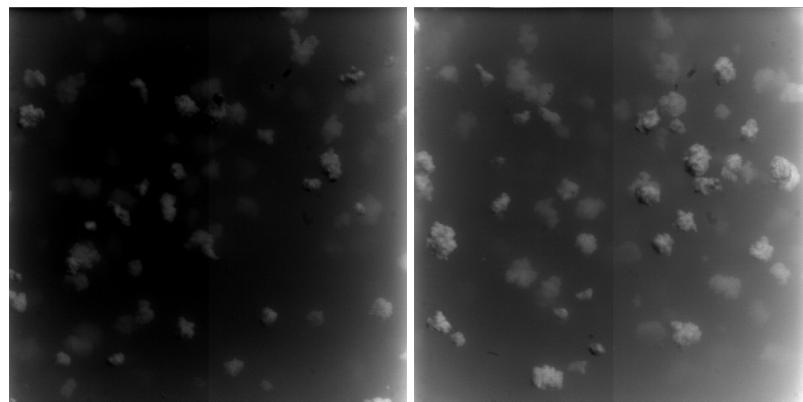


TB1 projet : U-NET segmentation Bellanca Ugo

1 - Pre-processing.....	1
2 - U-NET model.....	3
3 - Post-processing.....	4
4 - Résultats.....	5

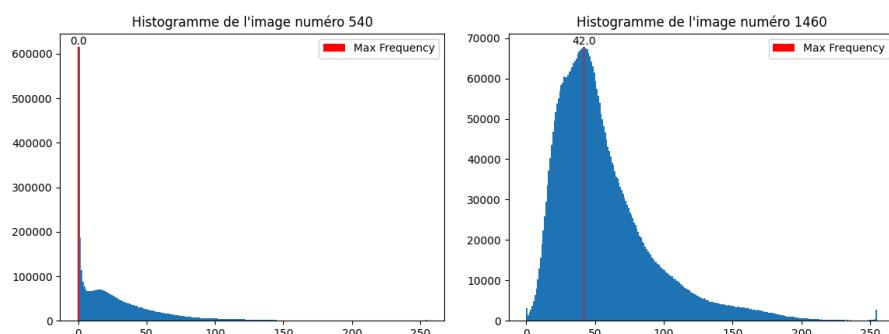
1 - Pre-processing

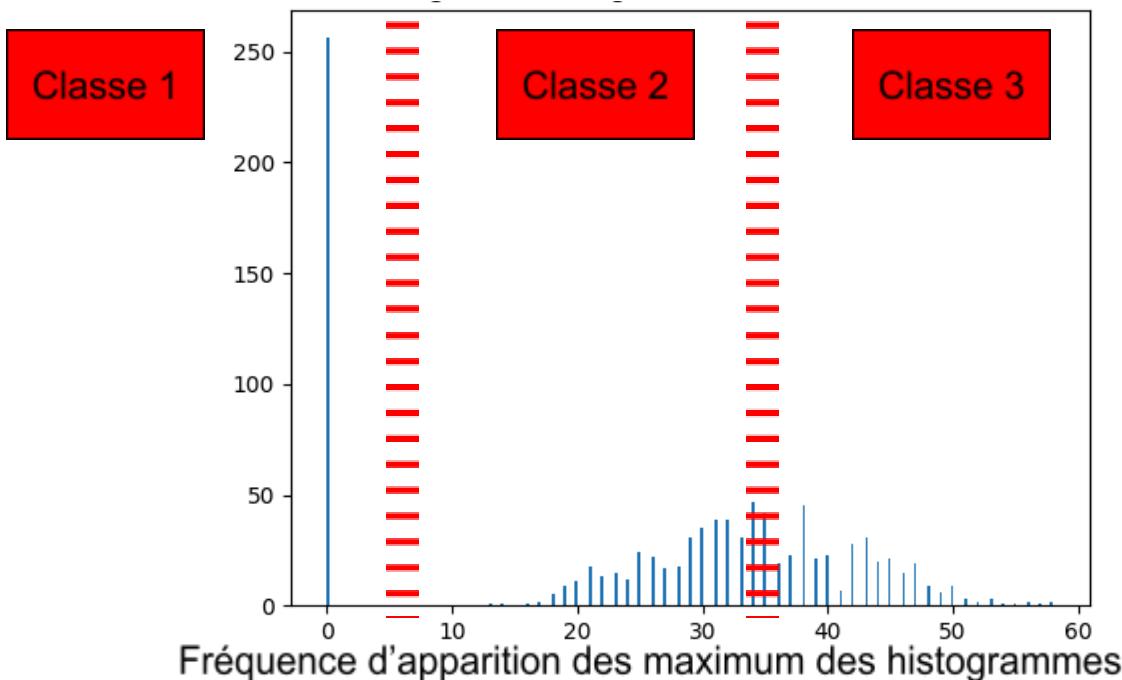
En regardant rapidement les images, il y a des images plus sombres que d'autres :



On va donc regrouper les images en classes selon leurs histogrammes.

On observe 2 principaux types d'histogrammes, ceux ayant la fréquence maximale en 0 (moins éclairées) et les autres (plus éclairées) ayant la fréquence maximale entre 10 et 60.



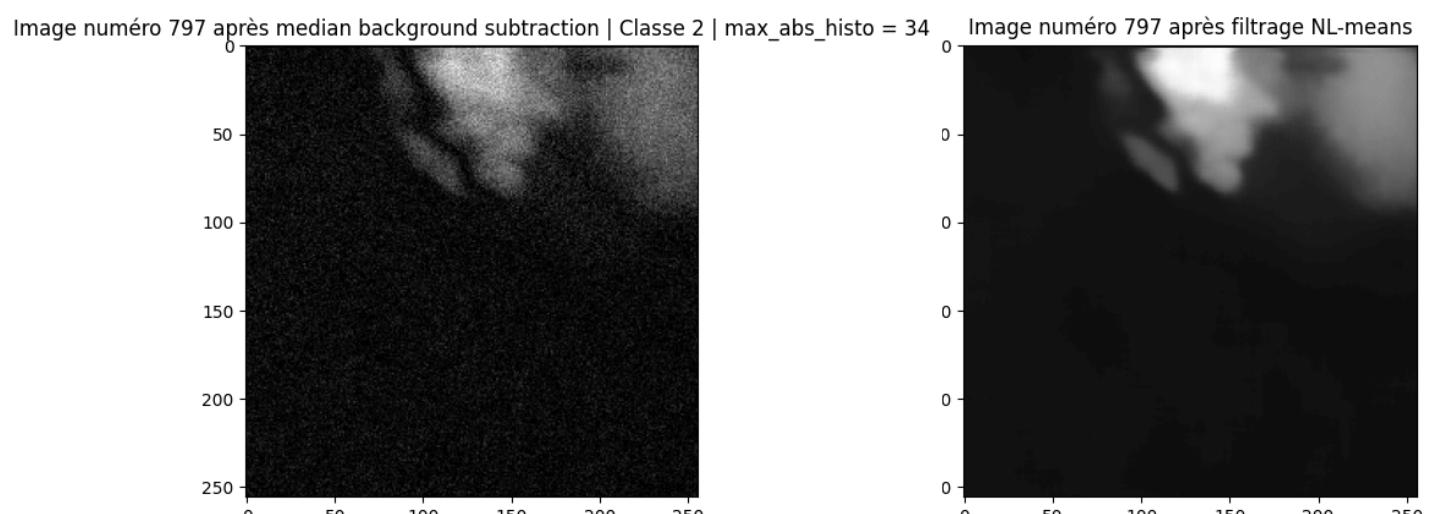


On décide de regrouper les images en 3 classes avec les seuils 5 et 35.

On fait cela pour pouvoir appliquer une soustraction du fond médian calculé sur 300 images prises au hasard parmi les 1619 images du dossier. (La médiane est plus robuste que la moyenne d'où ce choix, même si le résultat est identique). Cela règle aussi le problème de la ligne verticale et de la différence de luminosité gauche/droite. On a tenté de le régler ce problème par une soustraction de la partie réelle du plan de Fourier, ce qui règle effectivement la ligne mais introduit des "ombres" délicates à traiter. On se contentera de la soustraction par fond médian.

Après la soustraction du fond médian, on applique un filtre NL-means, qui prend en compte l'ensemble de l'image pour estimer le bruit et le supprime tout en conservant les contours de l'image.

On termine le pré-traitement par l'application d'un filtre gaussien pour légèrement flouter l'image, ce qui donne de meilleurs résultats en sortie d' U-NET en pratique dans notre cas.
Temps d'exécution pour 10 images = 10 minutes environ



2 - U-NET model

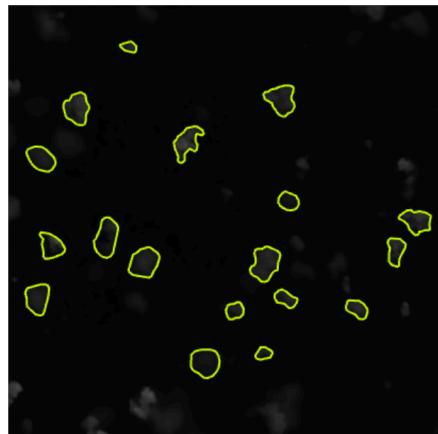
On pourrait penser utiliser U-NET sur les images 2048x2048 mais la RAM plante. Il faut donc soit réduire la taille des images de départ (en choisissant de modifier la variable SIZE=2048) soit patcher l'image de départ en 64 patches de taille 256x256.

C'est ce que l'on choisit de faire. On installe donc *patchify* pour éviter de patcher et depatcher manuellement.

- Notre modèle U-NET est un modèle classique pour image 256x256.
- On a annoté 50 masques 2048x2048 = 3200 masques 256x256.
- On fait un Train Test Split avec 30% de test.

Malheureusement, la RAM n'est pas suffisante et plante encore une fois. Nous pouvons utiliser 30 masques au maximum.

De même, le nombre d'epochs n'est que de 30. Le modèle peut être amélioré en prenant 100, ce qui est généralement fait dans la littérature. (J'ai dépassé mon quota Google Colab).



La luminosité excessivement basse est due au logiciel d'annotation (pour mieux voir les masques)

3 - Post-processing

Exemple d'image pré traitement et son masque

post_process_1 : temps d'exécution = 5 minutes

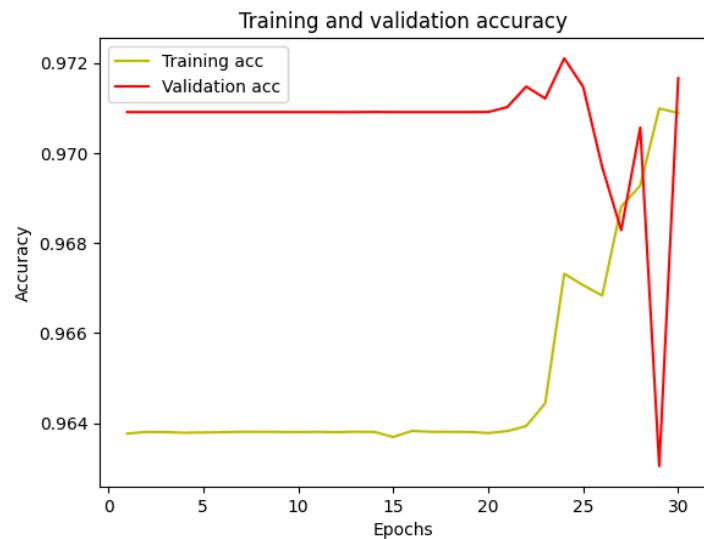
- Border cleaning
- 4 closing avec élément structurant en disque | bouche les petits trous dans les objets
- 5 opening | supprime les petits objets
- Watershed par sure fg & bg et markers | supprimer les particules connectées/superposées qui faussent l'analyse en binaire

post_process_2 : temps d'exécution = 15 secondes

- On labellise les objets
- Par la fonction regionprops de skimage, on détermine les caractéristiques de chaque label

Avantage 1 = on a des données morphologiques sur chaque label = nanoparticules.
On peut facilement en faire une étude statistique, ce n'est pas l'objet de ce présent compte rendu

Avantage 2 = on a éliminé facilement les objets trop gros pour être éliminés par opening en thresholdant df['area']



Analyse primaire du modèle U-NET :

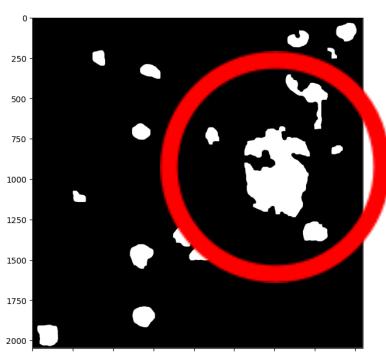
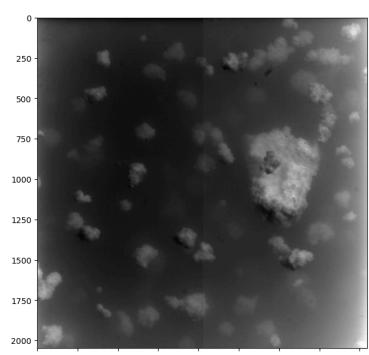
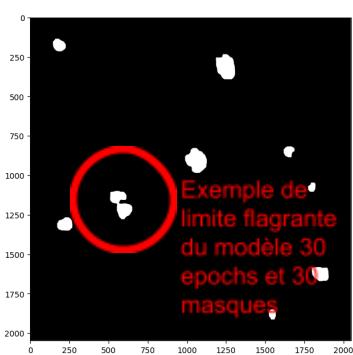
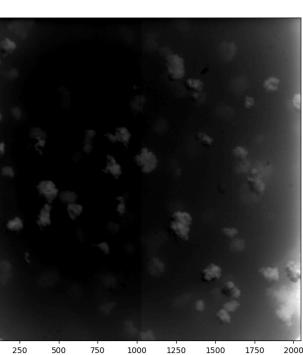
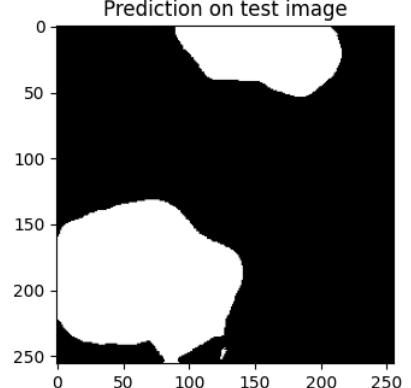
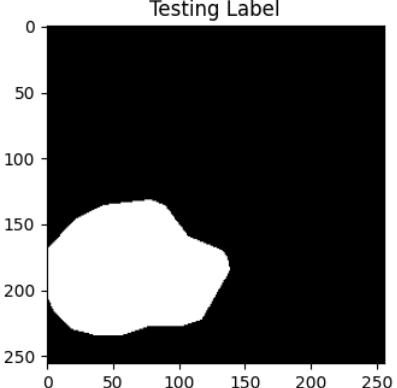
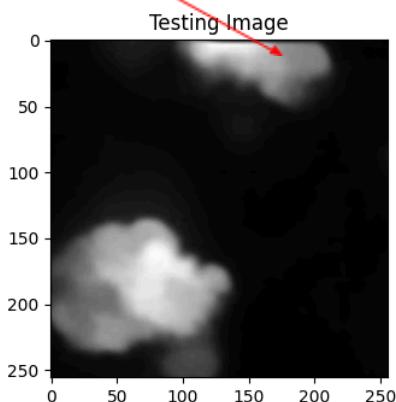
Avoir plus d'epochs sur un ordinateur plus puissant serait très intéressant au vu de la courbe training accuracy qui commençait à bien augmenter, même si dans le cas de la segmentation binaire, l'accuracy n'est pas le meilleur indicateur.

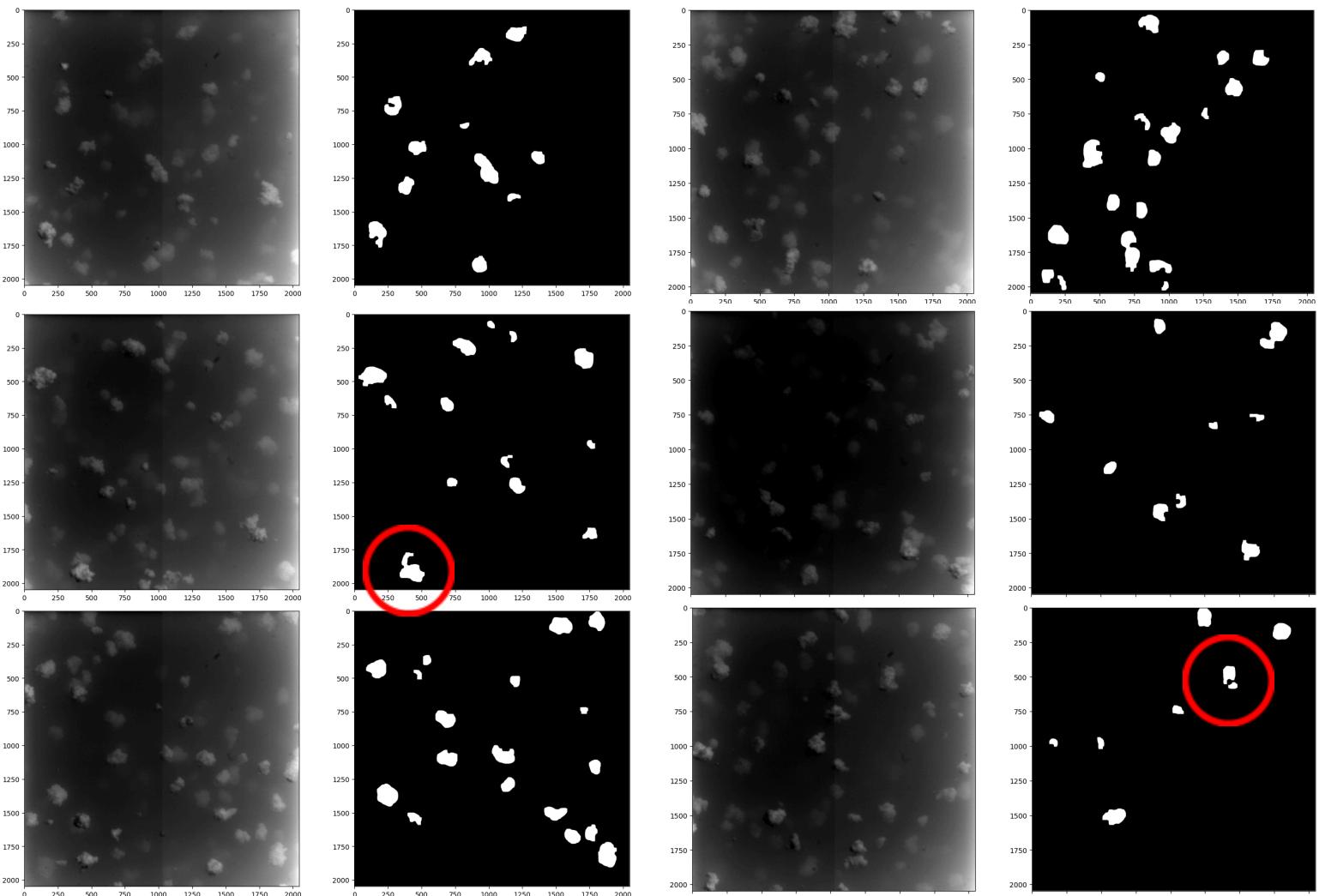
Un meilleur serait l'IoU (Intersection over Union), dont on oscille entre 35 et 55% selon le nombre de masques, d'epochs, du calcul du modèle etc..

Ce qui n'est pas si mauvais, sans être exceptionnel, sachant qu'on peut ajouter des masques et des epochs (un bon, très satisfaisant IoU est 80% à titre d'exemple)

4 - Résultats

particule qui se superpose avec une autre => non labellisé





Conclusion :

Notre but, et cela se voit au travers des masques, était d'avoir seulement les objets intéressants pour une étude statistique, donc pas de superposition voulu (connected objects), pas les objets coupés par le bord, pas obtenir le plus possible d'objets etc..

La méthode Deep Learning U-NET se révèle être ici très intéressante en ce sens : les résultats sont satisfaisants (une étude statistique pourrait avoir lieu et les résultats devraient être plutôt correctes) et ils peuvent sûrement être encore bien améliorés par l'emploi d'un ordinateur plus puissant pour ajouter des epochs et des masques avec leurs jeux d'augmentations (data augmentation, non fait ici car HS) pour éviter cet overfitting.