



**HAL**  
open science

# Vascular Segmentation of Functional Ultrasound Images using Deep Learning

Hana Sebia, Thomas Guyet, Mickaël Pereira, Marco Valdebenito, Hugues  
Berry, Benjamin Vidal

► **To cite this version:**

Hana Sebia, Thomas Guyet, Mickaël Pereira, Marco Valdebenito, Hugues Berry, et al.. Vascular Segmentation of Functional Ultrasound Images using Deep Learning. 2024. hal-04744271

**HAL Id: hal-04744271**

**<https://inria.hal.science/hal-04744271v1>**

Preprint submitted on 25 Oct 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# Vascular Segmentation of Functional Ultrasound Images using Deep Learning

Hana Sebia<sup>a,b</sup>, Thomas Guyet<sup>a,b</sup>, Mickaël Pereira<sup>b,c</sup>, Marco Valdebenito<sup>d</sup>, Hugues Berry<sup>a,b</sup>, Benjamin Vidal<sup>b,c,d</sup>

<sup>a</sup>*AIstroSight, Inria, Hospices Civils de Lyon, Villeurbanne, France*

<sup>b</sup>*University Claude Bernard Lyon 1, Villeurbanne, France*

<sup>c</sup>*Lyon Neuroscience Research Center Lyon, France*

<sup>d</sup>*CERMEP-Imaging Platform Lyon, France*

---

## Abstract

Segmentation of medical images is a fundamental task with numerous applications. While MRI, CT, and PET modalities have significantly benefited from deep learning segmentation techniques, more recent modalities, like functional ultrasound (fUS), have seen limited progress. fUS is a non-invasive imaging method that measures changes in cerebral blood volume (CBV) with high spatio-temporal resolution. However, distinguishing arterioles from venules in fUS is challenging due to opposing blood flow directions within the same pixel. Ultrasound localization microscopy (ULM) can enhance resolution by tracking microbubble contrast agents but is invasive, and lacks dynamic CBV quantification. In this paper, we introduce the first deep learning-based segmentation tool for fUS images, capable of differentiating signals from different vascular compartments, based on ULM automatic annotation and enabling dynamic CBV quantification. We evaluate various UNet architectures on fUS images of rat brains, achieving competitive segmentation performance, with 90% accuracy, a 71% F1 score, and an IoU of 0.59, using only 100 temporal frames from a fUS stack. These results are comparable to those from tubular structure segmentation in other imaging modalities. Additionally, models trained on resting-state data generalize well to images captured during visual stimulation, highlighting robustness. This work offers a non-invasive, cost-effective alternative to ULM, enhancing fUS data interpretation and improving understanding of vessel function. Our pipeline shows high linear correlation coefficients between signals from predicted and actual compartments in both cortical and deeper regions, showcasing its ability to accurately capture blood flow dynamics.

*Keywords:* functional ultrasound, segmentation, ultrafast ultrasound localization microscopy, preclinical, medical images, neuroscience

*PACS:* 43.35.Yb, 87.63.dk, 87.19.La

*2008 MSC:* 62H35, 68T05, 82C32

---

## 1. Introduction

Blood flow and volume are closely linked to neuronal activity changes in the brain, a phenomenon known as neurovascular coupling. This relationship has enabled functional neuroimaging techniques, such as functional ultrasound imaging, to successfully probe brain activity at the

mesoscopic scale in both animals and humans. Functional ultrasound imaging (fUS) is a neuroimaging technique that leverages hemodynamic signals to map brain activity, achieving high spatial resolution down to 100  $\mu\text{m}$  and temporal resolution as low as 400 ms [1, 2, 3]. This method captures changes in cerebral blood volume related to neuronal activity through ultra-fast Power Doppler imaging. These images are analyzed to understand brain functions and pathologies [4, 5, 6]. Accurately identifying whether signals originate from arterioles or venules in these images is crucial to improve our understanding of brain vascular dynamics, as these distinct vascular compartments are differently involved in spontaneous fluctuations of vessel tone and functional hyperemia [7, 8, 9, 10] as well as energy supply [11] and waste clearance [12]. Unfortunately, while the spatial resolution of fUS is relatively high for in vivo mesoscopic brain imaging compared to fMRI, it is insufficient to resolve individual blood vessels. Consequently, a single fUS pixel may encompass multiple vessel types, each responding differently to neurovascular changes, making it challenging to distinguish between these compartments. Although Color Doppler sequences in ultrasound imaging can determine blood flow direction – which helps in differentiating opposing flows in cortical arterioles and venules – this method still struggles to clearly identify these compartments. This limitation arises because upward and downward flows often coexist within the same 100-micron voxels, canceling each other out and obscuring clear compartmental distinction. To overcome this, ultrasound localization microscopy (ULM) is an advanced imaging modality that also employs ultra-fast sampling techniques to significantly enhance spatial resolution to just a few microns [13, 14]. This advanced approach not only improves spatial resolution but also enables the determination of blood flow direction. Nonetheless, it requires the intravenous injection of micro-bubble contrast agents and the tracking of their position along a protracted time period to create a detailed vascular map. This map differentiates between upward and downward flows based on the movement direction of the micro-bubbles. While ULM provides comprehensive insights into cerebral blood flow dynamics, it lacks the temporal resolution of functional ultrasound. Moreover, its deployment is complex and invasive, necessitating the injection of contrast agents and presenting significant challenges in managing large data volumes required for image reconstruction.

This study leverages deep learning to enhance fUS image analysis by circumventing the conventional use of ULM. Specifically, we aimed to differentiate between upward and downward blood flows, corresponding to arterial and venous structures in the cortical region, directly from fUS images, without relying on ULM or requiring a particular data processing of the intermediate raw data. Our approach focused on automating an image segmentation task by training a deep learning model on a dataset specifically annotated to identify these vascular compartments. The process of constructing annotations, based on available ULM images, is detailed in Section 3.3. Once the model is trained, it can be applied to new fUS images to accurately retrieve vascular structures without requiring ULM sessions or manual annotation. The challenge was to identify a deep learning architecture that could achieve high accuracy. We focused our investigation on UNet architectures, known for their effectiveness on medical images, even when trained on small datasets [15]. To address this, we proposed a benchmarking module that compares multiple segmentation architectures as detailed in Section 3.4.2. Our results, summarised in Section 4.4.1, demonstrated a segmentation quality comparable to that achieved for other imaging modalities (MRI, OCT,...) on vascular and tubular structures. Additionally, we investigated, in Section 4.4.2, how incorporating different temporal frames from fUS data affects the segmentation quality, exploring potential improvements in accuracy. Finally, we showed, in Section 4.4.3, that training a model on fUS images acquired in a resting state generalize to images acquired under visual stimulation with the same precision, without requiring re-training. In Section 4.4.4,

we illustrated how this segmentation approach can be used in a practical way to improve fUS signal interpretation.

## 2. Related work

### 2.1. *Different applications of Deep Learning on functional ultrasound*

Despite the potential benefits, fUS has seen few studies that make use of advanced computational methods, particularly for image segmentation task. The existing research primarily focuses on other aspects, such as improving image reconstruction from sparse data and enhancing sensitivity in neuroimaging applications. For instance, Di Ianni and Airan [16] have developed a deep learning platform known as Deep-fUS, which significantly reduces the amount of ultrasound data required while maintaining image quality. Their work demonstrates the potential for deep learning to significantly streamline fUS imaging using portable devices and resource-limited settings. This approach also highlights the innovative use of CNN in reconstructing power Doppler images from under-sampled sequences. Similarly to fUS, ULM has also seen limited exploration with deep learning. van Sloun et al. [17] have leveraged CNN to achieve super-resolution in ULM, significantly improving the precision of micro-bubble localization, while Milecki et al. [18] have developed a spatio-temporal framework that refines the process of micro-bubble tracking. While the aforementioned works have made substantial advancements in improving the quality of images produced by fUS and ULM, they primarily focus on enhancing image reconstruction and resolution.

### 2.2. *UNets for medical image segmentation*

Medical image segmentation has witnessed major advancements, particularly through the use of deep learning models [19, 20]. The UNet architecture has demonstrated remarkable success, providing precise segmentation across diverse imaging modalities such as Ultrasound [21], PET [22], MRI [23] and CT [24]. Its design, characterized by a symmetric encoder-decoder structure and extensive use of skip connections [25, 26], facilitates significantly the segmentation of various organs and tumors such as skin lesions [27], liver and rectal tumors in CT and MRI scans [28, 29], or breast tumors in ultrasound imaging [30]. Enhancements to UNet have also adapted it for volumetric data from MRI and CT, extending its structure to better analyze spatial relationships in 3D space [31, 32, 33], which is essential for organs and vascular segmentation [34]. Numerous reviews continue to discuss these advancements and their contributions to the field [35, 15, 36].

### 2.3. *Vessel segmentation*

Medical imaging segmentation has also expanded to encompass more complex and subtle structures, such as blood vessels. This task has primarily focused on the segmentation of cerebral and retinal vessels, crucial for diagnosing and monitoring neurological and ocular diseases [37, 38]. This field is broadly categorized into two main areas: 2D and 3D segmentation. Volumetric imaging like Magnetic Resonance Angiography (MRA) allows for constructing detailed 3D vascular tree structures, with recent advances using enhanced UNet architectures [39, 40]. However, given the specific task we aim to address, our research focuses particularly on 2D segmentation. This emphasis on 2D analysis led us to explore domain-specific optimizations that significantly improve segmentation outcomes. Notably, Shit et al. [41] introduced the *centerlineDice* metric, which measures similarity across segmentation masks and

their morphological skeletons, ensuring topological continuity. Furthermore, Zhou et al. [42] has advanced this area by embedding vessel density and fractal dimensions directly into the loss function, enabling refined multi-class vessel segmentation that categorizes pixels into veins, arteries, background or uncertain regions. This novel regularization has also improved biomarker quantification, directly contributing to downstream clinical tasks.

### 3. Method

#### 3.1. *fUS acquisition*

For the acquisition of functional ultrasound imaging, we performed experiments on the brain of 35 anesthetized rats. A few days before imaging, the rats underwent a skull-thinning procedure to enhance the signal-to-noise ratio, as described in [4]. The rats were anesthetized using a combination of subcutaneous medetomidine and isoflurane [43]. Imaging was performed using a preclinical ultrasound imaging system (Iconeus V1, Paris, France). Doppler vascular images were obtained using the Ultrafast Compound Doppler Imaging technique [44]. Each frame corresponded to a Compound Plane Wave frame [45] resulting from the coherent summation of backscattered echoes obtained after successive tilted plane waves emissions. Stacks acquired at a 500 Hz frame rate, were processed with a dedicated spatiotemporal filter based on Singular Value Decomposition (SVD) [46] to isolate the blood volume signal from tissue signals, thus generating Power Doppler images. These final images have a temporal resolution of 0.4 s, a spatial resolution of 100  $\mu\text{m}$  and are directly proportional to the cerebral blood volume.

In the initial phase, we conducted a 10-minute resting-state recording. This generated a 2D + time image stack consisting of 3 000 frames, each with a spatial resolution of  $112 \times 128$  pixels (100 microns/pixel). This first stack details the cerebral activity signals associated with hemodynamic changes at rest. Following the initial data collection, a second acquisition was conducted immediately. During this phase, we implemented visual stimulation by maintaining the anaesthetised rat in the dark and exposing it to periodic light flashes, as described in [5]. Stimulation runs consisted of black and white flickering on a screen placed in front of the animal (flickering at a frequency of 3 Hz and continuous black screen for rest). The stimulation pattern consisted in 30 s of initial rest followed by runs of 30 s of flicker and 45 s of rest, repeated four times for a total duration of 330 s. This session was shorter than the first, yielding an image stack of 825 frames at the same spatial resolution and acquisition frequency. This second stack captures therefore cerebral activity linked to hemodynamic changes during visual stimulation.

#### 3.2. *ULM acquisition*

Immediately following the initial fUS scans, we conducted an ultrasound localization microscopy session using the same equipment (Iconeus V1). For this, we injected contrast-enhancing micro-bubbles into the bloodstream of the rat via the tail vein and performed imaging over 5 minutes. The raw data collected using a dedicated Iconeus software, based on the methodology described by Errico et al. [13], were then used to construct three ultra-high resolution imaging modalities. These images, illustrated in Figure 1, detail the cerebral vascular tree at a fine scale with different types of information: (A) micro-bubble density, (B) velocity, and (C) axial velocity along the Z-axis. The resolution is 2 microns per pixel ( $7\,011 \times 5\,494$  pixels) for micro-bubble density and 10 microns per pixel ( $7\,008 \times 5\,490$  pixels) for velocity. The lower resolution for velocity results from applying a filter to remove outliers, based on the individual tracking of micro-bubbles. The construction of these modalities is performed by the Iconeus software and requires approximately 6 hours of computation per experiment.

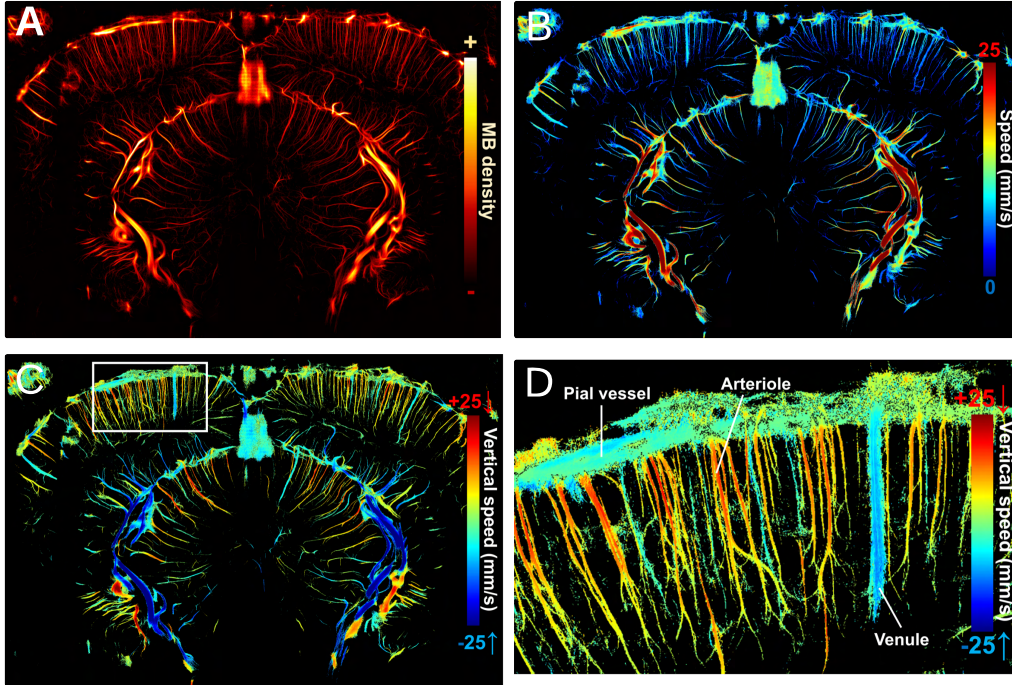


Figure 1: ULM modalities: (A) micro-bubble density, (B) micro-bubble velocity, (C) micro-bubble velocity over the Z-axis, (D) Zoom over arterioles and venules from micro-bubble velocity following the Z-axis.

### 3.3. fUS Automatic Annotation

The above experiments provided us with two stacks of fUS (resting state and visual stimulation), paired with one ULM image for each animal. We used the ULM image for the vascular annotation within the fUS stack, as it provides the necessary information. More specifically, the ULM Z-axis velocity modality is used to automatically distinguish different vascular compartments: in the cortex, positive velocity values indicate the movement of micro-bubbles through arterioles, while negative values reflect their movement through venules. For the sake of simplicity, we use the terms arteries to refer to downward flow compartments and veins to refer to upward flow compartments. This intentional oversimplification applies primarily to the cortical region, where this relationship can be easily made. We acknowledge that it may not apply in other regions. To automatically differentiate these vascular compartments, we applied thresholding to the Z-axis velocity image, as illustrated in Figure 2-A. The images were then resized using bilinear interpolation to match the spatial resolution of the fUS stack. Since our primary focus is on identifying vascular structures rather than quantifying velocity, the resulting images were converted into binary masks. To ensure robust annotations, we set a conservative threshold of 0.05 to exclude ultra-fine structures that fUS is unlikely to detect, thereby producing more accurate and useful vascular masks. These masks served as ground truth annotations for both resting and visual stimulation states, as they represent the anatomical structure of the brain vascular tree and do not contain functional information. It is important to note that resizing ultra-high resolution images introduced a small proportion of mixed pixels belonging to both artery and vein masks. However, since these mixed pixels constituted less than 1%, they were retained to avoid artifacts

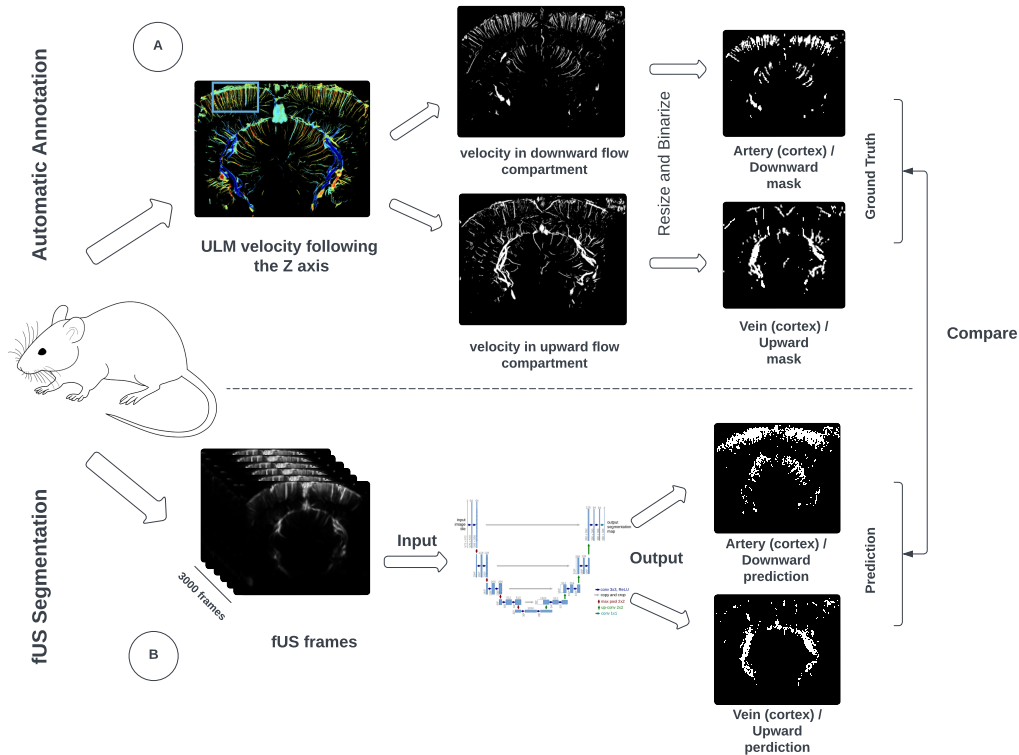


Figure 2: (A) Process of constructing fUS annotations – artery (cortex) /downward and vein (cortex) /upward masks – from ULM images. (B) Segmentation framework of fUS stacks based on UNet architecture.

that might arise from selective pixel exclusion.

### 3.4. Segmentation framework

We approached our problem of automatic compartmentalization of fUS imaging as a multi-class segmentation task where each pixel in the fUS image must be classified as vein, artery, or background. To this end, we trained a deep neural network to generate vein and artery masks for each fUS image stack input. These generated masks are then compared to the ground truth derived from ULM to adjust the weights of the network. This pipeline is detailed in Figure 2-B. We opted for a UNet architecture and some of its variants for the segmentation. Additionally, we evaluated various loss strategies to optimize feature and pixel-level classification. Our aim was to determine the most effective model and the most appropriate loss function for accurate fUS image segmentation. Once trained, the model is capable of generating masks based on new fUS images. Those masks are then projected on the fUS stack, enabling the visualization of cerebral blood volume evolution along the temporal axis while distinguishing the concerned compartment, whether it originates from a vein or an artery.

#### 3.4.1. UNet as a backbone architecture

The UNet architecture, introduced by Ronneberger et al. [47], has become a leading approach in the field of biomedical image segmentation due to its robustness and efficiency. The

architecture is fundamentally a convolutional neural network that features a “U-shaped” design. It consists of a contracting path (downsampling) to capture context and a symmetric expanding path (upsampling) that enables precise localization. The multiple convolutions, pooling layers, and concatenation steps within the UNet help capture multi-scale contextual information without the need for separate processing pathways. In addition to its specific architecture, the UNet benefits from skip connections that connect layers in the contracting path with corresponding layers in the expanding path. These connections are crucial in recovering spatial information lost during downsampling in order to achieve high accuracy in segmentation tasks. Originally designed to work well with very few training images, the UNet leverages extensive data augmentation techniques such as random rotations and elastic deformations, making it particularly suited for medical imaging scenarios where annotated data can be scarce.

### 3.4.2. Selected models

As discussed in Section 2.2, there have been significant advances and enhancements in the UNet architecture. We selected a benchmark reviewed in [15] from their Awesome-U-Net<sup>1</sup> GitHub repository. In addition to the classic UNet architecture [48], four architectures were chosen for their ability to address various complexities encountered in medical image segmentation:

- **Residual Net [49]** combines the strengths of UNet with ResNet, allowing for deeper networks without the risk of vanishing gradients. It uses recurrent convolutions for improved feature representation.
- **UNet++ [48]** introduces nested, dense skip pathways. These additional connections help in better feature propagation across the network, reducing the semantic gap between the feature maps of the encoder and decoder stages.
- **MultiresNet [50]** employs a multi-resolution convolutional approach to handle diverse image scales effectively within the segmentation process. It integrates layers that process multiple scales simultaneously, which helps in capturing both global and local contextual details more effectively.
- **Attention UNet [51]**, by incorporating attention gates, selectively emphasizes important features and suppresses irrelevant neuron activation, which is crucial in medical images where the focus area is often surrounded by a large amount of noise.

The selected architectures are originally designed to take 2D images as input and to generate 2D segmentation. However, to leverage the temporal richness of our fUS data, we opted to feed the UNet with multiple frames from a stack as many input channels. This approach allowed us to exploit the inter-frame dynamics while preserving the 2D structure of the existing architectures, avoiding the added complexity of adapting the networks. This trade-off ensures optimal capture of spatio-temporal information while maintaining the efficiency of the pre-existing models.

In the repository, additional implementation of architectures such as TransUNet [52], UC-TransNet [53], and MissFormer [54] were also provided. While these architectures have demonstrated their effectiveness on medical image segmentation, we did not train them on our data due to their requirement for square images. In our specific case, we already resize ULM images. Introducing further pre-processing by resizing also fUS images could add more biases, making it difficult to ensure the accuracy and reproducibility of the segmentation.

---

<sup>1</sup><https://github.com/NITRO98/Awesome-U-Net/tree/main>



### 3.4.3. Selected losses

In this section,  $S$  represents the segmentation mask produced by the model from fUS, and  $T$  represents the ground truth from ULM.  $T$  consists of a ternary mask derived from vein and artery masks. Specifically, when needed, it is expressed as  $T_c$  where  $c \in \{a, v, b\}$  – with  $a$  indicating arteries,  $v$  indicating veins, and  $b$  representing the background.  $S$  can be specified in a similar manner when necessary. The dimensions of the image are given by the height  $h$  and width  $w$ .

For the first loss function, we proposed to use a combination of Cross-Entropy and Dice loss. Cross-entropy is effective for pixel-wise segmentation tasks. For each pixel  $i$  and class  $c$ , the Cross-Entropy Loss is defined as:

$$\text{CE}_i(S, T) = - \sum_{c=1}^C T_{i,c} \log(S_{i,c}) \quad (1)$$

where  $T_{i,c}$  is the ground truth label for class  $c$  at pixel  $i$ , represented as a one-hot encoded vector, and  $S_{i,c}$  is the predicted probability for class  $c$  at pixel  $i$ . This approach penalizes incorrect classifications at the pixel level. To obtain the final Cross-Entropy term, the  $\text{CE}_i$  is averaged over all pixels in the image:

$$\text{Cross-Entropy Loss}(S, T) = \frac{1}{P} \sum_{i=1}^P \text{CE}_i(S, T) \quad (2)$$

where  $P$  is the total number of pixels in the image.

Dice loss is particularly useful for addressing class imbalances by focusing on the overlap between the predicted and ground truth masks. This is especially important in our case, as we have a significant class imbalance with far more background pixels compared to vein and artery pixels. Specifically, the Dice coefficient for a single class  $c$  is defined as follows:

$$\text{Dice}_c(S, T) = \frac{2 \sum_i S_{i,c} T_{i,c}}{\sum_i S_{i,c} + \sum_i T_{i,c} + \varepsilon} \quad (3)$$

where  $S_{i,c}$  is the predicted probability for class  $c$  at pixel  $i$ ,  $T_{i,c}$  is the ground truth label for class  $c$  at pixel  $i$ , and  $\varepsilon$  is a small constant added to avoid division by zero. The Dice Loss for each class is then calculated as:

$$\text{Dice Loss}_c(S, T) = 1 - \text{Dice}_c(S, T) \quad (4)$$

To obtain the final Dice term, we compute the Dice Loss for each class and then take the average across all classes  $C$ :

$$\text{Dice Loss}(S, T) = \frac{1}{C} \sum_{c=1}^C \text{Dice Loss}_c(S, T) \quad (5)$$

The final formulation of the combined loss function is given by a weighted sum of the previously detailed terms, as follows:

$$\text{LOSS}_{\text{DICE\_CE}}(S, T) = \alpha \text{Cross-Entropy Loss}(S, T) + \beta \text{Dice Loss}(S, T), \quad \alpha, \beta \in [0, 1] \quad (6)$$

where  $\alpha$  and  $\beta$  are weighting coefficients that balance the contribution of each loss term, and they are not required to sum to 1.

For the next loss functions, in combination with Cross-Entropy, we incorporate a regularization approach introduced in [42]. This approach includes two key terms: vessel density and fractal dimension. The vessel density measures a ratio of vessel area to a whole area as follows:

$$\text{Loss}_V(S, T) = \left| \frac{\sum_i S_{i,a} - \sum_i T_{i,a}}{h \times w} \right| + \left| \frac{\sum_i S_{i,v} - \sum_i T_{i,v}}{h \times w} \right|$$

In the above equation,  $\sum_i S_{i,a}$  and  $\sum_i S_{i,v}$  represent the counts of pixels classified as arteries and veins in the segmentation mask  $S$ , respectively, while  $\sum_i T_{i,a}$  and  $\sum_i T_{i,v}$  represent the corresponding counts from the ground truth  $T$ .

To determine the fractal dimension  $d_f$  of vessels, we rely on the Minkowski-Bouligand dimension [55], commonly referred to as the box-counting dimension. This measure evaluates the complexity of vessel morphology by covering the image with a grid of boxes of a given size and counting how many boxes contain a part of the vessel structure. As the box size  $\varepsilon$  decreases, more boxes are required to cover the vessels, and the scaling behavior  $N \propto \varepsilon^{d_f}$  with  $N$  the number of required boxes, defines the fractal dimension  $d_f$ . More explicitly, we set the box sizes  $\varepsilon = \{2^i | i \in \mathbb{Z}, 2 \leq 2^i \leq \min\{h, w\}\}$  and count the box number  $N_S(\varepsilon)$  needed to cover the segmentation mask and  $N_T(\varepsilon)$  to cover ground truth mask. The loss is then calculated as:

$$\text{Loss}_B(S, T) = \sum_c \frac{1}{\sqrt{\sum_i \varepsilon_i^2}} \cdot \sum_i \sqrt{\varepsilon_i \cdot \left( \frac{N_{T_c}(\varepsilon_i) - N_{S_c}(\varepsilon_i)}{N_{T_c}(\varepsilon_i)} \right)^2}$$

$\varepsilon_i$  was empirically configured to weight more on the error of large-size box.

To achieve accurate multi-class segmentation masks, we construct the *CF*-Loss (Clinically-relevant Feature-optimised Loss) by combining the feature-based loss functions,  $\text{Loss}_V$  and  $\text{Loss}_B$ , with the pixel-wise Cross-Entropy Loss. Following the approach proposed in [42], we formulate the *CF*-Loss in three configurations to balance clinical relevance with segmentation accuracy:

$$\text{Loss}_{CF,B}(S, T) = \alpha \text{Cross-Entropy Loss}(S, T) + \gamma \text{Loss}_B(S, T) \quad (7)$$

$$\text{Loss}_{CF,V}(S, T) = \alpha \text{Cross-Entropy Loss}(S, T) + \beta \text{Loss}_V(S, T) \quad (8)$$

$$\text{Loss}_{CF}(S, T) = \alpha \text{Cross-Entropy Loss}(S, T) + \gamma \text{Loss}_B(S, T) + \beta \text{Loss}_V(S, T) \quad (9)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are the loss weights for Cross-Entropy, vessel density and box count terms respectively. The values used for the weights are given in Section 4.3 below.

## 4. Experiments

### 4.1. Dataset

As mentioned in the Method Section, our data were derived from experiments on 35 different rats, providing us with 35 fUS stacks in both resting state and under visual stimulation, annotated as described in Section 3.3. No pre-processing was applied to these fUS images. To improve the robustness and generalizability of our models, we employed data augmentation techniques, including random horizontal and vertical flips and random rotations. This augmentation was applied in real-time during the training phase, where each frame of the input stack was modified with a random transformation at each training iteration. The corresponding ULM masks were augmented with the same transformations to maintain precise alignment with the fUS frames. To ensure accuracy in rotation, these flips and rotations were performed after resizing. Finally, we conducted a 7-fold cross-validation to ensure the reliability of our results, using data from 30 experiments randomly chosen for training and reserving 5 for testing.

#### 4.2. Evaluation metrics

To provide a comprehensive evaluation of the segmentation performance, we selected the metrics detailed above:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad \text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In this multi-class setting,  $TP$  (True Positives),  $TN$  (True Negatives),  $FP$  (False Positives), and  $FN$  (False Negatives) denote the counts for each pixel classification category.

In addition, we also used the Jaccard Index, also known as Intersection over Union (IoU). It is calculated for each class  $c$  individually, and then averaged to provide an overall score:

$$\text{Jaccard}_c = \frac{|T_c \cap S_c|}{|T_c \cup S_c|}, \quad \text{Jaccard Index} = \frac{1}{C} \sum_{c=1}^C \text{Jaccard}_c$$

$T$  and  $S$  denote the ground truth and segmentation masks respectively, as introduced before.

#### 4.3. Hyperparameter Setup

For the configuration of the selected deep learning models, we used the hyperparameters recommended in the GitHub repository from which the models were sourced. The only changes made were to the input and output channels; the output was consistently set to three to predict a ternary mask. For the input, as mentioned in Section 3.4.2, we treated the temporal dimension of the fUS stack as an image channel, setting it to 3 000 or 825 depending on the training set (resting state or under visual stimulation respectively). We chose weights for different losses as follows:  $\alpha$  and  $\beta$  at 0.5 in  $LOSS_{DICE\_CE}$ ,  $\alpha$  at 1 for all  $CF$  losses,  $\beta$  and  $\gamma$  at 1 in  $LOSS_{CF\_B}$  and  $LOSS_{CF\_V}$  respectively, and  $\gamma$  set at 0.5 for  $LOSS_{CF}$  as per the default settings in the reference paper. After various trials and errors, we settled on using the Adam optimizer with a learning rate of  $10^{-3}$  across 200 epochs for all training sessions.

#### 4.4. Results & Discussion

##### 4.4.1. Comparison among models and losses

In this experiment, we trained the selected models using the selected four loss functions on the resting state dataset, with the aim of identifying the most efficient model and the most appropriate loss function. We reported the average metrics across a 7-fold cross-validation in Table 1, page 13.

The highest performance metrics achieved were an accuracy of 89%, an F1 Score of 69%, and a Jaccard Index (IoU) of 0.56, aligning with results typically seen in vessel segmentation [42, 41]. The best results on average are predominantly achieved by the Attention UNet and UNet++, which seem better suited for this segmentation task due to their ability to focus on relevant regions within the image through attention mechanism. The UNet and ResNet performed slightly worse yet remained competitive, whereas the MultiResNet was noticeably less effective. This can be explained by its inability to effectively capture the fine details due to its more complex and less targeted architecture. In terms of loss functions,  $CF$  generally outperformed  $DICE\_CE$  for Attention UNet, UNet++, and UNet, with negligible differences among the  $CF$  variants,

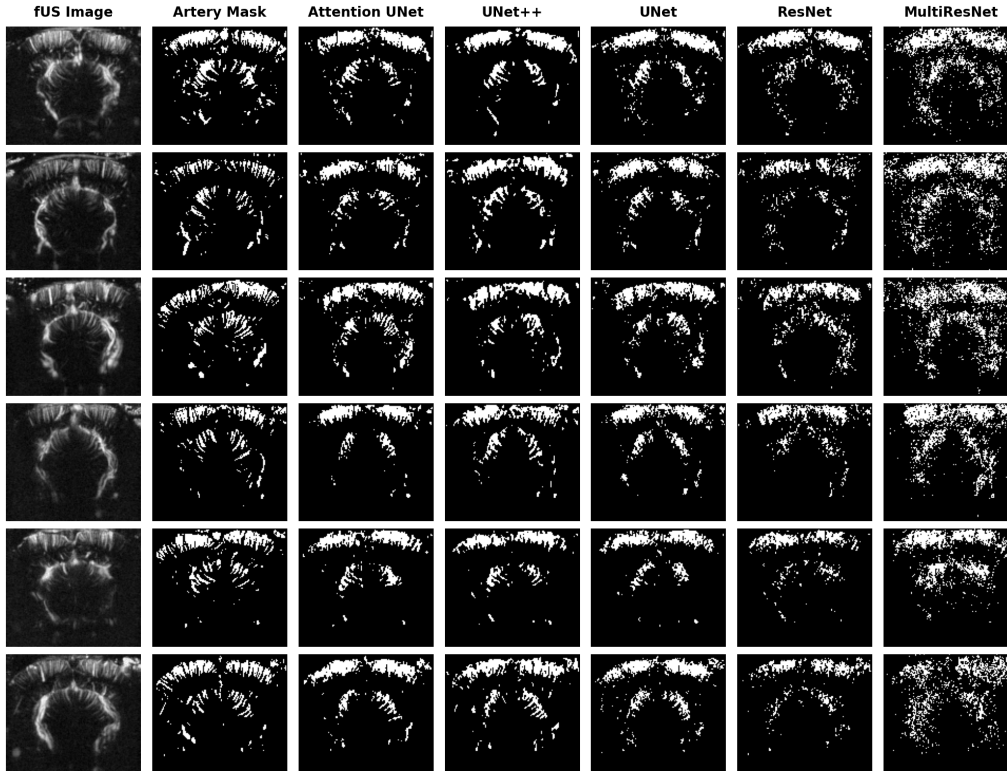


Figure 3: Artery/Downward prediction examples using CF Loss. Each row depicts a different image, starting with the first frame of the fUS stack in a log scale, followed by the constructed artery mask from ULM, and the segmentation proposed by the five benchmarked models.

suggesting that both fractal dimension and vessel density allowed the models to better identify vascular structures and distinguish vessel boundaries, improving segmentation accuracy.

For illustration, we presented predictions for six images in Figures 3 and 4 for arteries and veins, respectively. These images were selected to showcase different cerebral vascular structures. The predictions from Attention UNet and UNet++ align well with the performance metrics, closely matching the ground truth in the six examples. However, they struggle slightly with fine arterial structures at the cortical level and horizontal vessels, as seen in example 4. UNet and ResNet predictions respect the overall shape of the six images, but there are many missing structures, indicating that these models may not capture all the intricate details effectively. Finally, the segmentation of MultiResNet is notably imprecise, yet it shows a slightly better performance in segmenting veins, which have a simpler structure.

In the above experiment, we used the full fUS stack of 3 000 frames as input for training. We then repeated the experiment, this time using a single image per fUS stack, generated by averaging the 3 000 frames captured at rest. No statistically significant difference in overall performance was observed. Detailed results are provided in the appendix.

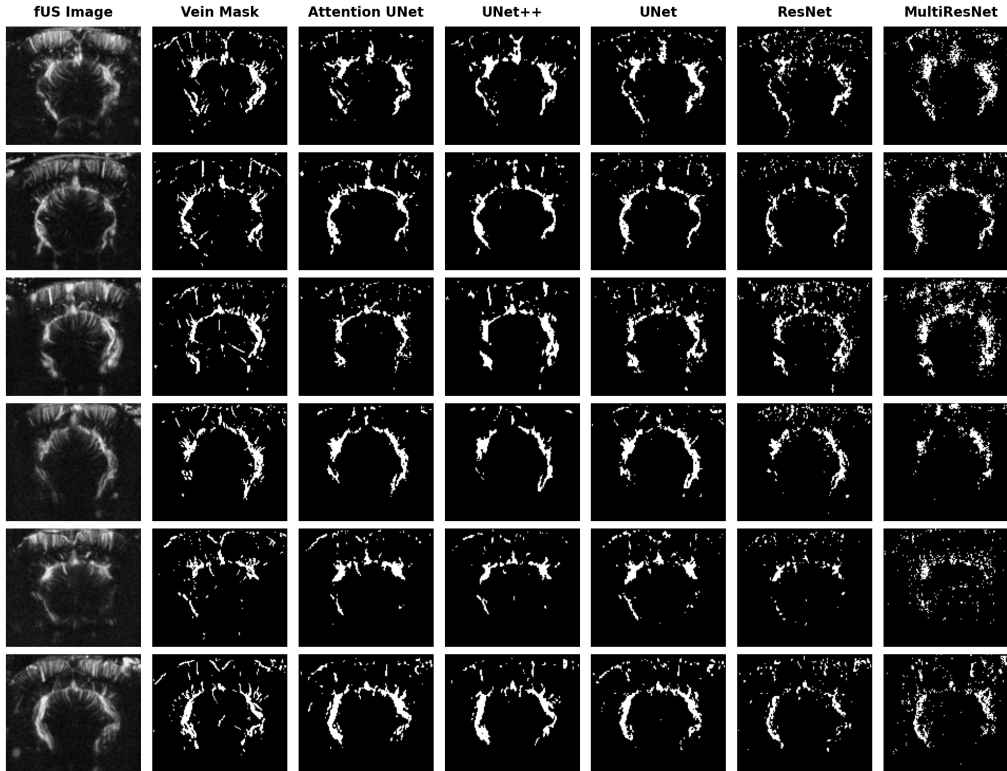


Figure 4: Vein/Upward prediction examples using CF Loss. Each row depicts a different image, starting with the first frame of the fUS stack in log scale, followed by the constructed vein mask from ULM, and the segmentation proposed by the five benchmarked models.

#### 4.4.2. Evaluating the Impact of fUS stack depth on segmentation performance

The primary goal of this experiment was to investigate how the depth of the fUS stack, defined by the number of frames used during training, impacts prediction quality. To address this question, we focused on the Attention UNet, one of the top-performing models from previous experiments, and trained it using CF loss on fUS stacks with varying numbers of frames. This analysis was conducted both at rest and under visual stimulation conditions, given the different signal evolution of cerebral blood volume observed in each state. We maintained the same sampling frequency for the stacks, varying only the number of frames,  $n$ , selected from the sets  $\{1, 100, 825, 3000\}$  for the resting state and  $\{1, 100, 825\}$  for visual stimulation. If the number of selected frames,  $n$ , was smaller than the total available frames,  $N$  (i.e., 3000 for resting state or 825 for visual stimulation), we randomly selected  $n$  contiguous frames from the full stack. Specifically, a random index  $i$  was drawn from the distribution  $i \sim \text{Uniform}(1, N - n)$  and the  $i + n$  frames were used. This random selection was performed at every training step, meaning that different frames were sampled from each stack in every epoch of the training process. The results for the F1 score and Jaccard Index are reported in Figure 5.

These results show that both metrics – F1 score and Jaccard Index – exhibited consistent trends across training scenarios in resting state and under visual stimulation, suggesting that

Table 1: Performance comparison of models and loss functions trained on full fUS stacks at rest. Bold values represent the best performance on average for each model across the loss functions. Statistical significance tests between model-loss pairs are detailed in Appendix A.

Attention UNet						
Loss	Accuracy $\uparrow$	F1 Score $\uparrow$	Precision $\uparrow$	Recall $\uparrow$	Jaccard Index $\uparrow$	Specificity $\uparrow$
<i>DICE_CE</i>	0.89 $\pm$ 0.01	0.67 $\pm$ 0.02	0.70 $\pm$ 0.02	0.65 $\pm$ 0.03	0.55 $\pm$ 0.02	0.86 $\pm$ 0.02
<i>CF_B</i>	0.89 $\pm$ 0.01	0.67 $\pm$ 0.02	0.70 $\pm$ 0.02	0.65 $\pm$ 0.03	0.54 $\pm$ 0.02	0.85 $\pm$ 0.01
<i>CF_V</i>	0.89 $\pm$ 0.01	0.68 $\pm$ 0.01	0.70 $\pm$ 0.03	0.67 $\pm$ 0.02	0.55 $\pm$ 0.01	0.86 $\pm$ 0.01
<i>CF</i>	<b>0.89 <math>\pm</math> 0.01</b>	<b>0.69 <math>\pm</math> 0.01</b>	<b>0.71 <math>\pm</math> 0.02</b>	<b>0.68 <math>\pm</math> 0.02</b>	<b>0.56 <math>\pm</math> 0.01</b>	<b>0.86 <math>\pm</math> 0.01</b>
UNet++						
<i>DICE_CE</i>	0.89 $\pm$ 0.01	0.67 $\pm$ 0.01	0.69 $\pm$ 0.01	0.66 $\pm$ 0.02	0.54 $\pm$ 0.01	0.86 $\pm$ 0.01
<i>CF_B</i>	0.88 $\pm$ 0.01	0.67 $\pm$ 0.01	0.68 $\pm$ 0.02	0.66 $\pm$ 0.01	0.54 $\pm$ 0.01	0.86 $\pm$ 0.00
<i>CF_V</i>	0.89 $\pm$ 0.01	0.67 $\pm$ 0.01	0.70 $\pm$ 0.02	0.67 $\pm$ 0.02	0.56 $\pm$ 0.01	0.87 $\pm$ 0.01
<i>CF</i>	<b>0.89 <math>\pm</math> 0.01</b>	<b>0.69 <math>\pm</math> 0.01</b>	<b>0.71 <math>\pm</math> 0.01</b>	<b>0.68 <math>\pm</math> 0.02</b>	<b>0.56 <math>\pm</math> 0.01</b>	<b>0.87 <math>\pm</math> 0.01</b>
ResNet						
<i>DICE_CE</i>	0.88 $\pm$ 0.01	0.63 $\pm$ 0.01	<b>0.68 <math>\pm</math> 0.02</b>	0.61 $\pm$ 0.02	0.51 $\pm$ 0.01	<b>0.84 <math>\pm</math> 0.01</b>
<i>CF_B</i>	<b>0.88 <math>\pm</math> 0.01</b>	<b>0.64 <math>\pm</math> 0.02</b>	0.67 $\pm$ 0.03	<b>0.62 <math>\pm</math> 0.02</b>	<b>0.51 <math>\pm</math> 0.01</b>	0.83 $\pm$ 0.01
<i>CF_V</i>	0.88 $\pm$ 0.01	0.63 $\pm$ 0.01	0.67 $\pm$ 0.02	0.61 $\pm$ 0.02	0.51 $\pm$ 0.01	0.84 $\pm$ 0.02
<i>CF</i>	0.87 $\pm$ 0.01	0.61 $\pm$ 0.01	0.65 $\pm$ 0.02	0.59 $\pm$ 0.03	0.49 $\pm$ 0.01	0.84 $\pm$ 0.02
UNet						
<i>DICE_CE</i>	0.88 $\pm$ 0.01	0.66 $\pm$ 0.01	<b>0.69 <math>\pm</math> 0.02</b>	0.64 $\pm$ 0.02	0.53 $\pm$ 0.01	0.85 $\pm$ 0.01
<i>CF_B</i>	0.88 $\pm$ 0.01	0.65 $\pm$ 0.01	0.67 $\pm$ 0.02	0.64 $\pm$ 0.01	0.52 $\pm$ 0.01	0.85 $\pm$ 0.01
<i>CF_V</i>	0.88 $\pm$ 0.01	0.67 $\pm$ 0.02	0.69 $\pm$ 0.03	0.66 $\pm$ 0.02	0.54 $\pm$ 0.02	0.86 $\pm$ 0.01
<i>CF</i>	<b>0.88 <math>\pm</math> 0.01</b>	<b>0.67 <math>\pm</math> 0.01</b>	0.68 $\pm$ 0.02	<b>0.67 <math>\pm</math> 0.02</b>	<b>0.54 <math>\pm</math> 0.01</b>	<b>0.86 <math>\pm</math> 0.01</b>
MultiresNet						
<i>DICE_CE</i>	0.84 $\pm$ 0.02	0.57 $\pm$ 0.02	0.57 $\pm$ 0.02	0.57 $\pm$ 0.01	0.45 $\pm$ 0.02	0.82 $\pm$ 0.01
<i>CF_B</i>	<b>0.84 <math>\pm</math> 0.01</b>	<b>0.57 <math>\pm</math> 0.02</b>	<b>0.57 <math>\pm</math> 0.03</b>	<b>0.59 <math>\pm</math> 0.02</b>	<b>0.45 <math>\pm</math> 0.02</b>	<b>0.84 <math>\pm</math> 0.01</b>
<i>CF_V</i>	0.83 $\pm$ 0.01	0.56 $\pm$ 0.02	0.57 $\pm$ 0.01	0.57 $\pm$ 0.03	0.44 $\pm$ 0.01	0.83 $\pm$ 0.03
<i>CF</i>	0.83 $\pm$ 0.01	0.56 $\pm$ 0.02	0.55 $\pm$ 0.02	0.58 $\pm$ 0.03	0.44 $\pm$ 0.02	0.83 $\pm$ 0.02

signal evolution does not significantly affect segmentation quality. In particular, reducing the number of frames remarkably improved these metrics, achieving maximal Jaccard Index of 0.60 and an F1 score above 0.72. A paired Wilcoxon test confirmed these improvements as statistically significant (p-value < 0.05) for the depths 1, 100 or 825 compared to 3000. This indicates that the model becomes easier to train with less parameters. Among the frame sets tested – 1, 100, and 825 – the boxplot using 100 frames ranks highest, particularly under visual stimulation. This suggests that a single frame lacks adequate information and using 825 frames might overly complicate the model, reducing its effectiveness. Thus, it appears that using 100 frames provides sufficient data for accurate segmentation without overburdening the model.

#### 4.4.3. Cross-condition efficacy of fUS-based models: Training on resting state for visual stimulation segmentation

We found above that brain signal changes captured by fUS during visual stimulation did not significantly influence the quality of segmentation, as similar results were achieved with or without visual stimulation. This led us to explore whether a model trained on fUS data from a resting state could effectively segment images acquired during visual stimulation. To this end, we trained the Attention UNet model on fUS images at rest (with 100 frames) and tested it on images from visually stimulated sessions (also with 100 frames). To ensure a clear separation between

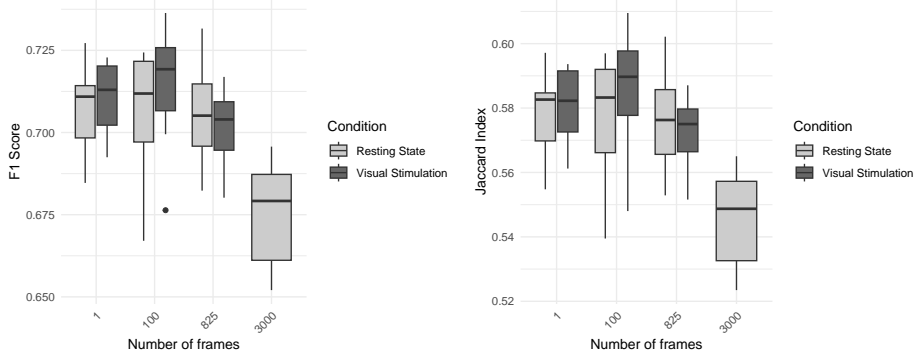


Figure 5: Box plots illustrating the F1 score and Jaccard Index for predictions made by an Attention UNet trained on fUS stacks of varying depths (1, 100, 825, 3 000), both in resting state and under visual stimulation. A Wilcoxon paired test showed a statistically significant difference between the depths of 1, 100, 825 and 3 000, whereas no significant difference is observed between the depths of 1, 100, 825.

Table 2: Performance of Attention UNet trained, with a  $CF$  Loss, on fUS stacks (100 frames) in resting state and tested on fUS stacks (100 frames) under visual stimulation.

Model	Accuracy $\uparrow$	F1 Score $\uparrow$	Precision $\uparrow$	Recall $\uparrow$	Jaccard Index $\uparrow$	Specificity $\uparrow$
Attention-UNet	$0.90 \pm 0.00$	$0.71 \pm 0.01$	$0.73 \pm 0.02$	$0.70 \pm 0.02$	$0.59 \pm 0.01$	$0.89 \pm 0.01$

training and testing data, we used different groups of rats for each condition: fUS images from resting state sessions were used for training, while only fUS images from visual stimulation sessions of separate rats were used for testing. The goal was to ensure that the model was evaluated on new structural data.

The results for various metrics are presented in Table 2, revealing an accuracy of 90%, an F1 score of 71%, and a Jaccard Index of 0.59. These outcomes are noteworthy as they align with previous results. This is particularly interesting because it shows that our model can be effectively trained on shorter fUS stacks in a resting state, and still perform well on data collected during visual stimulation without requiring retraining for the new conditions. This highlights the model applicability for accurately tracking differentiated activity in different vascular compartments.

#### 4.4.4. Visualizing vascular structures in fUS imaging

In this section, we illustrated that the UNet based inference of masks for veins and arteries (upward and downward compartments respectively) can be effectively used to enhance the interpretation of fUS signals.

Firstly, for a more visual rendering of our results, we overlaid real and model-predicted masks on an fUS image captured during visual stimulation (Figure 6, left). Arterial and venous signals are distinguished by red and blue colors, respectively, while the background is rendered in grayscale. The color intensity corresponds to the strength of the Power Doppler signal, indicative of Cerebral Blood Volume (CBV). Figure 6 illustrates the outcome of this visualization process. We surrounded regions where the model predictions were relatively accurate in green and areas where it struggled in yellow. It should be noted that the cortex is located at the top of the images. In the cortical region, the predicted mask accurately delineates arteries but does not capture all the veins, although it does detect a fair number of them. This difference in performance may be

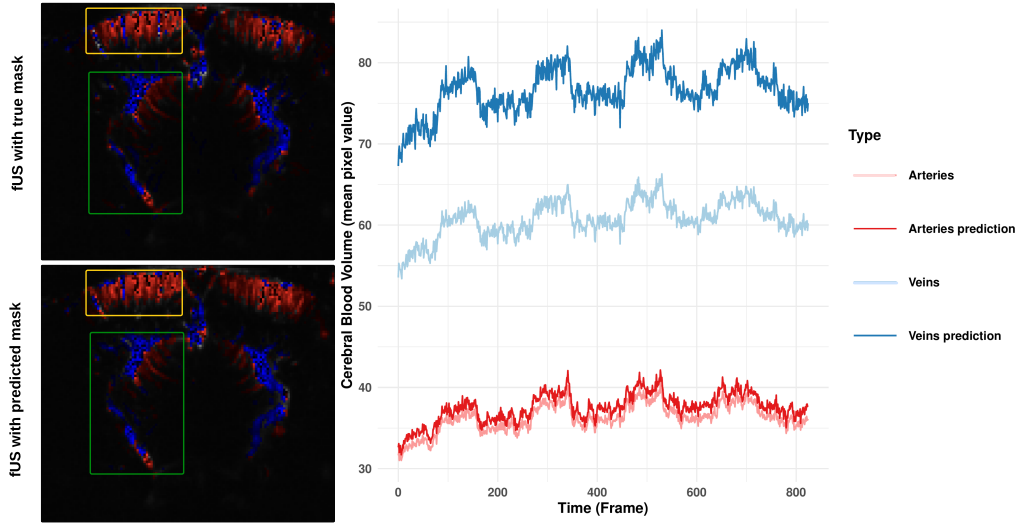


Figure 6: Images on the left show the projection of real masks of vein and artery compared to predicted ones on a random frame taken from a fUS stack acquired under visual stimulation. Veins are depicted in blue and arteries in red, with color intensity reflecting the cerebral signal strength during visual stimulation. Green rectangles highlight regions predicted with relative accuracy, while yellow rectangles indicate areas where the model struggled more. The figure to right shows the temporal evolution of the mean signal intensity in the cortical region from the same stack of fUS, separating signal from pixels classified as veins or arteries and displaying the predictions.

due to the denser presence of arteries compared to veins in this region. In contrast, the model performs particularly well in the lower regions, where the signal appears more visible, and it effectively identifies both downward and upward compartments.

This visualization was complemented by extracting the signal evolution in the cortical region across different compartments within the same fUS stack. Specifically, we averaged the pixels classified as veins or arteries per frame in the cortex, allowing us to visualize the evolution of the CBV in the graph to the right of Figure 6. Initially, we can distinguish four patterns corresponding to the response to four sessions of visual stimulation. The signals from both predicted and actual arteries are almost identical, achieving a Pearson correlation coefficient of 0.99. However, a difference in intensity between the signals from predicted and actual veins is noticeable. This discrepancy may be explained by the model inability to accurately identify veins in the cortical region due to their lower density in fUS images. This is consistent with the illustration showing the model limitations on the left. Aside from this, the signal follows the same trend, and the Pearson correlation coefficient is 0.98, indicating a strong linear correlation. We calculated the average correlation coefficients across all rats and achieved values of 0.98 for arteries and 0.55 for veins in the cortical region and 0.87 for downward flow and 0.98 for upward flow in the lower region.

In conclusion, while the model effectively identifies arterial structures within fUS imaging, it struggles more with accurately capturing venous structures in the cortical region. This limitation highlights the need for model enhancements to improve detection capabilities for less dense vascular features. Nonetheless, the correlation remains strong in both compartments, underscoring the model overall utility.



## 5. Conclusion and Perspectives

In this article, we introduced the first study on vascular segmentation of the brain using fUS imaging. Annotations were automatically derived from costly ultrasound localization microscopy, which is no longer required after model training to detect upward and downward flows. We evaluated several segmentation models, originally developed for other imaging modalities, and achieved competitive performance comparable to existing vascular analysis methods. Notably, using only 100 frames of the fUS stack during training yielded optimal results, with 90% accuracy and an F1 score of 71%. Furthermore, our findings showed that training the model on fUS data acquired in a resting state is sufficient for accurately segmenting images obtained during visual stimulation sessions. This demonstrates the model potential for analyzing brain activity during task-based experiments. The predicted masks also provided significant value in enhancing the interpretation of fUS Power Doppler signals, which is particularly useful for neuroscientists conducting preclinical studies. Our pipeline demonstrated a high level of agreement between predicted and actual signals, with average linear correlation coefficients of 0.98 for arteries and 0.55 for veins in the cortical region, and 0.87 for downward and 0.98 for upward flows in deeper regions. Although the model effectively captures arterial structures, it faces challenges with venous segmentation in the cortical region, highlighting potential areas for improvement. Nevertheless, the consistently high correlation underscores the model robustness and overall reliability. However, the model generalization to other brain regions requires staying within the same imaging plane, meaning a new model must be trained for each plane with ULM data. While this introduces certain constraints, the ongoing development of 3D fUS and ULM technologies is expected to mitigate this limitation in the near future. As a proof of concept, this work lays the foundation for further advances. Future improvements could include partitioning fUS images into patches and training the model to segment regions independently of their location. Additionally, a better use of the temporal dimension of fUS stacks could be achieved by transforming them into a suitable latent space using advanced techniques like the Inception Time model [56] to capture signal variations more effectively. Another promising direction is to develop a UNet architecture specifically tailored to the unique signal characteristics of fUS, particularly improving its sensitivity to less dense vascular structures, or to leverage advanced models such as the Dual Multi-Scale Attention UNet (DMSA-UNet) [57], which has shown strong potential in medical image segmentation.

## References

- [1] E. Macé, G. Montaldo, I. Cohen, M. Baulac, M. Fink, M. Tanter, Functional ultrasound imaging of the brain, *Nature methods* 8 (8) (2011) 662–664.
- [2] T. Deffieux, C. Demene, M. Pernot, M. Tanter, Functional ultrasound neuroimaging: a review of the preclinical and clinical state of the art, *Current opinion in neurobiology* 50 (2018) 128–135.
- [3] T. Deffieux, C. Demené, M. Tanter, Functional ultrasound imaging: a new imaging modality for neuroscience, *Neuroscience* 474 (2021) 110–121.
- [4] B. Vidal, M. Droguerre, L. Venet, L. Zimmer, M. Valdebenito, F. Mouthon, M. Charvériat, Functional ultrasound imaging to study brain dynamics: application of pharmaco-fUS to atomoxetine, *Neuropharmacology* 179 (2020) 108273.
- [5] M. Droguerre, B. Vidal, M. Valdebenito, F. Mouthon, L. Zimmer, M. Charvériat, Impaired local and long-range brain connectivity and visual response in a genetic rat model of hyperactivity revealed by functional ultrasound, *Frontiers in Neuroscience* 16 (2022) 865140.
- [6] B. Vidal, M. Pereira, M. Valdebenito, L. Vidal, F. Mouthon, L. Zimmer, M. Charvériat, M. Droguerre, Pharmaco-fUS in cognitive impairment: lessons from a preclinical model, *Journal of Psychopharmacology* 36 (11) (2022) 1273–1279.

- [7] H. Chen, S. Mirg, P. Gaddale, S. Agrawal, M. Li, V. Nguyen, T. Xu, Q. Li, J. Liu, W. Tu, X. Liu, P. J. Drew, N. Zhang, B. J. Gluckman, S.-R. Kothapalli, Dissecting multiparametric cerebral hemodynamics using integrated ultrafast ultrasound and multispectral photoacoustic imaging, *bioRxiv* (2023).
- [8] Y. He, M. Wang, X. Chen, R. Pohmann, J. R. Polimeni, K. Scheffler, B. R. Rosen, D. Kleinfeld, X. Yu, Ultra-slow single-vessel bold and cbv-based fmri spatiotemporal dynamics and their correlation with neuronal intracellular calcium signals, *Neuron* 97 (4) (2018) 925–939.
- [9] N. Renaudin, C. Demené, A. Dizeux, N. Ialy-Radio, S. Pezet, M. Tanter, Functional ultrasound localization microscopy reveals brain-wide neurovascular activity on a microscopic scale, *Nature methods* 19 (8) (2022) 1004–1012.
- [10] C. Bourquin, J. Poree, F. Lesage, J. Provost, In vivo pulsatility measurement of cerebral microcirculation in rodents using dynamic ultrasound localization microscopy, *IEEE Transactions on Medical Imaging* 41 (4) (2021) 782–792.
- [11] Y. Qi, M. Roper, Control of low flow regions in the cortical vasculature determines optimal arterio-venous ratios, *Proceedings of the National Academy of Sciences* 118 (34) (2021) e2021840118.
- [12] J. J. Iliff, M. Wang, D. M. Zeppenfeld, A. Venkataraman, B. A. Plog, Y. Liao, R. Deane, M. Nedergaard, Cerebral arterial pulsation drives paravascular csf–interstitial fluid exchange in the murine brain, *Journal of Neuroscience* 33 (46) (2013) 18190–18199.
- [13] C. Errico, J. Pierre, S. Pezet, Y. Desailly, Z. Lenkei, O. Couture, M. Tanter, Ultrafast ultrasound localization microscopy for deep super-resolution vascular imaging, *Nature* 527 (7579) (2015) 499–502.
- [14] O. Couture, V. Hingot, B. Heiles, P. Muleki-Seya, M. Tanter, Ultrasound localization microscopy and super-resolution: A state of the art, *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 65 (8) (2018) 1304–1320.
- [15] R. Azad, E. K. Aghdam, A. Rauland, Y. Jia, A. H. Avval, A. Bozorgpour, S. Karimijafarbigloo, J. P. Cohen, E. Adeli, D. Merhof, Medical image segmentation review: The success of U-Net, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024) 1–20.
- [16] T. Di Ianni, R. D. Airan, Deep-fUS: A deep learning platform for functional ultrasound imaging of the brain using sparse data, *IEEE Transactions on Medical Imaging* 41 (7) (2022) 1813–1825.
- [17] R. J. G. van Sloun, O. Solomon, M. Bruce, Z. Z. Khaing, H. Wijkstra, Y. C. Eldar, M. Mischi, Super-resolution ultrasound localization microscopy through deep learning, *IEEE Transactions on Medical Imaging* 40 (3) (2021) 829–839.
- [18] L. Milecki, J. Porée, H. Belgharbi, C. Bourquin, R. Damseh, P. Delafontaine-Martel, F. Lesage, M. Gasse, J. Provost, A deep learning framework for spatiotemporal ultrasound localization microscopy, *IEEE Transactions on Medical Imaging* 40 (5) (2021) 1428–1437.
- [19] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, A. K. Nandi, Medical image segmentation using deep learning: A survey, *IET Image Processing* 16 (5) (2022) 1243–1267.
- [20] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, A. K. Nandi, Medical image segmentation using deep learning: A survey, *IET Image Processing* 16 (5) (2022) 1243–1267.
- [21] L. Cai, Q. Li, J. Zhang, Z. Zhang, R. Yang, L. Zhang, Ultrasound image segmentation based on Transformer and U-Net with joint loss, *PeerJ Computer Science* 9 (2023) e1638.
- [22] E. Bourigault, D. R. McGowan, A. Mehranian, B. W. Papież, Multimodal PET/CT tumour segmentation and prediction of progression-free survival using a full-scale UNet with attention, in: *3D Head and Neck Tumor Segmentation in PET/CT Challenge, 2021*, pp. 189–201.
- [23] J. Dolz, C. Desrosiers, I. Ben Ayed, IVD-Net: Intervertebral disc localization and segmentation in MRI with a multi-modal UNet, in: *Proceedings of the International workshop and challenge on computational methods and clinical applications for spine imaging, 2018*, pp. 130–143.
- [24] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, P.-A. Heng, H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes, *IEEE transactions on medical imaging* 37 (12) (2018) 2663–2674.
- [25] F. Tang, L. Wang, C. Ning, M. Xian, J. Ding, CMU-Net: A strong ConvMixer-based medical ultrasound image segmentation network, in: *Proceedings of the 20th International Symposium on Biomedical Imaging (ISBI), 2023*, pp. 1–5.
- [26] Y. Peng, M. Sonka, D. Z. Chen, U-Net v2: Rethinking the skip connections of U-Net for medical image segmentation (2023).
- [27] X. Tong, J. Wei, B. Sun, S. Su, Z. Zuo, P. Wu, ASCU-Net: Attention gate, spatial and channel attention U-Net for skin lesion segmentation, *Diagnostics* 11 (3) (2021) 501.
- [28] Y. A. Ayalew, K. A. Fante, M. A. Mohammed, Modified U-Net for liver cancer segmentation from computed tomography images with a new class balancing method, *BMC Biomedical Engineering* 3 (1) (2021) 4.
- [29] K. Zhang, X. Yang, Y. Cui, J. Zhao, D. Li, Imaging segmentation mechanism for rectal tumors using improved U-Net, *BMC Medical Imaging* 24 (1) (2024) 95.
- [30] P. Pramanik, R. Pramanik, F. Schwenker, R. Sarkar, DBU-Net: Dual branch U-Net for tumor segmentation in breast ultrasound images, *PLOS ONE* 18 (11) (2023) e0293615.

- [31] S. Qamar, H. Jin, R. Zheng, P. Ahmad, M. Usama, A variant form of 3D-UNet for infant brain segmentation, *Future Generation Computer Systems* 108 (2020) 613–623.
- [32] F. Milletari, N. Navab, S.-A. Ahmadi, V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: *Proceedings of the International Conference on 3D vision (3DV)*, IEEE, 2016, pp. 565–571.
- [33] R. Raza, U. I. Bajwa, Y. Mehmood, M. W. Anwar, M. H. Jamal, dResU-Net: 3D deep residual U-Net based brain tumor segmentation from multimodal MRI, *Biomedical Signal Processing and Control* 79 (2023) 103861.
- [34] Q. Huang, J. Sun, H. Ding, X. Wang, G. Wang, Robust liver vessel extraction using 3D U-Net with variant dice loss function, *Computers in biology and medicine* 101 (2018) 153–162.
- [35] S. Liu, Y. Wang, X. Yang, B. Lei, L. Liu, S. X. Li, D. Ni, T. Wang, Deep learning in medical ultrasound analysis: A review, *Engineering* 5 (2) (2019) 261–275.
- [36] M. E. Rayed, S. M. S. Islam, S. I. Niha, J. R. Jim, M. M. Kabir, M. F. Mridha, Deep learning for medical image segmentation: State-of-the-art advancements and challenges, *Informatics in Medicine Unlocked* 47 (2024) 101504.
- [37] M. R. Goni, N. I. R. Ruhaiyem, M. Mustapha, A. Achuthan, C. M. N. Che Mohd Nassir, Brain vessel segmentation using deep learning – A review, *IEEE Access* 10 (2022) 111322–111336.
- [38] C. Chen, J. H. Chuah, R. Ali, Y. Wang, Retinal vessel segmentation using deep learning: A review, *IEEE Access* 9 (2021) 111985–112004.
- [39] Y. Wu, S. Qi, M. Wang, S. Zhao, H. Pang, J. Xu, L. Bai, H. Ren, Transformer-based 3D U-Net for pulmonary vessel segmentation and artery-vein separation from CT images, *Medical & Biological Engineering & Computing* 61 (10) (2023) 2649–2663.
- [40] X. Yang, L. Xu, S. Yu, Q. Xia, H. Li, S. Zhang, Segmentation and vascular vectorization for coronary artery by geometry-based cascaded neural network, *IEEE Transactions on Medical Imaging* (2024).
- [41] S. Shit, J. C. Paetzold, A. Sekuboyina, I. Ezhov, A. Unger, A. Zhylyka, J. P. W. Pluim, U. Bauer, B. H. Menze, cDice – a novel topology-preserving loss function for tubular structure segmentation, in: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 16555–16564.
- [42] Y. Zhou, M. Xu, Y. Hu, S. B. Blumberg, A. Zhao, S. K. Wagner, P. A. Keane, D. C. Alexander, CF-Loss: Clinically-relevant feature optimised loss function for retinal multi-class vessel segmentation and vascular feature measurement, *Medical Image Analysis* 93 (2024) 103098.
- [43] J. Grandjean, G. Desrosiers-Gregoire, C. Anckaerts, D. Angeles-Valdez, F. Ayad, D. A. Barrière, I. Blockx, A. Bortel, M. Broadwater, B. M. Cardoso, et al., A consensus protocol for functional connectivity analysis in the rat brain, *Nature neuroscience* 26 (4) (2023) 673–681.
- [44] J. Bercoff, G. Montaldo, T. Loupas, D. Savery, F. Mézière, M. Fink, M. Tanter, Ultrafast compound doppler imaging: Providing full blood flow characterization, *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 58 (1) (2011) 134–147.
- [45] G. Montaldo, M. Tanter, J. Bercoff, N. Bencech, M. Fink, Coherent plane-wave compounding for very high frame rate ultrasonography and transient elastography, *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 56 (3) (2009) 489–506.
- [46] C. Demeñé, T. Defieux, M. Pernot, B.-F. Osmanski, V. Biran, J.-L. Gennisson, L.-A. Sieu, A. Bergel, S. Franqui, J.-M. Correas, et al., Spatiotemporal clutter filtering of ultrafast ultrasound data highly increases doppler and ultrasound sensitivity, *IEEE transactions on medical imaging* 34 (11) (2015) 2271–2285.
- [47] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: *Proceedings of the conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.
- [48] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, UNet++: A nested U-Net architecture for medical image segmentation, in: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer International Publishing, Cham, 2018, pp. 3–11.
- [49] M. Drozdal, E. Vorontsov, G. Chartrand, S. Kadoury, C. Pal, The importance of skip connections in biomedical image segmentation, in: *Proceedings of Conference on Deep Learning and Data Labeling for Medical Applications (DLMIA)*, Springer International Publishing, 2016, pp. 179–187.
- [50] N. Ibtihaz, M. S. Rahman, MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation, *Neural Networks* 121 (2020) 74–87.
- [51] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, et al., Attention U-Net: Learning where to look for the pancreas, *arXiv preprint arXiv:1804.03999* (2018).
- [52] J. Chen, J. Mei, X. Li, Y. Lu, Q. Yu, Q. Wei, X. Luo, Y. Xie, E. Adeli, Y. Wang, et al., TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers, *Medical Image Analysis* (2024) 103280.
- [53] H. Wang, P. Cao, J. Wang, O. R. Zaiane, UCTransNet: Rethinking the skip connections in U-Net from a channel-wise perspective with transformer, *Proceedings of the AAAI Conference on Artificial Intelligence* 36 (3) (2022) 2441–2449.
- [54] X. Huang, Z. Deng, D. Li, X. Yuan, Y. Fu, MISSFormer: An effective transformer for 2D medical image segmen-

- tation, *IEEE Transactions on Medical Imaging* 42 (5) (2023) 1484–1494.
- [55] K. Falconer, *Fractal geometry: mathematical foundations and applications*, 2014.
- [56] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P. Muller, F. Petitjean, InceptionTime: Finding AlexNet for time series classification, *Data Mining and Knowledge Discovery* 34 (2020) 1936–1962.
- [57] X. Li, C. Fu, Q. Wang, W. Zhang, C.-W. Sham, J. Chen, DMSA-UNet: Dual multi-scale attention makes UNet more strong for medical image segmentation, *Knowledge-Based Systems* 299 (2024) 112050.

## Appendix A. Statistical Significance of Model Comparisons (trained on full fUS stacks) Using the Wilcoxon Test

Table A.3: P-values from the paired Wilcoxon significance test for the different metrics from the training on the full fUS stacks (values lower than 0.05, highlighted in bold in the table, indicate that the observed differences are significant)

Comparison	F1Score	Accuracy	Precision	Recall	IoU	Specificity
(AttentionUNet-DICE_CE) vs (MultiresNet-DICE_CE)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-DICE_CE) vs (UNet+-DICE_CE)	0.47	0.58	<b>0.047</b>	0.94	0.47	0.94
(AttentionUNet-DICE_CE) vs (ResNet-DICE_CE)	<b>0.016</b>	<b>0.047</b>	0.22	<b>0.016</b>	<b>0.016</b>	0.078
(AttentionUNet-DICE_CE) vs (UNet-DICE_CE)	0.3	0.3	0.38	0.47	0.22	0.22
(AttentionUNet-DICE_CE) vs (AttentionUNet-CF_B)	0.22	0.58	0.58	0.38	0.22	0.3
(AttentionUNet-DICE_CE) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(AttentionUNet-DICE_CE) vs (UNet+-CF_B)	0.22	0.3	<b>0.031</b>	1	0.22	0.58
(AttentionUNet-DICE_CE) vs (ResNet-CF_B)	<b>0.031</b>	<b>0.016</b>	0.078	0.11	<b>0.031</b>	0.16
(AttentionUNet-DICE_CE) vs (UNet-CF_B)	<b>0.031</b>	0.11	<b>0.016</b>	0.078	<b>0.031</b>	0.16
(AttentionUNet-DICE_CE) vs (AttentionUNet-CF_V)	0.16	0.47	1	0.22	0.16	0.38
(AttentionUNet-DICE_CE) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(AttentionUNet-DICE_CE) vs (UNet+-CF_V)	0.16	0.58	0.94	0.38	0.16	0.22
(AttentionUNet-DICE_CE) vs (ResNet-CF_V)	<b>0.031</b>	0.11	<b>0.016</b>	<b>0.031</b>	<b>0.031</b>	0.16
(AttentionUNet-DICE_CE) vs (UNet-CF_V)	0.47	0.22	0.3	0.69	0.47	0.81
(AttentionUNet-DICE_CE) vs (AttentionUNet-CF)	0.16	0.3	0.3	0.3	0.16	0.38
(AttentionUNet-DICE_CE) vs (MultiresNet-CF)	<b>0.022</b>	<b>0.022</b>	<b>0.022</b>	<b>0.022</b>	<b>0.022</b>	0.2
(AttentionUNet-DICE_CE) vs (UNet+-CF)	0.16	0.47	0.94	0.11	0.16	0.16
(AttentionUNet-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(AttentionUNet-DICE_CE) vs (UNet-CF)	0.81	0.22	<b>0.031</b>	0.47	0.81	0.22
(MultiresNet-DICE_CE) vs (UNet+-DICE_CE)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (ResNet-DICE_CE)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (UNet-DICE_CE)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (AttentionUNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (MultiresNet-CF_B)	0.94	1	0.94	0.22	1	<b>0.047</b>
(MultiresNet-DICE_CE) vs (UNet+-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (ResNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(MultiresNet-DICE_CE) vs (UNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(MultiresNet-DICE_CE) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (MultiresNet-CF_V)	0.81	0.47	0.81	0.94	0.69	0.3
(MultiresNet-DICE_CE) vs (UNet+-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.11
(MultiresNet-DICE_CE) vs (UNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (MultiresNet-CF)	0.69	0.81	0.16	0.3	0.58	0.22
(MultiresNet-DICE_CE) vs (UNet+-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.22	<b>0.016</b>	0.16
(MultiresNet-DICE_CE) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet+-DICE_CE) vs (ResNet-DICE_CE)	<b>0.016</b>	0.3	0.58	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet+-DICE_CE) vs (UNet-DICE_CE)	0.22	0.81	0.22	<b>0.031</b>	0.16	<b>0.031</b>
(UNet+-DICE_CE) vs (AttentionUNet-CF_B)	1	0.38	0.38	0.69	1	0.11
(UNet+-DICE_CE) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet+-DICE_CE) vs (UNet+-CF_B)	0.69	0.38	0.69	0.69	0.69	0.81
(UNet+-DICE_CE) vs (ResNet-CF_B)	<b>0.016</b>	<b>0.047</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet+-DICE_CE) vs (UNet-CF_B)	<b>0.031</b>	0.16	0.16	<b>0.031</b>	<b>0.031</b>	<b>0.031</b>
(UNet+-DICE_CE) vs (AttentionUNet-CF_V)	<b>0.031</b>	0.22	0.16	<b>0.047</b>	<b>0.031</b>	0.3
(UNet+-DICE_CE) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet+-DICE_CE) vs (UNet+-CF_V)	0.078	0.58	0.16	0.078	0.078	<b>0.016</b>
(UNet+-DICE_CE) vs (ResNet-CF_V)	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet+-DICE_CE) vs (UNet-CF_V)	0.94	0.58	1	0.94	0.94	0.94
(UNet+-DICE_CE) vs (AttentionUNet-CF)	<b>0.031</b>	0.11	<b>0.016</b>	0.58	<b>0.047</b>	0.69
(UNet+-DICE_CE) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet+-DICE_CE) vs (UNet+-CF)	<b>0.016</b>	0.3	<b>0.016</b>	0.078	<b>0.016</b>	0.16
(UNet+-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet+-DICE_CE) vs (UNet-CF)	0.94	1	0.69	0.58	0.94	0.38
(ResNet-DICE_CE) vs (UNet-DICE_CE)	<b>0.016</b>	0.38	0.3	<b>0.031</b>	<b>0.016</b>	0.58
(ResNet-DICE_CE) vs (AttentionUNet-CF_B)	<b>0.016</b>	0.22	0.3	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(ResNet-DICE_CE) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	0.94
(ResNet-DICE_CE) vs (UNet+-CF_B)	<b>0.016</b>	0.47	0.58	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (ResNet-CF_B)	0.69	0.81	0.94	0.94	0.81	0.94

(ResNet-DICE_CE) vs (UNet-CF_B)	0.078	0.58	0.94	0.078	0.11	0.22
(ResNet-DICE_CE) vs (AttentionUNet-CF_V)	<b>0.016</b>	0.16	0.3	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.69
(ResNet-DICE_CE) vs (UNet++-CF_V)	<b>0.016</b>	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (ResNet-CF_V)	0.47	0.94	0.38	0.58	0.3	0.94
(ResNet-DICE_CE) vs (UNet-CF_V)	<b>0.016</b>	0.3	0.69	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(ResNet-DICE_CE) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.047</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(ResNet-DICE_CE) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.16	<b>0.016</b>	0.81
(ResNet-DICE_CE) vs (UNet++-CF)	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	0.16	0.078	0.38	<b>0.016</b>	1
(ResNet-DICE_CE) vs (UNet-CF)	<b>0.016</b>	0.47	1	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-DICE_CE) vs (AttentionUNet-CF_B)	0.58	0.69	0.81	0.38	0.58	0.38
(UNet-DICE_CE) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.16
(UNet-DICE_CE) vs (UNet++-CF_B)	0.81	0.94	0.3	0.11	0.81	0.078
(UNet-DICE_CE) vs (ResNet-CF_B)	<b>0.031</b>	0.3	0.16	<b>0.031</b>	<b>0.031</b>	0.38
(UNet-DICE_CE) vs (UNet-CF_B)	0.078	0.58	0.16	0.58	<b>0.031</b>	0.81
(UNet-DICE_CE) vs (AttentionUNet-CF_V)	<b>0.031</b>	0.078	0.94	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>
(UNet-DICE_CE) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.3
(UNet-DICE_CE) vs (UNet++-CF_V)	0.078	0.16	0.58	<b>0.031</b>	0.078	<b>0.016</b>
(UNet-DICE_CE) vs (ResNet-CF_V)	<b>0.016</b>	0.3	0.078	<b>0.016</b>	<b>0.016</b>	0.16
(UNet-DICE_CE) vs (UNet-CF_V)	0.81	0.81	0.58	0.22	0.81	0.47
(UNet-DICE_CE) vs (AttentionUNet-CF)	<b>0.031</b>	<b>0.047</b>	0.3	<b>0.031</b>	<b>0.031</b>	<b>0.047</b>
(UNet-DICE_CE) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.16
(UNet-DICE_CE) vs (UNet++-CF)	<b>0.016</b>	0.22	0.38	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	0.3
(UNet-DICE_CE) vs (UNet-CF)	0.58	0.69	0.38	<b>0.047</b>	0.47	<b>0.031</b>
(AttentionUNet-CF_B) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(AttentionUNet-CF_B) vs (UNet++-CF_B)	0.58	0.3	0.47	0.58	0.58	0.16
(AttentionUNet-CF_B) vs (ResNet-CF_B)	<b>0.047</b>	<b>0.016</b>	0.078	0.078	<b>0.047</b>	0.16
(AttentionUNet-CF_B) vs (UNet-CF_B)	0.11	<b>0.031</b>	0.078	0.3	0.078	0.58
(AttentionUNet-CF_B) vs (AttentionUNet-CF_V)	0.078	0.38	0.81	<b>0.016</b>	0.078	<b>0.031</b>
(AttentionUNet-CF_B) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(AttentionUNet-CF_B) vs (UNet++-CF_V)	<b>0.016</b>	0.47	0.47	<b>0.031</b>	<b>0.016</b>	<b>0.031</b>
(AttentionUNet-CF_B) vs (ResNet-CF_V)	<b>0.047</b>	0.078	0.11	0.078	<b>0.047</b>	0.3
(AttentionUNet-CF_B) vs (UNet-CF_V)	0.47	0.38	0.69	0.3	0.47	0.22
(AttentionUNet-CF_B) vs (AttentionUNet-CF)	<b>0.016</b>	0.11	0.58	0.16	<b>0.016</b>	0.22
(AttentionUNet-CF_B) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	0.16
(AttentionUNet-CF_B) vs (UNet++-CF)	<b>0.016</b>	0.38	0.47	<b>0.047</b>	<b>0.016</b>	0.078
(AttentionUNet-CF_B) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	<b>0.016</b>	0.11
(AttentionUNet-CF_B) vs (UNet-CF)	0.81	1	0.3	0.22	0.69	0.16
(MultiresNet-CF_B) vs (UNet++-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (ResNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.16	<b>0.016</b>	0.69
(MultiresNet-CF_B) vs (UNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(MultiresNet-CF_B) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (MultiresNet-CF_V)	0.11	0.94	0.94	0.47	0.11	0.81
(MultiresNet-CF_B) vs (UNet++-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.11	<b>0.016</b>	1
(MultiresNet-CF_B) vs (UNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(MultiresNet-CF_B) vs (MultiresNet-CF)	0.47	0.22	0.22	1	0.47	0.58
(MultiresNet-CF_B) vs (UNet++-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	1	<b>0.016</b>	0.94
(MultiresNet-CF_B) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF_B) vs (ResNet-CF_B)	<b>0.016</b>	<b>0.031</b>	0.58	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF_B) vs (UNet-CF_B)	<b>0.016</b>	0.22	0.47	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet++-CF_B) vs (AttentionUNet-CF_V)	<b>0.031</b>	0.078	0.22	0.16	<b>0.031</b>	0.38
(UNet++-CF_B) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet++-CF_B) vs (UNet++-CF_V)	<b>0.016</b>	0.47	0.11	0.16	<b>0.016</b>	0.16
(UNet++-CF_B) vs (ResNet-CF_V)	<b>0.031</b>	0.078	0.58	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>
(UNet++-CF_B) vs (UNet-CF_V)	0.58	0.94	0.81	0.38	0.58	0.94
(UNet++-CF_B) vs (AttentionUNet-CF)	<b>0.031</b>	0.078	0.078	<b>0.047</b>	<b>0.031</b>	0.3
(UNet++-CF_B) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF_B) vs (UNet++-CF)	<b>0.016</b>	0.11	0.078	<b>0.031</b>	<b>0.016</b>	0.16
(UNet++-CF_B) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(UNet++-CF_B) vs (UNet-CF)	0.38	1	0.94	0.22	0.38	0.58
(ResNet-CF_B) vs (UNet-CF_B)	<b>0.031</b>	0.81	0.81	0.16	0.078	0.3
(ResNet-CF_B) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078	<b>0.016</b>	0.58
(ResNet-CF_B) vs (UNet++-CF_V)	<b>0.016</b>	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (ResNet-CF_V)	0.94	1	0.94	0.81	1	0.94

(ResNet-CF_B) vs (UNet-CF_V)	0.078	0.16	0.47	<b>0.016</b>	0.078	<b>0.016</b>
(ResNet-CF_B) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.031</b>
(ResNet-CF_B) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.94
(ResNet-CF_B) vs (UNet++-CF)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (ResNet-CF)	0.078	<b>0.031</b>	0.47	0.47	0.078	1
(ResNet-CF_B) vs (UNet-CF)	<b>0.016</b>	0.078	0.38	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-CF_B) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	<b>0.016</b>	0.078
(UNet-CF_B) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.16
(UNet-CF_B) vs (UNet++-CF_V)	<b>0.016</b>	<b>0.031</b>	0.11	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet-CF_B) vs (ResNet-CF_V)	<b>0.031</b>	0.94	0.94	<b>0.047</b>	<b>0.031</b>	0.3
(UNet-CF_B) vs (UNet-CF_V)	0.22	0.22	0.58	<b>0.047</b>	0.16	0.22
(UNet-CF_B) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-CF_B) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	0.38
(UNet-CF_B) vs (UNet++-CF)	<b>0.016</b>	<b>0.031</b>	<b>0.047</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet-CF_B) vs (ResNet-CF)	<b>0.016</b>	0.16	<b>0.047</b>	<b>0.016</b>	<b>0.016</b>	0.11
(UNet-CF_B) vs (UNet-CF)	<b>0.031</b>	0.22	0.38	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF_V) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF_V) vs (UNet++-CF_V)	0.94	0.81	1	0.81	0.69	0.47
(AttentionUNet-CF_V) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.031</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF_V) vs (UNet-CF_V)	0.16	0.22	0.47	0.38	0.16	0.69
(AttentionUNet-CF_V) vs (AttentionUNet-CF)	1	0.94	0.58	0.81	1	0.94
(AttentionUNet-CF_V) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF_V) vs (UNet++-CF)	0.38	0.47	0.81	0.47	0.3	0.3
(AttentionUNet-CF_V) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(AttentionUNet-CF_V) vs (UNet-CF)	0.078	0.16	0.16	0.47	<b>0.016</b>	1
(MultiresNet-CF_V) vs (UNet++-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_V) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.3	<b>0.016</b>	0.58
(MultiresNet-CF_V) vs (UNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(MultiresNet-CF_V) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(MultiresNet-CF_V) vs (MultiresNet-CF)	1	0.3	0.11	0.69	1	0.81
(MultiresNet-CF_V) vs (UNet++-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(MultiresNet-CF_V) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.22	<b>0.016</b>	0.69
(MultiresNet-CF_V) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet++-CF_V) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF_V) vs (UNet-CF_V)	0.078	0.22	<b>0.047</b>	0.22	0.078	0.22
(UNet++-CF_V) vs (AttentionUNet-CF)	0.58	1	0.58	0.58	0.58	0.47
(UNet++-CF_V) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet++-CF_V) vs (UNet++-CF)	0.94	0.94	0.94	1	0.94	0.81
(UNet++-CF_V) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF_V) vs (UNet-CF)	<b>0.016</b>	0.078	0.078	0.38	<b>0.016</b>	0.69
(ResNet-CF_V) vs (UNet-CF_V)	<b>0.031</b>	0.3	0.47	<b>0.016</b>	<b>0.031</b>	0.078
(ResNet-CF_V) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_V) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.11	<b>0.016</b>	0.81
(ResNet-CF_V) vs (UNet++-CF)	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_V) vs (ResNet-CF)	0.078	0.16	0.11	0.38	<b>0.047</b>	0.94
(ResNet-CF_V) vs (UNet-CF)	<b>0.016</b>	0.38	0.47	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-CF_V) vs (AttentionUNet-CF)	0.11	0.16	0.22	0.47	0.16	0.47
(UNet-CF_V) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-CF_V) vs (UNet++-CF)	0.078	0.38	0.38	0.16	0.078	0.22
(UNet-CF_V) vs (ResNet-CF)	<b>0.016</b>	<b>0.031</b>	0.078	<b>0.031</b>	<b>0.016</b>	0.11
(UNet-CF_V) vs (UNet-CF)	0.81	0.81	0.94	0.81	0.94	0.58
(AttentionUNet-CF) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(AttentionUNet-CF) vs (UNet++-CF)	0.58	1	0.47	0.38	0.58	0.3
(AttentionUNet-CF) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(AttentionUNet-CF) vs (UNet-CF)	0.078	0.11	<b>0.031</b>	0.81	0.078	0.58
(MultiresNet-CF) vs (UNet++-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.81	<b>0.016</b>	1
(MultiresNet-CF) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(UNet++-CF) vs (UNet-CF)	<b>0.047</b>	0.22	<b>0.047</b>	0.3	<b>0.047</b>	0.58
(ResNet-CF) vs (UNet-CF)	<b>0.016</b>	<b>0.047</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>

## Appendix B. Comparison among models and losses trained on averaged fUS stacks

Table B.4: Performance comparison across models and losses trained on averaged fUS stacks. Bold values represent the best performance for each model across the loss functions. Statistical significance tests between model-loss pairs are detailed in B.5.

Attention UNet						
Loss	Accuracy	F1 Score	Precision	Recall	Jaccard Index	Specificity
<i>DICE_CE</i>	0.88 ± 0.01	0.67 ± 0.02	0.70 ± 0.02	0.65 ± 0.02	0.54 ± 0.01	0.85 ± 0.01
<i>CF_B</i>	0.89 ± 0.01	0.67 ± 0.02	0.69 ± 0.02	0.65 ± 0.02	0.54 ± 0.02	0.85 ± 0.01
<i>CF_V</i>	0.89 ± 0.01	0.68 ± 0.01	<b>0.71 ± 0.02</b>	0.67 ± 0.02	0.55 ± 0.01	0.86 ± 0.01
<i>CF</i>	<b>0.89 ± 0.01</b>	<b>0.68 ± 0.01</b>	0.70 ± 0.01	<b>0.68 ± 0.02</b>	<b>0.55 ± 0.01</b>	<b>0.86 ± 0.01</b>
MultiresNet						
<i>DICE_CE</i>	<b>0.84 ± 0.02</b>	<b>0.58 ± 0.02</b>	<b>0.57 ± 0.02</b>	<b>0.60 ± 0.01</b>	<b>0.46 ± 0.02</b>	<b>0.84 ± 0.01</b>
<i>CF_B</i>	0.83 ± 0.01	0.56 ± 0.02	0.57 ± 0.01	0.57 ± 0.03	0.44 ± 0.01	0.83 ± 0.01
<i>CF_V</i>	0.82 ± 0.01	0.55 ± 0.03	0.55 ± 0.02	0.58 ± 0.03	0.43 ± 0.02	0.83 ± 0.02
<i>CF</i>	0.82 ± 0.03	0.56 ± 0.03	0.56 ± 0.03	0.59 ± 0.03	0.44 ± 0.03	0.84 ± 0.01
UNet++						
<i>DICE_CE</i>	0.88 ± 0.01	0.67 ± 0.01	0.69 ± 0.01	0.66 ± 0.02	0.54 ± 0.01	0.86 ± 0.01
<i>CF_B</i>	0.88 ± 0.01	0.66 ± 0.01	0.68 ± 0.02	0.65 ± 0.01	0.53 ± 0.01	0.85 ± 0.01
<i>CF_V</i>	<b>0.89 ± 0.01</b>	0.67 ± 0.01	<b>0.70 ± 0.02</b>	<b>0.67 ± 0.02</b>	<b>0.55 ± 0.01</b>	<b>0.87 ± 0.01</b>
<i>CF</i>	0.89 ± 0.01	<b>0.68 ± 0.02</b>	0.69 ± 0.02	0.67 ± 0.03	0.55 ± 0.02	0.86 ± 0.01
ResNet						
<i>DICE_CE</i>	<b>0.88 ± 0.01</b>	0.62 ± 0.01	<b>0.68 ± 0.02</b>	<b>0.61 ± 0.02</b>	<b>0.51 ± 0.01</b>	<b>0.84 ± 0.01</b>
<i>CF_B</i>	0.88 ± 0.01	<b>0.63 ± 0.01</b>	0.67 ± 0.02	0.61 ± 0.02	0.51 ± 0.01	0.84 ± 0.02
<i>CF_V</i>	0.87 ± 0.01	0.61 ± 0.01	0.65 ± 0.02	0.59 ± 0.03	0.49 ± 0.01	0.84 ± 0.02
<i>CF</i>	0.86 ± 0.01	0.58 ± 0.01	0.60 ± 0.03	0.57 ± 0.02	0.46 ± 0.01	0.84 ± 0.01
UNet						
<i>DICE_CE</i>	0.88 ± 0.01	0.66 ± 0.01	0.69 ± 0.02	0.64 ± 0.02	0.53 ± 0.01	0.85 ± 0.01
<i>CF_B</i>	0.88 ± 0.01	0.65 ± 0.01	0.67 ± 0.02	0.64 ± 0.01	0.52 ± 0.01	0.85 ± 0.01
<i>CF_V</i>	0.88 ± 0.01	0.67 ± 0.01	0.69 ± 0.03	0.66 ± 0.02	0.54 ± 0.01	0.86 ± 0.01
<i>CF</i>	<b>0.88 ± 0.01</b>	<b>0.67 ± 0.02</b>	<b>0.69 ± 0.03</b>	<b>0.66 ± 0.02</b>	<b>0.54 ± 0.02</b>	<b>0.86 ± 0.01</b>

Table B.5: P-values from the paired Wilcoxon significance test for the different metrics from the training on the averaged fUS stacks (values lower than 0.05, highlighted in bold in the table, indicate that the observed differences are significant)

Comparison	F1Score	Accuracy	Precision	Recall	IoU	Specificity
(AttentionUNet-DICE_CE) vs (MultiresNet-DICE_CE)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.11
(AttentionUNet-DICE_CE) vs (UNet++-DICE_CE)	1	0.81	0.3	0.58	1	0.3
(AttentionUNet-DICE_CE) vs (ResNet-DICE_CE)	<b>0.016</b>	0.11	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-DICE_CE) vs (UNet-DICE_CE)	0.22	0.16	0.47	0.3	0.22	0.47
(AttentionUNet-DICE_CE) vs (AttentionUNet-CF_B)	0.94	1	1	1	0.81	1
(AttentionUNet-DICE_CE) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-DICE_CE) vs (UNet++-CF_B)	0.16	0.69	<b>0.016</b>	0.94	0.16	0.38
(AttentionUNet-DICE_CE) vs (ResNet-CF_B)	<b>0.031</b>	<b>0.047</b>	0.16	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>
(AttentionUNet-DICE_CE) vs (UNet-CF_B)	0.078	0.58	0.3	0.38	0.078	0.94
(AttentionUNet-DICE_CE) vs (AttentionUNet-CF_V)	0.16	0.3	0.47	0.38	0.16	0.47
(AttentionUNet-DICE_CE) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(AttentionUNet-DICE_CE) vs (UNet++-CF_V)	<b>0.047</b>	0.69	0.69	0.078	<b>0.047</b>	0.078
(AttentionUNet-DICE_CE) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.047</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	0.11
(AttentionUNet-DICE_CE) vs (UNet-CF_V)	0.94	0.81	0.38	0.58	1	0.16
(AttentionUNet-DICE_CE) vs (AttentionUNet-CF)	0.3	0.58	0.16	0.58	0.3	0.3
(AttentionUNet-DICE_CE) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-DICE_CE) vs (UNet++-CF)	0.47	0.69	0.58	0.22	0.47	0.22
(AttentionUNet-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.22
(AttentionUNet-DICE_CE) vs (UNet-CF)	0.94	0.94	0.47	0.58	0.94	0.3
(MultiresNet-DICE_CE) vs (UNet++-DICE_CE)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (ResNet-DICE_CE)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	0.3	<b>0.016</b>	0.22



(MultiresNet-DICE_CE) vs (UNet-DICE_CE)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.16
(MultiresNet-DICE_CE) vs (AttentionUNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(MultiresNet-DICE_CE) vs (MultiresNet-CF_B)	0.16	0.47	0.22	0.16	0.16	0.81
(MultiresNet-DICE_CE) vs (UNet++-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(MultiresNet-DICE_CE) vs (ResNet-CF_B)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	0.69	<b>0.016</b>	0.58
(MultiresNet-DICE_CE) vs (UNet-CF_B)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(MultiresNet-DICE_CE) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (MultiresNet-CF_V)	0.16	0.38	0.22	0.38	0.16	0.94
(MultiresNet-DICE_CE) vs (UNet++-CF_V)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-DICE_CE) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	0.38	<b>0.016</b>	1
(MultiresNet-DICE_CE) vs (UNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(MultiresNet-DICE_CE) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(MultiresNet-DICE_CE) vs (MultiresNet-CF)	0.3	0.22	0.58	0.47	0.3	0.81
(MultiresNet-DICE_CE) vs (UNet++-CF)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.047</b>
(MultiresNet-DICE_CE) vs (ResNet-CF)	0.94	0.078	<b>0.016</b>	<b>0.016</b>	0.078	0.47
(MultiresNet-DICE_CE) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(UNet++-DICE_CE) vs (ResNet-DICE_CE)	<b>0.016</b>	0.16	0.22	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-DICE_CE) vs (UNet-DICE_CE)	0.078	0.94	0.69	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>
(UNet++-DICE_CE) vs (AttentionUNet-CF_B)	0.47	0.22	0.22	1	0.47	0.69
(UNet++-DICE_CE) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-DICE_CE) vs (UNet++-CF_B)	0.22	1	0.69	0.38	0.16	0.58
(UNet++-DICE_CE) vs (ResNet-CF_B)	<b>0.016</b>	0.16	0.3	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-DICE_CE) vs (UNet-CF_B)	<b>0.031</b>	0.94	0.81	<b>0.031</b>	0.078	<b>0.031</b>
(UNet++-DICE_CE) vs (AttentionUNet-CF_V)	<b>0.047</b>	0.078	<b>0.047</b>	0.22	<b>0.031</b>	0.69
(UNet++-DICE_CE) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet++-DICE_CE) vs (UNet++-CF_V)	<b>0.016</b>	0.69	0.3	0.078	<b>0.016</b>	<b>0.047</b>
(UNet++-DICE_CE) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-DICE_CE) vs (UNet-CF_V)	0.81	0.38	0.58	0.81	0.69	0.94
(UNet++-DICE_CE) vs (AttentionUNet-CF)	0.16	0.22	0.078	0.81	0.11	0.47
(UNet++-DICE_CE) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-DICE_CE) vs (UNet++-CF)	0.47	0.47	0.16	0.58	0.3	0.47
(UNet++-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-DICE_CE) vs (UNet-CF)	0.69	0.3	0.58	0.81	0.94	1
(ResNet-DICE_CE) vs (UNet-DICE_CE)	<b>0.016</b>	0.3	0.11	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (AttentionUNet-CF_B)	<b>0.016</b>	0.078	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.38	<b>0.016</b>	0.58
(ResNet-DICE_CE) vs (UNet++-CF_B)	<b>0.016</b>	0.11	0.58	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (ResNet-CF_B)	0.69	0.47	1	0.3	0.58	0.22
(ResNet-DICE_CE) vs (UNet-CF_B)	<b>0.016</b>	0.3	0.3	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.81	<b>0.016</b>	0.38
(ResNet-DICE_CE) vs (UNet++-CF_V)	<b>0.016</b>	0.16	<b>0.047</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (ResNet-CF_V)	0.94	0.16	0.47	0.3	1	0.16
(ResNet-DICE_CE) vs (UNet-CF_V)	<b>0.016</b>	0.11	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (MultiresNet-CF)	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	1	<b>0.016</b>	0.22
(ResNet-DICE_CE) vs (UNet++-CF)	<b>0.016</b>	0.11	0.11	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	<b>0.031</b>
(ResNet-DICE_CE) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-DICE_CE) vs (AttentionUNet-CF_B)	0.22	0.22	1	0.22	0.11	0.22
(UNet-DICE_CE) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-DICE_CE) vs (UNet++-CF_B)	1	0.81	0.22	0.38	1	0.16
(UNet-DICE_CE) vs (ResNet-CF_B)	<b>0.016</b>	0.11	0.22	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(UNet-DICE_CE) vs (UNet-CF_B)	1	1	0.38	0.47	0.94	0.47
(UNet-DICE_CE) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.031</b>
(UNet-DICE_CE) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(UNet-DICE_CE) vs (UNet++-CF_V)	<b>0.016</b>	0.47	0.94	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-DICE_CE) vs (ResNet-CF_V)	<b>0.016</b>	0.078	<b>0.016</b>	0.11	<b>0.016</b>	0.16
(UNet-DICE_CE) vs (UNet-CF_V)	0.47	<b>0.047</b>	1	0.22	0.38	0.11
(UNet-DICE_CE) vs (AttentionUNet-CF)	<b>0.031</b>	0.22	0.22	0.16	<b>0.031</b>	0.16
(UNet-DICE_CE) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	0.16
(UNet-DICE_CE) vs (UNet++-CF)	<b>0.047</b>	0.38	0.58	<b>0.031</b>	<b>0.031</b>	<b>0.016</b>
(UNet-DICE_CE) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.11
(UNet-DICE_CE) vs (UNet-CF)	0.16	0.3	0.69	0.22	0.16	0.3
(AttentionUNet-CF_B) vs (MultiresNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF_B) vs (UNet++-CF_B)	0.47	0.38	<b>0.047</b>	0.94	0.38	0.81
(AttentionUNet-CF_B) vs (ResNet-CF_B)	<b>0.031</b>	0.11	0.16	<b>0.031</b>	<b>0.031</b>	<b>0.031</b>
(AttentionUNet-CF_B) vs (UNet-CF_B)	0.3	0.3	0.69	0.47	0.3	0.69
(AttentionUNet-CF_B) vs (AttentionUNet-CF_V)	0.22	0.3	0.16	0.58	0.22	0.47
(AttentionUNet-CF_B) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>

(AttentionUNet-CF_B) vs (UNet+-CF_V)	0.16	0.81	1	<b>0.047</b>	0.16	<b>0.031</b>
(AttentionUNet-CF_B) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.047</b>	<b>0.047</b>	<b>0.016</b>	<b>0.016</b>	0.078
(AttentionUNet-CF_B) vs (UNet-CF_V)	1	0.94	0.69	0.94	1	0.22
(AttentionUNet-CF_B) vs (AttentionUNet-CF)	0.22	0.58	0.16	0.81	0.22	0.58
(AttentionUNet-CF_B) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF_B) vs (UNet+-CF)	0.69	0.58	1	0.47	0.69	0.38
(AttentionUNet-CF_B) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(AttentionUNet-CF_B) vs (UNet-CF)	0.81	0.69	0.69	0.69	0.81	0.69
(MultiresNet-CF_B) vs (UNet+-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (ResNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	0.81
(MultiresNet-CF_B) vs (UNet-CF_B)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (MultiresNet-CF_V)	0.94	0.58	1	0.47	0.94	0.38
(MultiresNet-CF_B) vs (UNet+-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.3	<b>0.016</b>	0.81
(MultiresNet-CF_B) vs (UNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (MultiresNet-CF)	0.94	0.81	0.94	0.81	1	0.47
(MultiresNet-CF_B) vs (UNet+-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_B) vs (ResNet-CF)	0.078	<b>0.016</b>	<b>0.031</b>	1	<b>0.031</b>	0.3
(MultiresNet-CF_B) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet+-CF_B) vs (ResNet-CF_B)	<b>0.031</b>	0.078	0.38	<b>0.031</b>	<b>0.031</b>	<b>0.016</b>
(UNet+-CF_B) vs (UNet-CF_B)	0.94	0.94	0.3	0.69	0.94	0.58
(UNet+-CF_B) vs (AttentionUNet-CF_V)	<b>0.031</b>	<b>0.031</b>	0.078	0.16	<b>0.031</b>	0.58
(UNet+-CF_B) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet+-CF_B) vs (UNet+-CF_V)	<b>0.031</b>	0.81	0.22	<b>0.047</b>	<b>0.031</b>	0.11
(UNet+-CF_B) vs (ResNet-CF_V)	<b>0.031</b>	0.078	0.3	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>
(UNet+-CF_B) vs (UNet-CF_V)	0.58	0.38	0.16	0.58	0.47	0.69
(UNet+-CF_B) vs (AttentionUNet-CF)	<b>0.031</b>	0.16	<b>0.016</b>	0.47	<b>0.031</b>	0.69
(UNet+-CF_B) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet+-CF_B) vs (UNet+-CF)	0.3	0.3	0.22	0.38	0.3	0.47
(UNet+-CF_B) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(UNet+-CF_B) vs (UNet-CF)	0.38	0.38	0.47	0.47	0.3	0.94
(ResNet-CF_B) vs (UNet-CF_B)	<b>0.016</b>	0.81	0.16	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (AttentionUNet-CF_V)	<b>0.016</b>	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.38	<b>0.016</b>	0.94
(ResNet-CF_B) vs (UNet+-CF_V)	<b>0.016</b>	0.16	0.22	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (ResNet-CF_V)	0.47	0.22	0.3	0.69	0.38	0.58
(ResNet-CF_B) vs (UNet-CF_V)	<b>0.016</b>	<b>0.016</b>	0.22	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (AttentionUNet-CF)	<b>0.016</b>	0.078	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.3	<b>0.016</b>	0.38
(ResNet-CF_B) vs (UNet+-CF)	<b>0.016</b>	0.16	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(ResNet-CF_B) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	0.3
(ResNet-CF_B) vs (UNet-CF)	<b>0.016</b>	0.078	0.078	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-CF_B) vs (AttentionUNet-CF_V)	<b>0.016</b>	0.11	<b>0.047</b>	0.11	<b>0.016</b>	0.22
(UNet-CF_B) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(UNet-CF_B) vs (UNet+-CF_V)	<b>0.016</b>	0.58	0.3	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet-CF_B) vs (ResNet-CF_V)	<b>0.016</b>	0.11	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	0.078
(UNet-CF_B) vs (UNet-CF_V)	0.3	0.3	1	0.11	0.22	0.16
(UNet-CF_B) vs (AttentionUNet-CF)	<b>0.031</b>	0.3	0.11	0.16	<b>0.031</b>	0.58
(UNet-CF_B) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(UNet-CF_B) vs (UNet+-CF)	0.078	0.38	0.22	<b>0.047</b>	0.078	<b>0.047</b>
(UNet-CF_B) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet-CF_B) vs (UNet-CF)	0.16	0.58	0.58	0.3	0.22	0.22
(AttentionUNet-CF_V) vs (MultiresNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(AttentionUNet-CF_V) vs (UNet+-CF_V)	1	0.47	<b>0.047</b>	0.22	0.94	0.078
(AttentionUNet-CF_V) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF_V) vs (UNet-CF_V)	0.3	0.38	0.078	0.58	0.3	0.94
(AttentionUNet-CF_V) vs (AttentionUNet-CF)	0.47	0.69	0.81	0.58	0.47	0.47
(AttentionUNet-CF_V) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(AttentionUNet-CF_V) vs (UNet+-CF)	0.47	0.69	0.16	1	0.38	0.94
(AttentionUNet-CF_V) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF_V) vs (UNet-CF)	0.16	0.22	0.16	0.38	0.078	0.94
(MultiresNet-CF_V) vs (UNet+-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_V) vs (ResNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078	<b>0.016</b>	0.94
(MultiresNet-CF_V) vs (UNet-CF_V)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_V) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_V) vs (MultiresNet-CF)	0.81	0.58	0.58	0.94	0.94	0.58
(MultiresNet-CF_V) vs (UNet+-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(MultiresNet-CF_V) vs (ResNet-CF)	0.078	<b>0.016</b>	<b>0.031</b>	0.58	0.078	0.58

(MultiresNet-CF_V) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(UNet++-CF_V) vs (ResNet-CF_V)	<b>0.016</b>	0.11	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF_V) vs (UNet-CF_V)	0.11	0.94	1	0.16	0.11	0.22
(UNet++-CF_V) vs (AttentionUNet-CF)	0.38	0.94	0.22	0.16	0.38	0.078
(UNet++-CF_V) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF_V) vs (UNet++-CF)	0.58	1	0.3	0.38	0.58	0.38
(UNet++-CF_V) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF_V) vs (UNet-CF)	0.16	0.81	0.69	0.22	0.078	0.16
(ResNet-CF_V) vs (UNet-CF_V)	<b>0.016</b>	<b>0.047</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	0.078
(ResNet-CF_V) vs (AttentionUNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.078
(ResNet-CF_V) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	0.3	<b>0.016</b>	1
(ResNet-CF_V) vs (UNet++-CF)	<b>0.016</b>	0.078	<b>0.016</b>	<b>0.047</b>	<b>0.016</b>	0.078
(ResNet-CF_V) vs (ResNet-CF)	<b>0.016</b>	0.078	<b>0.031</b>	0.078	<b>0.016</b>	0.94
(ResNet-CF_V) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet-CF_V) vs (AttentionUNet-CF)	0.3	0.58	<b>0.016</b>	0.94	0.3	0.38
(UNet-CF_V) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet-CF_V) vs (UNet++-CF)	0.81	0.38	0.81	0.58	0.81	1
(UNet-CF_V) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>
(UNet-CF_V) vs (UNet-CF)	0.94	0.81	0.94	0.94	1	0.58
(AttentionUNet-CF) vs (MultiresNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(AttentionUNet-CF) vs (UNet++-CF)	1	1	0.47	0.81	1	0.3
(AttentionUNet-CF) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>
(AttentionUNet-CF) vs (UNet-CF)	0.22	0.11	0.22	0.94	0.22	0.94
(MultiresNet-CF) vs (UNet++-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.031</b>	<b>0.016</b>	<b>0.031</b>
(MultiresNet-CF) vs (ResNet-CF)	0.3	<b>0.016</b>	<b>0.047</b>	0.69	0.11	1
(MultiresNet-CF) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF) vs (ResNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>
(UNet++-CF) vs (UNet-CF)	0.58	0.69	0.47	0.47	0.58	0.58
(ResNet-CF) vs (UNet-CF)	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.016</b>	<b>0.047</b>