nnMorpho, a PyTorch library for Mathematical Morphology operators

Gonzalo Romero-García, Carlos Agon, Isabelle Bloch

nnMorpho, a PyTorch library for Mathematical Morphology operators

Gonzalo Romero-García[1], Carlos Agón[1], Isabelle Bloch[2]
[1]Sorbonne Université, CNRS, IRCAM, STMS, Paris, France
[2]Sorbonne Université, CNRS, LIP6, Paris, France

Introduction

This poster presents nnMorpho, a Python library that implements Mathematical Morphology (MM) operators in the PyTorch framework.

A first motivation is to enable to plug MM operators in Deep Learning pipelines, allowing even the structuring elements to be learned.

The second motivation is to take advantage of PyTorch tensors to accelerate the operators with the capability of GPUs, since MM operators are highly parallelizable.

Related software

kornia – a PyTorch Open-Source library for Computer Vision
data structure: PyTorch tensors

SciPy – the standard scientific computing library for Python
data structure: NumPy arrays

OpenCV – a computer vision library with a Python wrapper
data structure: NumPy arrays

scikit-image – a library for image processing in Python
data structure: NumPy arrays

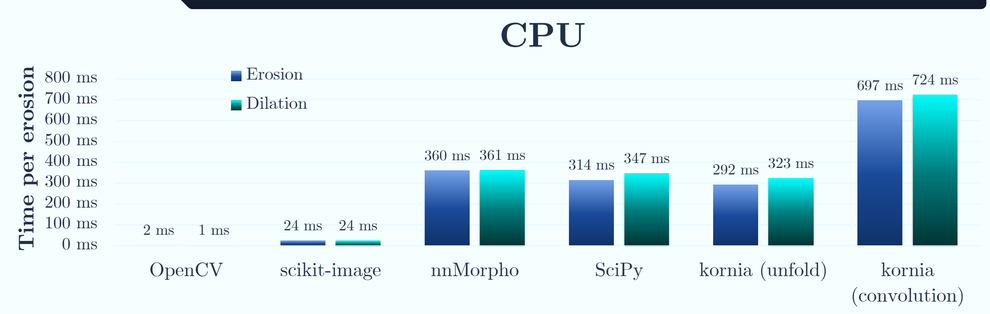
Motivation

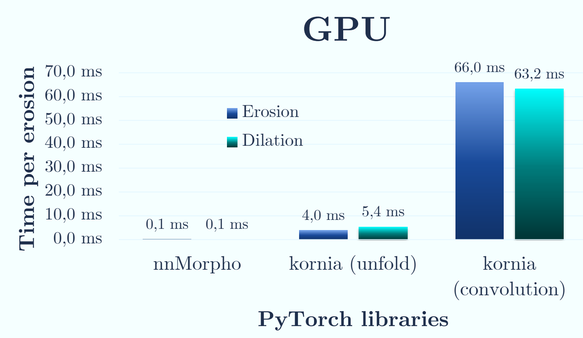
- Work with PyTorch tensors
- Plug MM operators in DL pipelines
- Improve speed with GPU computations
- Improve non-standard topologies (e.g., cylindric)
- Include all morphological parameters (border and origin)
- Implement parallel algorithms in CUDA kernels

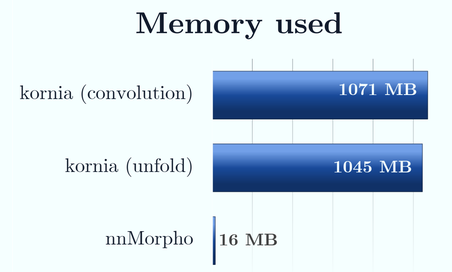
Content

- Greyscale MM operators in 3 flavors:
  - Pure Python functions
  - PyTorch functions (that compute and store gradients)
  - PyTorch modules (with the structuring element as a learning parameter
- Two types of border: geodesic and Euclidean
- Binary MM in two topologies: flat and cylindric

Performance







All the computations are made on a PC with:
OS: `Ubuntu 20.04.3 LTS`
CPU: `Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz`
GPU: `Nvidia GeForce GTX 1050 Mobile`

Parameters:
Input size: `(1000, 1000)`
Structuring element size: `(15, 15)`

Features

Feature	SciPy	scikit-image	OpenCV	kornia	nnMorpho
PyTorch	✗	✗	✗	✓	✓
Non-flat structuring element	✓	✓	✗	✓	✓
GPU capability	✗	✗	✗	✓	✓
Border parameter	✗	✗	✗	✓	✓
Cylindric topology	✗	✗	✗	✗	✓
Batch processing	✗	✗	✗	✓	✓
More than 2D	✓	✗	✗	✗	✗
Computation of gradients	✗	✗	✗	✓	✓
Origin parameter	✓	✓	✗	✓	✓

Future work

One of the intentions of nnMorpho is to extend the variety of MM operators to more complex ones trying, when possible, to use parallel algorithms implemented in CUDA.

Another intention is to extend the topologies (for instance, to toroidal topology) and the dimensions (3D or even n-D).

Another future line of work is to implement SIMD instructions for the CPU computations in order to match the speed of OpenCV or scikit-image without GPUs.