2024 IEEE First International Conference on Artificial Intelligence for Medicine, Health and Care (AIMHC) | 979-8-3503-7198-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIMHC59811.2024.00048

3D Reconstruction of the Carotid Artery from Handheld Ultrasound Videos

Abhishikta Bandyopadhyay†, Omkar Kulkarni†, Tim Oates
University of Maryland Baltimore County
Email: abhishb1@umbc.edu, omkark1@umbc.edu, oates@cs.umbc.edu

***Abstract*—Cardiovascular disease is one of the leading causes of death among people of all genders and races in the United States. According to the CDC, approximately 695,000 people died in the United States in 2021 due to poor cardiovascular conditions and approximately 17% of these deaths were due to a stroke. To increase patients' awareness and understanding of their cardiovascular health, this paper presents a method for extracting a 3D model of the carotid artery from videos taken with handheld ultrasound devices which, due to their relatively low cost, make it possible for primary care physicians to own and use them on a large number of patients. Technicians using the device move it up and down the neck, changing positions and angles frequently. Our approach extracts the artery from each frame using machine vision methods, reorders the frames spatially (from the bottom of the artery to the top) as opposed to the temporal order in the video, and then uses computer graphics methods to build a 3D model. The method used for reordering the frames yielded an accuracy of 86.34% when compared to the ground truth sequence. Our goal is to have an easily understandable representation of the state of the carotid artery to educate patients about their risks and thereby increase compliance with treatment.**

I. Introduction

The two carotid arteries are vital components of the cardiovascular system, supplying oxygenated blood to the neck and the head, including the brain. They start as a single vessel from the heart and bifurcate towards the upper end of the neck to form the internal and external carotid arteries.

Our goal in this work is to generate a 3D model of the carotid arteries from an ultrasound video to provide a clear indication of cardiovascular health. A common cardiovascular problem is Carotid Artery Stenosis. This occurs when a plaque buildup of fatty deposits clogs the artery, resulting in decreased blood flow to the brain and the head. Demonstrating a problem like this visually to patients could be a driver for them to take action, leading to an early and easily understandable diagnosis and increase compliance with treatment.

The Point-Of-Care Ultrasound (POCUS) device used in our dataset is the Butterfly iQ+. Each video is recorded by a skilled technician who scans the carotid artery multiple times from different angles. This paper presents a method for visualizing the exterior shape of the artery given these videos as inputs, which is a challenging, heretofore unsolved problem. Modifying the method to visualize the interior of the artery is left for future work.

† Equal contribution

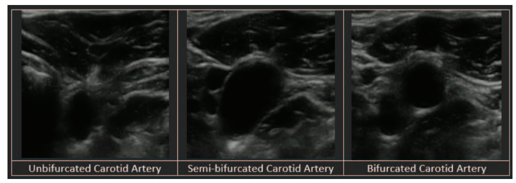

Fig. 1. The three phases of the carotid artery in the ultrasound video: unbifurcated, semi-bifurcated, and bifucated.

Figure 1 shows the transverse view of the artery as a circle or ellipse (unbifurcated) that begins to split apart (semi-bifurcated) and then becomes two circles/ellipses as the two component vessels branch off in different directions. This view makes it easier to detect and visualize plaque buildups on the artery walls. The goal is to take a video made by a technician, in which the device is moved up and down the neck of the patient, and extract a video such that the frames are ordered spatially, from the bottom of the artery to the top. One of the main challenges is detecting the exact position of the artery in the ultrasound video and determining when a frame is looking at the same part of the artery seen in other frames. Another problem is the low resolution and noisy nature of the videos, which makes detection of the artery difficult.

The approach used for this problem has three parts. The first is image segmentation, or finding the artery in a video frame. For this task we manually annotated six ultrasound videos of the carotid artery and trained deep neural networks to perform image segmentation. Each video consists of 449 frames, for a total of 2,694 frames to train and validate the models. Segmentation of the video frames generates masks over the transverse view of the carotid artery, and also draws contours around these masks. Given these masks, the second step is finding the correct order of frames according to the carotid artery's 3D structure.

We compare two approaches to ordering. In the first approach, a clustering algorithm finds frames that are probably from the same region of the artery in the video based on visual similarity. A model trained using the masks generated in the first part is then used to separate the clusters of frames into two classes: bifurcated and unbifurcated. Finally, a machine vision method uses Scale Invariant Feature Transform (SIFT) features of frames to first order the clusters and then the frames in each cluster to get the correct sequence from the bottom of the artery to the top, first working on the unbifurcated images

979-8-3503-7198-7/24/$31.00 ©2024 IEEE
DOI 10.1109/AIMHC59811.2024.00048

and then the bifurcated ones. In the second approach, which is simpler, we construct an ellipse around the mask for each frame and the attributes of each ellipse (e.g., area, perimeter, aspect ratio, and orientation) are compared to calculate a similarity score. This score is then used to order the frames, again using knowledge of how the shape of the artery changes as it bifurcates.

Given the segmentation masks, contours, and the sequence of frames from the above steps, we generate a 3D point cloud. Computer graphics algorithms are then used to compute normals on the point cloud and then surface reconstruction algorithms build a 3D model of the surface of the carotid artery.

To summarize, our main contributions are as follows:

1) Successful segmentation of the transverse view of the carotid artery in low-resolution ultrasound videos.
2) Creating and comparing two different approaches for generating sequential ultrasound video to evaluate which is more robust and accurate.
3) Building a 3D model of the artery using the segmentation masks and the ordering.

II. Related Work

A. 3D Ultrasound Reconstruction

There is some prior work on building 3D models from ultrasound videos. [15] describes a method to transform images and stack them into a 3D image cube, but its error rate was high and the target area of the body needed to be scanned multiple times. This prolonged the ultrasound procedure and was not feasible for all patients. [28] presents an approach to building 3D models from intravascular ultrasound to semi-automate the process of plaque detection. Intravascular ultrasound (IVUS) imaging is used along with X-ray coronary angiography to detect vessel pathologies, which makes it an intrusive procedure compared to ultrasound alone. Using methods from 3D ultrasound imaging [12], [16] attempts to study in vitro blood flow using three-dimensional color doppler ultrasound.

B. Carotid Artery Segmentation

Carotid artery segmentation is a long-standing problem and significant work has been done in this field. [25] is a good survey of much of this research up to 2006. It describes ten significant contributions to ultrasound segmentation, and claims that ultrasound segmentation using deep learning was first proposed back in 1999 [4]. Others have treated ultrasound segmentation as a spatio-temporal problem [13], which we discuss later in this paper. [24] presents a segmentation approach that takes into consideration missing object boundaries, as is the case with some of the instances in our dataset. [23] uses gray-level distribution and shape priors to find boundaries of the objects of interest. Except for [4], none of these methods use deep learning, and rely on classical computer vision algorithms such as contouring, edge detection, smoothing, etc., and encode prior information to eventually segment the object of interest in the ultrasound image.

A very recent algorithm presented in [14] shows impressive results without the use of machine learning. They use a dataset sourced from Brno University's SPLab, which is similar to our dataset. It contains 974 transverse and 84 longitudinal B-mode ultrasound images, respectively. They use a basis splines-based active contour method to find lumen and media adventitia boundaries in transverse as well as longitudinal B-mode ultrasound images, and their dice indices for segmentation of both boundaries in both types of images are over 92%.

One of the most recent approaches [34] uses contrast-enhanced MR angiography, not ultrasound, but only 61% of the segmentations were deemed to be usable. [5] worked on the same problem, and while the results may be impressive, their dataset is CT scans, which do not fall under Non-Destructive Testing (NDT) because they expose the body to radiation and, in the case of carotid arteries case, if focused on a highly sensitive region.

C. POCUS device for Ultrasound Videos

Our dataset is a group of videos that were recorded using the Point-of-Care Ultrasound device called the Butterfly iQ+. There are several papers highlighting the use of this device and its efficiency. [6] highlights the use of point-of-care ultrasound devices in the real world, citing their efficiency and accessibility. We found no significant prior work related to sequencing frames of handheld ultrasound videos.

III. Methodology

This section describes our processing pipeline, which has three steps:

1) carotid artery segmentation;
2) sequencing the frames in spatial order;
3) building a 3D model.

A. Carotid Artery Segmentation

1) Dataset, Annotations, and Preprocessing: As described earlier, each video in our dataset has 449 frames, with an average 25 fps rate. The videos did not indicate any changes in the orientation of the probe such as tilting or rocking therefore we were not able to assess the impact of such scenarios during our experiment. However, the data was gathered in a natural setting so there is some variation in orientation in the data. The video frames are first manually cropped from 480x640 to 280x500 resolution to remove extraneous information added by the device. Next, we use the VIA VGG [11] image annotator tool to draw ellipses on the artery walls. The artery typically goes through three stages when considering its trajectory from the heart to the head - unbifurcated, semi-bifurcated (or in transition), and bifurcated. The annotations are only ellipses, so they do not perfectly match the actual artery (especially in the transition stage). We later show that this inaccuracy does not affect segmentation performance. After annotation, the frames and annotations are resized to 384x384. The segmentation task has two classes: artery and background. Six videos were annotated, for a total of 2,694 frames, with five used for training (2,245 frames) and one for validation (449 frames). Even though our dataset contains videos consisting of 449 frames our code is capable of handling longer or shorter videos. In case the video is extremely large or there are too many frames we can always use clustering (described below in section III.B.1) to remove repetitive consecutive frames to generate an accurate 3D structure.

2) Model and Architecture: We compare the performance of two state-of-the-art segmentation architectures based on Convolutional Neural Networks (CNNs): Mask R-CNN [17] and U-Net [27]. Mask R-CNN is a popular architecture used for general-purpose classification, object detection, and segmentation tasks. It was originally trained on the ImageNet [9] and COCO [21] datasets for the ImageNet and COCO challenges. The Mask R-CNN pipeline starts with a ResNet [18] network which is followed by a Feature Pyramid Network (FPN) that serves as a Region Proposal Network (RPN). These networks propose a pre-determined number of regions for where the object(s) of interest might be in the image. One of the drawbacks of this architecture is that it is computationally expensive and slow, while also requiring significant space. We experimented with both ResNet-50 and ResNet-101 backbones.

U-Net [27] is a model architecture that shares some properties with Mask R-CNN[17] in that it includes ResNet-like skip connections. The unique property of this architecture is that it retains the features computed in earlier layers and uses them as as inputs to later layers. Individual blocks of this network, in addition to the input and output blocks, include double-convolutional blocks, downsampling blocks, and upsampling blocks. The double-convolutional block consists of two convolutional layers with the same number of output channels. Its output is sent to the downsampling block, where a max pooling layer performs downsampling. This is done sequentially four times. In the upsampling block, the output from the previous layer is upsampled, either using transposed convolution or interpolation. When upsampled, the output from the double-convolutional block corresponding to the dimensions of the newly upsampled volume is concatenated with the upsampled output. This is done sequentially until the original image resolution is restored. After the original resolution is restored, the output layer is simply a sigmoid activation layer with the same number of output channels as the number of classes, one channel for each class. For our problem there is just one channel as each pixel either belongs to the artery or doesn't. Fig 2 illustrates the U-Net architecture.

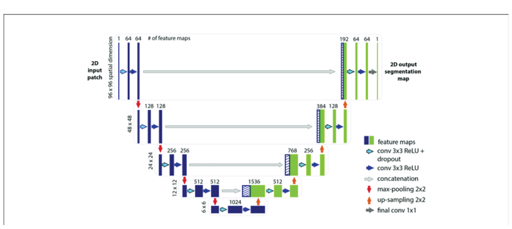

Fig. 2. The U-Net architecture is commonly applied to medical imagery.

B. Sequencing Frames in Handheld Ultrasound Videos

We compared two methods to sequence the frames in ultrasound videos to match the spatial structure of the artery as opposed to the temporal structure of the video.

1) Method 1: Clustering: After segmenting the ultrasound video, the next step is to identify frames that display the same area of the carotid artery. The first method does that by separating frames based on whether the artery is bifurcated, clustering frames within each category, and then sequencing the clusters based on centroids and then sequencing the frames in each cluster. These steps are described below.

Given that our goal is to sequence frames in spatial order from the bottom of the artery to the top, a preliminary step is to classify each frame as to whether it is bifurcated (top) or unbifurcated (bottom) to simplify downstream processing. CNNs were trained to solve this binary classification task, including pretrained VGG16, VGG19[30], and Resnet50 networks.

The features used by the classification models, after fine tuning on our data, were also used for clustering frames. The activations in the penultimate fully connected layer - 25,088 in VGG16 and VGG19, 100,352 in Resnet50 - were reduced further using Principle Component Analysis before clustering.

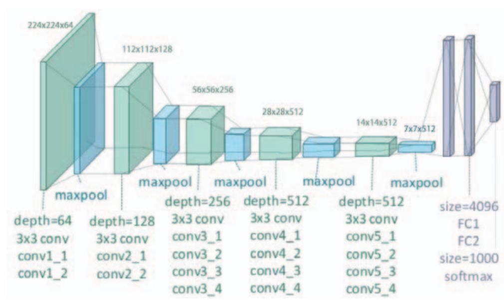

Fig. 3. Illustration of VGG19 Architecture

The clustering algorithms we used are k-Means and Gaussian Mixture Models (GMMs). k-Means is a method of vector quantization [2]. It aims to cluster n observations (in our case images) into k clusters in which each observation belongs to the cluster with the nearest mean. GMMs are another type of clustering algorithm that classifies data into different categories based on probability distributions [26]. GMMs are probabilistic models that assumes all the data points are generated from a mixture of Gaussian distributions with unknown parameters. We used full covariance matrices rather than assuming a diagonal covariance matrix.

Both k-Means and Gaussian Mixture models converge quickly to local optima and they both employ an iterative refinement approach. They both use cluster centers to model data. However, k-Means finds clusters of comparable spatial extent, while GMMs allow clusters to have different shapes [26]. We explored both to determine which was better suited to our task. The cluster size chosen in this case is 20. We arrived at this number by doing trial and error on our dataset and confirming that 20 clusters gave the most distinct frames but also did not miss important groups.

After this stage, we choose one image from each of the 20 clusters and start the process of putting them in the correct order. To ensure that the artery starts as unbifurcated and ends up being bifurcated we train a VGG16 model based on the

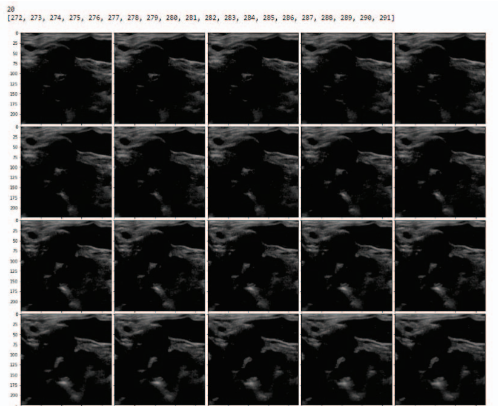

Fig. 4. Cluster obtained after implementing KMeans

As demonstrated in this image, the K-means algorithm clusters all the frames in the video that look identical or very similar. This helps us avoid repeated frames and simplifies the process of sequentialization.

masked images received from the U-Net to classify if the artery is bifurcated on unbifurcated. After this step, the 20 images (one from each cluster) are separated into two classes. Then we apply Scale Invariant Feature Transform (SIFT) to order the frames with respect to their similarity. SIFT is a computer vision algorithm to detect, describe, and match local features in images. SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on the Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of three or more features that agree on an object and its pose is then subject to further detailed model verification and, subsequently, outliers are discarded. Finally, the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence [22]. The motive behind using SIFT is that the parts of the artery closer to each other spatially will appear more visually similar than parts further away. Hence, after applying SIFT, we get the correct order of frames. After this, we can regenerate the video with that ordering. .

2) Method 2: Ellipse Features: : This method uses a more mathematical approach relying heavily on the attributes of the ellipse matching the artery such as the center coordinates, major axis, minor axis, and angle.

Cropping and Segregation: The first problem that we tackle in this approach is fixing the lateral shift of the artery in the ultrasound videos by locating the artery mask obtained from segmentation and extracting only the ellipse which indicates

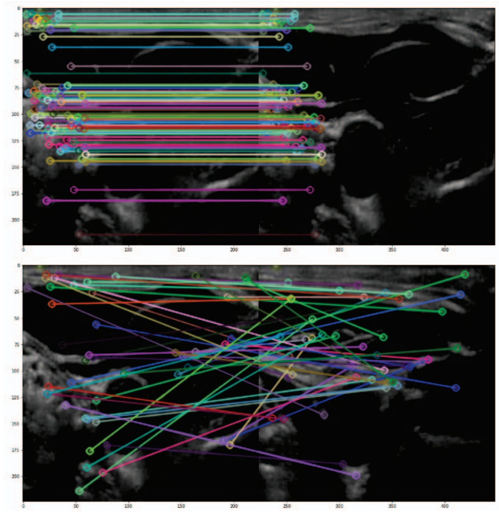

Fig. 5. Demonstration of Scale Invariant Feature Transform

the border of the carotid artery. This helps to get rid of the extra noise in the form of tissues scanned during the ultrasound. This process also helps segregate the images into bifurcated and unbifurcated categories based on the number of ellipses detected in each image.

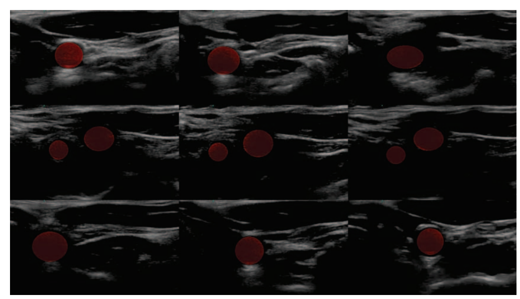

Fig. 6. Ultrasound Video Frames with Masks

Calculating the Similarity of Ellipses: We chose 4 different factors to calculate the similarity of two ellipses which are area similarity, perimeter similarity, aspect ratio similarity, and orientation similarity. All four similarity scores have equal weight and the sum of these factors yields the optimal similarity score for each ellipse pair. The similarity score is stored in an adjacency matrix which makes it easier to compare each ellipse with the other.

Sorting and Sequencing the Frames: To sort the adjacency matrix we used a modified version of topological sort. It starts at the most elongated ellipse which is on the verge of bifurcating and continues back to the most dissimilar ellipse which is the starting point of the artery. In the case of bifurcated artery frames, since it is common knowledge that the internal and external arteries move further apart as we move away from the common carotid artery, we have computed the distance between the two ellipses and sorted them accordingly. This gives us the complete sequence of

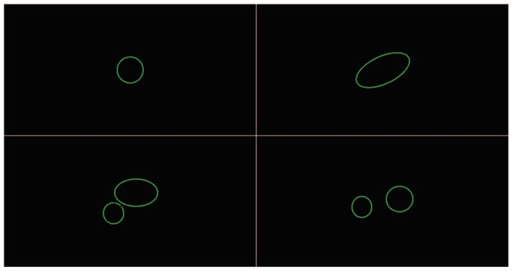

Fig. 7. Ultrasound Video Frames Displaying the Ellipses after Fixing the Lateral Shift

frames starting at the most bifurcated point in the artery to the starting point, i.e., the unbifurcated artery.

C. Generating a 3D visualization of the artery using the segmentation masks and the ultrasound video orderings

For building the 3D model, we need the orderings obtained from the previous step as well as the contours of the segmentation masks. After arranging the frames according to the ordering along with the mask contours, we stack them up and obtain the co-ordinates of all the points that form the walls of the carotid artery. We then use the open3d [33] package to generate a 3D point cloud. We then perform some pre-processing steps, such as cleaning the point cloud, and removing outliers and other asymmetrical points. This preprocessing step prevents unpredictable behaviour in the steps that follow, and ensures a smoother and more aesthetic end result.

After this, we experimented with two approaches to prepare for the final step of surface reconstruction. For the first approach we split the artery point cloud along the Z-axis from where it starts bifurcating into a few separate point clouds with some overlap, and separately compute a convex hull [3] on each of these point clouds. Having multiple convex hulls is especially useful in this scenario as it defines the artery while considering the bifurcation.

The other was to compute normals [20] on the final set of points. An important parameter to consider is the number of neighboring vertices to be considered when computing the normals for a given vertex. We conducted experiments using a variety of values, ranging from a minimum of 5 to a maximum of 500 neighbors.

Both of these steps were followed by a screened poisson surface reconstruction [19] to get to the penultimate step of generating the 3D model. Here we experimented with the reconstruction depth and octree depth parameters to find a combination that generated a surface that included a large portion of the point cloud while being fairly smooth. This was then followed by surface smoothing algorithms, such as Laplacian smoothing [31], and Taubin smoothing [32].

To perform these tasks, we experimented with copmuter graphics softwares such as the visualization toolkit (VTK) python library [29], Blender [8], and Meshlab[7]. We determined that Blender was an overkill for our problem, and while using VTK with python was useful, the preprocessing and cleaning of the point cloud was not easy. With Meshlab, we were able to manually preprocess our point cloud, and progressively visualize the 3D model with every step.

IV. Results

A. Artery segmentation

We use the frames of five videos for training and one video for validation. The quantitative metrics we have used are mask accuracy and network loss. For the U-Net[27], we have used the dice coefficient[10] as an additional metric for validation. We have used the same hardware and kept the hyperparameters (such as batch size) consistent across the models to make a fair comparison.

All three network architectures were trained on a NVIDIA RTX 6000 GPU with 24 GB VRAM. The batch size was fixed to 8 images. One epoch for the Mask R-CNN[17] is 100 train steps, which is equivalent to 800 randomly chosen training examples per epoch. For the U-Net, however, one epoch was one full run of the dataset, which is equivalent to 2245 training examples per epoch. Figures 8, 9, 10, and 11 visualize the performance of these models. We modified matterport's[1] implementation of the Mask R-CNN to suit our problem

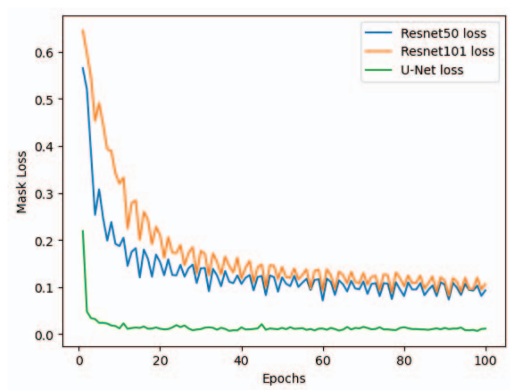

Fig. 8. Mask Loss vs epochs for all three architectures

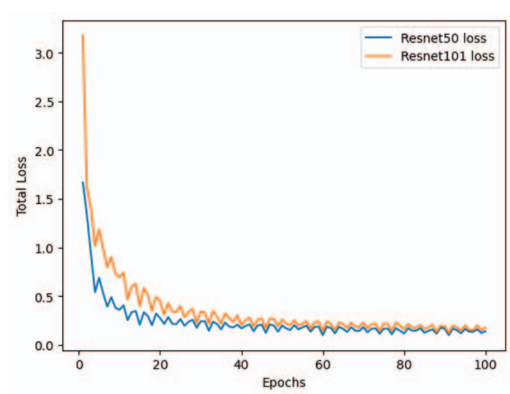

Fig. 9. Total Loss vs epochs for Mask-RCNN architectures

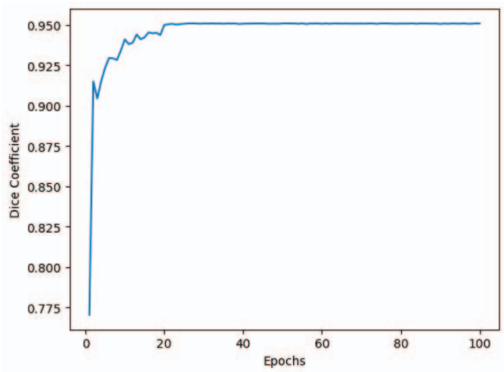

Fig. 10. U-Net dice coefficient vs epochs

Architecture	Train Time per Epoch	Video inference time
Resnet50	109s	56.684s
Resnet101	127s	58.211s
U-Net	46s	10.276s

Fig. 11. Architecture Performance

Fig 8 shows that the U-Net loss decreases a lot faster than that of the Mask R-CNN, and just by the twentieth epoch it starts plateauing out. However, the Mask RCNN continues to learn even by the end of the hundredth epoch. In our experiment, we trained the Mask RCNN for upto 1000 epochs, and that was when the mask validation loss plateaued at around 0.065; meaning that it takes 900 epochs of training for the batch loss value to decrease by an additional 0.035. This is very resource intensive compared to the U-Net.

Between Resnet50 and Resnet101 backends, we see that both perform similarly when it comes to validation loss, as well as the quality of the generated masks. We conclude that using the Resnet101 backbone is overkill.

The measured validation mask accuracy of the Mask RCNN doesn't exceed 83% either for any epoch, but the U-Net does significantly better, exceeding 94% after training for just 40 epochs.

After a qualitative analysis of the results (Fig 12 and 13), we conclude that the U-Net performs significantly better than the Mask R-CNN for this ultrasound segmentation task. The U-Net lives up to its purpose of specializing in medical imagery segmentation as mentioned in the original U-Net paper.

Also, the U-Net is more robust to inaccurate annotations. The U-Net was able to detect masks accurately in the connected region in the frame where the carotid artery had started bifurcating, even though the provided annotations did not include that region.

The U-Net was trained with a batch size of 8, however, we could have easily fit close to 80 images in a single batch on the same GPU. This could have enabled even faster training.

B. Frame Sequentializing

After doing a comparative study on which method yielded the best results we observed that the accuracy of the first method was 69.89% whereas that of the second method was 86.34%. We have been able to calculate this value by calculating the displacement of each frame to the sequence in the ground truth video. If a frame in the generated result is present in the 3rd position whereas in the ground truth video, it is present in the 1st position then 2 is added to the error. Thus, we have been able to calculate the accuracy percentage of both methods. The total error values obtained for method 1 is 60699 and method 2 is 27551. The maximum error possible is 201601.

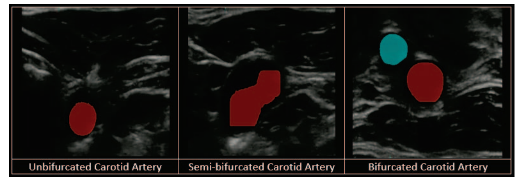

Fig. 12. Masks generated on Carotid Artery in different phases by Mask-RCNN

Fig. 13. Masks generated on Carotid Artery in different phases by U-Net

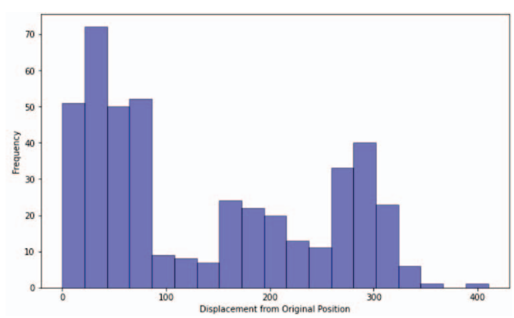

Fig. 14. This histogram shows the displacement of each frame from its original position for Method 1

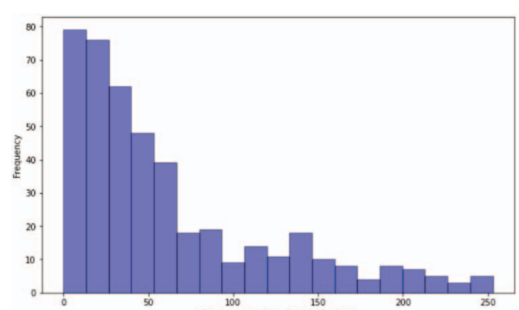

Fig. 15. This histogram shows the displacement of each frame from its original position for Method 2

By observing these 2 graphs we can conclude that Method 2 has performed better based on 2 major factors. In the first graph, we can see that the displacement of the frames that are not present in the correct position is majorly present between 0 and 80 and also 250 to 300 whereas, in the second graph, we can see that most of the frames that are not present in the correct position have an error between 0 to 50. Secondly, the range of x axis which indicates the displacement of each frame from the correct position is much larger in the first graph i.e. 400 as compared to the second graph i.e. 250.

C. 3D model of the Carotid Artery

After implementing the steps outlined in the methodology section, we were able to generate a comprehensible visualization of the carotid arteries.

Computing separate convex hulls[3] was comparatively much faster than computing normals, however this advantage is somewhat nullified due to the manual intervention when splitting the original point cloud. Using this method also reduces the effect that the reconstruction depth and octree depth parameters have during the final stage of surface reconstruction. Fig 16 shows an example of the visualization when using the convex hull approach. Computing normals[20] for the point cloud, however is more effective for this particular scenario. Meshlab[7] defaults to just ten neighbors for computing normals, and having just ten neighbors leaves out a large section of the vertices not having their normals computed. We experiment with an even lesser number of neighbors, and as many as up to 1000 neighbors, and we conclude that having 500 neighbors is sufficient to compute the normals for most of the vertices accurately. This ensures that the surface reconstruction algorithm covers all these vertices.

Fig. 16. 3D Artery visualization using the convex hull approach

We experiment with the reconstruction depth and octree depth parameters, and a reconstruction depth of 4 and an octree depth of 4 suffices to create the desired artery surface, and requires so little compute that it's done almost instantly. However, this surface is somewhat edgy and blunt, and also not smooth. A reconstruction depth of 5 and an octree depth of 8 generates a sufficiently smooth surface that incorporates all the intricacies, while going easy on the computation required; the required time for this is around 57 seconds. A reconstruction depth of 6 combined with an octree depth of 8 results in a surface that is unnecessarily intricate for a patient's visualization needs while also requiring significantly heavier compute; the required time for this is about 320 seconds. Hence, our pick for this problem is the second configuration. Fig 17 has the artery surfaces as per each of the described configurations.

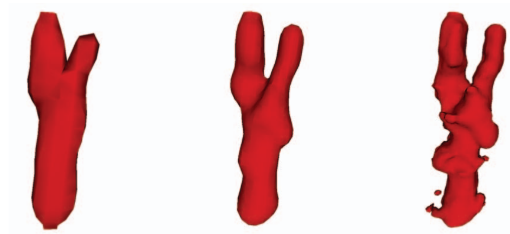

Fig. 17. 3D visualization of the artery with different reconstruction and octree depth configurations; from left to right (4,4), (5,8), (6,8)

V. Limitations and Future Scope

Our work is the first that we know that is an end-to-end pipeline for segmentation and sequencing of the artery from ultrasound data. We faced a few roadblocks, elaborated as follows.

The dataset is derived from a POCUS device, and the quality of the videos is blurry and noisy. To enhance the quality of the image, changes in contrast, brightness, sharpness, structure, and exposure need to be made to obtain clearer boundaries. The tissues moving in the ultrasound videos interfere with the clustering even though the artery position and width remain unchanged. Opting for advanced image enhancement techniques, such as AI-based noise reduction, to improve image quality would create the risk of losing crucial information hence we have opted for minimal image enhancement techniques.

Another challenge is the lateral movement of the device when the ultrasound is being taken. This leads to a shift in the location of the artery we see in the video, and we need to come up with algorithms that could find this shift so we can take that into account when processing the 3D structure.

In the future, we can consider using segmentation algorithms that are more geared towards video segmentation and that make use of the temporal nature of the data. This may lead to a more accurate visualization. Further fine-tuning of hyperparameters in K-Means algorithm can help improve the accuracy of the first frame sequentializing method. We also plan on correlating the results with ground truth data which will be available in the future, for now, we have focused on building the pipeline for solving the problem.

Another useful extension of this could be to detect plaque in the Carotid Artery as well and incorporate it in the 3D model, thus providing a visual representation of the state of the artery.

VI. Conclusion

In this paper, we have demonstrated that we can successfully reconstruct the sequential ultrasound video of the carotid artery from the original ultrasound video. We have successfully performed segmentation on the carotid artery and also have been able to find the correlation between the frames of the ultrasound video. We have effectively mapped each frame in the video to a particular part of the carotid artery and removed repetitive and redundant frames. We have been able to organize the frames in the correct order to move from the base of the neck up to the head and generate the ultrasound video that now moves in a single direction. We have demonstrated how we can generate a 3D model of the carotid artery using the orderings and the segmentation masks, and how this can be extended to modeling plaque in the carotid artery as well.

References

[1] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 5

[2] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. 3

[3] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, dec 1996. 5, 7

[4] T Binder, M Süssner, D Moertl, T Strohmer, H Baumgartner, G Maurer, and G Porenta. Artificial neural networks and spatial temporal contour linking for automated endocardial contour detection on echocardiograms: a novel approach to determine left ventricular contractile function. *Ultrasound in medicine & biology*, 25(7):1069–76, Sep 1999. 2

[5] Gerda Bortsova, Daniel Bos, Florian Dubost, Meike W. Vernooij, M. Kamran Ikram, Gijs van Tulder, and Marleen de Bruijne. Automated segmentation and volume measurement of intracranial internal carotid artery calcification at noncontrast ct. *Radiology: Artificial Intelligence*, 3(5):e200226, 2021. 2

[6] Stephanie L Burleson, Joseph F Swanson, Eric F Shufflebarger, Matthew E Lissauer, and Samuel C Chen. Evaluation of a novel handheld point-of-care ultrasound device in an african emergency department. *Ultrasound Journal*, 12(1):53, 2020. 2

[7] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. 5, 7

[8] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 3

[10] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 5

[11] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, New York, NY, USA, 2019. ACM. 2

[12] A. Fenster and D.B. Downey. 3-d ultrasound imaging: a review. *IEEE Engineering in Medicine and Biology Magazine*, 15(6):41–51, 1996. 2

[13] N. Friedland and D. Adam. Automatic ventricular cavity boundary detection from sequential ultrasound images using simulated annealing. *IEEE Transactions on Medical Imaging*, 8(4):344–353, 1989. 2

[14] J. H. Gagan, Harshit S. Shirsat, Grissel P. Mathias, B. Vaibhav Mallya, Jasbon Andrade, K. V. Rajagopal, and J. R. Harish Kumar. Automated segmentation of common carotid artery in ultrasound images. *IEEE Access*, 10:58419–58430, 2022. 2

[15] David G. Gobbi and Terry M. Peters. Interactive intra-operative 3d ultrasound reconstruction and visualization. In Takeyoshi Dohi and Ron Kikinis, editors, *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002*, pages 156–163, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 2

[16] Z Guo, M Moreau, DW Rickey, PA Picot, and A Fenster. Quantitative investigation of in vitro flow using three-dimensional color doppler ultrasound. *Ultrasound in Medicine & Biology*, 21(6):807–816, 1995. 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 3, 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[19] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), jul 2013. 5

[20] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Transactions on Graphics*, 21(3):10 p, 2002. Special issue : Proceedings of ACM SIGGRAPH 2002. 5, 7

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 3

[22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 4

[23] Max Mignotte, Jean Meunier, and Jean-Claude Tardif. Endocardial boundary e timation and tracking in echocardiographic images using deformable template and markov random fields. *Pattern Analysis & Applications*, 4(4):256–271, Nov 2001. 2

[24] Karol Mikula, Alessandro Sarti, and Fiorella Sgallari. *Co-Volume Level Set Method in Subjective Surface Based Medical Image Segmentation*, pages 583–626. Springer US, Boston, MA, 2005. 2

[25] J.A. Noble and D. Boukerroui. Ultrasound image segmentation: a survey. *IEEE Transactions on Medical Imaging*, 25(8):987–1010, 2006. 2

[26] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009. 3

[27] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 3, 5

[28] Roberto Sanz-Requena, David Moratal, Diego Ramón García-Sánchez, Vicente Bodí, José Joaquín Rieta, and Juan Manuel Sanchis. Automatic segmentation and 3d reconstruction of intravascular ultrasound images for a fast preliminar evaluation of vessel pathologies. *Computerized Medical Imaging and Graphics*, 31(2):71–80, 2007. 2

[29] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit (4th ed.)*. Kitware, 2006. 5

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 3

[31] Olga Sorkine. Laplacian Mesh Processing. In Yiorgos Chrysanthou and Marcus Magnor, editors, *Eurographics 2005 - State of the Art Reports*. The Eurographics Association, 2005. 5

[32] G. Taubin. Curve and surface smoothing without shrinkage. In *Proceedings of IEEE International Conference on Computer Vision*, pages 852–857, 1995. 5

[33] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 5

[34] Magnus Ziegler, Jesper Alfraeus, Mariana Bustamante, Elin Good, Jan Engvall, Ebo de Muinck, and Petter Dyverfeldt. Automated segmentation of the individual branches of the carotid arteries in contrast-enhanced mr angiography using deepmedic. *BMC Medical Imaging*, 21(1):38, Feb 2021. 2