

Détection d'objets dans des séquences d'images couleurs : compte rendu 3

Marie Bocquelet,
Fabien Caballero

12 Novembre 2023

I Changement dans l'organisation :

- Abandon de la méthode sea-thru de nettoyage des images sous-marines : nous avons estimé que cette méthode était trop conséquente pour le temps imparti,
- Implémentation de méthodes basiques de nettoyage d'images : augmentation de contraste, amélioration de la netteté, de la luminosité...
- Nous avons décidé d'utiliser finalement la version 8 de YOLO,
- Première étape : récupération d'un dataset de poissons passé au format YOLO sur internet, training et test sur des données totalement inconnues.
- Deuxième étape : utilisation du dataset fournit par Quantacell, training et test sur des vidéos inconnues.
- Troisième étape : comparaison des résultats obtenus en fonction des deux datasets.
- Les décisions concernant l'application n'ont pas changé. Cependant, suite à quelques problèmes rencontrés avec Qt Python, nous allons sûrement développer l'application en Qt C++ puis appeler les scripts Python depuis cette application.

II Avancement :

- Récupération d'un dataset sur internet d'environ 6000 images :
<https://storage.googleapis.com/openimages/web/visualizer/index.html>
- Passage de ce dataset sous le format pris en charge par YOLO. Le format est : un dossier train qui contient l'ensemble des images et un dossier labels qui contient les annotations correspondantes.

- Installation de ultralytics : développement de projets open source, dont une implémentation évolutive de YOLO.
- Nous avons choisit la version yolov8n, qui est la version nano car c'est la plus légère.
- Lancement d'un training sur l'ensemble des images avec un nombre d'epochs de 100 et un batch size de 16.
- Continuation de l'état de l'art.

Pour récupérer le dataset et le passer au format YOLO, nous nous sommes aidé d'un tutoriel disponible sur Youtube : <https://www.youtube.com/watch?v=Z-65nqxUd14>.

Les annotations au format YOLO se présentent sous la forme suivante :

```
1 0 0.5046875000000001 0.2303055 0.6118750000000001 0.351251
2 0 0.500625 0.649212 0.88375 0.46802600000000005
```

Le premier élément correspond à la classe, les deux suivants correspondent au centre du rectangle entourant l'objet correspondant à la classe sur l'image. Enfin, les deux derniers correspondent à la hauteur et à la largeur du rectangle. Dans un fichier texte, il y aura sur chaque ligne, un rectangle correspondant à un objet dans l'image correspondante.

Voici un exemple des images et de leurs annotations utilisées pour le training :

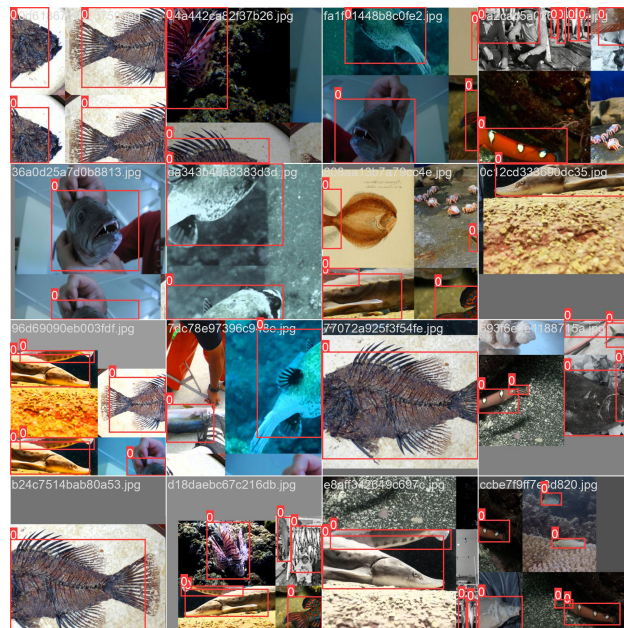


Figure 1: train batch 0

III Prochaines étapes :

- Recommencer l'application en version C++.
- Terminer l'état de l'art.
- Tester le modèle entraîné dont nous avons parlé précédemment sur une vidéo inconnue.
- Avancer sur les méthodes de nettoyage des images sous-marines.

IV Liens :

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