# Rapport TP3


Détection et suivi d'un objet d'intérêt

INF6804 – Vision par ordinateur

Travail présenté à Guillaume-Alexandre Bilodeau

Hiver 2023

Département de génie informatique et génie logiciel
École Polytechnique de Montréal

Dernière mise à jour : 16 avril 2023 à 17:48:48

Guillaume Girouard 1954899
Pierre-Emmanuel Rebours 2165286

# Table des matières

# 1. Introduction

Lors de ce laboratoire, il est demandé aux étudiants de réaliser une expérience de suivi d'objet multiple sur une séquence d'images personnalisées fournie par le professeur. De cette expérience, l'ensemble des étudiants sera classé sous forme de compétition par rapport à leur méthode soumise évaluée sous la métrique HOTA.

Dans ce présent document, un cadre plus théorique comprend d'abord une explication approfondie du modèle choisi par l'équipe, une description et une énumération des difficultés comprises dans la séquence vidéo fournie et la justification du choix de la méthode basé sur l'énumération des difficultés.

Par la suite, le document traite un cadre plus pratique présentant la description de l'implémentation réalisée, la présentation des résultats et l'analyse des performances de l'algorithme sur ces résultats.

Les résultats du suivi seront joints à ce document lors de la remise sous format texte pour participer à la compétition générale.

# 2. Description en profondeur de la solution

## 2.1. Présentation de YOLOv5

La méthode YOLO (You Only Look Once) est un 'one stage object detector' entrainé sur un unique réseau de neurones convolutif qui se veut être rapide sans pour autant perdre en précision. L'algorithme décompose l'image en grille et l'algorithme prédit des boites englobantes pour chaque région associée à ces quadrilles. Son architecture repose sur trois composantes principales, soit un backbone pour l'extraction d'information de l'image, un 'Neck' pour être robuste au changement de taille ou d'échelle d'un objet et une tête qui s'occupera de la prédiction des classes, des boites englobantes et du score de certitude. L'image suivante aide à visualiser l'architecture.

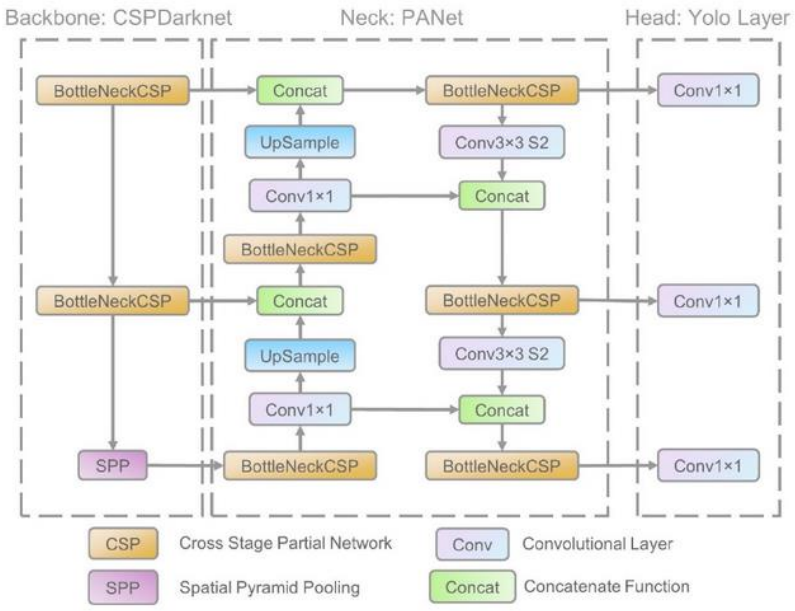



*Figure 2.1-1.* Architecture YOLOv5

Source https://iq.opengenus.org/yolov5/

Pour yolov5 présentement utilisé dans ce laboratoire. On aura que le backbone est représenté par un modèle de réseau préentrainé servant à extraire les informations pertinentes des images mises en entrée. À chaque niveau de profondeur du CNN, on aura une perte d'informations spatiale associée à l'image. Pour limiter ces dégâts, ce dernier prend l'architecture CSP-Darknet53. Cette architecture propose un CSPNet (Cross Stage Partial Network) qui permet de conserver les avantages d'un réseau densément connecté tout en contournant les effets de gradient redondant. Entre autres, une pratique commune pour acheminer correctement un contenu informationnel élevé dans les couches les plus profondes du réseau est d'utiliser de dense architecture de connexion entre chaque couche. Cependant, cette technique peut apporter des problèmes de gradients redondants et ainsi rendre moins efficace l'entrainement d'un modèle que voulu. Une façon de contourner ce problème est l'utilisation de CSP. Ces derniers vont tronquer en deux parties les informations de la couche précédente vers la suivante pour avoir d'une part une partie de l'information qui aura une connexion dense vers les autres couches et d'autre part une partie de l'information moins connectée comme le montre l'image suivante.

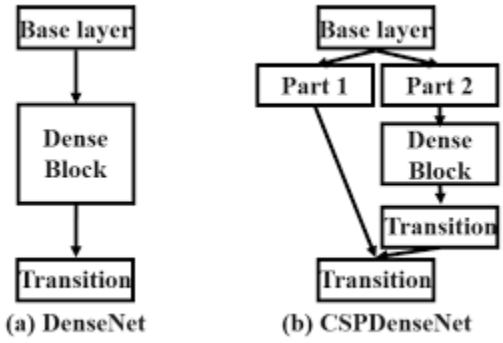



*Figure 2.1-2.* Graphe de concept CSP

Une fois les informations extraites du backbone, YOLOv5 implémente un modèle de 'Neck' basé sur l'utilisation de SPP(Spatial Pyramid Pooling) et d'un PANet(Path Aggregation Network). La partie du path aggregation network sert à fusionner les sorties de différentes couches pour raffiner la résolution spatiale et sémantique à rendre à la tête du réseau. Le spatial pyramid Pooling est un bloc assurant l'acheminement de l'information vers l'entrée de la tête sur une résolution fixe. Entre autres, elle se base sur la combinaison de plusieurs convolutions dilatées entre les échantillons pour permettre une récupération spatiale des couches du backbone avec une plus grande précision de l'information spatiale sans pour autant augmenter le temps de traitement de la méthode.

Finalement, la tête du réseau est composée de trois couches convolutionnelles qui s'occupent de la prédiction des dimensions des boites englobantes, de la classe des objets et du score de certitude de la détection d'un objet sur l'image. Les équations pour trouver les bonnes informations de la boite englobantes sont les suivantes :

$$b_x = (2 \cdot \sigma(t_x) - 0.5) + c_x$$

$$b_y = \left(2 \cdot \sigma\left(t_y\right) - 0.5\right) + c_y$$

$$b_w = p_w \cdot \left(2 \cdot \sigma(t_w)\right)^2$$

$$b_h = p_h \cdot \left(2 \cdot \sigma(t_h)\right)^2$$

Pour l'entrainement du réseau, la fonction de perte est représentée par la somme pondérée. On possède une fonction 'Binary cross Entropy' pour prédire la classe et la présence d'objet ainsi qu'une fonction de 'Complete Intersection over Union' pour la localisation de la boite englobante.

Au final, le détecteur yolov5 possède des avantages sur le fait que la méthode est: rapide, précise, facile d'intégration avec le format du laboratoire, préentrainé sur COCO, légère, possède architecture réunie à PyTorch et est associé par défaut au traqueur OC-SORT utilisé dans ce laboratoire.

Cependant, la méthode possède des désavantages. Entre autres, elle possède des difficultés quand l'objet est trop petit ou quand il y a une trop grande concentration d'objet très rapproché dû à son principe de décomposition de l'image en quadrillé de grille.

## 2.2. Présentation de la méthode OC-SORT

OC-SORT (Observation-Centric SORT) est un détecteur multiobjet basé sur un modèle de mouvement. C'est une méthode qui augmente les performances de la méthode SORT: elle se veut robuste à l'occlusion et au mouvement non linéaire, 2 problèmes qui impactent grandement les performances des traceurs traditionnels.

SORT repose sur l'utilisation d'un filtre de Kalman: C'est un filtre qui estime de façon linéaire l'état d'un système dynamique qui a été discrétisé dans le temps. Il est divisé en 2 étapes. Dans l'étape de prédiction, on estime l'état (ie position de l'objet) au temps présent à partir de l'estimation de l'état précédent ainsi que la matrice de covariance de l'erreur.
Il y a également une étape de mise à jour qui corrige l'état prédit précédemment à l'aide des observations (fournies par une méthode de détection) dans le temps présent et qui met à jour les paramètres du filtre:

$$predict \begin{cases} \hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} \\ P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \end{cases}$$

$$update \begin{cases} K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1} \\ \hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (z_t - H_t \hat{x}_{t|t-1}) \\ P_{t|t} = (I - K_t H_t) P_{t|t-1} \end{cases}$$

Avec :

- $\hat{x}_{t|t-1}$, état au temps présent à partir de l'estimation de l'état précédent
- $F_t$, modèle de transition d'état
- $H_t$, modèle d'observation
- $Q_t$, bruit lié à la prédiction
- $R_t$, bruit lié à la prédiction
- $K_t$, paramètre à postériori de filtre
- $z_t$, observation

Dans le cas de SORT, un état x et une observation z sont définis de la façon suivante:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$$

$$z = [u, v, w, h, c]^T$$

Avec :

- $(u, v)$, les coordonnées du centre de l'objet dans l'image
- $s$, l'aire de la boite englobante
- $r$, le rapport d'aspect de la boite englobante (supposé constant)
- $w, h$, la largeur et la taille de l'objet
- $c$, le seuil de confiance de la détection

Le problème de ce filtre, ce n'est pas temps son approximation du mouvement entre deux frames comme étant linéaire, mais c'est plutôt un problème qui survient lorsqu'il n'y a pas d'observation à une frame: objet caché, objet non détecté…. On ne peut alors que ce fier à l'estimation de la première phase du filtre ce qui peut conduire à des erreurs en cas de mouvement non linéaire. SORT présente d'autre limitations que tente de corriger OC-SORT, mais toute prenne leur importance lorsque l'on n'arrive pas à observer l'objet pendant plusieurs frames car elle résulte en une accumulation d'erreur de l'estimation qui n'a pu être corrigée par des observations.
OC-SORT va corriger cela en rajoutant une étape appelée ORU (Observation-centric Re-Update): Si une observation d'une cible est disponible de nouveau après ne pas avoir été détecté un certain temps, on réitère une boucle de l'algorithme sur la période de temps où l'on n'a pas eu d'observation. Pour cela on tire des "observation" d'une trajectoire virtuelle entre le moment où la cible a été observée pour la dernière fois et le moment où elle a de nouveau été détectée. La boucle exécutée est composée de la phase de prédiction évoquée plus haut suivie de la phase "re-update":

$$re-update \begin{cases} K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1} \\ \hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(\tilde{z}_t - H_t \hat{x}_{t|t-1}) \\ P_{t|t} = (I - K_t H_t) P_{t|t-1} \end{cases}$$

Avec : $\tilde{z}_t = Traj_{virtual}(z_{t_1}, z_{t_2}, t)$ où $t_1 < t < t_2$ et $z_{t_1}/z_{t_2} est\ la\ dernière/nouvelle\ observation$ de l'objet.

En plus de ORU, un terme a été rajouté dans le calcul la matrice de coût pour la phase d'association des traceurs aux objets détectés: il s'agit d'un terme qui calcule la constance (ici sous la forme d'une différence $\Delta\theta$ ) entre la direction liant 2 observations d'un suivi existant et la direction liant une observation d'un suivi avec une nouvelle observation comme illustré dans la figure ci-dessous (l'article parle d'OCM pour Observation-Centric Momentum). Les nouvelles observations sont représentées par des points rouges. La ligne verte indique un suivi existant. Les deux lignes jaune et bleu représentent respectivement les directions évoquées plus haut:

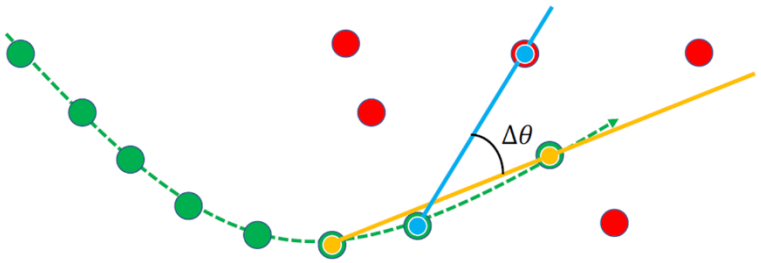



*Figure 2.2-1. Schéma de concept ORU [2]*

On a donc une matrice de coût calculé de la façon suivante:

$$C(\hat{X}, Z) = C_{IoU}(\hat{X}, Z) + \lambda C_v(\mathcal{Z}, Z)$$

Avec :

- $\hat{X} \epsilon \mathbb{R}^{N \times 7}$, l'ensemble des états d'un objet sur N frame
- $Z \epsilon \mathbb{R}^{M \times 5}$, l'ensemble des M observation(ie detection) fait à la frame N+1
- $\mathcal{Z}$, qui contient la trajectoire de toutes les observations fait jusqu'à présent
- $C_{IoU}(\hat{X}, Z)$, calcul l'IoU (Intersection over Union) par pair
- $C_v$, qui contient toutes les paires $\Delta\theta = \left|\theta^{track} - \theta^{intention}\right|$ où $\theta^{track}$ est la direction liant deux observations d'une trajectoire existante et $\theta^{intention}$ est celle liant une observation d'une trajectoire existante à une nouvelle observation

Enfin les auteurs rajoutent une heuristique, appelée OCR (Observation-Centric Recovery) afin de répondre au problème d'objet caché pendant une petite période de temps. Elle va tenter de faire l'association entre les nouvelles observations et la dernière observation de l'objet qui a été brièvement caché.

La méthode est assez récente et ne possède pas encore une analyse de performance bien définie. Cependant, la méthode essaie de corriger les lacunes de la méthode SORT et de l'utilisation d'un filtre de Kalman, sans complètement les régler. L'association est basée uniquement sur le mouvement ce qui peut poser problème si la prédiction n'est pas correcte. D'autres indices pourraient être rajouté comme l'apparence de l'objet: c'est ce qui a été fait dans la méthode Deep OC-SORT [8].

## 2.3 présentation de la métrique HOTA

La métrique HOTA est une métrique conçue dans l'objectif de répondre aux limitations de certaines métriques couramment utilisées en MOT comme MOTA. Elle mesure au même niveau la précision dans la détection ainsi que celle de l'association là où d'autres métriques mesurent davantage un aspect que l'autre.
En parallèle des concepts de TP (True Positive), FN (False Negative),FP(False Positive) propre à la détection d'objet, la métrique HOTA introduit de nouveaux concepts qui leur sont similaires, mais cette fois si orientés vers l'association d'objet lors du suivi. Pour un vrai positif (TP) donné :

- TPA (True Positive Association) est l'ensemble des vrais positifs qui ont, en prédiction et dans la vérité terrain, le même id que le vrai positif.
- FNA (False Négative Association) est l'ensemble des faux négatifs qui ont dans la vérité terrain le même id que le vrai positif, mais qui ont en prédiction un id différent ou pas d'id (car l'objet n'a pas été détecté).
- FPA (False Positive Association) est l'ensemble des Faux positifs qui ont en prédiction l'id du vrai positif, mais qui n'est pas le leur ou qui ne correspond à aucun objet dans la vérité terrain

Le schéma ci-dessous, tiré de l'article [5] qui introduit cette métrique, rend visible les définitions précédentes :

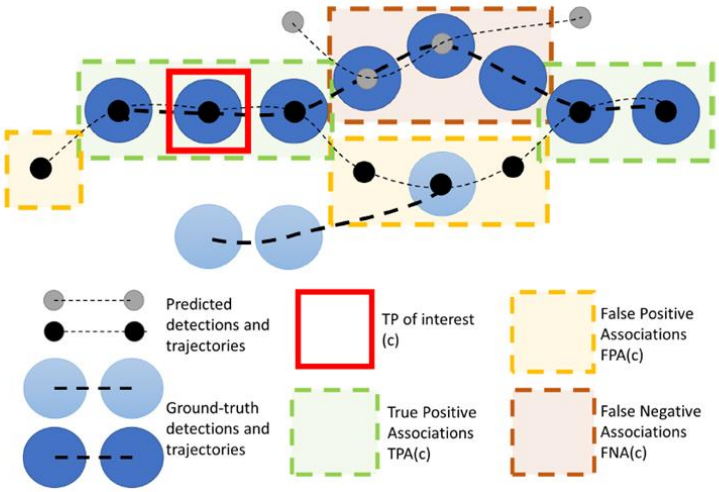



*Figure 2.3-1* Schéma de concept HOTA

La métrique prend la forme suivante :

$$\mathrm{HOTA}_\alpha = \sqrt{\frac{\sum_{c\in\{\mathrm{TP}\}} \mathcal{A}(c)}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|}}$$

$$\mathcal{A}(c) = \frac{|\mathrm{TPA}(c)|}{|\mathrm{TPA}(c)| + |\mathrm{FNA}(c)| + |\mathrm{FPA}(c)|}$$

Où $\alpha$ est le seuil à partir duquel, si une détection a un IoU entre sa boite englobante et celle de la vérité terrain plus grand que ce seuil, alors on considère que c'est un vrai positif.

## 3. Identification des difficultés de la vidéo donnée:

Le jeu de données fourni par ce laboratoire présente un suivi d'objet multiple de plusieurs tasses dispersées un peu partout dans un bureau. La caméra est tenue par une personne qui fait l'action de prendre une tasse isolée, de replacer cette dernière avec les autres, d'enlever rapidement une autre tasse du champ de vision de la caméra et de la replacer, de verser de l'eau d'un verre à un autre devant une tasse, et finalement, de faire un gros plan sur une autre tasse isolée se retrouvant au sommet d'un ordinateur.

De ce jeu de données, il est possible d'énumérer les difficultés potentielles et de faire un point sur celles devant être priorisées face à la méthode de suivi à choisir.

*Table 3-1 tables de description des problèmes de la séquence Moodle*

| Problème | Frames problématiques | Description du problème |
|---|---|---|
| **Caméra non fixe** | Toutes | Mouvement non linéaire et imprévisible |
| **Occlusion totale** | Frame 265 à 485 et 1035 à fin | Perte totale de l'instance d'un objet déjà identifié dans l'image |

| | | |
|---|---|---|
| **Déformation** | 565 à 1010 | Déformation de la tasse par réfraction avec l'eau ou le verre |
| **Translation** | Toutes | Déplacement des instances d'objets identifiés |
| **Rotation hors plan** | Toutes | Changement de profil des objets au cours du temps |
| **Couleur différente** | Toutes | Objets aux couleurs différents |
| **Changement d'échelle** | Frame 56 à 230 et 1050 à fin | Grossissement et rapetissement de l'objet |
| **Forme différente** | Toutes | Tasses aux formes différentes |
| **Occlusion partielle** | Toutes | Occlusions des tasses soit par un autre objet soit par les rebords de l'image |
| **Images floues** | Toutes | Par le focus de la caméra, les objets plus à l'avant sont plus flous que ceux en arrière |
| **Changement de luminosité** | 1300-fin | L'ombre de la personne qui filme obscurcit légèrement la tasse |
| **Concentration d'objet** | 550-1050 | Rapprochement ou superposition d'objets |
| **Coupure de frame** | 981-996 | Frames manquantes |

De cette liste, il est à noter que les éléments les plus susceptibles de nuire aux performances de suivit risque d'être le problème de caméra non fixe, les occlusions totales des objets pour réapparaitre plus tard dans la vidéo, les occlusions partielles et le changement d'échelle des objets.

## 4. Justification de la méthode par rapport à la séquence

En lien avec la section précédente, une justification de l'approche de résolution du problème est donnée dans cette partie.

La section précédente a, entre autres, permis de guider la recherche de solution vers une méthode répondant aux problèmes soumis. Une première idée soulevée pour le choix d'une méthode est de premièrement trouver un dataset similaire à notre séquence et de concentrer la recherche sur les algorithmes ressortant au sommet du classement d'après la métrique HOTA. La supposition suivante est alors soumise : le problème de caméra non fixe et de changement d'échelle de plusieurs objets dans une séquence peut être similaire à des séquences de conduite autonome. Effectivement, dans le cas de la conduite autonome, la caméra est fixée sur une voiture et bouge par rapport aux directions que prend la voiture. De plus, les autres voitures de la séquence peuvent se rapprocher de la caméra ou s'en éloigner et même disparaitre et réapparaitre dépendamment de la circulation et de la destination de chacune des voitures. Choisir ce type de jeux de données peut alors être justifiable face aux problèmes à traiter au vu des problèmes relevés dans la partie 3. Des séquences de conduite autonome populaire et disponible sur internet, la séquence 2D bounding Box KITTI, car est un choix reflétant bien la description énumérée. Des performances de suivi reliées à ce dataset, la méthode OC_SORT est celle qui est au sommet de ce classement autant sur la métrique MOTA que la métrique HOTA comme le montre l'image suivante :

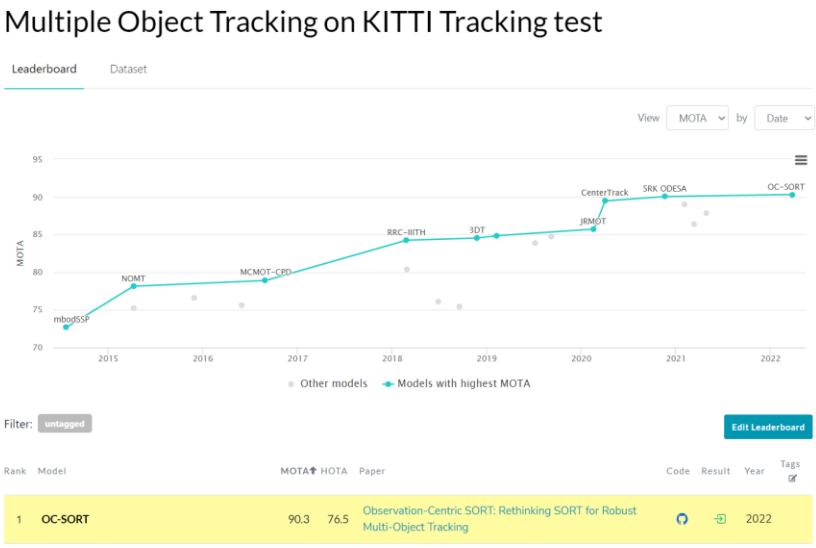



*Figure 4.1-1.* Classement des méthode MOT de KITTI Tracking test sous MOTA

Cette méthode est donc celle considérée pour résoudre la séquence donnée du laboratoire.

OC-SORT est une méthode qui se veut résiliente aux occlusions et au mouvement de caméra non linéaire. De ce fait, elle répond bien à des problèmes majeurs relevés en partie 3.
L'approximation des mouvements entre deux frames comme étant linéaire n'est pas une aberration, le problème survient lorsqu'il y a des occlusions. OC-SORT utilise un filtre de Kalman. Des occlusions entrainent la propagation d'erreur dans ce filtre, rendant la prédiction d'état erroné. C'est pour cela qu'OC-SORT rajoute une troisième phase de mise à jour (ORU) qui s'exécute en cas d'occlusion avérée.
Ces occlusions arrivent dans le jeu de donnée du TP. Pour avoir une prédiction de la trajectoire plus robuste, OC-SORT s'assure de la cohérence des directions lors de la phase d'association des détections (OCM). Il est légitime de se dire que, de cette façon, OC-SORT sera plus robuste aux mouvements parasites de l'objet qui sont liés à ceux de la caméra.
Pour retrouver des objets occultés brièvement, comme certains objets de la vidéo du TP, OC_SORT utilise son heuristique OCR (cf. partie 2.2)

OC_SORT est utilisé avec le détecteur YOLOv5. Cette méthode de détection possède de bonnes performances de précision sur l'apparition des boites englobantes générée. De plus, elle possède un bloc exerçant du SPP sur les extractions d'informations des objets, ce bloc de SPP permet sur ce point d'acquérir une meilleure information spatiale de l'objet et rendrait la détection d'objet sur plusieurs échelles plus robuste. Ainsi, le détecteur contournera les changements d'échelles des objets sur la séquence et la translation. De plus, il s'agit d'une intelligence artificielle entrainer sur le jeu de donnée COCO ce jeu de donnée comporte énormément d'images de différents types de tasse. Cet entrainement devrait alors faciliter la détection de tasse sur plusieurs plans comprenant des rotations en plan et hors plan en plus de faciliter la détection de tasses sous des formes et des couleurs variées. Cependant, cette méthode de détection repose sur un principe de grille quadrillé. Cette grille entraine potentiellement des problèmes lorsqu'une grosse concentration d'objet se retrouve dans le même carrée de grille ou lorsqu'un objet se retrouve couper entre deux grilles. La méthode pourrait dans ce cas continuer à avoir des problèmes pour identifier une grande concentration de plusieurs d'objets situés loin dans un arrière-plan. Cet effet reste présent dans la séquence, mais ne le présente pas de manière significative.

# 5. Description de l'implémentation

Le code utilisé repose en grande partie sur des implémentations existantes des méthodes.

L'implémentation d'OC-SORT est tirée d'une version packagée (https://github.com/kadirnar/ocsort-pip) du dépôt officiel ( https://github.com/noahcao/OC_SORT)

On a plusieurs hyper paramètres. Les plus importants sont deux seuils :

- det_tresh : il s'agit du seuil de confiance de détection. Pour les détections qui ont une confiance de détection supérieure à ce seuil, elles seront associées entre elles lors d'un premier matching.
  Si elles sont inférieures à cette limite, mais supérieures à 0.1, elles seront associées lors d'un second matching, séparément des premières. L'article dit suivre les pratiques courantes avec SORT en prenant une valeur entre 0.4 et 0.6 selon le dataset. Nous choisirons donc 0.5 en sa nature de juste milieu.
- iou_threshold : il s'agit du seuil de confiance pour l'association : une association aura lieu si son niveau de confiance est supérieur à ce seuil. Une fois encore nous suivrons les recommandations de l'article en gardant une valeur de 0.3

Il y a également le poids associé à la cohérence de la direction d'un objet entre deux frames dans le calcul du coût d'association. L'article mentionne choisir une valeur de 0.2. Cette valeur sera gardée.

Pour la métrique HOTA, nous avons utilisé l'implémentation de TrackEval en exécutant la commande suivante:

!python TrackEval/scripts/run_mot_challenge.py --BENCHMARK MOT17_tp --SPLIT_TO_EVAL train --TRACKERS_TO_EVAL OC-SORT --METRICS HOTA --USE_PARALLEL False --NUM_PARALLEL_CORES 1

Cette commande récupère les fichiers .txt qui contiennent les résultats pour notre traceur et celle de la vérité terrain pour pouvoir, à l'aide du script run_mot_challenge.py, déterminer les performances de OC-SORT selon la métrique HOTA et HOTA(0)

MOT17_tp est un dataset qui contient les séquences de MOTS suivantes: MOT17-02-DPM, MOT17-04-DPM, MOT17-05-DPM, MOT17-09-DPM, MOT17-10-DPM, MOT17-11-DPM, MOT17-13-DPM.

TrackEval doit être téléchargé localement dans l'environnement de travail pour être utilisé.

Ce que nous avons implémenté réside principalement dans des fonctions et des classes utilitaires pour pouvoir récupérer les données et générer les résultats.
Entre autres, un module du non de 'dataloader.py' est implémenté par l'équipe pour, dans un premier temps, récupérer les images d'une séquence sous format 'Mat' par rapport à l'index de l'image courante à traiter, et dans l'autre, inscrire les résultats du suivi d'objet dans un fichier texte sous le format requis des analyses MOT et du TP.

Le détecteur Yolov5 utilisé est tiré de la librairie PyTorch disponible au lien suivant :

https://pytorch.org/hub/ultralytics_yolov5/

Ce détecteur provient d'une intelligence artificielle entrainée sur le jeu de donnée COCO. Les concepteurs de yolov5 proposent sur leur site plusieurs complexités de configuration de modèle pour répondre à un plus grand cas d'utilisation de leurs algorithmes. De ces complexités, il est possible de prendre du format Nano permettant un temps d'exécution très vite, mais un score de précision assez médiocre, ou la complexité XLarge permettant un temps d'exécution plus lent, mais répondant à de plus hauts standards de précision de la détection. Parce que

notre séquence à traiter ne requiert pas de performance en temps d'exécution particulière, le choix du modèle YOLOv5x6 est alors favorisé.

## 6. Présentation des résultats

Notre configuration MOT est alors testée sur l'ensemble des données de Motchallenge disponibles au lien suivant :

https://motchallenge.net/data/MOT17/

Plus précisément, ces ensembles de données sont décrits dans le tableau qui suit :

*Table 6-1.* Description des séquences prises de MOT17

| Nom de la séquence | Nombre d'images | Description | Difficulté observée associée |
|---|---|---|---|
| **MOT17-02-DPM** | 600 | Personne marchant autour d'un large carré | Concentration d'objet |
| **MOT17-04-DPM** | 1050 | Piétons marchant dans la rue la nuit vue en hauteur | Concentration d'objet, occlusion partielle |
| **MOT17-05-DPM** | 837 | Scène de rue filmée d'une caméra mobile | Caméra non fixe, gros plan de l'objet, concentration d'objet, Occlusion partielle |
| **MOT17-09-DPM** | 525 | Piétons filmés dans une scène de rue pris par un angle bas | Gros plan de l'objet, collision d'objet et occlusion |
| **MOT17-10-DPM** | 654 | Scène de piéton pris la nuit par une caméra mobile | Caméra non fixe, concentration d'objet, obscurité, images floues |
| **MOT17-11-DPM** | 900 | Caméra bougeant vers l'avant dans un centre commercial | Caméra non fixe, concentration d'objet<br>Occlusion partielle |
| **MOT17-13-DPM** | 750 | Scène filmée par un bus dans une intersection occupée | Caméra non fixe, concentration d'objet |

Pour ces séquences, les résultats suivants sont capturés :

*Table 6-2.* Présentation des scores de HOTA pour les séquences MOT17 testées

| Séq | HOTA | DetA | AssA | DetRe | DetPr | AssRe | AssPr | LocA | OWTA | HOTA(0) | LocA(0) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **MOT17-02-DPM** | 31.414 | 19.568 | 50.553 | 19.937 | 84.89 | 52.86 | 88.433 | 86.617 | 31.729 | 36.477 | 83.803 |
| **MOT17-04-DPM** | 32.256 | 22.35 | 46.623 | 22.876 | 84.615 | 48.135 | 89.744 | 86.629 | 32.647 | 38.713 | 82.89 |
| **MOT17-05-DPM** | 42.515 | 41.532 | 43.671 | 44.625 | 74.022 | 51.255 | 67.652 | 80.447 | 44.137 | 58.689 | 73.718 |
| **MOT17-09-DPM** | 48.878 | 51.611 | 46.433 | 54.468 | 81.518 | 50.938 | 79.417 | 84.192 | 50.283 | 61.72 | 80.029 |
| **MOT17-10-DPM** | 35.615 | 33.355 | 38.125 | 34.892 | 76.238 | 42.407 | 76.028 | 80.71 | 36.462 | 47.514 | 75.116 |
| **MOT17-11-DPM** | 52.606 | 51.455 | 54.131 | 55.897 | 79.71 | 60.666 | 79.606 | 85.395 | 54.961 | 65.268 | 80.243 |
| **MOT17-13-DPM** | 30.081 | 21.038 | 43.226 | 21.617 | 78.745 | 45.502 | 79.72 | 82.668 | 30.528 | 36.965 | 78.453 |
| **Combinées** | 36.311 | 28.191 | 46.997 | 29.245 | 80.815 | 50.712 | 83.16 | 84.48 | 37.033 | 44.823 | 79.899 |

Les données suivantes sont aussi données pour mieux séparer les performances de détection face à celles de suivi :

*Table 6-3. Table du recensement des détections et identifications des séquences MOT17*

| Séquence | Détections | Détections GT | Identifications | Identifications GT |
|---|---|---|---|---|
| MOT17-02-DPM | 4364 | 18581 | 50 | 62 |
| MOT17-04-DPM | 12857 | 47557 | 76 | 83 |
| MOT17-05-DPM | 4170 | 6917 | 120 | 133 |
| MOT17-09-DPM | 3558 | 5325 | 88 | 57 |
| MOT17-10-DPM | 5876 | 12839 | 88 | 57 |
| MOT17-11-DPM | 6617 | 9436 | 90 | 75 |
| MOT17-13-DPM | 3196 | 11642 | 74 | 110 |

La courbe de la performance en fonction du coefficient $\alpha$ choisit est la suivante :

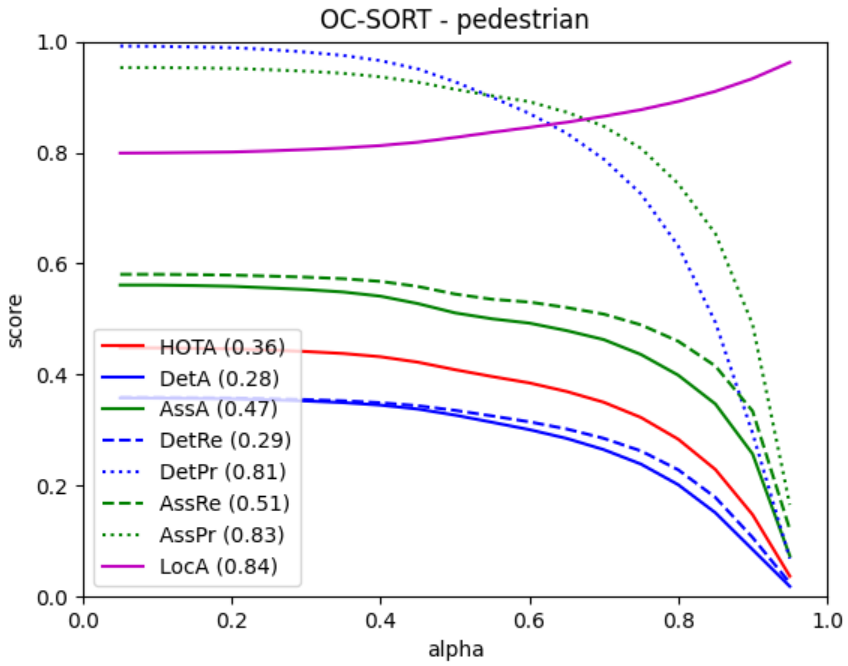
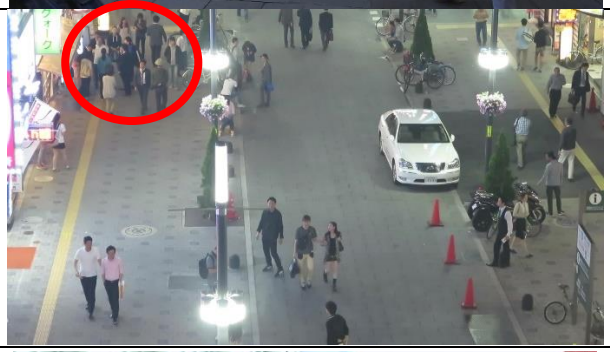
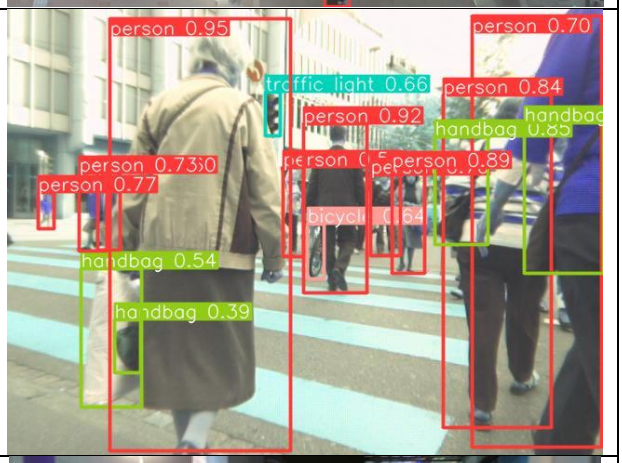
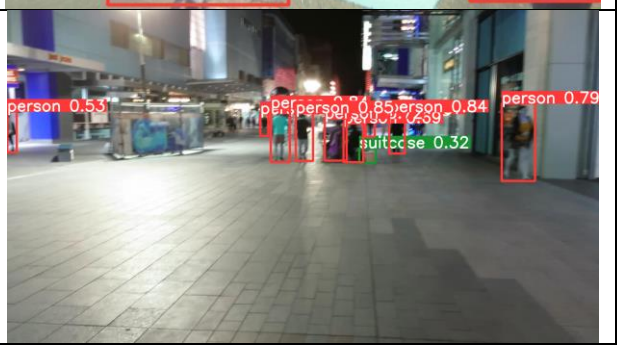
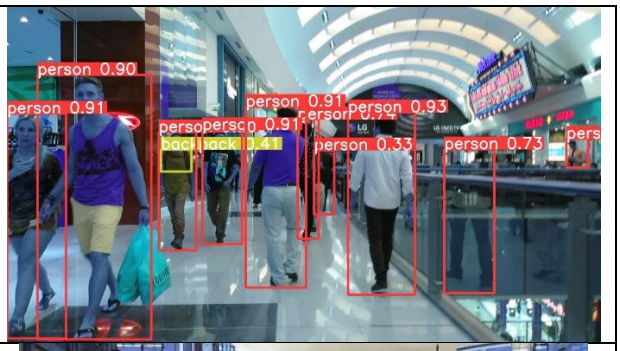
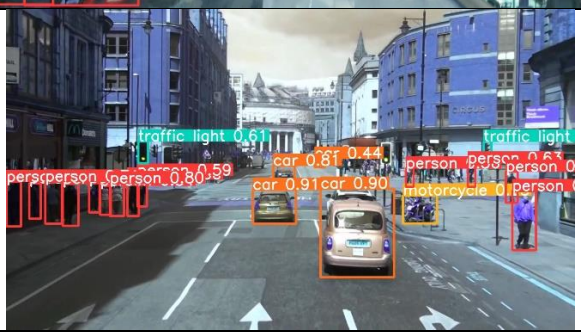



*Figure 6-1.* Graphe de variation des scores HOTA en fonction de $\alpha$

Finalement, un tableau ciblant les problèmes que l'on souhaite contourner avec notre méthode est présenté pour aider la validation.

*Table 6-4.* Tableau d'analyse des problèmes

| Séquence | Problèmes ciblés | Frame original | Résultat de la détection |
|---|---|---|---|
| 02 (img492) | Concentrations d'objets |  |  |
| 04 (img492) | Concentrations d'objets |  |  |
| 5 (img250) | Gros plan d'objet |  |  |
| 10 (img030) | Image floue |  |  |

| | | | |
|---|---|---|---|
| 11 (img250) | occlusion |  |   |
| 13 (img100 | Concentrations d'objets |  |   |

## 7. Analyse des résultats

Cette section se dédit à l'analyse des résultats présentée dans la section précédente

Premièrement, on remarque que HOTA(0) est beaucoup plus permissif que HOTA, car le seuil $\alpha$ (cf. partie 2.3) est de 0. De ce fait, il suffit que la boite englobante prédite touche la boite englobante de la vérité terrain pour que la détection soit considérée comme vrai positif. Là où, pour une valeur de [alpha] plus grande, il aurait fallu que les boites englobantes partagent une aire commune. Ce phénomène peut aussi être observé en regardant le graphique de la section précédente. De ce graphique, il est notamment possible de voir un début de diminution des performances du score de HOTA à partir d'un alpha d'environ 0.5.

On remarque par la suite que les séquences : MOT17-02-DPM, MOT17-04-DPM, MOT17-10-DPM et MOT17-13-DPM sont celles ayant obtenue les résultats HOTA les plus catastrophiques. Pour ses séquences, il est possible de directement lier ces performances à la détection. En effet, il est à remarquer déjà une grosse différence entre le nombre d'objets détectés et le nombre d'objets détectés de la vérité terrain retourné par le coefficient DetA. Effectivement, les séquences MOT17-02-DPM, MOT17-04-DPM et MOT17-13-DPM ont plus de 4 fois moins de détection recensée que celles de la vérité terrain et MOT17-10-DPM en a plus de deux fois moins. Ce problème est aussi observable par la comparaison des coefficients DetRe et DetPr. On nous indique qu'on possède un haut taux de précision pour la validité des vrais positifs, mais que le 'recall' nous montre qu'on arrive à détecter peu de positif dans l'ensemble.

En regardant similairement le tableau d'analyse des problèmes, on remarque effectivement que des objets sont mal détectés pour ces séquences. Comme supposé, une dense superposition d'objets se situant à l'arrière-plan est difficilement identifiable pour YOLO. La séquence MOT17-04-DPM montre que la masse de personnes situées dans le coin haut gauche de l'image n'est pas correctement détectée. De plus, les piétons situés dans l'arrière-plan de la séquence MOT17-02-DPM ne le sont pas non plus. D'un autre point de vue, le détecteur est capable d'identifier

individuellement chaque personne du groupe au second plan montré dans la séquence MOT17-13-DPM du tableau d'analyse des problèmes, mais n'est pas capable de correctement identifier une masse de personnes en arrière-plan sur le même frame. Il est alors possible de confirmer cette lacune de performance au niveau de la détection.

Un autre point d'intérêt concerne le comportement de notre méthode face à l'utilisation d'une caméra non fixe. Les séquences comportant cette caractéristique sont MOT17-05-DPM, MOT17-010-DPM, MOT17-11-DPM et MOT17-13-DPM. On remarque dans l'ensemble que ces séquences sont majoritairement celles qui se comportent le mieux par rapport à la métrique HOTA. Cependant, il est plus probable de penser que ce gain de performance se rapporte à l'angle de vue de la caméra. Le fait que la caméra soit tenue par une personne impose une vue plus droite des profils des objets facilitant leur détection. Il reste par contre possible d'affirmer que le fait que la caméra soit mobile ne diminue pas les performances de suivi. Un autre effet dénoté en lien avec l'utilisation d'une caméra non fixe est la présence de frame plus floue que les autres entrainer par un mouvement soudain de la caméra. Entre autres, ce phénomène est présenté dans l'échantillon de la détection de la séquence MOT17-010-DPM. Cette frame montre un mouvement flou des personnes à détecter. Cependant, le détecteur reste capable de correctement identifier les objets.

Au niveau des performances de suivi, on remarque qu'on obtient de meilleures performances que la détection pour chacune des séquences en comparant les scores AssA et DetA. On remarque aussi qu'on possède dans l'ensemble de bonnes performances de précisions pour le score AssPr. Ceci indiquerait que l'algorithme serait capable de correctement faire l'association entre les objets qu'il possède. Cependant, on possède au niveau du score AssA des lacunes sur le score de 'recall' AssRe. Ceci indiquerait qu'on possède une moins grande concentration de faux positifs sur l'ensemble total de positifs de la vérité terrain. Or, il est possible encore une fois d'associer ce comportement aux performances de détections. Effectivement, si le score de 'recall' de la détection est problématique et que la détection est directement donnée au suivi, il est compréhensible d'avoir une même tendance pour le suivi.

En outre, il est possible d'affirmer que les résultats à obtenir sur la séquence Moodle seront mieux que ceux obtenus présentement. Effectivement, de l'analyse effectuer, la plus grosse faiblesse de notre approche concerne la présence d'une dense concentration d'objet situé dans l'arrière-plan de l'image. Par exemple, le détecteur ne sera pas capable de détecter les personnes les plus éloignées dans une foule située à plus de 50 mètres de la caméra. Or, ce type de situation n'est pas vraiment présente dans la vidéo du TP. Malgré la présence d'une concentration de tasse dans quelques frames de la séquence, ces dernières sont plus en second plan de l'image et non à l'arrière-plan. On s'attend alors à de meilleures performances de détection.

# 8. Conclusion

En conclusion, lors de ce laboratoire, une introduction au suivi d'objet multiple a été effectuée sous la forme d'une compétition. Plus précisément, une séquence spécifique et une métrique d'évaluation ont été imposées en début de travail. Par ces contraintes, il a fallu identifier les problèmes potentiels de suivit relier à la séquence, trouver et justifier une méthode de suivi répondant aux besoins, l'implémenter et la valider en testant l'approche sur d'autres séquences publiques aux problèmes similaire possible de trouver en ligne. Finalement, le résultat de suivi sur la séquence spécifique imposée doit être soumis au classement final.

Pour notre approche, un essaie de combiner le détecteur yolov5 avec le traqueur OC_SORT a été réaliser pour réduire l'impact des mouvements non linéaires de la caméra, des gros plans effectuer sur les tasses et d'être capable de récupérer le suivit après une occlusion de courte durée. Des résultats obtenus sur des séquences MOT rassemblant ces caractéristiques, il a été possible de confirmer la robustesse à ces facteurs en particulier.

# Bibliographies


[1] Azevedo, P. (2022, June 17). Object Tracking State of the Art 2022 - Pedro Azevedo - Medium. *Medium*. https://medium.com/@pedroazevedo6/object-tracking-state-of-the-art-2022-fe9457b77382

[2] Cao, J. (2022, March 27). *Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking*. arXiv.org. https://arxiv.org/abs/2203.14360

[3] Contributeurs aux projets Wikimedia. (2023). Filtre de Kalman. *fr.wikipedia.org*. https://fr.wikipedia.org/wiki/Filtre_de_Kalman

[4] Imane, C. (2022). YOLO v5 model architecture [Explained]. *OpenGenus IQ: Computing Expertise & Legacy*. https://iq.opengenus.org/yolov5/

[5] Luiten, J., Osep, A., Dendorfer, P., Torr, P. H. S., Geiger, A., Leal-Taixé, L., & Leibe, B. (2021). HOTA: A Higher Order Metric for Evaluating Multi-object Tracking. *International Journal of Computer Vision*, *129*(2), 548–578. https://doi.org/10.1007/s11263-020-01375-2

[6] *PyTorch*. (n.d.). https://pytorch.org/hub/ultralytics_yolov5/

[7] Solawetz, J. (2023). What is YOLOv5? A Guide for Beginners. *Roboflow Blog*. https://blog.roboflow.com/yolov5-improvements-and-evaluation/

[8] Maggiolino, G., Ahmad, A., Cao, J., Kitani, K. (2023). *Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification.* ArXiv.org. https://arxiv.org/abs/2302.11813