Segmentation CNNs are denoising models

Luis A. Zavala-Mondragón [1] **Ruud van Sloun** [1] **Peter H.N. de With** [1] **Fons van der Sommen** [1]

Abstract

Encoder-decoder CNNs, such as the U-Net are the *de-facto* approach for image segmentation. Despite their good properties, U-Net-like models are often treated as black boxes, which hides the signal processing performed to the images, as well as their potential downsides/limitations. To address these disadvantages, this paper studies the signal processing performed by segmentation models such as the U-Net by employing a proxy CNN, in which its linear behavior can be analyzed. The suggested proxy model has been trained for image segmentation and its impulse response is computed for different training and test settings. The impulse and frequency responses show that the processing of U-Net-like models trained for segmentation are similar to sparse modeling techniques employed in image denoising and in signal detection. Furthermore, this simple approach of using a proxy CNN can indicate also properties of the filter banks that compose the CNN.

1. Introduction

Convolutional neural networks (CNNs) are the *de-facto* approach for many pixel-level tasks in image processing and computer vision, such as image denoising (Zhang et al., 2017), image super-resolution (Wang et al., 2020), image segmentation (Ronneberger et al., 2015), etc. The reason for this broad adoption can be attributed to a few factors, such as data availability, the high performance of DNNs and the ability of standard CNNs (e.g. the U-Net (Ronneberger et al., 2015)) to be repurposed to new tasks by simply re-training with different data and/or by performing minor architectural changes.

Despite the numerous advantages of conventional CNN approaches for image processing and computer vision, CNNs also have disadvantages that should be considered. For example, CNNs have high model complexity (McCann et al., 2017), they can be easily misled by adversarial attacks (Khamaiseh et al., 2022), they are prone to overfitting and their internal operation is often unknown. In addition, it has been found that producing marginal gains in performance comes at the cost of an exponential increase in computational complexity (Thompson et al., 2020).

The aforementioned disadvantages of CNNs have not remained unnoticed. Over the past years, significant efforts have been made to better understand CNNs employed for image processing and computer vision. Among these efforts, the studies of the signal processing behavior of denoising CNNs are particularly remarkable, since they have provided feasible explanations of the operation of CNNs based on concepts, such as wavelets (Zavala-Mondragón et al., 2023), low-rank approximation (Ye et al., 2018), convolutional sparse coding (Papyan et al., 2017), etc. These studies explain that U-Net-like models applied for image denoising have the following signal processing behavior. First, the convolution filters in the encoder separate the input signal into multiple bands. Second, the nonlinear part of the model (e.g. ReLUs or shrinkage functions), suppress parts of the signal in the bands, which are associated with the noise. Finally, the convolution filters in the decoder map the signal back to the original domain. It should be noted that this simple abstraction of the cognitive bias of U-Net-like models for this task has been exploited for performance improvements (Han & Ye, 2018), and for improving the interpretability of CNN models (Zavala-Mondragón et al., 2022).

In contrast with noise reduction, the signal processing executed by segmentation models is less understood. This is sensible because the connection of this task with other signal processing algorithms is less obvious, since segmentation models perform a more complex mapping of the input signal than denoising CNNs. However, we hypothesize that, just as denoising CNNs, the behavior of segmentation models is similar to noise reduction algorithms based on sparse modeling. Specifically, we hypothesize, that segmentation models learn multi-channel (sparse) representations, in which the image sections associated to the background are suppressed, while the signal of interest is preserved and/or boosted.

It should be noted that we are not the first in highlighting the possibility that denoising encoder-decoder mod-

[1] Department of Electrical Engineering, Eindhoven universiy of Technology, Eindhoven, The Netherlands. Correspondence to: Luis A. Zavala-Mondragón <l.a.zavala.mondragon@tue.nl>.

els are related to segmentation. For example, Kascenas and O'Neil (Kascenas et al., 2022) have employed a U-Net-based denoising autoencoder for anomaly detection. Their model is trained in such way that it would reject low-frequency noise from anomaly-free brain MRI slices. The reasoning behind this approach is based on the assumption that anomalies for that application have similarities with low-frequency noise. As a consequence, it can be observed that the specified signal modeling causes that the CNN is unable to reconstruct anomalies and the difference between the input and denoised images resemble a segmentation mask. Finally, it should be noted that this model is unsupervised and no further connections with supervised segmentation are made.

It can be observed that studying the signal processing behavior in U-Net-like segmentation models is challenging, because these models perform arbitrary scaling factors and they have normalization layers that can introduce offsets to the signal and have signal-dependent behavior. In addition, ReLU activations and max-pooling operations impede to independently study the linear part of the model (the convolutions) from the nonlinear sections (ReLU and max-pooling operations). The reason for this issue is that eliminating the nonlinearities from ReLU-based CNNs changes drastically the behavior of the model. In this paper, we are interested to understand the effect of both the linear and nonlinear processes that the CNN applies to the input signal in order to map it to a segmentation mask. This understanding should be realized by modeling these processes in a network that is suited for image segmentation.

In order to study the signal processing behavior of segmentation U-Net-like models, this paper introduces a simplified CNN in which the signal processing behavior can be more easily studied. This CNN is referred to as *simplified U-Net-proxy model* (SUM) and it is a model that avoids normalization layers and employs soft shrinkage activations as nonlinearities. These design choices allow the model to behave as a linear function when the threshold is set to zero, which allows to independently study the linear and nonlinear processes within the model (Zavala-Mondragón et al., 2022).

Our experiments based on the SUM model confirm our hypothesis that encoder-decoder models, such as the U-Net, achieve segmentation in a process that is reminiscent of wavelet/framelet approaches for image denoising. Specifically, the convolution filters of the encoder separate the signal into multiple bands and shrinkage functions are applied to remove/boost sections of the decomposed signal, while the decoder reconstructs the processed signal. The result of this operation is a signal which that is highly correlated to the area delimited by the segmentation mask, while other non-relevant parts are attenuated. Finally, the output layer is limited to saturate the previously referred signal to generate a final segmentation mask.

The structure of this paper is as follows. Section 3 discusses the method applied for investigating the signal processing behavior in the SUM model. Section 4 describes the dataset, experiments and their results. Moreover, Section 5 provides a discussion on the outcomes of the results. Finally, Section 6 highlights the main findings and contributions of this paper.

2. Background: Encoder-decoder CNNs for noise reduction

This section introduces the signal model involved in data-driven noise reduction and a high-level description of the operation of noise reduction encoder-decoder CNNs. Prior to addressing these concepts, the notation is first introduced. The notation follows the paper by Zavala-Mondragón *et al.* (Zavala-Mondragón et al., 2023) and it is briefly discussed here for self containment. In this paper, scalars are represented by lowercase letters (e.g. a), vectors by lowercase underlined letters (e.g. $\underline{b}$), matrices –i.e. images– by boldface lowercase letters (e.g. $\mathbf{x}$) and tensors –i.e. convolution kernels– by uppercase boldface letters (e.g. $\mathbf{K}$). Moreover, the convolution between two 2D signals (e.g. $\mathbf{f}$ and $\mathbf{x}$) is represented by $\mathbf{k} * \mathbf{f}$, the convolution between two tensors $\mathbf{F}$ and $\mathbf{K}$ by $\mathbf{FK}$ and, finally, the convolution between a tensor $\mathbf{K}$ and an image $\mathbf{x}$ is shown as $\mathbf{Kx}$.

In noise reduction problems, it is common to describe a noisy observed signal $\mathbf{x}$ by

$$\mathbf{x} = \mathbf{y} + \boldsymbol{\eta}, \tag{1}$$

where, $\mathbf{y}$ is a noiseless signal and $\boldsymbol{\eta}$ represents noise. It is common to estimate the noiseless signal in Eq. (1) with encoder-decoder CNNs. Akin to other work (Zavala-Mondragón et al., 2023), we assume that the simplest encoder-decoder module is defined by

$$\mathrm{G}(\mathbf{x}) = \mathrm{A}_{(\tilde{\underline{b}})}(\tilde{\mathbf{K}}^{\intercal}\mathrm{A}(\mathbf{Kx})_{(\underline{b})}), \tag{2}$$

in which $\mathrm{G}(\cdot)$ is a generic encoder-decoder block, while $\tilde{\mathbf{K}}^{\intercal}$ and $\mathbf{K}$ are the decoder and encoder convolution kernels, respectively. Furthermore, $\mathrm{A}_{(\underline{b})}(\cdot)$ and $\mathrm{A}_{(\tilde{\underline{b}})}(\cdot)$ are arbitrary activations that depend on their bias vectors $\underline{b}$ and $\tilde{\underline{b}}$. In Eq. (2), the *encoded/latent representation* $\mathbf{E}$ is defined by

$$\mathbf{E} = \mathbf{Kx}, \tag{3}$$

which is sparsified with the activation $\mathrm{A}_{(\underline{b})}(\cdot)$. It can be observed that $\mathrm{G}(\cdot)$ is a single encoder-decoder block and deeper models are achieved by nesting multiple basic encoder-decoder blocks. This is expressed by

$$\mathrm{ED}(\mathbf{x}) = \mathrm{G}_{N-1} \circ \mathrm{G}_{N-2} \ldots \mathrm{G}_1 \circ \mathrm{G}_0(\mathbf{x}), \tag{4}$$

where $\mathrm{G}_n(\cdot)$ is the n-th shallow encoder-decoder network and $\mathrm{ED}(\cdot)$ is a deep CNN. It should be noted that this iterative approach has drawn analogies between deep CNNs and iterative soft shrinkage algorithms (Jin et al., 2017; Daubechies et al., 2004; Gregor & LeCun, 2010).

Existing research shows that at least two elements enable noise reduction in CNNs (Zavala-Mondragón et al., 2023). (1) Linear filtering caused by the convolution filters. (2) The nonlinearities that remove noise by enforcing sparsity. To illustrate the linear filtering happening within the model, it is assumed that the activation functions are linear (i.e. $= \mathrm{A}_{(\cdot)}(\mathbf{x}) = \tilde{\mathrm{A}}_{(\cdot)}(\mathbf{x}) = \mathbf{x}$). Under these conditions, Eq. (2) becomes

$$\hat{\mathbf{y}} = \mathbf{k} * \mathbf{x} = \tilde{\mathbf{K}}^{\intercal}\mathbf{K}\mathbf{x}, \tag{5}$$

where the noiseless estimate $\hat{\mathbf{y}}$ is the result of minimizing the distance between $\mathbf{x}$ and the ground-truth signal $\mathbf{y}$ through the learned convolution filter $\mathbf{k}$ (i.e. a learned Wiener filter). In contrast with the previous analysis, when studying the effect of the sparsifying nonlinearities in denoising CNNs, it is easier to assume that, under certain conditions, the learned convolution kernels $\mathbf{K}$ and $\tilde{\mathbf{K}}$ may allow for perfect reconstruction, which is equivalent to

$$\mathbf{x} = \mathrm{A}_{(\tilde{\underline{b}}=0)}(\tilde{\mathbf{K}}^{\intercal}\mathrm{A}_{(\underline{b}=0)}(\mathbf{K}\mathbf{x})). \tag{6}$$

for any $\mathbf{x}$. If a model complies with this condition, then the biases $\underline{b}$ in combination with sparsifying activations $\mathrm{A}_{(\cdot)}(\cdot)$ (e.g. ReLUs or soft shrinkages) enable the noise reduction behavior of the model by enforcing sparsity in the encoded/latent representation, which is analogous to denoising algorithms based on sparse modeling (Ye et al., 2018; Papyan et al., 2017; Zavala-Mondragón et al., 2023). As concluding remark, note that trained models likely employ both, linear and nonlinear filtering to achieve noise reduction.

As a final remark, it can be observed that despite the fact that residual models are not discussed here, similar denoising behaviors can be attributed to them. However, in such cases, the CNN cancels the components attributed to the signal in such way that the signal reconstructed by the decoder approximates the noise present in the signal and allows it to be subtracted from the original input to generate the final noiseless estimate. For a more detailed description of this behavior we refer the reader to the paper of Zavala-Mondragón *et al.* (Zavala-Mondragón et al., 2023).

3. Methods

3.1. Hypothesized behavior of segmentation CNNs

Segmentation CNNs estimate a binary mask $\hat{\mathbf{m}}$ of the true segmentation $\mathbf{m}$ by

$$\hat{\mathbf{m}} = \mathrm{C}_{(\hat{\mathbf{K}},\hat{\tilde{\mathbf{K}}},\hat{\mathbf{K}}_{\Omega},\hat{\tilde{\underline{b}}},\hat{\underline{b}})}(\mathbf{x}), \tag{7}$$

where is the input image $\mathbf{x}$ and $\mathrm{C}_{(\cdot)}(\cdot)$ is a segmentation CNN. The learned encoder and decoder convolution kernels and biases of $\hat{\tilde{\mathbf{K}}}$ are $\mathrm{C}_{(\cdot)}(\cdot)$, $\hat{\mathbf{K}}$, $\hat{\underline{b}}$ and $\hat{\tilde{\underline{b}}}$. In addition, the learned convolution weight of the output layer is $\mathbf{K}_{\Omega}$. The parameters of $\mathrm{C}_{(\cdot)}(\cdot)$ are learned by minimizing

$$(\hat{\mathbf{K}},\hat{\tilde{\mathbf{K}}},\hat{\mathbf{K}}_{\Omega},\hat{\tilde{\underline{b}}},\hat{\underline{b}}) = \underset{(\mathbf{K},\tilde{\mathbf{K}},\mathbf{K}_{\Omega},\tilde{\underline{b}},\underline{b})}{\arg\min} \mathcal{L}\{\mathrm{C}_{(\mathbf{K},\tilde{\mathbf{K}},\mathbf{K}_{\Omega},\tilde{\underline{b}},\underline{b})}(\mathbf{X}),\mathbf{M}\}, \tag{8}$$

in which $\mathcal{L}\{\cdot\}$ is a loss function and the dataset is defined by a set of input images $\mathbf{X}$ and of ground-truth segmentations $\mathbf{M}$. It should be noted that this approach, at first glance, does not model the signal, which partly explains its lack of interpretability.

The segmentation CNN $\mathrm{C}_{(\cdot)}(\cdot)$ is generically described by the model

$$\mathrm{C}(\mathbf{x}) = \Omega\big(\mathrm{ED}(\mathbf{x})\big), \tag{9}$$

where $\mathrm{ED}(\cdot)$ is an encoding-decoding CNN, such as the denoising model described in Section 2. Moreover, the output layer is represented by $\Omega(\cdot)$ and it is often composed by a convolution layer and a saturating function (e.g. the sigmoid).

If it is assumed that the additive noise model shown in Eq. (1) applies to image segmentation as well, then the observed image $\mathbf{x}$ is the superposition of the signal of interest/foreground $\mathbf{y}_{\mathrm{Sig}}$ and the background $\boldsymbol{\eta}_{\mathrm{bg}}$. This is represented by the model

$$\mathbf{x} = \mathbf{y}_{\mathrm{Sig}} + \boldsymbol{\eta}_{\mathrm{BG}}. \tag{10}$$

In this paper, it is hypothesized that forward-propagating an input image $\mathbf{x}$ through the the encoder-decoder module $\mathrm{ED}(\cdot)$ of $\mathrm{C}(\cdot)$ results in

$$\hat{\mathbf{c}}_{\mathrm{Sig}} + \varepsilon = \mathrm{ED}(\mathbf{x}). \tag{11}$$

Here, signal $\hat{\mathbf{c}}_{\mathrm{Sig}}$ is correlated to the foreground $\mathbf{y}_{\mathrm{Sig}}$ and ε represents errors (e.g. signal offsets). Finally, propagating the output of $\mathrm{ED}(\cdot)$ through the output layer generates the estimated segmentation mask, which is described by

$$\hat{\mathbf{m}} = \Omega(\hat{\mathbf{c}}_{\mathrm{Sig}} + \varepsilon). \tag{12}$$

It can be observed that $\Omega(\cdot)$ is a saturating function. Consequently, the error signal ε can be easily pushed outside the dynamic range. It should be noted that under this assumption, segmentation CNNs could be considered as a nonlinear extension of matched filters (see Appendix I), where the encoder-decoder module improves the ability to detect the foreground signal, by suppressing the background (or in other words, to improve the foreground-to-background energy ratio). Finally the output layer simply performs a threshold operation.

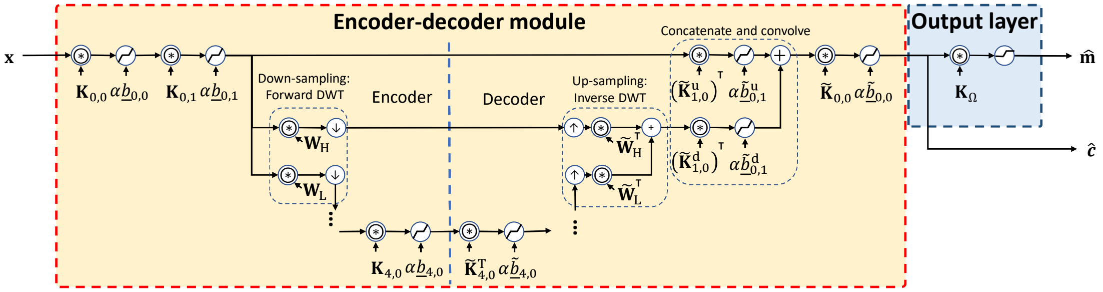
Encoder-decoder module
Output layer
x
$\mathbf{K}_{0,0}$ $\alpha\underline{b}_{0,0}$
$\mathbf{K}_{0,1}$ $\alpha\underline{b}_{0,1}$
Down-sampling:
Forward DWT
$\mathbf{W}_\mathrm{H}$
$\mathbf{W}_\mathrm{L}$
Encoder
Decoder
$\mathbf{K}_{4,0}$ $\alpha\underline{b}_{4,0}$
$\tilde{\mathbf{K}}^\mathrm{T}_{4,0}$ $\alpha\underline{\tilde{b}}_{4,0}$
Up-sampling:
Inverse DWT
$\tilde{\mathbf{W}}_\mathrm{H}^\mathsf{T}$
$\tilde{\mathbf{W}}_\mathrm{L}^\mathsf{T}$
Concatenate and convolve
$(\tilde{\mathbf{K}}^\mathrm{u}_{1,0})^\mathsf{T}$ $\alpha\underline{\tilde{b}}^\mathrm{u}_{0,1}$
$(\tilde{\mathbf{K}}^\mathrm{d}_{1,0})^\mathsf{T}$ $\alpha\underline{\tilde{b}}^\mathrm{d}_{0,1}$
$\tilde{\mathbf{K}}_{0,0}$ $\alpha\underline{\tilde{b}}_{0,0}$
$\mathbf{K}_\Omega$
$\hat{\mathbf{m}}$
$\hat{\mathbf{c}}$

Figure 1. Diagram of simplified U-Net-proxy model (SUM). Here, each encoding step is composed by a convolution layer with a kernel $\mathbf{K}_{i,l}$, where $i \in [0, 1]$ and l represents the decomposition level. Each convolution layer is followed by a soft shrinkage, where the threshold level/bias value is given by $\alpha\underline{b}_l$, where α is a scalar value. After the convolution plus activation layer, the signal is down-sampled with the discrete wavelet transform (DWT), which is implemented with the convolution kernels $\mathbf{W}_\mathrm{H}$ and $\mathbf{W}_\mathrm{L}$ that are the low-frequency and high-frequency filter banks, respectively. The decoder has an approximately mirrored structure with respect to the encoder and its convolution filters and biases are denoted by $\tilde{\mathbf{K}}_l$ and $\underline{\tilde{b}}_l$, respectively. Moreover, the channel concatenation and reduction performed in the U-Net is represented by the convolution and sum with the decoder kernels $(\mathbf{K}^\mathrm{u}_{1,0})^\intercal$ and $(\mathbf{K}^\mathrm{d}_{1,0})^\intercal$. In addition, the up-sampling is performed with the inverse discrete wavelet transform, whose kernels are represented by $\tilde{\mathbf{W}}_\mathrm{L}$ and $\tilde{\mathbf{W}}_\mathrm{H}$. Finally, the output layer is given by a 1×1 convolution which scales the output of the encoder-decoder network and saturates the value between zero and unity with a clipping operation. Note that the model is composed by four decomposition levels, but not all of them are displayed due to space constraints.

3.2. Simplified U-Net-proxy model

As mentioned in Section 1, to study the signal processing behavior in U-Net-like models is challenging, because this model is nonlinear and has signal-dependent behavior. In order to overcome these limitations, this paper employs a simplified architecture that is addressed as *simplified U-Net-proxy model* (SUM). The referred SUM model, shown in Fig. 1, is based on the tight frame U-Net (Han & Ye, 2018) and the *learned wavelet frame shrinkage network* (LWFSN) (Zavala-Mondragón et al., 2022). In this model, soft shrinkage functions are employed as nonlinearities and the discrete wavelet transform is employed for up and down-sampling.

The SUM model employed here has three main advantages when compared to the conventional U-Net for understanding the operation of this type of models. (1) The SUM employs the discrete wavelet transform (DWT) as up/down-sampling. This operation is linear, which contrasts with the nonlinear behavior of the max-pooling in the U-Net. Furthermore, the DWT overcomes the limited capacity of the pooling/unpooling in the original U-Net to propagate high-frequency information (Han & Ye, 2018; Etmann et al., 2020; Zavala-Mondragón et al., 2023). (2) The SUM model does not employ normalization layers, which eliminates signal-dependent behaviors and avoids the introduction of offsets to the signal. (3) The SUM model employs the soft shrinkage functions as nonlinearities, instead of the ReLUs employed in the U-Net. This design choice allows to independently study the linear and nonlinear processing performed by the SUM. Specifically, shrinkage functions become the identity operation when the threshold level/bias is set to zero, which allows to characterize the impulse and frequency responses of the filters of the CNN (Zavala-Mondragón et al., 2022). Additional discussion on ReLU and soft shrinkage functions, as well as on the impulse response in CNNs in Appendices A and B, respectively. In the subsequent equations, the SUM model is represented by $\mathrm{S}(\cdot)$, and its encoder-decoder module as $\mathrm{ED}_\mathrm{S}(\cdot)$.

To conclude this section, note in Fig. 1 that the SUM network has two outputs. The first output is the estimated segmentation $\hat{\mathbf{m}}$, while the second output is the intermediate signal $\hat{\mathbf{c}}$ that is produced by the encoder-decoder module $\mathrm{ED}_{\mathrm{S}(\alpha)}(\cdot)$, which is embedded within the SUM module. In addition, in Fig. 1 it is visible that the threshold levels in the activations of $\mathrm{ED}_{\mathrm{S}(\alpha)}(\cdot)$ are scaled by factor $\alpha \in [0, 1]$. This variable is added to allow to modulate the sparsity-enforcement property of the nonlinearities in the model.

3.3. Training the proposed model

The loss term $\mathcal{L}$ employed to train the models in this paper is defined by

$$\mathcal{L} = \mathcal{L}_\mathrm{Seg}(\hat{\mathbf{m}}, \mathbf{m}) + \lambda_\mathrm{Rec}\mathcal{R}_\mathrm{Rec}(\hat{\mathbf{x}}, \mathbf{x}). \tag{13}$$

Here, $\mathcal{L}_\mathrm{Seg}(\cdot)$ stands for the *segmentation* loss, which in this paper is the soft intersection over union (IoU) (Huang et al., 2019). The second term is the *reconstruction regularization* $\mathcal{R}_\mathrm{Rec}(\cdot)$. This term is is employed to highlight the similarity between the operation of denosing and segmentation models, by forcing the encoder and decoder filters of $\mathrm{EDS}_{(\cdot)}(\cdot)$ to only decompose and reconstruct the signal, just as sparse denoising models do (Donoho & Johnstone, 1994; Chang et al., 2000). The term $\mathcal{R}_{\text{Rec}}(\cdot)$ is defined by

$$\mathcal{R}_{\text{Rec}} = \|\mathbf{x} - \hat{\mathbf{x}}\|_1, \tag{14}$$

in which $\|\cdot\|_1$ is the L_1 norm and $\hat{\mathbf{x}}$ is defined by

$$\hat{\mathbf{x}} = \mathrm{ED}_{\mathrm{S}(\alpha=0)}(\mathbf{x}). \tag{15}$$

Here, as specified in Section 3.2, the scalar α modulates the sparsity-enforcing properties of the soft shrinkage function. Consequently, setting $\alpha = 0$ causes that the biases within the activation layers are set to zero, which disables their sparsifying behavior. On the other hand, a value of $\alpha = 1$ means that the model operates with the bias values that have been learned during training.

4. Experiments and results

This section defines the experiments performed in this paper. The code used for training the models and to generate the tables and figures is available at https://github.com/LuisAlbertZM/ICML-segmentation-CNNs-are-denoising.

4.1. Dataset

The experiments employ a toy dataset based on slices of the training set of the BRATS 2021 dataset (Baid et al., 2021; Menze et al., 2014; Bakas et al., 2017), which has been employed for segmentation of lesions in multi-modal brain MRI images. The dataset contains T1, T2, T1 post-contrast-weighted and T2-FLAIR images. The scope of this paper is limited to explore networks with single input/output channels. Consequently, we have selected T2-FLAIR images as input only and all label lesions have been merged into a single *anomalous* class. As pre-processing step, all the scans are resized to 128×128×128 voxels and the intensity is normalized within the range 0-255 and quantized to 8 bits. In addition, for this dataset, two slices of each of the 1,251 patients have been extracted, one where the largest part of the tumor is shown and a slice that contains only a few tumor voxels. The patients are split in about one third for training, validation and testing, which results in a total of 834 slices for training, 834 for validation and 832 for testing, the data splits do not have overlapping patients. Prior to propagating the slices through the CNNs, the image data are scaled between zero and unity.

4.2. Experiment description and reference models

In this paper, two main experiments are conducted. The first experiment has the purpose to show that segmentation models follow the signal processing behavior described in Section 3.1. For this test, two SUM model instances are trained, the first instance is trained only with the segmentation loss (i.e. λ_{Rec}=0) and it is referred to as SUM_{Seg}, while the filters of the second SUM model are regularized to reconstruct the input signal by setting λ_{Rec}=0.35 (referred to as SUM_{Rec}). Finally, the U-Net and its version without normalization layers ($\text{U-Net}_{\text{NoBN}}$) have been added as baseline reference and are trained with the segmentation loss.

In order to provide more evidence for our claim that denoising and segmentation models behave operate in a similar way, the second experiment in this paper considers to train the SUM CNN for image denoising. In this case, the input to the network is a slice contaminated with Gaussian noise with standard deviation equal to 10% of the dynamic range of the image, while the ground-truth is the noise-free picture. For this experiment, the loss function is the L1 loss between the processed and noise-free images.

All the models implemented in his paper contain 4 down/upsampling stages, the SUM models have 8 feature maps after the first convolution layer, while the U-Net models have 16 feature maps. The number of feature maps duplicate with every decomposition level, just as in the original U-Net (Ronneberger et al., 2015). In addition, all models contain a 1×1 convolution combined with a clipping operation as output layer. The method employed to compute the impulse response of the SUM model is described in Appendix B. In addition, to characterize the impulse and frequency responses, this paper proposes the space-domain and frequency-domain spreads (SDV and FDS, respectively), as well as the frequency-domain variation (FDV). The referred features are described in Appendix C.

4.3. Training

For the experiments, we have trained all the models for 300 epochs with AdamW optimization (Loshchilov & Hutter, 2017). For the specified optimizer, the learning rate linearly decays from an initial value of 3.5×10^{-4} to zero and the weight decay value is set to 0.25. The batch size employed for training for all models is set to 8 samples. Furthermore, as data augmentation, rotations by 90, 180 and 270 degrees and flips in the horizontal dimensions have been performed. Each of the data augmentations has a probability to happen of 50% for every sample in the training/validation sets. In Experiment 1 each model is trained 5 times to estimate the confidence intervals. Moreover, in Experiment 2 the model is trained only once since it provides a result of more qualitative nature.

4.4. Experiment 1: Signal processing behavior of U-Net-like models

Overall performance: With the described settings, we have trained all the specified models. Fig 2 displays an example slice that has been processed with each of the models being tested. In addition, the computed Dice score and Jaccard index/intersection over union computed (IoU) are

Table 1. Dice score and intersection over union (IoU) measured over the segmentation estimates generated by the U-Net, U-Net$_{\mathrm{NoBN}}$, SUM$_{\mathrm{Seg}}$ and SUM$_{\mathrm{Rec}}$ models. The performance values are obtained by averaging the performance after 5 training cycles.

Model	U-Net	U-Net$_{\mathrm{NoBN}}$	SUM$_{\mathrm{Seg}}$	SUM$_{\mathrm{Rec}}$
Dice Score	0.884 ± 0.004	0.859 ± 0.006	0.849 ± 0.010	0.859 ± 0.007
IoU	0.793 ± 0.007	0.752 ± 0.009	0.738 ± 0.015	0.752 ± 0.010

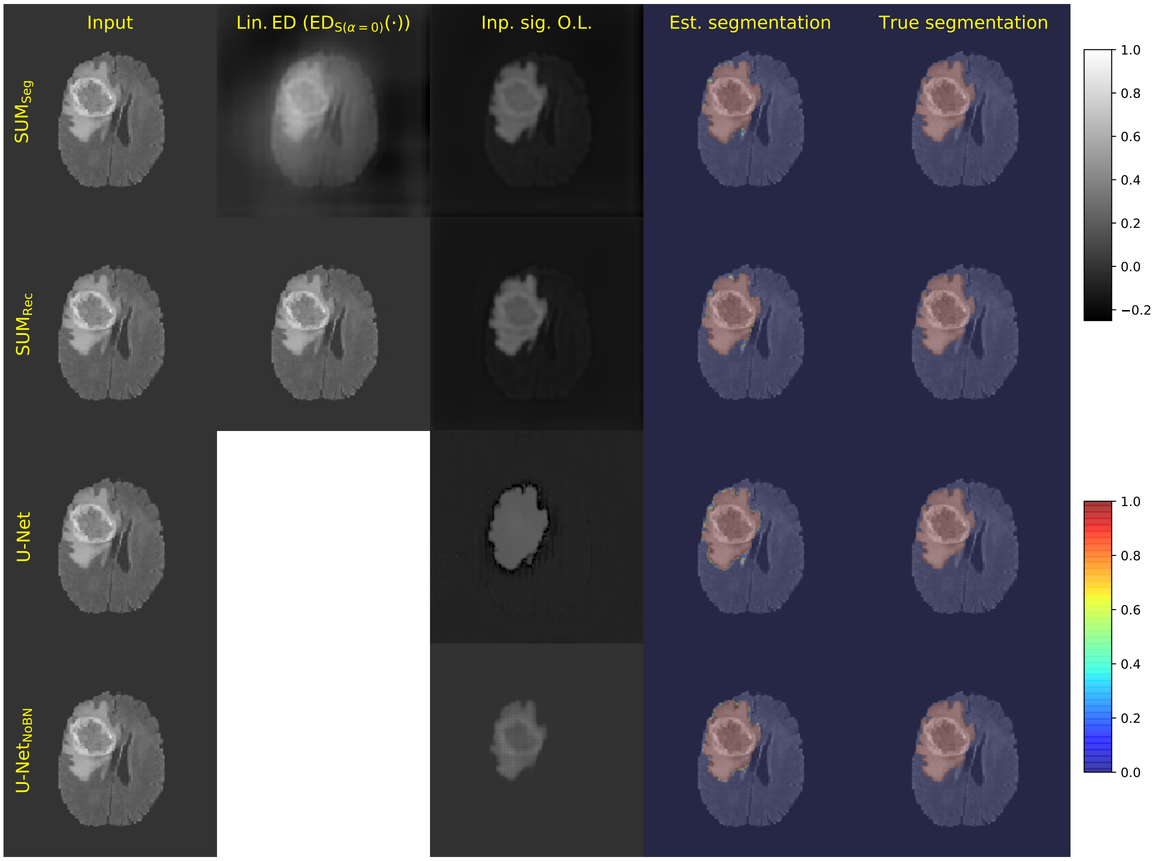

Figure 2. A slice of the test set processed with the SUM$_{\mathrm{Seg}}$, SUM$_{\mathrm{Est}}$, U-Net and U-Net$_{\mathrm{NoBN}}$ models. In the figure, column *Input* displays the input to each of the models. Column *Lin. ED. (ED$_{S(\alpha=0)}(\cdot)$)* refers to the output of the encoder-decoder module when the bias of the nonlinearities is set to zero. Column *Inp. sig. O.L.* is the input signal to the output layer. Finally, the last two columns (*Est. segmentation* and *True segmentation*) show the estimated and true segmentation masks, respectively. For display, images in the columns *Lin. ED.* (ED$_{S(\alpha=0)}(\cdot)$ and *Inp. sig. O.L.* are scaled in such way that the similarity with the input image is more visible. Finally, it should be noted that the *Reconstruction* signal is missing in the U-Net models, this signal has been excluded, because these models are too complex to be analyzed. Appendix F, shows the segmentations an additional subject. Furthermore, Appendix D provides complementary information in the signal flow employed to compute the images that compose this figure.

shown in Table 1. Here, it can be observed that the best performing model in terms of Dice score and intersection over union (IoU) is the U-Net, followed by the SUM$_{\mathrm{Rec}}$ model and the U-Net$_{\mathrm{NoBN}}$. This result is sensible because the SUM model is simpler and has less parameters than the conventional U-Net. In addition, it should be noted, that the absence of normalization layers in the U-Net$_{\mathrm{NoBN}}$ causes an important drop in performance. However, Appendix E shows that despite the fact that the U-Net is the best-performing model for this experiment, it is also the least robust to unseen distortions.

Signal processing within the model: The column *Inp. sig. O.L.* in Fig. 2 shows the signal that results from processing the input image with the encoder-decoder section of each of the tested CNNs. It can be observed that the encoder-decoder module of all the segmentation models generate a signal that is reminiscent of the area delimited by

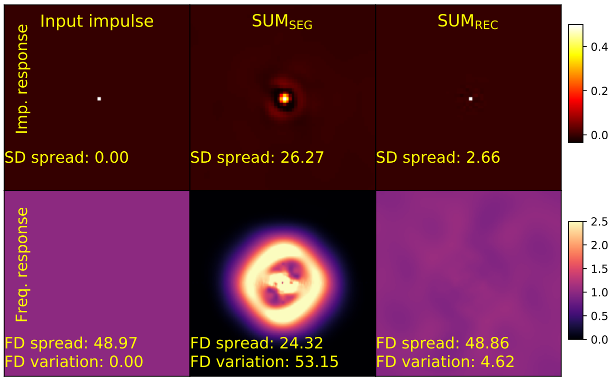

Figure 3. Impulse responses of the encoder-decoder model within the $\mathrm{SUM}_{\mathrm{Seg}}$ and $\mathrm{SUM}_{\mathrm{Rec}}$ CNNs for one of the training cycles. The first row (*Imp. response*) depicts the impulse response of the $\mathrm{SUM}_{\mathrm{Seg}}$ and $\mathrm{SUM}_{\mathrm{Rec}}$ models, respectively. The second row (*Freq. response*) shows the magnitude of the Fourier transform of the impulse response. It can be noted that the frequency response of the $\mathrm{SUM}_{\mathrm{Seg}}$ model focuses on a narrower frequency range that is more focused in the lower and intermediate frequency ranges. In contrast, filters of the $\mathrm{SUM}_{\mathrm{Rec}}$ propagate more frequencies.

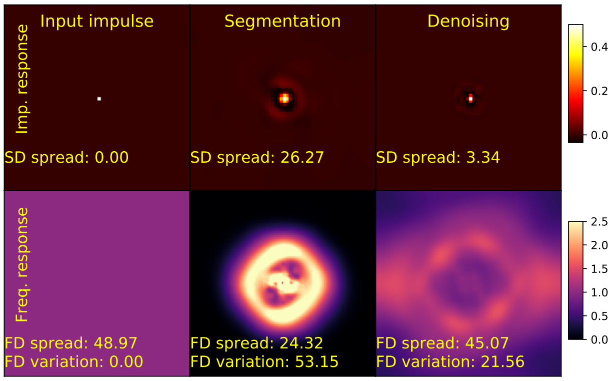

Figure 4. Impulse responses of a SUM model trained for image segmentation and denoising. Note that the model trained for segmentation has a wider response in space and it reconstructs less frequency bands. In contrast, the model trained for image denoising has a more localized impulse response and its frequency response is broader.

the segmentation mask, in which the non-anomalous regions are pushed towards negative values that are suppressed by the saturating output layer. This supports our hypothesis that the encoder-decoder part of the segmentation CNN behaves as some sort of matched filter which boosts the foreground signal (the anomaly in this case), and/or suppresses the rest of the image. Moreover, the column *Lin. ED.* $ED_{S_{(\alpha=0)}}(\cdot)$ in Fig. 2 shows that when the SUM model is trained only with segmentation loss ($\mathrm{SUM}_{\mathrm{Seg}}$), the filters of the encoder and decoder allow to propagate most of the signal content of the signal input. However, the filters of the encoder-decoder model of $\mathrm{SUM}_{\mathrm{Seg}}$ do not reconstruct the signal perfectly. This behavior is more obvious when observing the impulse response of the encoder-decoder filters of the $\mathrm{SUM}_{\mathrm{Seg}}$ and $\mathrm{SUM}_{\mathrm{Rec}}$ models, displayed in Fig. 3. In the figure, it is visible that the filters of the $\mathrm{SUM}_{\mathrm{Seg}}$ propagate only a narrow set of frequencies, while the filters of the $\mathrm{SUM}_{\mathrm{Rec}}$ propagate a broader frequency range. In spite of this, when enabling the nonlinearities of the model, the encoder-decoder part of the SUM suppresses and boosts part of the signal, which generates an image that is reminiscent of the anomaly despite the fact that the filtering that they perform differs.

In Fig. 2, it is visible that in the SUM model where signal reconstruction is enforced ($\mathrm{SUM}_{\mathrm{Rec}}$), the convolution filters of the model are able to approximate the input signal. This means that in this case, the nonlinearities are the main driver of the segmentation, which is analogous to image denosing models based on wavelets. Moreover, when analyzing the impulse and frequency responses of this model (see Fig. 3), it can be observed that –just as expected– the reconstruction regularization forces the CNN to reconstruct more frequencies than the model trained only with the segmentation loss.

4.5. Experiment 2: Contrasting the behavior of CNNs trained for image denoising and segmentation

The SUM model for image denoising has been trained with the procedure described in Section 4.2. With the resulting models, we have processed one of the slices in the dataset (see Fig. 5), where it is visible that the linear part of both models (see column *Lin ED* ($ED_{S_{\alpha=0}(\cdot)}$)) propagates most of the image content. However, in the case of the segmentation model, the reconstruction is less accurate than in the denoising model, which is a sensible result because perfect reconstruction may not be needed for producing segmentation estimates. In contrast, the denoising model propagates the input signal almost perfectly and it is even able to reconstruct the noise. Furthermore, when the nonlinear behavior of the models is enabled, the output of the encoder-decoder module of the SUM model trained for segmentation, becomes a signal where the segmentation model attenuates the background and pushes it towards negative values, while preserving the section of the image that corresponds to the foreground (see column *Inp. sig. O.L.*). In an analogous way, the non-linearities of the denoising model eliminate the noise, while allowing to propagate the signal of interest. The result shown in Fig. 5 is complemented by the impulse responses of the models trained for segmentation and denoising that is displayed in Fig. 4, where it can be observed that the impulse response of the segmentation model only propagates a narrow band of frequencies that has more emphasis in the low and mid-frequency bands, while the

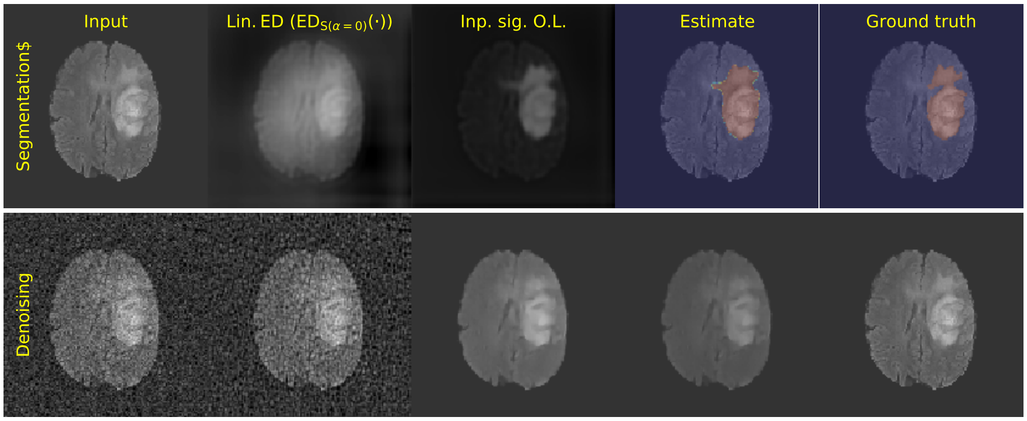

Figure 5. Comparison between the operation of a SUM model trained for image segmentation (top) and denoising (bottom). In column (*Lin. ED (ED*$_{S(\alpha=0)}(\cdot)$*))*, it can be observed that the convolution filters of the encoder and decoder of the segmentation model can reconstruct a blurred version of the input. In contrast, the model trained for image denosing, is able to propagate more frequencies and it is even able to reconstruct the noise. When the non-linearities are enabled (column *Inp. sig O.L.*), the encoder-decoder module within the SUM trained for image segmentation attenuates the non-anomalous areas and sends them to the negative image range. In a similar process, the model trained for image denosing suppresses the noise, while preserving the signal of interest. Finally, the output layer of both models scale the signal, and in the case of the segmentation model, the signal is saturated.

denoising model has a more uniform frequency response, which means that its convolution filters of this model can propagate a broader frequency range. Note that the impulse response of the model trained for image denoising is similar to the SUM model that is trained for image segmentation where the filters have been regularized for signal reconstruction SUM$_{\text{Rec}}$

5. Discussion

The results of the experiments in Section 4 confirm that segmentation encoder-decoder networks behave akin to noise reduction algorithms based on sparse modeling. However, in contrast with sparse modeling techniques employed in noise reduction (e.g. framelets), the filters banks in segmentation networks do not need to approximate perfect signal reconstruction. Furthermore, it can be observed that the described operation of segmentation CNNs is also similar to the matched filtering discussed in Appendix I.

It should be noted that the impulse responses shown here can be influenced by training parameters such as the learning rate, weight-decay settings and data augmentation, which means that, in practice, the model can converge to solutions with different signal processing behavior. Complementary experiments that have been performed to study this behavior are shown in Appendices G and H.

It can be observed that the similarity in the processing performed by models employed for image denoising and segmentation may explain the recent successes in co-learning (Buchholz et al., 2020; Ye et al., 2023), where it is shown that training a CNN for simultaneous denoising and segmentation benefits the performance and generalization of the segmentation part of the model. Furthermore, the authors hypothesize that this work can be used also as a first step to understand in more depth the reverse diffusion process performed in diffusion models (Sohl-Dickstein et al., 2015) for image generation and segmentation (Wu et al., 2024; Baranchuk et al., 2022).

The performed study has the following limitations. (1) The experiments are based on a proxy model that employs as nonlinearities soft shrinkages instead of ReLUs, which may not capture completely the behavior of ReLU-based models. However, we consider that this approximation is good enough to sufficiently characterize the behavior of this U-Net-like of models. (2) The studies presented here avoid batch normalization, which is a common element in CNNs. (3) This paper focuses only on binary segmentation models.

6. Conclusions and future work

This paper demonstrates that the signal processing in encoder-decoder CNNs employed in segmentation and image denoising is analogous and is described by the following steps. (1) The convolution filters of the encoder and decoder provide a decomposition and reconstruction which is useful for estimating the signal of interest. This is a process akin to the framelet decomposition. However, in encoder-decoder CNNs employed in image segmentation, the linear part of the model does not need to approximate perfect reconstruction, as framelet-based denoising models do. (2) The nonlinearities/activation functions suppress the background and/or boost the signal of interest in the feature maps, while the decoder reconstructs the signal. (3) The output layer provides scaling and saturation and sets the output values to the range between zero and unity. To the best of our knowledge, this is the first time that this behavior has been studied in segmentation models.

The authors envision two different research tracks as follow-up work. The first of these branches considers to further explore SUMs and to study the feasibility of providing designers with quantitative metrics that characterize the biases of these models for image segmentation. Specifically, we hypothesize that it is possible to discover biases that are learned to by the SUM model to specific textures and/or object sizes, which can be find by greedily finding the feature maps that are the most relevant to produce the segmentation and by masking them while obtaining the impulse response of the filters of the SUM model.

The second branch of envisioned future work considers to employ SUM models to approximate the behavior of trained U-Nets. In this way, it may be possible to study the operation of trained models, as well as their biases. Finally, further research will be performed to extend this analysis approach to multi-class segmentation problems as well as to more challenging segmentation datasets.

Impact Statement

The development of this work is motivated by the need to have better understanding of CNN models, which could potentially be employed in the chain of decision-making in medicine. Consequently, the authors envision only positive impacts for this research, since it could provide early insights on the operation and limitations of the CNNs used for critical applications. Furthermore, we envision that this work can be further employed to improve the robustness, reliability, and trustworthiness of deep neural networks.

Acknowledgements

The authors would like to thank the reviewers for the useful comments.

Author contributions statement

1. Luis A. Zavala-Mondragón developed the original concept of the article, designed and implemented the experiments and wrote the article.
2. Ruud van Sloun provided technical advise, suggested the experiments on the impulse response that are related to Appendices C, H and G and provided feedback on the article.
3. Peter de With provided technical advise and extensive reviews on the text of the paper.
4. Fons van der Sommen provided technical advise, feedback on the paper and suggested to write Appendices A and D.

References

Baid, U., Ghodasara, S., Mohan, S., Bilello, M., Calabrese, E., Colak, E., Farahani, K., Kalpathy-Cramer, J., Kitamura, F. C., Pati, S., et al. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv:2107.02314*, 2021.

Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J. S., Freymann, J. B., Farahani, K., and Davatzikos, C. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4(1):1–13, 2017.

Baranchuk, D., Voynov, A., Rubachev, I., Khrulkov, V., and Babenko, A. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=SlxSY2UZQT.

Benz, P., Zhang, C., Karjauv, A., and Kweon, I. S. Revisiting batch normalization for improving corruption robustness. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 494–503, 2021a.

Benz, P., Zhang, C., and Kweon, I. S. Batch normalization increases adversarial vulnerability and decreases adversarial transferability: A non-robust feature perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7818–7827, 2021b.

Buchholz, T.-O., Prakash, M., Schmidt, D., Krull, A., and Jug, F. Denoiseg: joint denoising and segmentation. In *European Conference on Computer Vision*, pp. 324–337. Springer, 2020.

Chang, S. G., Yu, B., and Vetterli, M. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, 2000.

Daubechies, I., Defrise, M., and De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Donoho, D. L. and Johnstone, J. M. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

Etmann, C., Ke, R., and Schönlieb, C.-B. iunets: Fully invertible u-nets with learnable up-and downsampling. *arXiv preprint arXiv:2005.05220*, 2020.

Gregor, K. and LeCun, Y. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pp. 399–406, 2010.

Han, Y. and Ye, J. C. Framing u-net via deep convolutional framelets: Application to sparse-view ct. *IEEE Transactions on Medical Imaging*, 37(6):1418–1429, 2018.

Huang, Y., Tang, Z., Chen, D., Su, K., and Chen, C. Batching soft iou for training semantic segmentation networks. *IEEE Signal Processing Letters*, 27:66–70, 2019.

Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26 (9):4509–4522, 2017.

Kascenas, A., Pugeault, N., and O'Neil, A. Q. Denoising autoencoders for unsupervised anomaly detection in brain mri. In *International Conference on Medical Imaging with Deep Learning*, pp. 653–664. PMLR, 2022.

Khamaiseh, S. Y., Bagagem, D., Al-Alaj, A., Mancino, M., and Alomari, H. W. Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification. *IEEE Access*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

McCann, M. T., Jin, K. H., and Unser, M. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6):85–95, 2017. doi: 10.1109/MSP.2017.2739299.

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

Papyan, V., Romano, Y., and Elad, M. Convolutional neural networks analyzed via convolutional sparse coding. *The Journal of Machine Learning Research*, 18(1):2887–2938, 2017.

Pinson, H., Lenaerts, J., and Ginis, V. Linear cnns discover the statistical structure of the dataset using only the most dominant frequencies. In *International Conference on Machine Learning*, pp. 27876–27906. PMLR, 2023.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Steven, K. Fundamentals of statistical signal processing: Detection theory. *University of Rhode Island: Prentice Hall PTR*, 1998.

Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

Wang, H., Zhang, A., Zheng, S., Shi, X., Li, M., and Wang, Z. Removing batch normalization boosts adversarial training. In *International Conference on Machine Learning*, pp. 23433–23445. PMLR, 2022.

Wang, Z., Chen, J., and Hoi, S. C. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020.

Wu, J., FU, R., Fang, H., Zhang, Y., Yang, Y., Xiong, H., Liu, H., and Xu, Y. Medsegdiff: Medical image segmentation with diffusion probabilistic model. In Oguz, I., Noble, J., Li, X., Styner, M., Baumgartner, C., Rusu, M., Heinmann, T., Kontos, D., Landman, B., and Dawant, B. (eds.), *Medical Imaging with Deep Learning*, volume 227 of *Proceedings of Machine Learning Research*, pp. 1623–1639. PMLR, 10–12 Jul 2024. URL `https://proceedings.mlr.press/v227/wu24a.html`.

Ye, J. C., Han, Y., and Cha, E. Deep convolutional framelets: A general deep learning framework for inverse problems. *SIAM Journal on Imaging Sciences*, 11(2):991–1048, 2018.

Ye, T., Wang, J., and Yi, J. Deep learning network for parallel self-denoising and segmentation in visible light optical coherence tomography of the human retina. *Biomedical Optics Express*, 14(11):6088–6099, 2023.

Zavala-Mondragón, L. A., Rongen, P., Bescos, J. O., de With, P. H., and van der Sommen, F. Noise reduction in CT using learned wavelet-frame shrinkage networks. *IEEE Transactions on Medical Imaging*, 41(8): 2048–2066, 2022.

Zavala-Mondragón, L. A., de With, P. H., and van der Sommen, F. A signal processing interpretation of noise-reduction convolutional neural networks: Exploring the mathematical formulation of encoding-decoding cnns. *IEEE Signal Processing Magazine*, 40(7):38–63, 2023. doi: 10.1109/MSP.2023.3300100.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. doi: 10.1109/TIP.2017.2662206.

A. ReLU and Shrinkage functions

This appendix discusses the similarities and differences between rectified linear units (ReLU) and soft shrinkage activations. To start this discussion, the ReLU activation $(\cdot)_+$ is defined by

$$(x-b)_+ = \max\{x-b, 0\}, \tag{16}$$

in which $\max\{\cdot\}$ the maximum operation. Moreover, the soft shrinkage $\mathcal{S}_{(\cdot)}(\cdot)$ activtion is mathematically described by

$$\mathcal{S}_{(b)}(x) = \text{sign}(x) \cdot \max\{|x| - b, 0\}, \tag{17}$$

which is equivalent to

$$\mathcal{S}_{(b)}(x) = (x-b)_+ - (-x-b)_+. \tag{18}$$

Here, it becomes evident that the soft-shrinkage is equivalent to the ReLU when $x - b \geq 0$. However, the soft shrinkage is an antisymmetric function, while the ReLU activation is not. The referred behavior is more evident when observing the graphical representations in Fig. 6.

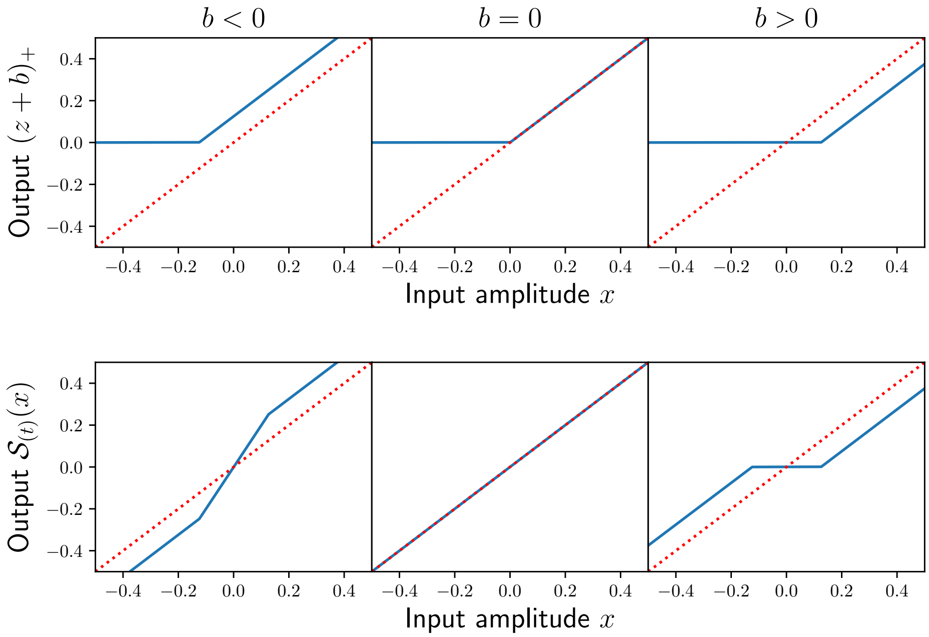

Figure 6. Input and output relationships for ReLU (top) and soft shrinkage functions (bottom) for negative, null and positive bias/threshold levels. It can be noted that the ReLU and soft shrinkage functions behave identically for positive inputs. However, the soft shrinkage is an antysimmetric function with respect to the line $x = 0$, while the ReLU sets to zero any negative values.

B. Impulse response of a shrinkage-based model

The impulse response of a system is a common technique to characterize the filtering behavior of linear time-invariant (LTI) systems. In the context of convolutional neural networks (CNNs), it has been used to study the frequency coverage of the filter banks that compose the encoder and decoder parts of a CNN (Zavala-Mondragón et al., 2022; Zavala-Mondragón et al., 2023). It should be noted that this technique is used exclusively to characterize linear systems. Consequently, only can be applied in CNNs based on linear, shrinkage or (soft) clipping activations, because these activation functions allow to obtain linear models by altering their bias/threshold level (Zavala-Mondragón et al., 2023).

In order to explain how the impulse response of a shrinkage-based CNN is computed, this appendix presents of an arbitrary CNN $\mathrm{C}(\cdot)$, which is defined by

$$\mathrm{C}(\mathbf{x}) = \mathrm{D}(\mathrm{E}(\mathbf{x})). \tag{19}$$

Here, the decoder $\mathrm{D}(\cdot)$ is

$$\mathrm{D}(\mathbf{y}) = \mathcal{S}_{(\underline{\tilde{b}}_0)}\Big(\tilde{\mathbf{K}}_0 \mathcal{S}_{(\underline{\tilde{b}}_1)}\Big(\tilde{\mathbf{K}}_1 \mathbf{y}\Big)\Big), \tag{20}$$

in which $\tilde{\mathbf{K}}_0$, $\tilde{\mathbf{K}}_1$ are the convolution filters of the decoder, while $\underline{\tilde{b}}_0$ and $\underline{\tilde{b}}_1$ are their respective bias/threshold vectors. In addition, $\mathcal{S}_{(\cdot)}(\cdot)$ represents the soft threshold function. Complementary to the decoder, the encoder $\mathrm{E}(\cdot)$ is mathematically described by

$$\mathrm{E}(\mathbf{x}) = \mathcal{S}_{(\underline{b}_1)}\Big(\mathbf{K}_1 \mathcal{S}_{(\underline{b}_0)}\Big(\mathbf{K}_0 \mathbf{x}\Big)\Big). \tag{21}$$

Akin to the decoder, the encoder filters and corresponding biases/threshold levels are defined by $\mathbf{K}_0$, $\mathbf{K}_1$, $\underline{b}_0$ and $\underline{b}_1$.

In order to characterize the impulse response of the model described in Eq. (19), the input $\mathbf{x}$ is replaced by the convolution identity (Dirac's delta function), represented by $\mathbf{I}$. Furthermore, the biases $\underline{\tilde{b}}_1$, $\underline{\tilde{b}}_0$, $\underline{b}_1$, $\underline{b}_0$ are set to zero. Consequently, under this conditions, the output of the model $\mathrm{C}_{(\underline{b}=0)}(\mathbf{I})$ is defined by

$$\mathrm{C}_{(\underline{b}=0)}(\mathbf{I}) = \mathcal{S}_{(0)}\Big(\tilde{\mathbf{K}}_0 \mathcal{S}_{(0)}\Big(\tilde{\mathbf{K}}_1 \mathcal{S}_{(0)}\Big(\mathbf{K}_1 \mathcal{S}_{(0)}\Big(\mathbf{K}_0 \mathbf{I}\Big)\Big)\Big)\Big), \tag{22}$$

where, $\mathbf{x} = \mathcal{S}_{(0)}(\mathbf{x})$ for any value of $\mathbf{x}$. Consequently, Eq. (22) becomes

$$\mathrm{C}_{(\underline{b}=0)}(\mathbf{I}) = \tilde{\mathbf{K}}_0 \tilde{\mathbf{K}}_1 \mathbf{K}_1 \mathbf{K}_0. \tag{23}$$

Here, it can be noted that the entire response of the system depends purely on linear convolutions. Therefore, Eq. (23) is reduced to

$$\mathrm{C}_{(\underline{b}=0)}(\mathbf{I}) = \mathbf{r}, \tag{24}$$

in which $\mathbf{r}$ is the impulse response of the shrinkage-based CNN $\mathrm{C}(\cdot)$. It should be noted that the impulse response $\mathbf{r}$ is a filter which captures the global behavior of the filter banks that compose $\mathrm{C}(\cdot)$. Finally, the frequency response of the system is obtained by computing $\mathcal{F}\{\mathrm{C}_{(\underline{b}=0)}(\mathbf{I})\}$, where $\mathcal{F}\{\cdot\}$ is the Fourier transform and its magnitude indicates the spatial frequencies that can be propagated through the filters of the CNN.

C. Signal spread and variation features

In order to establish a quantitative approach to characterize the impulse response $\mathbf{r}$ of the filter banks composing a CNN, this paper suggests to employ the signal spread of the impulse response in the frequency and spatial domains, as well as the variation of the frequency-domain representation of the impulse response.

The *frequency-domain variation* (FDV) of the impulse response employed in this paper is defined by

$$\mathrm{FDV}(\mathbf{f}) = \frac{\|\mathbf{f}\|_{\mathrm{TV}}}{\|\mathbf{f}\|_{\mathrm{F}}}, \tag{25}$$

where $\mathbf{f} = |\mathcal{F}\{\mathbf{r}\}|$ is the magnitude of the Fourier transform of the impulse response $\mathbf{r}$, while $\|\cdot\|_{\mathrm{F}}$ is the Frobenious norm and $\|\cdot\|_{\mathrm{TV}}$ is the total variation norm (Rudin et al., 1992)

$$\|\mathbf{f}\|_{\mathrm{TV}} = \sum_{n=0}^{N-1} \sqrt{(\mathbf{d}_h[n])^2 + (\mathbf{d}_v[n])^2}, \tag{26}$$

in which h and v denote the image indices in the horizontal and vertical directions, respectively. Moreover the gradient of $\mathbf{f}$ in horizontal and vertical directions $\mathbf{d}_h$ and $\mathbf{d}_v$ in this paper is defined by

$$\begin{aligned} \mathbf{d}_h &= \mathbf{s}_h * \mathbf{x}, \\ \mathbf{d}_v &= \mathbf{s}_v * \mathbf{x}. \end{aligned} \tag{27}$$

Here, the Sobel filters in the horizontal and vertical directions $\mathbf{s}_h$ and $\mathbf{s}_v$ are defined by

$$\mathbf{s}_h = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}; \quad \mathbf{s}_v = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}. \tag{28}$$

As mentioned earlier, two additional measurements employed to characterize the impulse response are the spread of the amplitude of space and frequency representations of the impulse response. These features are referred to as *frequency-domain spread* (FDS) and *spatial-domain spread* (SDS) and are defined by

$$\begin{aligned}\mathrm{FDS}(\mathbf{f}) &= \sum_{h=0}^{N_h-1}\sum_{v=0}^{N_v-1}\left(\frac{\mathbf{f}(h,v)}{\sum_{n=0}^{N_h-1}\sum_{n=0}^{N_v-1}\mathbf{f}(h,v)}\right)\cdot\sqrt{(h-c_h)^2+(v-c_v)^2},\\ \mathrm{SDS}(\mathbf{r}) &= \sum_{h=0}^{N_h-1}\sum_{v=0}^{N_v-1}\left(\frac{\mathbf{r}(h,v)}{\sum_{n=0}^{N_h-1}\sum_{n=0}^{N_v-1}\mathbf{r}(h,v)}\right)\cdot\sqrt{(h-z_h)^2+(v-z_v)^2},\end{aligned} \tag{29}$$

in which c_h and c_v denote the horizontal and vertical coordinates the zero-th frequency, while z_h and z_v are the coordinates that correspond to the input impulse/Dirac delta that is supplied to the CNN to compute its impulse response.

D. Signal flow to generate the images in Fig. 2

Fig. 2 shows an input slice processed with different parts of the SUM model. This is is performed with the aim of understanding better the internal processing that this model performs. This appendix shows complementary information on the signal flow in the SUM model used to produce such figure. Specifically, there are two main cases to be studied. The first case considers when the non-linear part of the model is enabled (see the top part of Fig. 7). The second case is when the non-linear part of the model is disabled, which is employed for analyzing the processing performed by the linear part of the system (bottom part of Fig. 7). The titles of the columns in Fig. 2 are shown as outputs of the SUM model.

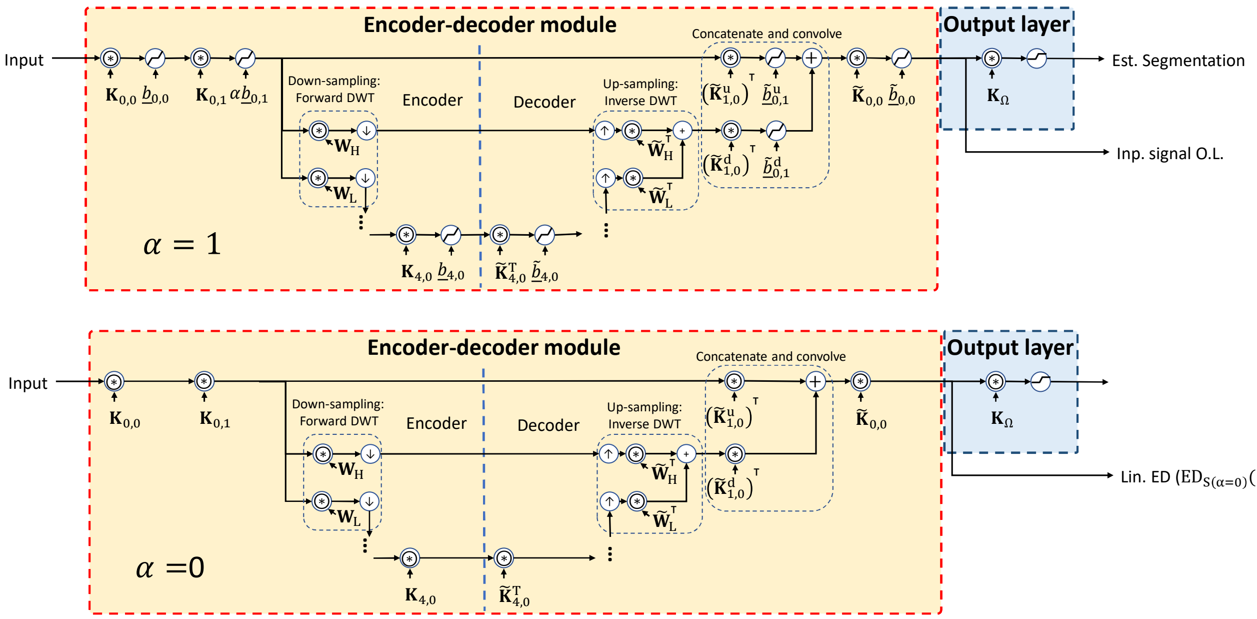
Encoder-decoder module
Output layer
Input
Concatenate and convolve
Est. Segmentation
$\mathbf{K}_{0,0}$ $\underline{b}_{0,0}$
$\mathbf{K}_{0,1}$ $\alpha\underline{b}_{0,1}$
Down-sampling: Forward DWT
Encoder
Decoder
Up-sampling: Inverse DWT
$(\tilde{\mathbf{K}}^{u}_{1,0})^T$ $\tilde{\underline{b}}^{u}_{0,1}$
$\tilde{\mathbf{K}}_{0,0}$ $\tilde{\underline{b}}_{0,0}$
$\mathbf{K}_{\Omega}$
Inp. signal O.L.
$\mathbf{W}_H$
$\mathbf{W}_L$
$\tilde{\mathbf{W}}_H^T$
$\tilde{\mathbf{W}}_L^T$
$(\tilde{\mathbf{K}}^{d}_{1,0})^T$ $\tilde{\underline{b}}^{d}_{0,1}$
$\alpha = 1$
$\mathbf{K}_{4,0}$ $\underline{b}_{4,0}$
$\tilde{\mathbf{K}}^T_{4,0}$ $\tilde{\underline{b}}_{4,0}$
Encoder-decoder module
Output layer
Input
Concatenate and convolve
$\mathbf{K}_{0,0}$
$\mathbf{K}_{0,1}$
Down-sampling: Forward DWT
Encoder
Decoder
Up-sampling: Inverse DWT
$(\tilde{\mathbf{K}}^{u}_{1,0})^T$
$\tilde{\mathbf{K}}_{0,0}$
$\mathbf{K}_{\Omega}$
Lin. ED ($\mathrm{ED}_{S(\alpha=0)}(\cdot)$)
$\mathbf{W}_H$
$\mathbf{W}_L$
$\tilde{\mathbf{W}}_H^T$
$\tilde{\mathbf{W}}_L^T$
$(\tilde{\mathbf{K}}^{d}_{1,0})^T$
$\alpha =0$
$\mathbf{K}_{4,0}$
$\tilde{\mathbf{K}}^T_{4,0}$

Figure 7. Signal flow in the SUM model to generate each of images in the columns shown in Fig. 2. The top figure shows the signal flow employed to compute the images in columns *Est. segmentation* and *Inp. sig. O.L.* in Fig. 2, in which the parameter α in Fig. 1 is set to unity. It can be noted that the column *Inp. sig. O.L.* is the input signal to the output layer, or the output of the encoder-decoder network. Moreover, signal *Est. segmentation* is the estimated segmentation by the network. The lower picture shows the signal flow to compute the column *Lin. ED ED*$_{S_{(\alpha=0)}}(\cdot)$. In this case, the parameter α is set to unity, which causes that all the shrinkage layers behave as the identity operation. Consequently, the images in column *Lin. ED ED*$_{S_{(\alpha=0)}}(\cdot)$ show the processing performed by linear part of the encoder-decoder network within the SUM model (i.e. the filter banks).

E. Robustness of the models in Experiment 1 to blur and noise

Experiment 1 in Section 4 shows that the U-Net model performs better than the U-Net$_{\text{NoBN}}$, SUM$_{\text{Seg}}$ and SUM$_{\text{Rec}}$ models when the test set images look similar to the training/validation sets (the images are not distorted). This appendix extends this result by evaluating the referred models in images that are blurred or noisy. Specifically, Table 2 summarizes the performance metrics when test set images have been corrupted with Gaussian blur ($\sigma_{\text{blur}} = 1$), or with additive Gaussian noise ($\sigma_{\text{noise}} = 10\%$ of the dynamic range of the image). The referred table shows that the U-Net is the least robust model to both, blurring and noise, whereas the other models are only affected marginally when the input is blurred. In the case when the input image is noisy, it can be observed that the most robust model is the SUM$_{\text{Rec}}$ model.

Note that the U-Net model with batch normalization (U-Net) layers is less robust to unseen corruptions than the U-Net model that does not have them (U-Net$_{\text{NoBN}}$). This observation suggests that, while batch normalization may improve the performance of the model, it makes the CNN less robust. This result matches the observations that have been made on the lack of robustness of batch-normalization-based models to adversarial attacks (Wang et al., 2022; Benz et al., 2021b;a).

Table 2. Dice score and intersection over union (IoU) measured over the segmentation estimates generated by the U-Net, U-Net$_{\text{NoBN}}$, SUM$_{\text{Seg}}$ and SUM$_{\text{Rec}}$ models. The performance values are obtained by averaging the performance after 5 training cycles. This table is an extension of Table 1, where the test set has been contaminated with image blur or noise in order to test the robustness of the compared models to unseen corruptions.

Not distorted input				
Model	U-Net	U-Net$_{\text{NoBN}}$	SUM$_{\text{Seg}}$	SUM$_{\text{Rec}}$
Dice Score	0.884 ± 0.004	0.859 ± 0.006	0.849 ± 0.010	0.859 ± 0.007
IoU	0.793 ± 0.007	0.752 ± 0.009	0.738 ± 0.015	0.752 ± 0.010
Gaussian blur ($\sigma_{\text{blur}} = 1$)				
Model	U-Net	U-Net$_{\text{NoBN}}$	SUM$_{\text{Seg}}$	SUM$_{\text{Rec}}$
Dice Score	0.793 ± 0.018	0.852 ± 0.005	0.841 ± 0.007	0.854 ± 0.005
IoU	0.657 ± 0.024	0.742 ± 0.008	0.726 ± 0.011	0.745 ± 0.007
Noise ($\sigma_{\text{Noise}} = 10\%$ of dynamic range of the images)				
Metric	U-Net	U-Net$_{\text{NoBN}}$	SUM$_{\text{Seg}}$	SUM$_{\text{Rec}}$
Dice Score	0.553 ± 0.100	0.776 ± 0.062	0.782 ± 0.038	0.804 ± 0.011
IoU	0.389 ± 0.099	0.638 ± 0.080	0.643 ± 0.049	0.673 ± 0.016

F. Additional slices processed with the models described in Experiment 1

Experiment 1 in Section 4 shows an example slice that has been processed with the SUM$_{\text{Seg}}$, SUM$_{\text{Rec}}$, U-Net and U-Net$_{\text{NoBN}}$ models. In order to show that the result is representative of other cases, this appendix shows an additional slice processed by the referred models, which is displayed in Fig. 8.

G. Impulse responses of the SUM model as a function of data augmentation

This appendix explores the effect of data augmentation in the characteristics of the impulse response of the SUM model. Specifically, the SUM model has been trained with the settings listed as follows. (1) No augmentations. (2) Only rotations and mirroring. (3) Only Gaussian blur with standard deviation σ_{blur} with unity value. (4) Only noise with standard deviation σ_{noise}=10% of the dynamic range of the image. (5) Rotations, mirroring, Gaussian blur and Noise.

The SUM model has been trained with the referred data augmentations and the performance metrics for all the models are summarized in Table 3. In the referred table, it is possible to observe that the model trained with all the augmentations (column *Rot. noi. blur*) is the best performing non-distorted and blurred images. Furthermore, the model trained with rotations and mirroring (column *Rot. mirr*) performs as second best when processing non-distorted and blurred inputs. In fact, the model trained with rotations performs better when presented with blurred images than the model only trained with blurred pictures. However, the model trained with rotations does not perform so good when processing noisy images.

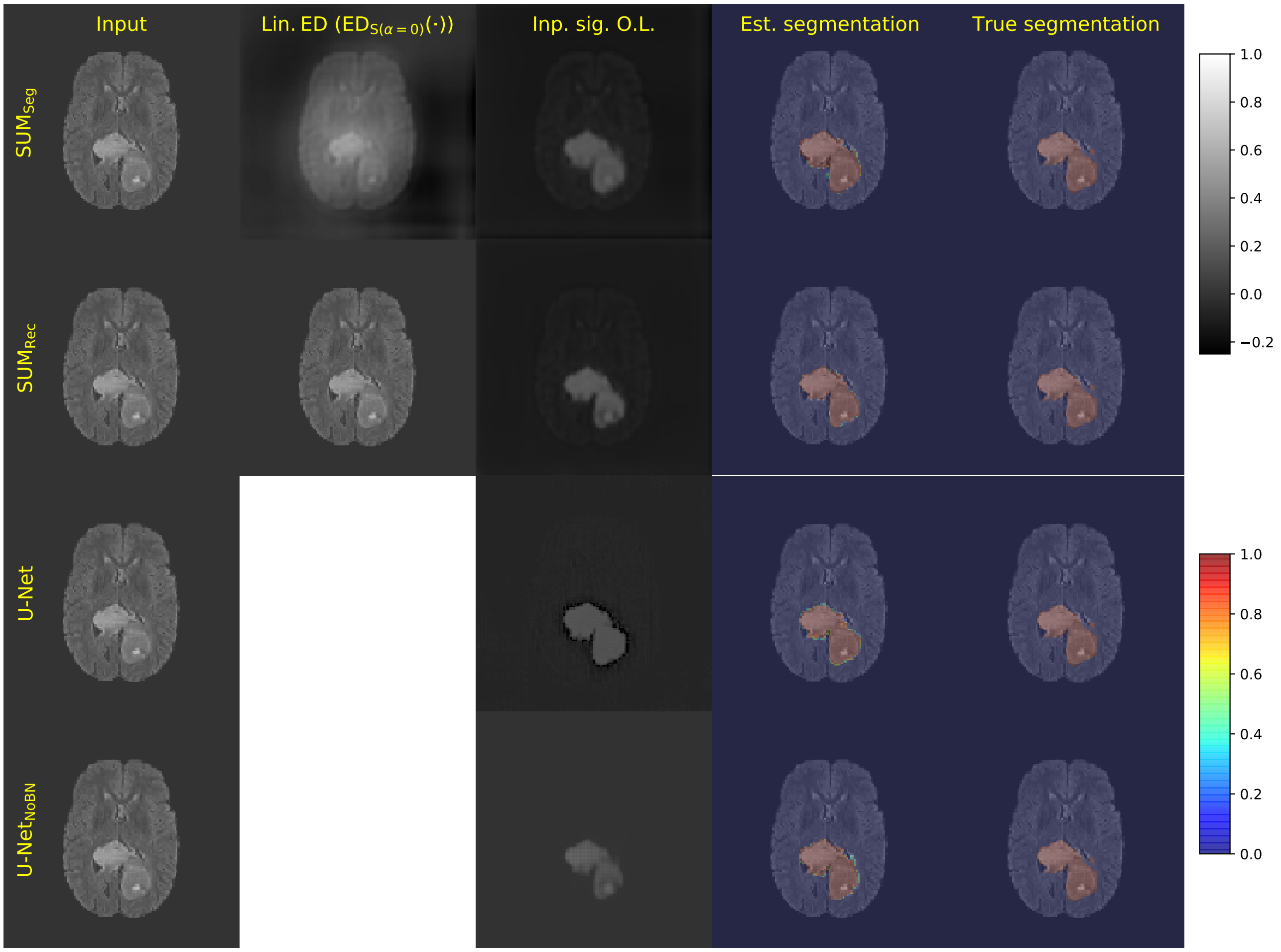

Figure 8. A slice of the test set processed with the SUM_{Seg}, SUM_{Est}, U-Net and $U\text{-}Net_{NoBN}$.

When observing the FDV and FDS features, it is possible to observe that the model trained with all the augmentations has the smallest frequency spread variation, this model is also the most robust to image blur. Conversely, it can be shown that the model with no augmentations is the least robust and has the highest values for the FDS feature. Complementary to Table 9, the figure in Appendix G shows the impulse responses of the models with the same augmentations for one training cycle. In the figure it is visible that the Fourier spectrums of the impulses that include rotations are smoother, while the Fourier spectrum of the model trained without any data augmentation is more irregular.

H. Effect of the weight decay in the impulse response of the SUM_{Seg} model

This appendix explores the effect of the *weight decay* in the signal processing behavior of segmentation models. To demonstrate the influence of this hyperparameter, the SUM_{Seg} model is trained with weight decay values of 0.5, 0.25, 0.125, 0.075 and 0. For this experiment, no data augmentations have been employed.

Table 4 summarizes the results of processing the test set that has been proceeded with non-corrupted slices, as well as with corrupted images with Gaussian blur ($\sigma_{\text{blur}} = 1$), or with additive Gaussian noise ($\sigma_{\text{noise}} = 10\%$ of the dynamic range of the image). In the referred table, it can be observed that space-domain spread (SDS) of the impulse response decreases the weight-decay value increases. However, lower spread in the space-domain does not seem to result in better performing models. However, just as in Appendix G, the best performing models have also the lowest frequency-domain spreads.

Table 3. Dice score and intersection over union (IoU) measured for segmentation estimates produced by the SEG model with different data augmentation techniques.

Not distorted input					
Augm.	No augment.	Rot. mirr.	Noise	Blur	Rot. noi. blur
Dice Sc.	0.709 ± 0.086	0.849 ± 0.010	0.833 ± 0.003	0.833 ± 0.003	0.852 ± 0.003
IoU	0.556 ± 0.103	0.738 ± 0.015	0.713 ± 0.005	0.714 ± 0.005	0.742 ± 0.005
Gaussian blur ($\sigma_{\text{blur}} = 1$)					
Augm.	No augment.	Rot. mirr.	Noise	Blur	Rot. noi. blur
Dice Sc.	0.695 ± 0.089	0.841 ± 0.007	0.828 ± 0.004	0.831 ± 0.003	0.849 ± 0.003
IoU	0.540 ± 0.106	0.726 ± 0.011	0.707 ± 0.005	0.711 ± 0.005	0.737 ± 0.004
Noise ($\sigma_{\text{Noise}} = 10\%$ of dynamic range of the images)					
Augm.	No augment.	Rot. mirr.	Noise	Blur	Rot. noi. blur
Dice Sc.	0.773 ± 0.048	0.780 ± 0.038	0.822 ± 0.003	0.802 ± 0.006	0.805 ± 0.019
IoU	0.632 ± 0.060	0.641 ± 0.050	0.698 ± 0.005	0.669 ± 0.008	0.674 ± 0.026
Impulse response characterization measurements					
Augm.	No augment.	Rot. mirr.	Noise	Blur	Rot. noi. blur
SDS	25.9 ± 0.88	26.8 ± 2.30	28.2 ± 2.47	26.8 ± 1.57	30.0 ± 2.75
FDV	53.9 ± 3.47	53.0 ± 2.12	55.2 ± 3.54	56.1 ± 0.88	50.9 ± 4.12
FDS	25.8 ± 0.52	23.6 ± 0.68	22.3 ± 0.71	23.8 ± 1.32	21.4 ± 1.37

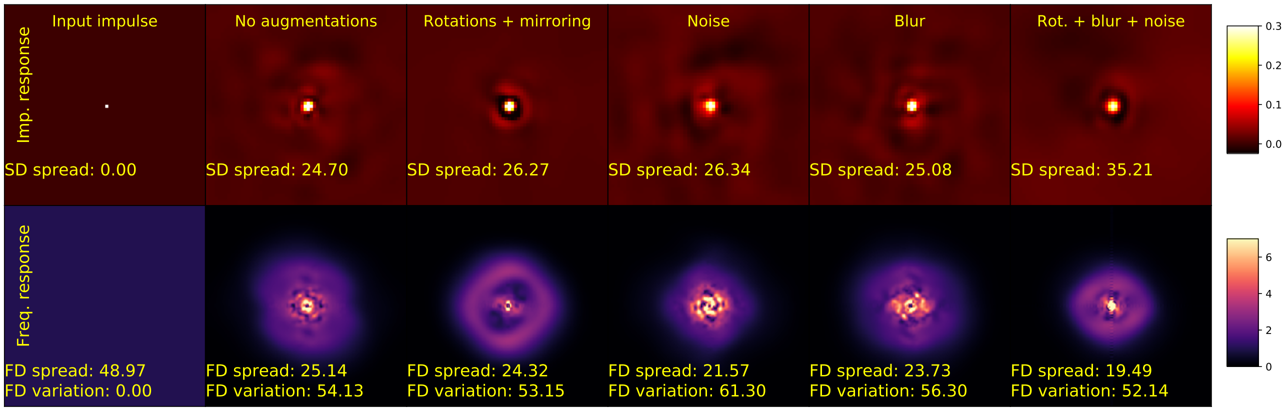

Figure 9. Impulse response for SUM models trained with different data augmentations. The top row depicts the impulse response, while the bottom row shows their corresponding Fourier spectrums.

I. Matched filters and segmentation models

Recent research has shown that the convolution kernels in shallow classification CNNs converge to filters that are correlated to the training samples (Pinson et al., 2023). Consequently, the convolution of such filters with the images being processed is analogous to the processes performed by matched filters used in signal detection and communication systems (Steven, 1998). In matched filtering, a known signal is to be detected. The signal may be corrupted by noise and/or distorted by the propagation medium. Since the signal is known, the optimal estimator is given by a template of the signal that is referred to as *matched filter*. Therefore, when convolving the input with the observed noisy/distorted observation, the result is the superposition of the auto correlation of the signal and a cross-correlation with the noise. If the noise is assumed to be uncorrelated to the signal, it can be noted that the amplitude of the resulting operation is small when the signal being detected is not present. In contrast, when the signal of interest is present, the correlation between the template and the signal is high, which results in more signal power.

Table 4. Dice score and intersection over union (IoU) measured for segmentation estimates produced by the SEG$_{\text{Seg}}$ with blurred and noisy images as a function of the weight decay values set for training the model. The performance values are obtained by averaging the performance after 5 training cycles.

Not distorted input					
W.D. val.	0.5	0.25	0.125	0.075	0
Dice Sc.	0.828 ± 0.002	0.835 ± 0.002	0.840 ± 0.006	0.839 ± 0.003	0.843 ± 0.007
IoU	0.707 ± 0.004	0.717 ± 0.004	0.725 ± 0.009	0.723 ± 0.005	0.729 ± 0.010
Gaussian blur ($\sigma_{\text{blur}} = 1$)					
W.D. val	0.5	0.25	0.125	0.075	0
Dice Sc.	0.825 ± 0.003	0.830 ± 0.002	0.830 ± 0.005	0.826 ± 0.004	0.828 ± 0.006
IoU	0.702 ± 0.004	0.709 ± 0.003	0.710 ± 0.007	0.704 ± 0.006	0.706 ± 0.009
Noise ($\sigma_{\text{Noise}} = 10\%$ of dynamic range of the images)					
W.D. val.	0.5	0.25	0.125	0.075	0
Dice Sc.	0.783 ± 0.006	0.804 ± 0.002	0.796 ± 0.017	0.815 ± 0.005	0.808 ± 0.008
IoU	0.643 ± 0.008	0.672 ± 0.003	0.662 ± 0.023	0.688 ± 0.007	0.678 ± 0.012
Space/frequency-domain and frequency-domain variation					
W.D. val.	0.5	0.25	0.125	0.075	0
SDS	23.4 ± 1.86	25.8 ± 1.95	27.5 ± 0.98	28.1 ± 0.57	30.0 ± 1.36
FDV	44.8 ± 3.66	57.5 ± 5.06	70.1 ± 3.46	71.3 ± 7.39	88.0 ± 4.10
FDS	24.5 ± 0.93	25.7 ± 1.24	26.0 ± 1.11	24.9 ± 2.11	24.6 ± 1.70

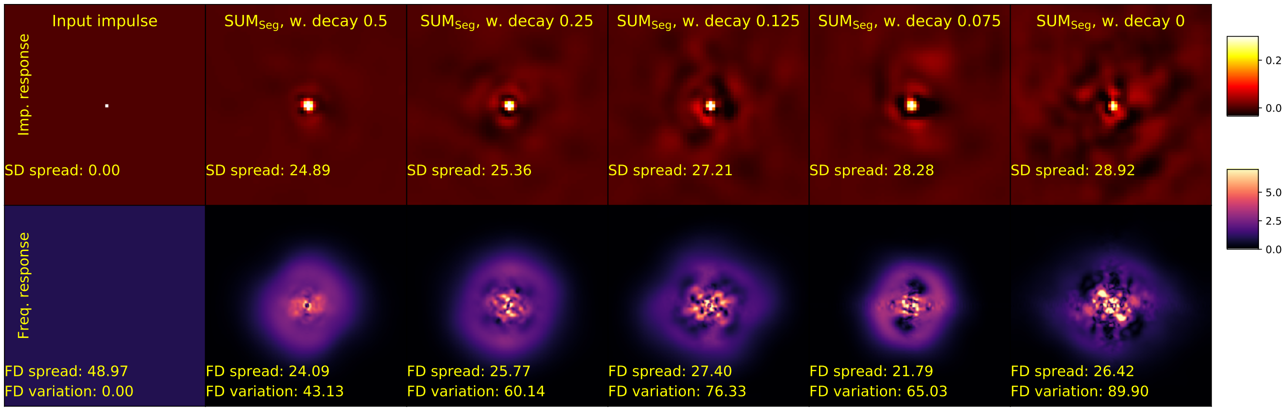

Figure 10. Effect of the weight decay on the signal processing behavior of the SUM$_{\text{Seg}}$ model. The first row shows the impulse response of each of the trained models, while the second row depicts the corresponding frequency responses. Note that the filtering caused by models trained with higher weight-decay values is more spatially localized and has a more uniform frequency spectrum.

As an example of the above discussion, Fig. 11 shows the high-level operation of a matched filter. Specifically, assume that it is desired to detect the sinusoidal wave $\underline{s}$ shown in the first frame (top left). In practice, this signal propagates trough a medium, which may introduce noise $\underline{\eta}$ (top center and top right). As discussed earlier, matched filtering considers that the optimal detection mechanism to detect the signal is to convolve it with a highly-correlated signal (e.g. another sinusoid). Since the signal may present phase shifts, then a quadrature component may be employed. The correlation between the observed noisy input $\underline{x}$ and a cosine and sine templates with the same frequency is shown the bottom left frame. The final process of matched filtering is to perform a threshold operation. Note that the resulting detected signal is similar to the ground-truth segmentation which shows the area in which the original sinusoid is active (bottom right).

In order to contrast the analogy of matched filtering with segmentation CNNs. We propose to train a simple CNN

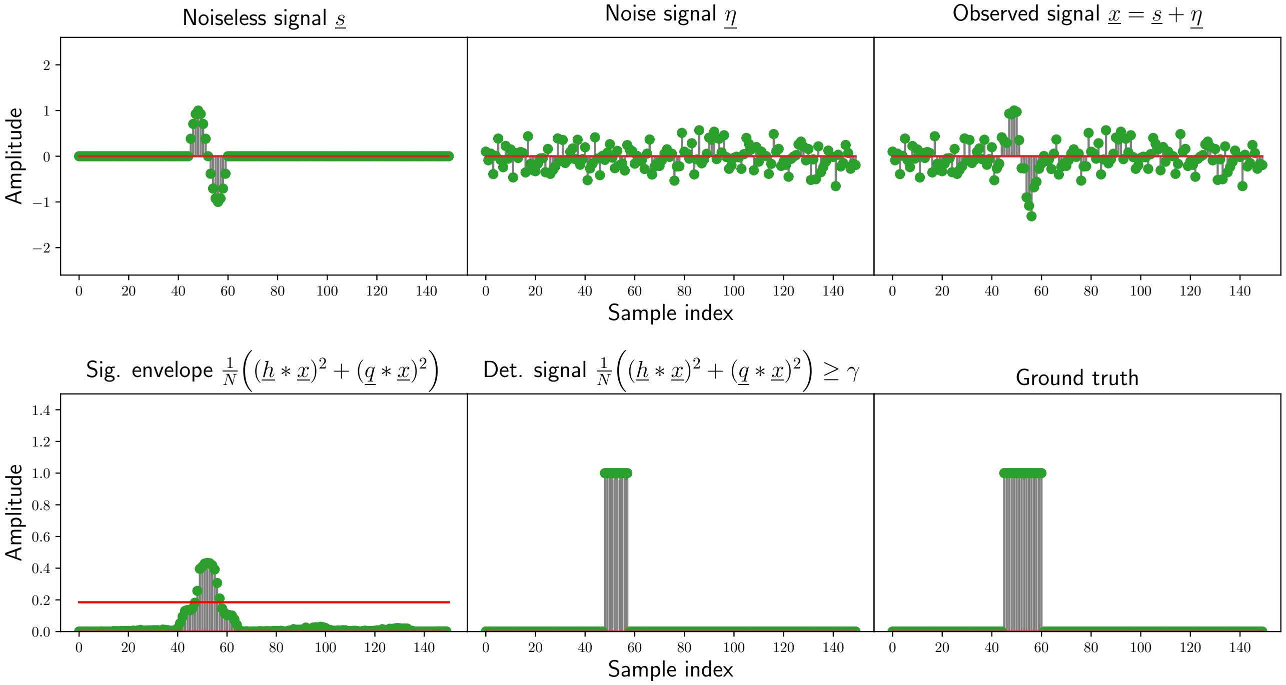

Figure 11. Example of matched filter employed to detect a sinusoidal signal with known frequency and unknown phase. The top left frame shows a one-cycle sinusoidal wave $\underline{s}$ that is to be detected. The top center frame shows noise $\underline{\eta}$ that contaminates the signal of interest $\underline{s}$. The resulting signal of the previous operation is the noise-contaminated signal $\underline{x}$ shown in the top right frame. The top left frame shows the convolution, squaring and sum of the signal $\underline{x}$ with two quadrature filters $\underline{h}$ and $\underline{q}$, where $\underline{h}$ is a cosine function and $\underline{q}$ is a sine with the same frequency. It should be noted that the result of this operation is a signal that is the squared of the envelope signal. Finally, the top center frame shows the result of applying the threshold operation to the signal in the bottom left frame (the threshold level is represented as an horizontal red line). Note that the detection signal is reminiscent of the ground truth signal/segmentation.

encoder-decoder with synthetic data to perform signal segmentation/detection. The proposed CNN is defined by

$$\text{CNN}(\underline{x}) = \mathcal{C}_{(0,1)}\Big(\tilde{\mathbf{C}}_0 \mathcal{S}_{(\underline{b}_0)}\big(\mathbf{C}_0 \underline{x}\big)\Big). \tag{30}$$

Here, $\tilde{\mathbf{C}}_0$ and $\mathbf{C}_0$ are tensors with dimensions $(2{\times}1{\times}18)$ and $(1{\times}2{\times}18)$, respectively. Moreover the bias vector $\underline{b}$ has dimensions $(1{\times}2)$. Finally, function $\mathcal{C}_{(0,1)}(\cdot)$ is a clipping operation between zero and unity.

The CNN described in Eq. (30) has been trained to segment/label every pixel in randomly placed pieces of sinusoidal signals that are contaminated with noise, the resulting trained convolution kernels are shown in Fig. 12, where it can be seen that the filters of the encoder $\mathbf{C}_0$ learn signals that are reminiscent of sinusoidal waveforms. Moreover, the filters of the decoder $\tilde{\mathbf{C}}_0$ resemble some sort of Gaussian filter (in blue) and a derivative of Gaussian (in green). Fig 13 shows the operation of the trained model that is evaluated in the signal $\underline{x}$ from Fig. 11. In the figure it can be observed that the convolution and activation of the encoder layer results in a sparse signal, were most of the energy is focused in the area where the signal of interest is active. Furthermore, where the decoder is applied to the encoded signal, it is visible that the resulting signal is smoother and more similar to the envelope signal shown in Fig. 11. However, in this case, all the values that are not considered to be part of the signal are pushed toward the negative region and are clipped by the output layer to produce the segmentation estimate $\underline{m}$.

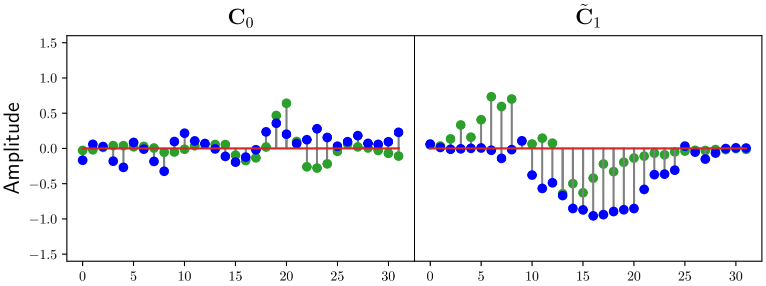

Figure 12. Filters learned by the simple CNN described in Eq. (30). The left image shows the filters learned by the encoder convolution kernel $\mathbf{C}_0$, whereas the right image are the corresponding filters of the decoder $\tilde{\mathbf{C}}_0$. Note that the encoder filters learn a sinusoidal function, while the decoder functions resemble a scaling and wavelet functions.

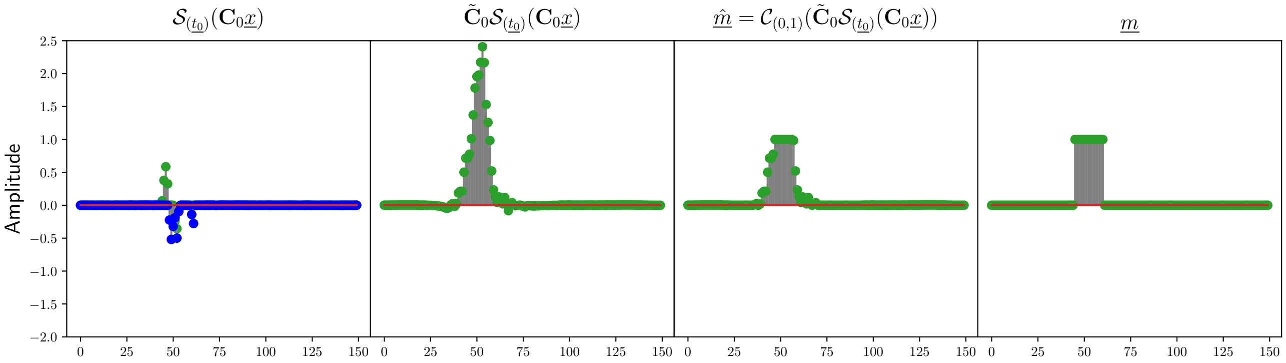

Figure 13. Operation of the CNN described in Eq. (30) when evaluated in the noisy observation $\underline{x}$ shown in Fig. 11. From left to right, the first frame shows the application of the activation signal over the convolution of signal $\underline{x}$ with the encoder filters. The second frame is the convolution of the encoded signal with the decoder filter. Note that the small elements in the encoded signal are pushed towards negative values. The third frame shows the result of applying the output layer to the decoded signal. The resulting signal $\underline{\hat{m}}$ is an estimate of the true segmentation $\underline{m}$ that is shown in the fourth and last frame