INF6804 - Vision par ordinateur

Hiver 2024

TP3 – SUIVI D'OBJET

Nicolas Dépelteau - 2083544

Sebastien Goll - 2231054

Soumis à : Guillaume-Alexandre Bilodeau

12 avril 2024

Table des matières

2 Introduction

Le but de ce TP est d'exécuter une tache de suivi d'objet, ce type de tâche consiste en deux parties :

- La détection, qui permet de détecter les objets d'intérêt sur une frame.
- Le suivi, qui consiste à associer les objets détectés sur l'image actuelle aux objets détectés sur les images précédentes.

La difficulté de la tache vient de la phase de suivi, les objets doivent pouvoir être identifiés de façon unique et doivent être suivi entre les différentes frames de la vidéo, ce qui nécessite des techniques et modèles qui sont différents et plus complexes qu'une simple tache de détection/segmentation.

Dans notre cas, nous allons devoir appliquer ce processus de suivi à une vidéo sur laquelle les objets d'intérêt sont de tasses.

Description de la solution

Afin de réaliser cette tâche de suivi d'objets, deux différentes approches se sont offertes à nous : la création d'un modèle et processus de zéro ou l'utilisation de modèles et processus déjà développés pour la tâche.

Nous nous sommes donc intéressés aux modèles ayant de bonnes performances sur les taches MOT17 et MOT20 (deux taches de suivi d'objets (multi-object tracking)). Nous nous sommes intéressés à la méthode ByteTrack qui est (au moment de l'écriture de ce rapport) 8ème sur MOT17 et 3ème sur MOT20.

ByteTrack est une méthode de suivi d'objet qui utilise un modèle de détection pour la détection des boites englobantes et une phase de suivi par la méthode BYTE qui est une technique algorithmique d'association d'objet.

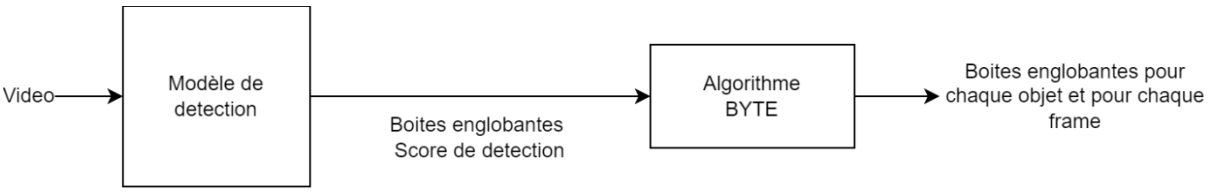

Figure 1 : Architecture de la méthode ByteTrack

Un des avantages de ByteTrack est l'adaptabilité de cette méthode, grâce a la structure de ByteTrack, n'importe quel modèle de détection peut être utilisé tant que le format de sortie du modèle est celui attendu par la méthode (une liste de boites englobantes et leurs valeurs de certitude). Cette flexibilité permet d'adapter facilement ByteTrack a tout type de situation puisque le modèle peut être choisi selon ses performances (fine tuned a la tache précise que l'on veut) et aussi selon le temps d'inférence du modèle selon le matériel utilisé (pour pouvoir attendre du 30fps par exemple).

L'algorithme BYTE a comme particularité de prendre en compte toutes les boites englobantes, quel que soit le score de détection de ces boites englobantes. BYTE fait d'abord

une phase d'association sur les boites englobante à haut score de détection, puis, essaye de faire des correspondances entre les boites englobantes ayant un score faible (en dessous d'un seuil) et les objets détectés sur les frames précédentes qui n'ont pas été associes sur l'image actuelle.

ByteTrack dépend donc énormément du modèle qui est utilisé afin de faire la détection, les performances de la méthode va directement dépendre de la précision de détection, si un des objets n'est plus détecté pendant quelques images, ByteTrack peut le considérer comme un nouvel objet étant apparu dans la scène.

Indentification des difficultés

Sur la vidéo, de nombreux passages peuvent être difficile pour le modèle et pour la tache de suivi d'objets.

Gros plan

Sur certaines images de la vidéo, les tasses sont filmées en très gros plan. Le modèle peut rencontrer des difficultés à détecter la tasse dans ces images puisqu'elle est filmée de manière inhabituelle.

Figure 2 : Frame 1350 : Gros plan sur une tasse

Tasse par transparence

Dans la vidéo, certaines tasses sont parfois visibles par transparence à travers les verres d'eau qui sont manipulés. Nous pensons que, comme les gros plans, ce sont des situations rarement filmées et qu'il est possible que le modèle de détection n'arrive pas à bien détecter les tasses visibles à travers ces verres.

Figure 3 : Frame 810 : Tasse vue par transparence

Objet ressemblant à une tasse

Sur la table, certains objets peuvent ressembler en forme a une tasse, nous pensons notamment à la bouilloire qui, comme les tasses, a une anse et qui peut donc ressembler a une tasse du point de vue du modèle de détection.

Objet sortant du champ

La tache que nous devons accomplir consiste à assigner un identifiant unique à chaque objet, cependant, ByteTrack n'est pas prévu pour tenir cette identification unique sur les objets se trouvant hors-champ. Nous nous attendons donc à observer un changement d'identifiant lors qu'un objet re-entre dans le champ. Pour garder un identifiant unique même hors champ, il est nécessaire d'ajouter un modèle d'identification supplémentaire permettant d'identifier uniquement chaque tasse.

Boites englobantes superposées

Comme l'algorithme BYTE ne se base que sur les positions des boites englobantes afin de réaliser l'association et le suivi, une difficulté pourrait provenir de la superposition de plusieurs boites englobantes. Dans ce cas, on peut se demander si BYTE arrive à maintenir l'identifiant donné à chaque tasse.

Justification de la méthode

Afin de surmonter les difficultés liées la détection des objets, nous avons décidé d'utiliser le modèle YOLOX puisque nous savons que les modèles YOLO sont des modèles très performant dans la détection d'objet en temps réel. Le modèle YOLOX est capable et entrainé pour détecter les objets de classe « cup », ce qui est obligatoire pour la tache que nous devons accomplir. Comme YOLOX est un modèle performant, nous pensons qu'il sera capable de surmonter la majorité des difficultés liée à la détection mentionnées dans la section 0. Nous nous attendons aussi que YOLOX ne sera peut-être pas capable de parfaitement détecter toutes les tasses dans les situations les plus compliquées, mais nous pensons que YOLOX sera quand même en capaciter de correctement détecter les objets dans la majorité des cas.

Pour la partie suivi/association d'objet, nous utilisons une des méthodes ayant les meilleurs résultats sur les taches de suivi d'objet. Nous sommes donc convaincus que cette méthode pourra fournir de très bons résultats quand appliqué a notre cas. Puisque les taches MOT17 et MOT20 proposent des situations similaires a celle discutée dans la section 0, nous nous attendons donc à observer une stabilité des identifiants lors de cette situation. Cependant, comme mentionné dans la section 0, la méthode ByteTrack et l'algorithme BYTE n'est pas du tout fait pour faire du suivi d'objets hors plan et ne peut donc pas du tout reconnaitre un objet revenant dans le champ de la caméra.

Les bons résultats attendus des deux parties clés de la méthode ByteTrack (modèle de détection YOLOX et algorithme BYTE pour le suivi d'objet) sont les raisons principales pour notre choix d'utiliser ces technologies pour la tache de suivie de tasses dans notre vidéo.

Description de l'implémentation

Implémentation

L'implémentation choisis provient d'un code bifurquer (Zhang, 2022) que nous avons modifiés pour utiliser le *backbone YOLOX* sur les 80 classes de *COCO*. On a effectué une modification quant à filtrer les prédictions pour garder seulement la classe *cup* qui correspond à la détection des tasses. Nous avons également réglé quelques soucis d'implémentation qui y étaient présent. Nous avons également dû utiliser notre propre déploiement local avec *GPU* du conteneur officiel de Google Colab pour l'exécution. Nous avons ensuite, tester sur l'ensemble de donnée *MOT17* pour obtenir les métriques. Pour cela, nous avons dû bifurquer un script d'évaluation (Jonathon Luiten, 2020) pour calculer les métriques selon le fichier de résultat obtenir par *ByteTrack*. Le tout est présent dans un *notebook jupyter*.

Paramètres principaux

Le paramètre principal est de quel *backbone* utilisé. Nous avons choisi *YOLOX* avec les poids préentraîner sur *COCO* pour des raisons que les poids étaient disponibles et que l'architecture *YOLOX* était déjà présente dans le code source utilisé. Les autres paramètres sont restés ceux par défauts déterminer par les auteurs du papier. Cela inclue entre autres un seuil de 0.7 pour l'algorithme de suppression non-maximale (NMS).

Présentation des résultats

Les métriques dans le Tableau 1 Métriques sur l'ensemble de donnée MOT17ont été obtenue sur l'ensemble de donnée MOT17 avec la détection sur seulement les piétons. Cela induit un biais à la baisse. Le modèle n'est également pas *finetune* à la tâche en question. Nous avons utilisé le même modèle pour notre résultat sur les vidéos de validation ainsi que celle de ce travail pratique. Ce sont donc les performances d'un modèle générique.

Tableau 1 Métriques sur l'ensemble de donnée MOT17

Métrique	Valeur (%)
HOTA	15.34
HOTA(0)	18.53
DetA	6.24
AssA	37.79
DetRe	6.27
DetPr	82.91
AssRe	40.46
AssPr	83.38
LocA	84.63
OWTA	15.39
LocA(0)	81.23
HOTALocA(0)	15.05

Les figures Figure 0.1 Exemple de détection avec une fausse détection et Figure 0.2 Exemple d'un changement d'identifiant lorsqu'une tasse re rentre dans le champ de vision montre un exemple où la tasse rouge est sortie du champ de vision de la caméra pour ensuite revenir dans le champ de vision par la suite.

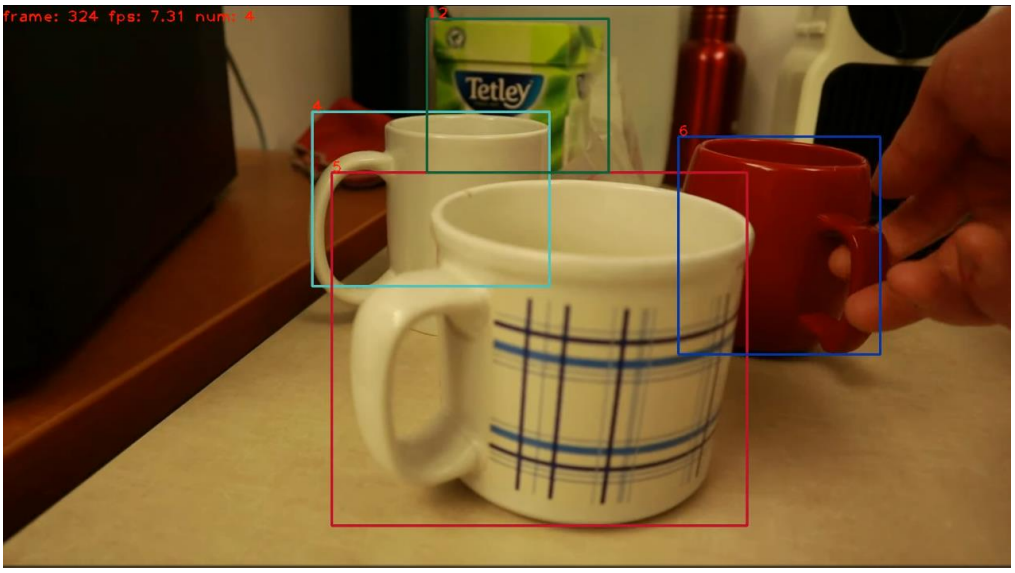

Figure 0.1 Exemple de détection avec une fausse détection

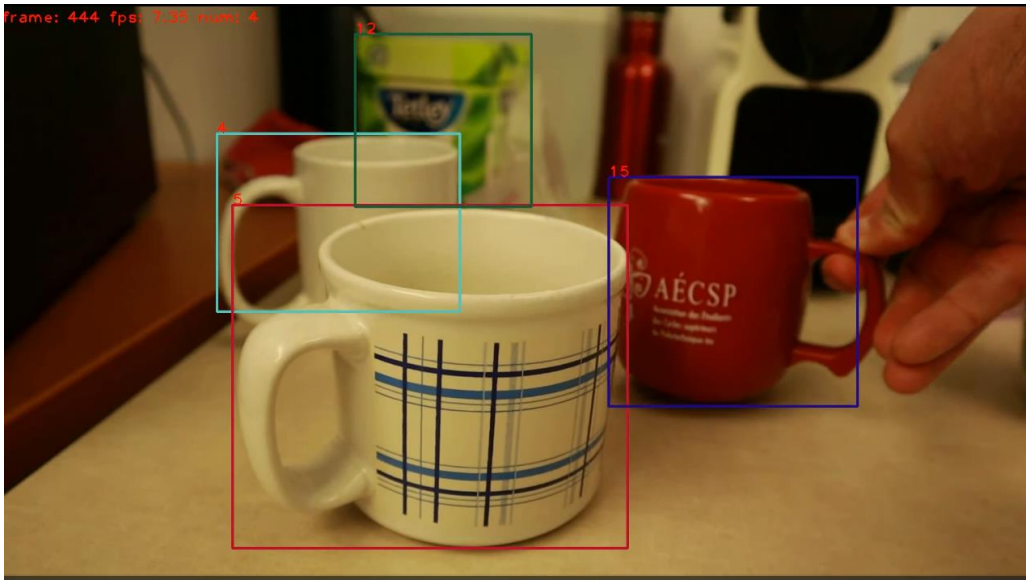

Figure 0.2 Exemple d'un changement d'identifiant lorsqu'une tasse re rentre dans le champ de vision de la caméra

La Figure 0.3 Exemple de fausse détection pour les objets qui ressemblent à une tasse montre un exemple d'une fausse détection sur la tasse numéro 29 qui est en fait une cafetière.

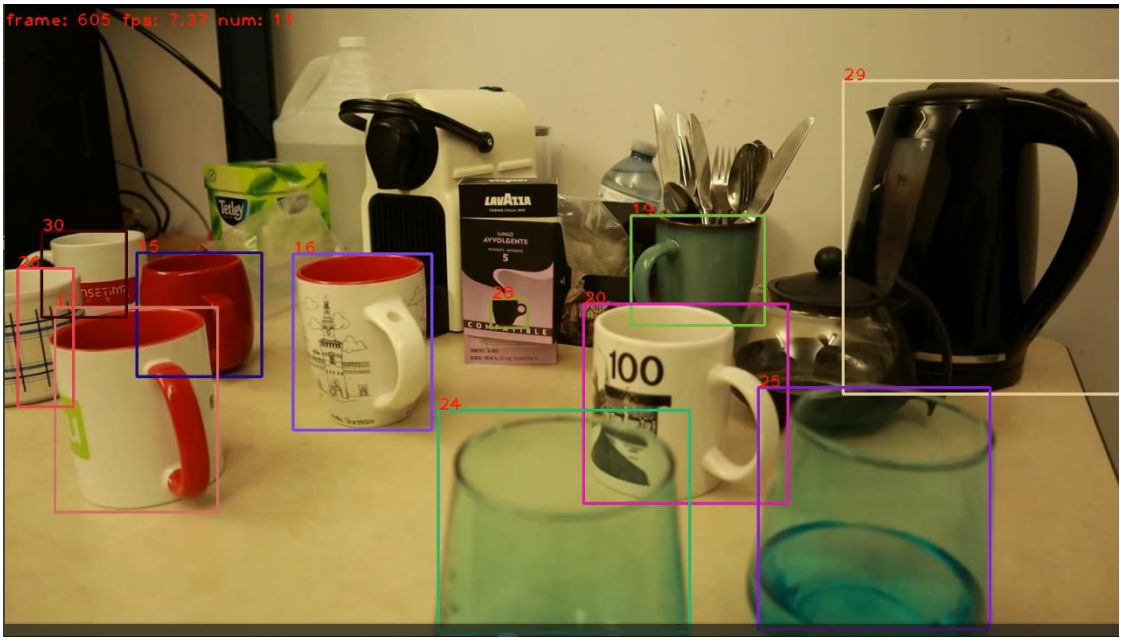

Figure 0.3 Exemple de fausse détection pour les objets qui ressemblent à une tasse

La Figure 0.4 Exemple d'occlusion et de rotation montre un exemple d'occlusion du verre bleu sur la tasse 20 en plus d'une rotation du verre en question.

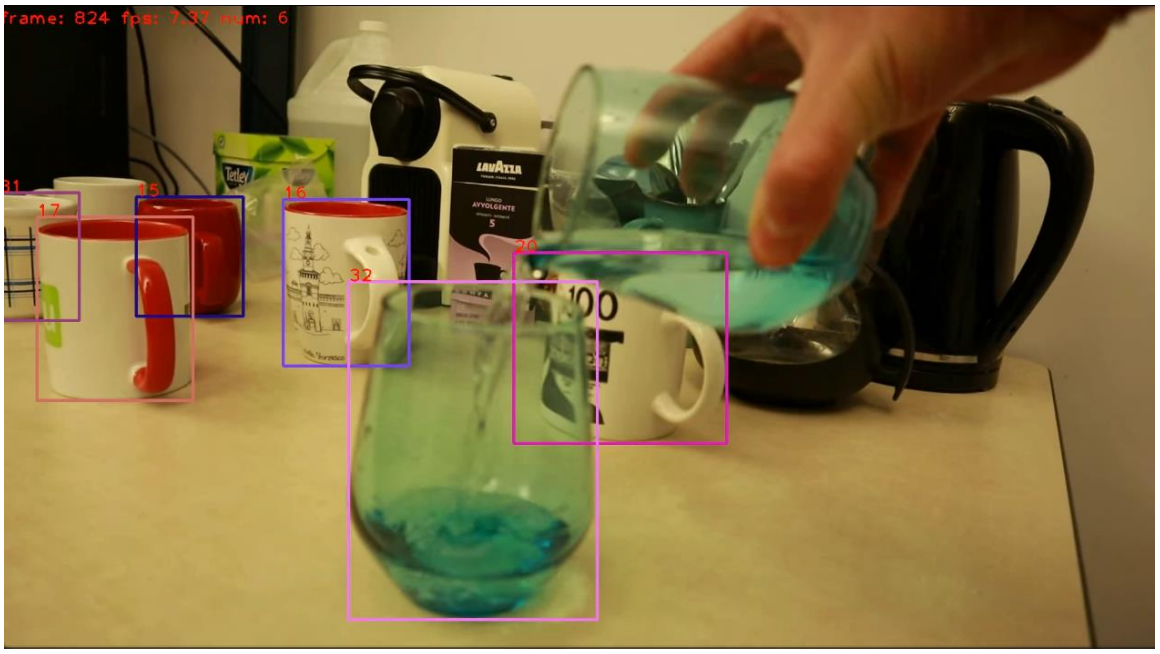

Figure 0.4 Exemple d'occlusion et de rotation

Les figures Figure 0.5 Exemple de mise à l'échelle de petite taille et Figure 0.6 Exemple de mise à l'échelle de grande taille sont des exemples où la mise à l'échelle de petit à grand est effectuée.

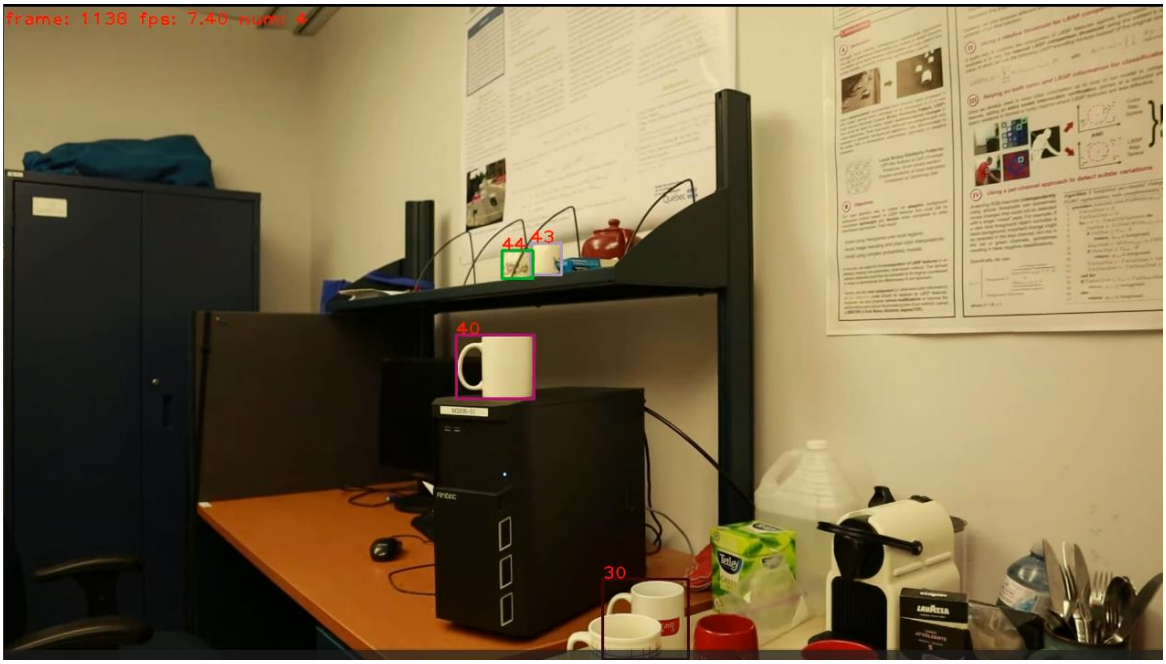

Figure 0.5 Exemple de mise à l'échelle de petite taille

Figure 0.6 Exemple de mise à l'échelle de grande taille

Discussion des résultats

Différence entre HOTA et HOTA (0)

La valeur d'HOTA est 15.34 et celle de HOTA (0) est de 18.53. La valeur est plus élevée pour HOTA (0) puisque le seuil de localisation α est de 5%, donc il est permis d'avoir les détections qui ont une localisation supérieure ou égale à 5%. Il y a donc plus d'association, donc la métrique est plus élevée.

Forces de notre méthode

Les forces de notre méthode sont les tasses d'une image à l'autre sont bien identifié et garde le bon identifiant lorsque la détection est bonne. Notre méthode est bonne pour les objets qui subissent une occlusion comme démontré dans la Figure 0.4 Exemple d'occlusion et de rotation.

Faiblesse de notre méthode

Les faiblesses de notre méthode qu'il est sensible aux fausses détections. Par exemple, dans la Figure 0.3 Exemple de fausse détection pour les objets qui ressemblent à une tasse, notre méthode fait le suivi d'un objet qui ne correspond pas à une tasse. Notre méthode est également sensible à la perte temporaire d'une tasse dans le champ de vision de la caméra. En effet, dans les figures Figure 0.1 Exemple de détection avec une fausse détection et Figure 0.2 Exemple d'un changement d'identifiant lorsqu'une tasse re rentre dans le champ de vision de la caméra, on peut remarquer que la tasse rouge change d'identifiant lorsqu'elle réapparait dans la vidéo, et cela est vrai pour tous les tasses qui ont le même comportement. Une dernière faiblesse de cette méthode est la mise à l'échelle rapidement peut arrêter la détection de la tasse en gros. En effet, dans les figures Figure 0.5 Exemple de mise à l'échelle de petite taille et Figure 0.6 Exemple de mise à l'échelle de grande taille, la tasse sur l'ordinateur n'est plus tracker lorsqu'elle est prend tout le champ de vision de la caméra.

Retour sur les difficultés

Avec les résultats obtenus et présentés dans la section 0, nous pouvons confirmer les hypothèses que nous avions émises dans la section 0 par rapport aux difficultés de la séquence vidéo que nous avons analysé.

Tasse en gros plan :

Comme nous pouvons le voir dans la Figure 0.6, le modèle de détection a effectivement des difficultés à détecter les tasses lorsque celle-ci sont en très gros plan sur l'image. On peut observer quand même la qualité du modèle de détection en regardant la dernière image sur laquelle la tasse est détecté, cette image se trouve dans la Figure 7. On remarque que le modèle est quand même capable de détecter la tasse lorsqu'elle se trouve très proche de la caméra, ce qui nous conforte dans notre choix de modèle.

Figure 7 : Limite de détection de la tasse en gros plan

Tasse par transparence :

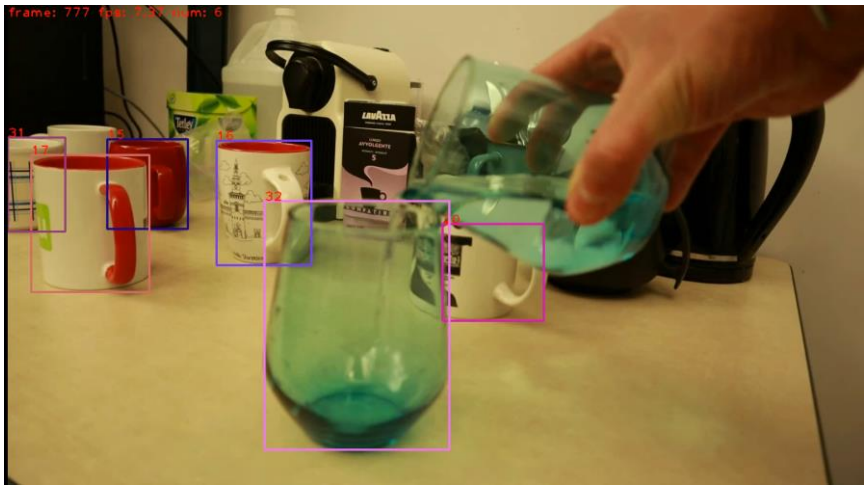

Figure 8 : Cas de détection de tasse par transparence

Dans la Figure 8, on peut observer comment le modèle réagit lorsque qu'une tasse (la numéro 20) est visible par transparence a travers les vers d'eau au premier plan. Bien que celle-ci soit

peu visible, le modèle arrive quand même à la détecter et ne perd jamais le tracking de cette tasse tout au long de la séquence. On remarque toutefois que, dans l'image ci-dessus, la boite englobante ne correspond pas tout à fait à la vraie tasse, cela est surement du au fait que, pour le modèle, la tasse est occluse et que la partie visible en transparence n'est pas considérée comme faisant partie de l'objet. Cette situation montre la force du modèle particulièrement lorsque l'objet est occlus mais n'arrive pas forcément à comprendre la partie visible en transparence.

Objet ressemblant à une tasse :

Comme le montre la Figure 0.3, la bouilloire se trouvant dans l'arrière-plan est détecté comme étant une tasse, ceci est surement du au fait que la forme des deux objets est similaire, et comme la bouilloire est en arrière-plan, le modèle la confond avec une tasse. Ceci est aussi du au fait que le modèle que nous utilisons n'est pas entrainé à détecter les bouilloires, expliquant d'autant plus pourquoi YOLOX a du mal à différencier les deux types objets.

Objets sortant du champ :

Sur la Figure 0.1, on peut voir la tasse rouge AÉCSP avec un identifiant initial de 6, puis la tasse est retirée du champ de la caméra et puis, sur la Figure 0.2, la tasse est remise dans le champ. On remarque que l'identifiant a changé lorsque la tasse a disparue, ceci est du aux limitations fondamentales de la méthode utilisé qui se base sur la position des boites englobantes précédentes afin de faire la correspondance temporelle entre les objets et entre les images. Comme la méthode n'a aucun moyen d'identifier uniquement les objets à partir de leur apparence seulement, ces résultats étaient tout à fait prévus et correspondent à nos prévisions.

Boites englobantes superposées :

Sur l'entièreté de la vidéo, le modèle ne montre aucune difficulté à identifier les différents objets même lorsque leurs boites englobantes sont superposées. Cela prouve la force de la méthode ByteTrack utilisée.

Performance spéculée sur la séquence sur Moodle

On s'attend à des performances accrues pour la séquence vidéo sur Moodle puisque le *backbone* identifie bien les tasses. On a seulement quelques fausses détections et les identifiants qui changent. Malgré cela, on peut voir visuellement sur les prédictions que ça performe bien. Les performances sur l'ensemble de validation sont moins bonnes puisque les détections étaient que sur les piétons, ce qui n'est pas le même type d'objet que les tasses.

Bibliographie

Jonathon Luiten, A. H. (2020). *TrackEval*. Récupéré sur https://github.com/Depdx/TrackEval

PyTorch. (s.d.). Récupéré sur PyTorch: https://pytorch.org/

wandb. (s.d.). *wandb*. Récupéré sur Github: https://github.com/wandb/wandb

Yadan, O. (2019). *Hydra - A framework for elegantly configuring complex applications*. Récupéré sur Github: https://github.com/facebookresearch/hydra

Zhang, Y. a. (2022). *ByteTrack: Multi-Object Tracking by Associating Every Detection Box*. Récupéré sur https://github.com/Depdx/ByteTrack

ZHANG, Y. S. (2022). ytetrack: Multi-object tracking by associating every detection box. *European conference on computer vision*, 1-21.