

# Détection d'objets dans des séquences d'images couleurs : compte rendu 5

Marie Bocquelet,  
Fabien Caballero

25 Novembre 2023

## I Avancement :

**-Implémentation des filtres.**

**-Liaisons des filtres à l'application :** Nous avons commencé à faire la liaison entre les scripts Python et l'application C++. Pour l'instant seulement les filtres sont liés.

**-Analyse de la méthode par soustraction sans améliorations :** Nous avons analysé comment évoluait la méthode utilisée sans aucune améliorations et avec variation du seuil utilisé pour la création du masque. Pour représenter comment le modèle évoluait, nous avons utilisé le calcul de l'accuracy. Pour cela, il nous a aussi fallu calculer les vp, vn, fp et fn. Or, sur les vidéos test, nous ne possédions pas la vérité terrain. Donc pour l'obtenir, nous avons annoté une vidéo, ce qui nous a permis de localiser les poissons grâce à des rectangles dont nous possédons les coordonnées du centre, la hauteur et la largeur dans des fichiers textes par image.

- *Calcul des vrais positifs* : dans le masque, les poissons sont censés correspondre aux parties blanches. Ainsi, pour calculer le nombre de pixels identifiés comme poisson et étant réellement des poissons, nous regardions si le pixel était blanc et si il appartenait à l'un des rectangles de la vérité terrain.
- *Calcul des vrais négatifs* : les vrais négatifs correspondent aux pixels noirs qui ne sont pas dans l'un des rectangles.
- *Calcul des faux positifs* : les faux positifs sont les pixels blancs qui ne se trouvent pas dans l'un des rectangles.
- *Calcul des faux négatifs* : les faux négatifs sont les pixels noirs qui se trouvent dans l'un des rectangles de la vérité terrain.

Il y aura forcément des faux négatifs car les rectangles entourant les poissons ne sont pas collés aux poissons, donc parfois des pixels noirs se trouveront forcément dans le rectangle. Aussi, nous n'avons pas réussi à améliorer le nettoyage des roches.

**-Lancement d'un training de 200 et 300 epochs, et test pour 100 epochs :** Pour 100 epochs, voici ce que l'on a pu obtenir :

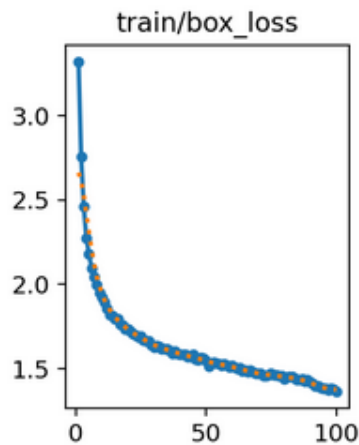


Figure 1: Courbe de loss en fonctions des epochs

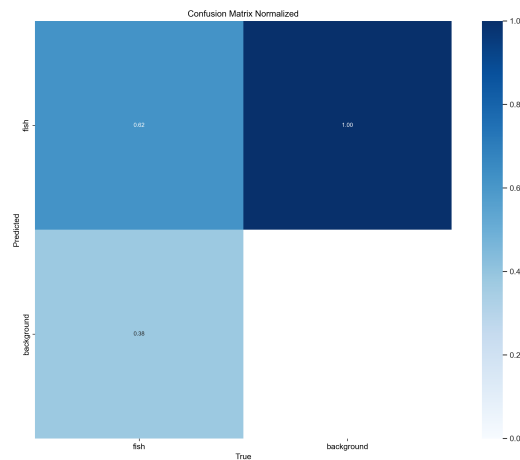


Figure 2: Matrice de confusion 100 epochs

On peut voir que la loss n'a pas encore atteint convergence, pour obtenir la meilleure loss possible, il faut augmenter le nombre d'epochs, d'où les training pour 200 et 300. On voit aussi sur la matrice de confusion une augmentation de 0.12 par rapport à 20 epochs pour la détection des poissons, ce qui est une nette amélioration et cela se ressent au niveau des vidéos.

## II Prochaines étapes :

- Finir de lier les scripts à l'application.
- Faire l'analyse de la méthode par soustraction avec améliorations.
- Lancer des training en changeant d'autres paramètres que le nombre d'epochs, par exemple la taille du batch.

## III Liens :

- <https://github.com/fcaball/Detection-Poissons>
- <https://trello.com/b/P8TKTbEG/projet-image>

## IV Citations :

Pour l'état de l'art nous nous sommes inspirés de l'article suivant : "Contribution à la détection et à la reconnaissance d'objet dans les images", de Hedi Harzallah.

Pour le deep learning nous avons utilisé le tutoriel de : <https://www.youtube.com/watch?v=Z-65nqxUdl4&list=LL&index=3&t=3451s> et du github associé <https://github.com/computervisioneng/yolov8-full-tutorial>

Pour la méthode par soustraction nous nous sommes aidés de : <https://www.youtube.com/watch?v=pkzT9MlICPE&list=LL&index=2&t=195s> et du github associé <https://github.com/L42Project/Tutoriels/tree/master/OpenCV/tutoriel13>