2D to 3D Transfer Learning for Medical Images

Noam Eshed[1] Roberto Halpin[1] Varsha Kishore[1] Jiaming Wen[2]
[1] College of Engineering, Computer Science
[2] College of Agricultural and Life Sciences, Soil and Crop Sciences
Cornell University
Emails: ne236@cornell.edu, rgh224@cornell.edu, vk352@cornell.edu, jw2495@cornell.edu

Abstract

Computer vision has long been of interest in the medical field. Training models to do tasks such as organ and tumor detection and segmentation reliably would give doctors a great advantage in battling illnesses such as cancer. One of the main issues with training such a segmentation model is the lack of both 2D and 3D training data. A promising approach to doing medical scan segmentation is transfer learning. In this paper we employ transfer learning to segment tumors in 3D medical volumes. The 3D volumes are CT scans of pancreas that may or may not contain tumors. There are few pre-trained 3D neural networks, and 2D networks have the advantage of much larger data sets from which to train. Therefore, we propose to use transfer learning to initialize 3D weight filters by using a 2D pre-trained model. We will compare different methods of converting the 2D filters to 3D, and their effects on the segmentation accuracy calculated as intersection over union.

1. Introduction

The medical community greatly depends on body imaging using 3D scanning techniques such as computer tomography (CT), magnetic resonance imaging (MRI), and ultrasounds to diagnose and treat patients. Being able to train a 3D neural network, that is a neural network that takes in 3D data, and segments organs and tumors within the body would assist doctors in tasks like tumor detection. 3D imaging data that could be used to train such a network is difficult to come by for a variety of reasons. Since these imaging techniques tend to be expensive, uncomfortable, and may emit harmful radiation, they are usually only used when necessary. Patients must also give consent for their medical scans to be shared for use in research that does not directly concern their treatment [9]. As a result, there is limited 3D medical data available.

To deal with this challenge, we propose to use transfer learning. However, there are not many pre-trained 3D networks. As mentioned previously, it is difficult to find enough data to pre-train a 3D network. One possible approach to dealing with this problem is to splice the 3D medical scans into many 2D slices and then use them in a 2D convolutional network. The concern behind doing so is the loss of relevant spatial information; any two consecutive slices will be similar, as they are spatially close to each other. Splicing the data into 2D images would cause a loss in this valuable spatial similarity information. Hence, our aim is to build a 3D convolutional neural network that will make use of the spatial information in the CT scans. Figure 1 shows the difference between a 2D convolution and a 3D convolution. In this paper we propose to use a pre-trained 2D convolutional neural network, and then use weights from that network to initialize a 3D convolutional neural network that will be used to segment 3D scans of the pancreas as tumor, pancreas and background. Hence, we will use transfer learning and convert the 2D network to a 3D network.

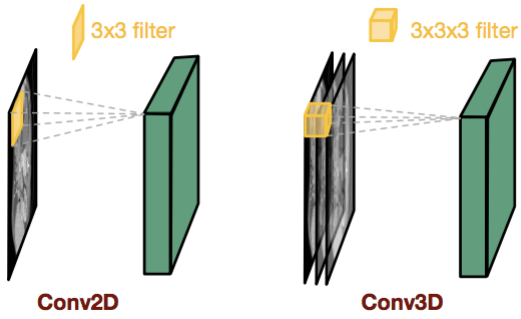


Figure 1: The left plot is a 2D convolution filter that takes as input a 2D region. The right plot is a 3D convolution filter that takes as input a 3D region. The output feature map is the same in both cases.[11]

2. Related Research

A lot of work has been done using transfer learning for 2D medical imaging. Limited work has been done in exploring 3D transfer learning. There has also been some research into transfer learning from 2D to 3D networks, but with different tasks and different motivations than ours. To our knowledge, no work has been done to transfer an out-of-domain 2D network to a 3D network for the purpose of medical image segmentation.

2.1. Transfer Learning and Medical Segmentation

Research done at the Division of Medical Image Computing, at the German Cancer Research Center in Heidelberg, proposed a U-Net model called the No-New-Net [5]. The No-New-Net comprised of multiple 2D and 3D U-nets and was used for medical image segmentation in the Medical Segmentation Decathlon challenge [1]. The No-New-Net is a self-adaptive model that can be used for segmenting medical scans of various organs. All the U-nets in this model were trained separately from scratch. No transfer learning was used, but the paper showed promising results for using 3D networks for medical segmentation.

There are promising results for using out-of-domain transfer learning for medical image data sets as well. Research done across medical imaging labs uses ImageNet pre-trained CNN models for recognition tasks [12]. The 2D ImageNet pre-trained network was transferred to a 2D medical image network for segmentation. This technique resulted in state-of-the-art performance in detection of thoraco-abdominal lymph nodes, and provides validity to our approach of applying 2D out-of-domain weights to 3D medical segmentation data.

A paper published in early 2018 proposed using 2D to 3D transfer learning in order to denoise low-dose CT scans [11]. Their motivation stemmed from the rising popularity of low-dose CT scans, which carry less radiation than regular CT scans, but produce lower-resolution images. They used a pre-trained model to initialize the 3D model weights, and noted that an advantage to this method is that it carries fewer parameters than a newly-trained 3D model. They trained a very specific 2D network called the Conveying Path-based Convolutional Encoder-decoder and used that to initialize a 3D model. We want to use a out-of-domain pre-trained network that is trained on large image data like ImageNet or RV-VOC12, fine-tune it on our 2D data, and use that to initialize a new 3D network, which makes our approach slightly different.

The most similar approach to the one presented here was done by Carreira et al at the Department of Engineering Science at the University of Oxford [2]. Researchers at Oxford used a 2D network pre-trained on the Kinetics data set [6] and transferred the model to 3D video data to better learn human actions between frames. In the 3D video data set, the extra dimension is time, but in our 3D medical data the third dimension is spatial. We believe that we can apply some of the techniques used in their work and build upon it. Furthermore, we want to see how well 2D to 3D transfer learning works when the extra dimension in the data is spatial and not temporal.

2.2. Initializing 3D Networks

In order to transfer a 2D neural network into 3D, a 3D weight filter must be initialized using the 2D weights filters. Mansimov et al introduced some approaches to this conversion in the paper 'Initialization Strategies of Spatio-Temporal Convolutional Neural Networks' [8]. Carreira et al used some of the techniques in this paper on the kinetics video data set and obtained promising results. The simplest 3D weight initialization is *Zero Weight Initialization (ZWI)* (1).

$$W^{(3D)} = \begin{cases} W^{(2D)}, & t = \text{middle slice of filter} \\ O, & \text{otherwise.} \end{cases} \quad (1)$$

In this approach, the middle slice of a 3D filter is initialized to have the same weights as a corresponding 2D filter. All other filter slices are initialized to matrices of zeros. Note that ZWI without fine tuning is equivalent to a 2D network because we are only using information from a single slice. With fine tuning, weights for adjacent slices will be learned. The concern with using ZWI is that it initializes very high weights to the center slice of the filter, while ideally all slices should yield a similar contribution. As such, using *Initialization by Averaging* (2) might yield better results.

$$W_t^{(3D)} = \frac{W^{(2D)}}{T}, \forall t \in 1, ..., T \quad (2)$$

A variation of *Initialization by Averaging* (2) is *Initialization by Scaling* (3). *Initialization by Scaling* scales each filter layer by a different weight that corresponds to its importance.

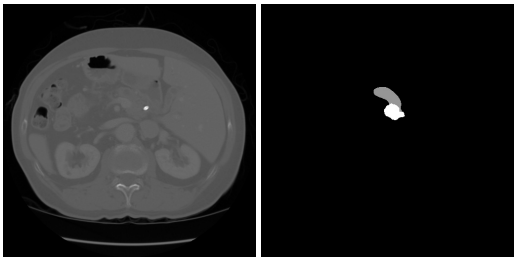
$$W_t^{(3D)} = \alpha_t * W^{(2D)}, \text{ where } \alpha_t > 0 \text{ and } \sum_{t=1}^{T} \alpha_t = 1 \quad (3)$$

We attempt all three methods of weight initialization, and compare the results to determine which method is the most successful.

3. Data set

For this project, we used 3D CT scan images of pancreas. This data was obtained from the Medical Segmentation Decathlon Challenge [1]. The data was originally collected by the Memorial Sloan Kettering Cancer Center. In the data

set, there are 282 3D volumes of portal venous phase CT scans. The dimension of the images is $512 \times 512 \times d$, where d is the depth of the image. The depth varies for different data samples. The data set also contains corresponding ground truth segmentation labels for each 3D image. The labels in this data set are the pancreas, pancreatic tumor, and background. It is especially hard to accurately segment the pancreas because the background is very large when compared to the size of the tumor itself. Figure 2 shows a sample 2d slice of the 3D data.



(a) Image (b) Ground Truth Label

Figure 2: The picture on the left (a) above is one slice of a 3D CT scan. The picture on the right (b) contains the segmentation labels for the slice on the left. In (b), black is background, grey is pancreas and white is tumor.

The provided images are gray-scale. We converted the images to RGB images because the pre-trained 2D network we use expects three-channel RGB images. Each CT image was copied three times to serve as the RGB channels. Furthermore, we re-scaled all the pixel values to be between 0 and 1.

4. Approach

The main method used in this paper is as follows:

1. *Train a network on 2D images:* The first step is to train an out-of-domain 2D neural network in order to obtain 2D filter weights. These weights will be used for transfer learning. Due to time restrictions, an already pre-trained network was used for this project. We used a dilated ResNet-18 [10][3] that was pre-trained on the PASCAL Visual Object Classes Challenge 2012 (RV-VOC12) data [4]. RV-VOC contains 20 classes of natural images of people, animals, vehicles, and household objects [4].

2. *Modify the 2D network:* The 2D neural network must be adapted to the three-class segmentation task at hand (segmenting background, organ, and tumor) and fine-tuned. The neural network was adapted by changing the last layer to output one of three classes (instead of one of twenty classes).

3. *Initialize a 3D network:* The 2D filter weights from the 2D neural network are used to initialize a 3D neural network. The 3D medical images are then fed into the neural network and the network is fine-tuned. Since this is a 3D network, it should make use of spatial information between consecutive slices and out perform the 2D network.

4.1. 2D segmentation as benchmark

To obtain a baseline for performance, we split our 3D images into 2D slices. The images were split along the third dimension, so every slice was a 512×512 2D image. As mentioned above, to accommodate the pre-trained model, we had to modify the last layer of the ResNet to output one of three classes. Since we are using an out-of-domain neural network we had to fine tune the network in order to obtain reasonable results. We fine-tuned the pre-trained ResNet-18 using the 2D slices to get a baseline performance. Then, the pre-trained model was applied to all 2D slices and the mean intersection over union (IoU) score was evaluated.

4.2. 3D transfer learning

The next step was to use the 2D dilated ResNet-18 network to initialize a 3D network. We made a 3D dilated ResNet-18 by changing all the 2D filters to their 3D equivalents. As shown in figure 1, a 2D filter and a 3D filter differ in input regions but output feature maps that have the same dimension. A 3D neural network needs more memory because of the extra dimension. As a result we could only use a batch size of 2 (we used a batch size of 32 for the 2D neural network). Since are batch size is low, training the 3D neural network was quite slow. An important choice we had to make was deciding how to use the 2D filter weights to initialize the weights of the 3D filters. We tested three different methods of initialization. The results of the methods are presented in the next section.

5. Experiments and Results

In this work, we tested three possible ways to assign the initial weights, i.e., padding with zero, averaging or customized scaling over the channels (see Section 2.2). For Initialization by Scaling, the α values were chosen as follows. For filters of kernel size 3: $\alpha_1 = 0.25, \alpha_2 = 0.5, \alpha_3 = 0.25$. For filters of kernel size 7: $\alpha_1 = \alpha_7 = 0.05, \alpha_2 = \alpha_6 = 0.1, \alpha_3 = \alpha_5 = 0.2, \alpha_4 = 0.3$. In our 3D network, all 3D filters have a kernel size of either 3 or 7.

The input to the 3D model was a 3D image. To ensure that each 3D image had the same dimensions, we set the depth of every image to 70. If the image had depth lower than 70, we removed the image from the data set. If the depth was greater than 70, we cropped the edges and only used the middle 70 slices. The slices on the edges were

Model	Mean IoU
2D net	0.3325
3D net + random initialization*	0.3329
3D net + zero weight initialization*	0.3565
3D net + initialization by averaging*	**0.3835**
3D net + initialization by scaling*	0.3689

Table 1: Results
*Performance improves w/ longer training

mostly background, so this does not greatly affect the results. We removed about 25 data samples because they had a depth lower than 70. The 3D neural network was fine-tuned on the 3D images and the mean IoU was computed. Table 1 shows the results obtained from the different experiments conducted. For each experiment, the neural network was trained for 30 epochs. The mean IoU values might seem low, but the pre-trained network on the RV-VOC12 data set only achieved a mean IoU of 0.59. Since we are adapting this pre-trained network for an out-of-domain task, our IoU scores are even lower, as is expected. Our main aim is to show that custom initialized 3D networks outperform 2D networks for our 3D segmentation task. Our experiments show that this is indeed true.

We observed that the 2D ResNet mean IoU converges to 33.25% in about 5 epochs. Even after continued training, the mean IoU did not increase past 33.25%. As such, this is the best IoU score that we could obtain by fine-tuning a RV-VOC12 pre-trained ResNet-18 with 2D images.

On the other hand, for the 3D ResNet the mean IoU kept increasing even after 30 epochs. We had to stop the training at 30 epochs due to time constraints. We tried using different learning rates to make the network converge faster, while keeping the learning rate low enough to prevent the model from diverging. We used the Adam optimizer [7] with a learning rate of 0.001, betas of 0.9 and 0.999, and a weight decay of 0.0001 to train our models. We see from the results that the mean IoU obtained using a 3D ResNet with random initialization is only marginally better than the mean IoU obtained from using a 2D network. The 3D network did seem to be improving but the mean IoU score did not increase by much and it seems like it would take a very long training time to get better results using the 3D network with random initialization.

In comparison, the custom initialized 3D neural networks performed better and the mean IoU improved at a faster rate. We see from the results in Table 1 that the 3D ResNet + Initialization by Averaging yields the best results. It is however possible that Initialization by Scaling produces better results if different α weights were used.

6. Conclusions and Future Work

In conclusion, we have shown that 3D neural networks have the capacity to outperform their 2D counterparts, likely because they take into consideration the extra spatial information between different slices of the 3D image. The transfer learning performed did result in an improved performance. However, one drawback of using 3D architectures is that they require more memory and time.

The work presented in this paper results in a number of possibilities for future research. We would use better pre-trained networks, that can then be used to initialize a 3D neural network. Using a pre-trained model that is specifically trained on medical data should yield better results. U-nets have been used to achieve state-of-the-art results in medical imaging tasks, so using a better suited pre-trained model like a U-net could also lead to better results. Another avenue for improvement is to explore other methods of using 2D filter weights to initialize 3D filter weights. Lastly, it would be interesting to apply the technique described in this paper to other 3D medical image segmentation data sets and evaluate how well the technique performs on different data sets.

References

[1] Medical Segmentation Decathlon. http://medicaldecathlon.com/.

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *arXiv:1705.07750v3 [cs.CV]*, 2017.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915 [cs.CV]*, 2016.

[4] M. Everingham, L. Van-Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012 (voc2012) results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/, 2012.

[5] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, and K. H. Maier-Hein. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv:1809.10486v1 [cs.CV]*, 2018.

[6] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv:1705.06950 [cs.CV]*, 2017.

[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980 [cs.LG]*, 2014.

[8] A. Mansimov, N. Srivastava, and R. Salakhutdinov. Initialization strategies of spatio-temporal convolutional neural networks. *arXiv:1503.07274v1 [cs.CV]*, 2015.

[9] U. D. of Health and H. Services. Your Medical Records. https://www.hhs.gov/hipaa/

for-individuals/medical-records/index.html. [Online; accessed 6-Dec-2018].

[10] D. Pakhomov, V. Premachandran, M. Allan, M. Azizian, and N. Navab. Deep residual learning for instrument segmentation in robotic surgery. *arXiv preprint arXiv:1703.08580*, 2017.

[11] H. Shan, Y. Zhang, Q. Yang, U. Kruger, M. K. Kalra, L. Sun, W. Cong, and G. Wang. Correction for 3d convolutional encoder-decoder network for low-dose ct via transfer learning from a 2d trained network. *arXiv:1802.05656v2*, 2018.

[12] H. Shin, H. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. Summer. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging, doi: 10.1109/TMI.2016.2528162*, 2016.