

Rapport : TP3 Cell analysis: segmentation and tracking

LABRI Ahlem

November 26, 2025

Mon dépôt GitHub est disponible ici : [Mon repo.](#)

1 Analyse des cellules HeLa

1.1 Segmentation des cellules

La segmentation a été réalisée sur le canal bleu de l'image *HeLa_cells.tif*. Les outils d'ICY utilisés sont les suivants :

- **Best Threshold (Otsu)** : permet de calculer automatiquement le seuil optimal et de générer un masque binaire des noyaux.

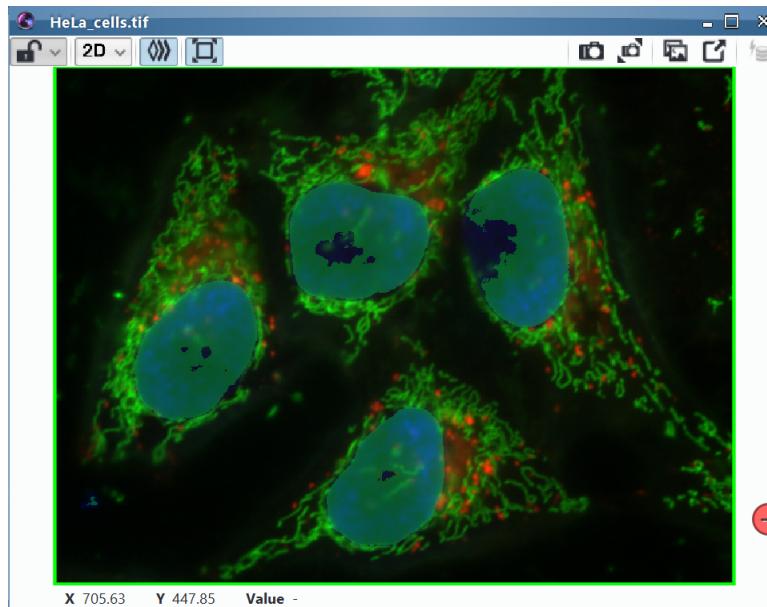


Figure 1: Segmentation des cellules après seuillage Otsu

- **Separate Components** : utilisé pour séparer chaque cellule en composante distincte et générer une ROI par cellule.

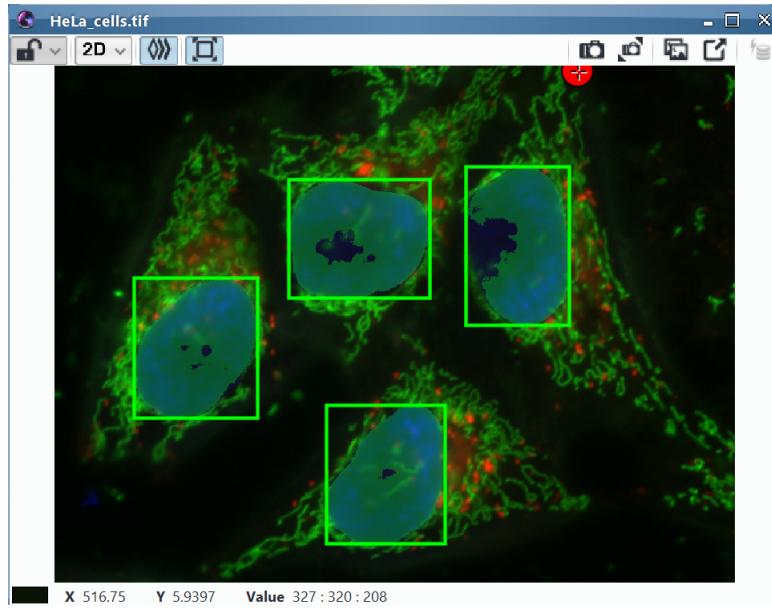


Figure 2: Séparation des composantes

1.2 Détection des spots

La détection des structures ponctuelles fluorescentes (canal rouge) a été réalisée à l'aide de l'outil **Spot Detector**. Les paramètres et modules utilisés dans ICY sont les suivants :

- **Spot Detection (Bright over Dark)** : détection des spots en considérant les points lumineux sur fond sombre.
- **Export to ROI** : conversion automatique des spots détectés en ROIs individuelles.
- **Export to Swimming Pool** : exportation des spots détectés dans le *Swimming Pool* pour une gestion centralisée et un comptage facilité.

La détection a été effectuée uniquement à l'intérieur des ROIs associées à chaque cellule afin de garantir un comptage précis.

Les résultats obtenus pour chaque cellule sont présentés dans le tableau suivant :

Cellule	Nombre de spots
Cellule 1	1
Cellule 2	5
Cellule 3	5
Cellule 4	11

Table 1: Nombre de spots détectés dans chaque cellule.

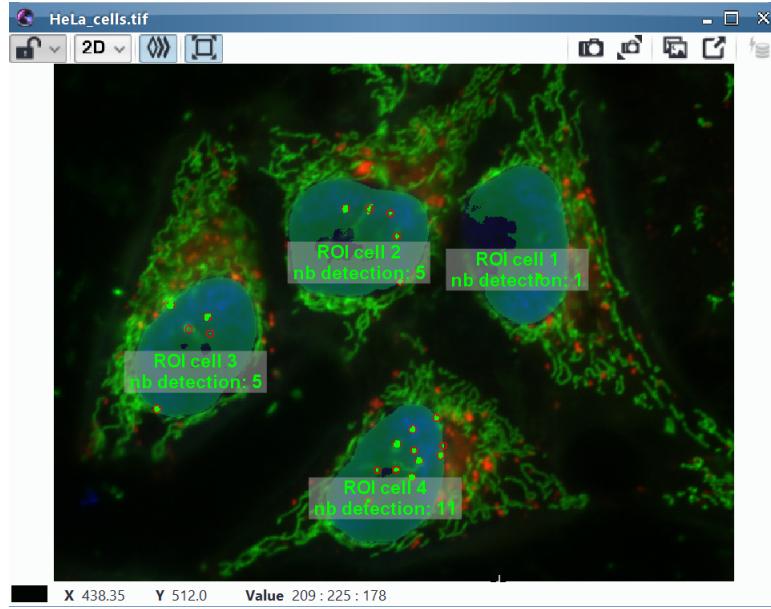


Figure 3: Spots détectés + comptage de spots dans les cellules

2 Segmentation et suivi automatique des cellules par CNN

Dans cette partie, la segmentation et le suivi temporel des cellules sont réalisés automatiquement à partir d'une séquence de la base *Fluo-N2DL-HeLa* du *Cell Tracking Challenge*. L'objectif est de segmenter chaque noyau sur chaque image puis d'associer ces noyaux au cours du temps afin d'obtenir des trajectoires de cellules.

2.1 Jeu de données et prétraitement

Nous utilisons les séquences 01 et 02 du jeu de données *Fluo-N2DL-HeLa* pour l'apprentissage, et une séquence de test pour l'évaluation. Chaque séquence est composée d'images 2D en niveaux de gris de noyaux de cellules HeLa acquises par fluorescence.

La structure des dossiers est la suivante :

- **train/01/** et **train/02/** : images brutes (**t000.tif**, **t001.tif**, ...).
- **train/01_GT/SEG/** et **train/02_GT/SEG/** : segmentations manuelles (**man_seg012.tif**, ...).
- **test/01/**, **test/02/** : séquences de test sur lesquelles on applique le modèle appris.

Les paires image/masque sont construites en associant chaque masque manuel **man_segXXX.tif** avec l'image correspondante **tXXX.tif**. Les prétraitements appliqués sont :

- redimensionnement des images et masques en 256×256 pixels ;
- conversion en niveaux de gris si nécessaire ;
- normalisation de l'intensité : soustraction de la moyenne et division par l'écart-type de chaque image ;
- conversion des masques d'instances en masque binaire : tous les labels > 0 sont considérés comme « cellule ».

2.2 Segmentation par réseau de neurones convolutionnel

La segmentation est réalisée avec un réseau de type U-Net implémenté en PyTorch. L'architecture utilisée est la suivante :

- entrée : image 1 canal de taille 256×256 ;
- chemin contractant : 4 blocs **DoubleConv** (deux convolutions 3×3 + BatchNorm + ReLU) avec nombres de canaux 32, 64, 128, 256, séparés par des max-poolings 2×2 ;
- goulot : bloc **DoubleConv** à 512 canaux ;
- chemin expansif : remontée symétrique par transposed-convolutions (2×2) et concaténation des skip connections correspondantes, suivies de blocs **DoubleConv** ;
- couche de sortie : convolution 1×1 produisant une carte de logits à un seul canal, transformée en probabilité par une sigmoïde.

Paramètres essentiels de la segmentation

Les paramètres d'apprentissage et de post-traitement les plus importants sont résumés ci-dessous :

Paramètres d'apprentissage.

- **Taille des patchs** : 256×256 (compromis entre contexte spatial et temps de calcul).
- **Batch size** : 8.
- **Taux d'apprentissage** : 10^{-3} , optimiseur Adam avec un *weight decay* de 10^{-5} .
- **Nombre d'époques** : 40.
- **Fonction de coût** : combinaison *Binary Cross-Entropy* + *Dice loss* :

$$\mathcal{L} = \alpha \text{BCE}(\hat{y}, y) + (1 - \alpha) (1 - \text{Dice}(\hat{y}, y)),$$

avec $\alpha = 0,5$.

Ce choix de perte est important car le jeu de données est fortement déséquilibré : les pixels de fond sont largement majoritaires. La composante Dice permet de mieux pénaliser les erreurs sur les petites régions positives (noyaux).

Paramètres de post-traitement.

- **Seuil de probabilité** : $\tau = 0,1$. Après apprentissage, les probabilités prédites pour les noyaux restent parfois inférieures à 0,5. Un seuil plus faible (0,1) permet de récupérer la plupart des noyaux tout en limitant les faux positifs, comme vérifié visuellement sur plusieurs images de test.
- **Étiquetage de composantes** : à partir du masque binaire, un étiquetage de composantes connexes (`cv2.connectedComponents`) est appliqué pour obtenir un masque d'instances où chaque cellule reçoit un identifiant entier distinct.

Ces paramètres (en particulier le seuil τ et la combinaison BCE + Dice) ont un impact direct sur la qualité de la segmentation et ont été ajustés de manière empirique à partir des visualisations.

2.3 Algorithme de suivi des cellules

À partir de la séquence de masques d'instances, le suivi temporel est réalisé image par image. Chaque cellule est représentée par son centroïde et son masque.

Entre deux frames consécutives, le problème est formulé comme une assignation bipartite résolue par l'algorithme hongrois :

1. extraction des centroïdes des IDs de track à la frame t et des labels de la frame $t + 1$;
2. calcul, pour chaque paire possible (i, j) , d'un coût combinant distance et recouvrement :

$$\text{cost}(i, j) = w_d \frac{d(i, j)}{d_{\max}} + w_{\text{iou}} (1 - \text{IoU}(i, j)),$$

où $d(i, j)$ est la distance euclidienne entre centroïdes, IoU l'intersection-sur-union des masques, et d_{\max} une distance de normalisation ;

3. l'algorithme hongrois fournit l'assignation de coût minimal ;
4. si le coût minimal pour une paire dépasse un seuil, la cellule de la frame $t + 1$ est considérée comme une nouvelle trajectoire.

Les paramètres de tracking utilisés sont :

- distance de normalisation $d_{\max} = 20$ pixels ;
- poids de la distance $w_d = 0,5$;
- poids de l'IoU $w_{\text{iou}} = 0,5$;
- coût maximal acceptable pour une correspondance : $\text{max_cost} = 0,8$.

Ces valeurs donnent un compromis raisonnable entre continuité des trajectoires et capacité à détecter l'apparition/disparition de cellules.

2.4 Illustration des résultats

Les résultats sont illustrés par deux types de figures.

Segmentation sur une image de test. La Figure 4 montre, pour une image de la séquence de test :

- l'image originale en niveaux de gris ;
- la carte de probabilité produite par le CNN ;
- le masque binaire obtenu après seuillage ($\tau = 0,1$) ;
- l'overlay image + masque binaire.

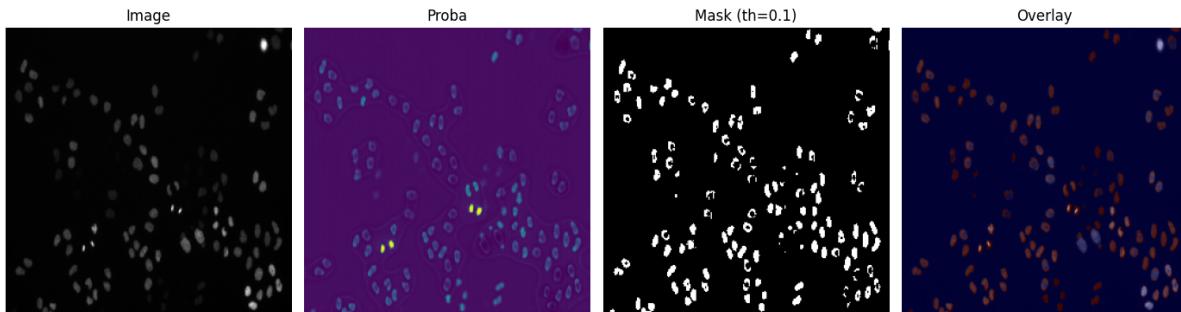


Figure 4: Exemple de segmentation par CNN sur une image de la séquence Fluo-N2DL-HeLa : image originale, probabilité par pixel, masque binaire et overlay.

Suivi de cellules au cours du temps. La Figure 5 présente des instantanés du suivi pour plusieurs frames (par exemple 0, 20, 40, 60, 80 et 91). Chaque cellule est représentée par une couleur, et les positions successives sont reliées pour visualiser sa trajectoire.

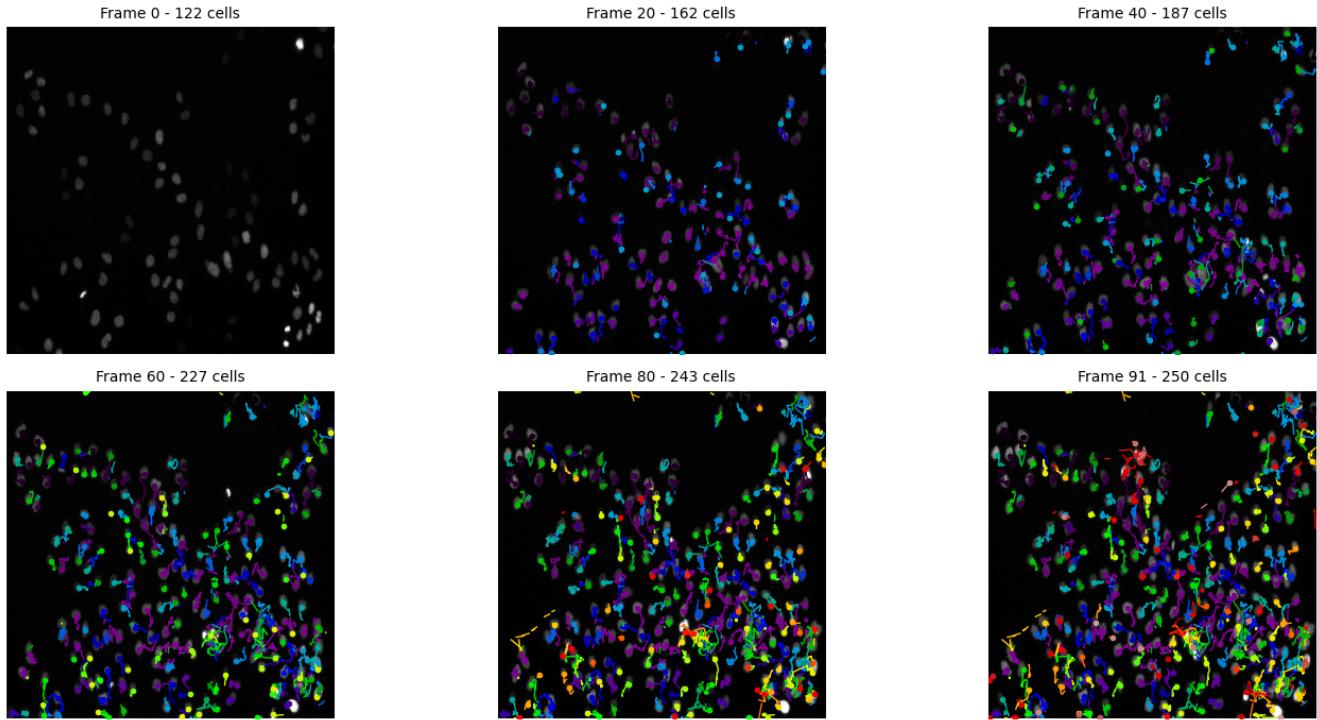


Figure 5: Exemples de suivi des cellules : snapshots de différentes frames avec trajectoires colorées.

Enfin, la Figure 6 synthétise les trajectoires des 20 cellules ayant la plus longue durée de vie sur l'ensemble de la séquence (92 images). Le fond correspond à une image moyenne de la séquence, et chaque trajectoire est codée avec une couleur distincte.

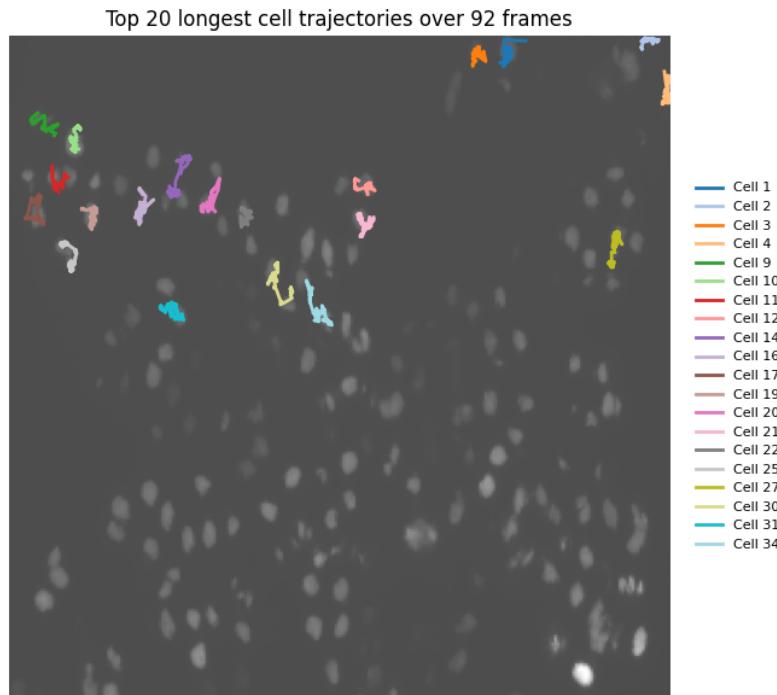


Figure 6: Trajectoires des 20 cellules les plus longues sur la séquence complète.

Ces résultats montrent que le CNN permet de segmenter correctement la majorité des noyaux de la séquence et que la combinaison distance + IoU dans l'algorithme de tracking permet de suivre efficacement les cellules au cours du temps.