Convolutional Neural Networks for Synovitis Segmentation in Rheumatoid Arthritis

Qitao Li, Xiaojuan Li, Valentina Pedoia

Abstract. Quantifying synovitis legions in Magnetic Resonance (MR) images of patients with Rheumatoid Arthritis (RA) is an important tool for diagnosis and treatment of RA. Manual segmentation of lesions is costly due to both the complex positioning of the lesions and the size of the MR volumes. In this work, we present an automatic segmentation method using a fully-convolutional Convolutional Neural Network (CNN) with an encoder-decoder architecture. The network is trained end-to-end from 51 $512 \times 512 \times 20$ MR volumes and outputs a voxel-wise segmentation of the input volume. Furthermore, we train with a loss function based on Dice coefficient to deal with class-imbalance. We find that the network performs with an average dice coefficient of 0.61 on a test set. The implementation and code are available at https://github.com/cosmicac/ucsf-mri-seg.

1 Introduction

Rheumatoid arthritis (RA) is a chronic inflammatory disease affecting 1.3 million Americans that causes destruction of articular and periarticular structures, of which wrist joints may be among the first affected [1]. Early diagnosis and monitoring of abnormalities within joint structures are important for successful therapy.

Magnetic Resonance (MR) images of patients' wrists can reveal synovitis lesions. The Outcome Measures in Rheumatology Clinical Trials (OMERACT) has developed a RA MRI Scoring (RAMRIS) system to assess these lesions in a reliable and reproducible manner. However, because RAMRIS does not use the direct volumes of the lesions but instead uses a discrete score to measure the severity of lesions, it is a semi-quantitative system that can result in considerable inter-reader variability [2]. Using the true volume of lesions would offer a more objective measurement of disease progression.

Manual delineation of lesions is a time-consuming and tricky task that requires a skilled operator. Comparison between multiple MR sequences is often neccessary. Moreover, an operator must take into consideration the three spatial dimensions of the MR volume while working through the volume one 2D slice at a time. Semi-automatic methods that combine rough manual outlining with thresholding can help speed up the process, but the choice of thresholding level can introduce biases and inaccuracies. A fuly-automatic, fast segmentation method that utilizes the information of 3 spatial dimensions of the MR volumes to segment synovitis lesions would be desirable.

A method for automatic segmentation of synovitis in the wrist was proposed by Czaplicka et al., but it relies on complex, less generalizable wrist-bone segmentation and thresholding [3]. In recent years, convolutional neural networks (CNNs) have been used in image classification and segmentation contexts to great success due to their ability to extract and learn heirarchical features. CNNs first achieved acclaim in their abilities to classify whole images as demonstrated by Krizhevsky et al. [4]. However, a segmentation context requires pixel-wise classification instead of whole-image classification. Long et al. addressed this with a fully-convolutional architecture that outputs dense pixel-wise predictions [5]. A recent, popular innovation to the fully-convolutional architecture that has produced great restuls is the ”deconvolutional”, or ”encoder-decoder” architecture [6,7,8]. This architecture features a symmetrical network that first learns an encoding by downsampling with convolutions and then learns to decode into a segmentation mask by upsampling with ”deconvolutions”.

In this work, we present a CNN with a 3D encoder-decoder architecture to automatically segment synovitis lesions in MR volumes. Similar to V-Net from Milletari et al., we equip our network with a loss function based on the Dice coefficient to deal with class-imbalance [8]. We show that our CNN can perform fast and accurately on test patients.

2 Materials

Our dataset consists of 61 MRI volumes of dimension $512 \times 512 \times 20$ from 31 patients at various timepoints. These volumes are available in both a T1-weighted pre-gadolinium injection sequence and a T1-weighted post-gadolinium injection sequence. The two T1-weighted sequences form the two input channels for our synovitis segmentation network. We split up our dataset into 51 volumes for training/validation use and 10 volumes for a test set.

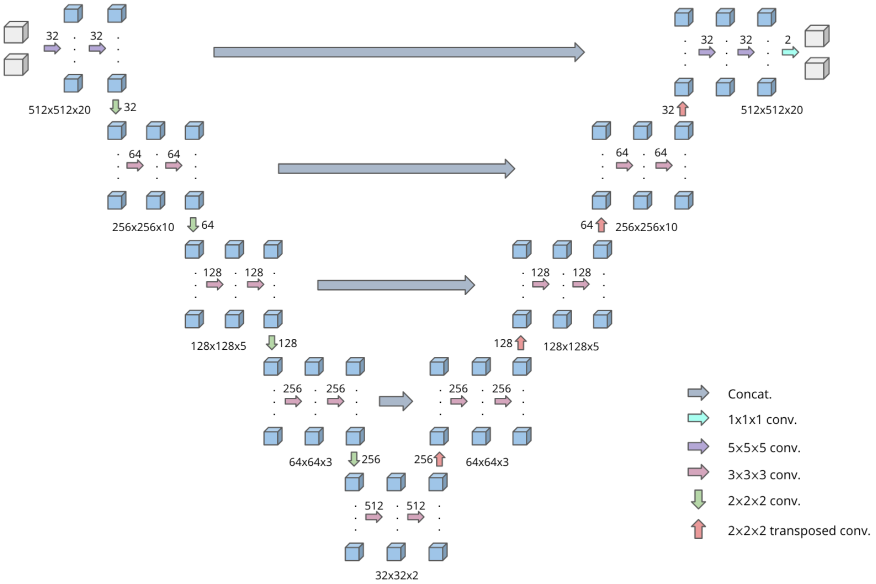

Figure 1: The architecture of our CNN. The numbers on top of the arrows are the number of filters. The spatial dimensions of each level of the CNN are on below.

3 Method

3.1 Architecture

Figure 1 gives a visual representation of our network architecture. The left half of the network is downsampling to learn an encoding, while the right half of the network decodes the learned encoding and outputs a segmentation mask at the end. There are 26 convolutional layers in total.

The network consists of five layers in terms of spatial dimensions, each one half the size of the previous layer. Each layer consists of two convolutional layers. Convolutions within layers have stride of 1 and are padded to preserve the same dimensions. On the left side of the network, $2 \times 2 \times 2$ convolutions with stride 2 are are used to downsample between layers, whereas on the right side, $2 \times 2 \times 2$ transposed convolutions with stride 2 are used to learn an upsampling. In the first layer, the kernel size of the convolutions is $5 \times 5 \times 5$ in order to capture more complex features on the lowest level of th e heirarchy of features. Within all the other layers, kernels of size $3 \times 3 \times 3$ are employed. All convolutional layers are equipped with ReLU non-linearities. A batch normalization is used after every convolutional layer.

As with a standard CNN, the receptive field of filters in consecutive layers is increased linearly within layers, and doubled when moving down a layer. Since the spatial dimensions are halved when moving down a layer, we can double the number of filters. Likewise, the spatial dimensions are doubled when moving up a layer so the number of filters is halved.

Learning an upsampling or interpolation scheme on the right side of the network can result in coarseoutlines and segmentations due to loss of detailed information on the left side encoding path. To help deal with this, we employ fine grained feature forwarding from each level on the left side to the corresponding level on the right side, like many previous works [5,6,7,8]. This allows for a more precise upsampling to be learned from the transposed convolutions.

In the final convolutional layer on the right side, we employ a two filters with kernel size of $1 \times 1 \times 1$ in order to output our two volumes of logits, each with the same spatial dimension as the input $(512 \times 512 \times 20)$, one for each class. These logits are then passed to a softmax operation to extract our final class probabilities for each voxel.

3.2 Training

We train with a loss function based on Dice coefficient as follows, where P is the length N ($N = 512 \times 512 \times 20$) vector of predicted probabilites (output of softmax) and Y is the length N vector of binary ground truth labels.

$$D = \frac{2 \sum_i^N p_i y_i}{\sum_i^N p_i + \sum_i^N y_i + \gamma}$$

Note that $p_i \in P$ are the raw probabilities and not the binary predictions that would be used to calculate the Dice coefficient when evaluating accuracy - this is because a non-differentiable *argmax* operation would be neccessary to extract the predictions from the probabilities, which would prevent gradient calculation by the optimizer.

We train using the Adam optimizer with initial learning rate $\alpha = 1e - 04$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $epsilon = 1e - 04$. We use batch size of 1 and decay the learning rate by a factor of 0.15 every 4 epochs. The small batch size of 1 was found neccessary due to memory limitations.

We implemented the model in Python using Tensorflow. We trained for 18 hours on a Nvidia K80 GPU with 12GB of VRAM, but found that the model converged in less than 10 hours.

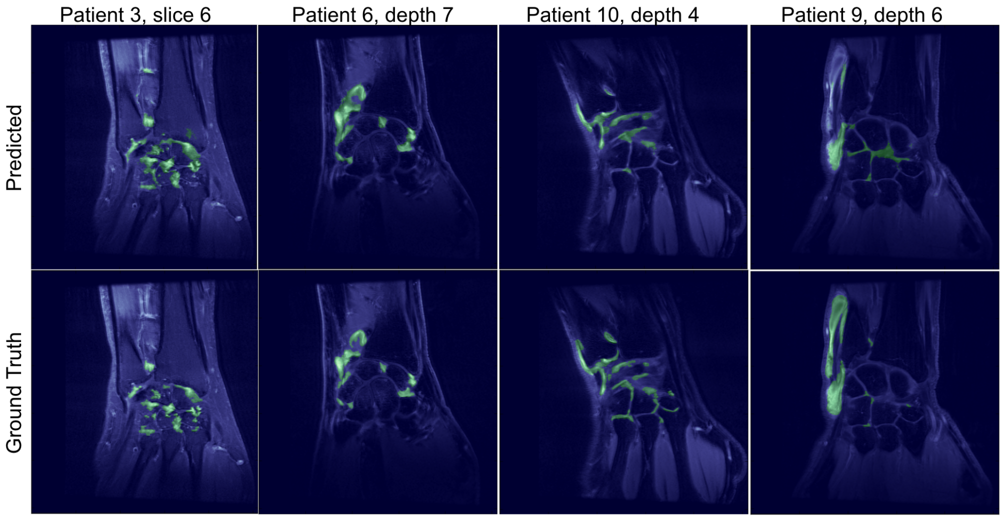

Figure 2: Four slices of four volumes from the test set. The model's predicted segmentations are on top, while the ground truth segmentations are below.

4 Results

We obtain an average Dice coefficient of 0.613 (± 0.078) across a test set of 10 images, compared against a manual ground-truth segmentation done by a radiologist.

Qualititavely, the model appears to have learned the general position of synovitis (i.e. the synovial membranes between wrist bones). It produces best results when the synovitis is contained to within the center of the volume, such as in *Patient 3, slice 6* of **Figure 2.**. The model is able to detect many small patches of synovitis, with respectable granularity - it is not limited to big, continuous patches.

The model fails to detect some synovitis outside of the center of the volume and performs worse, such as in *Patient 9, depth 6* of **Figure 2.**, where there is a large area of synovitis in the tendon. We notice that in this case the model also detects synovitis in the center membranes where there is none. Another shortcoming we noticed is that the model seems poor at detecting long, thin patches of synovitis, as is the case in *Patient 10, depth 4* of **Figure 2.**. This is a known weakness of CNNs.

The raw Dice Coefficients on the test patients are available in **Table 1.**

Patient	Dice Coefficient
1	0.558
2	0.675
3	0.682
4	0.651
5	0.615
6	0.616
7	0.680
8	0.562
9	0.433
10	0.655

Table 1. The raw Dice coefficents on a test set of 10 volumes.

5 Conclusion

We presented an approach to automatically and accurately segmenting synovitis lesions using a convolutional neural network. Our network had a symmetric, end-to-end, ”encoding-decoding” architecture, consisting of 26 convolutional layers in total. The left side of the network downsamples the input volume from $512 \times 512 \times 20$ to $32 \times 32 \times 2$, while the right side of the network upsamples back to a $512 \times 512 \times 20$ segmentation. We trained with a loss function based on the Dice coefficient, which was found to help with class imbalance. While we arrived at promising results with our model, future work can look to improve performance and segment other types of lesions as well.

References

1. Gabriel S.E., The epidemiology of rheumatoid arthritis, Rheum. Dis. Clin. North Am. 27 (2001) 269281.

2. Hodgson R.J., O'Connor P., Moots R. MRI of rheumatoid arthritis - image quantitation for the assessment of disease actiivty, progression and response to therapy. Rheumatology 47 (2008) 13-21.

3. Czaplicka K, Wojciechowski W, Wlodarczyk J, Urbanik A, Tabor Z. Automated assessment of synovitis in 0.2 T magnetic resonance images of the wrist. Computers in Biology and Medicine 67 (2015) 116-125.

4. Krizhesky A, Sutskever I, Hinton G.E. ImageNet classification with deep convolutional neural networks. NIPS (2012) 1106-1114.

5. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation (2014). arXiv:1411.4038 [cs.CV]

6. Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentaiton. Proceedings of the IEEE International Conference on Computer Vision (2015) 1520-1528.

7. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 (2015) 234-241.

8. Milletari F, Navab N, Ahmadi S. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation (2016). arXiV:1606.04797 [cs.CV]