Improving Morphological Networks for Learning Image-to-Image Transforms

Antoine Bottenmuller, Guillaume Tochon, Romain Hermary, Élodie Puybareau, Jesus Angulo

Improving Morphological Networks for Learning Image-to-Image Transforms

Antoine Bottenmuller[1], Guillaume Tochon[2], Romain Hermary[3], Élodie Puybareau[2], and Gustavo (Jesús) Angulo[4]

[1] Mines Paris, PSL University, Center for Mathematical Morphology (CMM), Fontainebleau, France
antoine.bottenmuller@minesparis.psl.eu
[2] EPITA Research Laboratory (LRE), EPITA, Le Kremlin-Bicêtre, France
{guillaume.tochon,elodie.puybareau}@epita.fr
[3] Computer Vision, Imaging & Machine Intelligence Research Group (CVI^2), University of Luxembourg, Luxembourg
romain.hermary@uni.lu
[4] Mines Paris, PSL University, Center for Applied Mathematics (CMA), Sophia Antipolis, France
jesus.angulo@minesparis.psl.eu

Abstract. Replacing convolution with morphological operations in trainable layers has received significant attention lately. Among the various strategies that have emerged, smooth morphological layers have shown strong potential and flexibility, as a single layer can behave either like a (pseudo-)erosion or a (pseudo-)dilation depending on the sign and value of its trainable control parameter. In this work, we build upon the so-called $\mathcal{S}$Morph layer by introducing a harmonized formulation that addresses previously identified asymptotic limitations when learning grayscale erosion and dilation. We also investigate and compare two strategies (a novel penalty term in the training loss and shared-weight layers) to improve the learning of grayscale opening and closing operations in two-layer networks. Finally, we evaluate the performance of this improved $\mathcal{S}$Morph layer on a salt-and-pepper denoising task in a four-layer network architecture, and compare it with other morphological and convolutional networks.

Keywords: mathematical morphology · morphological network · smooth morphological layer · grayscale transform · salt-and-pepper denoising.

1 Introduction

Over the past decade, convolutional neural networks (CNNs) have revolutionized numerous image processing tasks such as object recognition and semantic segmentation. By learning representative multiscale filter banks through trainable convolutional layers, CNNs capture both low-level and high-level features, enabling them to effectively capture complex visual patterns. As universal approximators (provided that the depth of the network is large enough [24]), they offer a powerful framework for approximating a broad class of image operators in a

data-driven way. Yet, convolutional layers also suffer from well-known limitations. As a matter of fact, they remain limited in terms of interpretability and they often struggle with small or low-contrast objects. In contrast, morphological operations are particularly well-suited to process such challenging features and offer geometrically meaningful and interpretable operations. This motivates the replacement of standard convolution layers with trainable morphological operations, leading to the development of morphological neural networks (MNNs). These networks thus enable the automatic learning of morphological operations sequences along with their optimal structuring elements. Beside, the Matheron-Maragos-Bannon-Barrera (MMBB) representation theory of non-linear operators [2, 14, 12] establishes theoretical guarantees regarding the decomposition of any (translation invariant) image operator as a composition of morphological operations. While this property plays a similar role to the universal approximation theory for neural networks, no general construction algorithm exists to derive in practice such decomposition for a given image operator. Thus, MNNs also offer an appealing data-driven approach to uncover such representations.

However, erosion and dilation operations (as well as their compositions) rely on pointwise computations of minimum and maximum over local neighborhoods, which are not differentiable everywhere, unlike standard convolution. A first workaround to this issue explored the use of smooth approximations of these operations. Leveraging the counter-harmonic mean [1], the so-called P-convolution ($PConv$) layer was proposed in [13] and was shown to effectively learn target morphological operations such as erosion, dilation, opening, closing and top-hat. Other smooth approximations of min and max were also explored, including the LogSumExp in [19, 18] or the α-softmax which led to the definition of the $\mathcal{S}$Morph layer in [8, 6]. This latter was notably shown to outperform the $PConv$ layer in learning grayscale morphological operation. A key advantage of these smooth formulations is that their asymptotic behavior is driven by the value of a trainable control parameter. Each layer can act either as an erosion when the parameter tends to $-\infty$, or as a dilation when it tends to $+\infty$. As almost-everywhere differentiable layers such as MaxPooling or ReLU became standard components in neural network architectures, exact morphological layers were proposed as an alternative to smooth approximations by adapting the backpropagation step [5, 15, 20, 16]. This enabled the design of morphological networks performing exact morphological operations, though at the cost of reduced flexibility in learning the optimal sequence of erosions and dilations from data. However, morphological networks, whether based on exact or smooth layers, are still challenging to train and prone to convergence issues [6], which restricts their actual practical applications. Recent studies have therefore investigated how these difficulties relate to the differentiability of the layers and to their initialization strategies [3, 4].

In this work, we extend the $\mathcal{S}$Morph layer proposed in [8, 6]. Specifically, we introduce a reformulated version that resolves the asymptotic shortcomings of $\mathcal{S}$Morph, and we propose constraint-based strategies that significantly improve convergence when learning opening and closing operations in two-layer networks. Our contributions are organized as follows. In Section 2, we recall the definition

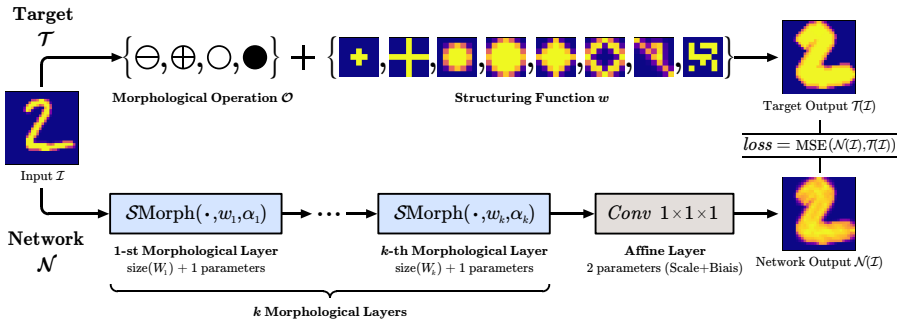


Fig. 1. Architecture of a k-layer morphological network, adapted from [6]. Upper path describes the *target morphological transform* $\mathcal{T} = \mathcal{O}_w$; lower path the *network* $\mathcal{N}$ itself.

of the $\mathcal{S}$Morph layer and introduce a novel harmonized formulation, referred to as $\mathcal{HS}$Morph. In Section 3, we examine the behavior of $\mathcal{HS}$Morph layers in two-layer networks for learning opening and closing operations. We also propose two architectural improvements, namely a novel penalty term in the training loss and shared-weight layers, to improve network convergence. In Section 4, we evaluate the performance of $\mathcal{HS}$Morph in four-layer networks for salt-and-pepper denoising tasks and compare it with other morphological layers. Finally, Section 5 concludes the paper and outlines future research directions.

2 Smooth Morphological Layers

Smooth layers are designed as fully differentiable approximations of true morphological operations. These layers rely on smooth approximations of the min and max functions, such as the counter-harmonic mean, LogSumExp, or α-softmax, with each formulation determining the asymptotic behavior of the layer. The behavior of each approximation, and thus the layer itself, depends on a control parameter. Its sign determines which function is approximated (min when negative, with the layer acting like a (pseudo-)erosion, and max when positive, with the layer acting like a (pseudo-)dilation), while its magnitude governs the accuracy of the approximation (the larger in absolute value, the closer the approximation is to the true operation). Smooth layers can then be integrated into neural network architectures in the same way as traditional convolutional layers. Figure 1 illustrates such a morphological network composed of k $\mathcal{S}$Morph layers, used to learn a target morphological operation and its associated structuring element.

2.1 The $\mathcal{S}$Morph Layer

The $\mathcal{S}$Morph layer is based on the $\alpha-$softmax function [9], denoted by $\mathcal{S}_\alpha$ and defined, for any $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, as

$$\mathcal{S}_\alpha(\mathbf{x}) = \frac{\sum_{i=1}^n x_i e^{\alpha x_i}}{\sum_{i=1}^n e^{\alpha x_i}}. \tag{1}$$

The control parameter α governs the behavior of $\mathcal{S}_\alpha(\mathbf{x})$: when $\alpha = 0$, $\mathcal{S}_\alpha(\mathbf{x})$ corresponds to the arithmetic mean of $\mathbf{x}$, while $\lim_{\alpha\to-\infty} \mathcal{S}_\alpha(\mathbf{x}) = \min_i x_i$ and $\lim_{\alpha\to+\infty} \mathcal{S}_\alpha(\mathbf{x}) = \max_i x_i$. Thus, $\mathcal{S}_\alpha$ provides an asymptotic approximation of the min and max functions depending on the value of α.

Let $f\colon V \to \mathbb{R}$ be an image, where $x \in V$ denotes the pixel coordinates on a 2D (or N-D) grid $V \subseteq \mathbb{Z}^2$, and $f(x) \in \mathbb{R}$ the corresponding pixel value, and let $w\colon W \to \mathbb{R}$ be a structuring function (or kernel), with spatial support $W \subseteq \mathbb{Z}^2$. The $\mathcal{S}$Morph operation of image f with kernel w and parameter $\alpha \in \mathbb{R}$ is defined as

$$\mathcal{S}\mathrm{Morph}(f, w, \alpha)(x) = \frac{\sum_{y\in W} \big(f(x+y) + w(y)\big) e^{\alpha(f(x+y)+w(y))}}{\sum_{y\in W} e^{\alpha(f(x+y)+w(y))}}. \tag{2}$$

This formulation directly follows from the definition of the α-softmax function, by substituting the entries x_i in Equation (1) with $f(x+y)+w(y)$. Thus, the asymptotic behavior of $\mathcal{S}\mathrm{Morph}(f, w, \alpha)$ mirrors that of the α-softmax:

$$\lim_{\alpha\to-\infty} \mathcal{S}\mathrm{Morph}(f, w, \alpha)(x) = \min_{y\in W} \big\{f(x+y) + w(y)\big\} = (f \ominus -w)(x), \tag{3}$$

$$\lim_{\alpha\to+\infty} \mathcal{S}\mathrm{Morph}(f, w, \alpha)(x) = \max_{y\in W} \big\{f(x+y) + w(y)\big\} = (f \oplus \breve{w})(x), \tag{4}$$

with $\breve{w}$ the symmetrized version of the kernel w with respect to the origin 0_n.

Consequently, when $\alpha \to -\infty$, $\mathcal{S}\mathrm{Morph}(f, w, \alpha)$ converges to the erosion of the image f with $-w$. Conversely, when $\alpha \to +\infty$, $\mathcal{S}\mathrm{Morph}(f, w, \alpha)$ converges to the dilation of f with $\breve{w}$. As observed in [8, 6], the $\mathcal{S}$Morph layer provides a good approximation of an erosion or a dilation when $|\alpha| > 5$. The performed operation is otherwise only considered to be a pseudo-erosion or pseudo-dilation.

2.2 Proposed $\mathcal{HS}$Morph Layer

The definition of $\mathcal{S}$Morph introduced in [8, 6] and recalled in Equation (2) has two shortcomings regarding its asymptotic behavior: given any kernel w, the layer either approximates the erosion of image f with negated kernel $-w$ when $\alpha < 0$, *i.e.*, $\mathcal{S}\mathrm{Morph}(f, w, \alpha) \simeq f \ominus -w$; or approximates the dilation of f with symmetrized kernel $\breve{w}$ when $\alpha > 0$, *i.e.*, $\mathcal{S}\mathrm{Morph}(f, w, \alpha) \simeq f \oplus \breve{w}$; but never with w itself. While the former behavior was explicitly reported in [8, 6], the latter was not discussed, as all filters considered in their experiments were symmetric. We propose here a revised definition of the $\mathcal{S}$Morph layer that ensures consistency, so that it approximates the erosion and dilation of f with the same kernel w. To this end, we replace the filter w in Equation (2) by the new filter $\tilde{w}_\alpha$ defined as

$$\tilde{w}_\alpha = \tanh(\alpha) \left(\frac{1 + \tanh(\alpha)}{2} \breve{w} + \frac{1 - \tanh(\alpha)}{2} w \right), \tag{5}$$

and define the $\mathcal{HS}$Morph (for Harmonized $\mathcal{S}$Morph) layer as

$$\mathcal{HS}\mathrm{Morph}(f, w, \alpha) = \mathcal{S}\mathrm{Morph}(f, \tilde{w}_\alpha, \alpha). \tag{6}$$

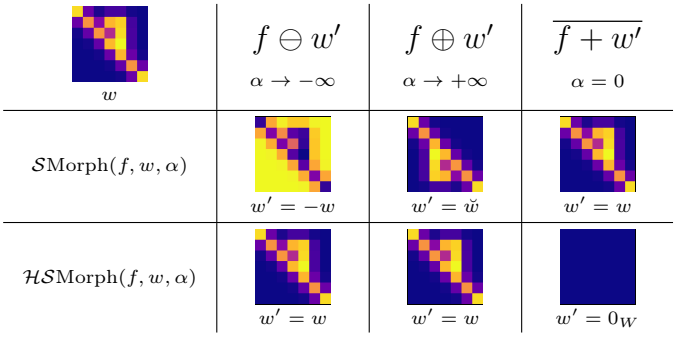


Fig. 2. Asymptotic behavior of $\mathcal{S}$Morph and $\mathcal{HS}$Morph layers. For a given kernel w (top left corner), the $\mathcal{S}/\mathcal{HS}\text{Morph}(\cdot, w, \alpha)$ layer behaves as a true erosion ($\alpha \to -\infty$) or dilation ($\alpha \to +\infty$) operation of input f with the associated structuring function w'. For $\alpha = 0$, the layer behaves as a local arithmetic mean operation of $f + w'$.

The asymptotic behavior of $\tilde{w}_\alpha$ follows directly from that of the hyperbolic tangent: $\lim_{\alpha \to -\infty} \tilde{w}_\alpha = -w$ and $\lim_{\alpha \to +\infty} \tilde{w}_\alpha = \check{w}$. Equations (3) and (4) then lead to

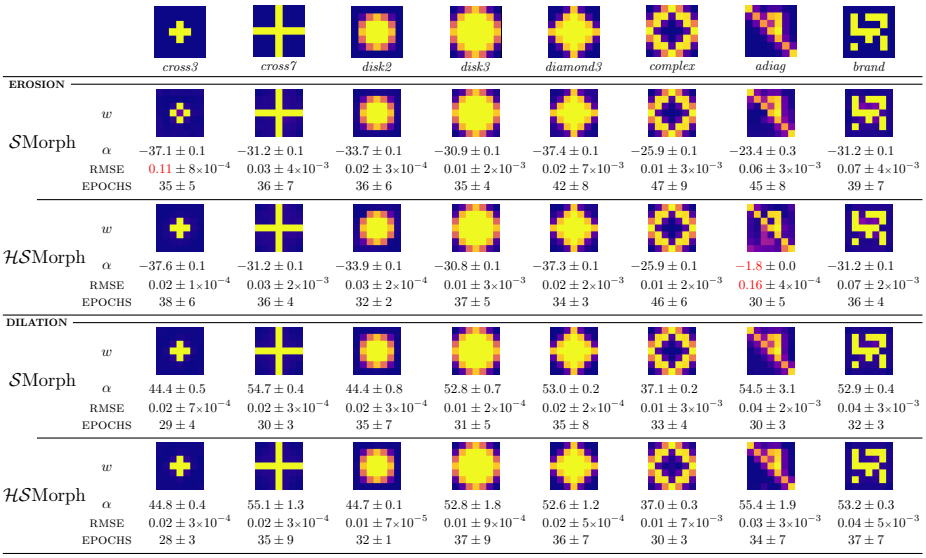
$$\lim_{\alpha \to -\infty} \mathcal{HS}\text{Morph}(f, w, \alpha) = \lim_{\alpha \to -\infty} \mathcal{S}\text{Morph}(f, \tilde{w}_\alpha, \alpha) = f \ominus w\,, \tag{7}$$

$$\lim_{\alpha \to +\infty} \mathcal{HS}\text{Morph}(f, w, \alpha) = \lim_{\alpha \to +\infty} \mathcal{S}\text{Morph}(f, \tilde{w}_\alpha, \alpha) = f \oplus w\,. \tag{8}$$

Figure 2 illustrates the asymptotic behavior of $\mathcal{S}$Morph and $\mathcal{HS}$Morph layers. Since $\tanh(5) = -\tanh(-5) \approx 0.9999$, the convergence regime of $\mathcal{HS}$Morph is essentially the same as that of $\mathcal{S}$Morph: the performed operation can be regarded as an erosion or dilation when $|\alpha| > 5$, and as a pseudo-operation otherwise.

2.3 Comparison of $\mathcal{S}$Morph and $\mathcal{HS}$Morph Layers

To assess whether the definition of the filter $\tilde{w}_\alpha$ in Equation (5) affects the ability of the $\mathcal{HS}$Morph layer to accurately learn a target morphological operation and its corresponding structuring function, we conducted a comparative experiment against the standard $\mathcal{S}$Morph layer. More specifically, we independently trained one-layer $\mathcal{S}$Morph and $\mathcal{HS}$Morph networks, following the architecture in Figure 1, to separately learn erosion and dilation operations. We used eight target structuring functions, as presented in Figure 3. In addition to the six symmetric functions used in [8, 6], we introduced two new asymmetric ones, named *adiag* and *brand*. Both erosion and dilation operations were applied with every structuring function to all 60k images contained in the MNIST dataset [10]. The original images were used as input to the network, and training was performed by minimizing the mean squared error (MSE) between network outputs and target images resulting from true morphological operations. We used the same training procedure as described in [8, 6]: a batch size of 32, and the Adam optimizer with an initial learning rate of 0.01, which was reduced by a factor of 10 when the loss plateaued



		cross3	*cross7*	*disk2*	*disk3*	*diamond3*	*complex*	*adiag*	*brand*
EROSION									
$\mathcal{S}$Morph	w								
	α	-37.1 ± 0.1	-31.2 ± 0.1	-33.7 ± 0.1	-30.9 ± 0.1	-37.4 ± 0.1	-25.9 ± 0.1	-23.4 ± 0.3	-31.2 ± 0.1
	RMSE	$0.11 \pm 8\times10^{-4}$	$0.03 \pm 4\times10^{-3}$	$0.02 \pm 3\times10^{-4}$	$0.01 \pm 2\times10^{-3}$	$0.02 \pm 7\times10^{-3}$	$0.01 \pm 3\times10^{-3}$	$0.06 \pm 3\times10^{-3}$	$0.07 \pm 4\times10^{-3}$
	EPOCHS	35 ± 5	36 ± 7	36 ± 6	35 ± 4	42 ± 8	47 ± 9	45 ± 8	39 ± 7
$\mathcal{HS}$Morph	w								
	α	-37.6 ± 0.1	-31.2 ± 0.1	-33.9 ± 0.1	-30.8 ± 0.1	-37.3 ± 0.1	-25.9 ± 0.1	-1.8 ± 0.0	-31.2 ± 0.1
	RMSE	$0.02 \pm 1\times10^{-4}$	$0.03 \pm 2\times10^{-3}$	$0.03 \pm 2\times10^{-4}$	$0.01 \pm 3\times10^{-3}$	$0.02 \pm 2\times10^{-3}$	$0.01 \pm 2\times10^{-3}$	$0.16 \pm 4\times10^{-4}$	$0.07 \pm 2\times10^{-3}$
	EPOCHS	38 ± 6	36 ± 4	32 ± 2	37 ± 5	34 ± 3	46 ± 6	30 ± 5	36 ± 4
DILATION									
$\mathcal{S}$Morph	w								
	α	44.4 ± 0.5	54.7 ± 0.4	44.4 ± 0.8	52.8 ± 0.7	53.0 ± 0.2	37.1 ± 0.2	54.5 ± 3.1	52.9 ± 0.4
	RMSE	$0.02 \pm 7\times10^{-4}$	$0.02 \pm 3\times10^{-4}$	$0.02 \pm 3\times10^{-4}$	$0.01 \pm 2\times10^{-4}$	$0.02 \pm 2\times10^{-4}$	$0.01 \pm 3\times10^{-3}$	$0.04 \pm 2\times10^{-3}$	$0.04 \pm 3\times10^{-3}$
	EPOCHS	29 ± 4	30 ± 3	35 ± 7	31 ± 5	35 ± 8	33 ± 4	30 ± 3	32 ± 3
$\mathcal{HS}$Morph	w								
	α	44.8 ± 0.4	55.1 ± 1.3	44.7 ± 0.1	52.8 ± 1.8	52.6 ± 1.2	37.0 ± 0.3	55.4 ± 1.9	53.2 ± 0.3
	RMSE	$0.02 \pm 3\times10^{-4}$	$0.02 \pm 3\times10^{-4}$	$0.01 \pm 7\times10^{-5}$	$0.01 \pm 9\times10^{-4}$	$0.02 \pm 5\times10^{-4}$	$0.01 \pm 7\times10^{-3}$	$0.03 \pm 3\times10^{-3}$	$0.04 \pm 5\times10^{-3}$
	EPOCHS	28 ± 3	35 ± 9	32 ± 1	37 ± 9	36 ± 7	30 ± 3	34 ± 7	37 ± 7

Fig. 3. Learning results for the *erosion* (first two rows) and *dilation* (last two rows) operations with one-layer $\mathcal{S}$Morph and $\mathcal{HS}$Morph networks on the MNIST dataset: learned filter w, corresponding control parameter α, RMSE between w and the target structuring element, and number of training epochs, for eight target elements (columns). Each metric comes with the *mean* $\pm$ *standard deviation* over five runs per experiment.

for five consecutive epochs. Training was stopped when the loss plateaued for ten consecutive epochs. Each filter was initialized using a centered normal distribution with standard deviation $\sigma = 0.01$, and $\alpha = 0$. Training was done simultaneously on w and α, and five independent runs were conducted per experiment.

Figure 3 presents the obtained results. For $\mathcal{S}$Morph, both the displayed filter and the reported root mean square error (RMSE) are obtained after applying the appropriate asymptotic correction (negation for erosion, symmetrization for dilation) to the original learned filter. The results obtained with both $\mathcal{S}$Morph and $\mathcal{HS}$Morph layers are generally similar in terms of filter accuracy, convergence speed and final loss (typically around 10^{-6}), with $\mathcal{HS}$Morph even outperforming $\mathcal{S}$Morph in the case of the *cross3* erosion task. The only exception is the *adiag* erosion scenario, where $\mathcal{HS}$Morph converged to a pseudo-erosion ($\alpha = -1.8$) with a slightly inaccurate filter. While the actual reason for this counter-performance remains unclear, we believe it is not due to the asymmetry of the target filter, as the new $\mathcal{HS}$Morph formulation successfully learned other asymmetric scenarios (dilation with *adiag* and both erosion and dilation with *brand*). In any case, the $\mathcal{HS}$Morph layer was not intended to improve $\mathcal{S}$Morph performance on asymmetric filters but rather to correct its asymptotic behavior. Overall, the obtained results confirm that replacing w with $\tilde{w}_\alpha$ in Equation (6) preserves the learning performance of $\mathcal{S}$Morph, while addressing its asymptotic limitations. Therefore, we exclusively use $\mathcal{HS}$Morph in place of $\mathcal{S}$Morph for the remainder of this article.

3 Improvement Strategies for Two-Layer Networks

While one-layer $\mathcal{S}$Morph networks can successfully learn erosion and dilation operations along with their associated structuring functions, two-layer networks encounter greater difficulty in learning their dual compositions: opening and closing [8, 6]. In this section, we introduce two strategies designed to enhance the learning capacity of two-layer networks on such operations.

3.1 Shared-Weight Dual Layer

When learning opening and closing operations, two-layer networks are expected to converge as follows: for openings, the first layer should converge to an erosion ($\alpha_1 \ll 0$) and the second layer to a dilation ($\alpha_2 \gg 0$); for closings, the roles are reversed. Both layers should specifically learn the same structuring function, in line with the definition of opening and closing. However, this is rarely the case, as observed in [8, 6] and illustrated in the top rows of the opening and closing sections of Figure 4, especially when the network fails to converge. Thus, enforcing strict weight sharing across the two layers is an appealing idea to improve learning results. Moreover, this allows the number of learnable parameters to be reduced.

Therefore, we introduce a dual-layer variant, referred to as $\mathcal{HS}$Morph*Dual*, designed to improve the learning of opening and closing operations in two-layer networks. For any input image $f\colon V \to \mathbb{R}$, this layer is defined as

$$\mathcal{HS}\text{Morph}\mathit{Dual}(f, w, \alpha_1, \alpha_2) = \mathcal{HS}\text{Morph}(\mathcal{HS}\text{Morph}(f, w, \alpha_1), w, \alpha_2), \quad (9)$$

where $w\colon W \to \mathbb{R}$ is the shared filter, and $\alpha_1, \alpha_2 \in \mathbb{R}$ the two control parameters. This dual layer consists of two successive $\mathcal{HS}$Morph layers with control parameters α_1 and α_2 and one shared filter w, enforcing kernel consistency across the two stages, in line with true opening and closing operations. The independence of α_1 and α_2 allows the layer to approximate a (pseudo-)opening ($\alpha_1 < 0 < \alpha_2$) or (pseudo-)closing ($\alpha_2 < 0 < \alpha_1$) operation, provided that $\text{sgn}(\alpha_1) \neq \text{sgn}(\alpha_2)$.

Two-layer networks that use this strategy contain a single $\mathcal{HS}$Morph*Dual* layer, as it effectively combines the roles of two conventional layers. Architectures composed of such dual layers are denoted $\mathcal{HS}$Morph–S, where "S" indicates "weight sharing" across two consecutive standard layers as described in Equation (9).

3.2 Constraint on Control Parameters

As originally observed in [8, 6], and further illustrated in the top rows of the opening and closing sections in Figure 4, a consistent pattern in failed training scenarios for learning opening and closing operations is that the control parameters α_1 and α_2 tend to remain close to zero. This results in pseudo-operations, often with an incorrect sign on one of the two control parameters.

To address this issue, we introduce a novel regularization term C (for "constraint") to encourage the control parameters α_1 and α_2 of two consecutive

$\mathcal{HS}$Morph layers to move away from zero and from each other during training. This penalty term is defined for any $\alpha_1, \alpha_2 \in \mathbb{R}$ (such that $\alpha_1\alpha_2 \neq 1$) as

$$C(\alpha_1, \alpha_2) = \frac{1}{1 - \alpha_1\alpha_2}. \tag{10}$$

The loss function used to train the network $\mathcal{N}$ with respect to a target morphological transform $\mathcal{T}$ (see Figure 1) is then given for any input $f : V \to \mathbb{R}$ by

$$loss = \text{MSE}(\mathcal{N}(f), \mathcal{T}(f)) + \lambda \cdot C(\alpha_1, \alpha_2), \tag{11}$$

where $\lambda > 0$ is the penalty coefficient, typically set to 0.01 in our experiments.

The function C remains positive as long as $\alpha_1\alpha_2 < 1$, which is always the case in practice since both parameters are initialized at zero. Minimizing $C(\alpha_1, \alpha_2)$ (*i.e.*, $\to 0^+$) is therefore equivalent to minimizing the product $\alpha_1\alpha_2$ toward $-\infty$. This penalty term has appealing gradient properties. Its partial derivatives are

$$\frac{\partial C}{\partial \alpha_1}(\alpha_1, \alpha_2) = \frac{\alpha_2}{(1 - \alpha_1\alpha_2)^2} \qquad \text{and} \qquad \frac{\partial C}{\partial \alpha_2}(\alpha_1, \alpha_2) = \frac{\alpha_1}{(1 - \alpha_1\alpha_2)^2}\,. \tag{12}$$

Notably, the partial derivative of C with respect to α_1 depends positively on α_2, and vice versa, with their ratio being exactly α_2/α_1. Thus, the gradient signs are controlled by the opposite parameter, naturally pushing α_1 and α_2 toward opposite signs during training, hence favoring opening or closing. Moreover, the parameter with the smaller magnitude receives a stronger update, encouraging a faster deviation from zero and preventing stagnation. Networks trained with this regularization term, as shown in Equation (11), are denoted $\mathcal{HS}$Morph–C.

3.3 Comparison of the Proposed Strategies for two-layer Networks

To evaluate the effectiveness of the proposed strategies for improving the learning of opening and closing operations in two-layer networks, we conducted a comparative experiment following the same procedure as in Section 2.3. More specifically, we compared the performances of the following network configurations:

$\mathcal{HS}$Morph: a standard two-layer $\mathcal{HS}$Morph network;
$\mathcal{HS}$Morph–C: the same two-layer $\mathcal{HS}$Morph architecture with the penalty term from Equation (10) added to the training loss, as in Equation (11);
$\mathcal{HS}$Morph–S: architecture with a single $\mathcal{HS}$Morph*Dual* layer from Equation (9), equivalent to two $\mathcal{HS}$Morph layers with shared weights;
$\mathcal{HS}$Morph–SC: with one $\mathcal{HS}$Morph*Dual* layer combined with the penalty term.

Figure 4 presents the obtained results for five target structuring functions: two challenging cases (*cross3* and *disk2*) from [8, 6] where standard two-layer $\mathcal{S}$Morph networks failed, one successful case (*complex*) from the same study, and the two asymmetric structuring functions (*adiag* and *brand*) introduced in this work. The three remaining targets (*cross7*, *disk3*, and *diamond3*) from [8, 6] were successfully learned by all four network configurations and are omitted from

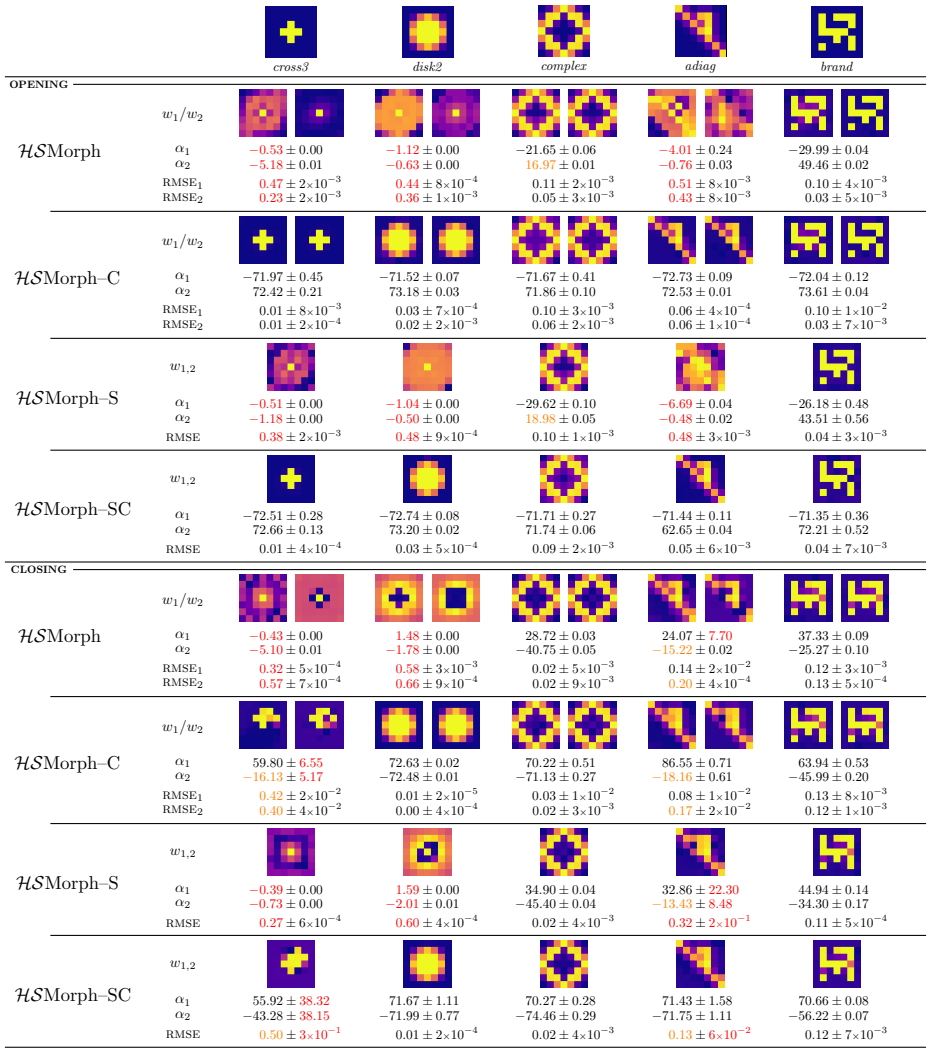


Fig. 4. Learning results for the *opening* (first four rows) and *closing* (last four) operations using four variants of a two-layer $\mathcal{HS}$Morph network on MNIST: learned filters w_1 and w_2, corresponding parameters α_1 and α_2, and RMSE between w_1, w_2 and the target structuring element across five elements (columns). Each metric shows *mean* $\pm$ *standard deviation* over five runs. Convergence failures appear in red; medium results in orange.

Figure 4 for brevity. In the three cases where the standard $\mathcal{HS}$Morph network fails (*cross3*, *disk2*, and *adiag*), both $\mathcal{HS}$Morph–C and $\mathcal{HS}$Morph–SC configurations systematically resolve the convergence issue, with the closing of *cross3* as only exception, where the reconstructed cross is incomplete and misaligned, yielding a poor RMSE. In contrast, the $\mathcal{HS}$Morph–S configuration, which relies on a dual

layer with shared weights, does not provide any convergence improvements. Thus, the control parameters α_1 and α_2 play a far more decisive role than enforcing a shared kernel w between consecutive layers for learning opening and closing operations. Convergent two-layer networks naturally align their filters w_1 and w_2 to the target filter without requiring weight sharing, while α_1 and α_2 directly determine the nature and quality of the operation performed by each layer. In contrast, enforcing shared weights in the $\mathcal{HS}$Morph*Dual* layer may overly restrict the solution space, making it harder to reach a suitable minimum, which likely explains the lack of improvement with the $\mathcal{HS}$Morph–S configuration. Overall, these results show that promoting divergence of control parameters has a much significant impact than strict weight sharing in ensuring convergence and accurate learning of opening and closing operations.

4 Application to a Denoising Task

We finally evaluate the improvement strategies proposed in the previous Section 3 for deeper network architectures. Specifically, we investigate the performance of four-layer $\mathcal{HS}$Morph networks on a salt-and-pepper denoising task, a classical setting for evaluating morphological approaches [5]. Given the nature of this noise, morphological networks are expected to converge toward morphological alternating filters [17], *i.e.*, an opening followed by a closing or vice versa. We compare the same four configurations as in Section 3, adapted to a four-layer architecture: a baseline $\mathcal{HS}$Morph network, a $\mathcal{HS}$Morph–C network with constrained control parameters, a $\mathcal{HS}$Morph–S network composed of two dual layers with shared weights, and a $\mathcal{HS}$Morph–SC network combining both the constraint and the dual layers. For the constrained configurations, we replace the penalty term $C(\alpha_1, \alpha_2)$ in Equation (11) with a new term $C(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ designed for a four-layer configuration:

$$C(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \frac{1}{3}\big(C(\alpha_1, \alpha_2) + C(\alpha_2, -\alpha_3) + C(\alpha_3, \alpha_4)\big) \tag{13}$$

Defined as such, this adapted constraint loss encourages the first pair (α_1, α_2) and the last pair (α_3, α_4) of layers to have opposite signs, and the middle pair (α_2, α_3) of layers to have the same sign, thus guiding the four-layer network to a behavior consistent with the expected morphological alternating filtering.

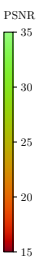
In addition to the four $\mathcal{HS}$Morph configurations, we also evaluate three other four-layer models: an exact morphological network (using the MorphoLayers library [21]), a smooth network using *PConv* layers [13], and a standard convolutional network [7]. Since each exact layer must be explicitly set as either an erosion or a dilation, the MorphoLayers network was configured to perform a closing followed by an opening (which is the configuration toward which $\mathcal{HS}$Morph–C networks naturally converge for most of the runs in these denoising experiments).

Table 1. Denoising results on the MNIST dataset: PSNR values (dB) between clean images and network outputs on the test set after training for denoising task, for seven four-layer networks (rows) and five salt-and-pepper noise levels (columns). Each cell contains the *mean* $\mu \pm$ *standard deviation* σ (top) and the *max* value (bottom) over five runs per experiment. Best result per column in **bold**, second best underlined.

Networks \ Noise (%)	1% 	5% 	10% 	25% 	50% 
$\mathcal{HS}$**Morph**	29.70 ± 0.12 max: 29.81	**28.33** ± 0.24 max: 28.61	25.58 ± 0.41 max: 26.31	20.52 ± 0.05 max: 20.59	18.44 ± 0.01 max: 18.45
$\mathcal{HS}$**Morph-C**	**31.51** ± 0.02 max: 31.53	27.64 ± 0.02 max: 27.66	**25.71** ± 0.26 max: 26.09	**22.38** ± 0.14 max: **22.73**	**20.12** ± 0.03 max: 20.16
$\mathcal{HS}$**Morph-S**	28.41 ± 0.22 max: 28.86	24.40 ± 0.01 max: 24.41	22.92 ± 0.01 max: 22.92	20.31 ± 0.07 max: 20.40	18.29 ± 0.01 max: 18.29
$\mathcal{HS}$**Morph-SC**	31.38 ± 0.53 max: 32.42	24.66 ± 0.47 max: 25.06	22.22 ± 0.01 max: 22.22	19.75 ± 0.01 max: 19.76	19.25 ± 0.32 max: 19.63
MorphoL [21]	24.56 ± 1.22 max: 26.26	23.26 ± 1.72 max: 26.59	24.89 ± 1.51 max: 26.61	21.98 ± 0.44 max: 22.46	19.00 ± 0.37 max: 19.26
PConv [13]	27.88 ± 1.41 max: 30.04	23.75 ± 1.15 max: 25.23	23.54 ± 1.11 max: 25.28	21.05 ± 1.08 max: 22.05	17.78 ± 0.62 max: 18.38
Conv [7]	26.80 ± 8.69 max: **34.70**	18.67 ± 9.90 max: **30.05**	18.92 ± 4.28 max: **26.88**	17.30 ± 2.49 max: 22.29	19.58 ± 1.74 max: **20.85**

Table 2. Denoising results on the FashionMNIST dataset: PSNR values (dB) between clean images and network outputs on the test set after training for denoising task, for seven four-layer networks (rows) and five salt-and-pepper noise levels (columns). Each cell contains the *mean* $\mu \pm$ *standard deviation* σ (top) and the *max* value (bottom) over five runs per experiment. Best result per column in **bold**, second best underlined.

Networks \ Noise (%)	1% 	5% 	10% 	25% 	50% 
$\mathcal{HS}$**Morph**	26.33 ± 0.01 max: 26.34	24.34 ± 0.01 max: 24.35	23.78 ± 0.02 max: 23.80	22.66 ± 0.15 max: 22.95	**20.39** ± 0.14 max: **20.51**
$\mathcal{HS}$**Morph-C**	27.81 ± 0.41 max: 28.58	**26.37** ± 0.01 max: 26.38	**25.42** ± 0.01 max: **25.43**	**23.17** ± 0.03 max: **23.22**	20.27 ± 0.02 max: 20.29
$\mathcal{HS}$**Morph-S**	**29.99** ± 0.01 max: 30.00	25.83 ± 0.97 max: 26.32	22.88 ± 0.01 max: 22.89	20.64 ± 0.01 max: 20.65	20.15 ± 0.01 max: 20.16
$\mathcal{HS}$**Morph-SC**	25.30 ± 0.01 max: 25.31	24.75 ± 0.04 max: 24.81	23.75 ± 0.09 max: 23.83	20.28 ± 0.04 max: 20.34	20.17 ± 0.01 max: 20.18
MorphoL [21]	22.79 ± 0.49 max: 23.31	22.29 ± 0.60 max: 23.18	22.83 ± 0.77 max: 24.15	21.33 ± 0.88 max: 22.61	19.51 ± 0.28 max: 19.80
PConv [13]	26.84 ± 1.31 max: 29.33	25.54 ± 0.67 max: 26.18	23.54 ± 0.63 max: 24.70	22.61 ± 0.38 max: 23.12	17.61 ± 0.89 max: 18.69
Conv [7]	16.36 ± 9.98 max: **30.95**	15.26 ± 8.46 max: **29.10**	16.26 ± 8.44 max: 25.05	16.06 ± 6.58 max: 22.11	15.22 ± 6.09 max: 19.76



The *PConv* network includes batch rescaling in $[1, 2]$ before each layer (see [8]). The convolutional baseline uses standard convolution layers, each being followed by the ReLU activation function, with the same basic structure as in [7, 11, 23], *i.e.*, a network made of four successive convolution-ReLU layers, but without batch normalization layers in order to keep the same architecture as morphological networks (see Figure 1). All models were trained independently on all 60k images in MNIST and Fashion-MNIST [22] datasets, separately, using salt-and-pepper noisy images as inputs and the corresponding clean images as targets. Performance was then evaluated on their test sets (containing 10k images each).

Tables 1 and 2 summarize the denoising performance of the seven network configurations at various noise levels, for MNIST and Fashion-MNIST, respectively. For each experiment, we report the mean and standard deviation of the peak signal-to-noise ratio (PSNR), computed over five runs and averaged over the entire test set, as well as the maximum PSNR achieved. Across nearly all experiments, the $\mathcal{HS}$Morph–C network consistently achieves the highest mean PSNR, regardless of the dataset or noise level. In the few cases where it does not, it ranks second best. Conversely, $\mathcal{HS}$Morph–S and $\mathcal{HS}$Morph–SC models offer no significant improvement over the baseline $\mathcal{HS}$Morph model, further supporting the conclusion from Section 3.3: promoting divergence between the control parameters with the penalty term added to the loss significantly improves the convergence and learning of smooth morphological networks with $\mathcal{HS}$Morph layers.

Moreover, the $\mathcal{HS}$Morph–C model consistently outperforms both the smooth *PConv* network and the exact MorphoLayers approach, suggesting that the flexible nature of smooth morphological layers, when properly regularized, offers a more effective alternative to both fixed morphological operators and earlier smooth formulations. Finally, the convolutional network presents poor performances on average, its high standard deviation indicating frequent convergence issues and unstable training. However, as highlighted by the maximum PSNR over five runs (displayed below the average and standard deviation values in each cell of Tables 1 and 2), successful runs often achieve the highest PSNR among all networks, sometimes significantly. This instability is a known limitation of convolutional architectures without batch normalization layers [7, 11]. While such layers generally improve convergence and denoising performance, they were omitted here to allow a fair comparison with the morphological models which all follow the architecture depicted in Figure 1. As the convolutional network without batch normalization and the $\mathcal{HS}$Morph network used for these denoising experiments have the same number of learnable parameters (each layer contains the filter weights, plus the biais or the control parameter, associated with the convolutional or smooth morphological layer, respectively), adding new learnable parameters through batch normalization layers only in the convolutional model could significantly advantage its convergence and learning capacities. In future work, new experiments could be conducted that integrate batch normalization layers added before convolutional and morphological layers, in both convolutional and smooth morphological architectures, respectively, to assess the superiority of the former with normalization layers for such denoising tasks.

5 Conclusion

In this work, we introduced a harmonized version of the $\mathcal{S}$Morph layer that overcomes asymptotic limitations while maintaining the performance of one-layer networks when learning erosion and dilation operations. We also proposed a novel penalty term added to the training loss function and shared-weight dual layers to improve the convergence of two-layer networks for learning opening and closing operations, with the former proving to be significantly more effective. Finally, we investigated the proposed $\mathcal{HS}$Morph layer and its improvement strategies in a four-layer network for salt-and-pepper denoising, and compared with other morphological (smooth and exact) and convolutional networks. The obtained results confirmed the superiority of our model with the penalty term, highlighting the importance of control parameter divergence in smooth-layer networks. Future work will focus on integrating these smooth layers into deeper architectures to assess their potential in broader vision tasks, as well as performing a comprehensive benchmarking of morphological networks with classical architectures.

Acknowledgments. This work was conducted during G. Angulo's academic visiting period to NYU, partially funded by the Fondation MINES Paris.

Disclosure of Interests. The authors declare no conflict of interest.

References

1. Angulo, J.: Pseudo-morphological image diffusion using the counter-harmonic paradigm. In: International Conference on Advanced Concepts for Intelligent Vision Systems. pp. 426–437. Springer (2010)
2. Banon, G.J.F., Barrera, J.: Minimal representations for translation-invariant set mappings by mathematical morphology. SIAM Journal on Applied Mathematics **51**(6), 1782–1798 (1991)
3. Blusseau, S.: Training morphological neural networks with gradient descent: some theoretical insights. In: International Conference on Discrete Geometry and Mathematical Morphology. pp. 229–241. Springer (2024)
4. Dimitrova, M., Blusseau, S., Velasco-Forero, S.: Learning morphological representations of image transformations: Influence of initialization and layer differentiability. In: International Conference on Discrete Geometry and Mathematical Morphology. Springer (2025)
5. Franchi, G., Fehri, A., Yao, A.: Deep morphological networks. Pattern Recognition **102**, 107246 (2020)
6. Hermary, R., Tochon, G., Puybareau, É., Kirszenberg, A., Angulo, J.: Learning grayscale mathematical morphology with smooth morphological layers. Journal of Mathematical Imaging and Vision **64**(7), 736–753 (2022)
7. Ilesanmi, A.E., Ilesanmi, T.O.: Methods for image denoising using convolutional neural network: a review. Complex & Intelligent Systems **7**(5), 2179–2198 (2021)
8. Kirszenberg, A., Tochon, G., Puybareau, É., Angulo, J.: Going beyond p-convolutions to learn grayscale morphological operators. In: International Conference on Discrete Geometry and Mathematical Morphology. pp. 470–482. Springer (2021)

9. Lange, M., Zühlke, D., Holz, O., Villmann, T., Mittweida, S.G.: Applications of lp-norms and their smooth approximations for gradient based learning vector quantization. In: ESANN. pp. 271–276 (2014)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
11. Mafi, M., Izquierdo, W., Martin, H., Cabrerizo, M., Adjouadi, M.: Deep convolutional neural network for mixed random impulse and gaussian noise reduction in digital images. IET Image Processing **14**(15), 3791–3801 (2020)
12. Maragos, P.: A representation theory for morphological image and signal processing. IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(6), 586–599 (1989)
13. Masci, J., Angulo, J., Schmidhuber, J.: A learning framework for morphological operators using counter–harmonic mean. In: Mathematical Morphology and Its Applications to Signal and Image Processing: 11th International Symposium, ISMM 2013, Uppsala, Sweden, May 27-29, 2013. pp. 329–340. Springer (2013)
14. Matheron, G.: Random sets and integral geometry. John Wiley & Sons (1975)
15. Mondal, R., Dey, M.S., Chanda, B.: Image restoration by learning morphological opening-closing network. Mathematical Morphology-Theory and Applications **4**(1), 87–107 (2020)
16. Nogueira, K., Chanussot, J., Dalla Mura, M., Dos Santos, J.A.: An introduction to deep morphological networks. IEEE Access **9**, 114308–114324 (2021)
17. Serra, J.: Image Analysis and Mathematical Morphology, vol. 2. Academic Press (1988)
18. Shen, Y., Shih, F.Y., Zhong, X., Chang, I.C.: Deep morphological neural networks. International Journal of Pattern Recognition and Artificial Intelligence **36**(12), 2252023 (2022)
19. Shih, F.Y., Shen, Y., Zhong, X.: Development of deep learning framework for mathematical morphology. International Journal of Pattern Recognition and Artificial Intelligence **33**(06), 1954024 (2019)
20. Velasco-Forero, S., Angulo, J.: MorphoActivation: Generalizing ReLU activation function by mathematical morphology. In: International Conference on Discrete Geometry and Mathematical Morphology. pp. 449–461. Springer (2022)
21. Velasco-Forero, S., Pagès, R., Angulo, J.: Learnable empirical mode decomposition based on mathematical morphology. SIAM Journal on Imaging Sciences **15**(1), 23–44 (2022)
22. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
23. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE Transactions on Image Processing **27**(9), 4608–4622 (2018)
24. Zhou, D.X.: Universality of deep convolutional neural networks. Applied and computational harmonic analysis **48**(2), 787–794 (2020)