A Look into Brain Tumor Segmentation with Deep Neural Networks

Arran Scaife AMS2UXK@VIRGINIA.EDU
Arthur Harris RAH9EU@VIRGINIA.EDU
Avni Garg AG9WSE@VIRGINIA.EDU

Abstract

This is a paper review for the research paper "Brain Tumor Segmentation with Deep Neural Networks" by Authors Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal,Pierre-Marc Jodoin, and Hugo Larochelle. Our paper review takes a look into the proposed model with regards to implementation, training time and differences with regards to previously used methodology and related work. We also run experiments using data provided in class and add some of our results obtained. In this paper, we also run the code on the given dataset in class and evaluate it critically with regards to described implementation. Furthermore, we discuss possible avenues of further research and critiques we have of this paper.

Keywords: Cascaded Architecture, Convolutional Neural Network, Brain Tumor Segmentation

1. Introduction (Arran Scaife)

This paper takes a look into how to implement brain tumor segmentation using deep neural networks for the BRATS 2013 competition. This is an especially difficult problem as no two brain tumors are exactly alike, making their segmentation unique with each case because of size, location and shape variation. We are also dealing with the issue of MRI machines creating different contrasts which leads to dissimilar data. The goal of this paper is to handle varying inputs with different representations to segment unique representations of tumors in the brain. To that end, there has been previous related work which has influenced the paper in its current form.

The first class of work done on this subject is Classical Generative models where the models relied on prior knowledge of healthy and tumorous tissue appearance in order to obtain the atlas, compute voxel wise posterior probabilities, and isolate abnormalities via thresholds for low posterior probabilities. To that end, some important work in this area was conducted by Prastawa. His particular paper aligns brains and tumors to the already given ICBM brain atlas, computes posterior probabilities and find clusters of low posterior probabilities either through K-Means ($K = 2$) or active contours that rely on brain symmetry features or alignment based features.

The second class of work done on this topic is Classical Discriminative Models which extract many low level features (models such as SVMs or Decision Forests). The most accurate discriminative methods have been Random Forests. Another factor of these models is hand engineered features. By their nature, many hand-engineered features exploit very generic edge-related information with no specific adaptation to the domain of brain tumors.

Ideally, one would like to have features that are composed and refined into higher-level, task adapted representations. Recently, preliminary investigations have shown that the use of deep CNNs for brain tumor segmentation makes for a very promising approach. One example is the work done by Zikic and Menze, where the output of a generative model is fed as input to a discriminative classifier. There is also related work employing classical CNN architectures by Zikic and Urban, where the architecture has a sequence of convolutional layers, non linear activation functions between each layer, and a softmax output layer. Then they created higher level representations without hand engineered features. This model is now state-of-the-art for brain tumor segmentation.

The ultimate challenge of this paper is solving how to quickly capture local information with global context to segment properly.

2. Background (Arran Scaife, Arthur Harris)

Here are some relevant terms and definitions to understanding the paper:

- Feature Maps: It can be thought of as a topological map of responses if a window frame (spatial non linear feature extractor) was taken and slid over the input such that each area of the input was treated identically.
- Patch: A window of the larger input pane that is fed as an input to the CNN
- Slice: A lengthwise segment of the MRI, which shows us the brain in axial view, which contains a certain amount of information of the image. In 2 dimensions, it translates to another image.

3. Methodology (Arthur Harris, Avni Garg)

This paper goes goes through a variety of architectures to test which one has the best segmentation capability. Each of these architectures was built using keras sequential in tensorflow and has multiple layers.

The first architecture described is two pathway, where the input is a patch which is then fed as input to a local pathway with two convolutional layers that then outputs a feature map. The input patch is also fed into a global pathway that has the ability to model the image with more flexibility and discreteness which also outputs a feature map. These feature maps are then concatenated and run through the softmax activation to get the tumor segmentation.

The second type of architecture described is cascaded architecture which consists of two convolutional neural networks where the output of one of them is fed as the input to the other one either as an input or as another layer. We have 3 types of cascade architectures

The first type is input concatenation. In this architecture, the output of convolutional neural network 1 is fed as input to convolutional neural network 2.

The next type is local pathway concatenation. In this case, the output of convolutional neural network 1 is connected to the first hidden layer in the second convolutional neural network.

The final type is pre output concatenation, where the output of the first convolutional neural network is connected to the feature maps right before the softmax layer which gives

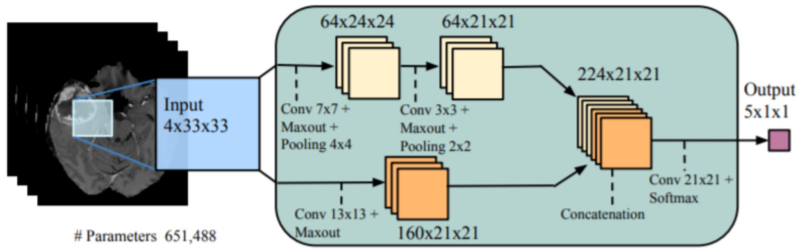
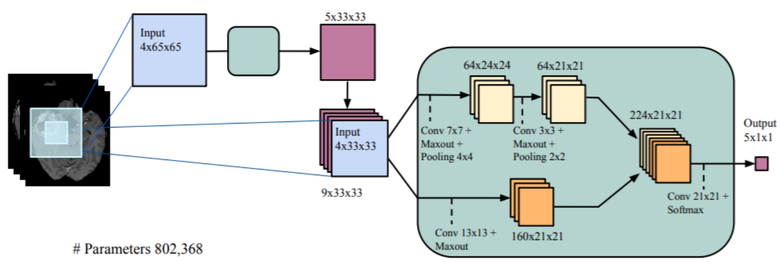
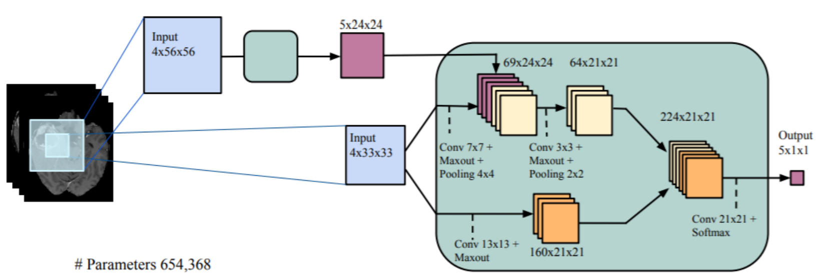


Figure 1: Two Path CNN



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

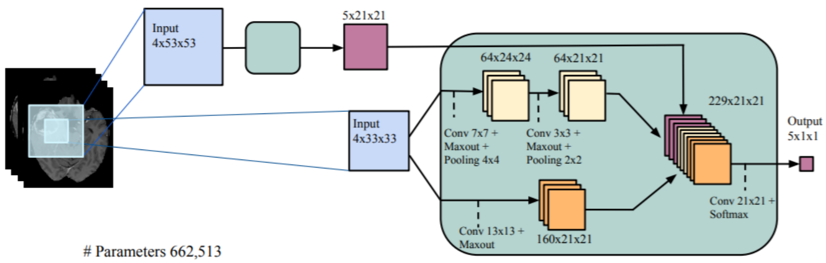
Figure 2: Input Concatenation



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

Figure 3: Local Concatenation

the output. The author ultimately went with input concatenation for the BRATS competition as it had the best performance.



(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).

Figure 4: Pre Output Concatenation

When it comes to training, the paper uses negative log probability as the loss measure, optimizes using stochastic gradient descent with momentum and prevents overfitting by using dropout stochastic regularization and elastic net regularization (L1 and L2 norm)

4. Experiment (Arthur Harris, Avni Garg)

When conducting experiments based on this paper, we used the segmentation dataset and built upon the implementation of this paper by Vaibhav Jade.

During the running of the implementation of the paper, we found many issues in the code and spent a lot of time resolving them. We started by changing the input to actually be able to receive the input and, once the code was able to actually process the code, we ran into many other difficulties.

At first, we saw that for a single image, we were generating 58 thousand slices based on the paper's patch methodology and knew we could never train that much data. It simply was not possible with the training time being so high for each slice. What we did to remedy this was first cut out slices that were essentially pointless to the training of the model. We then increased the step size from 1 to a third of the input size to further reduce the number of patches. In the end, we were able to cut the number of patches per image down to a range of 4.5 thousand to 5 thousand patches per image. To achieve this, we also removed 0 slices.

We also then lowered the number of epochs and batch size to get runnable training times of 12 minutes per epoch on average. This was so that we could at least see some results from the code when discussing the paper as the implementation of this paper does not lend well to being run all at once.

Here is one of the training runs on a singular image where we have 5748 patches, 5 weights, 2 epochs and batch size of 8 while using the Adam optimizer (Learning rate = 0.005) and Categorical Cross Entropy loss function.

```
Returned
Y_ints shape (5748,)
d[2] shape (5748, 5)
NP.unique yints [0 1 3]
d[2][0] [0 0 0 0 0]
Temp FIx {0: 8.57910447761194, 1: 15.966666666666667, 2: 1, 3: 8.57910447761194, 4: 3.3811764705882354}
{0: 0.3457235654998196, 1: 26.61111111111111, 2: 14.298507462686567}
(5748, 5)
Epoch 1/2
719/719 [==============================] - 712s 989ms/step - loss: 1.0595 - accuracy: 0.0421
Epoch 2/2
719/719 [==============================] - 711s 988ms/step - loss: 1.0020 - accuracy: 0.0423
```

Figure 5: Singular Train Run

5. Discussion (Arthur Harris, Avni Garg)

This paper was highly influential and has spawned many other research papers. At the time of this report, it has been cited at least 2000 times. There are some very interesting possible future research areas that came to mind when reading this paper.

One of those areas is 3+ way architecture. It would be quite interesting to see the resultant feature maps and whether or not more pathways leads to more accuracy.

Another area we think would be interesting to approach would be determining how many actual features are required to gain sufficient accuracy, as a lower number of filters would lower the training and evaluation time.

All of the cascaded architectures, if considered from another angle, are really 2 layer ResNets, as the output of the first network is propagated forward to a layer in the second neural network and then operations are performed on the initial input patch. If we look at the accuracy of the cascaded architectures while employing the ResNet perspective, the accuracy jump can be expected, however this paper would be strengthened if the authors provided mathematical backing for the architecture and why loss is minimized.

We noticed a discrepancy between the paper and its actual implementation with regards to the two path implementation. The means of concatenation of the output of two path and the the secondary neural network does not follow the paper's description. This leads to the question of what the results would be if the models were correctly implemented.

From a computational perspective, our group agreed that this model, in its current form, is quite impractical in terms of training a model. If each MRI scan creates about 58 thousand patches to train on, without access to very high and sophisticated computational power and coupled with an excess of time to train, the user is looking at a model that will inevitably crash their session due to the excessive RAM requirements and render the model useless. There is major scope for optimization in this model which would make this more accessible for users.

6. Conclusion (Avni Garg)

From these experiments, we can see that training is a very large and time consuming process in this model and quite inaccessible to users who do not have access to extreme computational power, considering that the primary users of this method would be hospitals that map tumors for patients. These hospitals may not be necessarily attached to academic institutions which may not necessarily possess the required capabilities to run a models like this, given the fact that they would need machine learning specialists who can deploy this to external sources to train. There is a lot of progress still to be made with regards to patch generation and creating meaningful data from the images we feed in so as to not waste a lot of training time and power on empty patches.

References

Davy, A., Havaei, M., Warde-Farley, D., Biard, A., Tran, L., Jodoin, P.M., Courville, A., Larochelle, H., Pal, C., Bengio, Y., 2014. Brain tumor segmentation with deep neural networks. in proc of BRATS-MICCAI .

Menze, B., Reyes, M., Leemput, K.V., 2014. The multimodal brain tumor image segmentation benchmark (brats). IEEE Trans. on Medical Imaging (accepted)

Prastawa, M., Bullit, E., Ho, S., Gerig, G., 2004. A brain tumor segmentation framework based on outlier detection. Medical Image Anaylsis 8, 275–283.

Urban, G., Bendszus, M., Hamprecht, F., Kleesiek, J., 2014. Multi-modal brain tumor segmentation using deep convolutional neural networks. in proc of BRATS-MICCAI .

Zikic, D., Ioannou, Y., Brown, M., Criminisi, A., 2014. Segmentation of brain tumor tissues with convolutional neural networks. in proc of BRATS-MICCAI.