

Fourreau Benjamin, Mounir Alaoui Youssef

Rapport TP3

Vision par ordinateur

Suivi multi-objets

Description de la méthode

La méthode que nous avons choisi pour prédire la séquence de donnée utilise DeepSORT comme méthode de suivi multi-objets et YOLO comme méthode de détection d'objets.

Nous avons choisi YOLO pour plusieurs raisons. C'est un modèle avec une vitesse de prédiction très élevée, en partie car le modèle réalise les prédictions de manière globale. Il considère la prédiction des boîtes englobantes et des classes des objets en une seule régression sur l'image entière. Cette particularité lui permet également d'obtenir de très bonne performance de détection. Ensuite, YOLO est très simple à implémenter. Il se couple très facilement à DeepSORT.

Désormais, DeepSORT est une méthode de suivi performante qui combine l'extraction de caractéristiques avec un réseau de neurone pour identifier précisément les objets et l'algorithme de Kalman Filter pour prédire l'emplacement futur des objets. Avec ces prédictions il le compare aux nouvelles détections pour réaliser la mise à jour du suivi. Grâce à son extraction des caractéristiques, le modèle est assez robuste aux occlusions, changement d'éclairage et changement de perspective. Des difficultés que l'on rencontre dans la séquence d'images. DeepSORT est cependant très coûteux et peut avoir du mal dans les situations avec beaucoup d'encombrements.

Nous avons également essayé une version très simplifier d'une méthode de suivi, pour la comparer à DeepSORT. Elle calcul simplement la distance entre deux prédictions à deux instants, si elles sont très proches alors on considère que c'est le même objet.

Identification des difficultés dans séquence

Occlusion, encombrement

Certains objets vont à plusieurs reprises cacher temporairement d'autres objets. Rajoutant de la difficulté dans la détection de suivi. Le modèle ne pourrait plus reconnaître l'objet d'une image à l'autre, si l'objet devient caché. De plus, par moment, les tasses seront très proches les uns des autres.



Image 1 Exemple d'occlusion de la séquence

Variation d'éclairage

La scène change, venant modifier les conditions d'éclairage et donc l'apparence des images. Encore une fois rendant plus difficile la reconnaissance d'objets et donc le suivi.

Objets similaires

Certains étant similaires, il peut être difficile de ne pas les confondre et maintenir à suivi cohérent.



Image 2 Exemple d'objets similaires dans la séquence

Échelle variable

Certains objets sont déplacés dans le champ, modifiant considérablement leur taille. Si cette taille change trop brusquement, le modèle ne pourrait pas maintenir le suivi.

Ci-dessous un exemple d'un changement d'échelle.



Image 3 Exemple d'un changement d'échelle 1



Image 4 Exemple d'un changement d'échelle 2

Déplacement de la caméra

La séquence d'image possède des mouvements de caméra. Lorsque que la caméra se déplace, cela entraîne des changements de perspective, pouvant changer l'apparence des objets.

Petites tailles

Certains modèles possèdent une certaine difficulté à détecter des petits objets. De plus, les petits objets sont naturellement plus difficile à différencier, leurs détails étant moins visibles.

Sortie de l'image

Lorsqu'un objet disparaît de l'image, le modèle doit toujours être capable de le re-identifier s'il revient. Seulement, s'il revient dans une position différente que lorsqu'il a quitté l'image, il peut être difficile de l'assigner au même id.

Justification de la méthode

Ces problèmes peuvent rendre le suivi multi-objets délicats. YOLO et DeepSORT vont nous permettre d'en atténuer certains.

Problèmes résolus

Les versions récentes de YOLO (v8 et v9), ont été entraîné sur des dataset avec des perturbations tel que la variation d'éclairage et variation d'échelle. YOLO est ainsi particulièrement robuste à des variations raisonnables d'éclairage et changement de taille.

DeepSORT ajoutant des informations d'apparence pour le suivi aide grandement à réduire l'impact de l'occlusion et les objets qui se ressemblent. Ces informations proviennent de caractéristiques extraites par un réseau de neurone convolutif pré-entraîné. Les CNN sont très efficace pour apprendre des caractéristiques pertinentes pour la distinction d'objets.

Le filtre de Kalman utilisé par DeepSORT est puissant pour la sortie d'un objet de l'image. Le filtre prédit la nouvelle position de l'objet lorsque ce dernier en sort. Il anticipe ainsi les mouvements de l'objet hors image, pour plus facilement le re-identifier lorsque ce dernier revient.

Problèmes persistants

Des variations extrêmes de changement d'éclairage ou d'échelle reste toujours un défi difficile pour les modèles telle que YOLO. De plus, YOLO par son principe de détection global possède l'inconvénient d'être moins performant pour des petits objets.

Malgré ces outils performant, l'encombrement reste également un problème difficile à traiter.

Description de l'implémentation

Nous avons testé trois différentes configurations. Les deux premières utilisent DeepSORT pour le suivi et YOLO pour la détection. Elle se distingue seulement par la version de YOLO, l'une utilise YOLOv8 et l'autre YOLOv9. La troisième configuration utilise YOLOv9 mais avec un tracker qui réalise le suivi basé sur la distance des objets d'une frame à l'autre.

Pour tous les modèles, nous importons le modèle YOLO du git Ultralytics. On instancie le modèle en lui indiquant la version du modèle. Nous utilisons « yolov8n » et « yolov9c ».

Modèle 1 et 2

Nous utilisons YOLO pour prédire pour chaque image, la position des boîtes englobante et le score de confiance associé à chacune. Nous fournissons ensuite à DeepSORT, les données prédites pour qu'il mette à jour les identifications des objets.

Pour utiliser DeepSORT, il est nécessaire de clone le projet, pour ensuite utiliser leur fonction. Ensuite nous instancions une classe Tracker dans le fichier tracker.py , pour implémenter le suivi avec DeepSORT. Le projet est disponible sur le git [nwojke/deep_sort: Simple Online Realtime Tracking with a Deep Association Metric \(github.com\)](https://github.com/nwojke/deep_sort).

Nous avons choisi d'essayer DeepSORT avec deux modèles YOLO, pour observer l'impact d'un bon détecteur sur les métriques d'HOTA. La version 9 est en générale plus performante que la version 8. L'exemple suivant nous montre que v8 manque certaines détections pour des petits éléments.



Modèle 3

Cette méthode plus simpliste s'appuie également sur les détections fournies par YOLO. De même nous instancions une classe disponible dans le fichier tracker_manu.py. À l'initialisation, elle initialise un dictionnaire pour stocker les points centraux des objets suivis et des ID pour ces objets. La méthode update prend en entrée les boîtes englobantes des objets détectés (sous forme de coordonnées x, y, largeur w, hauteur h) et met à jour le suivi des objets. Elle convertit d'abord ces rectangles en points centraux, calcule ensuite une matrice de coûts basée sur la distance euclidienne entre les nouvelles détections et les objets suivis précédemment, et résout le problème d'affectation pour minimiser le coût global. Les détections sont alors assignées aux objets suivis existants ou considérées comme de nouveaux objets si elles ne correspondent à aucun suivi existant, en fonction d'un seuil de distance. Les objets non assignés à une détection nouvelle sont retirés du suivi.


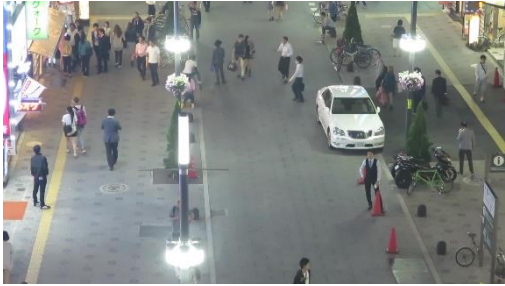

Avec cette méthode plus simpliste, on peut considérer pleinement les performances de DeepSort.

Nous nous sommes inspiré de deux nombreuses implémentations sur internet.

L'évaluation

Nous évaluons les performances des modèles sur la métrique HOTA. Nous utilisons la méthode TrackEval pour cela. De même que pour DeepSORT, nous clonons le projet pour pouvoir ensuite y utiliser ses fonctions. Le projet est disponible sur ce git [JonathonLuiten/TrackEval: HOTA \(and other\) evaluation metrics for Multi-Object Tracking \(MOT\). \(github.com\)](https://github.com/JonathonLuiten/TrackEval).

Nous avons évalué nos modèles sur 4 séquences du dataset MOT17. Nous avons choisi des séquences assez différentes pour tester les différentes hypothèses que nous avons pu faire précédemment.

Dataset	Image exemple	Description
MOT17-02-DPM		Cette séquence possède des objets de très petites tailles, des espaces avec éclairages différents mais relativement sombres et de nombreuses occultations.
MOT17-04-DPM		Cette séquence possède un encombrement de personnes ainsi que des variations de l'éclairages au sein de l'espace.
MOT17-05-DPM		Cette scène possède de l'occultation par le déplacement des voitures, cachant les personnes. Sinon, une séquence assez simple.

MOT17-09-DPM		Cette scène possède des variations d'échelles.
--------------	--	--

Pour pouvoir comparer qualitativement les résultats en plus des métriques HOTA, nous avons implémenter l'affichage des boîtes englobantes des groundtruths.

Voici par exemple, le groundtruth de la première image du dataset MOT17-04-DPM.



Image 5 Groundtruth de la première image du dataset MOT17-04-DPM

Résultats

Nous allons évaluer les performances selon la métrique HOTA, qui est dérivée de différentes métriques :

- DetA (Detection Accuracy) : mesure la précision de détection des objets.
- AssA (Association Accuracy) : mesure la capacité du système à suivre les objets identifiés.
- LocA (Localization Accuracy) : mesure la précision de la position des boîtes englobantes

Nous étudions ces métriques selon plusieurs seuils alpha, correspondant au seuil IoU pour considérer une détection valide.

Résultat globale

Nous avons tout d'abord procédé à une étude globale de nos différents modèles sur les quatre séquences du dataset MOT17. Pour avoir une vue d'ensemble et synthétiser les performances variées, nous nous sommes principalement concentrés sur la métrique HOTA(0), qui permet de résumer les capacités de détection d'objets et de suivi des modèles pour un seuil alpha de 0.05 soit très bas.

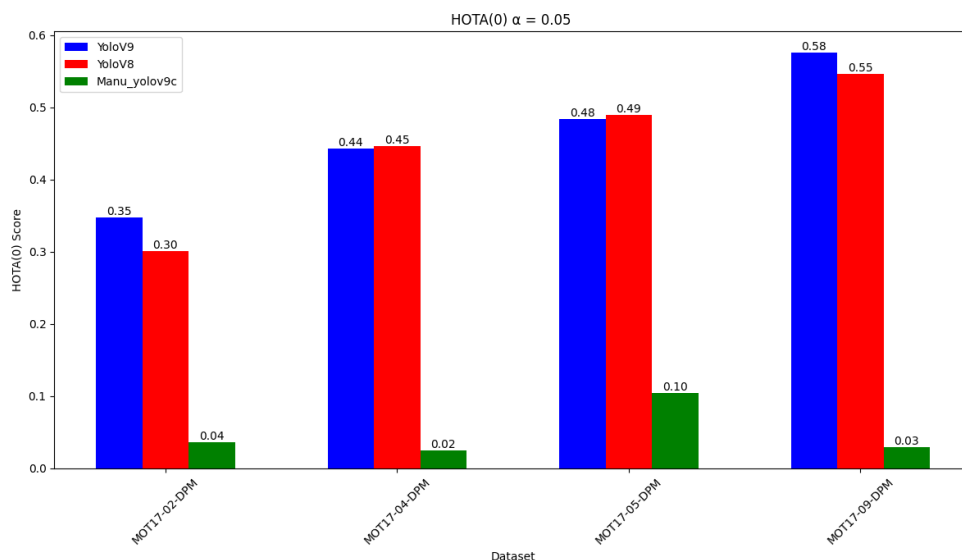


Figure 1 Bar chart des HOTA(0) de chaque modèle son chaque séquence

Le graphique ci-dessus montre une grande disparité de performances HOTA(0) entre les types de modèles. En effet, indépendamment de la séquence, le modèle manuel sans DeepSORT comme système de suivi présente de très faibles performances, avec des scores en dessous de 0.1, tandis que les modèles intégrant DeepSORT affichent de bonnes performances. Selon la séquence, nous observons toutefois un léger avantage pour le modèle utilisant YOLOv9.

Dans un second temps, pour obtenir plus de détails sur les aspects spécifiques sur lesquels nos différents modèles rencontrent des difficultés, nous avons décidé de mesurer l'évolution de nos différentes métriques en fonction de la valeur de alpha sur la séquence MOT17-05. Vous trouverez les résultats dans le tableau ci-dessous :

Model	Performances																																			
DeepSORT- YOLOv9	<p>Tracking Performance for : MOT17-05-DPM - yolov9c.pt</p> <table><thead><tr><th>Alpha</th><th>HOTA</th><th>DetA</th><th>AssA</th><th>LocA</th></tr></thead><tbody><tr><td>0.0</td><td>0.48</td><td>0.55</td><td>0.40</td><td>0.65</td></tr><tr><td>0.2</td><td>0.46</td><td>0.50</td><td>0.42</td><td>0.68</td></tr><tr><td>0.4</td><td>0.44</td><td>0.45</td><td>0.42</td><td>0.72</td></tr><tr><td>0.6</td><td>0.35</td><td>0.35</td><td>0.35</td><td>0.78</td></tr><tr><td>0.8</td><td>0.20</td><td>0.15</td><td>0.25</td><td>0.85</td></tr><tr><td>1.0</td><td>0.00</td><td>0.00</td><td>0.00</td><td>0.95</td></tr></tbody></table>	Alpha	HOTA	DetA	AssA	LocA	0.0	0.48	0.55	0.40	0.65	0.2	0.46	0.50	0.42	0.68	0.4	0.44	0.45	0.42	0.72	0.6	0.35	0.35	0.35	0.78	0.8	0.20	0.15	0.25	0.85	1.0	0.00	0.00	0.00	0.95
Alpha	HOTA	DetA	AssA	LocA																																
0.0	0.48	0.55	0.40	0.65																																
0.2	0.46	0.50	0.42	0.68																																
0.4	0.44	0.45	0.42	0.72																																
0.6	0.35	0.35	0.35	0.78																																
0.8	0.20	0.15	0.25	0.85																																
1.0	0.00	0.00	0.00	0.95																																
DeepSORT - YOLOv8	<p>Tracking Performance for : MOT17-05-DPM - yolov8n.pt</p> <table><thead><tr><th>Alpha</th><th>HOTA</th><th>DetA</th><th>AssA</th><th>LocA</th></tr></thead><tbody><tr><td>0.0</td><td>0.48</td><td>0.55</td><td>0.40</td><td>0.65</td></tr><tr><td>0.2</td><td>0.46</td><td>0.50</td><td>0.42</td><td>0.68</td></tr><tr><td>0.4</td><td>0.44</td><td>0.45</td><td>0.42</td><td>0.72</td></tr><tr><td>0.6</td><td>0.35</td><td>0.35</td><td>0.35</td><td>0.78</td></tr><tr><td>0.8</td><td>0.20</td><td>0.15</td><td>0.25</td><td>0.85</td></tr><tr><td>1.0</td><td>0.00</td><td>0.00</td><td>0.00</td><td>0.95</td></tr></tbody></table>	Alpha	HOTA	DetA	AssA	LocA	0.0	0.48	0.55	0.40	0.65	0.2	0.46	0.50	0.42	0.68	0.4	0.44	0.45	0.42	0.72	0.6	0.35	0.35	0.35	0.78	0.8	0.20	0.15	0.25	0.85	1.0	0.00	0.00	0.00	0.95
Alpha	HOTA	DetA	AssA	LocA																																
0.0	0.48	0.55	0.40	0.65																																
0.2	0.46	0.50	0.42	0.68																																
0.4	0.44	0.45	0.42	0.72																																
0.6	0.35	0.35	0.35	0.78																																
0.8	0.20	0.15	0.25	0.85																																
1.0	0.00	0.00	0.00	0.95																																
Manu – YOLOv9	<p>Tracking Performance for : MOT17-05-DPM - Manu_yolov9c.pt</p> <table><thead><tr><th>Alpha</th><th>HOTA</th><th>DetA</th><th>AssA</th><th>LocA</th></tr></thead><tbody><tr><td>0.0</td><td>0.10</td><td>0.55</td><td>0.02</td><td>0.65</td></tr><tr><td>0.2</td><td>0.10</td><td>0.50</td><td>0.02</td><td>0.68</td></tr><tr><td>0.4</td><td>0.10</td><td>0.45</td><td>0.02</td><td>0.72</td></tr><tr><td>0.6</td><td>0.08</td><td>0.35</td><td>0.02</td><td>0.78</td></tr><tr><td>0.8</td><td>0.05</td><td>0.15</td><td>0.02</td><td>0.85</td></tr><tr><td>1.0</td><td>0.00</td><td>0.00</td><td>0.00</td><td>0.95</td></tr></tbody></table>	Alpha	HOTA	DetA	AssA	LocA	0.0	0.10	0.55	0.02	0.65	0.2	0.10	0.50	0.02	0.68	0.4	0.10	0.45	0.02	0.72	0.6	0.08	0.35	0.02	0.78	0.8	0.05	0.15	0.02	0.85	1.0	0.00	0.00	0.00	0.95
Alpha	HOTA	DetA	AssA	LocA																																
0.0	0.10	0.55	0.02	0.65																																
0.2	0.10	0.50	0.02	0.68																																
0.4	0.10	0.45	0.02	0.72																																
0.6	0.08	0.35	0.02	0.78																																
0.8	0.05	0.15	0.02	0.85																																
1.0	0.00	0.00	0.00	0.95																																

Nous pouvons constater que les courbes des modèles DeepSORT YOLOv8 et DeepSORT YOLOv9 sont très similaires, ce qui est cohérent étant donné la similitude de principe entre ces deux modèles. En

comparant ces courbes avec celle du modèle manuel, qui n'utilise pas DeepSORT, nous remarquons tout d'abord des performances de détection d'objets comparables, comme en témoigne la similitude des courbes pour la métrique DetA. Cependant, le problème majeur de ce modèle réside dans la métrique AssA, qui évalue la capacité de suivi d'objets. Peu importe le seuil alpha choisi, cette dernière affiche des valeurs très faibles, ce qui entraîne également des scores très bas pour l'indice HOTA (0).

Hypothèses

Après avoir totalement écarté le modèle manuel nous avons également comparé plus précisément sur différentes situation l'utilisation des modèles Deepsort avec yolov8 et yolov9 et comparé également leur résultat selon HOTA(0) calculé plus haut et différents seuils de HOTA.

Occlusion / Variation d'éclairage/ Petits images (dataset 02)

Concernant nos résultats relatifs aux occlusions, nous nous attardons sur le premier dataset. Celui possédant de nombreuses occultations, variations d'éclairages et des petites images. D'emblée avec HOTA(0), le score est de 0.35 pour YOLOv9 contre 0.30 pour YOLOv8 sur ce dataset. Le graphique suivant nous montre cette différence sur les courbes correspondant à la valeur HOTA(0). Le graphique nous montre que cette différence est principalement attribuable à la performance de YOLOv9 sur la métrique DetA(0), c'est-à-dire à la détection des objets sur nos images.

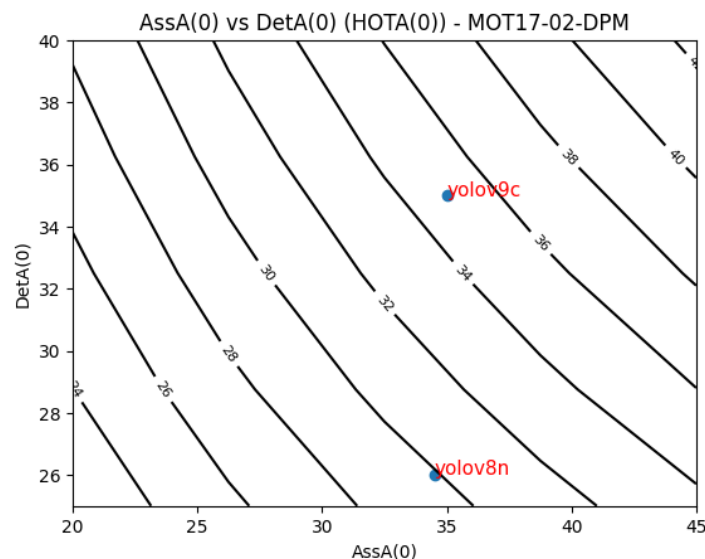
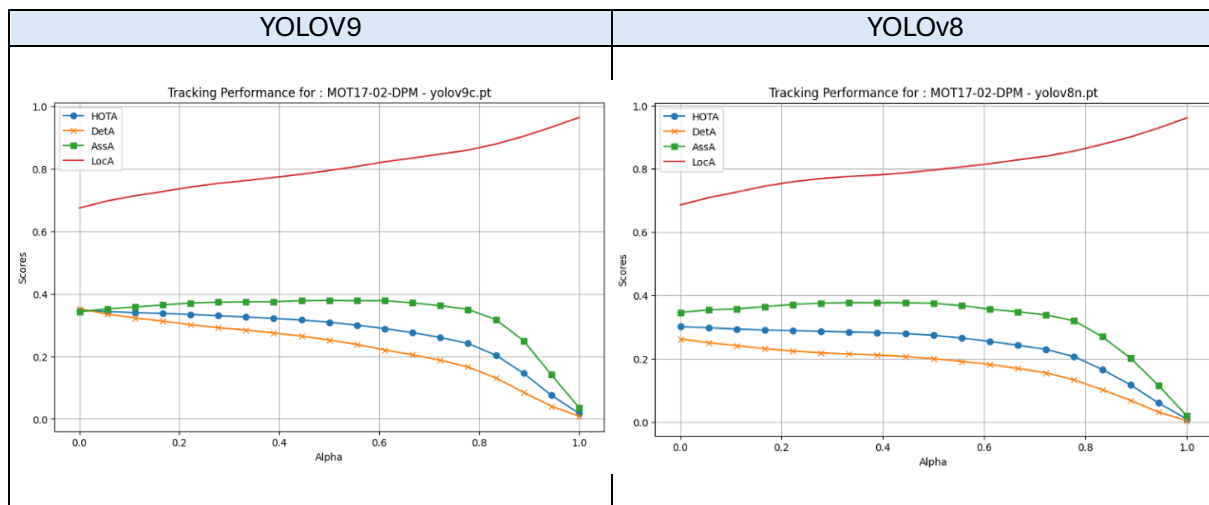


Figure 2 HOTA(0) des modèles selon AssA(0) et DetA(0) sur le dataset 02

Nous pouvons ensuite afficher les différentes métriques selon plusieurs seuils.



Nous remarquons que nos scores HOTA diminuent à mesure que le seuil alpha augmente, ce qui est attendu puisque relever alpha accroît la rigueur du critère de correspondance entre les boîtes englobantes prédites et réelles. Cependant, les modèles parviennent malgré tout à maintenir de bons scores HOTA jusqu'à un seuil de 0.6, démontrant ainsi une certaine précision dans la localisation de nos boîtes englobantes. Avec celles du modèle YOLOv9 qui sont légèrement meilleurs que celles de YOLOv8.

Nous affichons pour une trame de l'image les détections GT et celles des modèles.

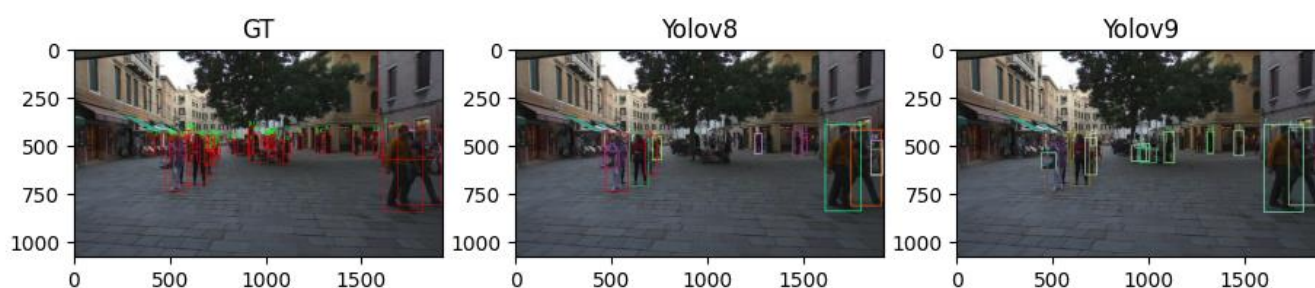


Image 6 Comparaisons des détections pour le dataset 02

En effet, à partir des exemples présentés, il est évident que la gestion des occlusions, permettant de détecter les personnes, a été plus performante avec YOLOv9. Cela est particulièrement notable lors de la comparaison des personnes identifiées au centre des images, ce qui n'est pas le cas avec YOLOv8.

Encombrement

Concernant la séquence du dataset 04, caractérisée par un environnement très encombré et de nombreuses occlusions, nous obtenons des scores HOTA(0) de 0.44 pour le modèle YOLOv9 et de 0.45 pour YOLOv8, ce qui est très similaire. Nous pouvons donc supposer que, dans ce cas, les deux modèles se valent, avec un léger avantage pour le modèle v8. Mais en analysant de plus près nos autres métriques on constate une grande différence de détection et de suivi entre eux.

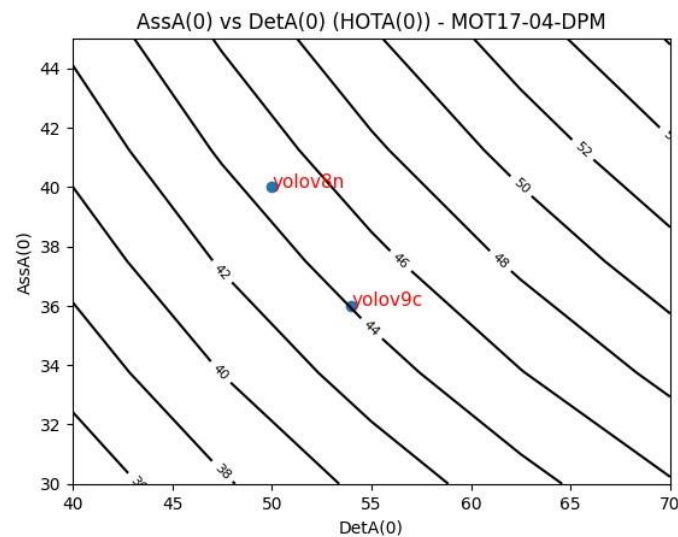
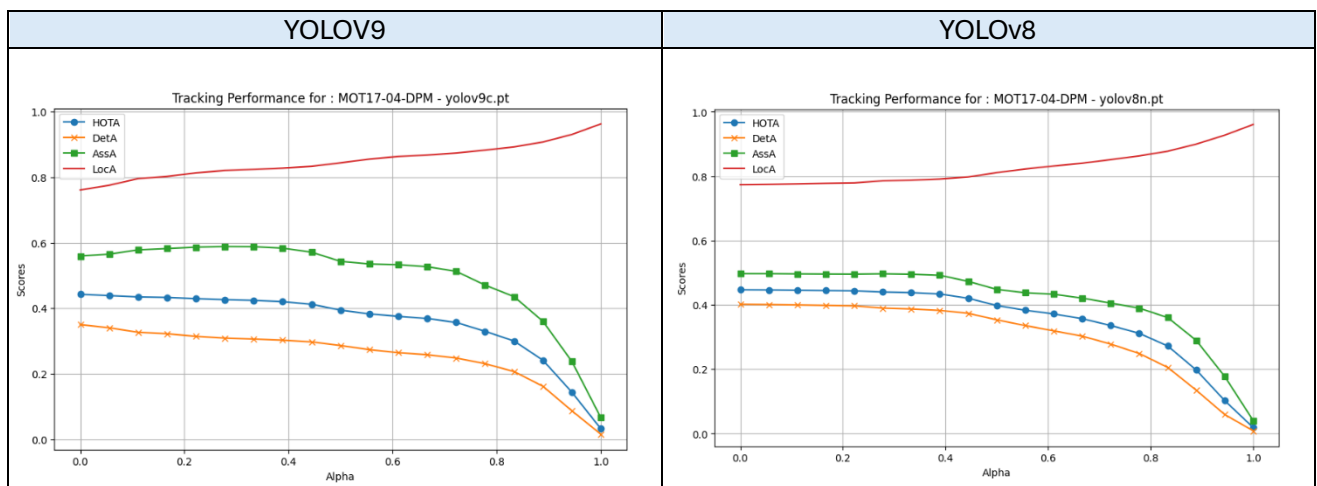


Figure 3 HOTA(0) des modèles selon AssA(0) et DetA(0) sur le dataset 04



Ceci peut également être remarqué en examinant de plus près les courbes spécifiques au HOTA. En effet, nous observons que le modèle YOLOv9 surpasse significativement YOLOv8 en termes de suivi d'objets, comme le montre la courbe verte. En revanche, pour la détection d'objets, YOLOv9 est nettement moins performant que YOLOv8, comme le souligne l'exemple de la séquence dans la zone de foule en haut à gauche. Dans ce cas, contrairement aux observations précédentes, l'augmentation de alpha a un effet moins uniforme et entraîne une chute plus rapide des performances.

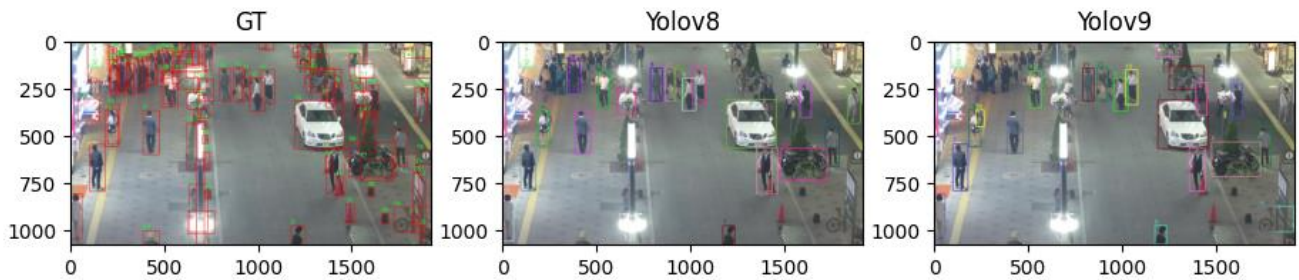
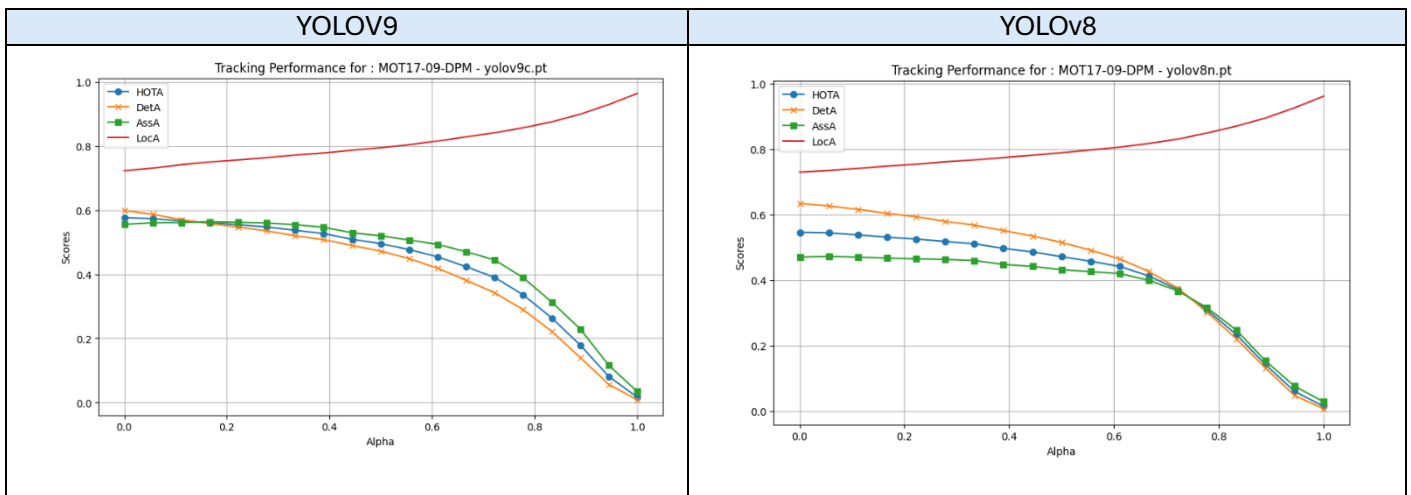


Image 7 Comparaisons des détections pour le dataset 04

Variation d'échelle

Dans la séquence où prédominent les variations d'échelle avec moins d'occasions d'occlusion et d'encombrement, nos modèles affichent des scores HOTA(0) de 0.58 pour YOLOv9 et 0.55 pour YOLOv8. Ces résultats sont nettement meilleurs que ceux obtenus sur les séquences précédentes, suggérant une moindre capacité des modèles à gérer l'occlusion en comparaison avec les simples changements d'échelle.



Nous observons un changement significatif par rapport aux analyses antérieures concernant la courbe de détection d'objets : celle-ci, en effet, surpasse maintenant celle du suivi, ce qui n'était pas le cas auparavant. La baisse observée n'est pas due à une détérioration du suivi mais plutôt à une amélioration de la détection.

Tout comme précédemment, YOLOv9 montre de meilleures performances de suivi par rapport à YOLOv8, mais l'inverse est vrai pour la détection d'objets. Malgré une meilleure performance globale par rapport aux autres séquences, l'impact de l'augmentation d'alpha se traduit ici par une baisse plus marquée des performances.

À l'issue de ces différents tests, nous avons constaté que le modèle DeepSORT couplé à YOLOv9 excelle dans l'aspect du suivi par rapport à celui basé sur YOLOv8. Cependant, il rencontre plus de difficultés dans la détection d'objets, particulièrement lors d'occlusions dans des situations de grande foule, ce qui est notable. Par ailleurs, ce modèle s'avère compétent pour identifier les objets lors de changements d'échelle, bien que ses performances soient légèrement inférieures à celles du modèle YOLOv8. Toutefois, lorsque l'on compare les deux, l'impact négatif du suivi moins précis de YOLOv8 sur les résultats HOTA(0) est plus prononcé, ce qui se traduit par des scores inférieurs.

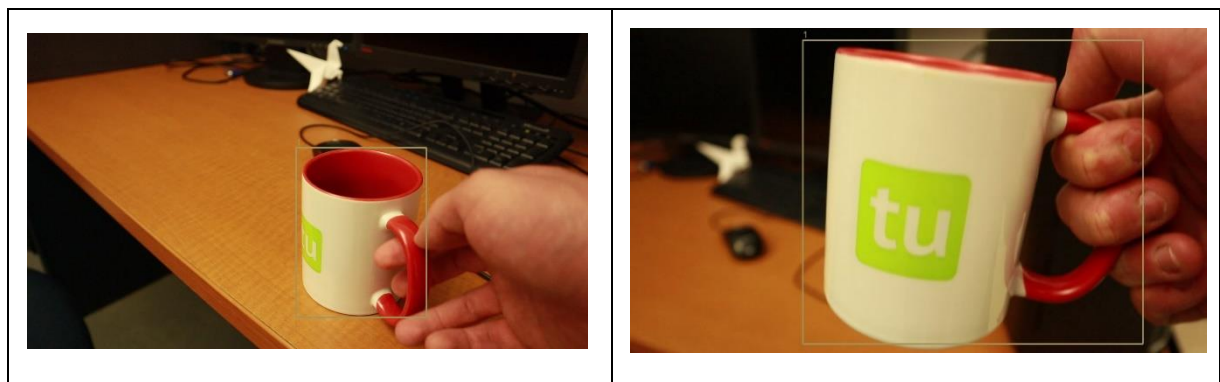
Résultats sur la séquence

Suite à nos différentes analyses précédentes, nous avons décidé d'opter pour le modèle DeepSORT et YOLOv9 pour l'analyse de la séquence sur Moodle. Ce choix est justifié par le fait que ce modèle s'est souvent avéré supérieur en termes de suivi par rapport à DeepSORT et YOLOv8. Bien qu'il y ait quelques différences en matière de détection, où YOLOv8 pourrait parfois être meilleur, ces variations peuvent être négligées étant donné que les séquences sur Moodle sont beaucoup moins encombrées que le dataset MOT-17 et la séquence 04. Nous préférons donc privilégier le suivi, surtout dans des situations où la détection ne semble pas être l'aspect le plus complexe de la séquence et sachant que YOLOv9 arrive tout de même à avoir de très bonnes performances dans certains types d'occlusion.

Échelle variable

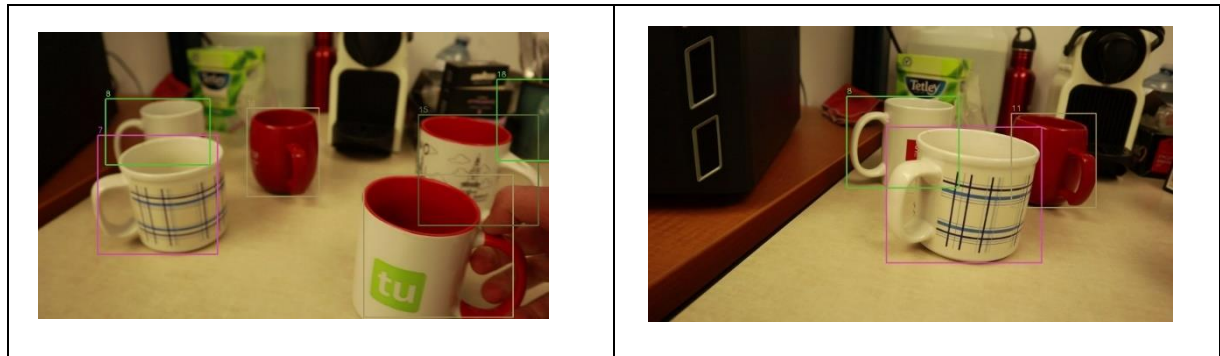
Suite aux différentes analyses réalisées sur le dataset MOT-17-09, nous avons pu constater que notre modèle ne rencontre pas de difficultés particulières avec les changements d'échelle et le suivi d'objets qui changent de perspective. Par conséquent, nous pouvons anticiper un très bon suivi des tasses lorsqu'elles sont déplacées vers la caméra dans notre séquence sur Moodle. Voici les résultats que nous pouvons observer pour ce cas.

Comme attendu, nous observons sur la séquence ci-dessus que la tasse est très bien encadrée et que son suivi est parfaitement fonctionnel. Cependant, lorsque la tasse est prise et qu'elle est alternativement rapprochée puis éloignée de la caméra, cela crée un phénomène de changement d'échelle.



Occultation

Pour ce cas, nos tests sur le dataset MOT-17 nous ont permis d'observer deux types de phénomènes. Nous avons noté, sur la séquence MOT-17-02, que notre modèle gère assez bien l'occlusion. Cependant, lorsque l'occlusion devient trop extrême, avec un encombrement important comme dans la séquence MOT-17-04, notre détection devient très mauvaise. Pour cette première séquence avec occlusion mais sans trop d'encombrement, nous devrions obtenir de bons résultats.

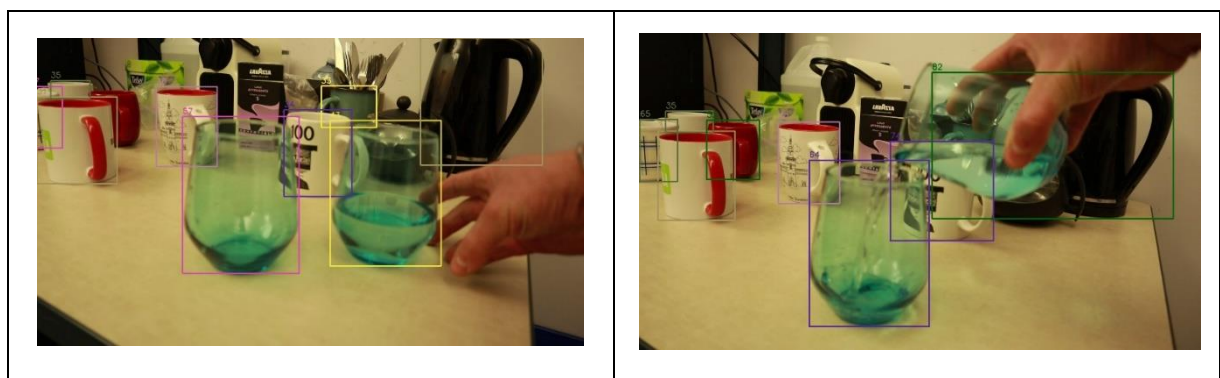


Comme prévu, nous constatons que toutes les tasses sont bien détectées malgré le fait qu'elles soient partiellement cachées. Le fait de les déplacer et de faire passer une tasse derrière une autre n'empêche également pas le suivi de celles-ci. Le seul point à souligner est que les boîtes englobantes ne sont pas parfaites et n'englobent pas parfaitement certaines tasses obstruées.

Objets similaires et Encombrement

Au vu des résultats obtenus avec le dataset MOT-17, notamment la séquence 04, nous pouvons raisonnablement supposer que notre modèle éprouvera des difficultés à identifier et suivre des objets dans un espace très encombré où plusieurs types d'objets se ressemblent. En ce qui concerne le suivi d'objets similaires, bien que nous n'ayons pas rencontré de véritables cas avec des objets parfaitement identiques, il est probable que le modèle aura du mal à distinguer deux objets, surtout s'ils sont proches l'un de l'autre.

La séquence ci-dessous illustre tous les cas mentionnés précédemment, et nous obtenons les résultats suivants pour la détection et le suivi de ces objets.



Comme prévu, les résultats obtenus ne sont vraiment pas satisfaisants, et même pires que ce que nous avions anticipé. En effet, entre les deux objets similaires, nous observons que de la première à la deuxième frame sur la droite, nos deux objets sont considérés comme différents de ce qu'ils étaient auparavant. Ce qui est le plus surprenant, c'est que même si l'objet de gauche n'a pas bougé, il a été identifié dans la seconde frame comme un tout autre objet, passant du numéro 57 au début à 64 ensuite. De plus, avec tout cet encombrement et ces occlusions répétées, nous avons un verre qui s'est fusionné avec la bouilloire pour créer un tout autre objet, ce qui expose clairement les faiblesses de ce modèle face à des occlusions en chaîne. Seules quelques tasses à l'arrière-plan, qui n'ont pas bougé et présentent des occlusions assez simples, ont été correctement suivies, ce qui reste très limité.

Mouvement de caméra / petit objet

Concernant les différents mouvements et l'identification de petits objets ou d'objets éloignés du champ, cela est souvent lié au principe de variation d'échelle, qui est assez bien maîtrisé par notre modèle. Cela permet donc de supposer que notre modèle pourrait avoir de bonnes performances sur la séquence ci-dessous.



Après analyse de nos résultats, nous constatons que notre modèle réussit à bien suivre la tasse sur l'ordinateur malgré les mouvements de la caméra et l'éloignement de celle-ci. Cependant, le point le plus complexe concerne les tasses situées plus bas. Nous parvenons à suivre certaines d'entre elles, mais dès qu'elles sont trop coupées, le suivi devient difficile et elles sont détectées comme étant d'autres objets.

Ces résultats confirment que notre modèle est performant lorsqu'il s'agit de mouvements, de mouvements de caméra, et de changements d'échelle, mais sa faiblesse se manifeste à un seuil d'occlusion trop élevé, qu'il ne parvient pas à gérer correctement. Ce qui est parfois compréhensible lorsque certaines tasses sont totalement coupées sur un plan comme nous pouvons le voir sur la seconde image.

Conclusion

Pour conclure, notre étude s'est initialement concentrée sur la sélection du meilleur modèle. Nous avons décidé de réaliser de nombreux tests et une analyse détaillée afin de trouver le modèle le plus adapté pour étudier notre séquence sur Moodle. À l'issue de ces analyses, nous avons opté pour la combinaison DeepSort et YOLOv9, ce qui nous a surpris à certains égards. En effet, contrairement à ce que nous pouvions penser, les plus grandes différences entre YOLOv9 et YOLOv8 ne résidaient pas tant au niveau de la détection, bien que ces soient des modèles de détection, mais plutôt dans le suivi. YOLOv9 semble mieux gérer son association avec DeepSort que YOLOv8. De plus, un point important de l'évaluation de nos modèles est la métrique HOTA qui permet de combiner les propriétés de détection et de suivi en une seule métrique et qui a souvent favorisé notre modèle choisi. En effet, même si nous avons souhaité examiner toutes les métriques, finalement, nous avons sélectionné le modèle avec le meilleur score HOTA(0).

D'autre part, selon le niveau de précision choisi, nous pouvons obtenir des résultats de HOTA très variés en fonction de notre hyperparamètre alpha, utilisé comme seuil de permissivité. Cela peut être particulièrement intéressant dans d'autres applications où une grande précision de la boîte englobante est requise. Nous observons souvent ce type de besoin dans le domaine sportif, par exemple pour la détection du premier arrivé dans des courses où la victoire peut se jouer à quelques centimètres, ou encore au football pour déterminer si le ballon a franchi la ligne de but.

Au cours de notre étude, nous avons observé différents phénomènes que notre modèle gère bien ou moins bien. Nous avons constaté que ce modèle excelle dans la gestion des mouvements, des mouvements de caméra et des changements d'échelle, ce qui a également été confirmé sur notre séquence sur Moodle, comme nous l'avons vu précédemment. Concernant l'occlusion, ce modèle peut présenter deux types de comportements très différents. D'une part, si l'occlusion n'est pas trop prononcée et qu'il y a peu d'encombrement, nous observons de très bons résultats, tant sur le dataset MOT-17 que sur notre séquence sur Moodle. Cependant, lorsque l'occlusion devient trop importante avec un encombrement conséquent, le modèle adopte un second comportement nettement moins efficace, confondant les objets entre eux ou utilisant des éléments du décor comme extension de notre boîte englobante. Cela pouvait même impacter d'autres objets ne bougeant pas mais ressemblant du coup à la nouvelle forme ou d'une forme similaire à un objet en mouvement, changeant l'identification de ces objets sans même avoir une interaction dessus comme nous avons pu le voir dans la séquence sur Moodle.