### Tests et mesures de performance des techniques abordées

#### Par langage :

Matlab

[Drive Laura Levraud - Recherches](https://drive.google.com/drive/folders/138VMV6KuIpHox_za9xacHmCljQ9jWMc2?usp=drive_link)

* Pixel pointing
* SORT
* SORTplusYOLOv8
* SURF
* YOLO

Python

[Drive Laura Levraud - Recherches](https://drive.google.com/drive/folders/138VMV6KuIpHox_za9xacHmCljQ9jWMc2?usp=drive_link)

* DeepSORT
* OpenCVTrackers
* OpenMMLab MMTracking
* Pixel pointing
* SURF
* YOLO

[Github Laura Levraud - RealTimeMOT](https://github.com/lauralvd01/RealTimeMOT)

* DeepSORT
* Live (mesures de performances du YoloV8 en mode temps réel)
* OpenCVTrackers
* YoloV8

C#

[Github Laura Levraud - GeoLiveDetect](https://github.com/lauralvd01/GeoLiveDectect)

* DeepSORT
* Interface Decklink-GeoRender/GeoWebSocket
* Yolov8

#### 

#### Par technique :

DeepSORT

[Drive Laura Levraud - Recherches - DeepSORT](https://drive.google.com/drive/folders/15JdtEqUnVGqnC7zlligUJQyNMhr4X5zq?usp=drive_link)

* Licence
* Notes et études, implémentations sur Google Colab
* Publication des recherches de Nicolaï Wokje
* Description détaillée des principes et algorithmes
* Mise en place et étude de l’implémentation de Nicolaï Wokje
* Premiers essais sur une vidéo de piétons

[Github Laura Levraud - RealTimeMOT \ DeepSORT](https://github.com/lauralvd01/RealTimeMOT/tree/main/DeepSORT)

* [RealTimeMOT\README.md](https://github.com/lauralvd01/RealTimeMOT/blob/main/README.md#deepsort) Mise en place et utilisation
* [.vscode\launch.json](https://github.com/lauralvd01/RealTimeMOT/blob/main/DeepSORT/.vscode/launch.json) configuration de vscode pour exécuter en debug les fichiers [tools\generate\_detections.py](https://github.com/lauralvd01/RealTimeMOT/blob/main/DeepSORT/deep_sort-master/tools/generate_detections.py), [deep\_sort\_app.py](https://github.com/lauralvd01/RealTimeMOT/blob/main/DeepSORT/deep_sort-master/deep_sort_app.py) et [generate\_videos.py](https://github.com/lauralvd01/RealTimeMOT/blob/main/DeepSORT/deep_sort-master/generate_videos.py) avec leurs arguments respectifs. Fichier à modifier pour pouvoir exécuter le DeepSORT sur ses propres fichiers.
* [deep\_sort-master](https://github.com/lauralvd01/RealTimeMOT/tree/main/DeepSORT/deep_sort-master) implémentation de NicolaÏ Wokje, corrigée et adaptée à l’utilisation en debug à partir des résultats de [RealTimeMOT\YoloV8\yolov8\_app.py](https://github.com/lauralvd01/RealTimeMOT/blob/main/YoloV8/yolov8_app.py)

OpenCVTrackers

[Drive Laura Levraud - Recherches - OpenCVTrackers](https://drive.google.com/drive/folders/1XpRhR_Rnm9fKmrNO1DjaBABVDGHr5hQ_?usp=drive_link)

* Implémentation sur Google Colab et tests sur les voitures
* Installation et implémentation python
* [\dataFiles](https://drive.google.com/drive/folders/1qDPApk-6TwK4_iakxdEdKPIU-jpxcVtv?usp=drive_link) contenant les fichiers csv servant pour les tests : init pour les initialisations des bounding boxes au début de chaque vidéo, tests pour les bounding boxes détectées au fur et à mesure de chaque vidéo
* [\outputVideos](https://drive.google.com/drive/folders/1_tIPsucr48DTh0fkRSSnt7ZqpKcONRmp?usp=drive_link) toutes les vidéos issues des tests des traqueurs pour chaque cible de chaque vidéo
* [OpenCV Trackers - Tests](https://docs.google.com/spreadsheets/d/1k4MN3ib_Ktv8BXrFAondOk85JN8LG-imyOQiMQh3DGw/edit?usp=drive_link) tables recensant les liens, les arguments utilisés et les données récoltées pour chacun des tests

[Github Laura Levraud - RealTimeMOT \ OpenCVTrackers](https://github.com/lauralvd01/RealTimeMOT/tree/main/OpenCVTrackers)

* [RealTimeMOT\README.md](https://github.com/lauralvd01/RealTimeMOT/blob/main/README.md#deepsort) Mise en place et utilisation
* [\dataFiles](https://drive.google.com/drive/folders/1qDPApk-6TwK4_iakxdEdKPIU-jpxcVtv?usp=drive_link) contenant les fichiers csv servant pour les tests : init pour les initialisations des bounding boxes au début de chaque vidéo, tests pour les bounding boxes détectées au fur et à mesure de chaque vidéo
* [initialize.py](https://github.com/lauralvd01/RealTimeMOT/blob/main/OpenCVTrackers/initialize.py), [test.py](https://github.com/lauralvd01/RealTimeMOT/blob/main/OpenCVTrackers/test.py) et [testAll.py](https://github.com/lauralvd01/RealTimeMOT/blob/main/OpenCVTrackers/testAll.py) respectivement pour créer les initialisations des bounding boxes de chaque vidéo, tester une vidéo pour une initialisation et tester chaque vid\Edge & Contour detection avec chaque initialisation

Open MMLab MMTracking

[Drive Laura Levraud - Recherches - Open MMLab MMTracking](https://drive.google.com/drive/folders/1K6wnpEtEm5qgODLRMXqoakao4DF78pwn?usp=drive_link)

* Notes et tests d’implémentation
* Gif de présentation
* [Post Linkedin Open MMPose](https://www.linkedin.com/embed/feed/update/urn:li:ugcPost:7149073578833969152)

Pixel pointing

[Drive Laura Levraud - Recherches - Pixel pointing](https://drive.google.com/drive/folders/13ffTAsW-8fEINsIVGLb0a2jnN9lcnA5K?usp=drive_link)

* Liste des pistes creusée et à creuser, et des liens associés
* Implémentations sur Google Colab et tests sur une image de bateau pour toutes les techniques envisagées
* [\BLOB](https://drive.google.com/drive/folders/1O2edjySryeVNAQKv13hMM1126ZguDaSh?usp=drive_link)
* [\Edge & Contour detection](https://drive.google.com/drive/folders/1H0aDB9wnmqhH0bjYf3AsG8B5UQqPkh1O?usp=drive_link)
* [\K-means](https://drive.google.com/drive/folders/1z2uO3wVtn1-RTMO6a1qewdCFoGef7C1n?usp=drive_link)
* [\OpenCV](https://drive.google.com/drive/folders/14V5muc_zOxEF7jiRlJygW8mLVMmwuw30?usp=drive_link), liens avec [OpenCVTrackers](https://drive.google.com/drive/folders/1XpRhR_Rnm9fKmrNO1DjaBABVDGHr5hQ_?usp=drive_link) et vidéos démos
* Raccourci vers le dossier [SURF](https://drive.google.com/drive/folders/1RZVHqX3GXEWQn3uE2zKei8NfeJgDdlMJ?usp=drive_link)

SORT

[Drive Laura Levraud - Recherches - SORT](https://drive.google.com/drive/folders/1nsB5ubw2BSOxfDgBjnko_F9-F2j5JP5G?usp=drive_link)

* Licence
* Notes et études, comparaisons de performances
* Publication des recherches d’Alex Bewley
* Description détaillée des principes et algorithmes
* Implémentation Matlab à base des tutoriels Matlab
* Premiers essais sur des vidéos de piétons et vidéos démos des tutoriels

SORTplusYOLOv8

[Drive Laura Levraud - Recherches - SORTplusYOLOv8](https://drive.google.com/drive/folders/15F-mw9k4X4HlzGcGkrIgUxjw-v7IUxZD?usp=drive_link)

* Implémentation Matlab de yoloV8
* Premiers essais de yoloV8 sur une vidéo de bateau et une de vélo
* Implémentation Matlab de yoloV8 + SORT
* Premiers essais sur les vidéos résultat de yoloV8

SURF

[Drive Laura Levraud - Recherches - SURF](https://drive.google.com/drive/folders/1RZVHqX3GXEWQn3uE2zKei8NfeJgDdlMJ?usp=drive_link)

* Notes, étude des principes et algorithmes, réflexions sur la construction d’une implémentation perso Matlab et journal de bord du travail de recherche et développement mené tout au long de l’implémentation, des tests et de l’implémentation des différentes évolutions
* Implémentation perso Matlab
* Outils et fonctions Matlab récupérées dans les tutoriels Matlab
* Notes sur les tests et les évolutions des 5 différentes versions de l’implémentation
* Table complète contenant les données de chaque test, allant de la version de l’implémentation testée aux arguments utilisés et commentaires émis
* Dossier complet des vidéos de chacun des 60 tests menés
* Vidéos utilisées en input
* Tutoriel python OpenCV

YOLO

[Drive Laura Levraud - Recherches - YOLO](https://drive.google.com/drive/folders/1TERdwrJMYzN5sEAfh6cnZOUf2f-OMs4W?usp=drive_link)

* Notes et études
* Implémentations sur Google Colab
* Mise en place et implémentation en Python
* Publication de recherche de la base de données COCO
* Premiers essais sur une vidéo de bateau
* [\YoloV8](https://drive.google.com/drive/folders/152_eXCudzhcippW0GouPYx0z0llDqgNH?usp=drive_link) sauvegarde des premiers résultats de l’implémentation Python, d’un test de merge (entrelacement) de résultats donnés par différents modèles, des modèles entraînés sur les tests de base de données 1, 6, 12, 13, 14 et 15 et de leurs résultats respectifs
* Lien vers l’implémentation Matlab et le couplage avec SORT
* Test de deep learning par Detectron 2

[Github Laura Levraud - RealTimeMOT \ YoloV8](https://github.com/lauralvd01/RealTimeMOT/tree/76677a5f1ec69de62049392c2b44dc6f37b24e57/YoloV8)

* [RealTimeMOT\README.md](https://github.com/lauralvd01/RealTimeMOT/blob/main/README.md#deepsort) Mise en place et utilisation
* [\COCO](https://github.com/lauralvd01/RealTimeMOT/blob/main/YoloV8/COCO) Modèles entraînés sur la base de données COCO et résultats
* [\Train14](https://github.com/lauralvd01/RealTimeMOT/blob/main/YoloV8/Train14) et [\Train15](https://github.com/lauralvd01/RealTimeMOT/blob/main/YoloV8/Train15) Modèles entraînés sur les version 14 et 15 de la base de données des sailboat
* [\train](https://github.com/lauralvd01/RealTimeMOT/blob/main/YoloV8) Test de Christophe, de l’entraînement et de l’utilisation de [yolov8\_detect.py](https://github.com/lauralvd01/RealTimeMOT/blob/main/YoloV8/yolov8_detect.py)
* [Roboflow - Train a model.pdf](https://github.com/lauralvd01/RealTimeMOT/blob/main/YoloV8/Roboflow%20-%20Train%20a%20model.pdf) Mode d’emploi pour utiliser Roboflow afin de créer une nouvelle base de données et pour entraîner un modèle de deep learning dessus
* Implémentations python de YoloV8 à utiliser selon le [README](https://github.com/lauralvd01/RealTimeMOT/blob/main/README.md#deepsort) du repository

### 

### Application pour Sébastien

[Drive - Laura Levraud](https://drive.google.com/drive/folders/115I249_MXybnibEoBjG5JS5MHDNIe1yj?usp=drive_link)

[Github - Laura Levraud](https://github.com/lauralvd01/Laura)