Sciatic Nerve Segmentation in MR Images of the Upper Leg via Convolutional Neural Networks

Shashank Manjunath[1], Richard D. Dortch[1,2]

[1]Vanderbilt University Institute of Imaging Science, Vanderbilt University, Nashville, TN, United States, [2]Radiology and Radiological Sciences, Vanderbilt University, Nashville, TN, United States

Introduction: Quantitative MRI of the sciatic nerve provides novel information that reports on disability in patients with inherited neuropathies[3]. Unfortunately, segmentation of the proximal sciatic nerve from MRI images requires researchers to spend significant amounts of time manually labeling the region of interest (ROI). Furthermore, raters must be extensively trained to minimize inter-rater variability. New technologies, such as convolutional neural networks (CNNs), may allow one to automate and standardize this segmentation process. CNNs operate by i) training a network of convolution kernels on an expertly labeled dataset[1] and ii) applying the trained CNNs on other images to identify certain features. Although CNNs are traditionally used for image recognition applications[2], they have recently been used for image segmentation[1]. Here we investigated the feasibility of applying CNNs to segment MRI images of the proximal sciatic nerve in patients with inherited neuropathies and matched controls.

Materials and Methods: The sciatic nerve was manually labeled using 3D proton-density MRI datasets (256x256x40 matrix) from 72 subjects (34 inherited neuropathy patients, 38 matched controls)[3]. Datasets were randomly split into training and validation sets, with 80% of subjects (N=58) being used to train, and 20% (N=14) being used to test. For training, each subject's data was rearranged into a 256x1280x8 matrix, or a 2D montage of 5 contiguous slices (Fig. 1, top panel). The number of slices was chosen based upon available memory and training time. This approach preserved information about how the ROI moves through the slice, while allowing the use of 2D convolutions in the network. The resulting 2D montages were then fed into the "U-Net"[4], which was optimized via stochastic gradient descent with a learning rate/momentum of 0.0001/0.99. This network was designed in Python 2.7, using Keras[5] for CNN and Theano[6,7] as the backend. Training took ≈4 days using a NVidia Titan X GPU. After training, the returned ROIs were defined as voxels nonzero values. CNN segmentation performance was evaluated via the Dice Coefficient[8], which quantifies the similarity between the manual and CNN derived ROIs and ranges from 0-1 (1 indicates a perfect CNN segmentation relative to the manual method).

Results and Discussion: The algorithm failed to produce a segmentation, defined as producing a dice coefficient of less than 0.05, in 20% of cases. However, for the 80% of cases where a segmentation was produced, the average dice coefficient was 0.69, and the variance was 0.04. Segmenting the 14 subjects' data, which consisted of 120 individual images which had not been seen by the net before, took roughly 90 seconds when run on a Titan X GPU. An example of a segmentation with a dice coefficient of 0.79 is shown below (Fig. 1, bottom right panel).

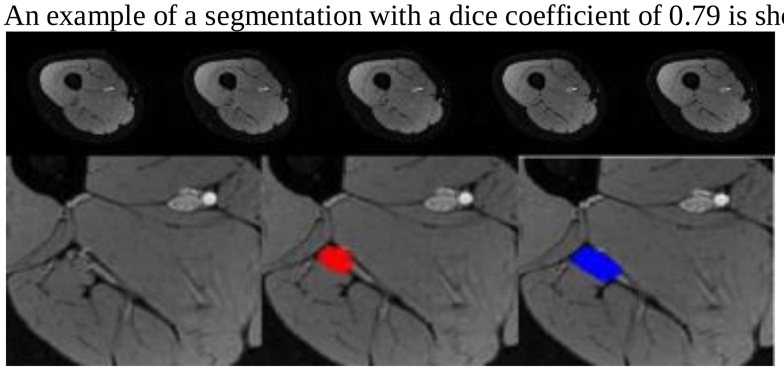

Figure 1. (Top). Subset of images stitched together into a 2D montage. (Bottom Left). Example of a zoomed region of MR image without ROI overlay. (Bottom Middle). Same MR image with true, user-defined ROI overlay (red). (Bottom Right). Same MR image with predicted ROI (blue).

Conclusions: We demonstrated that CNNs can be applied to medical image segmentation applications. Additionally, we proposed a novel method of conserving 3D ROI information when using 2D convolutions in a CNN, which improved computation efficiency and preserved information on edge slices that are lost with 3D convolutions. Future work includes i) implementing random image transformation to augment the training dataset fed into the CNN and ii) investigating methods to include information from all 40 slices simultaneously when during training (e.g., 3D CNNs or cropping slices around the sciatic nerve to reduce memory requirements).

Acknowledgements: NIH/NINDS R01 NS97821 and NIH/NIBIB K25 EB13659 for funding.

References: [1]. Long J. *arXiv:1605.06211*. 2016. [2] Krizhevsky A. *Adv Neur Inf Proc Sys*, 1097-1105. 2012 [3] Dortch RD. *Neurology*, 83, 1545–1553. 2014. [4] Ronneberger O. *arXiv: 1505.04597v1*. 2015. [5] Chollet F. *Keras, GitHub repository*. *2015*. [6] Bastien, F. "Theano: New Features and Speed Improvements." *NIPS Deep Learning Workshop*. 2012. [7] Bergstra, J. "Theano: A CPU and GPU Math Expression Compiler." *Proc Python Sci Comp Conf*. 2010. [8] Zijdenbos AP. *IEEE Trans Med Imag*, 13, 716-724. 1994.