SMRVIS: Point cloud extraction from 3-D ultrasound for non-destructive testing

Lisa Y.W. Tang

June 5, 2023

Abstract

We propose to formulate point cloud extraction from ultrasound volumes as an image segmentation problem. Through this convenient formulation, a quick prototype exploring various variants of the Residual Network, U-Net, and the Squeeze and Excitation Network was developed and evaluated. This report documents the experimental results compiled using a training dataset of five labeled ultrasound volumes and 85 unlabeled volumes that got completed in a two-week period as part of a submission to the open challenge "3D Surface Mesh Estimation for CVPR workshop on Deep Learning in Ultrasound Image Analysis". Based on external evaluation performed by the challenge's organizers, the framework came first place on the challenge's Leaderboard. Source code is shared with the research community at a public repository.

Keywords: Ultrasound volumes; Non-Destructive Testing; Pipe; Manufacture Defects; Attention U-Net; Recurrent-Residual U-Net; Squeeze-Excitation U-Net; U-Net++; W-Net

1 Introduction

As part of a submission to an open challenge entitled "3D Surface Mesh Estimation for Computer Vision and Pattern Recognition workshop on Deep Learning in Ultrasound Image Analysis", this report presents experimental results compiled using a training dataset of five labeled ultrasound volumes and 85 unlabeled volumes. Due to resource and time constraints [3], we focus on finding a workable solution that can be quickly prototyped and tested. Accordingly, we propose to formulate three-dimensional mesh estimation from ultrasound volumes as an image segmentation problem. Source code is shared with the research community at `https://github.com/lisatwyw/smrvis`. We nicknamed this proposed framework as SMRVIS (Surface Mesh Reconstruction Via Image Segmentation) partly to draw an analogy to an older framework called SMRFI [7] that formulated shape matching as a feature image registration problem. Unlike this prior work that embedded 2D shapes with feature values and assigned the computed feature values to the nearest voxel of the embedding space, the present framework simply encodes the positions of the reference mesh vertices with values in the range of [0, 1].

Rather than performing segmentation in 3D, our current approach performs a series of segmentations of thin sections by treating overlapping consecutive slices as 3-channel inputs. Our mesh-embedding scheme does not limit us to the deployment of two-dimensional models only. However, as the number of labeled data in this challenge dataset is relatively small (n =5), the choice of two-dimensional models allowed us to adopt a patch-based training approach, which has countless success cases as observed in previous open challenges [3]. To this end, we have explored and evaluated W-Net, R2 U-Net, SE U-Net, Attention U-Net, and U-Net++, which we review briefly in the next section.

1.1 Background

Nowadays, a vast majority of segmentation frameworks employ an encoder-decoder neural network structure that is popularized by the U-Net architecture [9] developed in 2015. In this architecture, network components called **skip connections** allow data from down-sampling layers to be rerouted back to the up-sampling layers.

Since its initial success, researchers have proposed countless variants of U-Net to enhance its performance [24, 25]. For instance, in 2016, U-Net was extended to 3-dimension via V-Net that uses convolutional layers in place of up-/down- sampling max-pooling layers as the former can better retain information than max-operations would. Xia et al. [15] also developed the W-Net in 2017 for unsupervised image segmentation. Their approach joins two fully convolutional neural network (CNN) branches with an autoencoder such that the first branch would encode input data into a fuzzy segmentation while the second branch would reconstruct the fuzzy segmentation to input data. Since its proposal, the W-Net is shown to have numerous successful applications such as retinal

vessel segmentation [10] and generation of Chinese characters [12].

In 2017, Hu et al. further enhanced U-Nets by incorporating the **squeeze and excitation blocks** at the end of each convolutional block in order to enhance the inter-dependencies between channels. In 2018, Oktay et al. proposed to incorporate an **attention module** within each skip connection that will drive the overall model to focus on input regions that garner more importance. Around the same time, Zhou et al. proposed U-Net++ with at least two novel features: 1) aggregating features across different semantic levels (i.e. scales) via new skip connections built of **dense convolution blocks**; 2) deep supervision [16]. Alom et al. [13] proposed the recurrent residual U-Net (R2Unet), which combines the strengths of recurrent connections and residual networks and was shown to have improved the quality of the feature representations that they could produce [13].

In 2022, Kugelman et al. [20] conducted a comparative study to benchmark some of the aforementioned U-Net variants in the context of retinal tissue extraction from optical coherence tomography and recommended the adoption of R2-U-Nets. We adopted their opensource implementation in this work; code listing 1 provides an abstraction of a basic U-Net implementation.

Listing 1: Schematics of basic U-Net

```
from tensorflow.keras.layers import BatchNormalization, Add, Multiply, Concatenate
from tensorflow.keras.layers import Input, ConvND, ConvNDTranspose
from tensorflow.keras.layers import GlobalAveragePoolingND, MaxPoolingND
from tensorflow.keras.models import Model

# above function names are abstractions only (ND instead of 2D or 3D)

def stack_bn_act( x, NF, KS ):
  for i, ks in enumerate( KS ):
    x = Conv2D( NF, ks, padding='same' )(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
  return x

def conv_block( x, NF, KS ):
  o = Conv2D( NF, kernel_size= KS[0], padding='same' )(x)
  o = BatchNormalization()(o)
  o = Activation( 'relu' )(o)
  o = stack_bn_act(o, NF, KS[1:]) # defined above
  return o

def deconv_block( x, NF, KS=2 ):
  o = Conv2DTranspose( NF, KS, padding='same' )(x)
  o = BatchNormalization()(o)
  o = Activation( 'relu' )(o)
  return o

# ---------------- hyperparameters tested in ablation study
AC, BS, NF = 'sigmoid', 8, 16
NX, NY NDIM, n_slices= 224, 224, 2, 3
ks=[(1,1)]*len(nfilters)

inp = Input( (NX, NY, 3) )

o1 = conv_block( inp, NF,    ks_s ); p1 = MaxPooling2D()( o1 )
o2 = conv_block( p1,  NF*2,  ks_s ); p2 = MaxPooling2D()( o2 )
o3 = conv_block( p2,  NF*4,  ks_s ); p3 = MaxPooling2D()( o3 )
o4 = conv_block( p3,  NF*8,  ks_s ); p4 = MaxPooling2D()( o4 )
o5 = conv_block( p4,  NF*16, ks_s ); p5 = MaxPooling2D()( o5 )

o6 = conv_block( Concatenate()( [deconv_block( o5, NF*8, strides=(2,2) ), o4]), NF*8, ks)
o7 = conv_block( Concatenate()( [deconv_block( o6, NF*4, strides=(2,2) ), o3]), NF*4, ks)
o8 = conv_block( Concatenate()( [deconv_block( o7, NF*2, strides=(2,2) ), o2]), NF*2, ks)
o9 = conv_block( Concatenate()( [deconv_block( o8, NF*1, strides=(2,2) ), o1]), NF*1, ks)

out = Activation( AC )( Conv2D( n_slices, 1 )( o9 ) )
model = Model(inp, outputs=out )
```

2 Methods

2.1 Materials

A training set of 90 ultrasound volumes were provided by the challenge organizers. Each scan captures piece(s) of a steel pipe, potentially containing artifacts inside these pipes. Corresponding Surface mesh of the pipe (pieces) were created by an "experienced data analyst" [6]. Five of these surface meshes (corresponding to volumes 1 to 5) were provided to the challenge participants and herein referred to as reference masks. Figures 1-5 show examples of the surface renderings of these reference meshes. As the reference labels for the remaining 85 volumes were not provided at the time of challenge, these volumes were mainly used in this study as test samples.

2.2 Preprocessing

Each of the reference mesh was first encoded into an image representation. To do so, the vertices of each mesh were read into memory. Next, a binary mask was created to encode the mesh vertices, taking into account the voxel spacing of the ultrasound volumes. To facilitate learning, the point cloud mask was dilated so that the edges of the binary mask softens. An alternative approach would be to apply Gaussian blur [7] but we found this simpler approach sufficient and computationally more efficient.

To read in the corresponding ultrasound volume, a meta file was created for each of the raw ultrasound files (example in Sec. B; script is also provided under `https://github.com/lisatwyw/smrvis/blob/main/utils/write_meta.sh`) and Python package `SimpleITK` was used (example code snippet follows). To ensure that the fitted models would be robust to noise, we only preprocessed the input ultrasound data with two steps: down-sampling the image resolution and rescaling their intensity values to [0,1].

Listing 2: Code listing for reading data using SimpleITK and Plydata

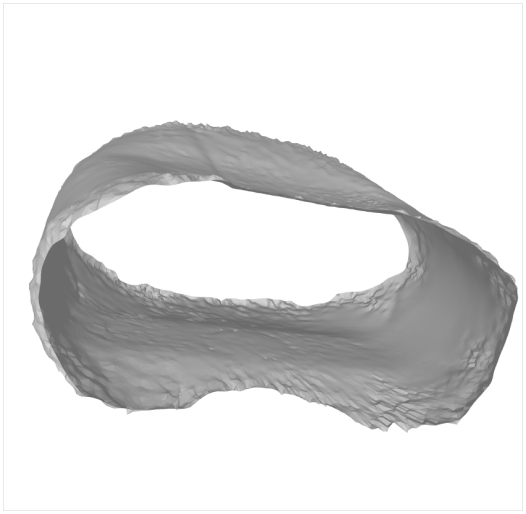
```
import SimpleITK as sitk

vols={}
i = 1 # sample_id
filename='train_data/training/volumes/scan_%03d.mhd'% i

header=sitk.ReadImage(filename)
vols[i]=sitk.GetArrayFromImage(header)

from plyfile import PlyData
plydata,verts,faces={},{},{}

filename='train_data/training/meshes/scan_%03d.ply' % i
vx = plydata[i]['vertex']['x']
vy = plydata[i]['vertex']['y']
vz = plydata[i]['vertex']['z']
verts[i] = [ (vx[d],vy[d],vz[d]) for d in range(len(vx)) ]
num_faces= plydata[i]['face'].count
faces[i] = [ plydata[i]['face'][d][0] for d in range(num_faces) ]
```

Listing 3: Code listing for reading in mesh data and generating their screenshots with Pyvista.

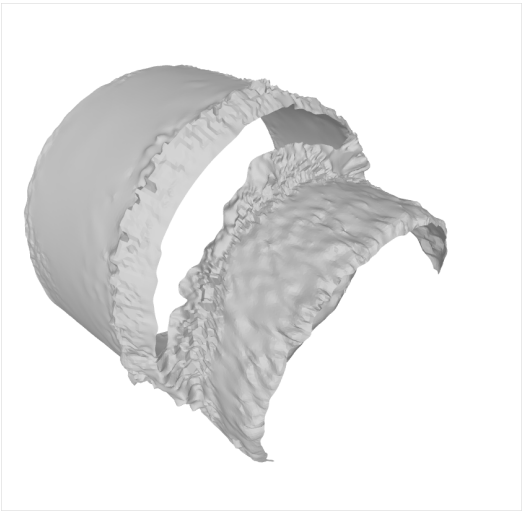
```
import pyvista
# we need to start the frame buffer even if plotting offline
pyvista.start_xvfb()
plotter = pyvista.Plotter(off_screen=True)
mesh = pyvista.read(ply_filename)
plotter.add_mesh( mesh, opacity=.3, color='grey' )
plotter.add_title('Estimated mesh for volume #%d'%i)
plotter.show( screenshot = out + '_screenshot.png')
```

2.3 Models

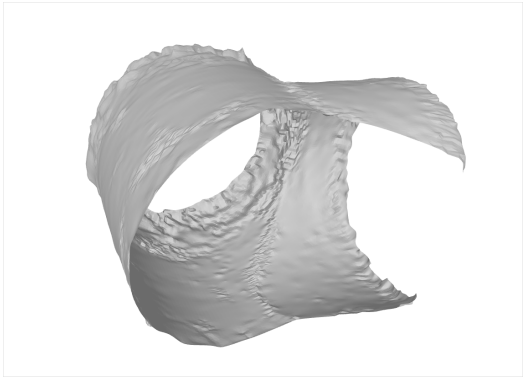
Based on observations and evidence from prior studies [20, 8], we elected to explore the following variants of the U-Net: W-Net (Wnet), the recurrent-residual U-Net (R2-Unet), Squeeze and Excitation U-Net (SE-Unet), U-Net++, and an Attention U-Net (Att-Unet).



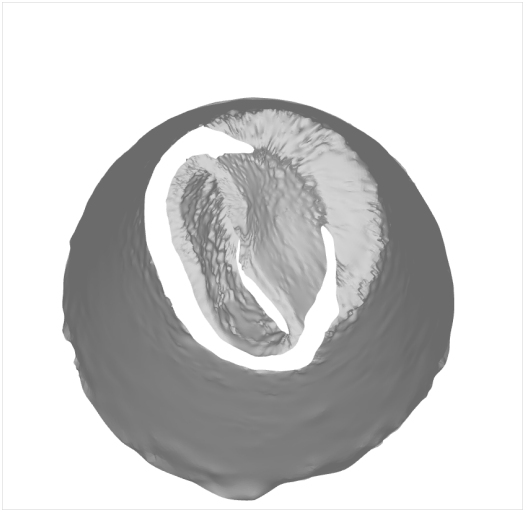
(a) Reference mesh of volume 1



(b) Reference mesh of volume 2



(c) Reference mesh of volume 3



(d) Reference mesh of volume 4

Figure 1: Reference meshes provided by the challenge organizers.

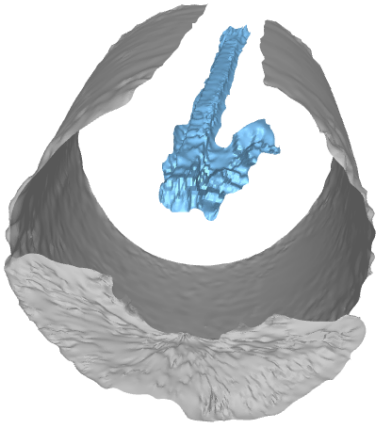


Figure 2: Surface rendering of a pipe (grey) containing artifacts (blue).

2.4 Training setup

We divided the provided labeled meshes with three non-overlapping subsets: validation set is used for early-stopping using slices from one volume while the test set is constructed from another volume for evaluation of the training progress, with the remaining volumes all used for training. For instance, a trial might involve volume 5 as the test volume, volume 4 as the validation set, and volumes 1-3 as the training set.

2.4.1 Data augmentation

We augmented the training dataset by perturbing the input training set on-the-fly with random translations, rotations, and mirroring on the x- and y-axes. We examined two approaches: applying these different types of transforms simultaneously, or exclusively. We also explored the effects of a training schedule wherein training volumes that have been reoriented entirely are only presented at a later phase.

2.4.2 Optimization

We employed the Adaptive Moment Estimator (ADAM) optimization algorithm with default parameters and explored values of 0.0001, 0.008, 0.001, and 0.1 as the initial learning rate in the context of three learning rate scheduling schemes, namely, cyclical decay, cosine decay, and polynomial decay [11].

Training was permitted to run for 300 epochs or terminated early when there is no reduction in the metrics computed on the validation set. We empirically explored the use of Dice and Jaccard as the validation metric but found training diverged when these two metrics were used, most likely due to scarcity of mesh vertices in relation to the its enclosing volume. We thus elected to use the same loss function (but computed on the validation set) for determining the termination criteria. Example progress plots are shown in the Appendix.

2.5 Ablation studies conducted

We conducted ablation trials to answer design questions surrounding the following components and present comparisons in the Results section and the Appendix.

1. Image resolution (IR): how might the image resolution used to train the models affect accuracy? (According to a 2019 review [5], ultrasound scanners typically acquire data that will be reconstructed to matrix size of 512 $\times$ 512 with 256 intensity levels. Hence, we explored the standard resolution of 256 $\times$ 256 and the non-standard resolution of 384 $\times$ 384, which correspond to down-sampling factors of three and two, respectively);

2. Activation function (AC): prior work advocate [17] the use of sigmoid, hard sigmoid, and hyperbolic tangent function based on datasets involving magnetic resonance and computed tomography data; do results generalize to ultrasonic data?

3. Loss function (LS): previous studies [17, 23, 18] have observed inter-plays between the loss function and activations for image classification; we hypothesize that their results may not generalize to ultrasound point extraction and hence explore Dice, binary cross entropy (BCE) and binary focal cross entropy loss terms (BFCE);

4. Selection scheme (SE): should all ultrasound sections be presented to the model during training (SE=1) or only sections containing the reference mesh (SE=2) be presented to ease training?

5. Random transform (RT): how the type(s) of random transformations affect the training progress? e.g. should all random transforms be allowed or only one type at a time?

6. Mesh encoding (EN): how should the mesh vertices be encoded in the image space? Would attenuating boundaries of the surface mesh help training when this encoding strategy is used in conjunction with alternative loss functions such as pixel-wise mean squared difference or absolute difference?

7. Number of filters (NF): would reducing the number of filters from the default size of 16 to 8 impact performance severely?

2.6 Evaluation metric and model selection

Following the Challenge's evaluation protocol, we employ Chamfer distance (CD) measure, which is defined as:

$$CD(\mathcal{S},\mathcal{T}) = \frac{1}{|\mathcal{S}|}\sum_{x\in S} min_{y\in T}||x-y||_2^2 + \frac{1}{|\mathcal{T}|}\sum_{y\in\mathcal{T}} min_{x\in S}||x-y||_2^2 \quad (1)$$

where $\mathcal{S}$ and $\mathcal{T}$ denote source and target point clouds, respectively.

To enable computation without needing a graphics card, we approximated the distance using v =10,000 points randomly drawn from each point cloud as we find the measured Chamfer distance to be relatively stable for this choice of v.

2.7 Mesh surface extraction

We employ Python packages `Veko` and `Pyvista` to respectively extract isosurfaces and visualize the extracted isosurfaces for volume rendering of the mesh surfaces (code listing 3). Figure 1-5 illustrate reference meshes rendered for volumes 1-5, while those from other volume (with no reference masks provided) are shown in the Appendix.

2.8 Implementation and deployment details

As mentioned earlier, opensource code for the U-Net variants [20] were adapted so that different activation functions could be tested.

All experiments were conducted in a virtual environment with Python 3.10, Tensorflow 2.12 and Torch 1.13.0. Graphical processing unit (GPU) cards explored include NVIDIA Tesla V100, Tesla T4, and P1000-SXM2 (CUDA Version 12.0).

To maximize reproducibility [21], source scripts will be updated on the repository with human- (and machine-) readable instructions for emulating the computing environment needed to run these scripts in the form of a docker container at `https://github.com/lisatwyw/smrvis/`.

3 Results & Discussions

When each ultrasound volume of 1281 slices was down-sampled to an axial resolution of 256 $\times$ 256, inference with a single GPU with V100 and single-core CPU for instance took less than one minute and 431.2 $\pm$0.9 seconds (about 7 minutes), respectively.

Figure 5-Figure 8 each illustrates the point cloud extracted from the training volumes by the proposed framework and the point clouds provided in the reference mesh files. Figure 3 visualizes results in two dimensions, while Figure 9 presents the isosurface computed from a randomly selected test volume.

We next present quantitative results with Table 2-6 that generally report the Chamfer distances between the reconstructed and reference point clouds.

More specifically, Table 2 presents trials that failed to converge while tables 3-6 present trials that yielded satisfactory Chamfer distances (below 95.0). In generating these tables, samples from select volumes were used as the training and validation set while a left out volume was used as a test (unseen) sample as described in Section 2.4; these are marked in the tables as '(val)' and '(tst)' to denote validation and test set, respectively.

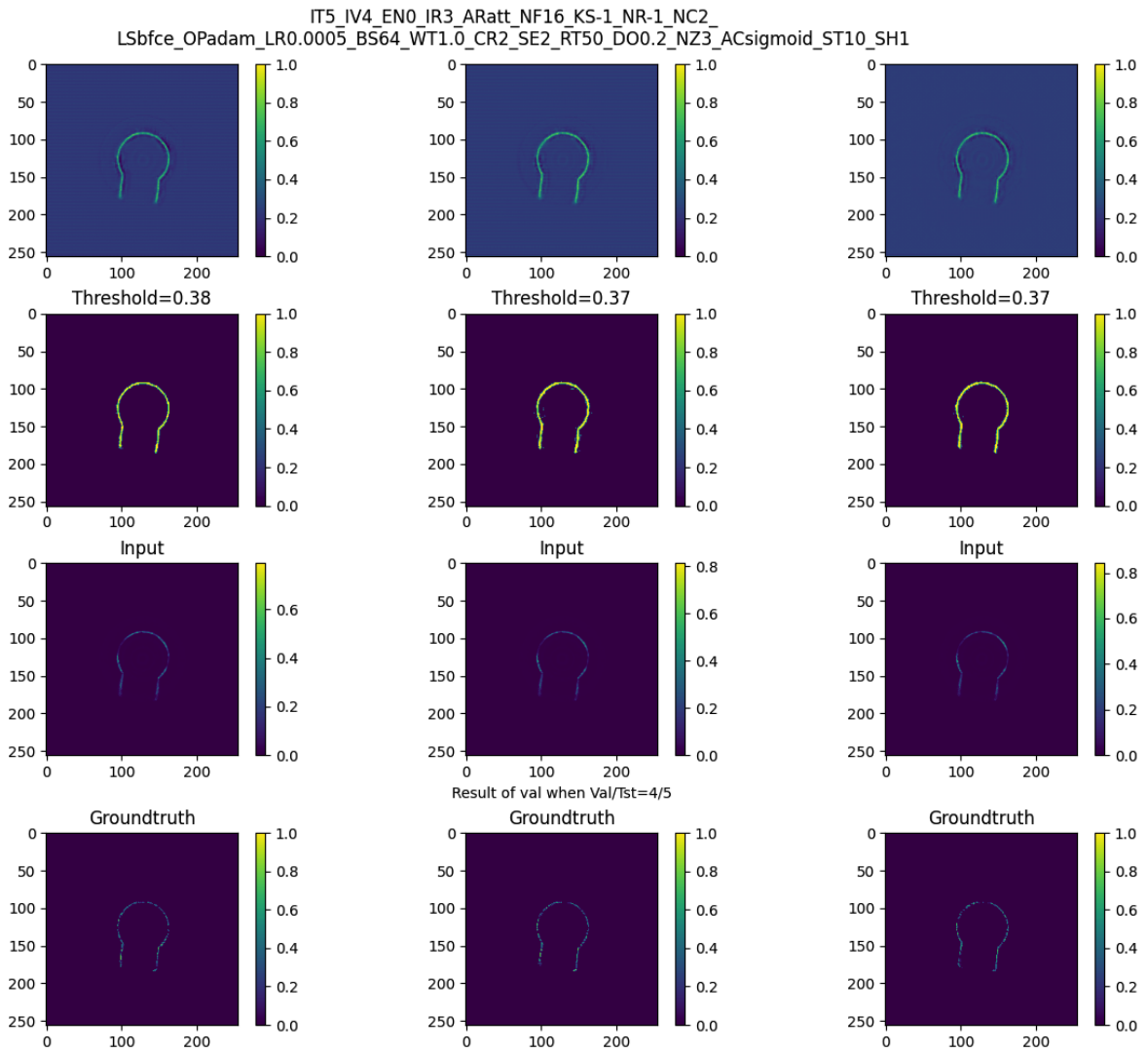


Figure 3: Example 3-slice input to each model. Top to bottom: probabilistic output, thresholded output, input ultrasound slice, and its corresponding reference mesh label.

Based on the results from the trials summarized by these tables, volumes 4 and 5 had the lowest and highest Chamfer distance, respectively. This may be explained by Figure 2, which shows that volume 5 captured a pipe with an object inside, rendering an obstacle to achieving low Chamfer distances between the extracted and reference point clouds.

Results from Table 4 suggest that the choice on the encoding schemes of the reference mesh labels did not impact performance in significant ways. Empirical results (Appendix) suggest that taking the average of contiguous slices maybe superior to taking the solution of the most confident slice.

Table 1 lists the hyper-parameters of the configurations tested for the U-Net variants and example statistics on training convergence.

More results are presented in the Appendix; in short, we did not find significant difference between the following:

1. 384×384 (IR=2) vs 256×256 (IR=3);
2. Sigmoid, hard sigmoid, and hyperbolic tangent will lead to slightly different effective sizes [19], which will impact the choice of threshold values;
3. We did not find improvement of accuracy due to the inclusion of Dice score nor did we find BFCE to be superior over BCE;
4. Use of 8 filters and 16 filters gave comparable results

Model (AR)	# of filters (NF)	# of layers	Model size	Training set	Activation (AC)	# of epochs	t
Att-Unet	16	190	1,989,767	2,3,4 [1/5]	sigmoid	54	0.31
	8		1,989,767	All	sigmoid	51	0.32
R2-Unet	16	158	1,999,571	3,4,5 [2/1]	sigmoid	35	0.32
	8, NR=NC=3	356	751,035	All	sigmoid	49	0.42
U-Net++	8	242	514,219	2,4,5 [3/2]	sigmoid	167	0.41
	16	242	2,050,387	1,2,3 [5/4]	sigmoid	121	0.39
SE-Unet	16	161	2,054,355	1,2,5 [4/3]	tanh	30	0.40
	8	161	515,179	All	tanh	36	0.40
W-Net	16	168	1,159,091	1,2,3 [5/4]	hard sigmoid	100	0.34

Table 1: Number of trainable parameters of the explored model configurations is shown under column 'Model size'. The last column records the threshold t applied to the probabilistic output in order to generate isosurfaces of the predicted mesh surface.

Model	Settings	Volume 1	Volume 2	Volume 3	Volume 4	Volume 5
-	-	167.6	167.6	167.6	167.6	167.6
R2Unet	BCE	105.8	103.8	102.7	105.8 (val)	118.6 (tst)
SE-Unet	BFCE	107.9	108.1	102.3	104.8 (val)	114.9 (tst)
R2Unet	Hard sigmoid/ 2-1 (t=0.32)	147.4 (tst)	137.6 (val)	144.4	134.1	149.8

Table 2: Baseline for subsequent evaluations based on Chamfer distance. For reference, treating every voxel position as part of the reconstructed mesh yields the worse possible distance as 167.6.

Conversely, the following affected the success of training:

1. Chance of training success was increased when the encoded reference meshes were dilated;
2. Chance of training success increased when the second scheme was adopted for sample selection (omit training samples that did not contain the mesh and thus less relevant);
3. Simultaneous application of different random transformations may render training more difficult; we found that a progressive scheme of introducing multiple transformations only in a later phase of training to be an effective solution.

In summary, the models did not appear to have over-fitted to the training set as including the hardest sample (volume 5) did not lead to reduction in error. There appears to be no significant difference due to the choice of model, image resolution, and random transformation schemes (more results presented in the Appendix). We observed that training diverged when reoriented volumes were presented too early during model training.

4 Conclusion

In this brief note, we explored the feasibility of surface mesh reconstruction via point cloud estimation as an image to mask generation problem. This initial framework opens door to possibility of leveraging (pretrained) deep and wide networks published in the wild. The source code developed during the course of this experimental prototyping period will be posted at `https://github.com/lisatwyw/smrvis`. We hope the research communities will find this quick prototype consisting of a few Python scripts approachable.

Acknowledgements

We sincerely thank DarkVision Technologies Inc. for provision of the ultrasound dataset and hosting this exciting challenge. The author also expresses deep gratitude to Tong Tsui Shan and Kim Chuen Tang for their support.

References

[1] Rao J, Wang J, Kollmannsberger S, Shi J, Fu H, Rank E. Point cloud-based elastic reverse time migration for ultrasonic imaging of components with vertical surfaces. Mechanical Systems and Signal Processing. 2022 Jan 15;163:108144.

[2] Verykokou S, Ioannidis C. An Overview on Image-Based and Scanner-Based 3D Modeling Technologies. Sensors. 2023 Jan;23(2):596.

[3] Eisenmann M, Reinke A, Weru V, Tizabi MD, Isensee F, Adler TJ, Ali S, Andrearczyk V, Aubreville M, Baid U, Bakas S. Why is the winner the best?. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023 (pp. 19955-19966).

[4] Virkkunen I, Koskinen T, Jessen-Juhler O, Rinta-Aho J. Augmented ultrasonic data for machine learning. Journal of Nondestructive Evaluation. 2021 Mar;40:1-1.

[5] Osipov, L.V., Kulberg, N.S., Leonov, D.V. et al. 3D Ultrasound: Visualization of Volumetric Data. Biomed Eng 54, 149–154 (2020). https://doi.org/10.1007/s10527-020-09993-3

[6] DarkVision Technologies Inc. Nov 2022. The Ultrasound Dataset Challenge. Retrieved Mar 2023 from https://www.cvpr2023-dl-ultrasound.com/.

[7] Tang L, Hamarneh G. SMRFI: Shape matching via registration of vector-valued feature images. In 2008 IEEE Conference on Computer Vision and Pattern Recognition 2008 Jun 23 (pp. 1-8). IEEE.

[8] Lisa YW Tang, Harvey O Coxson, Stephen Lam, Jonathon Leipsic, Roger C Tam, and Don D Sin, "Towards large-scale case-finding: training and validation of residual networks for detection of chronic obstructive pulmonary disease using low-dose CT," The Lancet Digital Health, vol. 2, no. 5, pp. e259–e267, 2020.

[9] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. InMedical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 2015 (pp. 234-241). Springer International Publishing.

[10] Galdran A, Anjos A, Dolz J, Chakor H, Lombaert H, Ayed IB. The little w-Net that could: state-of-the-art retinal vessel segmentation with minimalistic models. arXiv preprint arXiv:2009.01907. 2020 Sep 3.

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017

[12] Jiang H, Yang G, Huang K, Zhang R. W-Net: one-shot arbitrary-style Chinese character generation with deep neural networks. InNeural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part V 25 2018 (pp. 483-493). Springer International Publishing.

[13] Alom MZ, Hasan M, Yakopcic C, Taha TM, Asari VK. Recurrent residual convolutional neural network based on u-Net (r2u-Net) for medical image segmentation. arXiv preprint arXiv:1802.06955. 2018 Feb 20.

[14] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[15] Sonal Gore, Ashwin Mohan, Prajakta Bhosale, Prajakta Joshi, Ashley George. Brain Tumor Segmentation Using Deep Neural Networks. 2021 Retrieved May 2023 https://github.com/Brain-Tumor-Segmentation.

[16] Lee CY, Xie S, Gallagher P, Zhang Z, Tu Z. Deeply-supervised nets. InArtificial intelligence and statistics 2015 Feb 21 (pp. 562-570). PMLR.

[17] Nieradzik L, Scheuermann G, Saur D, Gillmann C. Effect of the output activation function on the probabilities and errors in medical image segmentation. arXiv preprint arXiv:2109.00903. 2021 Sep 2.

[18] Dubey SR, Singh SK, Chaudhuri BB. Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing. 2022 Jul 3.

[19] Gut D, Tabor Z, Szymkowski M, Rozynek M, Kucybała I, Wojciechowski W. Benchmarking of deep architectures for segmentation of medical images. IEEE Transactions on Medical Imaging. 2022 Jun 6;41(11):3231-41.

[20] Kugelman J, Allman J, Read SA, Vincent SJ, Tong J, Kalloniatis M, Chen FK, Collins MJ, Alonso-Caneiro D. A comparison of deep learning U-Net architectures for posterior OCT retinal layer segmentation. Scientific Reports. 2022 Sep 1;12(1):14888.

[21] Nüst D, Sochat V, Marwick B, Eglen SJ, Head T, Hirst T, Evans BD. Ten simple rules for writing Dockerfiles for reproducible data science. PLoS computational biology. 2020 Nov 10;16(11):e1008316.

[22] Iandola, Forrest N., Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size." arXiv preprint arXiv:1602.07360 (2016).

[23] Ma J, Chen J, Ng M, Huang R, Li Y, Li C, Yang X, Martel AL. Loss odyssey in medical image segmentation. Medical Image Analysis. 2021 Jul 1;71:102035.

[24] Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., Maier-Hein, K. H. (2021). nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods, 18(2), 203-211.

[25] Heinrich MP. Make nnUNets Small Again. InMedical Imaging with Deep Learning, short paper track 2023 Apr 28.

[26] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: Chasing higher flops for faster neural networks. arXiv preprint arXiv:2303.03667, 2023

[27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11976–11986, 2022

[28] Isensee F, Jäger PF, Full PM, Vollmuth P, Maier-Hein KH. nnU-Net for brain tumor segmentation. In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part II 6 2021 (pp. 118-132). Springer International Publishing.

[29] Nichol A, Jun H, Dhariwal P, Mishkin P, Chen M. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. arXiv preprint arXiv:2212.08751. 2022 Dec 16.

A More example visualizations

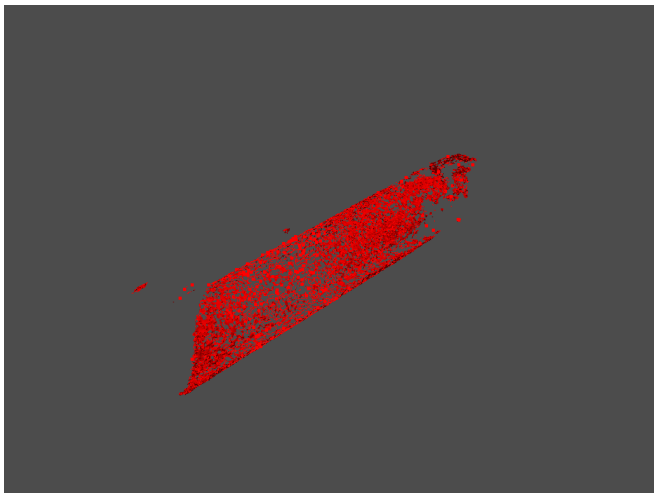


Figure 4: Example of a degenerate case.

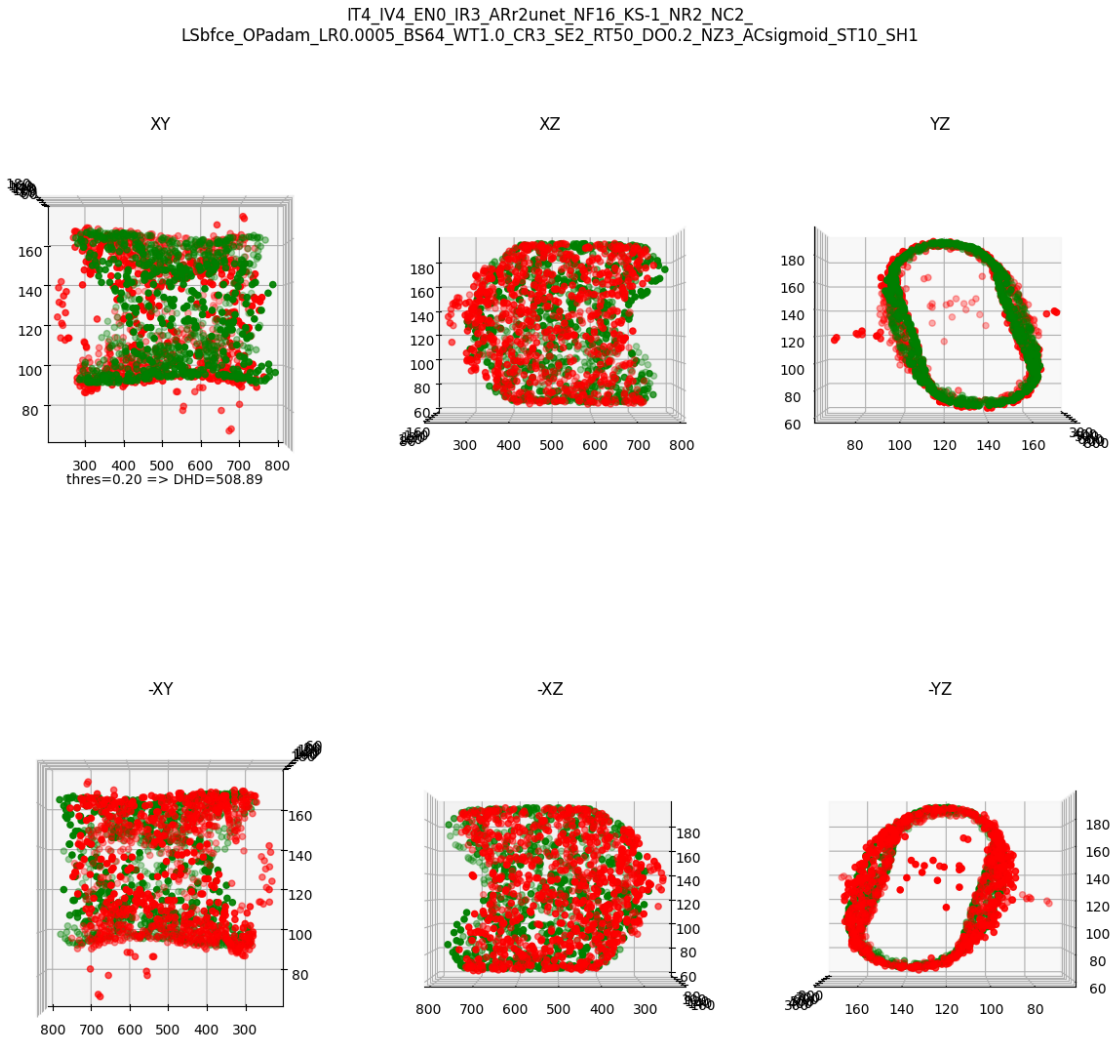


Figure 5: Results for volume 1. Point cloud of the reference mesh label (in green) and the point cloud extracted by the proposed framework using a R2-Unet (in red).

B Meta file

```
ObjectType = Image
NDims = 3
BinaryData = True
BinaryDataByteOrderMSB = False
CompressedData = False
TransformMatrix = 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0
Offset = 0.0 0.0 0.0
CenterOfRotation = 0.0 0.0 0.0
AnatomicalOrientation = RAI
ElementSpacing = 0.49479 0.49479 0.3125
DimSize = 768 768 1280
ElementType = MET_USHORT
ElementDataFile = scan_001.raw
```

When the ultrasound data is read using a meta header file as demonstrated in Code listing B, users may need to update the values of the Anatomical Orientation tag.

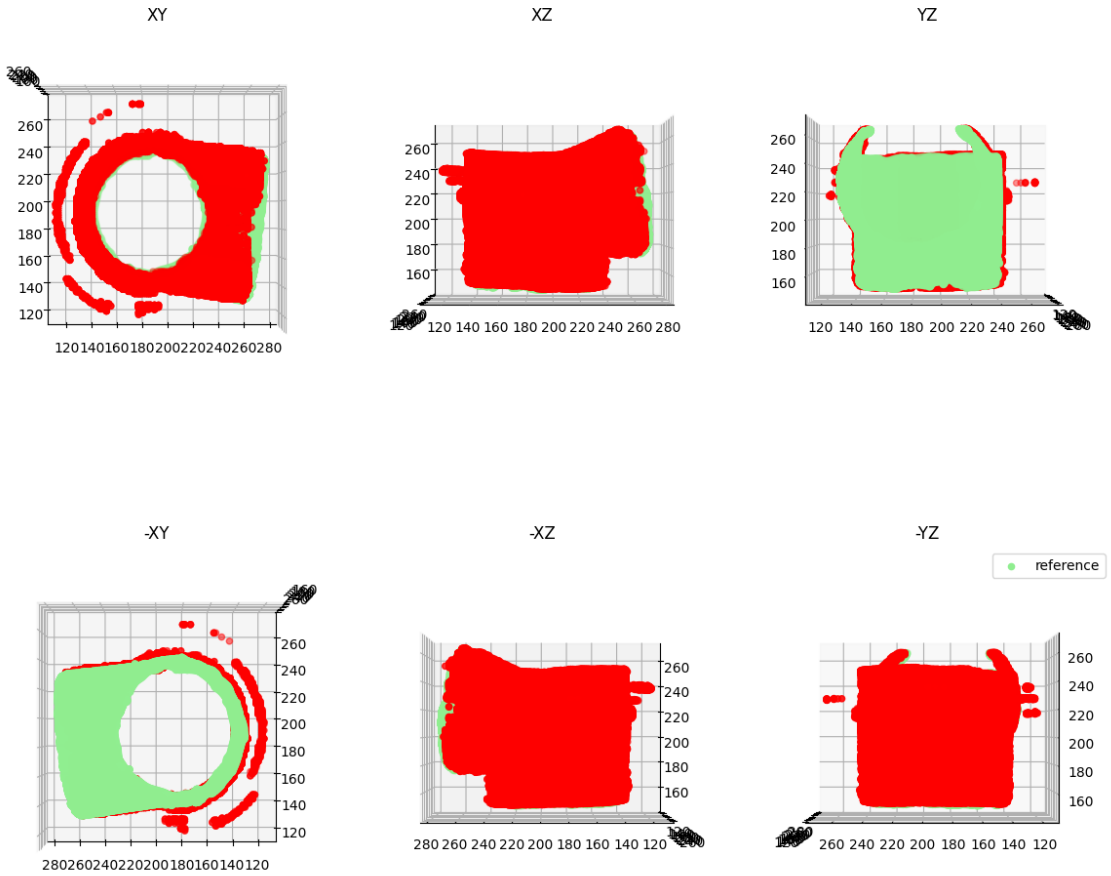


Figure 6: Results for volume 3.

C Training progress

Figure 17 and 19 show examples of training progress drawn from two randomly selected trials that involved a R2Unet and SE-Unet model.

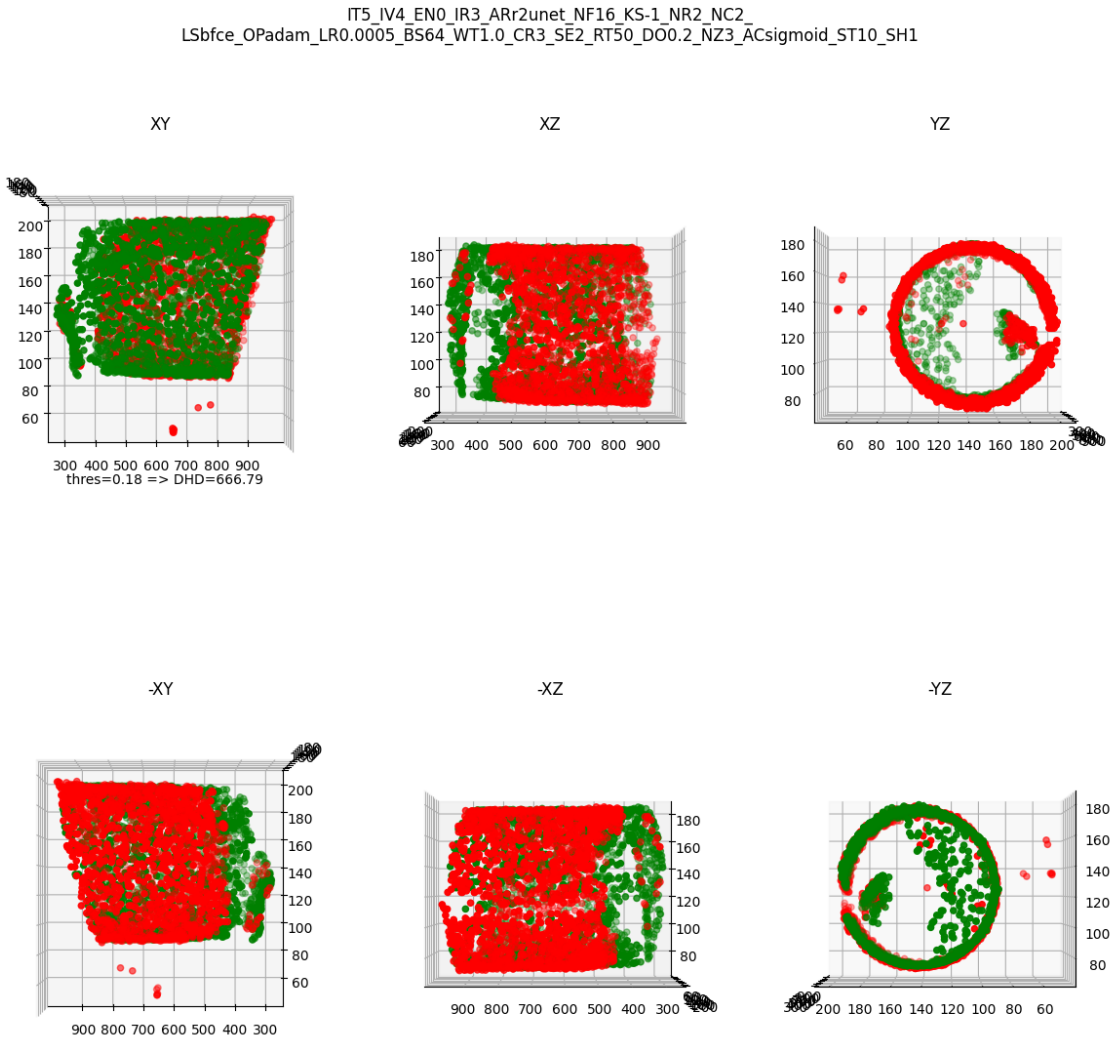


Figure 7: Results for volume 4 using an SE-Unet.

Model	Settings	t	Volume 1	Volume 2	Volume 3	Volume 4	Volume 5
Att-Unet	Trained on 1,2,3	0.19	84.0	75.1	75.9	68.4 (val)	94.0 (tst)
		0.28	82.1	73.4	72.5	66.3 (val)	93.2 (tst)
		0.37	83.2	73.7	70.5	66.1 (val)	93.2 (tst)
R2-Unet	Trained on 3,4,5	0.15	81.5 (tst)	73.3 (val)	73.3	65.1	91.8
		0.22	77.9 (tst)	73.1 (val)	72.3	65.3	91.8
	Trained on 1,4,5	0.2	83.1	74.6 (tst)	75.0 (val)	66.4	93.4
		0.3	80.4	73.7 (tst)	73.5 (val)	65.8	93.3
		0.4	73.8	73.9 (tst)	72.1 (val)	65.9	93.4
	Trained on 1,2,3	0.14	82.0	74.8	73.0	65.5 (val)	92.1 (tst)
		0.21	81.8	71.4	72.7	65.4 (val)	92.2 (tst)
		0.25	82.1	71.2	72.1	65.6 (val)	92.2 (tst)
		0.35	NIL	71.7	71.3	65.3 (val)	91.1 (tst)
SE-Unet	Trained on 3,4,5	0.2	82.0 (val)	72.7	73.0 (tst)	65.9	92.1
		0.3	80.3 (val)	72.4	72.7 (tst)	65.3	92.3
		0.4	78.3 (val)	73.3	71.1 (tst)	65.6	92.8
	Trained on 1,2,3	0.20	81.8	72.7	71.9	64.7 (val)	92.4 (tst)
		0.30	80.1	72.2	71.2	65.0 (val)	93.0 (tst)
		0.32	79.6	72.3	70.9	65.1 (val)	93.2 (tst)
Wnet	Trained on 1,2,5	0.338	72.8	72.6	72.3 (val)	65.7 (tst)	92.7
		0.375	71.7	72.8	72.1 (val)	65.9 (tst)	93.0
		0.413	69.9	73.1	71.9 (val)	66.1 (tst)	93.4

Table 3: Effects of training set on validation and test volumes. Performance evaluation of different networks based on Chamfer distance. For each model, the best test score selected by the lowest validation error (i.e. lowest Chamfer distance) is highlighted with blue cells.

Model	Settings	threshold	Volume 1	Volume 2	Volume 3	Volume 4	Volume 5
Att-Unet	Saturated mask	0.2	82.1	73.0	72.7 (val)	66.6 (tst)	92.5
		0.3	81.3	72.4	72.3 (val)	66.1 (tst)	92.4
		0.4	78.9	72.5	70.8 (val)	65.7 (tst)	92.7
	Solid mask	0.22	81.1	72.7	72.4 (val)	65.7 (tst)	92.3
		0.33	79.2	72.8	71.6 (val)	65.5 (tst)	92.6
		0.37	78.8	73.1	71.3 (val)	65.5 (tst)	92.6

Table 4: Comparison of the encoding schemes of the reference mesh labels.

Model	Settings	t	Volume 1	Volume 2	Volume 3	Volume 4	Volume 5
Att-Unet	Hard sigmoid	0.20	116.4	107.0	106.2	95.7	114.0
		0.30	81.4	73.8	74.3 (val)	66.4 (tst)	93.0
		0.40	78.5	73.7	73.2 (val)	65.9 (tst)	93.1
R2-Unet	Linear	0.28	76.2	73.0	72.6	65.1	93.8
SE-Unet	Hard sigmoid	0.18	81.9	73.3	74.6 (val)	66.8 (tst)	92.4
		0.27	80.3	73.0	72.9	65.8	92.7
		0.37	81.2	74.3	71.4 (val)	66.1 (tst)	93.4
	tanh	0.26	79.7	73.8	71.5	66.0	93.5
		0.29	80.4	74.0	70.9 (val)	66.0 (tst)	94.4
Wnet	Hard sigmoid (BFCE)	0.34	81.6	72.7	69.6	64.3	92.4
		0.37	80.1	72.8	69.1 (val)	64.1 (tst)	92.1
	Hard sigmoid with BCE	0.34	81.6	72.7	69.6	64.3	92.4
		0.37	80.1	72.8	69.1 (val)	64.1 (tst)	92.1

Table 5: Comparisons on the loss and activation functions.

Model	Settings	t	Volume 1	Volume 2	Volume 3	Volume 4	Volume 5
Att-Unet	Max	0.20	81.0	74.4	73.7	66.6	93.0
		0.30	79.9	74.0	73.3	66.3	93.0
		0.40	79.4	74.0	73.1 (val)	66.3 (tst)	93.2
	Mean	0.40	78.9	72.5	70.8 (val)	65.7 (tst)	92.7
SE-Unet	Max	0.33	79.9	73.4	72.4 (val)	65.8 (tst)	93.1
		0.36	80.4	73.9	71.7 (val)	65.9 (tst)	93.4
	Single slice only	0.33	80.6	73.9	72.0 (val)	65.8 (tst)	93.0
		0.36	81.2	74.4	71.3 (val)	66.3 (tst)	94.2
	Mean	0.37	81.2	74.3	71.4 (val)	66.1 (tst)	93.4
SE-Unet with tanh	Mean	0.23	77.4	73.8	71.8	65.9	93.6
		0.26	77.4	74.0	71.2 (val)	66.0 (tst)	94.8
	Max	0.23	78.2	73.7	72.2	66.0	93.3
		0.26	78.0	73.9	71.8	66.1	93.5
	Single slice only	0.23	79.5	73.7	72.0	65.8	93.5
		0.26	79.6	73.9	71.5 (val)	66.0 (tst)	93.5
W-Net	Max	0.338	89.0	80.7	84.2 (val)	74.5 (tst)	98.1
		0.375	81.6	76.5	78.1 (val)	69.5 (tst)	94.8
		0.413	77.5	75.2	75.4 (val)	67.7 (tst)	94.6
	Single slice only	0.338	82.9	77.3	79.4	70.2	95.8
		0.413	77.5	76.5	77.5 (val)	68.6 (tst)	95.2
	Mean (from Table 3)	0.413	69.9	73.1	71.9 (val)	66.1 (tst)	93.4

Table 6: Comparisons on the choice of aggregation scheme. Selection based on the validation set suggests that aggregation by mean achieves distance of 66.1 on the test volume, which is a slight improvement compared to alternative choices (67.7- 68.6) for the Wnet. For the SE-Unet, there is no distinctive impact from the aggregation choice (Chamfer distance ranged from 65.8-66.3).

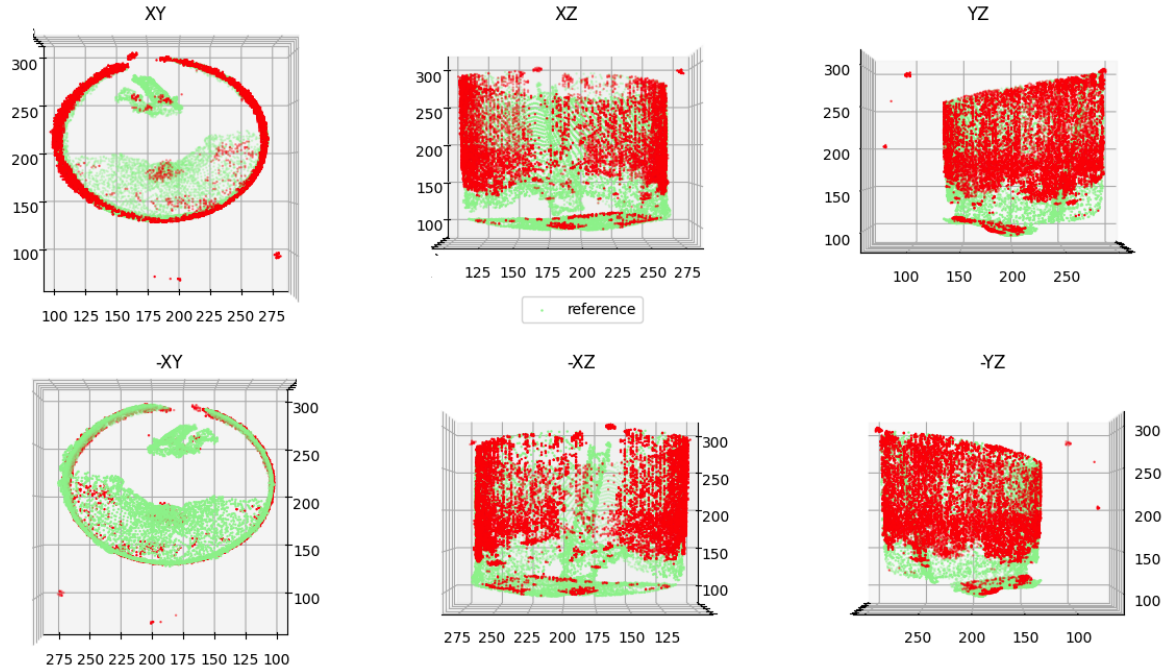


Figure 8: Results from volume 5. Point cloud of the reference mesh label (in green) and the point cloud extracted by the proposed framework using a R2-Unet (in red).

Model						(f to compute t, AG)
Att-Unet	77.5 (tst)	72.9	73.4	65.4	92.8 (val)	(0.9, 0)
	77.1	71.9	73.4	64.5	92.1	(1, 0)
	79.3	73.7	73.3	66.2	93.0	(0.9, 1)
	78.9	73.3	73.3	65.9	92.7	(1, 1)
	76.9	73.2	73.4	65.9	92.9	(0.9, 2)
	75.7	72.7	73.4	65.7	93.0	(1, 2)
R2-Unet	81.7	70.9	72.4	65.6 (val/tst)	92.3	(0.9, 0)
	82.8	71.1	72.1	65.6	92.1	(1, 0)
	81.7	71.7	72.5	65.4	92.5	(0.9, 1)
	81.8	72.1	72.2	65.6	92.6	(1, 1)
	82.3	70.5	72.9	65.4	92.1	(0.9, 2)
	83.1	70.8	72.7	65.4	91.9	(1, 2)
SE-Unet	77.4	73.8	71.8 (val)	65.9 (tst)	93.6	(0.9, 0)
	77.4	74.0	71.2 (val)	66.0 (tst)	94.8	(1, 0)
	78.2	73.7	72.2	66.0	93.3	(0.9, 1)
	78.0	73.9	71.8	66.1	93.5	(1, 1)
	79.5	73.7	72.0	65.8	93.5	(0.9, 2)
	79.6	73.9	71.5	66.0	93.5	(1, 2)
	74.4	73.8	71.2	66.4	94.2	
U-Net++; NF8/SE2/IR3	77.4	72.8	72.6 (tst)	65.1 (val)	92.4	(0.9, 0)
	75.4	73.0	72.4	65.5	92.6	(1, 0)
	78.5	72.9	72.6	65.3	92.5	(0.9, 1)
	76.3	73.1	72.4	65.4	92.6	(1, 1)
	78.3	72.8	72.6 (tst)	64.9 (val)	92.4	(0.9, 2)
	76.1	72.9	72.5	65.4	92.5	(1, 2)
U-Net++; NF16/SE1/IR2/RT=60	82.6	74.0	70.9	66.3	93.4	(0.9, 0)
	85.4	74.4	67.9	66.9	93.4	(1, 0)
	82.4	74.0	71.5	66.1	93.5	(0.9, 1)
	83.8	74.4	69.6	66.7	93.4	(1, 1)
	82.1	74.0	71.7	66.0 (val)	93.3 (tst)	(0.9, 2)
	83.1	73.9	70.6	66.5	93.5	(1, 2)
	82.4	73.4	72.7	65.9	93.2	(0.7, 0)
	82.3	73.7	72.1	65.9	93.3	(0.8, 0)
	82.3	73.4	72.9	65.8	93.2	(0.7, 1)
	82.3	73.7	72.3	65.9	93.5	(0.8, 1)
	82.7	73.4	72.8	65.9	93.2	(0.7, 2)
	82.4	73.7	72.3	65.8	93.3	(0.8, 2)
Att-Unet; SE=1; IR=2; RT=45	80.5	73.9	72.0	65.9 (val)	93.7 (tst)	(0.9, 0)
	81.4	74.4	70.6	66.5	94.9	(1, 0)
	80.5	73.9	72.5	66.0	93.4	(0.9, 1)
	80.6	74.3	71.4	66.3	94.7	(1, 1)
	81.9	74.1	71.8	66.1	93.5	(0.9, 2)
	82.0	74.5	70.9	66.4	93.8	(1, 2)
Wnet	82.3	72.4	70.2	64.4 (val)	92.8 (tst)	(0.9, 0)
	81.9	72.4	69.7	63.9 (val)	92.7 (tst)	(1, 0)
	82.6	72.6	70.3	64.7	92.8	(0.9, 1)
	82.4	72.6	69.9	64.5	92.7	(1, 1)

Table 7: Evaluation by Chamfer distance computed between the reconstructed and reference point clouds extracted from each ultrasound volume.

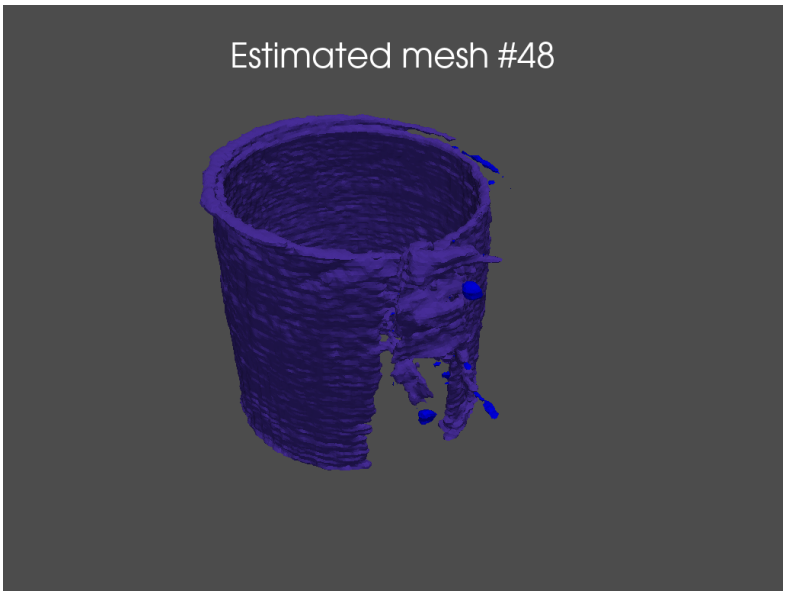


Figure 9: Isosurface rendering of the points extracted for test volume 48 (whose reference label not provided to challenge participants). Two colours (blue and purple) correspond to two thresholds applied on the probabilistic output masks predicted by a trained model.

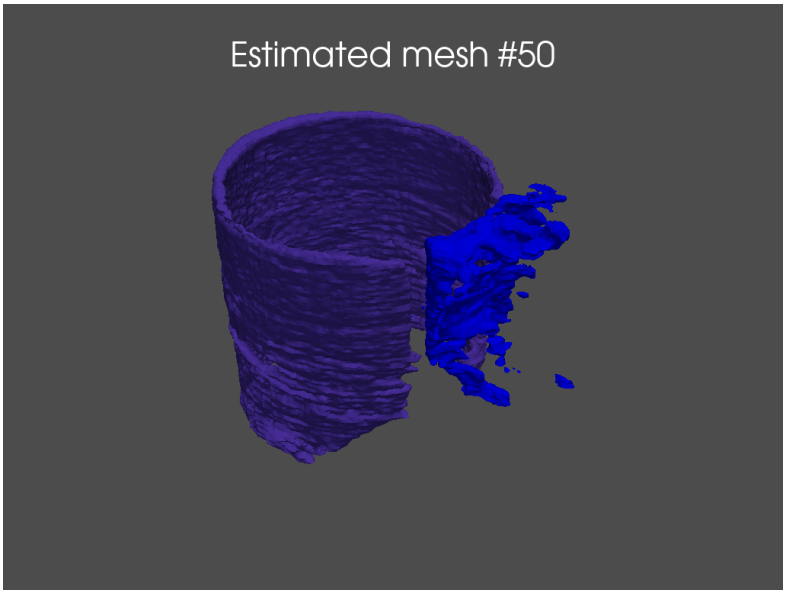


Figure 10: Test volume 49.

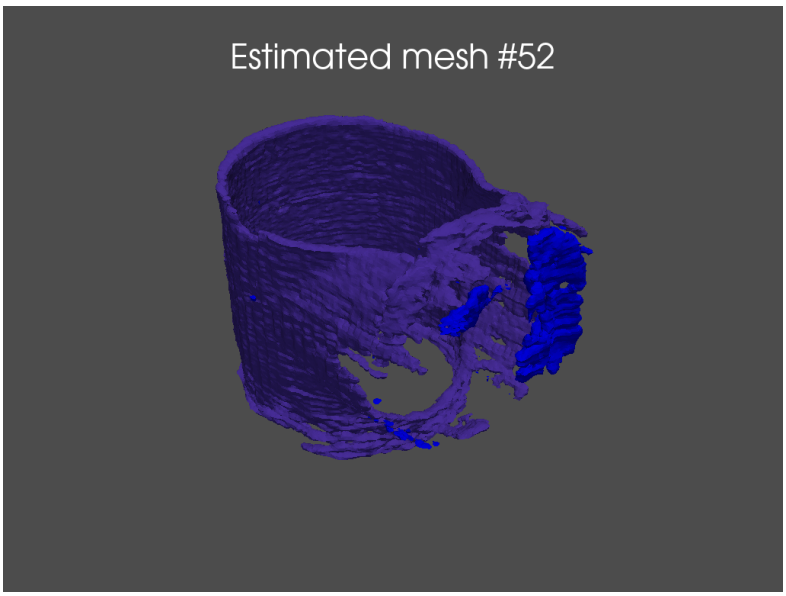


Figure 11: Test volume 51.

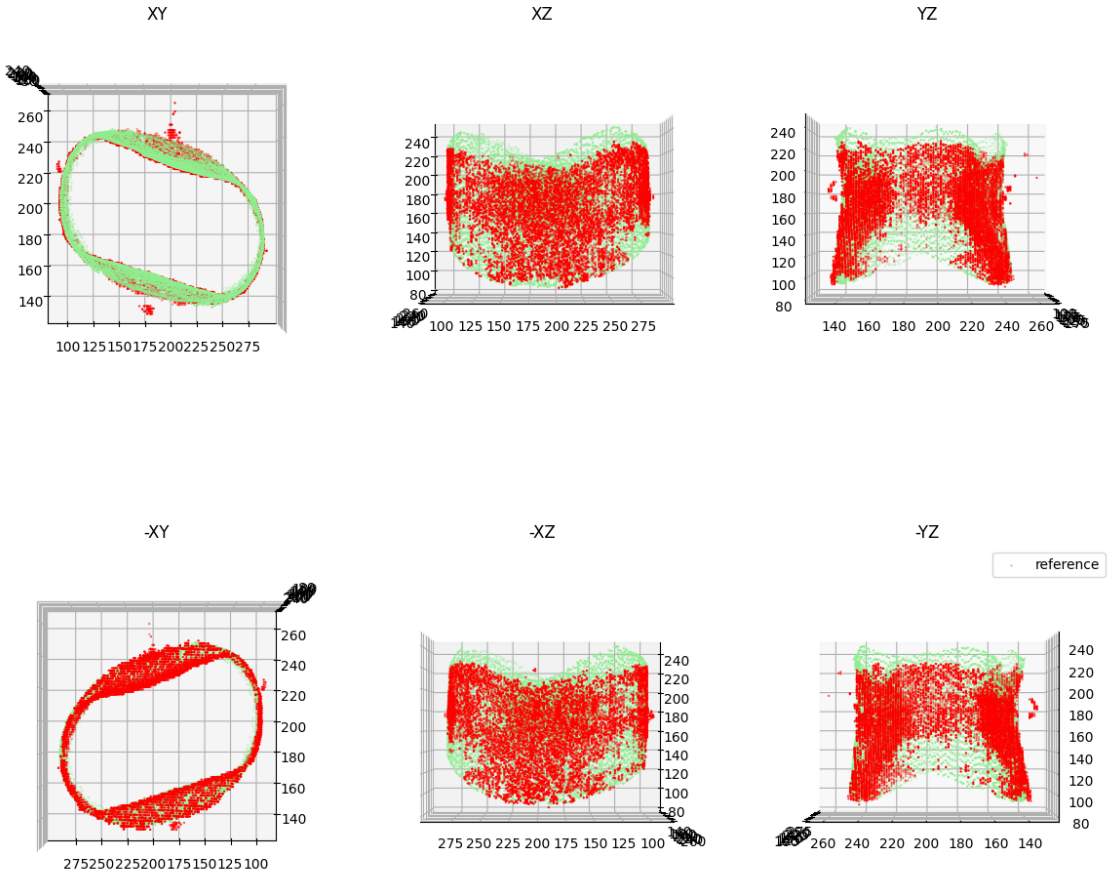


Figure 12: Results for volume 1 by SE-Unet.

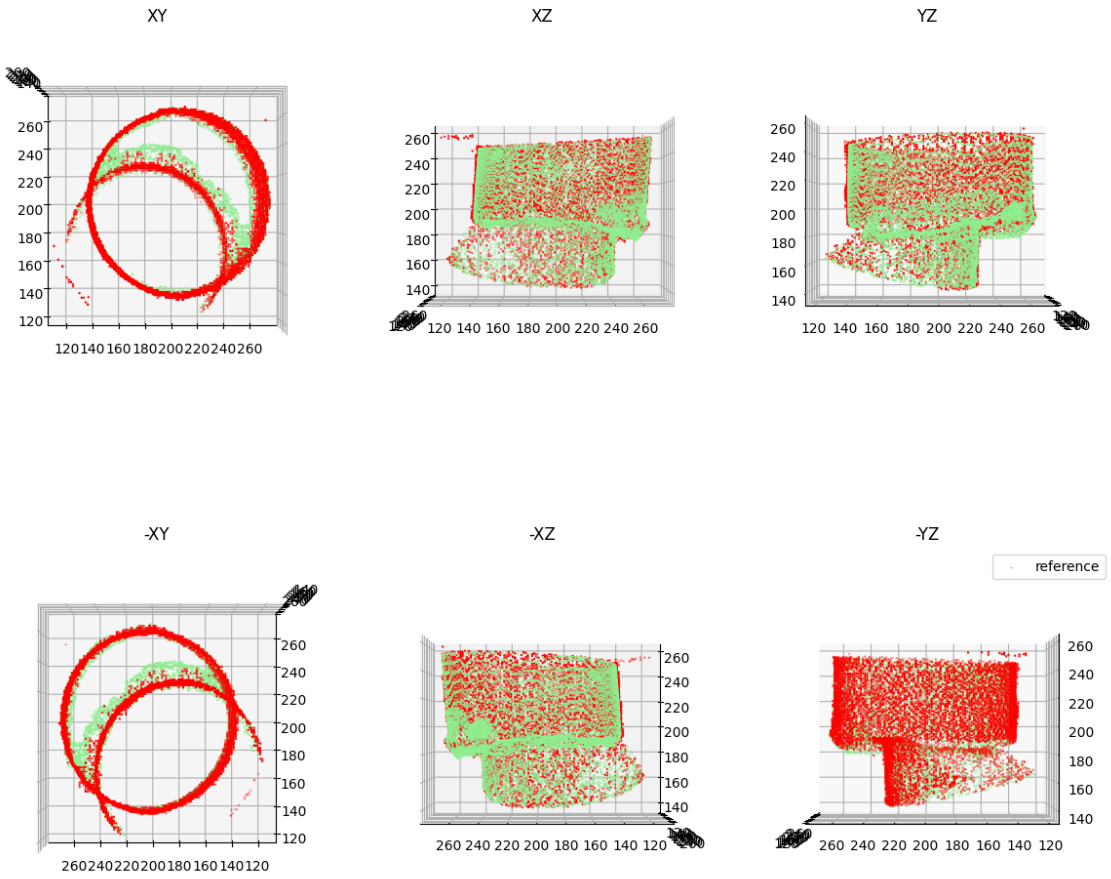


Figure 13: Results for volume 2 using an SE-Unet.

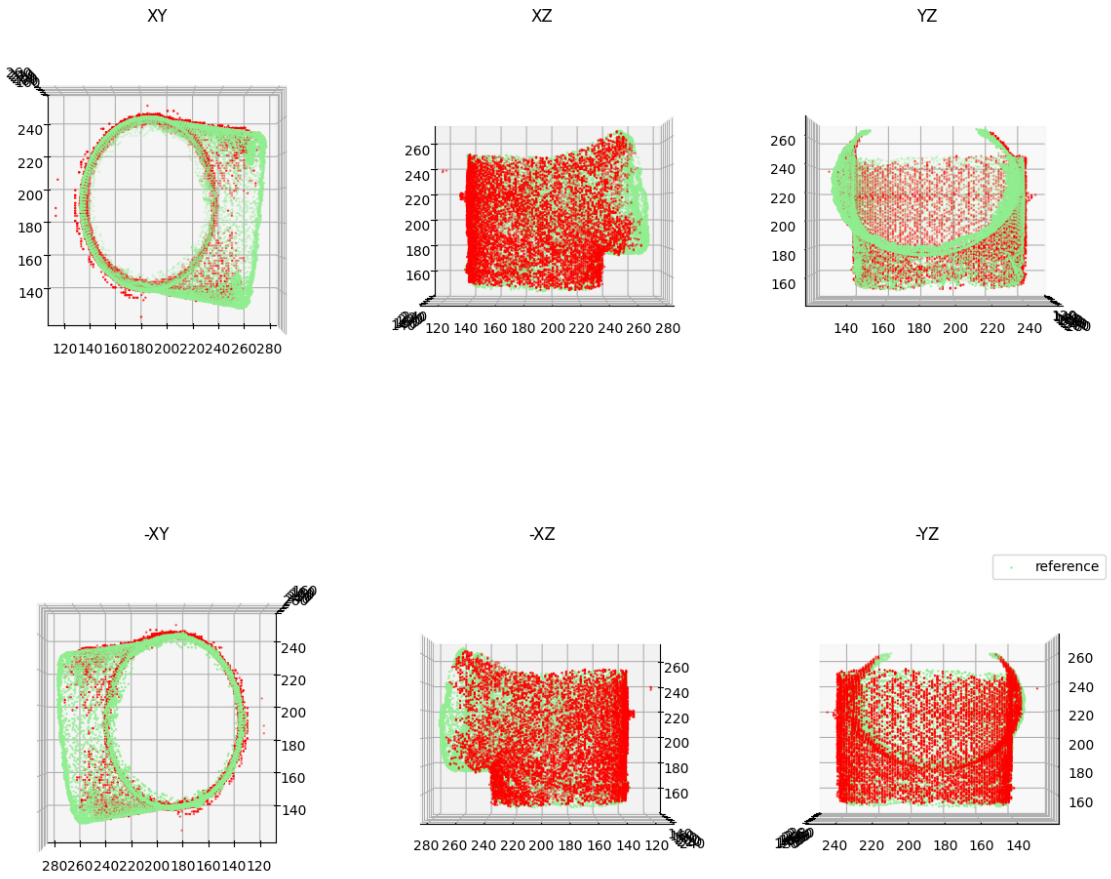


Figure 14: Results for volume 3 using an SE-Unet.

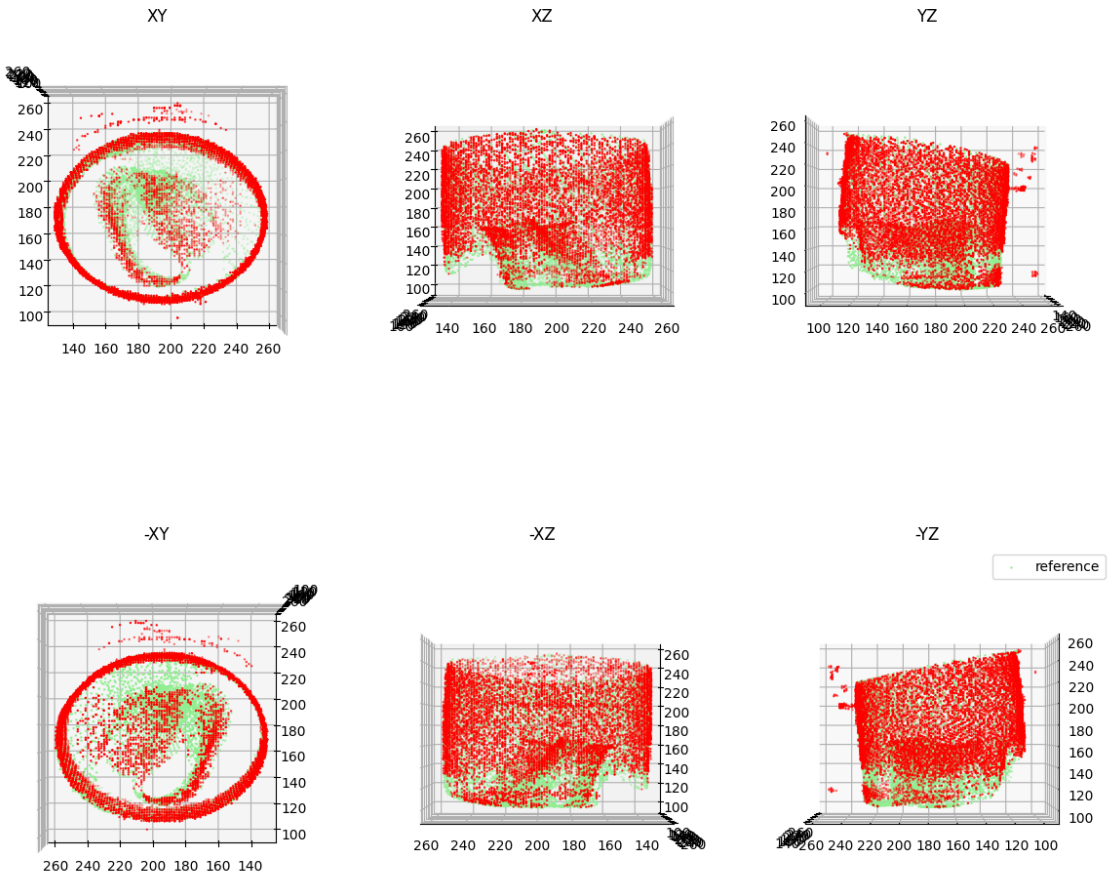


Figure 15: Results for volume 4 using an SE-Unet.

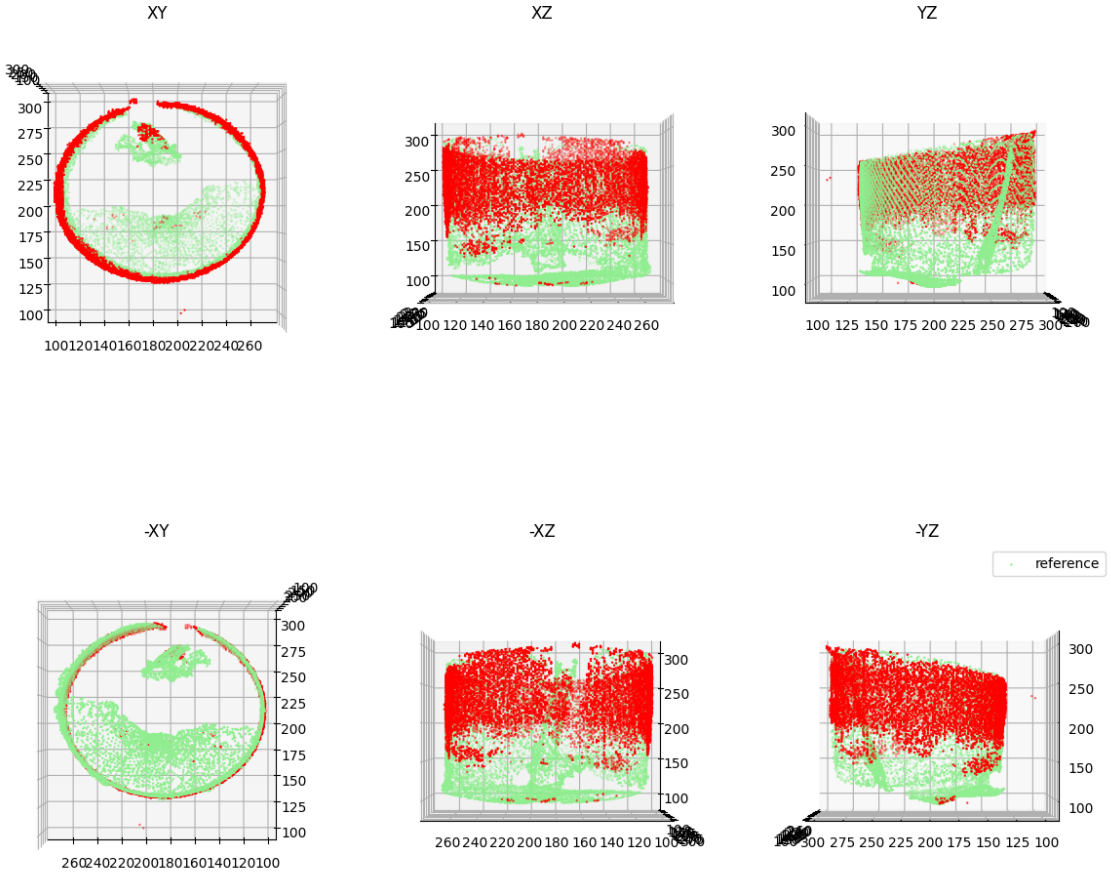


Figure 16: Results for volume 5 using an SE-Unet.

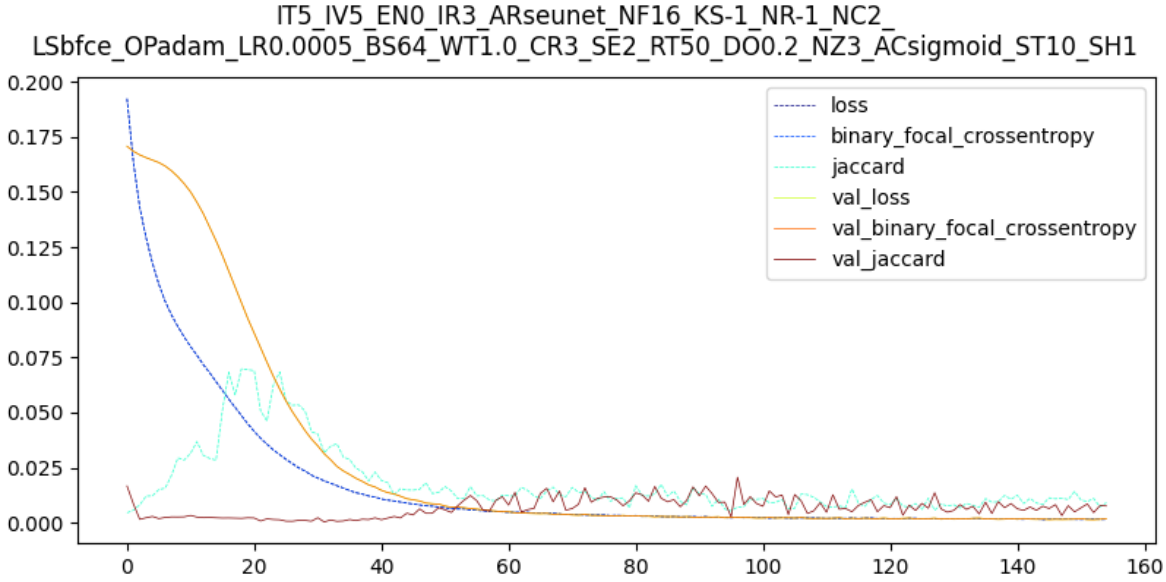


Figure 17: Example progress plot of training a R2-Unet.

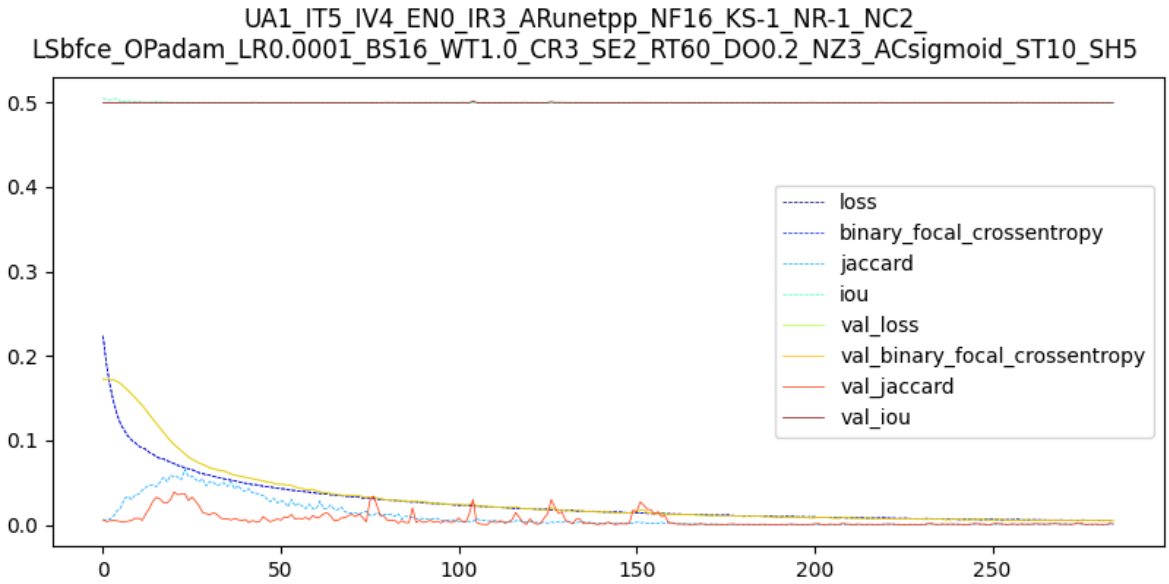


Figure 18: Example progress plot of training an U-Net++.

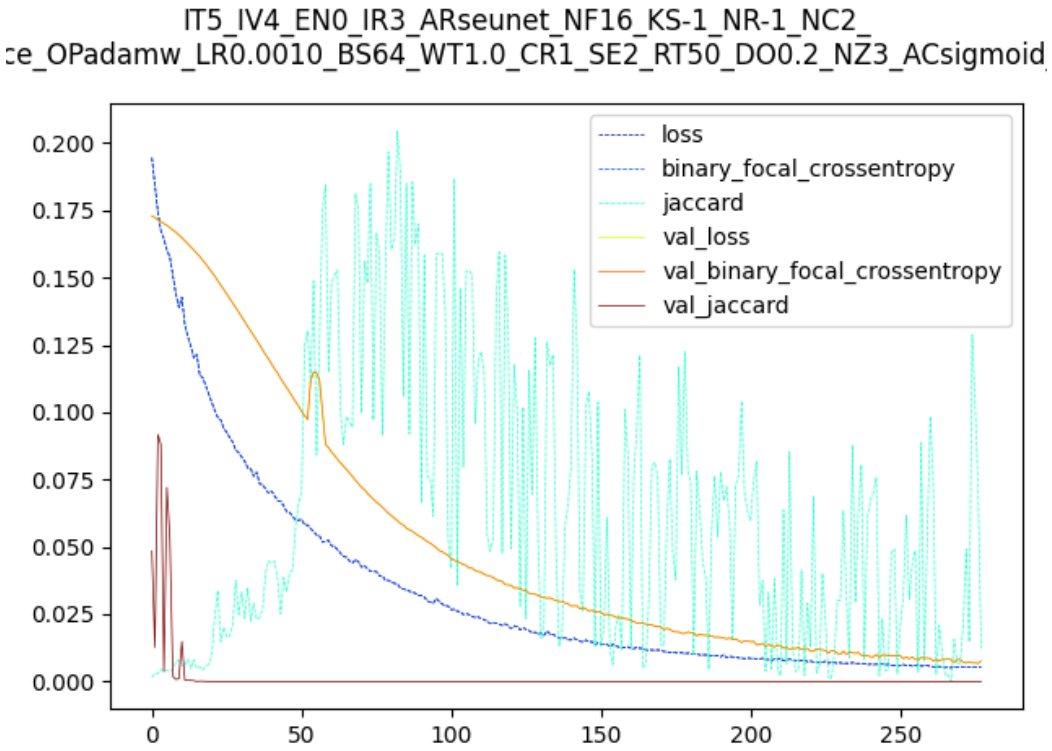


Figure 19: Example progress plot of training an SE-Unet.