Jakub Tomasz Micorek, BSc

Exploring a Learning Framework for Morphological Operations

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Information and Computer Engineering

submitted to

Graz University of Technology

Supervisors

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Axel Pinz

Dipl.-Ing. Moritz Kampelmühler, BSc.

Institute of Electrical Measurement and Sensor Systems

Graz, January 2020

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Abstract

Mathematical morphology is the study of shape and aims to remove/enhance structures in images using erosions/dilations and structuring elements. A learning based approach using the counter-harmonic mean shows that learning morphological operators is possible, though the resulting approximations are rather smooth which is caused by the positive valued constraint on structuring elements. Our novel robust counter-harmonic mean filter is more stable during learning and improves the approximation quality by avoiding clipping negative structuring element values. We show that our novel morphological layers are able to learn good approximations of morphological operators within purely morphological neural networks enabling further research on learning based mathematical morphology.

Kurzfassung

Mathematische Morphologie beschäftigt sich mit der Lehre von Form und Gestalt. Mit einem Strukturlement können Strukturen in einem digitalen Bild mit dem morphologischen Operator "Erosion" entfernt und mit dem morphologischen Operator "Dilatation" hervorgehoben werden. Ein Gradientenverfahren welches den "Counter-Harmonic Mean" nutzt, ermöglicht das Lernen von morphologischen Operatoren und Strukturelementen. Das Optimierungsverfahren ist instabil aufgrund von ausschließlich positiv beschränkten Werten der Strukturelemente. Die resultierenden angenäherten Bilder sind jedoch unscharf. Wir entwickeln eine robuste Variante des "Counter-Harmonic Mean" und ermöglichen ein stabileres Lernen und bessere Annäherungen an die morphologischen Operatoren. Weiters zeigen wir, dass unsere entwickelten morphologischen neuronalen Schichten in der Lage sind morphologische Operatoren in rein morphologischen neuronalen Netzwerken zu lernen. Unsere Arbeit ermöglicht ein genaueres Lernen von morphologischen Operatoren und ermöglicht mehrere morphologische Operatoren in einem morphologischen neuronalen Netzwerk zu erlernen.

Contents

1 Introduction

Recent research in the image processing domain highlights the importance of learning based approaches like Convolutional Neural Networks (CNNs) [10]. CNNs are trained using the backpropagation algorithm and are exceptional in object detection [5], image classification [6] and other applications - an overview is given in [4]. Mathematical morphology is another image processing research field and builds on top of two basic morphological operators, erosion and dilation [16]. Erosion and dilation apply min/max operators in the neighbourhood defined by a structuring element where the structuring element represents a kernel which is used to enhance or remove structure from an image. Combining multiple erosions/dilations forms more complex morphological pipelines and morphological operators. Finding appropriate structuring elements, morphological pipelines and other morphological operators is problem specific and requires a domain expert [15] to engineer them; therefore, we intend to learn well fitting configurations with the backpropagation algorithm. Due to the min/max operators used in erosion/dilation, gradient computation and error backpropagation is a challenging problem.

Erosion and dilation can be approximated with the counter-harmonic mean filter [18] where a single parameter controls between approximations of min/max and convolution operators. Another component of the counter-harmonic mean is a convolutional kernel which represents the structuring element. The innovative work of the counter-harmonic mean learning framework [12] enables learning of morphological operators with the backpropagation algorithm because the counter-harmonic mean is differentiable w.r.t. the structuring element and the morphological operator. However, the authors obtain rather smooth morphologically processed images. In our work we reimplement the proposed learning framework and study the learning behaviour using the counter-harmonic mean. Based on our findings we develop a more robust version of the counter-harmonic mean and achieve more stable learning and better approximation quality over the counter-harmonic mean.

The Morph-CNN proposed in [13] combines counter-harmonic mean filters and convolutions within a single network and the Morph-CNN is used for image classification. The authors fall short of addressing the problem how multiple morphologically processed images are combined within a layer using multiple counter-harmonic mean filters and they do not report if the counter-harmonic mean filters learn morphological operators or convolutions. In our work we propose two novel types of morphological network layers using our novel robust counter-harmonic mean filter. Inspired by depthwise separable convolutions [3], both of our proposed morphological layers combine multiple robust counter-harmonic mean filters within the morphological layers. We show that a morphological neural network purely built on top of our morphological layers is able to extract features for image classification. In addition, we show that our morphological layers are able to learn approximations of morphological operators within the network.

This thesis proposes a more robust learning framework for learning morphological operators and provides two novel morphological layers which are able to learn morphological operators. Our work enables further research on learning based mathematical morphology.

2 Fundamentals of Mathematical Morphology

Mathematical morphology is well defined with mathematical set theory [16] and is built on top of the two basic morphological operators, erosion and dilation. These operators are shift-invariant and utilize a structuring element to morphologically process an image. The structuring element represents a kernel which is used to enhance or remove structure from the image. Combinations of erosion and dilation are called morphological pipelines and the two most common pipelines are opening and closing, which consist of erosion followed by dilation, and dilation followed by erosion, respectively. Opening and closing are the basic morphological operators for morphological image filtering [17]. The mathematical morphology tools introduced above aim to extract shape based features from images. However, finding appropriate structuring elements, morphological pipelines and other morphological operators are problem specific and require a domain expert [15]; therefore, we intend to learn well fitting configurations with the backpropagation algorithm.

In order to better understand mathematical morphology and why it is hard to learn with the backpropagation algorithm, we provide the necessary theoretical background. We base our notation on [17], a pioneering work in the field of morphological image processing.

2.1 Disambiguation

We introduce the required mathematical background in order to define and classify morphological operators and to understand their specific properties. Knowing these properties helps us to understand the behaviour of a specific morphological operator. Feel free to skip this section and revisit it again later when we discuss and refer to specific properties.

The ordering and complementation properties are required for the increasing, (anti-)extensive, (self-)dual and idempotence properties below.

Ordering: Images f and g can be ordered in the following way, where $\mathbf{x}$ is the pixel position:

$$f \leqslant g \quad \Leftrightarrow \quad \forall \mathbf{x}, f(\mathbf{x}) \leqslant g(\mathbf{x}). \tag{2.1}$$

For example, if f is an all black image and g is an all white image the ordering relation holds.

Complementation: The complement f^c of an image f at pixel position $\mathbf{x}$ is the difference between the maximum value t_{max} of the image domain of f and the image f:

$$f^c(\mathbf{x}) = t_{max} - f(\mathbf{x}). \tag{2.2}$$

Complementing the image twice results in the initial image:

$$(f^c)^c = f. \tag{2.3}$$

Consider an image $f \in [0,1]$, where $t_{max} = 1$. We show the image f, its complement f^c and the complement of the complemented image $(f^c)^c = f$.

$$f = \begin{array}{|c|c|c|}\hline 0.0 & 0.1 & 0.2 \\ \hline 0.4 & 0.5 & 0.6 \\ \hline 0.8 & 0.9 & 1.0 \\ \hline \end{array} \qquad f^c = \begin{array}{|c|c|c|}\hline 1.0 & 0.9 & 0.8 \\ \hline 0.6 & 0.5 & 0.4 \\ \hline 0.2 & 0.1 & 0.0 \\ \hline \end{array} \qquad (f^c)^c = \begin{array}{|c|c|c|}\hline 0.0 & 0.1 & 0.2 \\ \hline 0.4 & 0.5 & 0.6 \\ \hline 0.8 & 0.9 & 1.0 \\ \hline \end{array} = f \tag{2.4}$$

The complementation property is crucial to define the (self-)duality property of morphological operators in (2.8) and (2.9).

Increasing: An operator Ψ transforms an image f by $\Psi(f)$. A transformation is called increasing if it does not change the ordering of images:

$$\forall f, g : \quad f \leqslant g \quad \Rightarrow \quad \Psi(f) \leqslant \Psi(g). \tag{2.5}$$

Erosion/dilation and opening/closing are increasing as they do not change the ordering of images.

Extensive: A transformation Ψ is called extensive if, for all images f, the transformed image is higher than or equal to the original image:

$$\forall f: \quad f \leqslant \Psi(f). \tag{2.6}$$

For example, let Ψ be a transformation which introduces just salt noise. Then the image with salt noise is brighter than the clean image and preserves the ordering. Dilation and closing also brighten the image and if the structuring element contains its origin the ordering is preserved, resulting in an extensive transformation.

Anti-Extensive: A transformation Ψ is called anit-extensive if, for all images f, the transformed image is lower than or equal to the original image:

$$\forall f: \quad \Psi(f) \leqslant f. \tag{2.7}$$

For example, let Ψ be a transformation which introduces just pepper noise. Then the image with pepper noise is darker than the clean image and preserves the ordering. Erosion and opening also darken the image and if the structuring element contains its origin the ordering is preserved, resulting in an anti-extensive transformation.

Duality: Two transformations Ψ and Φ are called dual if applying Ψ to an image f is equal to applying Φ to the complement image f^c and taking the complement of the resulting image:

$$\forall f: \quad \Psi(f) = \left(\Phi(f^c)\right)^c. \tag{2.8}$$

For example, min/max operators are dual, as we can see for $\Psi = \min$ and $\Phi = \max$ and utilize f from (2.4).

Erosion utilizes the min operator in the neighbourhood of the structuring element and dilation the max operator respectively. The min/max operators are dual and therefore the erosion can be expressed with dilation utilizing complements and vice versa.

Self-Duality: A transformation Ψ is called self-dual if applying Ψ to an image f is equal to applying Ψ to the complement image f^c and taking the complement of the resulting image:

$$\forall f : \quad \Psi(f) = (\Psi(f^c))^c . \tag{2.9}$$

Note that the difference between self-duality and duality is that the same transformation is used ($\Psi = \Phi$) twice. For example, a median filter is self-dual, as we can see for $\Psi =$ median applied to f from (2.4).

Idempotence: A transformation Ψ is called idempotent if applying Ψ multiple times equals to applying the transformation only once:

$$\forall f : \quad \Psi(\Psi(f)) = \Psi(f). \tag{2.10}$$

Opening/closing are morphological filters which remove peaks/valleys narrower than the structuring element. Applying the same morphological filter again has no impact, as all peaks/valleys were already removed. In contrast, erosion/dilation and convolution/correlation are not idempotent.

2.2 Structuring Element

A structuring element is a small set and is used to probe an image, similar to the filter kernel of a convolution or correlation operation. We distinguish between flat structuring elements B_f and non-flat (volumetric, grayscale) structuring elements B_v, shown in Figure 2.1.

Flat structuring elements are defined by a shape, a size and an origin. The shape of the flat structuring element may be a line, a disk, a cross, a square, or any arbitrary structure, which serves as a neighbourhood mask for the image to be processed. In contrast to flat structuring elements, non-flat structuring elements also have grayvalues assigned to the mask. A morphological operator determines the output value in the neighbourhood defined by the structuring element and outputs it to the location of the origin.

In the remaining sections of this chapter we are going to visualize how morphological operators process images. For visualization and explanatory purpose we will use 1D-signals and for this purpose utilize the 1D structuring elements shown in Figure 2.2.

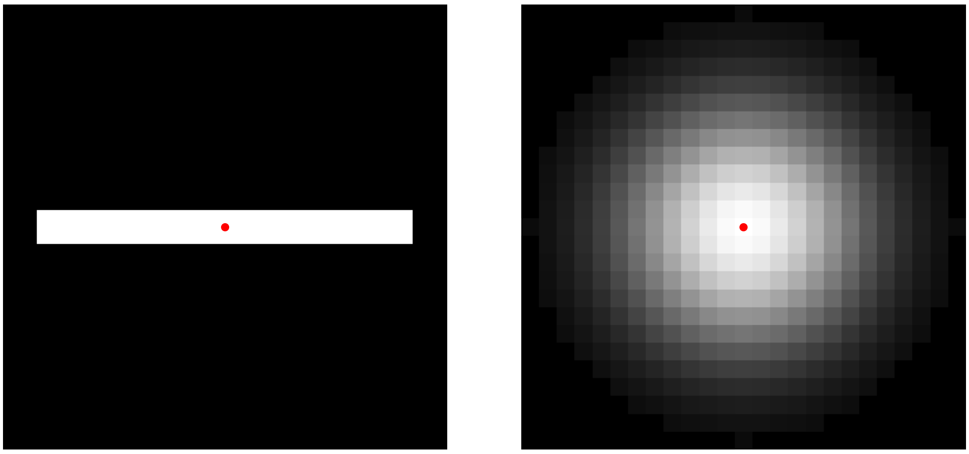


Figure 2.1: A flat structuring element is depicted on the left. Only pixels in the neighbourhood of the horizontal line are considered for the morphological operators. A non-flat structuring element is depicted on the right. It contains grayvalues. The origins are indicated with red dots, which lie in the centers. The black areas are for visualization purpose only, they are not considered part of the structuring element.

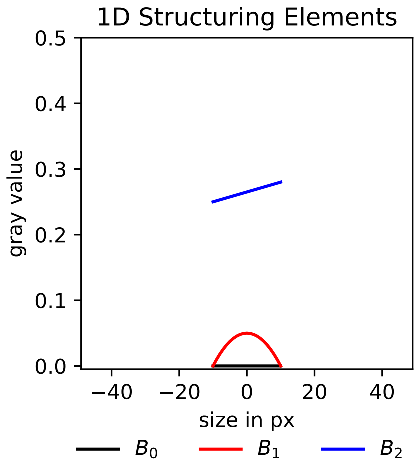


Figure 2.2: Three 1D structuring elements of size 21 are depicted as B_0, B_1 and B_2. B_0 is a flat structuring element, B_1 and B_2 are non-flat structuring elements forming a parabola and a slanted line.

2.3 Erosion and Dilation

Mathematical morphology stems from set theory and the two fundamental morphological operators are called erosion and dilation. Both operators utilize a structuring element to morphologically process an image.

The erosion ε of a set X with a flat structuring element B is defined as:

$$\varepsilon_B(X) = \{\mathbf{x} \mid B_{\mathbf{x}} \subseteq X\} . \tag{2.11}$$

At any location $\mathbf{x}$ where structuring element B fits into set X results in the eroded set.

The dilation δ of a set X with a flat structuring element B is defined as:

$$\delta_B(X) = \{\mathbf{x} \mid B_{\mathbf{x}} \cap X \neq 0\} . \tag{2.12}$$

At any location $\mathbf{x}$ where the structuring element B intersects with the set X results in the dilated set.

In Figure 2.3, we show an erosion and a dilation applied to a set. We clearly see that the erosion removes structure smaller than the disk shaped structuring element and shrinks the border area of larger structures. In case of dilation the border area grows and even closes narrow gaps. Erosion and dilation have no inverse, i.e. completely removed structure smaller than the structuring element caused by erosion can not be reconstructed, and the closed gap between the larger and smaller area caused by dilation can not bet opened again. Defining the right shape and size of the structuring element plays an important role in how the set will be modified.

Binary images can be interpreted as sets; i.e. foreground pixels define a set and the locations are discrete. The idea of binary morphology was later extended to grayscale images, where the set operators are replaced with min/max operators. While binary morphology only allows for flat structuring elements grayscale morphology also allows for non-flat structuring elements. From here on we will focus only on grayscale morphology, utilizing flat and non-flat structuring elements.

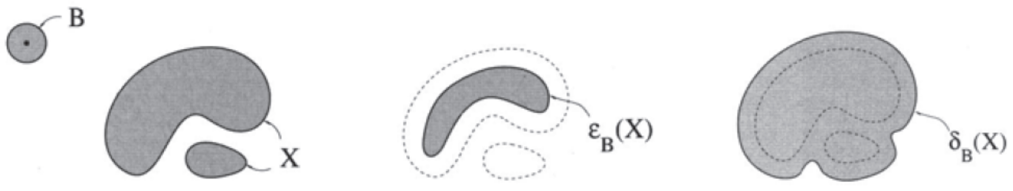


Figure 2.3: B is the structuring element where the origin is in the center. X is the set the morphological operator is applied on. In the center we show the resulting erosion (2.11) and on the right we show the resulting dilation (2.12) of set X. (Fig. 3.3, p.52 and Fig 3.6, p.54 from [17])

The erosion ε of an image f at location $\mathbf{x}$ in the neighbourhood defined by a flat structuring element B_f is the minimum in the neighbourhood:

$$[\varepsilon_{B_f}(f)](\mathbf{x}) = \min_{\mathbf{b} \in B_f} f(\mathbf{x} + \mathbf{b}). \tag{2.13}$$

In Figure 2.4, we show an erosion with a flat structuring element. Note that the full shape of the structuring element appears on local minima and local peaks narrower than the structuring element are attenuated. The erosion is an anti-extensive transformation (2.7); hence, the erosion darkens the image and the eroded image is lower than the initial image (2.1).

The dilation δ of an image f at location $\mathbf{x}$ in the neighbourhood defined by a flat structuring element B_f is the maximum in the neighbourhood:

$$[\delta_{B_f}(f)](\mathbf{x}) = \max_{\mathbf{b} \in B_f} f(\mathbf{x} + \mathbf{b}). \tag{2.14}$$

In Figure 2.5, we show a dilation with a flat structuring element. Note that the full shape of the structuring element appears on local maxima and local valleys narrower than the structuring element are filled. The dilation is an extensive transformation (2.6); hence, the dilation brightens the image and the dilated image is higher than the initial image (2.1).

Erosion ε and dilation δ are dual (2.8):

$$\varepsilon(f) = (\delta(f^c))^c \tag{2.15}$$

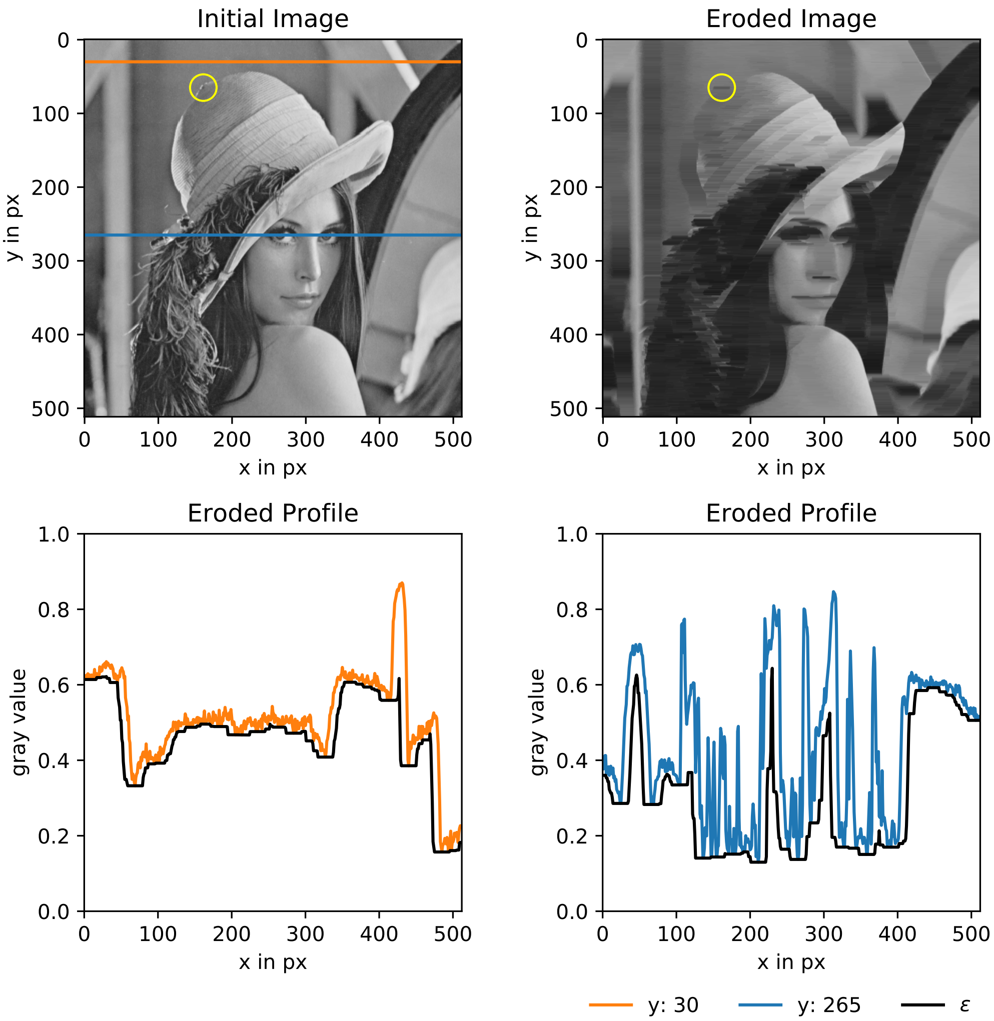


Figure 2.4: The initial image is shown in the top left. The grayvalue profiles, indicated by the horizontal lines in orange and blue, are shown at the bottom. We apply an erosion with a flat structuring element with the shape of a horizontal line (shown in Figure 2.1, left) of size 21 to the initial image. Only pixels in the horizontal neighbourhood are considered. The top right image is the resulting eroded image. The black graphs on the bottom indicate the eroded grayvalue profiles. An erosion darkens the image and attenuates local peaks narrower/smaller than the structuring element. Note how the full structuring element appears at single dark spots, an example is indicated by the yellow circle.

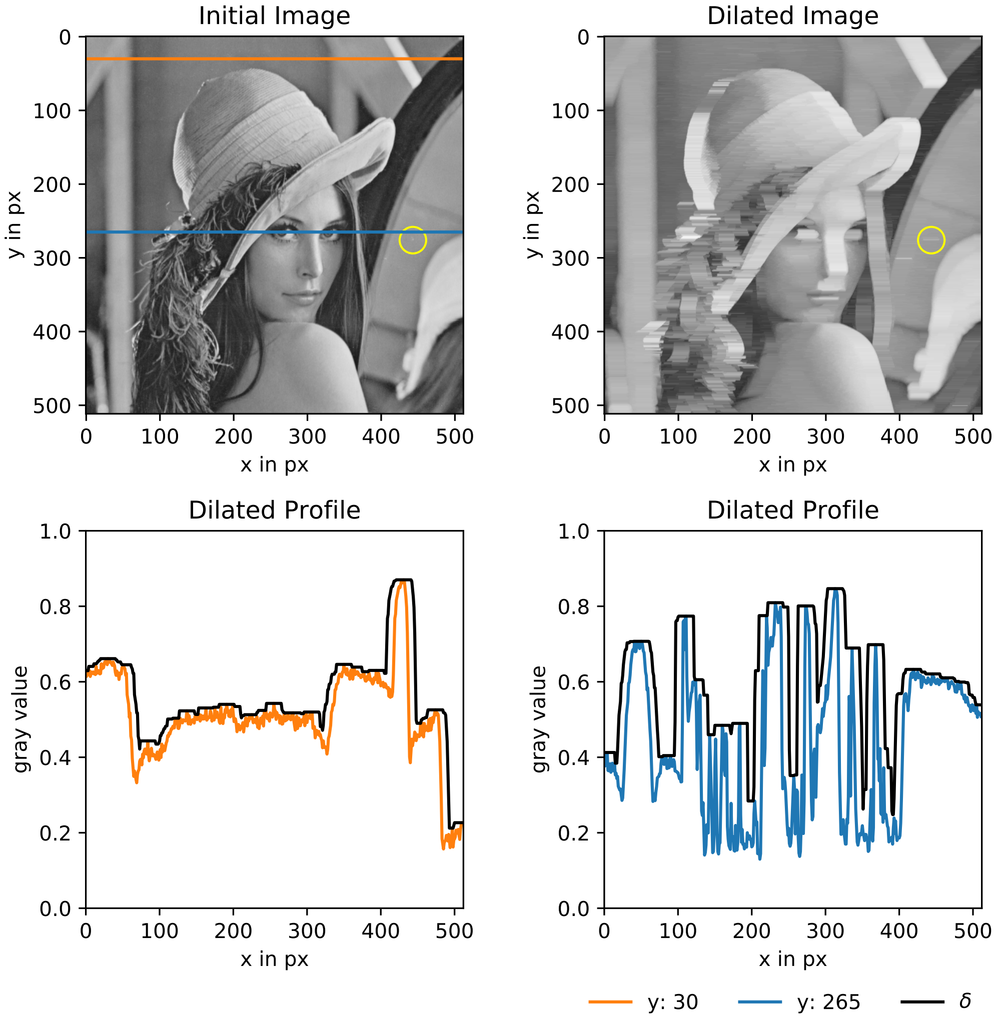


Figure 2.5: The initial image is shown in the top left. The grayvalue profiles, indicated by the horizontal lines in orange and blue, are shown at the bottom. We apply a dilation with a flat structuring element with the shape of a horizontal line (shown in Figure 2.1, left) of size 21 to the initial image. Only pixels in the horizontal neighbourhood are considered. The top right image is the resulting dilated image. The black graphs on the bottom indicate the dilated grayvalue profiles. A dilation brightens the image and fills local valleys narrower/smaller than the structuring element. Note how the full structuring element appears at single bright spots, an example is indicated by the yellow circle.

and increasing (2.5) operators.

In contrast to flat structuring elements B_f, non-flat structuring elements B_v contain grayvalues. Below we introduce erosion and dilation utilizing non-flat structuring elements.

The erosion ε of an image f at location $\mathbf{x}$ in the neighbourhood defined by a non-flat structuring element B_v is the minimum of the difference of the image and the structuring element:

$$[\varepsilon_{B_v}(f)](\mathbf{x}) = \min_{\mathbf{b} \in B_v} \{f(\mathbf{x}+\mathbf{b}) - B_v(\mathbf{b})\}\,. \tag{2.16}$$

The dilation δ of an image f at location $\mathbf{x}$ in the neighbourhood defined by a non-flat structuring element B_v is the maximum of the sum of the image and the structuring element:

$$[\delta_{B_v}(f)](\mathbf{x}) = \max_{\mathbf{b} \in B_v} \{f(\mathbf{x}+\mathbf{b}) + B_v(\mathbf{b})\}\,. \tag{2.17}$$

A non-flat structuring element B_v is set to the value range of the image domain. The erosion with a non-flat structuring element may be below the lower bound of the image domain and the dilation may exceed the upper bound. In both cases, the image is either clipped to the domain of the original image or the image domain is extended.

Note that a non-flat flat structuring element with all values equal to 0 is a flat structuring element; hence, $\varepsilon_{B_v}(f) = \varepsilon_{B_f}(f)$ and $\delta_{B_v}(f) = \delta_{B_f}(f)$ for $B_v \in \{0\}$.

In Figure 2.6, we compare erosion and dilation with flat and non-flat structuring elements on a 1D signal. The shapes of the structuring elements appear at local minima and maxima. In contrast to flat structuring elements which introduce horizontal lines to the eroded and dilated signal, non-flat structuring elements allow to introduce an arbitrary shape and add an offset. If the offset is too high, potentially undesired clipping to the domain of the signal can occur if the domain can not be expanded. We observe that erosions utilizing non-flat structuring elements are lower than or equal to erosions utilizing flat structuring elements of same size. Similarly dilations with non-flat structuring elements are lower than or equal to dilations with flat structuring elements;

hence, $\varepsilon_{B_v}(f) \leqslant \varepsilon_{B_f}(f) \leqslant f \leqslant \delta_{B_f}(f) \leqslant \delta_{B_v}(f)$.

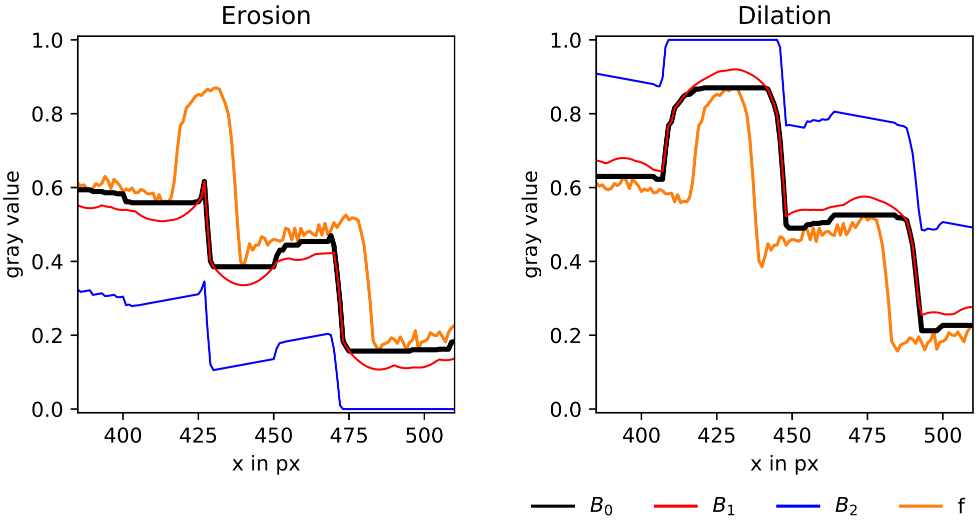


Figure 2.6: We show erosions (left) and dilations (right) of a 1D-signal f with flat (B_0) and non-flat (B_1, B_2) structuring elements introduced in Figure 2.2, where f is a magnified grayvalue profile from Figures 2.4 and 2.5. The shape of the structuring element appears at local minima (erosion) and maxima (dilation). While flat structuring elements do not add any offset and therefore only introduce horizontal lines, non-flat structuring elements add an offset and allow to add any arbitrary shape to peaks/valleys of the signal. The offset of structuring element B_2 is too large and introduces clipping if the signal is bounded between $[0, 1]$, in case of erosion from pixel 470 to 512 and in case of dilation from pixel 415 to 440.

2.4 Opening and Closing

A morphological pipeline is defined as a sequence of erosions and dilations, where opening and closing are the core representatives.

The opening operator is defined as an erosion followed by a dilation:

$$\gamma(f) = \delta(\varepsilon(f)). \tag{2.18}$$

In Figure 2.7, opening using a flat structuring element is shown. The erosion darkens the overall image and also removes local bright peaks which are narrower than the utilized structuring element. Then the following dilation compensates the darkening caused by erosion, while the removed bright peaks stay suppressed.

The closing operator is defined as a dilation followed by an erosion:

$$\phi(f) = \varepsilon(\delta(f)). \tag{2.19}$$

In Figure Figure 2.8, closing using a flat structuring element is shown. The dilation brightens the overall image and also fills local valleys which are narrower than the utilized structuring element. Then the following erosion compensates the brightening caused by dilation, while keeping the filled dark valleys.

Under the assumption that the structuring element is symmetrical the following properties hold: Opening is anti-extensive as it removes peaks and closing is extensive as it fills valleys. Both are dual (2.8), increasing (2.5) and idempotent (2.10) operators. Applying an idempotent operator multiple times has the same result as applying the operator only once. In other words, structures removed or enhanced by the same structuring element will not be affected again:

$$\begin{matrix} \gamma(f) = \gamma(\gamma(f)) \\ \phi(f) = \phi(\phi(f)) \end{matrix} \; . \tag{2.20}$$

A morphological filter selectively suppresses image structures, such as noise or irrelevant structures and is defined by the idempotence (2.10) and increasing (2.5) properties. Structures that are preserved should not be modified by further applications of the same filter [17]. While opening and closing fulfill these properties and therefore are morphological filters, erosion and dilation alone are not morphological filters.

In Figure 2.9, we compare opening and closing with flat and non-flat structuring element. The shape of the structuring element appears along the filtered signal. While flat structuring elements only allow for filtering with horizontal lines, non-flat structuring elements allow for filtering with any arbitrary shapes.

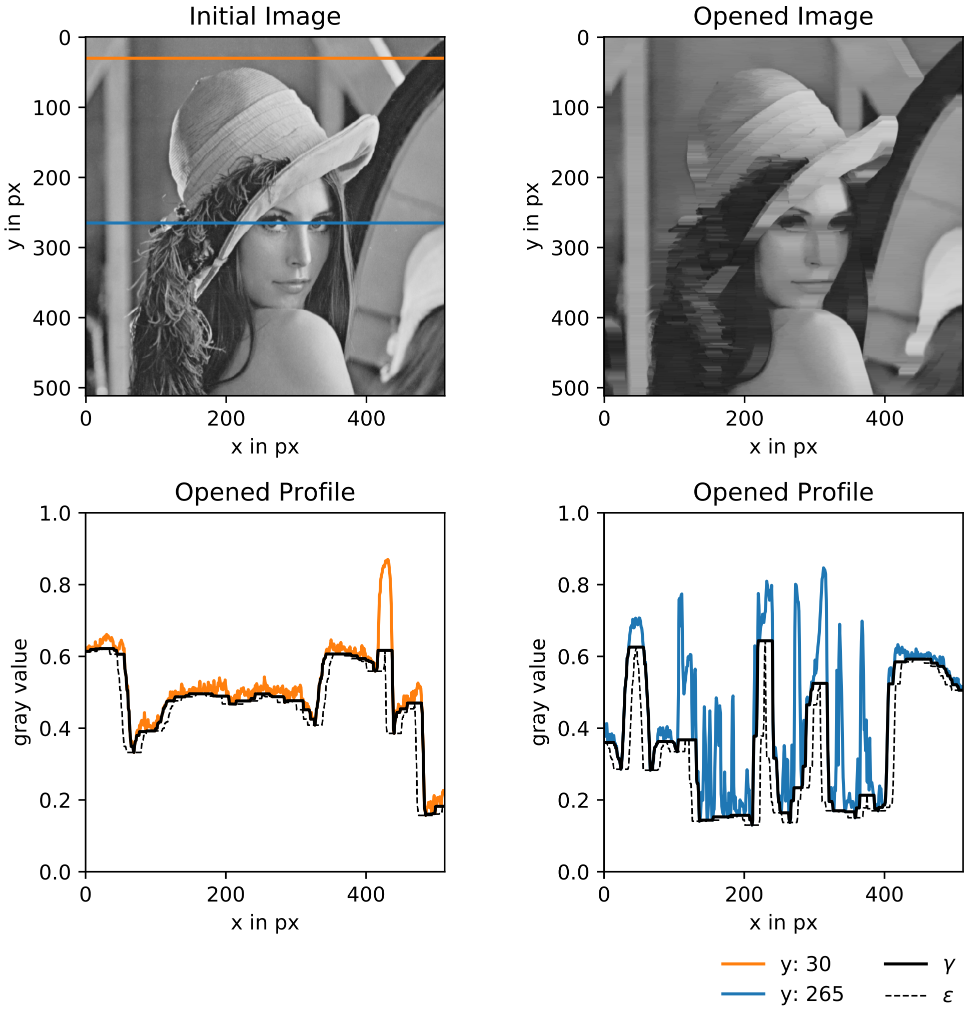


Figure 2.7: Opening of a grayscale image and grayvalue profiles from the image. We utilize a flat structuring element with the shape of a horizontal line (shown in Figure 2.1, left) of size 21. Opening γ is defined as an erosion ε followed by a dilation δ. The dashed graphs on the bottom indicate the eroded grayvalue profiles. Note how peaks narrower than the structuring element (< 21) are suppressed. Dilating the eroded grayvalue profile results in the opened profile in solid black. For opening we observe that structures smaller than the structuring element get suppressed while compensating the darkening of the erosion: $\varepsilon(f) \leqslant \gamma(f) \leqslant f$. The top right image is the resulting opened image. Note how, for example, white areas of the eyes get suppressed to darker neighbouring grayvalues.

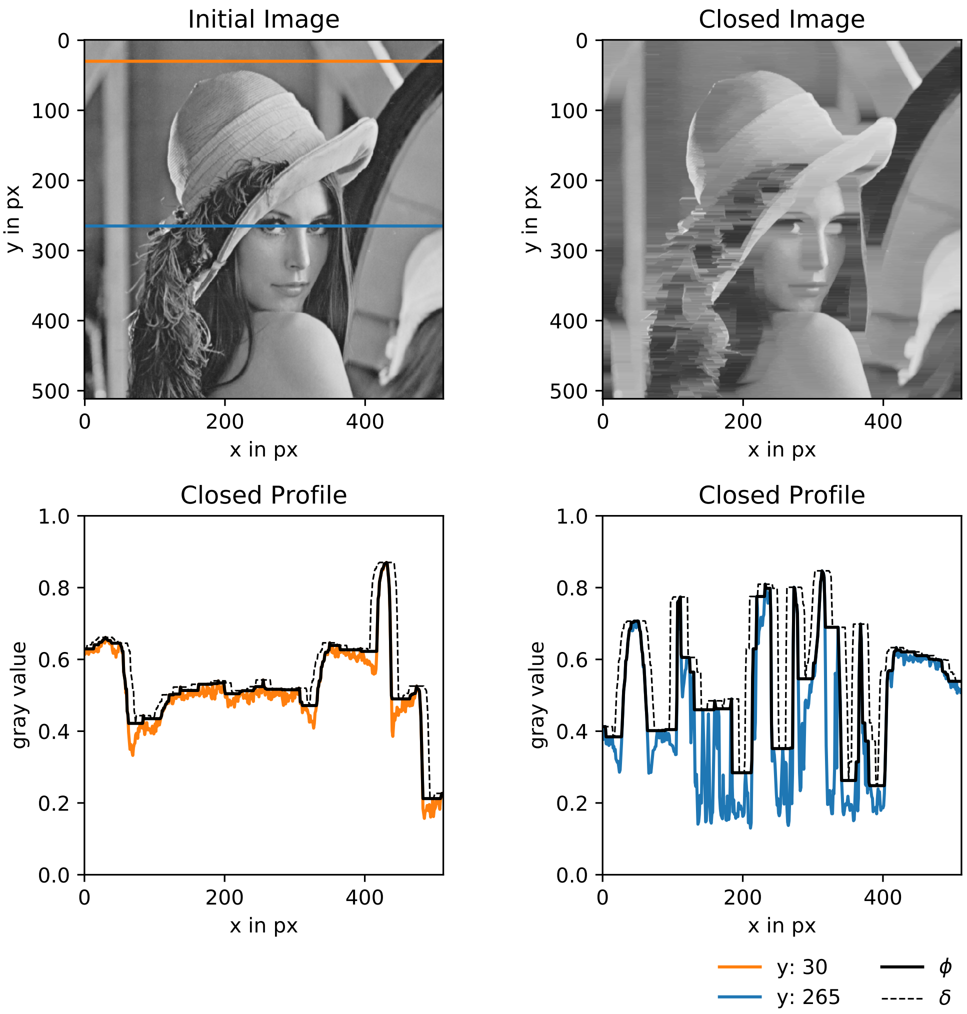


Figure 2.8: Closing of a grayscale image and grayvalue profiles from the image. We utilize a flat structuring element with the shape of a horizontal line (shown in Figure 2.1, left) of size 21. Closing ϕ is defined as a dilation δ followed by an erosion ε. The dashed graphs on the bottom indicate the dilated grayvalue profiles. Note how valleys narrower than the structuring element (< 21) are filled. Eroding the dilated grayvalue profile results in the closed profile in solid black. For closing we observe that structures smaller than the structuring element get suppressed while compensating the brightening of the dilation: $f \leqslant \phi(f) \leqslant \delta(f)$. The top right image is the resulting closed image. Note how, for example, dark areas like the iris get increased to brighter neighbouring grayvalues.

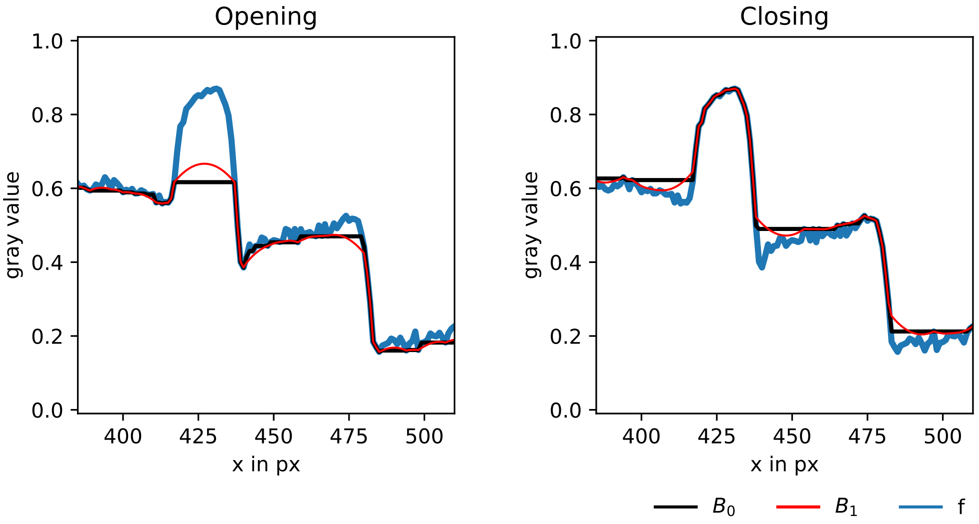


Figure 2.9: We compare openings (left) and closings (right) of a 1D-signal f with flat (B_0) and non-flat (B_1) structuring elements introduced in Figure 2.2, where f is a magnified grayvalue profile from Figures 2.7 and 2.8. The shape of the structuring element appears at local maxima (opening) and local minima (closing). Compared to the flat structuring element B_0 the parabolic structuring element B_1 results in a smoother signal.

2.5 Top-Hats

Morphological operators can be combined with residuals to form new morphological operators such as top-hat transformations.

The white top-hat transformation is the difference of the image f and the opening of f:

$$WTH(f) = f - \gamma(f). \tag{2.21}$$

Openings suppress bright peaks smaller than the utilized shape of the structuring element; hence, the white top-hat transformation results in the suppressed peaks as shown in Figure 2.10 left.

The black top-hat transformation is the difference of the closing of f and f itself:

$$BTH(f) = \phi(f) - f. \tag{2.22}$$

Closings fill dark valleys smaller than the utilized shape of the structuring element; hence, the black top-hat transformation results in the filled valleys as shown in Figure 2.10 right.

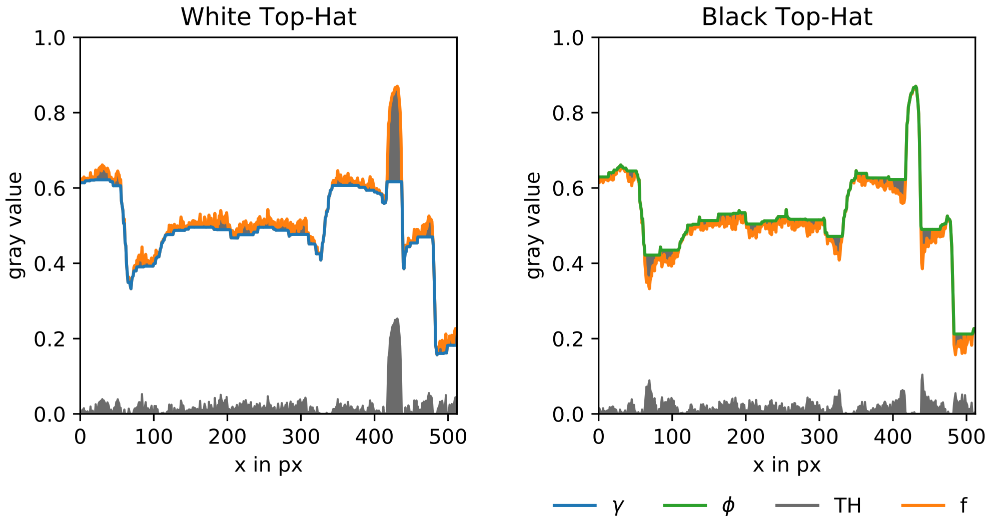


Figure 2.10: A flat structuring of size 21 is used to open (blue) and close (green) the image f. The top-hats obtained by the residuals are shown in gray. The white top-hat represents the removed bright structures and the black top-hat represents the closed valleys.

3 Related Work

Convolutional Neural Networks (CNNs) [10] are powerful methods used for feature extraction in the image processing and computer vision domain. The convolutional kernels in a CNN aim to extract meaningful features and are learned with the backpropagation algorithm. Inspired by the huge success of CNNs and the similarity between convolutional kernels and structuring elements, we aim to learn morphological pipelines by replacing the convolution operator with morphological operators. Note that a linear convolution, which calculates a weighted sum, is not able to calculate erosion or dilation due to the min and max operator used in the morphological operators.

3.1 Mathematical Morphology Learning Frameworks

One approach to learn structuring elements is to replace the convolution operator of a CNN with an erosion (2.16) or a dilation (2.17). The erosion and dilation units have to be predefined in the network structure. Flat and non-flat structuring elements are then learned with the backpropagation algorithm. However, not all values of each structuring element are updated simultaneously. Similar to max-pooling, the error is only backpropagated at the locations of minima and maxima. For this reason, only a single value of each structuring element is updated for a single training iteration. This method was applied in [15] to remove rain in images. The authors designed two morphological networks with predefined sequences of erosions and dilations and learned the structuring elements. The only difference between these networks is that one network uses fewer morphological operators to reduce the parameter count. They compared the morphological networks against a CNN, where the CNN performed marginally better w.r.t PSNR and structural similarity. A huge advantage of the morphological networks over the CNN is the parameter count. The morphological networks only require 0.27% (normal network) and 0.04% (smaller network) of the CNN parameter count; hence, learning morphological operators might be beneficial to applications with limited available computational power. To summarize, the drawbacks of such an approach are that the erosion and dilation units have to be predefined in each layer and that the structuring element updates are sparse.

Designing predefined pipelines in a network architecture may reduce the success of extracting meaningful features; therefore, we aim to also learn sequences of morphological operators, so that morphological pipelines can be learned through backpropagation. The counter-harmonic mean [18] can be used as an approximation of morphological operators with flat structuring elements. One parameter of the counter-harmonic mean allows to control between erosion, convolution and dilation. Another paramaeter is a convolutional kernel which represents the structuring element. A learning framework using the counter-harmonic mean was proposed in [12]. The authors show that learning the morphological operators and the structuring element simultaneously is possible, as the counter-harmonic mean is differentiable w.r.t. the structuring element and the parameter controlling the approximation of the morphological operators. In contrast to the min/max operators, another advantage of the counter-harmonic mean is that the error is backpropagated through all values of a structuring element and therefore the whole structuring element is updated simultaneously for each training sample. The authors show that the learning framework is able to learn erosions and dilations, openings and closings and the top-hat transformations. In addition, they are able to remove salt and pepper noise from images. In [13] the authors combine four counter-harmonic mean layers, five convolutional network-in-network layers [11] and two pooling layers to classify the MNIST [14] dataset. They compared their proposed architecture to smaller CNN architectures and outperformed the smaller networks. However, the MNIST dataset is a questionable benchmark for such a deep network architecture and they did not discuss if the counter-harmonic mean layers approximated morphological operators or convolutions.

The counter-harmonic mean learning framework seems most promising, as it meets our requirements to learn structuring elements and morphological operators, we choose to further investigate this learning approach. First, we implement the counter-harmonic mean learning framework proposed in [12] and investigate the results. Second, we aim to classify image data like in [13], but only utilizing counter-harmonic mean layers to investigate if morphological operators or convolutions are learned. In Section 3.2 we review the counter-harmonic mean in more detail.

During writing this thesis, [15] proposed a more robust version to learn structuring elements with a soft minimum operator for erosion and a soft maximum operator for dilation. The morphological operators still have to be predefined in the network architecture and only the structuring elements can be learned.

3.2 Counter-Harmonic Mean

Erosion and dilation without the notions of min/max was first proposed in [18], where both morphological operators are approximated [1] with the counter-harmonic mean (CHM) [2]. The approximation turns out to be very effective in image smoothing, edge detection and image sharpening in presence of noise because a parameter P allows to tune the trade-off between detection of extrema and smoothing [18].

The counter-harmonic mean of order $P \in \mathbb{R}$ is defined by :

$$\kappa_{\mathbf{w}}^{P}(f)(\mathbf{x}) = \frac{\left(f^{P+1} * \mathbf{w}\right)(\mathbf{x})}{\left(f^{P} * \mathbf{w}\right)(\mathbf{x})}, \tag{3.1}$$

where f is the real-valued image in the range $[0, 1]$, $\mathbf{x}$ being the pixel coordinate, f^P the pixel values raised to the power of P and $\mathbf{w} \in \mathbb{R}^+$ is the convolutional kernel with positive values. The resulting convolutions of the numerator and denominator are divided pixelwise. CHM can be interpreted as the P-deformed convolution and is denoted as $\kappa_{\mathbf{w}}^{P}(f)(\mathbf{x}) \equiv (f *_P \mathbf{w})(\mathbf{x})$. Note, when addressing individual values within the kernel $\mathbf{w}$ we use the vector notation using subscripts, e.g. $w_{i,j}$.

In short, parameter P controls the approximation between erosion, convolution and dilation, where $\mathbf{w}$ represents the flat structuring element (Section 2.2).

For $P \ll 0$ the smallest value in the local neighbourhood of the resulting convolution with $\mathbf{w}$ will dominate. Similar the largest value will dominate for

$P \gg 0$. As only finite numbers for P are feasible, we speak of a pseudo-erosion and pseudo-dilation, which can be interpreted as the non-flat erosion and non-flat dilation [12]. Then the structuring element B is approximately:

$$B \approx \frac{1}{P} \log(\mathbf{w}). \tag{3.2}$$

The convolutional kernel $\mathbf{w}$, which is usually defined as a square support window, represents a structuring element B where non-zero values in $\mathbf{w}$ define the shape of the structuring element B.

The CHM filtered image stays in the definition range while no clipping occurs. As P shrinks/grows to $\pm\infty$, the structuring element behaves like a flat structuring element, because $\lim_{P \to \pm\infty} \frac{1}{P} \log(\mathbf{w}) = 0$.

The limit cases $P = \pm\infty$ are flat erosion with:

$$\lim_{P \to -\infty} \left(f *_P \mathbf{w}\right)(\mathbf{x}) = \varepsilon(f)(\mathbf{x}), \tag{3.3}$$

and flat dilation with:

$$\lim_{P \to +\infty} \left(f *_P \mathbf{w}\right)(\mathbf{x}) = \delta(f)(\mathbf{x}). \tag{3.4}$$

The counter-harmonic mean is differentiable w.r.t. P and $\mathbf{w}$, and thus the backpropagation algorithm can be applied. In the learning framework proposed in [13], CHM filters combined with convolutions learn morphological operators and structuring elements simultaneously.

For P close to zero, the approximation of erosion and dilation is not sufficient because the CHM acts like a convolution with a positive kernel and leads to a blurry image. For salt and pepper noise removal, such smoothing behaviour might be better suited than a regular erosion/dilation.

We compare the CHM properties, introduced in [9], and the CHM approximation quality [1] to classical mathematical morphology (Section 2) below.

CHM pseudo-erosion ($P \ll 0$) converges to erosion faster than CHM pseudo-dilation ($P \gg 0$) converges to dilation, i.e., assuming $P > 0$:

$$\left|\kappa_B^{-P}(f) - \varepsilon_B(f)\right| \leq \left|\kappa_B^{P}(f) - \delta_B(f)\right|. \tag{3.5}$$

Another property of CHM is the following ordering relationship, i.e., assuming $P > 0$:

$$\kappa_B^{-P}(f) \leq \kappa_B^{P}(f). \tag{3.6}$$

However, the duality property, which is a fundamental property for erosion/dilation and for opening/closing, does not hold for the CHM filter, i.e., assuming $P > 0$:

$$\kappa_B^{P}(f) \neq \left(\kappa_B^{-P}(f^c)\right)^c. \tag{3.7}$$

In addition the (anit-)extensive, increasing and idempotence properties do also not hold; however, this should not be a limiting factor for robust approximation of erosion and dilation.

In Figure 3.1, we compare the CHM approximation quality for various values of P with erosion and dilation using a flat structuring element. Note that the flat structuring element for classical mathematical morphology is set to zero and the flat structuring element for CHM is set to one. In the case $P = 0$ we obtain a simple convolution and a low-pass filtering behaviour. For $|P| \gg 0$, the approximation quality to erosion and dilation is increasing. Because opening/closing are composed of erosion/dilation, we make similar observations for approximating opening/closing in Figure 3.2.

The authors using counter-harmonic mean to approximate erosion/dilation [1, 9] claim that good approximation quality of the operators can be achieved with $P = \pm 20$ and for their smoothing application $P = \pm 5$ is appropriate. In the learning framework proposed in [12], Masci et al. are able to learn erosions/dilations (no learned P values mentioned) and openings/closings ($5 \leq |P| < 10$) with the CHM learning framework. However, we rather observe noticeable smoothing behaviour, hence we aim to learn larger $|P|$ values.

To summarize, the counter-harmonic mean approximates *flat* erosion and *flat* dilation well enough for $|P| \gg 0$, where the morphological operator and the structuring element can be learned through the backpropagation algorithm. The properties of mathematical morphology do not hold for the CHM approximation which is not limiting in denoising applications [1, 9, 18]; however, the low-pass smoothing behaviour for P close to zero might be problematic for long morphological pipelines as higher frequency features get lost.

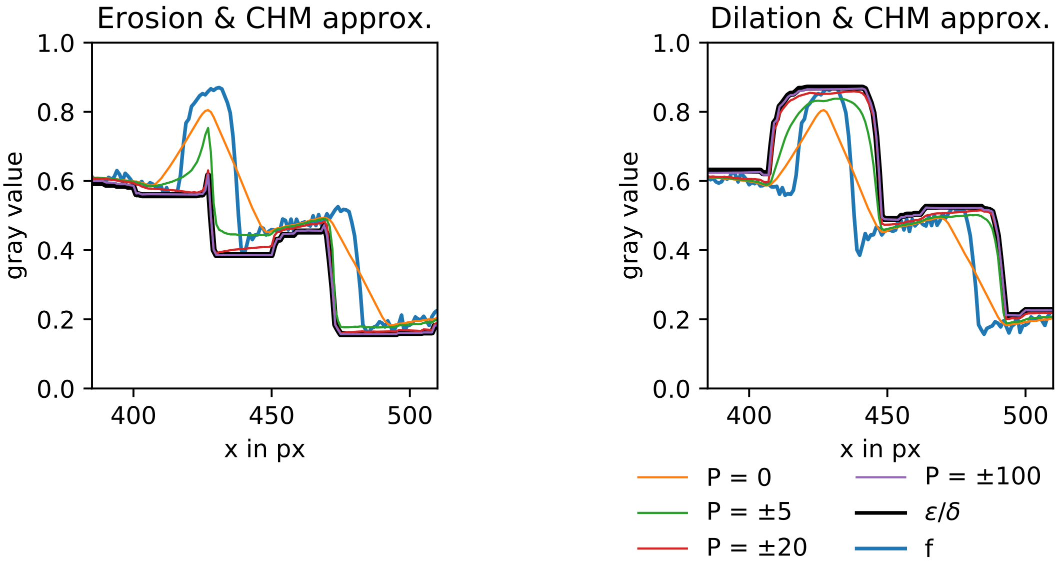


Figure 3.1: We use a magnified version of the same 1D-signal f (blue) as in the previous Figures 2.4, 2.5 and 2.6 with the same flat structuring element of size 21. In black, we show the classical erosion (left) and dilation (right), while the remaining graphs represent approximations with CHM. In case of erosion, P is negative and in case of dilation, P is positive. For $P = 0$ we obtain a regular convolution with low-pass like behaviour. For $|P| \gg 0$ approximation quality compared to flat erosion/dilation increases and for $P = \pm 100$ the approximation quality is close to classical erosion/dilation.

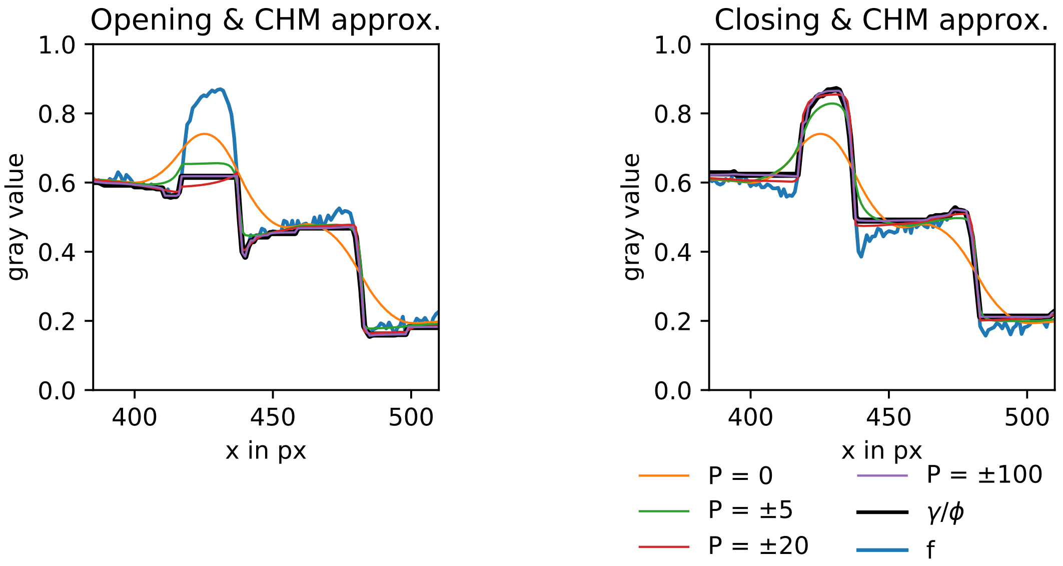


Figure 3.2: We use a zoomed in version of the same 1D-signal f (blue) as in the previous Figures 2.7, 2.8 and 2.9 with the same flat structuring element of size 21. In black, we show the proper opening (left) and closing (right), while the remaining graphs represent approximations with the counter-harmonic mean. In case of opening, a CHM erosion followed by a CHM dilation is applied and in case of closing vice versa. For $P = 0$ we obtain two regular convolutions with low-pass like behaviour. For $|P| \gg 0$ approximation quality compared to flat opening/closing increases and for $P = \pm 100$ the approximation quality is close to the proper opening/closing.

4 Method

The first part of this chapter focuses on learning morphological operators and structuring elements utilizing the counter-harmonic mean filter and a novel robust counter-harmonic mean filter. The second part focuses on image classification tasks utilizing the robust counter-harmonic mean filter as a feature extractor.

4.1 Learning Morphological Operators

The innovative work of the counter-harmonic mean learning framework [12] allows us to learn morphological operators and structuring elements simultaneously. However, the authors obtain rather smooth output images; therefore, we strongly focus on studying the learning behaviour of the counter-harmonic mean filter for erosions/dilations, openings/closings and top-hat transformations introduced in Section 2. Due to our findings, we develop a novel, more robust version of the counter-harmonic mean filter and compare both.

Robust Counter-Harmonic Mean

Learning of a counter-harmonic mean filter (3.1) is unstable depending on the target structuring element, mainly due to clipping negative $\mathbf{w}$ values because of the constraint $\mathbf{w} \geq 0$. The resulting output images are smoothed and contain unwanted artefacts.

Our main idea is to get rid of clipping negative values during learning due to the $\mathbf{w} \geq 0$ constraint of the counter-harmonic mean filter (CHM). We develop the robust counter-harmonic mean (RCHM) filter by restricting $\mathbf{w}$ to values between zero and one using the logistic sigmoid function σ and thus allow learning of arbitrary weight values:

$$\mathbf{w} = \sigma(\mathbf{w_r}) = \frac{1}{1+e^{-\mathbf{w_r}}} \qquad \sigma : \mathbb{R} \to [0,1], \quad \mathbf{w_r} \in \mathbb{R}, \quad \mathbf{w} \in [0,1]. \tag{4.1}$$

Inserting the modified structuring element of (4.1) into the counter-harmonic mean of (3.1) results in our robust version of the counter-harmonic mean filter:

$$\kappa^P_{\sigma(\mathbf{w_r})}(f)(\mathbf{x}) = \frac{\left(f^{P+1} * \sigma(\mathbf{w_r})\right)(\mathbf{x})}{\left(f^{P} * \sigma(\mathbf{w_r})\right)(\mathbf{x})}, \qquad \mathbf{w_r} \in \mathbb{R}. \tag{4.2}$$

The properties of the CHM filter in Section 3.2 also apply to the RCHM filter because the modified structuring element $\mathbf{w} = \sigma(\mathbf{w_r})$ is positive between zero and one. Note, $\sigma(0) = 0.5$, $\sigma(< 5) \approx 0$ and $\sigma(> 5) \approx 1$.

When mentioning the counter-harmonic mean filter (3.1) and the robust version (4.2) we always denote the structuring element with $\mathbf{w}$. In case we refer to robust counter-harmonic mean filter, we refer to the modified structuring element $\mathbf{w}$ in (4.1).

Loss Functions

For learning morphological operators two loss functions are used, the mean squared error (MSE) and the structural similarity (SSIM) [19].

The mean squared error is defined as:

$$\mathrm{MSE} = \frac{1}{MN} \sum_{i}^{M} \sum_{j}^{N} (f_{in}(i,j) - f_{out}(i,j))^2, \tag{4.3}$$

and takes the mean value of all pixelwise squared differences of the input image f_{in} and the output image f_{out}. Larger errors are penalized more than lower errors.

The structural similarity considers the luminance, contrast and structure of two image patches $\mathbf{x}$, $\mathbf{y}$ and is defined as:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}, \tag{4.4}$$

where μ_x denotes the mean value, σ_x the standard deviation of patch $\mathbf{x}$ and σ_{xy} denotes the covariance of both patches $\mathbf{x}$ and $\mathbf{y}$. C_1 and C_2 are constants for numerical stability. The SSIM is ≤ 1 and only if $\mathbf{x} = \mathbf{y}$ results in SSIM $= 1$.

Learning Erosion and Dilation

A single (robust) counter-harmonic mean filter is able to approximate an erosion and a dilation. Given an input image and a morphologically processed target image we learn an approximated morphological operator and the structuring element. An example for the input image and the dilated target image using a cross-shaped structuring element is shown in Figure 4.1.

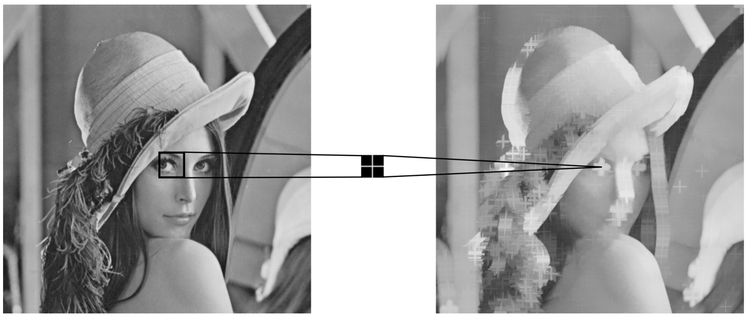


Figure 4.1: Illustration of an input image and a dilated target image. The cross-shaped structuring element and the dilation is learned using the (R)CHM.

Training is performed with the backpropagation algorithm using the Adam [7] optimizer with a learning rate of $\alpha = 0.005$ and the exponential decay rates of $\beta_1 = 0.9, \beta_2 = 0.999$. In each iteration, parameters P and $\mathbf{w}$ are updated simultaneously. Given dilated target images we initialize $P = 1$ and for eroded target images $P = -1$. We compare the experiments with two loss functions, namely the mean squared error (MSE) and the structural similarity (SSIM)[19] introduced above. All input images are of size 256×256, all structuring elements are of size 11×11 and the resulting approximated morphological output images are of size 246×246 because no padding is used. We do not use any activation function or bias term.

All structuring element values are initialized to $\mathbf{w} = \mathbf{0.5}$ in experiments using the counter-harmonic mean filter (3.1) and as the structuring element

$\mathbf{w}$ is constrained to $\mathbf{w} \geq 0$, all $w_{i,j} < 0$ are clipped to $w_{i,j} = 0$ after each optimization step if required. Using the robust counter-harmonic mean filter (4.2) all structuring element values in $\mathbf{w}$ are also initialized to $\mathbf{w} = \mathbf{0.5}$ by setting the parameters $\mathbf{w_r}$ to $\mathbf{w_r} = \mathbf{0}$ which results in $\mathbf{w} = \sigma(\mathbf{0}) = \mathbf{0.5}$. The robust filter does not require any clipping during learning.

Note, that we outperform the proposed method in [12] using stochastic gradient descent (SGD) with alternate updates of $\mathbf{w}$ and P in initial experiments using Adam and simultaneous updates of $\mathbf{w}$ and P and, therefore, we omit SGD and alternate updates.

Learning Opening and Closing

Two (R)CHM filters in sequence are able to approximate openings and closings. Given an input image and a morphologically processed target image we learn both morphological operators and both structuring elements. An example for an opening using a disc-shaped structuring element is shown in Figure 4.2 where the input image is eroded and the eroded image is dilated.

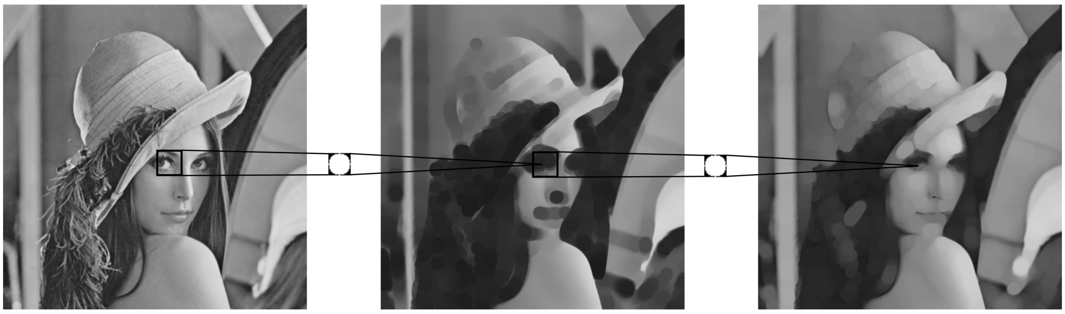


Figure 4.2: Illustration of a morphological opening. The input image is eroded with a disc-shaped structuring element and the eroded image is dilated with the same disc-shaped structuring element.

Learning openings/closings is performed as learning erosions/dilations discussed previously with the following changes: We use two Adam optimizers for learning RCHM filters, one optimizer updates P with a learning rate of $\alpha = 0.05$ and the second optimizer updates w with a learning rate of $\alpha = 0.005$. The target images are of size 236×236 because no padding is used in the additional second layer.

Learning Top-Hats

Learning openings and closings using two (R)CHM filters enables learning of top-hat transformations. A white top-hat transformation is shown in Figure 4.3 using the opening from Figure 4.2.

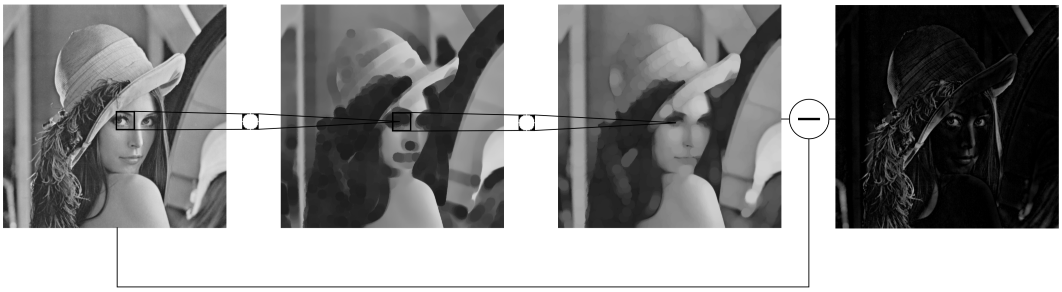


Figure 4.3: The white top-hat transformation is the difference of the input image with the opened input image.

The white top-hat (2.21) and the black top-hat (2.22) transformations differ in the ordering of the input image term and the opening/closing terms. The absolute difference layer, introduced in [12], generalizes the ordering for white and black top-hats and, therefore, allows us to learn both top-hat transformations with the same architecture consisting of two (R)CHM filters followed by the absolute difference with the input image.

$$\begin{aligned} WTH(f) &= f - \gamma(f) = |f - \gamma(f)| = |\gamma(f) - f|, \\ BTH(f) &= \phi(f) - f = |f - \phi(f)| = |\phi(f) - f| \end{aligned}$$

Given an input image and a top-hat transformed target image we learn both, morphological operators and the structuring elements.

Learning top-hat transformations is performed as learning openings/closings described previously and including the absolute difference layer. In case of CHM the learning rate is lowered to $\alpha = 0.0005$ and in case of RCHM the learning rates are lowered to $\alpha = 0.005$ for updating P and $\alpha = 0.001$ for updating $\mathbf{w}$.

Structuring Elements and Dataset Generation

To validate the approximation quality of the learned morphological operators we need to generate our own meaningful dataset with erosion/dilation, opening/closing and the top-hat transformations utilizing structuring elements containing various structures. The target images are generated with classical mathematical morphology (Section 2) and additionally our approach also includes target images generated with the counter-harmonic mean filter (Section 3.2) with varying approximation quality. First, we discuss the used structuring elements and finally we discuss the target image generation.

In the (robust) counter-harmonic mean filter the flat structuring element w is predefined to a squared grid window with a fixed size of 11×11 pixels. We define a new important property for measuring the structure size of a structuring element in such a squared window, namely the fullness property, which takes the ratio of occupied pixels to total pixels. In contrast to a full structuring element, which occupies all pixels in the window (a square of size 11×11), a non-full structuring element does not occupy all pixels in the squared window (a disk, a cross, a diagonal, a square smaller than 11×11, etc.). We utilize four different structuring elements with varying fullness throughout our experiments which are shown in Figure 4.4 and Table 4.1.

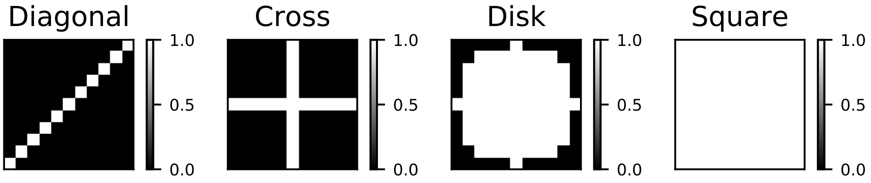


Figure 4.4: The diagonal-, cross-, disk- and square-shaped structuring elements. Pixels with value zero indicate that the pixels are not part of the structuring element and pixels with value greater than zero, in our case one, indicate that the pixels are part of the structuring element.

The authors of the counter-harmonic mean learning framework [12] did not mention any learned P values for erosion and dilation, but looking at their resulting images we observe smoothing behaviour which indicates rather low

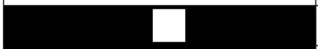
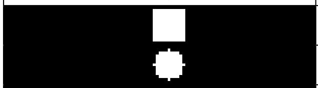
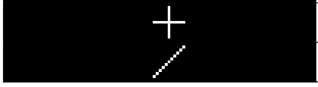
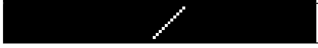
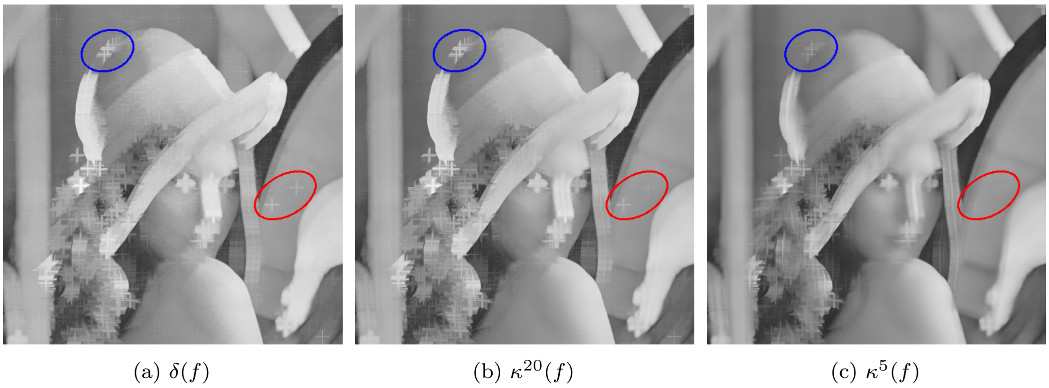
Structuring Element	Window Size	Occupied Pixels	Fullness
	11 × 11	121	100%
	11 × 11	81	67%
	11 × 11	21	17%
	11 × 11	11	9%

Table 4.1: Only the square is a full structuring element; the disk, cross and diagonal are not full structuring elements as not all pixels in the squared window are occupied. Fewer occupied pixels result in lower fullness.

learned $|P|$ values (see Section 3.2 on smoothing behaviour using the counter-harmonic mean). They reported learned P values for opening and closing in the range of $5 \leq |P| \leq 10$ and as opening and closing are compositions of erosion and dilation, we infer that also rather low $|P|$ values were learned for erosion/dilation. Learning large $|P|$ values, which result in a more accurate approximation of erosion/dilation and therefore less smoothing, apparently is a hard problem. For this reason, our method includes target images generated with counter-harmonic mean filters with various P values in addition to the classical morphological target images and we expect to learn the counter-harmonic mean target images much easier, especially target images generated with lower $|P|$ values. Our counter-harmonic mean target images are generated with $P = \pm 5, \pm 10, \pm 20, \pm 30$ using the structuring elements shown in Figure 4.4 and Table 4.1. Learning the counter-harmonic mean target images for $P = \pm 5$ and $P = \pm 10$ is our minimum requirement to reproduce the results in [12]. With $P = \pm 20$ and $P = \pm 30$ the counter-harmonic mean target images are much more similar to the classical morphological operators; hence, learning such $|P|$ values is desirable and our favourable goal. A comparison between classical morphological dilation and counter-harmonic mean approximations using $P = 20$ and $P = 5$ is shown in Figure 4.5 where we observe that the classical dilation target image contains more structure and detail than the target images generated with counter-harmonic mean using rather small P values.

All experiments in Section 5.1 use the same input image of size 256 × 256 with 11 × 11 structuring elements shown in Table 4.1. We show relevant examples explaining success and failure cases in the experiments section.



(a) $\delta(f)$

(b) $\kappa^{20}(f)$

(c) $\kappa^{5}(f)$

Figure 4.5: Target image comparison between classical dilation (a) and counter-harmonic mean with $P = 20$ (b) and $P = 5$ (c) where noticeable differences are shown in the red and blue ellipses. For $P = 20$ the cross-shaped structuring element does not appear as crisp as in the classical dilation, especially apparent in the red area, and for $P = 5$ we additionally notice smoothing behaviour of the overall image.

4.2 Vision Task

This section covers learning morphological neural networks using robust counter-harmonic mean filters within the networks. We study if morphological neural networks are able to extract meaningful features for image classification and whether morphological operators or convolutions are learned. In the previous section each layer consists of a single (R)CHM filter which is not sufficient for image classification. For this reason, we develop and compare two types of pseudo-morphological layers consisting of multiple RCHM filters.

Datasets

We use the MNIST [14] dataset consisting of handwritten digits ranging from digit zero to digit nine and the Fashion-MNIST [20] dataset consisting of ten different clothing items. Both datasets consist of grayscale images of size 28×28 and are split into 60000 training images and 10000 testing images. Fashion-MNIST is a more difficult alternative for benchmarking machine learning algorithms than the widely used regular MNIST.

Morphological Layers

The underlying convolution operation in a CNN is well suited for pattern matching where a single convolutional kernel sums up the spatial and depthwise convolutions to a single channel in the subsequent layer. However, erosions/dilations are not comparable to such pattern matching mechanisms. Thus, the convolutions in a CNN can not simply be swapped with RCHM filters. For this reason, we introduce two types of pseudo morphological layers which are inspired by depthwise separable convolutions [3].

Our MorphGroup (Figure 4.6) and MorphAll (Figure 4.7) layers pseudomorphologically process each input channel independently using the RCHM followed by 1 × 1 convolutions which combine the morphologically processed images. The MorphGroup and MorphAll layers differ in the depthwise 1 × 1 convolutions, the former performs a grouped depthwise convolution and the latter performs a depthwise convolution over all morphologically processed images.

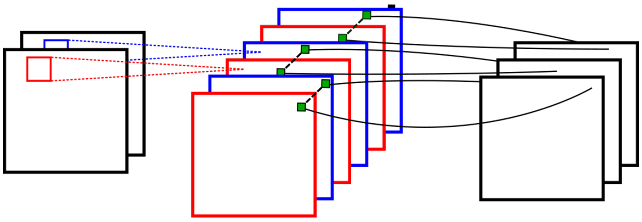


Figure 4.6: MorphGroup layer using two input channels and three output channels. Each input channel is morphologically processed independently three times using RCHM. Three 1 × 1 convolutions combine the morphologically processed images in groups of two, resulting in three output channels. The 1 × 1 convolutions require $c_{in} \cdot c_{out}$ parameters. A bias term is added for each output channel. Note the similarity to a convolutional layer, where instead of the 1 × 1 convolutions, summations are performed.

Network Architecture and Training

As the datasets have the same input size and the same amount of classes we use one general network architecture shown in Figure 4.8. In each layer, $L1$ to $L4$, the filter size is fixed to 7 × 7 and the number of output channels is set to

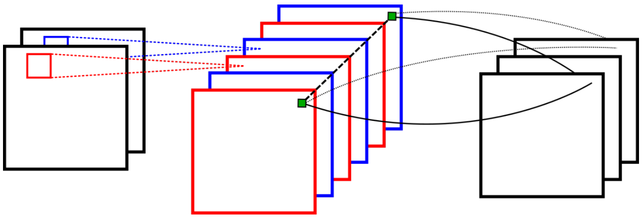


Figure 4.7: MorphAll layer using two input channels and three output channels. Each input channel is morphologically processed independently three times using RCHM. Three 1 × 1 convolutions combine all six morphologically processed images to three output channels. The 1 × 1 convolutions require $c_{in} \cdot c_{out}^2$ parameters, which is by a factor c_{out} more than MorphGroup. A bias term is added for each output channel.

8. A single activation in the last RCHM layer $L4$ is of size 4×4 because no padding is used. The flattened layer $L4$ output is the feature vector of length 128 which is then fully connected to 10 output neurons. Remember, (R)CHM is defined for images between zero and one. We clamp the input images to $[10^{-5}, 1 - 10^{-5}]$, otherwise divisions by zero may occur. For the same reason, we utilize sigmoid activation functions after each layer.

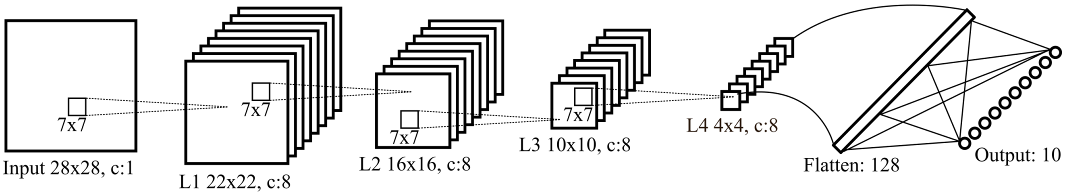


Figure 4.8: Proposed morphological network architecture for classifying the MNIST and Fashion-MNIST datasets.

We compare the performance of both morphological layer approaches for:

1. Pure MorphGroup layers in $L1, L2, L3, L4$
2. Pure MorphAll layers in $L1, L2, L3, L4$

As the sigmoid functions in the RCHM and the activation functions may output very small values, which in combination with $|P| \gg 0$ may cause numerical problems, we use `double` precision and clip all P at $P = \pm 50$ and the $\mathbf{w_r}$ parameters to $\mathbf{w_r} = \pm 100$ where required.

The $\mathbf{w_r}$ parameters of the structuring elements, P and the depthwise 1×1 convolutions of the MorphGroup are initialized uniformly between $[-1, 1]$. The depthwise 1×1 convolutions in the MorphAll layer are initialized uniformly between $[-6, 6]$, which accelerates learning in early epochs.

All parameters are updated using the Adam optimizer with a learning rate of $\alpha = 0.001$ and the exponential decay rates of $\beta_1 = 0.9, \beta_2 = 0.999$ based on the negative log likelihood loss. The depthwise 1×1 convolutions and the fully connected layer weights are L_2-regularized with a weight of 10^{-5}, no regularization is performed on P, $\mathbf{w_r}$ of the RCHM filters and bias terms.

5 Experiments

The first section shows the experimental results of learning predetermined morphological operators and structuring elements utilizing the counter-harmonic mean. Based on the drawbacks of the counter-harmonic mean we show the need for the robust counter-harmonic mean. In the second section a morphological corner detector is presented. The third section focuses on image classification tasks where we use predefined datasets and use the (robust) counter-harmonic mean as a feature extractor. No predetermined morphological operators are given and the network has to learn appropriate morphological operators from scratch.

5.1 Learning Morphological Operators

Our goal is to learn the morphological operators and structuring elements given an input image and the predetermined morphologically processed target image. Before applying the counter-harmonic mean filter to a larger network architecture we strongly focus on analysing the learning behaviour of the counter-harmonic mean. Due to our findings we develop a novel more robust version of the counter-harmonic mean and compare it to the counter-harmonic mean.

The numbers of training iterations are set according to the convergence of the loss. We require far fewer iterations using the robust counter-harmonic mean over using the counter-harmonic mean. Thus, we do not train for equal numbers of iterations for erosions/dilations, openings/closings and using `float/double` precision. Only the top-hat transformations are trained with equal numbers of iterations.

5.1.1 Learning Erosion and Dilation

Erosion and dilation are the basic and most fundamental morphological operators, so analysing and understanding the learning mechanism of erosion and dilation is important for more complex problems like opening/closing and top-hat transformations. First, we show that the counter-harmonic mean enables learning morphological operators with only a single filter. However, we find that the mathematical and numerical constraints of the counter-harmonic mean cause unstable learning for larger $|P|$ values and non-full structuring elements. Finally, we introduce our novel robust counter-harmonic mean filter and compare it to the counter-harmonic mean filter.

Learning with Counter-Harmonic Mean
The experimental results for learning counter-harmonic mean filters are shown in Table 5.1 with generated target images using $P = \pm 5, \pm 10, \pm 20, \pm 30$ and the classical morphological operators erosion ε/dilation δ and all full/non-full structuring elements from Figure 4.4. Only the best results over all iterations are shown and are based on the lowest MSE and highest SSIM with the corresponding learned P_{MSE} and P_{SSIM} values. We observe that learning target images generated with $P = \pm 5, \pm 10$ is an easily achievable task resulting in low MSE and high SSIM and also reaching the desired target P. For $P = \pm 20$ we observe that learning is harder, in terms of loss and target P value for non-full structuring elements. The learned P values using the SSIM loss are closer to the target P. Learning erosion/dilation using the full square-shaped structuring element is an easily achievable task. In general, learning dilations performs better than learning erosions. For target $P = \pm 30$ and erosion ε $(P \to -\infty)$/dilation δ $(P \to \infty)$ the learned P values for non-full structuring elements do not improve over target $P = \pm 20$ and also the loss is getting worse because the target images for $P = \pm 5, \pm 10, \pm 20$ are smoother than for $P = \pm 30$ and ε/δ. To summarize, learning desired large $|P|$ values is harder for structuring elements with lower fullness and additionally learning dilations is easier than learning erosions.

The corresponding learned structuring elements from the experiments in Table 5.1 are shown in Figure 5.1 for learning dilations based on the SSIM loss. Good results are achieved by targeting $P = 5, 10, 20$. Structuring elements with lower fullness and large P tend to learn very large $w_{i,j} \gg 0$ values, especially

		Erosion					Dilation				
SE	Fullness	Target	MSE	P_{MSE}	$SSIM$	P_{SSIM}	Target	MSE	P_{MSE}	$SSIM$	P_{SSIM}
⁄	9%	κ^{-5}	$2.09 \cdot 10^{-15}$	−5.0	1.0000	−5.0	κ^{5}	$3.12 \cdot 10^{-15}$	5.0	1.0000	5.0
+	17%		$3.09 \cdot 10^{-15}$	−5.0	1.0000	−5.0		$4.53 \cdot 10^{-15}$	5.0	1.0000	5.0
●	67%		$9.30 \cdot 10^{-15}$	−5.0	1.0000	−5.0		$1.40 \cdot 10^{-14}$	5.0	1.0000	5.0
■	100%		$1.34 \cdot 10^{-14}$	−5.0	1.0000	−5.0		$2.20 \cdot 10^{-14}$	5.0	1.0000	5.0
⁄	9%	κ^{-10}	$2.04 \cdot 10^{-15}$	−10.0	1.0000	−10.0	κ^{10}	$3.59 \cdot 10^{-15}$	10.0	1.0000	10.0
+	17%		$3.06 \cdot 10^{-15}$	−10.0	1.0000	−10.0		$5.16 \cdot 10^{-15}$	10.0	1.0000	10.0
●	67%		$9.01 \cdot 10^{-15}$	−10.0	1.0000	−10.0		$1.62 \cdot 10^{-14}$	10.0	1.0000	10.0
■	100%		$1.29 \cdot 10^{-14}$	−10.0	1.0000	−10.0		$2.66 \cdot 10^{-14}$	10.0	1.0000	10.0
⁄	9%	κ^{-20}	$4.70 \cdot 10^{-5}$	−12.7	0.9985	−14.3	κ^{20}	$2.07 \cdot 10^{-5}$	14.7	0.9999	18.5
+	17%		$5.40 \cdot 10^{-5}$	−13.3	0.9978	−14.9		$9.41 \cdot 10^{-6}$	16.7	0.9999	19.7
●	67%		$3.93 \cdot 10^{-5}$	−14.5	0.9988	−16.1		$3.41 \cdot 10^{-6}$	18.4	1.0000	19.9
■	100%		$1.31 \cdot 10^{-14}$	−20.0	1.0000	−20.0		$2.49 \cdot 10^{-14}$	20.0	1.0000	20.0
⁄	9%	κ^{-30}	$1.40 \cdot 10^{-4}$	−12.3	0.9945	−14.1	κ^{30}	$8.53 \cdot 10^{-5}$	14.5	0.9984	19.4
+	17%		$1.76 \cdot 10^{-4}$	−13.2	0.9906	−14.7		$7.39 \cdot 10^{-5}$	16.7	0.9980	20.7
●	67%		$1.63 \cdot 10^{-4}$	−15.1	0.9933	−17.2		$6.06 \cdot 10^{-5}$	19.9	0.9997	26.3
■	100%		$2.62 \cdot 10^{-6}$	−27.0	0.9997	−26.7		$2.34 \cdot 10^{-14}$	30.0	1.0000	30.0
⁄	9%	ε $P \to -\infty$	$3.95 \cdot 10^{-4}$	−11.6	0.9885	−12.9	δ $P \to \infty$	$3.35 \cdot 10^{-4}$	14.0	0.9923	18.9
+	17%		$5.19 \cdot 10^{-4}$	−13.1	0.9830	−14.4		$4.11 \cdot 10^{-4}$	16.2	0.9883	20.3
●	67%		$6.12 \cdot 10^{-4}$	−17.6	0.9829	−16.7		$5.82 \cdot 10^{-4}$	24.6	0.9886	22.4
■	100%		$2.80 \cdot 10^{-4}$	−26.9	0.9911	−24.2		$2.42 \cdot 10^{-4}$	38.3	0.9938	37.4

Table 5.1: Experimental results for learning erosions and dilations using the counter-harmonic mean filter with `float` precision. Only the best results out of 800000 training iterations are shown including the corresponding learned P values.

compared to the full square-shaped structuring element. In addition, black pixels are either very close to zero $w_{k,l} \gtrapprox 0$ or exactly $w_{m,n} = 0$ because of clipping during optimization and caused by the $\mathbf{w} \geq 0$ constraint of the counter-harmonic mean filter. For target $P = 30$ and the classical dilation, learning larger values on the outside and edges of the structuring elements are preferred and the values in the center are lower. The learned **w** give the impression that they are non flat structuring elements. However, remember that CHM only approximates flat erosion/dilation and the learned structuring elements are approximately flat structuring elements when applied to (3.2) discussed in Section 3.2.

We found two main problems using the counter-harmonic mean filter, one for learning full structuring elements and another learning non-full structuring elements. In Figure 5.2, we show the learning behaviour of parameters P and **w** for the first 30000 iterations targeting full structuring elements. In case of targeting $P = \pm 5, \pm 10, \pm 20$ and $P = 30$ with `float` precision, the $\min(\mathbf{w})/\max(\mathbf{w})$ values converge to similar values of ~ 0.5 forming a full structuring element. In case of learning a classical dilation, a `float` underflow occurs for a learned $P = 37.4$ and can be improved by using `double` precision with an underflow occurring for $P = 271.1$. Targeting the classical erosion and $P = -30$ using `float` precision are numerically unstable and can be solved by using `double` precision. Using `double` enables to learn the desired $P = -30$ and in case of the classical erosion the best performance is reached for $P = -199.5$. In general, learning full structuring elements is only limited by `float` and `double` precision and unstable learning can be solved by clipping P to appropriate values depending on the precision. In Figure 5.3, we show the learning behaviour for targeting non-full cross-shaped structuring elements and we observe no improvement using `double` precision over `float` precision, because the P values do not get large enough to notice numerical problems. Targeting $|P| >= 20$ is much harder for non-full structuring elements than for the full square-shaped structuring element. As already shown in Figure 5.1, large $\max(\mathbf{w})$ values are learned with larger $|P| \gg 0$ for non-full structuring elements. During optimization, negative **w** values are clipped to zero due to the $\mathbf{w} \geq 0$ constraint. The combination of $w_{i,j} = 0$, values close to zero $w_{k,l} \gtrapprox 0$, large values $w_{m,n} \gg 0$ and a large $|P| \gg 0$ value within the same structuring element may introduce unwanted artefacts in the output images making learning of non-full structuring elements hard. We show such failure cases later. In Table 5.2 the

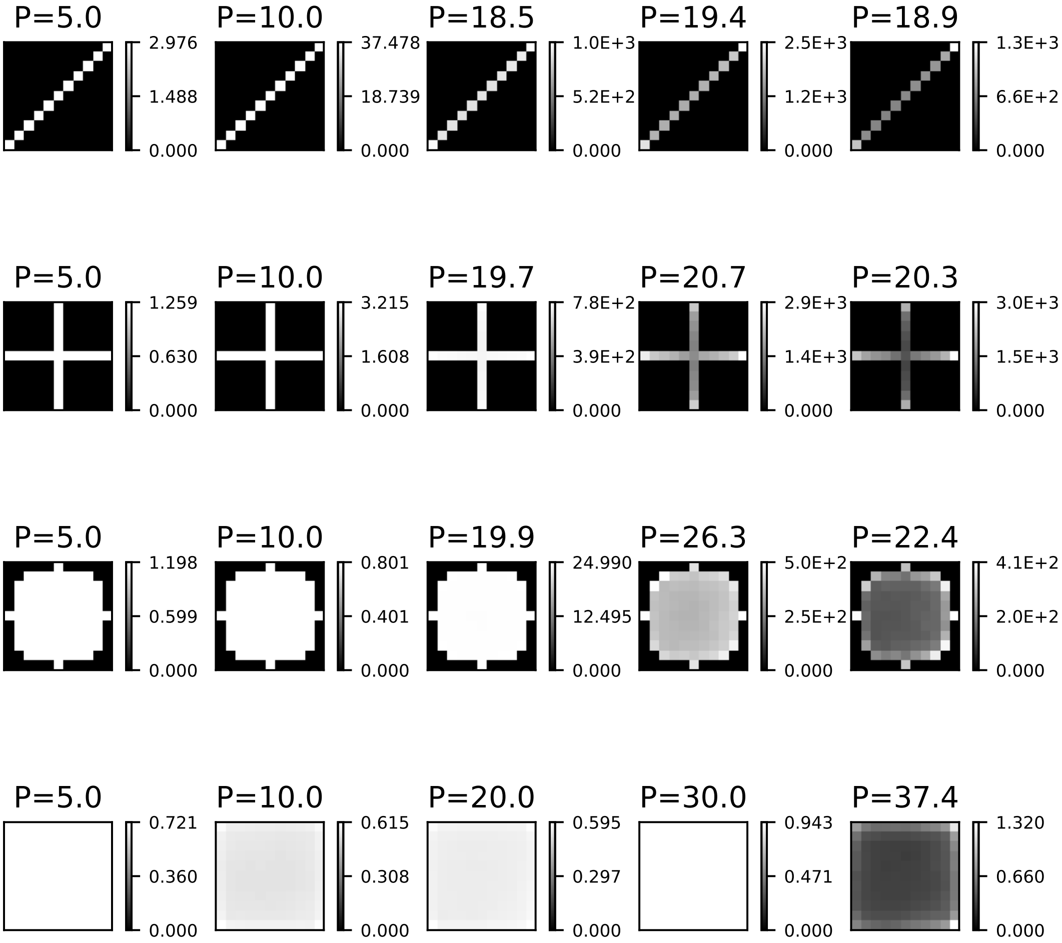


Figure 5.1: Learned structuring elements for dilations using SSIM from experiments in Table 5.1. The target P values are column wise in ascending order $P = 5, 10, 20, 30$ and in the last column the classical dilation δ $(P \to 0)$. Zero is visualized in black and $\max(\mathbf{w})$ in white.

best results of learning cross-shaped and square-shaped structuring elements using `double` precision are shown. As already mentioned, we observe that the full square-shaped structuring element benefits from using `double` precision as opposed to the cross-shaped structuring element having no improvement as shown in Table 5.1.

		Erosion			Dilation		
SE	Fullness	Target	$SSIM$	P_{SSIM}	Target	$SSIM$	P_{SSIM}
+	17%	κ^{-30}	0.9890	-14.0	κ^{30}	0.9976	20.0
□	100%		0.9999	-30.0		0.9999	30.0
+	17%	ε $P \to -\infty$	0.9817	-13.7	δ $P \to \infty$	0.9876	19.4
□	100%		0.9998	-199.5		0.9998	271.1

Table 5.2: Experimental results for learning erosions and dilations using the counter-harmonic mean filter with `double` precision. Only the best results out of 300000 training iterations are shown including the corresponding learned P values.

We compare learned structuring elements which produce successful and failure cases in the output images. First, we show the full square-shaped structuring elements at two different iterations in Figure 5.4 (a) and their resulting output images in Figure 5.5. At iteration 10500 the learned square-shaped structuring element and the corresponding $P = -24.5$ results in a structural similarity above 0.999 and outputs a visually pleasing image shown in Figure 5.5 (b). However, the structural similarity drops to 0.93 at iteration 24000 for a corresponding $P = -31.4$ as the structuring element is not a full square-shaped one anymore which is caused by numerical problems using `float` precision. The resulting output image is shown in Figure 5.5 (c) where we can see that the corrupted structuring element appears. As already mentioned, learning full structuring elements benefits from using `double` precision and results in exceptional approximations of classical erosion/dilation shown in Figure 5.7. Second, we show the non-full cross-shaped structuring elements in Figure 5.4 (b) at two iterations, both are very similar and both learned P values are $P = -12.7$. The structural similarity oscillates around 0.981 ±0.05 between iterations 63000 and 64500, but the output images have noticeable differences as we show in Figure 5.6 (b) and (c). In general, we found that the combination of $w_{i,j} = 0$, $w_{k,l} \gtrapprox 0$ and $w_{m,n} \gg 0$ within the same structuring element and a large exponent $|P| \gg 0$ causes those artefacts due to the $\mathbf{w} \geq 0$ constraint. For smaller $|P| < 10$ the combination of $w_{i,j} = 0$, $w_{k,l} \gtrapprox 0$ and $w_{m,n} \approx 1.5$ does not cause any problems. During training

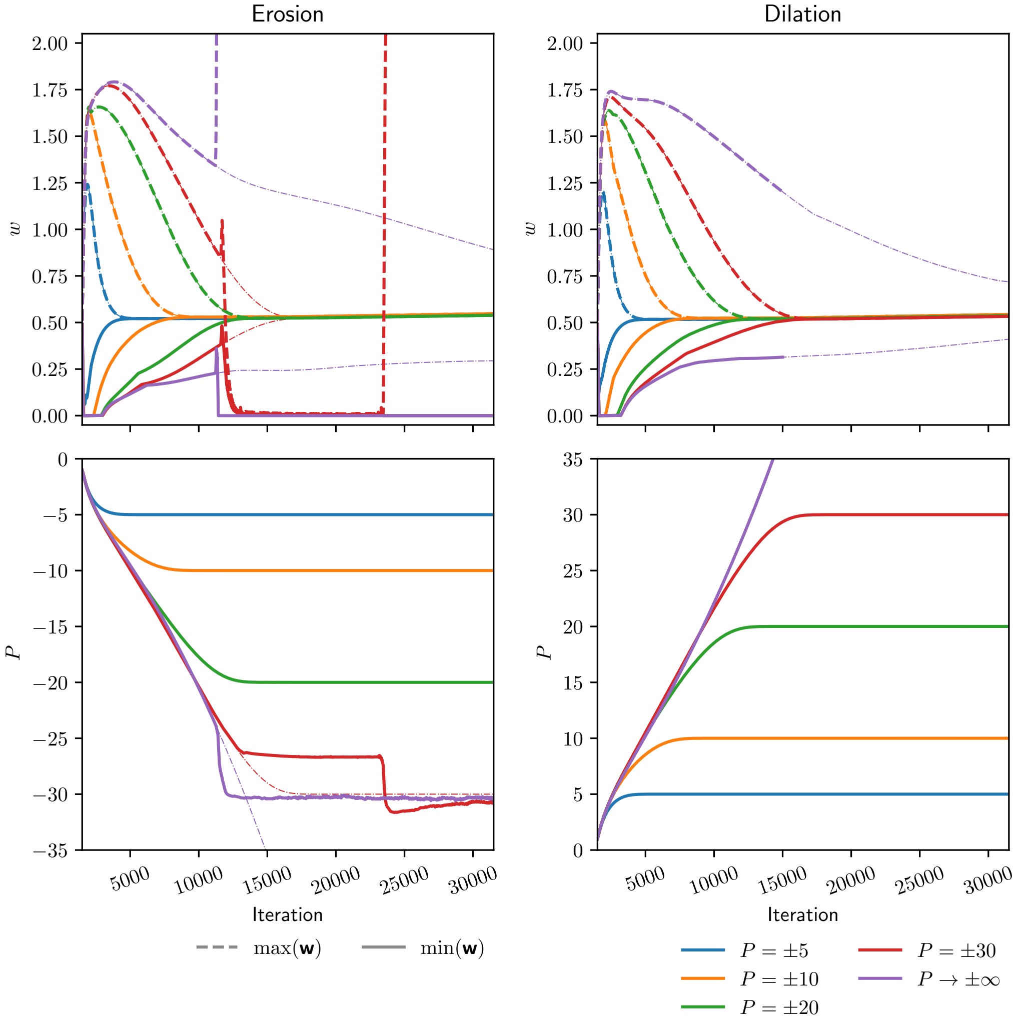


Figure 5.2: Each color represents the parameters P and $\mathbf{w}$ of a single counter-harmonic mean filter during learning of the *full square-shaped* structuring element using the SSIM loss. The left column shows the learning behaviour of erosions and the right column of dilations. In the top plots we show the maxima and minima of the structuring elements $\mathbf{w}$ and in the bottom the corresponding P values for the first 30000 iterations. Bold lines visualize values using `float` precision and the continuing dash-dotted lines visualize values using `double` precision.

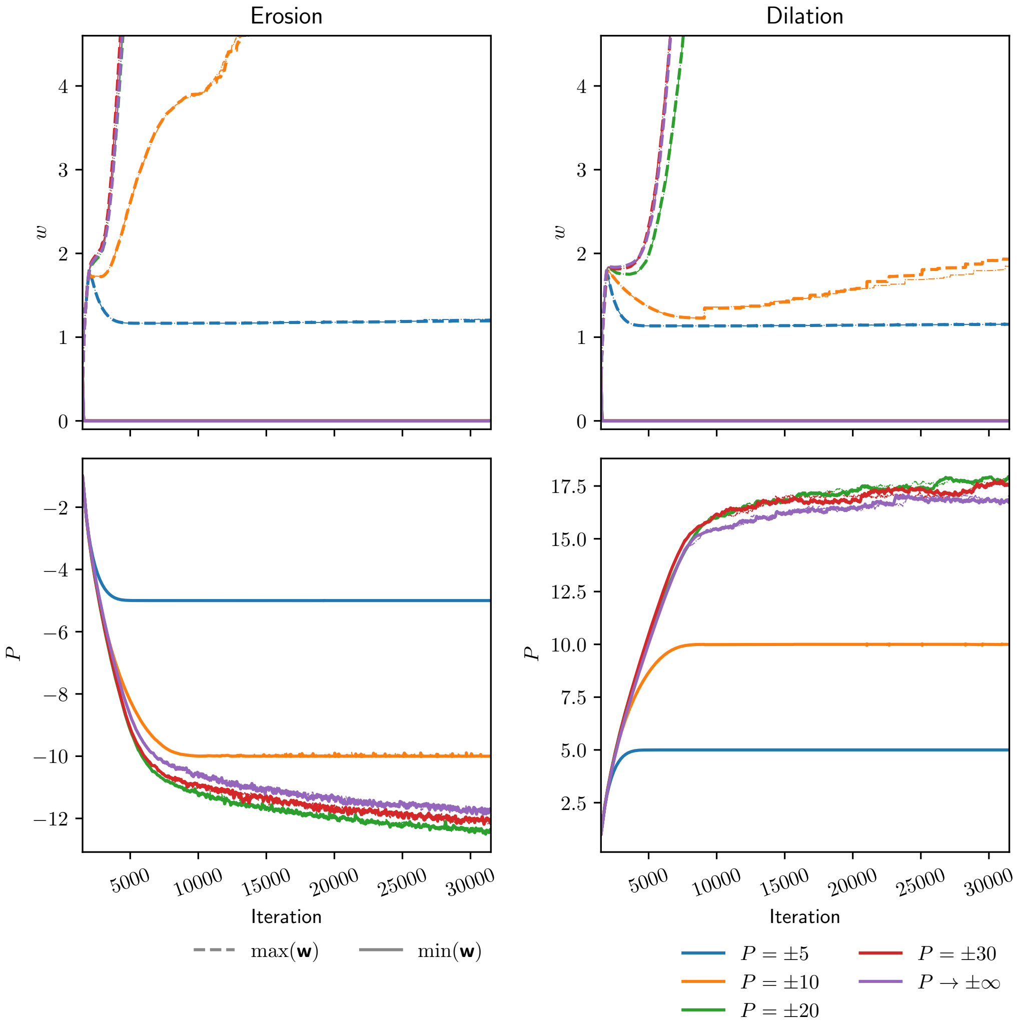
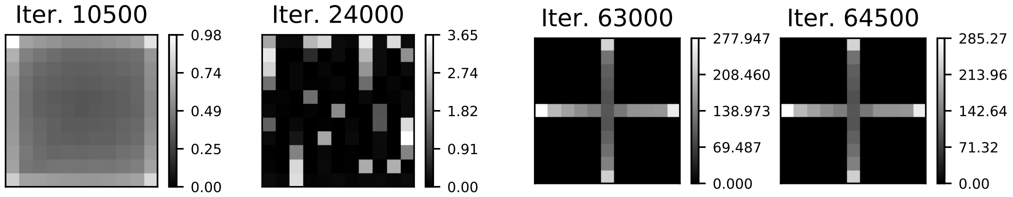


Figure 5.3: Each color represents the parameters P and $\mathbf{w}$ of a single counter-harmonic mean filter during learning of the *non-full cross-shaped* structuring element using the SSIM loss. The left column shows the learning behaviour of erosions and the right column of dilations. In the top plots we show the maxima and minima of the structuring elements $\mathbf{w}$ and in the bottom the corresponding P values for the first 30000 iterations. Bold lines visualize values using `float` precision and the continuing dash-dotted lines visualize values using `double` precision.

we observe that for $|P| > 10$ the P values start to oscillate from the time the first structuring element value is clipped to zero and therefore struggles to learn desired larger $|P|$ values. Note that all other non-full structuring elements, especially with lower fullness, have the same issues as the non-full cross-shaped structuring elements. As non-full structuring elements do not reach large $|P|$ values, `double` precision does not solve the problem of undesired artefacts in the output images; hence, the output images look the same as using `float`.



(a)

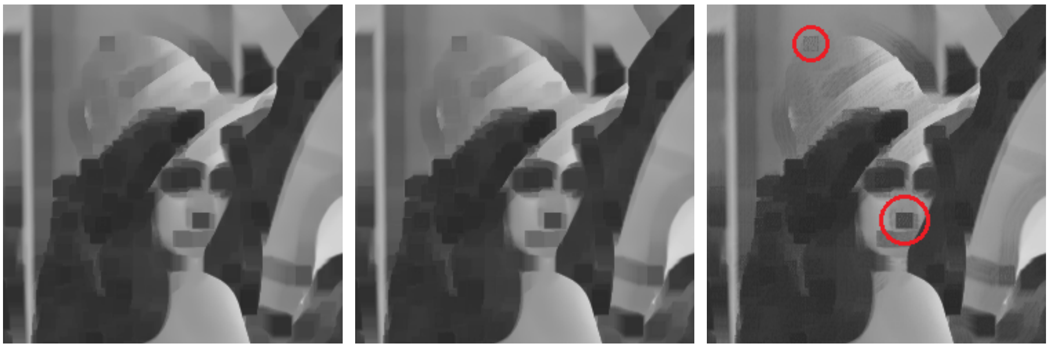
(b)

Figure 5.4: The learned structuring elements from Figures 5.2, 5.3 using `float` precision targeting $P = -30$ are shown for different iterations. In (a) the full square-shaped structuring elements and in (b) the non-full cross-shaped structuring elements. The structuring elements are scaled between 0 and $\max(\mathbf{w})$ for visualization purpose.

To conclude, learning approximations of erosions/dilations is possible using the counter-harmonic mean filter. However, learning of non-full structuring elements is a hard problem caused by the $\mathbf{w} \geq 0$ constraint making learning unstable and additionally undesired artefacts may appear in the output images. Using `double` precision only improves the performance of learning full structuring elements.

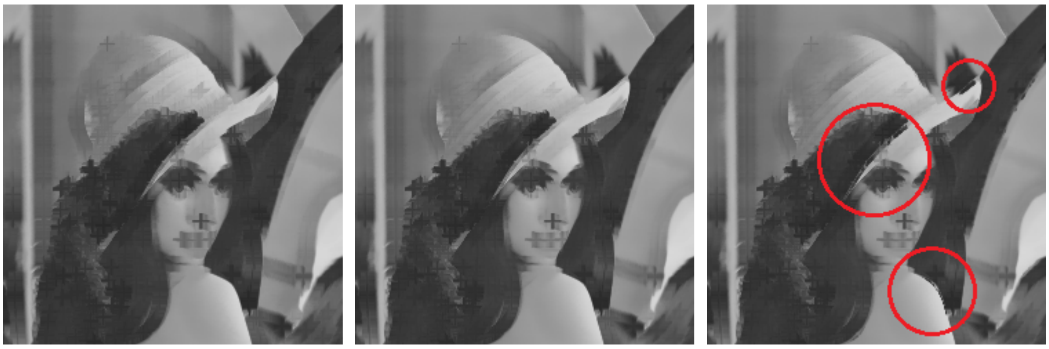
Learning with Robust Counter-Harmonic Mean

Learning counter-harmonic mean filters (CHM), introduced in (3.1), for $|P| \gg 0$ does not work properly for non-full structuring elements because of clipping due to the $\mathbf{w} \geq 0$ constraint we discussed previously. Our main idea is to avoid clipping using the robust counter-harmonic mean (RCHM) introduced in (4.2) and learn positive and negative values $\mathbf{w_r}$ and scaling them with the logistic sigmoid function between zero and one resulting in a positive structuring element $\mathbf{w} = \sigma(\mathbf{w_r})$.



(a) Target image κ^{-30} (b) Iter. 10500, $P = -24.5$ (c) Iter. 24000, $P = -31.4$

Figure 5.5: The counter-harmonic mean target image generated with a square-shaped structuring element with $P = -30$ is shown in (a). The output image for epoch 10500 with learned $P = -24.5$ is shown in (b) and the output image for epoch 24000 with learned $P = -31.4$ is shown in (c). In (c), the learned structuring element is not a full square-shaped one as shown in Figure 5.4 (a) and causes undesired output as we can see at the top of the hat and the nose indicated with red circles.



(a) Target image κ^{-30} (b) Iter. 63000, $P = -12.7$ (c) Iter. 64500, $P = -12.7$

Figure 5.6: The counter-harmonic mean target image generated with a cross-shaped structuring element with $P = -30$ is shown in (a). The output image in iteration 63000 with learned $P = -12.7$ is shown in (b) and the output image in iteration 64500 with a similar learned $P = -12.7$ is shown in (c). Unwanted artefacts appear in the output image in (c) caused by the learned structuring element and the unwanted artefacts are indicated with red circles.

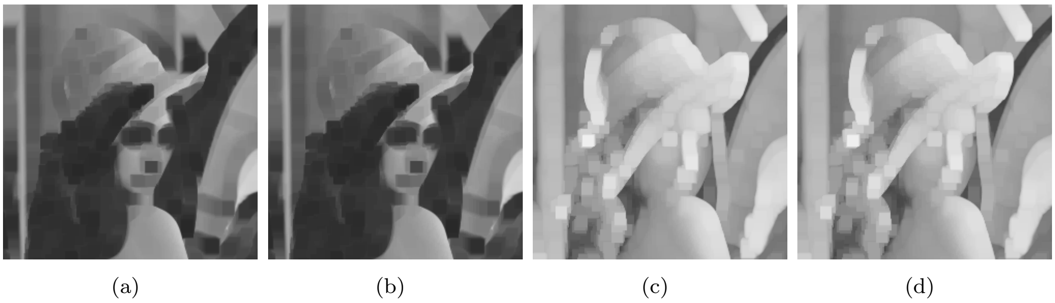


Figure 5.7: The classical erosion and dilation target images are shown in (a) and (c), and the learned counter-harmonic mean filters output the eroded image in (b) with $P = -199.5$ and the dilated image in (d) with $P = 271.1$.

The experimental results using the RCHM filter are shown in Table 5.3 and we observe that learning of non-full structuring elements clearly improved over using the CHM filter shown in Table 5.1. Note that target images are still generated with CHM. The RCHM filter enables us to learn much larger target P values. In addition, the RCHM required far fewer training iterations. However, learning non-full structuring elements now also causes numerical problems like learning full square-shaped structuring elements using the CHM with `float` precision. Again, only the best results in Table 5.3 are shown and do not show how stable learning using RCHM filters is. For this reason, we show the learning behaviour of parameters P and $\mathbf{w}$ $(= \sigma(\mathbf{w_r}))$ using `float` precision and `double` precision using full square-shaped structuring elements in Figure 5.8 and using non-full cross-shaped structuring elements in Figure 5.9. For the full square-shaped structuring elements using the RCHM we observe similar learning behaviour as using the CHM shown in Figure 5.2 and both struggle with numerical problems using `float` precision. Using `double` precision for the robust version also fixes numerical issues for $P = -30$ and the underflows occur much later for $P = -200.3$ and $P = 271.0$ shown in Table 5.4. In case of learning the non-full cross-shaped structuring elements the performance is also limited by `float` and `double` precision. Using `double` precision enables also to learn for $P = -30$ and the underflows occur for $P = -200.6$ and $P = 255.1$ shown in Table 5.4. Note that training time using `double` precision is much longer than using `float` precision on GPUs only supporting `float`.

The success and failure cases for learning full square-shaped structuring elements using counter-harmonic mean, see Figure 5.5, also apply to the robust counter-

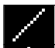
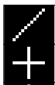
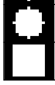
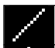
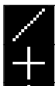
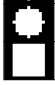
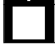
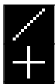
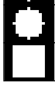
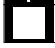
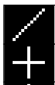
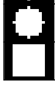
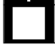
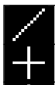
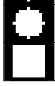
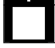
		Erosion					Dilation				
SE	Fullness	Target	MSE	P_{MSE}	$SSIM$	P_{SSIM}	Target	MSE	P_{MSE}	$SSIM$	P_{SSIM}
	9%	κ^{-5}	$1.46 \cdot 10^{-9}$	-5.0	1.0000	-5.0	κ^{5}	$1.43 \cdot 10^{-9}$	5.0	1.0000	5.0
	17%		$1.26 \cdot 10^{-9}$	-5.0	1.0000	-5.0		$1.27 \cdot 10^{-9}$	5.0	1.0000	5.0
	67%		$5.17 \cdot 10^{-10}$	-5.0	1.0000	-5.0		$5.09 \cdot 10^{-10}$	5.0	1.0000	5.0
	100%		$1.39 \cdot 10^{-14}$	-5.0	1.0000	-5.0		$2.33 \cdot 10^{-14}$	5.0	1.0000	5.0
	9%	κ^{-10}	$1.55 \cdot 10^{-9}$	-10.0	1.0000	-10.0	κ^{10}	$1.51 \cdot 10^{-9}$	10.0	1.0000	10.0
	17%		$1.35 \cdot 10^{-9}$	-10.0	1.0000	-10.0		$1.33 \cdot 10^{-9}$	10.0	1.0000	10.0
	67%		$5.51 \cdot 10^{-10}$	-10.0	1.0000	-10.0		$5.36 \cdot 10^{-10}$	10.0	1.0000	10.0
	100%		$1.37 \cdot 10^{-14}$	-10.0	1.0000	-10.0		$2.80 \cdot 10^{-14}$	10.0	1.0000	10.0
	9%	κ^{-20}	$1.98 \cdot 10^{-9}$	-20.0	1.0000	-20.0	κ^{20}	$1.87 \cdot 10^{-9}$	20.0	1.0000	20.0
	17%		$1.58 \cdot 10^{-9}$	-20.0	1.0000	-20.0		$1.53 \cdot 10^{-9}$	20.0	1.0000	20.0
	67%		$6.14 \cdot 10^{-10}$	-20.0	1.0000	-20.0		$5.79 \cdot 10^{-10}$	20.0	1.0000	20.0
	100%		$1.30 \cdot 10^{-14}$	-20.0	1.0000	-20.0		$2.58 \cdot 10^{-14}$	20.0	1.0000	20.0
	9%	κ^{-30}	$1.10 \cdot 10^{-7}$	-29.0	0.9998	-26.9	κ^{30}	$2.66 \cdot 10^{-9}$	29.9	1.0000	29.9
	17%		$1.23 \cdot 10^{-8}$	-29.8	0.9996	-26.1		$2.07 \cdot 10^{-9}$	30.0	1.0000	30.0
	67%		$1.55 \cdot 10^{-9}$	-30.0	1.0000	-30.0		$9.39 \cdot 10^{-10}$	30.0	1.0000	30.0
	100%		$3.93 \cdot 10^{-14}$	-30.0	1.0000	-30.0		$2.40 \cdot 10^{-14}$	30.0	1.0000	29.9
	9%	ε $P \to -\infty$	$3.82 \cdot 10^{-5}$	-39.6	0.9957	-25.5	δ $P \to \infty$	$7.62 \cdot 10^{-5}$	32.9	0.9960	30.5
	17%		$6.35 \cdot 10^{-5}$	-38.9	0.9928	-25.7		$1.16 \cdot 10^{-4}$	34.2	0.9942	34.5
	67%		$1.31 \cdot 10^{-4}$	-38.0	0.9914	-25.0		$2.58 \cdot 10^{-4}$	32.6	0.9940	36.3
	100%		$1.40 \cdot 10^{-4}$	-37.8	0.9913	-24.6		$3.07 \cdot 10^{-4}$	33.2	0.9938	37.3

Table 5.3: Experimental results for learning erosions and dilations using the *robust* counterharmonic mean filter with `float` precision. Only the best results out of 50000 training iterations are shown including the corresponding learned P values.

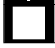
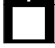
		Erosion			Dilation		
SE	Fullness	Target	$SSIM$	P_{SSIM}	Target	$SSIM$	P_{SSIM}
	17%	κ^{-30}	0.9999	-30.0	κ^{30}	0.9999	30.0
	100%		0.9999	-30.0		0.9999	30.0
	17%	ε $P \to -\infty$	0.9998	-200.6	δ $P \to \infty$	0.9998	255.1
	100%		0.9998	-200.3		0.9998	271.0

Table 5.4: Experimental results for learning erosions and dilations using the *robust* counterharmonic mean filter with `double` precision. Only the best results out of 100000 training iterations are shown including the corresponding learned P values.

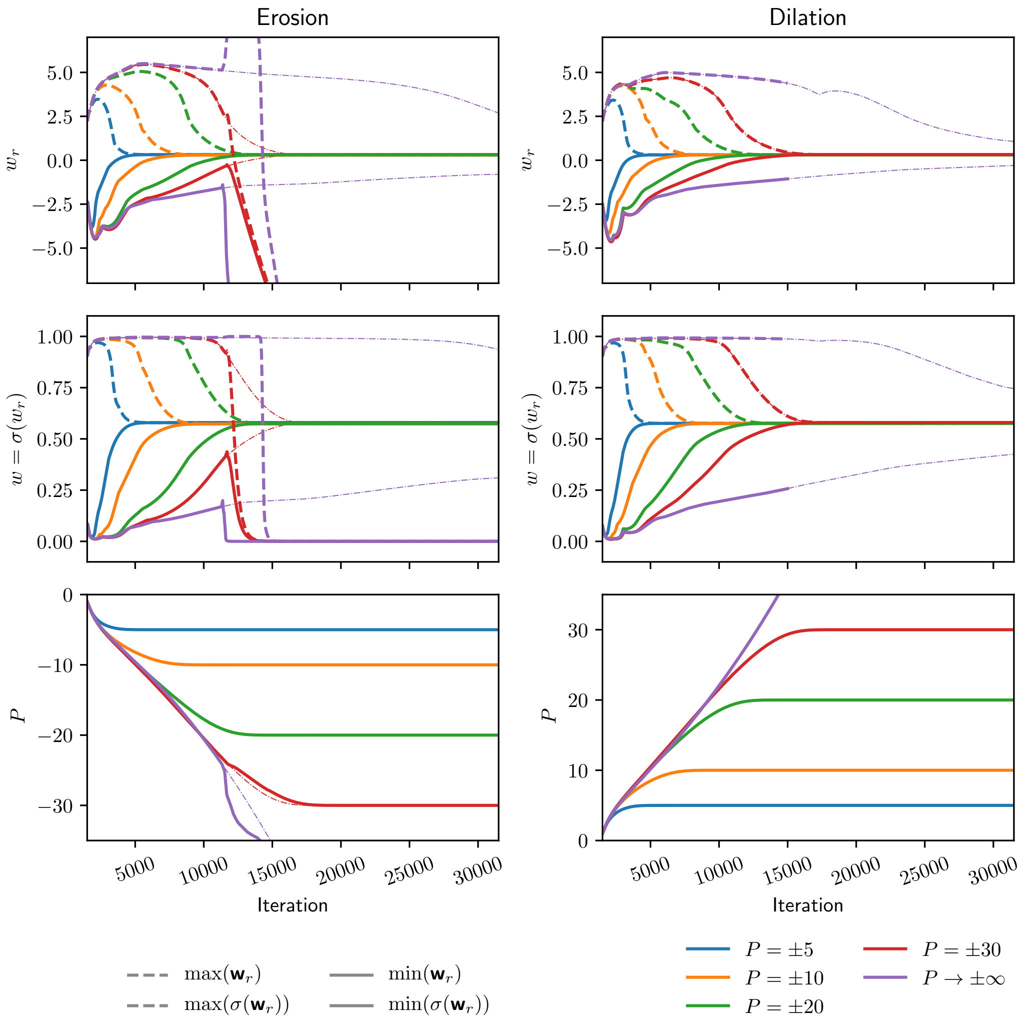


Figure 5.8: Each color represents the parameters P and $\mathbf{w}$ of a single robust counter-harmonic mean filter during learning of the *full square-shaped* structuring element using the SSIM loss. The left column shows the learning behaviour of erosions and the right column of dilations. In the top plots we show the maxima and minima of $\mathbf{w_r}$, in the middle plots we show the resulting sigmoid scaled structuring elements $\mathbf{w}$ and in the bottom the corresponding P values for the first 30000 iterations. Bold lines visualize values using `float` precision and the continuing dash-dotted lines visualize values using `double` precision.

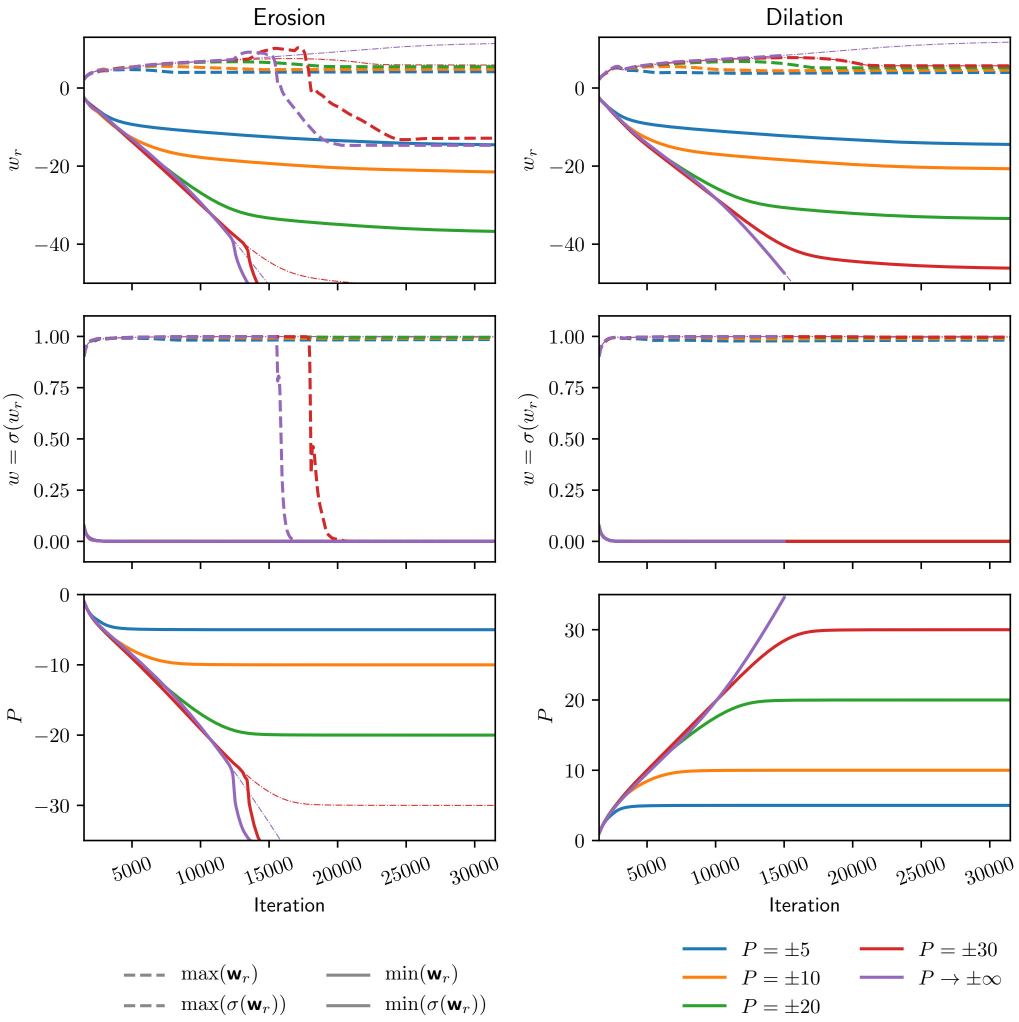
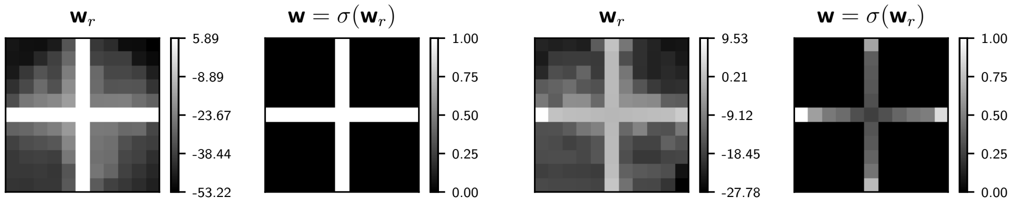


Figure 5.9: Each color represents the parameters P and $\mathbf{w}$ of a single robust counter-harmonic mean filter during learning of the *non-full cross-shaped* structuring element using the SSIM loss. The left column shows the learning behaviour of erosions and the right column of dilations. In the top plots we show the maxima and minima of $\mathbf{w_r}$, in the middle plots we show the resulting sigmoid scaled structuring elements $\mathbf{w}$ and in the bottom the corresponding P values for the first 30000 iterations. Bold lines visualize values using `float` precision and the continuing dash-dotted lines visualize values using `double` precision.

harmonic mean using full and non-full structuring elements. Stable learning is possible by constraining $|P| \gtrapprox 22$ to $P \cong \pm 22$ using `float` precision and by constraining $|P| \gtrapprox 180$ to $P \cong \pm 180$ using `double` precision at the cost of approximation quality.

Using `double` precision targeting $P = -30$ results in a perfect non-full structuring element as shown in Figure 5.10 (a) and using `float` precision constraining $P < -22$ to $P = -22$ to learn a classical erosion results in the learned structuring element shown in Figure 5.10 (b) which gives a very good approximation of an erosion as shown in Figure 5.11 (c). If no constraining of P is performed, undesired artefacts in the output images appear. An example is with a learned $P = -46$ is shown in Figure 5.11 (b). The classical erosion shown in Figure 5.11 (a) is approximated very well by the constraints $P \cong \pm 22$ using `float` precision and $P \cong \pm 180$ using `double` precision.

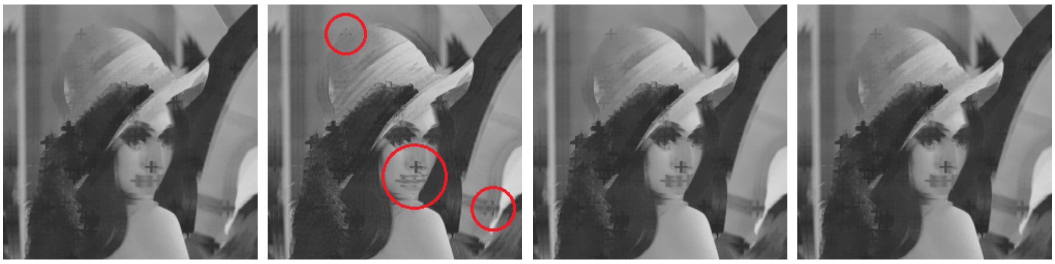


(a)

(b)

Figure 5.10: Learned cross-shaped structuring elements with targeting $P = -30$ using `double` precision is shown in (a) and targeting $P = -\infty$ using `float` precision and constraining $P < -22$ to $P = -22$ is shown in (b). For visualization purpose the $\mathbf{w_r}$ plots are scaled between $\min(\mathbf{w_r})$ and $\max(\mathbf{w_r})$ and the resulting structuring elements $\mathbf{w} = \sigma(\mathbf{w_r})$ plots are scaled between zero and one. By applying (3.2) on $\mathbf{w}$, an approximation of a flat structuring element is obtained.

The novel RCHM filter performs significantly better for learning non-full structuring elements compared to the CHM filter. Numerical problems have to be considered w.r.t. the desired approximation quality and in case of using `float` precision we recommend to constrain $|P| \gtrapprox 22$ to $P \cong \pm 22$ and using `double` precision we recommend to constrain $|P| \gtrapprox 180$ to $P \cong \pm 180$. Using the constraint on P does not introduce any unwanted artefacts in the output images, unlike with the counter-harmonic mean which introduces unwanted artefacts in the output images and does not even reach such large $|P|$ values.



(a) Target ε (b) $P = -46$ (c) $P = -22$ (d) $P = -180$

Figure 5.11: The target image generated with classical erosion is shown in (a). In (b) and (c) `float` precision is used during learning. The learned $P = -46$ causes unwanted output in (b). In (c), $|P| > 22$ are constrained to $P = \pm 22$ to prevent unwanted output caused by $|P| \gg 0$ values shown in (b). Using `double` precision allows for larger $P \cong \pm 180$.

5.1.2 Learning Opening and Closing

Two (R)CHM filters in sequence are required to learn openings/closings which is a much harder problem than learning erosions/dilations only requiring a single filter.

We compare the results of learning openings/closings using the RCHM and CHM in Table 5.5. RCHM outperforms CHM again, especially for learning non-full structuring elements. However, learning openings/closings is very unstable in terms of underflows of floating points. To prevent numerical problems during learning, we recommend to clip P to $P = \pm 22$ using `float` precision at the cost of approximation quality. See the experimental results of the approximation quality using clipping in Appendix Tables A.3 and A.6.

Learning RCHM filters using `double` precision enables larger $|P|$ values and clearly improves the approximation quality as we show in Table 5.6. Due to limited computing power and graphics card performance, we only trained for 100000 iterations and we expect to obtain even better results if larger $|P|$ are learned after longer training, similar to the case of learning erosions/dilations in the previous section achieving $P \approx -200$ and $P \approx 250$.

In Figures 5.12 and 5.13 we show the output images of the best cases using `float` precision and the output images are very similar to the target images. While closings perform very well for the square-shaped and cross-shaped structuring

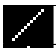
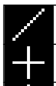
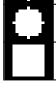
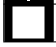
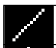
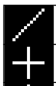
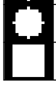
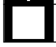
			CHM			RCHM		
SE	Fullness	Target	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
	9%	$\gamma(f)$	0.9743	−11.3	14.0	0.9893	−28.7	28.9
	17%		0.9660	−12.7	13.9	0.9859	−26.0	29.1
	67%		0.9752	−19.3	17.0	0.9834	−25.2	28.0
	100%		0.9867	−24.3	26.6	0.9820	−19.9	20.8
	9%	$\phi(f)$	0.9726	11.9	−15.5	0.9880	24.6	−27.8
	17%		0.9676	14.5	−18.3	0.9862	32.4	−31.9
	67%		0.9734	18.9	−20.0	0.9739	24.5	−30.3
	100%		0.9882	28.9	−33.4	0.9916	38.8	−43.4

Table 5.5: Comparison of experimental results using **CHM** and **RCHM** filters for learning classical opening γ and closing ϕ with `float` precision. Only the best results out of 400000 (CHM)/50000 (RCHM) training iterations are shown including the corresponding learned P_0 values in the first layer and P_1 in the second layer. See full comparison in Appendix in Tables A.1, A.2, A.4 and A.5.

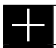
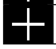
SE	Fullness	Target	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
	17%	$\gamma(f)$	0.9969	−134.0	166.9
	17%	$\phi(f)$	0.9953	141.5	−196.1

Table 5.6: Experimental results for learning classical opening/closing using the non-full cross-shaped structuring elements with `double` precision using RCHM. The best cases of 100000 iterations are shown.

elements, we observe smoothing behaviour and undesired artefacts in the openings. We found no improvement in using `double` precision or constraining $P = \pm 22$ or even lower to $P = \pm 15$ to get rid of the artefacts. For learning openings, we did not find any solutions to fix smoothing behaviour for square-shaped structuring elements and appearing artefacts for non-full structuring elements.

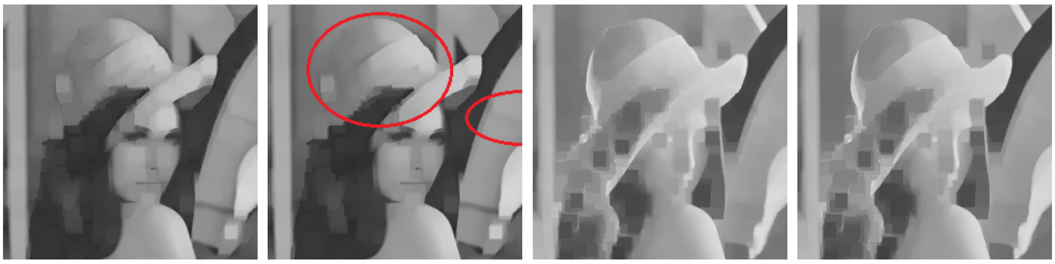

(a) Target opening γ (b) $P_0 = -19.9$ $P_1 = 20.8$ (c) Target closing ϕ (d) $P_0 = 38.8$ $P_1 = -43.4$

Figure 5.12: Comparison of target images (a) and (c) using square-shaped structuring elements with resulting output images in (b) and (d) using a learned robust counter-harmonic mean filter with `float` precision. Noticeable smoothing behaviour is indicated with red ellipses and occurs for learning the opening.

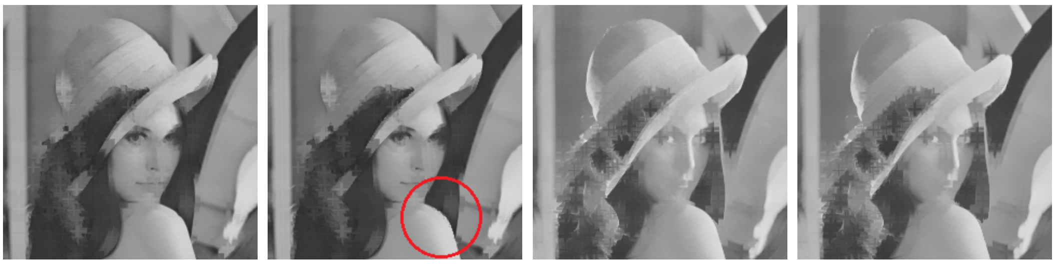
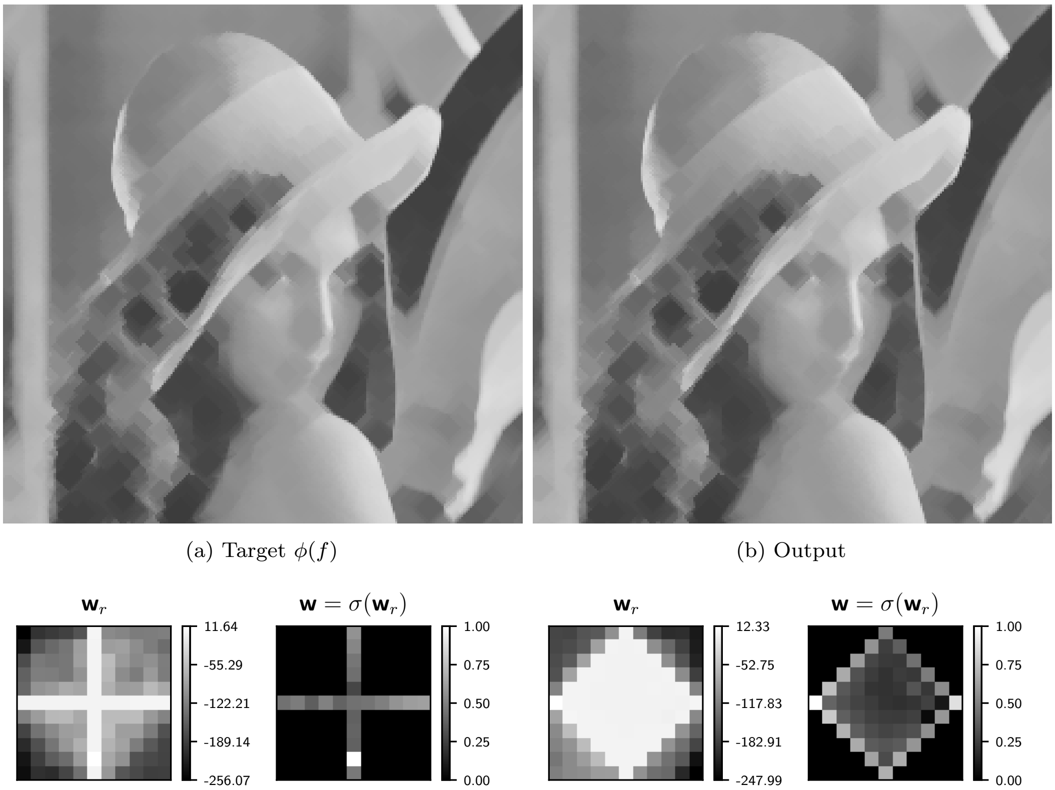

(a) Target opening γ (b) $P_0 = -26.0$ $P_1 = 29.1$ (c) Target closing ϕ (d) $P_0 = 32.4$ $P_1 = -31.9$

Figure 5.13: Comparison of target images (a) and (c) using cross-shaped structuring elements with resulting output images in (b) and (d) using a learned robust counter-harmonic mean filter with `float` precision. Noticeable artefacts for learning the openings are shown in the red circle.

A symmetrical opening/closing uses the same structuring element in both layers, while an asymmetrical opening/closing uses two different structuring elements. Until now we only show symmetrical openings/closings. For this reason, inspired by Laganière's morphological corner detector [8], we generate a target image by dilating with a cross-shaped structuring element followed by eroding with a diamond-shaped structuring element. In Figure 5.14, we show that the learning framework is capable to learn such asymmetrical closing.



(a) Target $\phi(f)$

(b) Output

(c) $P_0 = 200$

(d) $P_1 = -200$

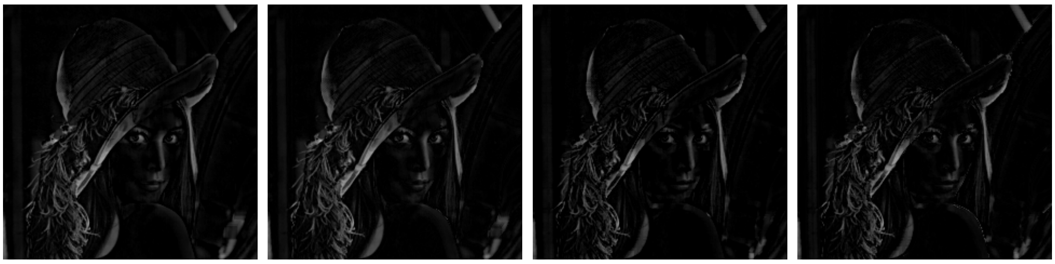
Figure 5.14: Asymmetrical closing using a dilation with a cross-shaped structuring element followed by an erosion with a diamond-shaped structuring element. The closed target image is in (a), the output after learning in (b), the learned cross-shaped structuring element from the first layer in (c) and the learned diamond-shaped structuring element from the second layer in (d). RCHM, `double` precision and a constraint on $|P| \ll 200$ is used.

To conclude, learning two (R)CHM filters for openings/closings is a more complex task than learning only a single filter. Learning with our RCHM approach clearly outperforms the work in [12] in learning openings/closings as they only obtain low P values which results in very smooth images, e.g., with $P_0 = -7.64$, $P_1 = 7.07$ for an opening and $P_0 = 6.8$, $P_1 = -8.85$ for a closing. Using our approach for openings, the output images still show some smoothing effects for the full structuring elements and few artefacts appear for non-full structuring elements. Nevertheless, the approximation quality to the true opening is sufficient. Furthermore, the output images for learning closings excel in terms of approximation quality.

5.1.3 Learning Top-Hats

Like in the previous section, two (R)CHM filters are required to learn openings/closings for the top-hat transformations. The absolute difference between input image and the learned openings/closings result in white/black top-hats.

Experimental results for learning white and black top-hats using CHM and RCHM are shown in Table 5.7. RCHM outperforms CHM for all experiments except for the black top-hat square case. However, again learning is very unstable without constraining P; hence, we provide experimental results with clipping $|P| > 25$ to $P = \pm 25$ in Appendix Tables A.9 and A.12. A comparison between the resulting output images of the constrained RCHM using square-shaped and cross-shaped structuring elements is shown in Figure 5.15 for white top-hats and Figure 5.16 for black top-hats. In both cases we observe a good approximation quality.



(a) Target WTH Square

(b) $P_0 = -25.0$ $P_1 = 16.3$

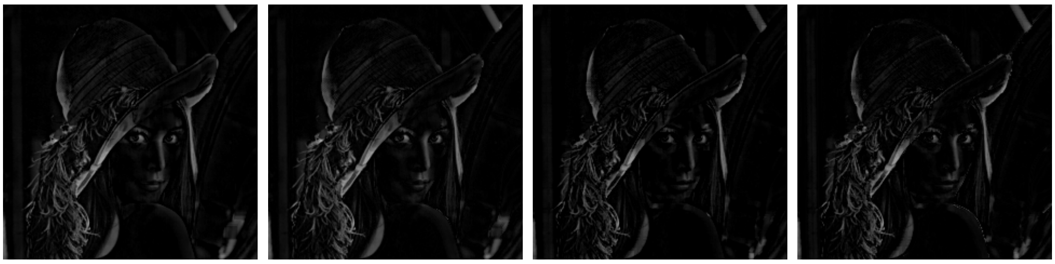
(c) Target WTH Cross

(d) $P_0 = -25.0$ $P_1 = 22.9$

Figure 5.15: The target images generated with the white top-hat transformation using square-shaped structuring elements is shown in (a) and using a cross-shaped structuring elements in (c). In (b) and (d) the resulting output images using a constraint on P during learning is shown.

			CHM			RCHM		
SE	Fullness	Target	$SSIM$	$\text{P}_{0,SSIM}$	$\text{P}_{1,SSIM}$	$SSIM$	$\text{P}_{0,SSIM}$	$\text{P}_{1,SSIM}$
╱	9%	$WTH(f)$	0.8707	−12.0	8.1	0.8892	−34.2	28.8
+	17%		0.8326	−24.1	24.2	0.9231	−28.8	29.0
●	67%		0.8888	−25.5	22.8	0.9376	−39.1	29.2
■	100%		0.9484	−28.5	20.1	0.9615	−45.0	27.3
╱	9%	$BTH(f)$	0.8011	27.3	−25.7	0.8965	34.5	−31.4
+	17%		0.8218	25.5	−28.0	0.9006	29.1	−30.9
●	67%		0.9025	30.4	−31.2	0.9409	34.3	−31.3
■	100%		0.9603	38.2	−31.8	0.9546	38.7	−30.1

Table 5.7: Comparison of experimental results using **CHM** and **RCHM** filters for learning classical white and black top-hat transformations with `float` precision. Only the best results out of 200000 training iterations are shown including the corresponding learned P_0 values in the first layer and P_1 in the second layer. See more detailed comparisons in Appendix Tables A.7, A.8, A.10 and A.11.



(a) Target BTH Square

(b) $P_0 = 25.0$ $P_1 = -20.7$

(c) Target BTH Cross

(d) $P_0 = 25.0$ $P_1 = -25.0$

Figure 5.16: The target images generated with the white top-hat transformation using square-shaped structuring elements is shown in (a) and using a cross-shaped structuring elements in (c). In (b) and (d) the resulting output images using a constraint on P during learning is shown.

To conclude learning of morphological operators, we found that learning larger $|P|$ is possible by using the robust counter-harmonic mean filter, which also improves the approximation quality over the counter-harmonic mean. However, constraining P according to floating point precision in expense of approximation quality is required for stable learning.

5.2 Morphological Corner Detection

A morphological corner detector was proposed in [8] which uses two asymmetrical closings with cross-, diamond-, star- and square-shaped structuring elements. The corners c are detected as the absolute difference of both asymmetrical closings and calculated as follows:

$$c(f) = |\phi(f)_{+,\diamond} - \phi(f)_{\times,\square}| \tag{5.1}$$

The subscripts indicate the structuring elements within the asymmetric closings.

In Figure 5.14, we already showed that learning asymmetrical closings is possible, actually we learned the $\phi(f)_{+,\diamond}$ pipeline with 11×11 structuring elements.

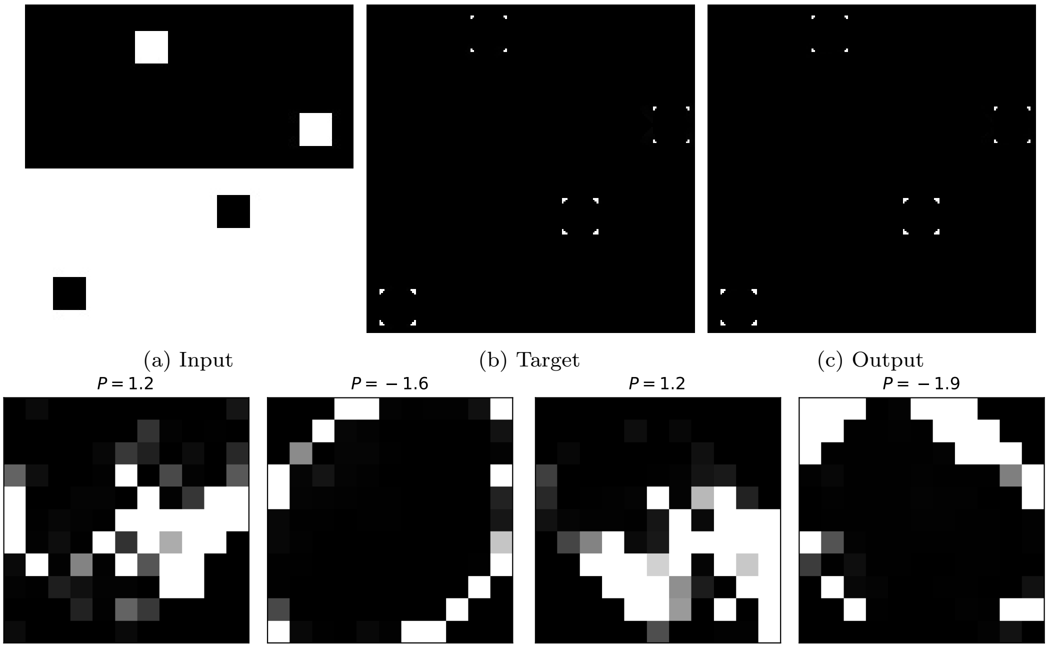
For corner detection we created two synthetic input images shown in Figure 5.17 (a) and Figure 5.18 (a) and generated target images in (b) according to (5.1) with 7×7 structuring elements. For learning, the structuring elements are of size 11×11 in the network architecture. The outputs after learning are shown in (c) and the two learned pipelines in (d) and (e). We observe that pseudo-closings are learned with rather low $|P|$ values and the learned structuring elements are neither cross-, diamond-, star- or square-shaped. In addition, we observe that learning to find corners with bright foreground and darker background to be harder than learning to find corners with dark foreground and brighter background, i.e. Figure 5.18 (c).

The most interesting findings of this experiment are:

(a) $|P|$ values are large enough to approximate morphological operators rather than convolutions.

(b) While the target image is generated according to [8] and (5.1), the network learns completely different asymmetrical closings with different structuring elements.

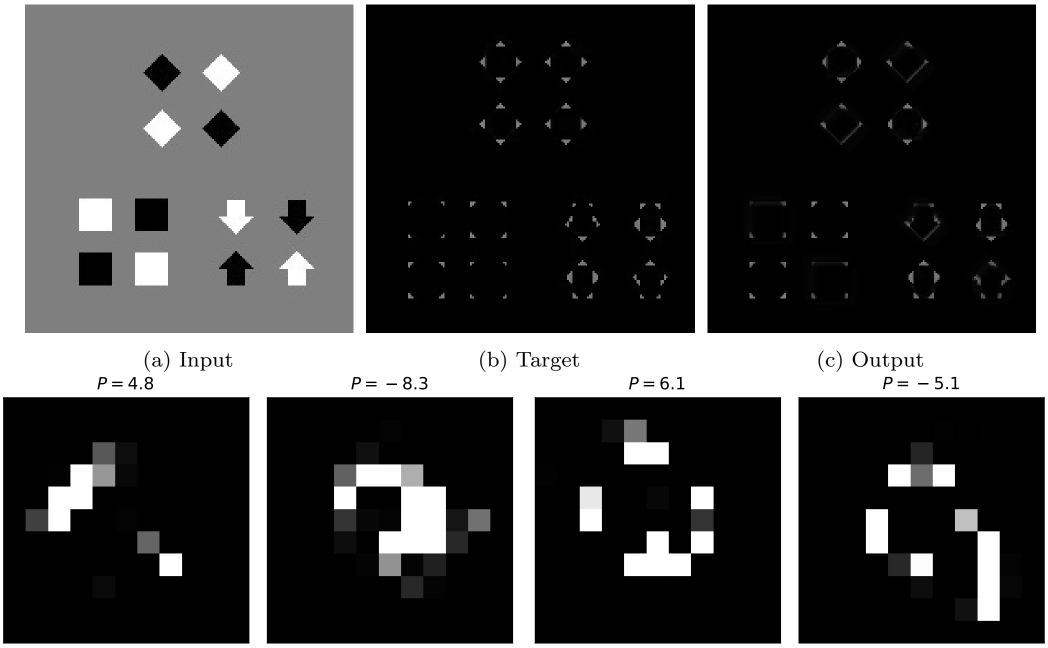
(c) The network architecture has actually learned a new morphological corner detector.



(a) Input (b) Target (c) Output

(d) Pipeline 1 (e) Pipeline 2

Figure 5.17: Given the input image (a) and the generated corner detection target image (b), the end-to-end learned network detects corners shown in (c). The learned pipeline 1 is shown in (d) and the parallel pipeline 2 in (e).



(a) Input

(b) Target

(c) Output

(d) Pipeline 1

(e) Pipeline 2

Figure 5.18: Given a more complex synthetic input image (a) and the generated corner detection target image (b), the end-to-end learned network detects corners shown in (c). The learned pipeline 1 is shown in (d) and the parallel pipeline 2 in (e).

5.3 Image Classification

In the previous sections our goal was to learn approximations of predetermined morphological operators. However, for image classification, we allow the network to learn any morphological operator. We study if a morphological neural network is able to extract meaningful features for image classification.

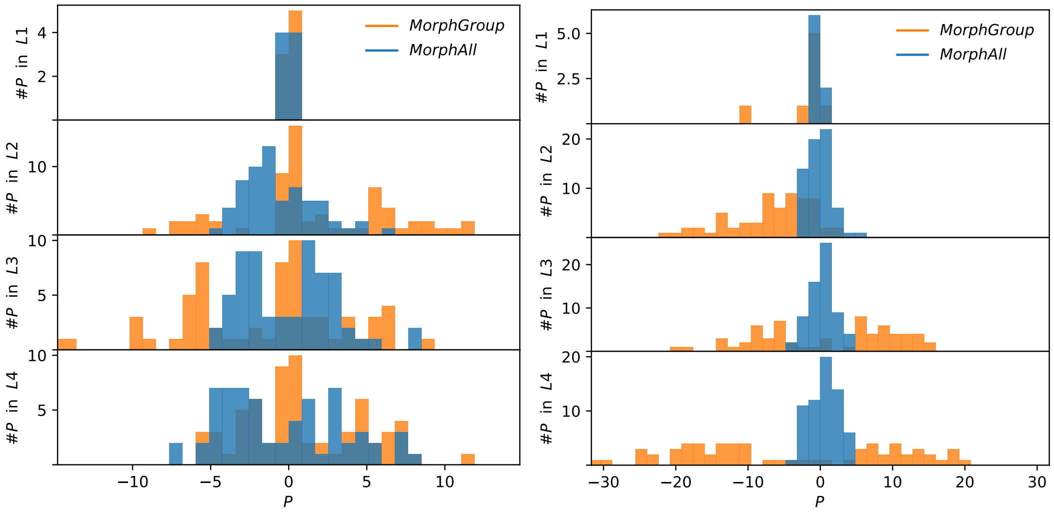
Our proposed morphological networks (Figure 4.8) using either MorphGroup layers or MorphAll layers are able to extract features for classifying the MNIST and Fashion-MNIST datasets and the test accuracies are shown in Table 5.8. The MorphAll layer outperforms the MorphGroup layer by using more parameters for the depthwise 1×1 convolutions, where using the MorphAll layers results in 12922 parameters and using the MorphGroup layers in 11522 parameters. Note that for each experiment the input images of size 28×28 are morphologically processed by four consecutive layers and reduced to a fixed vector of size 128, thus, the experiments are comparable based on the extracted features.

For comparison of morphological networks with CNNs, we also used convolutional layers [10] and depthwise separable convolution layers [3] in the same network architecture as illustrated in Figure 4.8. While the convolutions perform better in terms of test accuracy, we observe that the morphological networks are not far off. Note that we perform no hyperparameter optimization or architecture tuning.

Layers	MNIST	Fashion-MNIST
MorphGroup	98.41%	87.21%
MorphAll	**99.10%**	**88.40%**
Depthwise Separable Convolution	98.80%	**89.40%**
Convolution	**99.23%**	88.53%

Table 5.8: Test accuracies for classifying MNIST and Fashion-MNIST. We compare the performance of the morphological layers MorphGroup and MorphAll where MorphAll outperforms MorphGroup for both datasets. In addition, we show the test accuracies of convolutional layers using the same network architecture as for the morphological layers.

The learned P value distributions for both types of morphological layers are shown in Figure 5.19. From Section 3.2, Figure 3.1 we know that for $P = 0$ the (R)CHM is a convolution with a positve kernel, P values close to zero produce smooth morphologically processed images and $|P| \gg 0$ approximates erosions/dilations well. In $L1$ most of the values are between $[-0.5, 0.5]$ for MorphAll and MorphGroup layers. Thus, both types approximately learn convolutions and appear not to be well suited for the first layer of the network. For MorphAll most of the values are between $[-5, 5]$ in the remaining three layers and produce smooth morphologically processed images. The MorphGroup layer type is able to learn much larger $|P|$ values, in most cases between $[-20, 20]$ for Fashion-MNIST. From previous Section 5.1 we know that the RCHM for $|P| > 20$ approximates morphological operators very well. Nevertheless, the MorphAll layers achieve better test accuracies while the MorphGroup layer outperforms the MorphAll in terms of approximating morphological operators.



(a) MNIST

(b) Fashion MNIST

Figure 5.19: Distribution of learned P values for each layer of the models with the best test accuracies in Table 5.8. The distributions of P using the MorphGroup layers are depicted in orange and using MorphAll layers in blue. MorphGroup layers learn larger $|P|$ values than MorphAll layers, especially in layers $L2, L3$ and $L4$ for Fashion-MNIST.

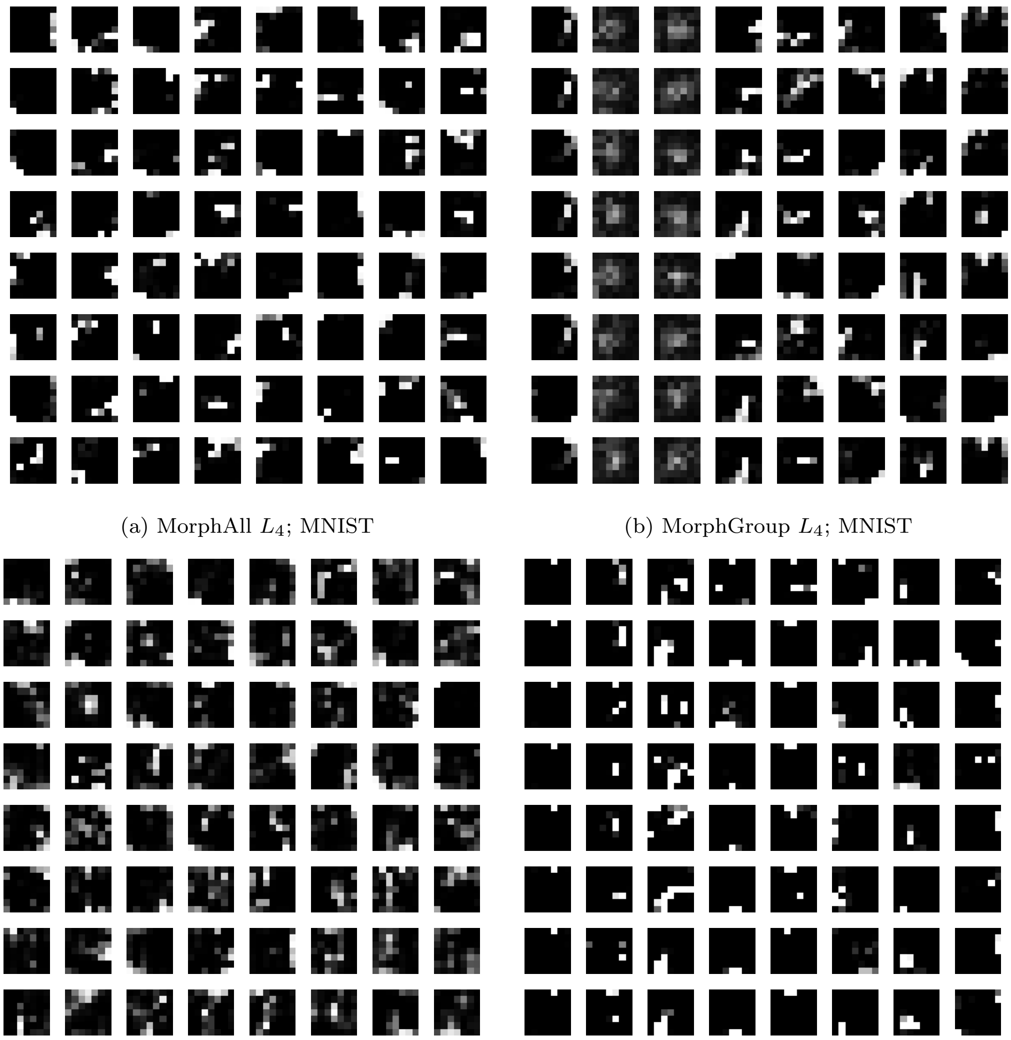
The learned structuring elements of layer $L4$ for all experiments are shown in Figure 5.20. For MorphAll the structuring elements have a high variety of different structures; thus, in combination with their lower $|P|$ values they act as smooth pseudo-morphological operator. For MorphGroup the structuring elements build groups of similar structures; thus, in combination for $|P|$ values around $P = \pm 20$ they act as excellent approximations of morphological operators. Note that the structuring elements in the second and third column in Figure 5.20 (b) are not learned well because the backpropagated error is very low. For the remaining three layers the properties of the structuring elements are similar to $L4$ and as shown in Appendix Figures A.1, A.2 and A.3.

It is worth mentioning that our morphological layers are similar to pooling layers and that they are able to learn the neighbourhood of the pooling region (learned with the structuring element) and the pooling operations min/avg./max (learned with P). With morphological layers, no pooling operation has to be predefined and the error is backpropagated through the whole neighbourhood as opposed to a widely used max pooling operation.

To conclude, using MorphAll layers outperforms MorphGroup layers in terms of test accuracy for image classification. On the other hand, MorphGroup layers outperform MorphAll layers in terms of approximating morphological operators. We show that both approaches are able to extract features for image classification.

Complexity Analysis

The number of parameters using the CNN architectures and morphological networks are comparable and are in the range of 11122 (convolution), 11522 (MorphGroup) and 12922 (MorphAll and depthwise separable convolution). The convolutional approaches are more stable while learning than the morphological ones, especially when using `float`. Morphological approaches using the (R)CHM require `double` precision to be more stable; therefore, this approach requires more memory, computing power and more expensive hardware. Calculating convolutions is very efficient on current GPUs while the (R)CHM approach requires exponent calculations which are less efficient. The RCHM also requires rescaling of images between $(0, 1)$ which we solve by using a sigmoid activation function.



(a) MorphAll L_4; MNIST

(b) MorphGroup L_4; MNIST

(c) MorphAll L_4; Fashion-MNIST

(d) MorphGroup L_4; Fashion-MNIST

Figure 5.20: Learned structuring elements $\mathbf{w} = \sigma(\mathbf{w_r})$ in layer $L4$ for all four experiments. All learned structuring elements are non-full structuring elements. MorphAll structuring elements have a large variety of structures and MorphGroup structuring elements have smaller variations of structures in groups of 8 due to the input/output channel size.

6 Discussion

We showed that erosions/dilations, openings/closings and top-hat transformations can be approximated with a gradient based learning framework using the counter-harmonic mean. However, we found that the CHM encounters numerical problems and unstable learning behaviour due to clipping of $\mathbf{w}$ because of the constraint $\mathbf{w} \geq 0$. No clipping of $\mathbf{w}$ is required for full square-shaped structuring elements; therefore, $|P| \gg 0$ are learned and underflows occur. Underflows can be prevented by clipping $P \cong \pm 22$ when using `float` precision and by clipping $P \cong \pm 180$ when using `double` precision. During learning non-full structuring elements, $\mathbf{w}$ may be clipped to zero, and in combination of $|P| > 10$, $w_{i,j} = 0$, $w_{k,l} \gtrapprox 0$ and $w_{m,n} \gg 0$ within the same structuring element, artefacts in the output appear causing unstable learning. Using `double` precision does not provide any advantage in learning non-full structuring elements. Our novel robust counter-harmonic mean filter avoids clipping of the structuring element by introducing a sigmoid scaled structuring element. The RCHM is able to learn non-full structuring elements with exceptional approximation quality and is only limited by the floating point precision. Like for CHM, we recommend clipping P according to the utilized precision.

As we generated own target images, we directly specified which morphological operators and structuring elements should be learned. For image classification, generalized morphological operators and structuring elements representing the data should be learned. While CNNs are exceptional for pattern matching over multiple channels, morphologically processed images have to be combined in a different manner than in CNNs. We proposed two pseudo-morphological layers, namely MorphAll and MorphGroup, which are inspired by depthwise separable convolutions, morphologically processing each input channel independently using RCHM filters followed by depthwise combinations using 1×1 convolutions. MorphAll combines over all possible morphological outputs and preferably learns pseudo-morphological operators with $-5 < P < 5$. The MorphGroup layer only combines groups of morphologically processed images and is able to learn much better approximations of morphological operators with $-30 < P < 30$. The test accuracies are better for using MorphAll layers over MorphGroup layers. We found that learning morphological networks in general is sensitive to initialization and additionally we recommend using `double` precision.

Overall, with the counter-harmonic mean learning framework we decided on an approach which is able to learn morphological operators with the backpropagation algorithm and updates all structuring element values at once. However, the (R)CHM approach is limited to only approximating *flat* erosions/dilations which might be a limiting aspect and should be tackled in future work.

7 Conclusion

Mathematical morphology aims to extract shape based features from images and requires a domain expert to find appropriate morphological operators for specific image processing problems. A learning based approach aims to find the best fitting morphological operators without the need of a domain expert. We showed that learning simple morphological operators using the counter-harmonic mean is possible, but is numerically unstable and inaccurate. Our novel robust counter-harmonic mean filter outperforms the counter-harmonic mean in terms of approximation quality, more stable learning and the ability to learn non-full structuring elements more accurately. We were able to learn morphological operators within a more complex morphological neural network, either learning pseudo-morphological operators or very good approximations. Our work provides a more accurate learning framework for morphological operators and two new approaches for learning morphological neural networks for further research on learning based mathematical morphology.

Morphological neural networks are by far not as mature as CNNs and only at the beginning, still we show that they are able to extract features for image classification. Our new building blocks may enable new approaches, i.e. replacing pooling layers where min/avg./max operators are learnable and not predefined, or even purely based on mathematical morphological theory, which may learn even more powerful morphological algorithms than engineered ones.

Bibliography

[1] J. Angulo. "Pseudo-Morphological Image Diffusion using the Counter-Harmonic Paradigm." In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer. 2010, pp. 426–437.

[2] P. S. Bullen. *Handbook of means and their inequalities*. Vol. 560. Springer, 2013.

[3] F. Chollet. "Xception: Deep learning with depthwise separable convolutions." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.

[4] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. "Mask R-CNN." In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[7] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014).

[8] R. Laganière. "Morphological corner detection." In: *International Conference on Computer Vision*. IEEE. 1998, pp. 280–285.

[9] J. Larrey-Ruiz, R. Verdú-Monedero, J. Morales-Sánchez, and J. Angulo. "Towards morphological image regularization using the counter-harmonic mean." In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer. 2013, pp. 317–328.

[10] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. “Gradient-based learning applied to document recognition.” In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[11] M. Lin, Q. Chen, and S. Yan. “Network in network.” In: *arXiv preprint arXiv:1312.4400, published in ICLR* (2013).

[12] J. Masci, J. Angulo, and J. Schmidhuber. “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. 2013, pp. 329–340.

[13] D. Mellouli, T. M. Hamdani, M. B. Ayed, and A. M. Alimi. “Morph-CNN: A Morphological Convolutional Neural Network for Image Classification.” In: *International Conference on Neural Information Processing.* Springer. 2017, pp. 110–117.

[14] *MNIST.* `http://yann.lecun.com/exdb/mnist/`. [Online; accessed 12-January-2020].

[15] R. Mondal, P. Purkait, S. Santra, and B. Chanda. “Morphological Networks for Image De-raining.” In: *International Conference on Discrete Geometry for Computer Imagery.* Springer. 2019, pp. 262–275.

[16] J. Serra. *Image analysis and mathematical morphology.* Academic Press, Inc., 1983.

[17] P. Soille. *Morphological Image Analysis: Principles and Applications.* Springer, 1999.

[18] L. J. Van Vliet. “Robust local max-min filters by normalized power-weighted filtering.” In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 1. IEEE. 2004, pp. 696–699.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. “Image quality assessment: from error visibility to structural similarity.” In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[20] H. Xiao, K. Rasul, and R. Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.” In: *arXiv preprint arXiv:1708.07747* (2017).

Appendix

Opening and Closing

More detailed comparison of all opening/closing experiments using MSE and SSIM including target images generated with all P values. The RCHM (Tables A.2, A.5) clearly outperforms the CHM (Tables A.1, A.4) in learning openings/closings for MSE and SSIM. The performance loss to the cost of stable learning using the RCHM can be compared in Tables A.3, A.6.

SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
╱	9%	$\kappa^5(\kappa^{-5}(f))$	$6.11 \cdot 10^{-5}$	-5.2	3.8	0.9977	-4.9	4.3
+	17%		$3.61 \cdot 10^{-5}$	-4.9	4.9	0.9939	-4.8	4.9
●	67%		$6.70 \cdot 10^{-7}$	-5.0	5.0	0.9999	-5.0	5.0
□	100%		$9.29 \cdot 10^{-6}$	-5.1	4.3	0.9997	-5.0	4.8
╱	9%	$\kappa^{10}(\kappa^{-10}(f))$	$1.69 \cdot 10^{-4}$	-9.3	5.5	0.9960	-9.8	8.1
+	17%		$4.63 \cdot 10^{-5}$	-9.8	9.8	1.0000	-10.0	10.0
●	67%		$4.04 \cdot 10^{-5}$	-10.1	9.4	0.9993	-10.2	10.4
□	100%		$8.00 \cdot 10^{-6}$	-10.1	9.3	0.9995	-9.8	9.4
╱	9%	$\kappa^{20}(\kappa^{-20}(f))$	$1.40 \cdot 10^{-4}$	-11.2	8.0	0.9890	-11.3	14.3
+	17%		$1.42 \cdot 10^{-4}$	-12.3	12.1	0.9926	-12.9	14.0
●	67%		$1.48 \cdot 10^{-4}$	-15.9	14.0	0.9921	-16.9	15.7
□	100%		$2.40 \cdot 10^{-14}$	-20.0	20.0	1.0000	-20.0	20.0
╱	9%	$\kappa^{30}(\kappa^{-30}(f))$	$1.67 \cdot 10^{-4}$	-11.6	11.3	0.9715	-13.0	14.4
+	17%		$1.90 \cdot 10^{-4}$	-12.9	12.3	0.9799	-14.0	14.9
●	67%		$1.73 \cdot 10^{-4}$	-16.9	14.4	0.9839	-20.5	17.3
□	100%		$7.23 \cdot 10^{-7}$	-27.8	28.9	0.9998	-26.6	29.0
╱	9%	$\gamma(f)$	$1.63 \cdot 10^{-4}$	-11.4	7.0	0.9743	-11.3	14.0
+	17%		$2.23 \cdot 10^{-4}$	-11.7	9.5	0.9660	-12.7	13.9
●	67%		$2.41 \cdot 10^{-4}$	-16.9	11.8	0.9752	-19.3	17.0
□	100%		$8.77 \cdot 10^{-5}$	-28.4	20.6	0.9867	-24.3	26.6

Table A.1: Experimental results for learning **openings** using **CHM** filters with `float` precision. Only the best results out of 400000 training iterations are shown including the corresponding learned P_0 values in the first layer and P_1 in the second layer.

SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
╱	9%	$\kappa^5(\kappa^{-5}(f))$	$2.40 \cdot 10^{-5}$	−5.1	4.7	1.0000	−5.0	5.0
┼	17%		$1.17 \cdot 10^{-5}$	−5.0	5.2	1.0000	−5.0	5.0
●	67%		$5.73 \cdot 10^{-6}$	−5.1	5.4	0.9997	−5.1	5.3
■	100%		$2.65 \cdot 10^{-14}$	−5.0	5.0	1.0000	−5.0	5.0
╱	9%	$\kappa^{10}(\kappa^{-10}(f))$	$5.71 \cdot 10^{-5}$	−10.4	9.4	1.0000	−10.0	10.0
┼	17%		$1.38 \cdot 10^{-5}$	−9.9	10.2	1.0000	−10.0	10.0
●	67%		$5.84 \cdot 10^{-5}$	−10.3	13.6	0.9974	−10.3	10.9
■	100%		$2.07 \cdot 10^{-14}$	−10.0	10.0	1.0000	−10.0	10.0
╱	9%	$\kappa^{20}(\kappa^{-20}(f))$	$8.74 \cdot 10^{-5}$	−20.5	15.4	0.9987	−20.1	19.9
┼	17%		$8.90 \cdot 10^{-5}$	−21.2	21.2	0.9976	−20.7	20.0
●	67%		$8.28 \cdot 10^{-5}$	−22.6	25.0	0.9971	−21.2	20.1
■	100%		$2.55 \cdot 10^{-14}$	−20.0	20.0	1.0000	−20.0	20.0
╱	9%	$\kappa^{30}(\kappa^{-30}(f))$	$1.00 \cdot 10^{-4}$	−22.5	17.2	0.9983	−28.3	26.6
┼	17%		$1.17 \cdot 10^{-4}$	−23.7	22.5	0.9960	−27.0	29.5
●	67%		$7.97 \cdot 10^{-5}$	−24.8	30.6	0.9952	−32.6	29.2
■	100%		$1.09 \cdot 10^{-5}$	−30.6	27.8	0.9998	−26.6	29.9
╱	9%	$\gamma(f)$	$1.11 \cdot 10^{-4}$	−27.1	16.2	0.9893	−28.7	28.9
┼	17%		$1.82 \cdot 10^{-4}$	−22.4	18.7	0.9859	−26.0	29.1
●	67%		$1.00 \cdot 10^{-4}$	−33.6	29.5	0.9834	−25.2	28.0
■	100%		$5.75 \cdot 10^{-5}$	−42.0	27.4	0.9820	−19.9	20.8

Table A.2: Experimental results for learning **openings** using **RCHM** filters with `float` precision. Only the best results out of 50000 training iterations are shown including the corresponding learned P_0 values in the first layer and P_1 in the second layer.

SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
╱	9%	$\kappa^{30}(\kappa^{-30}(f))$	$9.96 \cdot 10^{-5}$	−22.0	18.1	0.9973	−22.0	22.0
┼	17%		$1.15 \cdot 10^{-4}$	−22.0	22.0	0.9937	−22.0	22.0
●	67%		$8.78 \cdot 10^{-5}$	−22.0	22.0	0.9943	−22.0	22.0
■	100%		$2.53 \cdot 10^{-5}$	−22.0	22.0	0.9972	−22.0	22.0
╱	9%	$\gamma(f)$	$1.23 \cdot 10^{-4}$	−22.0	14.1	0.9882	−22.0	22.0
┼	17%		$1.17 \cdot 10^{-4}$	−22.0	21.3	0.9820	−22.0	22.0
●	67%		$1.51 \cdot 10^{-4}$	−22.0	20.2	0.9821	−22.0	22.0
■	100%		$1.30 \cdot 10^{-4}$	−22.0	15.4	0.9840	−22.0	22.0

Table A.3: Experimental results from learning **openings** shown in Table A.2 using RCHM filters and constraining $|P| > 22$ to $P = \pm 22$.

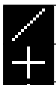
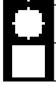
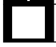
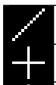
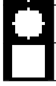
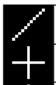
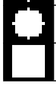
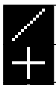
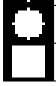
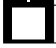
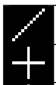
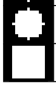
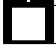
SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
	9%	$\kappa^{-5}(\kappa^{5}(f))$	$3.71 \cdot 10^{-5}$	5.1	−4.0	0.9958	5.0	−4.1
	17%		$4.49 \cdot 10^{-15}$	5.0	−5.0	1.0000	5.0	−5.0
	67%		$7.01 \cdot 10^{-7}$	5.0	−5.0	0.9999	5.0	−5.0
	100%		$1.78 \cdot 10^{-6}$	5.0	−4.8	1.0000	5.0	−5.0
	9%	$\kappa^{-10}(\kappa^{10}(f))$	$2.26 \cdot 10^{-4}$	10.2	−5.9	0.9905	9.9	−7.2
	17%		$4.93 \cdot 10^{-6}$	10.0	−10.2	0.9650	9.2	−8.4
	67%		$2.51 \cdot 10^{-5}$	10.3	−8.9	1.0000	10.0	−10.0
	100%		$1.09 \cdot 10^{-4}$	10.5	−6.9	0.9988	9.9	−9.3
	9%	$\kappa^{-20}(\kappa^{20}(f))$	$3.47 \cdot 10^{-4}$	8.4	−6.5	0.9848	12.1	−11.4
	17%		$8.28 \cdot 10^{-5}$	13.6	−14.9	0.9884	14.8	−16.9
	67%		$9.57 \cdot 10^{-5}$	18.1	−17.1	0.9979	19.1	−19.2
	100%		$2.42 \cdot 10^{-14}$	20.0	−20.0	1.0000	20.0	−20.0
	9%	$\kappa^{-30}(\kappa^{30}(f))$	$2.82 \cdot 10^{-4}$	9.3	−10.4	0.9803	12.7	−15.4
	17%		$1.26 \cdot 10^{-4}$	13.3	−15.6	0.9806	14.9	−18.2
	67%		$1.72 \cdot 10^{-4}$	20.3	−19.9	0.9913	20.8	−24.8
	100%		$2.71 \cdot 10^{-14}$	30.0	−30.0	1.0000	30.0	−30.0
	9%	$\phi(f)$	$1.99 \cdot 10^{-4}$	11.4	−7.7	0.9726	11.9	−15.5
	17%		$1.90 \cdot 10^{-4}$	12.6	−14.3	0.9676	14.5	−18.3
	67%		$2.42 \cdot 10^{-4}$	18.0	−14.0	0.9734	18.9	−20.0
	100%		$8.18 \cdot 10^{-5}$	32.2	−31.6	0.9882	28.9	−33.4

Table A.4: Experimental results for learning **closings** using **CHM** filters with `float` precision. Only the best results out of 400000 training iterations are shown including the corresponding learned P_0 values in the first layer and P_1 in the second layer.

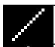
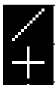
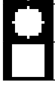
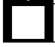
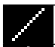
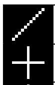
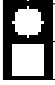
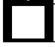
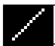
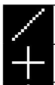
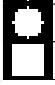
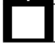
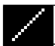
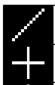
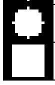
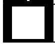
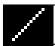
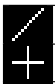
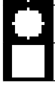
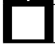
SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
	9%	$\kappa^{-5}(\kappa^{5}(f))$	$2.77 \cdot 10^{-9}$	5.0	-5.0	1.0000	5.0	-5.0
	17%		$2.46 \cdot 10^{-9}$	5.0	-5.0	1.0000	5.0	-5.0
	67%		$2.87 \cdot 10^{-6}$	5.0	-5.3	1.0000	5.0	-5.0
	100%		$2.04 \cdot 10^{-14}$	5.0	-5.0	1.0000	5.0	-5.0
	9%	$\kappa^{-10}(\kappa^{10}(f))$	$8.32 \cdot 10^{-6}$	10.1	-10.3	0.9992	10.1	-10.2
	17%		$5.06 \cdot 10^{-6}$	10.0	-10.1	1.0000	10.0	-10.0
	67%		$1.46 \cdot 10^{-5}$	10.2	-10.7	1.0000	10.0	-10.0
	100%		$2.01 \cdot 10^{-14}$	10.0	-10.0	1.0000	10.0	-10.0
	9%	$\kappa^{-20}(\kappa^{20}(f))$	$8.80 \cdot 10^{-6}$	20.3	-20.8	0.9971	21.1	-20.5
	17%		$1.45 \cdot 10^{-5}$	21.0	-20.8	0.9989	20.3	-20.1
	67%		$4.44 \cdot 10^{-5}$	21.2	-22.6	0.9995	20.2	-20.1
	100%		$2.41 \cdot 10^{-14}$	20.0	-20.0	1.0000	20.0	-20.0
	9%	$\kappa^{-30}(\kappa^{30}(f))$	$1.15 \cdot 10^{-5}$	29.3	-28.3	0.9917	28.4	-28.2
	17%		$2.45 \cdot 10^{-5}$	31.5	-30.2	0.9982	30.4	-29.7
	67%		$1.45 \cdot 10^{-4}$	25.5	-29.5	0.9988	30.4	-30.2
	100%		$7.82 \cdot 10^{-9}$	29.8	-29.9	1.0000	30.0	-30.0
	9%	$\phi(f)$	$4.85 \cdot 10^{-5}$	30.4	-27.0	0.9880	24.6	-27.8
	17%		$7.10 \cdot 10^{-5}$	26.0	-26.6	0.9862	32.4	-31.9
	67%		$1.99 \cdot 10^{-4}$	25.4	-26.6	0.9739	24.5	-30.3
	100%		$8.53 \cdot 10^{-5}$	31.7	-29.1	0.9916	38.8	-43.4

Table A.5: Experimental results for learning **closings** using **RCHM** filters with `float` precision. Only the best results out of 50000 training iterations are shown including the corresponding learned P_0 values in the first layer and P_1 in the second layer.

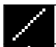
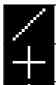
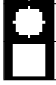
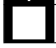
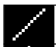
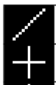
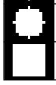
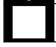
SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
	9%	$\kappa^{-30}(\kappa^{30}(f))$	$1.87 \cdot 10^{-5}$	22.0	-22.0	0.9904	22.0	-22.0
	17%		$2.76 \cdot 10^{-5}$	22.0	-22.0	0.9969	22.0	-22.0
	67%		$7.96 \cdot 10^{-5}$	22.0	-22.0	0.9976	22.0	-22.0
	100%		$1.94 \cdot 10^{-5}$	22.0	-22.0	0.9977	22.0	-22.0
	9%	$\phi(f)$	$7.02 \cdot 10^{-5}$	22.0	-22.0	0.9861	22.0	-22.0
	17%		$8.35 \cdot 10^{-5}$	22.0	-22.0	0.9818	22.0	-22.0
	67%		$1.21 \cdot 10^{-4}$	22.0	-22.0	0.9812	22.0	-22.0
	100%		$1.52 \cdot 10^{-4}$	22.0	-22.0	0.9815	22.0	-22.0

Table A.6: Experimental results from learning **closings** shown in Table A.5 using RCHM filters and constraining $|P| > 22$ to $P = \pm 22$.

Top-Hats

More detailed comparison of all top-hat experiments using MSE and SSIM including target images generated with $P = \pm 30$ and a constraint for more stable learning. The RCHM (Tables A.8, A.11) clearly outperforms the CHM (Tables A.7, A.10) in learning top-hats for MSE and SSIM. The performance loss to the cost of stable learning using the RCHM can be compared in Tables A.9, A.12.

SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
⟋	9%	$WTH(f)_{P,\pm 30}$	$2.59 \cdot 10^{-4}$	-10.4	9.9	0.7632	-8.6	7.3
+	17%		$2.90 \cdot 10^{-4}$	-11.6	10.9	0.7422	-25.2	31.0
●	67%		$2.96 \cdot 10^{-4}$	-17.2	14.0	0.7805	-25.5	29.1
■	100%		$9.69 \cdot 10^{-5}$	-27.4	29.3	0.8694	-27.8	29.3
⟋	9%	$WTH(f)$	$1.82 \cdot 10^{-4}$	-9.9	6.0	0.8707	-12.0	8.1
+	17%		$2.14 \cdot 10^{-4}$	-11.7	9.8	0.8326	-24.1	24.2
●	67%		$2.21 \cdot 10^{-4}$	-16.0	11.2	0.8888	-25.5	22.8
■	100%		$8.12 \cdot 10^{-5}$	-28.6	20.9	0.9484	-28.5	20.1

Table A.7: Experimental results for the white top-hat transformation using the **CHM** with `float` precision. Best results from 200000 iterations are shown.

SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
⟋	9%	$WTH(f)_{P,\pm 30}$	$2.33 \cdot 10^{-4}$	-27.9	25.3	0.8344	-31.0	28.6
+	17%		$2.92 \cdot 10^{-4}$	-29.0	30.3	0.8228	-29.7	27.2
●	67%		$1.83 \cdot 10^{-4}$	-25.4	27.0	0.8451	-36.1	28.9
■	100%		$1.10 \cdot 10^{-4}$	-35.5	29.7	0.8776	-36.2	28.8
⟋	9%	$WTH(f)$	$2.08 \cdot 10^{-4}$	-26.1	20.1	0.8892	-34.2	28.8
+	17%		$2.31 \cdot 10^{-4}$	-26.5	25.1	0.9231	-28.8	29.0
●	67%		$1.22 \cdot 10^{-4}$	-26.7	23.5	0.9376	-39.1	29.2
■	100%		$7.81 \cdot 10^{-5}$	-41.8	24.9	0.9615	-45.0	27.3

Table A.8: Experimental results for the white top-hat transformation using the **RCHM** with `float` precision. Best results from 200000 iterations are shown.

SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
⟋	9%	$WTH(f)$	$1.70 \cdot 10^{-4}$	-25.0	19.2	0.8740	-25.0	25.0
+	17%		$1.10 \cdot 10^{-4}$	-25.0	23.2	0.8957	-25.0	22.9
●	67%		$1.53 \cdot 10^{-4}$	-25.0	24.0	0.9167	-25.0	19.6
■	100%		$1.30 \cdot 10^{-4}$	-25.0	14.7	0.9366	-25.0	16.3

Table A.9: Experimental results for the white top-hat transformation using the **RCHM** mean with `float` precision and constraining $|P| > 25$ to $P = \pm 25$. Best results from 200000 iterations are shown.

SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
⟋	9%	$BTH(f)_{P,\pm 30}$	$4.29 \cdot 10^{-4}$	9.0	-7.4	0.8152	20.2	-17.9
+	17%		$3.19 \cdot 10^{-4}$	12.9	-14.9	0.8047	24.4	-31.3
●	67%		$2.52 \cdot 10^{-4}$	18.3	-17.8	0.8613	25.2	-26.8
■	100%		$6.03 \cdot 10^{-5}$	30.7	-30.7	0.9248	33.0	-31.6
⟋	9%	$BTH(f)$	$1.99 \cdot 10^{-4}$	11.2	-7.9	0.8011	27.3	-25.7
+	17%		$5.77 \cdot 10^{-4}$	11.4	-7.4	0.8218	25.5	-28.0
●	67%		$2.66 \cdot 10^{-4}$	17.9	-14.1	0.9025	30.4	-31.2
■	100%		$7.64 \cdot 10^{-5}$	32.0	-31.8	0.9603	38.2	-31.8

Table A.10: Experimental results for the black top-hat transformation using the **CHM** with `float` precision. Best results from 200000 iterations are shown.

SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
⟋	9%	$BTH(f)_{P,\pm 30}$	$2.80 \cdot 10^{-4}$	34.2	-28.9	0.8570	34.5	-28.8
+	17%		$7.64 \cdot 10^{-5}$	35.6	-30.0	0.9049	35.8	-30.3
●	67%		$1.31 \cdot 10^{-4}$	37.2	-30.4	0.9057	37.1	-30.4
■	100%		$5.48 \cdot 10^{-5}$	38.0	-34.0	0.9225	36.7	-32.0
⟋	9%	$BTH(f)$	$2.70 \cdot 10^{-4}$	25.9	-29.7	0.8965	34.5	-31.4
+	17%		$1.64 \cdot 10^{-4}$	27.1	-29.9	0.9006	29.1	-30.9
●	67%		$1.40 \cdot 10^{-4}$	30.4	-30.4	0.9409	34.3	-31.3
■	100%		$8.61 \cdot 10^{-5}$	36.8	-31.8	0.9546	38.7	-30.1

Table A.11: Experimental results for the black top-hat transformation using the **RCHM** with `float` precision. Best results from 200000 iterations are shown.

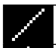
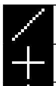
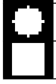
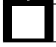
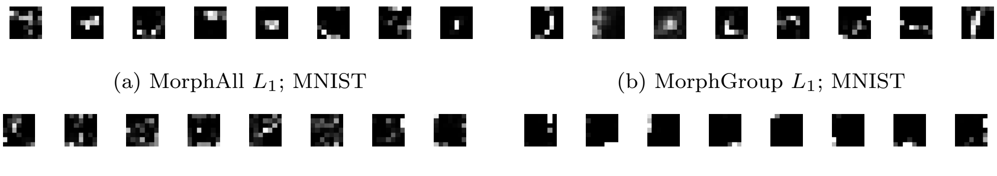
SE	Fullness	Target	MSE	$\mathrm{P}_{0,MSE}$	$\mathrm{P}_{1,MSE}$	$SSIM$	$\mathrm{P}_{0,SSIM}$	$\mathrm{P}_{1,SSIM}$
	9%	$BTH(f)$	$2.65 \cdot 10^{-4}$	25.0	−25.0	0.9199	25.0	−25.0
	17%		$1.10 \cdot 10^{-4}$	25.0	−25.0	0.9327	25.0	−25.0
	67%		$1.52 \cdot 10^{-4}$	25.0	−25.0	0.9280	25.0	−25.0
	100%		$1.14 \cdot 10^{-4}$	25.0	−25.0	0.9347	25.0	−20.7

Table A.12: Experimental results for the black top-hat transformation using the **RCHM** with `float` precision and constraining $|P| > 25$ to $P = \pm 25$. Best results from 200000 iterations are shown.

Structuring Elements

Learned structuring elements $\mathbf{w} = \sigma(\mathbf{w_r})$ in layer $L1, L2, L3$ for all four experiments are shown in Figures A.1, A.2 and A.3. All learned structuring elements are non-full structuring elements. MorphAll structuring elements have a large variety of structures and MorphGroup structuring elements have smaller variations of structures.



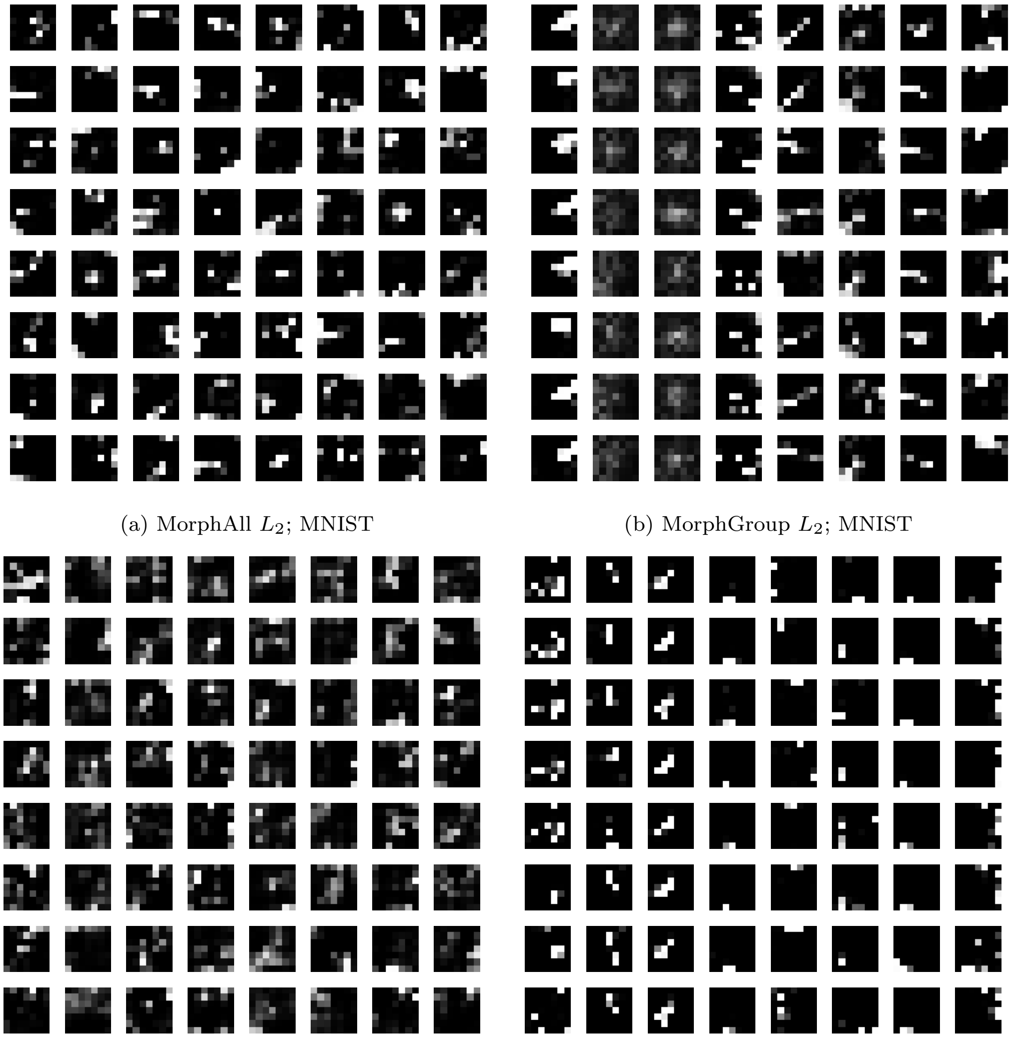
(a) MorphAll L_1; MNIST

(b) MorphGroup L_1; MNIST

(c) MorphAll L_1; Fashion-MNIST

(d) MorphGroup L_1; Fashion-MNIST

Figure A.1: Learned structuring elements $\mathbf{w} = \sigma(\mathbf{w_r})$ in layer $L1$ for all four experiments.



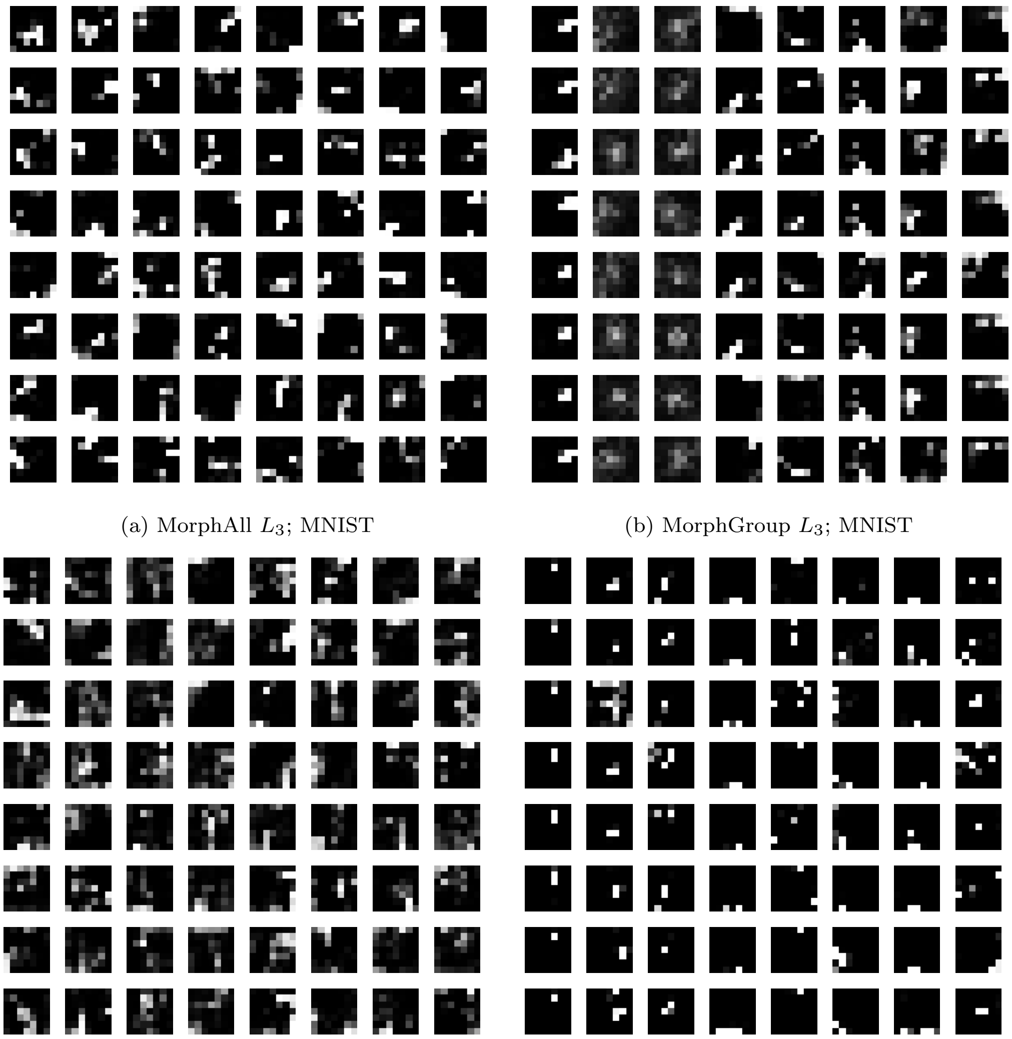
(a) MorphAll L_2; MNIST

(b) MorphGroup L_2; MNIST

(c) MorphAll L_2; Fashion-MNIST

(d) MorphGroup L_2; Fashion-MNIST

Figure A.2: Learned structuring elements $\mathbf{w} = \sigma(\mathbf{w_r})$ in layer $L2$ for all four experiments.



(a) MorphAll L_3; MNIST

(b) MorphGroup L_3; MNIST

(c) MorphAll L_3; Fashion-MNIST

(d) MorphGroup L_3; Fashion-MNIST

Figure A.3: Learned structuring elements $\mathbf{w} = \sigma(\mathbf{w_r})$ in layer $L3$ for all four experiments.