Journey from 2D to 3D Biomedical Image Segmentation using Convolutional Neural Networks

Asees Kaur

Capstone submitted in partial satisfaction of the requirements for the degree of Master of Science in Applied Mathematics

Applied Mathematics
University of California, Merced

Committee in charge:

Dr. Erica M. Rutter
Dr. Suzanne Sindi
Dr. Tomas Rube
Dr. Arnold Kim

This is to certify that I have examined a copy of a technical report by

Asees Kaur

and found it satisfactory in all respects, and that any and all revisions
required by the examining committee have been made.

Applied Mathematics
Graduate Studies Chair:

_____
Professor Chrysoula Tsogka

Committee Co-Chair / Research Advisor:

_____
Professor Erica Rutter

Committee Member:

_____
Professor Suzanne Sindi

Committee Member:

_____
Professor Tomas Rube

Committee Member:

_____
Professor Arnold Kim

_____
Date

Contents

1 Chapter 1: Introduction and Background

1.1 Abstract

Accurately segmenting biomedical images presents a significant challenge due to their complexity, compounded by the labor-intensive nature of manual segmentation and the issue of insufficient data in the healthcare industry. To tackle these challenges, my work focuses on leveraging machine learning techniques, specifically Convolutional Neural Networks (CNNs), to address data scarcity while preserving the characteristics of biomedical images. Chapter 1 provides essential background on biomedical image segmentation and the machine learning techniques used to segment biomedical images. Building on this foundation, Chapter 2 presents a real-world application of region-growing algorithms applied to Digital Subtraction Angiography (DSA) images. Using cross-validation to evaluate the accuracy of these methods is assessed, offering insights into their efficacy for biomedical image segmentation tasks. Finally, Chapter 3 outlines future directions, including extending the algorithm built in Chapter 2 to three-dimensional biomedical imaging data. Expanding to include more data types, this study aims to enhance biomedical image segmentation techniques, benefiting clinical decision-making and patient care.

1.2 Introduction and Background for Biomedical Image Segmentation

Image segmentation is the process of dividing an image into multiple regions, facilitating a deeper understanding and analysis of digital images [2]. Each pixel in an image is categorized into its respective class through pixel-to-pixel prediction [19]. There are two broad categories of image segmentation: semantic segmentation and instance segmentation. Semantic segmentation categorizes an image into foreground and background, assigning the same class to all similar objects within the image, thereby grouping together parts belonging to the same category [19]. In contrast to this, instance segmentation not only categorizes individual objects within the foreground but also performs semantic segmentation simultaneously, thereby categorizing each object distinctly. Figure 1a displays an example of a microscopy image of Glioblastoma-astrocytoma U373 cells on a polyacrylamide substrate from the ISBI Cell Tracking Dataset [1, 15]. In Figure 1b, shows the semantic segmentation of the cell image shown in Figure 1a and in Figure 1c shows the instance segmentation of the same cell image.

(a) Cell Image **(b)** Semantic Segmentation **(c)** Instance Segmentation

Figure 1: Example of instance and semantic image segmentation: (a) a microscopy image of Glioblastoma-astrocytoma U373 cells on a polyacrylamide substrate, (b) the semantic segmentation of the cell image, and (c) the instance segmentation of the same cell image from the ISBI Cell Tracking Dataset.

Image segmentation is applicable across various image types, with significant implications in biomedical imaging. With advancements in medical imaging techniques such as X-rays, ultrasound, CT scans, and MRI, computer-aided image segmentation can extract specific regions of interest (RoIs) for further evaluation, such as organ identification and tumor detection [6]. Medical imaging is essential in diagnosing illnesses, planning treatments, and monitoring progress. It allows healthcare providers to see inside the body and identify any irregularities. Additionally, analyzing medical images can assist in various tasks such as diagnosing diseases, planning treatments, localization of areas of interest, and even aiding in drug development [17].

Biomedical image segmentation, which involves partitioning images into meaningful segments, is particularly significant in both research and clinical settings[31, 26]. By segmenting medical images, clinicians can identify and isolate regions of interest, such as organs or lesions, from complex CT and MRI scans[37]. This process not only facilitates the extraction of critical information regarding the shape, size, or volume of regions like tumors but also aids in accurate diagnosis and treatment planning. However, manual segmentation can prove tedious, expensive, and prone to human error, highlighting the pressing need for automated and reliable segmentation techniques to enhance efficiency and precision in medical imaging analysis.

Despite numerous successful approaches, image segmentation remains one of the most challenging aspects of computer vision. This difficulty stems from the challenge of effectively representing features, especially in medical images which often suffer from issues like blur, noise, and low contrast. While traditional segmentation techniques such as edge detection and thresholding have historically been employed in medical imaging, the arrival of deep learning has revolutionized image segmentation across various domains, including biomedicine [40]. Deep learning techniques, particularly convolutional neural networks (CNNs) offer advantages over traditional methods, particularly in handling complex tasks with large datasets. CNNs excel at hierarchical feature representation, making them a prominent research focus in image processing and computer vision, and providing excellent segmentation results even in the presence of image noise, blur, or low contrast. [39, 37]

1.3 What are Convolutional Neural Networks?

Convolutional Neural Networks (CNNs) are widely used for image analysis tasks, such as image classification, image segmentation, object detection, facial recognition, and much more. For example, CNNs have achieved remarkable accuracy in image classification tasks, particularly with architectures like AlexNet [16], which significantly advanced the field. In image segmentation, the U-Net [29] architecture has been especially effective for biomedical applications. CNNs are also essential in object detection frameworks like YOLO (You Only Look Once) [27] and facial recognition systems such as FaceNet [30], demonstrating their versatility and powerful performance in various image analysis domains.

CNNs were first introduced in 1980 as Neocognitron that laid some foundational ideas in the context of pattern recognition and image processing. [5]. The concept of convolution, which involves sliding a filter (or kernel) over an input image and computing element-wise multiplications, is central to CNNs. This operation allows the network to extract features from the input data, mimicking the human brain's ability to recognize patterns. In a CNN, the convolutional layer applies multiple filters to the input image, each filter extracting different features. These filters, also known as kernels, are essentially arrays of learnable parameters that are adjusted during the training process to enhance the network's ability to recognize specific patterns or features. Images, being two-dimensional grids of pixel values, are well-suited for convolutional operations. As the filters slide over the input image, they compute the element-wise product between their weights (parameters) and the corresponding region of the image, producing feature maps that represent different aspects of the input. Overall, the convolutional layer plays a crucial role in feature extraction, enabling CNNs to learn hierarchical representations of the input data, ultimately leading to effective pattern recognition, just as the human brain does. [38]

1.4 Convolutional Neural Networks for Biomedical image segmentation

Deep Learning, particulary Convolutional Neural Networks(CNNs) plays a crucial role in Biomedical Image Segmentation due to their ability to learn complex features adapt to the variability inherent in medical images [36]. Biomedical images, derived from sources such as CT scans, MRIs, X-rays, and ultrasounds, offer valuable insights into various anatomical structures, including organs, blood vessels, and bones [12].

CNNs excel at automatically learning intricate patterns and representations directly from image data, alleviating the time-consuming process of manual interpretation. Among deep learning methods, CNNs are

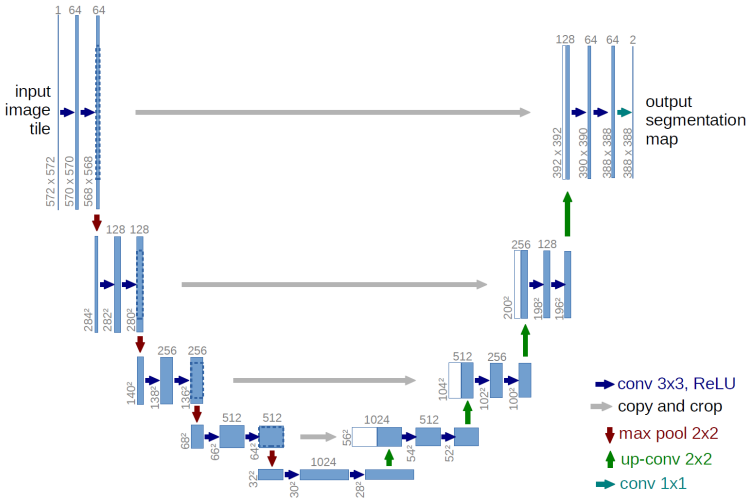


Figure 2: Unet architecture

commonly employed for feature extraction in medical image analysis [12, 7]. Notably, the U-Net architecture has emerged as a prominent choice for biomedical image segmentation [29].

(**I will replace the unet image later**) The U-Net architecture comprises of an encoder-decoder network, as illustrated in Figure 2. The encoder, located on the contracting path (left side), consists of convolutional layers followed by ReLU(Rectified Linear Unit is an activation function used in neural networks that outputs the input directly if it is positive, otherwise it outputs zero - it adds non-linearity to the process) activation functions and max-pooling operations that aids in feature extraction. On the other hand, the decoder, situated on the expansive path (right side), aims to upsample feature maps obtained from the encoder to preserve contextual information obtained from the encoder at higher resolutions. Crucially, skip connections (seen as arrows in Figure 2) facilitate the transfer of information between corresponding encoder and decoder layers, aiding in precise segmentation [29]. The output U-Net is probability maps of the likelihood of each pixel belonging to a particular class.

U-Net has demonstrated remarkable success in medical image segmentation, attributed not only to its capacity to capture contextual information efficiently but also its ability to learn from minimal annotated data, often through a tile-based approach [18, 8]. U-Net, while powerful for biomedical image segmentation, faces several challenges. It often produces patchy probability maps , leading to inconsistent and fragmented segmentations that lack smooth and contiguous boundaries. Additionally, U-Net requires extensive annotated training data to generalize effectively, which is labor-intensive and resource-demanding to obtain [29].

Patch-based methods help solve the issue of fragmented segmentation by breaking down large images into smaller, manageable patches. These patches are individual segments of the larger image that the model processes independently, which creates more diverse training data. This approach allows the model to learn detailed features better, leading to smoother and more accurate segmentations. In response to these challenges, my research focuses on a novel patch-based technique, the region growing algorithm [10], to address segmentation complexities in medical images.

2 Chapter 2: Region Growing Algorithm for two-dimensional Digital Subtraction Angiography Images

2.1 Overview of Pipeline

In this chapter, we will explore a novel segmentation pipeline designed to enhance the connectivity of segmented regions in Digital Subtraction Angiography (DSA) images, which are crucial for visualizing blood vessels. This pipeline effectively combines the strengths of patch-based methods and the region-growing algorithm to address the inherent limitations of the U-Net architecture, particularly in segmenting intricate structures like blood vessels. DSA images are used to visualize blood vessels and diagnose various vascular conditions. Accurate segmentation of these images is crucial for identifying abnormalities such as blockages, or aneurysms. Connectivity in the context of blood vessel segmentation refers to the ability to represent the vascular network as continuous and uninterrupted structures. This is vital for ensuring that the segmented blood vessels accurately reflect the true anatomy, which is essential for reliable diagnosis and treatment planning.

To enhance connectivity, we start by employing a convolutional neural network (CNN) inspired by the U-Net architecture to predict 3x3 masks for the input DSA images. The image is divided into smaller patches (80 X 80), allowing the model to focus on detailed features within each patch. This process generates diverse and detailed training data, which is essential for accurately capturing the fine structures of blood vessels. The 3x3 masks predicted by the CNN provide localized context for the next stage of the pipeline. These masks serve as seed points for the region-growing algorithm, ensuring that the initial segmentation captures essential details while providing a basis for further refinement.

The region-growing algorithm then takes over, using the 3x3 masks as seed points to iteratively expand the segmented regions based on pixel similarity. Importantly, the expansion only occurs in the direction of the foreground, which corresponds to the blood vessels in the DSA images. By iteratively expanding these seeds, the algorithm ensures smooth and contiguous boundaries for the blood vessels, effectively mitigating the boundary artifacts and inconsistencies typically introduced by patch-based methods alone. This combined approach ensures that the segmented blood vessels are continuous, reducing the risk of fragmented segments that can obscure the true vascular structure. Accurate and connected vessel segmentation improves the ability to diagnose vascular conditions by providing a clear and complete representation of the vascular network.

In summary, the goal of enhancing connectivity in the segmentation of DSA images is achieved by integrating the detailed focus of patch-based methods with the iterative expansion of the region-growing algorithm. This approach results in accurate, cohesive, and clinically valuable segmentations of blood vessels, ultimately improving diagnostic and treatment outcomes. By addressing the limitations of traditional U-Net architectures, this pipeline ensures that the segmented vascular structures are continuous and reflective of true anatomical connections, which is crucial for effective medical imaging analysis.

2.2 Introduction and Motivation

Cerebrovascular diseases (CVDs) are conditions that affect blood flow and blood vessels in the brain, encompassing aneurysms, strokes, arteriovenous malformations, and arteriovenous fistulas. Patients with CVDs are at risk of significant morbidity and mortality, making these diseases one of the leading causes of death today [41]. The main motivation for this project stems from the challenges faced by stroke patients. Strokes, which occur when blood flow to the brain is blocked or reduced, can lead to permanent neurological damage, disability, or even death. Symptoms of strokes include dizziness, sudden confusion, and weakness in the arms or legs. Strokes are broadly categorized into two types: ischemic and hemorrhagic. Ischemic strokes, the most common type, are caused by the blockage of an artery due to clogged blood vessels. Hemorrhagic strokes occur when a blood vessel in the brain leaks and bleeds into or around the brain.

Even though Computed Tomography (CT) and Magnetic Resonance (MR) imaging techniques are used to detect strokes, Digital Subtraction Angiography (DSA) is an imaging method particularly valuable for diagnosing and guiding treatment in cases of strokes and other CVDs. DSA is a fluoroscopy technique used in interventional radiology to visualize blood vessels clearly in a bony or dense soft tissue environment. This involves inserting a small catheter, usually in an artery in the leg, and passing it up to the blood vessels in the brain. Images are produced using a contrast medium by subtracting a "pre-contrast image" or mask from subsequent images once the contrast medium has been introduced into the system, hence the term "digital subtraction angiography." Subtraction angiography was first described in 1935 and, in English sources, in 1962 as a manual technique. Digital technology made DSA practical in the 1970s [9].

Accurate image segmentation is crucial for clearly visualizing blood vessels in DSA images, which helps guide clinicians in making objective decisions. However, manually segmenting each region is very time-intensive and subject to human error. In recent years, machine learning methods, especially Convolutional Neural Networks (CNNs), have been extensively used for biomedical image segmentation because of their capability to learn complex features. Biomedical images can be quite complex and show significant variability, making them time-consuming to understand. CNNs, such as the U-net architecture, can automatically learn important and complex patterns, adapting to different scenarios. [29].

However, the traditional U-net has some drawbacks. Since it processes the entire image at once, it can blur the local information of the blood vessels and their connections, producing patchy probability maps [13, 8]. Additionally, in medical image analysis, there is often insufficient labeled training data to train machine learning models adequately. One method to counteract this problem is to create many training image-label pairs from a single image. This process of creating patches helps preserve localized information and handle different resolutions of the images. Most importantly, using patch-based methods can address the insufficient data problem by generating more data from a single image.

This project aims to use a patch-based method known as Region Growing Algorithm (RGCNet) on DSA images. This approach aims to preserve localized information, handle different image resolutions, and address the issue of insufficient data by creating more training samples from a single image. RGCNet offers the potential to improve segmentation accuracy by focusing on local regions of interest within the images, thus enhancing the overall performance of automated segmentation techniques in the context of cerebrovascular diseases.

2.3 Data and Preprocessing

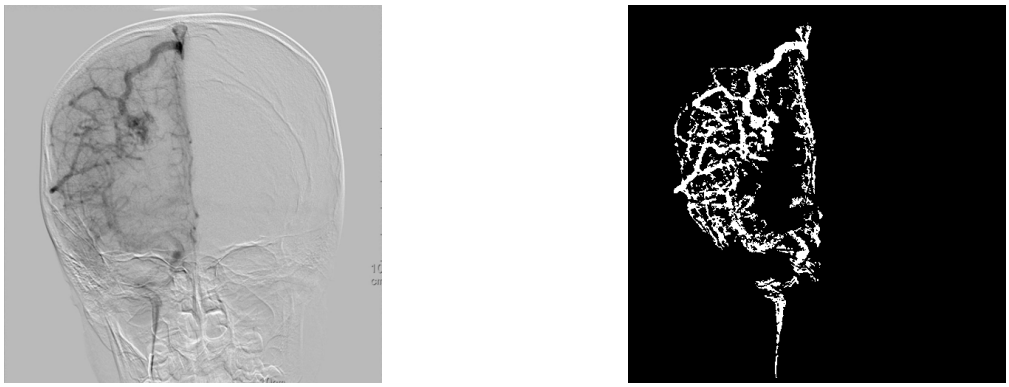
2.3.1 DSA Data

The 26 DSA images were manually traced from a randomly selected group of patients to label the ground truth and the images were de-identified for each patient. There is only one DSA image per patient. The contrast agent was iodixanol (Visipaque, General Electric), administered by intra-arterial power injection at 5 ml/s for a total amount of 8 ml into the cervical carotid arteries, or at 3 ml/s for a total amount of 5 ml into the cervical vertebral arteries. The patients that were included in this study had either pathological abnormalities from arteriovenous malformations, arteriovenous fistula, and cerebrovascular aneurysm, or normal DSA findings [41].

Figure 3 displays a typical DSA image and its corresponding mask.

2.3.2 Cross Validation

Cross-validation is a statistical method used to evaluate the performance of machine learning algorithms and improve their generalizability. This process typically involves splitting the dataset into two sets: one for training the model and the other for testing. These sets are rotated in successive rounds, ensuring that each data point is used for testing at least once. This technique helps ensure that the algorithm is unbiased



(a) DSA Image **(b)** DSA Mask

Figure 3: On the left (a) is an example of a typical DSA Image, acquired during angiography, and on the right (b) is its corresponding mask highlighting regions of interest.

towards unseen data and also provides information about how sensitive the learned model is to the training data. The most basic type of cross validation method is k-fold cross validation. In k-fold cross validation, the data is divided into k equal parts or folds. The model is then trained and tested k times, each time using a different fold for testing and the remaining k-1 folds for training [28].

In this work, I used 5-fold cross-validation. The dataset consisted of 26 image and mask pairs, which were randomly split into 5 sets. Each fold had 21 pairs for training and 5 pairs for testing. The training set was further split into training and validation sets with an 80-20 ratio via images/masks. The model, described in section Section 2.4, was trained on 80% of the data and validated on 20%. The validation was used to select the best weights for the model. The trained models were then tested using the region growing algorithm described in section Section 2.5 on the corresponding unseen testing sets.

After dividing the dataset for 5-fold cross-validation, each image and mask pair in the training set underwent balancing, as detailed in Section 2.3.3. This process ensured that all classes were equally represented, enhancing the model's ability to learn effectively. Following balancing, patches from all images were pooled together to improve data representation and facilitate robust learning. Pooling helped ensure adequate representation of rare categories, thereby enhancing the model's resilience to data variations.

2.3.3 Creation of Training Data and Labels

For pre-processing the data, we have normalized the images by dividing each pixel by the maximum pixel value in the image. To create the training data, we needed an (80, 80) block around each pixel in the image. We slide through every pixel in the image and get an (80, 80) block around it, considering it as the center pixel. This means using offsets of ± 40 pixels from the center pixel to capture the full block. Thus, for each image of size (512, 512), a total of 262,144 image-label pairs are constructed, one for each pixel in the image. To avoid complications around the boundaries of the image, the images and masks were padded by 40 zeros in each direction. For example, the original image was of the size (512, 512), and after padding, the size of the new image becomes (592, 592). This padding ensures that all (80, 80) blocks have valid pixels at the edges of the image. The same is done for the corresponding masks.

The training dataset is constructed of these mini tiles of the size (80, 80). In other words, here is where we use patch based methods. These mini tiles that are created for training data are the patches. Figure 4 portrays the creation of 80 X 80 patches of the training image. The left panel shows the original image, the middle panel shows a zoomed in area from the original image and the right panel is illustrating the 80 X 80 patches for the zoomed in image. Similarly, we constructed a testing data-set by forming tiles of the from the testing image.

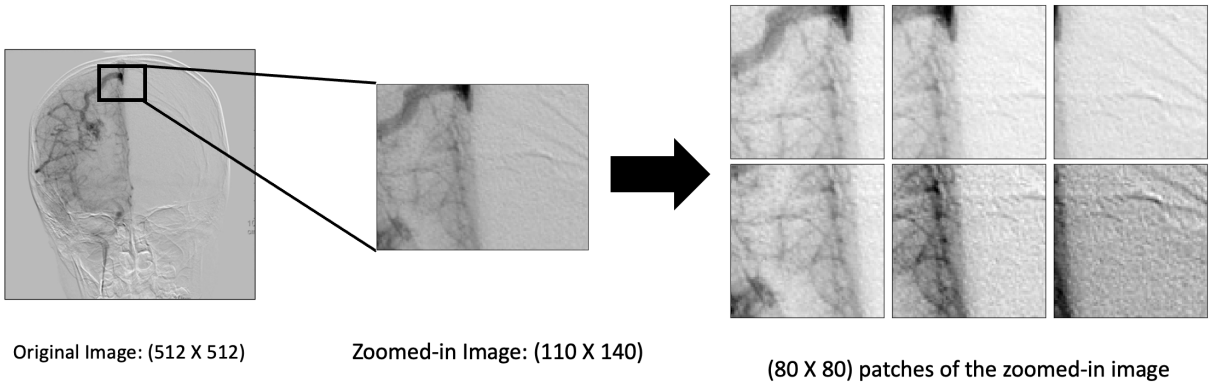


Figure 4: Example illustrating the creation of patches for training images. The left panel displays a sample training image, while the middle panel offers a close-up view of a referenced section from the left image. The right panel demonstrates the generation of 80 x 80 patches from the zoomed-in section shown in the middle panel.

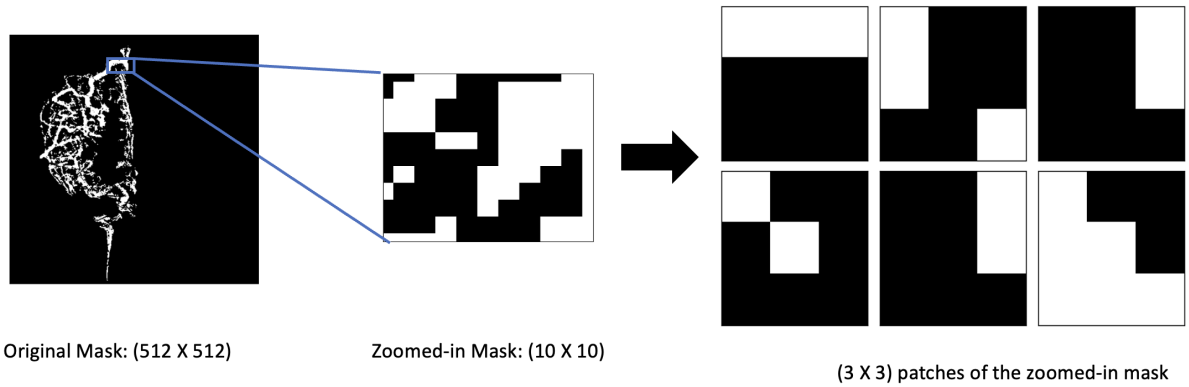


Figure 5: Example illustrating the creation of patches for training masks. The left panel displays a sample training mask, while the middle panel offers a close-up view of a referenced section from mask in the left panel. The right panel demonstrates the generation of 3 x 3 patches from the zoomed-in section shown in the middle panel.

In the similar way, the training mask was composed by taking a (3, 3) block around each pixel for each corresponding mask. A mask is an image the same size as the original, where each pixel indicates the class or object of the corresponding pixel in the original image. Masks are used in image segmentation tasks to classify each pixel in the image. This 3x3 block captures the local neighborhood around each pixel, providing context for the central pixel, which is crucial for segmentation tasks. An example of how this was implemented is shown in **??**. The left panel displays a sample training mask, the middle panel offers a close-up view of a referenced section from the mask in the left panel, and the right panel demonstrates the generation of 3 x 3 patches from the zoomed-in section shown in the middle panel.

It is evident from the above DSA image and mask Figure 4 and Figure 5, that there is more background pixels(zeros), than foreground pixels(ones) in the image. This class imbalance problem can influence what is learned and prioritized in the network and thus we will balance the data. Since directly assessing the class imbalance from the images could be complex, we opted to analyze the training masks. We calculated the sum of each 3x3 mask in the training data, with sums ranging from 0 (all background) to 9 (all foreground). We then down-sampled the data to the lowest number of patches per class of the masks and their corresponding image patches to achieve balance.

Sum	Number of patches per sum
0.0	235316
1.0	3803
2.0	3209
3.0	3212
4.0	2404
5.0	2186
6.0	2634
7.0	2131
8.0	2018
9.0	5231

Table 1: Distribution of Training Mask Patch Sums: Table showing the counts of training image/mask patches for each sum, from 0 to 9, highlighting a predominance of background patches (sum 0) and the fewest patches at sum 8.

Table 1 shows a table of sums of training mask patches, ranging from 0 to 9 in the left column and in the right is the number of training image/mask patches for that corresponding sum. As we can see from the table most number of patches belong to the category of sum 0, which means most of the patches are just

from the background of the image. Analyzing the table further, we notice that the lowest number of patches belong to sum 8. In our case, we down-sampled the data to address the class imbalance issue. This means that we deliberately reduced the number of training samples by selecting a subset of patches, ensuring a more balanced representation of foreground and background classes. To achieve this, we identified the class with the fewest samples (in our case, the foreground class with sum 8 in the training mask patches) and sampled an appropriate number of patches from each class to achieve a balanced dataset. By doing so, we ensured that the model receives sufficient training data from each class to learn effectively without being biased towards the majority class.

2.4 Architecture and Training

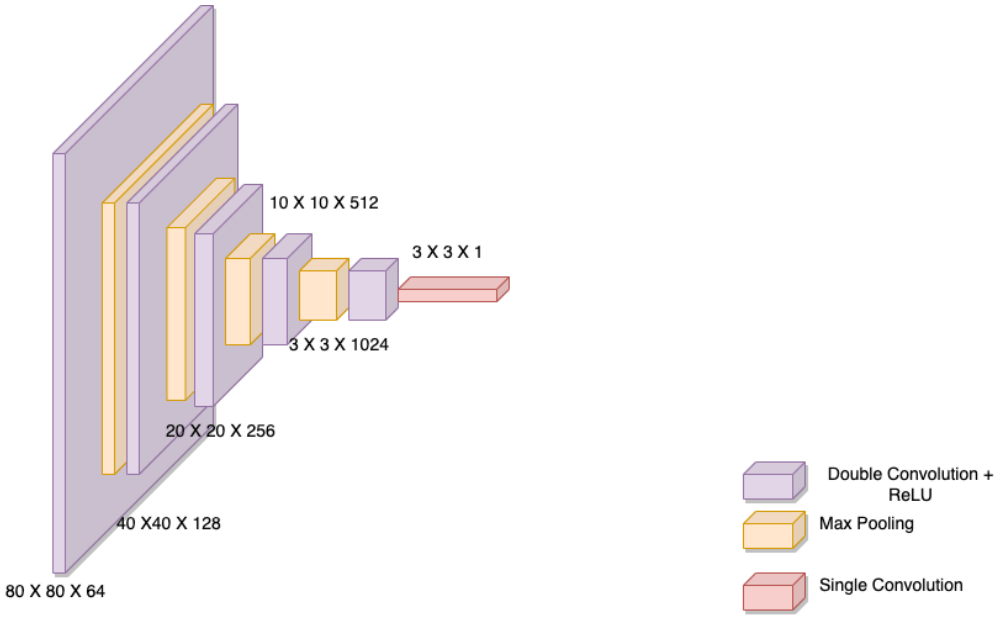


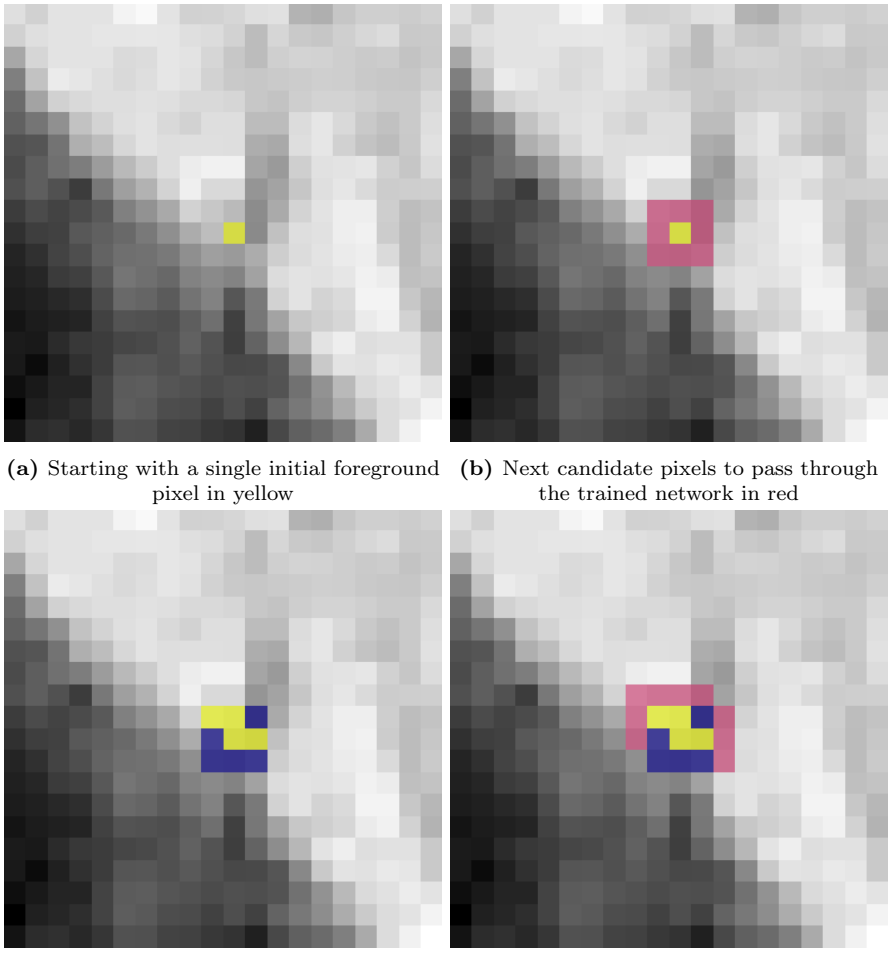
Figure 6: Network Architecture: Convolutional Neural Network (CNN) architecture consisting of layers with double convolution and ReLU (purple), max pooling (yellow), and a final single convolution (red). The dimensions of each layer are indicated, starting from 80x80x64 to 3x3x1.

I used a 25-layer Neural Network that resembles the encoder part of U-net[29]; there are 5 double convolutional layers activated by ReLu and each double convolutional layer is followed by a maxpooling layer, the last double convolution layer is followed by a single convolution to reshape into (3,3,1) - where 1 represents if a pixel is in foreground or background. The architecture is exhibited in Figure 6. This model was trained on the training data created earlier in the data preprocessing step. During training, the model was optimized using the Adam optimizer [14] and trained with the Binary Cross Entropy loss function. Training was conducted on a GPU available on the MERCED Cluster. The training process spanned 50 epochs, with a batch size of 50 images for the training data.

In general, Convolutional Neural Networks (CNNs) are trained to produce output masks of the same size as the input. However, in this case, we map an 80×80 input image to a 3×3 output mask. This approach captures all the contextual data around each pixel, ensuring contiguity. The 80×80 patch provides a wide range of information crucial for learning features and recognizing patterns. By predicting a 3×3 region, the network ensures spatial consistency within this smaller area, helping to maintain smooth transitions and reduce jagged edges in the segmentation map. The central pixels are influenced by a coherent set of features derived from the larger context.

2.5 Region Growing Algorithm

In the testing phase of our network evaluation, we utilize the Region Growing Algorithm (RGC) for efficient segmentation. Unlike the training phase, where the focus is on parameter optimization, testing aims for accurate results within a reasonable timeframe. The RGC's iterative approach, expanding around foreground pixels, ensures contiguous regions without processing the entire image. This targeted strategy significantly reduces processing time compared to comprehensive image analysis. This algorithm grows iteratively around each foregorund pixel towards the direction of foreground pixels to ensure the contiguity of the predicted region.



(a) Starting with a single initial foreground pixel in yellow

(b) Next candidate pixels to pass through the trained network in red

(c) Foreground (yellow) and background (blue) predicted pixels

(d) Next iteration of candidate pixels (red) to pass through the network

Figure 7: Example illustrating the region growing method. The process begins with a single initial foreground pixel (yellow) as shown in the first panel. The second panel identifies the next candidate pixels (red) that will be evaluated by the trained network. The third panel shows the predicted classification of pixels into foreground (yellow) and background (blue). Finally, the fourth panel highlights the next set of candidate pixels (red) for the subsequent iteration of the network evaluation. Yellow indicates foreground pixels, blue indicates background pixels, and red indicates the next candidate pixels for classification.

Figure 7 shows the working of the region growing algorithm in detailed steps. In all of the images, yellow represents predicted foreground pixels, purple represents predicted background pixels, and red represents the pixels that will be predicted in the next iteration of the algorithm. For example, Figure 7a shows one foreground pixel on the testing image in yellow. Figure 7b shows the candidate pixels that will be predicted in the next iteration. To determine these candidates, we extract an 80 X 80 patch around each center pixel and pass it through the trained network. The network outputs a 3 X 3 probability mask, indicating the

likelihood of each pixel being classified as foreground or background. The algorithm makes a prediction based on the trained model, which can be seen in Figure 7c, which shows the predictions from the above candidate pixels. The pixels in yellow were categorized as foreground and the pixels in purple were categorized as the background. To avoid unnecesary computation, the algorithm only expands in directions of foreground pixels and discards the background ones. Thus, in the next iteration of the algorithm, the pixels that will be predicted into foreground and background (Figure 7d, red), only surround the centroids which were predicted as foreground in the previous iteration (Figure 7c, yellow). We can see in Figure 7d that the next set of candidate pixels in red are either adjacent or diagonal to the foreground pixels. Here I am using a trial and error threshold of 0.1 on the predictions. Pixels with a prediction probability greater than 0.1 are considered foreground. This threshold was selected based on experimentation.

In practice, however, we do not start with one candidate initial pixel, we instead start with many to ensure computational speed. Instead, we start with 500 initial randomly selected pixels in the image. We keep track of the indices whose prediction is 1 i.e. the foreground pixels and only those indices go into the next iteration and the others are removed. We follow this procedure until there are no new indices are left to run. Figure 8 displays the construction of the predicted test image after every 82 iterations of the algorithm.

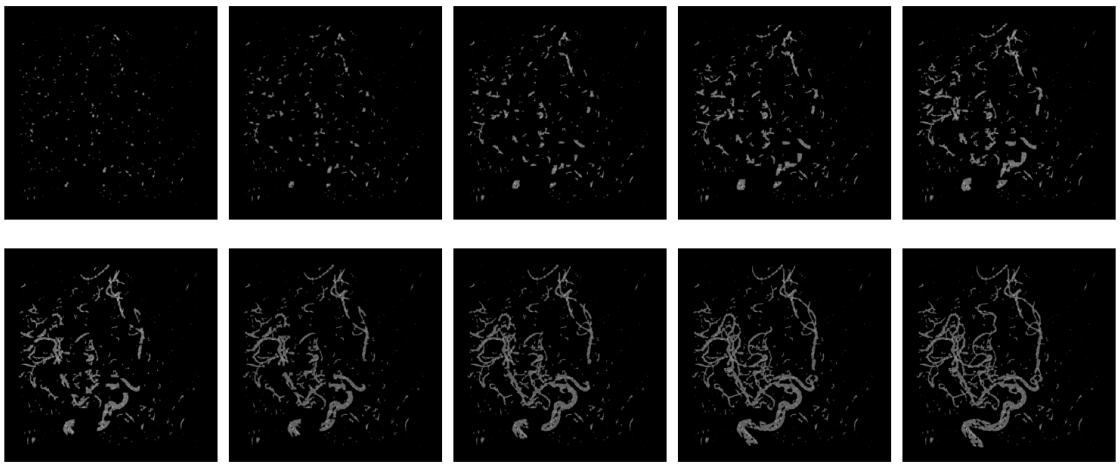
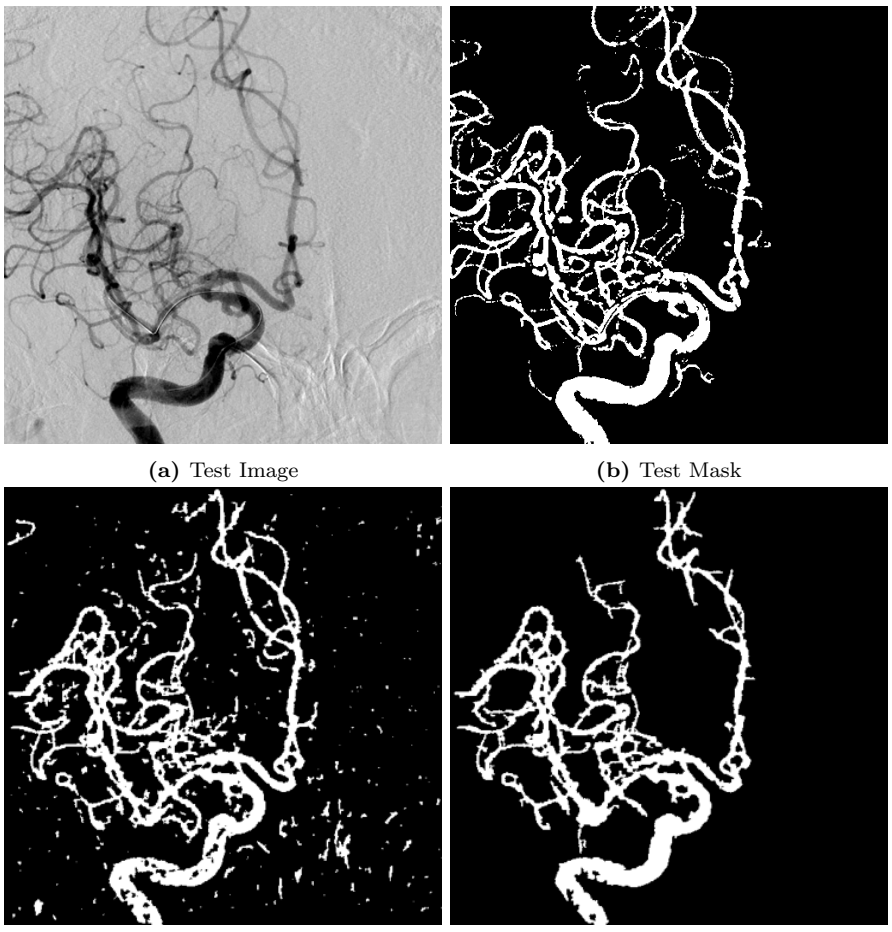


Figure 8: Reconstruction process of the same predicted test image at intervals of every 82 iterations of the algorithm. The series of subplots illustrates the progressive refinement and development of the image as the algorithm iteratively updates the predictions. Both the top and bottom rows show the same image evolving over time.

2.6 Post-processing and Analysis

After the testing process, we can compare our predicted segmentation to the ground truth segmentation. The objective was to use the region growing algorithm to predict the flow of the veins in the brain, ensuring that the segmentation accurately represents the most relevant anatomical structures for further analysis. The largest connected component is used to ensure that the predicted segmentation focuses on the most significant and continuous structure, which is critical for accurately identifying and analyzing the flow of veins in the brain. Figure 9a shows the original test image used for evaluation, while Figure 9b displays the test mask applied to delineate specific regions of interest within the test image. This mask guides the reconstruction process by indicating areas to be segmented. In Figure 9(c), we observe the fully reconstructed image after the algorithm processes the test image and mask. This reconstructed image represents the predicted segmentation over the entire region of interest. Figure 9d displays the largest connected component of the reconstructed image, highlighting the most significant contiguous area identified by the algorithm. Additionally, any holes within the largest connected component have been filled to ensure a more accurate

and contiguous representation of the vascular structure. This component is crucial for understanding the vein structure as it provides a clear and focused representation of the primary vascular pathways.



(a) Test Image **(b)** Test Mask

(c) Reconstructed Mask **(d)** Largest Connected Component

Figure 9: Final Results: A comprehensive overview of the input and output stages in the reconstruction process. These subfigures collectively illustrate the test image, test mask, fully reconstructed mask, and the largest connected component identified in the reconstruction.

In the Figure 10 we analyze our reconstruction by comparing it with the test mask. The white parts in this image shows the true positives, i.e. the portion that was foreground in the original test mask and was also predicted as foreground by the algorithm. The black here shows the true negatives, i.e. the background in the test mask that was also predicted as background by the algorithm. The false positives, i.e. the foreground predicted by the algorithm but was background in the original mask is depicted in yellow. Red shows the false negatives, i.e. part that was predicted as background by the algorithm but was not background in the original mask.

Upon examining the results in the figure, it appears that there are more false negatives (red) than false positives (yellow). This indicates that the algorithm tends to under-predict the foreground regions, missing significant parts of the vein structures. The presence of false positives, although less prevalent, suggests that there are some areas where the algorithm incorrectly identifies background regions as part of the vein structure.The higher prevalence of false negatives indicates that the algorithm struggles more with sensitivity (detecting all actual vein structures) than specificity (correctly identifying background regions). Overall, the results suggest that while the algorithm effectively identifies many true positives and true negatives, it needs refinement to better capture all true vein structures and reduce the occurrence of false negatives. Addressing these issues could significantly improve the segmentation accuracy and reliability.

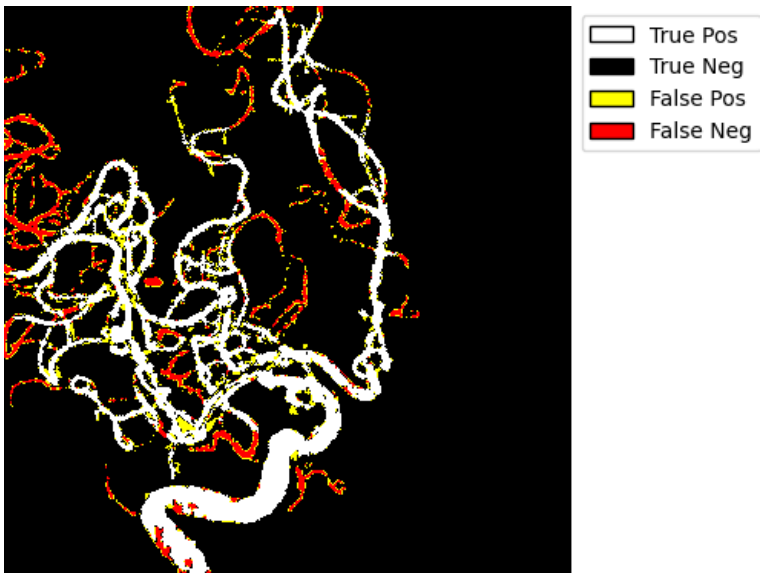


Figure 10: Analysis: Image shows comparison of the original test mask with the largest component of reconstruction. White represents true positives, black represents true negatives, yellow represents false positives, and red represents false negatives

In addition to qualitative analysis, it is imperative to include quantitative analysis to measure how accurate the segmentation masks are. In the world of biomedical image segmentation, jaccard index and dice score have become the most popular measures of evaluation[3]. The Jaccard score calculates the similarity between two masks, predicted and the original by dividing the intersection between both by the union of both the masks, as shown in Equation (1).

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

where A is the predicted mask and B is the ground truth of the test mask. The closer the score is to 100%, the closer are the predictions to the original mask. In this case, the Jaccard score was approximately 65%, indicating a moderate level of overlap between the predicted and ground truth masks. The dice score is another statistical measure to check the similarity between two masks. It is calculated by dividing twice the intersection between both the masks by the sum of number of elements in each mask, as shown in Equation (2)

$$\text{Dice Score} = \frac{2|A \cap B|}{|A| + |B|} \tag{2}$$

The closer the score is to 100%, the closer are the predictions to the original mask. The Dice score was approximately 79% in this case. These quantitative results indicate that the segmentation algorithm performs reasonably well, but there is still room for improvement. A Jaccard score of 65% and a Dice score of 79% show that while the algorithm can accurately capture many of the vein structures, it still misses some areas or incorrectly identifies background regions as veins.

In addition to the qualitative and quantitative analyses using Jaccard index and Dice score, we also calculated the accuracy of our segmentation algorithm. The accuracy is defined as the proportion of true results (both true positives and true negatives) among the total number of cases, as shown in Equation (3)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives, we calculated the accuracy of our segmentation algorithm to be approximately

94%. Comparing our results to those reported in Zhang et al. [41], where a U-net based deep learning approach achieved an average Dice coefficient of 82.68% and an accuracy of 97.8%, highlights the limitations and potential of our current approach. While there is a gap in performance, our results are notable given the significant constraint of training on a single image. The results demonstrate that even with a limited training dataset, the combination of a patch-based approach and the region growing algorithm can yield reasonable segmentation performance. This suggests that the methodology has strong potential, particularly when a small amount of data is available.

The primary objective of this study was to enhance the contiguity of the region. Examination of Figure 11 reveals that the largest connected component in the reconstructed mask is larger than that in the original test mask.

(a) Largest Connected Component of the original Test mask

(b) Largest Connected Component of the Reconstructed Mask

Figure 11: Comparison of the largest connected components in the original test mask and the reconstructed mask. The left image shows the largest connected component of the original test mask, while the right image depicts the largest connected component of the reconstructed mask, highlighting the improvements in connectivity achieved through the reconstruction process.

Visually, the Region Growing CNN reconstruction demonstrates greater connectivity compared to the original mask. One metric used for evaluation is the centerline Dice (clDice) [33]. The clDice assesses the accuracy of tubular structure segmentation by emphasizing the preservation of centerline topology. It quantifies how well the segmented centerlines align with the ground truth, which is crucial for tasks in medical imaging such as vascular analysis and nerve tracking. The clDice is computed as follows:

$$clDice = \frac{2 \times |\text{Skeleton}_{\text{seg}} \cap \text{Skeleton}_{\text{gt}}|}{|\text{Skeleton}_{\text{seg}}| + |\text{Skeleton}_{\text{gt}}|} \tag{4}$$

Here, $\text{Skeleton}_{\text{seg}}$ represents the skeleton of the segmented image, and $\text{Skeleton}_{\text{gt}}$ represents the skeleton of the ground truth mask. The notation $|\cdot|$ denotes the cardinality or number of pixels in their respective skeleton representations. Skeletonization is a process that reduces tubular structures to their medial axes while maintaining their connectivity. In the context of this study, the centerline Dice (clDice) metric was utilized to quantitatively evaluate the accuracy of tubular structure segmentation. The clDice score of 71% indicates how well the segmented centerlines align with those in the ground truth mask. Specifically, a clDice score of 71% implies that the segmented centerlines overlap with the ground truth centerlines by 71% on average. This metric is crucial in medical imaging for tasks such as vascular analysis and nerve tracking, where accurately identifying and delineating tubular structures is essential for diagnosis and treatment planning.

Another aspect investigated in this study was the ratio of the largest component of the reconstructed structure to the total area of the reconstruction. This ratio is calculated as follows:

$$\text{Ratio of Largest Connected Component} = \frac{\text{Length of the largest connected component}}{\text{Total area of the reconstructed structure}} \tag{5}$$

Here, Length of the largest connected component metric measures the longest continuous path of pixels and total area of the reconstructed structure refers to the entirety of the area covered by the reconstructed image or segmented region, encompassing all pixels that compose the segmented structure. It quantifies the overall size and spatial coverage of the reconstructed object or area of interest.

This came out to be 70%, which means 70% of the reconstruction is connected. In this context, the "Length of Largest Component" refers to the cumulative size or extent of the largest connected segment within the reconstructed image. The "Total Reconstructed Area" denotes the entire area covered by the reconstructed structure. For instance, a calculated ratio of 70% indicates that the largest connected component comprises 70% of the total reconstructed area. This metric provides valuable insights into the structural integrity and continuity of the reconstructed region. A higher ratio typically suggests a more cohesive and connected reconstruction, which is beneficial for accurate representation and analysis in medical imaging applications.

2.7 Discussion

In this chapter, we have demonstrated the application of the Region Growing Algorithm (RGCNet) on two-dimensional Digital Subtraction Angiography (DSA) images. Our approach shows compelling results, particularly in terms of accuracy, despite being trained and tested with limited data. The primary goal of our work was to enhance the segmentation accuracy of DSA images to aid in the diagnosis and treatment of cerebrovascular diseases (CVDs). By using a patch-based method combined with an iterative flood fill algorithm, we aimed to preserve localized information and handle varying image resolutions more effectively. Our results indicate that RGCNet significantly improves the clarity and precision of segmented blood vessels in DSA images. This improvement is crucial for clinicians as it facilitates more accurate assessments and informed decision-making. The ability to generate multiple training image-label pairs from a single image has proven beneficial in addressing the common challenge of insufficient labeled training data in medical imaging.

In the next part of my research, I will expand this work to the 3D DSA images using a 3D CNN architecture. This expansion will allow us to capture more complex spatial relationships within the cerebrovascular structures, potentially leading to even more accurate and informative segmentation results.

2.8 Acknowledgements

I would like to thank Xiaoyin Xu for providing the DSA images used in this study [41].

3 Chapter 3: Region Growing Algorithm for 3D Biomedical Images

3.1 Importance of 3D Biomedical Image Segmentation

Medical images, including computerized tomography (CT), magnetic resonance imaging (MRI), and diffusion tensor imaging (DTI), are fundamental to modern medical practices and research. These images serve as critical data sources for AI and VR applications, which are integral to strategic advancements in healthcare technologies. Consequently, the analysis and understanding of medical images are essential for driving forward medical innovations [34]. 3D image analysis, in particular, plays a pivotal role in accurately diagnosing and planning treatments by providing comprehensive views of anatomical structures. This technology enhances the precision of surgical procedures and radiotherapy by allowing for detailed visualization of complex anatomies. Additionally, 3D image analysis supports the development of personalized medicine by facilitating the study of individual anatomical and pathological variations. The future of healthcare increasingly relies on 3D medical images, which offer more depth and volume compared to traditional 2D images. This added dimension provides more detailed information about medical issues, significantly improving diagnostic accuracy and treatment efficacy [42].

3D biomedical image segmentation is a crucial aspect of 3D image analysis. It involves partitioning a 3D medical image into segments to isolate regions of interest, such as organs, tissues, or tumors. This segmentation is essential for various medical applications: enhancing diagnostic accuracy by clearly delineating anatomical structures and pathological regions [43], improving treatment planning by allowing for precise targeting of affected areas [23], and supporting personalized treatment plans by providing detailed images tailored to an individual's unique anatomy [11]. Moreover, segmentation enables quantitative assessments, such as measuring tumor volumes or monitoring disease progression, which are crucial for research and clinical decision-making. Segmented images also provide annotated datasets that are essential for training AI and machine learning models, driving innovations in automated diagnosis and treatment planning. In summary, 3D biomedical image segmentation is a key technology in modern healthcare, enhancing diagnostic accuracy, treatment precision, and the development of personalized medicine, while also supporting advancements in AI and machine learning, underscoring its critical role in advancing medical technology and improving patient care.

3.2 3D CNNs for Biomedical Image Segmentation

As discussed in Chapter 1, Convolutional Neural Networks are the best way to segment biomedical images. With advancements in neural network architectures, data augmentation techniques, and high-end GPUs, analyzing volumetric medical data with 3D deep learning has become feasible [34, 24]. The effective use of 3D CNNs emerged after the breakthrough success of AlexNet in 2012, which was made possible by advanced parallel computing architectures.[34].

While 1D CNNs extract spectral features and 2D CNNs extract spatial features from data, 3D CNNs can simultaneously capture both spectral and spatial features from input volumes. This capability makes 3D CNNs particularly useful for analyzing volumetric data in medical imaging. The mathematical formulation of a 3D CNN is similar to that of a 2D CNN, but with an additional dimension [34]. A 3D CNN-based segmentation model processes 3D images and produces 3D prediction masks, leveraging a network structure similar to standard 2D CNN models but extended to handle volumetric data. The convolution and pooling layers operate in three dimensions, allowing the model to capture spatial features across all axes. This capability makes 3D CNNs particularly effective for analyzing volumetric medical data [24].

Several medical image segmentation approaches use 2D deep learning methods, which effectively learn spatial relationships within a 2D plane. This works well for 2D medical images, however in 3D volumetric data such as MRI, CT, or USG, regions of interest often span multiple slices, making inter-slice information crucial. 2D CNNs cannot capture this inter-slice information, potentially limiting segmentation performance.

In contrast, 3D CNNs use convolution kernels in three dimensions, allowing them to learn spatial features across slices, resulting in better segmentation results for volumetric data [24].

3.3 Proposal: Expanding Region Growing Algorithm in 3D

In the medical imaging domain, creating a sufficiently large 3D dataset is highly challenging due to the invasive nature of some imaging techniques (e.g., CT), the extended duration required for imaging, and the labor-intensive process of annotating 3D data, [4]. This is where I want to leverage the idea of using patch based methods and Region Growing algorithm. We only require one image mask pair to use the patch based methods. The data created from that can be used to train a 3D CNN model, similar to the 2D CNN used in Chapter 2.

3.3.1 Preparation of 3D Training Data

The training dataset will be constructed from small 3D patches of size (80, 80, 80) now instead of (80, 80) in 2D. This involves using patch-based methods to create these mini volumetric tiles from the training data. Similarly, masks will be constructed in 3D patches of size (3,3,3) instead of (3,3). By utilizing these 3D patches, the model can learn spatial relationships across multiple slices, enhancing the segmentation performance for volumetric medical images such as MRI, CT, or ultrasound scans.

3.3.2 3D CNN Architecture for training 3D data

I am planning to use similar CNN architecture as used for 2D, 25-layer Neural Network that will consists of 5 double 3D convolutional layers activated by ReLU, each followed by a 3D max-pooling layer to progressively reduce the spatial dimensions while capturing essential features. The final double convolution layer is followed by a single 3D convolution to reshape the output into a size of (3, 3, 3, 1), where 1 indicates whether a voxel (a value on a grid in three-dimensional space, analogous to a pixel in 2D images) is in the foreground or background. This will allow the network to learn and predict the spatial relationships within the volumetric data effectively.

3.3.3 Region Growing Algorithm in 3D

In the testing phase of our 3D network evaluation, we propose using the 3D Region Growing Algorithm (RGA) for efficient segmentation of volumetric medical images. This approach ensures accurate results while optimizing processing time by focusing on contiguous regions rather than the entire volume. The 3D RGA operates by iteratively expanding around foreground voxels. Starting from initial foreground voxels, the algorithm grows by evaluating neighboring voxels in all three dimensions (x, y, and z), ensuring that the predicted regions remain contiguous. This method is particularly effective in handling the complex structures often present in medical volumetric data. The workflow begins with an initial set of foreground voxels identified by the trained 3D CNN model. For each iteration, the algorithm examines neighboring voxels around the current foreground voxels. A 3D patch, for instance, 80x80x80, around each central voxel is extracted and passed through the trained network to generate a probability map. This map predicts the likelihood of each voxel being part of the foreground or background. Voxels with a probability above a certain threshold (e.g., 0.1) are classified as foreground and added to the region. This process continues iteratively, expanding the region by adding adjacent voxels that meet the foreground criteria, thereby ensuring that only the necessary parts of the volume are processed. By focusing only on regions around foreground voxels, the algorithm significantly reduces computational load compared to processing the entire volume. This targeted approach allows for faster and more efficient segmentation, particularly beneficial for large volumetric datasets such as MRI, CT, or ultrasound scans.

4 Timeline

Semester	**Goal**
Fall 2024	• 3D Region Growing Algorithm • Look into measures of evaluating Connectivity • Write paper on 2D Region Growing
Spring 2025	Apply 3D Region Growing to 3D DSA data
Summer 2025	Possible Internship
Fall 2025	Look into Mathematical models for image segmentation
Spring 2026	Defend Thesis

References

[1] Arbelle, A., and Raviv, T. R. Microscopy cell segmentation via convolutional LSTM networks. *CoRR abs/1805.11247* (2018).

[2] Aslanzadeh, R., Qazanfari, K., and Rahmati, M. An efficient evolutionary based method for image segmentation, 2017.

[3] Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., and Blaschko, M. B. *Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory and Practice*. Springer International Publishing, 2019, p. 92–100.

[4] Chen, S., Ma, K., and Zheng, Y. Med3d: Transfer learning for 3d medical image analysis. *arXiv preprint arXiv:1904.00625* (April 2019).

[5] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics 36*, 4 (1980), 193–202.

[6] Guo, Y., and Ashour, A. S. 11 - neutrosophic sets in dermoscopic medical image segmentation. In *Neutrosophic Set in Medical Image Analysis*, Y. Guo and A. S. Ashour, Eds. Academic Press, 2019, pp. 229–243.

[7] Hassanpour, N., and Ghavami, A. Deep learning-based bio-medical image segmentation using unet architecture and transfer learning, 2023.

[8] Ibtehaz, N., and Rahman, M. S. Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation. *arXiv preprint arXiv:1902.04049* (2020).

[9] Jeans, W. D. The development and use of digital subtraction angiography. *The British Journal of Radiology 63*, 747 (1990), 161–168.

[10] John H. Lagregren and Erica M. Rutter and Kevin B. Flores. Region growing with convolutional neural networks for biomedical image segmentation.

[11] Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. A., Kane, A. D., Menon, D. K., Rueckert, D., and Glocker, B. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis 36* (2017), 61–78.

[12] Kayalibay, B., Jensen, G., and van der Smagt, P. Cnn-based segmentation of medical imaging data, 2017.

[13] Kimura, R., Teramoto, A., Ohno, T., Saito, K., and Fujita, H. Virtual digital subtraction angiography using multizone patch-based u-net. *Physical and Engineering Sciences in Medicine 43*, 4 (2020), 1305–1315.

[14] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations* (2015).

[15] Kirsten, L. N., and Jung, C. R. Cell tracking-by-detection using elliptical bounding boxes, 2023.

[16] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (2012), vol. 25, pp. 1097–1105.

[17] Li, M., Jiang, Y., Zhang, Y., and Zhu, H. Medical image analysis using deep learning algorithms. *Frontiers in Public Health 11* (2023), 1273253.

[18] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. A survey on deep learning in medical image analysis. *Medical Image Analysis 42* (2017), 60–88.

[19] Liu, X., Deng, Z., and Yang, Y. Recent progress in semantic image segmentation. *Artificial Intelligence Review 52*, 2 (2019), 1089–1106.

[20] Ltd., J. draw.io. Online diagramming tool.

[21] Luo, Y., and Sun, L. Digital subtraction angiography image segmentation based on multiscale hessian matrix applied to medical diagnosis and clinical nursing of coronary stenting patients. *Journal of Radiation Research and Applied Sciences 16*, 3 (2023), 100603.

[22] Meng, C., Sun, K., Guan, S., Wang, Q., Zong, R., and Liu, L. Multiscale dense convolutional neural network for dsa cerebrovascular segmentation. *Neurocomputing 373* (2020), 123–134.

[23] Milletari, F., Navab, N., and Ahmadi, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)* (2016), IEEE, pp. 565–571.

[24] Niyas, S., Pawan, S., Anand Kumar, M., and Rajan, J. Medical image segmentation with 3d convolutional neural networks: A survey. *Neurocomputing 493* (2022), 397–413.

[25] O'Shea, K., and Nash, R. An introduction to convolutional neural networks, 2015.

[26] Papanastasiou, G., Dikaios, N., Huang, J., Wang, C., and Yang, G. Is attention all you need in medical image analysis? a review. *IEEE Journal of Biomedical and Health Informatics 28*, 3 (Mar 2024), 1398–1411.

[27] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 779–788.

[28] Refaeilzadeh, P., Tang, L., and Liu, H. *Cross-Validation*. Springer US, Boston, MA, 2009, pp. 532–538.

[29] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015.

[30] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 815–823.

[31] Seo, H., Badiei Khuzani, M., Vasudevan, V., Huang, C., Ren, H., Xiao, R., Jia, X., and Xing, L. Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications. *Medical physics 47*, 5 (2020), e148–e167.

[32] Sharma, N., and Aggarwal, L. M. Automated medical image segmentation techniques. *Journal of medical physics/Association of Medical Physicists of India 35*, 1 (2010), 3.

[33] Shit, S., Paetzold, J. C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Pluim, J. P. W., Bauer, U., and Menze, B. H. cldice - a novel topology-preserving loss function for tubular structure segmentation. *Technical University of Munich and Eindhoven University of Technology* (2021).

[34] Singh, S., Wang, L., Gupta, S., Goli, H., Padmanabhan, P., and Gulyás, B. 3d deep learning on medical images: A review. *Sensors (Basel) 20*, 18 (Sep 2020), 5097.

[35] Singh, S. P., Wang, L., Gupta, S., Goli, H., Padmanabhan, P., and Gulyás, B. 3d deep learning on medical images: A review. *Sensors (Basel, Switzerland) 20*, 18 (2020), 5097.

[36] Suganyadevi, S., Seethalakshmi, V., and Balasamy, K. A review on deep learning in medical image analysis. *International Journal of Multimedia Information Retrieval 11*, 1 (2022), 19–38.

[37] Wang, R., Lei, T., Cui, R., Zhang, B., Meng, H., and Nandi, A. K. Medical image segmentation using deep learning: A survey. *IET Image Processing 16*, 5 (Jan. 2022), 1243–1267.

[38] Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging 9*, 4 (2018), 611–629.

[39] Yao, W., Bai, J., Liao, W., Chen, Y., Liu, M., and Xie, Y. From cnn to transformer: A review of medical image segmentation models, 2023.

[40] Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., and Gao, M. Techniques and challenges of image segmentation: A review.

[41] Zhang, M., Zhang, C., Wu, X., Cao, X., Young, G. S., Chen, H., and Xu, X. A neural network approach to segment brain blood vessels in digital subtraction angiography. *Computer methods and programs in biomedicine 185* (2020), 105159.

[42] Zhou, L., Fan, M., Hansen, C., Johnson, C. R., and Weiskopf, D. A review of three-dimensional medical image visualization. *Health Data Science 2022* (2022), 9840519.

[43] Özgün Çiçek, Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. 3d u-net: Learning dense volumetric segmentation from sparse annotation, 2016.