

# Intégration des Opérateurs de la Morphologie Mathématique dans un Réseau de Neurones

Romain Hermary

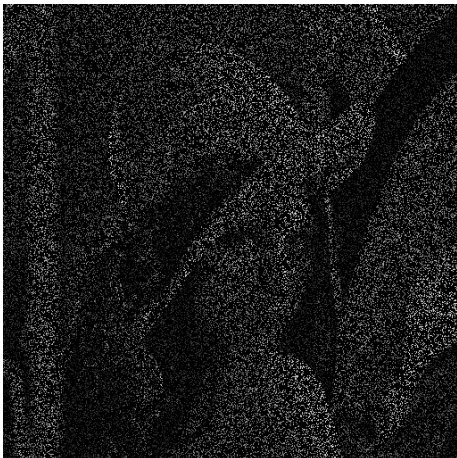
Laboratoire de Recherche et Développement de l'EPITA (LRDE)

Encadrants : Élodie Puybareau, Guillaume Tochon

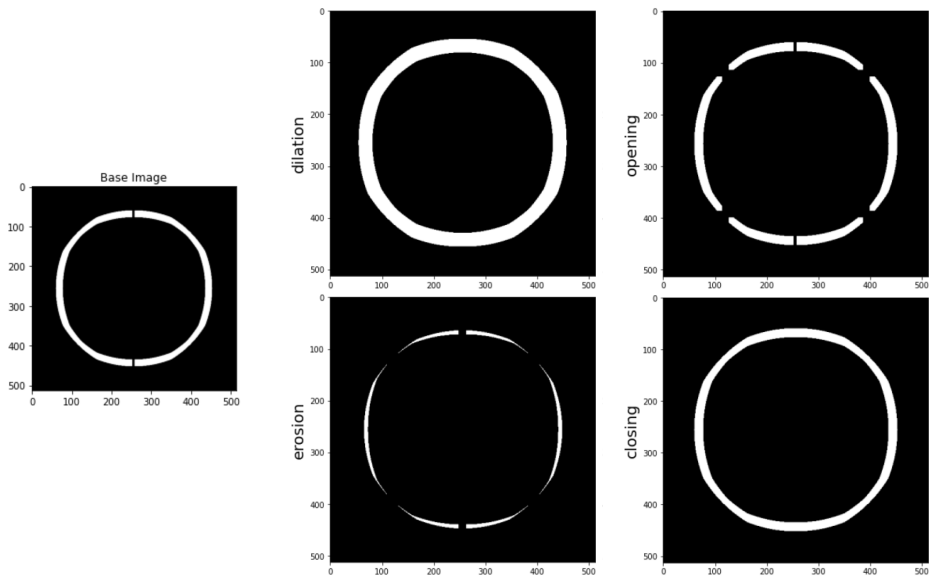
Mars 2021



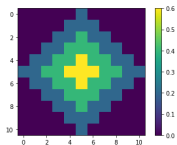
# À quoi ça sert ?



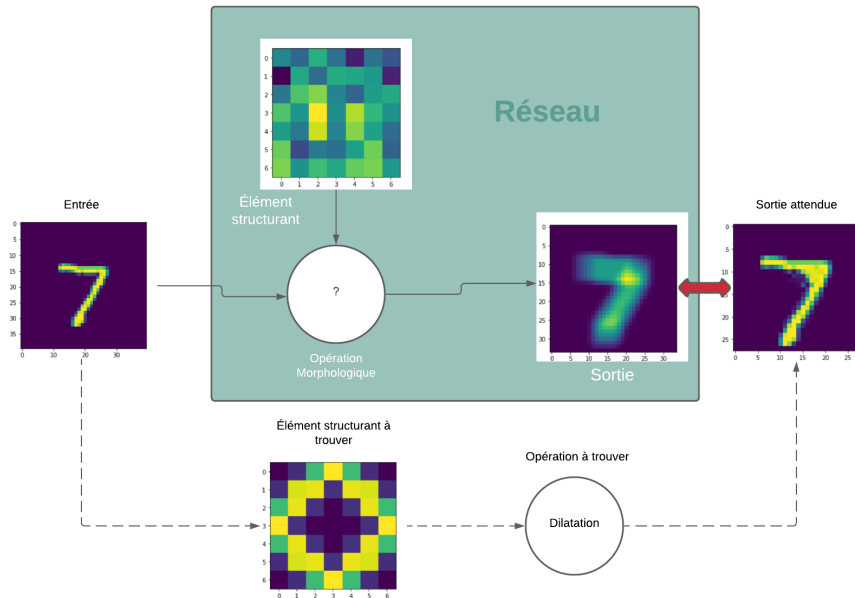
# Morphologie Mathématique : Opérations de base



# Exemple Concret



# Un Réseau Morphologique



## Travail Effectué :

- Lire et comprendre en détail les articles recommandés, chercher d'autres sources
- Découvrir les bibliothèques scikit-learn, PyTorch, CUDA
- Coder et optimiser un algorithme de MM pour un élément structurant en niveaux de gris
- Commencer à prendre en main le code et essayer de l'installer sur *node9*

## Travail à venir :

- Prendre en main tout le code
  - Python (PyTorch) : 1.3k
  - C++ (CUDA, PyTorch) : 650
  - Rust : 1.7k
- Réparer tout le code
- Refaire les tests
- Avoir le point de vue extérieur d'un chercheur du Centre de Morphologie Mathématique de MINES ParisTech

# PConv Layer [Masci et al., 2012]

- Moyenne Contre Harmonique d'ordre  $p$   
(Moyenne de Lehmer)

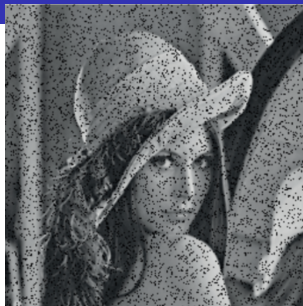
Soit  $n \in \mathbb{N}$ ,  $\forall i \in \{1, 2, \dots, n\}$  et  $x_i, w_i \in \mathbb{R}^+$ ,

$$X := \{x_1, x_2, \dots, x_n\}, \quad \forall p \in \mathbb{R}$$

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

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

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

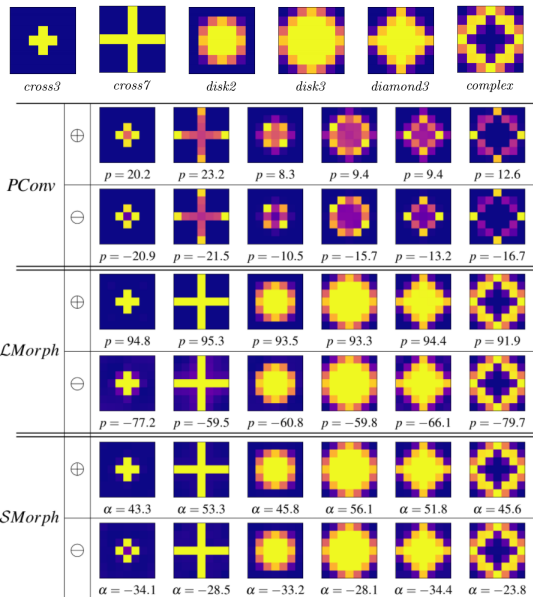


$$\mathcal{LMorph}(f, w, p)(x) = \frac{\sum_{y \in W(x)} (f(y) + w(x - y))^{p+1}}{\sum_{y \in W(x)} (f(y) + w(x - y))^p}$$

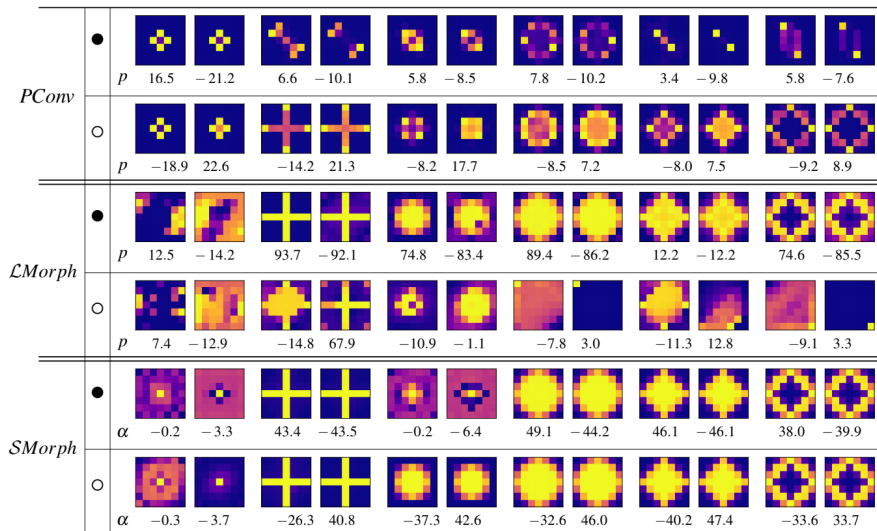
$$\mathcal{SMorph}(f, w, \alpha)(x) = \frac{\sum_{y \in W(x)} (f(y) + w(x - y)) e^{\alpha(f(y) + w(x - y))}}{\sum_{y \in W(x)} e^{\alpha(f(y) + w(x - y))}}$$



# Going Beyond P-Convolutions [Kirszenberg et al., 2021]



# Going Beyond P-Convolutions [Kirszenberg et al., 2021]



## Travail Effectué :

- Documentation CUDA et programmation GPU
- Documentation PyTorch
- Prise en main du code :
  - Python (PyTorch) : 1.3k
  - C++ (CUDA, PyTorch) : 650
- Réparation du code
- Installation des bibliothèques
- Début des tests d'apprentissage sur MNIST
- Documentation HDF et visualisation des résultats de l'apprentissage

## Travail à venir :

- Continuer les tests pour arriver aux résultats précédents
- Commencer à chercher comment améliorer les fonctions proposées (principalement *Smorph*)
- Avoir le point de vue extérieur d'un chercheur du Centre de Morphologie Mathématique de MINES ParisTech

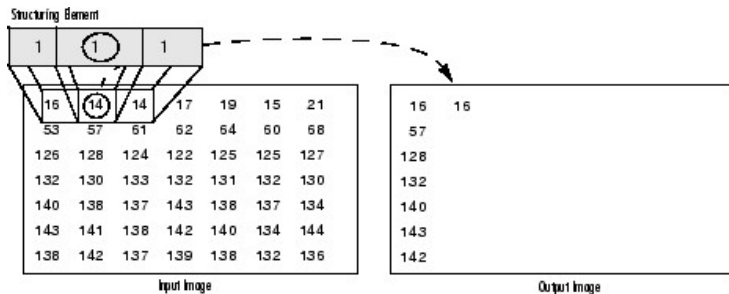


Kirszenberg, A., Tochon, G., Puybureau, E., and Angulo, J. (2021).  
Going beyond p-convolutions to learn grayscale morphological operators.  
*arXiv preprint arXiv:2102.10038v1*.



Masci, J., Angulo, J., and Schmidhuber, J. (2012).  
A learning framework for morphological operators using counter-harmonic mean.  
*arXiv preprint arXiv:1212.2546v1*.

# Opération Morphologique



$$(f \oplus b)(x) = \sup_{y \in E} [f(y) + b(x - y)]$$

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

$$f \circ b = (f \ominus b) \oplus b$$

$$f \bullet b = (f \oplus b) \ominus b$$

Source : Wikipedia

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

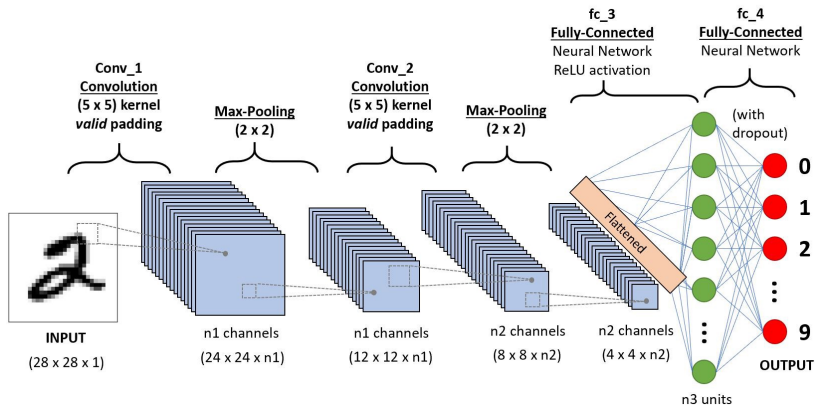
$$\lim_{p \rightarrow -\infty} \mathcal{LMorph}(f, w, p)(x) = \inf_{y \in W(x)} \{f(y) + w(x - y)\} = (f \oplus -w)(x)$$

$$\lim_{\alpha \rightarrow +\infty} \mathcal{SMorph}(f, w, \alpha)(x) = (f \oplus w)(x)$$

$$\lim_{\alpha \rightarrow -\infty} \mathcal{SMorph}(f, w, \alpha)(x) = (f \ominus -w)(x)$$

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

# Réseau de Neurones Convolutif (CNN)



Source : Towards Data Science



## Travail Effectué :

- Lire et comprendre en détail les articles recommandés, chercher d'autres sources
- Découvrir les bibliothèques scikit-learn, PyTorch, CUDA
- Coder et optimiser un algorithme de MM pour un élément structurant en niveaux de gris
- Commencer à prendre en main le code et essayer de l'installer sur *node9*

## Travail à venir :

- Prendre en main tout le code
  - Python (PyTorch) : 1.3k
  - C++ (CUDA, PyTorch) : 650
  - Rust : 1.7k
- Réparer tout le code
- Refaire les tests
- Contacter J. Angulo pour avoir son point de vue