

Trimaran - Georacing

Laura Levraud - Stage 09/2023 - 01/2024

Optimisation suivi de compétiteurs (vélo et bateaux) en temps réel via tracking 2D vidéo, associé à un tracking GPS.

- Mise en place de tracking 2D en temps réel de patterns dans une vidéo sur la base d'une étude et de la sélection de codes déjà publiés et dispos en open source.
- Associer des positions GPS pour tenter de pré-positionner la zone de pattern à tracker sur un compétiteur donné et d'améliorer la qualité de tracking 2D.

Résultats

Après plusieurs études et mesures de performances de techniques dites de **pixel pointing**, traçant le trajet d'un objet dans l'image au cours d'une vidéo grâce à la recherche de pixels caractéristiques de cet objet, les conclusions sont que le tracking par **pixel pointing** d'une part requiert l'action d'un opérateur afin de sélectionner dans l'image l'objet -ou la zone de pixels- à tracer, d'autre part est trop lent ou trop imprécis pour convenir à un tracking en temps réel, i.e avec un flux vidéo constant à plus de 25 images par secondes.

L'objectif premier étant de permettre le tracking en temps réel, et le second de l'automatiser, ne nécessitant la présence d'un opérateur que pour vérifier le bon déroulement et/ou choisir d'afficher ou non les rendus obtenus pour chaque bateau, le **pixel pointing** a été déclaré inadéquat.

Les résultats obtenus par les méthodes utilisant du **deep learning** étant bien plus encourageants, tant au niveau rapidité que précision, il a été décidé de commencer à développer un **outil** utilisant d'une part la détection par **deep learning** afin de détecter simultanément tous les objets présents dans l'image, et d'autre part un algorithme d'association afin d'attribuer à chaque objet une identité puis de recouper les informations entre chaque image pour pouvoir tracer les objets tout au long de la vidéo.



YoloV8

Pour la partie détection, le modèle **YoloV8** d'[Ultralytics](#) est utilisé. Celui proposé en ligne a été entraîné sur la base de données **COCO (Common Objects)** et ne convient pas aux formes particulières des voiliers de course qui devront être détectés lors des courses du **Vendée Globe** ou des **52 Super Series**. Il pourrait convenir pour détecter les vélos de course d'intérieur à tracer lors des **UCI TCL**, mais pour plusieurs raisons le tracking de ces vélos est beaucoup plus difficile que le tracking des bateaux. Pour le moment, l'entraînement de modèles adaptés s'est donc concentré sur la détection des voiliers.

Une base de données d'environ 13 500 images a été créée sur [Roboflow](#), à partir de vidéos du **Vendée Globe** et de lives enregistrés des **52 Super Series**. Ces images ont été annotées afin de détecter tous les `sailboat` présents. Une fois la base de données exportée, un modèle basé sur le modèle `yo1ov8n.pt` a été entraîné.

C'est ce modèle qui est utilisé, par l'intermédiaire d'un code écrit en C# inclus dans l'outil implémenté.

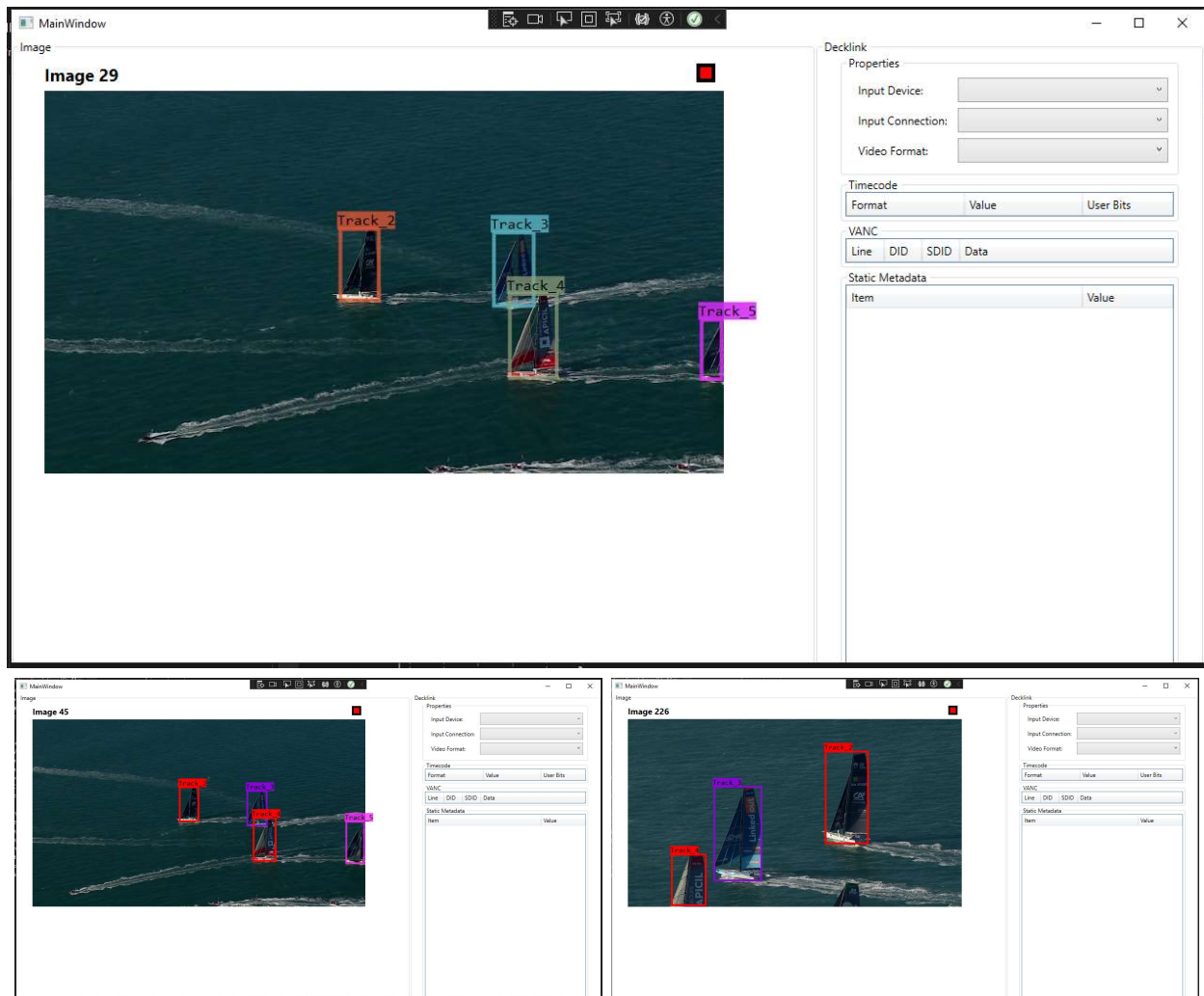


DeepSORT

Pour la partie recoupement des informations entre les différentes images de la vidéo, une version C# de l'algorithme de **DeepSORT** a été adaptée. Elle permet d'associer les détections d'une image à l'autre grâce à un filtre de Kalman et une mesure alliant distance et apparence grâce à un autre modèle de deep learning spécialisé dans l'identification. Une fois qu'un bateau a été détecté sur 3 images d'affilée (nombre modulable), lui est attribué un numéro d'identité et les informations liées à ce bateau sont alors stockées jusqu'à ce qu'il disparaisse de l'image.



Les informations ainsi récupérées permettent de localiser dans l'image, tout au long de la vidéo, tous les bateaux. Grâce à elles, l'outil mis en place peut afficher, sur la vidéo dans l'interface de debug, des rectangles autour des bateaux détectés, désignés par leur numéro d'identité. L'interface, reliée à une interface en ligne qui offre les mêmes fonctionnalités et qui sera à terme celle utilisée pour faire le pont avec les machines de rendu de réalité augmentée, permet également de cliquer sur l'un (ou plus) des bateaux détectés afin de choisir les bateaux pour lesquels les informations seront envoyées à la machine de réalité augmentée.



Structure du dossier

Application pour Sébastien

Sauvegarde du repository public [Github - Interface de commande de renders](#).

DEMOS

Sélection de vidéos résultats des différentes techniques testées.

Dev

README_R&D, récapitulatif de tous les programmes écrits au cours du stage et de l'endroit où on peut les trouver ou trouver leur documentation.

Copie pdf de chacun des README des repository Github créés pendant le stage :

- [Github - Interface de commande de renders \(public\)](#)
- [Github - GeoLiveDetect \(privé\)](#)
- [Github - RealTimeMOT \(public\)](#)

DOC

Documentation sur

- Roboflow et la création de bases de données
- Comment entraîner un modèle YoloV8
- Installer et utiliser l'application pour Sébastien

et liens vers les README du dossier [Dev](#).

Recherches

Ensemble des pistes creusées, des notes prises pour chacune et des tests commentés réalisés au cours du stage.

GEORACING - Pixel Tracking

Table résumant sommairement les techniques testées et leur appréciation.

README_Trimaran_Georacing

Le présent README, compilé avec grip puis imprimé en pdf. Pour visualiser les gifs, voir le dossier éponyme dans le dossier [DEMOS](#). Pour avoir accès aux liens, regarder le code correspondant dans le fichier .md.