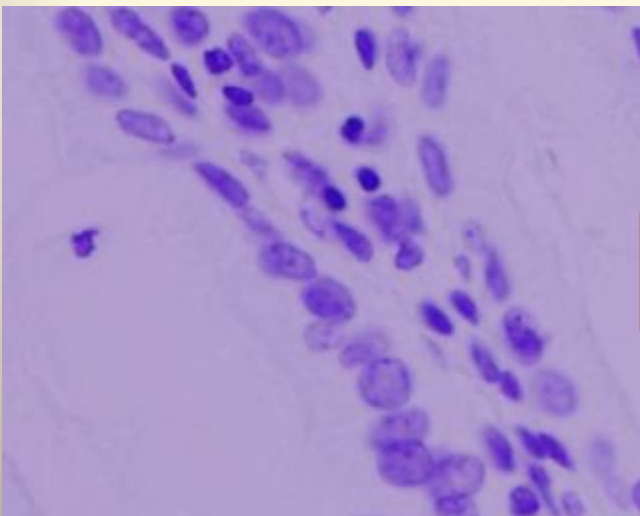


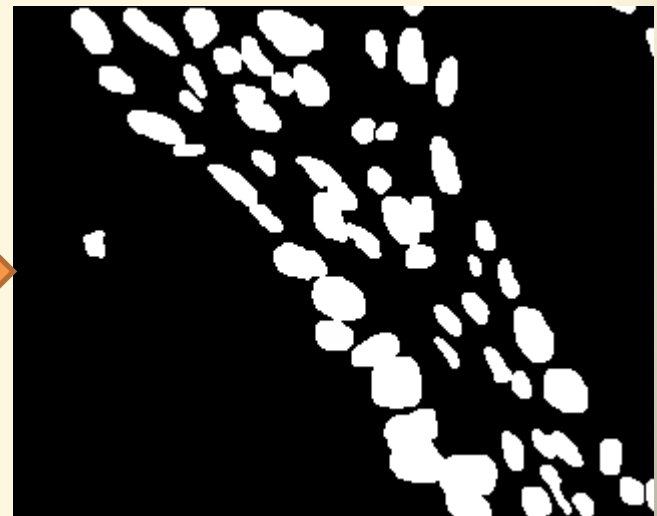


# Parcours Data Scientist

## Projet 9 : Compétition Kaggle

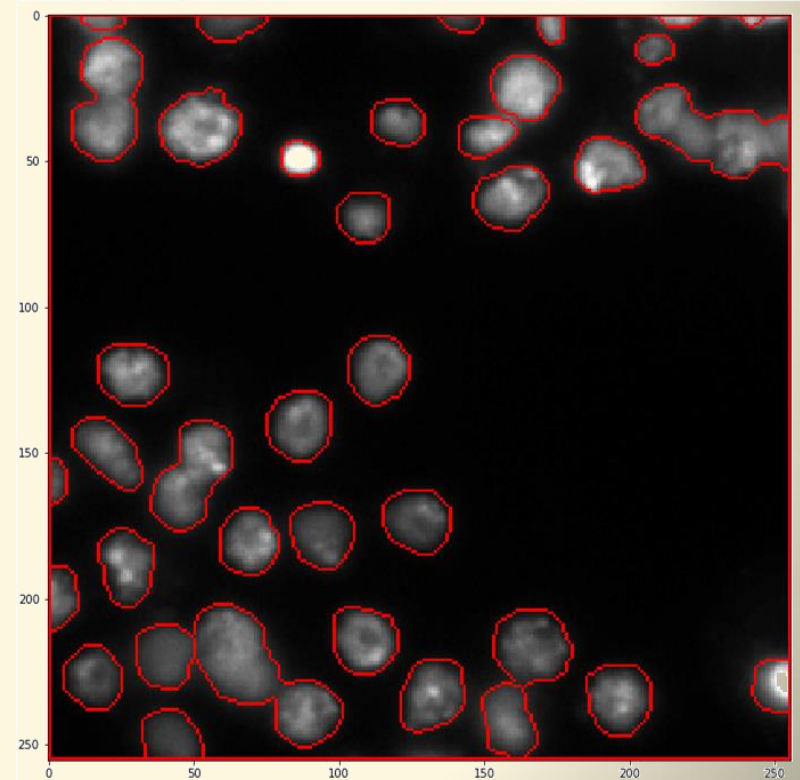


Prédiction de  
masque



# Sommaire

- Présentation de la compétition
- Exploration
- Pré-processing
- Modèles
  - Classique
  - U-net
  - U-net extended (multi arm)
- Post-Processing
- Résultats
- Améliorations
- Conclusion

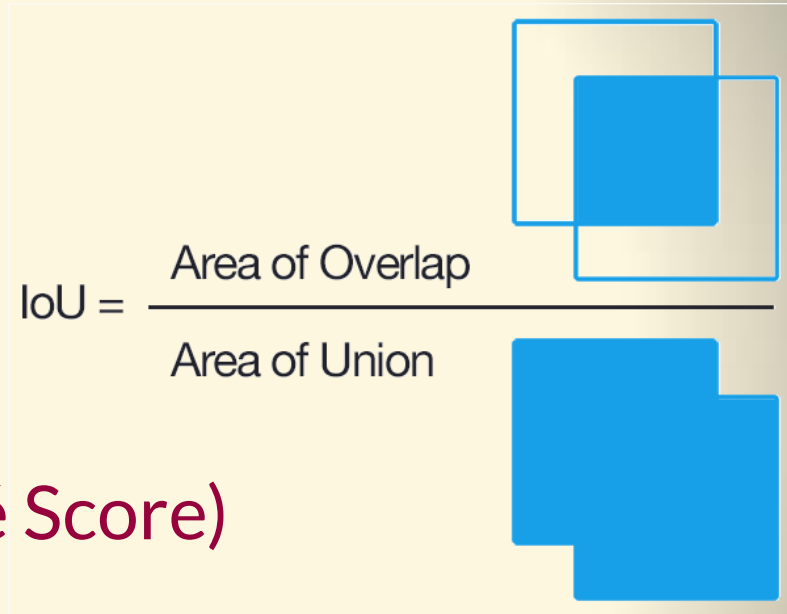


# Présentation

- Objectif
  - Détection de noyaux de cellules
  - Fournir 1 masque par cellule
  - N masques par images
- Dataset
  - 3 types d'images
  - 670 images d'entraînement
  - 65 images de test

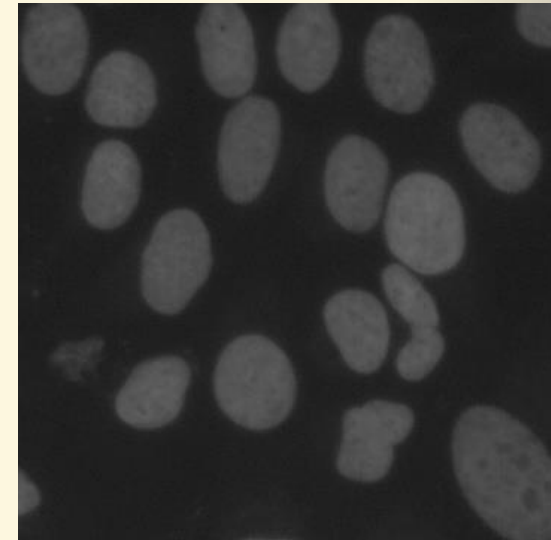
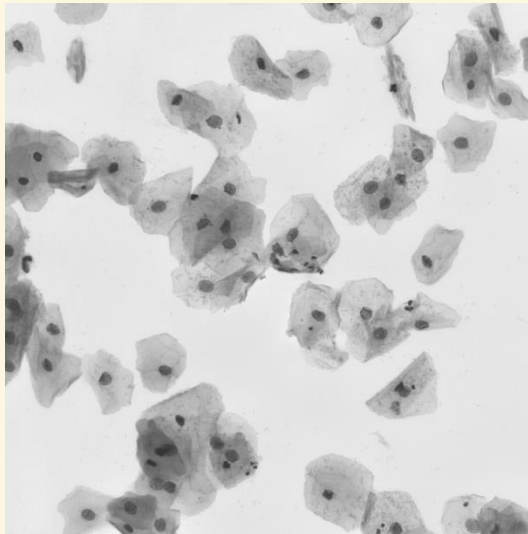
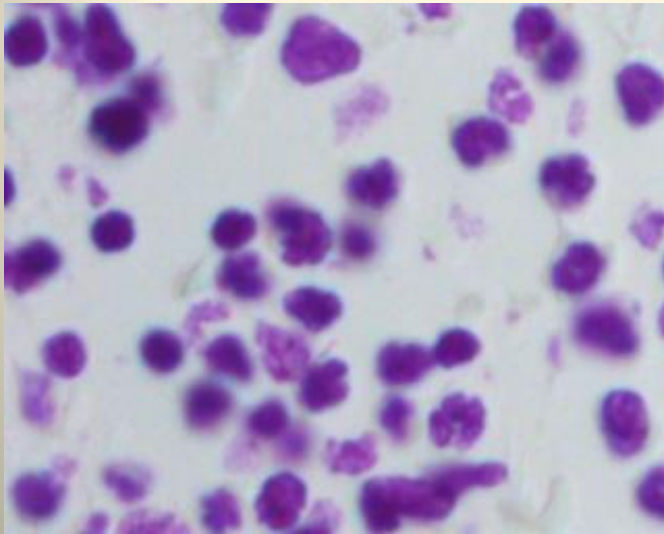
# Présentation

- Evaluation
  - 1 masque par cellule
  - 1 résultat de IoU/cellule
  - Moyenne du nombre de cellules  $\text{IoU} > T$  (appelé Score)
  - $T : 0,5 \Rightarrow 0,95$  (step 0,05)
  - Encodage Run Length Encoded
  - Overlap interdit


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

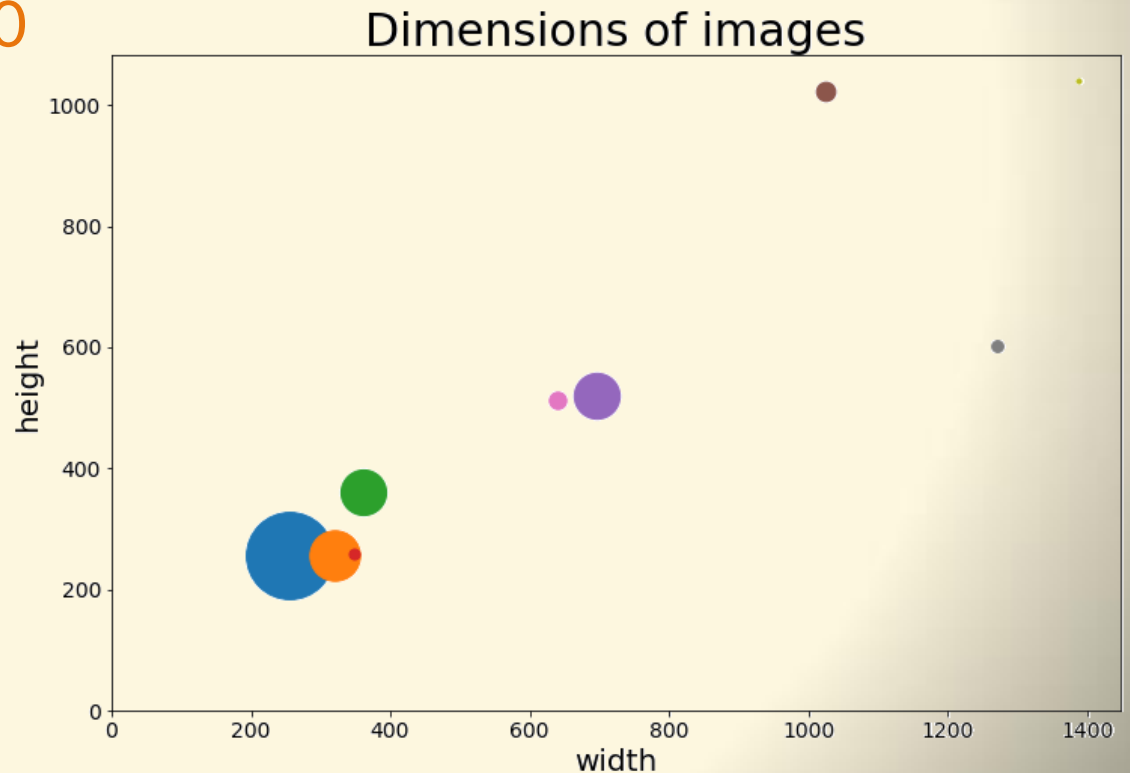
# Exploration

- Types d'images
  - 3 types selon la méthode d'acquisition



# Exploration

- Taille des images
  - Mini: 256x256
  - Maxi: 1388x1040
  - 80 % < 360x360
  - 97 % < 700x520

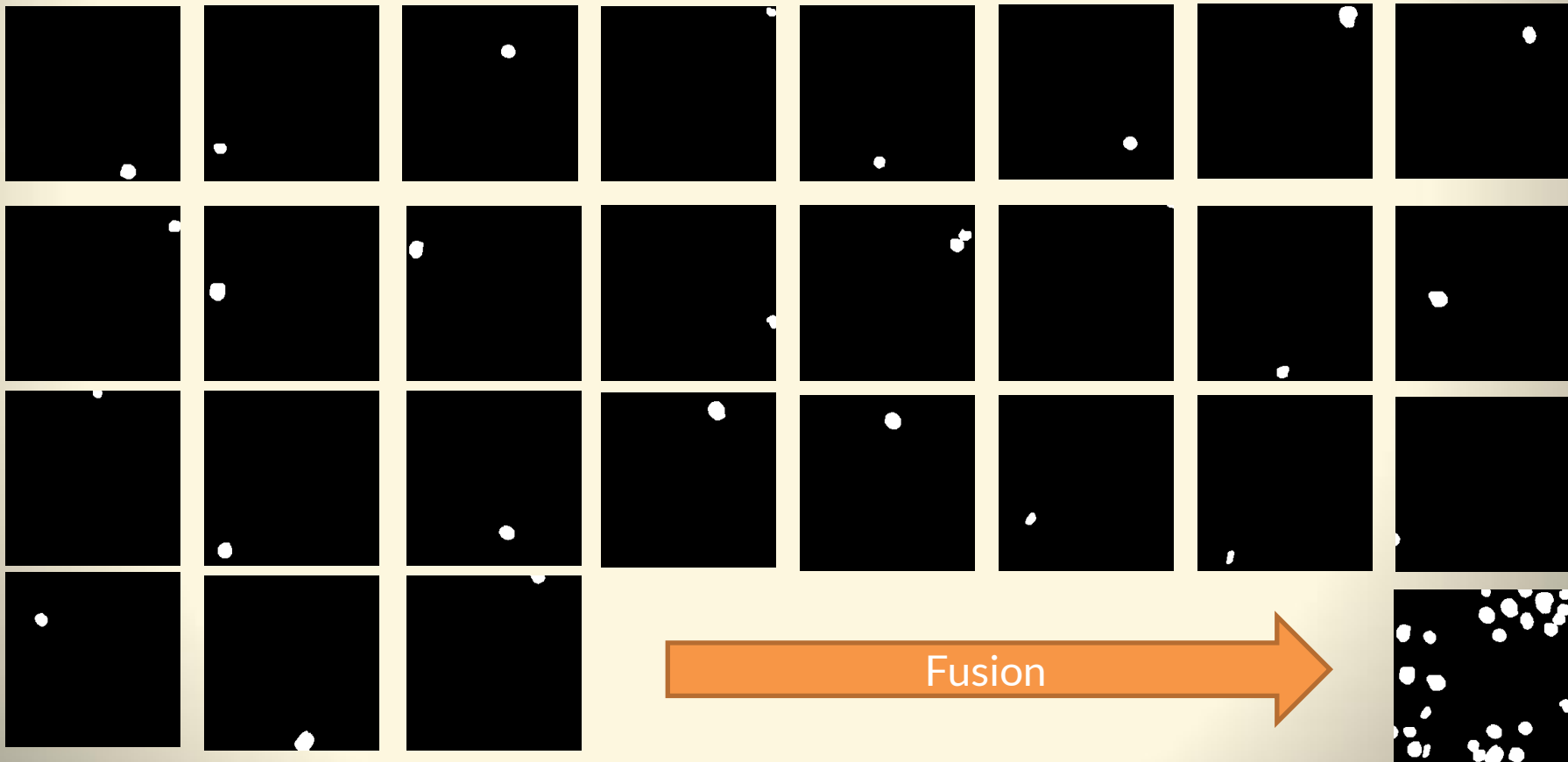


# Pré-processing

- Masques
  - Fusionnement
- Images
  - Plusieurs idées testées
    - Saturation
    - ~~Canny Edge Detection~~
    - ~~Laplacien~~
    - Adaptative Threshold
    - ~~Simple Binarisation~~
    - ~~Histogram Equalization~~
    - Redimensionnement

# Pré-processing

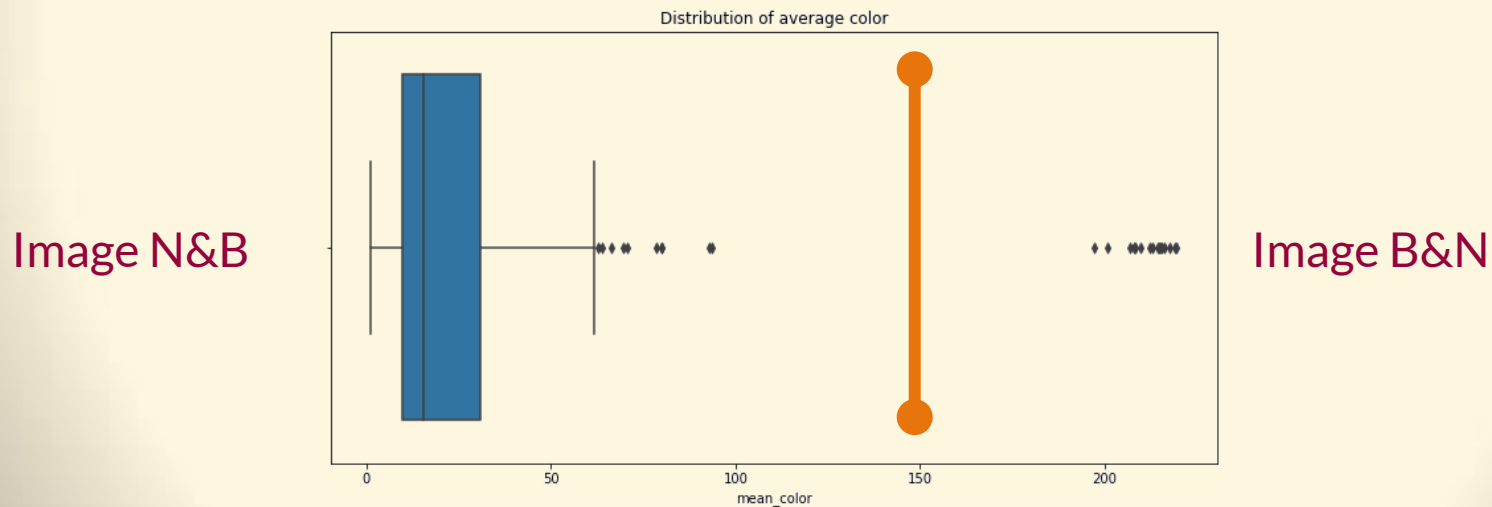
- Fusion des Masques





# Pré-processing

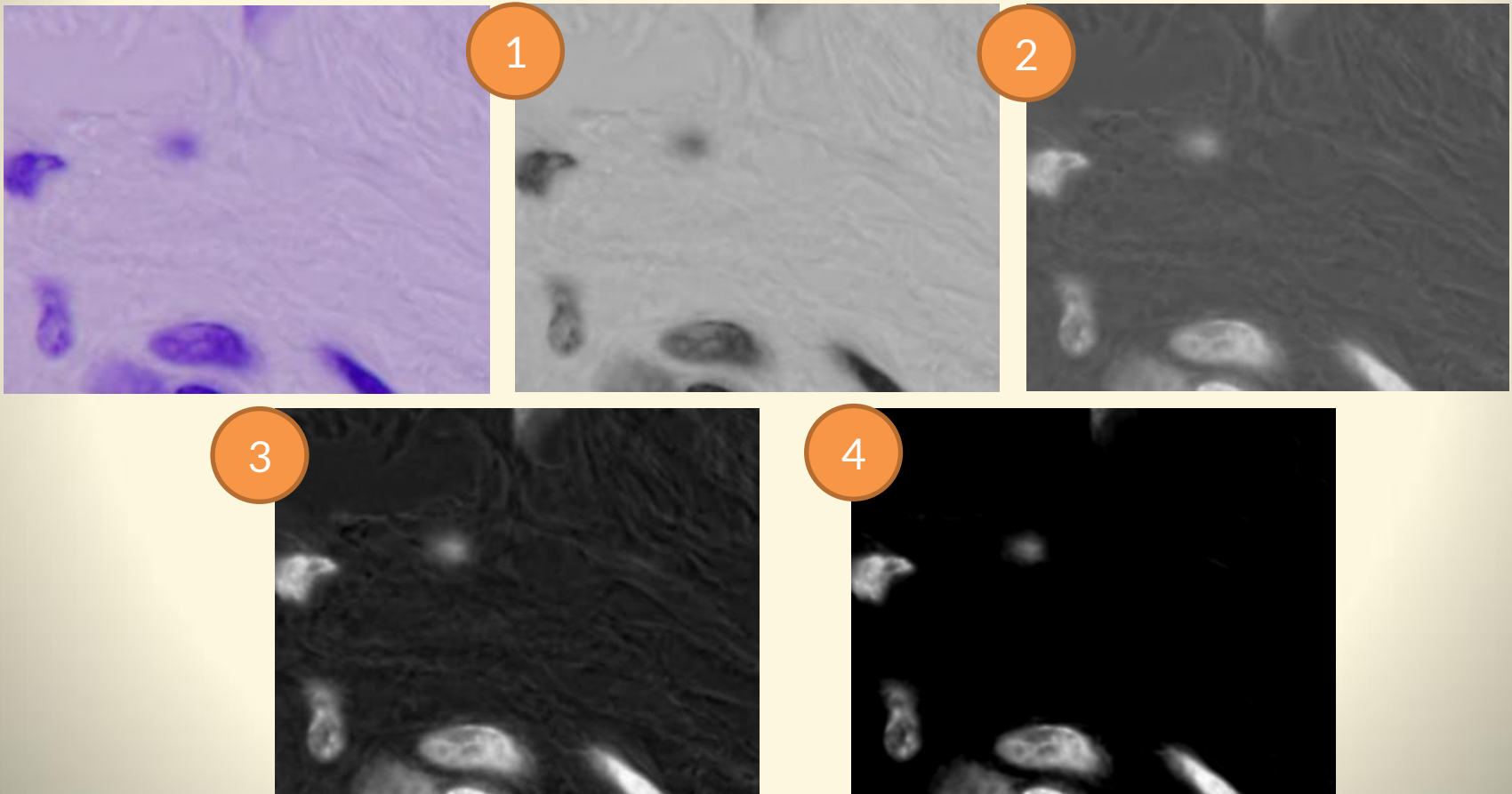
- Saturation
  1. Conversion en N&B
  2. Inverser N&B selon moyenne



## 3. Booster Contrastes

# Pré-processing

- Saturation



# Pré-processing

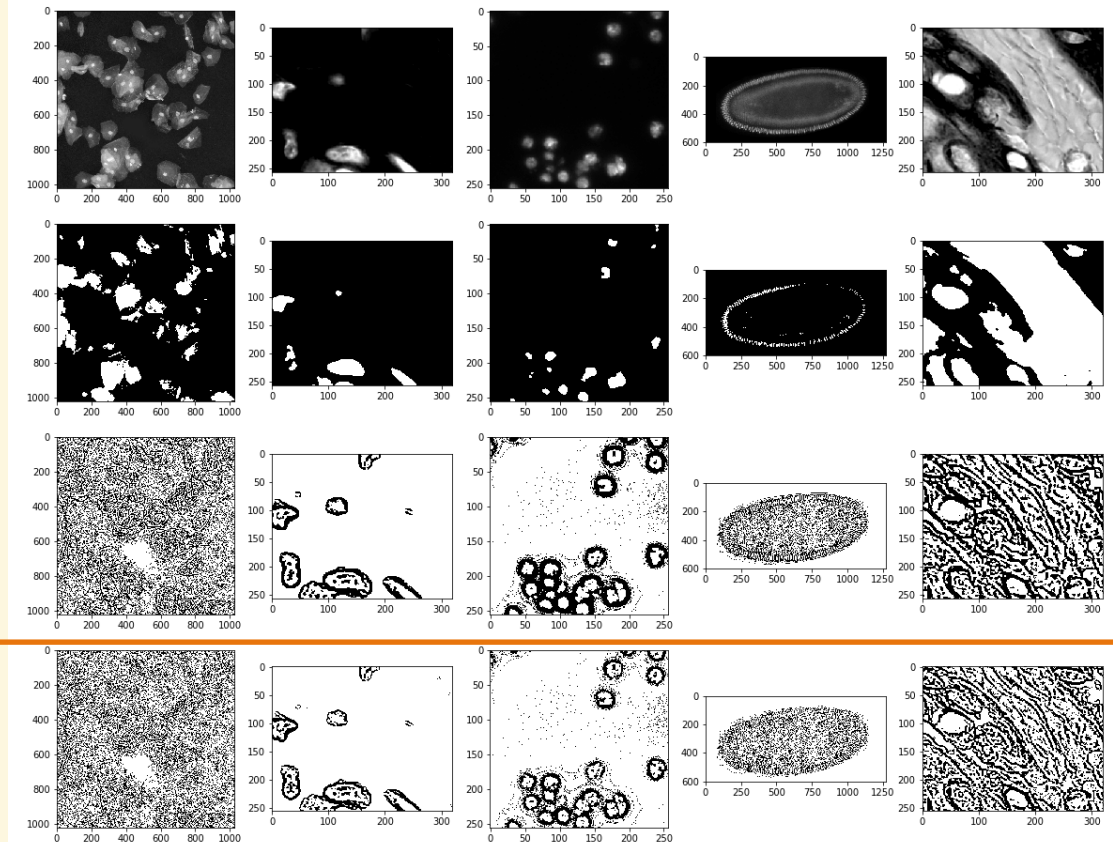
- Adaptative Threshold

Image de base

Seuil Simple

Seuil basé sur la moyenne des pixels voisins

Seuil basé sur la moyenne des pixels voisins avec un facteur basé sur la gaussienne



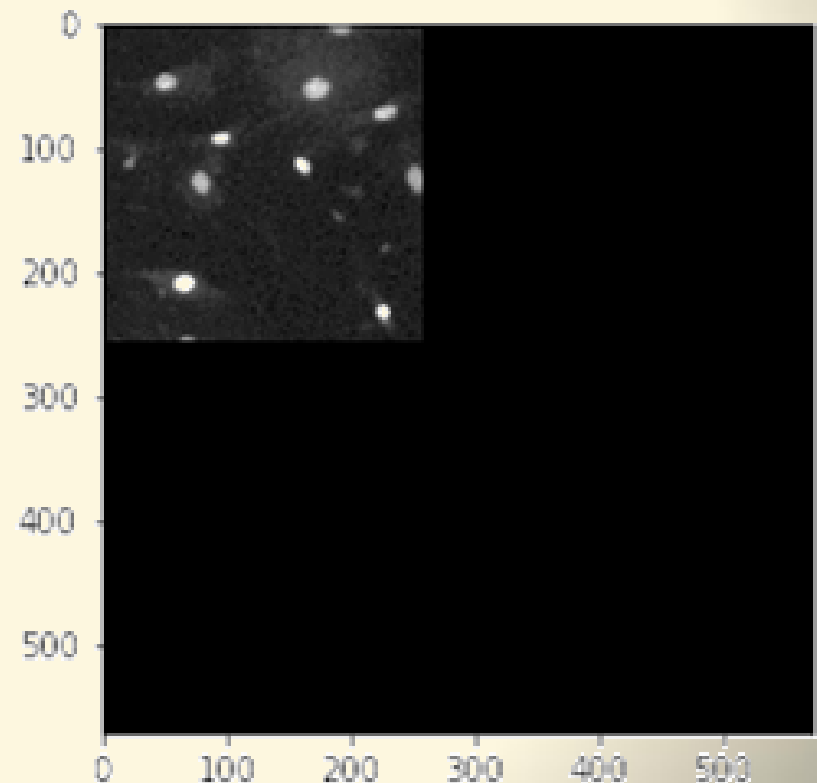
# Pré-processing

- Redimensionnement

- ✓ Classique

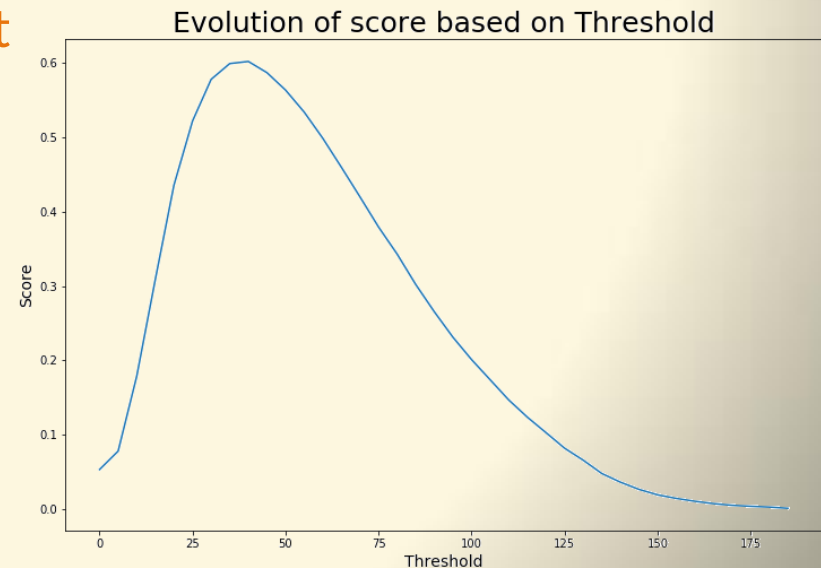
- Sans déformation

- Si  $img < objectif$ 
      - Complétion en noir
    - Si  $img > objectif$ 
      - Scaling constant
      - Complétion en noir



# Modèles

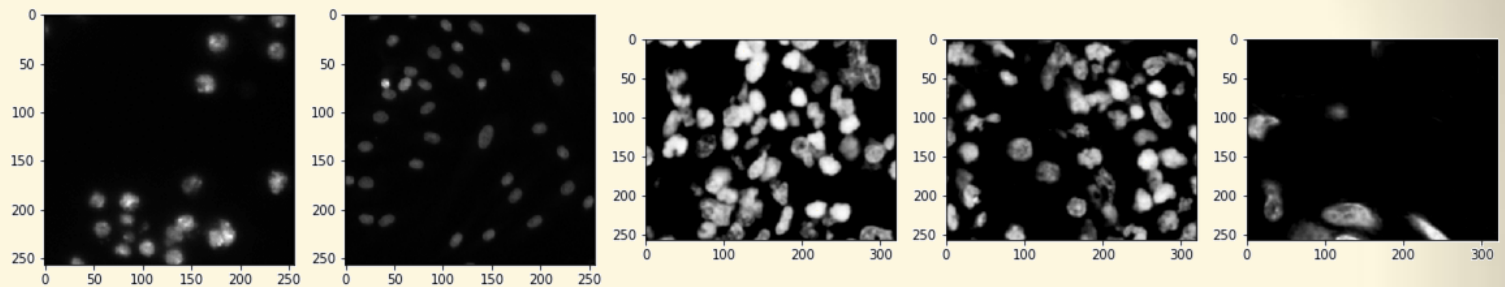
- Classique
  - Images pré-processées
  - Filtrage binaire
    - Seuil déterminé en fct du score sur le dataset
      - Fait sur le masque complet
    - Pas d'entraînement
  - Meilleur score :  $t=40$



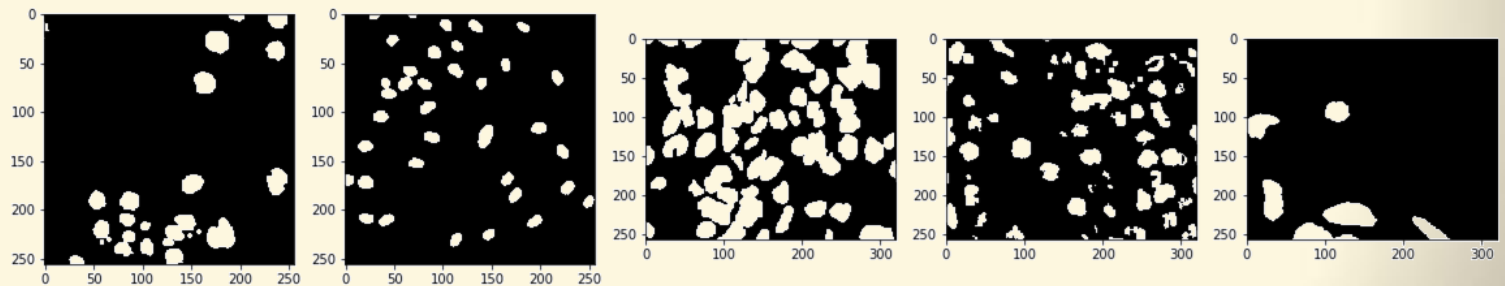
# Modèles

- Classique

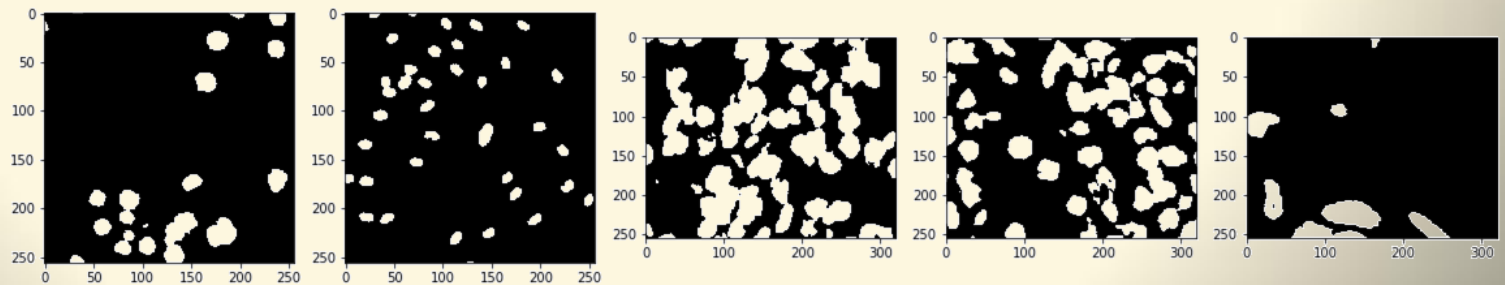
Image de base



Masque prédit

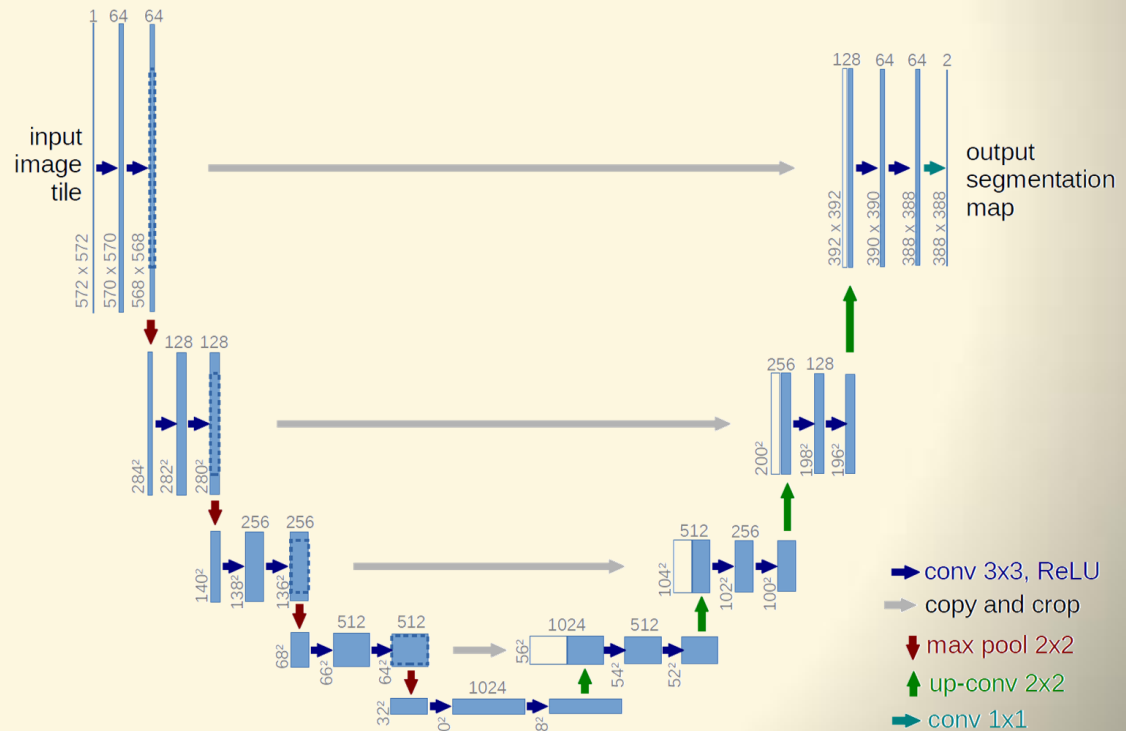


Vrai masque



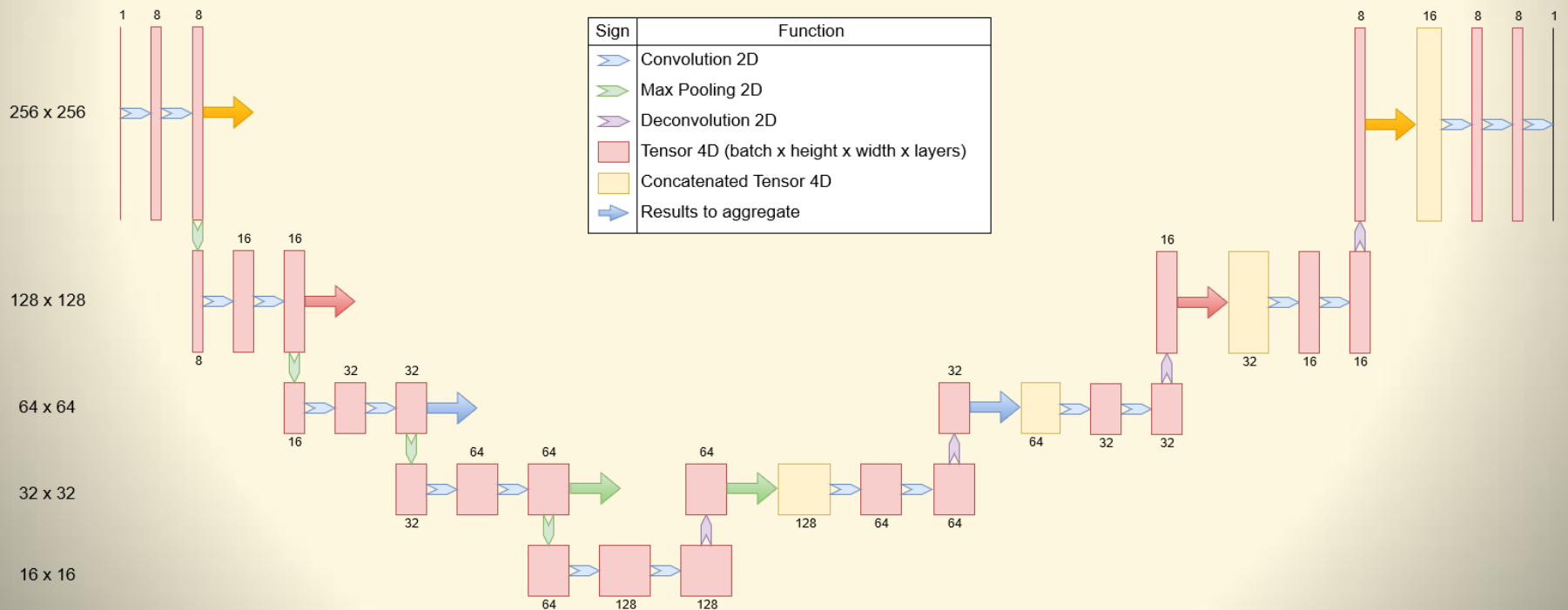
# Modèles

- CNN (version 1)
  - Basé sur U-net
  - Vainqueur
    - ISBI 2015
  - Entrée
    - $572 \times 572$
    - 94px bordure
  - Sortie
    - $388 \times 388$
  - Cause
    - Padding Valid
    - Image non  $2^n$



# Modèles

- CNN (version 1)
  - Entrée = Sortie
    - 256 x 256



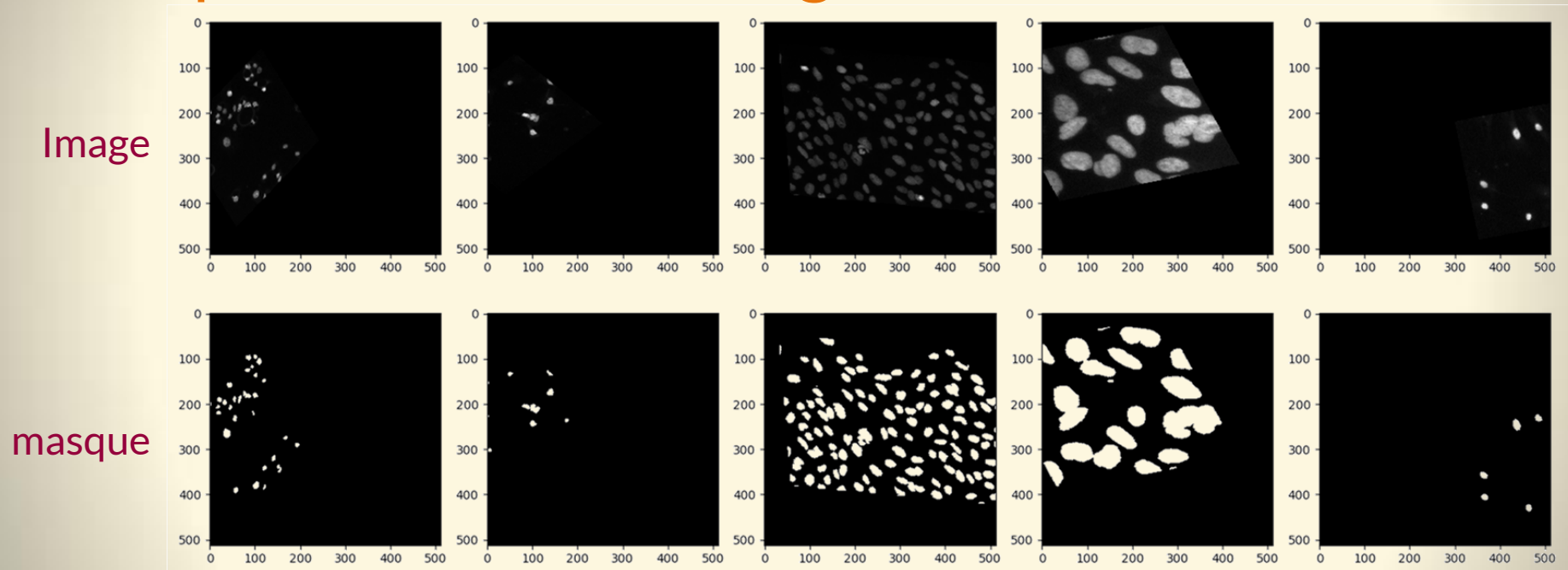


# Modèles

- CNN (version 1)
  - Test avec et sans data augmentation
    - Apprentissage très mauvais
  - Test avec images de base
    - Apprentissage correct mais cellules peu séparée
  - Test avec images pré-processées
    - Apprentissage correct mais cellules peu séparée
  - Enregistrement par Epoch de 4 test images

# Modèles

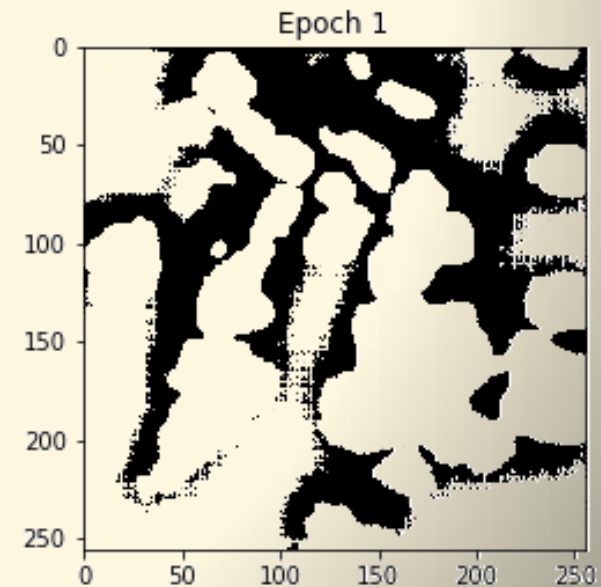
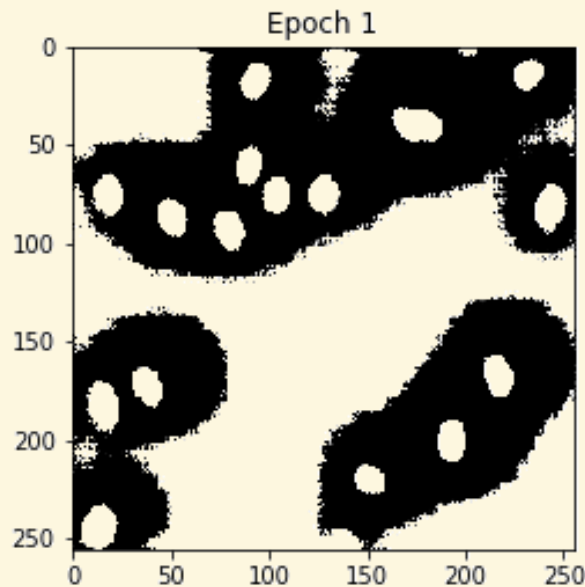
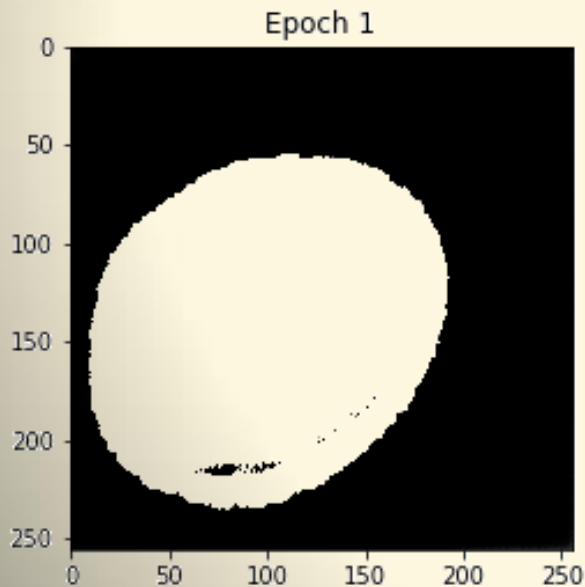
- Aparté sur la data augmentation



– Création de zones blanches => casse l'apprentissage

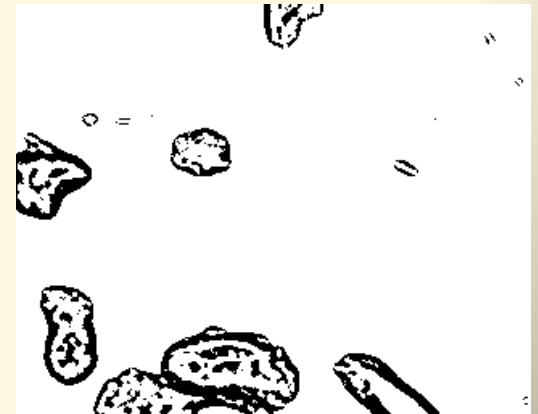
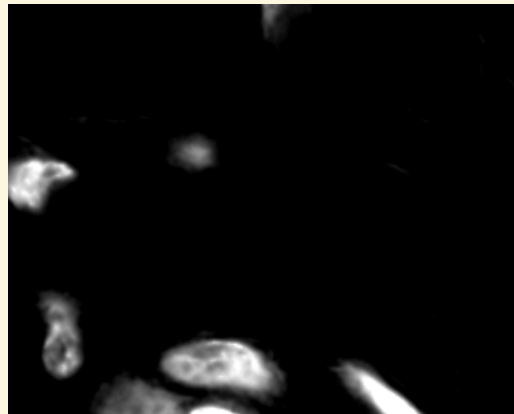
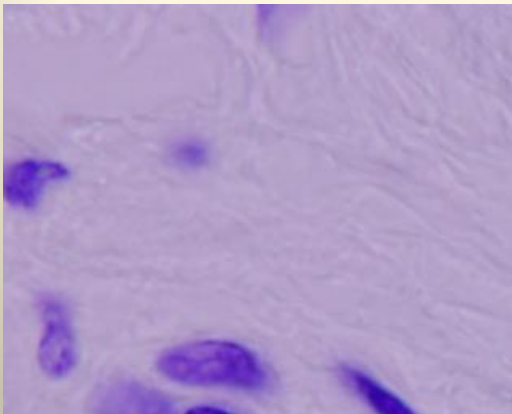
# Modèles

- CNN (version 1)
  - Redimensionnement classique (faible facteur)
  - Images pré-processées en 256x256



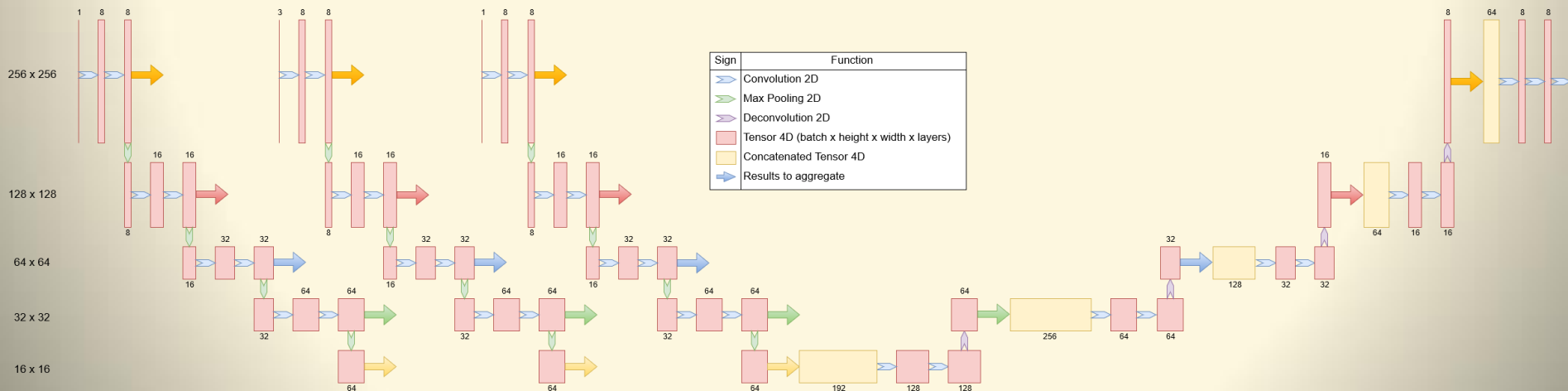
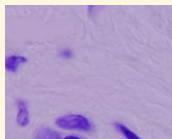
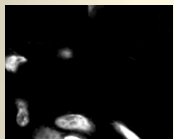
# Modèles

- CNN (version 2)
  - Postulat
    - Images pré-processées aide mais perte d'infos
    - Images de base un peu complexe ?
    - Adaptative Threshold aidera a splitter les cellules



# Modèles

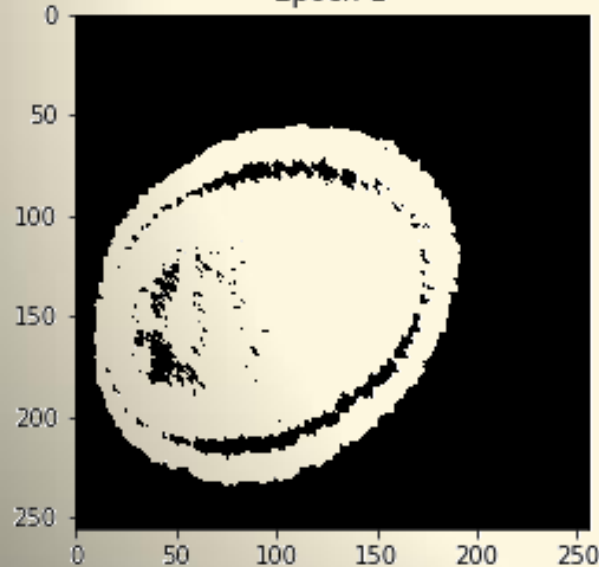
- CNN (version 2)
  - Idée : U-net extended (multi arm)
  - Entrée = Sortie =  $256 \times 256$



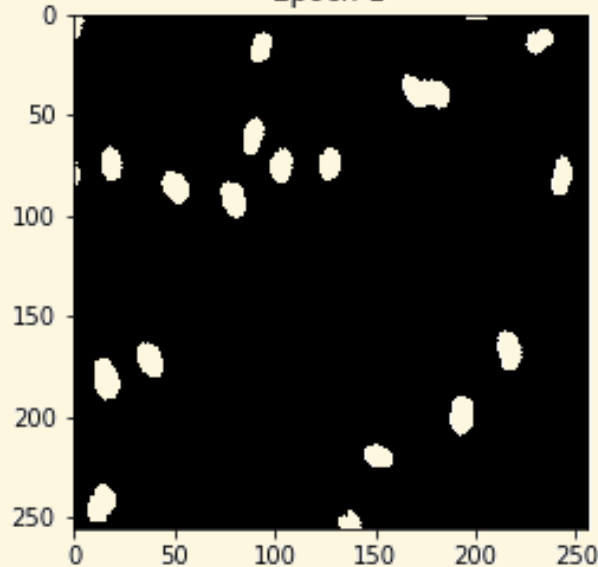
# Modèles

- CNN (version 2)
  - Training semble meilleur
  - Score inférieur (intersection faible, union correcte ?)

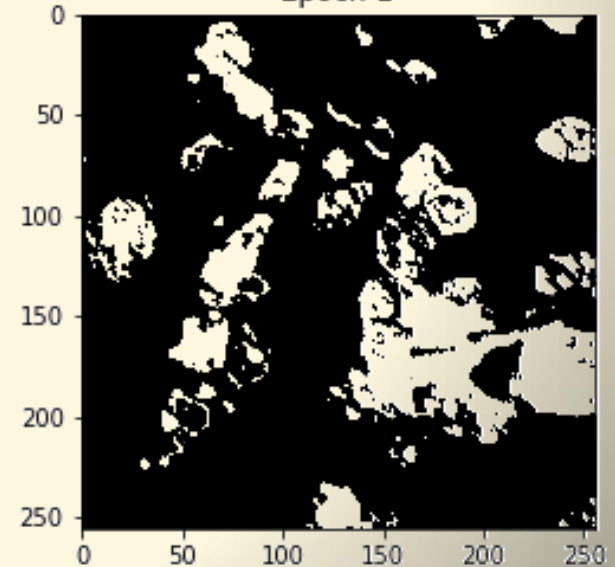
Epoch 1



Epoch 1



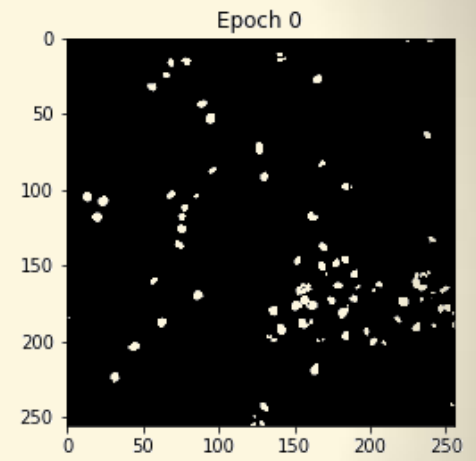
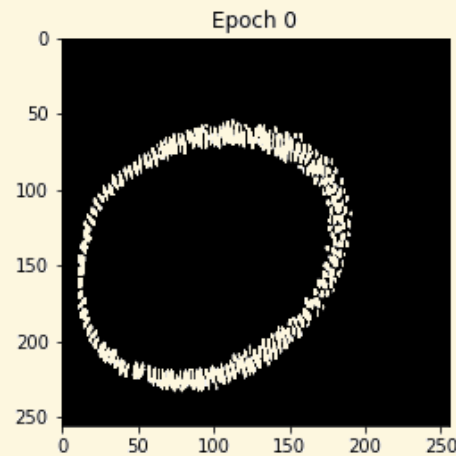
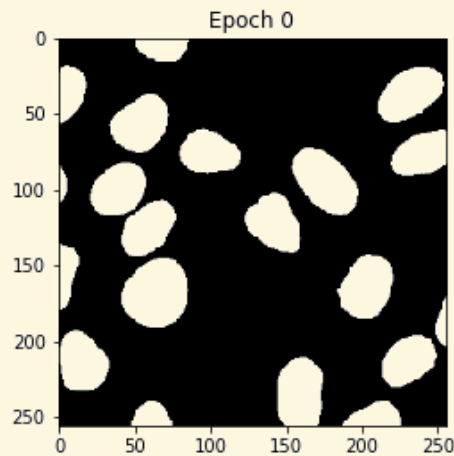
Epoch 1



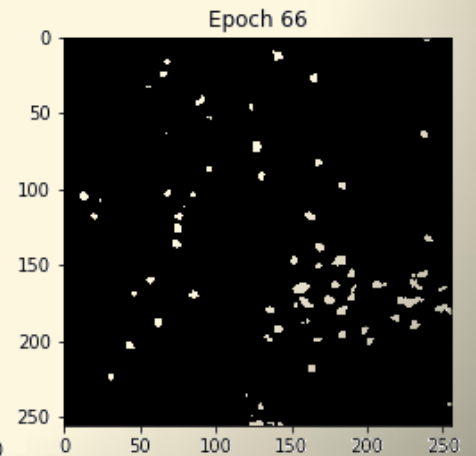
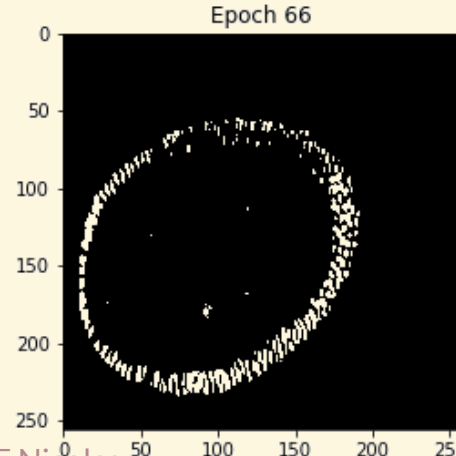
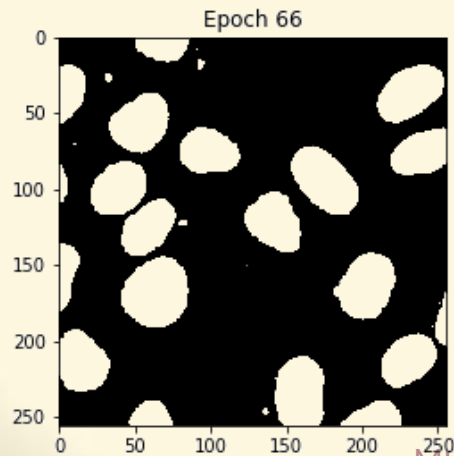
# Modèles

- CNN (version 2)

Vrai masque

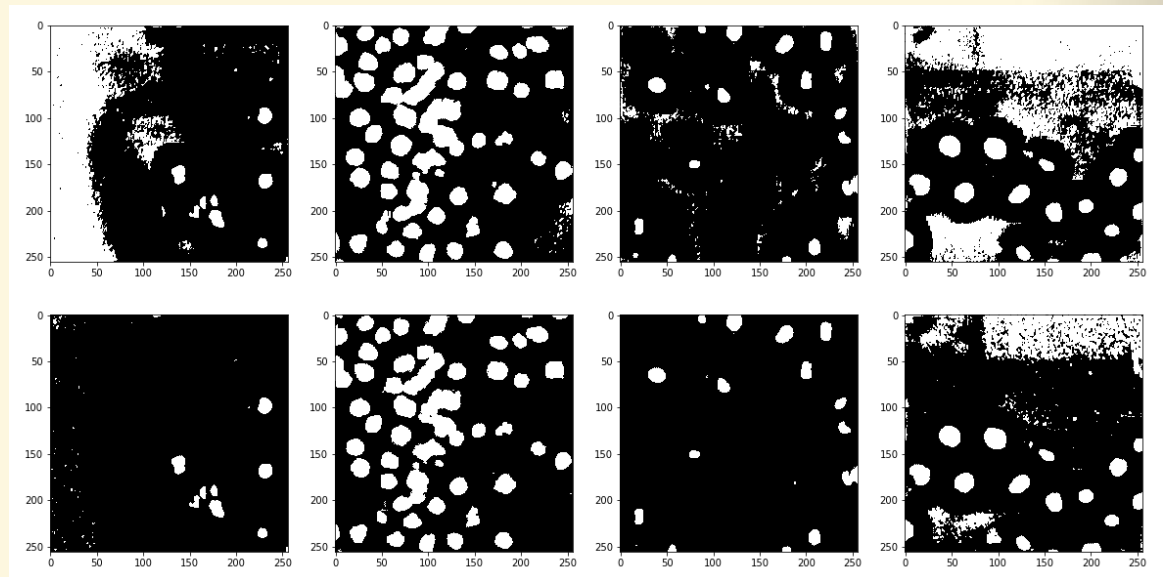


Masque prédit



# Post Processing

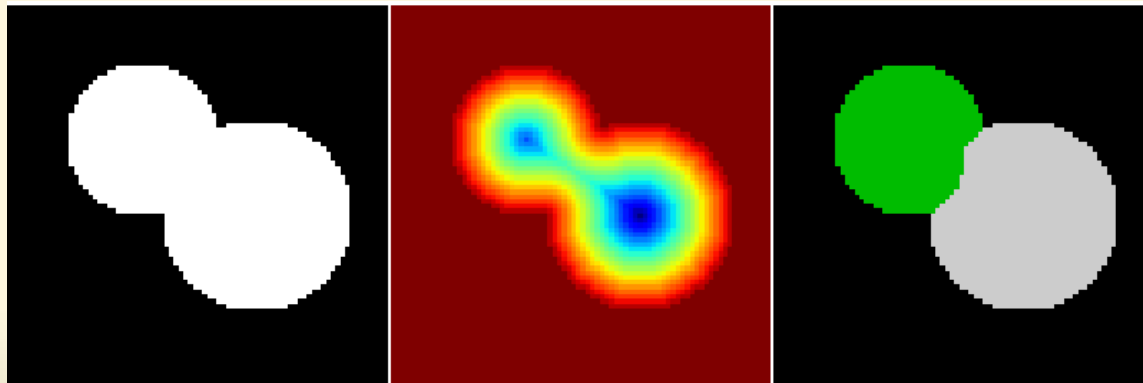
- Recherche du meilleur Threshold
  - Chaque pixel est issue d'une sigmoïde
  - Regarde évolution du Score en fonction du seuil
  - Modèle Classique
    - $T = 40/255$
  - CNN V1
    - $T = 0,9999$
  - CNN V2
    - $T = 0,4$



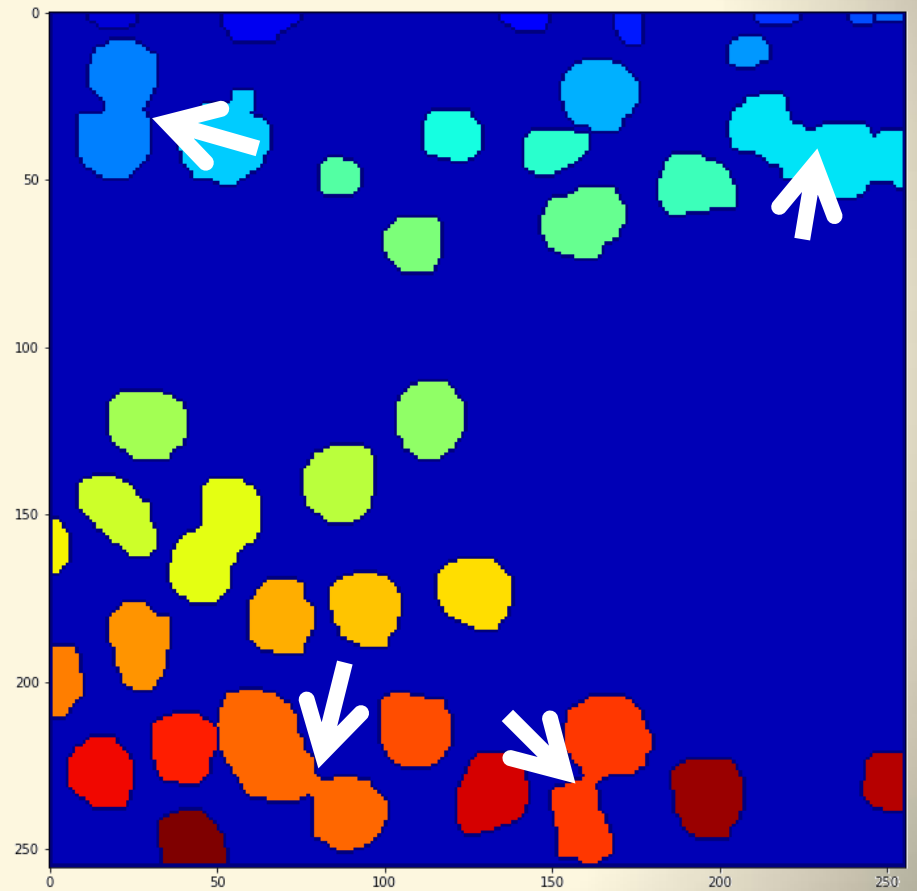
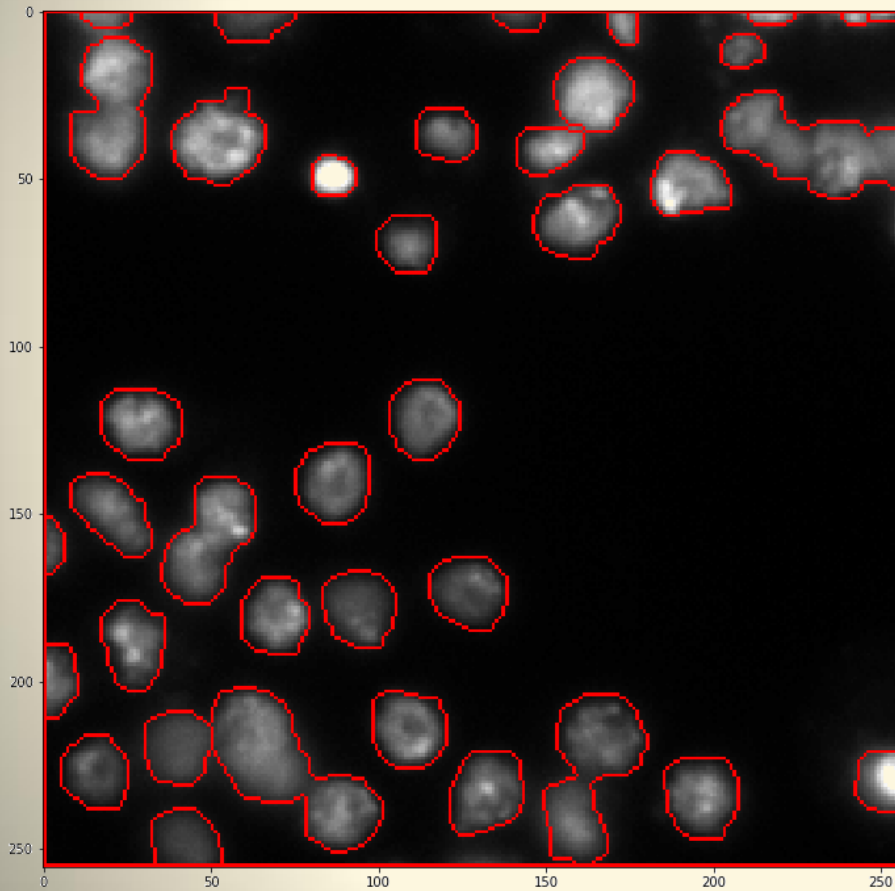


# Post Processing

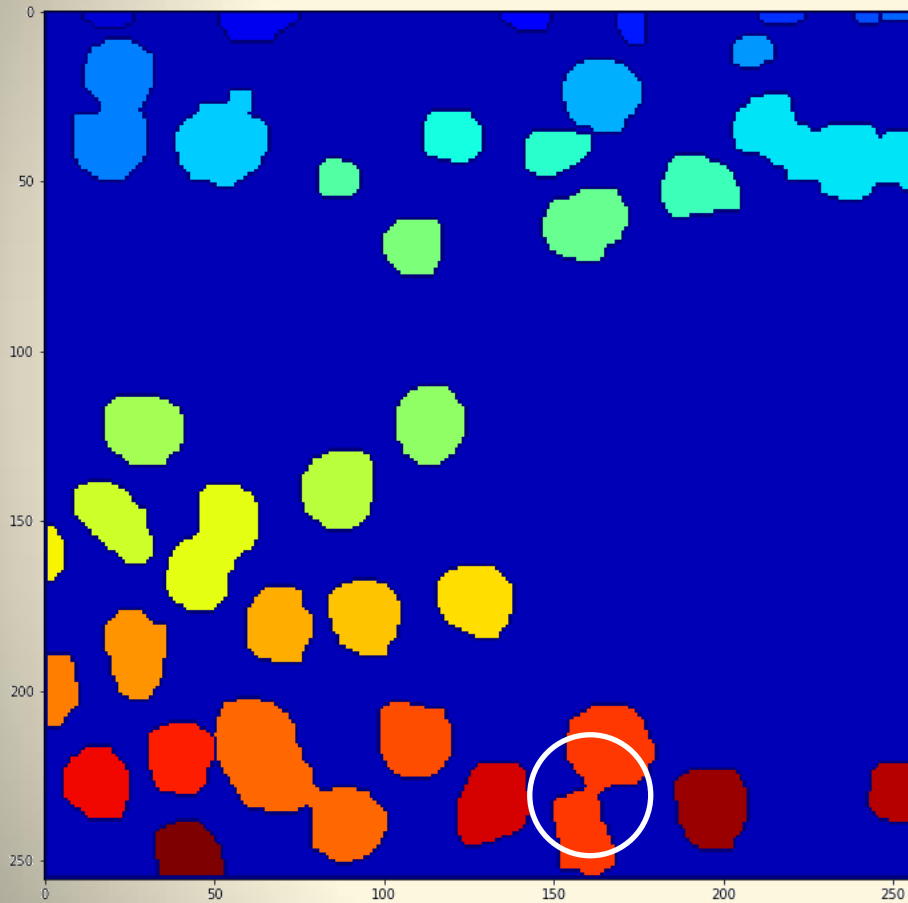
- Essai de suppression du bruit
  - `cv2.morphologyEx`
  - Erode et Dilate
- Essai de séparation des cellules « collées »
  - Watershed
  - Filtre oval



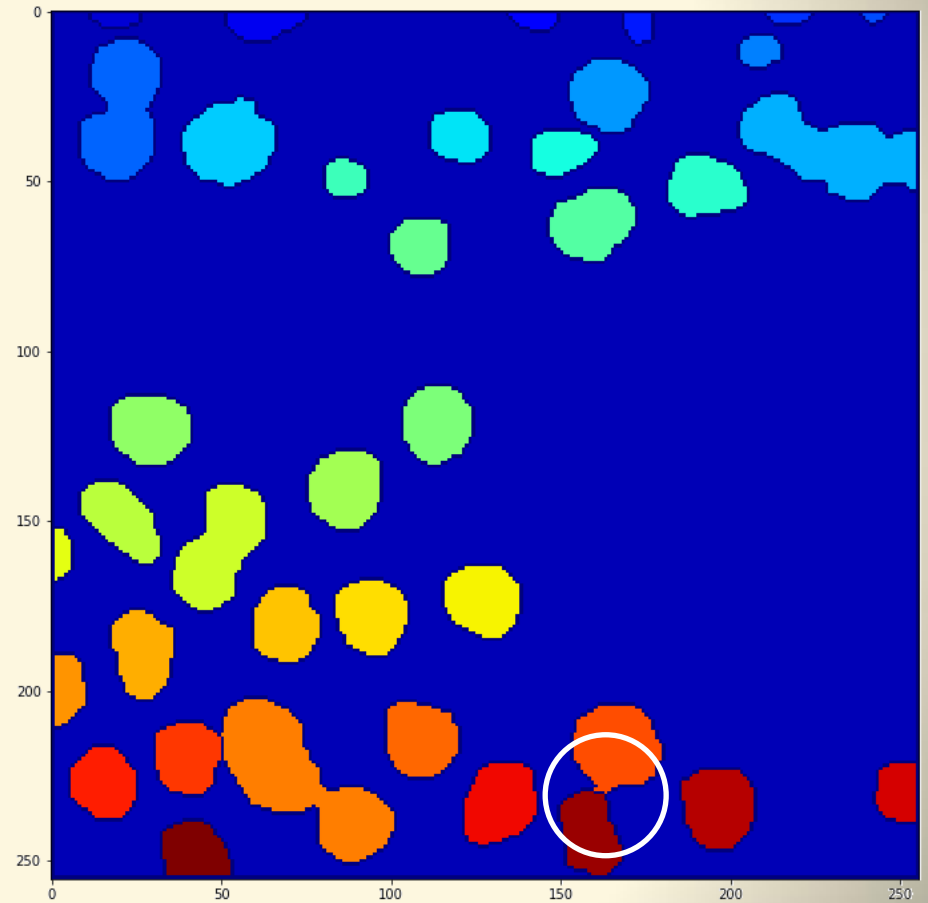
# Post Processing



# Post Processing



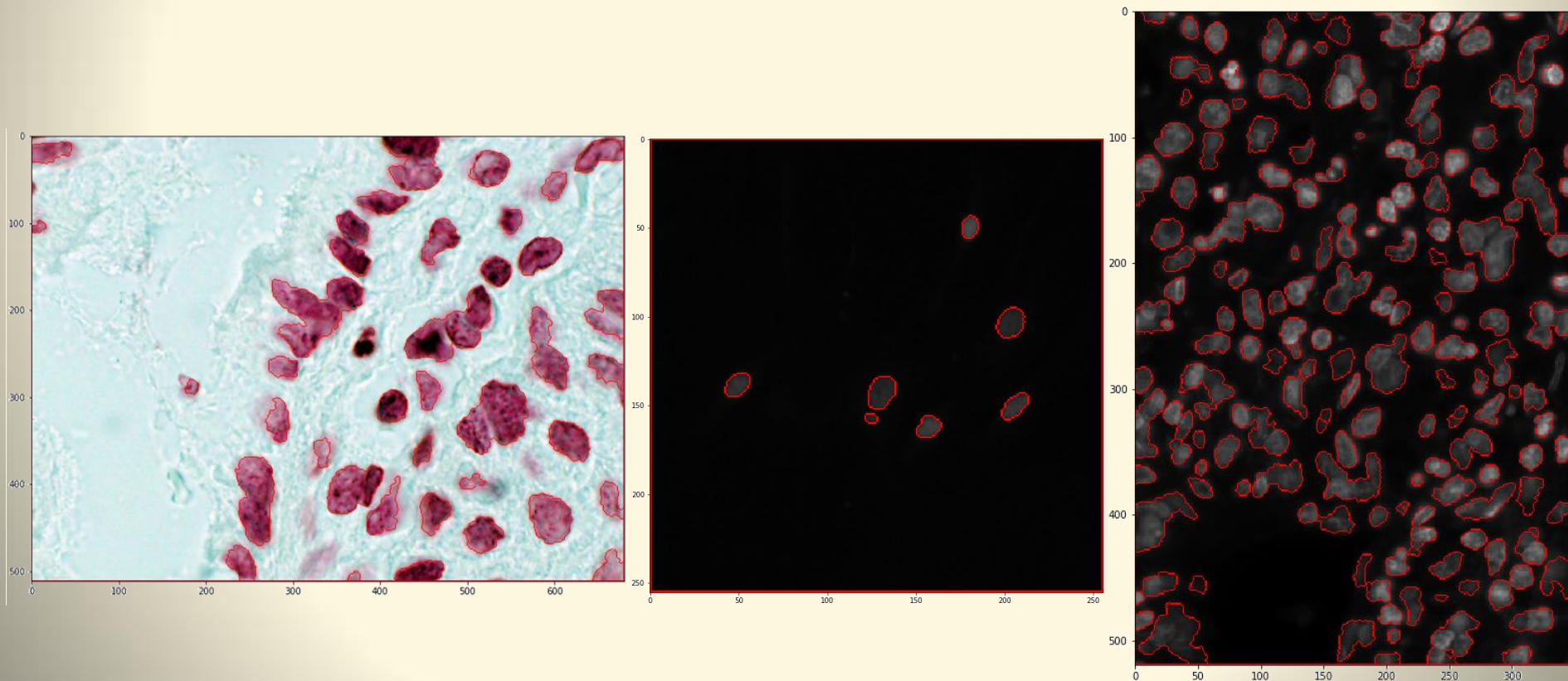
Filtre ovale = Off



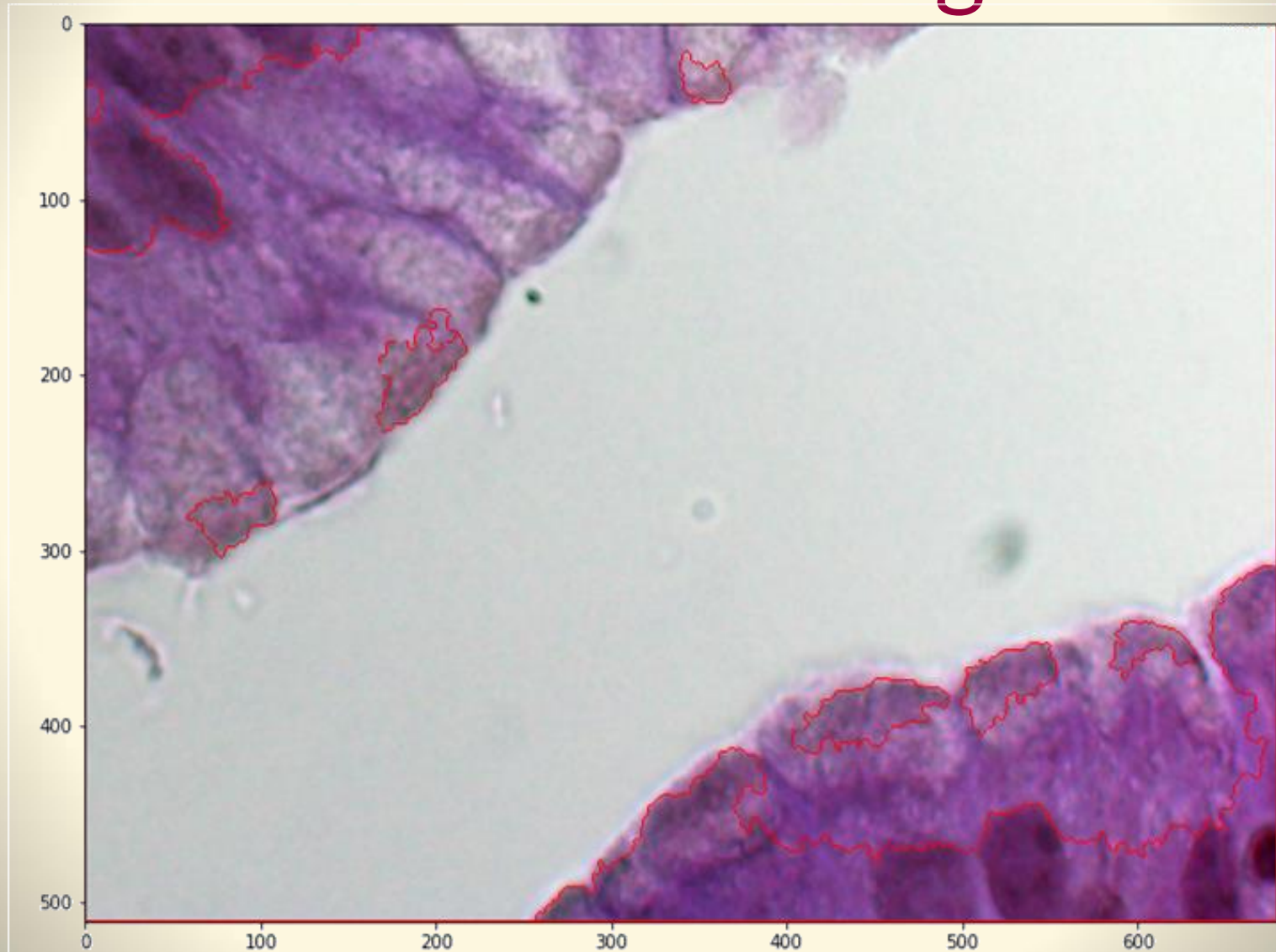
Filtre ovale = On

# Post Processing

- Résultat visuels corrects



# Post Processing



03/02/2018

MINE Nicolas

# Résultats

- Score sur masque complet:
  - Modèle Classique : 60,2 %
  - CNN V1 : 42,1% (très variable)
  - CNN V2 : 71,9%
- Score Kaggle
  - Modèle Classique : 0,226
  - CNN V1 : 0,189 à 0,301
  - CNN V2 : 0,208 à 0,243

# Résultats

- Explications :
  - Privilégier débordement sans overlap
    - Intersection : Linéaire
    - Union : inversement proportionnel
  - Evaluation sur certains masques uniquement
    - Images plus complexes
  - Mauvaise décompositions des cellules en contact
    - $IoU < 0,5$  très rapidement

# Améliorations possibles

- Utiliser le modèle Extended en 512 x 512
- Retester Watershed sur images plus précises
- Elastic Transformation sur la data augmentation
  - Utilisé sur U-net 2015
  - Non implémenté sur Keras
- Ensemble de U-net
  - Prédiction variable d'une epoch à l'autre
- U-net de 2015 avec images de 388 x 388 au centre d'une image de 572 x 572



# Conclusion

- Découverte : Image Segmentation
- Approfondissement des techniques de traitement d'images par rapport au P7
- Bon résultats avec une méthode simpliste
- CNN un peu au dessus (sauf parfois au score)
- Découverte « auto-encoder » pour images
  - U-net
- Résultats moyen sur Kaggle
  - Evaluation complexe à comprendre/debugger

