Automatic Registration and Segmentation of Magnetic Resonance Images Using Deep Neural Networks

Rajeswari Sivakumar
Institute for Artificial Intelligence
University of Georgia
Athens, GA, 30605
rs12419@uga.edu

Maulik N. Shah
Institute for Artificial Intelligence
University of Georgia
Athens, GA, 30605
mns28652@uga.edu

Abstract

The present paper examines the efficacy of different deep learning techniques for registering and segmenting brain magnetic resonances images (MRI) from 100 different subjects. The goal is to develop separate networks that can be used to complete the tasks of registration and segmentation.This will ultimately help researchers better quantify differences between health and pathogenic brains as well as better understand early disease markers in brain disorders. To this end our research implements two different networks, a version of fully convolutional neural networks for registration, and the V-net architecture for segmentation. While we were not able to train the V-net architecture due to memory limitations, we were able to succesfully train 1000 epochs of the registration network. Due to the limited success of both networks, we believe the that this work requires further development that will be discussed below.

1 Introduction

1.1 Automatic Processing

In recent years advances in computing have allowed for implementation of increasingly of increasingly advanced algorithms for the processing and classification of biomedical images. In particular there have been great strides in the modeling of brain images.

Classification and qualification of severity of diseased brains has seen significant improvement. Much of this improvement has been seen in tumor segmentation where clear differences can be made between cancerours and healthy cells as well as between subtypes of pathogenic cells.

This however becomes more tricky in chronic illness that result in degeneration of cells, but no obvious changes in brain structure, such as clusters of pathogenic cells. In particular, illnesses like Parkinson's disease and Alzheimer's show only slight changes in relative volume of cortical structures. More obviously differentiated features like protein build up or pathogenic tissue are not visible until until the illness has progressed significantly. At this point, diagnosis from brain imaging is a moot point and algorithmic approaches are redundant. Thus to gain a more nuanced understanding of the effects of these diseases on the brain as well as to aid in earlier diagnosis, we must examine registration and segmentation of brain regions. Both of these techniques are important steps in processing and quantifying variance in brain images between individuals.

Preprint. Work in progress.

1.2 Registration

Registration traditionally involves the alignment of the brain images to a statistical atlas. Atlases are built from several brain images collected from subjects of different brain images collected from subjects of different ages, genders and races. This stratified sampling approach is used to ensure that the atlas balanced in it's representation of each group. These collected brain images, typically magnetic resonances images (MRI) are combined to form a matrix of distributions, corresponding to voxel values. The distribution matrix (atlas) is then used as a template to adjust the relative position of voxels in a target image. This transformation is represented as an affine matrix transformation. (Mazziotta et al., 2001)

More recently researchers have experimented with a variety of approaches to improve the speed of registration while maintaining accuracy of segmentation and identification of key landmarks in the brain images, such as the Quicksilver network (Yang et al., 2017). This approach was interesting because it uses the moving image(unprocessed) and fixed(registered version of moving image) to train a network. The Quicksilver algorithm iterates over patches of the moving and fixed image, and concatenates their convolved features to then learn a final transformation of each patch. While this approach was novel, and the tandem approach to training the algorithm was interesting, we wanted to build a network that would learn spatial relationships of the voxels over the whole brain. Thus we turned out sights to Li and Fan (2018). The fully convolutional network they implement mimics U-net architecture developed by Ronneberger et al. (2015). Their network however followed a significantly scaled down approach illustrated in Figure 1.

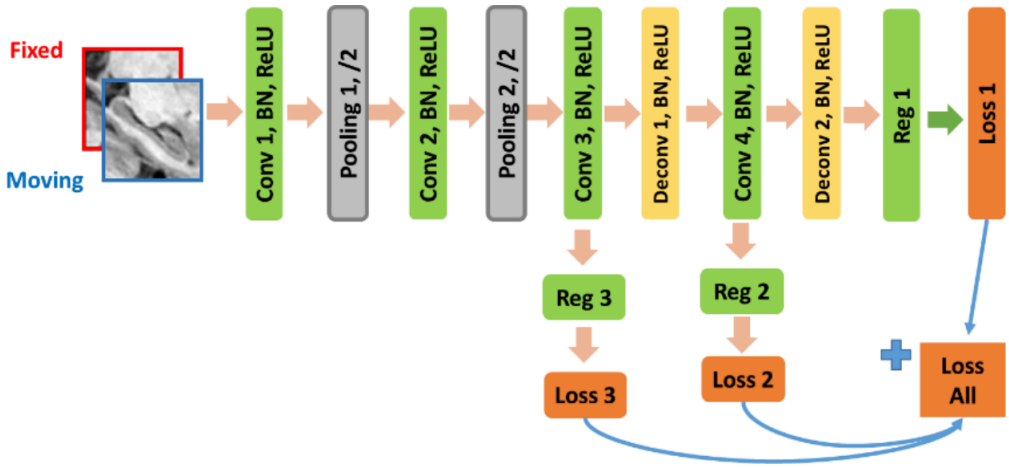

Figure 1: Architecture implemented by Li and Fan (2018). We zero-padded our image arrays to the max dimensions across all images. Due to limitations, we did not include as many kernel layers in the hidden layers of the network in our implmentation.

1.3 Segmentation

Segmentation of medical images for tracking certain substances in human organs is a very useful application to diagnose some diseases like Parkinsons disease. Convolutional neural networks are widespread in practice for solving such problems due to their high accuracy in finding the patterns in the 2-D as well as 3-D images. To perform segmentation tasks on 3-D images using convolutional neural networks (CNN), networks are generally trained on 2-D patches of 3-D images (U-Net, VGG, etc.). Although they yield some reasonable results, they were unable to learn the spatial relationships across all dimensions effectively.

The V-Net is designed to find volumetric patterns using 3-D images as training data, which was originally applied to segment prostate MRI image volumes. (Milletari et al., 2016) In this paper, we are trying to use this network to train a set of fMRI images by dividing them into 3-D patches and extracting the patterns to predict the Parkinsons disease as a final goal.

The V-Net has a V shaped architecture, in which the left portion of the network reduces the resolution of the input image using convolution and appropriate stride at each stage, while the right part of the network increases the resolution of the input by deconvolving the combination of the input and the extracted features from the left stage of the network at the same height. The use of extracted features from the left-portion stages in the right-portion of the network allows retaining features learned at the early left-stages of the network, which would otherwise be forgotten. Intuitively, it would allow the network to learn the patterns in the image more accurately and would reduce the overall learning loss. A very important structural advantage is that the inner part of this CNN captures the content of the whole volume because the computation at each layer has the much larger special support of the images than generally being sought to find the best results.

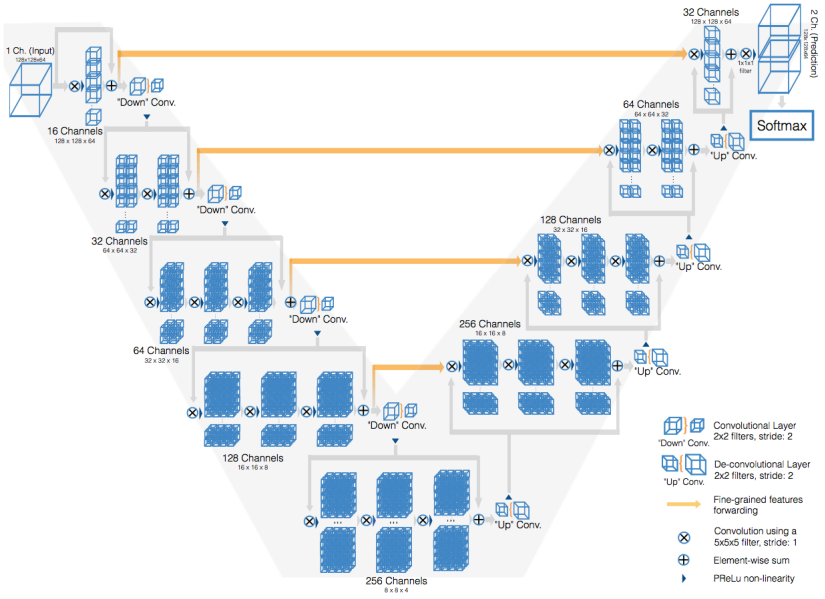

Figure 2: Architecture implemented by Milletari, Navab, and Ahmadi (2016).

As stated in the original paper, each stage consists of 3 convolution layers is designed to learn the residual function by using the input to it (1) into all the convolution layers to learn non-linearities and (2) adding the input to the last convolution layer to enable learning the residual function. The kernels used at each stage in the convolution layers are of the size 5 x 5 x 5 voxels, and the down sampling is done using a 2 x 2 x 2 voxel kernel with a stride value of 2, which would reduce the sample size by half. The same idea is used for increasing the input size to double using the deconvolution on the right portion of the architecture. Replacement of the pooling operation with the convolution structure gives a great memory advantage by reducing mapping components from output to input in the back-propagation process.

The final layer of the Softmax function is being fed with just two sets of feature maps produced by applying 1 x 1 x 1 voxel kernel over the input of the actual patch size. The Softmax function segments the image into foreground and background regions by computing probability of each voxel. One more innovation in the V-Net is the Dice Loss function, which tries to calculate loss without applying any sort of weighting to the foreground or the background regions. The function and its derivation are given as Equations 1 and 2 respectively.

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N {p_i}^2 + \sum_i^N {g_i}^2} \tag{1}$$

$$\frac{\partial D}{\partial p_j} = 2\left[\frac{g_j\left(\sum_i^N {p_i}^2 + \sum_i^N {g_i}^2\right) - 2p_j\left(\sum_i^N p_i g_i\right)}{\left(\sum_i^N {p_i}^2 + \sum_i^N {g_i}^2\right)}\right] \tag{2}$$

2 Methods

2.1 Data

We used the Mindboggle-101, manually labeled MRI images (Klein and Tourville, 2012). From these we dropped one subject due to labels and used the remaining 100 to train our registration and segmentation network. For the registration network, we used the skull-stripped registered and unregistered brain images as targets and training instances respectively. For the segmentation network, we used the skull-stripped brains and respective Desikan-Killany-Tourville(DKT) 31 labels. This schema developed by the authors essentially maps the brain in three dimensions to 31 labels corresponding to various coritical regions. The DKT protocol is a variation of the Desikan-Killany (DK) protocol (Desikan et al., 2006) adapted to be more precise and unambiguous in the labels produced.

Since the data were of irregular sizes there were all padded with zeros to meet uniform dimensions of 256 x 256 x 256. This was done as a necessity for training in both convolutional neural network architectures.

2.2 Registration Network: Fully Convolutional Network (FCN)

The zero-padded images were fed through a variation of the network described by **?**. We maintained the layers of the architecture and the weights of each loss (1, 2, and 3) as (0.3, 0.6, and 1) respectively. We trained our network for 1000 epochs. In our network we used zero-padding to maintain size after the convoultional layers. Convolutions 1 and 2 had kernel size 6 and stride 2 while convolutions 3 had kernel sizes 3 and strides 2. Convolution 4 had kernel size and stride of 1. The max-pooling layers both had kernel size and stride of 2. The deconvolutional layers both had kernel sizes of 6 and strides of 2. We reached this configuration after some iterative testing to see which kernel dimensions would allow the network to closely resemble the same relative proportions as the original literature.

Initially we had intended to test our network on Google Cloud Compute Engine, but we were unable to do so due to technical difficulties accessing our data in Storage Buckets. Consequently we ran our network only on local machines.

2.3 Segmentation Network: V-net

The preprocessed Mindboggle dataset images of 256 x 256 x 256 size are chunked into 16 patches of 128 x 128 x 64 size, which is the standard input for the V-Net according to the original paper. The implementation was done in PyTorch, with some variation than the actual V-Net architecture (Milletari et al., 2016), solely to fit the network in the given infrastructure.

Initially, the V-Net was fed with the batch of 16 patches for training, which is indeed a 3-D image in the form of volumetric patches, but due to huge memory overload the idea didnt work as the number of parameters in the network exceeded 20 million and the network crashed due to insufficient memory. We then tried to reduce the batch size to just one patch in the following experiments to optimize the memory overload, which also failed for the same reason. To run the network with the available infrastructure, we also tried to reduce the number of stages and layers in each stage. This still failed to yield improvement.

3 Results

3.1 Registration

Upon scoring outputs from our network, we found an average KL divergence score of -71.67. Given that these values are negative, we believe that our scoring function may need to be amended. Additionally, there was significant variance among the scores of outputs, suggesting that our model has not converged. We therefore believe that with additional training, we might be able to obtain better results.

3.2 Segmentation

Our tracking of the issue inferred that the crash occurred at the final up layers of the network in almost all experiments, even in case of different infrastructure memory sizes. Though it is not natural to happen in the given architecture, the input size at the crashing layers was in accordance with expectations. After researching forums for the similar issues in PyTorch, it gives a sense of a bug in the library, which is not yet resolved. However can not be sure of the exact point of error. Although the implementation couldnt give the results by the time of the deadline, we intend to update some configuration to find the solution using the given infrastructure.

References

R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3): 968–980, 2006.

A. Klein and J. Tourville. 101 labeled brain images and a consistent human cortical labeling protocol. *Frontiers in neuroscience*, 6:171, 2012.

H. Li and Y. Fan. Non-rigid image registration using self-supervised fully convolutional networks without training data. In *International Symposium on Biomedical Imaging 2018 (ISBI'18)*. IEEE, 2018.

J. Mazziotta, A. Toga, A. Evans, P. Fox, J. Lancaster, K. Zilles, R. Woods, T. Paus, G. Simpson, B. Pike, et al. A probabilistic atlas and reference system for the human brain: International consortium for brain mapping (icbm). *Philosophical Transactions of the Royal Society B: Biological Sciences*, 356(1412):1293–1322, 2001.

F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

X. Yang, R. Kwitt, M. Styner, and M. Niethammer. Quicksilver: Fast predictive image registration–a deep learning approach. *NeuroImage*, 158:378–396, 2017.