

INF6804 - TP3

Miquel Florensa Tommy Pou
2251214 190154

Hiver 2024

Table des matières

1	Introduction	2
2	Méthodologie	2
2.1	YOLO	2
2.2	Byte-track	4
3	Expérience	5
3.1	Difficultés de la tâche	5
3.2	Justification de la méthode	5
4	Implémentation utilisée	6
4.1	Modèles de détection et de suivi	6
4.2	Métriques d'évaluation	7
5	Résultats des expériences	7
6	Discussion des résultats	10
6.1	Retour sur les difficultés	10
6.2	Analyse des graphiques	11
6.3	Discussion des métriques	11
6.4	Hyperparamètres	12
6.5	Hypothèse sur la séquence sur Moodle	12
7	Conclusion	12

1 Introduction

Dans le domaine de vision par ordinateur, le suivi de multiples objets (*MOT*) est une tâche qui consiste à identifier les objets dans une séquence d'images et à suivre le déplacement de ceux-ci à travers le temps. Dans le cadre de ce travail, nous implémenterons une approche permettant de résoudre ce problème en effectuant un suivi d'objets sur la vidéo qui nous a été assignée. Dans ce document, nous aborderons dans un premier temps la méthodologie que nous proposons, des difficultés en lien avec la tâche et de nos hypothèses sur la performance de notre solution. Ensuite, nous présenterons l'implémentation de l'approche et des résultats aboutis. Enfin nous analyserons la performance de notre algorithme.

2 Méthodologie

Dans ce travail, nous utiliserons le suivi Yolo qui est une méthode de suivi multi-objets basée sur la détection d'objets YOLO et dans laquelle différentes méthodes de suivi peuvent être ajoutées. Le méthode que nous utiliserons sera ByteTrack. Cette méthode constitue actuellement l'état de l'art pour les suiveurs d'objets multiples (*MOT*). Mais tout d'abord, expliquons tous les composants un par un :

2.1 YOLO

YOLO, ou "You Only Look Once", est un algorithme efficace de détection d'objets introduit par Joseph Redmon et al. Plutôt que de faire glisser une fenêtre ou d'utiliser des propositions de régions, YOLO divise l'image d'entrée en une grille et fait des prédictions pour les boîtes de délimitation et les probabilités de classe pour les objets à l'intérieur de chaque cellule de la grille en un seul passage à travers un réseau neuronal convolutionnel (CNN). Cette approche permet à YOLO d'obtenir des performances en temps réel. Pour chaque cellule de la grille, YOLO prédit les coordonnées de la boîte englobante, les dimensions, les scores de confiance et les probabilités de classe. Une suppression non maximale est ensuite appliquée pour éliminer les boîtes de délimitation redondantes, ce qui permet d'obtenir un ensemble final de boîtes de délimitation avec les étiquettes de classe et les scores de confiance associés. YOLO est largement utilisé dans diverses applications telles que la conduite autonome et la surveillance en raison de sa rapidité et de sa précision dans les tâches de détection d'objets.

YOLOv8 est l'avant-dernière version de YOLO, et celle dont l'implémentation est la plus efficace et la plus stable. L'architecture de YOLOv8 commence par des couches convolutives suivies de couches entièrement connectées dans sa tête qui sont responsables de la prédiction des boîtes de délimitation, des scores d'objectivité et des probabilités de classe pour les objets détectés dans chaque image. En outre, YOLOv8 intègre un mécanisme d'auto-attention dans sa tête, ce qui permet au modèle de hiérarchiser dynamiquement les caractéristiques de l'image en fonction de leur pertinence. En outre, il est spécialisé dans la détection d'objets à plusieurs échelles, en s'appuyant sur un réseau de pyramides de caractéristiques pour détecter des objets de tailles et d'échelles différentes. Ce réseau, composé de plusieurs couches, permet à YOLOv8 de détecter des objets allant de la plus grande à la plus petite taille dans une image.

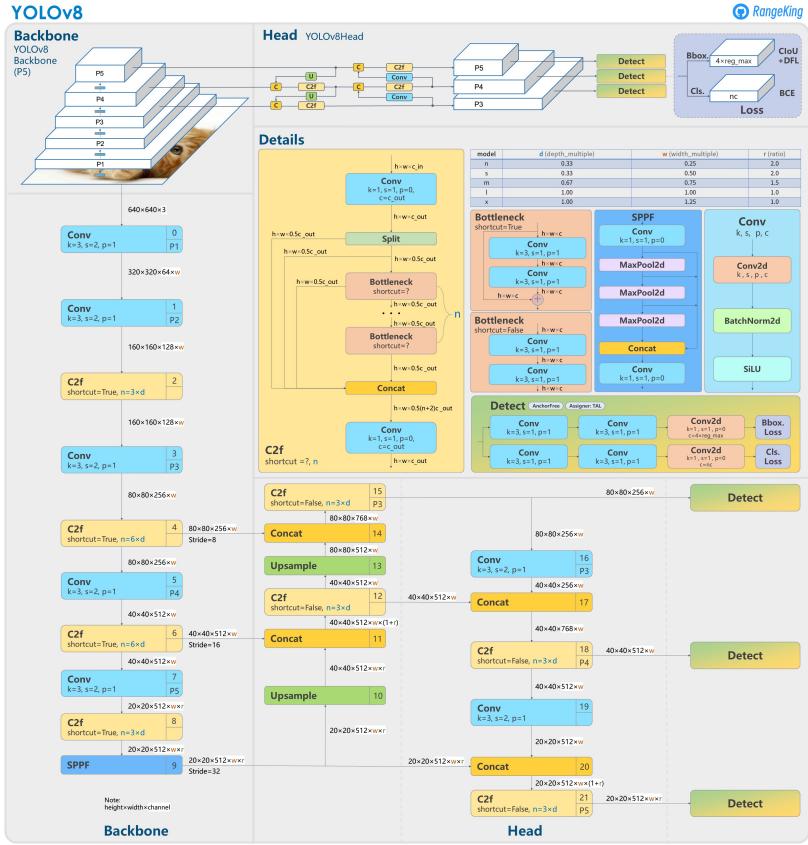


FIGURE 1 – Architecture neuronale de YOLOv8 [1].

Dans ce travail pratique, nous utiliserons aussi YOLOv9, la dernière version qui améliore les versions précédentes en surmontant les problèmes de perte d'information inhérents aux réseaux neuronaux profonds. Elle intègre l'information de gradient programmable (PGI) et une architecture de réseau d'agrégation de couches efficace généralisée (GELAN), ce qui rend le modèle très léger et efficace et le rend plus performant que n'importe quel autre modèle.

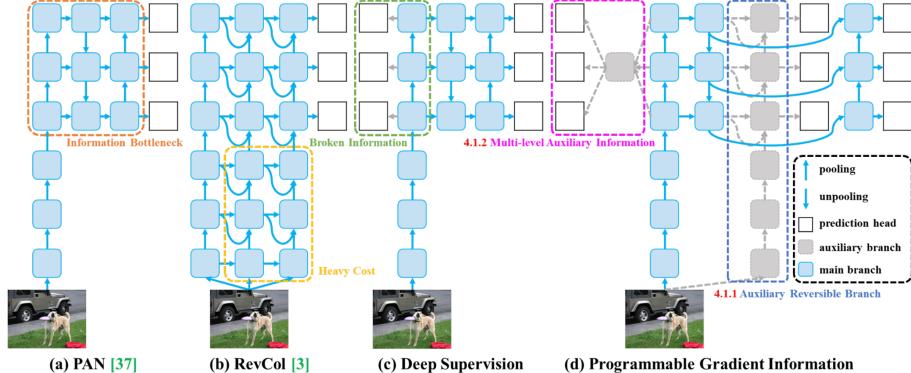


Figure 3. PGI and related network architectures and methods. (a) Path Aggregation Network (PAN)) [37], (b) Reversible Columns (RevCol) [3], (c) conventional deep supervision, and (d) our proposed Programmable Gradient Information (PGI). PGI is mainly composed of three components: (1) main branch: architecture used for inference, (2) auxiliary reversible branch: generate reliable gradients to supply main branch for backward transmission, and (3) multi-level auxiliary information: control main branch learning plannable multi-level of semantic information.

FIGURE 2 – Architecture neuronale de YOLOv9 [2].

2.2 Byte-track

D'autre part, nous devons également ajouter une méthode de suivi qui permet de suivre des cibles mobiles et de les identifier à l'aide d'un identifiant. Le méthode de suivi que nous utiliserons est donc Byte-track :

ByteTrack est une méthode de suivi multi-objets simple mais efficace qui atteint des performances de pointe sur plusieurs points de référence. Sa principale innovation réside dans la manière dont elle associe les boîtes de détection au cours du processus de suivi.

Le fonctionnement de base de la méthode repose sur :

- Obtenir des boîtes de détection : ByteTrack obtient d'abord les boîtes de détection et les scores d'un détecteur d'objets comme YOLOX pour chaque image de la vidéo.
- Séparer les cases à score élevé de celles à score faible : ByteTrack sépare les boîtes de détection en boîtes à score élevé (au-dessus d'un seuil) et en boîtes à score faible.
- Première association : Les cases à score élevé sont associées aux "tracklets" existants (objets suivis dans les images précédentes) en utilisant la similarité de mouvement ou d'apparence et l'algorithme hongrois. Cette méthode est similaire aux méthodes de suivi traditionnelles.
- Deuxième association : Les tracklets non appariés de la première étape sont ensuite associés aux boîtes de détection à faible score en utilisant uniquement des indices de mouvement tels que l'intersection sur l'union (IoU). Cela permet de récupérer des objets dont le score de détection est faible en raison d'une occlusion ou d'un flou de mouvement.
- Gestion des pistes : Les cases non appariées à faible score sont considérées comme de l'arrière-plan et éliminées. Les tracklets non appariés sont conservés pendant quelques images en vue d'une éventuelle réassociation. De nouveaux tracklets sont initialisés à partir de toutes les détections à score élevé non appariées.

3 Expérience

La vidéo que nous devons analyser consiste en un court-métrage d'une personne qui déplace une tasse dans une pièce. Tout au long de la séquence, la personne déplace la caméra. En arrière-plan, on retrouve d'autres tasses et d'autres objets typiques que l'on retrouverait dans un bureau. La méthode que nous allons développer doit être en mesure d'identifier et de suivre correctement toutes les tasses qui se retrouvent dans la vidéo.

3.1 Difficultés de la tâche

Les difficultés en lien avec cette tâche sont les suivantes.

- **Mouvement de la caméra.** La première difficulté émane du fait que la caméra bouge, rendant ainsi la scène dynamique. Par conséquent, les objets subissent une translation. Bien que les objets soient statiques, l'écart entre les éléments et la référence, c'est-à-dire la caméra, varie à chaque image. De plus, les objets subissent une rotation puisque le point de vue change. Enfin, le mouvement de la caméra peut flouter l'image, réduisant ainsi la qualité de l'image.
- **Occlusions.** Dans cette vidéo, la présence d'occlusions revêt plusieurs formes. Tout d'abord, la séquence débute avec la personne qui tient la tasse par la poignée. Ses doigts cachent une partie de la tasse, ce qui peut engendrer une difficulté pour le modèle à reconnaître la forme entière de la tasse. Ensuite, au fur et à mesure que la vidéo progresse, on constate que certaines tasses se chevauchent. Cela pose un problème, car certaines tasses sont partiellement visibles. Leurs boîtes englobantes peuvent être moins précises.
- **Variance intra-classe et changement d'apparence.** Les tasses ne sont pas uniformes. Certaines tasses sont opaques et d'autres sont transparentes. Les tasses opaques comportent des couleurs différentes : certaines sont rouges et sur les tasses blanches, on retrouve des illustrations différentes. De plus, les tasses transparentes possèdent une forme différente : elles ne comportent pas de poignée. On peut même voir le contenu des verres transparents. Vers la fin de la vidéo, la personne verse de l'eau dans une tasse transparente. Le modèle devra être suffisamment robuste pour suivre cet objet malgré la transformation subie.
- **Ré-identification.** En raison du déplacement de la caméra, certaines tasses disparaissent de la vidéo, mais ré-apparaissent quelques images plus tard. Ce constat engendre une difficulté au niveau de l'identification des tasses, car le modèle devra être capable de reconnaître les tasses qui ne sont pas nouvelles.
- **Débordement et échelle.** À un moment donné, la personne rapproche la caméra d'une tasse jusqu'au point où l'image ne peut plus contenir toute la forme de la tasse. Il est difficile d'identifier un objet partiellement visible, surtout lorsque son échelle est très grande à cause de sa proximité à la caméra. Inversement, lorsque l'objet est loin de caméra, il apparaît petit dans l'image et devient difficile à détecter.

3.2 Justification de la méthode

Nous pensons que notre solution sera bien affectée par la plupart des problèmes énumérés ci-dessus.

- **Mouvement de la caméra.** Dans la vidéo que nous devons analyser, nous n'avons trouvé aucun mouvement brusque. Ainsi, la qualité des trames devrait être consistante et suffisamment bonne à travers toute la séquence, ce qui est désiré pour la détection d'objets. Pour la

méthode de suivi, nous pensons que l'algorithme Byte-Track aura des résultats satisfaisants, car le déplacement de la caméra est relativement lent. Les déplacements des boîtes englobantes ne devraient pas être trop grands, ce qui facilite l'association.

- **Occlusions.** Les modèles de YOLO ont été entraînés sur le jeu de données COCO, qui contient une immense variété d'images contenant des verres. Certaines de ces images incluent des verres partiellement visibles, donc le modèle de détection devrait être en mesure d'identifier jusqu'à un certain point les verres légèrement cachés. Les occlusions présentent toutefois un problème pour l'étape de suivi. Lorsque les occlusions sont mélangées avec le mouvement de caméra, cela peut perturber les boîtes englobantes des objets concernés. L'association devient alors plus difficile.
- **Variance intra-classe et changement d'apparence.** Le jeu de données sur lequel le modèle a été entraîné comporte plusieurs sortes de verres. Nous pourrions ainsi nous attendre à ce que YOLO identifie suffisamment bien les différents types de verres. Si le modèle de détection est robuste à ce défi visuel, alors cela ne devrait pas affecter la qualité des boîtes englobantes. Par contre, les changements visuels peuvent causer un problème pour l'étape d'association puisqu'ils réduisent la similarité entre les boîtes correspondantes.
- **Ré-identification.** Byte-Track repose surtout sur la métrique d'IoU pour ré-identifier les objets. Lorsque les objets sortent de la trame et entrent à nouveau, il faut que les boîtes englobantes apparaissent près de l'endroit où elles ont disparu. Il existe une autre solution, qui est de remplacer l'association de IoU par la métrique ReID. Nous pouvons ainsi effectuer une association basée sur l'apparence plutôt que le IoU. Ces deux métriques rendent l'approche davantage robuste au problème de ré-identification puisqu'elles performent mieux dans des contextes différents. [3]
- **Débordement et échelle.** Lorsqu'un objet déborde dans l'image, le niveau de confiance est largement réduit parce que l'objet n'affiche pas la totalité de ses textures. Cela est problématique pour le modèle de détection parce que le modèle peut manquer l'objet avec un haut seuil de confiance ou détecter des faux positifs en raison du faible niveau de confiance. Les objets de petite échelle sont problématiques pour les mêmes raisons évoquées.

4 Implémentation utilisée

4.1 Modèles de détection et de suivi

Comme nous l'avons indiqué précédemment, nous utiliserons le détecteur YOLO pour détecter les tasses et le Byte-track tracker.

Comme il s'agit de modèles complexes avec des implémentations difficiles, nous utiliserons l'implémentation d'Ultralytics [4]. Ultralytics YOLOv8 est un modèle de détection d'objets SOTA qui offre un large éventail de fonctionnalités, notamment la détection d'objets, la segmentation d'instances, l'estimation de la pose, la détection d'objets orientés et la classification d'images. Il excelle dans la détection et la localisation d'objets dans les images et les vidéos, ce qui le rend parfait pour des applications telles que la surveillance, les véhicules autonomes, la robotique, etc. En outre, YOLOv8 peut effectuer une segmentation des instances, délimitant les frontières précises des objets, et estimer la pose ou les points clés des objets détectés, ce qui permet des applications dans des domaines tels que l'interaction homme-machine, l'analyse des mouvements et la réalité augmentée. Il peut également détecter des objets avec des orientations arbitraires et classer des images entières dans des classes prédéfinies. En ce qui concerne les licences, Ultralytics propose une licence open-source

AGPL-3.0 pour les étudiants, les chercheurs et les passionnés, favorisant la collaboration ouverte et le partage des connaissances, ainsi qu'une licence d'entreprise conçue pour un usage commercial, permettant l'intégration transparente de YOLOv8 dans des produits et services commerciaux.

Cette implémentation supporte également YOLOv9 mais n'est pas encore optimisée pour ce nouveau modèle.

En plus du détecteur YOLO, nous utiliserons le tracker Byte-track de Supervision [5]. Supervision est une bibliothèque qui se concentre sur le comptage et le suivi d'objets. Elle est compatible avec les détecteurs YOLOv8 et YOLOv9 et inclut le tracker Byte-track. Bien que cette bibliothèque soit en version bêta, elle a montré de bons résultats et une grande facilité d'utilisation. Enfin, nous effectuerons une recherche d'hyperparamètres afin d'obtenir les meilleurs résultats dans le défi MOT20 dans la section suivante. Cette recherche consistera principalement à trouver les meilleures valeurs pour le seuil de confiance pour les détections et le seuil Intersection over Unit (IoU).

4.2 Métriques d'évaluation

Pour évaluer le modèle, nous nous servirons du score *Higher Order Tracking Accuracy* (HOTA). Ce score d'évaluation est spécialement conçu pour le suivi d'objets multiples et combine plusieurs métriques. Il évalue à quel point les trajectoires des détections correspondantes sont bien alignées et pénalise les correspondances erronées. Il combine la précision de détection *DetA*, la précision d'association *AssA* et la précision de localisation *LocA*. Le score HOTA pour un seuil de localisation α est donné par : [6]

$$HOTA_\alpha = \sqrt{DetA_\alpha \times AssA_\alpha}$$

Le score final de HOTA calcule la moyenne de tous les scores $HOTA_\alpha$ où la métrique $LocA = \int_0^1 \frac{1}{|TP_\alpha|} \sum_{c \in TP_\alpha} \times \text{Score de similarité } S(c) d\alpha$ dépasse le seuil donné :

$$HOTA = \int_0^1 HOTA_\alpha d\alpha$$

L'utilisation du score HOTA a été directement tirée l'implémentation officielle de cette métrique. [7] Pour nos expériences, nous aurons tout simplement besoin d'installer les *ground truths* du jeu de données. Le format d'annotation de notre jeu de données, MOT20, est compatible avec le script d'évaluation présent dans le code.

5 Résultats des expériences

Dans la section suivante, nous verrons les résultats obtenus avec les différentes techniques expliquées précédemment dans l'ensemble de données MOT20, qui est un ensemble de données pour le suivi d'objets multiples réalisé en 2020. L'ensemble de données contient 8 séquences vidéo difficiles (4 pour le train, 4 pour le test) dans des environnements sans contraintes, dans des lieux très fréquentés tels que des gares, des places publiques et un stade. En outre, nous testerons différents hyperparamètres pour chaque configuration.

Métriques HOTA avec YOLOv8, Byte-track, confiance= 0.3, seuil IoU=0.7												
Dataset	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA	RHOTA	HOTA(0)	LocA(0)	HOTALocA(0)
MOT20-01	33.958	25.224	45.947	25.929	76.281	48.769	76.643	79.97	34.476	43.698	75.714	33.085
MOT20-02	24.841	22.633	27.51	23.25	76.703	28.996	76.919	80.607	25.222	31.53	76.257	24.044
MOT20-03	20.996	17.364	25.498	17.725	71.89	26.333	76.124	76.822	21.234	28.821	71.307	20.551
MOT20-05	7.8051	4.7824	12.787	4.8004	80.455	12.99	83.795	82.003	7.8217	9.5158	78.977	7.5153
COMBINED	16.312	11.116	24.078	11.26	75.345	25.029	79.105	79.242	16.429	21.232	74.601	15.839

TABLE 1 – Mesures HOTA pour les vidéos MOT20 avec YOLOv8, Byte-track et confiance= 0.3.

Métriques HOTA avec YOLOv8, Byte-track, confiance= 0.1, seuil IoU=0.7												
Dataset	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA	RHOTA	HOTA(0)	LocA(0)	HOTALocA(0)
MOT20-01	35.36	28.072	44.884	29.106	74.442	47.993	74.149	79.151	36.079	46.343	74.17	34.373
MOT20-02	26.201	25.108	27.653	25.993	74.965	29.257	75.057	79.918	26.72	33.681	74.931	25.237
MOT20-03	26.38	23.494	29.772	24.249	69.77	31.11	73.707	75.818	26.837	37.328	69.574	25.971
MOT20-05	11.344	7.8919	16.363	7.9482	78.681	16.695	81.274	80.972	11.389	14.165	77.359	10.958
COMBINED	19.744	14.991	26.16	15.286	73.575	27.308	77.19	78.374	19.957	26.326	73.111	19.247

TABLE 2 – Mesures HOTA pour les vidéos MOT20 avec YOLOv8, Byte-track et confiance= 0.1.

Métriques HOTA avec YOLOv9, Byte-track, confiance= 0.1, seuil IoU=0.5												
Dataset	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA	RHOTA	HOTA(0)	LocA(0)	HOTALocA(0)
MOT20-01	39.236	30.393	50.90	31.74	80.13	53.47	81.02	82.22	39.79	48.69	78.92	38.42
MOT20-02	29.287	24.555	35.11	25.20	79.50	36.82	81.50	82.59	29.71	36.09	79.09	28.55
MOT20-03	22.192	16.103	30.64	16.41	73.70	31.52	78.21	78.26	22.41	29.74	73.24	21.78
MOT20-05	13.902	9.765	19.19	9.84	80.30	20.34	82.48	82.03	13.97	17.02	78.84	13.42
COMBINED	19.885	13.925	28.55	14.13	77.87	29.58	81.46	80.95	20.05	25.08	77.01	19.32

TABLE 3 – Mesures HOTA pour les vidéos MOT20 avec YOLOv9, Byte-track et confiance= 0.1.

IoU seuil (%)											
Conf. seuil (%)	70		90		95		HOTA	HOTA(0)	HOTA	HOTA(0)	
	HOTA	HOTA(0)	HOTA	HOTA(0)	HOTA	HOTA(0)					
10	19.74	26.32	19.73	26.34	17.52	24.41					
20	18.03	23.74	17.20	22.76	16.01	21.74					
30	16.31	21.23	15.47	20.18	14.41	19.19					

TABLE 4 – Résultats combinés avec HOTA et HOTA(0) pour différents IoU et seuils de confiance avec YOLOv8 et Byte-track.

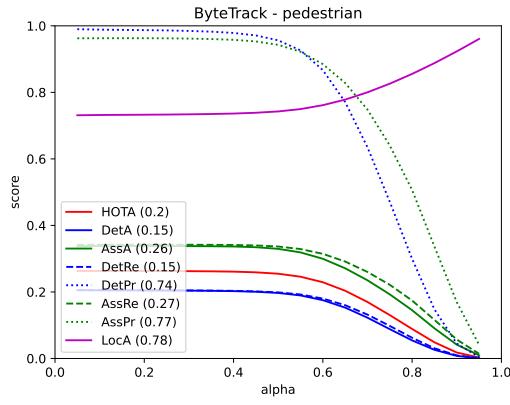


FIGURE 3 – Résultats MOT20 challenge avec YOLOv8, Byte-track, confiance 0.1 et IoU seuil 0.7.

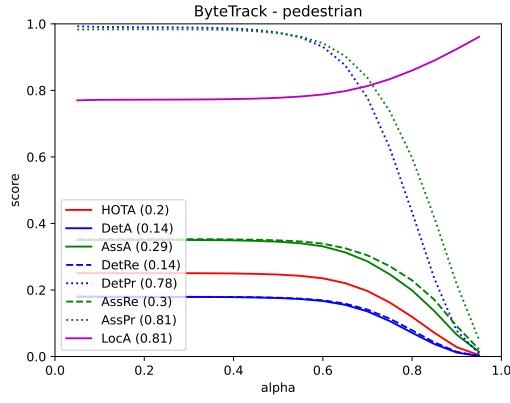


FIGURE 4 – Résultats MOT20 challenge avec YOLOv9, Byte-track, confiance 0.1 et IoU seuil 0.5.

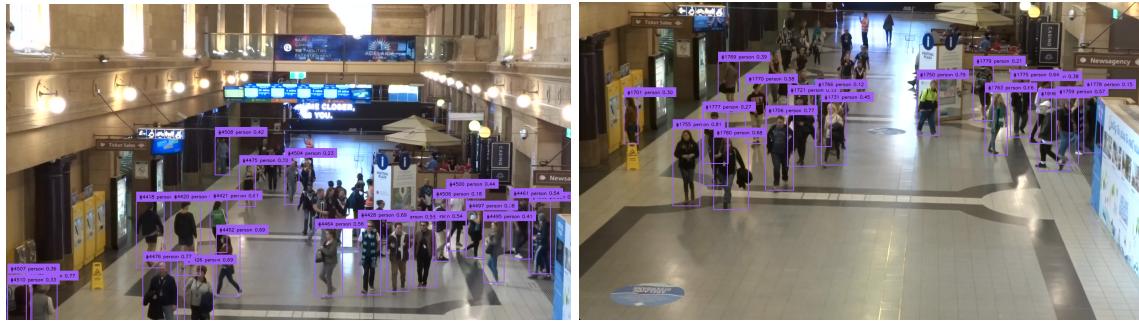


FIGURE 5 – Résultats MOT20-01 et MOT20-02.

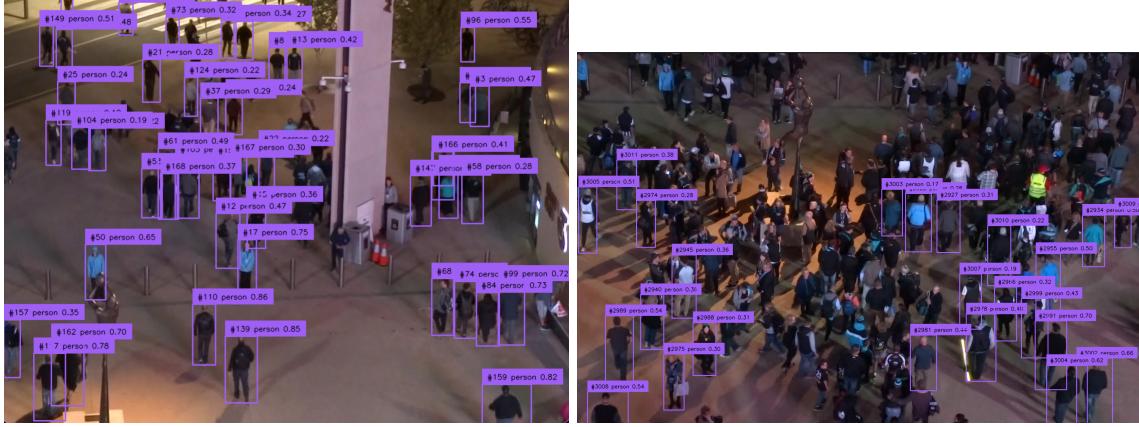


FIGURE 6 – Résultats MOT20-03 et MOT20-05.

6 Discussion des résultats

6.1 Retour sur les difficultés

Tout d’abord, nous pouvons clairement voir dans les tableaux 1 et 3 que le HOTA pour YOLOv8 avec Byte-track dépend fondamentalement de la séquence vidéo que nous analysons, donc pour la première vidéo le résultat est correct, mais avec les autres séquences la performance diminue. Ce problème est simplement dû à la complexité de chaque séquence. Ainsi, comme nous pouvons le voir dans les figures 5 et 6, dans les premières vidéos, le modèle est capable de détecter et de suivre chaque personne, mais dans la figure 6, dans les vidéos MOT20-03 et MOT20-05, le modèle n’est pas capable de détecter tous les humains car il y en a beaucoup et la lumière de la vidéo n’est pas bonne.

Nous allons analyser chacune des difficultés évoquées dans la section 3 par rapport à la performance de notre modèle sur le jeu de données MOT20, si applicable.

- **Mouvement de la caméra.** Dans les vidéos du jeu de données MOT20, la caméra est statique, ce qui limite la comparaison entre la performance sur MOT20 et celle sur la séquence de Moodle. Toutefois, dans les deux ensembles de données, les objets se déplacent à travers les trames dans des directions variées. Le suivi des personnes ne semble pas avoir été si difficile pour le modèle.
- **Occlusions.** Les occlusions paraissent très problématiques dans la figure 6. L’image de droite est saturée de personnes qui se chevauchent, ce qui nous empêche de les voir au complet. Par conséquent, un très grand nombre de personnes n’ont pas été identifiées par le modèle. Ce facteur semble être la raison principale du faible score lors de l’évaluation, puisqu’il a causé un très grand nombre de boîtes englobantes non identifiées.
- **Variance intra-classe et changement d’apparence.** Dans l’ensemble de données MOT20, le changement d’apparence n’a pas pu être testé. Toutes les personnes ont les mêmes caractéristiques du début à la fin des séquences. Cependant, nous avons pu la variance intra-classe a pu être observée. Les personnes ont des poses et des textures différentes. Sur la figure 5, cela ne semble pas poser d’erreur grave pour les personnes en avant-plan, malgré leurs vêtements distincts ou le fait qu’elles tiennent des objets différents. Il reste à préciser que la

performance du modèle est très dépendante du jeu de données sur lequel il a été entraîné.

- **Ré-identification.** La première vidéo de l'ensemble MOT20 contient des personnes qui marchent derrière un panneau d'informations. Le modèle a dû ré-identifier les personnes avant et après avoir été caché par cet obstacle. Néanmoins, le modèle a obtenu le meilleur score sur cette séquence, donc la ré-identification ne semble pas être particulièrement difficile pour cette approche.
- **Débordement et échelle.** Nous n'avons trouvé aucune instance de débordement dans les vidéos de MOT20. Par contre, nous avons pu observer des personnes de très petite échelle dans les séquences. Ces éléments ont été problématiques pour le modèle tel que vu dans les images de droites des deux figures. Plusieurs personnes situées dans l'arrière-plan n'ont pas été identifiées. Cependant, dans d'autres cas comme la vidéo MOT20-03, les personnes situées au loin ont été détectées.

6.2 Analyse des graphiques

Si nous examinons les figures 3 et 4, nous verrons que les deux graphiques sont presque identiques, mais que dans le cas de YOLOv9, nous obtenons des résultats légèrement meilleurs pour chaque métrique. Maintenant, si nous examinons l'une des figures, par exemple la figure 4, nous verrons que toutes les mesures sont stables pour toute valeur du seuil de l'IoU de localisation jusqu'à ce qu'elles se situent autour de $\alpha = 0.55$, où toutes les mesures commencent à diminuer au même rythme. Cela signifie que notre modèle ne fonctionne bien que si nous considérons que la précision de l'IoU de la localisation n'est pas supérieure à 0.55.

6.3 Discussion des métriques

Afin de vérifier réellement les performances du modèle, nous allons examiner chaque composante de la métrique HOTA pour le YOLOv9 :

- **DetA** : la détection mesure l'alignement entre l'ensemble des détections prédictes et l'ensemble des détections réelles. En examinant les résultats dans le tableau 3, nous constatons que DetA est très médiocre en général, en particulier dans la dernière séquence vidéo qui obtient un score de seulement 7.9. Ces résultats s'expliquent par le fait que le modèle n'est parfois pas en mesure de capturer toutes les personnes présentes dans la scène et qu'il y a donc beaucoup de faux négatifs (détections de vérité de terrain qui ne correspondent pas). Encore une fois, si nous regardons une image de la détection dans la séquence MOT-05 [6, nous verrons un grand nombre de FN].
- **AssA** : l'association mesure l'efficacité avec laquelle un tracker relie les détections au fil du temps dans les mêmes identités (ID). Dans la séquence MOT20, le modèle fait généralement un bon travail de suivi des mêmes identifiants pour la même personne au cours des vidéos. Une fois de plus, le score diminue à mesure que la difficulté augmente et que le nombre de personnes présentes dans la scène s'accroît.
- **LocA** : la localisation mesure l'alignement spatial entre une détection prédictive et une détection réelle en calculant l'indice d'utilité. C'est la meilleure mesure que nous ayons obtenue en termes de score, avec une moyenne de 80% dans toutes les vidéos. Cela signifie que les personnes détectées sont bien détectées et que leurs boîtes de délimitation correspondent en grande partie aux boîtes de vérité. Dans ce cas, YOLOv9 obtient donc de très bons résultats.

Si nous examinons d'autres indicateurs :

- **DetRe** : Le rappel de détection mesure la capacité d'un tracker à trouver toutes les détections de vérité au sol. Nous pouvons constater que le mode ne parvient pas à trouver toutes les

détections, avec un score moyen de 14.13 dans YOLOv9.

- DetPr : La précision de la détection mesure la capacité d'un tracker à ne pas prédire des détections supplémentaires qui n'existent pas. Dans ce cas, la moyenne combinée est de 77.87, ce qui signifie que le mode ne détecte généralement pas d'objets supplémentaires.
- De même, si nous examinons maintenant le rappel d'association (AssRe) et la précision d'association (AssPr), nous constatons un comportement similaire à celui de la détection : le modèle n'est pas très bon pour mesurer la capacité des suiveurs à éviter de diviser le même objet en plusieurs pistes plus courtes (AssRe), mais il est en revanche très bon pour mesurer la capacité des suiveurs à éviter de fusionner plusieurs objets en une seule piste (AssPr).
- Enfin, nous pouvons observer la métrique RHOA qui est calculée en prenant le rappel de détection au lieu de la précision de détection et qui nous donne presque le même résultat puisque DetA et DetRe sont presque identiques.

6.4 Hyperparamètres

Comme nous pouvons le voir dans le tableau 4, les meilleurs HOTA utilisant YOLOv8 sont obtenus en utilisant un seuil de confiance de 10% et un seuil d'incertitude de 70%. D'autre part, le meilleur HOTA(0) est obtenu avec un seuil de confiance de 10% également, mais avec un seuil d'incertitude de 90%. Malgré cela, les résultats sont presque similaires, c'est pourquoi nous utiliserons un seuil de 70% pour la détection des tasses. Le problème des seuils élevés est qu'ils créent généralement plusieurs boîtes de délimitation pour le même objet, ce qui rend le modèle moins précis et l'empêche de correspondre à chaque objet et de le suivre. D'autre part, nous pouvons sélectionner un seuil de confiance de 10%, car il n'a pas d'effet négatif sur les performances et n'entraîne généralement pas de faux positifs, mais il permet de détecter les objets éloignés qui ne sont pas très clairs pour le modèle.

6.5 Hypothèse sur la séquence sur Moodle

Malgré le faible score de HOTA obtenu, nous pensons que le modèle a mieux performé sur la séquence de Moodle. Les principaux problèmes liés à la performance sur le jeu de données MOT20 ne sont pas aussi présents que ceux dans la vidéo des verres. Par exemple, les verres sont beaucoup plus rapprochées de la caméra que les personnes, ce qui réduit le niveau de difficulté lié à l'échelle. De plus, l'éclairage est beaucoup plus intense dans la vidéo des verres, ce qui facilite la détection d'objets.

7 Conclusion

Le but de ce travail était d'implémenter une approche de suivi d'objet multiples (MOT). Nous nous sommes servi du modèle YOLO effectuer une détection d'objets, suivi de l'algorithme Byte-Track pour suivre les objets. La séquence que nous avons eu à analyser est un enregistrement d'une personne qui déplace des verres dans son bureau. Nous avons testé notre méthode sur l'ensemble de données MOT20 et avons obtenu des scores de HOTA assez faibles, principalement à cause de problèmes d'échelle et d'occlusions.

Nous pourrions sans doute apporter des améliorations à notre approche. Par exemple, nous pourrions effectuer un ré-entraînement du modèle YOLO sur un jeu de données contenant des verres similaires à ceux dans la vidéo. Ceci permettrait de raffiner la détection des objets ainsi que la précision des boîtes englobantes.

Références

- [1] RangeKing. Model structure of yolov8 detection models(p5). <https://github.com/ultralytics/ultralytics/issues/189>, 2023.
- [2] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9 : Learning what you want to learn using programmable gradient information, 2024.
- [3] Ben Le. An introduction to bytetrack : Multi-object tracking by associating every detection box. <https://www.datature.io/blog/introduction-to-bytetrack-multi-object-tracking-by-associating-every-detection-box>, 2023.
- [4] Glenn Jocher. Ultralytics yolov8. <https://github.com/ultralytics/ultralytics>, 2024.
- [5] Roboflow. Supervision : Democratising computer vision. <https://github.com/roboflow/supervision>, 2024.
- [6] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota : A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2) :548–578, October 2020.
- [7] Patrick Dendorfer Philip Torr Andreas Geiger Laura Leal-Taixe Jonathon Luiten, Aljosa Osep and Bastian Leibe. Hota : A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 2020.