

# Segmentation & Tracking de cellules HeLa sous ICY et Python (Cellpose & TrackPy)

Wassim Chikhi

Master 2 Vision et Machine Intelligente – 2025/2026

## 1. Objectifs du TP

Ce TP vise à construire deux pipelines d'analyse d'images cellulaires complémentaires :

- **Question 1** : segmenter des cellules HeLa et détecter des *spots* fluorescents sous ICY, puis compter les spots par cellule via des ROIs.
- **Question 2** : segmenter automatiquement des cellules dans une séquence 2D+temps (Cell Tracking Challenge) à l'aide de Cellpose, puis suivre leurs trajectoires avec TrackPy.

L'objectif final est d'obtenir un pipeline reproductible, combinant méthodes interactives (ICY) et deep learning (Python).

## 2. Données et outils

**Données ICY (Q1)** : image multicanal HeLa\_cells.tif.

**Données Python (Q2)** : séquence DIC DIC-C2DH-HeLa (série 01 du Cell Tracking Challenge).

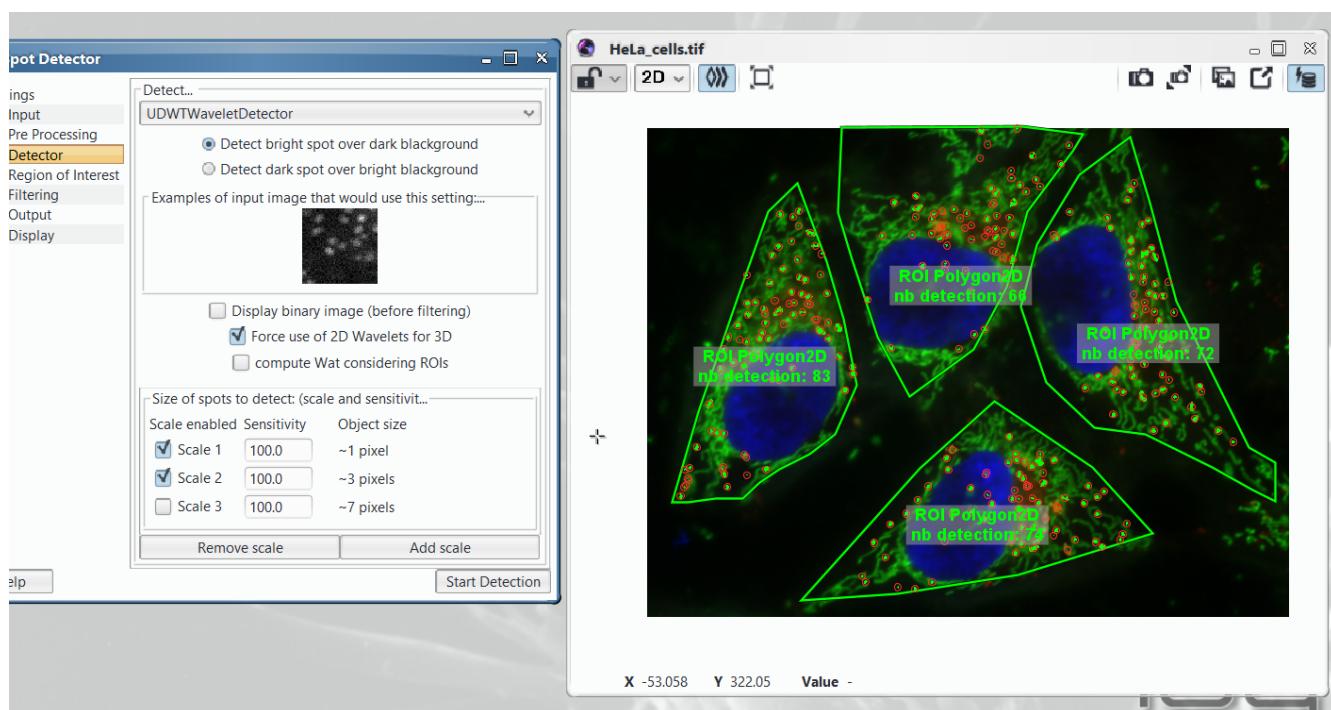
**Outils** : ICY — Cellpose v4 — TrackPy — Scikit-Image — Kaggle GPU.

## 3. Question 1 – Segmentation et détection de spots sous ICY

### 3.1. Pipeline ICY

Le pipeline comprend :

- filtrage médian (réduction du bruit),
- seuillage automatique (Otsu / *Best Threshold*),
- opérations morphologiques (Fill Holes, Dilation/Erosion),
- détection de spots par ondelettes (*Spot Detector*),
- comptage des spots par cellule via ROI (analyse d'inclusion / comptage par ROI).

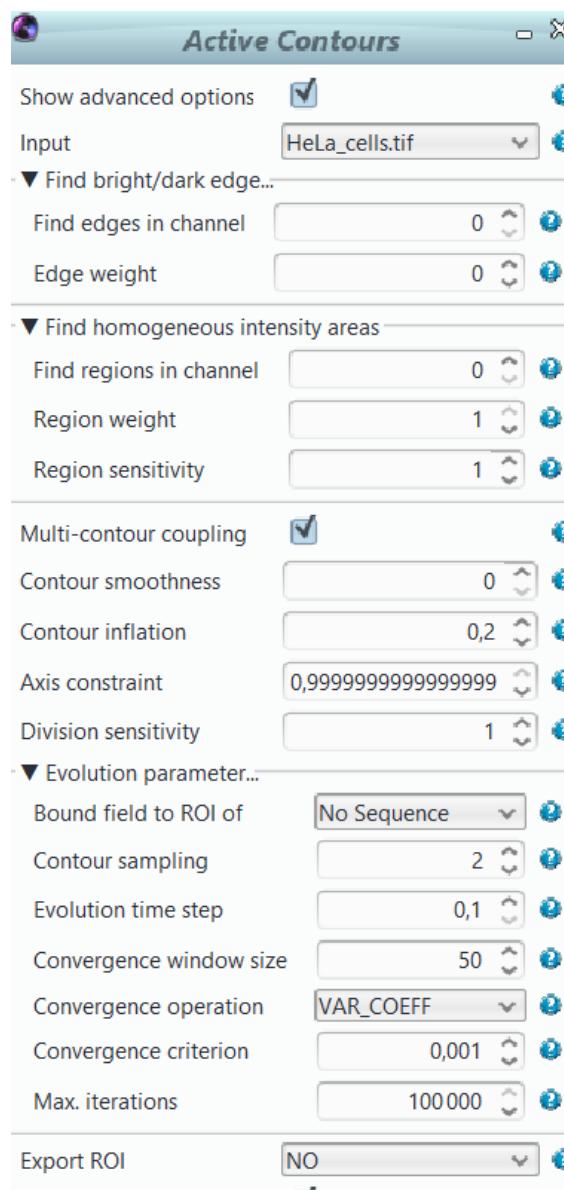


**FIGURE 1.** Q1 — ROIs cellulaires (vert) et spots détectés (rouge) sous ICY sur HeLa\_cells.tif.

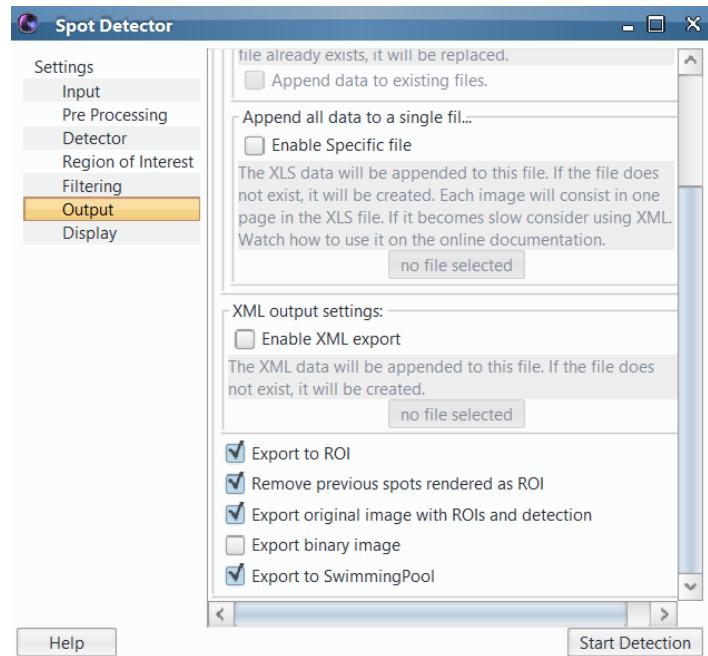
### 3.2. Paramétrage ICY (réglages clés et reproductibilité)

Les paramètres ICY ont été ajustés empiriquement afin (i) d'obtenir des contours cellulaires fermés et (ii) de limiter les faux positifs lors de la détection des spots. Les captures des réglages finaux sont fournies en Figures 2 et 3.

- **Prétraitement** : filtre médian (réduction du bruit impulsionnel) pour stabiliser la segmentation et la détection.
- **Segmentation cellules** : seuillage (Otsu/*Best Threshold*) suivi d'opérations morphologiques (Fill Holes, Dilation/Erosion) pour obtenir des régions cellulaires denses et fermées (ou, selon les essais, contours actifs initialisés par ROI).
- **Détection des spots** : détecteur par ondelettes (*Spot Detector*), avec ajustement du *threshold* afin d'obtenir une détection centrée sur les spots et de limiter le bruit.
- **Comptage par cellule** : association des spots aux ROIs cellulaires (inclusion) afin d'obtenir le nombre de spots par cellule.



**FIGURE 2.** Réglages ICY utilisés pour la segmentation des cellules (prétraitements + seuillage + morphologie / ou Active Contours si utilisé).



**FIGURE 3.** Réglages ICY du détecteur de spots (ondelettes / seuil).

### 3.3. Résultats quantitatifs : spots par cellule (moyenne & écart-type)

Le nombre de spots détectés dans quatre cellules est :

Cellule	Nombre de spots
1	83
2	69
3	72
4	74

**TABLE 1.** Comptage des spots par cellule (Q1).

À partir de ces mesures (sur  $n = 4$  cellules), la statistique globale est :

$$\mu = 74.5 \text{ spots/cellule}, \quad \sigma = 6.03 \text{ spots/cellule}$$

où  $\mu$  est la moyenne et  $\sigma$  l'écart-type (estimé sur échantillon).

### 3.4. Discussion (Q1)

On observe une variabilité modérée du nombre de spots selon la cellule (Table 1), cohérente avec des différences de marquage, d'épaisseur optique ou de bruit local. Le *threshold* du détecteur de spots est un paramètre déterminant : un seuil trop bas augmente les faux positifs, tandis qu'un seuil trop haut sous-estime le comptage. La reproductibilité est assurée en conservant les paramètres finaux (captures et/ou protocole ICY exporté).

## 4. Question 2 – Segmentation CNN & Tracking (Python)

### 4.1. Chargement et visualisation des données

La séquence DIC est chargée sous la forme d'un tenseur  $(T, H, W)$ . La Figure 4 présente une image brute représentative (frame 0), utilisée comme référence visuelle pour évaluer la qualité de segmentation et du tracking.

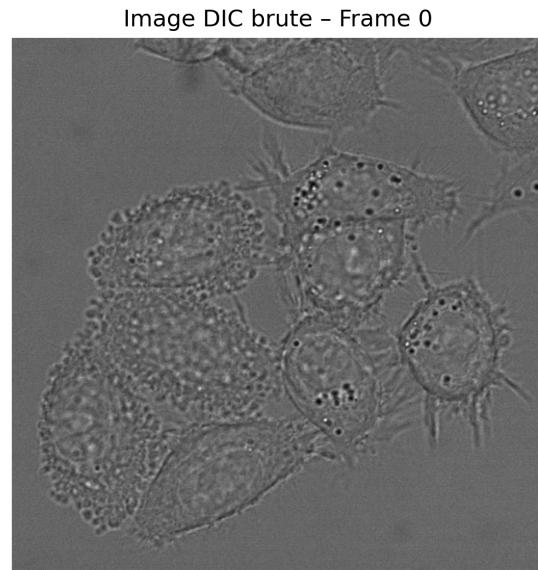


FIGURE 4. Q2 — Image DIC brute (frame 0) de la séquence DIC-C2DH-HeLa.

### 4.2. Segmentation CNN (Cellpose)

La séquence DIC est normalisée en  $[0, 1]$ . La segmentation est réalisée avec Cellpose (v4) sur GPU. Les paramètres utilisés sont :

- diameter=30,
- flow\_threshold=0.4,
- cellprob\_threshold=0.0.

La Figure 5 illustre les masques de segmentation produits sur la frame 0.

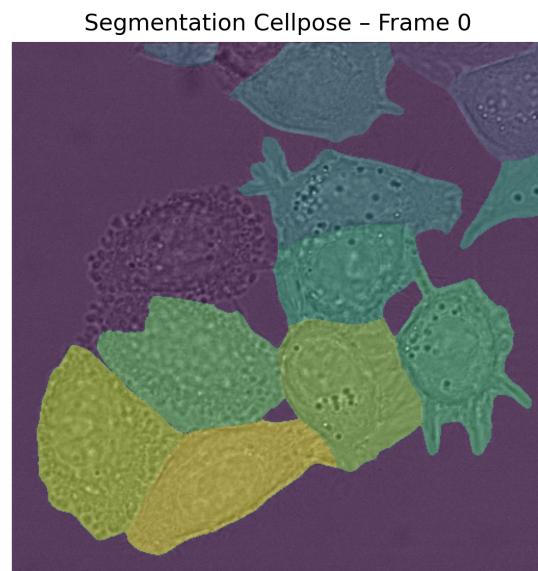
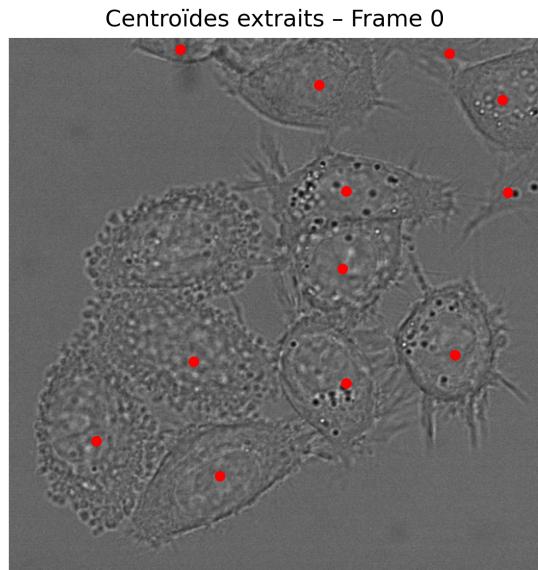


FIGURE 5. Q2 — Masques cellulaires obtenus par Cellpose (frame 0) superposés à l'image DIC.

#### 4.3. Extraction des centroïdes et suivi (TrackPy)

Les centroïdes des régions segmentées sont extraits à partir des masques à l'aide de `regionprops_table`. La Figure 6 montre les centroïdes obtenus sur la frame 0, utilisés ensuite comme observations d'entrée pour le tracking.



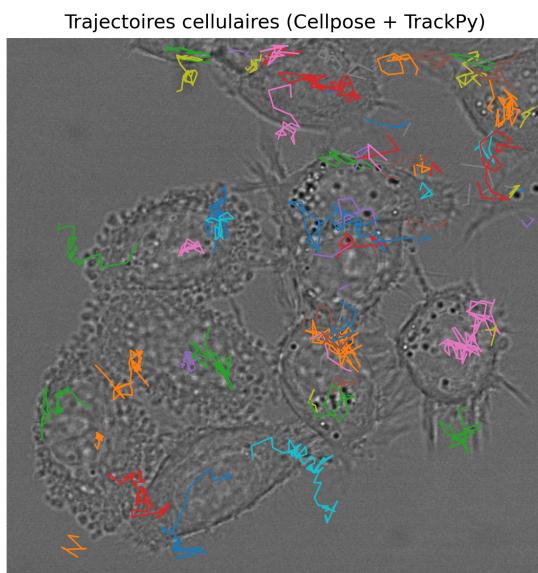
**FIGURE 6.** Q2 — Centroïdes extraits à partir des masques Cellpose (frame 0).

Le tracking est ensuite réalisé avec TrackPy (*Nearest Neighbour linking*) :

- `search_range=25 px`,
- `memory=3 frames`,

permettant de reconstruire des trajectoires même en présence de détections manquantes sur quelques frames.

#### 4.4. Visualisation des trajectoires



**FIGURE 7.** Q2 — Trajectoires Cellpose + TrackPy superposées à l'image DIC (frame 0 en fond).

Les mouvements observés sont lents et confinés, ce qui est cohérent avec des cellules HeLa adhérentes.

#### 4.5. Notebook Kaggle (reproductibilité)

L'intégralité du pipeline Python (chargement des données, segmentation Cellpose, extraction des centroïdes, tracking TrackPy et génération des figures) est disponible publiquement :

### Notebook Kaggle : TP3 – CellSegmentation & Tracking (Q2)

Ce notebook permet :

- d'exécuter le pipeline sur GPU,
- de visualiser la segmentation (Figure 5),
- d'inspecter les observations utilisées pour le tracking (Figure 6),
- de reproduire les trajectoires (Figure 7).

## 5. Conclusion

Ce TP montre la complémentarité entre :

- **ICY** : adapté à l'analyse interactive, au contrôle visuel et au comptage précis des spots par cellule,
- **Cellpose + TrackPy** : pipeline automatique scalable pour des séquences temporelles entières.

L'intégration des deux approches fournit un workflow robuste pour analyser la structure et la dynamique de cellules HeLa.