DualNetFC: Lesion Segmentation on ATLAS via a Dual-Pathway Deep Network

Weston Ungemach[1] and Beite Zhu[2]

Abstract— **This work implements a dual-pathway 10-layer neural network (Fig. I.) for brain lesion segmentation on the ATLAS (Anatomical Lesions After Stroke) dataset, newly compiled and released by the University of Southern California, contains 229 T1-weighted MRI scans each labeled with a 3D mask segmenting (possibly) multiple lesions. We process these scans in two-dimensional horizontal slices and achieve sharper performance than a baseline encoder/decoder network with a significantly smaller network.**

I. INTRODUCTION

The analysis of stroke lesions informs medical treatment at two separate stages: the acute stage ($<$ 24 hours after stroke) and the longer term subacute/chronic stages. During the acute stage, clinicians face important time sensitive decisions, such as whether to perform surgery to prevent further brain damage or other impairment. During the subacute/chronic stages, clinicians typically work with patients to rehabilitate lost speech or motor function. In both of these cases, neurologists and neuroradiologists frequently analyze brain lesions as a result of the stroke throughout the decision-making process. While there exist several automated and semi-automated procedures for mapping these lesions, the industry standard is still expert hand-labeling, which is time-intensive and inefficient.

Here we analyze the ATLAS dataset of T1-weighted MRI scans complete with segmentation of brain lesions. We implement a dual-pathway deep neural network which processes two-dimensional slices from these scans of size 232×196 and predicts lesion masks for these slices.

II. RELATED WORK

Early work in brain lesion and tumor segmentation used traditional machine learning techniques (such as random forests) which treated these tasks as anomaly detection [1]. In these approaches, a new brain scan is compared directly to a healthy one and structural similarities and differences are compared. While this is generally an effective method for classification, it is much less useful for segmentation. Brain lesions are often large enough to change the large-scale brain structure, which makes direct comparison of anatomical features difficult. This renders such methods ineffective for segmentation.

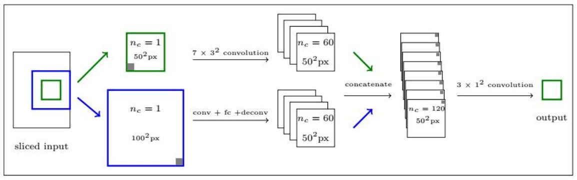

Fig. I. Crops from an input image are successively passed through our dual-pathway network before being synthesized into a final predicted mask.

Many recent approaches to these problems use neural network techniques. The inspiration for our network comes from tumor segmentation rather than lesion segmentation. In [2], the authors find the most effective means of tumor segmentation to be an ensemble learning approach with separate networks designed to target specific regions of the brain. This approach was synthesized in Kamnitsas et al. [3] with a dual network design very similar to ours. The idea is to combine the multiple region specific models by passing both local and global information into the network. This allows the network to learn the context surrounding the input image and utilize this information during segmentation, eliminating the need for location-specific models.

III. THE ATLAS DATASET

The ATLAS (Anatomical Lesions After Stroke) dataset, newly compiled and released by the University of Southern California, contains 229 T1-weighted MRI scans each labeled with a three-dimensional mask segmenting (possibly) multiple lesions. By taking horizontal cross-sections, this gives a collection of 43,281 two-dimensional T1-weighted MRI slices. Note that while every three-dimensional scan contains at least one lesion, many of the two-dimensional cross sections do not intersect these lesions. Labels were provided by a team of 11 trained labelers and verified by a neuroradiologist and an expert labeler. While there exist other large datasets of brain lesion data, ATLAS is relatively unique in that it uses high-resolution T1-weighted scans which are usually reserved for academic research and analysis of long-term damage and rehabilitation. This stands in contrast to the ISLES data set, which does not have T1-weighted scans. The compilers of ATLAS do not expect techniques imported from the study of ISLES to perform optimally.

IV. BASELINE ENCODER/DECODER NETWORK

As a point of comparison for DualNetFC, we introduct a baseline model which we will refer to at the “Encoder/Decoder Network”. In our code we refer it as the Atlas Model (and to avoid confusion from the data set and

*This work was not supported by any organization

[1] Department of Mathematics, Stanford University, SUNetID: westonu

[2] Department of Mathematics, Stanford University, SUNetID: jupiterz

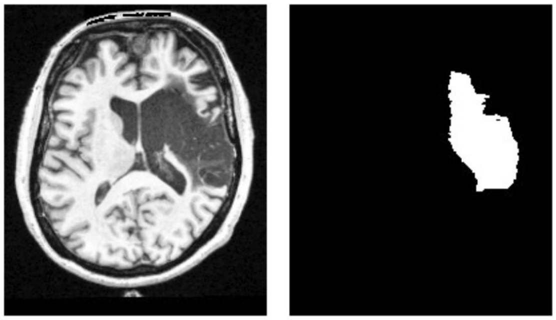

Fig. II. A sample two-dimensional slice of an MRI scan in the ATLAS dataset along with its segmentation label.

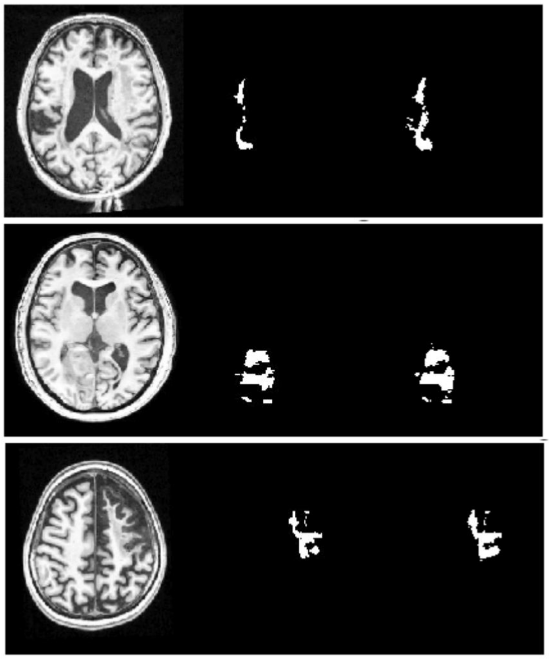

Fig. III. Some sample outputs for DualNetFC. In each image, the left is the input scan, the middle is the target mask, and the right is the predicted mask.

the organization, which have the same names, we will call it the"Encoder/Decoder Network" in this report). This is a simple feed-forward network which takes in a full two-dimensional slice, passes it through $3 \times (3 \times 3$ convolution, 2×2 max-pooling), 3 fully connected layers of sizes 1024, 256, and 1024, and finally $2 \times (2 \times 2$ upsampling, 3×3 deconvolution). The Sørensen-Dice coefficient of this model converges to 0.40 when trained *by_slice* and 0.16 when trained *by_scan* (see the *Training* section for details on these methods). This model has 93,705,297 parameters. We will use these scores as our baseline performance for comparison. We note that the performance of this baseline model decreases sharply when the 1024-node fully connected layers are decreased in size. This stands in contrast to the analysis of DualNetRC256 detailed in the *Experiments* section below.

V. DualNetFC Design and Implementation

A. Architecture

DualNetFC processes a 232×196 input as follows (Fig. IV.). A series of twelve 100×100 "blue" crops are taken from the image, and from each the central 50×50 "green" crop is extracted. The crops are then processed as follows:

1) Each "blue"/"green" pair is fed to the network in succession.
2) The "green" crop is passed through the upper pathway of the network, which consists of 7 convolutional layers (all of which have 3×3 filters and "SAME" padding).
3) The corresponding "blue" crops are passed through the lower pathway of the network, which consists of $2 \times ((2 \times 2)$ pooling, then 3×3 convolution), four fully-connected layers with dropout of sizes 512, 128, 512, and $25*25*30$, upsampled to $50*50*30$, passed through $2 \times (2 \times 2$ deconvolutional layer).
4) The upper and lower pathways are then concatenated along the channel direction and passed through three 1×1 convolutions. This output is then zero-padded so that the precicted mask lines up with the location of the input crops in the input image.
5) Predictions for each slice are then simply added together and passed through a pixel-wise sigmoid function.
6) The final mask is generated by taking those pixels for which the network has predicted a probability > 0.5.

B. Objective Function and Accuracy

The loss function here is a weighted pixel-wise cross-entropy:

$$\mathscr{L} = -\sum_{p \ \in \ \text{pixels}} 5\hat{p}\log(p) + (1-\hat{p})\log(1-p)$$

Accuracy in image segmentation is usually measured with respect to the Sørensen-Dice coefficient (SDC) of a prediction-target pair, rather than just from the loss function. For two masks X, Y this coefficient is given by:

$$SDC(X,Y) = \frac{\text{Area of } \ X \cap Y}{\text{Area of } \ X \cup Y} \in [0,1].$$

This measure of accuracy takes into account both false positives and false negatives.

C. Training

There are two distinct ways of dividing the ATLAS dataset into training/dev sets:

1) In a *by_scan* split, the 229 three-dimensional scans are divided into training/dev sets, which are then processed in two-dimensional slices. Thus, during training the network does not see slices from any of the scans present in the development set
2) In a *by_slice* split, the 43,281 two-dimensional slices are shuffled and divided into training/dev sets. In this

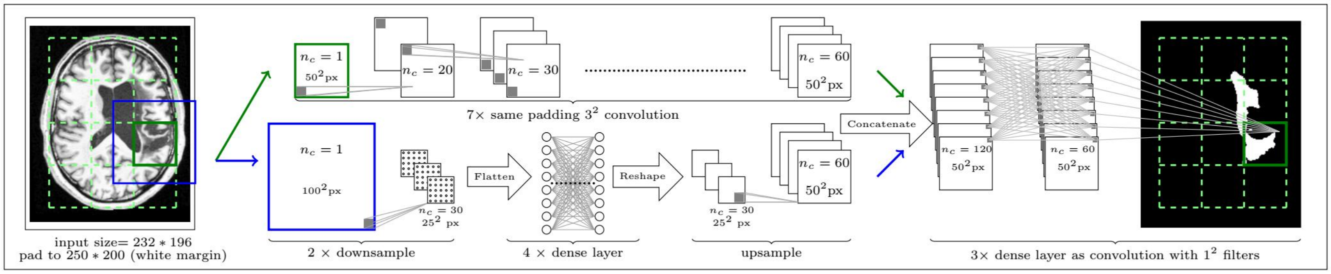

Fig. IV. The input is sliced into 12 green boxes according to the dashed green grid. Each green box is 50^2 pixels, centered by a 100^2 pixels blue box (possibly overlapping with other blue boxes). The model neglects a $16*23$ margin from the input, which mostly contains no useful data. The green boxes are passed through the top path, which aims to scrutinize the detailed contours from the local information (thus no max pooling is used). On the other hand blue boxes provide contextual information for their central green boxes through the bottom path. The output from each pair of green and blue box will be a 50^2 pixels box, located exactly where the input green box is. The final prediction results from patching all these boxes together with necessary paddings.

case, the network does train on slices from scans placed in the development set.

It is clear that a model trained *by_scan* should be expected to have worse performance than one trained *by_slice*; the latter offers the network the opportunity to memorize the training data and apply that knowledge directly to the development set, as many of the cross-sections will look similar. We trained separate models *by_scan* and *by_slice*.

In both cases, we allocated 30% of the data to a development set, yielding a train/dev split of 30197/12984. The network was trained on a single GPU and converged in approximately 60 epochs, each of which took approximately 6 minutes to run. We employed Adam optimization with the standard parameter values, a dropout rate of 0.15, and the Xavier initialization. At early plateaus and dips in performance we implemented learning rate decay with starting learning rate 0,001, cutting the learning rate in half at each application. We stopped when the model appeared to converge.

D. Hyperparameter choices

There were various hyperparameters in our network architecture to optimize. The most notable were the following:

1) The *pos_weight* multiplier weights the first term in the loss function. (It is 5 in our final model, as listed above.) A higher *pos_weight* encourages the network to make positive predictions, which effectively helps to balance class sizes between positive and negative. This imbalance is a priori the heart of this segmentation problem as most lesions take up only a small portion of the given MRI slice. We tested *pos_weight* values of 100, 50, 25, 5 and 1.
2) The size of the largest fully connected layers has a strong impact on performance. In our final implementation this layer has 512 nodes, but we tested both 1024 and 256 as well. The 1024 model performs exceptionally poorly, but the 256 model displays interesting behavior, explored in the *Experiments* section below.

Network	Training Method	SDC
Encoder/Decoder	*by_slice*	0.40
DualNetFC	*by_slice*	0.38
Encoder/Decoder	*by_scan*	0.16
DualNetFC	*by_scan*	0.21

Fig. V. The Sørensen-Dice coefficients for DualNetFC and the baseline Encoder/Decoder network when trained *by_slice* and *by_scan*.

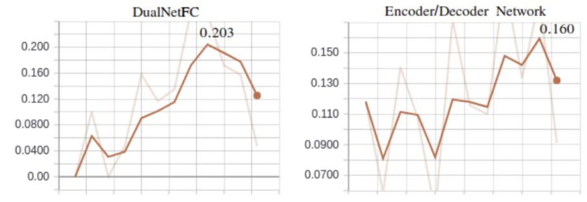

Fig. VI. Plots of the Sørensen-Dice coefficient vs iteration for DualNetFC and the standard Encoder/Decoder network

E. Results

In Fig. V. we have compiled the Dice coefficients for DualNetFC and the baseline Encode/Decoder network, both trained *by_slice* and *by_scan*. We see here that while the Encoder/Decoder network attains slightly better performance when trained *by_slice*, DualNetFC performs better when trained *by_scan*, which is the significantly harder and more practical task. This is somewhat surprising given that DualNetFC is about $1/5$ the size of the Encoder/Decoder network.

F. Discussion

Due to the constrained size of the dataset, we did not use a test set. We believe that this severely limits the scope of our results. While containing $43,281$ horizontal slices, the ATLAS dataset only contains 229 complete scans. Each scan contains 188 horizontal slices. Within a scan, those slices that contain a lesion typically contain strongly correlated lesion masks. When trained *by_slice*, we suspect that our network simply memorizes the shapes of these 229 different lesions and the bottom pathway provides context for which memorized shape to choose from. While this is a bit of a simplification of what is actually happening, the dramatic decrease in performance of our model when we move to *by_scan* training suggests that some amount of this sort of memorization occurs. It is unclear if this is the fault of the network architecture or simply due to the constrained data size. Note that the same problem is present in the baseline Encoder/Decoder Network to a much greater extent.

A concern which can be more directly attributed to the architecture is the fact that our model severely underfits the training dataset. The Sørensen-Dice coefficient on the training set converges to approximately 0.60. Even when trained on a small selection of 200 examples, this coefficient stays below 0.80 even after several hundred epochs. Similarly, the loss on the training set converges to 0.24. All of the other models in the *Experiments* section below converge to this value as well, which suggests that the network architecture is at fault.

VI. Experiments

The network detailed above was just one in a series of experiments we performed with the dual-pathway architecture. We considered the following alternate architectures for the two pathways. Graphs of the Sørensen-Dice Coefficient for each are shown in Fig. VIII.

1) **DualNetVeryFC**: This model introduces four smaller fully connected layers in the middle of the upper pathway. This model was so large that it was essentially unable to converge.
2) **DualNetBigBlue**: This model passes the entire image through the lower pathway, rather than take local crops. The sharp decrease in the Sørensen-Dice coefficient suggests that DualNetFC does use the upper pathway and is learning from the local croppings.
3) **DualNetOverlap**: This model most directly resembles the one implemented in [3]. It uses 64×64 crops for the "green" window which are passed through convolutional layers using "VALID" padding, so that the final output size at concatenation was still 50×50. This has the effect of overlapping the regions considered by the different croppings so that there is an extra layer of context for the network. Interestingly, this also saw a decrease in performance. It seems that the introduction of any sort of pooling in the upper pathway (whether that be via honest pooling layers or though convolutional layers with "valid" padding) forgets too much information and make accurate segmentation difficult.
4) **DualNetFC256**: This model is identical to DualNetFC except that the largest fully connected layers now only have 256 nodes rather than 512. This effectively decreases the number of parameters in the network by a factor of two. The plots in Fig. VIII. show that this model achieves a high SDC when trained *by_slice*. Unfortunately, when trained *by_scan*, this model achieves a DSC of near zero and seems to be guessing randomly. We suspect that the smaller fully connected layers are bottle-necking the model and forcing it to memorize the data. This renders it useless when training *by_scan*.

Network	Maximum SDC	Num. Parameters
Encoder/Decoder	0.40	93,705,297
DualNetFC	0.38	19,469,835
DualNetVeryFC	0.26	45,233,281
DualNetBigBlue	0.20	39,064,203
DualNetOverlap	0.31	19,592,925

Fig. VII. Encoder/decoder is a simple conv/fc/deconv, DualNetVeryFC has fully connected layers in the upper pathway, DualNetBigBlue passes the entire input as the blue crop, and DualNetOverlap overlaps the green crops.

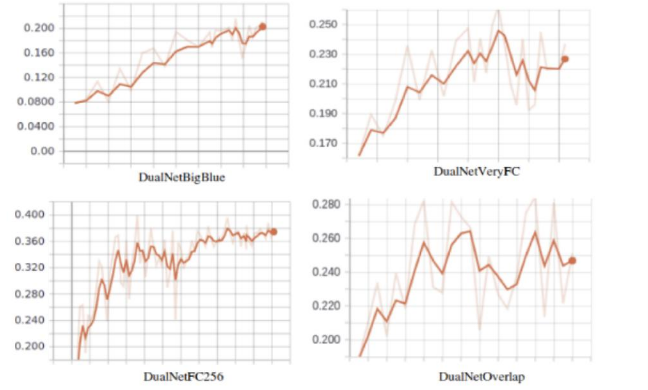

Fig. VIII. Plots of the Sørensen-Dice coefficient vs iteration for each of our experimental models when trained *by_slice*. Note that only DualNetFC256 achieves performance at the level of DualNetFC.

VII. Conclusions/Future Directions

While there is still much work to be done analyzing the ATLAS dataset, we believe that our discussion of dual-pathway networks as least begins to explore their possible application here. The success of DualNetFC over the Encoder/Decoder network when trained *by_scan* offers at least some evidence that this approach is worth pursuing. A natural next step would be to continue following the methods of [3] and employ a three-dimensional variant of DualNetFC which operates on small crops of an entire brain scan, rather than just one two dimensional slices. This will, however, even further exacerbate the problems that we have already experienced regarding the small size of this dataset. One possible solution would be to do some sort of transfer learning directly from the model in [3], as they achieve a very high DSC of approximately 0.60 when training their model. An improvement using this approach would be interesting given the qualitative difference between the T1-weighted MRI scans in ATLAS and the unweighted ones in the ISLES dataset that they train on.

Code

The code for this project is available at `https://github.com/wungemach/atlas`.

Acknowledgment

We would like to thank our mentor, David Eng, for his guidance throughout this project.

References

[1] Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig. A brain tumor segmentation framework based on outlier detection. *Medical Image Analysis*, 8(3):275 – 283, 2004. Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003.

[2] B. H. Menze, A. Jakab, and S. Bauer et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, Oct 2015.

[3] Konstantinos Kamnitsas, Christian Ledig, and Virginia F. J. Newcombe et al. Efficient multi-scale 3d CNN with fully connected CRF for accurate brain lesion segmentation. *CoRR*, abs/1603.05959, 2016.

[4] Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig. A brain tumor segmentation framework based on outlier detection*1. 8:275–83, 10 2004.

[5] Ezequiel Geremia, Bjoern H. Menze, Olivier Clatz, Ender Konukoglu, Antonio Criminisi, and Nicholas Ayache. Spatial decision forests for ms lesion segmentation in multi-channel mr images. In Tianzi Jiang, Nassir Navab, Josien P. W. Pluim, and Max A. Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, pages 111–118, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[6] Yann Lecun, Leon Bottou, Y Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. 86:2278 – 2324, 12 1998.

[7] Rongjian Li, Wenlu Zhang, Heung Il Suk, L. Wang, Jiang Li, Dinggang Shen, and Shuiwang Ji. Deep learning based imaging data completion for improved brain disease diagnosis. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 17(Pt 3):305–312, 2014.