Travail pratique #2

INF6804 - Vision par ordinateur

18 mars 2024

Émile Poirier 2014031

Ariste Kwawe-K-Arnov 2317869

Table des matières

Introduction

Ce deuxième rapport de laboratoire dans le cadre du cours de vision par ordinateur porte sur le sujet de la segmentation d'objets vidéo. Cette tâche consiste à identifier dans un vidéo la position et le déplacement de régions d'intérêts selon différentes classifications. L'objectif du travail était de se familiariser avec deux différents algorithmes de segmentation de régions d'intérêts, soit : la soustraction d'arrière-plan et la segmentation d'instances. Ce rapport contient donc les détails de nos implémentations ainsi que de nos résultats des deux méthodes en utilisant la vase de données CDNET 2012. Nous décrirons nos hypothèses initiales et nos conclusions face à ces dernières et nous comparerons les avantages et désavantages des deux méthodes.

Présentation des deux méthodes

Soustraction d'arrière-plan

La soustraction d'arrière-plan est une des façons les plus intuitives de s'attaquer au problème de détection de mouvement. Comme son nom l'indique, le but est de soustraire une référence connue d'arrière-plan à une image où on pourrait y retrouver du mouvement. Ceci permet donc d'isoler les pixels qui sont modifiés et ainsi obtenir de l'information sur les objets dans l'image.

Il s'agit donc ici d'une tâche de segmentation à deux niveaux : l'arrière-plan et l'avant-plan (Bilodeau, 2024). Il est important de noter que cette tâche ne peut pas être réalisée avec une seule image. Il faut minimalement une image de référence. Dans le contexte de segmentation de vidéo, la méthode demande donc de sélectionner une ou plusieurs trames ne contenant aucun sujet afin de générer la référence d'arrière-plan. L'implémentation la plus simple serait de seulement soustraire une image à la référence et garder les pixels qui sont au-delà d'un seuil, mais cette façon de procéder est très vulnérable au bruit d'une caméra. Comme mentionné dans les notes de cours (Bilodeau, 2024), il existe deux familles de techniques permettant d'obtenir la référence afin d'améliorer drastiquement les résultats :

1. La méthode paramétrique : on assume ici qu'un bruit blanc gaussien affecte les images et il faut donc prendre plusieurs trames afin d'identifier la moyenne et la variance du bruit pour créer une fonction de probabilité autour de la référence.
2. La méthode non paramétrique : on n'assume pas que le bruit est gaussien et on utilise que des échantillons d'image de la vidéo pour créer la référence. On vient donc comparer une trame à plusieurs références et on pose un seuil basé sur des histogrammes mesurés.

Segmentation d'instances

La segmentation d'instances est une tâche d'identification d'objets d'intérêts où les différentes classes ainsi qu'individus par classes sont identifiés indépendamment. Cette tâche est généralement accomplie par un réseau de neurones convolutifs (Bilodeau, 2024). Il existe de multiples modèles pouvant réaliser cette tâche avec différentes architectures, mais pour ce laboratoire, nous nous sommes concentrés sur le modèle discuté en classe, soit les R-CNN.

Cet algorithme fonctionne en séparant l'image en région qui sont qui pourraient contenir des objets pour ensuite passées dans un réseau convolutif qui vient extraire des vecteurs caractéristiques qui sont alors utilisés dans un algorithme de classification (Potrimba, 2023). On identifie donc deux étapes : l'extraction des régions puis la classification des objets.

Dans le contexte de notre travail, nous ne nous intéressons qu'à un certain type d'objets. Ainsi, nous ajoutons des filtres sur type de classification et le taux de certitude par rapport à cette classification pour extraire les masques qui définissent nos régions d'intérêts. La figure ci-dessous démontre bien ces étapes :

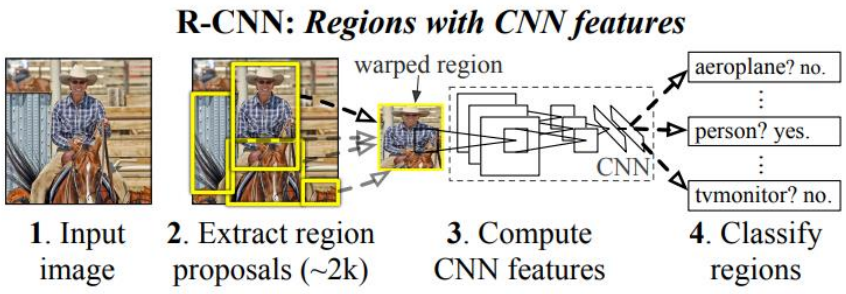


Figure 1 OBJECT DETECTION SYSTEM OVERVIEW. De « Rich feature hierarchies for accurate object detection and semantic segmentation » par Girshick, R., Donahue, J., Darell, T., Malik, J. (2014). UC Berkeley.

Puisque cette méthode utilise un algorithme de classification sur chaque région identifiée, on obtient alors des masques indépendants pour chaque instance d'objet détecté dans l'image originale. On n'obtient donc pas une classification binaire par pixel comme avec la soustraction d'arrière-plan, mais plutôt une classification selon des labels pour des regroupements de pixels.

Hypothèses de performances dans des cas spécifiques

Occultation

Si elle est partielle, la segmentation d'instance par les réseaux convolutifs apprend à mieux reconnaître les formes de chaque instance et alors donnera un meilleur résultat que la soustraction d'arrière-plan qui prend en considération chaque pixel. En effet, la soustraction d'arrière-plan ne pourra jamais déduire le reste de la forme détectée, le masque de détection sera toujours limité par ce qui est visible. Cependant, si l'occultation est trop importante, la segmentation d'instance ne sera pas capable de détecter le type d'objet et indiquera donc une non-détection. Dans cette situation, la soustraction d'arrière-plan serait donc supérieure puisque même si l'objet n'est pas complètement identifié, une détection est meilleure que de ne rien avoir.

Mouvement d'objets

Lorsqu'une caméra filme un objet en mouvement, si la vitesse d'obturation est trop basse, regarder les images trame par trame aura pour effet de montrer un sujet très flou. Ainsi, pour un algorithme de segmentation d'instance, le sujet pourrait être complètement inidentifiable puisqu'il faudrait le contexte des autres images pour comprendre de quoi l'objet s'agit. On peut donc poser l'hypothèse qu'une méthode de soustraction d'arrière-plan serait beaucoup plus utile pour détecter des objets en mouvement rapide.

Vulnérabilité face au bruit

Puisque la méthode de soustraction d'arrière-plan se base sur une référence d'arrière-plan connue et statique, on peut poser l'hypothèse que cette méthode sera beaucoup plus vulnérable au bruit. Effectivement, si la luminosité de la scène change plus loin dans la vidéo, soustraire l'arrière-plan aura pour effet d'indiquer des changements pour des pixels pas réellement affectés par du mouvement. La segmentation d'instance ne se base pas sur une référence statique donc l'algorithme devrait fonctionner correctement, peu importe la quantité de bruit blanc dans l'image.

Description des expériences

Pour tester et comparer les deux méthodes, la stratégie est d'utiliser les deux méthodes sur l'ensemble des trames disponibles et de comparer les résultats avec les trames fournies dans les dossiers groundtruth. Pour faire une évaluation qualitative, nous nous baserons sur la qualité des vidéos produites, en comparant la forme des objets identifiés, la quantité d'erreurs et les effets non voulus tels que le "flickering" ou la fusion de plusieurs objets.

Pour comparer quantitativement les performances des deux différentes méthodes utilisées, nous avons choisi deux métriques : l'erreur quadratique totale et l'intersection sur union (IoU). La première mesure est obtenue en faisant la somme de l'erreur au carré de chaque pixel :

$$Err = \sum_{i=1}^{m} \sum_{j=1}^{n} (image(i,j) - groundtruth(i,j))^2$$

Cette norme nous permet d'avoir une idée de la distance entre l'image avec les masques obtenue et celle avec les masques de référence. Plus l'erreur est grande, plus les masques obtenus sont différents. La deuxième métrique permet de connaître le pourcentage de la surface des masques de référence qui a été correctement identifiée:

$$IoU = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} (image(i,j) \cap groundtruth(i,j))}{\sum_{i=1}^{m} \sum_{j=1}^{n} (image(i,j) \cup groundtruth(i,j))}$$

Cette métrique permet de savoir le pourcentage des masques correctement identifiés, lorsqu'aucun masque n'est présent dans l'image de référence, nous avons simplement mis le score à 100%.

De plus, pour valider nos hypothèses, nous avons sélectionné des trames que nous jugions pertinentes où des événements spéciaux arrivaient et où le comportement des deux méthodes diffère. Pour tester l'occlusion, nous avons choisi des trames dans « pedestrians » et « PETS2006 » puisqu'on peut y retrouver des moments où des sujets sont cachés derrière les autres. Pour tester le mouvement des objets, nous avons choisi des trames dans « Highway » et dans "pedestrians" puisque dans ces deux vidéos des objets se déplacent assez rapidement (voiture et vélo). Pour observer l'effet du bruit, il faudra se fier à l'entièreté des vidéos obtenues avec les deux méthodes puisque cette caractéristique est indépendante de ce qu'il se passe dans les scènes.

Description des deux implémentations utilisées

Soustraction d'arrière-plan

Pour cette méthode, nous nous sommes inspirés du code fourni dans les exemples "SingleGaussianBGS" et "TemporalAvgBGS". La stratégie était de trouver des images dans les vidéos ou ne se trouvaient aucun sujet, donc juste de l'arrière-plan. Pour les vidéos « Highway », « Office » et « Pedestrians », il était possible de trouver des trames vides au début, au milieu ou à la fin des images fournies. Le seul cas où il fut plus difficile d'identifier des trames vides fut le la vidéo « PETS2006 » où seules les 6 premières images étaient vides, ce qui venait réduire le nombre d'échantillons pour ce cas.

Par la suite, en utilisant une douzaine d'images d'arrière-plan (lorsque disponible), on peut obtenir la moyenne des images pour obtenir une meilleure référence, moins affectée par le bruit de la caméra. On peut également trouver la variance de chaque pixel en utilisant :

$$\mu = \frac{1}{k}\sum_{n=1}^{k} I(n)$$

$$\sigma^2 = \frac{1}{k}\sum_{n=1}^{k} I^2(n) - \mu^2$$

Pour obtenir les masques d'avant-plan, nous devons ensuite soustraire de chaque image la moyenne et garder tous les pixels en haut d'un certain seuil $n\sigma^2$, où n'est pas un paramètre permettant d'ajuster le filtrage. Cependant, en réalisant nos tests, nous nous sommes aperçus qu'intégrer la variance dans l'équation venait nuire au résultat. En effet, avec la variance, beaucoup de pixels devenaient identifiés comme faisant partie de l'avant-plan même si aucun objet n'était vraiment visible. Un filtre commun pour chaque pixel permettait donc d'avoir des résultats plus consistants. Pour choisir la valeur du filtre, nous avons testé différentes valeurs pour chaque vidéo et subjectivement choisi les résultats qui nous semblaient les meilleurs. Voici les valeurs finales choisies :

Vidéo	Filtre avec le meilleur résultat
Highway	50
Office	75
Pedestrians	50
PETS2006	25

Segmentation d'instance

Pour la segmentation d'instance, nous nous sommes encore inspirés du code du cours, particulièrement l'exemple « Mask_RCNN » et ainsi du code de la référence [4] (Chen, 2020) : « Instance Segmentation using Mask-RCNN and PyTorch ». Le modèle utilisé est un inclus dans la librairie de « torchvision », soit « maskrcnn_resnet50_fpn ». Le modèle est également préentrainé.

Pour chaque vidéo, on prend par la suite chaque image une à la fois et on les transforme en tensor et on vient obtenir les masques de prédiction à partir du modèle. On passe un filtre sur les labels pour aller chercher le type de sujet dont nous avons besoin puis on utilise un second filtre sur le score afin de seulement garder les prédictions qui sont réellement recherchées.

Une difficulté rencontrée fut lors de la sélection du filtre sur le score. Lorsque le filtre est trop bas, les prédictions peuvent se tromper et identifier des masques sur l'image qui ne correspondent pas réellement aux sujets recherchés. Lorsque le filtre est trop haut, les sujets peuvent ne pas être identifiés à toutes les trames, ce qui cause un effet de flickering lorsqu'on compile les images en vidéo. Le tableau suivant montre les filtres sur les labels et les scores choisis :

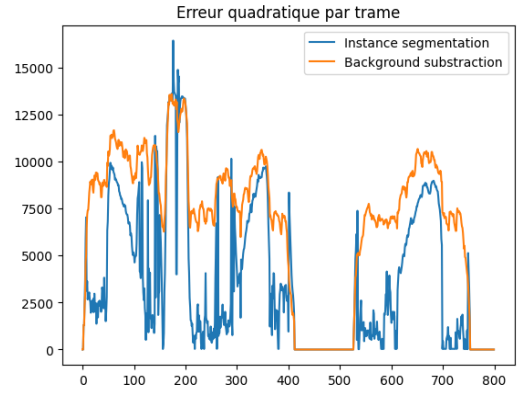
Vidéo	Filtre sur le label	Filtre sur le score
Highway	3 (voiture)	90 %
Office	1 (personne), 84 (livre)	98 %
Pedestrians	1 (personne), 2 (vélo)	96 %
PETS2006	1 (personne), 27 (sac à dos)	65 %

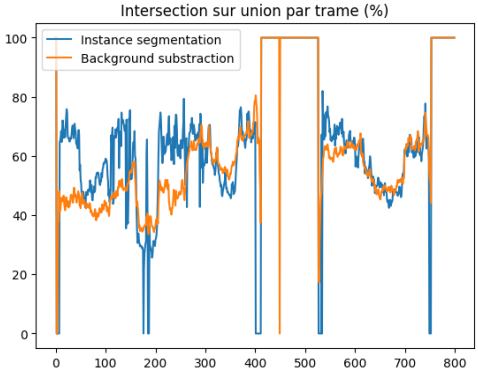
Le seul cas où le choix du filtre n'a pas permis d'avoir des résultats jugés satisfaisants était celui de la vidéo « Highway ». Pour ce cas, une ombre dans le côté de gauche de l'image était identifiée comme étant une voiture lorsque le filtre était trop bas, mais lorsqu'on montait le filtre assez haut pour l'ignorer, nous obtenions un gros effet de « flickering » sur les voitures détectées. Nous avons comme théorie que c'est le reflet sur les vitres des voitures qui vient causer ce problème. L'algorithme de segmentation ne peut pas savoir que le reflet n'est pas inclus avec la voiture donc il le supprime. Nous avons donc abouti sur un taux de 90%, mais sur une certaine trame le problème de l'ombre et de la non-détection des voitures persiste.

Présentation des résultats de tests

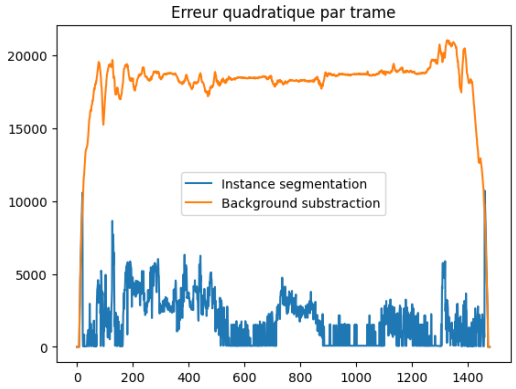
Les graphiques suivants sont générés en utilisant les deux métriques mentionnées plus haut, soit l'erreur quadratique et l'IoU. Pour générer des graphiques de comparaison, nous avons trouvé la valeur de ces métriques pour chaque trame des quatre vidéos fournies, en comparant les résultats obtenus avec les méthodes de soustraction d'arrière-plan et de segmentation d'instance par rapport aux images de « groundtruth » fournies.

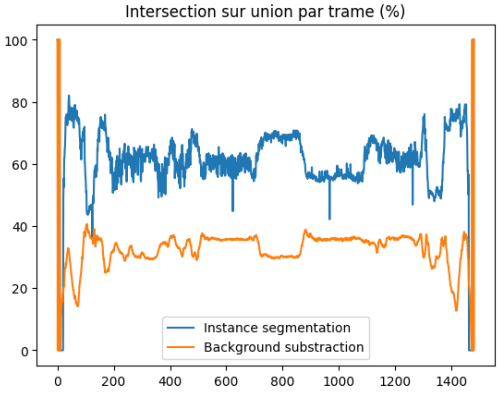
Pedestrians:



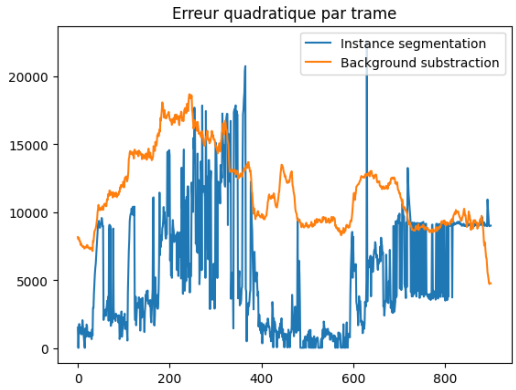


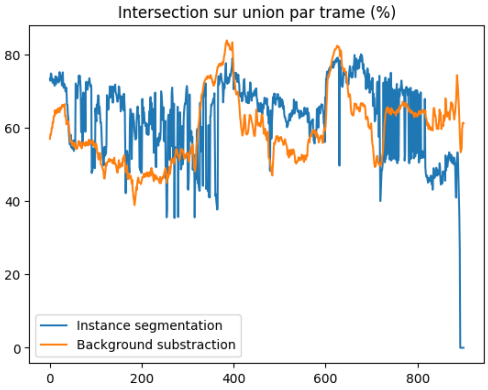
Office:



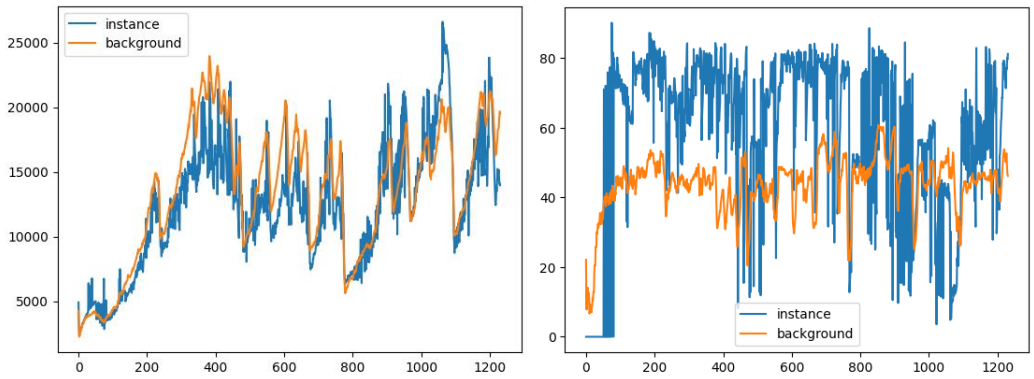


PETS2006 :



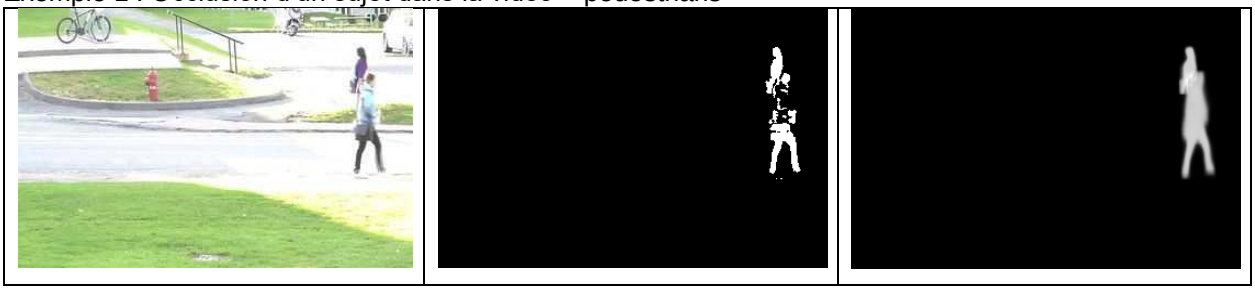


Highway :

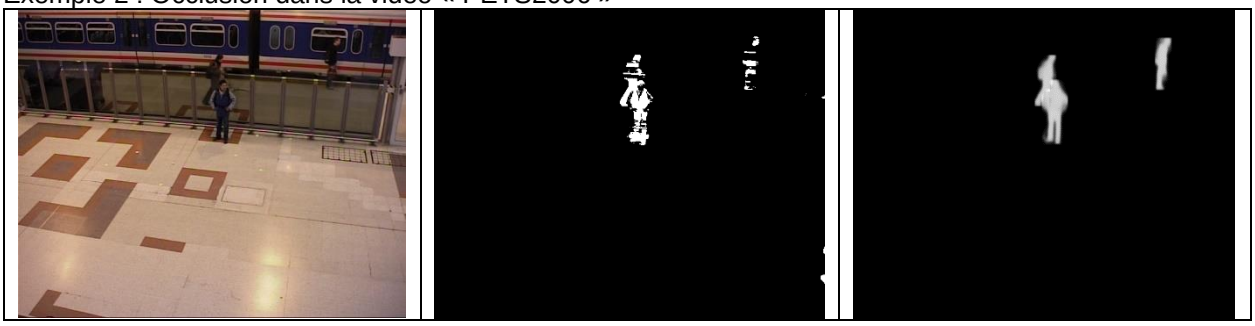


Les exemples suivants identifiant des trames où les résultats de deux méthodes nous permettent de conclure sur les hypothèses posées. Pour chaque exemple, nous avons mis dans l'ordre l'image originale, le résultat de la soustraction d'arrière-plan puis le résultat de la segmentation d'instance.

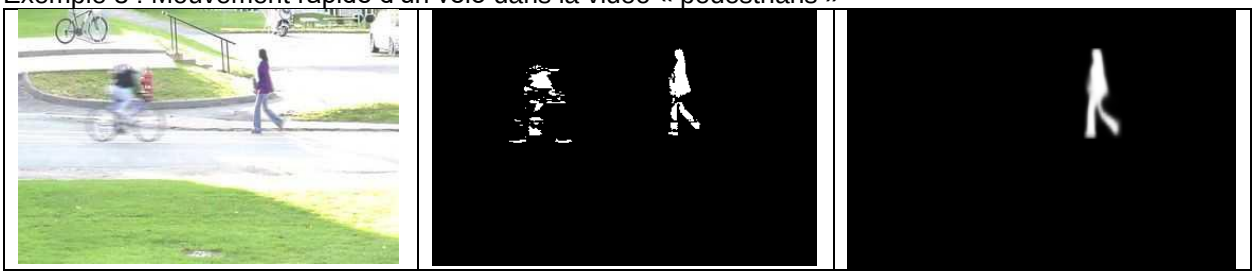
Exemple 1 : Occlusion d'un sujet dans la vidéo « pedestrians »



Exemple 2 : Occlusion dans la vidéo « PETS2006 »



Exemple 3 : Mouvement rapide d'un vélo dans la vidéo « pedestrians »



Exemple 4 : Bruit blanc sur la vidéo « Highway »

Exemple 5 : Bruit non identifié sur la vidéo « Highway »

Exemple 6 : Bruit causé par l'ombre sur « Pedestrians »

Finalement, un autre résultat pertinent à mentionner est l'aspect performances des deux méthodes. La méthode de soustraction d'arrière-plan se réalisait assez rapidement, avec un temps d'exécution autour de 20 secondes par vidéos, tout dépendamment du nombre de trames ainsi que de la taille des images. En revanche, la méthode de segmentation d'instances pouvait prendre jusqu'à 2h pour être exécutée sur une vidéo, ce qui est un temps drastiquement plus important qu'avec l'autre méthode.

Discussion des résultats et retour sur hypothèses

Les résultats quantitatifs présentés nous démontrent très bien l'erreur quadratique par trame est plus généralement plus élevé pour la méthode de soustraction d'arrière-plan. Ceci est explicable par le fait que la méthode de soustraction est très sensible au bruit et que même quand un objet est identifié, des parties du contour désiré seront quand même manquantes, comme on peut le voir dans les exemples précédents.

Pour la métrique de l'union sur l'intersection, on observe également une nette supériorité avec la segmentation d'instance. Le seul cas où le résultat est assez équivalent serait celui de la vidéo « Highway ». Comme mentionné plus tôt, l'ombre apparaissant comme une voiture ainsi que la non-détection des voitures à cause du filtre explique la grande oscillation dans les résultats. Une solution aurait été de mettre le masque plus bas et de venir manuellement ignorer le masque de l'ombre lorsqu'il était détecté.

Concernant les occultations, les exemples 1 et 2 des résultats nous montrent des situations où un sujet est entre partiellement et beaucoup caché. Pour les deux méthodes, on obtient des résultats où les deux sujets sont identifiés. Cependant, la segmentation nous donne deux masques indépendants nous permettant de bien séparer les deux instances alors que la soustraction d'arrière-plan ne nous donne qu'une région avec beaucoup de mouvement, mais sans être capable de distinguer les deux sujets. On peut donc conclure la segmentation d'instance est une méthode plus performante dans les cas d'occultation.

Pour le mouvement rapide des objets, le seul exemple que nous avons pu utiliser est celui où la bicyclette se déplace rapidement dans l'exemple 3. Les résultats concordent alors avec notre hypothèse pour ce cas spécifique : la méthode de segmentation d'instance n'est pas capable de correctement identifier le sujet alors que la méthode de soustraction d'arrière-plan l'est.

Notre dernière hypothèse concernait la présence de bruit dans les images. Les exemples 4 et 6 montrent bien comment du bruit blanc ou tout simplement un changement sur un sujet qui ne nous intéresse pas (ombre du piéton) sont des changements qui seront détectés par la soustraction d'arrière-plan même si on n'est pas intéressé par ces derniers. En revanche on remarque dans l'exemple 5 qu'une ombre est également détectée comme étant une voiture dans le résultat de la segmentation. On peut donc conclure qu'en général la segmentation d'instance est meilleure pour ignorer le bruit et les changements non pertinents puisque les labels des objets voulus doivent être sélectionnés, mais que malgré cela, il peut tout de même y avoir des erreurs.

Conclusion

Pour conclure, la comparaison des résultats nous a montré que la méthode de segmentation d'instance était supérieure lorsque nous nous concentrions sur des cas avec de l'occlusion ou du bruit, mais que la méthode de soustraction d'arrière-plan était mieux adaptée pour les cas où un sujet se déplaçait très rapidement. De plus, nous avons observé que la méthode de segmentation était nettement supérieure en termes de score IoU et en termes d'erreur quadratique, mais que ces résultats impliquaient également un temps d'exécution drastiquement plus long. Ce laboratoire nous a donc permis de nous familiariser avec l'implémentation de deux différentes méthodes d'extraction de régions d'intérêts sur vidéo en nous faisant comprendre leurs avantages et désavantages.

Références

[1] OpenCV. How to Use Background Substraction Methods [Comment utiliser les méthodes de soustraction d'arrière-plan]. https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html

[2] IBM. What is instance segmentation ? [Qu'est-ce que la segmentation d'instance ?]. https://www.ibm.com/topics/instance-segmentation.

[3] Bilodeau, G. (2024) INF6804-Vision par ordinateur. chap4_trackingobjects. https://moodle.polymtl.ca/pluginfile.php/213245/mod_resource/content/15/Chap4_Tracking%20objects.pdf

[4] Statistiques Canada. Modélisation du contexte à l'aide de transformateurs : reconnaissance des aliments. https://www.statcan.gc.ca/fr/science-donnees/reseau/reconnaissance-aliments

[5] Chen, E. (6 mai 2020). Instance Segmentation using Mask-RCNN and PyTorch. Eric Chen's Blog. https://haochen23.github.io/2020/05/instance-segmentation-mask-rcnn.html

[6] Girshick, R., Donahue, J., Darell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. UC Berkeley. https://arxiv.org/pdf/1311.2524.pdf?ref=blog.roboflow.com

[7] Potrimba, P. (25 septembre 2023). What is R-CNN?. Roboflow Blog: https://blog.roboflow.com/what-is-r-cnn/