University of Amsterdam
computer vision lab

BMVC 2022

MorphPool: Efficient Non-linear Pooling & Unpooling in CNNs

Rick Groenendijk, Leo Dorst & Theo Gevers

Goal

Down-sampling using max pooling can be formalised and improved using *mathematical morphology*, the mathematics of contact. In addition to down-sampling, encoder-decoder networks also require up-sampling. Seminal works [1, 2, 3] use a combination of unpooling and transposed convolutions to this end. In our article, both pooling *and* unpooling in CNNs are formalised as fully morphological operations without any linear interpolation scheme; the full procedure is called **MorphPool**.

Code is available at https://github.com/rickgroen/morphpool

Method

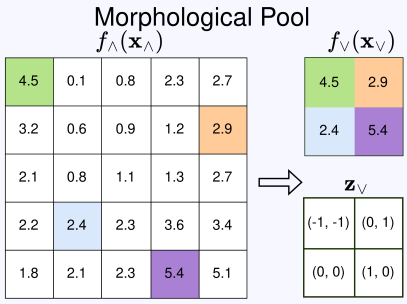
Max pool is a discrete local strided maximum operator

$$f_\vee(\mathbf{x}) = \max_{\mathbf{z}\in\mathbf{Z}} \left(f_\wedge(s\mathbf{x} + \mathbf{z}) \right), \quad (1)$$

where s is the stride and z denotes the indicator set over the local neighbourhood. By algebraic manipulation it can be shown that the max pool is in fact a special case of the morphological dilation, defined as

$$f_\vee(\mathbf{x}_\vee) = \bigvee_{\mathbf{z}} f_\wedge(\mathbf{x}_\wedge - \mathbf{z}) + h(\mathbf{z}). \quad (2)$$

Here h is the structuring element, which is flat in the case of max pooling, but can be freely parameterised.

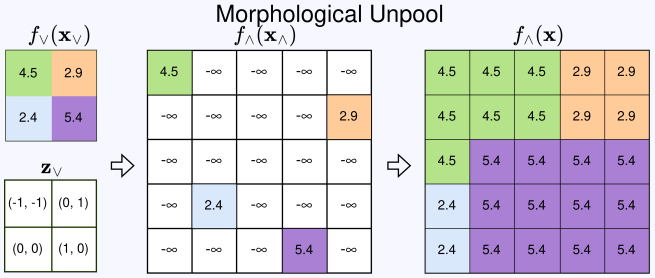


Morphological unpooling is a two-stage process. First, place back values by means of provenance:

$$f_\wedge(\mathbf{x}_\wedge - \mathbf{z}_\vee) = \begin{cases} f_\vee(\mathbf{x}_\vee) & \forall \mathbf{x}_\vee | (\exists \mathbf{x}_\wedge \in \mathbf{x}) \; [\mathbf{x}_\wedge = s\mathbf{x}_\vee] \\ -\infty & \text{otherwise.} \end{cases} \quad (3)$$

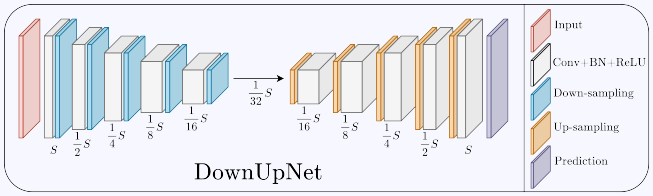
Second, fill $-\infty$ using the upper bound of the sparse values by morphological interpolation:

$$f_\wedge(\mathbf{x}) = \bigvee_{\mathbf{w}} f_\vee(\mathbf{x} - \mathbf{z}_\vee - \mathbf{w}) + h(\mathbf{w}), \quad (4)$$



Results

MorphPool was evaluated on semantic segmentation and depth auto-encoding for 3 datasets: NYUv2, SUN-RGBD, and 2D-3D-S. It was compared to strided convolution and up-sampling by interpolation (Linear) and to pooling combined with transposed convolution (Standard Pool) as in [2, 3]. We use a general-purpose encoder-decoder network without any task-specific modules to isolate performance purely due to sampling.



	#params	Semantic Segmentation mIoU ↑	acc ↑	bf ↑	Depth Auto-encoding ARD ↓	RMS ↓	$\delta_t < 1.25$ ↑
Linear	12.5M	0.349	0.711	0.308	0.193	0.563	0.715
Standard Pool	0	0.208	0.643	0.255	0.239	0.751	0.605
MorphPool	0	**0.382**	**0.735**	**0.346**	**0.177**	**0.534**	**0.734**

For depth input, MorphPool outperforms the other sampling methods at no extra parameter cost. In the case of segmentation, semantic boundaries are more clearly delineated as measured by the Boundary F1 score (bf).

It is possible to parameterise the MM kernel $h(\cdot)$ in **MorphPool** to make it more expressive.

	#params	SUN-RGBD mIoU ↑	acc ↑	bf ↑	NYUv2 mIoU ↑	acc ↑	bf ↑
Flat	0	0.382	0.735	0.346	0.323	0.607	0.200
Parabolic	4.0K	0.396	0.741	0.351	0.348	0.627	0.205
General	67.5K	**0.398**	**0.745**	**0.352**	**0.353**	**0.632**	**0.207**

RGB does not just express geometry but also illumination, which has aspects that are well described linearly, and then **MorphPool** has comparatively less effect. However, semantic boundaries are of significantly better quality for both RGB and depth input. In any case, the number of parameters required by **MorphPool** is significantly less than the Linear method.

	depth mIoU ↑	acc ↑	bf ↑	RGB mIoU ↑	acc ↑	bf ↑
Linear	0.349	0.711	0.308	0.381	0.694	0.314
Standard Pool	0.208	0.643	0.255	0.298	0.646	0.270
MorphPool (General)	**0.398**	**0.745**	**0.352**	**0.388**	**0.698**	**0.337**

Conclusions

We show that **MorphPool** can increase performance at much reduced parameter counts for semantic segmentation and depth auto-encoding. The beneficial effects are most pronounced on depth data and at semantic boundaries; mathematical morphology appears to be the natural language to express such nonlinear geometry.

References

[1] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *2011 international conference on computer vision*, pp. 2018–2025, IEEE, 2011.

[2] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.