

Article

# FAS-UNet: A Novel FAS-Driven UNet to Learn Variational Image Segmentation

Hui Zhu <sup>1,2</sup>, Shi Shu <sup>1,3</sup> and Jianping Zhang <sup>1,4,\*</sup><sup>1</sup> The School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China<sup>2</sup> Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education, Xiangtan 411105, China<sup>3</sup> Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan 411105, China<sup>4</sup> Hunan National Applied Mathematics Center, Xiangtan 411105, China

\* Correspondence: jpzhang@xtu.edu.cn

**Abstract:** Solving variational image segmentation problems with hidden physics is often expensive and requires different algorithms and manually tuned model parameters. The deep learning methods based on the UNet structure have obtained outstanding performances in many different medical image segmentation tasks, but designing such networks requires many parameters and training data, which are not always available for practical problems. In this paper, inspired by the traditional multiphase convexity Mumford–Shah variational model and full approximation scheme (FAS) solving the nonlinear systems, we propose a novel variational-model-informed network (FAS-UNet), which exploits the model and algorithm priors to extract the multiscale features. The proposed model-informed network integrates image data and mathematical models and implements them through learning a few convolution kernels. Based on the variational theory and FAS algorithm, we first design a feature extraction sub-network (FAS-Solution module) to solve the model-driven nonlinear systems, where a skip-connection is employed to fuse the multiscale features. Secondly, we further design a convolutional block to fuse the extracted features from the previous stage, resulting in the final segmentation possibility. Experimental results on three different medical image segmentation tasks show that the proposed FAS-UNet is very competitive with other state-of-the-art methods in the qualitative, quantitative, and model complexity evaluations. Moreover, it may also be possible to train specialized network architectures that automatically satisfy some of the mathematical and physical laws in other image problems for better accuracy, faster training, and improved generalization.

**Keywords:** model-informed deep learning; interpretable network; variational image segmentation; full approximation scheme

**MSC:** 68T05; 68T07; 68U10; 94A08



**Citation:** Zhu, H.; Shu, S.; Zhang, J. FAS-UNet: A Novel FAS-Driven UNet to Learn Variational Image Segmentation. *Mathematics* **2022**, *10*, 4055. <https://doi.org/10.3390/math10214055>

Academic Editor: Jakub Nalepa

Received: 30 September 2022

Accepted: 24 October 2022

Published: 1 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Image segmentation is one of the most important problems in computer vision and also is a difficult problem in the medical imaging community [1–3]. It has been widely used in many medical image processing fields such as the identification of cardiovascular diseases [4], the measurement of bone and tissue [5], and the extraction of suspicious lesions to aid radiologists. Therefore, image segmentation has a vital role in promoting medical image analysis and applications as a powerful image processing tool [5,6].

Deep learning (DL) has achieved great success in the field of medical image segmentation [5,7,8]. One of the most important reasons is that the convolutional neural networks (CNNs) can effectively extract image features. Therefore, much work at present involves design a network architecture with strong feature extraction ability, and many well-known CNN architectures have been proposed such as UNet [9], V-Net [10], UNet++ [11], 3D UNet [12], Y-Net [13], Res-UNet [14], KiU-Net [15], DenseUNet [16], and nnU-Net [17].

More and more studies based on data-driven methods have been reported for medical image segmentation. Although UNet and its variants have achieved considerably impressive performance in many medical image segmentation datasets, they still suffer two limitations. One is that most of researchers have introduced more parameters to improve the performance of medical image segmentation, but have tended to ignore the technical branch of the model's memory and computational overhead, which makes it difficult to popularize the algorithm to industry applications [18]. The other disadvantage is that these variants only design many suitable architectures through the researcher's experience or experiments, but do not focus on the mathematical theoretical guidance of network architectures such as explainability, generalizability, etc., which limits the application of these models and the improvement of task-driven medical image segmentation methods [19,20].

Recently, many works on image recognition and image reconstruction have been focusing on the interpretability of the network architecture. Inspired by some mathematical viewpoints, many related unroll networks have been designed and successfully applied. He et al. [21] proposed the deep residual learning framework, which utilizes an identity map to facilitate training; it is well known that it is very similar to the iterative method solving ordinary differential equations (ODEs) and also achieves promising performance on image recognition. G. Larsson et al employed the fractal idea to design a self-similar FractalNet [22], also discovering that its architecture is similar to the Runge–Kutta (RK) scheme in numerical calculations. According to the nature of polynomials, Zhang et al. designed PolyNet [23] by improving ResNet to strengthen the expressive ability of the network, and Gomez et al. [24] proposed RevNet by using some ideas of the dynamic system. Chen et al. [25] analyzed the process of solving ODEs, then proposed Neural ODE, which further shows that mathematics and neural networks have a strong relationship. Meanwhile, He et al. designed a network architecture for the super-resolution task based on the forward Euler and RK methods of solving ODEs [18] and achieved good performance. Sun et al. [26] designed ADMM-Net through the alternating direction method to learn an image reconstruction problem. Inspired by a multigrid algorithm for solving inverse problems, He et al. [27] proposed a learnable classification network denoted as MgNet to extract image features  $u$ , which uses a few parameters to achieve good performance on the CIFAR datasets. Alt et al. [28] analyzed the behavior and mathematical foundations of UNet, and interpreted them as approximations of continuous nonlinear partial differential equations (PDEs) by using full approximation schemes (FASs). Experimental evaluations showed that the proposed architectures for the denoising and inpainting tasks save half of the trainable parameters and can thus outperform standard ones with the same model complexity.

Unfortunately, only a few studies based on model-driven techniques have been reported for the segmentation task. In this paper, we mainly focus on the explainable DL framework combining the advantages of the FAS and UNet for medical image segmentation.

### 1.1. Problem

H. Helmholtz proposed that the ill-posed problem of producing reliable perception from fuzzy signals can be solved through the process of “unconscious inference” (the **Helmholtz Hypothesis**) [29]. This theory implies that human vision is incomplete and that details are inferred by the unconscious mind to create a complete image. That is, our perception system can also integrate the fuzzy evidence received from the senses into the situation based on its own environmental model.

Let  $p(u|f; \alpha)$  be a probabilistic distribution for feature representations  $u$  of the source image  $f$ . The prior probability of  $u$  can be modeled as the multivariate normal distribution. In general,  $u$  can be extracted from a given image  $f$  by optimizing the maximum a posteriori (MAP) estimation as

$$\arg \max_u \log p(u|f; \alpha), \quad (1)$$

where  $\alpha$  is the environmental parameter in classical “unconscious inference” or the inverse problem, and this problem leads to the nonlinear system defined by

$$\mathcal{F}(u; \alpha) = b, \tag{2}$$

where the nonlinear operator  $\mathcal{F}(\cdot; \alpha)$  is employed to generate the image  $b$ , e.g.,  $b = \mathcal{A}^T f$  is a deconvoluted image of  $f$  in the image deblurring problem with a convolution operator  $\mathcal{A}$ .

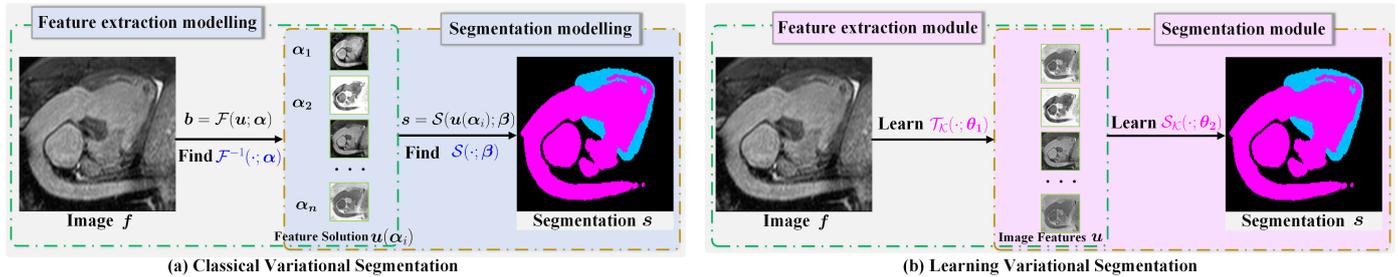
We consider that image segmentation refers to a composite process of feature extraction (6) and feature fusion segmentation. Here, the fusing process for feature  $u$  is defined by

$$s = \mathcal{S}(u; \beta), \tag{3}$$

where  $\mathcal{S}(\cdot; \beta)$  denotes a fusing segmentation with a fixed conscious parameter  $\beta$ , and  $s$  is the segmentation results or probability maps.

Such strongly interpretable segmentation models [30–32] are so general that, depending on the amount of well-predefined sparsity priors of the input image, they have the advantages of theoretical support and strong convergence. The total flowchart of classical variational segmentation can be summarized as shown in Figure 1a. However, they usually require expensive computations, but also have to face the problems of the selection of suitable regularizers  $\phi(\cdot)$  and model parameters  $(\alpha, \beta)$ . Consequently, some reconstructed results are unsatisfactory.

It is well known that the solution  $u$  usually has the multiscale property, so a natural idea is to exploit the multi-layer convolution and multigrid architecture, which can describe multiscale features to learn  $u$ . Based on the above facts, we propose a two-stage segmentation framework for learning feature  $u$  in Stage 1 and segmentation  $s$  in Stage 2, which is shown in Figure 1b.



**Figure 1.** Classical variational image segmentation and model-inspired learning method. (a) The first stage solves the nonlinear differential equations using the classical iterative method, and then, the second stage thresholds the smooth solution in the first stage to extract objects. (b) The first stage learns the solution mapping  $\mathcal{T}_K(f; \theta_1)$  by optimizing the convolution kernel  $\theta_1$  to extract image features. The second stage learns the feature fusion and segmentation thresholding parameter.

1.2. Contributions

In this work, we focus on analyzing the feature extraction inverse problem (2) and the feature fusion segmentation (3) to design an explainable deep learning network. It is well known that the unrolled iterations of the classical solution algorithm can be considered as the layers of a neural network, so we propose a novel FAS-driven UNet (FAS-UNet), which integrates image data and a multiscale algorithm for solving the nonlinear inverse problem (7). The major differences with our approach are that MgNet is not a U-shaped architecture and is only used for image classification, which leads to the output result not being able to be converted to the segmentation prediction of the input image. Besides, the proposed network was inspired by the traditional multiphase convexity Mumford–Shah variational model [30] and FAS algorithm for solving nonlinear systems [33], which exploits the model and algorithm priors’ information to extract the image features. Indeed, the goal of our work is to show that, under some assumptions about the operators, it is possible to interpret the smoothing operations of the FAS and image geometric extracting operations

of the variational model as the layers of a CNN, which in turn, provide fairly specialized network architectures that allow us to solve the standard nonlinear system (7) for a specific choice of the parameters involved.

Our main contributions are summarized as follows:

1. We propose a novel variational-model-informed two-stage image segmentation network (FAS-UNet), where an explainable and lightweight sub-network for feature extraction is designed by combining the traditional multiphase convexity Mumford–Shah variational model and FAS algorithm for solving nonlinear systems. To the best of our knowledge, it is the first unrolled architecture designed based on model and algorithm priors in the image segmentation community.
2. The proposed model-informed network integrates image data and mathematical models, and it provides a helpful viewpoint for designing the image segmentation network architecture.
3. The proposed architecture can be trained from additional model information obtained by enforcing some of the mathematical and physical laws for better accuracy, faster training, and improved generalization. Extensive experimental results show that it performs better than the other state-of-the-art methods.

The rest of the paper is organized as follows. The novel FAS-UNet framework for solving nonlinear inverse problems by analyzing variational segmentation theory and the FAS algorithm is proposed in Section 2. We show experimental results in Section 3. Finally, we conclude this work in Section 4.

## 2. Variational Segmentation via the CNN Framework

The goal of image segmentation is to partition a given image  $f : \Omega \rightarrow \mathbb{R}$  into  $r$  regions  $\{\Omega_i\}_{i=1}^r$  that contain distinct objects and satisfy  $\Omega_i \cap \Omega_j = \emptyset, j \neq i$ , and  $\bigcup_{i=1}^r \Omega_i = \Omega$ , where the image domain  $\Omega$  is a bounded and open subset of  $\mathbb{R}^2$ . Assume that  $\Gamma = \bigcup \partial\Omega_i$  is the union of boundaries of  $\Omega_i, |\Gamma|$ , denoting the arc length of curve  $\Gamma$ .

### 2.1. Multiphase Variational Image Segmentation

As mentioned, various ways of variational image segmentation have been proposed. Below, we review a few of them.

#### 2.1.1. Variational Image Segmentation

The Mumford–Shah (M-S) model is a well-known variational image segmentation method proposed by Mumford and Shah [34], which can be defined as follows:

$$\min_{u, \Gamma} \left\{ \tau_1 \int_{\Omega} (f - u)^2 dx + \tau_2 \int_{\Omega - \Gamma} |\nabla u|^2 dx + |\Gamma| \right\},$$

where  $\tau_1$  and  $\tau_2$  are the weight parameters. The first term requires that  $u : \Omega \rightarrow \mathbb{R}$  approximates  $f$ , the second term that  $u$  does not vary much on each  $\Omega_i$ , and the third term that the boundary  $\Gamma$  is as short as possible. This shows that  $u$  is a piecewise smooth approximation of  $f$ .

In particular, Chan and Vese considered the special case of the M-S model where the function  $u$  is chosen to be a piecewise constant function; thus, the minimization for two-phase segmentation is given as

$$\min_{\Gamma, c_1, c_2} \lambda_1 \int_{\text{inside}(\Gamma)} |f - c_1|^2 dx + \lambda_2 \int_{\text{outside}(\Gamma)} |f - c_2|^2 dx + |\Gamma|,$$

where  $c_1$  and  $c_2$  are the average image intensities inside and outside of boundary  $\Gamma$ , respectively, and  $\lambda_1$  and  $\lambda_2$  are the weight parameters.

Sometimes, the given image is degraded by noise and problem-related blur operator  $\mathcal{A}$ . Therefore, Cai et al. [30] extended the two-stage image segmentation strategy using a convex variant of the Mumford–Shah model as

$$\min_{u \in W^{1,2}(\Omega)} \int_{\Omega} \left( \kappa_1 (f - \mathcal{A}u)^2 + \kappa_2 |\nabla u|^2 + |\nabla u| \right) dx, \tag{4}$$

where  $\kappa_1$  and  $\kappa_2$  are positive parameters, and the existence and uniqueness of  $u$  were analyzed in their work.

We assume the image features  $\mathbf{u} = (u_1, \dots, u_d)^T : \Omega \rightarrow \mathbb{R}^d$ , where  $u_i : \Omega_i \rightarrow \mathbb{R}$  is a smooth mapping defined on the tissue or lesion  $\Omega_i$ . In this work, we extend the above model (4) to the multiphase case, which can deal with  $d$ -phase segmentation (multiple objects), which refers to a two-stage composite process of feature extraction (6) and feature fusion segmentation (3).

### 2.1.2. Feature Extraction

The first stage is to extract image features  $\mathbf{u}$  by maximizing a posterior probabilistic distribution (6) for feature representations  $\mathbf{u}$  of a given image  $f$  as

$$\begin{aligned} \arg \max_{\mathbf{u}} p(\mathbf{u}|f; \alpha) &= \arg \max_{\mathbf{u}} \log p(\mathbf{u}|f; \alpha) \\ &= \arg \max_{\mathbf{u}} \log \frac{p(f|\mathbf{u}; \alpha)p(\mathbf{u}; \alpha)}{p(f)} \\ &= \arg \max_{\mathbf{u}} \log p(f|\mathbf{u}; \alpha)p(\mathbf{u}; \alpha), \end{aligned} \tag{5}$$

where  $\alpha$  is the environmental parameter in classical “unconscious inference” or the inverse problem. Especially, the likelihood probability  $p(f|\mathbf{u}; \alpha)$  and the prior probability  $p(\mathbf{u}; \alpha)$  can be modeled as normal distributions, respectively, denoted by

$$p(f|\mathbf{u}; \alpha) \propto e^{-\frac{1}{2\sigma^2} \int_{\Omega} (\mathcal{A}\mathbf{u}-f)^2 d\Omega} = e^{-\gamma \int_{\Omega} (\mathcal{A}\mathbf{u}-f)^2 d\Omega}, \quad p(\mathbf{u}; \alpha) \propto e^{-\lambda \int_{\Omega} \phi(\nabla \mathbf{u}) d\Omega};$$

thus, the first stage is to find a smooth approximation  $\mathbf{u}$  by minimizing the multiphase generalizability (TS-MCMS) of (4), which can be rewritten as

$$\min_{\mathbf{u} \in W^{1,2}(\Omega)} \left\{ \int_{\Omega} (f - \mathcal{A}\mathbf{u})^2 dx + \mu \int_{\Omega} \phi(\nabla \mathbf{u}) dx \right\}, \tag{6}$$

where  $\mathcal{A} : \mathbb{R}^d \rightarrow \mathbb{R}$  is a convolutional blur operator,  $\phi(\nabla \mathbf{u}) = \nu |\nabla \mathbf{u}|^2 + |\nabla \mathbf{u}|$  is a geometric prior of  $\mathbf{u}$ , and  $\mu = \frac{\lambda}{\gamma}$ . Hence, this leads to the nonlinear system as

$$\mathcal{F}(\mathbf{u}; \alpha) := \mathcal{A}^T \mathcal{A}\mathbf{u} - \mu \nabla \cdot (\phi'(\nabla \mathbf{u})) = \mathbf{b}, \tag{7}$$

where  $\mathbf{b} = \mathcal{A}^T f$  and  $\alpha = (\mathcal{A}, \nabla, \mu, \nu)$ .

### 2.1.3. Feature Fusion Segmentation

Once the features  $\mathbf{u}$  are obtained, the segmentation is performed by fusing  $\mathbf{u}$  properly in the second stage; for example, many novel image segmentation methods [30–32] have been proposed based on thresholding the smooth solution  $\mathbf{u}$ . Then, the fusing process for feature  $\mathbf{u}$  is finished in (3).

The model-driven methods introduce prior knowledge regarding many desirable mathematical properties of the underlying anatomical structure, such as phase field theory,  $\Gamma$ -approximation, smoothness, and sparseness. The informed priors may help to render the segmentation method more robust and stable. However, these model-inspired methods generally solve the optimization problem in the image domain, while the numerical minimization method for the feature representations  $\mathbf{u}$  is very slow because the regularization of the TV-norm, the high dimensionality of  $\mathbf{u}$ , as well as the nonlinear relationship between

the images and the parameters pose significant computational challenges. Furthermore, it is challenging to introduce priors flexibly under different clinical scenes. These limitations make it hard for purely model-based segmentation to obtain the solutions efficiently and flexibly.

The goal of this work was to learn powerful solvers of (7) and (3) to aggregate a variety of mechanisms to address the medical image segmentation problem efficiently.

### 2.2. Proposed Learnable Framework of TS-MCMS Algorithm

We summarized the two-stage algorithm to formulate medical image segmentation based on the TS-MCMS model, inspired by the CNN architectures of unrolled iterations, and we propose a learnable framework with two CNN modules on multiscale feature spaces, FAS-UNet (see Figure 1b), aimed at learning the nonlinear inverse operators of (7) and (3) in the context of the variational inverse problem to segment a given image  $f$ .

It is already well known that the unrolled iterations of many classical algorithms can be considered as the layers of a neural network [22–26]. In this part, we are not interested in designing another approach for inferring the classes in MgNet [27], but rather, we aim at extracting the features of a given image  $f$ .

Inspired by the variational segmentation model (6), one of the key ideas in the proposed architecture is that we split our framework into a solution module  $\mathcal{T}_{\mathcal{K}}(f; \theta_1)$  and a feature fusion module  $\mathcal{S}_{\mathcal{K}}(\mathbf{u}; \theta_2)$ , where  $\mathcal{T}_{\mathcal{K}}(f; \theta_1)$  is the feature extraction part of the framework (in the multi-stage case) and  $\mathcal{S}_{\mathcal{K}}(\cdot; \theta_2)$  is the stage fusion part to be learned. Therefore, how to design the effective function maps  $\mathcal{T}_{\mathcal{K}}$  for approximately solving (7) and  $\mathcal{S}_{\mathcal{K}}$  for approximating (3) is an important problem.

This work applies a nonlinear multigrid method to design FAS-UNet for explainable medical image segmentation by learning the two following modules:

$$\begin{cases} \mathbf{u} = \mathcal{T}_{\mathcal{K}}(f; \theta_1) \\ \mathbf{s} = \mathcal{S}_{\mathcal{K}}(\mathbf{u}; \theta_2), \end{cases} \tag{8}$$

where  $f$  is an input image,  $\mathbf{u}$  is the feature maps, and  $\mathbf{s}$  is the prediction for the truth partitions, leading to the overall approximation function as

$$\mathbf{s} = \mathcal{S}_{\mathcal{K}}(\mathcal{T}_{\mathcal{K}}(f; \theta_1); \theta_2), \tag{9}$$

where  $\theta_1$  and  $\theta_2$  are parameters to be learned in the proposed explainable FAS-UNet architecture.

To understand the approximation ability of the proposed modules  $\mathcal{T}_{\mathcal{K}}(f; \theta_1)$  and  $\mathcal{S}_{\mathcal{K}}(\mathbf{u}; \theta_2)$  generated by the FAS-UNet architecture, we refer the readers to D. Zhou’s work [35], which answers an open question in CNN learning theory about how deep CNN can be used to approximate any continuous function to an arbitrary accuracy when the depth of the neural network is large enough.

### 2.3. FAS-Module for Feature Extraction

In this part, we discuss how the multigrid method can be used to solve nonlinear problems. The Helmholtz Hypothesis [29] demonstrates that the extracted features can also be represented by solving the equation:

$$\mathcal{F}(\mathbf{u}, \alpha) = \mathbf{b} := \mathcal{A}^T f, \tag{10}$$

subject to

$$\min_{\mathbf{u}} \|\mathcal{S}_{\mathcal{K}}(\mathbf{u}; \theta_2) - \mathbf{y}\|,$$

where  $\mathcal{F}$  denotes the transformation of combining feature  $\mathbf{u}$  with a deblurred image  $\mathbf{b} = \mathcal{A}^T f$ ,  $\mathbf{u}$  is the unknown features, and  $\mathbf{y}$  is the ground-truth of image  $f$ . Our starting point is the traditional FAS algorithm solving (10).

### 2.3.1. The Full Approximation Scheme

The multigrid method is usually used to solve nonlinear algebraic systems (10). For simplicity, the parameter  $\alpha$  in  $\mathcal{F}(\mathbf{u}, \alpha)$  is omitted when only the classical FAS algorithm is involved, i.e.,

$$\mathcal{F}(\mathbf{u}) = \mathbf{b}. \tag{11}$$

The multigrid ingredients including the error smoothing and the coarse grid correction ideas are not restricted to the linear situation, but can be immediately used for the nonlinear problem itself, which leads to the so-called FAS algorithm. The fundamental idea of the nonlinear multigrid is the same as in the linear case, and the FAS method can be recursively defined on the basis of a two-grid method. We start with the description of one fine-coarse cycle (finer grid layer  $\ell$  and coarser grid layer  $\ell + 1$ ) of the nonlinear two-grid method for solving (11). To proceed, let the fine grid equation be written as

$$\mathcal{F}^\ell(\mathbf{u}^\ell) = \mathbf{b}^\ell.$$

Firstly, we compute an approximation  $\bar{\mathbf{u}}^\ell := \mathbf{u}_m^\ell$  of the fine grid problem by applying  $m$  pre-smoothing steps to  $\mathbf{u}^\ell$  as follows

$$\begin{aligned} &\mathbf{u}_0^\ell = \mathbf{u}^\ell; \\ &\text{for } k = 1 : m \text{ do} \\ &\quad \mathbf{u}_k^\ell = \mathbf{u}_{k-1}^\ell + (\mathcal{F}^\ell)'(\mathbf{b}^\ell - \mathcal{F}^\ell(\mathbf{u}_{k-1}^\ell)), \\ &\text{end for} \\ &\bar{\mathbf{u}}^\ell = \mathbf{u}_m^\ell, \end{aligned}$$

which can be obtained via solving the least-squares problem defined by

$$\min_{\mathbf{u}} \left\{ \mathcal{E}(\mathbf{u}) := \frac{1}{2} \|\mathcal{F}^\ell(\mathbf{u}) - \mathbf{b}^\ell\|^2 \right\}.$$

Secondly, the errors to the solution have to be smoothed such that they can be approximated on a coarser grid. Then, the defect  $\mathbf{r}^\ell = \mathbf{b}^\ell - \mathcal{F}^\ell(\bar{\mathbf{u}}^\ell)$  is computed, and an analog of the linear defect equation is transferred to the coarse grid, which is defined by

$$\mathcal{F}^{\ell+1}(\mathbf{u}^{\ell+1}) = \mathbf{b}^{\ell+1} := I_\ell^{\ell+1} \mathbf{r}^\ell + \mathcal{F}^{\ell+1}(I_\ell^{\ell+1} \bar{\mathbf{u}}^\ell). \tag{12}$$

The coarse grid corrections are interpolated back to the fine grid by

$$\bar{\mathbf{u}}^\ell \leftarrow \bar{\mathbf{u}}^\ell + I_{\ell+1}^\ell \left[ \mathbf{u}^{\ell+1} - I_\ell^{\ell+1} \bar{\mathbf{u}}^\ell \right], \tag{13}$$

where  $\mathbf{u}^{\ell+1}$  is a solution of the coarse grid equations, then the errors are finally post-smoothed by

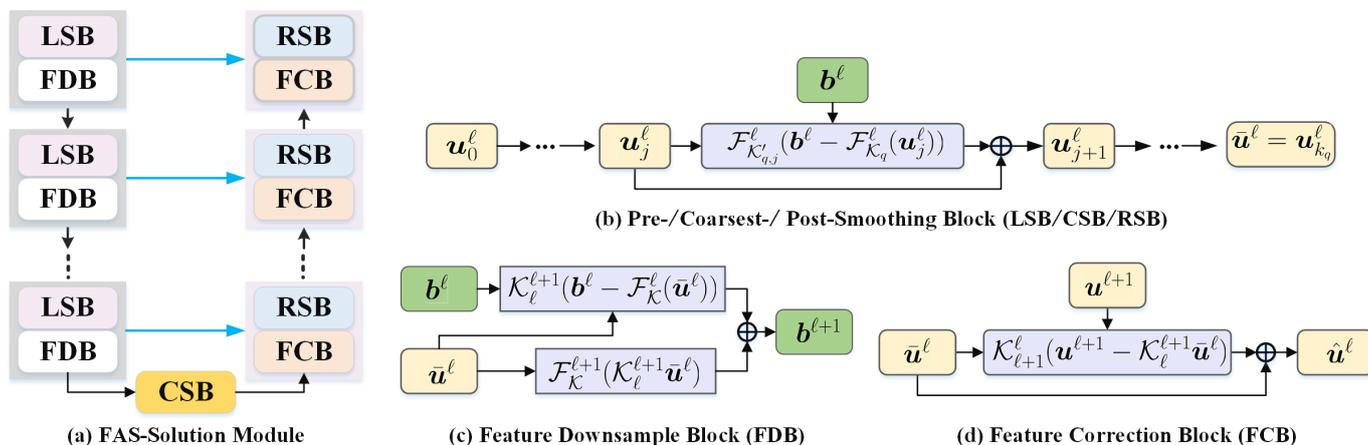
$$\begin{aligned} &\mathbf{u}_0^\ell = \bar{\mathbf{u}}^\ell; \\ &\text{for } j = 1 : m \text{ do} \\ &\quad \mathbf{u}_j^\ell = \mathbf{u}_{j-1}^\ell + (\mathcal{F}^\ell)'(\mathbf{b}^\ell - \mathcal{F}^\ell(\mathbf{u}_{j-1}^\ell)), \\ &\text{end for} \\ &\hat{\mathbf{u}}^\ell = \mathbf{u}_m^\ell. \end{aligned}$$

This means that, once the solution of the fine grid problem is obtained, the coarse grid correction does not introduce any changes via interpolation. We regard this property as an essential one, and in our derivation of the coarse grid optimization problem, we make sure that it is satisfied.

### 2.3.2. FAS-Solution Module—A Learnable Architecture for FAS Solution

In this part, we unroll the multiscale correction process of the multigrid method and design a series of deep FAS-Solution modules to propagate the features of input image  $f$ . Figure 2 demonstrates the cascade of all ingredients at each FAS iteration of our propagative network.

We now consider a decomposition of the image representation into partial sums, which correspond to the multiscale feature sequence, having the following idea in mind. To learn features that are invariant to noise and uninformative intensity variations, we propose a generative feature module  $u = \mathcal{T}_{\mathcal{K}}(f; \theta_1)$  allowing for a significant reduction of the number of parameters involved, which involves a learnable FAS update for solving the nonlinear system (10). Next, we analyze its three key components as follows.



**Figure 2.** The overall flowchart of the proposed feature extraction module (FAS-Solution module) with the multigrid architecture. It consists of three major ingredients, i.e., pre-/coarsest-/post-smoothing blocks (LSB/CSB/RSB), feature downsample block (FDB), and feature correction block (FCB).

**Pre-/coarsest-/post-smoothing block:** Error smoothing is one of the two basic principles of the FAS approach. At each pre-smoothing (or coarsest-processing or post-processing) step, to establish an efficient error correction and reduce the computational costs, we propose to generate the error-based iterative scheme. The main motivation of the learnable pre-/coarsest-/post-smoothing blocks (LSB/CSB/RSB) is to provide another way to robustly resolve the ambiguity of the feasible solutions. Therefore, we further unroll the Newton update process for calculating the approximate solution of the feature maps, then design a series of deep pre-/coarsest-/post-smoothing blocks as

$$\begin{aligned}
 & \text{for } j = 0 : k_q - 1 \text{ do} \\
 & \quad u_{j+1}^l = u_j^l + \mathcal{M}(u_j^l; \mathcal{K}_{q,\ell}, \mathcal{K}'_{q,\ell,j}, b^l) \\
 & \quad \quad = u_j^l + \mathcal{F}_{\mathcal{K}'_{q,j}}^l(b^l - \mathcal{F}_{\mathcal{K}_q}(u_j^l)), \\
 & \text{end for} \\
 & \bar{u}^l = u_{k_q}^l;
 \end{aligned}$$

where the residual network block  $\mathcal{M}(u^l; \mathcal{K}, \mathcal{K}', b^l)$  is a trainable feature correction network; here,  $\ell = 1, \dots, \mathcal{L}$  with  $\mathcal{L}$ -grid cycles. The above deep smoothing series denotes the pre-smoothing block when  $q := l$ , the  $\mathcal{L}^{th}$  coarsest-smoothing when  $q := m$ , and the post-smoothing when  $q := r$ .  $\mathcal{F}_{\mathcal{K}'_{q,j}}^l$  (or  $\mathcal{F}_{\mathcal{K}_q}^l$ , which means that the convolution  $\mathcal{K}_{q,\ell}$  will share the same weights in the overall pre-/post-precessing smoothing steps of the  $\ell^{th}$  grid cycle) represents operations consisting of three main components, including convolution  $\mathcal{K}'_{q,\ell,j}$  (or  $\mathcal{K}_{q,\ell}$ ) with  $p$  filters, ReLU function  $\varphi(\cdot)$ , and batch normalization  $\psi(\cdot)$ , such that

$\mathcal{F}_{\mathcal{K}'_{q,j}}^\ell(\cdot) := \psi(\varphi(\mathcal{K}'_{q,\ell,j}(\cdot)))$ . Especially,  $\mathbf{b}^0 := \mathcal{K}^0 f$  is an initial feature in the finest grid, which is obtained by learning the convolution  $\mathcal{K}^0$  with  $p$  filters.

**Feature downsample block:** The choice of restriction and interpolation operators  $I_\ell^{\ell+1}$  and  $I_{\ell+1}^\ell$  in the FAS algorithm, for the intergrid transfer of grid functions, is closely related to the choice of the coarse strategy. Here, we design the learnable convolution for transfer operators, i.e., the grid transfers between the finer grid  $\ell$  and the coarser grid  $\ell + 1$ .

The low-frequency components represent meaningful image features on a coarse grid  $\ell + 1$ , whereas the high-frequency components do not because they are not “visible” on the coarse grid, which means that the frequency information on the coarse grid can be extracted from the right-side term defined by

$$\mathbf{b}^{\ell+1} = \mathcal{K}_\ell^{\ell+1}(\mathbf{b}^\ell - \mathcal{F}_{\mathcal{K}}^\ell(\bar{\mathbf{u}}^\ell)) + \mathcal{F}_{\mathcal{K}}^{\ell+1}(\mathcal{K}_\ell^{\ell+1}\bar{\mathbf{u}}^\ell), \tag{14}$$

where  $\mathbf{b}^\ell$  and  $\bar{\mathbf{u}}^\ell$  are the inputs of the downsample block and  $\mathbf{b}^{\ell+1}$  is the output of the downsampling module in the feature space; here,  $\ell = 1, \dots, \mathcal{L} - 1$  with  $\mathcal{L}$ -grid cycles. Similar to  $\mathcal{F}_{\mathcal{K}_j}^\ell$ ,  $\mathcal{K}_\ell^{\ell+1}$  is a learnable downsample operation that would be used to approximate the restriction function  $I_\ell^{\ell+1}$  in (12) or (13), such as convolution with a stride of 2 and  $p$  filters.  $\mathcal{F}_{\mathcal{K}}^\ell$  and  $\mathcal{F}_{\mathcal{K}}^{\ell+1}$  are the nonlinear convolutional blocks in the fine and coarse layers, respectively. In general,  $\mathcal{F}_{\mathcal{K}}^\ell$  denotes the operator consisting of three main components, including convolution  $\mathcal{K}_\ell$  with  $p$  filters, ReLU function  $\varphi(\cdot)$ , and batch normalization  $\psi(\cdot)$ , such that  $\mathcal{F}_{\mathcal{K}}^\ell(\cdot) := \psi(\varphi(\mathcal{K}_\ell(\cdot)))$ . Note that  $\mathbf{b}^\ell - \mathcal{F}_{\mathcal{K}}^\ell(\bar{\mathbf{u}}^\ell)$  is equivalent to the residual of the images in the fine layer, then  $\mathcal{F}_{\mathcal{K}}^{\ell+1}(\mathcal{K}_\ell^{\ell+1}\bar{\mathbf{u}}^\ell)$  is added to reduce the loss of image information compared with directly pooling the image. The feature downsample block (FDB) architecture is shown in Figure 2b.

**Feature correction block:** The purpose of the feature correction block (FCB) is to take the detailed information extracted from the coarser grid into account and help to compensate the encoded features  $\bar{\mathbf{u}}^\ell$ . The coarse grid corrections are interpolated back to the fine grid, i.e.,

$$\hat{\mathbf{u}}^\ell \leftarrow \bar{\mathbf{u}}^\ell + \mathcal{K}_{\ell+1}^\ell \left[ \mathbf{u}^{\ell+1} - \mathcal{K}_\ell^{\ell+1}\bar{\mathbf{u}}^\ell \right], \tag{15}$$

where  $\mathcal{K}_{\ell+1}^\ell$  is a learnable upsampling operation that would be used to approximate the interpolation function  $I_{\ell+1}^\ell$  in (13), such as the transposed convolution with a stride of 2 and  $p$  filters; here,  $\ell = 1, \dots, \mathcal{L} - 1$  with  $\mathcal{L}$ -grid cycles. Obviously,  $\mathbf{e}^{\ell+1} = \mathbf{u}^{\ell+1} - \mathcal{K}_\ell^{\ell+1}\bar{\mathbf{u}}^\ell$  is the residual features on the coarse grid. Compared to directly upsampling  $\mathbf{u}^{\ell+1}$ , the transposed convolution  $\mathcal{K}_{\ell+1}^\ell \mathbf{e}^{\ell+1}$  of the residual feature maps  $\mathbf{e}^{\ell+1}$  is used as the error corrections to update the fine grid approximation  $\hat{\mathbf{u}}^\ell$ , which will compensate the information of feature maps  $\bar{\mathbf{u}}^\ell$ . Such a transposed convolution could learn a self-adaptive mapping to restore features with more detailed information.

Based on these designs for nonlinear operator  $\mathcal{F}_{\mathcal{K}}^\ell$  and two grid transfer convolutions  $\mathcal{K}_\ell^{\ell+1}$  and  $\mathcal{K}_{\ell+1}^\ell$  with  $p$  filters, we aimed to approximate the feature solution of (6) by learning these feature extraction parameters as

$$\boldsymbol{\theta}_1 = \left\{ \left( \mathcal{K}_{\ell+1}^\ell, \mathcal{K}_\ell^{\ell+1}, (\mathcal{K}_{q,\ell}), (\mathcal{K}'_{q,\ell,j})_{j=1}^{k_q}, \mathcal{K}_\ell \right)_{\ell=1}^{\mathcal{L}-1}, \mathcal{K}^0, (\mathcal{K}_{m,\mathcal{L}}), (\mathcal{K}'_{m,\mathcal{L},j})_{j=1}^{k_m} \mid q \in \{l, r\} \right\} \tag{16}$$

in  $\mathbf{u} = \mathcal{T}_{\mathcal{K}}(f; \boldsymbol{\theta}_1)$ , thus further improving image segmentation.

#### 2.4. Learning Feature Fusion Segmentation

It is well known that, in the segmentation task, each pixel is labeled as either 0 or 1 so that organ pixels can be accurately identified within the tight bounding box. In the second stage of the two-stage multiphase variational image segmentation (6), the traditional method is that users manually set one or more thresholds according to their professional prior, with all pixels in the same object sharing the threshold, and then, filter the feature to

obtain the segmentation result. Another method is to obtain the final segmentation result by  $k$ -means clustering (the number of categories is given artificially, and the initial clustering center is adjusted continuously during the clustering process) [30]. This approach leads to a large amount of computation (recalculation of the metrics for each iteration) and unstable segmentation results (only considering the relationship between pixels and centers, not the relationship between pixels).

The second key component of our proposed FAS-UNet framework is how to design the segmentation module  $\mathcal{S}_{\mathcal{K}}(\mathbf{u}; \theta_2)$  to compute segmented mask  $s$ . However, the fusion segmentation module takes a batch of multiscaled features from the FAS module as the input and outputs the mass segmentation masks. Finally, the pixel segmentation computes the mapping from smaller-scale possibility predictions to binary masks.

Based on this idea, the feature fusion segmentation module is constructed, which comprises a convolutional operation and an activation function. Intuitively, the module  $\mathcal{T}_{\mathcal{K}}(f; \theta_1)$  of the feature extraction based on deep learning obtains the multi-channel feature maps  $\mathbf{u}$  (much larger than the number of categories) in the first stage. Then, a shallow convolutional network is constructed to learn the parameters  $\mathcal{K}_p$  corresponding to the mapping  $\rho(\mathcal{K}_p(\cdot)) : \mathbb{R}^p \rightarrow \mathbb{R}^c$  from the feature maps  $\mathbf{u}$  to the segmentation probability maps through *softmax* function  $\rho(\cdot)$ , which improves the traditional practice and has better generalization.

Based on these designs for channel transfer convolution  $\mathcal{K}_p$  with  $c$  filters ( $c$  is the number of segmentation categories), we aimed to approximate the final multiphase segmentation probability maps  $\mathcal{S}(\cdot; \beta)$  of (6) by learning these fusion parameters as

$$\theta_2 = \{\mathcal{K}_p\} \tag{17}$$

in  $s = \mathcal{S}_{\mathcal{K}}(\mathbf{u}; \theta_2)$ , thus further refining the segmentation mask.

### 2.5. Loss Function

The proposed FAS-UNet architecture can be rewritten as

$$s = \mathcal{S}_{\mathcal{K}}(\mathcal{T}_{\mathcal{K}}(f; \theta_1); \theta_2),$$

which requires the loss function  $\mathcal{L}(\theta; \mathcal{D}_{\text{train}})$  to optimize the model parameters  $\theta := \{\theta_1, \theta_2\}$ . It can measure the error between the prediction and labels, and the gradients of the weights in the loss function can be back-propagated to the previous layers in order to update the model weights.

To proceed, we considered the training data  $\mathcal{D}_{\text{train}} = \{(f_i, y_i)\}_{i=1}^n$  from a set of classes  $\mathcal{C}_{\text{train}} = \{0, \dots, c-1\}^d$  used for training a pixel classifier, where  $f_i$  is an image sample,  $y_i \in \mathcal{C}_{\text{train}}$  is the corresponding label,  $c$  is the number of object categories segmented in the datasets,  $d$  indicates the number of image pixels, and  $n$  denotes the number of training samples. We employed the cross-entropy as the loss function, leading to the optimization problem as follows:

$$\min_{\theta} \left\{ \mathcal{L}(\theta; \mathcal{D}_{\text{train}}) := \sum_{i=1, j=1}^{n, d} \left( \bar{s}_i^{(j)} \cdot \log(s_i^{(j)}) + (1 - \bar{s}_i^{(j)}) \cdot \log(1 - s_i^{(j)}) \right) \right\}, \tag{18}$$

where  $s_i^{(j)}$  denotes the predicted probabilistic vector of the  $j^{\text{th}}$  pixel in the  $i^{\text{th}}$  sample and  $\bar{s}_i^{(j)}$  corresponds to the one-hot-encoded label of the ground-truth  $y_i^{(j)}$  at the  $j^{\text{th}}$  pixel in the  $i^{\text{th}}$  sample.  $s_i^{(j)} \in \mathbb{R}^c$ . Finally, the predicted class of the  $j^{\text{th}}$  pixel of the  $i^{\text{th}}$  image would be given by

$$y_i^{(j)} = \arg \max_k \{s_i^{(j)}(1), \dots, s_i^{(j)}(k), \dots, s_i^{(j)}(c)\},$$

where  $s_i^{(j)}(k) \in \mathcal{C}_{\text{train}}$ .

### 3. Datasets and Experiments

In this section, we first introduce the evaluation metrics for medical image segmentation and also describe the datasets and the experimental settings that we used for 2D CT image segmentation and 3D medical volumetric segmentation. Next, we analyzed the sensitivity of 2D FAS-UNet to each hyperparameter configuration by a series of experiments. Finally, we evaluated the effectiveness of the proposed 2D/3D FAS-UNet through comparative experiments.

#### 3.1. Evaluation Metrics

There are many metrics to quantitatively evaluate segmentation accuracy, each of which focuses on different aspects. In this work, we employed the average Dice similarity coefficient (a-DSC), average precision (a-Preci), and average symmetric surface distance (a-SSD), which are widely used in the segmentation task as evaluation metrics to evaluate the performance of the model. The a-DSC/a-Preci/a-SSD are calculated by averaging the DSC/Preci/SSD of each category [36].

The Dice score is the most-used metric in validating medical image segmentation, also called the DSC score [37], defined by

$$\text{DSC}(S, Y) = 2 \times \frac{|S \cap Y|}{|S| + |Y|},$$

where  $S$  and  $Y$  denote the automatically segmentation set of images and the manually annotated ground-truth, respectively.  $|\cdot|$  denotes the measure of a set. The above formulas compute the overlap between the prediction and the ground-truth, to evaluate the overall effect of the segmentation results. However, it is fairly insensitive to the precise boundary of the segmented regions. Precision effectively describes the purity of the prediction relative to the ground-truth or measures the number of those pixels actually having a matching ground-truth annotation by calculating

$$\text{Preci} = \frac{TP}{TP + FP},$$

where a true positive (TP) is observed when a prediction–target mask pair has a score that exceeds some predefined threshold and a false positive (FP) indicates that a predicted object mask has no associated ground-truth. The SSD value between two finite point sets  $S$  and  $Y$  is defined as follows:

$$\text{SSD}(\partial S, \partial Y) = \frac{\sum_{s \in \partial S} d(s, \partial Y) + \sum_{y \in \partial Y} d(y, \partial S)}{|\partial S| + |\partial Y|},$$

where  $d(v, X) = \min_{x \in X} \|v - x\|$  denotes the minimum Euclidean distance from point  $v$  to all points of  $X$ .

#### 3.2. Datasets and Experimental Setup

We evaluated the proposed method and other state-of-the-art methods on the 2D SegTHOR datasets [38], 3D HVSMR-2016 datasets [39], and 3D CHAOS-CT datasets [40,41], respectively. We introduce their details and data processing methods as follows.

##### 3.2.1. Data Preparation

For the SegTHOR datasets, in order to reduce the GPU memory cost and reduce the image noise, we first split the original 3D data into many 2D images along the Z-axis. Secondly, we used  $f = I[96 : 400, 172 : 396]$  as the input image, where  $I$  denotes the original 2D slice. We removed the slices of the pure black ground-truth when it was in training on the datasets.

For the 3D HVSMR-2016 datasets, we directly used the sliding window cropping method with strides of  $64 \times 64 \times 32$  to crop the volumes. In general, before cropping the 3D whole-volume into several overlapping sub-volumes of size  $128 \times 128 \times 64$ , a  $(32, 32, 16)$ -voxel padding with zero filling was first added to each direction of the 3D whole-volume. Then, after these operations, all remaining sub-volumes whose sizes were smaller than  $128 \times 128 \times 64$  were resized to  $128 \times 128 \times 64$  with zero-filling, and the intensity values of all patches were in  $[0, 4808]$ .

For the 3D CHAOS-CT datasets, which were used as for the liver segmentation experiments, we first cropped the volumes in the  $x, y$  directions to obtain an ROI with a size of  $380 \times 440 \times z$  and then used the above sliding window cropping method to crop out several 3D sub-volumes, where those intensity values were in  $[-1200, 1096]$ .

Although the noise problem can be improved by data pre-processing, our aim was not to pursue the best performance of the network on these datasets; we compared the performance of each method under fairer conditions. Using some data pre-processing techniques may be particularly beneficial to some methods, while at the same time, they may degrade the performance of others, so we did not use more complex data pre-processing techniques.

### 3.2.2. Experimental Configurations

We used mini-batch stochastic gradient descent (SGD) to optimize the proposed model, in which the initial learning rate, momentum parameter, and weight decay parameter were set to 0.01, 0.99, and  $10^{-4}$ , respectively. We set the batch size as 16, 4, and 4 for the 2D SegTHOR datasets, 3D HVSMR-2016 datasets, and 3D CHAOS-CT datasets, respectively. The maximum epochs of the three datasets were set to 150, 150, and 300, respectively. We used also the decay strategy to update the learning rate. The network initialization method was defined as Kaiming initialization, and the activation function was set as ReLU. The numbers of grid cycles for 2D and 3D FAS-UNet were  $\mathcal{L} = 5$  and  $\mathcal{L} = 4$ , respectively. The kernel sizes of the 2D and 3D networks were set to  $3 \times 3$  and  $3 \times 3 \times 3$  as the defaults, respectively. We did not use the weight-sharing scheme for the convolution  $\mathcal{K}'_{q,\ell,j}$  (within the outer-level nonlinear operator  $\mathcal{F}_{\mathcal{K}'_{q,\ell,j}}$ ) within one smoothing block. Table 1 shows the details of the 2D FAS-UNet framework, and 3D FAS-UNet has a similar architecture, except for replacing the 2D convolution with a 3D convolution.

### 3.2.3. Parameter Complexity

To compute the number of parameters of the proposed model, we first denote the number of parameters of 2D convolution kernel  $\mathcal{K}_{2d}$  with the shape  $p \times p \times k_c \times k_c$  as follows:

$$\eta(\mathcal{K}_{2d}) = p^2 N = p^2 (k_c)^2.$$

Similarly, the number of parameters of 3D convolution  $\mathcal{K}_{3d}$  with the shape  $rp \times rp \times k_c \times k_c \times k_c$  is defined as

$$\eta(\mathcal{K}_{3d}) = r^2 p^2 (k_c)^3.$$

Especially if the channel ratio satisfies that  $r < \frac{\sqrt{3}}{k_c}$ , one has  $\eta(\mathcal{K}_{2d}) > \eta(\mathcal{K}_{3d})$ . Thus, the number of parameters of convolution kernel set  $\theta_1$  in the proposed FAS-UNet can be computed by

$$\begin{aligned} \eta(\theta_1) &= cp(k_c)^2 + \left( (k_m + 1) + (\mathcal{L} - 1)(k_l + k_r + 5) \right) \eta(\mathcal{K}) \\ &\approx ((k_m + 1) + (\mathcal{L} - 1)(k_l + k_r + 5)) \eta(\mathcal{K}), \end{aligned}$$

where  $c$  is the channel number of the input image and  $\eta(\mathcal{K})$  is the number of parameters of each 2D or 3D convolution  $\mathcal{K}$ .

If setting  $k_c = 3, p = 64, \{k_l, k_m, k_r\} = \{3, 7, 4\}$  and  $\mathcal{L} = 5$ , one has  $\eta(\mathcal{K}_{2d}) = 36,864$ ; hence, 2D FAS-UNet has approximately  $\eta(\theta_1) = 56\eta(\mathcal{K}_{2d}) = 2,064,384$  parameters. In addition, one also has  $\eta(\mathcal{K}_{3d}) = 27,648$  when  $k_c = 3, p = 32$ . If setting  $\{k_l, k_m, k_r\} = \{3, 5, 2\}$

and  $\mathcal{L} = 4$ , thus 3D FAS-UNet has approximately  $\eta(\theta_1) = 36\eta(\mathcal{K}_{3d}) = 995,328$  parameters. Here, we did not compute  $\eta(\theta_2)$ , where a small amount of parameters are involved.

Our experiments were implemented on the PyTorch framework and two NVIDIA Geforce RTX 2080Ti GPUs with 11GB memory. For each quantitative result in the experiments, we repeated the experiment twice and chose the best one to compute the mean/std. Note that we used the same pipeline for all these experiments of each dataset for a fair comparison. The networks under comparison were trained from scratch.

**Table 1.** A standard configuration of the proposed 2D FAS-UNet.  $p \times H \times W$  denotes the number of channels and the height and width of the feature maps, respectively. “conv3, s = 1 or 2” indicates the convolution with a kernel size of  $3 \times 3$  and the stride of convolution being 1 or 2, respectively. ReLU and BN are the ReLU activation function and batch normalization.

Name	2D Operation	Output Size	Name	2D Operation	Output Size
<b>Input</b>	$f, \text{ initialization } u_0^1$	$p \times H \times W$			
LSB1	$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_l$	$p \times H \times W$	RSB1	$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_r$	$p \times H \times W$
FDB1	conv3, s = 2 for $f$ conv3, s = 2 for $u$	$p \times \frac{H}{2} \times \frac{W}{2}$	FCB1	deconv3, s = 2	$p \times H \times W$
LSB2	$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_l$	$p \times \frac{H}{2} \times \frac{W}{2}$	RSB2	$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_r$	$p \times \frac{H}{2} \times \frac{W}{2}$
FDB2	conv3, s = 2 for $f$ conv3, s = 2 for $u$	$p \times \frac{H}{4} \times \frac{W}{4}$	FCB2	deconv3, s = 2	$p \times \frac{H}{2} \times \frac{W}{2}$
LSB3	$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_l$	$p \times \frac{H}{4} \times \frac{W}{4}$	RSB3	$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_r$	$p \times \frac{H}{4} \times \frac{W}{4}$
FDB3	conv3, s = 2 for $f$ conv3, s = 2 for $u$	$p \times \frac{H}{8} \times \frac{W}{8}$	FCB3	deconv3, s = 2	$p \times \frac{H}{4} \times \frac{W}{4}$
LSB4	$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_l$	$p \times \frac{H}{8} \times \frac{W}{8}$	RSB4	$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_r$	$p \times \frac{H}{8} \times \frac{W}{8}$
FDB4	conv3, s = 2 for $f$ conv3, s = 2 for $u$	$p \times \frac{H}{16} \times \frac{W}{16}$	FCB4	deconv3, s = 2	$p \times \frac{H}{8} \times \frac{W}{8}$
CSB		$\begin{pmatrix} \text{conv3, s = 1} \\ \text{ReLU + BN} \\ \text{conv3, s = 1} \\ \text{ReLU + BN} \end{pmatrix} \times k_m$			$p \times \frac{H}{16} \times \frac{W}{16}$

### 3.3. Ablation Studies

We conducted four groups of ablation studies on the 2D SegTHOR datasets to optimize the hyperparameter configurations of the proposed framework.

#### 3.3.1. Blocks’ Sensitivity

Firstly, we assessed the effect of smoothing block configurations  $\{k_l, k_m, k_r\}$ , where  $\{k_l, k_m, k_r\}$  denote the  $k_l, k_m$ , and  $k_r$  smoothing iterations in the LSB, CSB, and RSB, respectively. Here, we first fixed the channel configuration with  $p = 32$  as the default. Table 2 shows a quantitative results of different block parameter sets, and we observed from the pre-smoothing experiments (fixing  $k_m$  and  $k_r$ , varying  $k_l$ ) that 2D FAS-UNet with  $k_l = 3$

iterations achieved the best a-DSC score and precision of 85.60% and 85.99%, respectively. For the post-smoothing (fixing  $k_l$  and  $k_m$ , varying  $k_r$ ), we saw that the model with  $k_r = 4$  iterations achieved the highest values of a-SSD and precision, the a-DSC score being slightly lower than the model with the block configuration  $\{3, 7, 5\}$  by 0.04%. To balance the prediction performance and computational costs, we set the block configuration as  $\{3, 7, 4\}$  in all 2D experiments.

**Table 2.** Quantitative assessment with the a-DSC, a-SSD, and a-Preci values of different pre-/coarest-/post-smoothing iteration numbers  $\{k_l, k_m, k_r\}$  using the proposed 2D FAS-UNet framework on the 2D SegTHOR validation datasets with four organs: esophagus, heart, trachea, and aorta. “Mean” denotes an average score segmenting all organs. The best and second places are highlighted in bold font and underline, respectively.

Number of Smoothing Steps		Fixing $k_m$ and $k_r$ , Varying $k_l$			Fixing $k_l$ and $k_m$ , Varying $k_r$		
		{2, 7, 2}	{3, 7, 2}	{4, 7, 2}	{3, 7, 3}	{3, 7, 4}	{3, 7, 5}
Params		<b>0.41 M</b>	<u>0.44 M</u>	0.48 M	<b>0.48 M</b>	<u>0.52 M</u>	0.56 M
a-DSC (%)	Esophagus	73.56 ± 9.58	<b>74.55 ± 8.83</b>	71.82 ± 9.42	<u>74.06 ± 9.80</u>	<b>74.24 ± 8.01</b>	73.19 ± 9.79
	Heart	<b>93.63 ± 2.22</b>	92.96 ± 3.46	93.21 ± 2.28	<u>92.98 ± 1.55</u>	<b>94.22 ± 1.52</b>	92.64 ± 4.72
	Trachea	83.06 ± 4.33	<b>85.78 ± 4.32</b>	<u>84.22 ± 4.04</u>	<b>86.43 ± 4.01</b>	84.47 ± 5.33	<u>84.54 ± 6.81</u>
	Aorta	<u>89.78 ± 4.66</u>	89.09 ± 4.9	<b>89.96 ± 5.78</b>	88.27 ± 5.13	<u>89.39 ± 7.26</u>	<b>92.11 ± 1.36</b>
	Mean	<u>85.01</u>	<b>85.60</b>	84.80	85.44	<u>85.58</u>	<b>85.62</b>
a-SSD (mm)	Esophagus	4.16 ± 1.91	<b>2.67 ± 0.98</b>	<u>3.35 ± 1.34</u>	3.22 ± 1.14	<b>2.67 ± 0.79</b>	<u>2.69 ± 1.06</u>
	Heart	<b>4.61 ± 3.97</b>	<u>12.48 ± 16.53</u>	16.54 ± 24.38	<u>2.66 ± 0.94</u>	<b>1.94 ± 0.66</b>	18.5 ± 27.98
	Trachea	<u>5.22 ± 2.37</u>	<b>3.18 ± 1.15</b>	6.61 ± 4.57	<u>4.80 ± 4.70</u>	<b>3.01 ± 2.48</b>	6.14 ± 4.48
	Aorta	<u>4.14 ± 2.33</u>	5.41 ± 3.32	<b>2.16 ± 1.22</b>	<b>2.26 ± 0.75</b>	<u>2.82 ± 1.33</u>	5.27 ± 4.90
	Mean	<b>4.53</b>	<u>5.94</u>	7.17	<u>3.24</u>	<b>2.61</b>	8.15
a-Preci (%)	Esophagus	<b>79.32 ± 7.98</b>	<u>77.58 ± 6.96</u>	74.22 ± 11.25	76.66 ± 7.94	<u>79.67 ± 6.67</u>	<b>82.10 ± 6.82</b>
	Heart	<b>95.73 ± 3.57</b>	92.84 ± 6.13	95.10 ± 4.60	96.34 ± 3.64	<b>96.60 ± 2.27</b>	92.70 ± 8.71
	Trachea	77.01 ± 8.83	<b>85.75 ± 6.56</b>	<u>78.43 ± 6.78</u>	<u>84.60 ± 7.82</u>	<b>90.98 ± 6.73</b>	80.29 ± 10.06
	Aorta	<u>89.70 ± 3.38</u>	87.79 ± 5.68	<b>90.51 ± 3.84</b>	<b>92.28 ± 2.98</b>	<u>91.76 ± 2.78</u>	90.87 ± 3.10
	Mean	<u>85.44</u>	<b>85.99</b>	84.56	<u>87.47</u>	<b>89.75</b>	86.49

### 3.3.2. The Input Feature Initialization

We also evaluated the initialization configuration of the input feature  $b^0 = \mathcal{K}^0 f$  on the finest cycle  $\ell = 1$  to verify its sensitivity. We compared different variants of the initialization method, such as zero initialization, random normal distribution initialization, and  $\psi(\mathcal{K}^0 f)$  initialization with the batch normalization operation  $\psi(\cdot)$ , where  $\mathcal{K}^0$  was obtained by learning the convolutional kernel with a size of  $p \times 3 \times 3$  for 2D segmentation or  $p \times 3 \times 3 \times 3$  for 3D segmentation.

Table 3 shows the quantitative comparison of these variants. The model with  $\psi(\mathcal{K}^0 f)$  initialization achieved a-DSC, a-SSD, and a-Preci values of 85.58% (ranked second), 2.61 mm (top-ranked), and 89.75% (top-ranked), respectively. Although the model with random initialization had the highest a-DSC score, the a-SSD and a-Preci scores were significantly lower than the model with  $\psi(\mathcal{K}^0 f)$  initialization. To this end, we set the proposed framework with  $\psi(\mathcal{K}^0 f)$  initialization as the default in this work.

### 3.3.3. Weight Sharing

To demonstrate the flexibility of the proposed framework, which does not have to be the different  $\mathcal{K}'_{q,\ell,j}$  parameter configurations in different nonlinear  $\mathcal{F}_{\mathcal{K}'_{q,j}}^\ell$  (with respect to  $j$ ) within the  $\ell^{th}$  pre-smoothing ( $q = l$ ) or post-smoothing ( $q = r$ ) block, we conducted several variants that had different sharing settings among  $\mathcal{K}'_{q,\ell,j}$  (with respect to the iteration step  $j$ ).

We only varied the weight sharing settings to verify the sensitivity of the model. We compared four variants, and the results are shown in Table 4. From the evaluation metrics, we see that the model without the weight sharing configuration had more parameters

and achieved the best performance on the a-DSC, a-SSD, and a-Preci values, respectively. The performance of the other three models did not show a significant differences. Therefore, we adopted the default unshared version in the rest of this paper.

**Table 3.** Quantitative assessment with the a-DSC, a-SSD, and a-Preci values of different initialization techniques (zero initialization, random initialization, and feature extraction initialization  $\psi(\mathcal{K}^0(f))$ ) using the proposed 2D FAS-UNet framework on the 2D SegTHOR validation datasets with four organs: esophagus, heart, trachea, and aorta. “Mean” denotes an average score segmenting all organs. The best and second places are highlighted in bold font and underlined, respectively.

Feature Initialization		Zero Initi	Random Initi	$\psi(\mathcal{K}^0(f))$ Initi
Params		0.52 M	0.52 M	0.52 M
a-DSC (%)	Esophagus	72.56 ± 11.66	<b>75.84 ± 10.02</b>	<u>74.24 ± 8.01</u>
	Heart	<u>92.40 ± 4.48</u>	92.37 ± 5.06	<b>94.22 ± 1.52</b>
	Trachea	84.21 ± 4.27	<b>85.08 ± 3.38</b>	<u>84.47 ± 5.33</u>
	Aorta	<b>91.17 ± 4.49</b>	<u>89.81 ± 4.22</u>	89.39 ± 7.26
	Mean	85.08	<b>85.77</b>	<u>85.58</u>
a-SSD (mm)	Esophagus	<b>2.39 ± 1.2</b>	2.85 ± 0.87	<u>2.67 ± 0.79</u>
	Heart	<u>15.38 ± 21.8</u>	16.82 ± 32.15	<b>1.94 ± 0.66</b>
	Trachea	5.67 ± 2.98	<u>4.66 ± 3.59</u>	<b>3.01 ± 2.48</b>
	Aorta	<u>2.85 ± 1.97</u>	7.59 ± 8.94	<b>2.82 ± 1.33</b>
	Mean	<u>6.57</u>	7.98	<b>2.61</b>
a-Preci (%)	Esophagus	<b>81.8 ± 6.41</b>	<u>81.3 ± 6.49</u>	79.67 ± 6.67
	Heart	92.53 ± 8.43	<u>93.31 ± 8.25</u>	<b>96.6 ± 2.27</b>
	Trachea	79.92 ± 8.34	<u>79.0 ± 5.46</u>	<b>90.98 ± 6.73</b>
	Aorta	<b>91.92 ± 2.75</b>	88.57 ± 5.28	<u>91.76 ± 2.78</u>
	Mean	<u>86.55</u>	85.55	<b>89.75</b>

**Table 4.** Quantitative assessment with the a-DSC, a-SSD, and a-Preci values of using the proposed 2D FAS-UNet framework with/without weight sharing of pre-/post smoothing steps on the 2D SegTHOR validation datasets with four organs: esophagus, heart, trachea, and aorta. “Mean” denotes an average score segmenting all organs. The best and second places are highlighted in bold font and underlined, respectively.

Block Weight Sharing		w/o	LSB	RSB	LSB and RSB
Params		0.52 M	0.45 M	<u>0.41 M</u>	<b>0.34 M</b>
a-DSC (%)	Esophagus	<b>74.24 ± 8.01</b>	<u>72.53 ± 10.17</u>	73.01 ± 9.91	72.53 ± 10.28
	Heart	<b>94.22 ± 1.52</b>	90.51 ± 6.98	89.33 ± 5.11	<u>92.00 ± 3.97</u>
	Trachea	<b>84.47 ± 5.33</b>	84.14 ± 5.21	<u>84.55 ± 2.60</u>	83.37 ± 4.10
	Aorta	89.39 ± 7.26	<b>90.46 ± 2.99</b>	<u>90.17 ± 4.30</u>	90.03 ± 2.40
	Mean	<b>85.58</b>	84.41	84.26	<u>84.48</u>
a-SSD (mm)	Esophagus	<b>2.67 ± 0.79</b>	3.22 ± 0.97	<u>2.69 ± 0.72</u>	4.26 ± 1.91
	Heart	<b>1.94 ± 0.66</b>	22.37 ± 16.99	34.91 ± 31.76	<u>14.51 ± 19.82</u>
	Trachea	<b>3.01 ± 2.48</b>	6.30 ± 4.50	<u>5.17 ± 3.36</u>	5.96 ± 3.38
	Aorta	<b>2.82 ± 1.33</b>	4.64 ± 3.87	<u>4.61 ± 1.78</u>	8.12 ± 5.49
	Mean	<b>2.61</b>	9.13	11.85	<u>8.21</u>
a-Preci (%)	Esophagus	<b>79.67 ± 6.67</b>	<u>77.94 ± 7.67</u>	77.36 ± 6.60	74.64 ± 8.14
	Heart	<b>96.60 ± 2.27</b>	88.05 ± 11.50	88.72 ± 9.14	<u>92.19 ± 7.80</u>
	Trachea	<b>90.98 ± 6.73</b>	<u>82.34 ± 9.26</u>	78.14 ± 5.00	<u>77.36 ± 8.26</u>
	Aorta	<b>91.76 ± 2.78</b>	<u>88.84 ± 3.57</u>	87.08 ± 6.09	85.82 ± 3.77
	Mean	<b>89.75</b>	<u>84.29</u>	82.82	82.50

### 3.3.4. Effects of Varying the Number of Channels

In this section, we analyze the segmentation performance of the proposed method with varying the number  $p$  of feature channels. In the FAS-Solution module, the pre-smoothing

and post-smoothing steps (with  $k_l$  and  $k_r$  iterations, respectively) had the same update structure with the coarsest smoothing step, which included  $k_m$  iterations. Therefore, we set the number of smoothing iterations as  $\{k_l, k_m, k_r\} = \{3, 7, 4\}$  and adopted a series of channel parameters  $p = 80, 64, 48, 32, 16$ , respectively, for comparison. Here,  $p = 80$  means that, in each convolutional operation of the FAS-Solution module, there were 80 filters with the same kernel size of  $3 \times 3$ .

Table 5 shows the quantitative comparison of different  $p$ -configurations. It reveals that, as the number of channels increased, the parameter of our model squarely increased. Additionally, the networks with the numbers of 64 and 16 achieved a-DSC scores of 86.83% (ranking first) and 84.88% (ranking lowest), respectively. When the number of channels was less than 64, increasing the number of channels could improve the a-DSC value, and one can see from Table 5 that the number of channels had a significant impact on the performance of the model. Based on this observation, the configuration with 64 channels is a preferable setting to balance the segmentation performance and computational costs, and we fixed  $p = 64$  throughout all the 2D experiments.

**Table 5.** Quantitative assessment with the a-DSC, a-SSD, and a-Preci values of different channel numbers  $p$  using the proposed 2D FAS-UNet framework on the 2D SegTHOR validation datasets with four organs: esophagus, heart, trachea, and aorta. “Mean” denotes an average score segmenting all organs. The best and second places are highlighted in bold font and underlined, respectively.

Number of Channels		80	64	48	32	16
Params		3.24 M	2.08 M	1.17 M	<u>0.52 M</u>	<b>0.13 M</b>
a-DSC (%)	Esophagus	<u>75.20 ± 11.53</u>	<b>75.34 ± 12.59</b>	74.50 ± 11.62	74.24 ± 8.01	70.30 ± 11.04
	Heart	<u>94.19 ± 1.95</u>	93.79 ± 1.79	92.04 ± 5.66	<b>94.22 ± 1.52</b>	93.17 ± 2.41
	Trachea	<u>84.77 ± 4.08</u>	<u>86.97 ± 5.05</u>	<b>87.29 ± 3.65</b>	84.47 ± 5.33	85.86 ± 3.62
	Aorta	<b>91.46 ± 5.14</b>	<u>91.20 ± 3.45</u>	90.20 ± 6.76	89.39 ± 7.26	90.17 ± 5.29
	Mean	<u>86.41</u>	<b>86.83</b>	86.01	85.58	84.88
a-SSD (mm)	Esophagus	2.72 ± 0.98	<u>2.49 ± 1.22</u>	<b>2.37 ± 1.26</b>	2.67 ± 0.79	3.18 ± 1.71
	Heart	8.33 ± 12.2	<u>8.45 ± 16.54</u>	12.79 ± 26.80	<b>1.94 ± 0.66</b>	12.11 ± 18.35
	Trachea	7.52 ± 4.88	4.61 ± 3.50	3.66 ± 2.63	<b>3.01 ± 2.48</b>	<u>3.20 ± 1.95</u>
	Aorta	<b>1.99 ± 1.02</b>	5.20 ± 4.92	2.80 ± 1.73	2.82 ± 1.33	<u>2.38 ± 1.33</u>
	Mean	<u>5.14</u>	5.19	5.40	<b>2.61</b>	5.22
a-Preci (%)	Esophagus	<u>81.02 ± 6.21</u>	<b>81.19 ± 10.70</b>	80.33 ± 8.23	79.67 ± 6.67	77.39 ± 8.61
	Heart	<u>95.84 ± 3.90</u>	<u>95.85 ± 3.45</u>	91.52 ± 8.89	<b>96.60 ± 2.27</b>	94.31 ± 2.92
	Trachea	80.30 ± 8.44	<u>84.64 ± 9.50</u>	<u>86.50 ± 8.41</u>	<b>90.98 ± 6.73</b>	85.14 ± 9.38
	Aorta	<b>93.04 ± 2.36</b>	91.17 ± 3.81	<u>91.93 ± 3.61</u>	91.76 ± 2.78	89.85 ± 3.56
	Mean	87.55	<u>88.21</u>	87.57	<b>89.75</b>	86.67

To provide insights into the model hyperparameter configurations of the proposed 3D FAS-UNet version on the 3D HVSMR-2016 datasets and 3D CHAOS-CT datasets, we also carried out a series of ablation experiments to investigate the influence of two key design variables, the number of channels and the number of convolutional blocks. The evaluation indicated that the network performed better with the configurations  $p = 32$ ,  $\{k_l, k_m, k_r\} = \{3, 6, 2\}$  for the 3D HVSMR-2016 datasets and  $\{3, 5, 2\}$  for the 3D CHAOS-CT datasets as the default. Here, we do not detail these comparisons.

Finally, we illustrate the hyperparameter configurations of the proposed FAS-UNet on each dataset throughout all experiments, as shown in Table 6.

**Table 6.** The hyperparameter configurations of the proposed FAS-UNet framework on different datasets.

Datasets	Layer Number $\mathcal{L}$	Feature Number $p$	$\{k_l, k_m, k_r\}$	Weight Sharing on $\mathcal{K}'_{q,\ell,j}$	Feature Initialization	Params
2D SegTHOR	5	64	{3, 7, 4}	No	$\psi(\mathcal{K}^0(f))$	2.08 M
3D HVSMR 2016	4	32	{3, 6, 2}	No	$\psi(\mathcal{K}^0(f))$	1.01 M
3D CHAOS CT	4	32	{3, 5, 2}	No	$\psi(\mathcal{K}^0(f))$	1.00 M

3.4. The 2D FAS-UNet for the SegTHOR Datasets

We evaluated the proposed network on the 2D SegTHOR datasets and compared the visualizations and quantitative metrics with the existing state-of-the-art segmentation methods, including 2D UNet [9], CA-Net [20], CE-Net [42], CPFNet [43], ERFNet [44], UNet++ [11], and LinkNet [45].

In Table 7, we show the quantitative results of each organ’s segmentation compared with the other seven models. We can see that the segmentation performance of the heart was the best among all organs, and its a-DSC score was more than 92% for each method, followed by the aorta, and the worst was the esophagus. The main reason for the good performance in extracting the heart was that the heart region is the largest, and its inner pixel value changes little, while its boundary is more obvious (see Figure 3), while the esophageal region is the smallest among all organs, which increased the segmentation difficulty.

**Table 7.** Performance comparisons between the proposed 2D FAS-UNet and the popular networks using the a-DSC, a-SSD, and a-Preci values on the 2D SegTHOR validation datasets with four organs: esophagus, heart, trachea, and aorta. “Mean” denotes an average score segmenting all organs. The best and second places are highlighted in bold font and underlined, respectively.

Method	2D UNet [9]	CA-Net [20]	CE-Net [42]	CPFNet [43]	ERFNet [44]	UNet++ [11]	LinkNet [45]	2D FAS-UNet	
<b>Params</b>	17.26 M	2.78 M	29.00 M	30.65 M	<b>2.06 M</b>	9.05 M	21.79 M	<u>2.08 M</u>	
a-DSC (%)	Esophagus	73.77 ± 13.23	<b>76.26 ± 8.72</b>	64.17 ± 10.78	66.58 ± 10.02	69.68 ± 5.93	65.19 ± 9.18	<u>75.34 ± 12.59</u>	
	Heart	93.42 ± 3.270	93.61 ± 2.45	<b>94.01 ± 1.17</b>	92.05 ± 4.39	92.59 ± 5.29	93.46 ± 2.36	<u>93.79 ± 1.79</u>	
	Trachea	86.71 ± 2.69	85.58 ± 4.38	85.12 ± 4.37	<b>87.41 ± 2.84</b>	79.10 ± 5.79	85.63 ± 5.78	<u>86.97 ± 5.05</u>	
	Aorta	90.42 ± 3.45	<b>91.2 ± 1.59</b>	88.38 ± 3.95	88.92 ± 4.98	90.08 ± 4.06	90.61 ± 2.66	<u>91.2 ± 3.45</u>	
	Mean	86.08	<u>86.66</u>	82.92	83.74	82.86	84.83	83.37	<b>86.83</b>
a-SSD (mm)	Esophagus	2.78 ± 1.36	4.14 ± 1.46	<u>2.54 ± 0.92</u>	2.57 ± 0.79	3.92 ± 0.93	4.77 ± 1.33	2.88 ± 1.22	<b>2.49 ± 1.22</b>
	Heart	14.37 ± 22.78	<u>4.87 ± 7.09</u>	<b>4.57 ± 5.31</b>	16.02 ± 22.26	7.76 ± 15.63	9.99 ± 11.17	8.42 ± 17.02	8.45 ± 16.54
	Trachea	6.05 ± 6.14	6.85 ± 4.87	<u>3.81 ± 2.05</u>	4.46 ± 3.91	8.62 ± 4.26	5.84 ± 5.38	<b>2.57 ± 1.35</b>	4.61 ± 3.50
	Aorta	6.36 ± 6.51	<u>3.64 ± 1.68</u>	5.42 ± 1.62	7.51 ± 5.53	<b>3.36 ± 2.31</b>	4.41 ± 2.29	5.85 ± 4.02	5.2 ± 4.92
	Mean	7.39	<u>4.88</u>	<b>4.08</b>	7.64	5.92	6.25	4.93	5.19
a-Preci (%)	Esophagus	<b>82.82 ± 4.69</b>	<u>82.67 ± 5.46</u>	79.14 ± 7.39	73.92 ± 7.29	68.78 ± 7.88	79.83 ± 7.72	74.1 ± 7.32	81.19 ± 10.70
	Heart	93.47 ± 5.39	94.91 ± 3.79	<u>95.26 ± 2.66</u>	91.85 ± 8.39	92.61 ± 9.62	94.27 ± 4.58	95.00 ± 4.71	<b>95.85 ± 3.45</b>
	Trachea	<u>85.21 ± 5.91</u>	81.72 ± 7.29	84.81 ± 9.74	84.48 ± 6.90	72.46 ± 9.71	<b>87.59 ± 9.05</b>	83.88 ± 9.26	84.64 ± 9.50
	Aorta	90.01 ± 2.81	88.84 ± 4.04	86.10 ± 4.73	88.84 ± 2.92	88.07 ± 3.35	<u>90.39 ± 3.57</u>	88.91 ± 5.28	<b>91.17 ± 3.81</b>
	Mean	87.88	87.03	86.33	84.77	80.48	<u>88.02</u>	85.47	<b>88.21</b>

The proposed method achieved an a-DSC value of 86.83% (ranked first) and an a-Preci value of 88.21% (ranked first) with only 2.08 M parameters. Compared with the state-of-the-art models CA-Net (second-ranked in the a-DSC score) and UNet++ (ranked second in the a-Preci value), the proposed 2D FAS-UNet obtained a 0.17% improvement of the a-DSC score with only 75% as many parameters as CA-Net and a 0.19% improvement of the a-Preci value with only 22% as many parameters as UNet++. Our method also had a higher a-DSC score than the third-ranked UNet by 0.75%, but had far fewer parameters than the

17.26 M of UNet. Compared with ERFNet, which achieved a-DSC and a-Preci values of 82.86% and 80.48% with the fewest parameters, respectively, FAS-UNet was higher in the overall a-DSC rankings. The a-DSC score of ERFNet ranked last, so we think it may fall into an under-fitting situation. This shows that ERFNet reduces the performance of the network while saving the parameters. CA-Net obtained a good a-DSC value because the attention mechanism may improve the segmentation results of small organs.

The evaluation results were also measured in terms of the a-SSD value for segmentation predictions of the eight methods. The proposed method achieved an a-SSD value of 5.19 mm (ranked fourth); CE-Net ranked first, which achieved an a-SSD value of 4.08 mm; ERFNet with the fewest parameters ranked fifth with 5.92 mm. CA-Net and Link-net ranked second and third, which had 2.78 M and 21.79 M parameters, respectively. The good results of CA-Net in terms of the a-SSD metric may be due to the use of multiple attention mechanisms, which enables the network to suppress the background region, and the network has a stronger ability to recognize the object region. The a-SSD score of UNet++ was much higher than that of 2D FAS-UNet, which shows that the over-segmented pixels were less than the under-segmented pixels. Meanwhile, we observed that the a-SSD value of our method was close to that of LinkNet in three organs; only the tracheal region was significantly worse than it, which makes it better than our method in the mean a-SSD score.

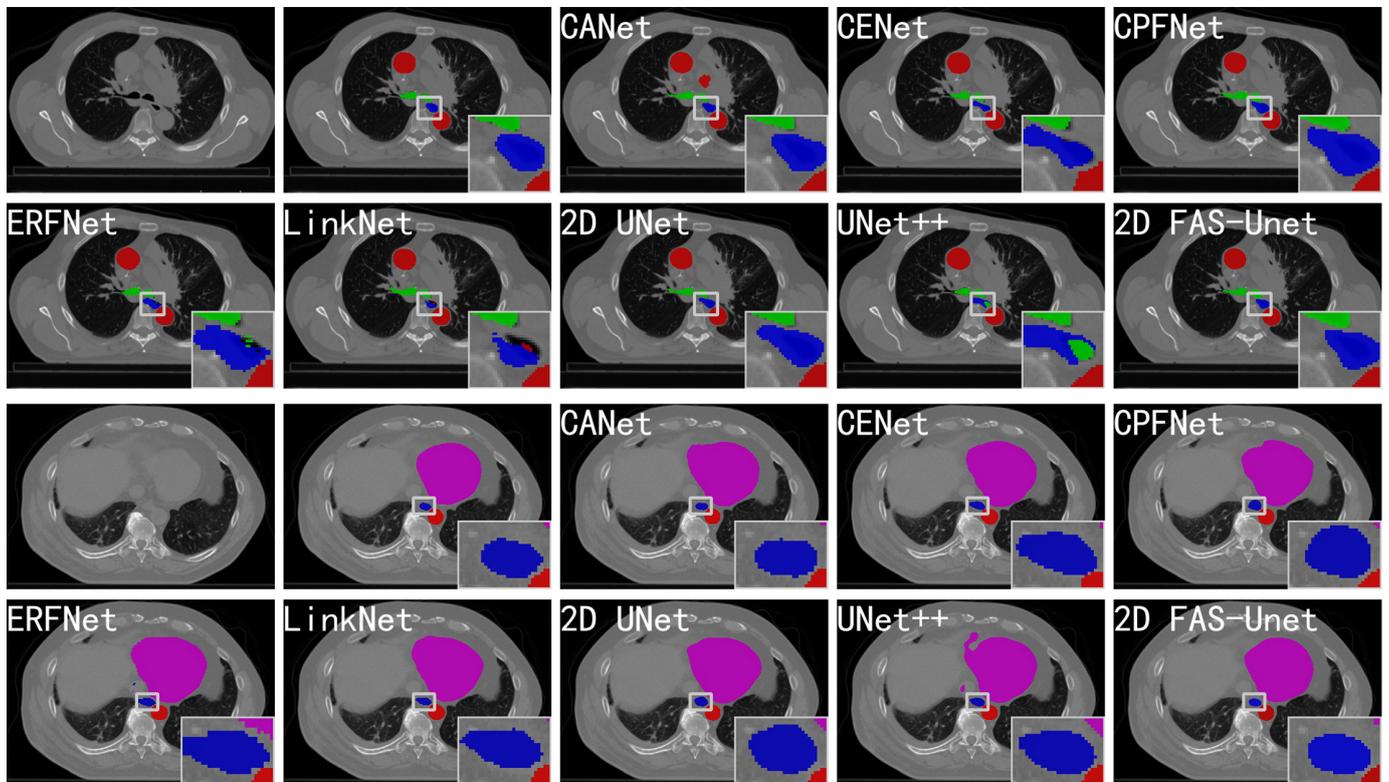
Figure 3 evaluates the visualizations of the segmented predictions obtained by the popular methods. One can observe that all methods except CA-Net (with an obvious over-segmentation) can accurately segment the aortic region (red). The reason may be that, during the imaging process, the aortic organ may be assigned to the very same image pixel, which leads to a small difference of the internal pixel value; in particular, the network can accurately learn its features. Almost all methods can also approximately segment the trachea organ shape (green); only the organ boundary is not clearly visible. This may be a common problem in small object segmentation because the hard-to-detect small-scale feature will be degraded rapidly with convolution and pooling. All methods were able to extract the heart location (magenta), while the visual quality of the proposed method was significantly better than the state-of-the-art methods. Our approach achieved the least missing pixels at the organ boundary, which resulted in substantially better performance than the existing results; the visualization remained comparable. It should also be emphasized that only a few methods performed well on the left boundary because the left side of the heart's boundary is very blurry. For example, UNet++ presented over-segmentation, and CA-Net on the left boundary was significantly different from the ground-truth, while the results of ERFNet and LinkNet had significant differences compared with the ground-truth in the shape aspect. This also shows that our method is robust to the heavy occlusion of illumination and large background clutters.

For the esophagus organ (blue) in Figure 3, the segmented results of CE-Net, CPFNet, and UNet had significant differences compared with the ground-truth in the shape aspect. The results of ERFNet and UNet++ had some trachea pixels within the esophagus, and LinkNet's prediction had some aortic pixels, all of which were clearly error-segmented. The predictions of CA-Net and FAS-UNet were similar to manual segmentation, but the result of CA-Net had a small aorta patch in the background region, while our method obviously performed better. The reasons for the esophagus's bad results were the fuzzy boundary and the small pixel value difference; in particular, the esophagus and aorta almost overlap in the second slice. Therefore, the proposed method is more robust, and it is more difficult for it to be affected by noise. This further shows that our method is effective in medical image segmentation.

### 3.5. The 3D FAS-UNet for the HVSMR-2016 Datasets

We also conducted the segmentation experiments of our 3D FAS-UNet on the HVSMR-2016 datasets. We compared our predictions with seven baseline models including 3D UNet [46], AnatomyNet [47], DMFNet [48], HDC-Net [49], RSANet [50], Bui-Net [51], and

VoxResNet [52]. Table 8 shows the segmentation results of different methods. Clearly, our method with fewer parameters ranked second in both the a-DSC and a-SSD values and obtained the top rank in mean precision.



**Figure 3.** Comparison with the other state-of-the-art networks on the validation set of the 2D SegTHOR datasets. The blue, pink, green, and red regions represent the esophagus, heart, trachea, and aorta, respectively. Form left to right in the first and third rows: the original images, ground-truth, and segmentation results of CANet, CE-Net, and CPFNet, respectively. Form left to right in the second and fourth rows: the segmentation results of ERFNet, LinkNet, 2D UNet, UNet++, and 2D FAS-UNet, respectively.

**Table 8.** Performance comparisons of the proposed 3D FAS-UNet and the popular networks using the a-DSC, a-SSD, and a-Preci values on the 3D HVSMR-2016 validation datasets with both organs: myocardium and blood pool. “MY” and “BP” denote myocardium and blood pool, respectively, “Mean” denotes an average score segmenting all organs. The best and second places are highlighted in bold font and underlined, respectively.

Method	3D UNet [46]	AnatomyNet [47]	DMFNet [48]	HDC-Net [49]	RSANet [50]	Bui-Net [51]	VoxResNet [52]	3D FAS-UNet	
Params	8.16 M	<u>0.73 M</u>	3.87 M	<b>0.30 M</b>	24.55 M	2.53 M	1.70 M	1.01 M	
a-DSC (%)	MY	<b>76.42 ± 3.32</b>	69.50 ± 4.69	70.30 ± 1.03	69.77 ± 5.10	71.94 ± 0.69	69.84 ± 0.47	65.79 ± 6.00	76.42 ± 4.38
	BP	89.21 ± 1.61	86.01 ± 1.51	<b>89.45 ± 0.18</b>	88.03 ± 2.24	84.71 ± 0.93	<u>89.4 ± 0.46</u>	84.66 ± 1.67	89.01 ± 0.18
	Mean	<b>82.82</b>	77.76	79.88	78.90	78.33	79.62	75.23	<u>82.72</u>
a-SSD (mm)	MY	<b>2.06 ± 0.73</b>	4.03 ± 0.31	2.74 ± 0.20	3.36 ± 0.32	<u>2.31 ± 0.30</u>	2.56 ± 0.22	2.52 ± 1.06	2.32 ± 0.27
	BP	3.05 ± 1.53	2.98 ± 1.21	<b>2.07 ± 0.48</b>	2.65 ± 0.02	<u>4.01 ± 0.55</u>	2.58 ± 0.51	3.47 ± 0.04	<u>2.57 ± 1.15</u>
	Mean	2.56	3.50	<b>2.40</b>	3.00	3.16	2.57	2.99	<u>2.44</u>
a-Preci (%)	MY	81.02 ± 9.13	71.25 ± 6.22	80.27 ± 4.22	67.41 ± 12.94	76.76 ± 9.00	<u>83.94 ± 6.73</u>	77.94 ± 10.40	<b>84.61 ± 7.14</b>
	BP	87.44 ± 2.62	89.73 ± 4.61	<b>91.78 ± 0.54</b>	87.84 ± 2.09	82.86 ± 6.08	89.66 ± 0.56	84.29 ± 1.13	<u>91.18 ± 2.36</u>
	Mean	84.23	80.49	86.03	77.62	79.81	<u>86.80</u>	81.11	<b>87.90</b>

The proposed method achieved an a-DSC value of 82.75% (ranked second) with only 1.01 M parameters and followed the first-ranked 3D UNet by 0.1% in the a-DSC score with

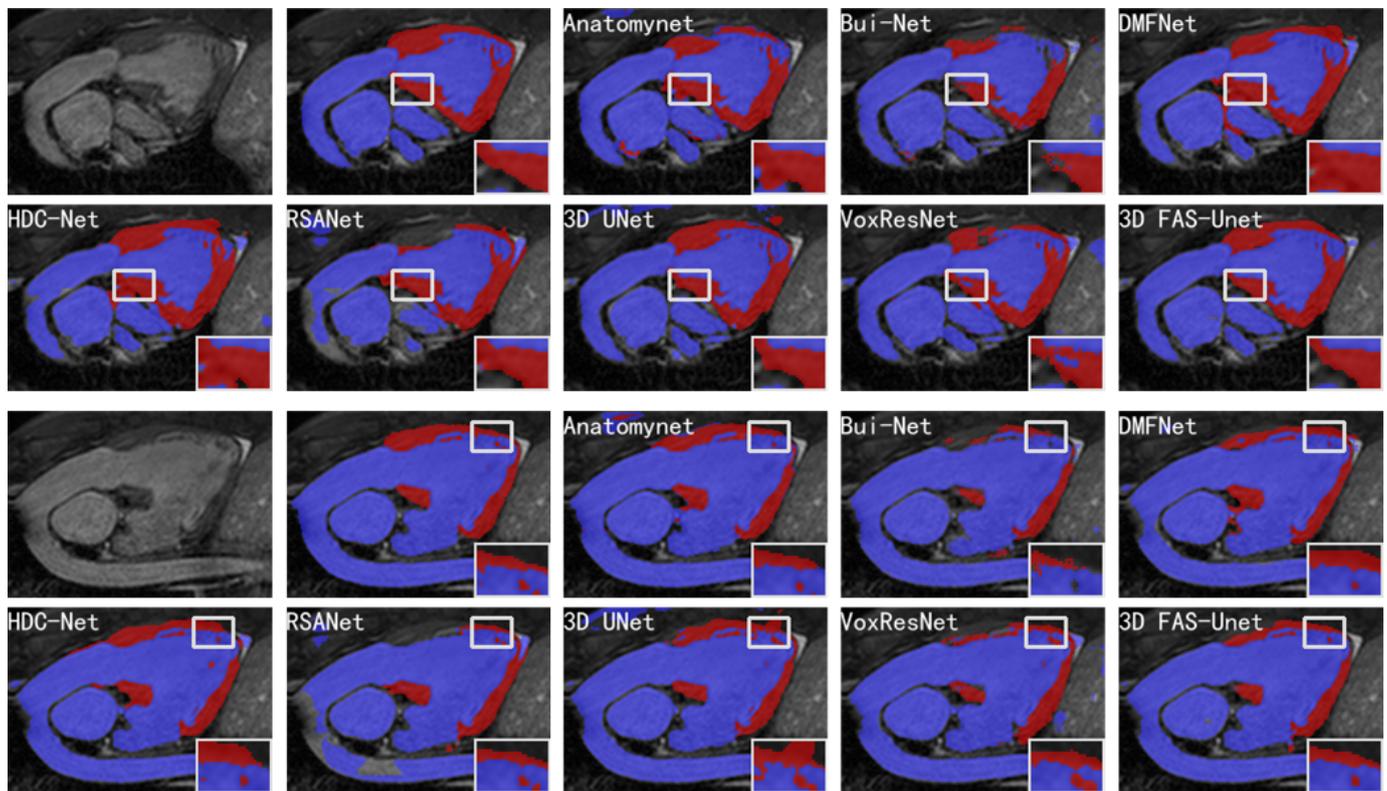
only 15% as many parameters as 3D UNet. Meanwhile, our method outperformed the third-ranked DMFNet by 2.84%. Although the numbers of parameters of HDC-Net and AnatomyNet were lower than that of our method, the a-DSC score of 3D FAS-UNet was 3.82% and 4.96% higher than theirs, respectively. One may notice that a black-box (unexplainable) network with a small number of parameters has low segmentation performance, which may be due to under-fitting. However, the number of parameters in RSANet is a bit large, and the effect was also not good enough. This may be due to too little training data, so the model appears to be over-fitting. Our method also obtained an a-SSD value of 2.44 mm (2nd rank), which was lower than DMFNet's 2.40 mm (1st rank) by 0.12 mm, and it slightly improved compared with 3D UNet, whose a-SSD value was 2.56 mm (3rd ranked). Although 3D FAS-UNet had a slightly lower a-SSD than DMFNet, it had 2.86 M fewer parameters. Compared with HDC-Net and AnatomyNet with fewer parameters, our method performed better on the a-SSD metric for myocardium and blood pool. The result of the a-SSD value shows that our method has good performance in segmenting object boundaries. The proposed method obtained the best a-Preci score of 87.90%, which was higher than the second-ranked Bui-Net by 1.10%. Although 3D FAS-UNet ranked third in the number of parameters, all three metrics were better than HDC-Net and AnatomyNet with fewer parameters. Therefore, our method achieves a good balance between the number of parameters and performance. Experiments on these datasets showed that the proposed FAS-driven explainable model can be robustly applied to 3D medical segmentation tasks.

Figure 4 visualizes the segmentation results of different methods on two slices of the HVSMR-2016 datasets, and it can be clearly observed that the proposed model highlighted less over-segmented regions outside the ground-truth compared with other methods. Meanwhile, it also can show that it was hard for our method to be affected by the voxels in the background region, where it did not predict the voxels of the background region as blood pools or myocardium, but most other methods predicted more background voxels as the object. The results showed that these methods are easily affected by noise in the background region.

In general, one can observe that AnatomyNet, VoxResNet, and 3D UNet showed obvious segmentation noise (over-segmented region). The reason is that the network collects much noise information in the interactions from input data of the network due to a too simple data pre-processing method, which affects the feature extraction. Several methods presented over-segmentation in the myocardial zoom-in, because the pixel value of this organ is very close to the background. The myocardium is structurally distorted, which makes the shape of the myocardium completely different compared to normal/healthy myocardium. Although VoxResNet did not have this phenomenon, it divided the middle part of the myocardial region into blood pools, which was also an obvious error segmentation. Only 3D UNet and our method performed better; especially, our method was closer to the ground-truth in shape. Further, all methods had poor segmentation results in the upper myocardial region; the intensity homogeneity between this organ and the upper background indicates that this region is very difficult to segment. We can see from the zoom-in results that many methods have obvious over-segmentation or under-segmentation for the myocardium and blood pool. Compared with Bui-Net and RSANet, we observed that the proposed learnable specialized FAS-UNet network still had obvious advantages in this region, and the results were very close to the ground-truth in the myocardial region (red) with respect to the shape and size. For the blood pool region (blue), our results did not show significant differences with other methods.

The proposed network integrates medical image data and the variational convexity MS model and algorithm (FAS scheme), and implements them through convolution-based deep learning, so it may be possible to design specialized modules that automatically satisfy some of the physical invariants for better accuracy and robustness. Qualitative and quantitative experimental results demonstrated the effectiveness and superiority of our method. It can not only correctly locate the position of the myocardium, but also segment the myocardium and blood pool in the complex marginal region. Moreover, the integrity

and continuity of our method in the object were well preserved. Overall, it performed better than the existing state-of-the-art methods in 3D medical image segmentation.



**Figure 4.** Visualizations of different methods for cardiovascular MR segmentation of different slices. From left to right in the first and third rows: the original images, the ground-truth, and the segmentation results of AnatomyNet, Skip-connected 3D DenseNet, and DMFNet, respectively. From left to right in the second and fourth rows: the segmentation results of HDC-Net, RSANet, 3D UNet, VoxResNet, and 3D FAS-UNet, respectively. The blue and red colors represent blood pool and myocardium, respectively.

### 3.6. The 3D FAS-UNet for the CHAOS-CT Datasets

In this part, the proposed 3D FAS-UNet was compared with seven baseline models on the 3D CHAOS-CT datasets, including 3D UNet [46], Bui-Net [51], DMFNet [48], 3D ESPNet [53], RSANet [50], RatLesNetV2 [54], and HDC-Net [49].

Firstly, we used a post-processing technique to improve the prediction results, where small undesirable clusters of voxels separated from the largest connection component may be over-segmented or the “holes” inside the liver may also be under-segmented. Table 9 shows the prediction results of different methods. Before post-processing, the proposed method achieved an a-DSC of 96.69% (top-ranked) with only 1.00 M parameters, which is slightly higher than the second-ranked RSANet by 0.08% in the mean a-DSC score with only about 4% as many parameters as RSANet, and further outperformed the third-ranked DMFNet by 0.28%. Our method also obtained an a-SSD value of 4.04mm (ranked second), which is the same as RSANet (top-ranked). Although 3D FAS-UNet had a lower mean a-Preci value than 3D ESPNet by 1.54%, it had 2.57 M fewer parameters.

The a-DSC scores of all methods were significantly improved by post-processing techniques. The 3D FAS-UNet achieved an a-DSC score of 97.11% (ranked second) and followed the first-ranked 3D UNet by 0.05% in the mean a-DSC with only 15% as many parameters as 3D UNet. Our method also obtained an a-SSD value of 1.16 mm (top-ranked), and it was better than 3D UNet (ranked 2nd) and RSANet (ranked 3rd) by 0.07 mm and 0.10 mm, respectively. The 3D FAS-UNet achieved an a-Preci of 96.73%, which is lower than 3D ESPNet and RatLesNetV2. Although 3D FAS-UNet ranked third in the parameter evalua-

tion, all five metrics were better than HDC-Net and RatLesNetV2 with fewer parameters; only the a-Preci value with post-processing was lower than that of RatLesNetV2. Thus, our method achieves a good balance between the number of parameters and segmentation performance.

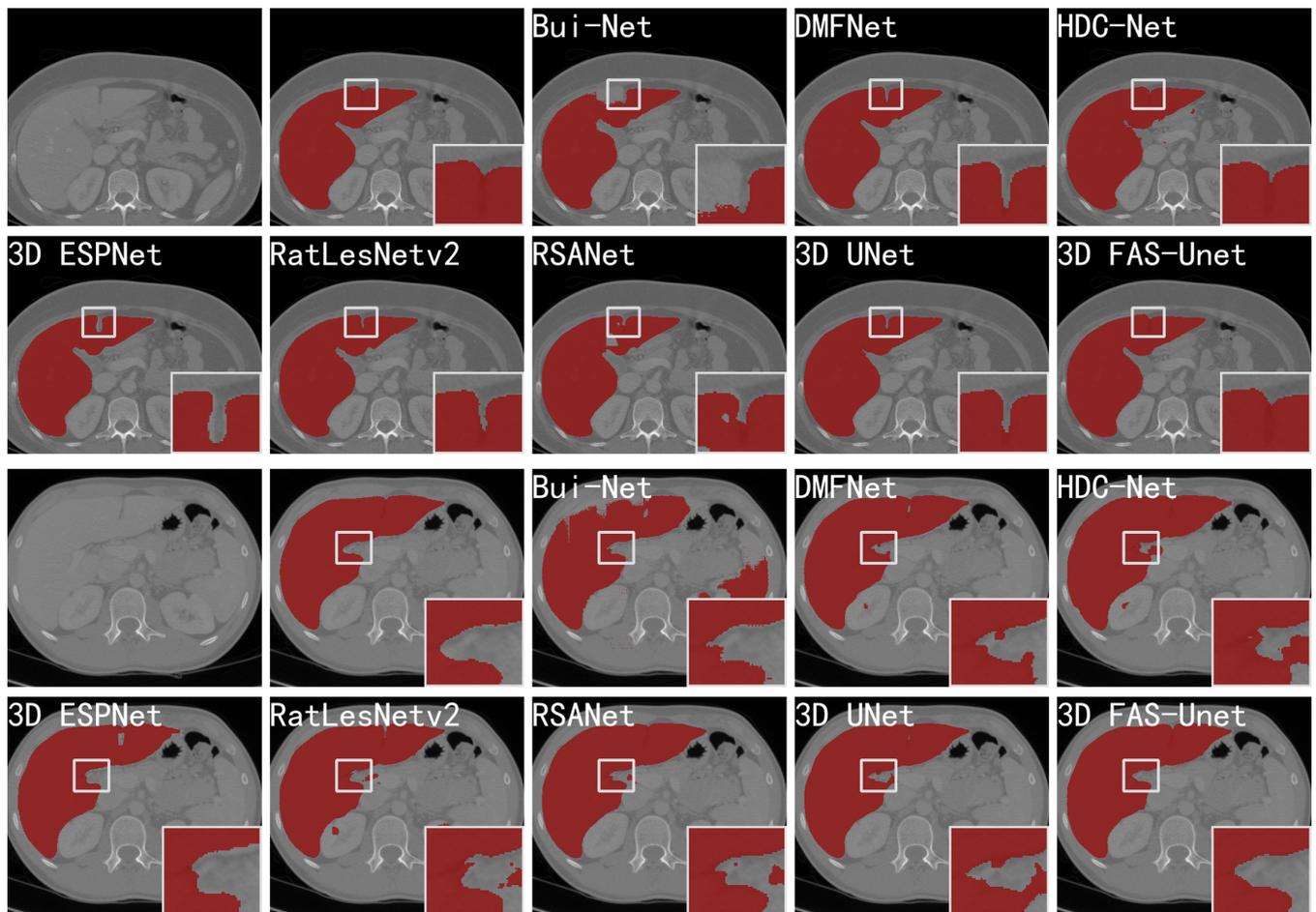
**Table 9.** Performance comparisons of liver segmentation between the different 3D networks with/without post-processing using the a-DSC, a-SSD, and a-Preci values on the 3D CHAOS-CT validation datasets. The best and second places are highlighted in bold font and underlined, respectively.

Method	Params	Without Post-Processing			With Post-Processing		
		a-DSC (%)	a-SSD (mm)	a-Preci (%)	a-DSC (%)	a-SSD (mm)	a-Preci (%)
3D UNet [46]	8.16 M	96.18 ± 1.72	5.65 ± 5.15	94.87 ± 4.12	<b>97.16 ± 0.54</b>	<u>1.23 ± 0.28</u>	96.71 ± 2.01
Bui-Net [51]	2.24 M	89.41 ± 4.86	11.54 ± 4.52	85.90 ± 8.20	92.50 ± 4.02	4.42 ± 3.60	91.85 ± 7.92
DMFNet [48]	3.87 M	96.41 ± 0.99	4.45 ± 2.98	95.34 ± 3.28	96.93 ± 0.47	1.29 ± 0.11	96.31 ± 2.42
3D ESPNet [53]	3.57 M	95.72 ± 0.81	4.32 ± 1.74	<b>97.46 ± 2.15</b>	96.15 ± 0.80	1.68 ± 0.67	<b>98.33 ± 1.31</b>
RSANet [50]	24.54 M	<u>96.61 ± 1.00</u>	<b>4.04 ± 2.30</b>	95.23 ± 2.70	97.08 ± 0.67	1.26 ± 0.33	96.11 ± 1.95
RatLesNetV2 [54]	<u>0.83 M</u>	95.73 ± 1.22	7.04 ± 3.12	95.13 ± 3.40	96.82 ± 0.69	1.27 ± 0.26	<u>97.23 ± 2.12</u>
HDC-Net [49]	<b>0.29 M</b>	95.30 ± 1.64	7.50 ± 4.92	93.77 ± 4.23	96.42 ± 0.67	1.30 ± 0.17	95.87 ± 2.51
3D FAS-UNet	1.00 M	<b>96.69 ± 0.76</b>	<u>4.04 ± 3.11</u>	<u>95.92 ± 2.78</u>	<u>97.11 ± 0.32</u>	<b>1.16 ± 0.13</b>	96.73 ± 2.08

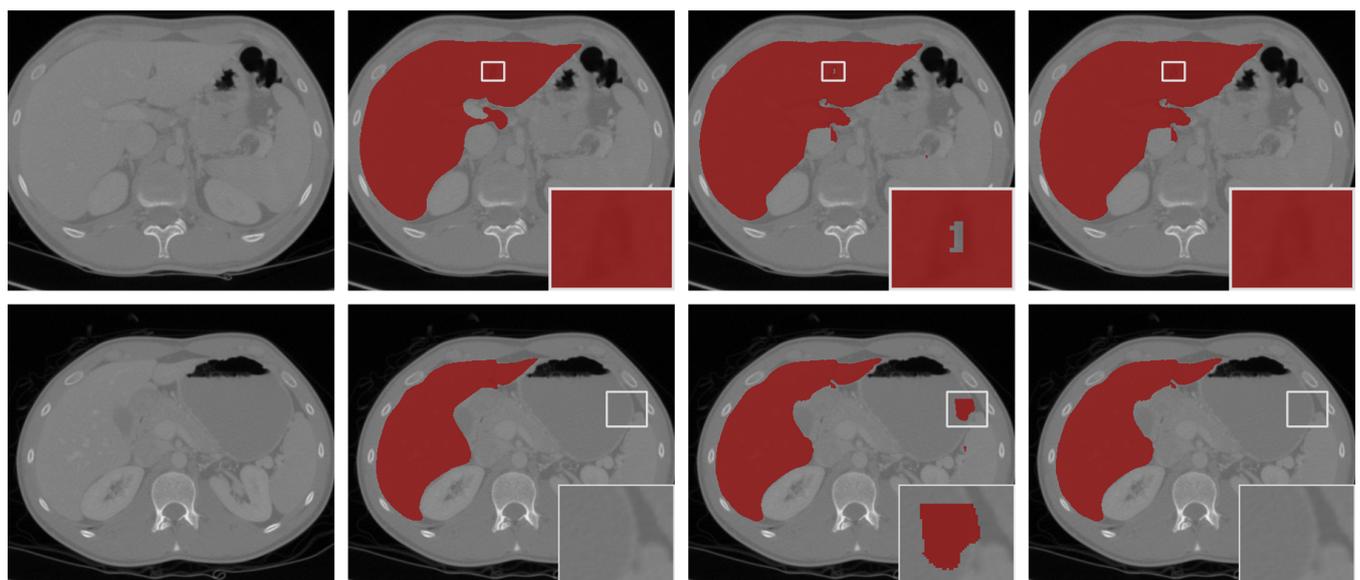
Figure 5 visualizes the prediction results of different networks on two slices of the CHAOS-CT datasets, and it can be clearly observed that the other networks highlighted more over-segmented regions outside the ground-truth compared with our network. Further, we can observe that the results for Bui-Net, DMFNet, HDC-Net, RatLesNetV2, and RSANet showed obvious segmentation noise in the background region. Although 3D ESP-Net did not present this phenomenon, the result showed an obvious “hole” inside of the liver, which was an obvious error-segmentation. However, our approach did not show these evident inaccurate results. In addition, most methods showed over-segmentation or under-segmentation on the boundaries of liver because it is very blurred in the CT image. From the zoom-in results of the first two rows of Figure 5, we can see that all mentioned methods had obvious under-segmentation except HDC-Net and our method, but our method had less noise in the background region. From the zoom-in results of the last two rows in Figure 5, we observe that many methods showed obvious over-segmentation on the liver boundaries. Only our method and 3D ESPNet achieved a good performance in this region, but 3D ESPNet extracted a “hole” in the liver region. In summary, compared to other methods, the boundary results obtained by our method were smoother, and the shape of the liver was more similar to the ground-truth.

We show visual comparisons before and after post-processing in Figure 6. The results demonstrated that the “hole” (the first row in Figure 6) was effectively filled, and the “island” (the second row in Figure 6) in the background was removed by post-processing. The experiments indicated that our model-driven approach with post-processing was more effective.

Qualitative and quantitative results demonstrated the effectiveness and advantages of the proposed method. Our method achieved a robust and accurate performance compared with the existing state-of-the-art methods in the 3D liver segmentation task, which can be applied to 3D medical image segmentation.



**Figure 5.** Visualizations (without post-processing) of different methods for liver CT segmentation of two slices. From left to right in the first and third rows: the original images, ground-truth, the segmentation results of skip-connected 3D DenseNet, DMFNet, and HDC-Net, respectively. From left to right in the second and fourth rows: the segmentation results of 3D ESPNet, RatLesNetv2, RSANet, 3D UNet, and 3D FAS-UNet, respectively.



**Figure 6.** The segmentation results with post-processing. From left to right: the selected scans of the validation set (first column), ground-truths (second column), segmentation results of 3D FAS-UNet without post-processing (third column), and results with post-processing (fourth column).

#### 4. Conclusions

In this work, we proposed a novel deep learning framework, FAS-UNet, for 2D and 3D medical image segmentation by enforcing some of the mathematical and physical laws (e.g., the convexity Mumford–Shah model and FAS algorithm), which focuses on learning the multiscale image features to generate the segmentation results.

Compared with other existing works that analyzed the connection between the multi-grid and CNN, FAS-UNet integrates medical image data and mathematical models and enhances the connection between data-driven and traditional variational model methodologies; it provides a helpful viewpoint for designing image segmentation network architectures. Compared with UNet, the proposed FAS-UNet introduces the concept of the data space, which exploits the model prior information to extract the features. Specifically, the feature extraction task leads to solving nonlinear equations, and an iterative scheme of numerical algorithms was designed to learn the features. Our experimental results showed that the proposed method is able to improve medical image segmentation in different tasks, including the segmentation of thoracic organs at risk, the whole-heart and great vessel, and liver segmentation. It is believed that the approach is a general one and can be applied to other image processing tasks, such as image denoising and image reconstruction. In addition, we found that the topological interaction module proposed by [55] can effectively improve the performance of many segmentation methods. Therefore, we will use this module in FAS-UNet to improve its performance in the future work.

**Author Contributions:** Conceptualization, H.Z., S.S. and J.Z.; methodology, H.Z., S.S. and J.Z.; software, H.Z.; validation, H.Z. and J.Z.; formal analysis, H.Z., S.S. and J.Z.; investigation, H.Z., S.S. and J.Z.; resources, H.Z., S.S. and J.Z.; data curation, H.Z.; writing—original draft preparation, H.Z. and J.Z.; writing—review and editing, H.Z., S.S. and J.Z.; visualization, H.Z.; supervision, S.S. and J.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (NSFC) under Grants 11971414 and 11771369 and partly by grants from the Natural Science Foundation of Hunan Province (Grants 2018JJ2375, 2018XK2304, and 2018WK4006).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The code and datasets are available at <https://github.com/zhuhui100/FASUNet> (accessed on 2 October 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [[CrossRef](#)] [[PubMed](#)]
2. Boveiri, H.R.; Khayami, R.; Javidan, R.; Mehdizadeh, A. Medical Image Registration Using Deep Neural Networks: A Comprehensive Review. *Comput. Electr. Eng.* **2020**, *87*, 106767. [[CrossRef](#)]
3. Cai, L.; Gao, J.; Zhao, D. A Review of the Application of Deep Learning in Medical Image Classification and Segmentation. *Ann. Transl. Med.* **2020**, *8*, 713. [[CrossRef](#)] [[PubMed](#)]
4. Chen, C.; Qin, C.; Qiu, H.; Tarroni, G.; Duan, J.; Bai, W.; Rueckert, D. Deep Learning for Cardiac Image Segmentation: A Review. *Front. Cardiovasc. Med.* **2020**, *7*, 25. [[CrossRef](#)]
5. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.; van Ginneken, B.; Sánchez, C.I. A Survey on Deep Learning in Medical Image Analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [[CrossRef](#)] [[PubMed](#)]
6. Miotto, R.; Wang, F.; Wang, S.; Jiang, X.; Dudley, J.T. Deep Learning for Healthcare: Review, Opportunities and Challenges. *Briefings Bioinform.* **2018**, *19*, 1236–1246. [[CrossRef](#)] [[PubMed](#)]
7. Fu, Y.; Lei, Y.; Wang, T.; Curran, W.J.; Liu, T.; Yang, X. Deep Learning in Medical Image Registration: A Review. *Phys. Med. Biol.* **2020**, *65*, 20TR01. [[CrossRef](#)]
8. Liu, X.; Song, L.; Liu, S.; Zhang, Y. A Review of Deep-Learning-Based Medical Image Segmentation Methods. *Sustainability* **2021**, *13*, 1224. [[CrossRef](#)]
9. Ronneberger, O.; Fischer, P.; Brox, T. UNet: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Springer: Cham, Switzerland, 2015.

10. Milletari, F.; Navab, N.; Ahmadi, S.A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571. [[CrossRef](#)]
11. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* **2019**, *39*, 1856–1867. [[CrossRef](#)]
12. Cicek, O.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. 3D UNet: Learning Dense Volumetric Segmentation from Sparse Annotation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2016: 19th International Conference, Athens, Greece, 17–21 October 2016*; Lecture Notes in Computer Science; Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 9901, pp. 424–432. [[CrossRef](#)]
13. Mehta, S.; Mercan, E.; Bartlett, J.; Weaver, D.; Elmore, J.G.; Shapiro, L. Y-Net: Joint Segmentation and Classification for Diagnosis of Breast Biopsy Images. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2018*; Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 893–901.
14. Xiao, X.; Lian, S.; Luo, Z.; Li, S. Weighted Res-UNet for High-Quality Retina Vessel Segmentation. In Proceedings of the 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, China, 19–21 October 2018; pp. 327–331. [[CrossRef](#)]
15. Valanarasu, J.M.J.; Sindagi, V.A.; Hacihaliloglu, I.; Patel, V.M. KiU-Net: Towards Accurate Segmentation of Biomedical Images Using Over-Complete Representations. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2020*; Lecture Notes in Computer Science; Martel, A.L., Abolmaesumi, P., Stoyanov, D., Mateus, D., Zuluaga, M.A., Zhou, S.K., Racoceanu, D., Joskowicz, L., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 12264, pp. 363–373. [[CrossRef](#)]
16. Li, X.; Chen, H.; Qi, X.; Dou, Q.; Fu, C.W.; Heng, P.A. H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes. *IEEE Trans. Med. Imaging* **2018**, *37*, 2663–2674. [[CrossRef](#)]
17. Isensee, F.; Jaeger, P.F.; Kohl, S.A.A.; Petersen, J.; Maier-Hein, K.H. nnU-Net: A Self-Configuring Method for Deep Learning-Based Biomedical Image Segmentation. *Nat. Methods* **2021**, *18*, 203–211. [[CrossRef](#)] [[PubMed](#)]
18. He, X.; Mo, Z.; Wang, P.; Liu, Y.; Yang, M.; Cheng, J. ODE-Inspired Network Design for Single Image Super-Resolution. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1732–1741.
19. Lu, Y.; Zhong, A.; Li, Q.; Dong, B. *Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations*; StockholmÅdssan: Stockholm Sweden, 2018; Volume 80, pp. 3276–3285.
20. Gu, R.; Wang, G.; Song, T.; Huang, R.; Aertsen, M.; Deprest, J.; Ourselin, S.; Vercauteren, T.; Zhang, S. CA-Net: Comprehensive attention convolutional neural networks for explainable medical image segmentation. *IEEE Trans. Med. Imaging* **2020**, *40*, 699–711. [[CrossRef](#)] [[PubMed](#)]
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. Larsson, G.; Maire, M.; Shakhnarovich, G. FractalNet: Ultra-Deep Neural Networks without Residuals. In Proceedings of the ICLR, Toulon, France, 24–26 April 2017.
23. Zhang, X.; Li, Z.; Change Loy, C.; Lin, D. Polynet: A pursuit of structural diversity in very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 718–726.
24. Gomez, A.N.; Ren, M.; Urtasun, R.; Grosse, R.B. The reversible residual network: Backpropagation without storing activations. *arXiv* **2017**, arXiv:1707.04585.
25. Chen, R.T.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. Neural ordinary differential equations. *arXiv* **2018**, arXiv:1806.07366.
26. Yang, Y.; Sun, J.; Li, H.; Xu, Z. Deep ADMM-Net for Compressive Sensing MRI. In Proceedings of the Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016; Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 10–18.
27. He, J.; Xu, J. MgNet: A unified framework of multigrid and convolutional neural network. *Sci. China Math.* **2019**, *62*, 1331–1354. [[CrossRef](#)]
28. Alt, T.; Schrader, K.; Augustin, M.; Peter, P.; Weickert, J. Connections between numerical algorithms for PDEs and neural networks. *arXiv* **2021**, arXiv:2107.14742.
29. Heyer, D.; Mausfeld, R. *Perception and the Physical World: Psychological and Philosophical Issues in Perception*; John Wiley & Sons, Ltd.: New York, NY, USA, 2002.
30. Cai, X.; Chan, R.; Zeng, T. A Two-Stage Image Segmentation Method Using a Convex Variant of the Mumford–Shah Model and Thresholding. *SIAM J. Imaging Sci.* **2013**, *6*, 368–390. [[CrossRef](#)]
31. Liu, C.; Ng, M.K.P.; Zeng, T. Weighted Variational Model for Selective Image Segmentation with Application to Medical Images. *Pattern Recognit.* **2018**, *76*, 367–379. [[CrossRef](#)]
32. Ma, Q.; Peng, J.; Kong, D. Image Segmentation via Mean Curvature Regularized Mumford–Shah Model and Thresholding. *Neural Process. Lett.* **2018**, *48*, 1227–1241. [[CrossRef](#)]
33. McCormick, S.F. *Multigrid Methods*; SIAM: Philadelphia, PA, USA, 1987.

34. Mumford, D.B.; Shah, J. Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **1989**, *42*, 577–685. [[CrossRef](#)]
35. Zhou, D.X. Universality of deep convolutional neural networks. *Appl. Comput. Harmon. Anal.* **2020**, *48*, 787–794. [[CrossRef](#)]
36. Taha, A.A.; Hanbury, A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med. Imaging* **2015**, *15*, 29. [[CrossRef](#)] [[PubMed](#)]
37. Dice, L.R. Measures of the Amount of Ecologic Association Between Species. *Ecology* **1945**, *26*, 297–302. [[CrossRef](#)]
38. Lambert, Z.; Petitjean, C.; Dubray, B.; Kuan, S. SegTHOR: Segmentation of Thoracic Organs at Risk in CT images. In Proceedings of the 2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA), Paris, France, 9–12 November 2020; pp. 1–6. [[CrossRef](#)]
39. Pace, D.F.; Dalca, A.V.; Geva, T.; Powell, A.J.; Moghari, M.H.; Golland, P. Interactive Whole-Heart Segmentation in Congenital Heart Disease. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Springer: Cham, Switzerland, 2015; Volume 9351, pp. 80–88.
40. Kavur, A.E.; Gezer, N.S.; Barış, M.; Aslan, S.; Conze, P.H.; Groza, V.; Pham, D.D.; Chatterjee, S.; Ernst, P.; Özkan, S.; et al. CHAOS challenge-combined (CT-MR) healthy abdominal organ segmentation. *Med. Image Anal.* **2021**, *69*, 101950. [[CrossRef](#)] [[PubMed](#)]
41. Kavur, A.E.; Selver, M.A.; Dicle, O.; Barış, M.; Gezer, N.S. CHAOS—Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge Data. *Zenodo* **2019**. [[CrossRef](#)]
42. Gu, Z.; Cheng, J.; Fu, H.; Zhou, K.; Hao, H.; Zhao, Y.; Zhang, T.; Gao, S.; Liu, J. CE-Net: Context Encoder Network for 2D Medical Image Segmentation. *IEEE Trans. Med. Imaging* **2019**, *38*, 2281–2292. [[CrossRef](#)]
43. Feng, S.; Zhao, H.; Shi, F.; Cheng, X.; Wang, M.; Ma, Y.; Xiang, D.; Zhu, W.; Chen, X. CPFNet: Context Pyramid Fusion Network for Medical Image Segmentation. *IEEE Trans. Med. Imaging* **2020**, *39*, 3008–3018. [[CrossRef](#)]
44. Romera, E.; Alvarez, J.M.; Bergasa, L.M.; Arroyo, R. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 263–272. [[CrossRef](#)]
45. Chaurasia, A.; Culurciello, E. Linknet: Exploiting encoder representations for efficient semantic segmentation. In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; pp. 1–4.
46. Cicek, O.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*; Springer: Cham, Switzerland, 2016; pp. 424–432.
47. Zhu, W.; Huang, Y.; Zeng, L.; Chen, X.; Liu, Y.; Qian, Z.; Du, N.; Fan, W.; Xie, X. AnatomyNet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. *Med. Phys.* **2019**, *46*, 576–589. [[CrossRef](#)]
48. Chen, C.; Liu, X.; Ding, M.; Zheng, J.; Li, J. 3D Dilated Multi-Fiber Network for Real-time Brain Tumor Segmentation in MRI. In Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Shenzhen, China, 13–17 October 2019.
49. Luo, Z.; Jia, Z.; Yuan, Z.; Peng, J. HDC-Net: Hierarchical decoupled convolution network for brain tumor segmentation. *IEEE J. Biomed. Health Inform.* **2020**, *25*, 737–745. [[CrossRef](#)]
50. Zhang, H.; Zhang, J.; Zhang, Q.; Kim, J.; Zhang, S.; Gauthier, S.A.; Spincemaille, P.; Nguyen, T.D.; Sabuncu, M.; Wang, Y. Rsanet: Recurrent slice-wise attention network for multiple sclerosis lesion segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Shenzhen, China, 13–17 October 2019; pp. 411–419.
51. Bui, T.D.; Shin, J.; Moon, T. Skip-connected 3D DenseNet for volumetric infant brain MRI segmentation. *Biomed. Signal Process. Control* **2019**, *54*, 101613. [[CrossRef](#)]
52. Chen, H.; Dou, Q.; Yu, L.; Qin, J.; Heng, P.A. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *NeuroImage* **2018**, *170*, 446–455. [[CrossRef](#)]
53. Nuechterlein, N.; Mehta, S. 3D-ESPNet with pyramidal refinement for volumetric brain tumor image segmentation. In Proceedings of the International MICCAI Brainlesion Workshop, Granada, Spain, 16 September 2018; pp. 245–253.
54. Valverde, J.M.; Shatillo, A.; De Feo, R.; Gröhn, O.; Sierra, A.; Tohka, J. Ratlesnetv2: A fully convolutional network for rodent brain lesion segmentation. *Front. Neurosci.* **2020**, *14*, 610239. [[CrossRef](#)] [[PubMed](#)]
55. Gupta, S.; Hu, X.; Kaan, J.; Jin, M.; Mpoy, M.; Chung, K.; Singh, G.; Saltz, M.; Kurc, T.; Saltz, J.; et al. Learning Topological Interactions for Multi-Class Medical Image Segmentation. *arXiv* **2022**, arXiv:2207.09654.