

Quels logiciels pour l'apprentissage en anatomie ?

IABM 2023, Institut Curie

Jean Feydy

Équipe HeKA, Inria Paris, Inserm, Université Paris-Cité

Jeudi 30 mars 2023

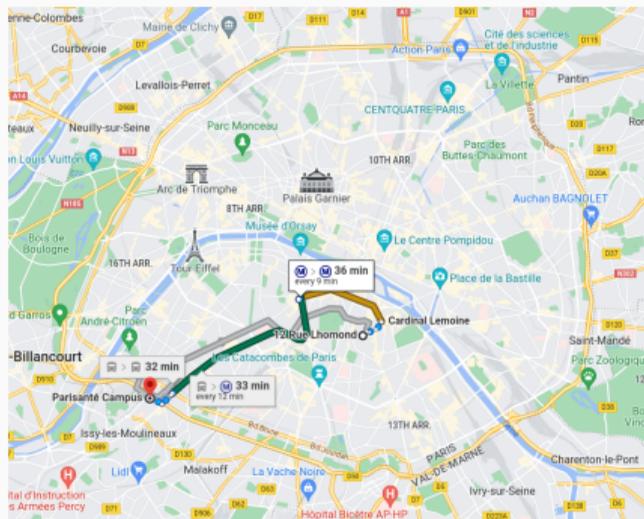
Hôpitaux

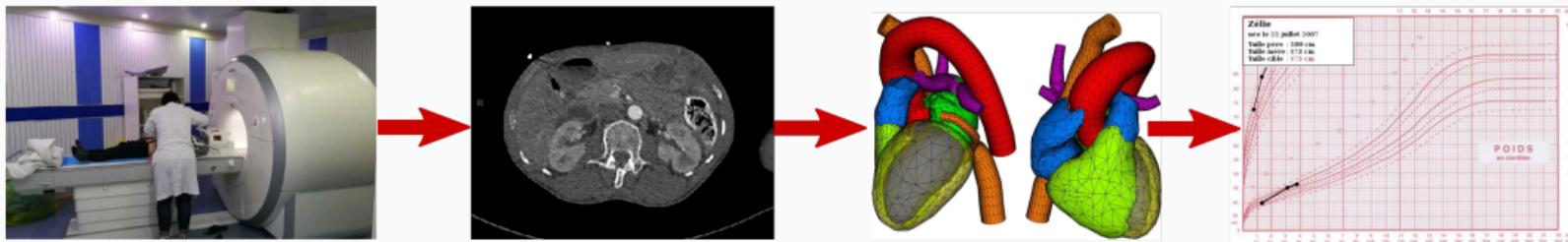
Inria Inserm

Universités



PariSanté Campus



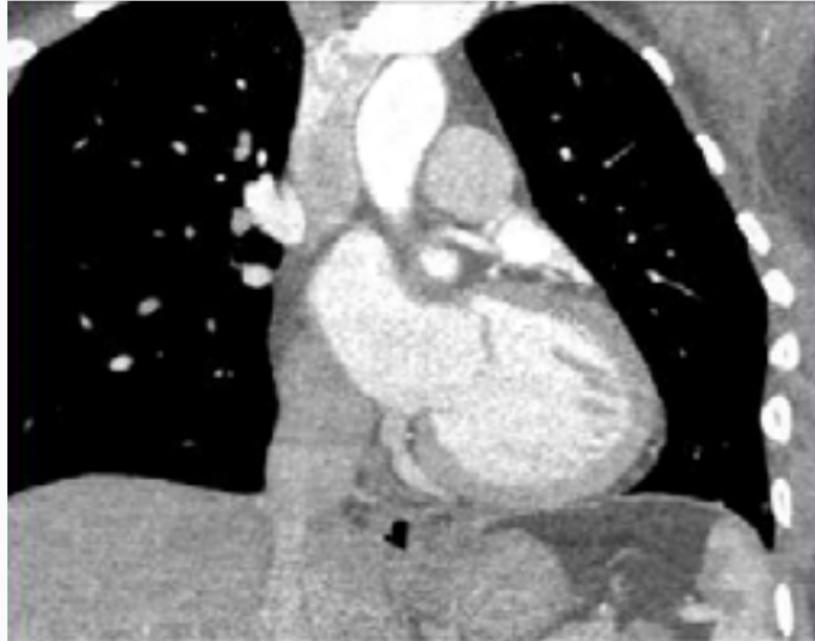


Comment passer d'un **tableau de nombres** à une représentation **intelligible** ?

1. Qu'est-ce qu'une **image** ?
2. Quels **logiciels pour l'IA** ?
3. **Complétons** notre boîte à outils.

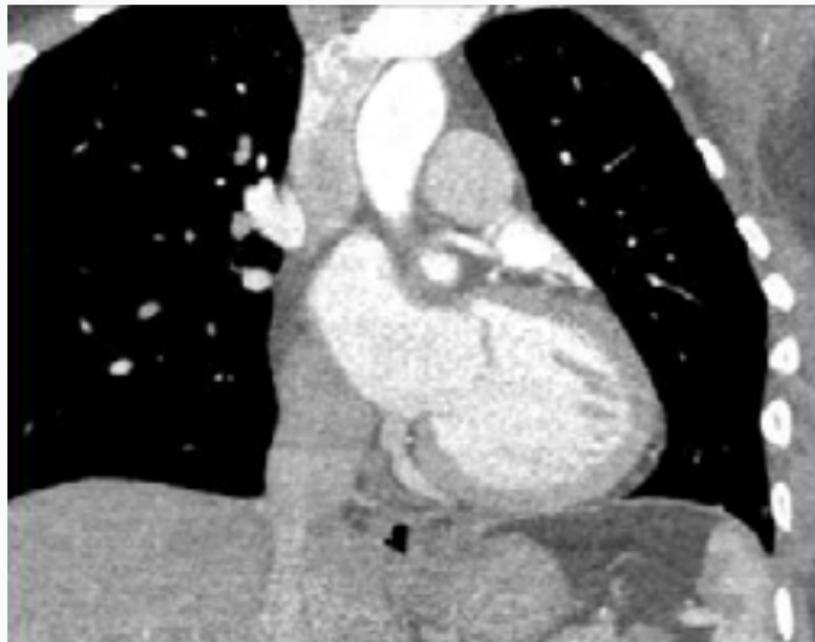
Qu'est-ce qu'une image ?

Une image : trois niveaux d'analyse [EPW11, Man11]



Une image : trois niveaux d'analyse [EPW11, Man11]

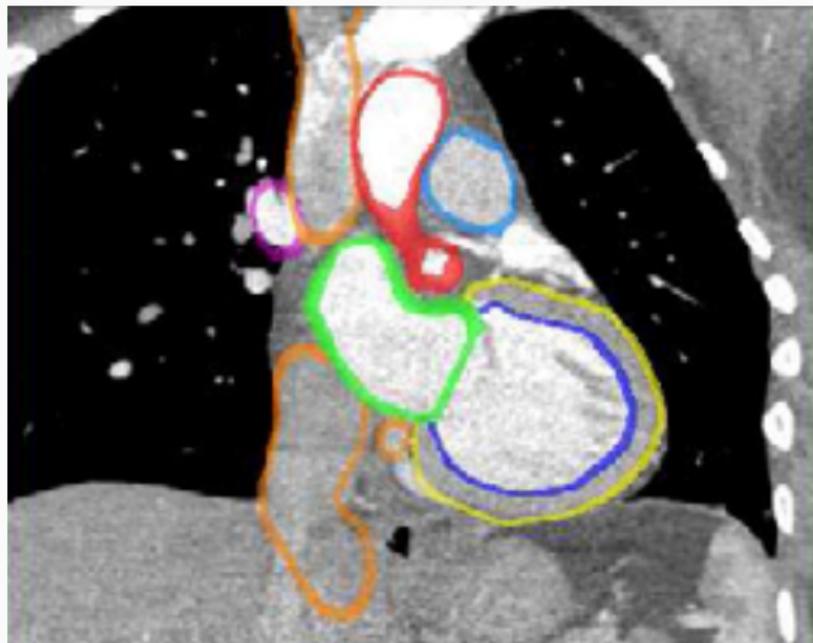
1. Pixels



Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

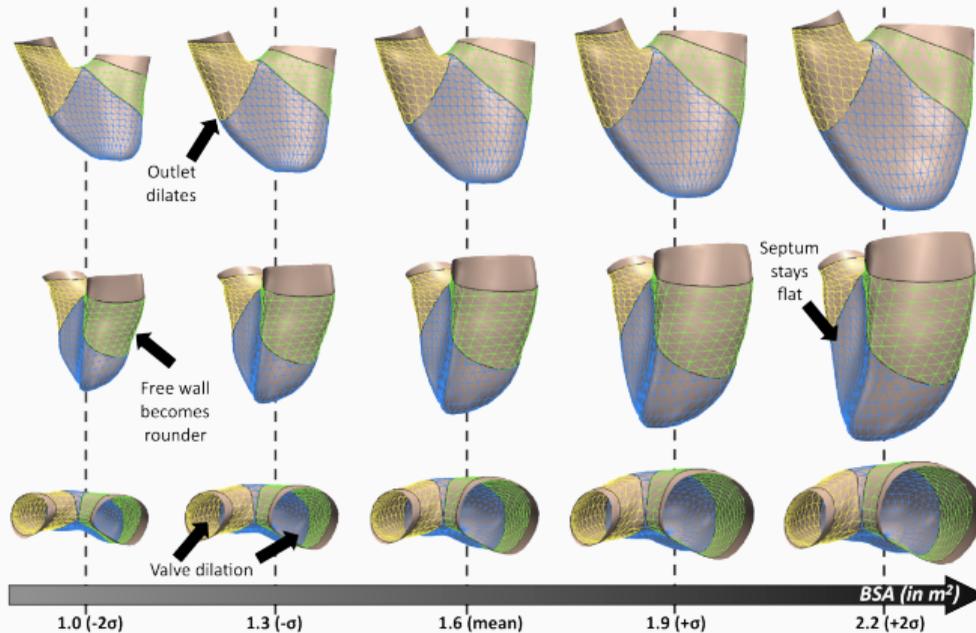
2. Anatomie



Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

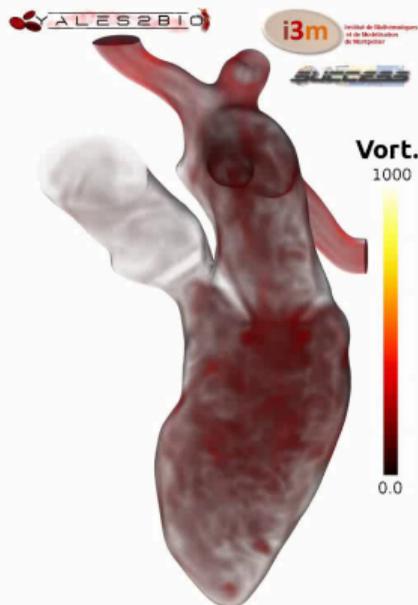


Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



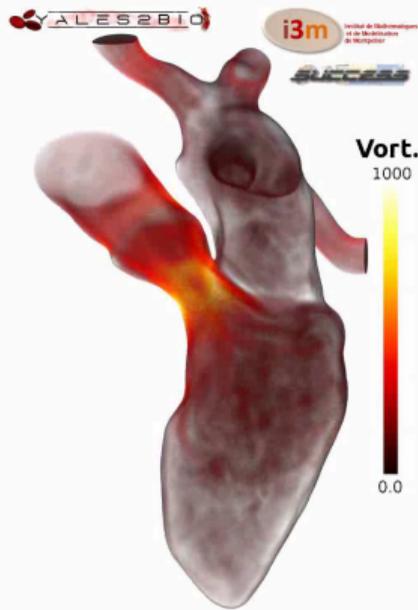
Time: 0 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



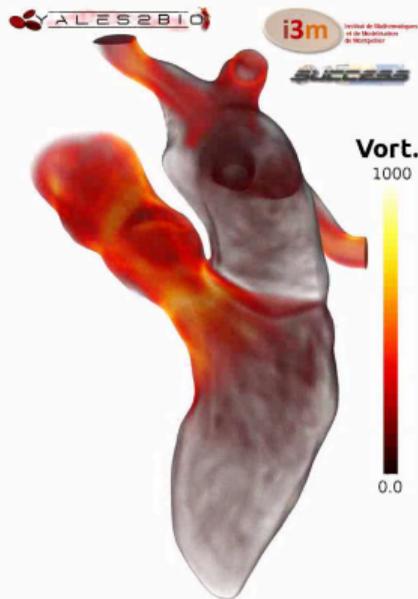
Time: 100 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



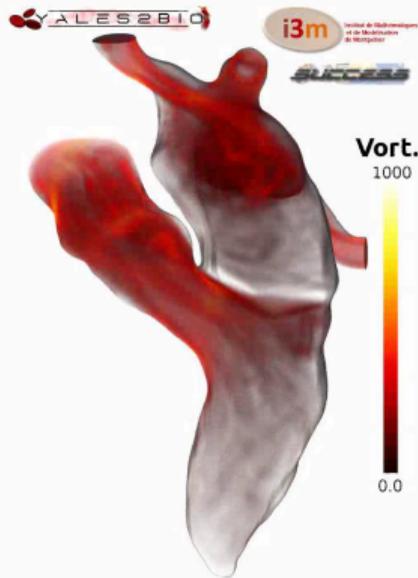
Time: 200 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



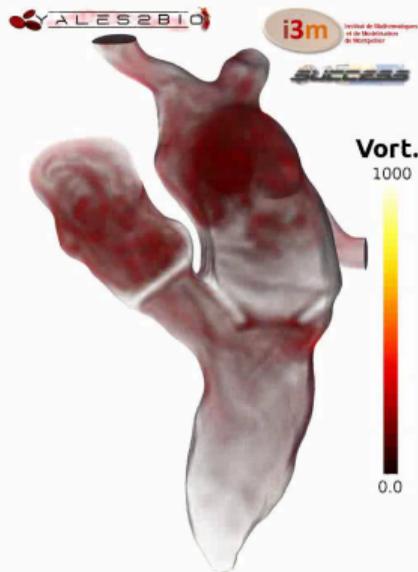
Time: 300 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



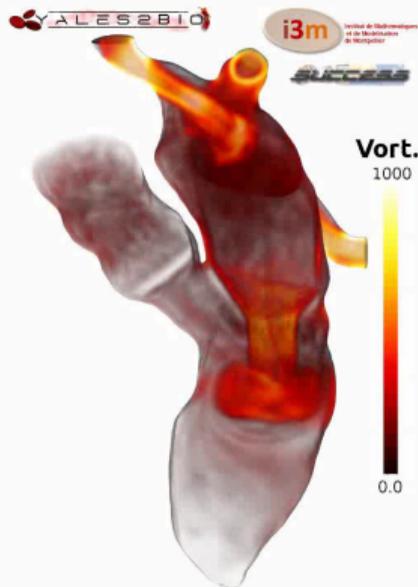
Time: 400 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



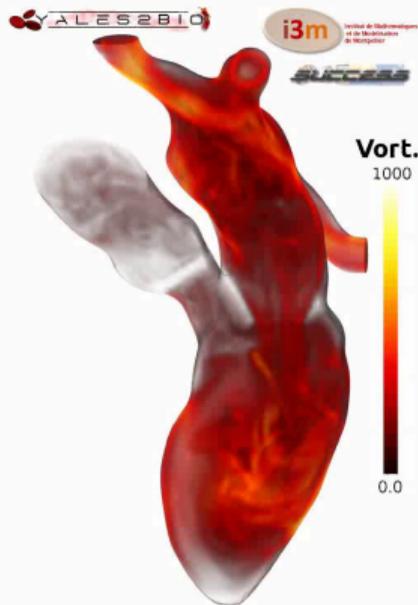
Time: 500 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



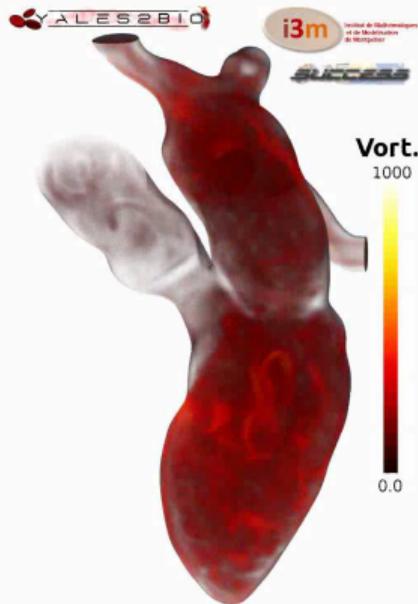
Time: 600 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



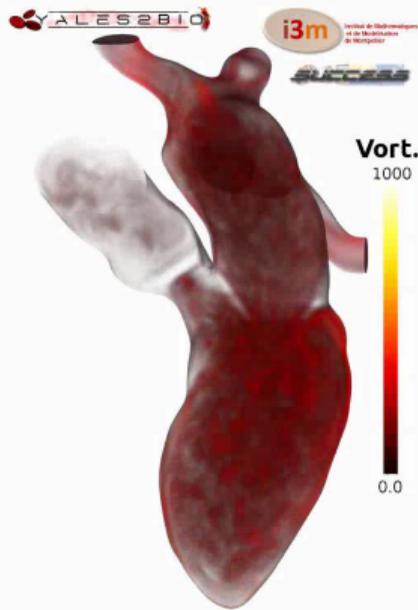
Time: 700 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



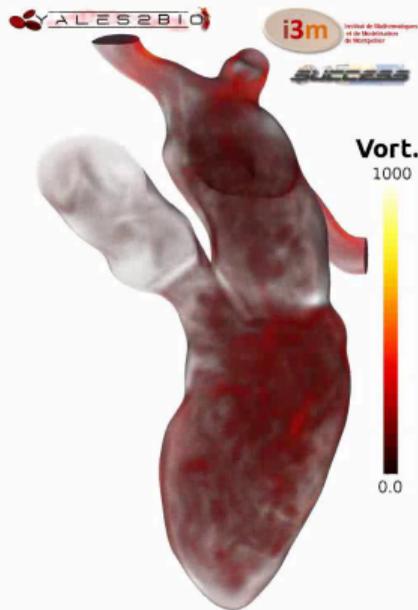
Time: 800 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



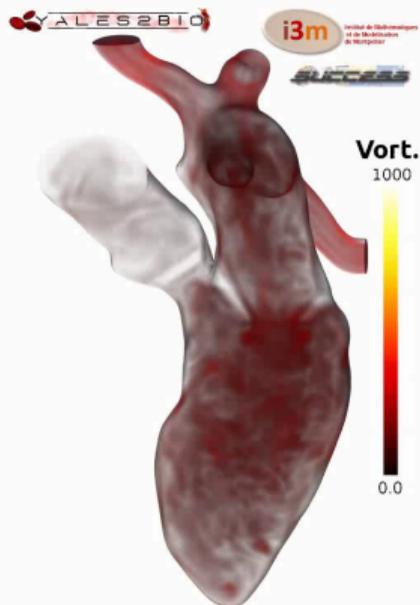
Time: 900 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



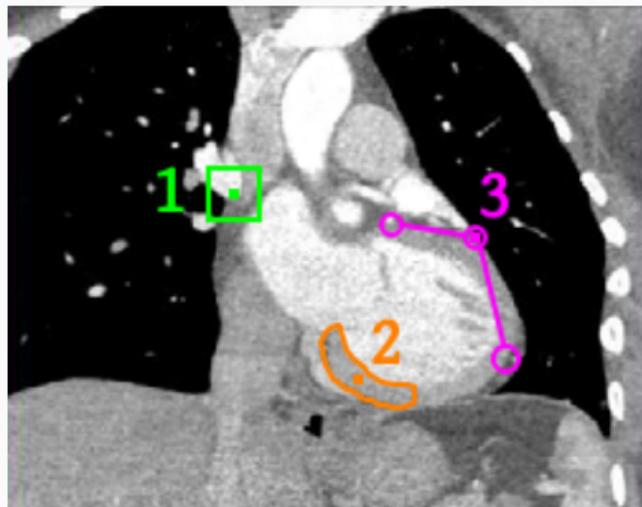
Time: 0 ms

Une image : trois niveaux d'analyse [EPW11, Man11]

1. Pixels

2. Anatomie

3. Fonction



En simplifiant beaucoup, chaque niveau de description correspond à une manière de **regrouper les pixels**.

1er niveau : grille de pixels

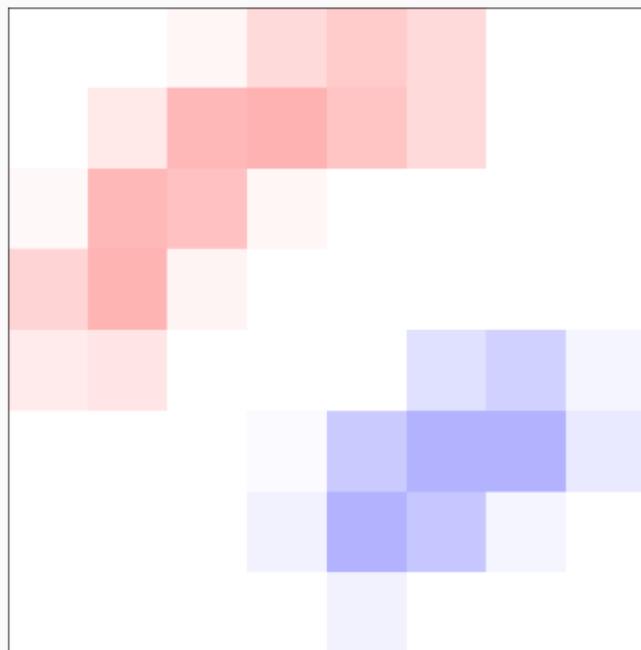


Tableau $N_x \times N_y \times N_z$ de pixels.

Images et volumes **bitmap** :

- .bmp, .png, .jpg
- Standard en **radiologie**.

- + Structure mémoire ordonnée.
- + Voisinages explicites.
- + **Convolutions** rapides.

- Analyse de **texture**.
- **Segmentation** des tissus.
- **Détection** de motifs.

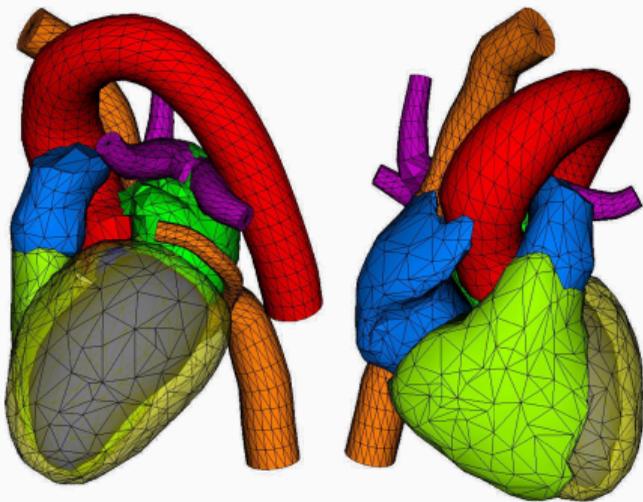
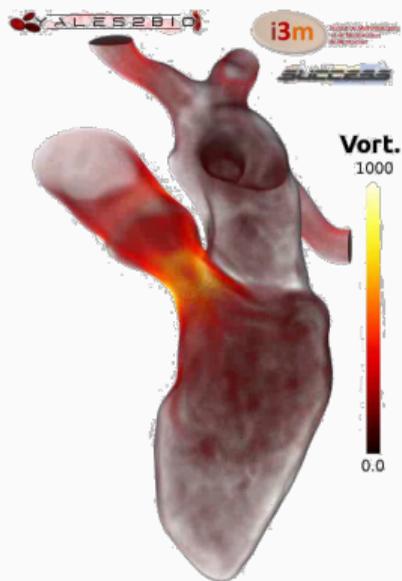


Tableau $N_{\text{points}} \times 3$
de coordonnées (x, y, z) .

Nuages de points (\pm triangles) :

- .svg
 - Standard pour le **jeu vidéo**.
- + Représentation compacte.
+ Géométrie de haute précision.
+ **Facile à déformer**.
- **Visualisation 3D**.
→ **Atlas** anatomiques.
→ Analyse des **formes**.



Mesh volumique,
graphe d'interactions.

Modèle physique, biologique :

- Éléments finis, réseaux.
 - Standard pour la **CAO**.
-
- + **Connaissance** a priori.
 - + **Robuste** au bruit.
 - + Comportement **réaliste**.
-
- Interprétation **physiologique**.
 - **Inférer** ce qui est invisible.
 - **Simuler** une intervention.

Forces et faiblesses de ces représentations

Comment chercher les **voisins** d'un point de l'espace ?

- Sur une **grille** : **immédiat**.
- Avec N **points** (x, y, z) : **calcul** de N distances, ou **construction** d'un arbre.

Comment faire **pivoter** un organe de 10° ?

- Sur une **grille** : **artefacts**, perte de détails, transferts de mémoire.
- Avec N **points** (x, y, z) : arithmétique **simple** sur les coordonnées.

Efficacité numérique \iff Entraînement sur de **grandes bases de données**.

En imagerie médicale, on peut travailler avec:

- Une **grille de pixels** 2D ou 3D.
- Un **tableau de coordonnées** (x, y, z) .
- Un **réseau d'interactions** complexes.
- Les trois à la fois!

Dans tous les cas, il s'agit de définir une grande **formule structurée** :

$$\text{image} \xrightarrow{F} F(\text{image}) \simeq \text{diagnostic}$$

On parle d'**architecture** de calcul paramétrique
 \simeq **modèle** à fitter \simeq **réseau** à entraîner.

Quels outils logiciels pour l'IA ?

Le matériel est au cœur de la “révolution de l’IA”

L’**imagerie** et l’**apprentissage** automatique sont étudiés depuis des **décennies**.

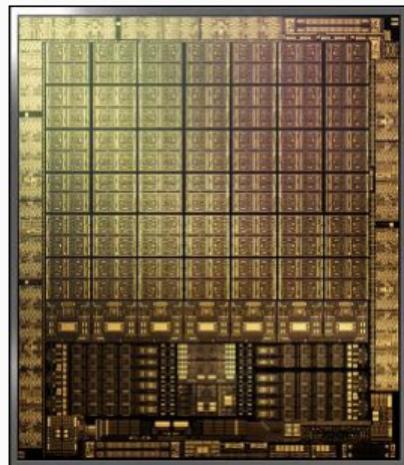
Percée en 2010-15 : utiliser des **PlayStations** pour faire de la **science** est devenu **facile**.

Effort de recherche à tous les niveaux vers:

- Un **matériel** toujours plus puissant.
- Des **bibliothèques** toujours plus souples.
- Des **modèles** toujours plus pertinents.

Résultats **spectaculaires** obtenus dans quelques domaines

⇒ grands **investissements** publics et privés.



7,000 cœurs sur un GPU.

Pour aller au-delà des prototypes, une suite logicielle complète est nécessaire

Graphisme : Imprimante + Driver + **Photoshop** \Rightarrow Dessins

Données tabulaires : GPU + **cuBLAS** + **PyTorch** / TensorFlow \Rightarrow Réseaux de neurones **“classiques”**

Grilles de pixels : GPU + **cuDNN** + **PyTorch** / TensorFlow \Rightarrow Réseaux de neurones **convolutifs**

Nuages de points et graphes : GPU + **CUDA** + **??** \Rightarrow Réseaux de neurones **géométriques**

Complétons notre boîte à outils

Comment développe-t-on un logiciel d'imagerie médicale ?

2000–2010 : bases de code **“monolithiques”** en C++

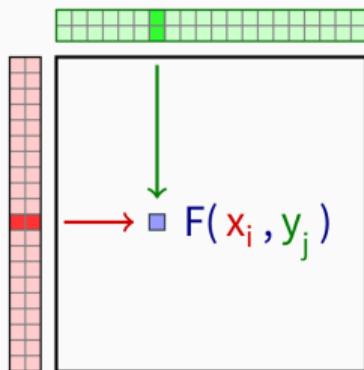
- Le C++ est incontournable pour les performances en 3D.
- **Difficile à maintenir** et à faire **évoluer**.

2010–2020 : développement **modulaire** en Python/C++

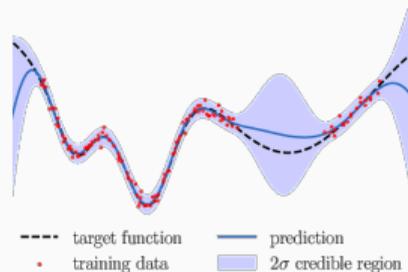
- **Appeler une bibliothèque C++** depuis Python est devenu facile.
- Alliance de la souplesse du Python et des performances du C++.
- Échanges entre les communautés, **facilité d'adoption** par les nouveaux venus.

2020–2030 : **dépasser les limites des bibliothèques** C++ existantes

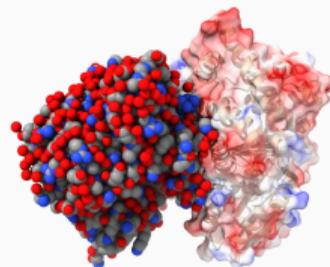
- **Générer du code C++** pour GPU et le compiler à la volée est devenu facile.
- Maintenir une extension de PyTorch “optimisée pour ...” devient raisonnable.



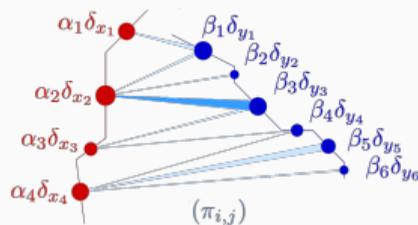
Matrices symboliques :
distances, noyaux,
transformées discrètes,
point convolutions,
attention...



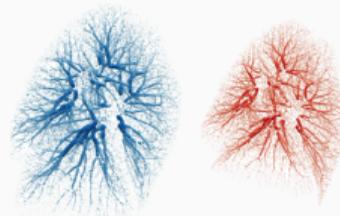
Processus gaussiens.



Docking de protéines.



Transport optimal.

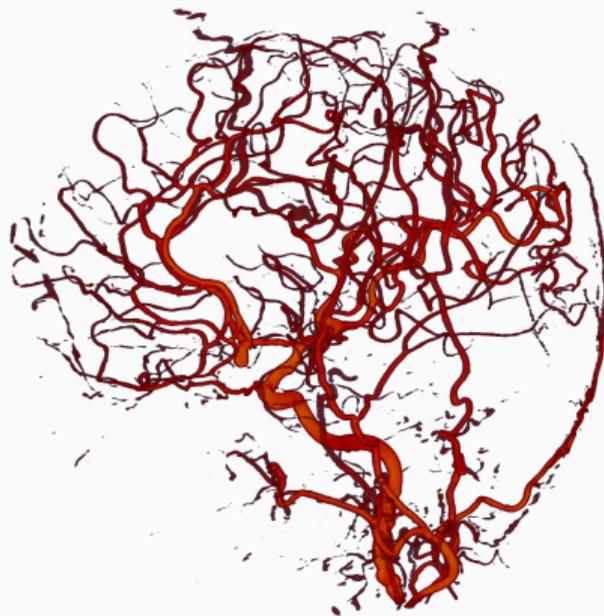


Suivi de la respiration.

Des interfaces haut niveau pour passer du prototype à la production

Un écosystème en construction :

- **PyG** : depuis 2018
 - **Deep learning sur les graphes.**
 - Basé sur torch-scatter, etc.
- **PyVista** : depuis 2019
 - **Visualisation 3D.**
 - Basé sur VTK.
- **scikit-shapes** : en cours
 - **Analyse de formes.**
 - Basé sur KeOps.



Réseau artériel cérébral.
Comment apprendre de cet objet ?

Conclusion

- Les **cartes graphiques** (GPUs) sont le moteur de l'IA.
- Un **environnement logiciel complet** est nécessaire, du C++ au Python.
- Depuis 2015, l'**imagerie médicale** a pu surfer sur une vague d'investissement des **GAF**A pour le traitement d'**images naturelles**.

Avancées : **segmentation**, traitement de la **texture** et **détection** de lésions.

- Quid de la planification **chirurgicale**, la **morphométrie**, l'analyse **vasculaire**... ?

Un **investissement dans les fondations numériques** du domaine est en cours.

References

-  Olivier Ecabert, Jochen Peters, and Matthew Walker.
Segmentation of the heart and great vessels in ct images using a model-based adaptation framework.
Medical Image Analysis, (15):863–876, 2011.
-  Tomaso Mansi.
A statistical model for quantification and prediction of cardiac remodelling: Application to tetralogy of fallot.
IEEE transactions on medical imaging, 2011.