

Imagerie racinaire

Traitement et segmentation automatique
d'images de racines



CONTEXTE & PROBLÉMATIQUES

Problématiques

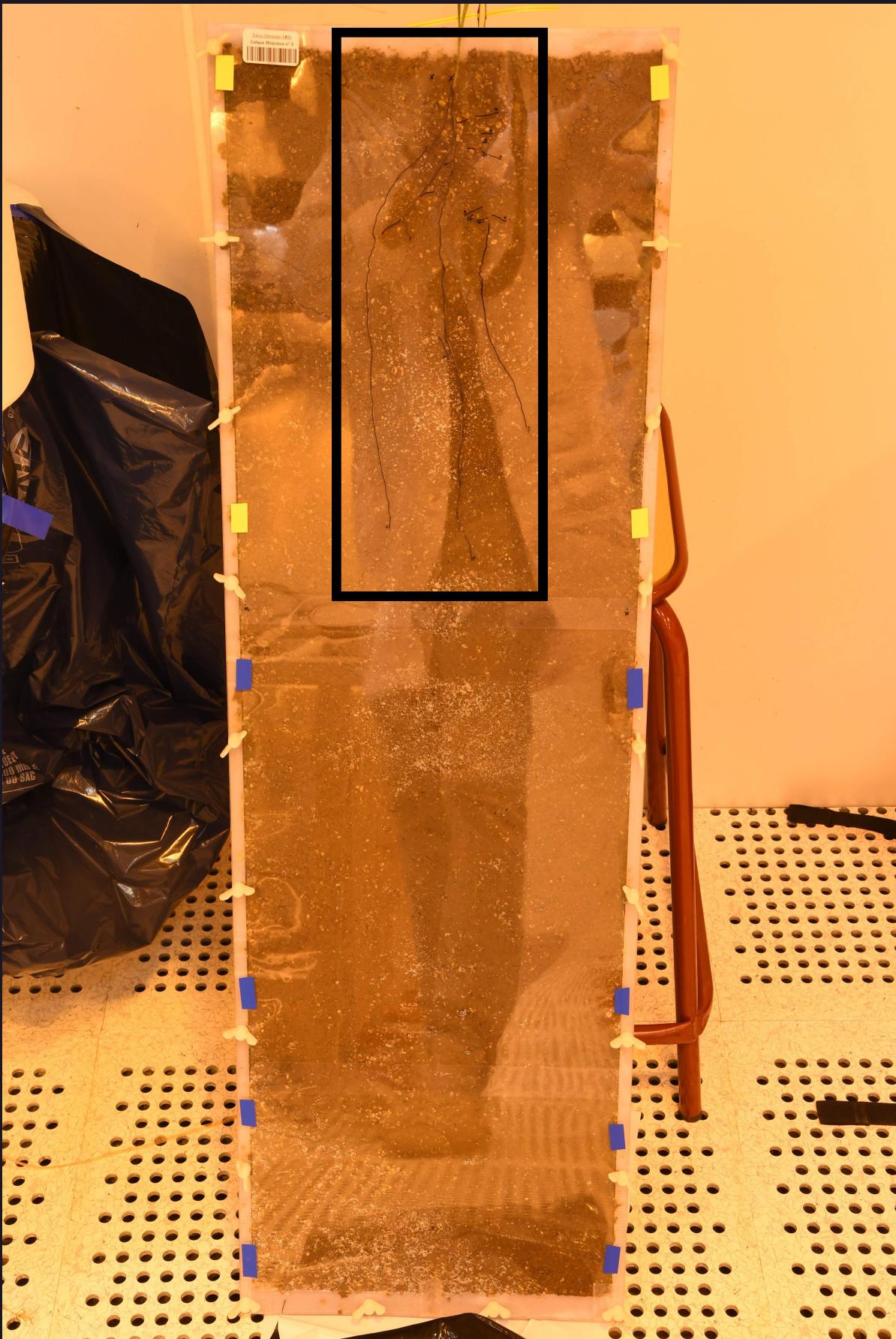


Photo d'un rhizobox à un instant t

Comment extraire la longueur des racines primaires
à partir de l'image ?

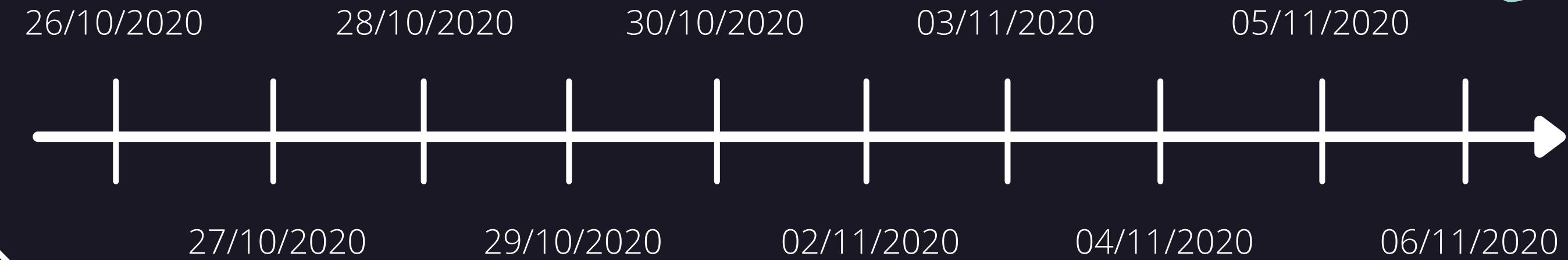
A partir d'un jeu de données d'images brutes peut-on automatiquement calculer
la longueur des racines ?

Le protocole expérimental doit-il être modifié pour pouvoir traiter ces images ?

Jeu de données



- 54 rhizoboxs
- Photos prises tous les jours pendant 10 jours pour observer l'évolution des racines



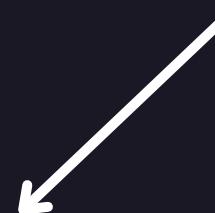
593 photos au total

Etiquette permettant d'identifier le rhizobox

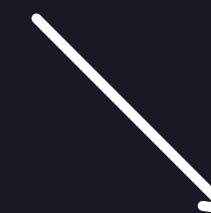
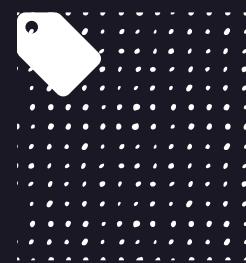
Racine primaire

Zoom sur un rhizobox

COMMENT EXTRAIRE DE L'INFORMATION DE NOS IMAGES ?



A partir de l'étiquette pour structurer
le jeu de données



A partir des racines pour en calculer la
longueur

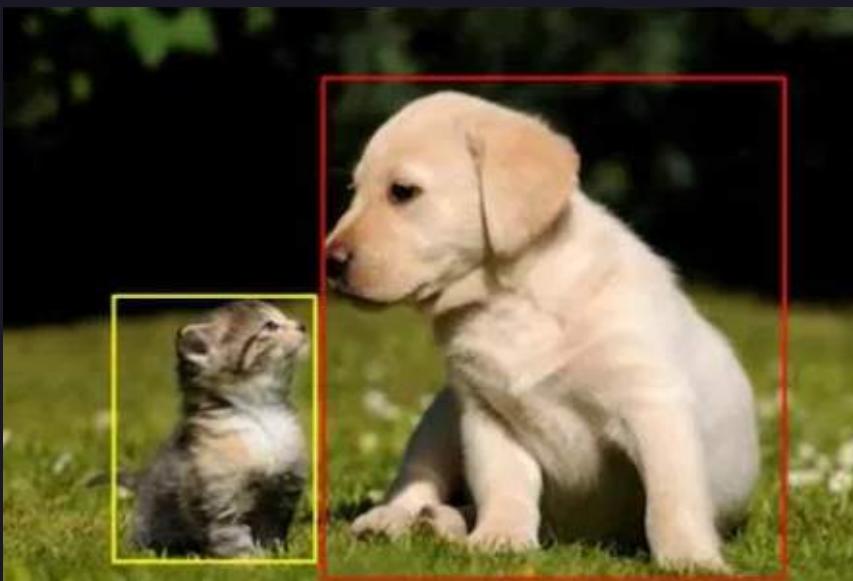


Détection VS Segmentation

Détection :

Une boîte de délimitation (bounding box) ou une région environnante est déterminée de la manière la plus étroite possible afin de localiser la position exacte de l'objet dans l'image.

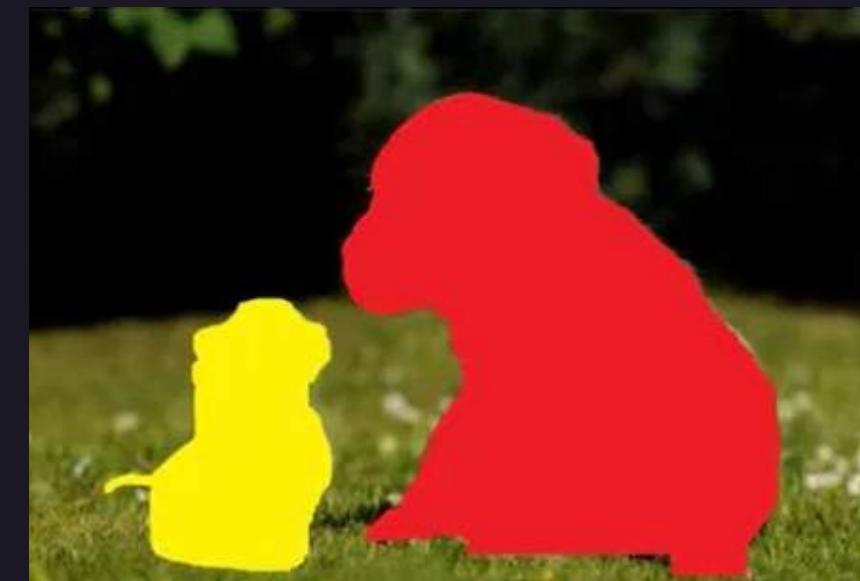
- Moins précis pour trouver les racines
- Utilisée dans ce projet pour structurer le jeu de données (étiquette)
- Exemple de modèle : YOLO



Segmentation Sémantique :

Processus consistant à associer chaque pixel d'une image donnée à une étiquette de classe particulière.

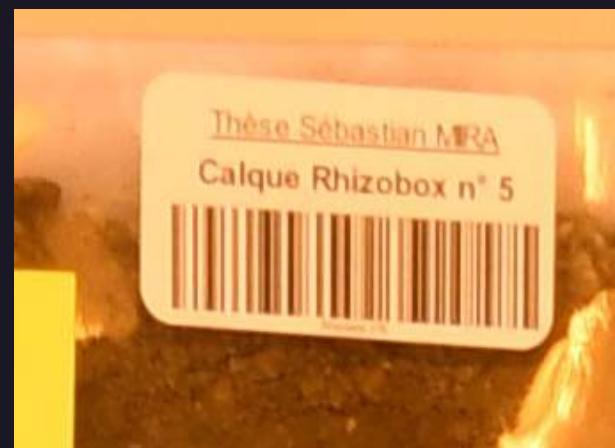
- Identification de masques qui permettent au pixel près d'identifier les racines.
- Permet éventuellement d'aller plus loin que le calcul de la longueur en observant l'architecture du réseau racinaire
- Exemple de modèle : U-net





CONSTITUTION D'UNE BANQUE DE DONNÉES

Constitution d'une banque de données - Individualisation des rhizobox par Deep Learning



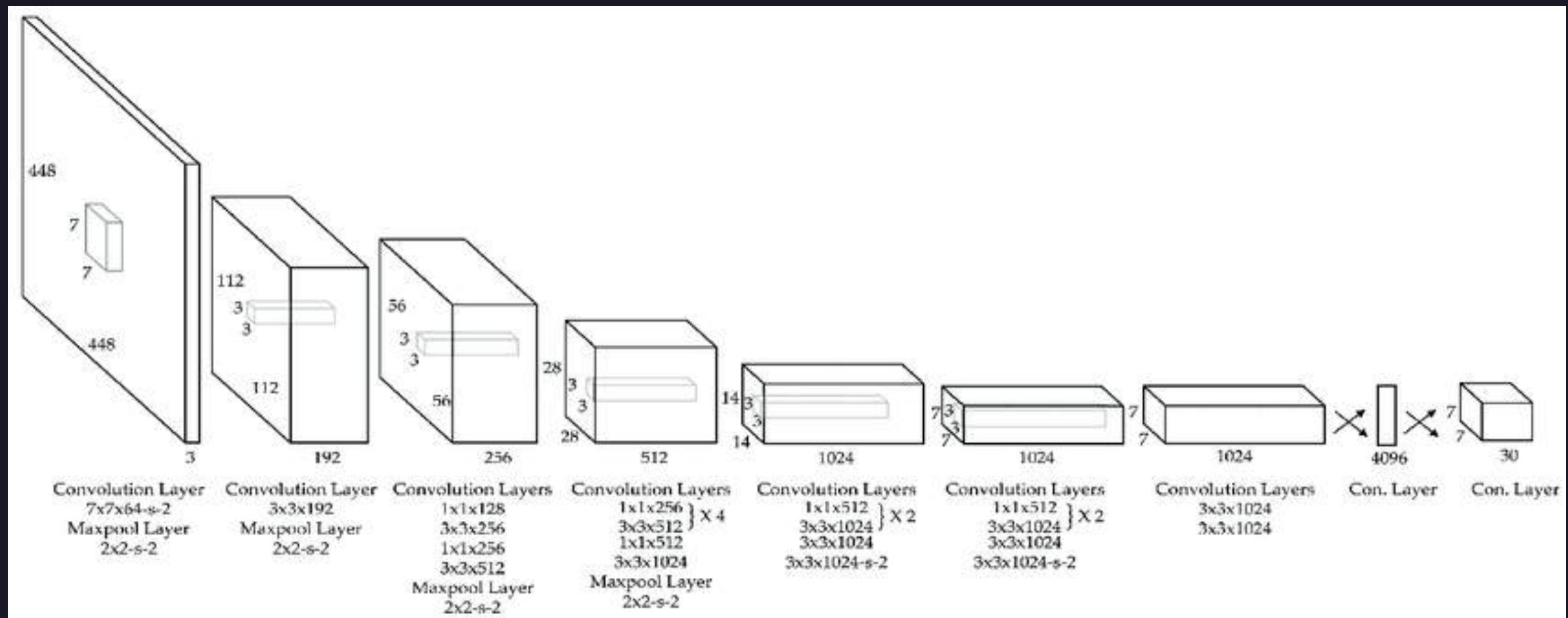
Comment lire le numéro du
rhizobox ?

Méthode 1 : recherche et lecture du
numéro (OCR) en parcourant l'image
entière

Méthode 2 : détection et zoom sur
l'étiquette (YOLO), puis lecture (OCR)

Constitution d'une banque de données - Tri

Le réseau de neurones : You Look Only Once (YOLO)



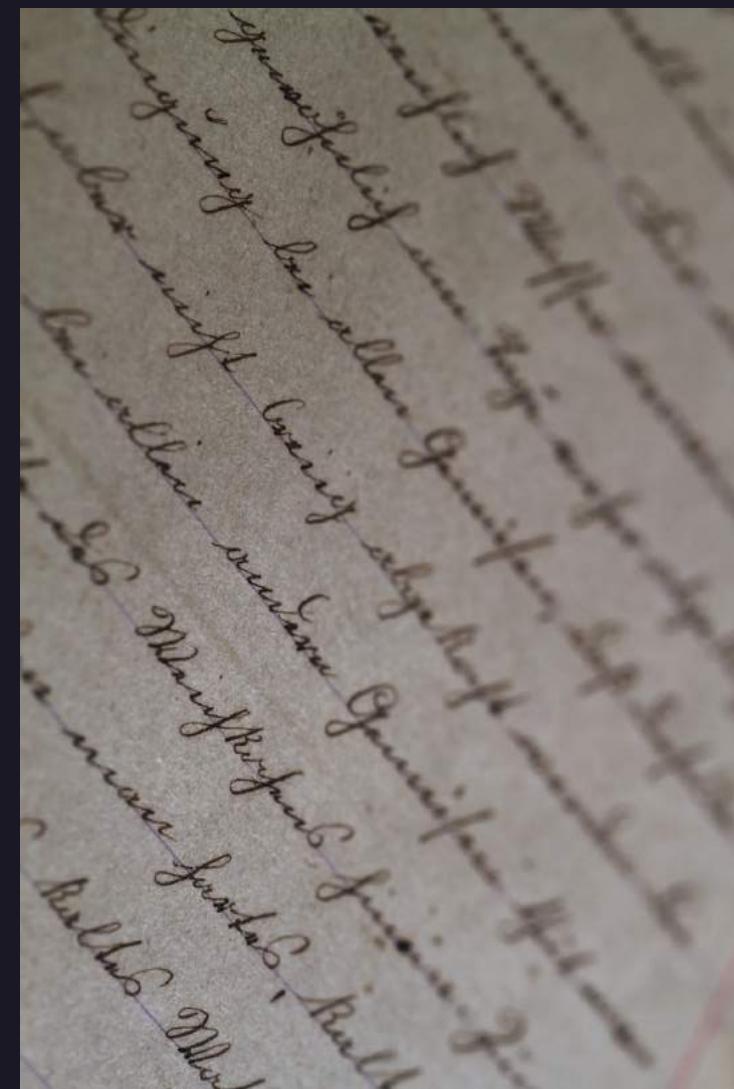
Architecture du réseau de neurones

- > Détection d'objets en deux étapes
 - > Utilisation d'un modèle pré-entraîné à la détection de zones de texte non-structurées

Redmon, J., Divvala, S.K., Girshick, R.B., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

Constitution d'une banque de données - Tri

Optical Character Recognition (OCR) avec Tesseract



- Outil open-source permettant de convertir une image contenant du texte en format texte
- Plusieurs sous-process, dont une étape de reconnaissance de caractères (Deep Learning)
- Utilisation des modèles pré-entraînés de Tesseract : entraînement avec 4 millions lignes de texte (manuscrit et imprimé) avec 4500 polices différentes, dans 10 langues différentes.

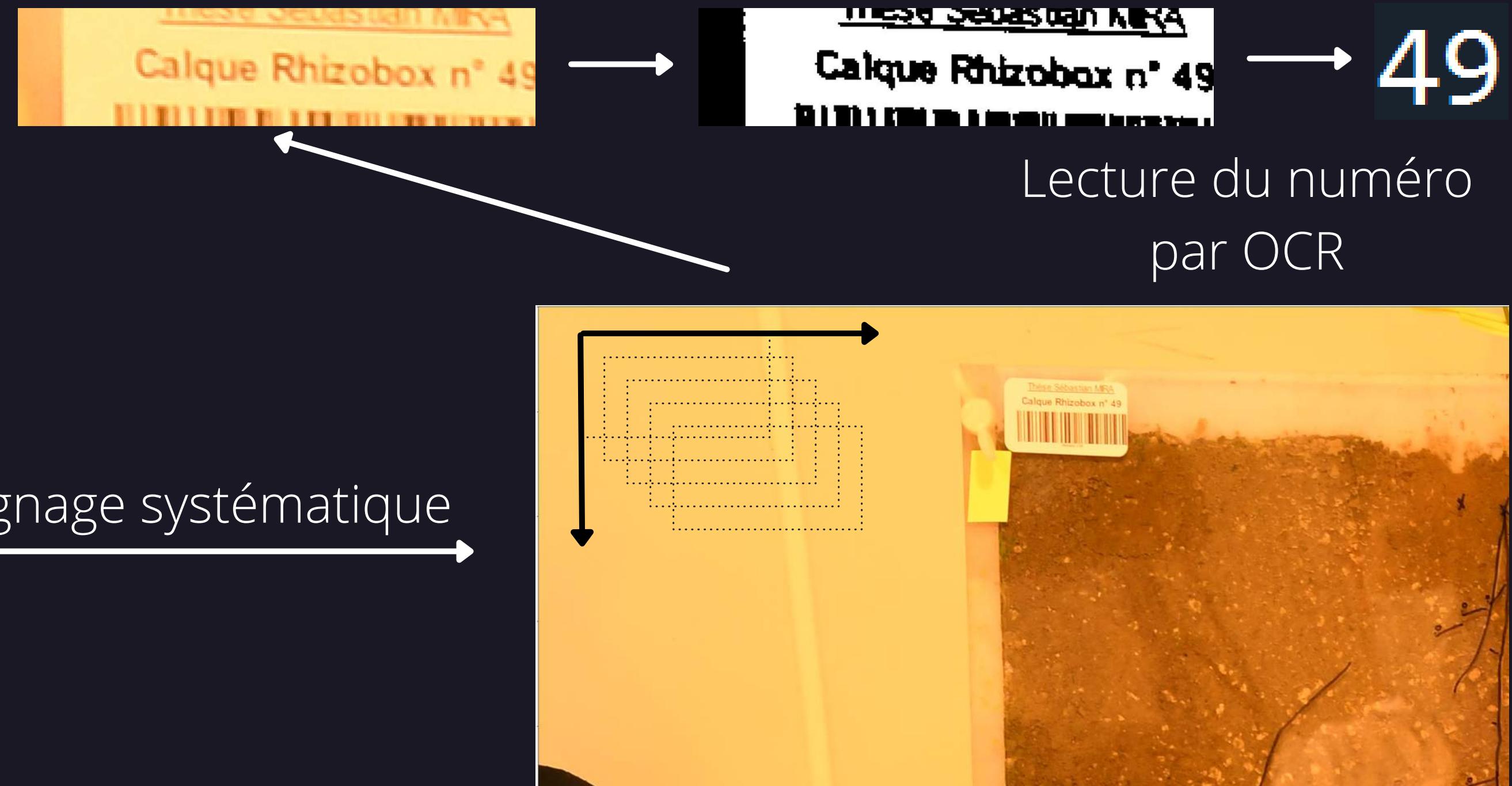
Constitution d'une banque de données - Tri

Méthode 1

Recherche du numéro
en parcourant l'image
entière



Image initiale



Cadre parcourant l'image

Constitution d'une banque de données - Tri

Méthode 2

Détection des zones de texte et lecture du numéro



Image initiale

Rognage systématique



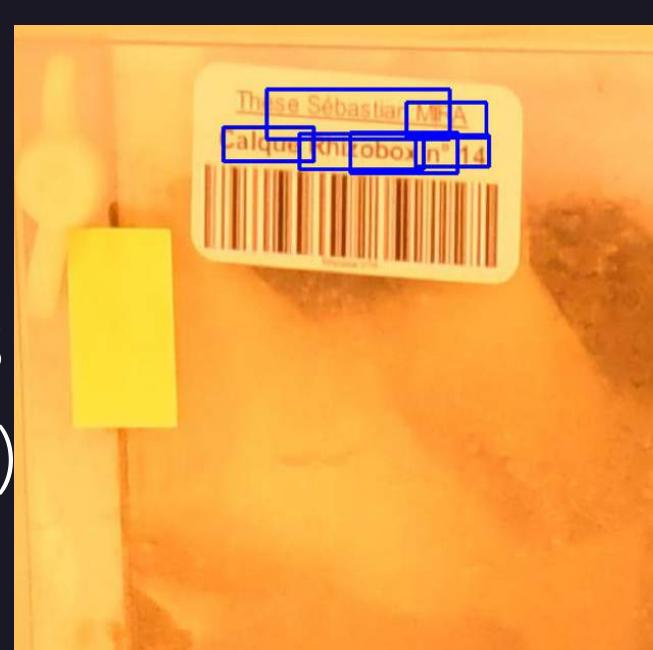
Image rognée



Zoom sur
l'étiquette (YOLO)



Bounding boxes
de texte (YOLO)



Constitution d'une banque de données - Tri

Méthode 2



Bounding boxes de texte

Binarisation en noir et blanc

Lecture du numéro
par OCR

Constitution d'une banque de données - Tri

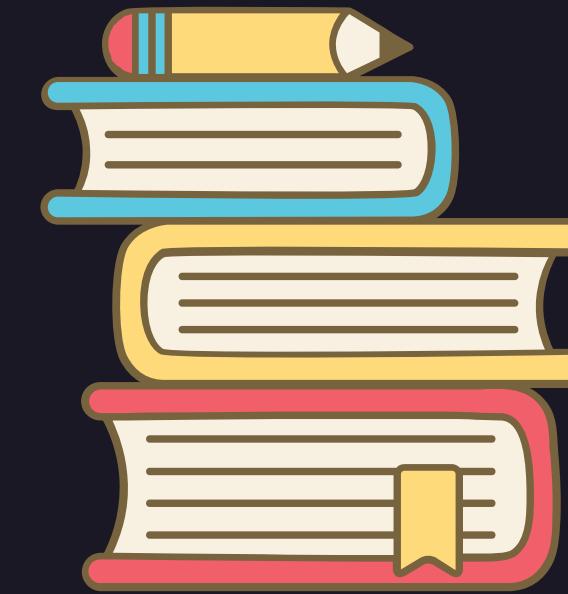
Comparaison des méthodes 1 et 2 pour la lecture du numéro

	Méthode 1	Méthode 2
Accuracy	0.11	0.04
Temps d'exécution moyen (sec)	103	4

Qualité des photos ?



SEGMENTATION ET LONGUEUR DES RACINES



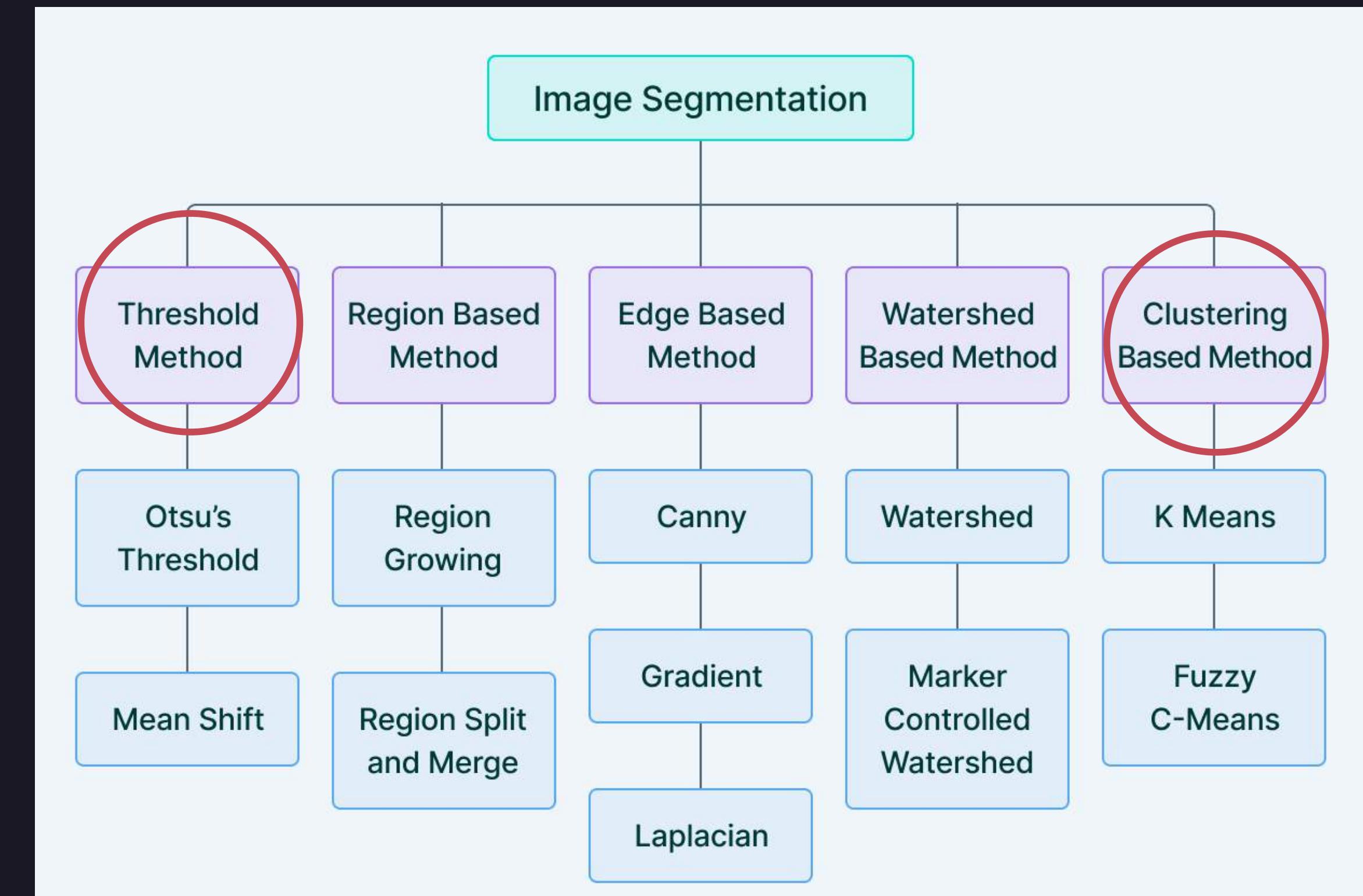
MODÈLES ET MÉTHODES

Méthodes traditionnelles de segmentation

Segmenter

Regrouper des régions ou des segments similaires d'une image sous leurs étiquettes de classe respectives.

La segmentation d'image est une extension de la classification d'image avec une étape supplémentaire de localisation permettant de tracer le contour des objets.



Thresholding et K-means clustering

threshold_local

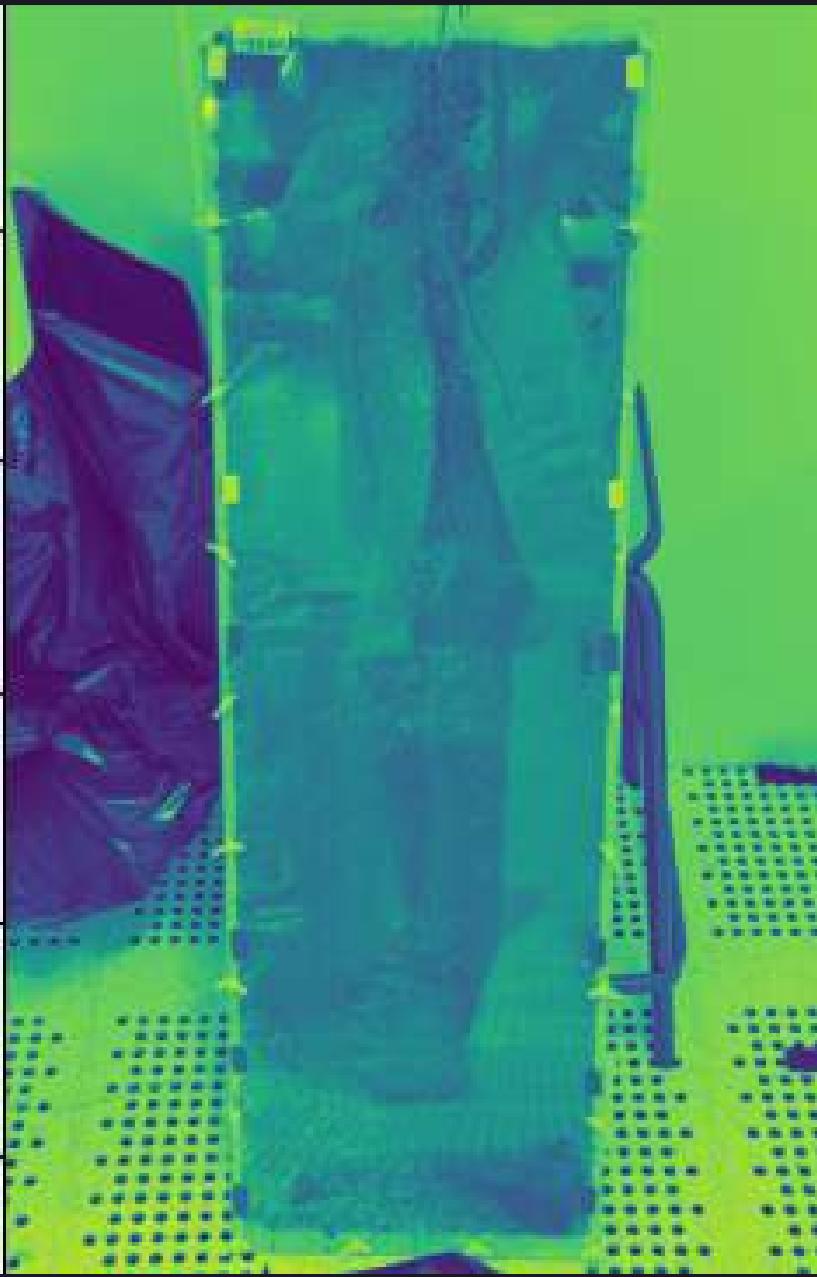


Image (masque de seuil) basée sur le voisinage local des pixels.

La valeur du seuil est la moyenne pondérée du voisinage local d'un pixel soustraite par une constante.

cv2.kmeans



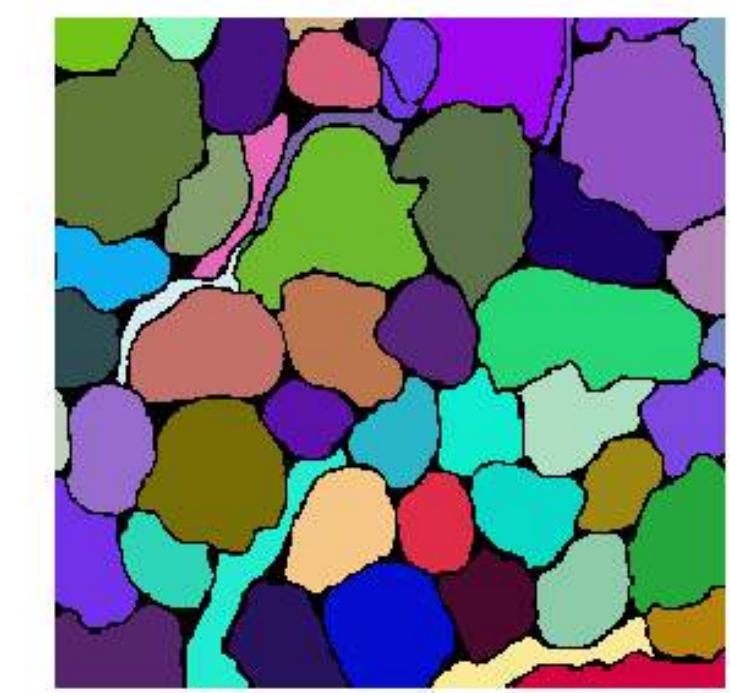
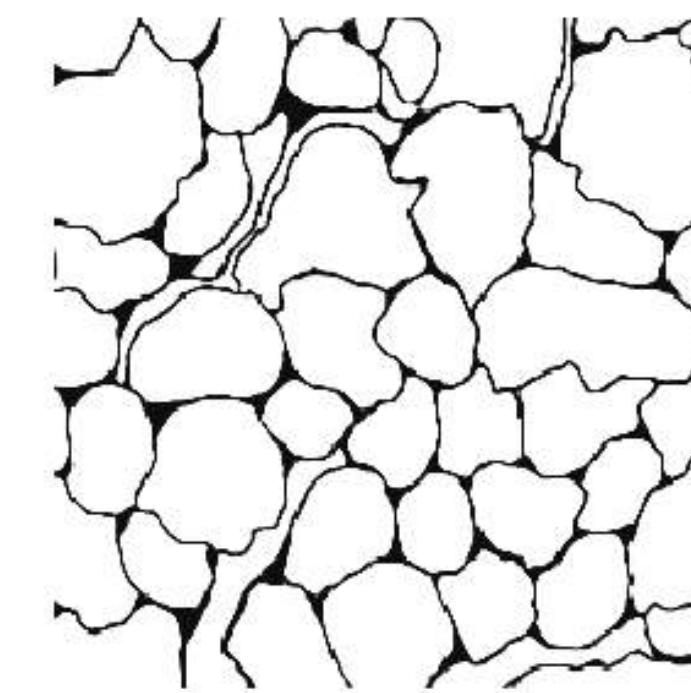
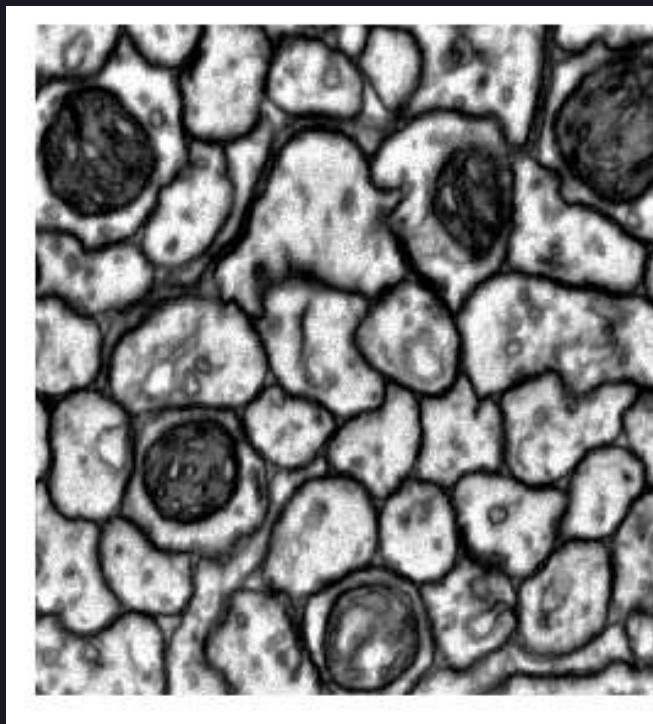
Nombre de classes : 4
Nombre d'itérations: 10

Segmentation sémantique avec U-net

U-net : Un des plus courants et des plus simples d'accès (code, compréhension)

Utilisé pour des images biomédicales
Un des plus précis

Utilisé sur des jeux de données avec un faible nombre d'images (environ une cinquantaine)



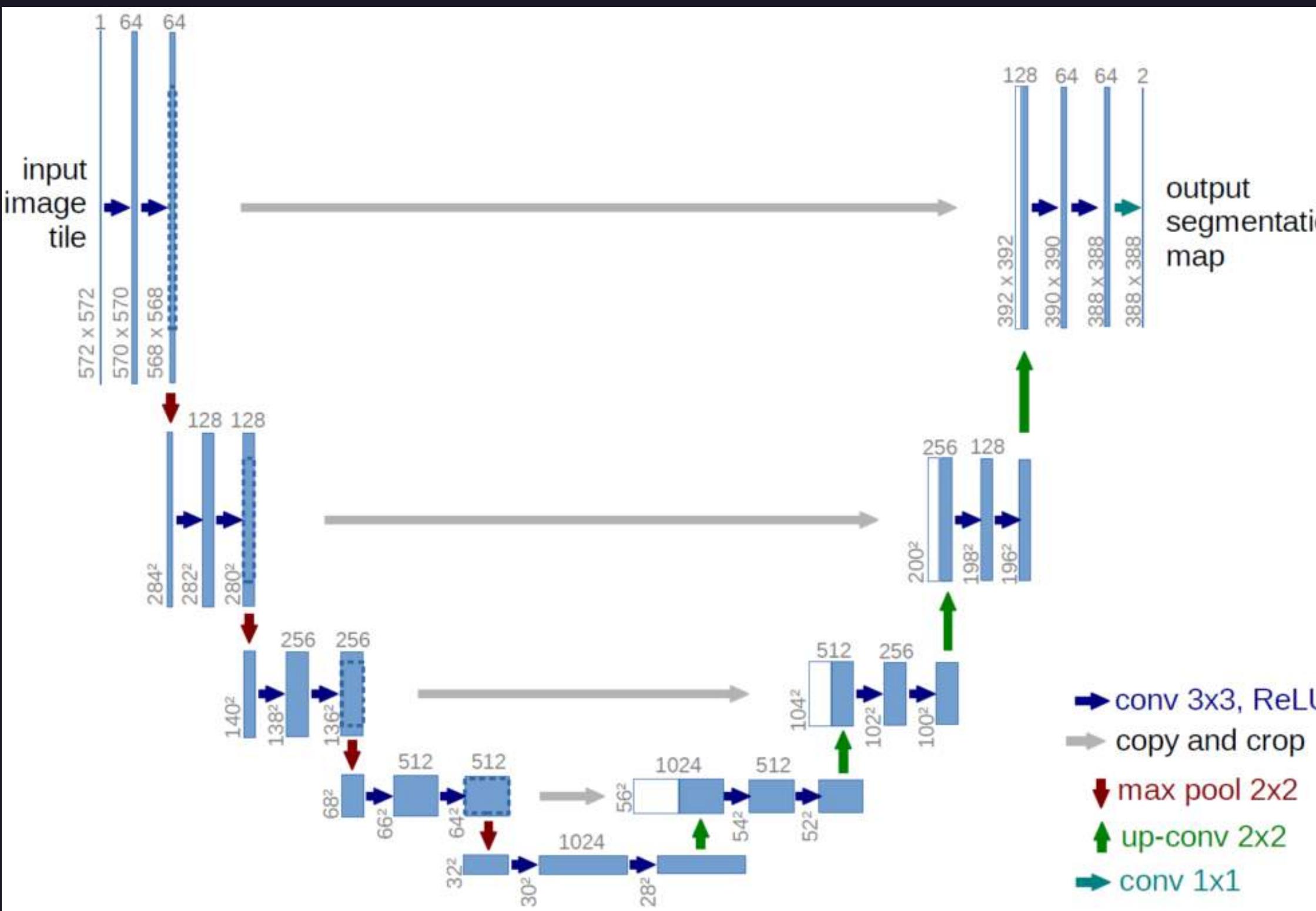
Alom et al. Nuclei Segmentation with Recurrent Residual Convolutional Neural Networks based U-Net (R2U-Net). 2018

Alom et al. Recurrent residual U-Net for medical image segmentation, mars 2019

Ronneberger et al. U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015

Falk et al. U-Net: deep learning for cell counting, detection, and morphometry, janv 2019

Segmentation avec 2D U-net - architecture



2 blocs :

- **Encoder (DownBlock)** : identifier les features
- **Décoder (UpBlock)** : localiser les features

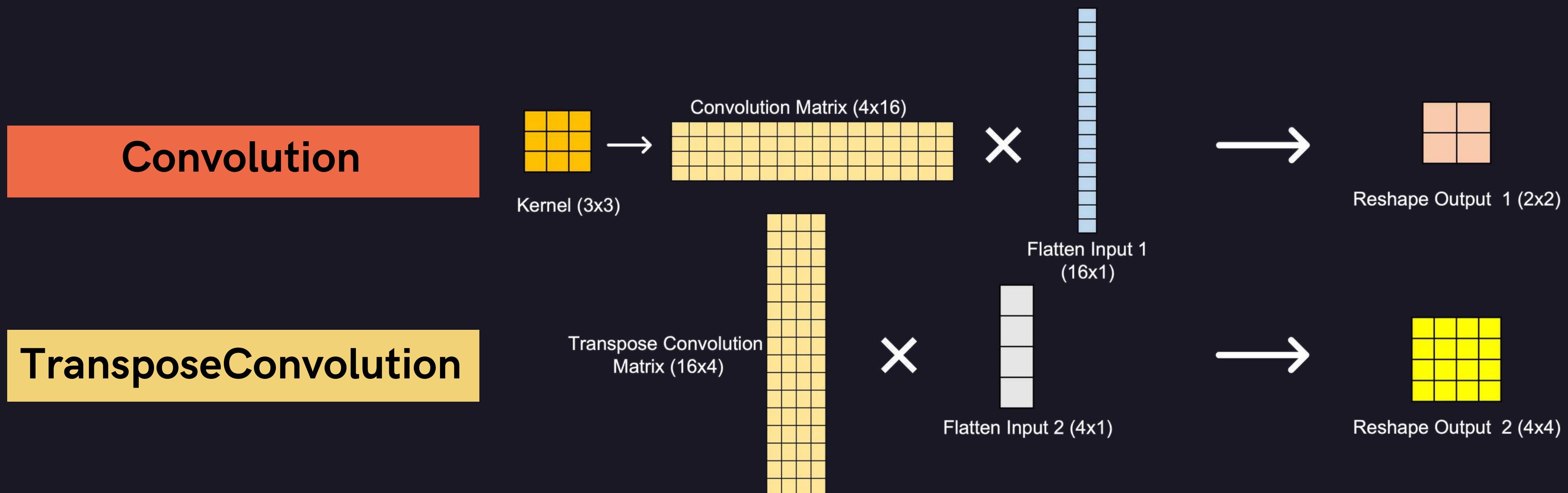
Couches de convolution

Couches de Maxpooling

Upsampling ou TransposeConvolution

Skip connections = transférer l'information apprise vers le block qui localise les features

2D U-net - les couches de convolution



Segmentation avec 3D U-net

Traitement d'images en couleurs - réseau utilisé pour la segmentation de racines

(Smith et al. « *Segmentation of roots in soil with U-Net* ». *Plant Methods* 16, no 1 (8 février 2020): 13.)

Différences avec 2D U-net :

Batch normalization
stabilité et rapidité

Couche SeparableConv2D
au lieu de Conv2D la division du kernel
(filtre) en deux permet de gérer des photos
couleurs avec 3 canaux (RGB)

3D U-net : Separable convolution et batch normalization

Separable Convolution

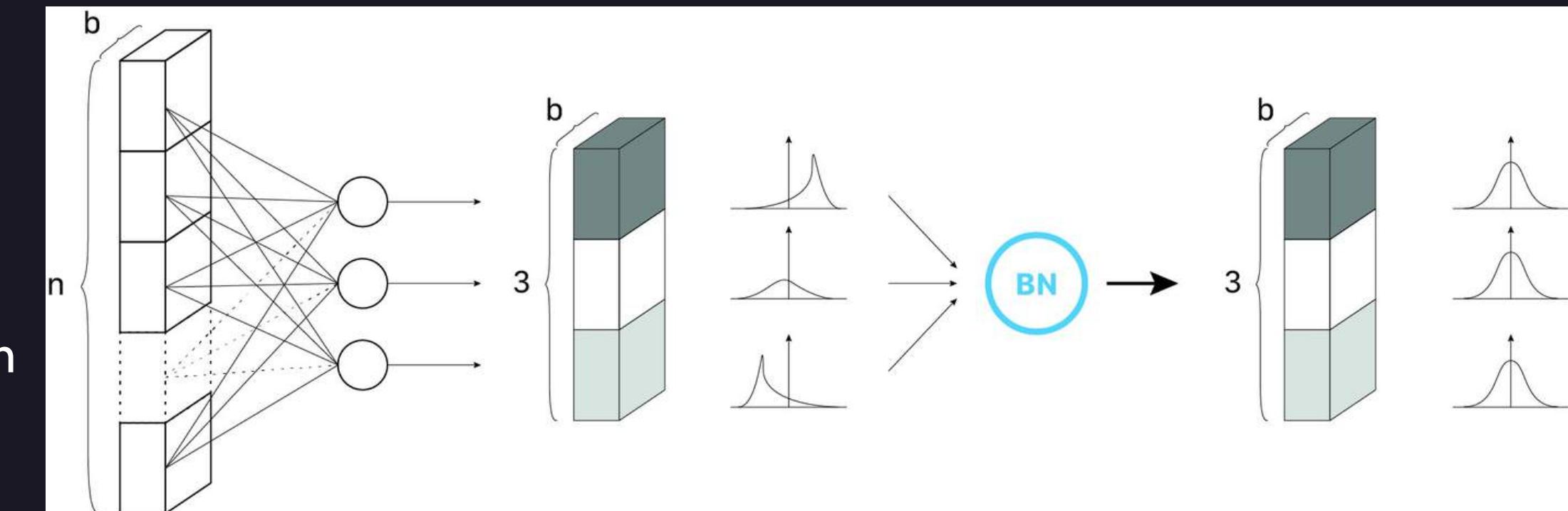


BatchNorm

Taille du Batch = hyperparamètre qui définit le nombre d'échantillons à traiter avant de mettre à jour les paramètres du modèle interne.

Normalisation des vecteurs d'activation des couches cachées en utilisant la moyenne et la variance du batch courant.

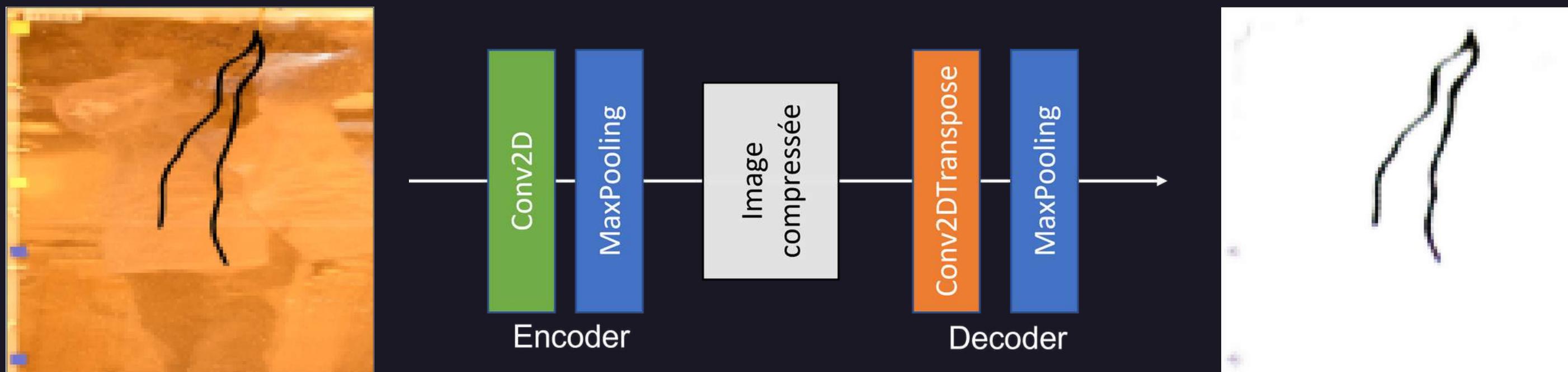
La sortie de chaque neurone suit une distribution normale standard à travers le batch.



Débruitage par autoencoder

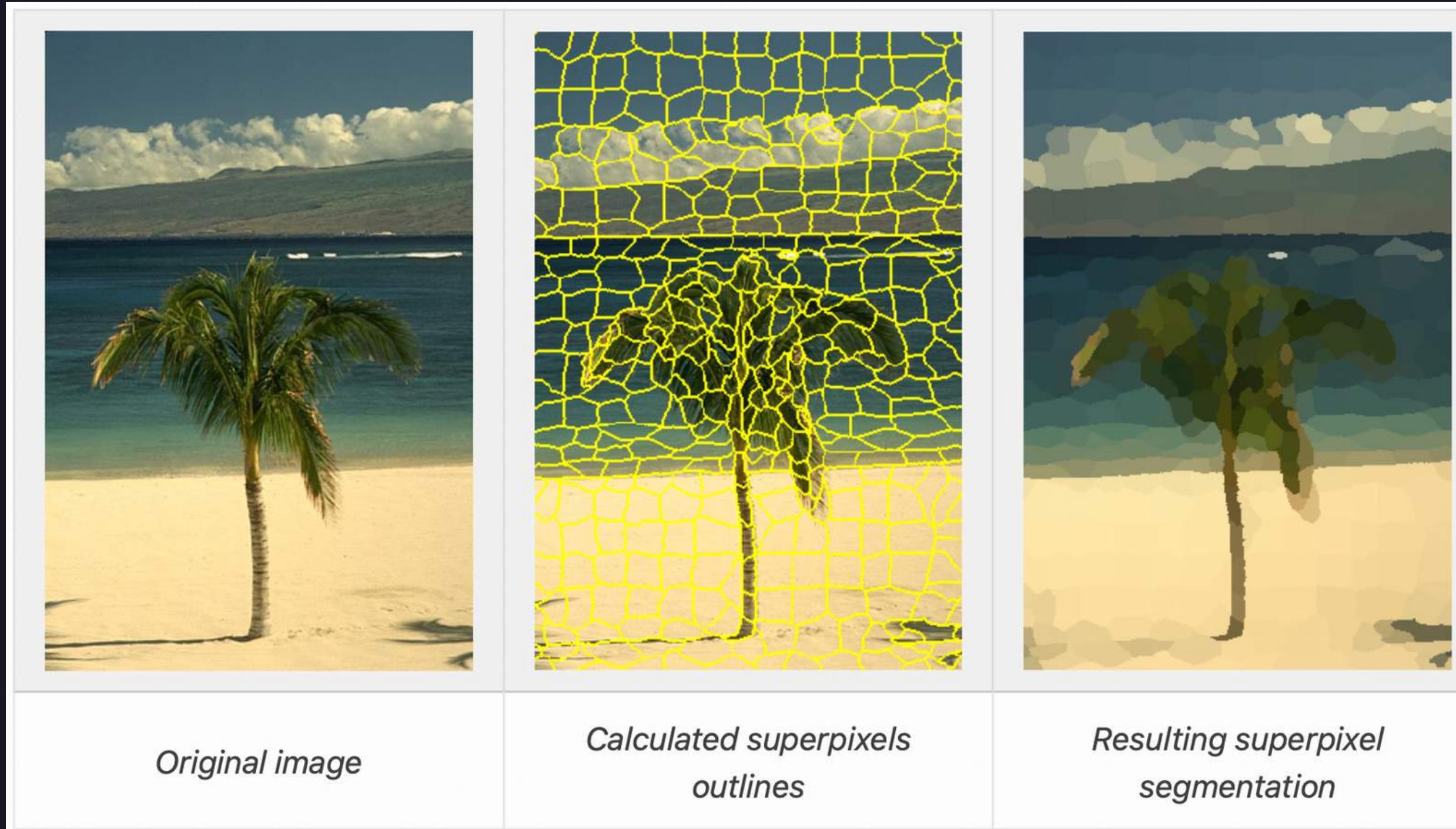
Etape 1 : Construction de l'autoencoder
(capacité à reproduire une image initiale)

Etape 2 : Débruitage des photos



Clustering K-means et Superpixel

Package R : SuperpixelsImageSegmentation



Superpixels = petits groupes homogènes de pixels voisins

Nouveau jeu de données

Jeu de données initial

Pas d'annotations

593 photos répartis dans 54
rhizoboxs

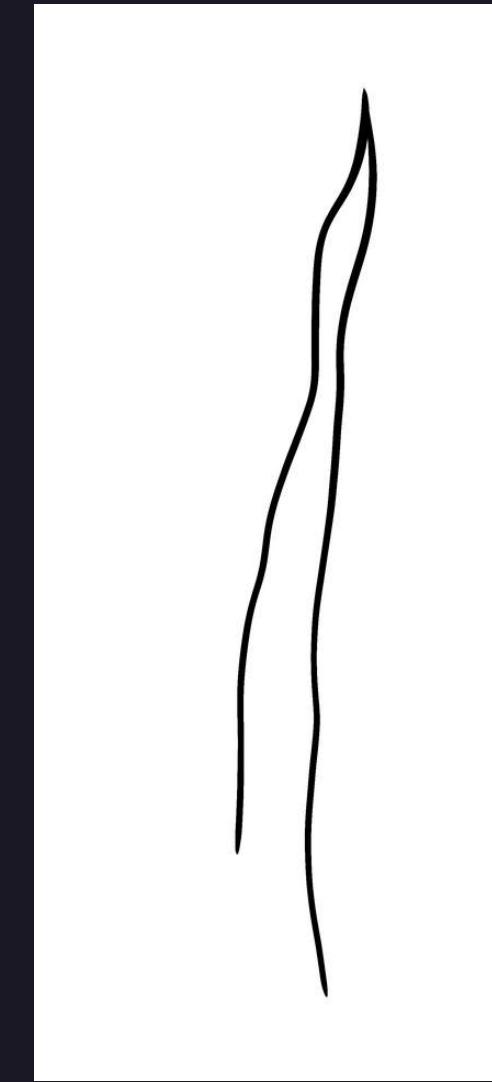


Jeu de données final

Annotations disponibles

Racines re-dessinées avec un trait épais
Rognage manuel du fond

360 photos , 36 rhizoboxs





RÉSULTATS

Comparaison et performance des modèles

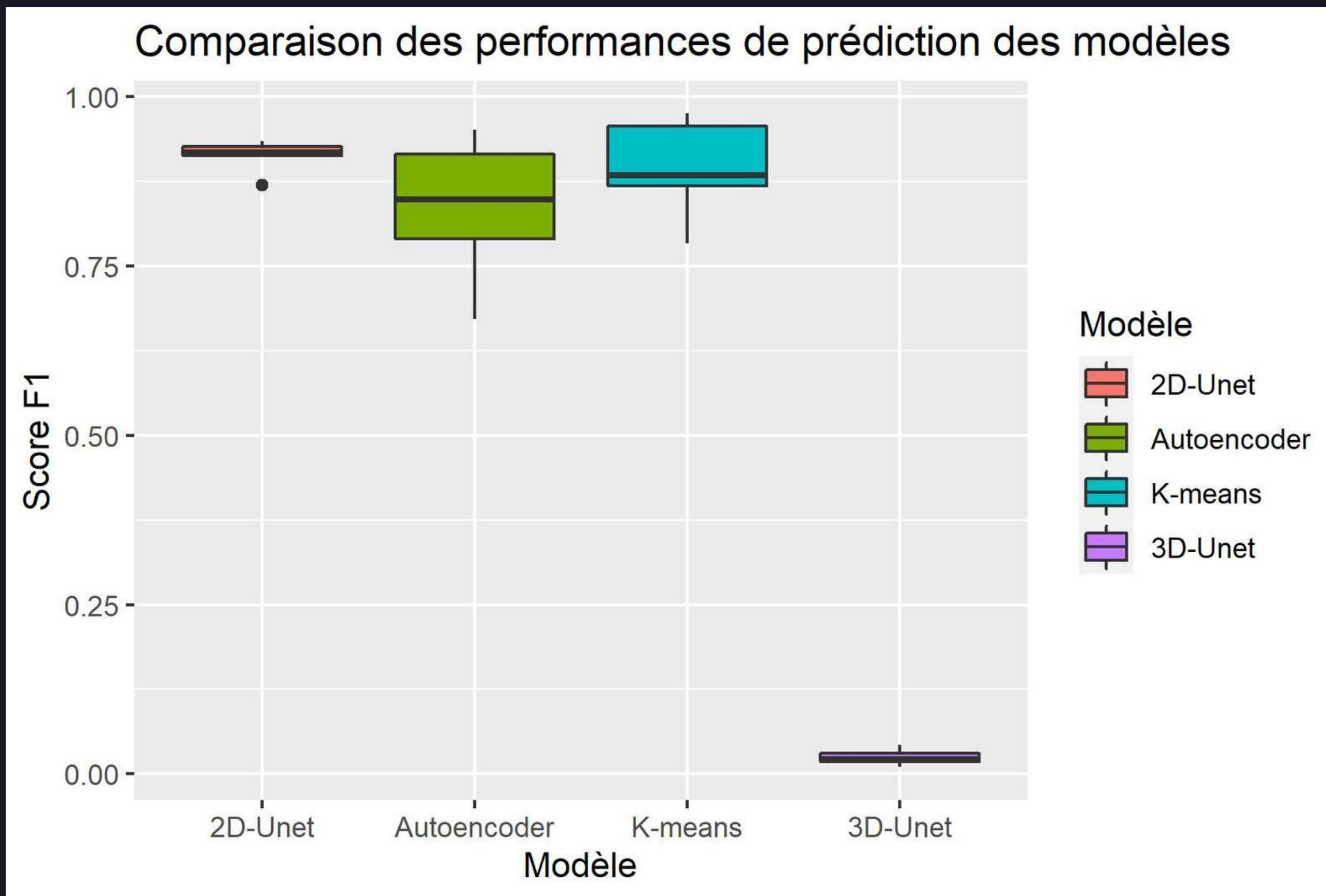
- Deux classes déséquilibrées (peu de racines)
- Binarisation de l'image
- Création de la matrice de confusion pour 10 images par modèle
- Calcul du F1 score pour chacune de ces photos

		Réalité / label	
		Positif (1 = racine)	Négatif (0 = fond)
Prédiction	Positif (1 = racine)	TP (Vrais positifs) = racines	FP (Faux positifs) ex : cadre
	Négatif (0 = fond)	FN (Faux négatifs) ex : bordures des racine	TN (Vrais négatifs) = fond

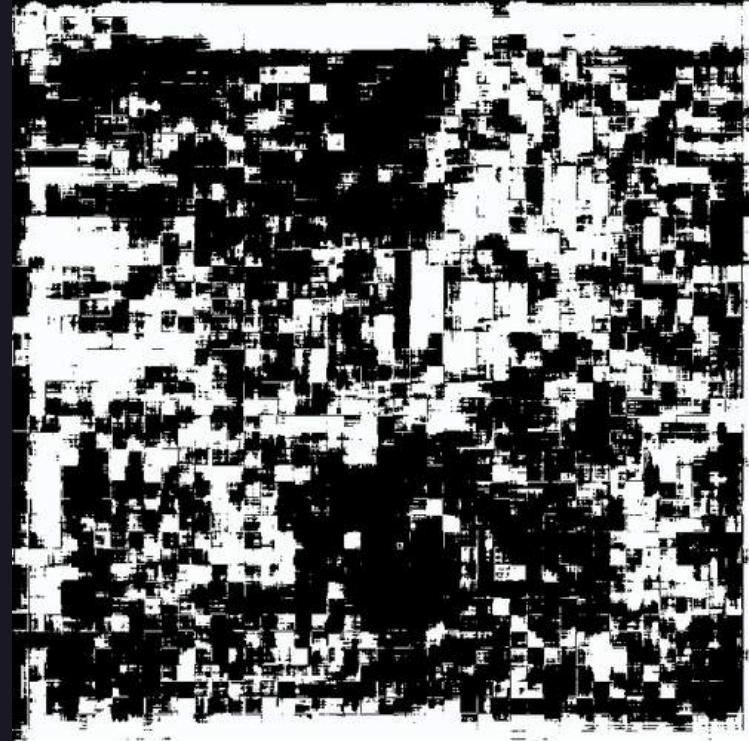
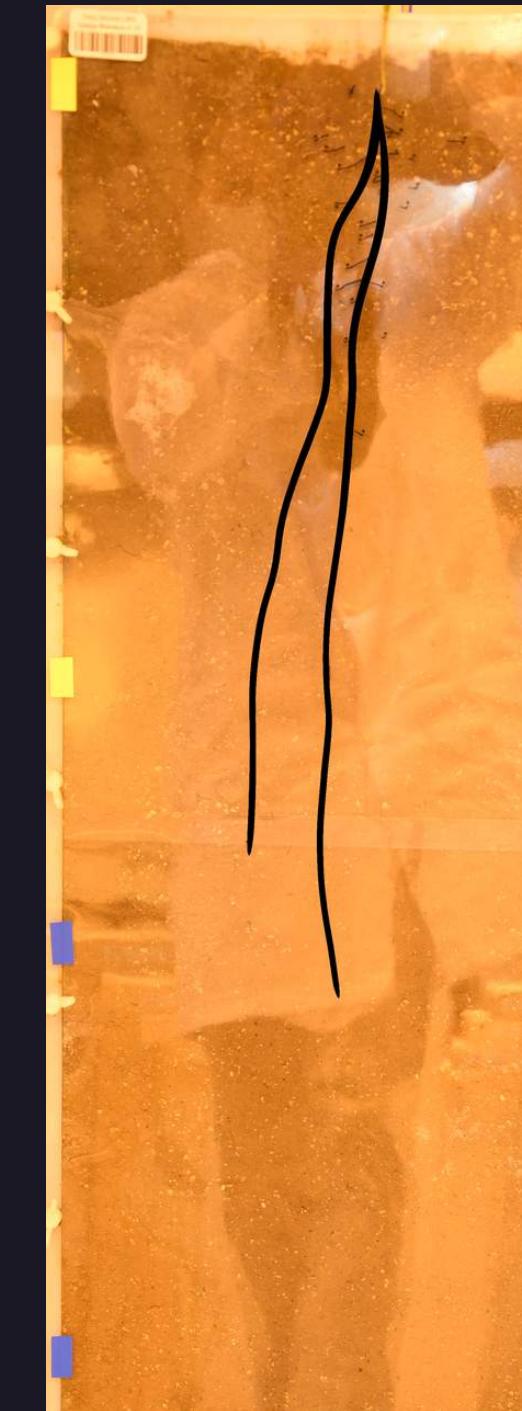
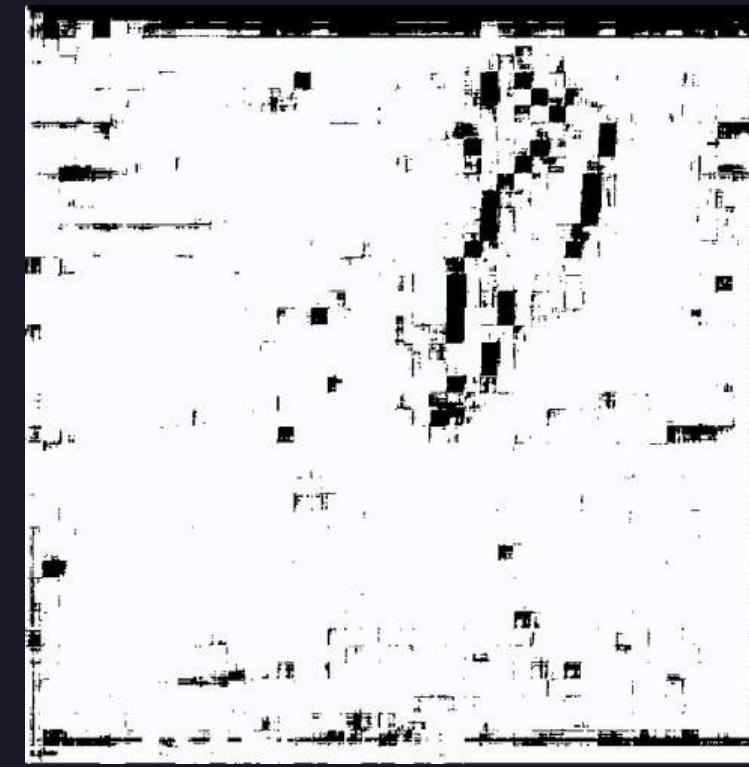
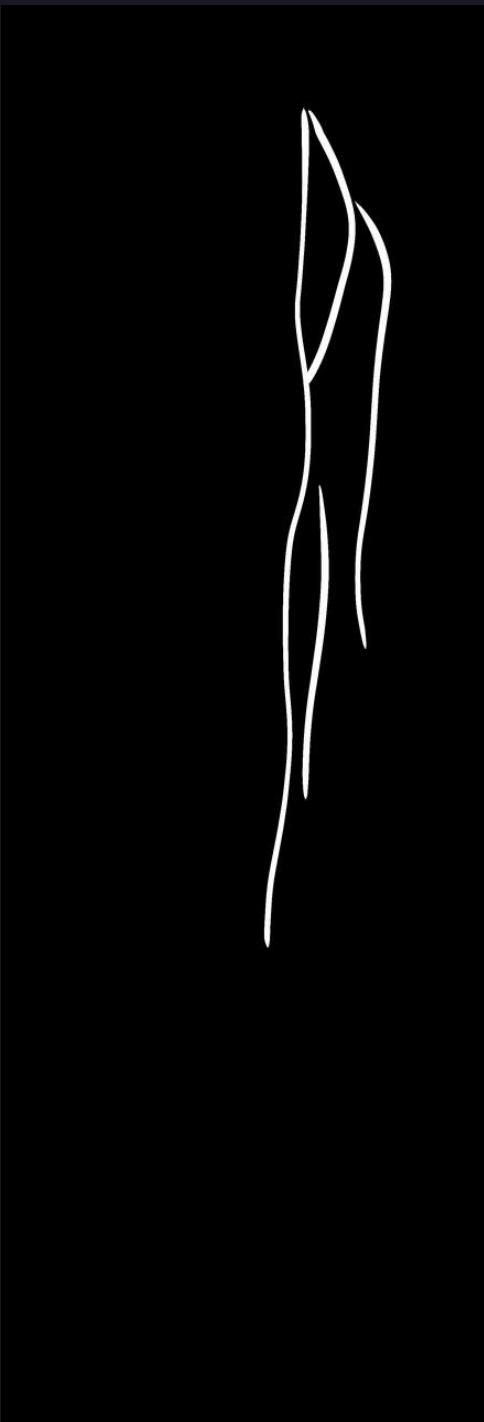
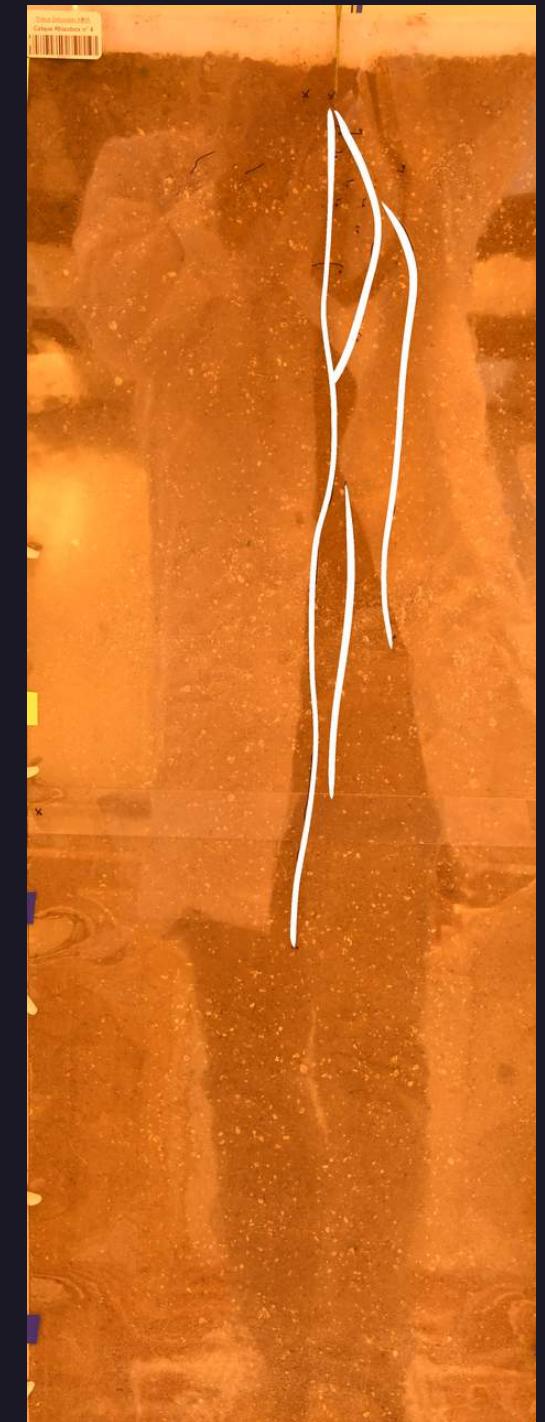
$$F1score = \frac{TP}{TP + \frac{1}{2}(FN + FP)}$$

$$F1score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

Comparaison et performance des modèles



Résultats 3D U-net

*photo**masque attendu**prédiction**photo**masque attendu**prédiction*

Optimizer : RMSprop ($lr=10^{-5}$)

Fonction de perte : sparse_categorical_crossentropy

Activation : Sigmoid

Epochs : 30

3D U-net : Combinations testées

	A	B	C	D	E	F	G	H	I
Optimizer	RMSprop (lr=0,01)	RMSprop (lr=10^-5)	RMSprop (lr=10^-5)	RMSprop (lr=10^-5)	RMSprop (lr=10^-5)	Adam (lr=10^-5)	Adam (lr=10^-5)	Adam (lr=10^-5)	Adam (lr=10^-5)
Fonction de perte	SCCE	SCCE	SCCE	SCCE	SSCE	BCE	SSCE	SSCE	BCE
Activation	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Softmax	Sigmoid	Sigmoid	Softmax	Softmax
Epoch	30	30	60	40	30	30	30	30	30
Accuracy	0	10^-7	10^-7	10^-7	10^-5	10^-4	10^-6	10^-8	10^-8

SCCE : SparseCategorical CrossEntropy

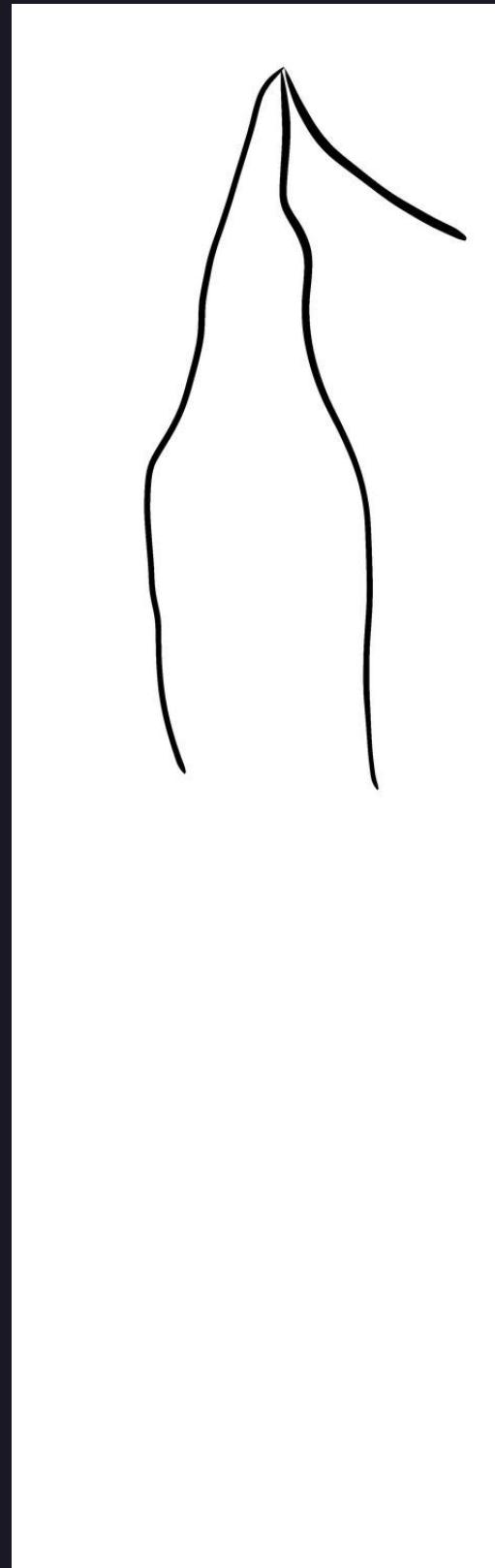
BCE : Binary CrossEntropy

Résultats : 2D U-net

photo



masque attendu

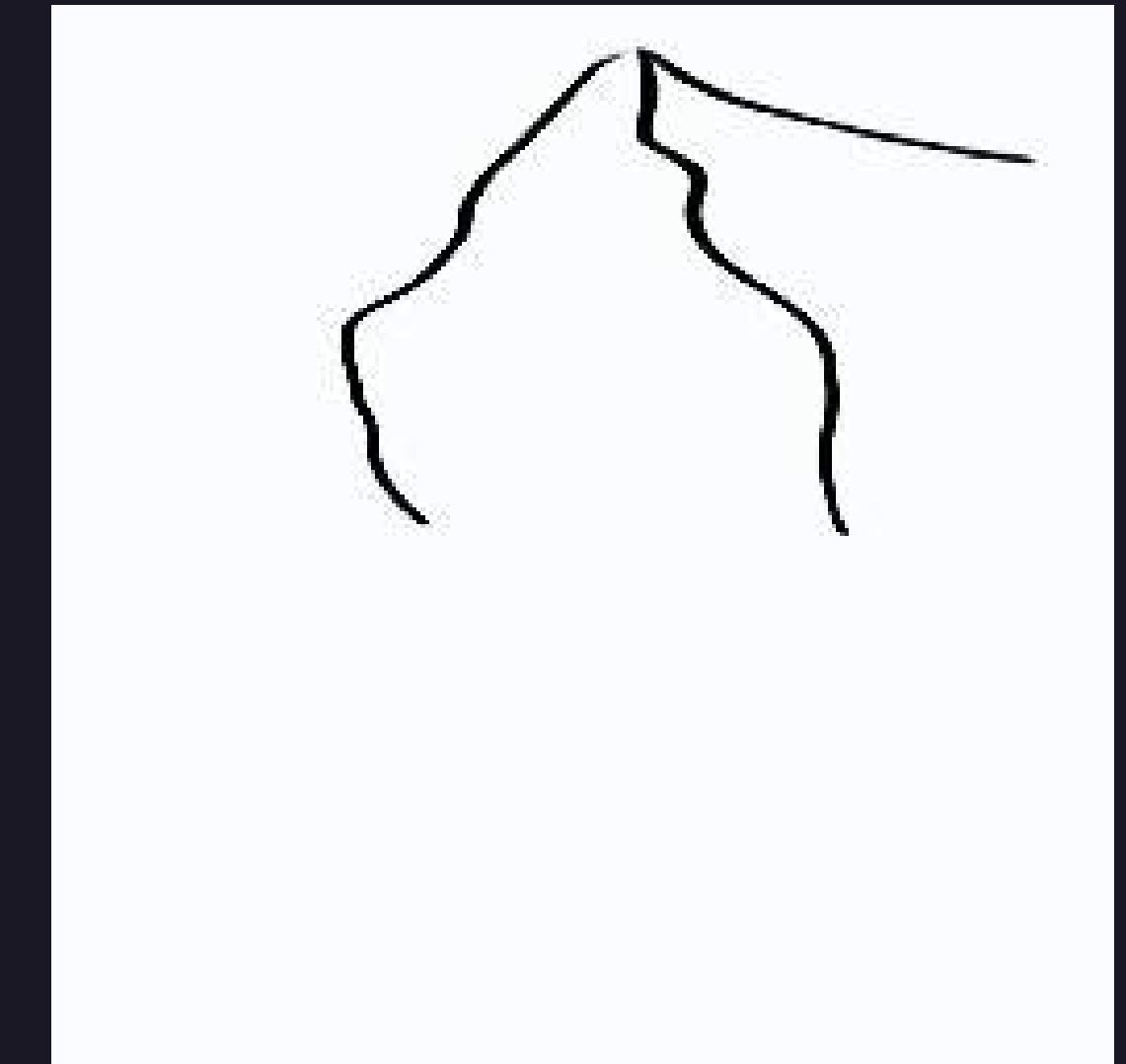


Optimizer : Adam ($lr=10^{-5}$)
Fonction de perte : binary_crossentropy

Activation : Sigmoid
Epochs = 15

Accuracy = 0,98

prédiction

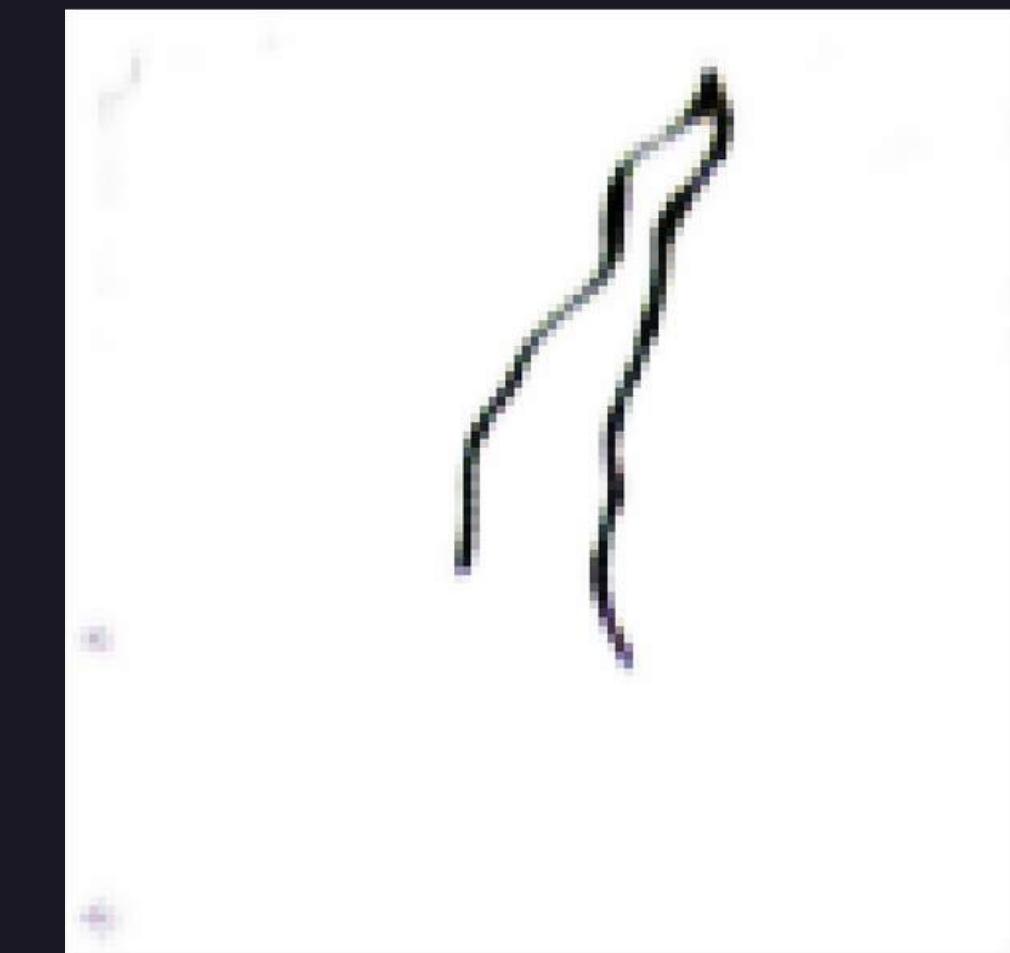


Résultats : autoencoder

photo



prédiction

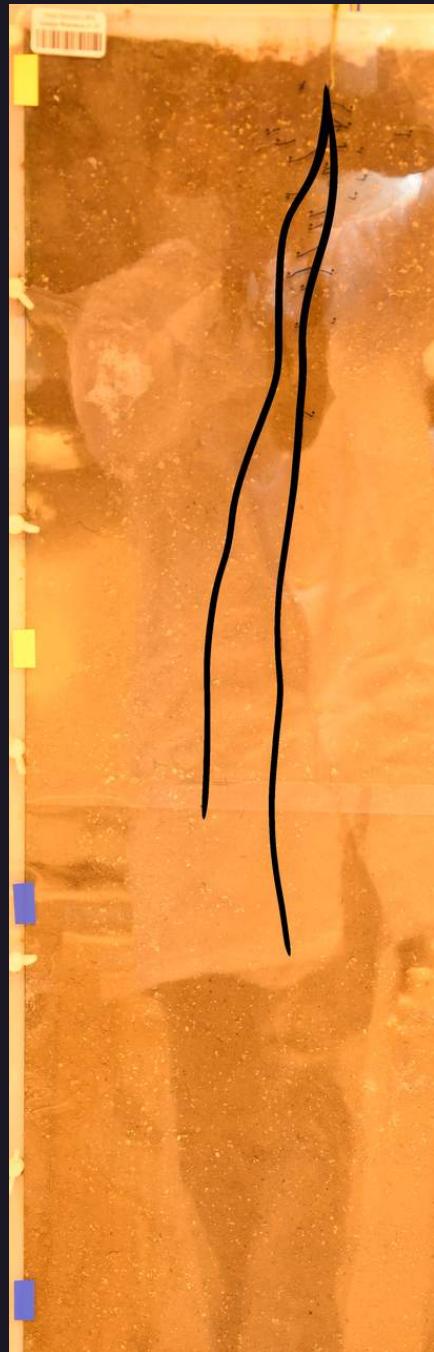


Optimizer : Adam ($lr=10^{-5}$)
Fonction de perte : binary_crossentropy
Activation : Sigmoid
Epochs = 50

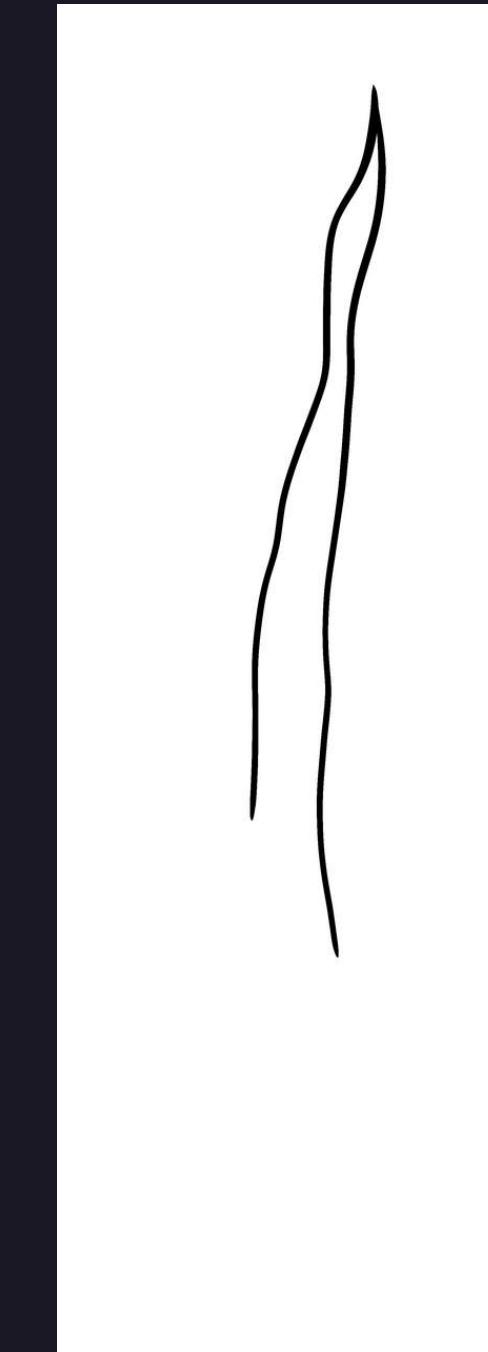
Résultats : Clustering Superpixel

Jeu de données avec racines épaisses noires

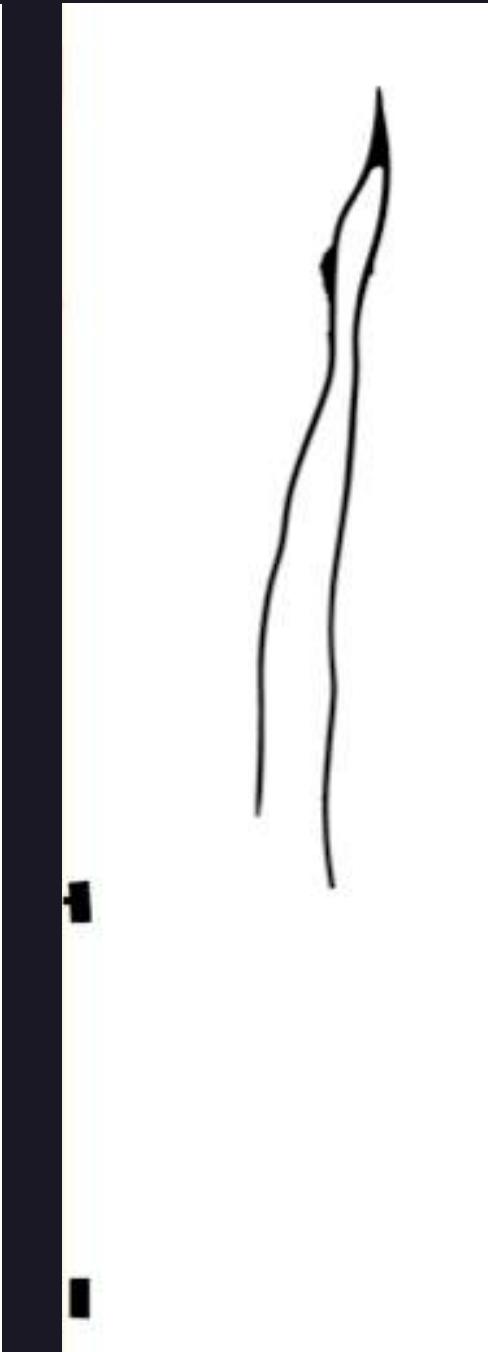
photo



masque attendu



prédiction



Résultats sur le jeu de données initial

Jeu de données initial avec rognage autour du cadre

photo



prédiction superpixel



prédiction 2D U-net



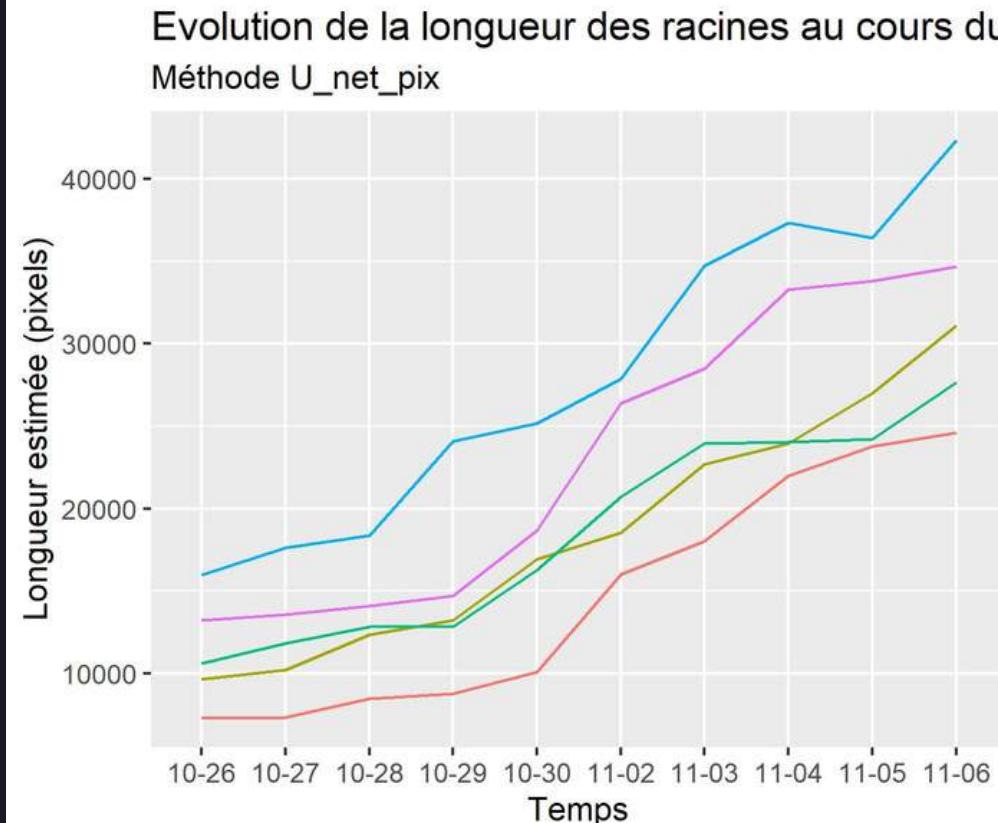
prédiction autoencoder



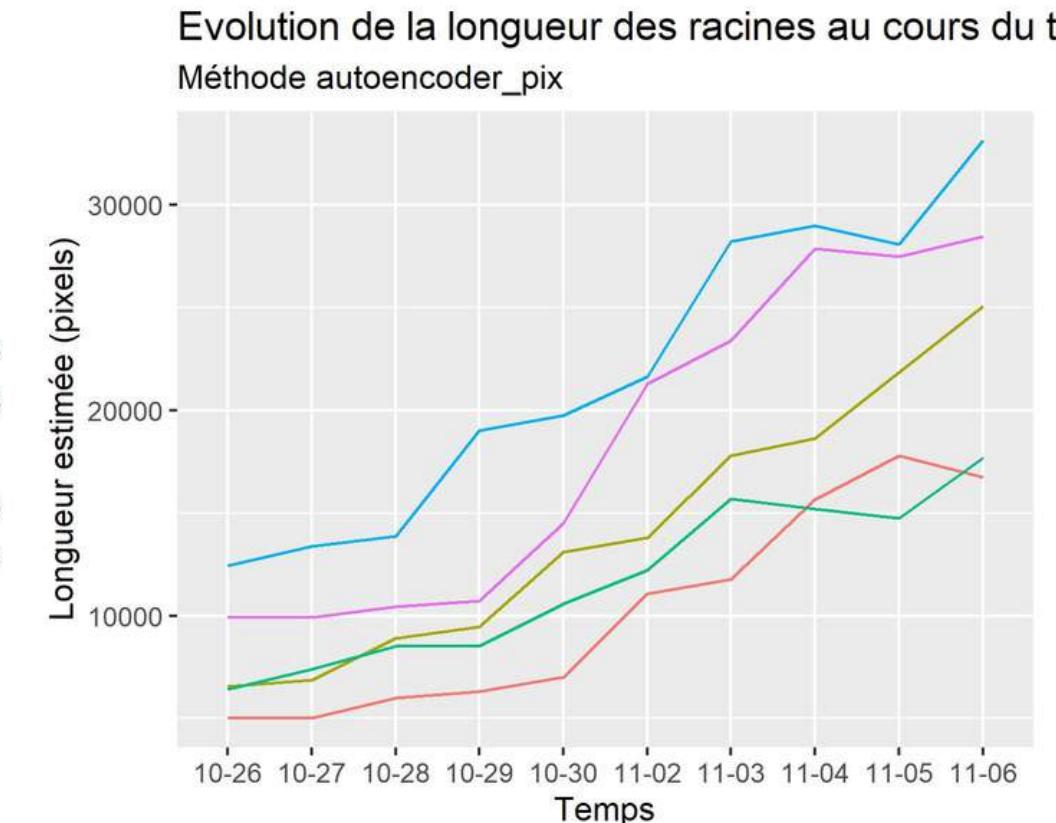
Résultats : évolution de la surface des racines pour les différents modèles

Nombre total de pixels noirs ("méthode pix")

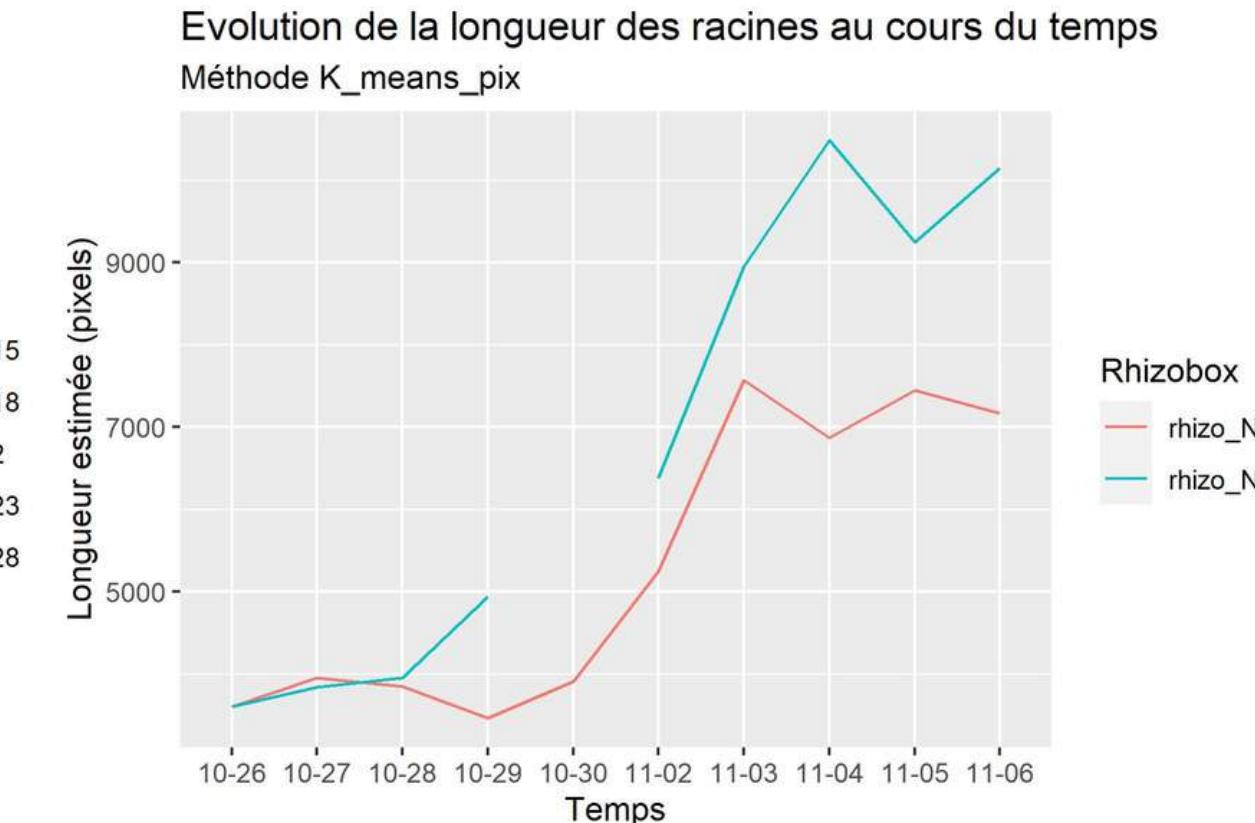
2D U-net



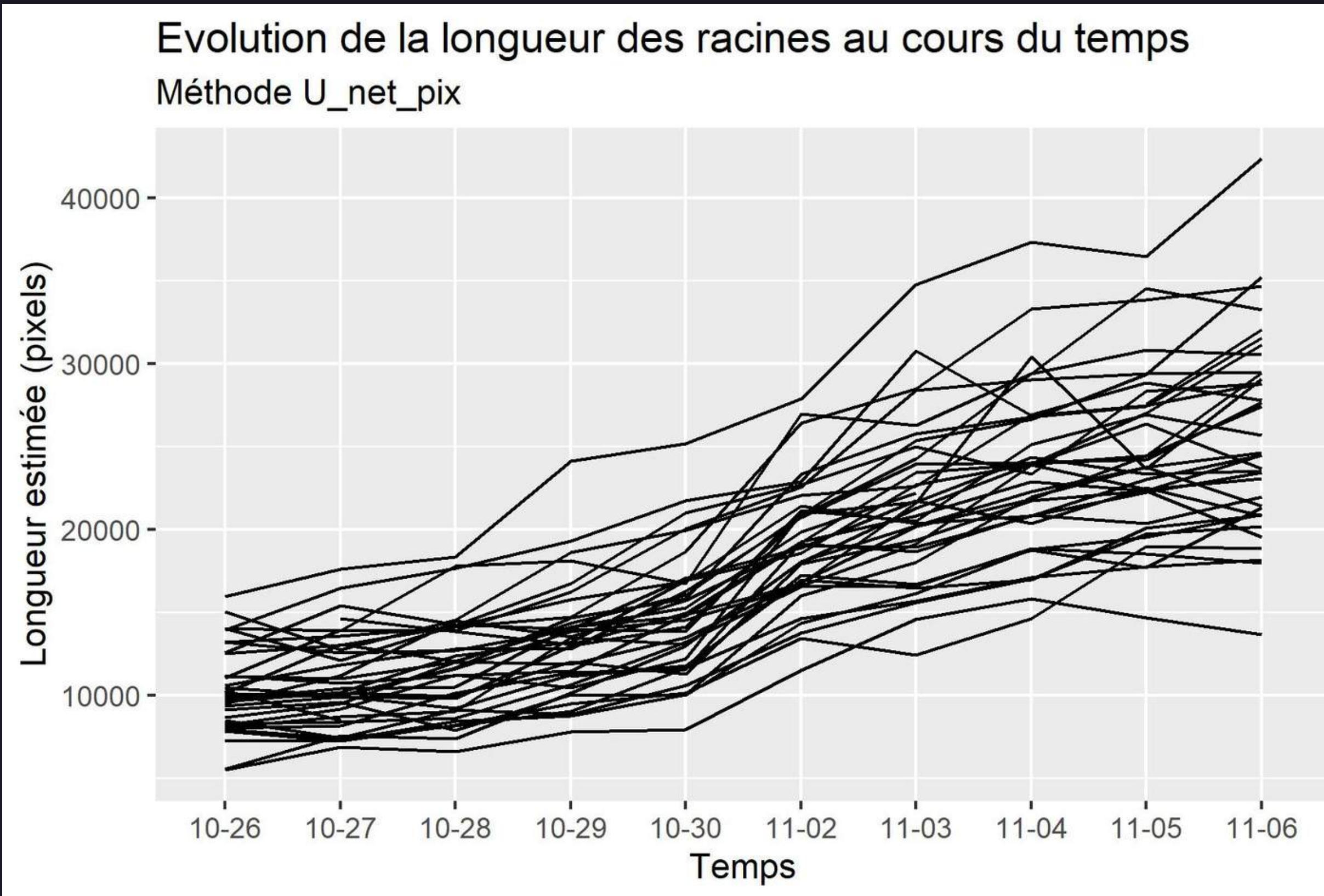
Autoencoder



Superpixel



Evolution de la longueur des racines pour les 36 rhizoboxs au cours du temps



Conditions expérimentales identiques pour chaque rhizobox.



CONCLUSION

Jeu de données initial trop complexe à cause de la couleur et de la qualité des photos

Modifications possibles du protocole expérimental :

- Changer la couleur du cadre pour pouvoir rogner le fond de la photo
- Utiliser un trépied bien réglé afin de prendre les photos droites
- Repasser les racines sur le rhizobox avec un feutre plus épais et plus opaque

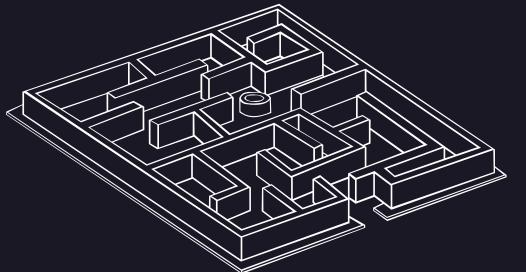
Le modèle 2D U-net, plus précis et plus rapide est à préconiser

Calculer le nombre de pixels total pour estimer la longueur permet de contourner le biais dû aux faux positifs (pixels mal prédits appartenant au cadre)

Pour aller plus loin...



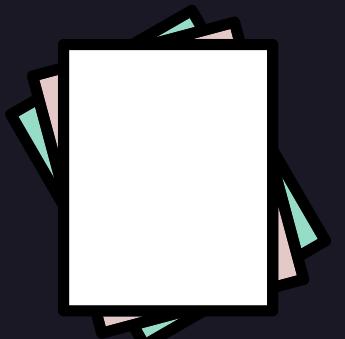
- Data augmentation - Augmentation du nombre de données



- Réseaux de neurones plus complexes



- Autre métrique de comparaison adaptée aux labels en niveau de gris

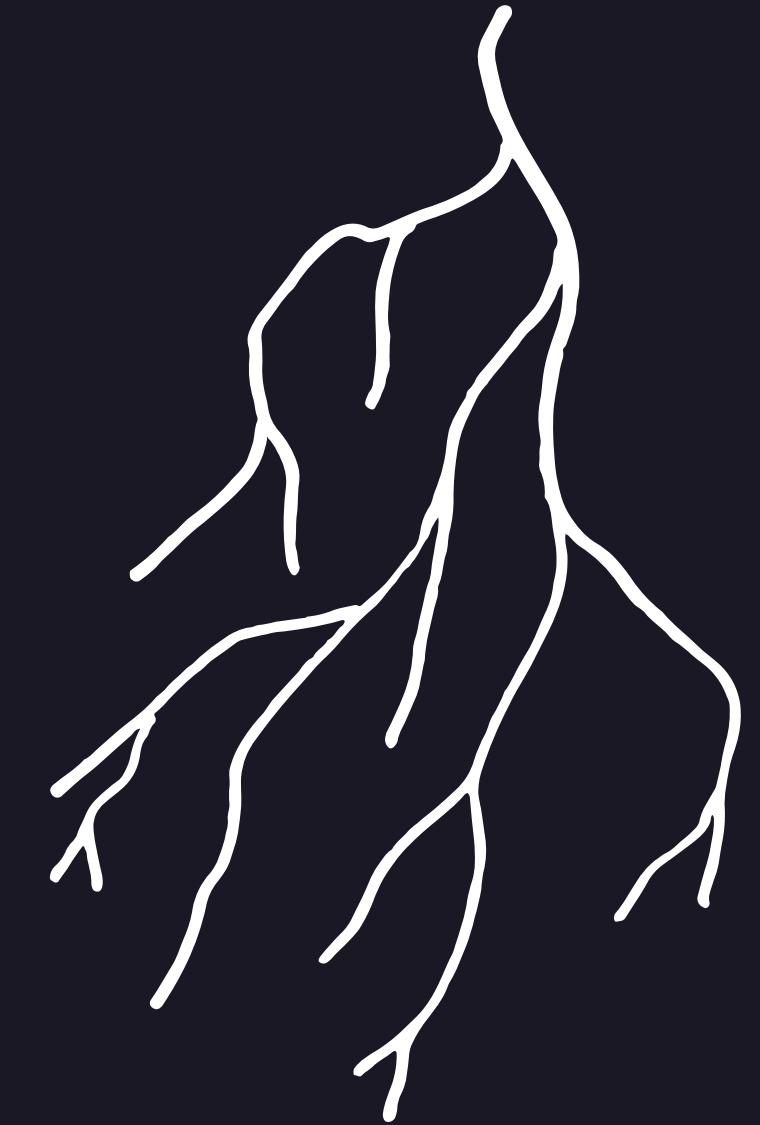


- Annotation d'image pour entraîner un réseau capable de calculer précisément la longueur

Bibliographie

- 1.Yang, G., Pennington, J., Rao, V., Sohl-Dickstein, J. & Schoenholz, S. S. A Mean Field Theory of Batch Normalization. (2019).
- 2.Yu, E. M., Iglesias, J. E., Dalca, A. V. & Sabuncu, M. R. An Auto-Encoder Strategy for Adaptive Image Segmentation. arXiv:2004.13903 [cs, eess] (2020).
- 3.Adaloglou, N. An overview of Unet architectures for semantic segmentation and biomedical image segmentation. AI Summer <https://theaisummer.com/unet-architectures/> (2021).
- 4.Huber, J. Batch normalization in 3 levels of understanding. Medium <https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338> (2022).
- 5.Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. (2015).
- 6.Zhou, X. et al. EAST: An Efficient and Accurate Scene Text Detector. arXiv:1704.03155 [cs] (2017).
- 7.Santurkar, S., Tsipras, D., Ilyas, A. & Madry, A. How Does Batch Normalization Help Optimization? (2018).
- 8.Alom, M. Z., Yakopcic, C., Taha, T. M. & Asari, V. K. Nuclei Segmentation with Recurrent Residual Convolutional Neural Networks based U-Net (R2U-Net). in NAECON 2018 - IEEE National Aerospace and Electronics Conference 228–233 (2018). doi:10.1109/NAECON.2018.8556686.
- 9.Alom, M. Z., Yakopcic, C., Hasan, M., Taha, T. M. & Asari, V. K. Recurrent residual U-Net for medical image segmentation. JMI 6, 014006 (2019).
- 10.Smith, A. G., Petersen, J., Selvan, R. & Rasmussen, C. R. Segmentation of roots in soil with U-Net. Plant Methods 16, 13 (2020).
- 11.Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (eds. Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F.) 234–241 (Springer International Publishing, 2015).
- 12.Falk, T. et al. U-Net: deep learning for cell counting, detection, and morphometry. Nat Methods 16, 67–70 (2019).

Merci pour votre
attention !



ANNEXES

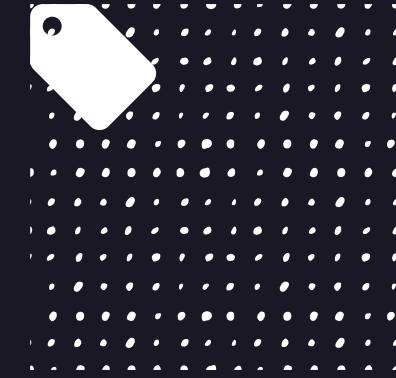


Logiciel SmartRoot : annotation semi-automatique des
images = création d'un masque

1 image = 3h de travail

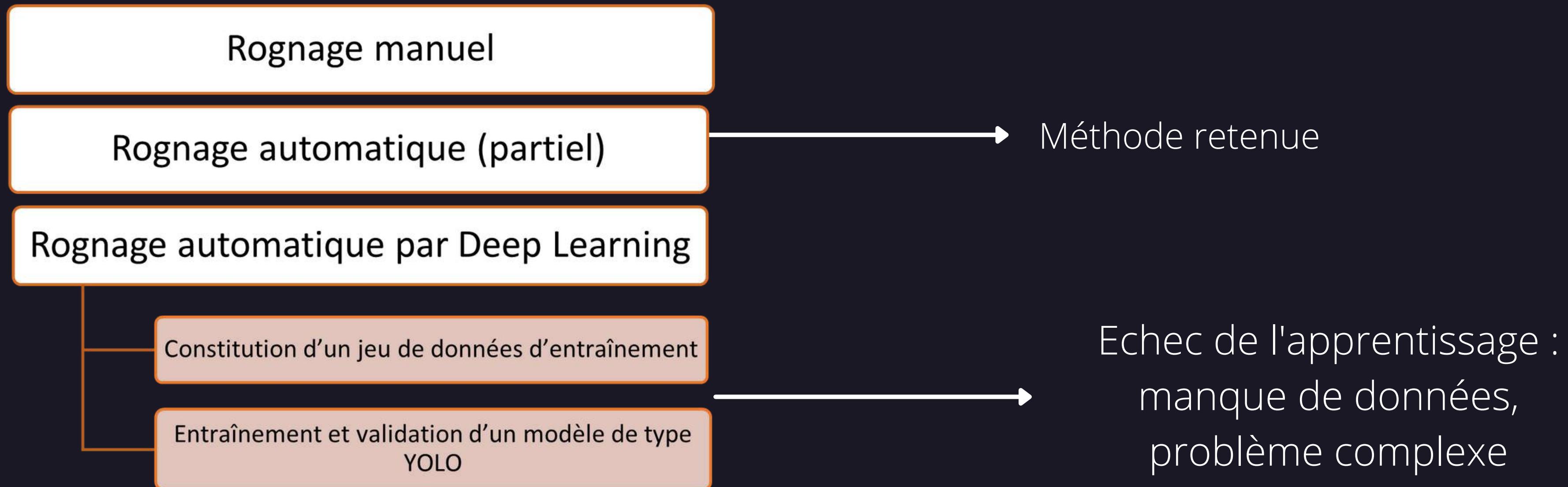


Problème complexe à cause de la qualité et de la couleur des images ainsi que du manque d'annotation de nos photos.



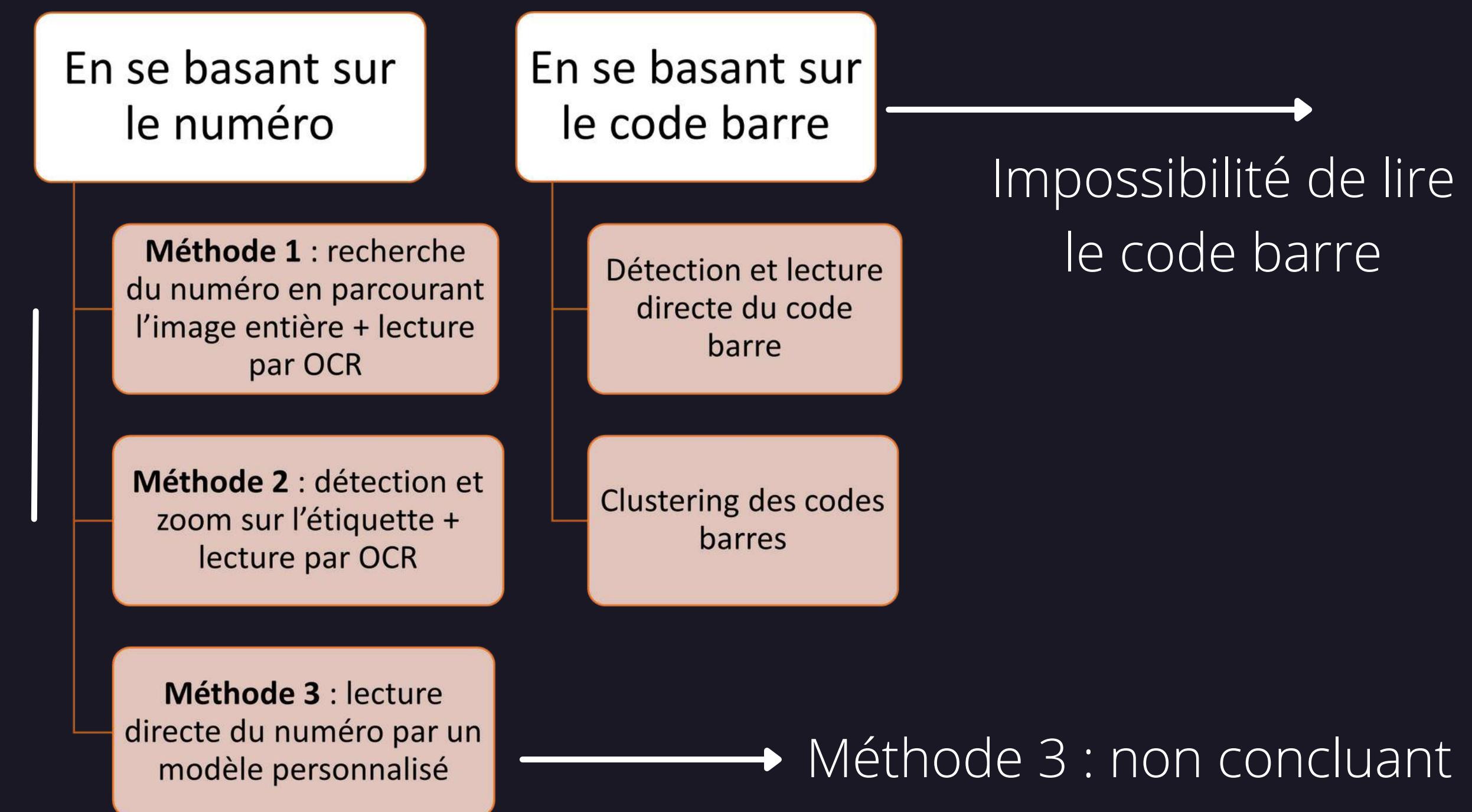
- Premier problème rencontré lié au pré-traitement des données et la construction d'une banque d'images :
 - Identification des rhizoboxs (photos non triées)
 - Redimensionnement au niveau du cadre.
- Deuxième étape : segmentation des racines dans le sol :
 - Quel(s) modèle(s) permet(tent) de segmenter au mieux les racines ? Quelles sont les caractéristiques de ce(s) modèle(s) ?
 - Est-ce que les modèles de Deep Learning permettent une meilleure segmentation des racines ?
 - Quelles sont les caractéristiques du jeu de données qui permettent d'obtenir un résultat satisfaisant avec les différents modèles ?
- Troisième étape :
 - Calcul de la longueur des racines, quelle(s) méthode(s) ?
 - Pistes d'amélioration pour faciliter l'analyse des images.

Constitution d'une banque de données - Rognage des zones en dehors du cadre

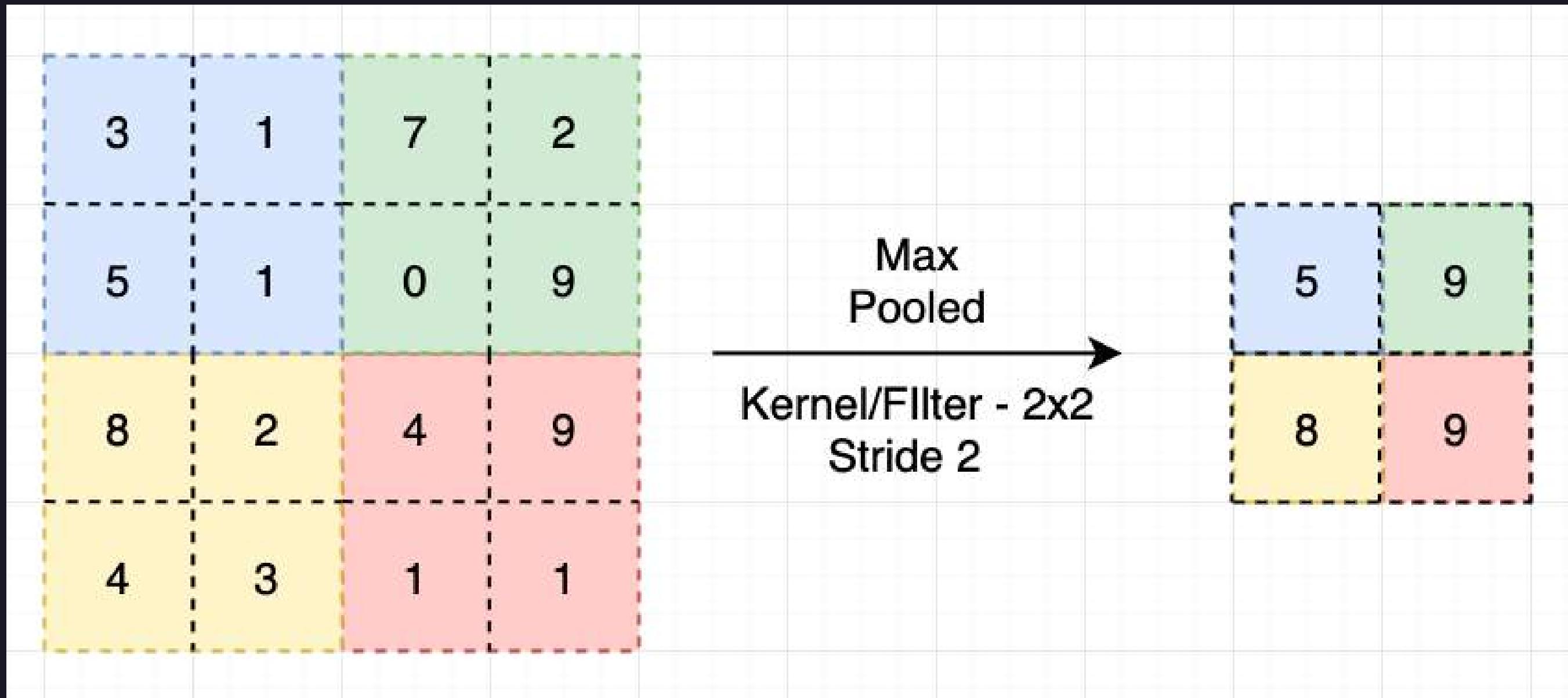


Constitution d'une banque de données - Individualisation des rhizobox par Deep Learning

COMPARAISON



U-net : les couches de MaxPooling



Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;

Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{mini-batch mean}$$

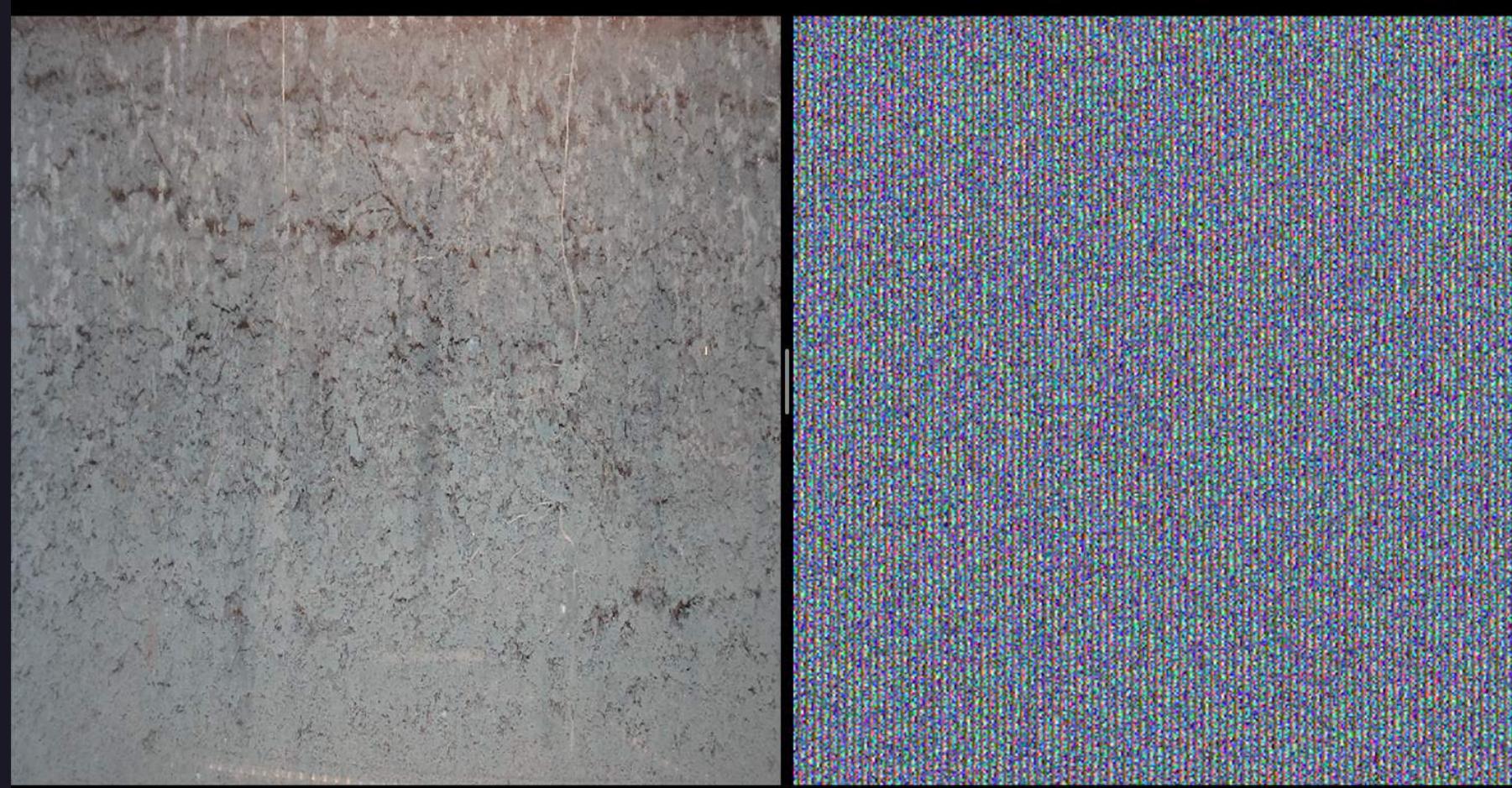
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Segmentation sémantique avec 3D U-net - Keras



Accuracy proche de 0

Accuracy

=

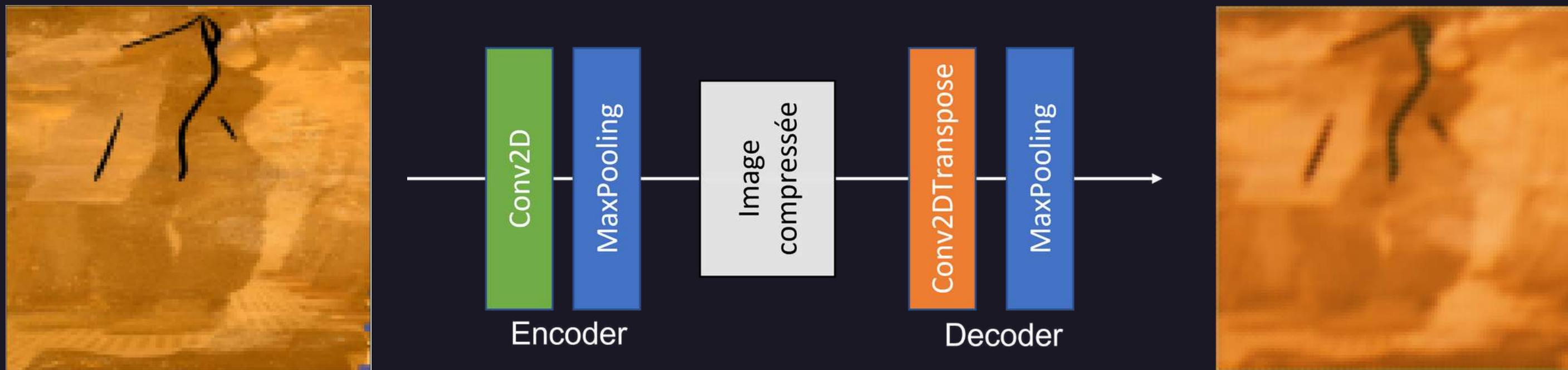
Nombre de pixels bien prédis /
nombre de pixels total

2 hypothèses :

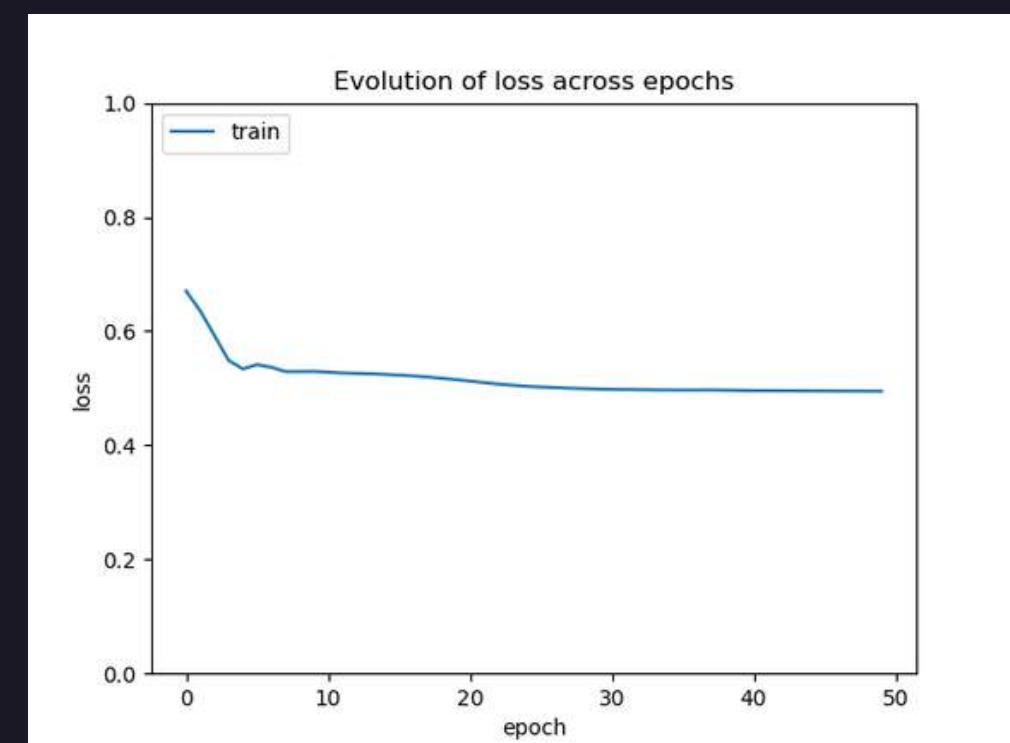
- Jeu de données trop complexe - pas assez de données
- Problème avec le réseau 3D - U-net et le traitement d'image couleur

Débruitage par autoencoder

Etape 1 : vérification de l'autoencoder



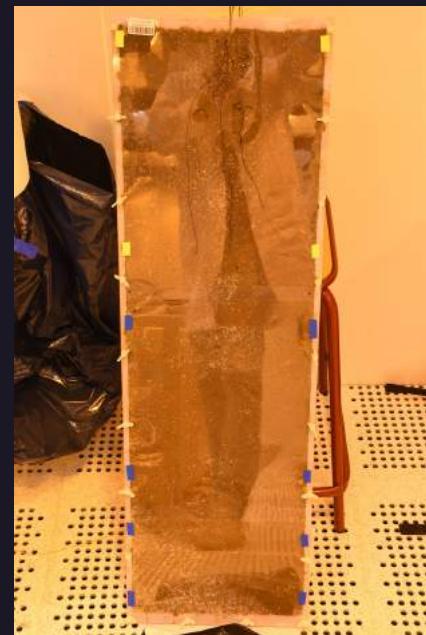
- epochs : 50
- optimizer : adam
- loss : BCE
- batch size : 12
- image size : (112, 112)



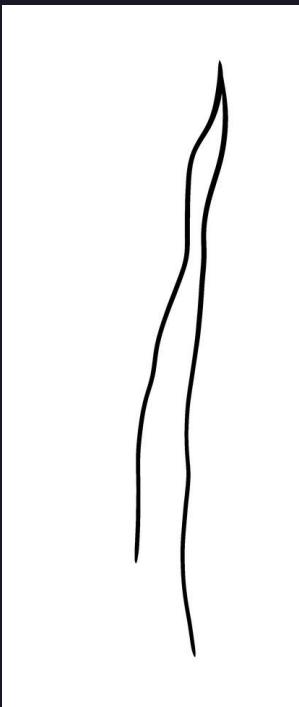
Jeux de données initial

Pas d'annotations

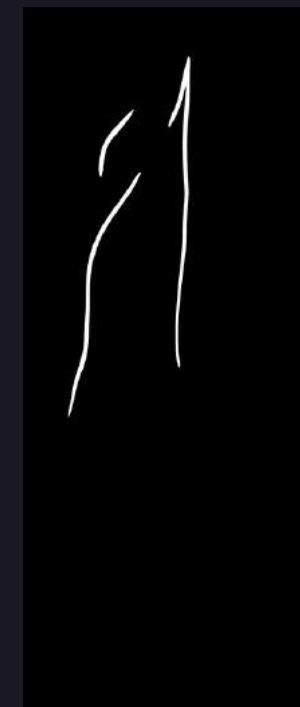
593 photos répartis dans 54 rhizoboxs



Jeu de données avec racines épaisses noires masque fond blanc (60 photos)



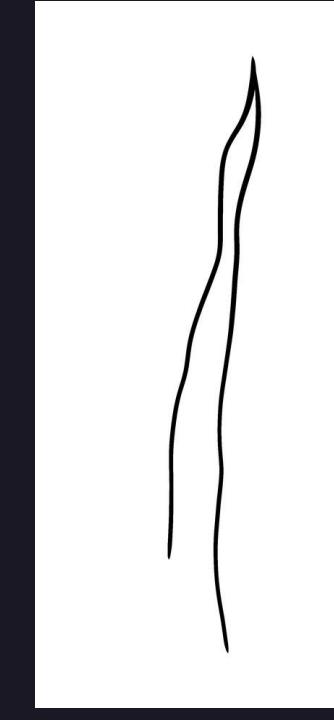
Jeu de données avec racines épaisses blanches masque fond noir (60 photos)



Jeu de données avec racines épaisses beige masque fond blanc (30 photos)



Jeu de données final (360 photos , 36 rhizoboxs)



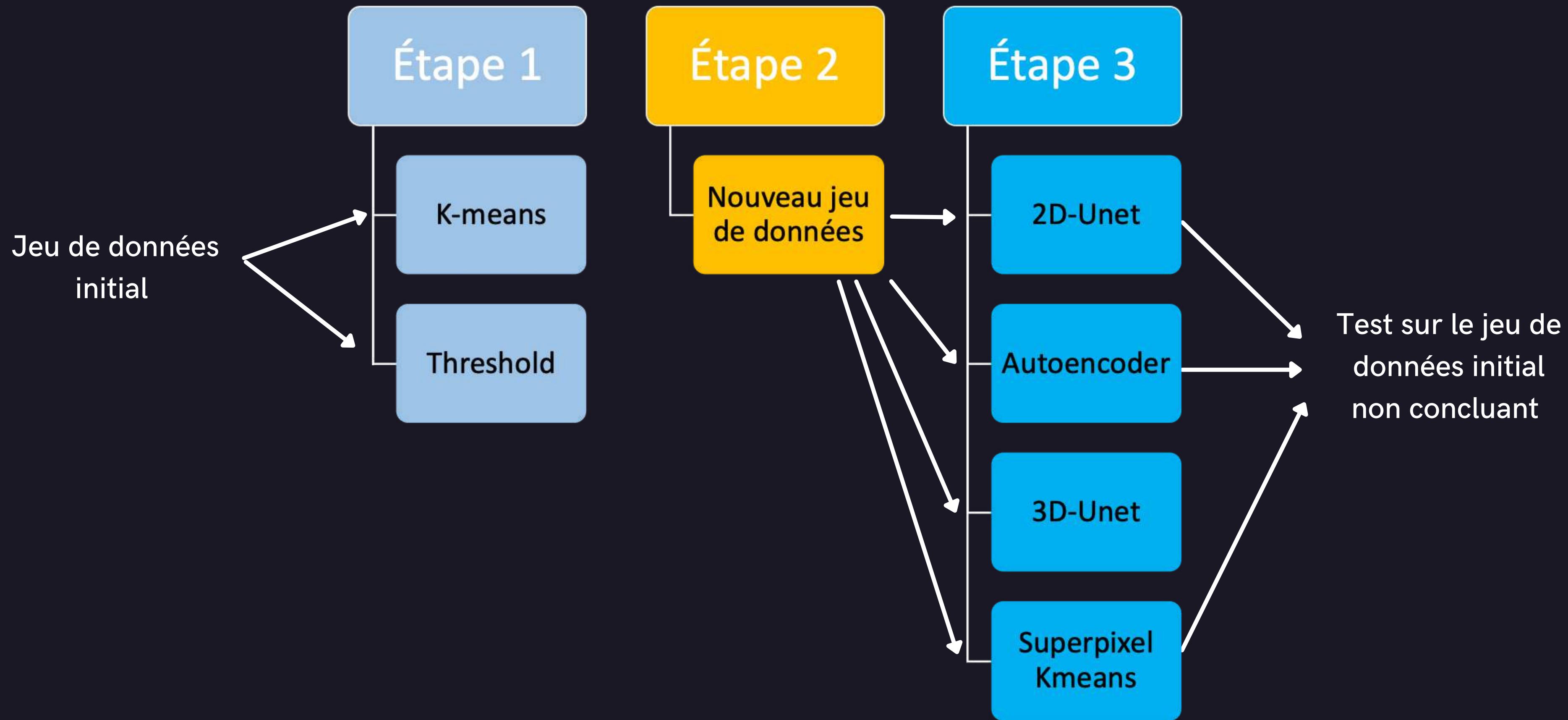
Jeux de données

Jeu de données issu de l'article "Segmentation of Roots in Soil with U-Net"

48 photos annotées



Résumé des modèles et jeux de données



Comparaison des 2 modèles U-net

	A	B	C	D	E	F	G	H	I
Optimizer	RMSprop (lr=0,01)	RMSprop (lr=10^-5)	RMSprop (lr=10^-5)	RMSprop (lr=10^-5)	RMSprop (lr=10^-5)	Adam (lr=10^-5)	Adam (lr=10^-5)	Adam (lr=10^-5)	Adam (lr=10^-5)
Fonction de perte	SCCE	SCCE	SCCE	SCCE	SSCE	BCE	SSCE	SSCE	BCE
Activation	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Softmax	Sigmoid	Sigmoid	Softmax	Softmax
Epoch	30	30	60	40	30	30	30	30	30
Accuracy	0	10^-7	10^-7	10^-7	10^-5	10^-4	10^-6	10^-8	10^-8

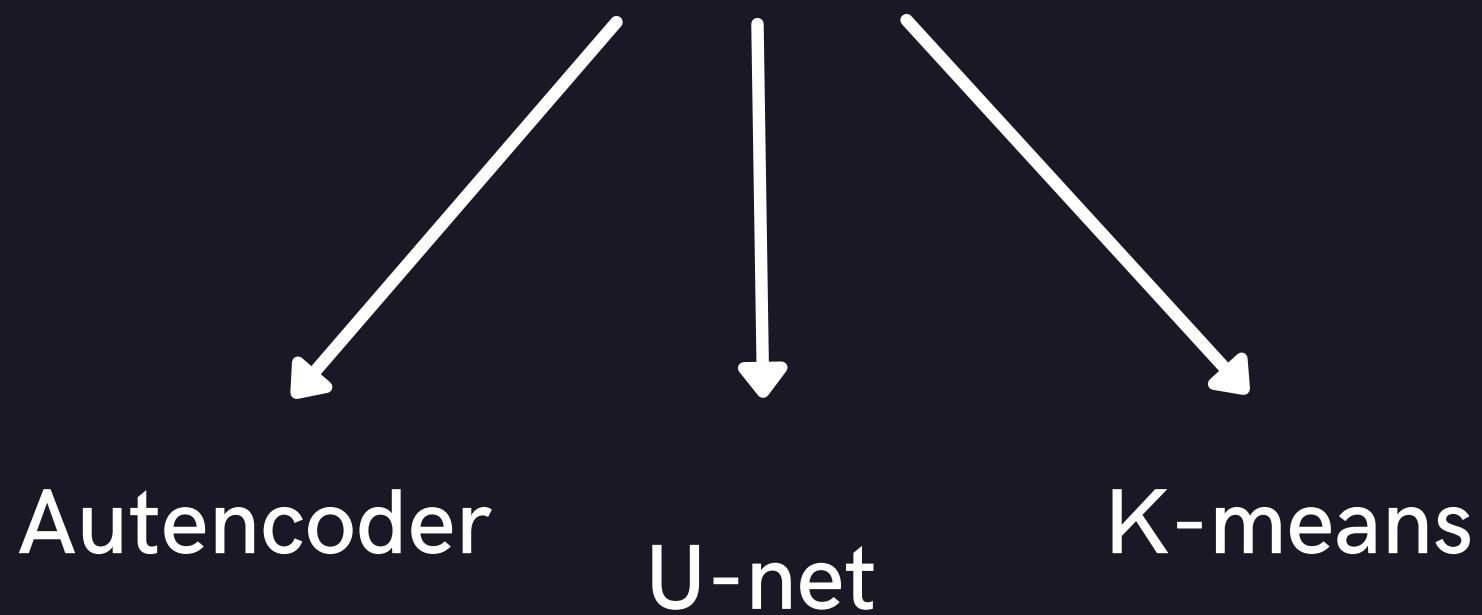
Mêmes caractéristiques que le réseau 2D U-net



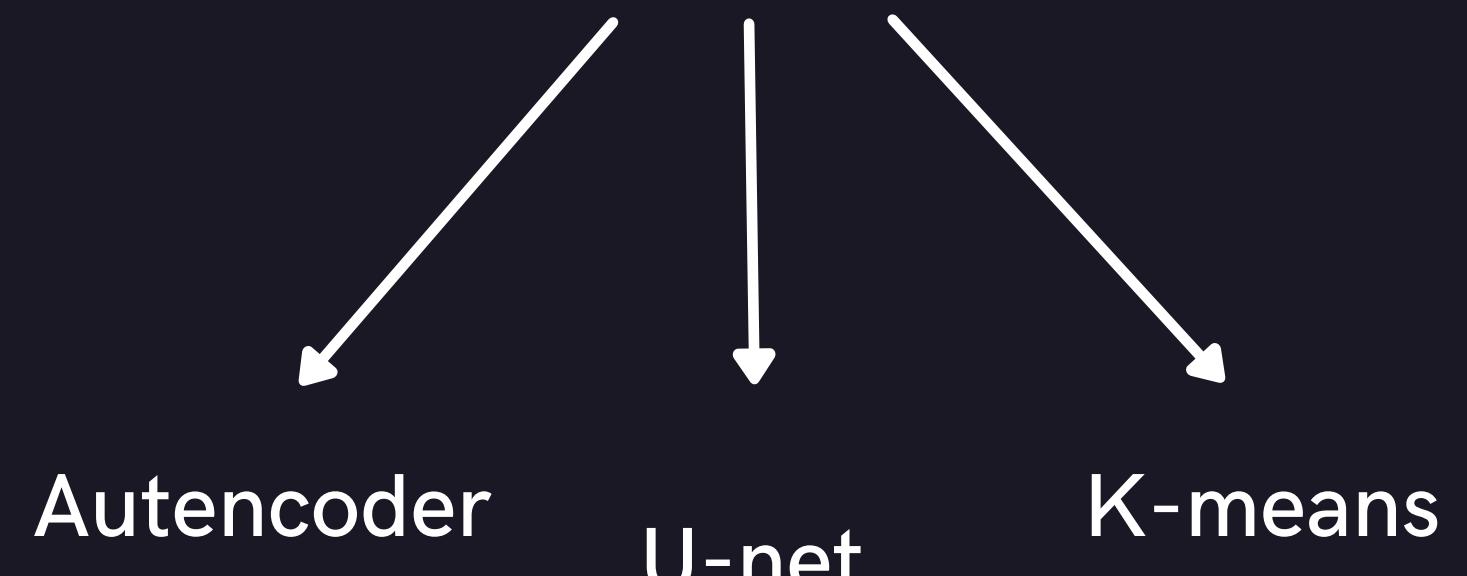
Les petites différences de structure du modèle semblent être le point important ici. En effet, les caractéristiques présentées n'influencent que très peu l'apprentissage du réseau sur nos photos de racines.

Calcul de la longueur des racines

Nombre total de pixels noirs
(méthode pix)



Différence entre le pixel noir le plus haut
et le plus bas
(méthode diff)

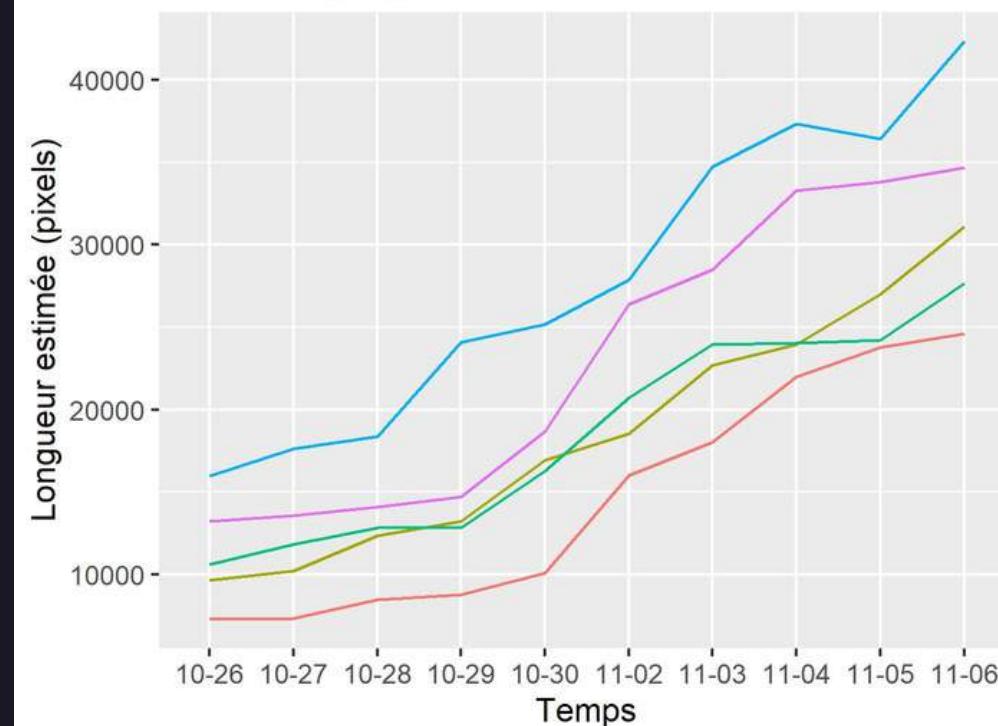


Comparaison des méthodes

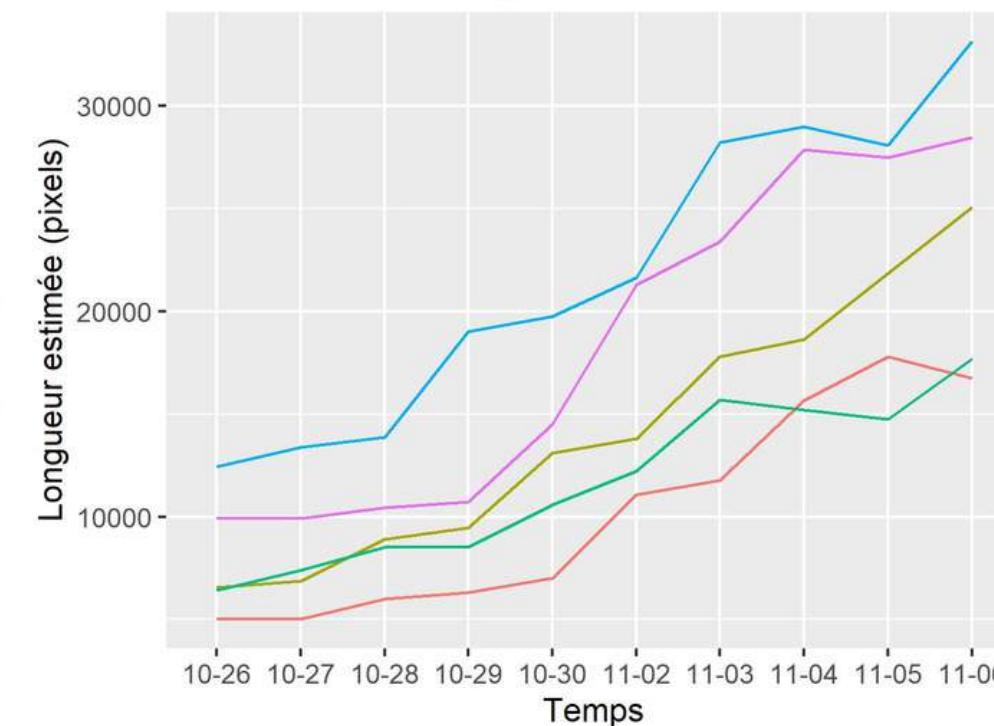
Méthode Pix	Méthode diff
Peu de variations dû aux faux positifs (cadre)	Beaucoup de variations et de valeurs inutilisables à cause des faux positifs
Calcul de la surface et non de la longueur	Calcul plus proche de la longueur
Possible de calculer une erreur moyenne grâce à l'estimation de la performance des modèles.	Calcul de l'erreur plus complexe

Surface (méthode pix)

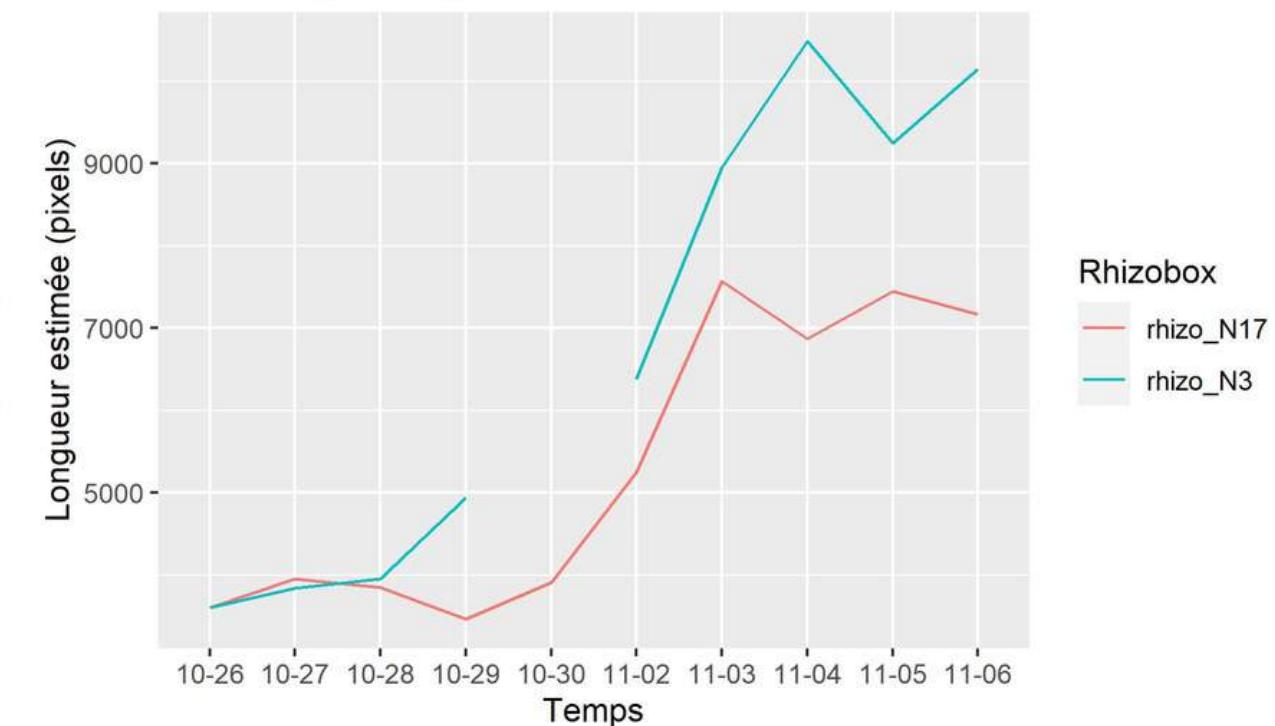
Evolution de la longueur des racines au cours du temps
Méthode U_net_pix



Evolution de la longueur des racines au cours du temps
Méthode autoencoder_pix

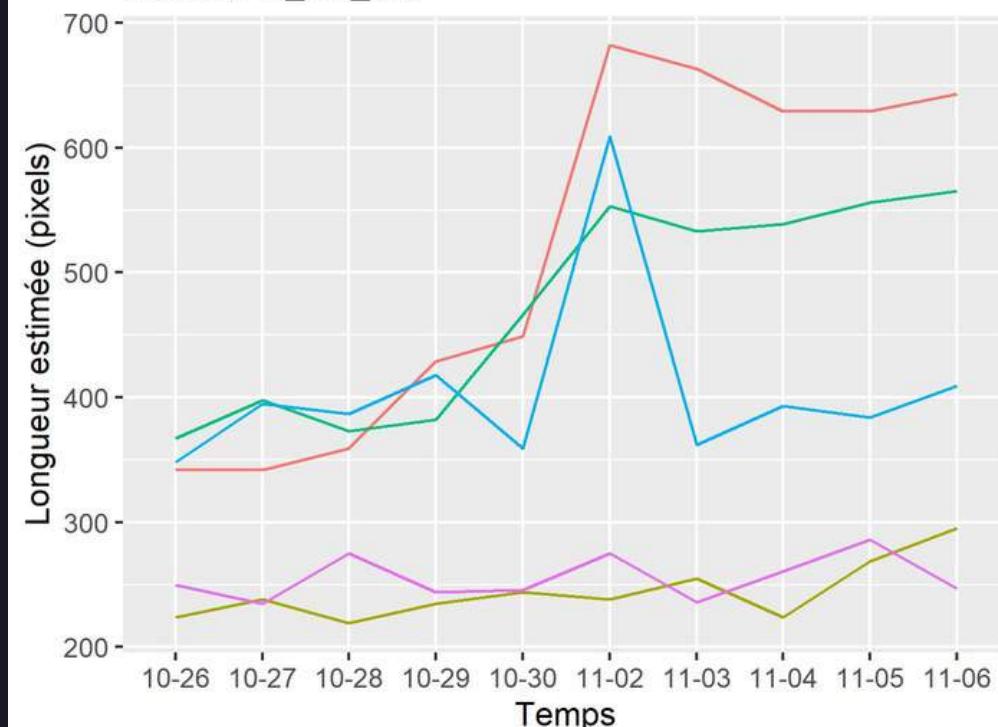


Evolution de la longueur des racines au cours du temps
Méthode K_means_pix

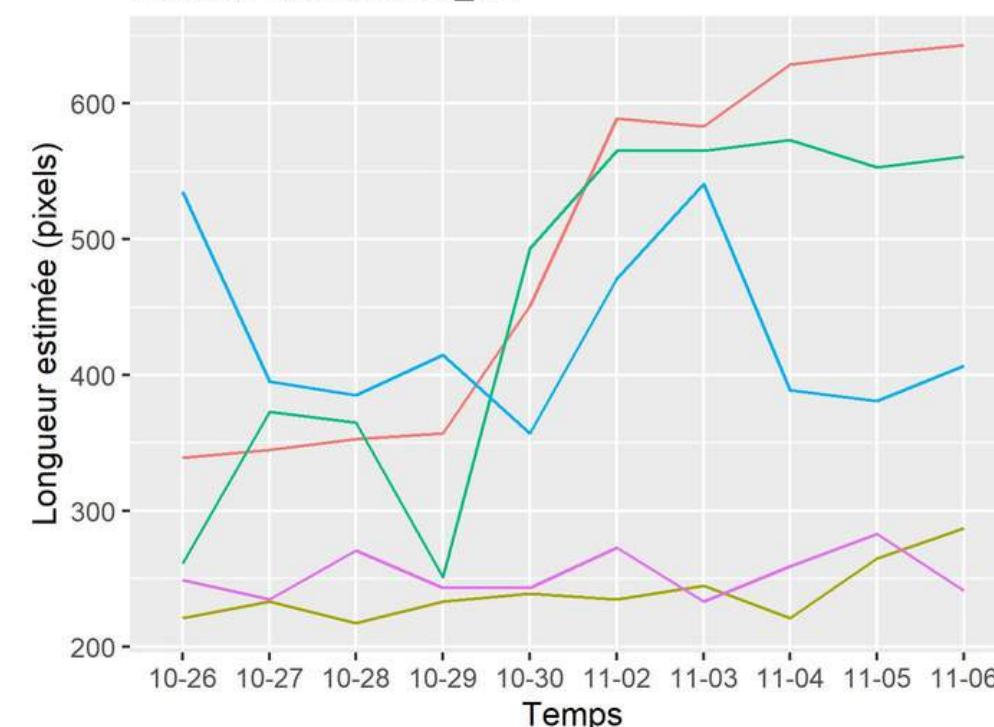


Longueur (méthode diff)

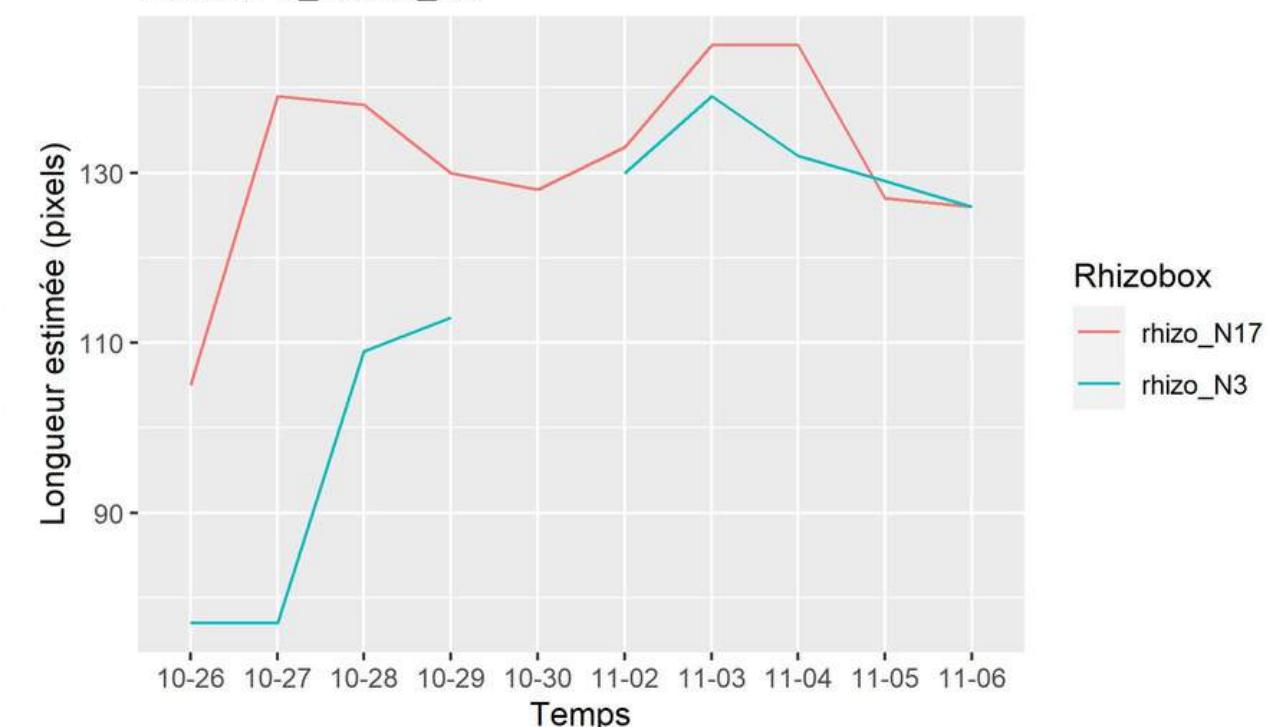
Evolution de la longueur des racines au cours du temps
Méthode U_net_diff



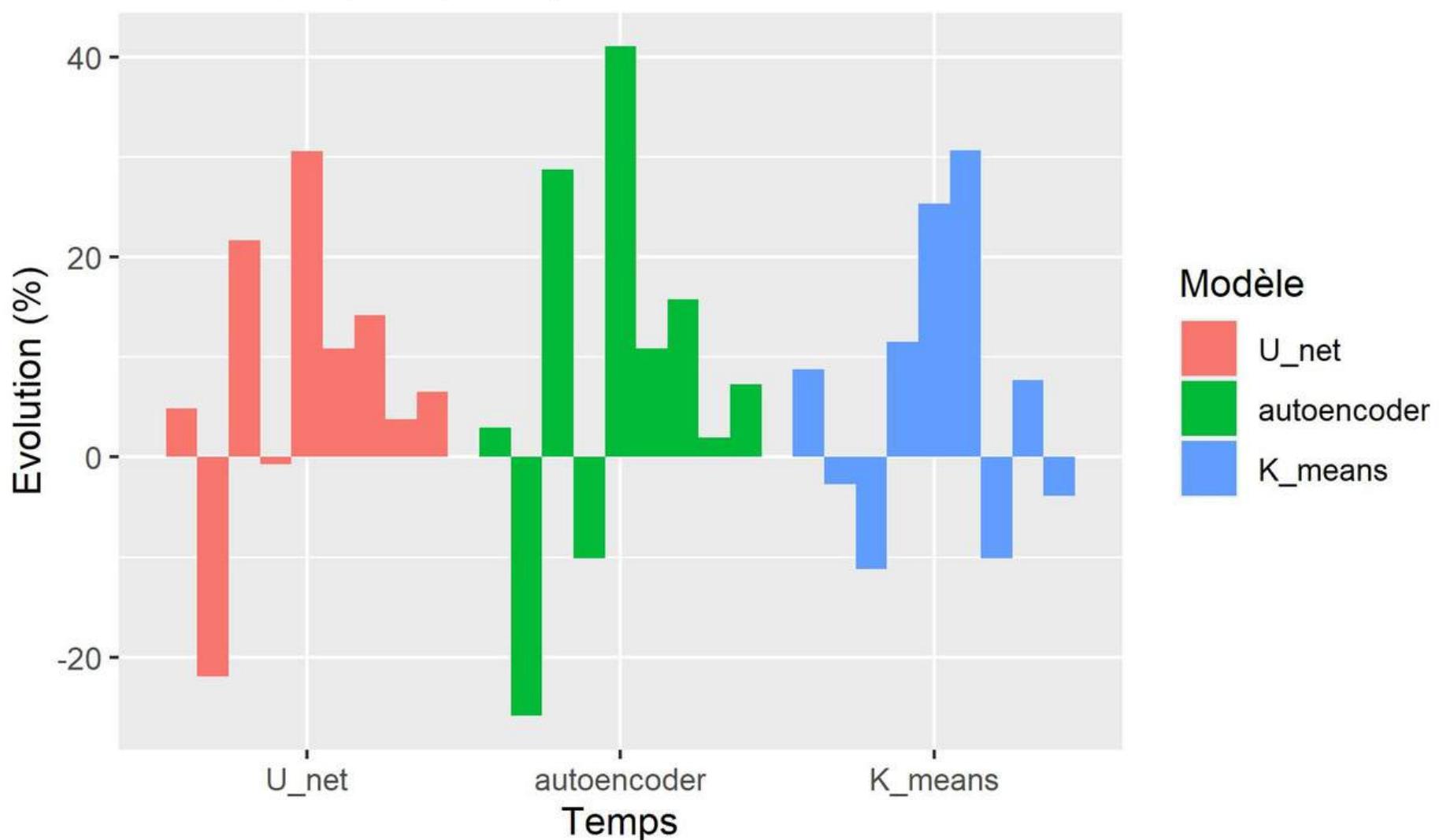
Evolution de la longueur des racines au cours du temps
Méthode autoencoder_diff



Evolution de la longueur des racines au cours du temps
Méthode K_means_diff

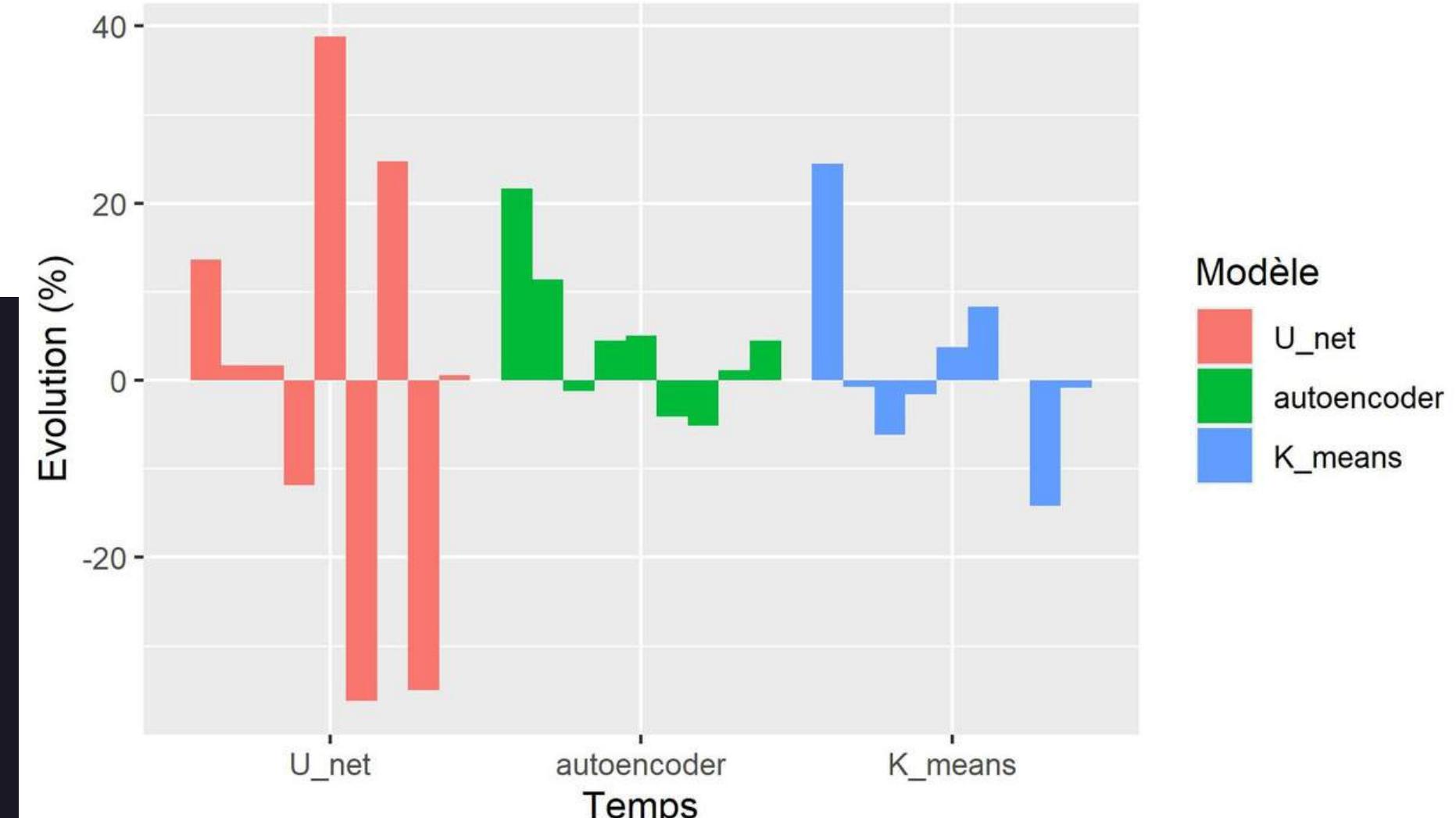


Evolution de la longueur des racines d'un jour à l'autre Méthode pix (rhizo_N17)



Pourcentage d'évolution jour après jour

Evolution de la longueur des racines d'un jour à l'autre Méthode diff (rhizo_N17)



Des anomalies sont observées à cause
de la repasse du jeu de données
(épaisseur du trait, etc.)