Using cascaded networks for post-stroke lesion detection in the ATLAS dataset

Diana D. Chin
Mechanical Engineering
Stanford University
ddchin@stanford.edu

William R. T. Roderick
Mechanical Engineering
Stanford University
wrtr@stanford.edu

Karen M. Wang
Mechanical Engineering
Stanford University
kmwang14@stanford.edu

Abstract

Post-stroke lesion detection is a process that currently takes skilled tracers up to an hour per scan. Automating lesion detection will not only help radiologists catch problematic lesions in many clinical domains, but will also enable us to rapidly expand neuroimaging datasets, which can then be used to improve our understanding of how MRI brain scans relate to recovery prognoses and suitable treatments. In this work, we test deep learning methodologies used in prior medical image segmentation studies on USC's new Anatomical Tracings of Lesions After Stroke (ATLAS) dataset. Given a series of MRI slices of the brain as input, our model predicts segmentation masks identifying the locations of post-stroke lesions as output. We demonstrate how a U-Net architecture, applying dilation, or using Gaussian blurring are relatively ineffective for improving the dice coefficient of our predictions, while the greatest performance can be derived by cascading an encoding/decoding neural network architecture.

1 Introduction

We are developing a deep learning diagnosis system for the USC Anatomical Tracings of Lesions After Stroke (ATLAS) dataset [1]. Segmentation of brain lesions is an inherently difficult task, which currently requires a manual process that takes skilled tracers up to an hour per scan [1]. Automating post-stroke lesion identification will enable researchers to gather much larger neuroimaging datasets, which can then be used in improving our understanding of how MRI brain scans relate to recovery prognoses and/or suitable treatments. Additionally, the machine learning methodology developed for this purpose can be generalized to improve the accuracy and efficiency of detecting abnormalities in other volumetric medical imaging data.

Our algorithm takes as input 2D MRI brain scans of height and width 232x196 pixels. We then use a neural network to output a segmentation mask of the same size identifying the locations of post-stroke lesions.

2 Related work

The ATLAS dataset was only released a few months ago, so there are no published research papers that use it. However, prior work has been done on lesion segmentation using MRI scans with various techniques, including U-Net [2], cascading [3], and other statistical algorithms and machine learning methods [4, 5, 6]. It is challenging to compare the performance of the different methods, because the different works use different performance metrics and different datasets that can vary greatly in difficulty. Despite the 3D nature of MRI scans, most work on lesion segmentation focuses on 2D images (or fusing outputs on 2D images into 3D volumes) because of the high computational cost of volumetric segmentation. Some studies have also explored dilation techniques [3] or Gaussian blurring on T1 and T2 MRI images as a preprocessing method for removing noise to retain the most

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

important details of an image [7, 8]. In this paper, we build on previous work in deep learning and compare different neural network architectures involving a U-Net [2] and a cascaded network [3]. In addition, we compare our results with and without dilation and a simple Gaussian blurring technique.

3 Dataset and Features

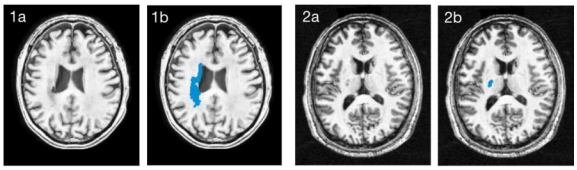

Figure 1: Examples of two pairs of MRI slices (1a,2a) with the corresponding lesion mask overlaid in blue (1b,2b) from the ATLAS dataset.

The ATLAS (Anatomical Tracings of Lesions After Stroke) dataset includes 229 T1-weighted MRI scans (from n=220 patients from 9 different sites) with segmented lesions, and is publicly available at `http://fcon_1000.projects.nitrc.org/indi/retro/atlas.html`. Each scan includes a series of MRI slices and one or more series of lesion masks. Some examples from the dataset are shown in fig 1. These MRI scans are split and shuffled randomly by slice such that there is the same distribution of slices in both the training and dev sets. The training set consists of 12,984 examples and the dev set consists of 1000 examples, which is a $93\%/7\%$ split. The original images are converted to Numpy arrays of pixel intensity values, which are then normalized to values between 0 and 1 to be input into the neural network. Each brain scan is 232x196 pixels, which is a total of 45,472 features per image. We chose not to introduce any additional features in this project in order to assess the ability of our network to detect lesions based only on the original MRI scans.

4 Methods [1]

4.1 ATLAS Model

Our mentor, David Eng, provided us with baseline code to become familiar with the dataset and learn about the challenges of the task. The baseline ATLAS Model, shown in fig 2, consists of a convolutional encoder (conv-pool-drop-conv-pool-drop-fc-drop-fc), followed by a deconvolutional decoder (fc-drop-fc-drop-up-deconv-up-deconv).

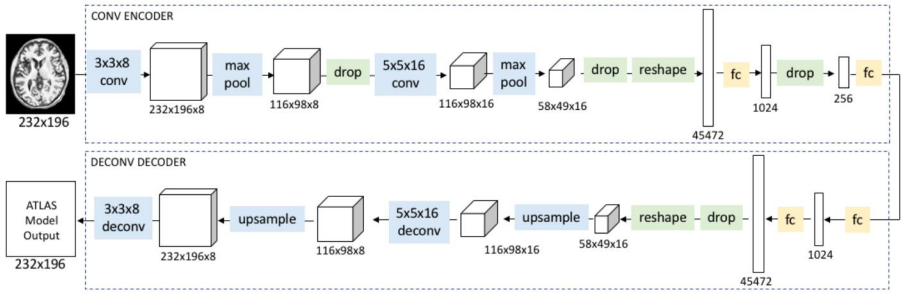

Figure 2: The encoder and decoder architecture in the baseline ATLAS Model.

4.2 U-Net Model

The U-Net architecture, based on [2], is shown in fig 3. We also reduced the network size in a "medium" version, by reducing the number of filters by a factor of 4, and a "small" version, which additionally replaced the 2x2 max pools with one 4x4 max pool and the 2x2 up-conv steps with one 4x4 up-conv.

4.3 Cascaded ATLAS Model

The cascaded network architecture is shown in fig 4 for a twice cascaded system. In each cascade, the next network is trained on the original input images masked with the previous network's output.

[1] All of the models were implemented with Tensorflow [9] and use pixel-wise weighted cross-entropy loss.

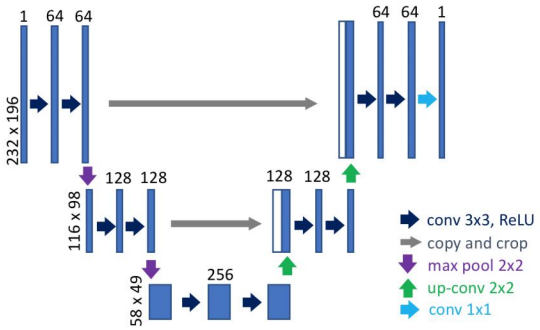

Figure 3: The U-Net architecture, figure adapted from [2].

During testing, the input is a raw MRI scan, which we found worked better than using the images that were fed through the previous network(s). By choosing the relative weighting in the cross entropy loss to strongly favor recall and reducing the weighting at each cascade, each succeeding network effectively trains on a smaller and smaller region of each image. The Atlas Model performed significantly better than the U-Net for the same number of epochs, so we chose to implement the Atlas Model in each cascaded network. We also tried dilating the output of the 1st network before masking, and blurring the masked images input into the second network, as shown in fig 4.

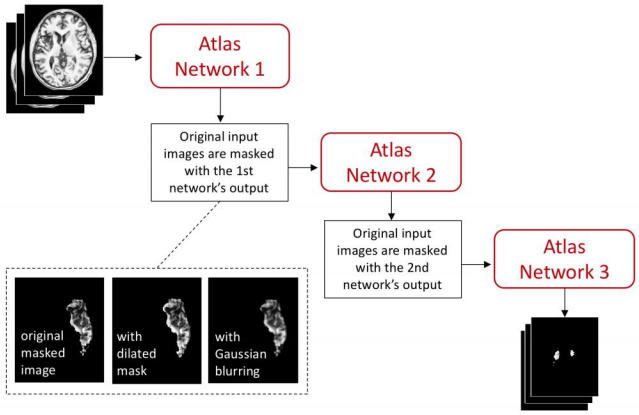

Figure 4: The cascaded network architecture, shown here for a twice cascaded system. Inset: Examples of images input into the 2nd network after being masked with the output of the 1st network.

5 Experiments

5.1 Hyperparameters

The primary hyperparameter that we tuned is the relative weighting of recall to precision in the cross entropy loss. A more detailed discussion of tuning this parameter will be in the results section. A batch size of 100 worked well and was a reasonable size for fast learning. Gradient norms above 5.0 were clipped to this maximum value in order to prevent the exploding gradient problem. The dropout value, which sets the fraction of units randomly dropped on non-recurrent connections, was set to 0.15 for a slight regularizing effect to prevent overfitting. The Adam optimizer was chosen for its ability to efficiently solve a broad array of deep learning problems, and its parameters (learning rate, β_1, β_2, ϵ) were left at their default settings [10] (summarized in the following table).

Hyperparameter	Value
Positive Weighting	Varied (1, 25, 50, 75, 100)
Batch size	100
Maximum gradient norm	5.0
Dropout while training	0.15
Optimizer	Adam ($lr = 10^{-3}$, $\beta_1 = 0.9$, $\beta_1 = 0.999$,$\epsilon = 10^{-8}$)

5.2 Performance Metrics

To evaluate the performance of our algorithm, we use the dice coefficient ($DICE$), a commonly used metric for assessing segmentations of medical volumetric data [11]. From [12], the definition of the dice coefficient is:

$$DICE = \frac{2TP}{2TP + FP + FN}$$

where TP, FP, and FN are true positive, false positive, and false negative pixel counts of the predicted lesion masks as compared to the ground-truth lesion masks. This coefficient approaches 1 with more correctly predicted lesion mask pixels (TP), but decreases if the algorithm predicts a lesion where one does not exist (FP) or does not identify a lesion where one does exist (FN).

While the $DICE$ score was the primary metric we were optimizing with network design decisions, we also considered other metrics that could be relevant, especially in a clinical setting. Recall and precision on a pixel per pixel basis were defined as $Recall_{pix} = \frac{TP}{TP+FN}$ and $Prec_{pix} = \frac{TP}{TP+FP}$,respectively. Furthermore, $Recall_{img}$ is the number of images where the predicted mask overlaps with the target divided by the number of images where there is a target. $Prec_{img}$ is the number of images the predicted mask overlaps the target divided by the number of images where we predicted a mask. We report these four metrics and discuss their relevance in the following section.

6 Results and Discussion

A summary of the dev set results for all of the tested models are shown below. Training set DICE scores (based on 100 random samples) are shown in parentheses.

Model	$DICE$	$Recall_{pix}$	$Prec_{pix}$	$Recall_{img}$	$Prec_{img}$
ATLAS Model (baseline, 20 epochs)	0.35 (0.46)	0.72	0.31	0.95	0.64
U-Net, small (3 epochs)	0.017 (0.020)	0.49	0.010	0.80	0.20
U-Net, medium (3 epochs)	0.024 (0.030)	0.36	0.013	0.87	0.23
U-Net, large (3 epochs)	0.028 (0.071)	0.29	0.020	0.63	0.16
Single Cascaded Atlas (20 epochs/network)	0.42 (0.55)	0.58	0.48	0.92	0.74
Single Cascaded Atlas + dilation (10 epochs/network)	0.34 (0.39)	0.62	0.31	0.83	0.76
Single Cascaded Atlas + gaussian blur (10 epochs/network)	0.35 (0.47)	0.71	0.31	0.94	0.62
Double Cascaded Atlas + dilation (10 epochs/network)	0.36 (0.44)	0.56	0.36	0.82	0.75
Double Cascaded Atlas (20 epochs/network)	0.44 (0.58)	0.55	0.51	0.89	0.77

Cascading the Atlas model successfully improved segmentation performance, which was expected because the cascaded structure simplifies the task for each network, making the networks easier to train. Further, we found that cascading two networks performed better than just training a single network for the same total amount of time. However, additional cascades produced diminishing returns, as shown in fig 5 (left). This is expected since if the network predicts a false negative, each cascade would continue to mispredict that image. Dilating was likely less helpful because the relative recall/precision weightings in the loss function of the first network favored false positives, so the predicted mask was generally already larger than the target. Blurring may have been ineffective because it leads to the loss of potentially useful information. As expected, running more epochs also initially increased the DICE score, but only for 45 epochs, and with diminishing returns after about 20, as shown in fig 5 (middle). The U-Net performed worse than expected as it appeared to only be thresholding, but we suspect that it would improve after much more training.

We also noted a trade-off between recall and precision (defined above), which were adjusted by changing the weighting in our cross entropy loss function (see fig 5 right). Tuning this weighting also affected the resulting DICE score. In a real clinical application, the image level recall may be particularly useful for a doctor since the doctor would likely check the algorithm's output regardless of the next steps for the patient. The doctor would want to be confident that the algorithm is catching a very high percentage of the lesions and outputting masks that at least overlap them.

Finally, to gain insight into what image features were the most influential in our network, we generated saliency maps, which illustrate the gradient of the predicted mask probabilities with respect to the input image pixels (see 6 left). High gradients, represented by brighter pixels in the saliency map,

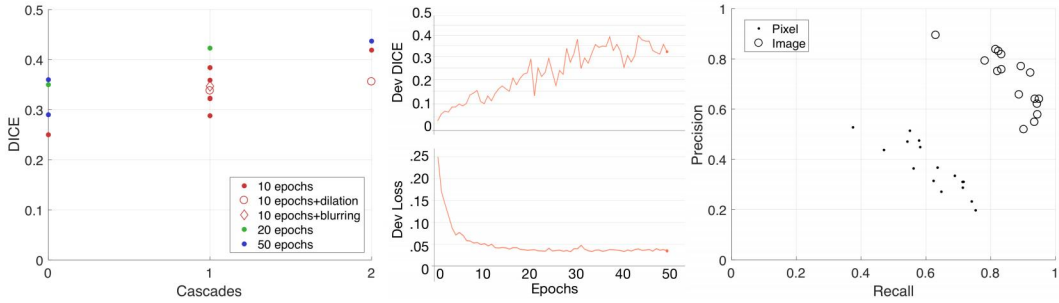

Figure 5: Left: Dice score as a function of the number of cascades for all of the models. Middle: The DICE score and loss as a function of the number of epochs trained for the baseline Atlas model. Right: The tradeoff between precision and recall.

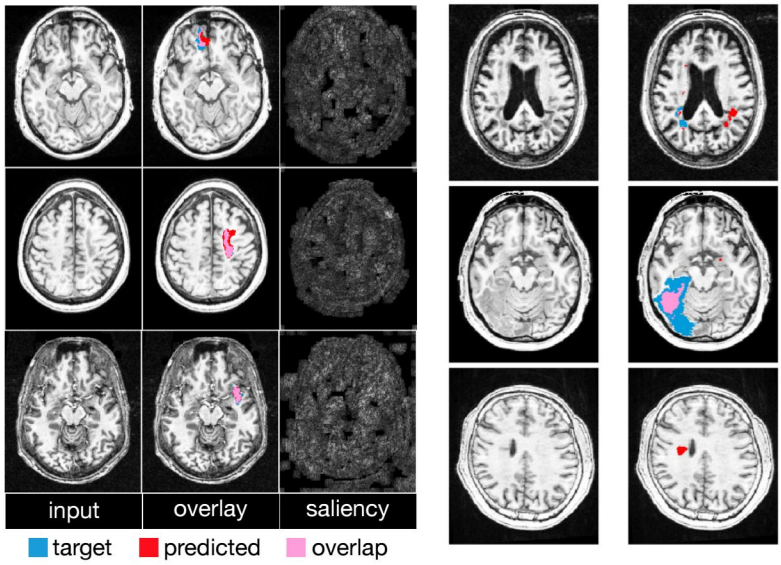

Figure 6: Left: Examples of inputs, outputs, and saliency maps. Right: Examples of mispredicted results.

indicate the importance of that input pixel in determining the output probabilities. The most salient pixels seem to cluster around gray and high contrast areas.

It makes sense, then, that mispredictions tend to occur most often when lesions are a lighter gray color, appearing more similar to healthy brain folds (see fig 6). Large lesions were segmented with the best accuracy/precision, likely because they tend to be darker and easier to delineate from healthy tissue. For small and medium lesions, the algorithm usually gets an area larger than the target.

7 Conclusion and Future Work

Out of all the methods we tried, we found that cascading the Atlas model twice yielded the best segmentation performance. However, the difference between the training and dev set performances indicate that the model has high variance, so a promising next step would be to use more training data to resolve this discrepancy. Additional regularization and further tuning the hyperparameters would also likely help. To improve the training and dev performance further, we would recommend training a larger model, exploring volumetric segmentation models from sparsely labeled images, or incorporating lesion metadata (ex. primary stroke location and hemisphere or vascular territory).

8 Code

Our code is available at: `https://github.com/wroderick/atlas.git`

9 Contributions

All team members contributed to generating ideas, editing the code, running experiments with the different models, and documenting our work.

References

[1] S.-L. Liew, J. M. Anglin, N. W. Banks, M. Sondag, K. L. Ito, H. Kim, J. Chan, J. Ito, C. Jung, N. Khoshab, S. Lefebvre, W. Nakamura, D. Saldana, A. Schmiesing, C. Tran, D. Vo, T. Ard, P. Heydari, B. Kim, L. Aziz-Zadeh, S. C. Cramer, J. Liu, S. Soekadar, J.-E. Nordvik, L. T. Westlye, J. Wang, C. Winstein, C. Yu, L. Ai, B. Koo, R. C. Craddock, M. Miham, M. Lakich, A. Pienta, and A. Stroud, "A large, open source dataset of stroke anatomical brain images and manual lesion segmentations," *bioRxiv*, 2017.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[3] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks," *CoRR*, vol. abs/1709.00382, 2017. [Online]. Available: http://arxiv.org/abs/1709.00382

[4] J. Cai, Y. Tang, L. Lu, A. P. Harrison, K. Yan, J. Xiao, L. Yang, and R. M. Summers, "Accurate weakly supervised deep lesion segmentation on ct scans: Self-paced 3d mask generation from recist," *arXiv preprint arXiv:1801.08614*, 2018.

[5] S. Jain, D. M. Sima, A. Ribbens, M. Cambron, A. Maertens, W. Van Hecke, J. De Mey, F. Barkhof, M. D. Steenwijk, M. Daams *et al.*, "Automatic segmentation and volumetry of multiple sclerosis brain lesions from mr images," *NeuroImage: Clinical*, vol. 8, pp. 367–375, 2015.

[6] S. González-Villà, S. Valverde, M. Cabezas, D. Pareto, J. C. Vilanova, L. Ramió-Torrentà, À. Rovira, A. Oliver, and X. Lladó, "Evaluating the effect of multiple sclerosis lesions on automatic brain structure segmentation," *NeuroImage: Clinical*, vol. 15, pp. 228–238, 2017.

[7] L. Spies, A. Tewes, P. Suppa, R. Opfer, R. Buchert, G. Winkler, and A. Raji, "Fully automatic detection of deep white matter t1 hypointense lesions in multiple sclerosis," *Physics in Medicine and Biology*, vol. 58, pp. 8323–8337, 2013.

[8] H.-J. Huppertz, C. Grim, S. Fauser, J. Kassubek, I. Mader, A. Hochmuth, J. Spreer, and A. Schulze-Bonhage, "Enhanced visualization of blurred gray–white matter junctions in focal cortical dysplasia by voxel-based 3d mri analysis," *Epilepsy Research*, vol. 67, pp. 35–50, 2005.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[10] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015.

[11] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool," *BMC medical imaging*, vol. 15, no. 1, p. 29, 2015.

[12] D. Eng, "Cs 230: Atlas dataset project description," 2018. [Online]. Available: https://docs.google.com/document/d/1qdTKDv3g203rFcSKaGsfPBLuPI30QSTu_rFTgWJvMfI/