Convolutional Neural Networks in Medical Imaging

Mitchell Finzel
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
finze008@morris.umn.edu

ABSTRACT

Currently unchanged from section draft

Ever since the boom in convolutional neural network popularity due to the success of their use in the 2012 ImageNet competition, there has been a large uptake in their use in the field of medical imaging segmentation and classification. Over the past 5 years there has been a surge in their success, achieving state-of-the-art performance across a broad variety of medical imaging systems ranging from the segmentation of knee cartilage all the way to the detection of Alzheimer's disease in MRIs. In this paper we will go over some of the cutting edge architecture techniques being used specifically for the tasks of brain segmentation and classification. The results are proving to be quite promising and could be a step towards the adoption of automatic segmentation systems in day to day medical diagnosis.

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

In 2012 convolutional neural networks or CNNs, were used to great success improving drastically over the previous state-of-the-arts in the ImageNet computer vision competition. [3] Since their success in image recognition CNN's have seen a rise in popularity, finding their way into more complex computer vision challenges such as medical imaging. In the past 5 years there has been a large uptick in the use of CNNs in biological segmentation tasks. These tasks extend across a wide variety of human anatomy. For example CNNs have been used for the automated detection of lymph nodes [6], the segmentation of knee cartilage [5] and Alzheimer's detection [4] to name a few. For this paper we will be focusing on two specific examples of CNN use in medical imaging segmentation [1] by Havaei, et al. and [2] by Kamnitsas, et al. These two papers show different approaches to CNN architecture applied to the segmentation of MRIs. While the results obtained by both approaches is not directly comparable we will go through what makes their approaches unique and where there is some overlap. After discussing their features we will take a look at their state-of-the-art results on a few different segmentation challenges.

UMM CSci Senior Seminar Conference, April 2017 Morris, MN.

2. BACKGROUND

Before delving into the specifics of the network architecture approaches that are being employed we will look at some of the fundamentals of CNNs and segmentation in general. When talking about segmentation in the field of medical imaging we are talking about being able to classify different parts of a medical image. For instance in an MRI of a brain we might want to segment base on what's white versus grey matter. These "labels", white matter and grey matter, are the outputs of the CNN. Currently most of this segmentation is done by hand by medical professionals, but advances in automatic segmentation have made great strides in the past few years.

2.1 Neural Networks 101

At their most basic form neural networks are essentially pattern recognizers. We train the neural network on a large amount of input and then it learns to recognize features and associates them with the output labels. Neural networks are generally composed of a number of layers that perform a variety of operations, each layer learning from the previous ones.

2.2 Convolutional Layers

Convolutional Layers are what differentiate convolutional neural networks from other neural networks, the first layer of all CNNs is a convolutional one. The convolutional layer takes an array of values that represents either the pixels or voxels of the input image. The layer then uses what is interchangeably called a filter, neuron or kernel, which is another array with the same depth as the input array. The kernel is then aligned to the upper left corner of the input, the area it covers is called the receptive field. The array contained within the receptive field is the multiplied with the array in the kernel using element-wise multiplication. The multiplications are then summed up and stored in the same relative position of what's called a feature map. The kernel then slides over a specified distance on the input and performs the same operation. What we end up with after all of the possible convolutions of the kernel and the input is a completed feature map. The feature map is an array that contains all of the results of the convolutions between the kernel and the input.

2.3 Kernels

Kernels, as described above, are arrays of values that are meant to represent features to be recognized. For instance a kernel could contain a feature such as a curve. This might

be represented by a pattern of numbers in the kernels array. When the kernel is multiplied with the input the result will be a higher number if the feature in the kernel is similar to the feature explained by the receptive field. If the feature described by the kernel is not present in the receptive field then the result of the multiplication will be relatively smaller. These recognitions of features are then stored in the feature map where they will likely be used as the input to the next layer. To increase the depth of feature recognition in a layer, just add more kernels. As these feature maps are used in future layers of similar operations a hierarchy of features is created with more complex features being represented in later layers.

2.4 Pooling Layers

Another oft used layer type is called a pooling layer. These layers have a relatively straightforward purpose, they take clusters from the input feature map and reduce them down to a single feature. For instance, an example of a pooling operation is max-pool. A max-pool pooling layer will divide the input into clusters and place the highest value of each cluster in their corresponding place in the pooling layer's feature map. Pooling layers are used to drastically reduce the amount of spacial data by eliminating a large portion of the input in one step. They also reduce the effects of overfitting, which is essentially when the neural network becomes to finely tuned to the training data and fails to generalize well.

Need sections on fully connected layers, softmax classifiers and training.

2.5 Softmax Layers

3. METHODS

3.1 Multi-modality approach in infant isointense brains

3.2 Novel approaches in brain tumor segmentation

Only minor edits since the section draft; inclusion of a figure

In [1] Havaei et al created a novel two path approach for a CNN trained on the Multimodal Brain Tumor Segmentation (BRATS 2013) challenge data set. Their approach can be broken down into three main components, the use of two pathways, the concatenation of one CNN's output into a seconds and the use of a two-phase training approach.

To setup their two pathway CNN Havaei et al created an architecture with two streams, a local pathway with a 7 x 7 receptive field and a global pathway with a 13 x 13 receptive field. By combining a localized and global perspective the architecture has the ability to detect visual detail around the centered pixel while also capturing data about the greater context of that pixel's location within the brain.

These two pathways are concatenated together after going through a series of convolutional layers, 2 layers for the local pathway and 1 for the global pathway. This final concatenation of the two pathways is then sent through the output layer to be interpreted as segmentation labels.

An issue with traditional CNN segmentation systems is that they predict the segmentation labels independent of one another, ignoring the possibility of joint segmentation label models. To address this issue [1] propose three different cascaded CNN architectures, where the output of one CNN is concatenated into one of the layers in a second CNN. By using a cascaded architecture Havaei et al allow the second CNN to learn from the values of nearby labels.

The three different cascaded architectures implemented by Havaei et al are variations on their two pathway approach described above; both of the CNNs, the original and the one being concatenated in later, use two pathways. The main variation between these three different architectures is the location of the concatenation of the first CNN with the second. As can be seen in 1 the first implementation, InputCascadeCNN, takes the output of the first CNN and directly concatenates it to the input of the second CNN. The second implementation, LocalCascadeCNN, takes the output of the first CNN and concatenates it to the first convolutional layer of the second CNNs local pathway. The final implementation, MFCascadeCNN, concatenates the output of the first CNN to the final layer of the second CNN, directly before its output.

The third main component in the research done by [1] is the use of a two-phase training system. One of the large issues in training CNNs for segmentation is the relative abundance of healthy tissue as compared to the small quantities of tissue that fall under each label. This is especially true of brains where labels can be comprised of less than 1 percent of the total images composition. To alleviate this problem Havaei, et al. first train the CNN on a data set of image patches where all of the labels are equally probable. They then retrain the final output layer taking into account the relative probabilities of the labels,thereby keeping the discriminatory capability of the previous layers intact while maintaining proper output probabilities.

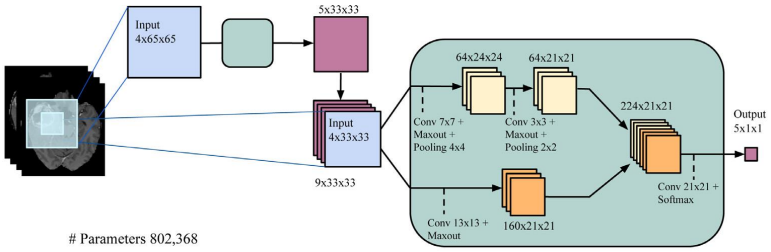
3.3 3D multi-scale approach in 3 lesion segmentation tasks

In [2] Kamnitsas, et al. provide one of the most recent architectural approaches to CNNs in the field of medical imaging segmentation. Their work can be boiled down to a few main techniques drawing from a wealth of past research. These main techniques include the use of 3D CNNs, dense-inference for network training, 3D CRFs for the final processing of the networks soft segmentation maps, deep networks for better discrimination and two pathways using a multi-scale approach.

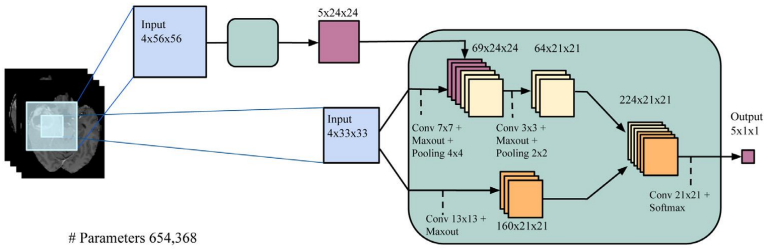
Much like the work done by Havaei, et al. Kamnitsas, et al. uses a two pathway approach to better capture local information while maintaining the broader context of the entire image. However, while Havaei, et al. accomplishes this two pathway approach by using a different sized receptive field in each pathway, Kamnitsas, et al. downsamples the image itself.

I need to better understand if there is a difference between these methods

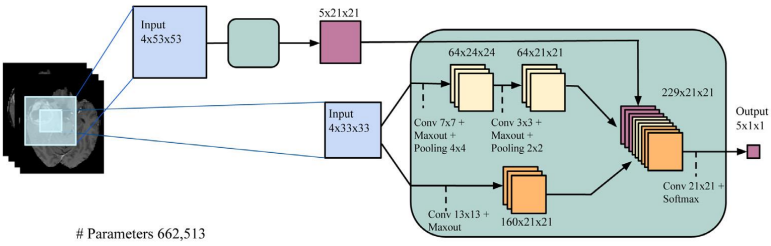
The network displayed in 2 is a simplified version of Kamnitsas full blown network "DeepMedic". As can be seen in the figure the network has two pathways, one with the normal resolution image patch and the other with a downsampled patch. These patches then go through a series of convolutional layers before two fully connected layers and the final classifier layer. DeepMedic has twice as many con-



(a) Cascaded architecture, using input concatenation (InputCascadeCNN).



(b) Cascaded architecture, using local pathway concatenation (LocalCascadeCNN).



(c) Cascaded architecture, using pre-output concatenation, which is an architecture with properties similar to that of learning using a limited number of mean-field inference iterations in a CRF (MFCascadeCNN).

Architectures.pdf

Figure 1: The three different "cascaded" architectures used by Havaei, et al. [1]

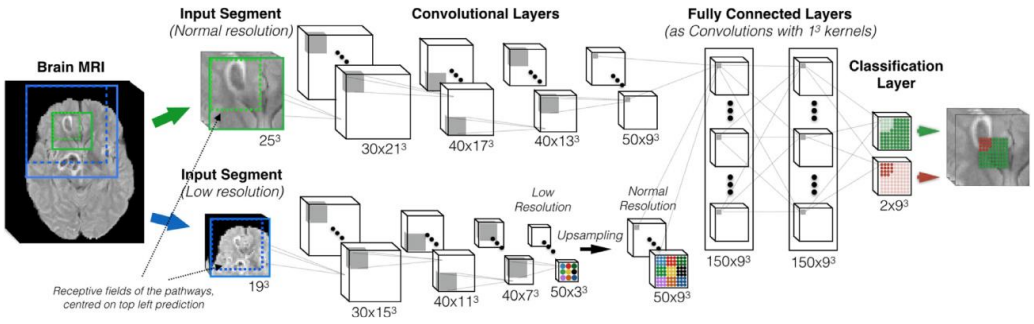


Figure 2: The basic neural network architecture used by Kamnitsas, et al. in [2]

volutional layers as the depicted figure, using a smaller 3^3 kernel size.

One of the biggest attributes of the work done by Kamnitsas, et al. is the use of 3D CNNs for the basis of their architecture. In the past, research has avoided the use of full blown 3D CNNs due to their increase in parameters and computational requirements. Prior research such as that done by Havaei, et al. [1] used 3D slices of the medical image. Hybrid approaches such as training on three orthogonal slices (coronal, sagittal and axial) have been used to cheaply approximate 3D CNNs [6]. The 3D CNNs used by Kamnitsas, et al. use three-dimensional kernels in the convolutional layers to create the feature maps. This can be thought of as a cube traversing the volume of the image at hand. The use of these 3D CNNs allow for better use of the volumetric data at hand and allows for the full exploitation of dense-inference.

Need to give more in depth explanation of 3D CNNs

Explain Dense Inference, also known as better understand what dense inference is

In CNNs it has been shown that deeper networks have greater discriminitive capability due to their additional non-linearities and better defined local optima. While deeper networks have greater discriminitive power it is also true that they have greater computational costs, which is further exaggerated by the use of 3D convolutions. There is also an increase in the number of trainable parameters that is associated with the use of 3D CNNs. To combat these barriers Kamnitsas, et al. replace convolutional layers that have a standard 5^3 kernal size with multiple layers that use a smaller 3^3 kernel size. These smaller kernels require fewer parameters and lead to a significant decrease in element-wise multiplications per layer. [2]

Deeper networks also suffer from being harder to train. As the network becomes deeper it becomes harder to preserve the signal. This is caused by the multiplication of the signals variance as it propagates through each layer. To alleviate this Kamnitsas, et al. initialize their kernel weights to the normal distribution

Add formula

. They also use a technique called batch normalization to deal with the similar issue of covariate shift, but this is outside of the scope of this paper.

Explain the 3D CRF, also known as better understand 3D CRFs

3.4 3D approach in Alzheimer's

4. RESULTS

Explain what dice, specificity and sensitivity are

As mentioned before the work done by Havaei, et al. was applied to the BRATS 2013 challenge. They also attempted to use their system on the BRATS 2014 dataset, but were unable to get the system to work due to problems with the dataset itself. For the BRATS 2013 Challenge contenders used an online evaluation tool that provided Dice, specificity and sensitivity scores for the three categories the complete tumor region, the core tumor region and the enhancing tumor region. Their highest scoring network configuration improved on the state of the art in both accuracy and speed.

Add table of results for the for Havaei BRATS 2013

The architectures proposed by Kamnitsas, et al. were applied to three different brain related challenges. The first was a challenge involving a database of MRIs from people who had suffered traumatic brain injuries, TBI. The second was on the BRATS 2015 dataset, which much like BRATS 2013 measured dice, specificity and sensitivity on the three categories of brain tumor hierarchies. The last was on the ISLES 2015 challenge that dealt with the segmentation of brain lesions caused by stroke. In all cases the DeepMedic + CRF architecture they propose beat the scores of the previous state-of-the-art approaches. In all cases the addition of the 3D CRF was found to have a statistically significant affect on performance.

More tables of results, probably don't need to show all of them

Figure 3 shows an example pulled from [2] of DeepMedic's segmentation of three different MRIs. As can be seen in the figure the network does a good job of segmenting both small and large lesions. In the second row however, it undersegments a contusion, possibly mistaking it for background. The third row also shows one of the greater challenges of this segmentation task, where post-surgical sub-dural debris is mistaken as a relevant lesion. [2]

5. CONCLUSIONS

As demonstrated by the work of Havaei, et al. and Kamnitsas, et al. convolutional neural networks show a lot of continued promise going forward. The use of multiple pathways for local and global context was shown to be effective in both works. Havaei, et al. has further shown that cascaded architectures with two-phase training are also useful in improving the state-of-the-art. Meanwhile Kamnitsas, et al. have shown that 3D CNNs have become computationally feasible and that Dense-inference and 3D CRFs show promise as well.

These two works are unfortunately not comparable when it comes to the challenges they partook in, but nevertheless both show a breadth of space for further improvement when it comes to CNNs and medical imaging. Future work could combine some of the techniques used such as using a cascaded architecture with DeepMedic or attempting two-phase training.

Acknowledgments

6. REFERENCES

[1] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18 – 31, 2017.

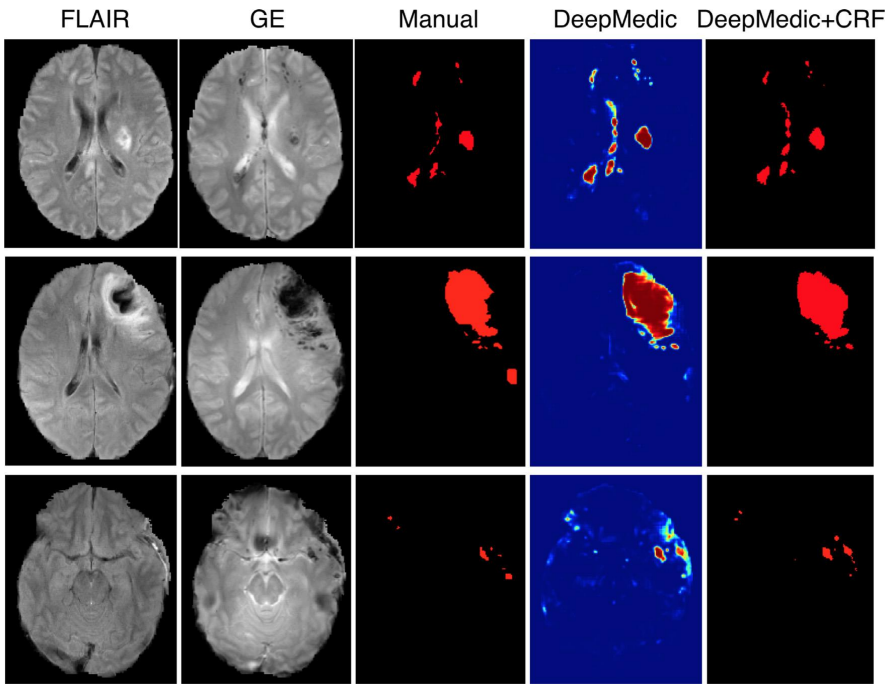
[2] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. Efficient multi-scale 3d {CNN} with fully connected {CRF} for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61 – 78, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[4] A. Payan and G. Montana. Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. *CoRR*, abs/1502.02506, 2015.

[5] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen. *Deep Feature Learning for Knee Cartilage Segmentation Using a Triplanar Convolutional Neural Network*, pages 246–253. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[6] H. R. Roth, L. Lu, A. Seff, K. M. Cherry, J. Hoffman, S. Wang, J. Liu, E. Turkbey, and R. M. Summers. A new 2.5d representation for lymph node detection using random sets of deep convolutional neural network observations. *CoRR*, abs/1406.2639, 2014.



Visuals.pdf

Figure 3: Three examples from the TBI dataset. The first two columns show the original MRIs with the third column showing the images segmented manually and the last two columns showing the segmentation performed by DeepMedic and DeepMedic with the CRF. Kamnitsas, et al. [2]