Deconver: A Deconvolutional Network for Medical Image Segmentation

Pooya Ashtari, Shahryar Noei, Fateme Nateghi Haredasht, Jonathan H. Chen, Giuseppe Jurman, Aleksandra Pižurica, *Senior Member, IEEE*, and Sabine Van Huffel

***Abstract*— While convolutional neural networks (CNNs) and vision transformers (ViTs) have advanced medical image segmentation, they face inherent limitations such as local receptive fields in CNNs and high computational complexity in ViTs. This paper introduces Deconver, a novel network that integrates traditional deconvolution techniques from image restoration as a core learnable component within a U-shaped architecture. Deconver replaces computationally expensive attention mechanisms with efficient nonnegative deconvolution (NDC) operations, enabling the restoration of high-frequency details while suppressing artifacts. Key innovations include a backpropagation-friendly NDC layer based on a provably monotonic update rule and a parameter-efficient design. Evaluated across five datasets (ISLES'22, Spleen, BraTS'23, GlaS, and FIVES) covering both 2D and 3D segmentation tasks, Deconver achieves state-of-the-art performance in Dice scores and Hausdorff distance while reducing computational costs (FLOPs) by up to 90% compared to leading baselines. By bridging traditional image restoration with deep learning, this work offers a practical solution for high-precision segmentation in resource-constrained clinical workflows.**

***Index Terms*— Deconvolution, Medical Image Segmentation, U-Net.**

The research was partially funded by the Flanders AI Research Program and the National Plan for Complementary Investments to the NRRP (D34H project, code: PNC0000001).

Pooya Ashtari is with the Department of Electrical Engineering (ESAT), STADIUS Center, KU Leuven, Belgium, and with the Department of Telecommunications and Information Processing, Ghent University, B-9000 Gent, Belgium (e-mail: pooya.ashtari@esat.kuleuven.be, pooya.ashtari@ugent.be). Corresponding author.

Shahryar Noei is with the Data Science for Health Unit, Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento, Italy (e-mail: snoei@fbk.eu).

Fateme Nateghi Haredasht is with the Center for Biomedical Informatics Research, Stanford University, Stanford, CA, USA (e-mail: fnateghi@stanford.edu).

Jonathan H. Chen is with the Center for Biomedical Informatics Research and with the Division of Hospital Medicine, Stanford University, Stanford, CA, USA (e-mail: jonc101@stanford.edu).

Giuseppe Jurman is with the Data Science for Health Unit, Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento, Italy and the Department of Biomedical Sciences, Humanitas University, Via Rita Levi Montalcini, 4, 20072 Pieve Emanuele MI (e-mail: giuseppe.jurman@fbk.eu).

Aleksandra Pižurica is with the Department of Telecommunications and Information Processing, Ghent University, B-9000 Gent, Belgium (e-mail: aleksandra.pizurica@ugent.be).

Sabine Van Huffel is with the Department of Electrical Engineering (ESAT), STADIUS Center, KU Leuven, Belgium and with Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium (e-mail: sabine.vanhuffel@esat.kuleuven.be).

Pooya Ashtari and Shahryar Noei contributed equally to this work.

The project is available at https://github.com/pashtari/deconver.

I. Introduction

Medical image segmentation is a fundamental task in modern healthcare, enabling precise delineation of anatomical structures and pathological regions essential for computer-assisted diagnosis, treatment planning, and surgical guidance. Despite advancements, achieving accurate segmentation remains challenging due to inherent complexities of medical images, such as low contrast, heterogeneous textures, and acquisition artifacts such as motion blur or noise.

Convolutional Neural Networks (CNNs), particularly U-Net [1] and its variants, have dominated medical image segmentation due to their ability to hierarchically extract spatial features. Extensions like 3D U-Net [2] and nnU-Net [3] further improved performance by adapting to volumetric data and automating architecture configurations. However, CNNs are inherently limited by their local receptive fields, hindering their ability to model long-range spatial dependencies, often critical for segmenting anatomically dispersed or structurally complex regions.

Recent efforts to address this limitation include enlarging kernel sizes [4] or adopting Vision Transformers (ViTs). ViT-based models like nnFormer [5] and MISSFormer [6] excel at capturing global context via self-attention but suffer from quadratic computational complexity relative to input resolution. This restricts their practicality in high-resolution medical imaging and resource-constrained clinical environments. Hybrid architectures, such as TransUNet [7] and Swin UNETR [8], attempt to balance locality and globality by combining convolutional and self-attention layers but often come at the cost of increased architectural complexity. To address this issue, more recently, SimPoolFormer [9] replaces the computationally intensive multi-headed self-attention in visual transformers with a lightweight SimPool operation, complemented by a vision multilayer perceptron stream to enhance efficiency. Similarly, MGCET [10] integrates MLP-mixer blocks with graph convolutional modules to improve representation power for hyperspectral image classification. While not originally designed for medical segmentation, these models illustrate the broader trend of replacing costly attention mechanisms with efficient alternatives.

Deconvolution is a classical technique in image processing widely used for deblurring and image restoration through methods such as Wiener filtering [11] and the Richardson-Lucy algorithm [12], the latter of which iteratively refines

estimates of latent sources under physical constraints such as nonnegativity. While effective as a pre-processing step in traditional pipelines, the integration of deconvolution into deep learning frameworks remains underexplored, where it could synergize data-driven feature learning with the image enhancement capability for improved segmentation performance.

In this work, we propose **Deconver**, a novel segmentation network that integrates deconvolution as a core learnable layer within a U-shaped architecture. Our key insight is to replace computationally expensive attention mechanisms with efficient deconvolution operations, enabling the restoration of high-frequency details while suppressing artifacts. The main contributions of this work are threefold:

- **Architectural Innovation:** Deconver is the first network to incorporate deconvolution principles as a learnable component within a deep architecture.
- **Nonnegative deconvolution layer:** We introduce a backpropagation-friendly, differentiable layer based on a provably monotonic update rule for nonnegative deconvolution, enabling stable end-to-end training using current deep learning frameworks.
- **Performance and efficiency:** Deconver achieves state-of-the-art performance on both 2D and 3D segmentation tasks with substantially fewer computational costs and parameters than leading baselines.

Extensive experiments across five datasets (ISLES'22, Spleen, BraTS'23, GlaS, and FIVES) demonstrate Deconver's superiority in Dice scores and boundary accuracy (Hausdorff distance). By bridging classical image restoration with modern deep learning, Deconver provides a practical solution for high-precision segmentation, particularly in resource-constrained clinical workflows.

II. Related Work

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have played a central role in medical image segmentation, primarily due to their ability to extract hierarchical spatial features. The introduction of U-Net [1] established an encoder–decoder architecture with skip connections that has since become the backbone of many segmentation models. Variants have since emerged to improve performance across different settings: 3D U-Net [2] extends U-Net to volumetric data using 3D operations; UNet++ [13] incorporates nested dense skip connections; and nnU-Net [3] presents a self-adapting framework capable of configuring itself to a wide range of tasks. Other notable CNN-based advances include SegResNet [14], a residual U-Net variant that won the Brain Tumor Segmentation Challenge (BraTS) 2018.

Despite their effectiveness, a key limitation of CNNs is their inherently local receptive field, which restricts their capacity to capture long-range spatial dependencies. This poses challenges for segmenting anatomically complex or spatially dispersed structures. Two major approaches have emerged to address this issue. One involves using large kernels, as seen in MedNeXt [15], which expands the receptive field by iteratively increase kernel sizes by upsampling small kernel networks. The other approach incorporates attention mechanisms to explicitly model global context, paving the way for Transformer-based and hybrid segmentation architectures.

B. Transformers

Originally introduced for natural language processing, Transformer architectures have been successfully adapted to computer vision tasks through Vision Transformers (ViTs) [16], which model images as sequences of patch tokens. Fully Transformer-based segmentation models incorporate self-attention mechanisms in both the encoder and decoder, offering a shift from traditional convolution-based designs.

Among these, nnFormer [5] proposes an interleaved architecture that combines local and global self-attention layers with convolutional downsampling. MISSFormer [6] builds a hierarchical Transformer-based encoder-decoder tailored for medical image segmentation, enhancing both local precision and long-range contextual reasoning. Self-attention mechanisms are known to be computationally intensive, especially on long sequences, which can hinder their scalability in high-resolution medical imaging tasks. One solution has been proposed in [17] by replacing attention with non-negative matrix factorization (NMF) which was shown to reduce the computational cost significantly while maintaining high performance. Other approaches include using hybrid models.

C. Hybrid Models

To balance the strengths of convolutional and attention-based methods, hybrid architectures have emerged as a practical solution. These models typically use a Transformer in the encoder and adopt a CNN-based decoder. One of the earliest hybrid models in medical imaging, TransUNet [7], incorporates a ViT encoder into the bridge of a U-Net. Extending this work, UNETR [18] employs a full Transformer-based encoder directly connected to a convolutional decoder via skip connections. Swin UNETR [8] further improves upon this by replacing the ViT encoder with Swin Transformer blocks [19], introducing a hierarchical structure that models both local and global dependencies efficiently across scales. While hybrid models are effective they still come with increased complexity and computational demands.

D. Deconvolution

Deconvolution is a fundamental technique in image processing aimed at reversing the effects of blurring and restoring an image closer to its original form [20]. Early methods such as Wiener deconvolution [11] applied frequency-domain filtering to restore degraded signals, while iterative approaches like the Richardson-Lucy algorithm [12, 21] refined the image estimate through successive likelihood-based updates.

Deconvolution methods are widely used in medical imaging to reduce blurring and enhance resolution across modalities. In fluorescence microscopy, they help restore fine cellular structures from blurred images [22], while in magnetic resonance imaging (MRI) [23], computed tomography (CT) [24, 25], and positron emission tomography (PET) [26], they improve image

clarity and diagnostic utility. Traditional pipelines, however, treat restoration and segmentation as separate steps, each optimized independently. This approach fails to leverage the potential synergy between these tasks. Therefore, We propose an end-to-end architecture incorporating deconvolution-inspired layers for holistic segmentation, enabling the restoration process to be optimized specifically for segmentation performance rather than general image quality metrics.

III. Methods

A. Notation

We denote vectors by boldface lowercase letters (e.g., $\boldsymbol{x}$), matrices by boldface uppercase letters (e.g., $\boldsymbol{X}$), and tensors by boldface calligraphic letters (e.g., $\boldsymbol{\mathcal{X}}$). For clarity, a 2D input image is represented as a 3D tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{C\times H\times W}$, where C denotes the number of channels, and H and W represent the spatial height and width. A 3D volumetric input is represented as a 4D tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{C\times H\times W}$, with D denoting depth. Individual elements in a tensor are accessed via indices matching its dimensions, such as $\boldsymbol{\mathcal{X}}[c,h,w]$ for a 3D tensor or $\boldsymbol{\mathcal{X}}[c,h,w,d]$ for a 4D tensor.

The inner product between two tensors $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ of identical dimensions is denoted by

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_{i_1,\ldots,i_N} \boldsymbol{\mathcal{X}}[i_1,\ldots,i_N]\ \boldsymbol{\mathcal{Y}}[i_1,\ldots,i_N],$$

where N is the number of tensor dimensions. The Frobenius norm is defined as $\|\boldsymbol{\mathcal{X}}\|_{\mathrm{F}} = \sqrt{\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle}$.

B. Revisiting Deconvolution

Deconvolution is a fundamental technique in image processing that seeks to reverse the effects of convolution to restore images degraded by blurring. In medical image segmentation, deconvolution is particularly valuable for enhancing fine anatomical details and mitigating acquisition artifacts. This enables more precise delineation of structures such as tumors and blood vessels by recovering high-frequency components often lost during image acquisition.

a) Problem Formulation: Let a 2D input image be represented as $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{C\times H\times W}$, where C denotes the number of channels, and H and W represent the height and width of the image, respectively. The objective is to recover the latent *source image* $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{E\times H\times W}$ from the observed $\boldsymbol{\mathcal{X}}$, given a known *filter tensor* $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{C\times E\times M'\times N'}$, where E represents the number of channels of the source image, and $M' = 2M+1$ and $N' = 2M+1$ are the spatial dimensions of the filter. Deconvolution aims to find the optimal tensor $\boldsymbol{\mathcal{S}}$ that best approximates the observed data $\boldsymbol{\mathcal{X}}$ as the cross-correlation of $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{V}}$, i.e., $\boldsymbol{\mathcal{X}} \approx \hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}$, defined as

$$\hat{\boldsymbol{\mathcal{X}}}[c,h,w] \triangleq \sum_{e=0}^{E-1}\sum_{m=0}^{2M}\sum_{n=0}^{2N} \boldsymbol{\mathcal{S}}_p[e,h+m,w+n]\boldsymbol{\mathcal{V}}[c,e,m,n], \tag{1}$$

for $c \in \{0,\ldots,C-1\}$, $h \in \{0,\ldots,H-1\}$, and $w \in \{0,\ldots,W-1\}$. Here, $\boldsymbol{\mathcal{S}}_p = \mathrm{pad}(\boldsymbol{\mathcal{S}},(M,N)) \in \mathbb{R}^{E\times(H+2M)\times(W+2N)}$ denotes the zero-padded source image, ensuring to preserve spatial dimensions post-filtering (note that this definition aligns with CNN conventions, applying the filter without flipping). This formulation extends standard convolution by allowing multi-channel inputs and outputs, making it suitable for modern deep learning architectures.

b) Nonnegative Deconvolution (NDC): In this work, we focus on the *nonnegative deconvolution (NDC)*, where $\boldsymbol{\mathcal{X}} \geq 0$, $\boldsymbol{\mathcal{V}} \geq 0$, and $\boldsymbol{\mathcal{S}} \geq 0$. The nonnegativity constraint aligns with the physical nature of medical imaging systems, where intensities are inherently positive, helping suppress negative artifacts that could mislead segmentation models. The goal is to estimate the source image $\boldsymbol{\mathcal{S}} \geq 0$ by minimizing the reconstruction error:

$$\begin{aligned} \underset{\boldsymbol{\mathcal{S}}}{\text{minimize}} \quad & \mathcal{E}(\boldsymbol{\mathcal{S}}) = \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}\|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \boldsymbol{\mathcal{S}} \geq 0, \end{aligned} \tag{2}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. This formulation implicitly assumes additive Gaussian noise, which approximates complex noise in many imaging modalities while ensuring computational tractability.

c) Multiplicative Update Rule: To address problem (2), we can derive an iterative update rule inspired by Richardson-Lucy algorithm [12] and nonnegative matrix factorization [27]. Starting with an initial guess $\boldsymbol{\mathcal{S}}^{(0)} \geq 0$, the source image at iteration $t+1$ is updated as:

$$\boldsymbol{\mathcal{S}}^{(t+1)} = \boldsymbol{\mathcal{S}}^{(t)} \odot \frac{\boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{V}}^{-}}{(\boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}}) * \boldsymbol{\mathcal{V}}^{-}}, \tag{3}$$

where $\odot$ denotes element-wise multiplication, $\boldsymbol{\mathcal{V}}^{-}$ is the *adjoint filter*, which is transposed and spatially flipped (i.e., $\boldsymbol{\mathcal{V}}^{-}[d,c,m,n] = \boldsymbol{\mathcal{V}}[c,d,2M-m,2N-n]$), and the division is element-wise. The numerator correlates residuals with the filter, amplifying regions where $\boldsymbol{\mathcal{S}}$ underestimates $\boldsymbol{\mathcal{X}}$, while the denominator normalizes the update to prevent overshooting. This multiplicative form inherently preserves nonnegativity when $\boldsymbol{\mathcal{S}}^{(0)} \geq 0$.

d) Monotonicity: The key advantage of the multiplicative update (3) is its guarantee of the reduction of the reconstruction error, which we will prove in theorem 1.

Theorem 1 (Monotonicity): Let $\boldsymbol{\mathcal{S}}^{(t)}$ be the source image at iteration t. With a nonnegative initial source $\boldsymbol{\mathcal{S}}^{(0)} \geq 0$ and under the update (3), the reconstruction error $e^{(t)} \triangleq \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}}\|_{\mathrm{F}}^2$ is non-increasing, i.e., $e^{(t+1)} \leq e^{(t)}$ for all $t \geq 0$.

Proof: See Appendix for a detailed proof. ∎

The update rule (3) also generalizes naturally to 3D volumes, making it suitable for modalities like MRI and CT. In Section III-F, we integrate this deconvolution technique as a learnable layer within a deep neural network, enhancing multi-scale feature maps to improve segmentation performance.

C. Overall Architecture

The Deconver architecture adopts a U-shaped structure (see Fig. 1), comprising an encoder and decoder with skip connections in between at equal resolutions. Given a 2D input image $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{C_{in}\times H\times W}$ with C_{in} input channels and spatial dimensions (H,W), the network outputs a *logit* map of shape (C_{out},H,W), where C_{out} denotes the number of target classes.

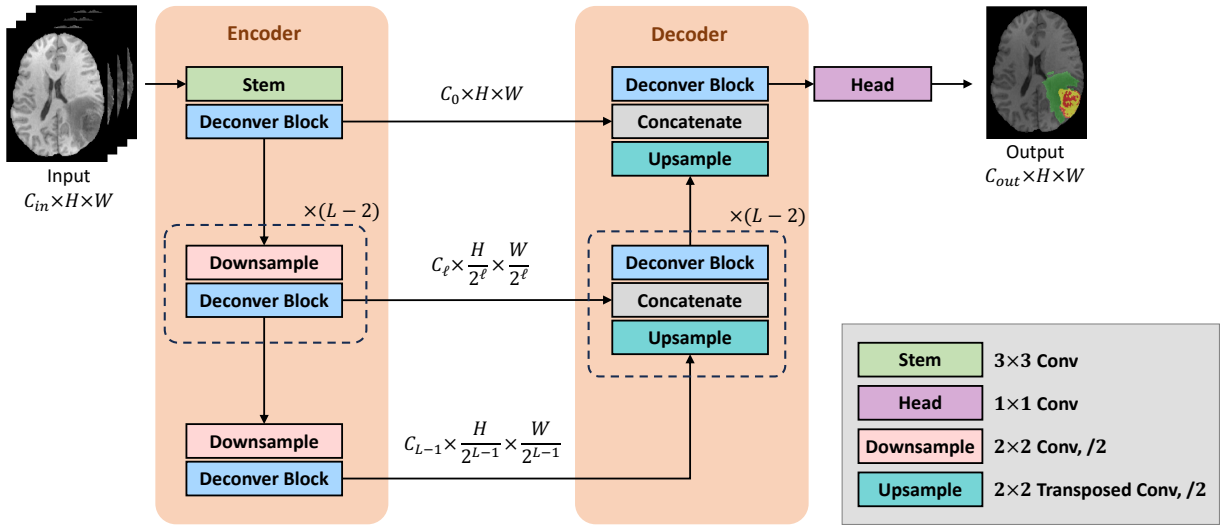
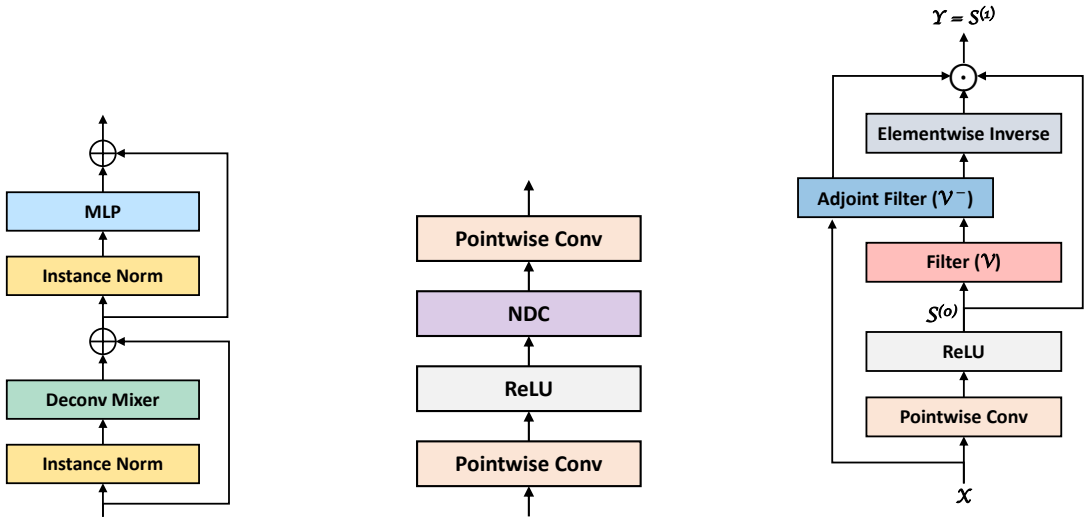


Fig. 1: Overview of Deconver architecture.



(a) Deconver block (b) Deconv Mixer (c) NDC layer

Fig. 2: Overview of Deconver block and its components.

The encoder consists of L stages, each containing a *Deconver block* described in Section III-D. At the initial stage, a *stem* layer increases the input channels to C_0 (typically 32 or 64, depending on the dataset) using a single convolutional layer with a kernel size of $(3, 3)$. Subsequent stages downsample feature maps using strided convolutions (stride=2), halving the spatial dimensions while doubling the channel count until a maximum of 512 channels is reached. Formally, the number of output channels at stage ℓ is set to $C_\ell = \min(C_0 \times 2^\ell, 512)$. This design balances computational efficiency with the capacity to learn abstract, high-level representations. As the encoder deepens, the growing receptive field enables the capture of global contextual relationships essential for distinguishing semantically similar but spatially distant structures.

The decoder mirrors the encoder's hierarchical structure but reverses the spatial reduction via transposed convolutions (stride=2) to upsample feature maps. At each stage, the decoder incorporates skip connections that concatenate upsampled features with their encoder counterparts at corresponding resolutions. These skip connections mitigate information loss during downsampling and enhance feature reuse, ensuring more precise localization of fine-grained structures. At the deepest decoder layer, a pointwise convolution head (1×1 kernel) generates the final *logit* map, which can be activated via sigmoid or softmax to produce class probabilities for segmentation.

D. Deconver Block

A Deconver block forms the main building unit of the proposed model. In contrast to Vision Transformer (ViT) blocks [16] that rely on attention mechanism, our Deconver block replaces the multi-head self-attention module with a

learnable *Deconv Mixer* module (presented in Section III-E), and substitutes layer normalization with instance normalization [28] to better accommodate the small batch sizes often used with 3D or high-resolution medical images.

As illustrated in Fig. 2a, the block consists of two sequential sub-modules: *Deconv Mixer* and Multi-Layer Perceptron (MLP). Each sub-module is preceded by instance normalization and followed by a residual connection. Formally, given an intermediate feature map $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{C_\ell \times H \times W}$ at stage ℓ, the block's operations are defined as

$$\begin{aligned} \boldsymbol{\mathcal{Z}} &= \text{DeconvMixer}(\text{InstanceNorm}(\boldsymbol{\mathcal{X}})) + \boldsymbol{\mathcal{X}}, \\ \boldsymbol{\mathcal{Y}} &= \text{MLP}(\text{InstanceNorm}(\boldsymbol{\mathcal{Z}})) + \boldsymbol{\mathcal{Z}}, \end{aligned} \tag{4}$$

where the MLP is composed of two pointwise convolution layers separated by a Gaussian Error Linear Unit (GELU) activation:

$$\text{MLP}(\boldsymbol{\mathcal{X}}) = \text{PointwiseConv}(\text{GELU}(\text{PointwiseConv}(\boldsymbol{\mathcal{X}}))). \tag{5}$$

The MLP expands the channel dimension by a factor of α before projecting back to the original dimension, enabling nonlinear interaction across channels while preserving spatial structure.

E. Deconv Mixer

As illustrated in Fig. 2b, the Deconv Mixer module processes input features through three sequential stages: an initial pointwise convolution, an *NDC layer*, and a final pointwise convolution.

First, Deconv Mixer applies a pointwise convolution to linearly project each position. The output is then passed through a ReLU activation function, enforcing nonnegativity to ensure compatibility with the subsequent *NDC layer* (described in Section III-F). This nonnegative feature map is then processed by the NDC layer, which captures spatial dependencies and restores high-frequency details that may have been lost in previous layers. Finally, a second pointwise convolution is applied to produce the output. Formally, given an input feature map $\boldsymbol{\mathcal{X}}$, the Deconv Mixer can be expressed as:

$$\begin{aligned} \boldsymbol{\mathcal{X}}^1 &= \text{PointwiseConv}(\boldsymbol{\mathcal{X}}), \\ \boldsymbol{\mathcal{X}}^2 &= \text{NDC}(\text{ReLU}(\boldsymbol{\mathcal{X}}^1)), \\ \text{DeconvMixer}(\boldsymbol{\mathcal{X}}) &= \text{PointwiseConv}(\boldsymbol{\mathcal{X}}^2), \end{aligned} \tag{6}$$

where the intermediate tensors $\boldsymbol{\mathcal{X}}^1$ and $\boldsymbol{\mathcal{X}}^2$, and the final output share the same shape as the input $\boldsymbol{\mathcal{X}}$.

F. Nonnegative Deconvolution Layer

The nonnegative deconvolution (NDC) layer forms the core innovation of Deconver, incorporating nonnegative deconvolution (presented in Section III-B) as a learnable layer to enhance feature representations.

The NDC layer operates in a grouped manner, partitioning an input feature map $\boldsymbol{\mathcal{X}} \in \mathbb{R}_{\geq 0}^{C \times H \times W}$ into G groups $\{\boldsymbol{\mathcal{X}}_g\}_{g=1}^{G}$ along the channel dimension. Each group $\boldsymbol{\mathcal{X}}_g \in \mathbb{R}_{\geq 0}^{C_g \times H \times W}$, where $C_g = C/G$, is processed independently, maintaining its own learnable filter $\boldsymbol{\mathcal{V}}_g \geq 0$ and source image $\boldsymbol{\mathcal{S}}_g \geq 0$.

The layer introduces a source channel ratio R, defined as $R = E/C$, where E denotes the number of source channels. This ratio controls the channel expansion of the source image relative to the input.

The initial source image $\boldsymbol{\mathcal{S}}_g^{(0)} \in \mathbb{R}_{\geq 0}^{RC_g \times H \times W}$ is derived from the input feature map $\boldsymbol{\mathcal{X}}$ through a pointwise convolution followed by ReLU activation (See Fig. 2c). This ensures nonnegativity while providing an adaptive and learnable initialization of the source. The filter $\boldsymbol{\mathcal{V}}_g \in \mathbb{R}_{\geq 0}^{C_g \times RC_g \times M' \times N'}$ is initialized using the Kaiming uniform distribution [29] and clamped to nonnegative values via ReLU before being plugged into the update rule. Note that the filters $\boldsymbol{\mathcal{V}}_g$ are not fixed; they are treated as learnable parameters optimized jointly with the other network parameters during training.

The NDC layer applies a single iteration of the multiplicative update rule (3) to refine the source image. For computational efficiency, we empirically found one iteration sufficient to achieve a good trade-off between accuracy and computational cost. The update for group g is:

$$\boldsymbol{\mathcal{S}}_g^{(1)} = \boldsymbol{\mathcal{S}}_g^{(0)} \odot \frac{\boldsymbol{\mathcal{X}}_g * \boldsymbol{\mathcal{V}}_g^- + \epsilon}{\left(\boldsymbol{\mathcal{S}}_g^{(0)} * \boldsymbol{\mathcal{V}}_g\right) * \boldsymbol{\mathcal{V}}_g^- + \epsilon}, \tag{7}$$

where $\epsilon = 10^{-8}$ is a small positive constant to avoid division by zero. The numerator amplifies regions where the source underestimates the input, while the denominator normalizes the update to prevent overshooting. The final output is the channel-wise concatenation of all group outputs $\{\boldsymbol{\mathcal{S}}_g^{(1)}\}_{g=1}^{G}$, preserving spatial resolution and expanding channel dimensions by R. The NDC layer is fully differentiable and backpropagation-friendly, enabling end-to-end training using modern deep learning frameworks. Additionally, the filter $\boldsymbol{\mathcal{V}}_g$ and its adjoint $\boldsymbol{\mathcal{V}}_g^-$ share the same learnable parameters, reducing parameter overhead and potentially improving generalization performance.

IV. Experiments

This section details the experimental setup, comparative analyses, and ablation studies conducted to assess Deconver's effectiveness in 2D and 3D, as well as binary and multi-class medical image segmentation tasks. We evaluate its performance across five datasets, benchmark it against state-of-the-art baselines, and analyze the impact of key architectural design choices.

A. Experimental Setup

1) Datasets: We evaluated Deconver on five publicly available datasets, covering both 3D (ISLES'22, Spleen, and BraTS'23) and 2D (GlaS and FIVES) medical imaging modalities:

- **ISLES'22** [30]: This dataset includes 250 multi-center MRI scans, targeting ischemic stroke lesions via diffusion-weighted imaging (DWI) and apparent diffusion coefficient (ADC) maps. We excluded FLAIR images to simplify the pipeline and avoid registration challenges.
- **Spleen** [31]: This dataset comprises contrast-enhanced abdominal CT volumes with manual spleen annotations

from Medical Segmentation Decathlon. In our experiments, we used only the official 41 training subset and did not use the test set.

- **BraTS'23** [32–34]: This dataset consists of 1,251 multiparametric MRI (mpMRI) scans, including native T1-weighted, post-Gadolinium T1-weighted, T2-weighted, and FLAIR sequences. Ground truth labels delineate three tumor subregions: enhancing tumor (ET), tumor core (TC), and whole tumor (WT).
- **GlaS** [35, 36]: Containing 165 high-resolution H&E-stained colorectal histopathology images, this dataset features expert-annotated gland segmentations.
- **FIVES** [37]: This dataset includes 800 high-resolution fundus photographs with manual segmentation of retinal blood vessels.

2) Baseline Models: We compare Deconver against several state-of-the-art baseline models, including CNNs such as nnU-Net [3] and SegResNet [14]; hybrid convolution-transformer architectures like UNETR [18] and Swin UNETR [8]; and Factorizer [17], which leverages non-negative matrix factorization (NMF). These baselines represent diverse approaches to medical image segmentation.

3) Implementation Details: All models were implemented using PyTorch and the MONAI framework, trained on a single NVIDIA H100 GPU. We used the AdamW optimizer with an initial learning rate of 0.0001 and weight decay of 0.00001. A cosine annealing learning rate scheduler was used, incorporating a 1% warm-up phase during which the learning rate was scaled by 10. Training was conducted for 500 epochs for ISLES'22, GlaS, and FIVES, and 300 epochs for BraTS'23. The batch size was set to 2 for BraTS'23 and 8 for all other datasets. Random patches were extracted during training, with sizes of $(64, 64, 64)$ for ISLES'22, $(96, 96, 96)$ for Spleen, $(128, 128, 128)$ for BraTS'23, $(256, 256)$ for GlaS, and $(512, 512)$ for FIVES. Data augmentation included random affine transformations, flipping, Gaussian noise, Gaussian smoothing, and intensity scaling/shifting. We use the sum of soft Dice [38] and cross-entropy losses as the training objective. For inference, a patch-based sliding window approach with a 50% overlap and the same patch size as training was adopted. The final binary segmentation maps were obtained by thresholding predicted probabilities (sigmoid outputs).

4) Model Configurations: Deconver was configured with dataset-specific hyperparameters to balance model capacity and computational efficiency. The encoder depth (L) was set to 4 for ISLES'22 and Spleen, 5 for BraTS'23, and 6 for GlaS and FIVES. The base number of channels (C_0) set to 64 for ISLES'22 and Spleen, and 32 for BraTS'23, GlaS, and FIVES. At each encoder stage (ℓ), the channel dimension (C_ℓ) was determined by the formula $C_\ell = \min(C_0 \times 2^\ell, 512)$, doubling the channels after each downsampling step until reaching a maximum of 512 channels.

In the NDC layers, the number of groups (G) was set equal to the number of input channels by default. Inspired by depth-wise separable convolutions and validated through ablation studies (Section IV-D), this design choice reduces computational complexity while preserving the ability to model diverse spatial patterns. Additionally, the source channel ratio (R) was fixed at 4, as our experiments found this to optimally balance accuracy and efficiency. The MLP expansion factor (α) was fixed at 4 across all experiments.

5) Evaluation Metrics: We performed stratified 5-fold cross-validation to assess generalization to unseen data. Segmentation performance was quantified using two metrics: Dice Similarity Coefficient (DSC) and Hausdorff Distance 95th percentile (HD95). DSC is defined as:

$$\mathrm{DSC}(\boldsymbol{g}, \boldsymbol{y}) = \frac{2\sum_{n=1}^{N} \boldsymbol{g}[n]\,\boldsymbol{y}[n]}{\sum_{n=1}^{N} \boldsymbol{g}[n] + \sum_{n=1}^{N} \boldsymbol{y}[n]}, \tag{8}$$

where $\boldsymbol{g}[n], \boldsymbol{y}[n] \in \{0, 1\}$ denote the ground truth and predicted labels for voxel n, respectively, and N represents the total number of voxels. DSC is defined as 1 when both the ground truth and the prediction contain only zeros. . Hausdorff Distance is computed as:

$$\mathrm{HD}(\mathbb{G}, \mathbb{Y}) = \max\left\{ \max_{\boldsymbol{g}\in\mathbb{G}} \min_{\boldsymbol{y}\in Y} d(\boldsymbol{g}, \boldsymbol{y}), \max_{\boldsymbol{y}\in\mathbb{Y}} \min_{\boldsymbol{g}\in\mathbb{G}} d(\boldsymbol{g}, \boldsymbol{y}) \right\}. \tag{9}$$

where $d(\boldsymbol{g}, \boldsymbol{y})$ represents the Euclidean distance between points $\boldsymbol{g}$ and $\boldsymbol{y}$; and $\mathbb{G}$ and $\mathbb{Y}$ are sets of all pixel (or voxel) positions on the surface of the ground truth and prediction, respectively. The HD95 metric computes the 95th percentile of distances rather than the maximum, providing a more robust measure against outliers. All the results are reported as the average over the 5-fold cross-validation.

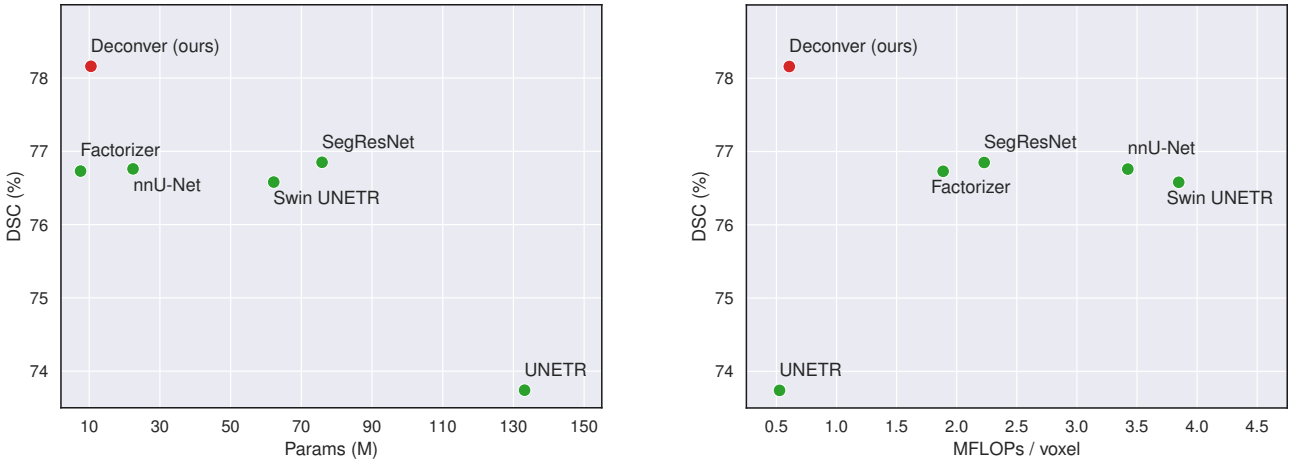
B. Results: 3D Segmentation

1) Binary segmentation (ISLES'22 and Spleen): Table I presents the quantitative results for ISLES'22 and Spleen. Both variants of Deconver outperform all the baselines in terms of DSC, with the version using a kernel size of $3\times3\times3$ achieving the highest DSC (78.16%) followed closely by the variant using a larger kernel size of $5 \times 5 \times 5$ (77.37%). In terms of boundary delineation accuracy, measured by the HD95 metric, Deconver demonstrated superior results, with HD95 values of 4.99 and 4.89 for the $3\times3\times3$ and $5\times5\times5$ kernels, respectively. On spleen segmentation, Deconver $5 \times 5 \times 5$ delivers the top DSC (89.20%) followed by the $3\times3\times3$ variant which reaches 86.73%, while SegResNet records the best HD95 (76.03), with Deconver remaining competitive (80.59–81.08).

Notably, both variants of Deconver significantly reduce computational complexity compared to the best-performing baselines. Deconver requires over 70% fewer FLOPs per voxel compared to the SegResNet, one of the best performing baselines. Additionally, Deconver uses around 85% fewer parameters, resulting in a highly compact architecture without sacrificing segmentation performance. Refer to Fig. 3 for a comparison of model performance versus computational efficiency on ISLES'22. To better understand the source of these differences at a finer granularity, we profiled each encoding block using the same input dimensions and number of channels as in the first encoding layer of the architecture for the ISLES'22 dataset. Under this setting, a Deconver block requires only 143.2K FLOPs per voxel, compared with 443.6K for an nnUNet block, 443.9K for a SegResNet block, and 1602.2K for a Swin UNETR block.

TABLE I: Segmentation performance comparison on ISLES'22 and Spleen. The best results are **bold**, and the second-best are underlined.

Model	Params	ISLES'22			Spleen segmentation		
		FLOPs / voxel	DSC (%)	HD95	FLOPs / voxel	DSC (%)	HD95
nnU-Net	22.4M	3423.5K	76.76	5.54	1014.4K	77.05	120.11
SegResNet	75.9M	2228.6K	76.85	5.18	660.3K	84.63	**76.03**
UNETR	133.2M	525.9K	73.74	6.54	155.8K	66.50	142.04
Swin UNETR	62.2M	4356.8K	76.58	5.55	1290.9K	85.34	109.74
Factorizer	7.5M	2266.7K	76.73	5.93	671.6K	63.65	135.06
Deconver (3×3×3)	10.5M	607.0K	**78.16**	4.99	179.9K	86.73	80.58
Deconver (5×5×5)	11.0M	607.0K	77.37	**4.89**	179.9K	**89.20**	81.08



(a) (b)

Fig. 3: Comparison of DSC against the number of parameters (left) and FLOPs/voxel (right) for different models on ISLES'22. Deconver (the variant using $3 \times 3 \times 3$ kernel) manages to maintain the highest DSC with fewer parameters and FLOPs/voxel.

We also qualitatively examined the segmentation results, as shown in Fig. 4. The figure presents a visual comparison of a representative slice from the ISLES'22 (top row) and the Spleen (bottom row) datasets, showcasing the predictions of different models. For Deconver, we illustrate the results form the $3 \times 3 \times 3$ kernel variant. As evident from the figure, Deconver provides superior segmentation quality compared to other baselines, achieving a more accurate delineation of the lesion while maintaining minimal false positives and false negatives.

The qualitative results show that nnU-Net, Swin UNETR, and UNETR undersegment in both examples, leading to clinically relevant false negatives. SegResNet captures the ground truth more completely but has introduced false-positive areas (in the first example marked by the orange circle).

2) Multi-class Segmentation (BraTS'23): Table II presents segmentation performance on BraTS'23, comparing different models across three tumor subregions: enhancing tumor (ET), tumor core (TC), and whole tumor (WT). Both Deconver variants outperform all baselines in average DSC. Notably, Deconver with a kernel size of 3 achieves the highest DSC for ET and TC, while Deconver with a kernel size of 5 achieves the top DSC for WT, maintaining strong overall performance.

In terms of HD95, Deconver variants demonstrate comparable or superior performance relative to the leading, particularly excelling in TC segmentation. Even when not ranked first, Deconver consistently produces results on par with the best-performing methods.

Similar to before, these competitive segmentation results are achieved with significantly reduced computational complexity. Compared to the second-best performing baseline (SegResNet), Deconver uses approximately 85% fewer parameters and over 90% fewer FLOPs.

Fig. 5 provides qualitative segmentation results on the BraTS'23 dataset. In the first row example, Deconver successfully captures both the centrally located enhancing tumor and the smaller enhancing spot at the lower half, while all other baseline models miss one or both of them (marked by the orange circles). In the second row example, Deconver provides a more accurate delineation of the edema region. Conversely, nnU-Net, Swin UNETR, and UNETR significantly undersegment the edema area, leading to incomplete tumor coverage. SegResNet and Swin UNETR falsely predict the normal brain tissue marked by the orange circle as edema.

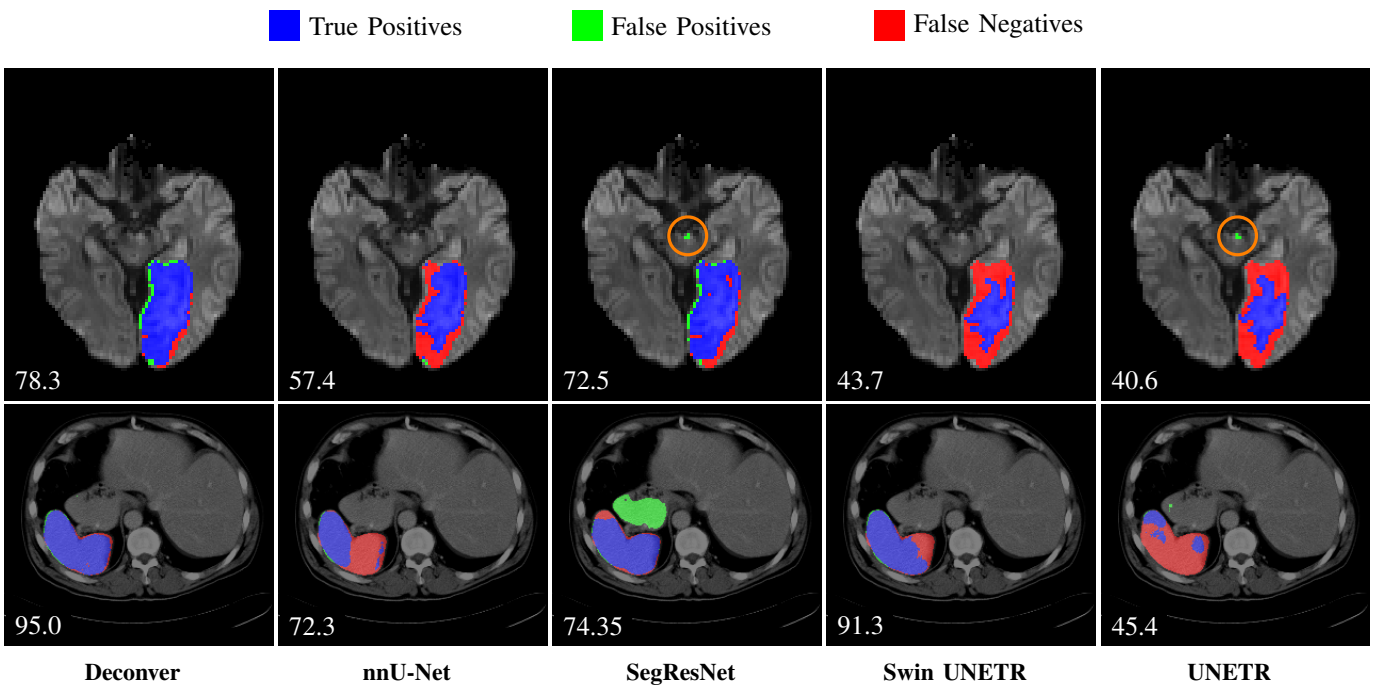


Fig. 4: Qualitative results for ISLES'22 (top row) and Spleen (bottom row). The regions of true positives are marked in blue, false positives in green, and false negatives in red. DSC is presented for each case. SegResNet introduces false positives in both examples (marked with the orange circle in the first row), while Swin UNETR, UNETR, and nnU-Net consistently under-segment in both examples.

TABLE II: Segmentation performance comparison on BraTS'23. The best results are **bold**, and the second-best are underlined.

Model	Params	FLOPs / voxel	DSC (%)				HD95			
			ET	TC	WT	Avg.	ET	TC	WT	Avg.
nnU-Net	22.6M	921.0K	86.73	91.40	93.25	90.46	3.33	3.88	5.54	4.34
SegResNet	75.9M	2235.7K	86.68	91.49	93.44	90.53	**3.26**	3.92	**5.21**	**4.17**
UNETR	139.8M	536.8K	85.47	89.92	92.64	89.34	4.22	4.98	6.93	5.47
Swin UNETR	62.2M	4098.6K	86.71	91.27	93.41	90.46	3.42	3.94	5.42	4.36
Factorizer	7.6M	1087.5K	85.99	90.51	93.13	89.88	3.69	4.36	5.68	4.67
Deconver (3×3×3)	10.6M	167.5K	**87.01**	**91.56**	93.42	**90.66**	3.30	**3.80**	5.63	4.45
Deconver (5×5×5)	11.0M	167.5K	86.97	91.41	**93.47**	90.62	3.50	3.99	5.59	4.49

C. Results: 2D Segmentation

We further evaluated Deconver on 2D medical image segmentation tasks using the GlaS and FIVES datasets. Table III presents the quantitative results. Deconver (5×5) achieved the highest DSC on both datasets, with 92.12% on GlaS and 92.72% on FIVES. Furthermore, it obtained the lowest HD95 value on GlaS (60.49) and the second-best FIVES (30.26). While Deconver (3×3) demonstrated comparable performance relative to baseline methods, it remained outperformed by its 5×5 counterpart. These results suggest that larger kernel sizes lead to improved performance in 2D medical image segmentation.

Consistent with results on 3D datasets, Deconver demonstrates an excellent trade-off between accuracy and computational efficiency on the GlaS and FIVES datasets. Both Deconver variants (3×3 and 5×5) achieve superior or competitive DSC compared to SegResNet, while requiring over 75% fewer FLOPs per pixel. Furthermore, the parameter count for Deconver remains close to that of nnU-Net and SegResNet, and is nearly six times lower than UNETR.

TABLE III: Segmentation performance comparison on 2D datasets (GlaS and FIVES). The best results are **bold**, and the second-best are underlined.

Model	Params	FLOPs / pixel	GlaS		FIVES	
			DSC (%)	HD95	DSC (%)	HD95
nnU-Net	20.6M	874.8K	91.61	87.17	92.65	33.01
SegResNet	25.5M	1928.8K	91.23	69.64	92.71	**28.80**
UNETR	120.2M	1195.1K	90.45	73.38	90.98	35.40
Swin UNETR	25.1M	1406.0K	91.70	67.27	92.69	30.87
Deconver (3×3)	20.6M	422.8K	91.52	68.52	92.48	35.45
Deconver (5×5)	20.8M	422.8K	**92.12**	**60.49**	**92.72**	30.26

In line with our analysis of 3D datasets, we also conducted a qualitative evaluation of the 2D segmentation results, as

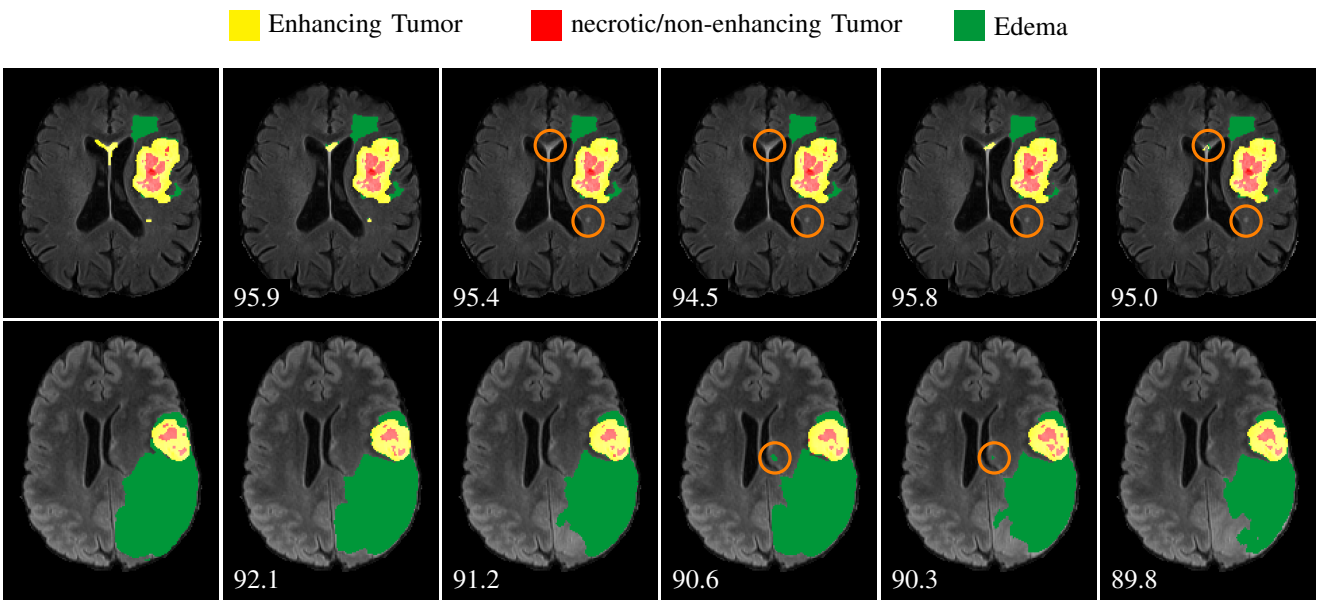


Fig. 5: Qualitative results of brain tumor segmentation on BraTS'23. Tumor core (TC) is the union of red (NCR/NET) and yellow (ET) regions, and whole tumor (WT) is the union of green (edema), red, and yellow regions. Each row displays a sample slice from a subject in the validation set. Average DSC is presented for each case. In the first row example, all of the baselines fail to detect part of the enhancing tumor marked by the orange circle. In the second row example, nnU-Net, Swin UNETR and UNETR do not capture fully the edema, while SegResNet and Swin UNETR falsy predict the area marked by the orange circle as edema.

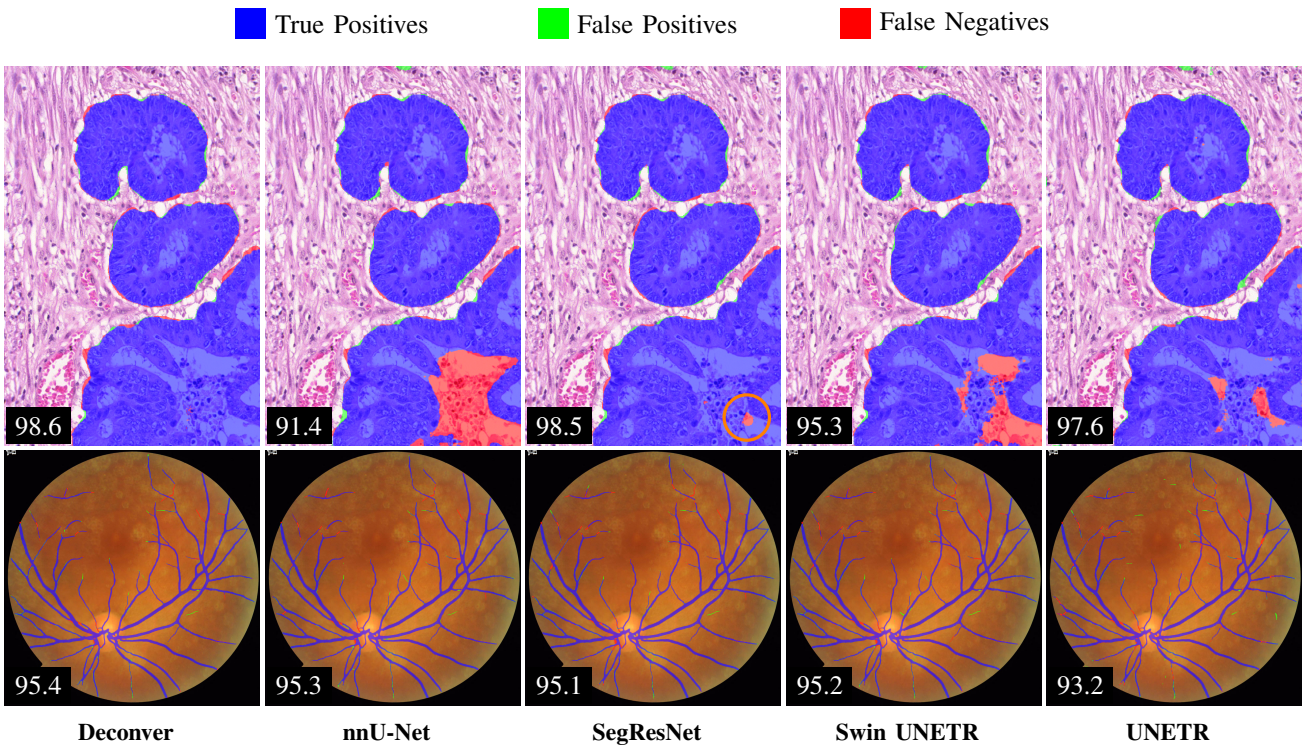


Fig. 6: Qualitative results of 2D segmentation on GlaS (first row) and FIVES (second row). The regions of true positives are marked in blue, false positives in green, and false negatives in red. DSC is presented for each case. The first row presents an example from the validation set of the GlaS dataset, where all baseline models undersegment the gland. The second row shows an example from the FIVES validation set, where, consistent with the quantitative results, most models perform similarly.

TABLE IV: Ablation study on the channel ratio (R) parameter of Deconver using ISLES'22 dataset. The best results are **bold**, and the second-best are underlined.

Ratio (R)	Params	FLOPs / voxel	DSC (%)	HD95
1.0	7.8M	425.8K	77.39	5.07
2.0	8.7M	486.2K	77.32	5.17
4.0	10.5M	607.0K	**78.16**	**4.99**

TABLE V: Ablation study on the number of groups (G) of Deconver using ISLES'22 dataset. The best results are **bold**, and the second-best are underlined.

Groups (G)	Params	FLOPs / voxel	DSC (%)	HD95
1	57.21M	607.4K	77.76	5.08
8	16.18M	607.1K	77.90	5.05
Channels	10.48M	607.0K	**78.16**	**4.99**

TABLE VI: Ablation study on the number of iterations of the multiplicative update rule using ISLES'22 dataset. Best results are **bold**.

Iterations	Params	FLOPs /voxel	DSC (%)	HD95
1	10.48M	607.0K	**78.16**	**4.99**
2	10.48M	609.06K	77.19	5.17

TABLE VII: Ablation study on the number of encoding stages (L) using ISLES'22 dataset. The best results are **bold**, and the second-best are underlined.

Stages (L)	Params	FLOPs / voxel	DSC (%)	HD95
3	2.6M	542.25K	77.46	5.09
4	10.48M	607.0K	**78.16**	**4.99**
5	24.23M	630.78K	78.01	5.06

shown in Fig. 6. The first row illustrates an example from the GlaS dataset, highlighting clear distinctions among the models. All baseline methods, notably nnU-Net and Swin UNETR, fail to completely segment the lowest gland, resulting in substantial false negatives, particularly in its central region. In contrast, Deconver accurately captures the full glandular structure, achieving an almost perfect segmentation mask. We used Deconver with the kernel size of 5 for this experiment.

The second row presents segmentation results from the FIVES dataset. Here, as anticipated by quantitative metrics, most models produce consistently accurate segmentations, with minimal observable differences between their outputs. Just one notable observation is that UNETR produces a higher number of false positives compared to the other methods.

D. Ablation Studies

To better understand the impact of parameter and key architectural choices on Deconver, we performed five ablation studies on the ISLES'22 dataset. We evaluated three key hyperparameters of NDC layers: the source channel ratio (R), the number of groups (G), and the number of iterations of our multiplicative update (3). We also evaluated two architectural aspects: (i) the number of encoding stages (L) in the U-shaped backbone, and (ii) the contribution of different components in the Deconver block by selectively removing the Deconv Mixer or the MLP modules (see Fig. 2a). In all ablation experiments, we varied one parameter while keeping the others fixed to assess its independent effect. We also used the kernel size of $(3, 3, 3)$, which was found to be optimal for the ISLES'22 dataset.

1) Channel Ratio (R): In this experiment, we fixed G to the number of channels and varied R to analyze its effect on the performance (Table IV). We observe that increasing R from 1 to 2 does not change the results significantly, while increasing it from 1 to 4 leads to major improvements in DSC and HD95. However, these gains come at the cost of increased computational complexity as both the number of parameters and FLOPs increase by 35.23% and 42.55% respectively.

2) The Number of Groups (G): In this experiment, we fixed $R = 4$ and varied G to assess its impact (Table V). The results reveal a surprising pattern. Decreasing G leads to a significant increase in parameters without proportional improvements in DSC. In particular, setting $G = 1$ drastically inflates the size of the model (almost 5 times more parameters) while providing no gains, setting G to the number of channels achieves the best DSC and HD95 while keeping the model compact. Notably, changing the Groups parameter does not lead to major changes in the FLOPs.

In general, these studies show that higher channel ratios improve segmentation quality but increase computational cost, while a higher number of groups significantly reduces the number of parameters while improving the performance. The best results are achieved when G is set to the number of channels and $R = 4$, as this configuration produces optimal segmentation accuracy with minimal overhead.

3) The Number of Iterations of Multiplicative Update: In this experiment, we report the results of increasibng the number of iterations of the multiplicative update (3) from one to two while keeping all other settings fixed to see the effect on the performance (Table VI). We observe that using a single iteration already provides the highest DSC (78.16%) with the lowest computational overhead, confirming the efficiency of our default design choice. Increasing the iterations to two reduces segmentation accuracy while incurring extra FLOPs, suggesting that additional updates may introduce redundancy rather than improving feature refinement.

4) Model Depth: In this experiment, we varied the number of encoding stages (L) in the U-shaped backbone (Fig. 1), with the number of decoding blocks (i.e., L) being one less than that of encoding (i.e., $L + 1$). The results in Table VII show that moving from $L = 3$ to $L = 4$ stages improves DSC from 77.46% to 78.16%, while the parameter count increases by around 300% (from 2.6M to 10.5M). Further increasing the depth to $L = 5$ stages reduces the performance (78.01%), while raising the parameter count by over 130% (from 10.5M to 24.2M). These findings suggest that our inital choice of $L = 4$ stages achieve the best balance between accuracy and model complexity.

5) Subblocks of Deconver Block: In this experiment, we analyzed the contribution of each component in the Deconver

TABLE VIII: Ablation study on the effect of different components in the Deconver block using ISLES'22 dataset. The best results are **bold**, and the second-best are underlined.

Included block	Params	FLOPs / voxel	DSC (%)	HD95
MLP	6.41M	333.7K	72.08	9.90
Deconv Mixer	6.99M	336.4K	77.30	**4.85**
Deconv Mixer + MLP	10.48M	607.0K	**78.16**	4.99

block by testing configurations with only MLP, only Deconv Mixer, and the full combination (see Fig. 2a). As shown in Table VIII, using only MLP results in the weakest performance (72.08% DSC), given that the parameter count is 39% lower compared to the full block (6.4M vs. 10.5M). Deconv Mixer alone improves DSC to 77.30% and achieves the best HD95 (4.85), with 33% fewer parameters than the full model (7.0M vs. 10.5M). Combining both modules yields the highest DSC (78.16%) and competitive HD95 (4.99), at the cost of increased complexity. These findings confirm that both components are complementary: Deconv Mixer is crucial for spatial detail, while MLP models channel interactions, and together they provide the strongest overall performance.

V. Conclusion and Discussion

In this work, we introduce Deconver, a powerful segmentation network that integrates nonnegative deconvolution (NDC) as a learnable module within a U-shaped architecture. By replacing computationally expensive attention mechanisms with efficient deconvolution operations, Deconver restores high-frequency details while effectively suppressing artifacts. Compared with other methods like Wiener filtering, our deconvolution method integrates more naturally into deep networks as it operates in the spatial domain using simple convolution and elementwise multiplication operations. Moreover, it enforces the non-negativity inherent to image data and avoids dependence on additional Fourier transforms and frequency domain priors.

Extensive experiments on five diverse medical imaging datasets (ISLES'22, Spleen, BraTS'23, GlaS, and FIVES) demonstrate that Deconver consistently achieves state-of-the-art segmentation performance, outperforming or matching leading CNN- and Transformer-based models while significantly reducing computational costs. Notably, Deconver reduces FLOPs by up to 90% compared to attention-based baselines, making it well-suited for resource-constrained clinical applications. Ablation studies further highlight the importance of key design choices, such as the source channel ratio and grouping strategy in NDC layers, in balancing accuracy and efficiency. We believe Deconver represents a promising step toward high-precision, computationally efficient medical image segmentation, bridging the gap between classical image restoration and modern deep learning. While our results are consistent across our benchmarks, clinical data inevitably contain fluctuations such as scanner-related noise or patient motion, which may influence segmentation in practice. Beyond the datasets used here, evaluating Deconver on additional modalities such as ultrasound and X-ray will be important to further establish generalizability. The choice of hyperparameters could also affect performance when adapting to new datasets. Future work, includes extending the framework beyond segmentation to other tasks such as classification and to further optimize the implementation for faster, more memory-efficient training and inference.

Appendix

Proof of Theorem 1

We prove the theorem using the majorization-minimization (MM) framework. This involves iteratively minimizing a surrogate function that upperbounds the original objective. The MM approach guarantees a monotonic decrease in the reconstruction error $\mathcal{E}(\boldsymbol{\mathcal{S}})$ through two key steps:

1) **Majorization**: Construct a surrogate function $Q(\boldsymbol{\mathcal{S}} \mid \boldsymbol{\mathcal{S}}^{(t)})$ that satisfies:
$$Q(\boldsymbol{\mathcal{S}} \mid \boldsymbol{\mathcal{S}}^{(t)}) \geq \mathcal{E}(\boldsymbol{\mathcal{S}}), \tag{10}$$
for all $\boldsymbol{\mathcal{S}}$, with equality when $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{S}}^{(t)}$.
2) **Minimization**: Update the source to minimize the surrogate:
$$\boldsymbol{\mathcal{S}}^{(t+1)} = \arg\min_{\boldsymbol{\mathcal{S}} \geq 0} Q(\boldsymbol{\mathcal{S}} \mid \boldsymbol{\mathcal{S}}^{(t)}) \tag{11}$$

Combining these, we directly obtain:

$$\mathcal{E}(\boldsymbol{\mathcal{S}}^{(t+1)}) \leq Q(\boldsymbol{\mathcal{S}}^{(t+1)} \mid \boldsymbol{\mathcal{S}}^{(t)}) \leq Q(\boldsymbol{\mathcal{S}}^{(t)} \mid \boldsymbol{\mathcal{S}}^{(t)}) = \mathcal{E}(\boldsymbol{\mathcal{S}}^{(t)}),$$

proving the reconstruction error is non-increasing across iterations.

Step 1: Majorization

Let's first expand the reconstruction error as

$$\begin{aligned}\mathcal{E}(\boldsymbol{\mathcal{S}}) &= \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}\|_{\mathrm{F}}^2 \\ &= \|\boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2 - 2\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}} \rangle + \|\boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}\|_{\mathrm{F}}^2. \end{aligned} \tag{12}$$

The main challenge lies in majorizing the quadratic term $\|\boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}\|_F^2$. To achieve this, we apply the elementwise Cauchy–Schwarz inequality to $(\boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}})^2$. For each output element (c, h, w), define:

$$A_{d,m,n} = \frac{\boldsymbol{\mathcal{S}}_p[d, h+m, w+n]\boldsymbol{\mathcal{V}}[c, d, m, n]}{\sqrt{\boldsymbol{\mathcal{S}}_p^{(t)}[d, h+m, w+n]\boldsymbol{\mathcal{V}}[c, d, m, n]}},$$
$$B_{d,m,n} = \sqrt{\boldsymbol{\mathcal{S}}_p^{(t)}[d, h+m, w+n]\boldsymbol{\mathcal{V}}[c, d, m, n]},$$

where $\boldsymbol{\mathcal{S}}_p^{(t)} = \mathrm{pad}(\boldsymbol{\mathcal{S}}^{(t)}, (M, N))$ and $\boldsymbol{\mathcal{S}}_p = \mathrm{pad}(\boldsymbol{\mathcal{S}}, (M, N))$. By Cauchy–Schwarz, we have:

$$\left(\sum_{d,m,n} A_{d,m,n} B_{d,m,n}\right)^2 \leq \left(\sum_{d,m,n} A_{d,m,n}^2\right)\left(\sum_{d,m,n} B_{d,m,n}^2\right).$$

Substituting back and using the definition of cross-correlation, we obtain:

$$(\boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}})^2 \leq \left(\frac{\boldsymbol{\mathcal{S}}^2}{\boldsymbol{\mathcal{S}}^{(t)}} * \boldsymbol{\mathcal{V}}\right) \odot \left(\boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}}\right). \tag{13}$$

Where $(\cdot)^2$ denotes elementwise squaring. Summing over all elements gives:

$$\|\boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}\|_{\mathrm{F}}^2 \leq \sum_{c,h,w} \left(\frac{\boldsymbol{\mathcal{S}}^2}{\boldsymbol{\mathcal{S}}^{(t)}} * \boldsymbol{\mathcal{V}} \right) [c,h,w] \cdot \left(\boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}} \right) [c,h,w]$$
$$= \langle \frac{\boldsymbol{\mathcal{S}}^2}{\boldsymbol{\mathcal{S}}^{(t)}} * \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}} \rangle. \tag{14}$$

This constructs the majorizing surrogate function:

$$Q(\boldsymbol{\mathcal{S}} \mid \boldsymbol{\mathcal{S}}^{(t)}) = \|\boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2 - 2\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}} \rangle + \langle \frac{\boldsymbol{\mathcal{S}}^2}{\boldsymbol{\mathcal{S}}^{(t)}} * \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}} \rangle. \tag{15}$$

Step 2: Minimization

To minimize the surrogate function $Q(\boldsymbol{\mathcal{S}} \mid \boldsymbol{\mathcal{S}}^{(t)})$, we first derive its gradient with respect to $\boldsymbol{\mathcal{S}}$.

Linear Term Gradient: The linear term $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}} \rangle$ has gradient:

$$\nabla_{\mathcal{S}} \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}} \rangle [d', h', w'] = \sum_{c,h,w} \boldsymbol{\mathcal{X}}[c,h,w] \frac{\partial (\boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}})[c,h,w]}{\partial \boldsymbol{\mathcal{S}}[d', h', w']}. \tag{16}$$

Expanding the cross-correlation $(\boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}})[c,h,w]$, we find that the partial derivative is nonzero when:

$$h + m = h' + M, \quad w + n = w' + N. \tag{17}$$

Thus,

$$\frac{\partial (\boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}})[c,h,w]}{\partial \boldsymbol{\mathcal{S}}[d', h', w']} = \begin{cases} \boldsymbol{\mathcal{V}}[c,d',m,n], & \text{if (17) holds,} \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Substituting back, $\nabla_{\mathcal{S}} \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}} \rangle [d', h', w']$ simplifies to:

$$\sum_{c,m,n} \boldsymbol{\mathcal{X}}[c, h' - m + M, w' - n + N] \boldsymbol{\mathcal{V}}[c,d',m,n].$$

Recognizing this as a cross-correlation operation, we obtain:

$$\nabla_{\mathcal{S}} \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}} \rangle = \boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{V}}^{-}. \tag{19}$$

Quadratic Term Gradient: Using (19) together with chain rule, the gradient of the quadratic term can be derived as

$$\nabla_{\mathcal{S}} \langle \frac{\boldsymbol{\mathcal{S}}^2}{\boldsymbol{\mathcal{S}}^{(t)}} * \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}} \rangle = \frac{2\boldsymbol{\mathcal{S}}}{\boldsymbol{\mathcal{S}}^{(t)}} \odot \left[(\boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}}) * \boldsymbol{\mathcal{V}}^{-} \right]. \tag{20}$$

Solving for $\boldsymbol{\mathcal{S}}^{(t+1)}$: Combining both gradients and setting the total gradient $\nabla_{\mathcal{S}} Q(\boldsymbol{\mathcal{S}} \mid \boldsymbol{\mathcal{S}}^{(t)}) = 0$ yields

$$-2(\boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{V}}^{-}) + \frac{2\boldsymbol{\mathcal{S}}}{\boldsymbol{\mathcal{S}}^{(t)}} \odot \left[(\boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}}) * \boldsymbol{\mathcal{V}}^{-} \right] = 0. \tag{21}$$

Solving for $\boldsymbol{\mathcal{S}}$, we derive the multiplicative update rule:

$$\boldsymbol{\mathcal{S}}^{(t+1)} = \boldsymbol{\mathcal{S}}^{(t)} \odot \frac{\boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{V}}^{-}}{(\boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}}) * \boldsymbol{\mathcal{V}}^{-}}.$$

If for any index (d,h,w), we have $(\boldsymbol{\mathcal{S}}^{(t)} * \boldsymbol{\mathcal{V}} * \boldsymbol{\mathcal{V}}^{-})[d,h,w] = 0$, then the nonnegativity of $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{V}}$, and $\boldsymbol{\mathcal{S}}^{(t)}$ implies $(\boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{V}}^{-})[d,h,w] = 0$. In this case, the indeterminate form $0/0$ is resolved by setting $\boldsymbol{\mathcal{S}}^{(t+1)}[d,h,w] = 0$, thereby preserving nonnegativity.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.

[2] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 424–432, 2016.

[3] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature methods*, vol. 18, no. 2, pp. 203–211, 2021.

[4] S. Roy *et al.*, "MedNeXt: Transformer-driven scaling of convnets for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2023, pp. 405–415.

[5] H.-Y. Zhou *et al.*, "nnFormer: Interleaved transformer for volumetric segmentation," *IEEE Transactions on Image Processing*, vol. 32, pp. 1–14, 2023.

[6] X. Huang, Z. Deng, D. Li, X. Yuan, and Y. Fu, "MISSFormer: An effective transformer for 2d medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 42, no. 5, pp. 1484–1494, 2023. DOI: 10.1109/TMI.2022.3230943.

[7] J. Chen *et al.*, "TransUNet: Rethinking the u-net architecture design for medical image segmentation through the lens of transformers," *Medical Image Analysis*, vol. 97, p. 103 280, 2024.

[8] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, "Swin UNETR: Swin transformers for semantic segmentation of brain tumors in mri images," in *International MICCAI brainlesion workshop*, Springer, 2021, pp. 272–284.

[9] S. K. Roy, A. Jamali, J. Chanussot, P. Ghamisi, E. Ghaderpour, and H. Shahabi, "SimPoolFormer: A two-stream vision transformer for hyperspectral image classification," *Remote Sensing Applications: Society and Environment*, vol. 37, p. 101 478, 2025.

[10] M. A. Al-qaness, G. Wu, and D. AL-Alimi, "Mgcet: Mlp-mixer and graph convolutional enhanced transformer for hyperspectral image classification," *Remote Sensing*, vol. 16, no. 16, p. 2892, 2024.

[11] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. MIT Press, 1949.

[12] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *The Astronomical Journal*, vol. 79, p. 745, 1974.

[13] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested u-net architecture for medical image segmentation," in *Deep learning in*

medical image analysis and multimodal learning for clinical decision support: 4th international workshop, DLMIA 2018, and 8th international workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, proceedings 4, Springer, 2018, pp. 3–11.

[14] A. Myronenko, "3D MRI brain tumor segmentation using autoencoder regularization," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*, Springer, 2019, pp. 311–320.

[15] E. de la Rosa *et al.*, *A robust ensemble algorithm for ischemic stroke lesion segmentation: Generalizability and clinical utility beyond the isles challenge*, 2024. arXiv: 2403.19425 [eess.IV]. [Online]. Available: https://arxiv.org/abs/2403.19425.

[16] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[17] P. Ashtari, D. M. Sima, L. De Lathauwer, D. Sappey-Marinier, F. Maes, and S. Van Huffel, "Factorizer: A scalable interpretable approach to context modeling for medical image segmentation," *Medical image analysis*, vol. 84, p. 102 706, 2023.

[18] A. Hatamizadeh *et al.*, "UNETR: Transformers for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022, pp. 574–584.

[19] Z. Liu *et al.*, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 10 012–10 022.

[20] P. Satish, M. Srikantaswamy, and N. K. Ramaswamy, "A comprehensive review of blind deconvolution techniques for image deblurring.," *Traitement du Signal*, vol. 37, no. 3, 2020.

[21] W. H. Richardson, "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America*, vol. 62, no. 1, pp. 55–59, 1972.

[22] K. Katoh, *Recent applications of deconvolution microscopy in medicine*, 2024.

[23] A. Debnath, H. M. Rai, C. Yadav, A. Agarwal, and A. Bhatia, "Deblurring and denoising of magnetic resonance images using blind deconvolution method," *International Journal of Computer Applications*, vol. 81, no. 10, pp. 7–12, 2013.

[24] P. Sharma, S. Sharma, and A. Goyal, "An mse (mean square error) based analysis of deconvolution techniques used for deblurring/restoration of mri and ct images," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016, pp. 1–5.

[25] Y. Liu, Y. Liang, G. Mu, and X. Zhu, "Deconvolution methods for image deblurring in optical coherence tomography," *Journal of the Optical Society of America A*, vol. 26, no. 1, pp. 72–77, 2008.

[26] C. Sample *et al.*, "Neural blind deconvolution for deblurring and supersampling psma pet," *Physics in Medicine & Biology*, vol. 69, no. 8, p. 085 025, 2024.

[27] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13, MIT Press, 2000. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

[28] D. Ulyanov, A. Vedaldi, and V. Lempitsky, *Instance normalization: The missing ingredient for fast stylization*, 2017. arXiv: 1607.08022 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1607.08022.

[29] K. He, X. Zhang, S. Ren, and J. Sun, *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*, 2015. arXiv: 1502.01852 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1502.01852.

[30] M. R. Hernandez Petzsche *et al.*, "ISLES 2022: A multi-center magnetic resonance imaging stroke lesion segmentation dataset," *Scientific data*, vol. 9, no. 1, p. 762, 2022.

[31] M. Antonelli *et al.*, "The medical segmentation decathlon," *Nature communications*, vol. 13, no. 1, p. 4128, 2022.

[32] B. H. Menze *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015. DOI: 10.1109/TMI.2014.2377694.

[33] U. Baid *et al.*, *The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification*, 2021. arXiv: 2107.02314 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2107.02314.

[34] S. Bakas *et al.*, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific Data*, vol. 4, 2017. DOI: 10.1038/sdata.2017.117.

[35] K. Sirinukunwattana *et al.*, "Gland segmentation in colon histology images: The glas challenge contest," *Medical image analysis*, vol. 35, pp. 489–502, 2017.

[36] K. Sirinukunwattana, D. R. J. Snead, and N. M. Rajpoot, "A stochastic polygons model for glandular structures in colon histology images," *IEEE Transactions on Medical Imaging*, vol. 34, no. 11, pp. 2366–2378, 2015. DOI: 10.1109/TMI.2015.2433900.

[37] K. Jin *et al.*, "FIVES: A fundus image dataset for artificial intelligence based vessel segmentation," *Scientific data*, vol. 9, no. 1, p. 475, 2022.

[38] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571. DOI: 10.1109/3DV.2016.79.