Journal of Mathematical Imaging and Vision (2022) 64:736–753
https://doi.org/10.1007/s10851-022-01091-1

Learning Grayscale Mathematical Morphology with Smooth Morphological Layers

Romain Hermary[1] · **Guillaume Tochon**[1] · **Élodie Puybareau**[1] · **Alexandre Kirszenberg**[1] · **Jesús Angulo**[2]

Received: 7 July 2021 / Accepted: 4 April 2022 / Published online: 14 May 2022

Abstract
The integration of mathematical morphology operations within convolutional neural network architectures has received an increasing attention lately. However, replacing standard convolution layers by morphological layers performing erosions or dilations is particularly challenging because the min and max operations are not differentiable. P-convolution layers were proposed as a possible solution to this issue since they can act as smooth differentiable approximation of min and max operations, yielding pseudo-dilation or pseudo-erosion layers. In a recent work, we proposed two novel morphological layers based on the same principle as the p-convolution, while circumventing its principal drawbacks, and showcased their capacity to efficiently learn grayscale morphological operators while raising several edge cases. In this work, we complete those previous results by thoroughly analyzing the behavior of the proposed layers and by investigating and settling the reported edge cases. We also demonstrate the compatibility of one of the proposed morphological layers with binary morphological frameworks.

Keywords Morphological layer · P-convolution · Counter-harmonic mean · Grayscale mathematical morphology · Morphological neural network

1 Introduction

Mathematical morphology is concerned with the nonlinear filtering of images [23], in which the elementary operations amount to compute the minimum or maximum of all pixel values within a neighborhood of some given shape and size of the pixel under study. This neighborhood is the so-called *structuring element*, and applying the maximum (resp. minimum) yields the well-known morphological dilation (resp. erosion) operation. Combining those elementary operations, one can define more advanced (but still nonlinear) filters, such as openings and closings, top-hat transforms or alternate sequential filters, which have many times proven successful for various image processing tasks such as filtering, segmentation or edge detection [6,23,24,27]. However, finding the optimal sequence of operations to chain and designing the shape and size of their respective structuring element is always quite arduous and generally done in a tedious and time-consuming trial-and-error fashion, since it also depends on the pursued application. Thus, there is a practical need to automatize the identification of the right sequence of operations to use and their structuring element. Recent advances in machine learning techniques, and particularly in the field of neural networks, are an attractive direction to investigate in that respect.

As a matter of fact, the structural similarity between neuron operations (weighted linear combination of input values, potentially mapped by nonlinear activation functions) and elementary morphological operations such as erosion and dilation makes it tempting to substitute the former by the latter. This results in the adaptation of traditional multi-layer

✉ Guillaume Tochon
guillaume.tochon@lrde.epita.fr

Romain Hermary
romain.hermary@lrde.epita.fr

Élodie Puybareau
elodie.puybareau@lrde.epita.fr

Alexandre Kirszenberg
alexandre.kirszenberg@lrde.epita.fr

Jesús Angulo
jesus.angulo@mines-paristech.fr

[1] EPITA Research and Development Laboratory (LRDE), Le Kremlin-Bicêtre, France

[2] Centre for Mathematical Morphology, Mines ParisTech, PSL Research University, Fontainebleau, France

perceptron architectures [9] into morphological perceptrons and convolutional layers implemented in deep convolutional neural networks [13] (CNNs) into morphological layers. Deep CNNs appear as particularly appealing solutions since the weights of each layer filter could be interpreted as (non-necessary flat) structuring elements, provided that the conventional convolution operation has been replaced by erosion or dilation and that the layer has a way to learn which operation to use. However, as tempting as the idea of replacing the standard convolution by morphological operations might be, it comes in practice with several challenges, the major one being that the min and max operations are not differentiable and thus do not lend well to stochastic gradient descent optimization approaches [13]. Based on asymptotic properties of counter-harmonic mean [1,3] (CHM), the p-convolution (hereafter abbreviated as $PConv$) layer has been proposed by Masci et al. [15] as a workaround to overcome this non-differentiability issue since it is not only smooth, but it can also approximate non-flat dilations, erosions and classical convolutions, depending on the value of its (trainable) inner parameter p. Very promising results were presented by Masci et al. [15] in terms of learning morphological operations and their structuring element, but never further investigated. Following this idea, we recently proposed in Kirszenberg et al. [11] two new morphological layers, namely the $\mathscr{L}$Morph layer (also based on the CHM framework) and the $\mathscr{S}$Morph layer (based on the regularized softmax approximation), both compatible with grayscale mathematical morphology and non-flat structuring elements. In this preliminary study our motivation was, much alike in the famous XOR-learning experiments for neural networks, to determine whether the particular layers we proposed can capture some particular representation of interest, as a sanity test prior to further experiments in larger architectures. In this sense, we showcased the ability of these two new layers to learn grayscale morphological operations and reported better results than the $PConv$ layer. However, several edge cases were also identified and the conducted experiments remained relatively shallow in terms of practical validation.

In this article, we investigate more in-depth the $\mathscr{L}$Morph and $\mathscr{S}$Morph layers introduced by Kirszenberg et al. [11], in terms of both morphological and learning-based behaviors. More specifically, the novel contributions of this article with respect to Kirszenberg et al. [11] are the following:

– We thoroughly analyze the performances of $\mathscr{L}$Morph and $\mathscr{S}$Morph layers for non-flat structuring elements. In particular, we now conduct several runs per learning scenario, allowing to evaluate the stability of the learned solutions.
– The edge cases regarding some convergence issues for some scenarios reported by Kirszenberg et al. [11] are investigated and (partially) settled.
– Although it was initially designed for grayscale morphological operations with non-flat structuring elements, the $\mathscr{S}$Morph layer is also able to learn flat (binary) structuring elements. Thus, we also examine the performances of the $\mathscr{S}$Morph layers in binary mathematical morphology scenarios.

This article is organized as follows. In Sect. 2, we review the related works devoted to the integration of mathematical morphology within neural network architectures, with a focus on convolutional-like morphological networks. In Sects. 3 and 4, we present the $PConv$ layer introduced by Masci et al. [15] and the $\mathscr{L}$Morph and $\mathscr{S}$Morph layers that we proposed in Kirszenberg et al. [11]. Sections 5 and 6 gather the conducted experiments and analyses for grayscale morphological operations learning for the former and binary morphological operations learning for the latter. Finally, Sect. 7 draws some conclusions and perspectives from our contributions.

2 Related Work

The interaction between mathematical morphology and neural network architectures is not a novel topic [2]. As a matter of fact, morphological neural networks were introduced in the late 1980s with a definition of neurons as weighted rank filters [32] or, in a less general form, as performing dilations or erosions [7]. Replacing the multiplication and addition of linear perceptron units with addition and minimum/maximum operators induces a new geometry of decision surfaces, referred to as *bounding box* geometry [21,28,30,33], and alternative (or complementary) strategies to gradient descent in networks training. Hybrid approaches mixing linear and morphological layers have also been developed for an even richer geometry [5,10,20,29,34], and dilation layers showed interesting pruning properties when located after linear layers [5,31,34]. The latter studies only considered dense layers and are little suited to image analysis, in contrast to convolutional neural networks which can handle large images.

The rise of deep CNN architectures in the early 2010s motivated the exploration of integrating elementary morphological operations in such networks to automatically learn their optimal shape and weights. A first workaround to overcome the non-differentiability of the min and max operations of erosions and dilations in convolutional-like approaches is to replace them by smooth differentiable approximations, making them suited to the conventional gradient descent learning approach via back-propagation [13]. In their seminal work, Masci et al. [15] provided the p-convolution ($PConv$) layer by relying on some properties of the counter-harmonic mean [3] (CHM). The $PConv$ layer depends on a trainable parameter p, whose value allows the layer to behave as a classical convolution or as a smooth approximation of either dilation or erosion (depending on the sign of p). The CHM was also used as an alternative to the standard max-pooling layer in classical CNN architectures [16] and applied to digit recognition tasks [17]. LogSumExp functions [4] (also known as multivariate softplus) were proposed as replacements of min and max operations to learn binary [26] and grayscale [25] structuring elements.

Alternatively, non-smooth operators such as the ReLU activation function or max-pooling layers (to name a few) have also been efficiently trained with stochastic gradient descent algorithms. As a matter of fact, those operators are actually differentiable almost everywhere, and a descent direction can be defined even in their zero-measure non-smooth regions. Thus, translation-invariant morphological layers were recently optimized just as usual convolutional ones [8,18,19,22] and applied to image classification, denoising, restoration as well as edge detection.

With our approach, which is using fully differentiable functions and pseudo-morphological operations, we expect the proposed layers to be more flexible than the previously cited works using exact operations since we are adding an extra degree of freedom. Increasing the learning possibilities and potency of morphological layers is part of our motivation, but, as mentioned in the previous section, our work is still preliminary and performance comparison with advanced networks on user-oriented problematic cannot be expressed pertinently in this publication.

As noted by Kirszenberg et al. [11], the integration of morphological operations in place of standard convolutions in deep CNN architectures is clearly a hot topic since all cited approaches exploring this research direction are very recent (apart from the article of Masci et al. [15], all other works date back to no later than 2017).

3 The P-Convolution Layer

This section is devoted to mathematical morphology preliminaries as well as the presentation of the $PConv$ layer introduced by Masci et al. [15].

3.1 Grayscale Mathematical Morphology

In mathematical morphology, it is common to represent an image as a 2D function $f : E \to \mathbb{R}$ with $x \in E$ being the pixel coordinates in the 2D grid $E \subseteq \mathbb{Z}^2$ and $f(x) \in \mathbb{R}$ being the pixel value. In this formalism, a binary structuring element $B \subseteq E$ defines a neighborhood of a given shape (square, circle, diamond, etc.), size and origin on the pixel grid. If B_x denotes the translation of B by x (that is, the origin of B is translated on x), then the erosion $f \ominus B$ and dilation $f \oplus B$ of image f with the binary structuring element B are

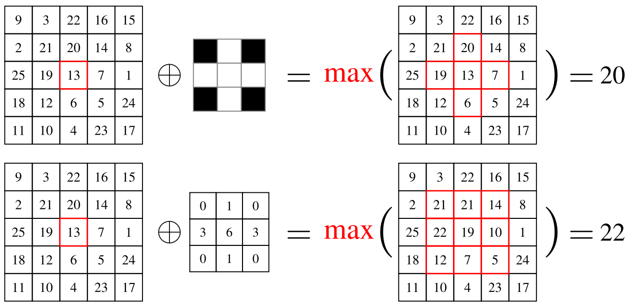


Fig. 1 Top: illustration of the dilation of the pixel 13 (with red borders) with a cross-shaped binary structuring element. Bottom: illustration of the dilation of the same pixel 13 with a grayscale structuring element (Color figure online)

defined as:

$$(f \ominus B)(x) = \inf \{f(y), y \in B_x\} \tag{1}$$

$$(f \oplus B)(x) = \sup \{f(y), y \in B_x\}\,. \tag{2}$$

In grayscale mathematical morphology, the structuring element b is defined as a real-valued function $b : E \to \mathbb{R}$, and erosion $f \ominus b$ and dilation $f \oplus b$ operations are written as:

$$(f \ominus b)(x) = \inf_{y \in E} \{f(y) - b(x-y)\} \tag{3}$$

$$(f \oplus b)(x) = \sup_{y \in E} \{f(y) + b(x-y)\}\,. \tag{4}$$

This formalism also encompasses the use of a flat (binary) structuring element B by defining the structuring function b as

$$b(x) = \begin{cases} 0 & \text{if } x \in B \\ -\infty & \text{otherwise} \end{cases}. \tag{5}$$

In that case, Eq. (3) (resp. Eq. (4)) coincides with Eq. (1) (resp. Eq. (2)). Figure 1 illustrates the difference between a dilation with a binary structuring element and a grayscale structuring element. In the first case, the binary structuring element defines a neighborhood around a given pixel, and the dilation amounts to taking the maximum value within this neighborhood. In the case of a grayscale dilation, the grayscale structuring element again defines a neighborhood around the pixel to be dilated, but the maximum value is taken on the sum of the image and structuring element contents.

3.2 The Counter-Harmonic Mean

Let $\mathbf{x} = (x_1, \ldots, x_n) \in (\mathbb{R}^+)^n$ and $\mathbf{w} = (w_1, \ldots, w_n) \in (\mathbb{R}^+)^n$ be two non-negative vectors, and $p \in \mathbb{R}$. The weighted counter-harmonic mean (CHM), also known as the Lehmer mean [3] of order $p \in \mathbb{R}$ of vector $\mathbf{x}$ with weights $\mathbf{w}$, is defined as

$$CHM(\mathbf{x}, \mathbf{w}, p) = \frac{\sum_{i=1}^{n} w_i x_i^p}{\sum_{i=1}^{n} w_i x_i^{p-1}} . \tag{6}$$

Defined as such, the Lehmer mean $CHM(\mathbf{x}, \mathbf{w}, p)$ corresponds to the weighted harmonic mean (resp. weighted arithmetic mean) for $p = 0$ (resp. $p = 1$) and to the contraharmonic mean when $p = 2$ and $\mathbf{w} = \mathbf{1}$ (all entries of $\mathbf{w}$ being equal to 1). In addition, because both vectors $\mathbf{x}$ and $\mathbf{w}$ have non-negative entries, the maximum (resp. minimum) among all entries of $\mathbf{x}$ dominates the numerator and denominator of Eq. (9) when p goes to $+\infty$ (resp. $-\infty$) , yielding the following asymptotic behavior for the CHM:

$$\lim_{p\to+\infty} CHM(\mathbf{x}, \mathbf{w}, p) = \sup_{i\in[\![1,n]\!]} x_i , \tag{7}$$

$$\lim_{p\to-\infty} CHM(\mathbf{x}, \mathbf{w}, p) = \inf_{i\in[\![1,n]\!]} x_i . \tag{8}$$

Thus, the CHM also allows for smooth approximations of min and max operations for large negative and positive values of p.

3.3 The P-Convolution Layer

The p-convolution (also known as *p-deformed convolution*) of an image f at pixel x for a given non-negative convolution kernel $w : W \subseteq E \to \mathbb{R}^+$ and some $p \in \mathbb{R}$ has been defined by Masci et al. [15] as:

$$\begin{aligned} PConv(f, w, p)(x) &= (f *_p w)(x) \\ &= \frac{(f^{p+1} * w)(x)}{(f^p * w)(x)} \\ &= \frac{\sum_{y\in W(x)} f^{p+1}(y) w(x-y)}{\sum_{y\in W(x)} f^{p}(y) w(x-y)} \end{aligned} \tag{9}$$

where $f^p(x)$ denotes the pixel value $f(x)$ raised to the power of p and $W(x)$ is the spatial support of kernel w centered at x

Based on the asymptotic properties of the CHM, the morphological behavior of the $PConv$ operation with respect to p has notably been studied by Angulo et al. [1]. More precisely, the $PConv$ operation coincides with a classical convolution with filter w when $p = 0$. When $p > 0$ (resp. $p < 0$), the $PConv$ operation can be interpreted as a pseudo-dilation (resp. pseudo-erosion), and when $p \to \infty$ (resp. $-\infty$), the largest (resp. smallest) pixel value in the local neighborhood $W(x)$ of pixel x dominates the weighted sum (9) and the $PConv(f, w, p)(x)$ acts as a non-flat grayscale dilation (resp. a non-flat grayscale erosion) with the structuring function $b(x) = \frac{1}{p}\log(w(x))$:

$$\lim_{p\to+\infty} (f *_p w)(x) = \sup_{y\in W(x)} \left\{ f(y) + \frac{1}{p} \log\left(w(x-y)\right) \right\} \tag{10}$$

$$\lim_{p\to-\infty} (f *_p w)(x) = \inf_{y\in W(x)} \left\{ f(y) - \frac{1}{p} \log\left(w(x-y)\right) \right\} . \tag{11}$$

In practice, Eqs. (10) and (11) hold true for $|p| > 10$. The flat structuring function (5) can be recovered by using constant weight kernels, i.e., $w(x) = 1$ if $x \in W$ and $w(x) = 0$ if $x \notin W$ and $|p| \gg 0$. As stated by Masci et al. [15], the $PConv$ operation is differentiable, thus compatible with gradient descent learning approaches via back-propagation. Thus, a $PConv$ layer implementing operation (9) can readily be integrated to CNN-like neural network architectures and its kernel w can be optimized in the same fashion as the weights of a classical convolution layer. However, the $PConv$ layer has an additional trainable parameter p that controls its morphological behavior. The learning scheme can alternate between learning p while keeping w fixed, and vice-versa, or the learning can be performed simultaneously on p and w.

3.4 Limits of the P-Convolution Layer

The definition and training of the $PConv$ layer comes with several numerical issues. As a matter of fact, w and f must be strictly positive in order for the $PConv$ to be defined on all its possible input parameters. Otherwise, the following numerical errors can occur:

- If $f(x)$ contains null values and p is negative, $f^p(x)$ is not defined;
- If $f(x)$ contains negative values and p is a non-null, non-integer real number, $f^p(x)$ can contain complex numbers;
- If $w(x)$ or $f^p(x)$ contain null values, $\frac{1}{(f^p * w)(x)}$ is not defined.

Therefore, in order to avoid any aforementioned issue, it is necessary to rescale the input image before feeding it to the $PConv$ layer. In practice, we experienced some numerical instabilities during training when rescaling the input within the range [0, 1], so we rescale between [1, 2] instead:

$$f_r(x) = 1.0 + \frac{f(x) - \min_{x\in E} f(x)}{\max_{x\in E} f(x) - \min_{x\in E} f(x)} . \tag{12}$$

Moreover, if several $PConv$ layers are concatenated one behind the other (to achieve (pseudo-) opening and closing operations for instance), a rescaling must be performed before each layer. Particular care must also be taken with the output of the last $PConv$ layer of the network. Indeed, because of these rescaling operations throughout the network, the range of the output may not match the range of the target, which can be problematic for MSE-based training losses. Thus, to avoid this effect, a trainable scale/bias $1 \times 1 \times 1$ convolution layer is added at the end of the network.

Last but not least, a notable drawback of the $PConv$ layer when it comes to learning a specific (binary or non-flat) structuring element is that it tends to be hollow and flattened out in the center (see further presented results in Sect. 5).

4 Proposed $\mathscr{L}$Morph and $\mathscr{S}$Morph Layers

In this section, we detail two novel morphological layers introduced by Kirszenberg et al. [11]. Similar to the $PConv$ operation, those two layers intend to act like trainable smooth approximations of min and max operators, thus achieving (pseudo-) erosion and dilation while being compatible with general grayscale mathematical morphology.

4.1 The $\mathscr{L}$Morph Layer

The first proposed layer is named $\mathscr{L}$Morph (for $\mathscr{L}$ehmer-mean-based morphological operation). As the $PConv$, it is designed to rely on the asymptotic behavior of the CHM to approximate min and max operations in a smooth and differentiable manner. Specifically, if $w : W \to \mathbb{R}^+$ is the structuring function and $p \in \mathbb{R}$, the $\mathscr{L}$Morph operation is defined as:

$$\mathscr{L}\text{Morph}(f, w, p)(x) = \frac{\sum_{y \in W(x)} (f(y) + w(x-y))^{p+1}}{\sum_{y \in W(x)} (f(y) + w(x-y))^{p}}. \tag{13}$$

Defined as such, we can identify $\mathscr{L}\text{Morph}(f, w, p)$ with the CHM defined by Eq. (6): All weights w_i (resp. entries x_i) of Eq. (6) correspond to 1 (resp. $f(y) + w(x-y)$) in Eq. (13). Therefore, from the following the asymptotic behavior of the CHM explicit by Eqs. (7) and (8), we can deduce that:

$$\lim_{p \to +\infty} \mathscr{L}\text{Morph}(f, w, p)(x) = \sup_{y \in W(x)} \{f(y) + w(x-y)\} = (f \oplus w)(x) \tag{14}$$

$$\lim_{p \to -\infty} \mathscr{L}\text{Morph}(f, w, p)(x) = \inf_{y \in W(x)} \{f(y) + w(x-y)\} = (f \ominus -w)(x). \tag{15}$$

As it was the case for the $PConv$, one can achieve either pseudo-dilation (if $p > 0$) or pseudo-erosion (if $p < 0$) with the $\mathscr{L}$Morph operation, depending on the sign of its control parameter p. In addition, when $p \to +\infty$, the $\mathscr{L}$Morph layer asymptotically converges toward the dilation of image f with structuring function w. Note, however, that when $p \to -\infty$, the $\mathscr{L}$Morph operation converges toward the erosion of f with structuring function $-w$. However, this is not a problem in a learning scenario since we can easily retrieve w from $-w$ by checking whether the sign of p is negative. Figure 2 displays examples of applying the $\mathscr{L}$Morph function with a given non-flat structuring element for differ-

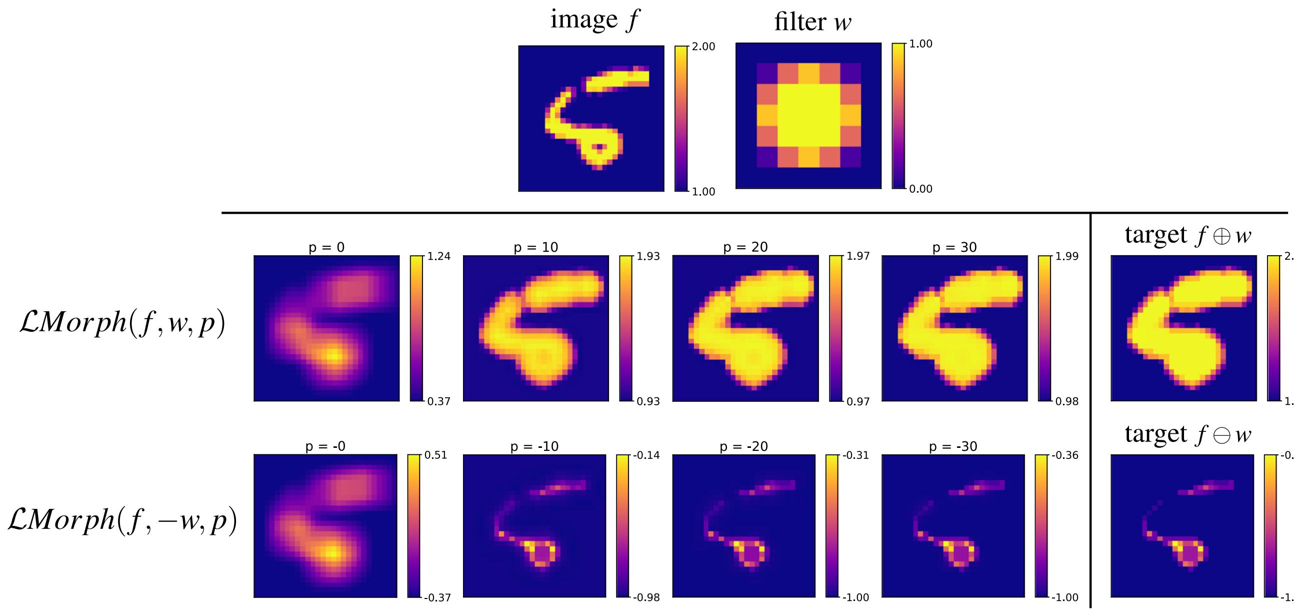


Fig. 2 Top row: input image from the MNIST database and non-flat structuring element w. Middle row: $\mathscr{L}$Morph pseudo-dilation for $p \in \{0, 10, 20, 30\}$, and target dilation $f \oplus w$. Bottom row: $\mathscr{L}$Morph pseudo-erosion for $p \in \{0, -10, -20, -30\}$ and target erosion $f \ominus w$. Note that for the erosion, $-w$ is used in $\mathscr{L}$Morph instead of w to approximate the target $f \ominus w$

ent values of p. As it can be seen, $|p| > 20$ is sufficient in practice to reproduce non-flat grayscale dilation or non-flat grayscale erosion.

Relying on the CHM like the $PConv$ layer brings over some shared limitations for the proposed $\mathscr{L}$Morph layer: The input image f must be positive and rescaled following Eq. (12), and the structuring function w must be positive or null.

4.2 The $\mathscr{S}$Morph Layer

Both the $PConv$ and $\mathscr{L}$Morph layers rely on the asymptotic behavior of the CHM to approximate min and max operations. The major drawback is that the input must be rescaled within the range $[1, 2]$. In order to circumvent this issue, we leverage the α-softmax function [12]. Given some $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, the α-softmax $\mathscr{S}_\alpha(\mathbf{x})$ is defined as:

$$\mathscr{S}_\alpha(\mathbf{x}) = \frac{\sum_{i=1}^n x_i e^{\alpha x_i}}{\sum_{i=1}^n e^{\alpha x_i}} \tag{16}$$

For $\alpha = 0$, $\mathscr{S}_0(\mathbf{x})$ coincides with the arithmetic mean of $\mathbf{x} = (x_1, \ldots, x_n)$. In addition, $\lim_{\alpha\to+\infty} \mathscr{S}_\alpha(\mathbf{x}) = \sup_i x_i$ and $\lim_{\alpha\to-\infty} \mathscr{S}_\alpha(\mathbf{x}) = \inf_i x_i$. This function is less restrictive than the CHM since it does not require the entries of $\mathbf{x}$ to be strictly positive. A major benefit is that it is no longer necessary to rescale its input.

Exploiting this property, we define the $\mathscr{S}$Morph (standing for smooth morphological) operation of an input image f with structuring function $w : W \to \mathbb{R}$ as:

$$\mathscr{S}\text{Morph}(f, w, \alpha)(x) = \frac{\sum_{y\in W(x)} (f(y) + w(x-y)) e^{\alpha(f(y)+w(x-y))}}{\sum_{y\in W(x)} e^{\alpha(f(y)+w(x-y))}}. \tag{17}$$

The $\mathscr{S}$Morph operation simply follows from the definition of the α-softmax function, replacing entries x_i in Eq. (16) by $f(y) + w(x - y)$. Thus, the asymptotic behavior of $\mathscr{S}\text{Morph}(f, w, \alpha)$ is the same as this of the α-softmax:

$$\lim_{\alpha\to+\infty} \mathscr{S}\text{Morph}(f, w, \alpha)(x) = \sup_{y\in W(x)} \{f(y) + w(x-y)\} = (f \oplus w)(x) \tag{18}$$

$$\lim_{\alpha\to-\infty} \mathscr{S}\text{Morph}(f, w, \alpha)(x) = \inf_{y\in W(x)} \{f(y) + w(x-y)\} = (f \ominus -w)(x). \tag{19}$$

Like the $PConv$ and $\mathscr{L}$Morph layers, the proposed $\mathscr{S}$Morph operation can perform a pseudo-dilation ($\alpha > 0$) or a pseudo-erosion ($\alpha < 0$), depending on the sign of its parameter α. Moreover, the $\mathscr{S}$Morph layer asymptotically converges toward the dilation $f \oplus w$ of image f with structuring function w when $\alpha \to +\infty$. When $\alpha \to -\infty$, the behavior of $\mathscr{S}$Morph is the same as $\mathscr{L}$Morph, namely that it converges toward the erosion $f \ominus -w$ of f with structuring function $-w$.

Figure 3 shows examples of applying the $\mathscr{S}$Morph function with a given non-flat structuring element for different values of α. The operation better approximates the target

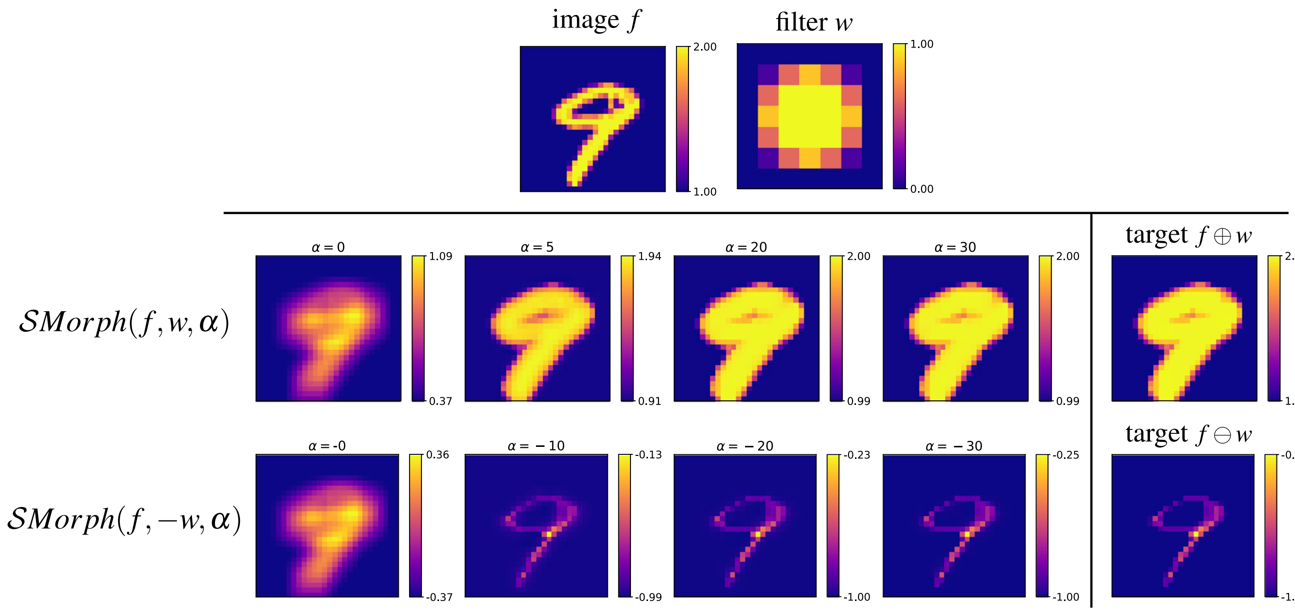


Fig. 3 Top row: input image from the MNIST database and non-flat structuring element w. Middle row: $\mathscr{S}$Morph pseudo-dilation for $\alpha \in \{0, 5, 20, 30\}$, and target dilation $f \oplus w$. Bottom row: $\mathscr{S}$Morph pseudo-erosion for $\alpha \in \{0, -5, -20, -30\}$ and target erosion $f \ominus w$. Note that for the erosion, $-w$ is used in $\mathscr{S}$Morph instead of w to approximate the target $f \ominus w$

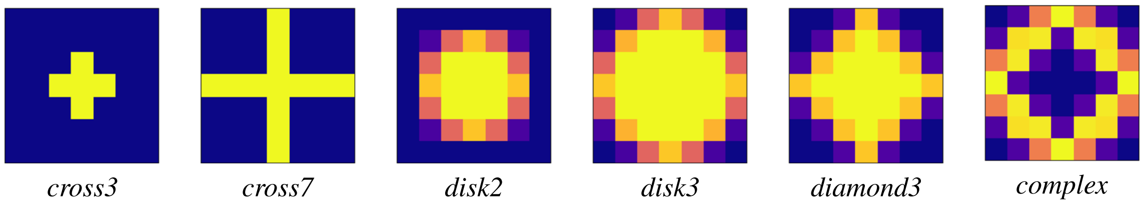


Fig. 4 7×7 target grayscale structuring elements. All values range between 0 (deep blue) and 1 (yellow) (Color figure online)

dilation or erosion when $|\alpha|$ increases. Note that the input image has also been rescaled in the range $[1, 2]$ to allow for a fair visual comparison with the results obtained for $\mathscr{L}$Morph and is displayed in Fig. 2 (although the rescaling is not necessary for the $\mathscr{S}$Morph layer).

5 Learning Grayscale Morphological Operators

In this section, we evaluate the ability of the $\mathscr{L}$Morph and $\mathscr{S}$Morph layers to properly learn a target grayscale structuring element and a morphological operation, and compare with the results obtained by the $PConv$ layer (which serves as a baseline due to its structural and conceptual similarities with the two proposed morphological layers).

In any case, the overall idea is the following: We apply one morphological operation among erosion $\ominus$, dilation $\oplus$, opening $\circ$ and closing $\bullet$ with one of the grayscale structuring elements displayed in Fig. 4 to all 60,000 digit images of the MNIST database [14], and we challenge the competing morphological layers to retrieve the correct target structuring element and morphological operation when fed with couples composed of original images and their transformed image. Hereafter, a *scenario* will denote a combination morphological operation/structuring element/morphological layer, hence resulting in a total of $4 \times 6 \times 3 = 72$ different investigated scenarios.

5.1 Learning Grayscale Erosions and Dilations

5.1.1 Experimental Protocol

In this first experiment, we focus on morphological erosions $\ominus$ and dilations $\oplus$, thus reducing to 36 different scenarios. As a matter of fact, all three morphological layers $PConv$, $\mathscr{L}$Morph and $\mathscr{S}$Morph should natively be able to approximate both operations, depending on the value of their inner parameter p (for both $PConv$ and $\mathscr{L}$Morph layers) or α (for the $\mathscr{S}$Morph layer). Thus, the implemented network architecture for this first set of scenarios is straightforward, as it is composed of a single morphological layer. Nevertheless, both $PConv$ and $\mathscr{L}$Morph require the input images to be rescale in the range $[1, 2]$ as discussed in Sects. 3.4 and 4.1. This constraint enforces the addition of a classical rescaling block before the morphological layer, and a trainable scale/biais $Conv$ $1 \times 1 \times 1$ after the layer, in order to rescale the network output into the range of the target images. Although the $\mathscr{S}$Morph layer does not suffer from this rescaling drawback, we use the same architecture to allow for a fair comparison between all three morphological layers. Figure 5 displays such architecture. It is also worth mentioning that, due to the shared limitation of $PConv$ and $\mathscr{L}$Morph layers to oper-

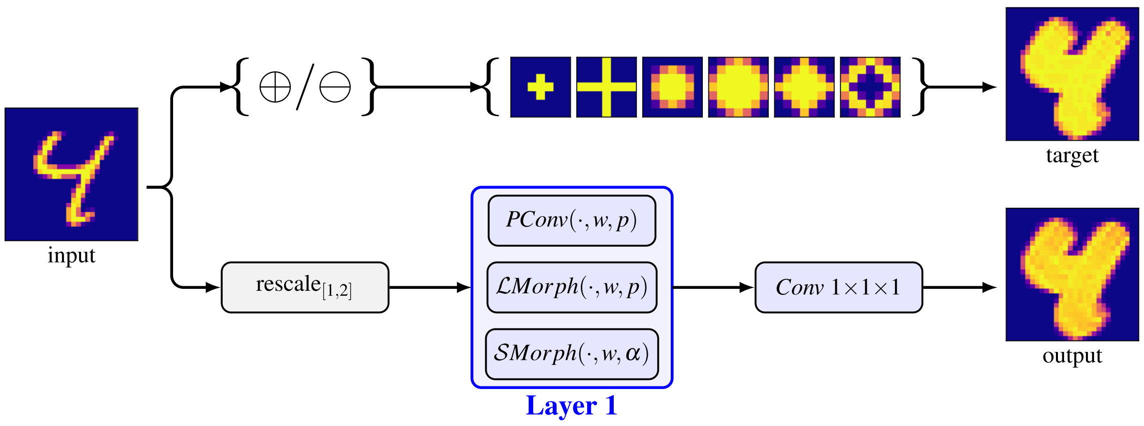


Fig. 5 Network architecture used for the erosion/dilation scenarios. Blue blocks are trainable units. A scenario is defined as the choice of $\oplus$ or $\ominus$ and one of the six target structuring elements in the upper path, and the choice of one layer among $PConv$, $\mathscr{L}$Morph and $\mathscr{S}$Morph in the lower path (Color figure online)

ate on filters with non-negatives weights only, the values of all six grayscale structuring elements displayed in Fig. 4 are comprised between 0 and 1.

All networks are trained with a batch size of 32, optimizing for the mean squared error (MSE) loss with the Adam optimizer (with starting learning rate $\eta = 0.01$). The learning rate of the optimizer is scheduled to decrease by a factor of 10 when the loss plateaus for five consecutive epochs. Convergence is reached when the loss plateaus for 10 consecutive epochs. The maximal number of training epochs is set to 1000. For the $PConv$ layer, the filter is initialized with 1s and $p = 0$. For $\mathscr{L}$Morph, the filter is initialized with a folded normal distribution with standard deviation $\sigma = 0.01$, and $p = 0$. For the $\mathscr{S}$Morph layer, the filter is initialized with a centered normal distribution with standard deviation $\sigma = 0.01$ and $\alpha = 0$. In all instances, the training is done simultaneously on the weights and the parameter p or α.

In comparison with the results presented by Kirszenberg et al. [11] where only one run per scenario was conducted, we now perform five training runs per scenario in order to investigate the performances of the morphological networks in terms of stability and repeatability. More specifically, we report the average and standard deviation over the five runs of the value of parameter p/α at convergence, the root mean square error (RMSE) between the filter learned by the morphological layer at convergence and the target filter, the training loss as well as the number of training epochs. It is worth mentioning that because of the initial rescaling of the input images in the range [1, 2] combined with the final scale/bias $Conv$ $1 \times 1 \times 1$ in the network, there is no guarantee that the range of the learned filter is [0, 1] as it is the case for the target structuring elements. Thus, the reported RMSE values are computed between the target structuring element and the learned filter scaled in the range [0, 1].

5.1.2 Obtained Results

Figure 6 presents the filter learned by the $PConv$, $\mathscr{L}$Morph and $\mathscr{S}$Morph layers along with the quantitative metrics defined at the end of Sect. 5.1.1 for the erosion $\ominus$ scenarios. Its analysis leads to several comments: First, looking at the sign of parameters p/α, all three morphological layers succeed at finding the correct morphological operation (namely, an erosion). Besides, the magnitude of the parameter at convergence also confirms that the operation applied by all layers can be well considered as an erosion (and not simply a pseudo-erosion).

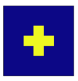
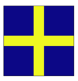
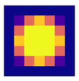
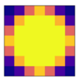
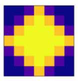
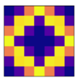
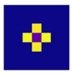
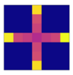
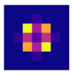
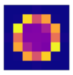
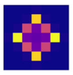
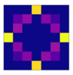
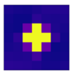
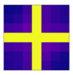
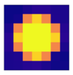
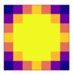
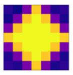
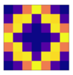
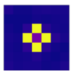
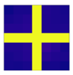
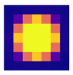
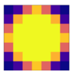
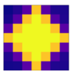
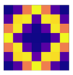
		 *cross3*	 *cross7*	 *disk2*	 *disk3*	 *diamond3*	 *complex*
PConv	w						
	p	-20.86 ± 0.45	-20.35 ± 0.20	-10.52 ± 0.004	-15.64 ± 0.06	-13.21 ± 0.02	-16.45 ± 0.07
	RMSE	$0.82 \pm 7\times10^{-4}$	$1.55 \pm 1\times10^{-3}$	$2.82 \pm 1\times10^{-4}$	$3.77 \pm 6\times10^{-4}$	$3.64 \pm 1\times10^{-4}$	$3.19 \pm 6\times10^{-3}$
	LOSS	$2.4\times10^{-5} \pm 1\times10^{-6}$	$6.2\times10^{-6} \pm 1\times10^{-7}$	$1.3\times10^{-4} \pm 1\times10^{-8}$	$2.6\times10^{-5} \pm 5\times10^{-8}$	$5.2\times10^{-5} \pm 2\times10^{-8}$	$1.2\times10^{-5} \pm 4\times10^{-8}$
	EPOCHS	209 ± 45	169 ± 22	49 ± 3	92 ± 7	66 ± 8	130 ± 18
$\mathscr{L}$*Morph*	w						
	p	-77.09 ± 0.15	-59.58 ± 0.20	-67.78 ± 0.30	-59.46 ± 0.18	-66.25 ± 0.17	-79.36 ± 0.93
	RMSE	$\mathbf{0.44} \pm 3\times10^{-4}$	$0.60 \pm 1\times10^{-2}$	$0.37 \pm 4\times10^{-2}$	$0.30 \pm 7\times10^{-3}$	$0.38 \pm 4\times10^{-2}$	$\mathbf{0.04} \pm 1\times10^{-3}$
	LOSS	$\mathbf{1.1\times10^{-6}} \pm 5\times10^{-9}$	$\mathbf{3.6\times10^{-7}} \pm 4\times10^{-9}$	$1.3\times10^{-6} \pm 6\times10^{-8}$	$\mathbf{3.7\times10^{-7}} \pm 4\times10^{-9}$	$5.8\times10^{-7} \pm 3\times10^{-9}$	$1.1\times10^{-6} \pm 5\times10^{-8}$
	EPOCHS	158 ± 4	164 ± 20	146 ± 16	256 ± 3	231 ± 9	876 ± 139
$\mathscr{S}$*Morph*	w						
	α	-33.95 ± 0.03	-28.52 ± 0.013	-30.61 ± 0.016	-28.10 ± 0.006	-34.38 ± 0.003	-23.79 ± 0.011
	RMSE	$0.78 \pm 2\times10^{-2}$	$\mathbf{0.18} \pm 3\times10^{-2}$	$\mathbf{0.10} \pm 2\times10^{-3}$	$\mathbf{0.04} \pm 2\times10^{-2}$	$\mathbf{0.13} \pm 4\times10^{-2}$	$0.08 \pm 2\times10^{-2}$
	LOSS	$2.1\times10^{-5} \pm 5\times10^{-9}$	$4.0\times10^{-7} \pm 8\times10^{-9}$	$\mathbf{4.6\times10^{-7}} \pm 8\times10^{-9}$	$4.2\times10^{-7} \pm 6\times10^{-9}$	$\mathbf{4.3\times10^{-7}} \pm 3\times10^{-9}$	$\mathbf{9.5\times10^{-7}} \pm 7\times10^{-9}$
	EPOCHS	40 ± 4	45 ± 4	38 ± 8	38 ± 7	42 ± 5	55 ± 9

Fig. 6 Learned filter w, corresponding parameter p/α, RMSE between the learned filter and the target structuring element, MSE training loss at convergence and number of training epochs for $PConv$, $\mathscr{L}$Morph and $\mathscr{S}$Morph layers on an erosion $\ominus$ task. Reported values correspond to the average $\pm$ standard deviation over the five runs. Best (lowest) results are in bold

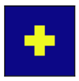
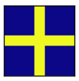
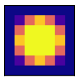
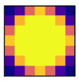
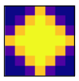
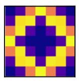
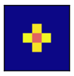
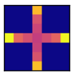
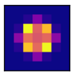
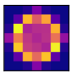
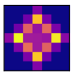
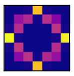
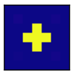
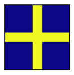
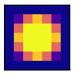
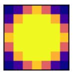
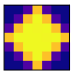
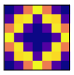
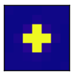
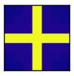
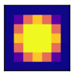
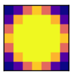
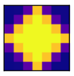
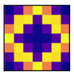
		 *cross3*	 *cross7*	 *disk2*	 *disk3*	 *diamond3*	 *complex*
PConv	w						
	p	19.87 ± 0.23	22.58 ± 0.44	8.27 ± 0.002	9.43 ± 0.002	9.39 ± 0.001	12.57 ± 0.005
	RMSE	$0.41 \pm 2\times10^{-3}$	$1.42 \pm 6\times10^{-4}$	$2.22 \pm 3\times10^{-4}$	$3.05 \pm 3\times10^{-4}$	$2.79 \pm 3\times10^{-4}$	$2.81 \pm 7\times10^{-5}$
	LOSS	$4.8\times10^{-5} \pm 1\times10^{-6}$	$9.0\times10^{-5} \pm 4\times10^{-6}$	$5.1\times10^{-4} \pm 5\times10^{-8}$	$6.3\times10^{-4} \pm 6\times10^{-9}$	$6.8\times10^{-4} \pm 5\times10^{-9}$	$3.3\times10^{-4} \pm 1\times10^{-8}$
	EPOCHS	194 ± 17	222 ± 38	46 ± 8	41 ± 3	38 ± 4	70 ± 7
$\mathscr{L}$*Morph*	w						
	p	94.92 ± 0.18	95.89 ± 0.29	94.16 ± 0.17	94.25 ± 0.96	94.67 ± 0.31	91.27 ± 0.64
	RMSE	$\mathbf{0.02} \pm 9\times10^{-5}$	$\mathbf{0.003} \pm 8\times10^{-6}$	$\mathbf{0.01} \pm 8\times10^{-5}$	$\mathbf{0.05} \pm 1\times10^{-4}$	$\mathbf{0.01} \pm 2\times10^{-4}$	$\mathbf{0.04} \pm 3\times10^{-4}$
	LOSS	$8.4\times10^{-6} \pm 4\times10^{-8}$	$1.1\times10^{-5} \pm 9\times10^{-8}$	$7.6\times10^{-6} \pm 4\times10^{-8}$	$1.2\times10^{-5} \pm 3\times10^{-7}$	$1.1\times10^{-5} \pm 1\times10^{-7}$	$2.1\times10^{-5} \pm 2\times10^{-7}$
	EPOCHS	119 ± 12	159 ± 22	206 ± 11	193 ± 27	206 ± 9	295 ± 35
$\mathscr{S}$*Morph*	w						
	α	41.97 ± 0.016	52.01 ± 0.027	40.75 ± 0.014	50.06 ± 0.044	49.47 ± 0.04	32.79 ± 0.021
	RMSE	$0.1 \pm 3\times10^{-3}$	$0.14 \pm 2\times10^{-3}$	$0.09 \pm 4\times10^{-4}$	$0.08 \pm 1\times10^{-3}$	$0.09 \pm 2\times10^{-3}$	$0.05 \pm 1\times10^{-2}$
	LOSS	$\mathbf{8.5\times10^{-7}} \pm 3\times10^{-8}$	$\mathbf{1.5\times10^{-6}} \pm 2\times10^{-8}$	$\mathbf{1.2\times10^{-6}} \pm 1\times10^{-8}$	$\mathbf{1.8\times10^{-6}} \pm 4\times10^{-8}$	$\mathbf{1.5\times10^{-6}} \pm 9\times10^{-9}$	$\mathbf{1.6\times10^{-6}} \pm 2\times10^{-8}$
	EPOCHS	39 ± 8	39 ± 9	44 ± 9	45 ± 4	44 ± 4	49 ± 7

Fig. 7 Learned filter w, corresponding parameter p/α, RMSE between the learned filter and the target structuring element, MSE training loss at convergence and number of training epochs for $PConv$, $\mathscr{L}$Morph and $\mathscr{S}$Morph layers on a dilation $\oplus$ task. Reported values correspond to the average $\pm$ standard deviation over the five runs. Best (lowest) results are in bold

However, except for *cross3*, the RMSE values for all other target structuring elements are notably higher for the $PConv$ than for both $\mathscr{L}$Morph and $\mathscr{S}$Morph. This is indeed confirmed when looking at the shape of the learned filters for all layers: It is clear that the $PConv$ layer suffers from the hollow effect mentioned in Sect. 3.4, while both $\mathscr{L}$Morph and $\mathscr{S}$Morph layers accurately retrieve the target structuring element. Nevertheless, the small standard deviations for both the parameter and the RMSE values clearly indicate that all five training runs have converged toward the same solution for all three layers and all six target structuring elements. In terms of MSE training loss, both $\mathscr{L}$Morph and $\mathscr{S}$Morph layers outperform the $PConv$ for all scenarios. Regarding the required number of epochs to reach convergence, the $\mathscr{S}$Morph layer converges faster and with more consistency than $PConv$ and $\mathscr{L}$Morph.

Figure 7 presents similar results for the dilation $\oplus$ scenarios and allows to draw comparable conclusions. The sign and magnitude of the learned parameters p/α confirm that all three layers correctly identified the dilation operation and replicated it. In terms of learned filters, $PConv$ again suffers from the hollow effect while both $\mathscr{L}$Morph and $\mathscr{S}$Morph perfectly recover the shape of the target structuring element, with the former performing slightly better than the latter according to the corresponding RMSE values. In any case, the standard deviation of the reported metrics again confirm the convergence of all three networks toward the same solution. The MSE training loss at convergence is consistently lower for the $\mathscr{S}$Morph layer (by an order of magnitude of 10 with respect to $\mathscr{L}$Morph and up to 10^2 with respect to $PConv$). Moreover, $\mathscr{S}$Morph again requires a lower number of epochs to reach convergence.

5.2 Learning Grayscale Openings and Closing

5.2.1 Experimental Protocol

We now investigate the capacity of the proposed layers to learn more advanced morphological operations such as the opening $\circ$ (composition of an erosion followed by a dilation with the same structuring element) or the closing $\bullet$ (composition of a dilation followed by an erosion, again with the same structuring element). The network architecture is thus adapted as displayed in Fig. 8. It is now composed of two chained morphological layers: The output of the first layer is fed as input to the second layer. Note that we do not consider hybrid networks here where the two morphological layers are not defined by the same operation $PConv/\mathscr{L}$Morph/$\mathscr{S}$Morph. Besides, it is worth mentioning that the two filters evolve independently from each other.

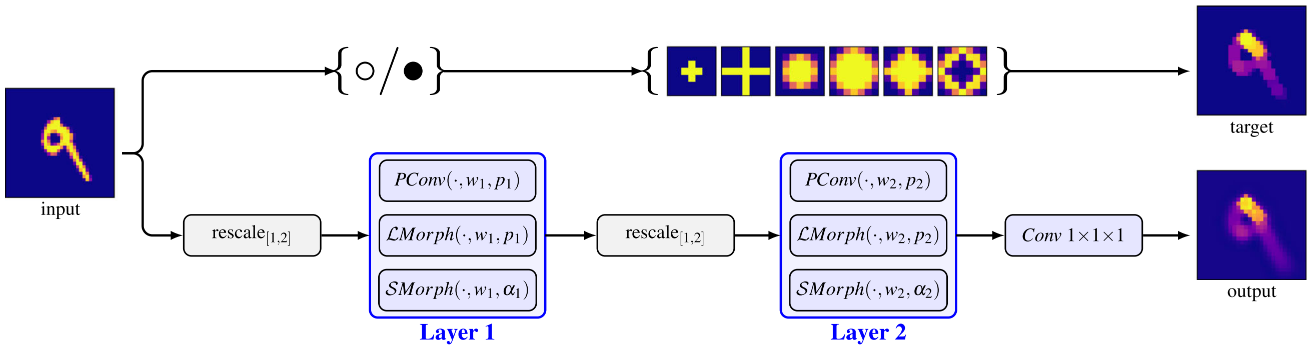


Fig. 8 Network architecture used for the opening/closing scenarios. Blue blocks are trainable units. A scenario is defined as the choice of $\circ$ or $\bullet$ and one of the six target structuring elements in the upper path, and the choice of $PConv$, $\mathscr{L}$Morph or $\mathscr{S}$Morph for both consecutive layers in the lower path (Color figure online)

Nevertheless, the two morphological layers are expected to learn filters having exactly the same shape, with parameter p or α of opposite signs, once training has converged. For the same reasons as the erosion/dilation scenarios described in Sect. 5.1.1, rescaling blocks are again added before both morphological layers, and the final scale/bias $Conv$ $1 \times 1 \times 1$ is placed at the output of the network. All other parameters (batch size, loss, optimizer and learning rate, convergence criterion and filters initialization) remain the same as described in Sect. 5.1.1. We again perform five runs per scenario and report the average (and standard deviation) of the parameter p/α as well as the RMSE at convergence for both layers, along with the MSE training loss at convergence and the number of training epochs.

5.2.2 Obtained Results

Figure 9 presents the qualitative and quantitative results obtained for the two layers of $PConv$, $\mathscr{L}$Morph and $\mathscr{S}$Morph morphological networks in opening $\circ$ scenarios. The $PConv$ network always succeeds at learning the right morphological operation since the first (resp. second) layer always converges to $p_1 < 0$ (resp. $p_2 > 0$). However, except for *cross3* and *cross7*, $|p| < 10$, indicating that the layer is applying pseudo-dilation or pseudo-erosion only. In addition, the learned structuring element suffers again from the hollow effect. A pathological issue arises for $\mathscr{L}$Morph and $\mathscr{S}$Morph with the *cross3* structuring element: $\mathscr{L}$Morph actually performs a closing operation instead of an opening (since the first layer converges toward $p_1 > 0$ and the second one toward $p_2 < 0$), while $\mathscr{S}$Morph achieves two consecutive pseudo-erosions in the network ($\alpha_1 \lesssim 0$ and $\alpha_2 < 0$). $\mathscr{L}$Morph also struggles with *disk2* since the second layer also converges toward an erosion instead of a dilation. Apart from those obvious failure cases, a visual analysis of the learned filters yields another observation: While the second network layer converges toward a filter whose shape is very close to the target structuring element, it might not be the case for the first layer that appears slightly flawed (this is noticeably the case for *cross7*, *diamond3* and *complex* for $\mathscr{L}$Morph). These convergence issues are further detailed and analyzed in Sect. 5.2.3. Apart from those edge cases, all networks again appear very stable in terms of convergence results, judging by the low standard deviation values for all reported quantitative metrics, with $\mathscr{S}$Morph almost consistently outperforming $PConv$ and $\mathscr{L}$Morph in terms of RMSE for both layers, MSE loss at convergence and number of training epochs.

Similar results for closing $\bullet$ scenarios are presented in Fig. 10. Several convergence issues arise for all three investigated layers for the scenarios whose reported metrics appear in red. $\mathscr{S}$Morph again converges toward two consecutive erosions on *cross3* and *disk2*, with the shape of the learned filters being nowhere near the expected target structuring elements. For $PConv$/*cross7* and $\mathscr{L}$Morph/*cross3*, the high standard deviations for the parameters values come from a divergent behavior: In both cases, the network converged three times out of the five runs toward a solution and the two remaining times toward another solution. These divergent solutions, displayed in Fig. 11, are the only two cases among all investigated scenarios where the networks do not repeatedly converge toward a stable solution. $\mathscr{L}$Morph/*disk2* and $\mathscr{L}$Morph/*complex* scenarios also appear to have some relatively high standard deviation values for the layer parameters, but the associated RMSE for the learned filters (coupled with a visual examination of those learned filters) confirms that the network still converged toward a stable solution. For those precise cases, the observed variations in terms of final parameters values have a negligible effect on the output of the $\mathscr{L}$Morph layers since the parameters absolute magnitudes ensure that the applied operations are very good approximations of erosions or dilations. Thus, those variations have little effect on the networks learning behaviors and the shapes of the learned filters. The $\mathscr{L}$Morph/*complex* scenario also features some important variations in terms of number of epochs to reach convergence, although all five runs did converge to a solution in less than the maximal number of epochs

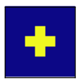
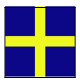
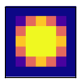
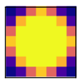
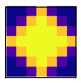
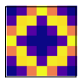
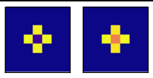
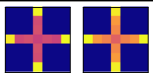
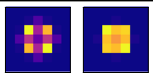
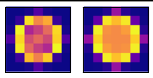
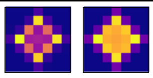
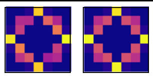
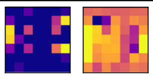
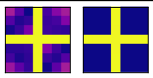
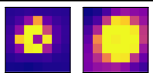
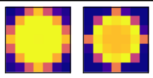
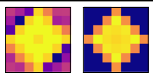
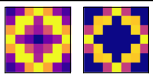
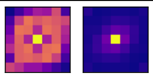
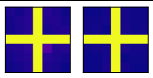
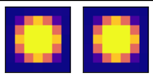
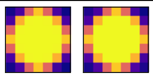
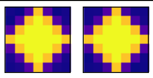
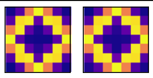
		cross3	*cross7*	*disk2*	*disk3*	*diamond3*	*complex*
							
PConv	w_1/w_2						
	p_1	-18.65 ± 0.29	-14.17 ± 0.32	-8.13 ± 0.04	-8.59 ± 0.06	-8.05 ± 0.01	-9.26 ± 0.15
	p_2	22.11 ± 0.35	21.21 ± 0.47	17.72 ± 0.15	7.03 ± 0.10	7.49 ± 0.01	8.87 ± 0.10
	RMSE$_1$	$\mathbf{0.91} \pm 4\times10^{-4}$	$1.45 \pm 1\times10^{-3}$	$2.69 \pm 7\times10^{-3}$	$3.13 \pm 3\times10^{-2}$	$3.17 \pm 5\times10^{-3}$	$2.53 \pm 3\times10^{-2}$
	RMSE$_2$	$\mathbf{0.32} \pm 1\times10^{-3}$	$0.87 \pm 2\times10^{-3}$	$2.49 \pm 2\times10^{-3}$	$2.51 \pm 1\times10^{-2}$	$2.43 \pm 4\times10^{-3}$	$2.49 \pm 1\times10^{-2}$
	LOSS	$\mathbf{8.9\times10^{-5}} \pm 3\times10^{-6}$	$7.5\times10^{-5} \pm 1\times10^{-6}$	$4.9\times10^{-4} \pm 2\times10^{-6}$	$3.9\times10^{-4} \pm 2\times10^{-5}$	$4.4\times10^{-4} \pm 4\times10^{-6}$	$2.2\times10^{-4} \pm 6\times10^{-6}$
	EPOCHS	167 ± 43	114 ± 8	61 ± 8	47 ± 13	50 ± 5	27 ± 12
$\mathscr{L}$*Morph*	w_1/w_2						
	p_1	$\mathbf{7.39} \pm 0.14$	-22.68 ± 0.57	-10.74 ± 0.14	-10.88 ± 0.11	-12.82 ± 0.08	-9.54 ± 0.03
	p_2	$\mathbf{-12.89} \pm 0.06$	66.71 ± 0.91	$\mathbf{-1.07} \pm 0.003$	8.18 ± 0.08	12.55 ± 0.07	9.79 ± 0.10
	RMSE$_1$	$3.17 \pm 2\times10^{-3}$	$1.02 \pm 6\times10^{-2}$	$1.99 \pm 7\times10^{-4}$	$0.17 \pm 8\times10^{-3}$	$1.92 \pm 5\times10^{-2}$	$1.15 \pm 8\times10^{-2}$
	RMSE$_2$	$4.72 \pm 8\times10^{-3}$	$\mathbf{0.02} \pm 3\times10^{-4}$	$1.61 \pm 8\times10^{-3}$	$1.80 \pm 4\times10^{-2}$	$0.70 \pm 4\times10^{-3}$	$0.85 \pm 8\times10^{-3}$
	LOSS	$8.7\times10^{-3} \pm 9\times10^{-6}$	$4.3\times10^{-5} \pm 2\times10^{-6}$	$2.0\times10^{-3} \pm 1\times10^{-5}$	$3.8\times10^{-4} \pm 1\times10^{-5}$	$3.2\times10^{-4} \pm 4\times10^{-6}$	$2.3\times10^{-4} \pm 2\times10^{-6}$
	EPOCHS	51 ± 6	111 ± 8	165 ± 42	21 ± 5	46 ± 6	44 ± 4
$\mathscr{S}$*Morph*	w_1/w_2						
	α_1	$\mathbf{-0.36} \pm 0.0009$	-24.71 ± 0.16	-34.71 ± 0.01	-29.15 ± 0.07	-38.36 ± 0.08	-18.69 ± 0.02
	α_2	$\mathbf{-3.72} \pm 0.05$	40.62 ± 0.01	41.38 ± 0.008	45.41 ± 0.02	47.13 ± 0.02	16.08 ± 0.02
	RMSE$_1$	$3.30 \pm 1\times10^{-2}$	$\mathbf{0.22} \pm 1\times10^{-2}$	$\mathbf{0.06} \pm 8\times10^{-3}$	$\mathbf{0.04} \pm 8\times10^{-3}$	$\mathbf{0.06} \pm 5\times10^{-2}$	$\mathbf{0.11} \pm 3\times10^{-2}$
	RMSE$_2$	$1.63 \pm 5\times10^{-3}$	$0.06 \pm 3\times10^{-2}$	$\mathbf{0.03} \pm 1\times10^{-2}$	$\mathbf{0.05} \pm 1\times10^{-2}$	$\mathbf{0.06} \pm 2\times10^{-2}$	$\mathbf{0.11} \pm 3\times10^{-2}$
	LOSS	$4.8\times10^{-3} \pm 4\times10^{-6}$	$\mathbf{4.6\times10^{-7}} \pm 9\times10^{-8}$	$\mathbf{5.5\times10^{-7}} \pm 4\times10^{-8}$	$\mathbf{6.2\times10^{-7}} \pm 8\times10^{-8}$	$\mathbf{5.5\times10^{-7}} \pm 4\times10^{-8}$	$\mathbf{1.8\times10^{-6}} \pm 4\times10^{-8}$
	EPOCHS	22 ± 3	41 ± 5	35 ± 3	42 ± 4	42 ± 6	42 ± 4

Fig. 9 Learned filters w_i, corresponding parameter p_i/α_i and RMSE$_i$ between the learned filter and the target structuring element for both layers ($i \in \{1, 2\}$), MSE training loss at convergence and number of training epochs for $PConv$, $\mathscr{L}$Morph and $\mathscr{S}$Morph layers on an opening $\circ$ task. Best (lowest) results are in bold. Abnormal results are in red (Color figure online)

allowed during training (set to 1000). Except for failure cases on *cross3* and *disk2* previously discussed, $\mathscr{S}$Morph again consistently outperforms $PConv$ and $\mathscr{L}$Morph in terms of shape and RMSE of learned filters, MSE loss at convergence and number of epochs to reach convergence.

5.2.3 Convergence Issues

The analysis in Sect. 5.2.2 of the obtained results for opening and closing scenarios raised several issues in terms of convergence behavior for $\mathscr{L}$Morph and $\mathscr{S}$Morph layers, particularly for *cross3* and *disk2* target structuring elements. In Kirszenberg et al. [11], we conjectured that this edge case may be a consequence of the final scale/biais $Conv$ $1 \times 1 \times 1$ layer, over-compensating for the gain or loss of average pixel intensities and impeding a correct error flow while back-propagating the error during the learning phase. This explanation has, however, two weaknesses: These convergence/learning issues only affect $\mathscr{L}$Morph and $\mathscr{S}$Morph and seemingly not the $PConv$ layer, although $\mathscr{L}$Morph and $PConv$ both rely on the CHM asymptotic behavior, and they only concern *cross3* and *disk2* target structuring elements, although *cross7* and *disk3* have the same shape (despite not the same size within the 7×7 spatial support). This latter observation led us to conduct a more in-depth analysis of the gradual convergence of the layer filters along the epochs during the training phase. Some representative results of this analysis are shown in Fig. 12, which displays the filter weights of two consecutive $\mathscr{S}$Morph layers at initialization, 1%, 2%, 3%, 5%, 7%, 10%, 20%, 50% and 100% of total number of training epochs in a closing $\bullet$ scenario, with the target structuring elements being *diamond3*, *cross7* and *disk2*. For the first two target structuring elements, the convergence succeeded (as shown in Fig. 10 for those particular scenarios, as well as the corresponding values of α_1 and α_2 during convergence displayed in Fig. 12, but it failed for the latter one (the $\mathscr{S}$Morph network converging toward two successive pseudo-erosions with $\alpha_1 = -0.282$ and $\alpha_2 = -6.481$). The way the filter weights update throughout the learning process for *diamond3* and *cross7* share several similarities: The convergence toward the correct shape is faster for the second layer than the first one (since its weights are updated before those of the first layer during back-propagation), so is the convergence

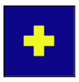
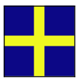
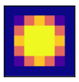
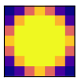
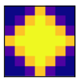
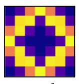
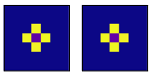
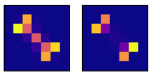
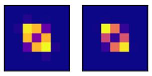
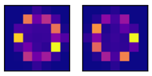
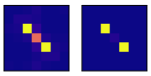
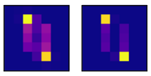
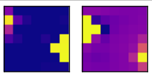
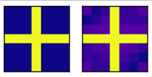
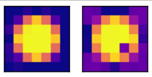
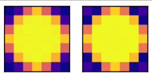
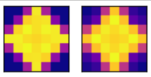
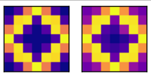
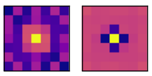
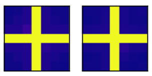
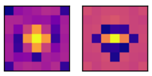
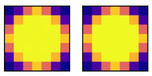
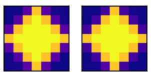
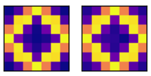
		 *cross3*	 *cross7*	 *disk2*	 *disk3*	 *diamond3*	 *complex*
PConv	w_1/w_2						
	p_1	16.36 ± 0.12	5.99 ± 0.5	5.78 ± 0.004	7.85 ± 0.21	3.43 ± 0.002	5.84 ± 0.003
	p_2	-20.95 ± 0.14	-8.07 ± 1.65	-8.54 ± 0.002	-10.22 ± 0.36	-9.73 ± 0.04	7.64 ± 0.002
	RMSE$_1$	$\mathbf{0.82} \pm 6\times10^{-4}$	$3.82 \pm 1\times10^{-1}$	$2.82 \pm 2\times10^{-4}$	$4.47 \pm 3\times10^{-3}$	$4.34 \pm 5\times10^{-5}$	$4.16 \pm 3\times10^{-5}$
	RMSE$_2$	$\mathbf{0.81} \pm 1\times10^{-3}$	$3.87 \pm 9\times10^{-2}$	$3.07 \pm 3\times10^{-4}$	$4.63 \pm 2\times10^{-1}$	$4.51 \pm 6\times10^{-5}$	$4.13 \pm 4\times10^{-6}$
	LOSS	$\mathbf{8.9\times10^{-5}} \pm 1\times10^{-6}$	$6.5\times10^{-3} \pm 7\times10^{-4}$	$1.7\times10^{-3} \pm 6\times10^{-8}$	$1.0\times10^{-3} \pm 6\times10^{-5}$	$5.4\times10^{-3} \pm 5\times10^{-7}$	$5.0\times10^{-3} \pm 1\times10^{-7}$
	EPOCHS	194 ± 21	56 ± 8	50 ± 4	60 ± 7	35 ± 8	49 ± 6
$\mathscr{L}$*Morph*	w_1/w_2						
	p_1	46.55 ± 29.7	93.67 ± 0.43	73.33 ± 2.49	89.50 ± 0.55	12.19 ± 0.009	72.29 ± 1.84
	p_2	-41.33 ± 22.2	-92.077 ± 0.38	-82.37 ± 1.84	-86.29 ± 0.26	-12.24 ± 0.02	-84.01 ± 1.32
	RMSE$_1$	$3.17 \pm 2\times10^{-1}$	$\mathbf{0.01} \pm 7\times10^{-5}$	$\mathbf{0.14} \pm 3\times10^{-3}$	$\mathbf{0.07} \pm 3\times10^{-4}$	$0.75 \pm 7\times10^{-4}$	$\mathbf{0.05} \pm 1\times10^{-3}$
	RMSE$_2$	$3.29 \pm 9\times10^{-1}$	$0.63 \pm 2\times10^{-3}$	$\mathbf{1.16} \pm 2\times10^{-2}$	$0.40 \pm 1\times10^{-3}$	$0.95 \pm ''\times10^{-4}$	$0.75 \pm 5\times10^{-2}$
	LOSS	$3.7\times10^{-3} \pm 6\times10^{-4}$	$1.3\times10^{-5} \pm 2\times10^{-7}$	$\mathbf{2.0\times10^{-4}} \pm 1\times10^{-6}$	$1.7\times10^{-5} \pm 2\times10^{-7}$	$6.2\times10^{-4} \pm 1\times10^{-7}$	$1.5\times10^{-5} \pm 8\times10^{-7}$
	EPOCHS	72 ± 3	300 ± 22	157 ± 12	123 ± 17	61 ± 8	329 ± 94
$\mathscr{S}$*Morph*	w_1/w_2						
	α_1	-0.25 ± 0.0003	42.47 ± 0.02	-0.28 ± 0.002	41.42 ± 0.26	43.02 ± 0.04	23.82 ± 0.04
	α_2	-3.44 ± 0.01	-41.60 ± 0.02	-6.48 ± 0.02	-39.96 ± 0.06	-42.94 ± 0.03	-29.73 ± 0.04
	RMSE$_1$	$1.99 \pm 1\times10^{-3}$	$0.16 \pm 1\times10^{-2}$	$2.77 \pm 2\times10^{-3}$	$0.09 \pm 1\times10^{-3}$	$\mathbf{0.09} \pm 3\times10^{-3}$	$0.15 \pm 4\times10^{-2}$
	RMSE$_2$	$3.77 \pm 3\times10^{-3}$	$\mathbf{0.09} \pm 8\times10^{-3}$	$3.81 \pm 9\times10^{-3}$	$\mathbf{0.01} \pm 3\times10^{-3}$	$\mathbf{0.10} \pm 5\times10^{-3}$	$\mathbf{0.11} \pm 6\times10^{-2}$
	LOSS	$1.9\times10^{-3} \pm 6\times10^{-7}$	$\mathbf{8.0\times10^{-7}} \pm 3\times10^{-8}$	$4.5\times10^{-3} \pm 4\times10^{-6}$	$\mathbf{1.1\times10^{-6}} \pm 3\times10^{-8}$	$\mathbf{8.7\times10^{-7}} \pm 3\times10^{-8}$	$\mathbf{1.2\times10^{-6}} \pm 5\times10^{-8}$
	EPOCHS	18 ± 2	39 ± 6	15 ± 1	44 ± 4	39 ± 4	47 ± 3

Fig. 10 Learned filters w_i, corresponding parameter p_i/α_i and RMSE$_i$ between the learned filter and the target structuring element for both layers ($i \in \{1, 2\}$), MSE training loss at convergence and number of training epochs for $PConv$, $\mathscr{L}$Morph and $\mathscr{S}$Morph layers on a closing • task. Best (lowest) results are in bold. Abnormal results are in red (Color figure online)

of the parameter α toward a range of values ensuring that the operation applied by the layer well approximates a dilation or an erosion (and no longer a pseudo-dilation/erosion), and the weights updating scheme seem to start on the filter edges and then spread toward the center. This last observation may be the reason why the convergence failed for *disk2* since the extent of this structuring element is smaller than its spatial support, thus not providing any anchor point on the edges for the weights to update toward the center of the filter, also explaining the same failure cases for *cross3*.

In order to evaluate the soundness of this previous hypothesis, we reduced the size of the spatial support of *cross3* and *disk2* from 7×7 to 3×3 and 5×5, respectively (for the extent and the spatial support of both structuring elements to be the same), and increase this of *disk3* from 7×7 to 9×9, 11×11 and 13×13 (to make the extent of *disk3* smaller than its spatial support). Figure 13 displays the filters that have been learned by $\mathscr{L}$Morph and $\mathscr{S}$Morph for those updated target structuring elements on opening ∘ and closing • scenarios. For 3×3 *cross3* and 5×5 *disk2*,

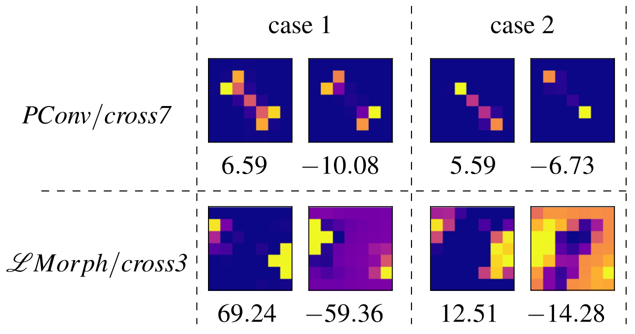


Fig. 11 Examples of network divergent behavior for $PConv/cross7$/• and $\mathscr{L}$Morph/*cross3*/• scenarios, with average parameter values at convergence

$\mathscr{L}$Morph performs well on closing • and $\mathscr{S}$Morph on opening ∘, but both morphological layers fail on the opposite operation. For *disk3* in an increased spatial support, $\mathscr{S}$Morph achieves much better results on the opening ∘ than $\mathscr{L}$Morph. Both layers, however, struggle with the closing • operation.

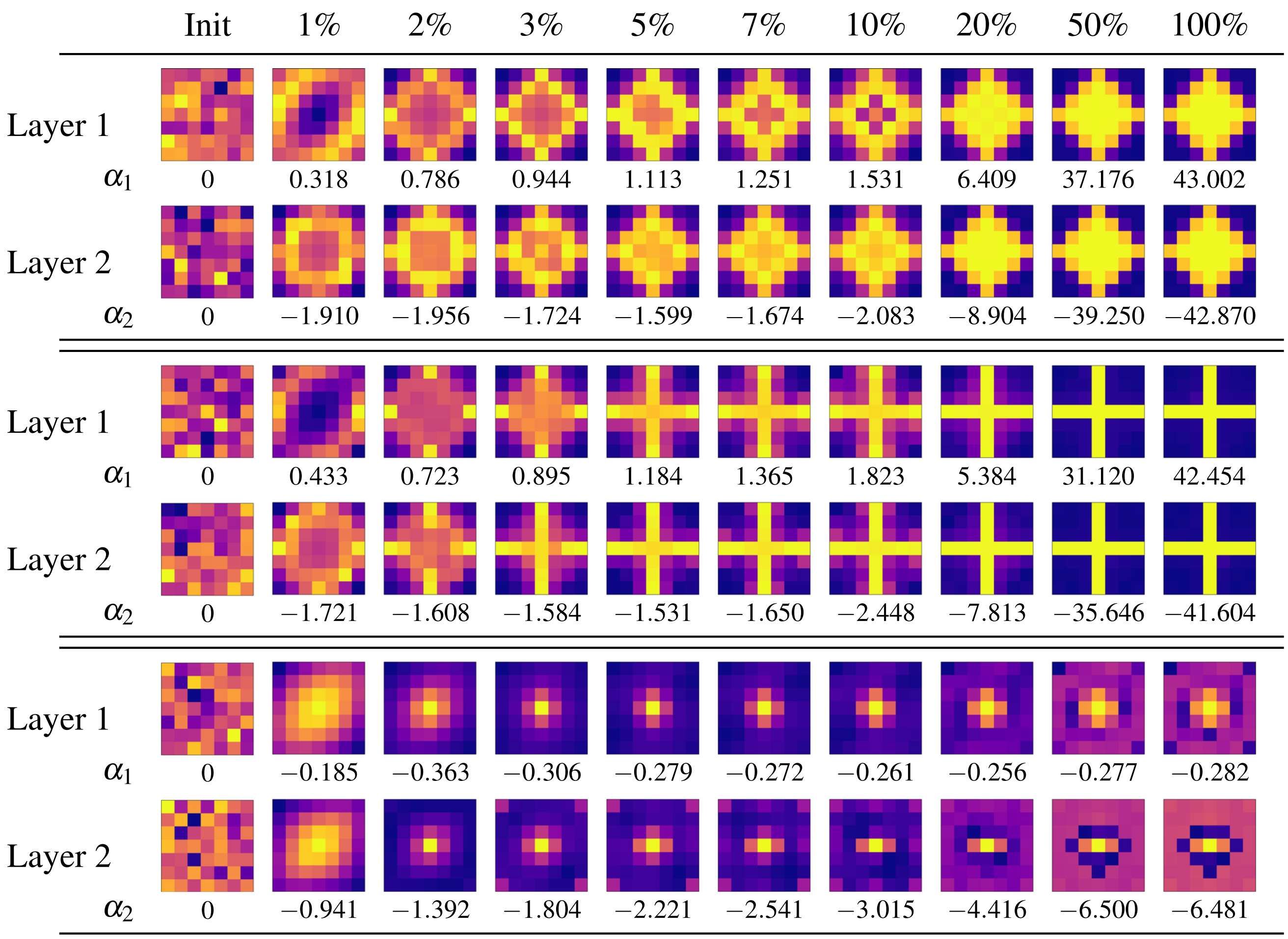


Fig. 12 Convergence of two consecutive $\mathscr{S}$Morph layers in a closing $\bullet$ scenario, with corresponding values of α_1 and α_2. Target structuring elements are *diamond3* (top row), *cross7* (middle row) and *disk2* (bottom row). Layers are shown at initialization, 1%, 2%, 3%, 5%, 7%, 10%, 20%, 50% and 100% of total number of training epochs

As a conclusion, the extent of a structuring element relatively to the size of its spatial support seems to be connected to the ability of $\mathscr{L}$Morph and $\mathscr{S}$Morph morphological layers to properly learn this structuring element for opening $\circ$ and closing $\bullet$ scenarios, but it is insufficient on its own to explain all presented failure cases. Convergence to local minima might also be a plausible explanation since the identification of a structuring element based on original and transformed images only is an under-determined task as several solutions might exist. In particular, symmetrically shifted filters as $\mathscr{S}$Morph/13×13 *disk3*/$\circ$ compensate each other out and are as valid as a solution as two centered filters (which were expected on this case). Integrating any prior information (symmetry, sparsity, spatial extent) on the sought structuring element within the optimized training loss function could be a potential solution, but this prior information might be hard to have access to on practical use cases.

6 Learning Binary Morphological Operators

6.1 Experimental Protocol

In this section, we evaluate the capacity of the proposed morphological layers to operate with binary morphological operations and binary structuring elements. For that purpose, we slightly depart from the framework exposed in Sect. 5: While the overall idea remains the same (identifying a morphological operation and its associated structuring element based on couples of original and transformed images), the input MNIST images and the target structuring elements are now binary. The former correspond to the original 60,000 MNIST images, used in the previous section, thresholded to $\frac{1}{2}$, while the latter are depicted in Fig. 14. The output target images are thus also binary, obtained using Eq. (1) for the erosion $\ominus$, Eq. (2) for the dilation $\oplus$, and their composition for the opening $\circ$ and closing $\bullet$.

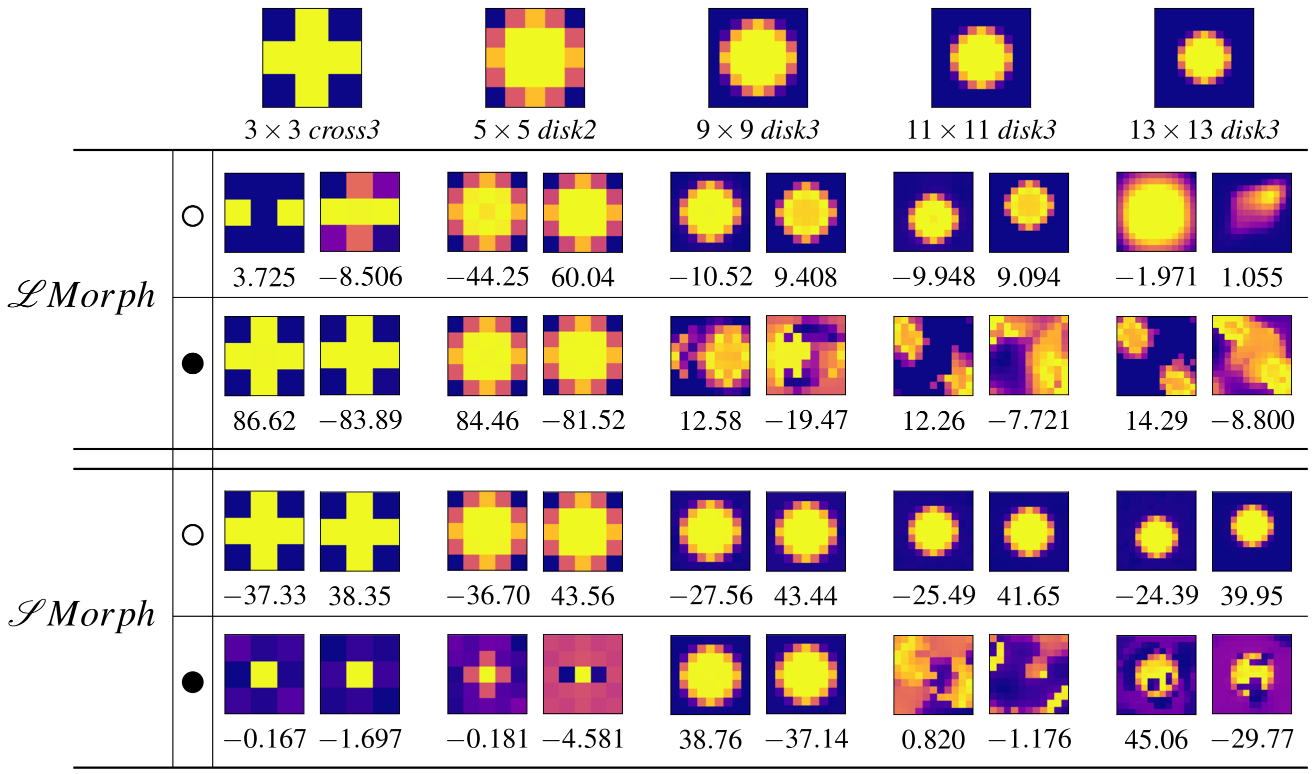


Fig. 13 Learned filters for opening ∘ and closing • operations for $\mathscr{L}$Morph and $\mathscr{S}$Morph for *cross3* in a 3 × 3 spatial support, *disk2* in a 5 × 5 spatial support, as well as *disk3* in 9 × 9, 11 × 11 and 13 × 13 spatial supports

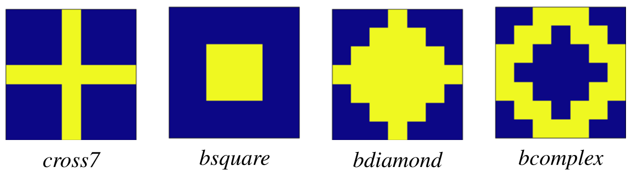


Fig. 14 7 × 7 target binary structuring elements. Yellow (resp. blue) corresponds to Boolean TRUE (resp. FALSE) (Color figure online)

However, both $PConv$ and $\mathscr{L}$Morph are not compatible with the processing of binary images since they require their input to be rescaled in the range [1, 2]. Therefore, working in this binary framework restricts to the use of the $\mathscr{S}$Morph morphological layer only. Nevertheless, the use of the rescaling block is no longer mandatory: It is thus removed from the network architecture. The scale/biais $Conv1 \times 1 \times 1$ is also discarded for the same reason. This results in very simple network architectures, merely composed of a single $\mathscr{S}$Morph layer for erosion and dilation scenarios, and comprising two consecutive $\mathscr{S}$Morph layers for opening and closing scenarios. The other network parameters (batch size, learning loss, optimizer and learning rate, convergence criterion and filter initialization) are the same as those described in Sect. 5. As the results presented in Sect. 5 for $\mathscr{S}$Morph showed to be very stable in terms of repeatability among several runs, we this time conduct a single run per investigated scenario. Therefore, the presented results should only be considered as a proof of concept that the $\mathscr{S}$Morph layer is able to operate on a binary morphological framework.

6.2 Expected Learned Filters

Even though the target structuring elements are binary (made of Boolean TRUE/FALSE pixel values), the grayscale formalism for non-flat structuring element described in Sect. 3.1 encompasses the use of binary structuring elements. If B stands for the spatial position of TRUE pixels in the binary structuring element, then this latter is equivalent to the grayscale structuring function described by Eq. (5), recalled as follows:

$$b : x \mapsto \begin{cases} 0 & \text{if } x \in B \\ -\infty & \text{otherwise} \end{cases}. \qquad (5)$$

Thus, the filter learned by the $\mathscr{S}$Morph layer should have a similar structure, namely weights close to 0 at the position of target TRUE pixels, and large negative weights for binary FALSE pixels. However, the input image values are in practice bounded ($f(x) \in \{0, 1\}$ since the input images f are binary).

Therefore, the structuring function (5) is equivalent to

$$b : x \mapsto \begin{cases} 0 & \text{if } x \in B \\ \leq -1 & \text{otherwise} \end{cases}. \qquad (20)$$

Moreover, because $\mathscr{S}$Morph$(f, w, \alpha < 0)(x) \approx (f \ominus -w)(x)$ (see Eq. (19)), we finally expect the $\mathscr{S}$Morph filter to learn

$$w(x) = \begin{cases} = 0 & \text{if } x \in B \\ \leq -1 & \text{otherwise} \end{cases} \quad \text{for a dilation} \qquad (21)$$

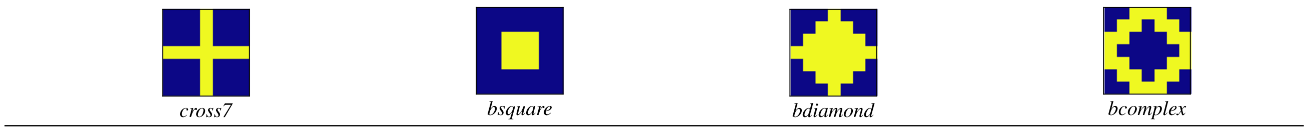
$$w(x) = \begin{cases} = 0 & \text{if } x \in B \\ \geq 1 & \text{otherwise} \end{cases} \quad \text{for an erosion.} \qquad (22)$$

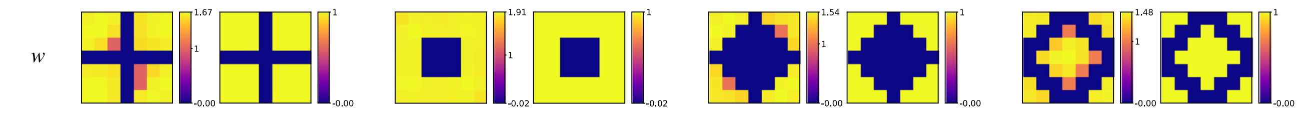
As a consequence, the Boolean TRUE/FALSE values of the target binary structuring element displayed in Fig. 14 are converted to $0/-1$ for the dilation (resp. $0/1$ for the erosion) and the learned filter weights are clipped to -1 if $\alpha > 0$ (resp. $+1$ if $\alpha < 0$) before computation of the RMSE between the learned filter and the target structuring element.

6.3 Obtained Results

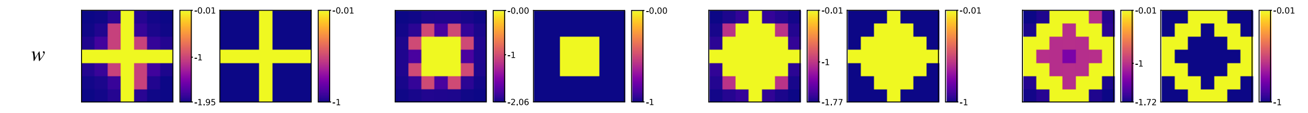
Figure 15 shows the obtained results by a single $\mathscr{S}$Morph layer on binary erosion and dilation scenarios. In any case, it can be seen from the sign and magnitude of α that the layer converged toward the correct operation and performs an accurate approximation of it. Furthermore, it took approximately 30 epochs for the layer to reach convergence, which is even faster than the reported learning performances in the grayscale framework. Last but not least, it was perfectly able to recover the shape of all target binary structuring elements, as expected by Eqs. (21) and (22): All TRUE pixels correspond to weights close to 0 in the learned filters, while all FALSE pixels indeed appear to be greater than 1 for the erosion and lower than -1 for the dilation. This can be qualitatively appraised with the clipped versions of the learned filters, and it is quantitatively confirmed by the low RMSE values.

Figure 16 presents the qualitative and quantitative results obtained by a network composed of two consecutive $\mathscr{S}$Morph layers for binary opening $\circ$ and binary closing $\bullet$ scenarios. For the opening case, the network failed on *bsquare* as it converged toward a closing operation ($\alpha_1 > 0$ and $\alpha_2 < 0$) and the shape and values of the learned filters are considerably different from the expected ones. For the three other target structuring elements, however, the obtained results are excellent, as both layers perfectly recover the shape of the target (the second layer performing even better than the first, replicating the behavior discussed in Sects. 5.2.2 and 5.2.3). Results, however, degrade for the closing: Although the network converged toward a pseudo-closing for all four target structuring elements (the first layer only achieves pseudo-dilation and the second layer performs pseudo-erosion), the learned filters are only satisfactory for *bsquare*. While those results are rather intriguing (in particular, why *bsquare* is the only failure case for the opening but the only success case for the closing), an in-depth analysis of the convergence behavior of the layers, similar to what is conducted in Sect. 5.2.3, would be necessary to further investigate those failure cases.



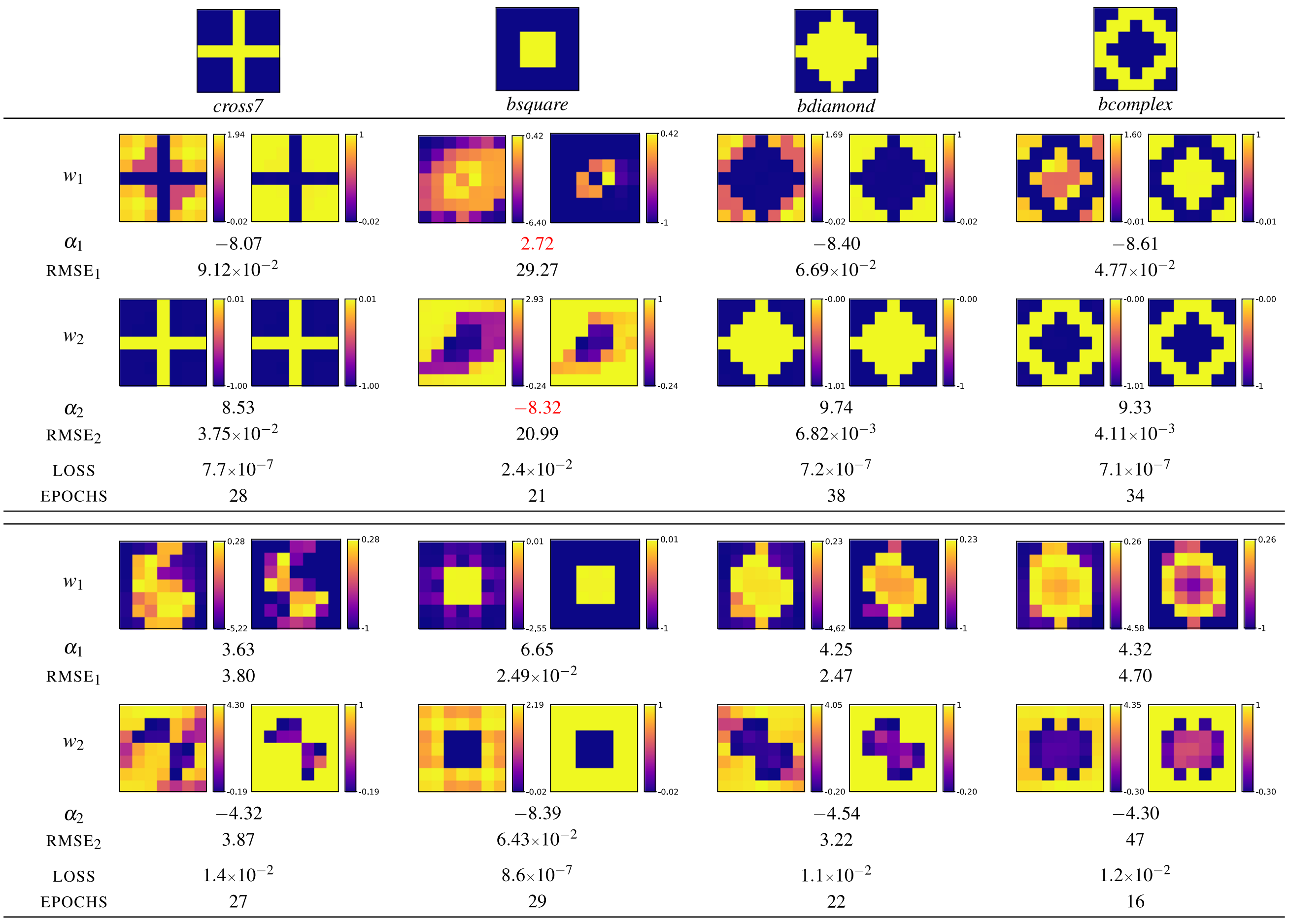


	cross7	bsquare	bdiamond	bcomplex
α	-10.12	-9.80	-10.59	-10.43
RMSE	1.75×10^{-2}	4.65×10^{-2}	1.72×10^{-2}	1.08×10^{-2}
LOSS	2.9×10^{-7}	3.1×10^{-7}	2.9×10^{-7}	2.9×10^{-7}
EPOCHS	31	31	26	32



	cross7	bsquare	bdiamond	bcomplex
α	9.10	8.88	9.96	10.11
RMSE	3.96×10^{-2}	1.78×10^{-2}	6.57×10^{-2}	5.45×10^{-2}
LOSS	4.5×10^{-7}	4.0×10^{-7}	4.6×10^{-7}	4.9×10^{-7}
EPOCHS	34	38	48	31

Fig. 15 Learned filter w and its clipped version with associated color bars, and quantitative metrics for a $\mathscr{S}$Morph layer for binary erosion $\ominus$ (first row) and binary dilation $\oplus$ (second row) scenarios



	cross7	*bsquare*	*bdiamond*	*bcomplex*
w_1				
α_1	-8.07	2.72	-8.40	-8.61
RMSE$_1$	9.12×10^{-2}	29.27	6.69×10^{-2}	4.77×10^{-2}
w_2				
α_2	8.53	-8.32	9.74	9.33
RMSE$_2$	3.75×10^{-2}	20.99	6.82×10^{-3}	4.11×10^{-3}
LOSS	7.7×10^{-7}	2.4×10^{-2}	7.2×10^{-7}	7.1×10^{-7}
EPOCHS	28	21	38	34
w_1				
α_1	3.63	6.65	4.25	4.32
RMSE$_1$	3.80	2.49×10^{-2}	2.47	4.70
w_2				
α_2	-4.32	-8.39	-4.54	-4.30
RMSE$_2$	3.87	6.43×10^{-2}	3.22	47
LOSS	1.4×10^{-2}	8.6×10^{-7}	1.1×10^{-2}	1.2×10^{-2}
EPOCHS	27	29	22	16

Fig. 16 Learned filter $w_{i\in\{1,2\}}$ and its clipped version, and quantitative metrics for two $\mathscr{S}$Morph layers for binary opening $\circ$ (first row) and binary closing $\bullet$ (second row) scenarios. Abnormal results are in red (Color figure online)

7 Conclusion

In conclusion, this paper extends the preliminary results presented in Kirszenberg et al. [11] on $\mathscr{L}$Morph and $\mathscr{S}$Morph morphological layers. The rationale behind both layers is the same, namely to provide smooth and differentiable approximations of min and max operators to achieve trainable grayscale erosions and dilations. However, $\mathscr{L}$Morph relies on the asymptotic properties of the CHM (similarly to the $PConv$ layer of Masci et al. [15]), while $\mathscr{S}$Morph leverages the α-softmax function to reach the same goal, thus sidestepping some of the limitations shared by the $PConv$ and $\mathscr{L}$Morph layers. In Kirszenberg et al. [11], promising results were reported in favor of $\mathscr{L}$Morph and $\mathscr{S}$Morph morphological layers regarding their capacity to learn morphological operations with grayscale structuring elements. However, a single run per scenario was conducted, and several edge cases were raised but left unanswered.

In this present article, we remedy to the shallow preliminary results reported by Kirszenberg et al. [11]. More specifically, we now perform multiple runs per learning scenario and thoroughly evaluate the obtained results in terms of qualitative and quantitative performances as well as robustness to initialization. Those results confirm that morphological networks based on $\mathscr{L}$Morph and $\mathscr{S}$Morph layers are indeed able to learn morphological operations along with their grayscale structuring element, almost all the time outperforming the $PConv$ layer (that serves as a baseline in terms of performances), with $\mathscr{S}$Morph also surpassing $\mathscr{L}$Morph. Besides, we show that the investigated morphological layers are well stable since they always converge toward the same solution (up to some isolated cases). We also further focus on the convergence edge case reported by Kirszenberg et al. [11] and conduct some additional experiments to settle it. While we are up to now not able to definitively settle those convergence issues, a plausible solution has been identified. Finally, in contrast to $PConv$ and $\mathscr{L}$Morph layers that are limited by their requirement of a rescaled input within the range [1, 2], the $\mathscr{S}$Morph layer is also compatible with a fully binary morphological framework. Thus, we evaluate the capacity of $\mathscr{S}$Morph to operate with binary morphology and to learn binary structuring element. Even though the results reported for opening and closing scenarios would require much analysis, those obtained on erosion and dilation confirm that the $\mathscr{S}$Morph layer well accommodates to binary morphological frameworks. Yet, some additional efforts are still needed to fully resolve those convergence edge cases. Finally, while the presented results are very encouraging, those should be only taken as a proof of concept regarding the behavior soundness and the performances of the proposed $\mathscr{L}$Morph and $\mathscr{S}$Morph layers. As a matter of fact, recovering a morphological operation and its associated structuring element is a situation that is seldom encountered in a practical image processing application. Therefore, the integration of $\mathscr{L}$Morph and $\mathscr{S}$Morph into more complex network architectures and their evaluation on concrete image processing applications are part of our future research avenues.

References

1. Angulo, J.: Pseudo-morphological image diffusion using the counter-harmonic paradigm. In: International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, pp. 426–437 (2010)
2. Bloch, I., Blusseau, S., Pérez, R.P., Puybareau, É., Tochon, G.: On some associations between mathematical morphology and artificial intelligence. In: International Conference on Discrete Geometry and Mathematical Morphology. Springer, pp. 457–469 (2021)
3. Bullen, P.S.: Handbook of Means and Their Inequalities, vol. 560. Springer, Berlin (2013)
4. Calafiore, G.C., Gaubert, S., Possieri, C.: Log-sum-exp neural networks and posynomial models for convex and log-log-convex data. IEEE Trans. Neural Netw. Learn. Syst. **31**(3), 827–838 (2019)
5. Charisopoulos, V., Maragos, P.: Morphological perceptrons: geometry and training algorithms. In: International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing. Springer, pp. 3–15 (2017)
6. Dalla Mura, M., Benediktsson, J.A., Bruzzone, L.: Alternating sequential filters with morphological attribute operators for the analysis of remote sensing images. In: Image and Signal Processing for Remote Sensing XVI, vol. 7830. International Society for Optics and Photonics, p. 783006 (2010)
7. Davidson, J.L., Ritter, G.X.: Theory of morphological neural networks. In: Digital Optical Computing II, vol. 1215. International Society for Optics and Photonics, pp. 378–388 (1990)
8. Franchi, G., Fehri, A., Yao, A.: Deep morphological networks. Pattern Recogn. **102**, 107246 (2020)
9. Hassoun, M.H., et al.: Fundamentals of Artificial Neural Networks. MIT Press, Cambridge (1995)
10. Hernández, G., Zamora, E., Sossa, H., Téllez, G., Furlán, F.: Hybrid neural networks for big data classification. Neurocomputing **390**, 327–340 (2020)
11. 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. Springer, pp. 470–482 (2021)
12. 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)
13. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
14. LeCun, Y., Cortes, C., Burges, C.J.: The MNIST database of handwritten digits, **10**(34), 14. http://yann.lecun.com/exdb/mnist (1998)
15. Masci, J., Angulo, J., Schmidhuber, J.: A learning framework for morphological operators using counter–harmonic mean. In: International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing. Springer, pp. 329–340 (2013)
16. Mellouli, D., Hamdani, T.M., Ayed, M.B., Alimi, A.M.: Morph-cnn: a morphological convolutional neural network for image classification. In: International Conference on Neural Information Processing. Springer, pp. 110–117 (2017)
17. Mellouli, D., Hamdani, T.M., Sanchez-Medina, J.J., Ayed, M.B., Alimi, A.M.: Morphological convolutional neural network architecture for digit recognition. IEEE Trans. Neural Netw. Learn. Syst. **30**(9), 2876–2885 (2019)
18. Mondal, R., Dey, M.S., Chanda, B.: Image restoration by learning morphological opening–closing network. Math. Morphol. Theory Appl. **4**(1), 87–107 (2020)
19. Nogueira, K., Chanussot, J., Dalla Mura, M., Schwartz, W.R., Santos, J.A.d.: An introduction to deep morphological networks. arXiv preprint arXiv:1906.01751 (2019)
20. Pessoa, L.F., Maragos, P.: Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition. Pattern Recogn. **33**(6), 945–960 (2000)
21. Ritter, G.X., Sussner, P.: An introduction to morphological neural networks. In: Proceedings of 13th International Conference on Pattern Recognition, vol. 4. IEEE, pp. 709–717 (1996)
22. Roy, S.K., Mondal, R., Paoletti, M.E., Haut, J.M., Plaza, A.J.: Morphological convolutional neural networks for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. (2021)
23. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, New York (1983)
24. Serra, J., Vincent, L.: An overview of morphological filtering. Circuits Syst. Signal Process. **11**(1), 47–108 (1992)
25. Shen, Y., Zhong, X., Shih, F.Y.: Deep morphological neural networks. arXiv preprint arXiv:1909.01532 (2019)
26. Shih, F.Y., Shen, Y., Zhong, X.: Development of deep learning framework for mathematical morphology. Int. J. Pattern Recogn. Artif Intell. **33**(06), 1954024 (2019)
27. Soille, P.: Morphological Image Analysis: Principles and Applications. Springer, Berlin (2013)
28. Sussner, P.: Morphological perceptron learning. In: Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC). IEEE, pp. 477–482 (1998)
29. Sussner, P., Campiotti, I.: Extreme learning machine for a new hybrid morphological/linear perceptron. Neural Netw. **123**, 288–298 (2020)
30. Sussner, P., Esmi, E.L.: Morphological perceptrons with competitive learning: lattice-theoretical framework and constructive learning algorithm. Inf. Sci. **181**(10), 1929–1950 (2011)
31. Valle, M.E.: Reduced dilation-erosion perceptron for binary classification. Mathematics **8**(4), 512 (2020)

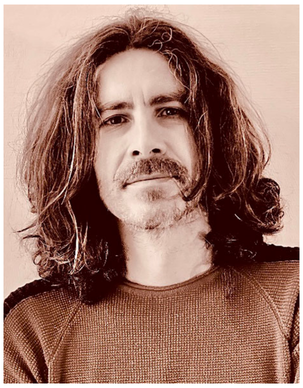
32. Wilson, S.S.: Morphological networks. In: Visual Communications and Image Processing IV, vol. 1199. International Society for Optics and Photonics, pp. 483–495 (1989)
33. Zamora, E., Sossa, H.: Dendrite morphological neurons trained by stochastic gradient descent. Neurocomputing **260**, 420–431 (2017)
34. Zhang, Y., Blusseau, S., Velasco-Forero, S., Bloch, I., Angulo, J.: Max-plus operators applied to filter selection and model pruning in neural networks. In: International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing. Springer, pp. 310–322 (2019)

Romain Hermary is currently pursuing an Engineering degree in Computer Science at EPITA in Paris where he specializes in image processing and machine learning. He also works as a graduate research student within EPITA Research and Development Laboratory (LRDE). His research interests include the development and optimization of deep learning methods and algorithms applicable on computer vision problems.

Guillaume Tochon received the Engineering and Ph.D. degrees in signal and image processing from the Grenoble Institute of Technology (Grenoble-INP) and from the University Grenoble Alpes in 2012 and 2015, respectively. He is now an associate professor at EPITA in Paris, and conducts his research within EPITA Research and Development Laboratory (LRDE). His research interests include the use of advanced mathematical morphology tools (such as hierarchical representations) and their integration within machine and deep learning framework, mostly for satellite and spatial imagery applications.

Èlodie Puybareau received a Ph.D. degree in signal and image processing from the University Paris Est in 2016. She is now an associate professor at EPITA and conduct her reserch with EPITA Research and Development Laboratory (LRDE). She is an expert in mathematical morphology and deep learning with a particular interest in feature extraction and classification for image diagnosis, especially for small or fuzzy object detection.

Alexandre Kirszenberg received the Engineering degree in Computer Science from EPITA in 2020. He also was a graduate research student at the EPITA Research and Development Laboratory (LRDE) where he worked on applying the theory of mathematical morphology within deep neural networks.



Jesús Angulo was born in Cuenca, Spain, in 1975. He received a degree in Telecommunications Engineering from the Polytechnical University of Valencia, Spain, in 1999, with a Master Thesis on Image and Video Processing. He obtained his Ph.D. in Mathematical Morphology and Image Processing in 2003 from the Ecole des Mines de Paris (France) under the guidance of Prof. Jean Serra. He got his Habilitation degree (French HDR) from the Université Paris-Est in 2012. He is currently senior research scientist (Directeur de Recherche) at the Center for Mathematical Morphology, in charge of the Ph.D. studies and head of the Department of Mathematics and Systems at MINES ParisTech, PSL Université Paris. He has contributed to more than 50 journal papers, 120 international conferences and 1 patent. His research interests are in the areas of mathematical morphology, max-plus and max-min mathematics, interaction between machine learning and structured data/image processing, non-conventional imaging and applications to the development of theoretically-sound and high-performance algorithms and software in the fields of Biomedicine/Biotechnology, Remote Sensing and Radar, Material Science and Industrial Vision.