Integrating Morphological Operators in Neural Networks

Romain Hermary

September 5, 2021

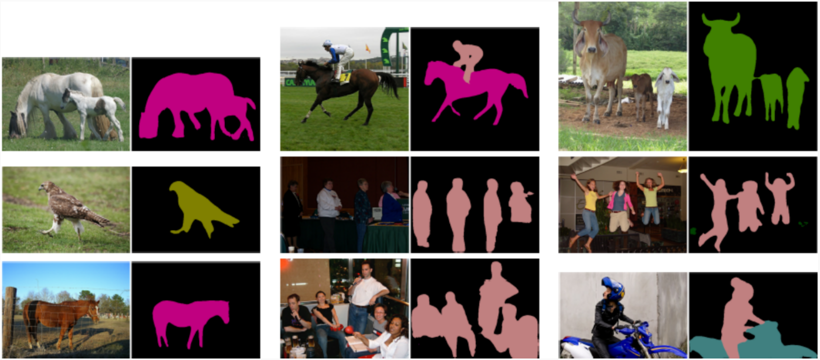
Laboratoire de Recherche et Développement de l'EPITA (LRDE)

Supervisors: Guillaume Tochon, Élodie Puybareau

Context

Image Segmentation

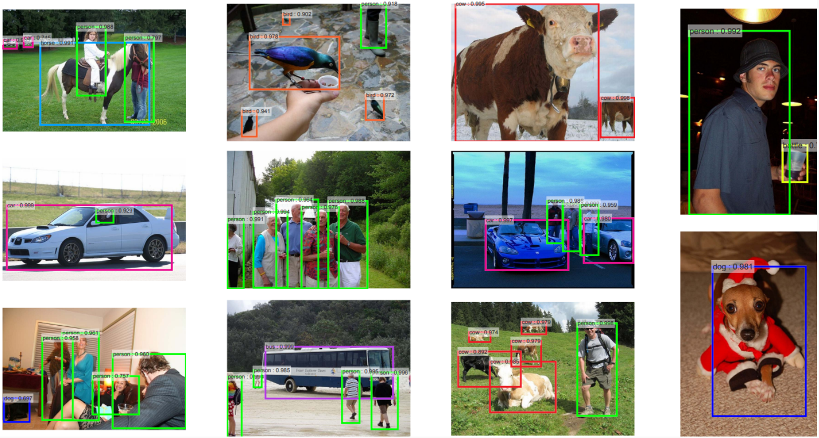
Detecting objects automatically within an image



Chen et al. (2017)

Object Detection and Classification

Detecting and classifying objects automatically



Ren et al. (2015)

Image Denoising

Reducing the *noise* in an image, increasing its quality

Liu et al. (2017)

Binary Mathematical Morphology

Binary Mathematical Morphology: Definitions

- A binary transformation of X by B in E
  - E is an Euclidean space
  - X a subset of E (*e.g.* a form, an image)
  - B a set, called a *structuring element*

$$\varepsilon_B(X) = \{z \in E, B_z \subseteq X\} \qquad \text{(erosion)}$$

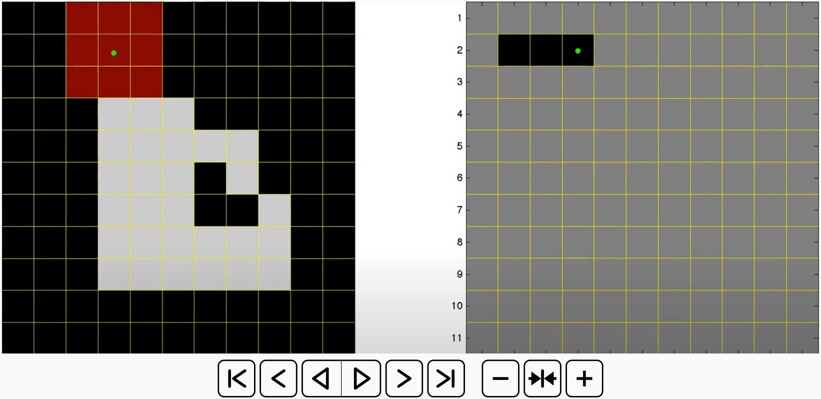
$$\delta_B(X) = \{z \in E, B_z \cap X \neq \emptyset\} \qquad \text{(dilation)}$$

- Where B_z is the *structuring element* centered in z

$$B_z = \{b + z \mid b \in B\}$$

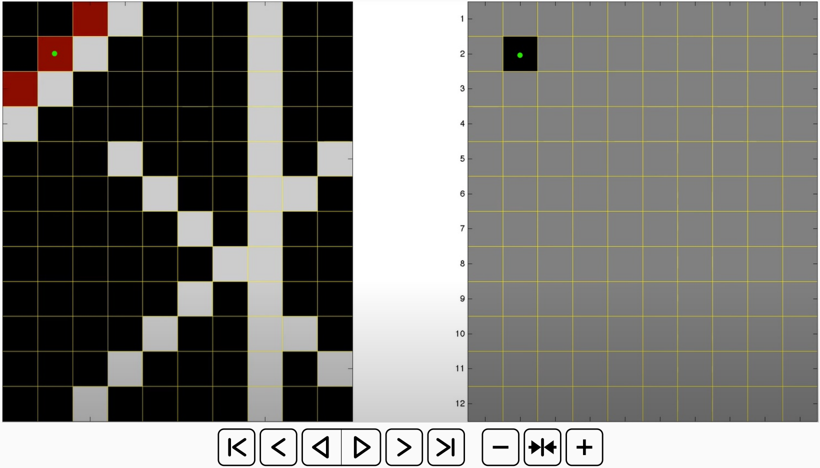
Matheron (1975); Serra (1983)

Binary Mathematical Morphology: Exemple



Peter Corke (YouTube)

Binary Mathematical Morphology: Exemple



Peter Corke (YouTube)

Higher Order Operators

We can combine these filters to create higher order operators:

- The opening $\gamma = \delta \circ \varepsilon$
  - Removes small objects (usually bright pixels) of an image
- The closing $\phi = \varepsilon \circ \delta$
  - Removes (fill) small holes

Ritter and Wilson (1996)

Grayscale Mathematical Morphology

Image are not usually binary

- Conventional images are stored as matrices of values between 0 and 255, you can see them as 3D landscapes
- ...

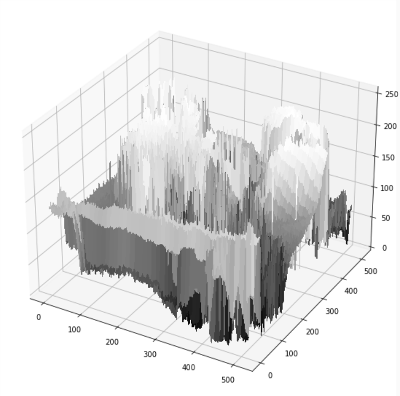
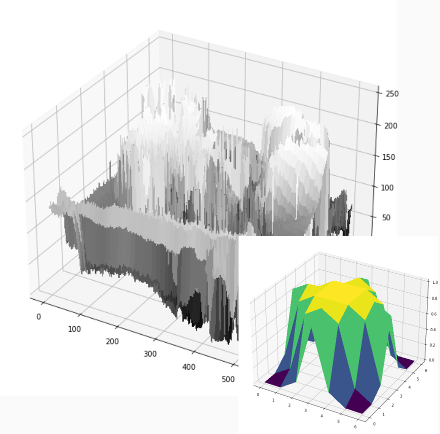


Image are not usually binary

- Conventional images are stored as matrices of values between 0 and 255, you can see them as 3D landscapes
- A structuring element too!



Grayscale Mathematical Morphology: Definitions

- For an image $f\colon E \subseteq \mathbb{Z}^2 \to \mathbb{R}$ and a structuring function $b : B \subseteq E \to \mathbb{R}$, operations are written as

$$(f \ominus b)(x) = \inf_{y \in E} \{f(y) - b(x-y)\} \qquad \text{(erosion)}$$

$$(f \oplus b)(x) = \sup_{y \in E} \{f(y) + b(x-y)\} \qquad \text{(dilation)}$$

- Also allows for binary structuring elements

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

Sternberg and Serra, 1980s

Grayscale Mathematical Morphology: Definitions

- For an image $f\colon E \subseteq \mathbb{Z}^2 \to \mathbb{R}$ and a structuring function $b : B \subseteq E \to \mathbb{R}$, operations are written as

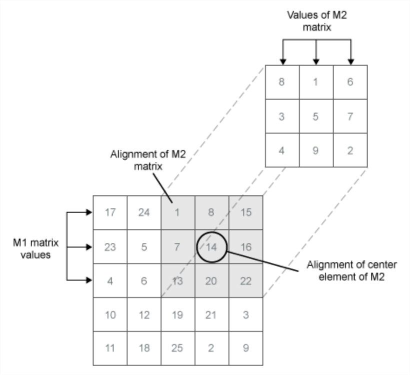
$$f \circ b = (f \ominus b) \oplus b \qquad \text{(opening)}$$

$$f \bullet b = (f \oplus b) \ominus b \qquad \text{(closing)}$$

- Every operation has the same properties than its binary homologous

Sternberg and Serra, 1980s

Superposition of matrices, taking the *maximum* (resp. *minimum*) of the additions (resp. subtractions) of the overlapped values

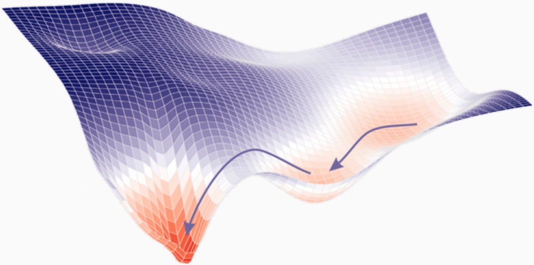


MathWorks

Supervised Learning

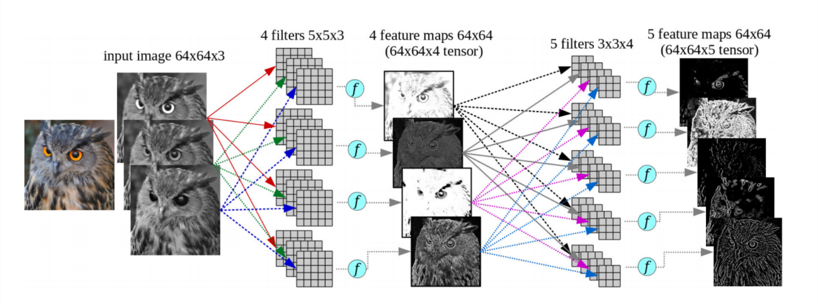
Intuition

- Automate complex processes (deduction, decision, distinction...)
- Give an input, compute the output, calculate the error, adjust the network parameters to minimize the error
- It is about finding the global minimum of the error function
- The method used is called Gradient Descent



Convolutional Neural Networks (CNN) (LeCun et al., 1989)

- Feature extraction
- **AlexNet** (Krizhevsky et al., 2012), **VGG-16** (Simonyan and Zisserman, 2015), **ResNet** (He et al., 2016), **GoogLeNet** (Szegedy et al., 2015)...



Ponti et al. (2017)

Mathematical Morphology and Supervised Learning

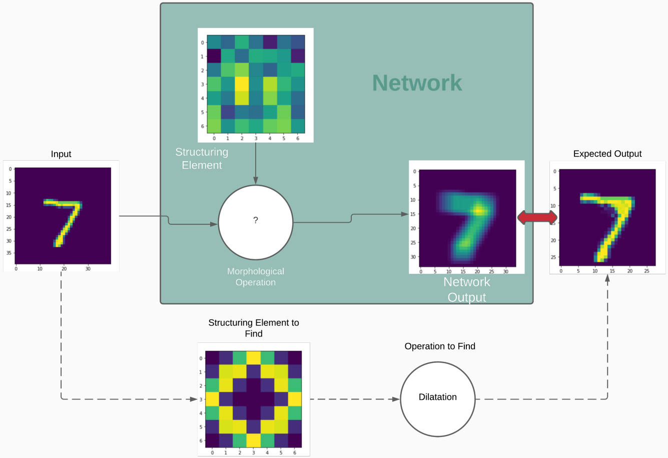
Problematic: Approach

Integrating Morphological Operators in Neural Networks

- For complex tasks, you need specific combinations of operators and structuring elements
- Finding this perfect combination is a tedious and very complicated process
- Letting a neural network learn them, is a very appealing strategy

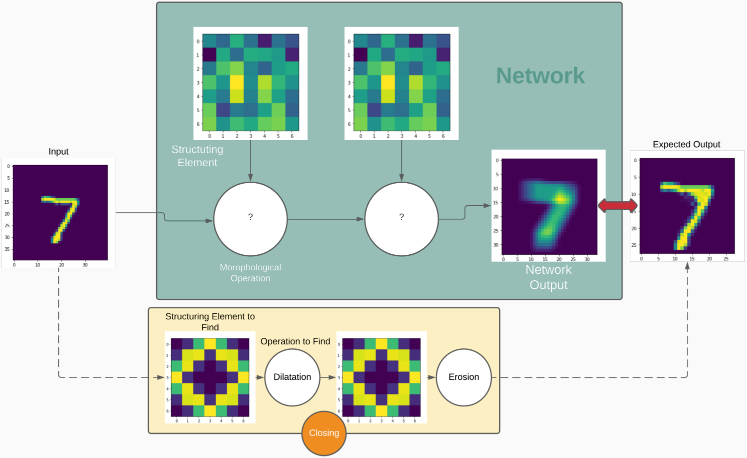
Objective

Design a neural network able to learn morphological operators and structuring elements



Objective

Design a neural network able to learn morphological operators and structuring elements



Problematic: Stake

Reminder of erosion and dilation formulas, which are operations we want the network to learn:

$$(f \ominus b)(x) = \inf_{y \in E} \{f(y) - b(x-y)\} \qquad \text{(erosion)}$$

$$(f \oplus b)(x) = \sup_{y \in E} \{f(y) + b(x-y)\} \qquad \text{(dilation)}$$

- But... min and max operations are not differentiable
- How can we do? Approximate those functions by differentiable ones

Approximation

$PConv$ (Masci et al., 2013)

- Counter-harmonic mean (CHM) (also known as Lehmer mean)

$$L_{p,w}(X) := \frac{\sum_{k=1}^{n} w_k.x_k^p}{\sum_{k=1}^{n} w_k.x_k^{p-1}}$$

$$\lim_{p\to+\infty}(L_{p,w}(X)) = max(X),$$

$$\lim_{p\to-\infty}(L_{p,w}(X)) = min(X)$$

- The p-convolution of an image f at pixel x for a given (positive) convolution kernel $w: W \subseteq E \to \mathbb{R}^+$ and parameter $p \in \mathbb{R}$ is defined as

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

$PConv$ (Masci et al., 2013)

$$\lim_{p \to +\infty} PConv(f, w, p)(x) = \sup_{y \in W(x)} \{f(y) + \frac{1}{p} \log\left(w(x-y)\right)\} = (f \oplus \frac{1}{p} \log(w))(x)$$

$$\lim_{p \to -\infty} PConv(f, w, p)(x) = \inf_{y \in W(x)} \{f(y) - \frac{1}{p} \log\left(w(x-y)\right)\} = (f \ominus \frac{1}{p} \log(w))(x)$$

Pitfalls of $PConv$ layer:

- $f(x) = 0$ and $p < 0 \Rightarrow f^p(x) \equiv NaN$
- $f(x) < 0$ and $p \in \mathbb{R} \Rightarrow f^p(x) \in \mathbb{C}$
- If $w(x) = 0$ or $f^p(x) = 0 \Rightarrow \frac{1}{(f^p * w)(x)} \equiv NaN$
- Need to rescale images between $[1, 2]$ before feeding them to any $PConv$ layer
- Tends to learn hollow structuring elements

$\mathcal{L}Morph$ (Kirszenberg et al., 2021)

- Also based on Counter-harmonic mean (CHM)

$$\mathcal{L}Morph(f, w, p)(x) = \frac{\sum_{y \in W(x)} (f(y) + w(x-y))^{p+1}}{\sum_{y \in W(x)} (f(y) + w(x-y))^{p}}$$

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

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

- Same issues as the $PConv$ layer

$\mathcal{S}Morph$ (Kirszenberg et al., 2021)

The $\alpha-$softmax function for some $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$

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

$$\lim_{\alpha \to +\infty} \mathcal{S}_\alpha(\mathbf{x}) = \max_i x_i$$

$$\lim_{\alpha \to -\infty} \mathcal{S}_\alpha(\mathbf{x}) = \min_i x_i$$

$$\mathcal{S}Morph(f, w, \alpha)(x) = \frac{\sum_{y \in W(x)} (f(y) + w(x-y))\, e^{\alpha(f(y)+w(x-y))}}{\sum_{y \in W(x)} e^{\alpha(f(y)+w(x-y))}}$$

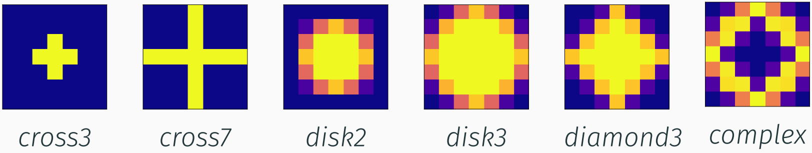
$$\lim_{\alpha \to +\infty} \mathcal{S}Morph(f, w, \alpha)(x) = (f \oplus w)(x)$$

$$\lim_{\alpha \to -\infty} \mathcal{S}Morph(f, w, \alpha)(x) = (f \ominus -w)(x)$$

No numerical issues

Experiments Environment

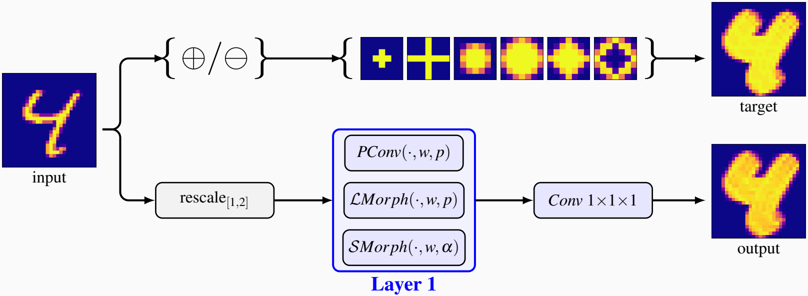
Structuring Elements



- 7×7 target grayscale structuring elements. All values range between 0 (deep blue) and 1 (yellow)
- Used to create target images and we expect the networks to learn them

Pipeline

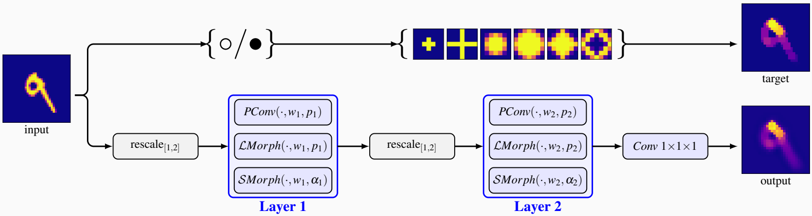
- Insertion of $\mathcal{L}Morph$, $\mathcal{S}Morph$, and $PConv$ layers into a network
- Using the **MNIST**[1] and **Fashion-MNIST**[2] databases for training



[1]Yann LeCun (LeCun et al., 1989)
[2]Zalando Research

Pipeline

- Loss function (MSE) $\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$
- RMSE function $\sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}}$



Semester Work: Initiation and Objectives

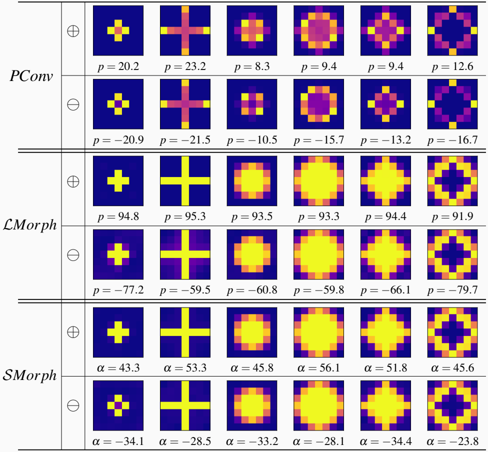
Introduction to the Subject

- Understanding and experiment mathematical morphology
- Refresh of deep learning mechanisms
- Documentation, literature
- Frameworks introductions (*scikit-learn*, *PyTorch*, *CUDA*…)
- Code appropriation, bug resolution

April, 9

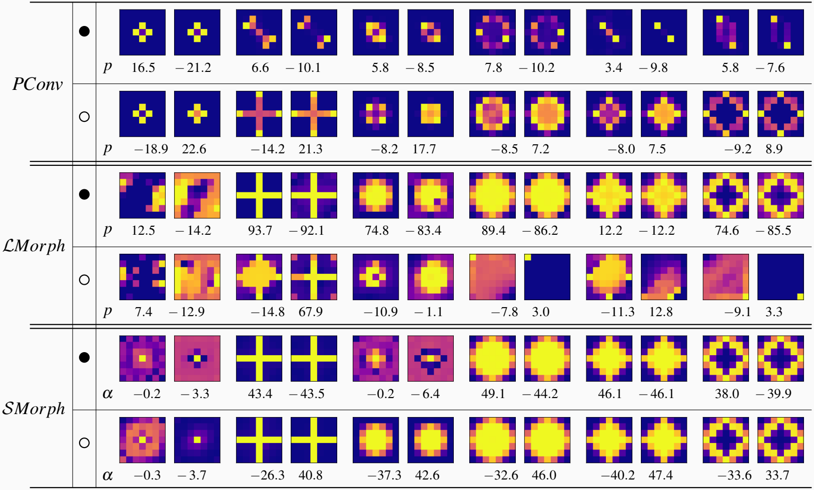
Previous Results (Kirszenberg et al., 2021)

Better results than $PConv$ and almost perfect learning for simple operations



Previous Results (Kirszenberg et al., 2021)

Globally better than $PConv$ but fails on some operations, especially for $\mathcal{L}Morph$



Main Objectives

- Reconduct the experiments
- Investigate on failing cases
- Make other and more extensive experiments
- Target of 30% additional data for publication

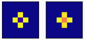
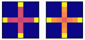
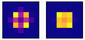
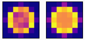
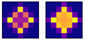
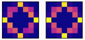
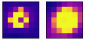
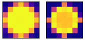
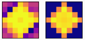
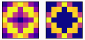
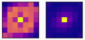
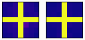
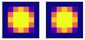
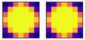
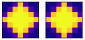
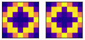
Semester Work: Experiments

Overview

- Writing scripts to automate runs and thus expend the results database
- Writing scripts to visualize the results and learning processes
- Compare actual results to those of the paper
- Try different approaches to study the results and eventually explain errors
- Conduct new experiments

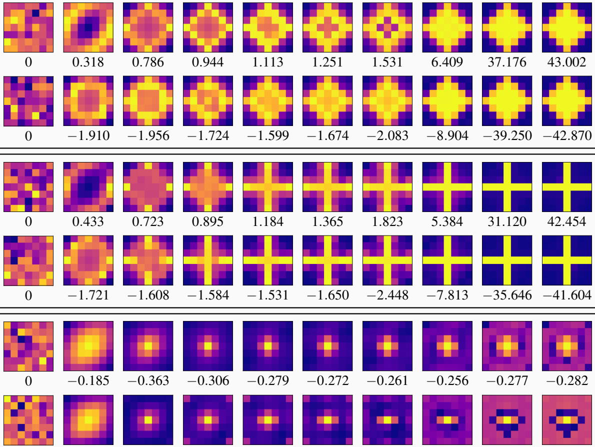
Comparison: Opening

- Could not reproduce aberrant results from $\mathcal{L}Morph$ (over 15 runs)
- $\mathcal{S}Morph$ and $\mathcal{L}Morph$ superior to $PConv$

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

Learning Structuring Elements: Evolution

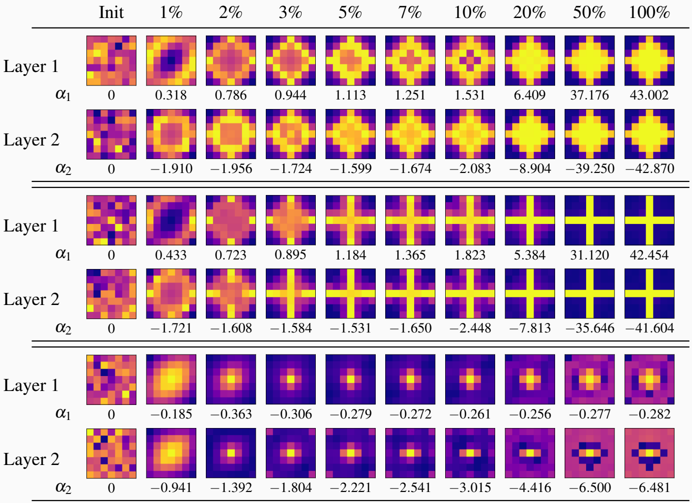
Runs on $\mathcal{S}Morph$; target structuring elements are *diamond3* (top row), *cross7* (middle row) and *disk2* (bottom row)



		Init	1%	2%	3%	5%	7%	10%	20%	50%	100%
Layer 1	α_1	0	0.318	0.786	0.944	1.113	1.251	1.531	6.409	37.176	43.002
Layer 2	α_2	0	−1.910	−1.956	−1.724	−1.599	−1.674	−2.083	−8.904	−39.250	−42.870
Layer 1	α_1	0	0.433	0.723	0.895	1.184	1.365	1.823	5.384	31.120	42.454
Layer 2	α_2	0	−1.721	−1.608	−1.584	−1.531	−1.650	−2.448	−7.813	−35.646	−41.604
Layer 1	α_1	0	−0.185	−0.363	−0.306	−0.279	−0.272	−0.261	−0.256	−0.277	−0.282
Layer 2	α_2	0	−0.941	−1.392	−1.804	−2.221	−2.541	−3.015	−4.416	−6.500	−6.481

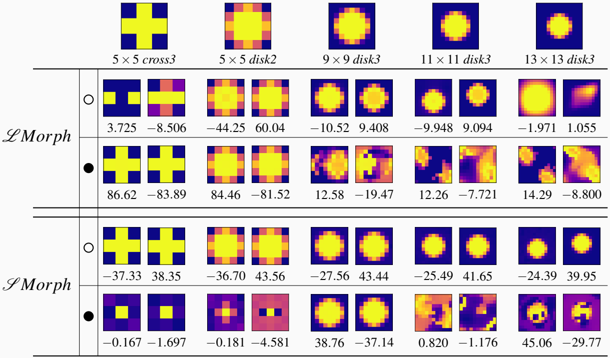
Learning Structuring Elements: Evolution

- Structuring elements seem to be learned from the support edges to the inside
- Is a structuring element not touching the support edges harder to learn?



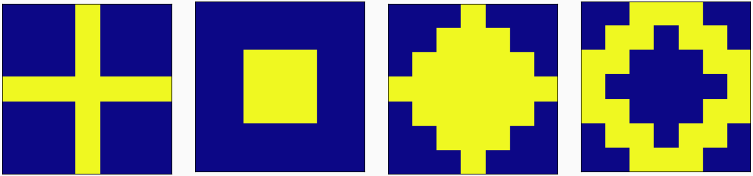
Border Track

- Augmenting or decreasing the filter size seems to affect the learning
- For a more precise conclusion it still needs a comparison to a *touching* structuring element with a bigger size
- Other tracks are promising, especially for the closing difficulties



$\mathcal{S}Morph$ and Binary Operations

- Only $\mathcal{S}Morph$ is equipped for those operations
- The following 7×7 target binary structuring elements were used. Yellow (resp. blue) corresponds to boolean TRUE (resp. FALSE)



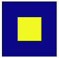
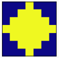
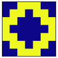
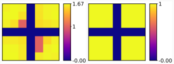
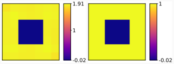
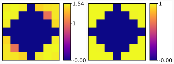
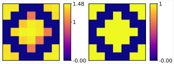
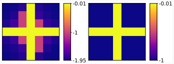
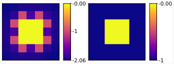
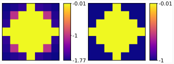
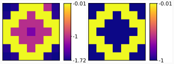
cross7 *bsquare* *bdiamond* *bcomplex*

- The network is expected to learn 0 for TRUE parts and $-\infty$ otherwise

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

Binary Operations: Results

$\mathcal{S}Morph$ is a *champ*

	 *cross7*	 *bsquare*	 *bdiamond*	 *bcomplex*
w				
α	-10.12	-9.80	-10.59	-10.43
RMSE	1.75×10^{-2}	4.65×10^{-2}	1.72×10^{-2}	1.08×10^{-2}
LOSS	2.9×10^{-7}	3.1×10^{-7}	2.9×10^{-7}	2.9×10^{-7}
EPOCHS	31	31	26	32
w				
α	9.10	8.88	9.96	10.11
RMSE	3.96×10^{-2}	1.78×10^{-2}	6.57×10^{-2}	5.45×10^{-2}
LOSS	4.5×10^{-7}	4.0×10^{-7}	4.6×10^{-7}	4.9×10^{-7}
EPOCHS	34	38	48	31

First row is erosion, second row is dilation

Journal of Mathematical Imaging and Vision (JMIV), 2021

Journal of Mathematical Imaging and Vision manuscript No.
(will be inserted by the editor)

Learning grayscale mathematical morphology with smooth morphological layers

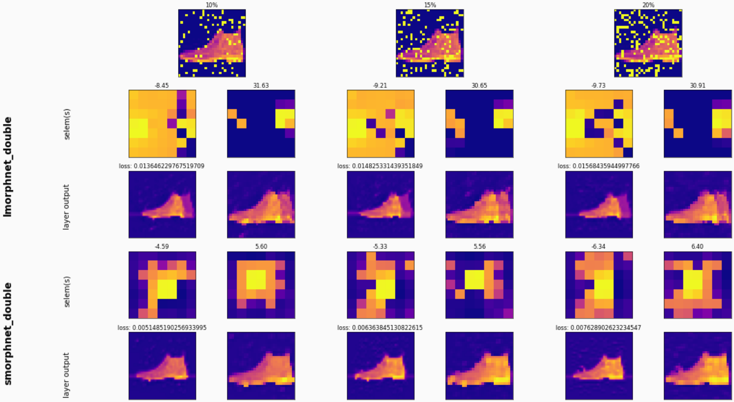
Romain Hermary · Guillaume Tochon · Élodie Puybareau · Alexandre Kirszenberg · Jesús Angulo

What Is Not In The Paper

Ranges Problems

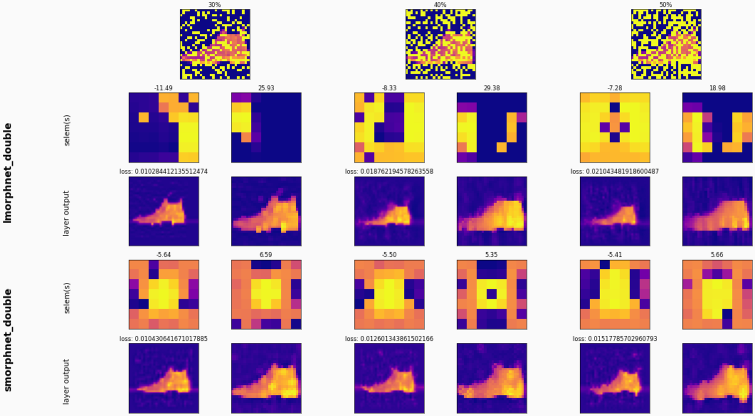
- Learned filters are not in the $[0;1]$ range, but have the correct visual shape (and give great results)
- Scale bias layer seems to compensate everything
- Especially needed because of image rescaling between $[1;2]$, input normalization and formulas definition domain ($\mathcal{L}Morph$, $PConv$)

Promising results



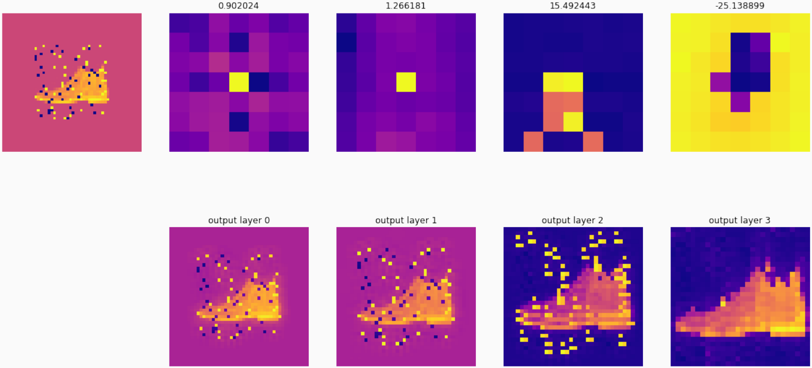
Denoising: Salt

Promising results



Denoising: Salt and Pepper ($\mathcal{S}Morph$ 10%)

Promising results



What's next?

- Fail reproduction ($\mathcal{S}Morph$)
- Changing/tweaking the formulas to solve some issues
- Other losses
- Other morphological operations (top-hat, bot-hat, hit-miss...)
- Symmetric and asymmetric structuring elements
- Real pictures
- Other approximation formulas

References

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587.

Franchi, G., Fehri, A., and Yao, A. (2020). Deep morphological networks. *Pattern Recognit.*, 102:107246.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Kirszenberg, A., Tochon, G., Puybareau, É., and Angulo, J. (2021). Going beyond p-convolutions to learn grayscale morphological operators. In *DGMM*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. In *NIPS*.

Liu, S., Wu, G., Liu, H., and Zhang, X. (2017). Image restoration approach using a joint sparse representation in 3d-transform domain. *Digit. Signal Process.*, 60:307–323.

Masci, J., Angulo, J., and Schmidhuber, J. (2013). A learning framework for morphological operators using counter-harmonic mean. In *ISMM*.

Matheron, G. (1975). Random sets and integral geometry.

Ponti, M., Ribeiro, L. S. F., Nazaré, T. S., Bui, T., and Collomosse, J. (2017). Everything you wanted to know about deep learning for computer vision but were afraid to ask. *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pages 17–41.

Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149.

Ritter, G. and Wilson, J. N. (1996). Handbook of computer vision algorithms in image algebra.

Serra, J. (1983). Image analysis and mathematical morphology.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.

The End

Any Questions?

Previous Results (Kirszenberg et al., 2021)

			cross3	*cross7*	*disk2*	*disk3*	*diamond3*	*complex*	
PConv	$\oplus$	LOSS	0.5	0.8	5.1	6.4	6.8	3.3	$\times 10^{-4}$
		RMSE	0.41	1.42	2.22	3.09	2.80	2.38	
	$\ominus$	LOSS	2.4	0.62	13	2.6	5.2	1.2	$\times 10^{-5}$
		RMSE	0.82	1.55	2.82	3.77	3.63	2.76	
$\mathcal{L}$*Morph*	$\oplus$	LOSS	0.84	1.2	0.78	1.2	1.2	2.1	$\times 10^{-5}$
		RMSE	**0.02**	**0.00**	**0.02**	**0.05**	**0.01**	**0.48**	
	$\ominus$	LOSS	1.1	**0.37**	1.3	**0.37**	0.58	**1.1**	$\times 10^{-6}$
		RMSE	**0.44**	0.63	0.37	0.29	0.38	**0.48**	
$\mathcal{S}$*Morph*	$\oplus$	LOSS	**1.8**	**4.0**	**2.5**	**4.9**	**3.9**	**1.7**	$\times \mathbf{10^{-7}}$
		RMSE	0.10	0.13	0.08	0.06	0.09	0.51	
	$\ominus$	LOSS	210	4.1	**0.8**	4.1	**4.3**	**9.5**	$\times \mathbf{10^{-7}}$
		RMSE	0.78	**0.15**	**0.09**	**0.05**	**0.09**	0.51	

Better scores in **bold**

Previous Results (Kirszenberg et al., 2021)

$\mathcal{S}Morph$ gets the majority of bests results

			cross3	*cross7*	*disk2*	*disk3*	*diamond3*	*complex*	
PConv	●	LOSS	0.09	5.1	1.7	1.0	5.4	4.9	$\times 10^{-3}$
		RMSE	**0.83** | **0.81**	3.94 | 3.97	2.82 | 3.07	4.44 | 4.64	4.34 | 4.51	3.68 | 3.65	
	○	LOSS	**0.86**	0.76	4.8	3.9	4.4	2.3	$\times 10^{-4}$
		RMSE	**0.91** | **0.32**	1.45 | 0.87	2.70 | 2.49	3.10 | 2.52	3.15 | 2.43	2.10 | 2.08	
$\mathcal{L}Morph$	●	LOSS	**0.5**	0.13	**2.0**	0.18	6.2	0.14	$\times 10^{-4}$
		RMSE	3.58 | 5.12	**0.01** | 0.63	**0.14** | **1.16**	**0.07** | 0.40	0.75 | 0.95	**0.47** | 0.99	
	○	LOSS	8.7	36	2.0	2.7	4.6	2.5	$\times 10^{-3}$
		RMSE	3.17 | 4.74	2.90 | 1.20	1.99 | 1.61	2.81 | 5.56	1.66 | 3.54	2.72 | 3.98	
$\mathcal{S}Morph$	●	LOSS	1700	**0.31**	4400	**0.42**	**0.37**	**0.17**	$\times 10^{-6}$
		RMSE	2.10 | 3.59	0.14 | **0.08**	2.78 | 3.68	0.08 | **0.01**	**0.08** | **0.10**	0.52 | **0.52**	
	○	LOSS	4700	**0.14**	**0.22**	**0.25**	**0.24**	**0.09**	$\times 10^{-6}$
		RMSE	3.40 | 1.63	**0.18** | **0.02**	**0.05** | **0.01**	**0.02** | **0.02**	**0.02** | **0.01**	**0.51** | **0.52**	

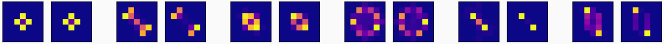
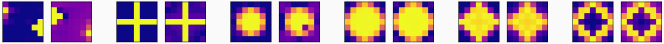
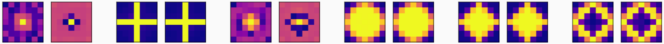
Comparison: Erosion

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

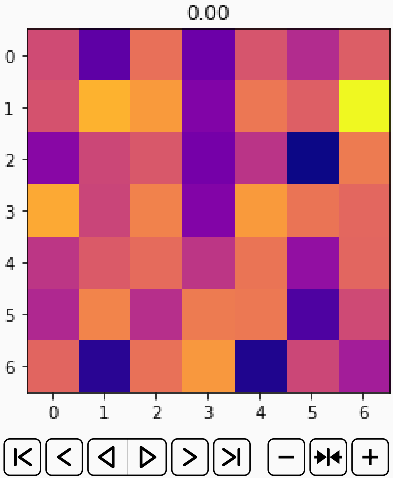
Comparison: Dilation

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

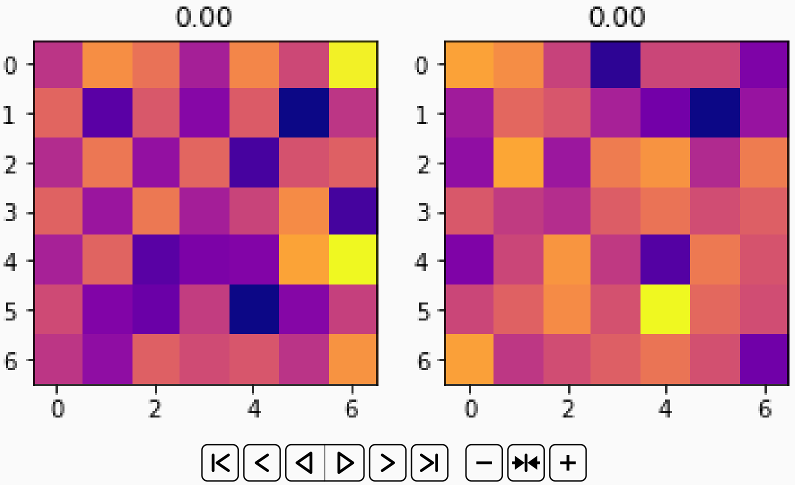
Comparison: Closing

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

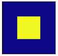
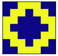
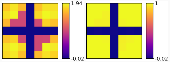
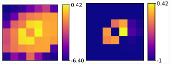
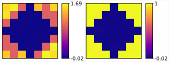
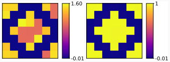
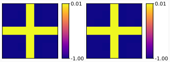
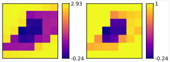
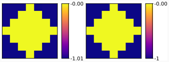
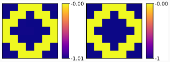
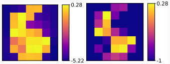
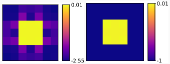
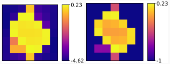
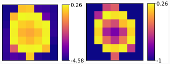
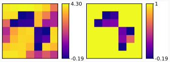
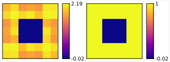
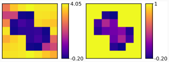
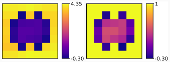
Learning A Structuring Element



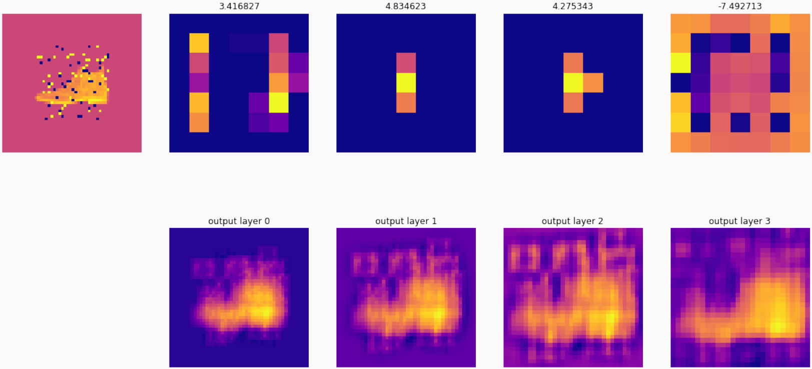
Learning A Structuring Element



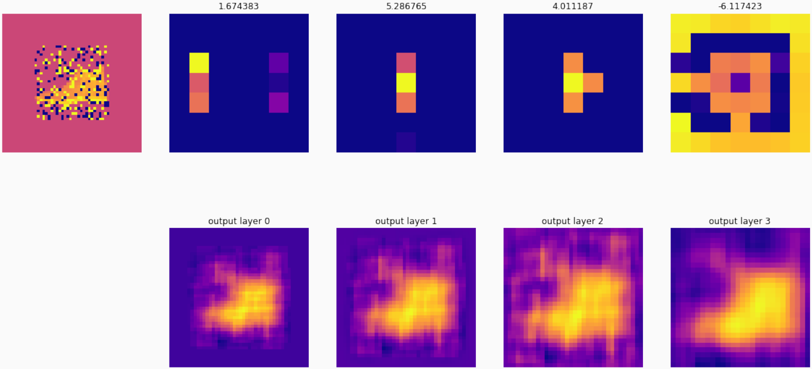
Binary Operations: Results

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

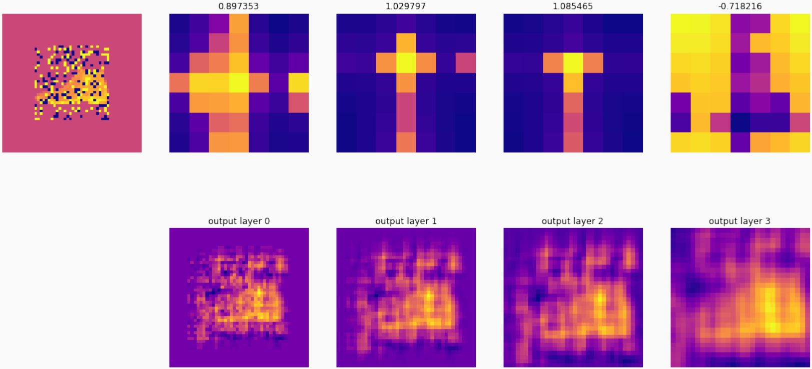
Denoising: Salt and Pepper ($\mathcal{L}Morph$ 10%)



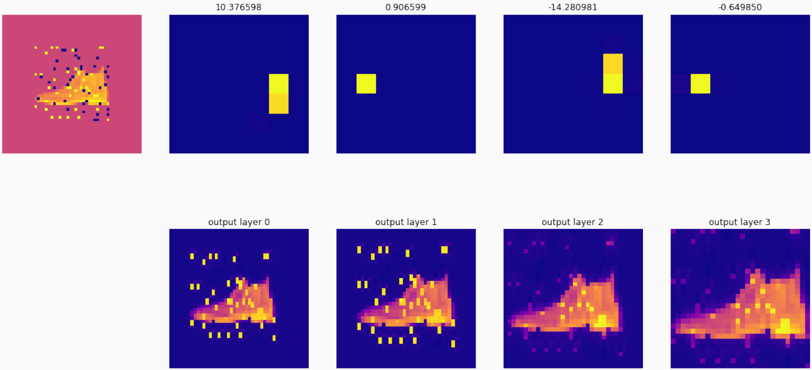
Denoising: Salt and Pepper ($\mathcal{L}Morph$ 30%)



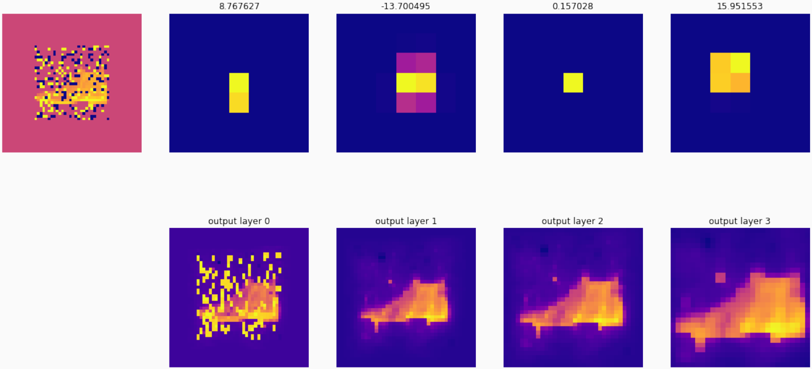
Denoising: Salt and Pepper ($\mathcal{S}Morph$ 30%)



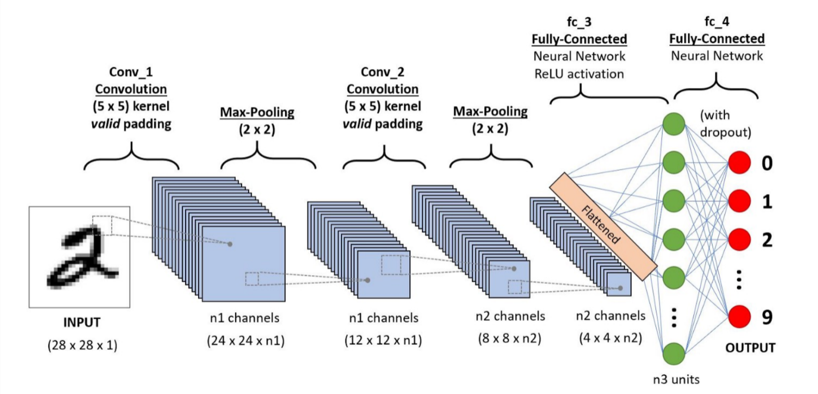
Denoising: Salt and Pepper ($PConv\ 10\%$)



Denoising: Salt and Pepper ($PConv\ 30\%$)

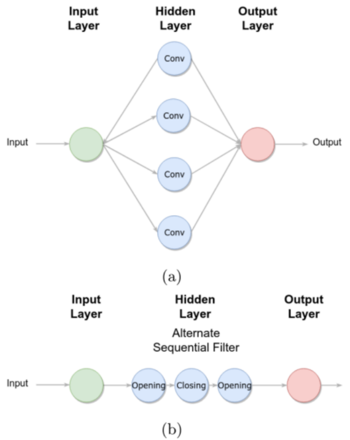


Convolutional Neural Network (CNN)



Towards Data Science

Deep Morphological Networks (Franchi et al., 2020)



	MSE (↘)		PSNR (↗)		SSIM (↗)	
Noise level	DCNN	DMNN	DCNN	DMNN	DCNN	DMNN
10%	0.015	**0.009**	18.71	**20.89**	0.9994	**0.9995**
20%	0.022	**0.016**	16.64	**18.34**	0.9988	**0.9990**
30%	0.030	**0.022**	15.38	**16.89**	0.9980	**0.9985**
40%	0.038	**0.029**	14.35	**15.66**	0.9972	**0.9978**
50%	0.047	**0.040**	13.44	**14.26**	0.9961	**0.9966**
60%	0.058	**0.051**	12.59	**13.19**	0.9945	**0.9953**
70%	0.069	**0.063**	11.86	**12.29**	0.9930	**0.9934**
80%	0.080	**0.079**	11.25	**11.36**	0.9907	**0.9908**
90%	0.089	**0.088**	10.83	**10.85**	0.9888	**0.9894**