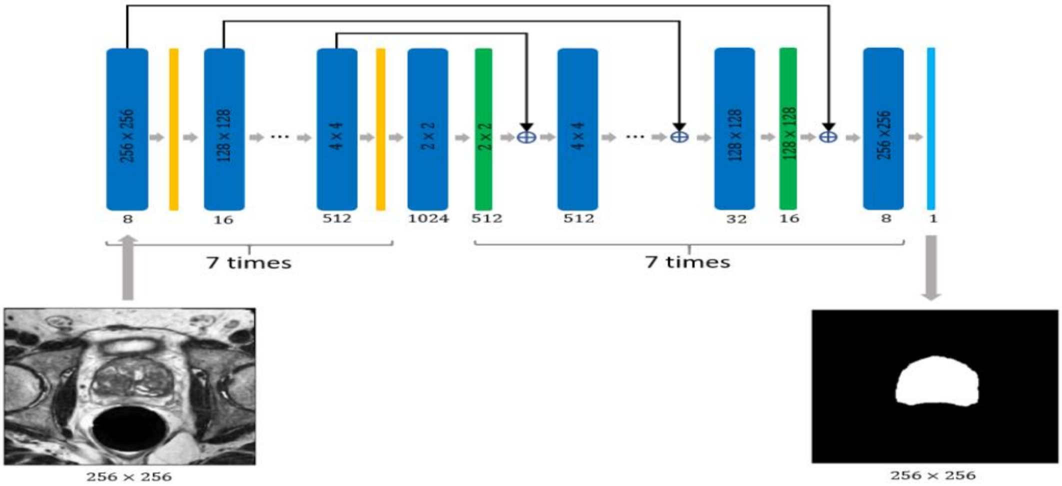


Fig 1 : Fully Convolutional Neural Network with Residuals

Combined with the cross entropy loss function, the energy function is determined by a pixel-wise soft-max over the final feature diagram.The soft-max is defined as

$$E = \sum_{\mathbf{x}\in\Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x}))$$

$\vee$

where ak(x) denotes the activation in feature channel k at the pixel position x belongs to Gamma. K is the number of classes and pk(x) is the approximated maximum-function. I.e. pk(x) _ 1 for the k that has the maximum activation ak(x) and pk(x) _ 0 for all other k.

HyperDense-Net

The concept of “the deeper the better” is considered as a key principle in deep learning. Nevertheless, one obstacle when dealing with deep architectures is the problem of vanishing or exploding gradients, which hampers convergence during training. To address these limitations in very deep architectures, we investigated densely connected networks. DenseNets are built on the idea that adding direct connections from any layer to all the subsequent layers in a feed-forward manner makes training easier and more accurate. This is motivated by three observations. First, there is an implicit deep supervision thanks to the short paths to all feature maps in the architecture. Second, direct connections between all layers help improving the flow of information and gradients throughout the entire network. Third, dense connections have a regularizing effect, which reduces the risk of over-fitting on tasks with smaller training sets.

Inspired by the recent success of densely-connected networks in medical image segmentation works, we propose a hyper-dense architecture for multi-modal image segmentation that extends the concept of dense connectivity to the multi-modal setting: each imaging modality has a path, and dense connections occur not only between layers within the same path, but also between layers across different paths. HyperDenseNet introduces a more general concept of connectivity, in which we link outputs from layers into different streams, each associated with a different modality of image. Our hyper-dense networking in the multi-modal setting provides a far more efficient feature representation than early / late fusion as the network learns the dynamic relationships between the modalities inside and between all the abstraction levels. Let us consider the scenario of two image modalities for simplicity, 4 while extension to N modalities is possible. Let x1l and x2l denote the lth layer outputs for streams 1 and 2, respectively. In general, the output of the lth layer in a stream s can then be defined as follows:

$$\mathrm{x}_l^s = H_l^s([\mathrm{x}_{l-1}^1, \mathrm{x}_{l-1}^2, \mathrm{x}_{l-2}^1, \mathrm{x}_{l-2}^2, \ldots, \mathrm{x}_0^1, \mathrm{x}_0^2]).$$

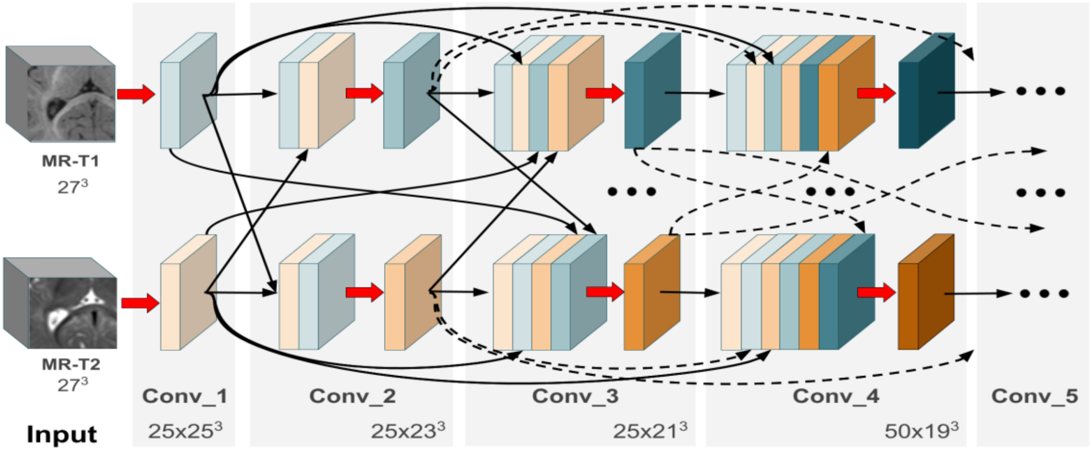


Figure 2: Hyper Dense Net Architecture

Shuffling and interleaving feature map elements in a CNN was recently found to enhance the efficiency and performance, while serving as a strong regularizer. This is motivated by the fact that intermediate CNN layers perform deterministic transformations to improve the performance. However, relevant information might be lost during these operations. To overcome this issue, it is therefore beneficial for intermediate layers to offer a variety of information exchange while preserving the aforementioned deterministic functions. Motivated by this principle, we thus concatenate feature maps in a different order for each branch and layer.

Figure 2 shows a section of the proposed architecture, where each gray region represents a convolutional block. For simplicity, we assume that the red arrows indicate convolution operations only, whereas the black arrows represent the direct connections between feature maps from different layers, within and in-between the different streams. Thus, the input of each convolutional block (maps before the red arrow) is the concatenation of the outputs (maps after the red arrow) of all the preceding layers from both paths. FCNNs usually use complete images as input for providing a wide receptive area. Then, the number of parameters is restricted by pooling / unpooling layers. One problem with this method is the lack of resolution resulting from repeated down-sampling. We follow the strategy in the proposed process, where sub-volumes are used as inputs, thus avoiding pooling layers. Although sub-volumes of size 27 27 27 are considered for preparation, during inference we used 35 35 35 non-overlapping sub-volumes, as in [5],[26]. This technique has two major advantages to it. Firstly, it reduces our network's memory requirements and thus removes the need for spatial pooling. More significantly, it reduces the number of training examples considerably, and thus does not require data increase.
Using cross-entropy as cost function, the network parameters are optimized through the RMSprop optimiser. Let θ denotes the network parameters (i.e., convolution weights, biases and ai from the parametric rectifier units), and yvs the label of voxel v in the s-th image segment. We optimize the following:

$$J(\boldsymbol{\theta}) = -\frac{1}{S \cdot V} \sum_{s=1}^{S} \sum_{v=1}^{V} \sum_{c=1}^{C} \delta(y_s^v = c) \cdot \log p_c^v(\mathbf{x}_s),$$

To initialize the network's weights, we adopted a strategy that gives rapid convergence for very deep architectures. A zero-mean Gausian standard deviation distribution is used in this technique to initialize the weights in layer l.

4. Training

Fully Convoluted Neural Network with Residuals

The input images and their corresponding segmentation maps are used to train the network with the implementation of Caffe's stochastic gradient descent. The output image is smaller than the input by a constant border distance, due to the unpadded convolutions. In order to minimize the overhead and optimize the use of the GPU memory, we prefer large input tiles over a large batch size and thus reduce the batch to one image. Accordingly, we use a high momentum (0.99) that defines the change in the current optimization phase by a large number of the previously seen training samples. A good initialization of the weights is extremely important in deep networks with several convolutionary layers and different paths through the network. Otherwise, network sections can offer excessive activations, while other sections will never contribute. Ideally the initial weights will be modified such that each function map in the network has a variance of approximately the unit.For a network with our architecture (alternating convolution and ReLU layers) this can be achieved by drawing the initial weights from a Gaussian distribution with a standard deviation of $\sqrt{2/N}$, where N denotes the number of incoming nodes of one neuron.

Hyper DenseNet

To initialize the network's weights, we followed a strategy that gives rapid convergence for very deep architectures. In this technique, a Gaussian zero-mean standard deviation distribution is used to initialize the weights in layer l, where nl denotes the number of connections within that layer to the units. The momentum was set at 0.6 and the initial learning rate at 0.001, being that after every 5 epochs by a factor of 2 (starting from epoch 10). Trained the network for 30 epochs, each consisting of 20 subepochs. A total of 1000 samples were randomly selected from the training images at each subepoch, and processed in size 5 batches.

NAS UNet

We use the SGD optimizer with momentum 0.95, cosine learning rate decaying from 0.025 to 0.01, and weight decay 0.0003 [20] while learning network weight w. We use Adam optimizer [51] with learning rate 0.0003 and weight decay 0.0001 while learning the architecture alpha. We find empirically that when we optimize alpha after a constant epoch (such as 50), the mean intersection over Union (mIoU) and the pixels accuracy (pixAcc) increase gradually. So at the beginning, we optimize the alpha

A NOVEL FOCAL TVERSKY LOSS FUNCTION WITH IMPROVED ATTENTION U-NET

The model with a batch size of 16 was conditioned for 5 epochs. Both models have been optimized using stochastic downward gradient with momentum, using an initial learning rate of 0.01 which decays by 10 to power -6 at each epoch. Via a grid search method these parameters were optimised. But the dice coefficient was around.804 given this.

5. Conclusions

Out of the 4 different approaches tried, below is what we are able to infer. The approach using Naas U-Net was promising in theory and it seemed to show good results on paper. But when implemented, despite doing hyper parameter tuning the DSC was around 0.68. The Attention U Net was able to produce the same result as shown in the paper (0.084) but tuning did not help much here with results around (0.812). The Fully convolutional neural network with residual connections is the best of the approaches which showed excellent results of DSC 0.881 after hyper parameter tuning against the paper claim of 0.8312. Hyperdense-Net is a new approach and seemed to be an effective one on paper atleast. But dye to the performance intensive operations involved in it, we were not able to complete a single run.

References

1. Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural net-
2. works segment neuronal membranes in electron microscopy images. In: NIPS. pp.
3. 2852{2860 (2012)
4. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative un-

5. supervised feature learning with convolutional neural networks. In: NIPS (2014)
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for ac-
7. curate object detection and semantic segmentation. In: Proceedings of the IEEE
8. Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
9. Hariharan, B., Arbelez, P., Girshick, R., Malik, J.: Hypercolumns for object seg-
10. mentation and _ne-grained localization (2014), arXiv:1411.5752 [cs.CV]
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into recti_ers: Surpassing human-
12. level performance on imagenet classi_cation (2015), arXiv:1502.01852 [cs.CV]
13. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadar-
14. rama, S., Darrell, T.: Ca_e: Convolutional architecture for fast feature embedding
15. (2014), arXiv:1408.5093 [cs.CV]
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classi_cation with deep con-
17. volutional neural networks. In: NIPS. pp. 1106{1114 (2012)
18. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.,
19. Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural
20. Computation 1(4), 541{551 (1989)
21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic
22. segmentation (2014), arXiv:1411.4038 [cs.CV]
23. [2] X. Llad´ o, A. Oliver, M. Cabezas, J. Freixenet, J. C. Vilanova,
    A. Quiles, L. Valls, L. Ramio´ -Torrenta`, and A` . Rovira, “Segmentation
24. of multiple sclerosis lesions in brain MRI: a review of
25. automated approaches,” Information Sciences, vol. 186, no. 1, pp.
26. 164–185, 2012.
27. [3] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani,
28. J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest et al., “The multimodal
29. brain tumor image segmentation benchmark (BRATS),”
30. IEEE Transactions on Medical Imaging, vol. 34, no. 10, pp. 1993–2024,
31. 2015.
32. [4] M. Havaei, N. Guizard, N. Chapados, and Y. Bengio, “HeMIS:
33. Hetero-modal image segmentation,” in International Conference on
34. MICCAI. Springer, 2016, pp. 469–477.
35. [5] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D.
36. Kane, D. K. Menon, D. Rueckert, and B. Glocker, “Efficient multiscale
37. 3D CNN with fully connected CRF for accurate brain lesion
38. segmentation,” Medical image analysis, vol. 36, pp. 61–78, 2017.

39. [6] L. Fidon, W. Li, L. C. Garcia-Peraza-Herrera, J. Ekanayake,
40. N. Kitchen, S. Ourselin, and T. Vercauteren, “Scalable multimodal
41. convolutional networks for brain tumour segmentation,” in International
42. Conference on MICCAI. Springer, 2017, pp. 285–293.
43. [7] M. Prastawa, J. H. Gilmore, W. Lin, and G. Gerig, “Automatic
44. segmentation of MR images of the developing newborn brain,”
45. Medical image analysis, vol. 9, no. 5, pp. 457–466, 2005.
46. [8] N. I. Weisenfeld, A. Mewes, and S. K. Warfield, “Segmentation of
47. newborn brain MRI,” in Biomedical Imaging: Nano to Macro, 2006.
48. 3rd IEEE International Symposium on. IEEE, 2006, pp. 766–769.
49. [9] P. Anbeek, K. L. Vincken, F. Groenendaal, A. Koeman, M. J.
50. Van Osch, and J. Van der Grond, “Probabilistic brain tissue
51. segmentation in neonatal magnetic resonance imaging,” Pediatric
52. research, vol. 63, no. 2, pp. 158–163, 2008.
53. [10] N. I. Weisenfeld and S. K. Warfield, “Automatic segmentation of
54. newborn brain MRI,” Neuroimage, vol. 47, no. 2, pp. 564–572, 2009.
55. [11] L. Wang, F. Shi, W. Lin, J. H. Gilmore, and D. Shen, “Automatic
56. segmentation of neonatal images using convex optimization and
57. coupled level sets,” NeuroImage, vol. 58, no. 3, pp. 805–817, 2011.
58. [12] V. Srhoj-Egekher, M. Benders, K. J. Kersbergen, M. A. Viergever,
59. and I. Isgum, “Automatic segmentation of neonatal brain MRI
60. using atlas based segmentation and machine learning approach,”
61. MICCAI Grand Challenge: Neonatal Brain Segmentation, vol. 2012,
62. 2012.
63. [13] S. Wang, M. Kuklisova-Murgasova, and J. A. Schnabel, “An atlasbased
64. method for neonatal MR brain tissue segmentation,” Proceedings
65. of the MICCAI Grand Challenge: Neonatal Brain Segmentation,
66. pp. 28–35, 2012.
67. [14] L. Wang, F. Shi, G. Li, Y. Gao, W. Lin, J. H. Gilmore, and D. Shen,
68. “Segmentation of neonatal brain MR images using patch-driven
69. level sets,” NeuroImage, vol. 84, pp. 141–158, 2014.
70. [15] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen,
71. “Deep convolutional neural networks for multi-modality isointense
72. infant brain image segmentation,” NeuroImage, vol. 108, pp.
73. 214–224, 2015.

74. [16] D. Nie, L. Wang, Y. Gao, and D. Sken, “Fully convolutional
75. networks for multi-modality isointense infant brain image segmentation,”
76. in 13th International Symposium on Biomedical Imaging
77. (ISBI), 2016. IEEE, 2016, pp. 1342–1345.
78. [17] J. Dolz, C. Desrosiers, L. Wang, J. Yuan, D. Shen, and I. Ben Ayed,
79. “Deep CNN ensembles and suggestive annotations for infant
80. brain MRI segmentation,” arXiv preprint arXiv:1712.05319, 2017.
81. [18] A. M. Mendrik, K. L. Vincken, H. J. Kuijf, M. Breeuwer, W. H.
82. Bouvy, J. De Bresser, A. Alansary, M. De Bruijne, A. Carass, A. El-
83. Baz et al., “MRBrainS challenge: online evaluation framework
84. for brain image segmentation in 3T MRI scans,” Computational
85. intelligence and neuroscience, vol. 2015, p. 1, 2015.
86. [19] H. Chen, Q. Dou, L. Yu, J. Qin, and P.-A. Heng, “VoxResNet: Deep
87. voxelwise residual networks for brain segmentation from 3D MR
88. images,” NeuroImage, 2017.
89. [20] S. C. Deoni, B. K. Rutt, A. G. Parrent, and T. M. Peters, “Segmentation
90. of thalamic nuclei using a modified k-means clustering
91. algorithm and high-resolution quantitative magnetic resonance
92. imaging at 1.5T,” Neuroimage, vol. 34, no. 1, pp. 117–126, 2007.
93. [21] O. Commowick, F. Cervenansky, and R. Ameli, “MSSEG Challenge
94. proceedings: Multiple Sclerosis Lesions Segmentation Challenge
95. using a data management and processing infrastructure,” in
96. MICCAI, 2016
97. [1] O. C. Eidheim, L. Aurdal, T. Omholt-Jensen, T. Mala, and B. Edwin,
98. ``Segmentation of liver vessels as seen in MR and CT images,'' Int.
99. Congr. Ser., vol. 1268, pp. 201_206, Jun. 2004. [Online]. Available:
100. http://www.sciencedirect.com/science/article/pii/S0531513104006132.
101. doi: 10.1016/j.ics.2004.03.184.
102. [2] A. Bert et al., ``An automatic method for colon segmentation in
103. CT colonography,'' Computerized Med. Imag. Graph., vol. 33, no. 4,
104. pp. 325_331, Jun. 2009. [Online]. Available: http://www.sciencedirect.
105. com/science/article/pii/S0895611109000226
106. [3] T. Klinder, J. Ostermann, M. Ehm, A. Franz, R. Kneser, and C. Lorenz,
107. ``Automated model-based vertebra detection, identi_cation, and segmentation

108. in CT images,'' Med. Image Anal., vol. 13, no. 3, pp. 471_482,
109. Jun. 2009. [Online]. Available: http://www.sciencedirect.com/science/
110. article/pii/S1361841509000085
111. [4] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, ``Brain intelligence:
112. Go beyond arti_cial intelligence,'' Mobile Netw. Appl., vol. 23, no. 2,
113. pp. 368_375, Apr. 2018. doi: 10.1007/s11036-017-0932-8.
114. [5] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, ``Deep features
115. learning for medical image analysis with convolutional autoencoder neural
116. network,'' IEEE Trans. Big Data, to be published.
117. [6] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, ``Health-
118. CPS: Healthcare cyber-physical system assisted by cloud and big data,''
119. IEEE Syst. J., vol. 11, no. 1, pp. 88_95, Mar. 2017.
120. [7] J. Long, E. Shelhamer, and T. Darrell, ``Fully convolutional networks
121. for semantic segmentation,'' in Proc. IEEE Conf. Comput. Vis. Pattern
122. Recognit. (CVPR), Jun. 2015, pp. 3431_3440.
123. [8] O. Ronneberger, P. Fischer, and T. Brox, ``U-net: Convolutional networks
124. for biomedical image segmentation,'' in Proc. Int. Conf. Med. Image
125. Comput. Comput.-Assist. Intervent. (MICCAI), Nov. 2015, pp. 234_241.
126. [9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ``Gradient-based learning
127. applied to document recognition,'' Proc. IEEE, vol. 86, no. 11,
128. pp. 2278_2324, Nov. 1998.
129. [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ``Imagenet classi_cation
130. with deep convolutional neural networks,'' Commun. ACM, vol. 60, no. 6,
131. pp. 84_90, Jun. 2012.
132. [11] K. Simonyan and A. Zisserman, ``Very deep convolutional networks
133. for large-scale image recognition,'' in Proc. Int. Conf. Learn. Repre-
134. sent. (ICLR), Apr. 2015, pp. 1_14.
135. [12] C. Szegedy et al., ``Going deeper with convolutions,'' in Proc. IEEE Conf.
136. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1_9.
137. [13] K. He, X. Zhang, S. Ren, and J. Sun, ``Deep residual learning for image
138. recognition,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR),
139. Jun. 2016, pp. 770_778.
140. [14] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, ``Densely
141. connected convolutional networks,'' in Proc. IEEE Conf. Comput. Vis.
142. Pattern Recognit. (CVPR), Jul. 2017, pp. 2261_2269.