Automated Vessel Segmentation in MRI Data

Quarter 2 Project

Medha Pappula and Kade Yen

Thomas Jefferson High School for Science and Technology

Machine Learning 1

Dr. Yilmaz

January 15, 2025

Abstract

The accurate segmentation of blood vessels in the brain is a critical task with significant implications for understanding neurovascular health and disease. This paper addresses the challenge of automatically identifying blood vessels in 3D Magnetic Resonance Imaging (MRI) data, which is traditionally performed manually by experts. This is both time-consuming and prone to variability, impacting results. Our motivation stems from the potential of such automated methods to improve diagnostics, inform prevention strategies, and reduce healthcare costs associated with cardiovascular diseases. We employ a U-Net architecture, an advanced form of convolutional neural network (CNN) designed for image segmentation tasks, to analyze MRI images and identify blood vessels. Our methodology includes processing 3D MRI volumes into smaller sub-volumes to enhance computational efficiency and training a U-Net model on a curated dataset of these sub-volumes. The results obtained demonstrate that our U-Net architecture outperforms other models used in prior research for this task, achieving a higher Dice score (0.681) and lower false positive (0.194) and false negative (0.289) rates.

Introduction

Vessel segmentation, the process of identifying and delineating blood vessels from medical images, is a challenging task within the medical image analysis space. This task is a fundamental step that supports a wide array of medical applications. Accurate segmentation allows medical professionals to gain critical insights into the structure and function of the vascular system, responsible for the transport of oxygen and nutrients throughout the body. Disruptions in these networks can indicate or even contribute to a wide range of diseases. We were motivated by:

- *Clinical Need:* Manual segmentation of blood vessels from MRI data is a labor-intensive, time consuming process requiring specialized training, making it difficult to scale.
- *Research Advancement:* Researchers can use this automated tool for larger scale studies of neurovascular diseases, analyzing vascular networks of many subjects and looking at the effect of vascular pathology across the whole brain instead of a small section.
- *Reduced Healthcare Costs and Personalized Medicine:* Automating this process can reduce costs for specialized personnel and also quickly personalize any treatment plans.

The inputs to our algorithm are 3D MRI volumes represented in the *.nii* file format, which consists of a grid of voxels, each pixel representing .2 microns of information. The output of our U-Net is a predicted segmentation mask, a 3D volume the same size as the input. The mask colors each voxel as either part of a blood vessel or not. Thus, our task is a binary segmentation problem, where each voxel is classified into one of two categories: vessel or non-vessel.

Previous Work

Prior studies have explored various approaches to blood vessel segmentation, which is typically done by hand. Below is a summary of previous works, looking at accuracy and limitations of each study. Many used structural modeling techniques, other ML techniques had variable accuracy, from supervised CNN, FCNN to unsupervised K-means for segmentation.

Type	Source	Accuracy	Limitations
K means	Gehad Hassan	0.75	Dataset image quality was high, ungeneralizable to global image datasets. Only Diabetic Retinopathy.
CNN	LMBiS-net	0.7873	Low number of learnable parameters (0.172 million)
Math Modeling	Stanford vessel segment	0.7832	higher sensitivity (misses fewer small vessels) at the cost of a lower precision (more false positives)
Structural Modeling	IterNET	0.726	Low true positive rate of 0.34
Math Modeling	Cross-modality learning approach	0.7726	Algorithms can generate spurious results (dots).
Full-Connect NN	Stanford vessel segment	0.7047	Small percentage (0.61%) of pixels in the training images

Dataset Description

This project uses a dataset derived from MRI scans. The original dataset consists of 600 3D images converted into the *.nii* format, which is a common format for medical imaging data. These images are either 128 or 256 voxel cubes, each voxel being .2 microns. These voxels could be considered as 'pixels' in a 3D image. To reduce computational complexity and time, these larger images were split into smaller 64-voxel cube images. Out of a total of around 2000 64 voxel cubes, we selected 1000 based on the clarity of image, amount of vessels we can see, amount of noise in the image, and amount of variability (good representability of all features).

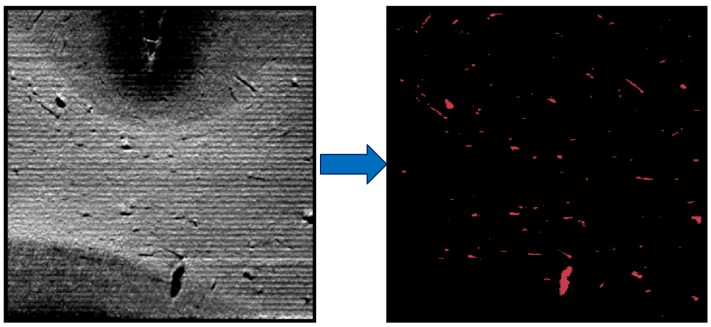
```
'shape': (128, 128, 128),
'voxel_size': 0.02,
'nb_levels': RandInt(1, 4),
'tree_density': Uniform(0.1, 0.2),
'tortuosity': Uniform(1, 5),
'radius': Uniform(0.01, 0.1),
'radius_change': Uniform(0.8, 1.2),
'nb_children': RandInt(1, 4),
'radius_ratio': Uniform(0.25, 1),
'device': 'cuda'
```

In developing the model, we give the original image, the mask, and these attributes. In previous studies like IterNET, these attributes are primarily used for structural analysis. In our method, we will include, but not place as much importance on them.

Feature	Definition
nb_levels	The amount of parent vessels / starting vessels the image has.
tree_density	The starting thickness of the parent vessel.
tortuosity	The amount of curl a blood vessel has / how spirally a blood vessel is.
radius	The thickness of the blood vessel.
radius_change	The change in radius from a parent blood vessel to a branching blood vessel.
nb_children	The amount of branching blood vessels that a parent blood vessel has.
radius_ratio	The change in radius in the blood vessel throughout its length.

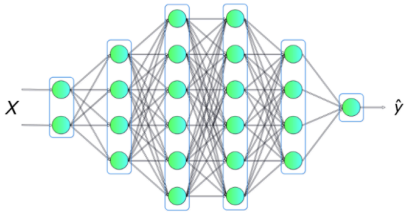
This data was then split into 80% testing and 20% validation. The 80% testing was further split into 80% training and 20% testing using stratified random sampling.

An example MRI scan, before and after manual segmentation.



Methods

Our focus is to implement a U-Net algorithm, an improvement on CNNs and compare it to both CNN and FCNN implementations seen in Matloob Abbasi et al. and Chollet et al. respectively.



Fully Connected Neural Networks (FCNNs)

Chollet et al. used FCNNs for this task. A FCNN is when each neuron is connected to every neuron in the previous layer. Given input vector, weights, and biases, the output is a non-linear activation function such as ReLU. FCNNs are effective for learning global patterns but struggle with data that exhibits strong spatial locality, such as images.

Convolutional Neural Networks (CNNs)

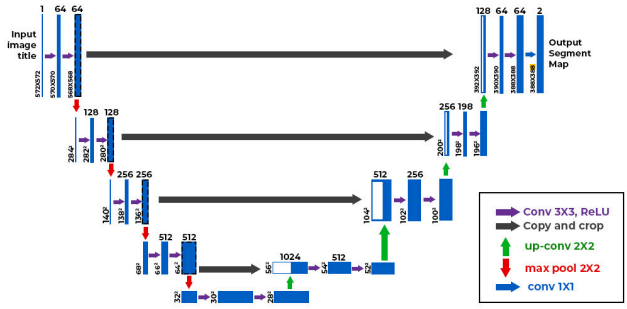
Matloob Abbasi et al. used CNNs by using convolutional layers that exploit local spatial structure in data. A single layer applies a set of learnable filters over the input, producing feature maps. For an input image and filter, the convolution operation (see right) reduces the number of trainable parameters by sharing weights across different spatial locations, increasing efficiency. Pooling layers, such as max-pooling, further reduce dimensionality and introduce translational invariance.

$$S_1 = \sum_{k=1}^{3} \beta\left(\mathrm{ReLU}\left(f^{n\times n}\left(I_{in}\right)\right)\right)$$

(1)

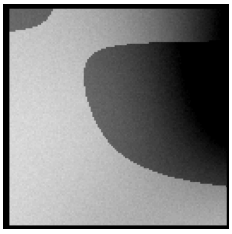
where β is the batch normalisation operation, ReLU is the activation function, $f^{n\times n}$ is the kernel size convolution operation $n \times n$, and $n = 2k - 1$.

$$I_{out} = \sum_{k=1}^{3} \beta\left(\mathrm{ReLU}\left(f^{n\times n}\left(S_1\right)\right)\right)$$



U-Nets: Extending CNNs

Our method uses U-Nets to build on top of CNNs, specifically for image segmentation tasks, by using a symmetric encoder-decoder structure with skip connections. The encoder extracts hierarchical features while progressively reducing spatial resolution, such as the size of a blood vessel or the amount of blood vessels.



After each feature extraction, the image resolution is decreased. Each pixel with an averaged value of nearby pixels. The decoder then samples the feature maps to reconstruct a segmented image. Skip connections retain fine-grained details by merging features from earlier layers (see right/left), keeping smaller information safe. We predict that this architecture will significantly improve segmentation accuracy.

For training, we had these parameters:

- **Number of Volumes:** 1000 preprocessed 64 voxel cube volumes were used for training
- **Training to Validation Split:** The data was split into 80% training (80% training and 20% testing) and 20% validation set for results.
- **Training Steps:** The model was trained for 100,000 steps, this was to ensure that the model was fine-tuned to the intricacies of the vessels. Previous runs showed that the model was typically underfitting to the data.
- **Batch Size:** A batch size of 1 was used, processing one image at a time during training. A batch size of 1 may be slower, but it allows for faster parameter updates and may be more robust for optimization.
- **Learning Rate:** A learning rate of 0.001 was used. This was decided due to the complexity of the data, hoping the small rate would ensure the model would incorporate every small vessel into learning.

We used a personal GPU to train this model, running it over a weekend period for multiple hours.

Discussion

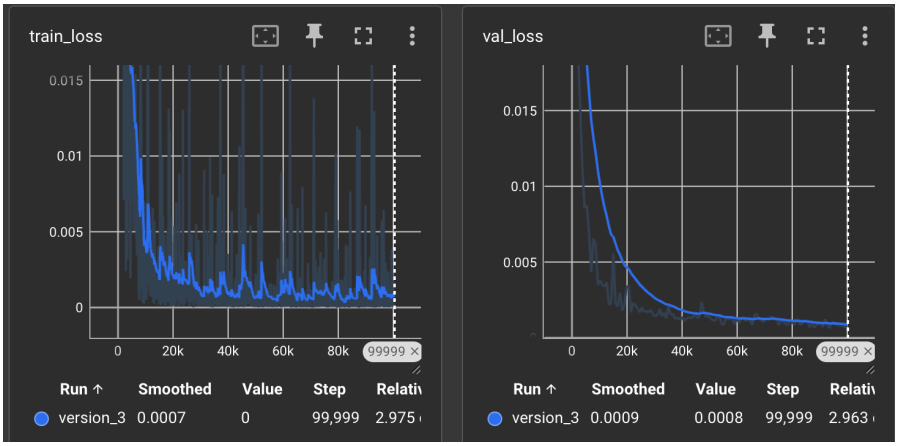
Include visualizations of results, examples of where your algorithm failed and a discussion of why certain algorithms failed or succeeded. In addition, explain whether you think you have overfit to your training set and what, if anything, you did to mitigate that.

After the 100,000 steps the *val_loss* seemed to converge and plateau, showing that it found an optimal solution. The *train_loss* despite being more variable, also seemed to reach a stable value.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

This is the equation for train loss and what this does is measure how well a model fits to the data. We want the model to fit well with the data but not too well to the point where it is overfitting. We can see it in the graphs below where a steady curve downward is good because that curve means that we are getting closer and fitting better to the data. However, if the curve goes back up again, this implies that the model is overfitting and is bad, but we don't see that in our graphs.



With the model completed, we compared our model's performance to the models previously mentioned in the introduction and methods. To do this, we used:

The F1-score, also known as the Dice similarity coefficient (DSC), is a popular metric used to evaluate the performance of a classification model. It takes into account both precision and recall, aiming to strike a balance between them, and can be expressed as:

$$F_1 - Score = \frac{2 \times T_P}{(2 \times T_P) + F_P + F_N}$$

We also use False Positive Rate (FPR) and False Negative Rate (FNR)

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP} \qquad FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

Given that the model was predicting the probabilities of the vessel or not, we only included test cases where the model was >80% confident in its prediction. Our model predicts each pixel and the percentage that the pixel is or isn't a vessel. We can set a threshold for each pixel which means that this would tell us how many vessels will get colored in. Based on the colored-in vessels, we can compare them with the "ground truth" and see which vessels are the same, human-only, model-only, and none. The scores are down below:

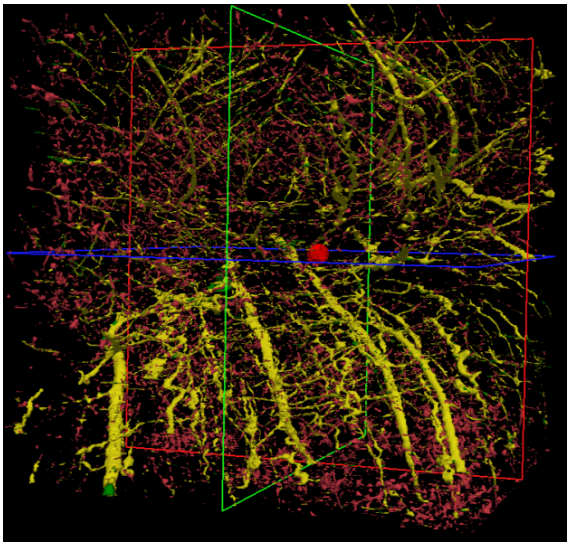
Model	**Dice**	**FPR**	**FNR**
Chollet et al.	(avg) 0.55	(avg) 0.2	(avg) 0.5
Our method	0.681	0.194	0.289

Our method had a higher Dice score, meaning its segmentation is closer to the ground truth labeled by humans. There were, however, still some false positive classifications, where regions

not part of a blood vessel were classified as one. This resulted in “small blobs” within the segmentation mask. There was also a high false negative rate, which meant that some regions of the blood vessels were missed during segmentation. Visual analysis shows that these regions were usually smaller vessels that were most likely not represented well in the CNN pooling algorithm. Compared with the FCNN implementation, our results are significant improvements in multiple areas. The U-Net mode’s ability to preserve spatial details and capture both global and local features resulted in more accurate vessel segmentations.

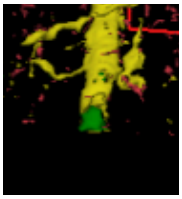
Given the small learning rate and large step size, the risk of overfitting was large. To try and identify any signs of overfitting we looked at FPR and FNR rates. Based on the lower FPR and FNR rates on the validation data, we don’t believe that the model is overfitting the data. Given that the model never interacted with the validation data, any overfitting would easily be seen with a high FPR and FNR.

The output (see right) is colored such:



- Red: Model predicted vessel; Human predicted not vessel (False Positive)
- Green: Model predicted no vessel; Human predicted vessel (False Negative)
- Yellow: Both Model and Human agree on vessel (True Positive)
- Black: Both Model and Human agree no vessel (True Negative)

The model is more likely to predict no vessel for larger vessels, whereas predicting the precedence of a vessel for smaller ones.



The model also seems to run into the issue of mispredicting the presence of smaller vessels, predicting random blobs not connected to the main

Summary of Improvements

The original paper implemented a standard FCNN, which we extended using CNNs for better spatial feature extraction and further refined with U-Nets for image segmentation. The inclusion of skip connections and the encoder-decoder structure led to a noticeable performance boost, showcasing the power of these architectural enhancements in complex tasks.

Conclusion

This research shows how U-Net based architecture can improve the automated segmentation of blood vessels from MRI images, achieving higher Dice scores compared to an FCNN-based approach. This high Dice score indicates that our output is highly similar to the human labelled set. However, the false negative rate was still high, with the algorithm missing some actual blood vessels, instead classifying blobs as blood vessels. In future research, we can work on reducing the false negative rate and improving the overall segmentation.

Based on the train loss, training could have continued a few more steps. Maybe around 10,000 more steps to attempt to find minimum value. One concerning feature of our model is the tendency to highlight small little blobs on the images that had no correlation to the blood vessel, increasing false negative rate. We can alleviate this by placing more weight on the data's attributes, such as *nb_children*, which can inform the model whether there should be any vessels continuing or not. Another potential solution is utilizing *radius* to cut off a prediction after a certain threshold. These 'child' vessels, or blobs, can then be removed, lowering FNR.

Team Member Contributions

Medha: Initial Idea and previous research for Project Proposal, Abstract, Previous Work, Introduction, Dataset Description, Experimental Results, Conclusion/Future Work, proofreading over Kade's sections

Kade: Vessel Segmentation Research and Presentation, Abstract, Dataset Description, Methods, Experimental Results, Discussion, Conclusion/Future Work, proofreading over Medha's sections

Works Cited

Chollet, E., Balbastre, Y., Mauri, C., Magnain, C., Fischl, B., & Wang, H. (2024, July 1). *Neurovascular Segmentation in sOCT with Deep Learning and Synthetic Training Data.* Unpublished. https://arxiv.org/pdf/2405.13757

Gehad Hassan a e, a, e, b, c, d, f, & disease, A. retinopathy is a. (2015, October 9). *Retinal blood vessel segmentation approach based on mathematical morphology*. Procedia Computer Science. https://www.sciencedirect.com/science/article/pii/S1877050915028355

Blood vessel segmentation in retinal fundus images. (n.d.-b). https://stacks.stanford.edu/file/druid:yt916dh6570/Bergen_Boin_Sanjani_Blood_Vessel_Segmentation.pdf

A cross-modality learning approach for vessel segmentation in retinal images | IEEE Journals & Magazine | IEEE Xplore. (n.d.-a). https://ieeexplore.ieee.org/document/7161344/

IterNet: Retinal image segmentation utilizing structural redundancy in Vessel Networks | IEEE Conference Publication | IEEE Xplore. (n.d.-c). https://ieeexplore.ieee.org/document/9093621/

Matloob Abbasi, M., Iqbal, S., Aurangzeb, K., Alhussein, M., & Khan, T. M. (2024, July 2). *LMBiS-net: A lightweight bidirectional skip connection based multipath CNN for Retinal Blood Vessel Segmentation*. Nature News. https://www.nature.com/articles/s41598-024-63496-9