INF6804

Vision par ordinateur

TP 2

Segmentation d’objets vidéo

Félix Auger (2067772)

Arnaud Dalbec-Constant (2014288)

18 mars 2024

Introduction

Le but de ce TP est de caractériser deux méthodes de segmentation de régions d'intérêts dans des vidéos, soit par soustraction d'arrière-plan et par segmentation d'instances. Elles seront comparées afin de déterminer laquelle est la meilleure et dans quelles circonstances.

Présentation des deux méthodes

Soustraction d'arrière-plan

La soustraction d'arrière-plan (*background substraction*) est une méthode de segmentation où on cherche à obtenir deux segments, soient l'arrière-plan (*background*) et le premier plan (*foreground*), ou encore ce qui est en mouvement. Cette méthode consiste en une opération de soustraction avec une image de référence, c'est-à-dire un modèle d'arrière-plan où il y n'y a pas d'objets d'intérêt. La méthode permet de détecter à la fois les objets en mouvement et ceux qui se sont immobilisés. Toutefois, la méthode présente quelques problèmes en lien avec le bruit et l'illumination.

La soustraction d'arrière est la différence $F(x,y)$ entre un modèle de référence $B(x,y)$ et l'image actuelle $Image_i(x,y)$:

$$F(x,y) = |Image_i(x,y) - B(x,y)| > T$$

Où T est le seuil de détection et $B(x,y)$ est le résultat de statistiques, comme la moyenne, la variance ou la médiane de différentes images. Le principe de soustraction d'arrière-plan est montré à la Figure 1. *Difference* correspond à l'image $F(x,y)$ et *Reference* correspond à $B(x,y)$.

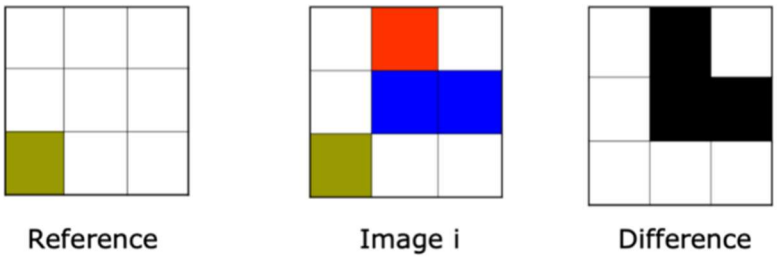

Figure 1: Principe de la soustraction d'arrière-plan [1]

Segmentation d'instances

La segmentation d'instances (*instance segmentation*) est une méthode de segmentation qui se rapproche beaucoup de la détection d'objets avec un réseau de neurones profond. La segmentation d'instances exige une détection précise de tous les objets dans une image, tout en segmentant chaque instance de manière précise. Elle combine ainsi des aspects des tâches classiques de vision par ordinateur telles que la détection d'objets, qui vise à classifier et localiser chaque objet individuel à l'aide de boîtes englobantes, et la segmentation sémantique, qui vise à classifier chaque pixel dans un ensemble prédéfini de catégories sans différencier les instances d'objets. La méthode *Mask R-CNN*, qui est elle-même basée sur *Faster R-CNN*, consiste à ajouter une tête de segmentation à un détecteur d'objets de base. La Figure 2 montre le framework Mask R-CNN pour la segmentation d'instances.

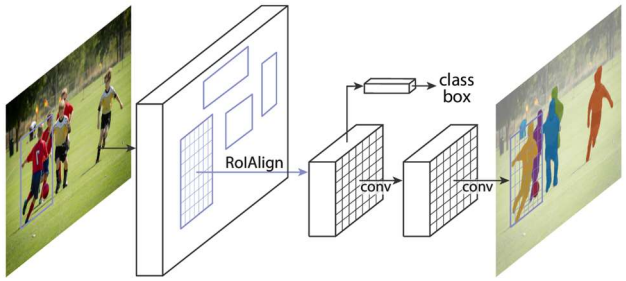

Figure 2: Framework Mask R-CNN pour la segmentation d'instances [2]

Cette architecture utilise un réseau convolutif afin d'analyser chacune des boites identifiées par la première couche. La suite agit comme un modèle de classification classique, mais en considérant des sous images indépendamment.

Hypothèses de performances dans des cas spécifiques

Cas d'utilisation 1 : Objet principal partiellement occulté

Le premier cas d'utilisation est lorsqu'un objet est partiellement occulté. Cela se produit par exemple dans la vidéo «highway» quand deux véhicules sont l'un derrière l'autre, ou encore dans la vidéo «pedestrians» quand deux piétons se croisent. La méthode de soustraction d'arrière-plan devrait mieux performer étant donné que les pixels d'intérêts représentent la section visible des objets en mouvement dans la vidéo. Le risque avec la méthode de segmentation est qu'elle combine deux objets ensemble et les associent à un objet ne faisant pas partie des objets d'intérêt. Par exemple, elle pourrait voir deux piétons et croire qu'il s'agit d'un objet plus large puisqu'ils sont partiellement superposés. Cette détection serait filtrée de la prédiction puisque les piétons sont les seuls sujets d'intérêts.

Cas d'utilisation 2 : Luminosité élevée

Le deuxième cas d'utilisation est lorsque la luminosité de l'image est élevée. Par exemple, les images de la base de données « pedestrians » sont exposées à une très grande quantité de lumière, causant une saturation des pixels dans certaines régions. Le modèle mask RCNN devrait mieux performer, puisqu'en identifiant les objets avec une boite englobante, il retrouve des images relativement similaires à sa base d'entrainement qui contient probablement des images à haute saturation. Ainsi, il possède une capacité à détecter des objets même s'il manque des pixels. Cependant, la méthode de soustraction serait pénalisée, car les pixels saturés des objets d'intérêt ne seraient pas différents des pixels saturés de l'arrière-plan, donc ils passeraient sous le seuil de détection.

Cas d'utilisation 3 : Distinction d'objets similaires dans l'image

Le troisième cas d'utilisation est lorsqu'il y a présence d'objets similaires dans l'image. Dans la base de données « office », le manteau accroché derrière la porte possède un contour similaire à l'humain qui est le sujet d'intérêt de cette série d'images. La méthode de soustraction d'arrière-plan devrait mieux performer dans les cas où il y a présence d'objets similaires, étant donné que le manteau est statique, ainsi il ne sera pas détecté comme un objet d'intérêt. Cependant, par la méthode de segmentation d'instances, le modèle pourrait associer le manteau ou d'autres objets comme étant un objet d'intérêt, puisque qu'il effectue sa prédiction et sa classification sur une seule image, sans prendre compte du fait que l'image est statique ou dynamique.

Description des expériences, séquences de la base de données et critères d'évaluation

Description des expériences

Tout d'abord, chaque méthode est optimisée en analysant la relation entre les hyperparamètres et les performances obtenues (*grid search*, voir section résultats). Une fois que les meilleurs hyperparamètres ont été retenus pour chaque base de données, les trois cas d'utilisation sont analysés en évaluant la performance de chaque méthode sur des séquences spécifiquement choisies de la base de données.

Séquences de la base de données

Cas 1 : Pour évaluer les performances pour le cas d'utilisation 1, on mesure et compare les performances de chaque méthode sur une image présentant des objets partiellement occultés. Par exemple, on utilise l'image «in001565» de la bande de données «highway», où plusieurs voitures se suivent de près. On utilise également l'image «in000428» de la bande de données «pedestrians» où deux piétons se croisent.

Cas 2 : Pour évaluer les performances pour le cas d'utilisation 2, on mesure et compare les performances de chaque méthode sur une image avec une luminosité élevée et des régions saturées. Par exemple, on utilise les images «in000412» et «in000499» de la bande de données «pedestrians», où respectivement un piéton et un cycliste sont difficilement distinguables à cause de saturation dans des régions de l'image.

Cas 3 : Pour évaluer les performances pour le cas d'utilisation 3, on mesure et compare les performances de chaque méthode sur l'entièreté de la vidéo «office», puisqu'il y a présence d'objets similaires, soit le manteau et l'humain.

Critère d'évaluation

Deux métriques ont été utilisés pour quantifier la performance des deux méthodes par rapport à leur vraisemblance au « groundthruth ». Premièrement, l'IOU ou « Intersect Over Union » permet de quantifier la proportion du nombre de pixels identifiés dans le *groundthruth* qui sont également identifiés par la méthode d'identification (intersection) par rapport à tous les pixels identifiés (union).

$$IOU = \frac{Intersect}{Union}$$

Cependant, il est difficile de différencier les faux positifs des vrais négatifs. Ainsi, un modèle identifiant plus de pixels que nécessaire pourrait avoir un score similaire à un modèle qui prédit une petite section des pixels à identifier. Pour pallier à ce défaut et pouvoir différencier ces cas spécifiques, le score F1 est utilisé. Ce métrique prend en compte les vrai positif (TP), les faux positifs (FP) ainsi que faux négatifs (FN).

$$Precision = \frac{TP}{TP + FP}, \qquad Recall = \frac{TP}{TP + FN}, \qquad F1\ score = \frac{2}{\frac{1}{precision} + \frac{1}{Recall}},$$

Ce score permet d'avoir une meilleure idée de la vraisemblance des pixels *groundtruth* par rapport aux pixels identifiés comme un objet d'intérêt, puisque la valeur est pénalisée s'il y a un débordement de la prédiction.

Description des deux implémentations utilisées

Soustraction d'arrière-plan :

Pour cette méthode, on commence par générer les arrière-plans. Il suffit généralement de trouver une image sur laquelle il n'y a aucun objet d'intérêt. Cependant, pour la vidéo «highway», il y a au moins une voiture dans chaque image. L'arrière-plan a donc été construit en combinant le bas d'une image qui montrait seulement une voiture en haut et le haut d'une autre image qui montrait seulement une voiture en bas.

La soustraction d'arrière-plan se fait simplement en soustrayant la matrice de l'arrière-plan à la matrice de l'image à tester. On compare ensuite si la valeur d'intensité de chaque pixel suite à la soustraction est supérieure à une valeur de seuil. Si le résultat de la soustraction est supérieur au seuil, le pixel devient un 1 binaire (blanc), sinon il devient 0 (noir). On peut ensuite compter le nombre de 1 dans l'image testée et l'arrière-plan afin de calculer les différents scores pertinents à l'analyse. Le code a été repris de l'exemple fournit par le professeur Bilodeau [3] et légèrement adapté pour les besoins de ce laboratoire.

Le seuil de sensibilité de détection est le seul hyperparamètre pour cette méthode. On le fait varier de 1 à 160, 1 étant la plus faible différence mesurable, 160 étant une borne supérieure choisie arbitrairement. La borne supérieure n'a peu d'importance, puisque les performances à ce niveau sont toujours faibles.

Pour identifier le meilleur hyperparamètre, un *grid search* est réalisé. Pour chaque valeur d'hyperparamètre, soit de 1 à 160, on calcule les scores obtenus sur toutes les images de la banque de données et on enregistre la moyenne. On retient la valeur d'hyperparamètre qui permet d'obtenir les meilleurs scores en moyenne, et c'est cette valeur qui est utilisée pour réaliser les expériences.

Segmentation d'instance :

Pour cette méthode, un modèle maskrcnn_resnet50_fpn pré-entraîné est chargé dans le script afin d'effectuer la tâche de segmentation d'instance. Le code utilisé pour importer le modèle a été tiré du notebook Mask_RCNN.ipynb fournit dans le cours. [4] Le reste de la structure du code a été ajouté afin d'effectuer une recherche du paramètre « Threshold » optimal pour chaque base de données indépendamment.

Ce modèle fourni une détection d'objets en plus de préciser les coordonnées de la boite englobante ainsi qu'une étiquette sur l'objet détecté. Puisque plusieurs catégories d'objets sont détectées, il faut filtrer les résultats pour retenir seulement la catégorie d'intérêt. Par exemple, pour la base de données Pedestrians, les piétons sont la seule catégorie d'objet à identifier, cela correspond à la catégorie 1 identifiée par le modèle, alors que les voitures pour « Highway » correspondent à l'étiquette 3.

Cette méthode utilise l'hyperparamètre « Threshold » pour quantifier le degré de confiance que doit avoir le modèle pour identifier un objet. Une série de tests est réalisée avec cette valeur de seuil allant de 0.20 à 0.95. À toutes les itérations, des images sont chargées et analysées par ce modèle afin d'en ressortir l'identification des pixels correspondant à tous les objets détectés. Elles sont ensuite comparées à l'image correspondant de type « groundtruth » afin de calculer l'IOU et le métrique F1. Tous les résultats sont enregistrés dans un tableau pour ensuite calculer la moyenne de ces métriques sur l'ensemble des images de la vidéo en fonction de la valeur de l'hyperparamètre.

Présentation des résultats de tests

Les figures suivantes représentent les résultats moyens obtenus sur chaque base donnée en fonction des hyperparamètres.

1.1 Soustraction d'arrière-plan sur Highway

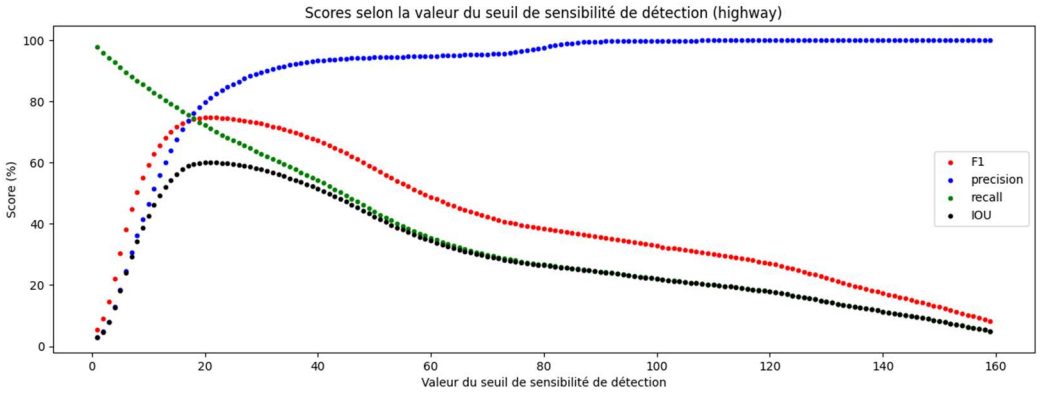

Figure 3: Scores obtenus par soustraction d'arrière-plan sur les données Highway en fonction du seuil de sensibilité de détection

1.2 Soustraction d'arrière-plan sur Office

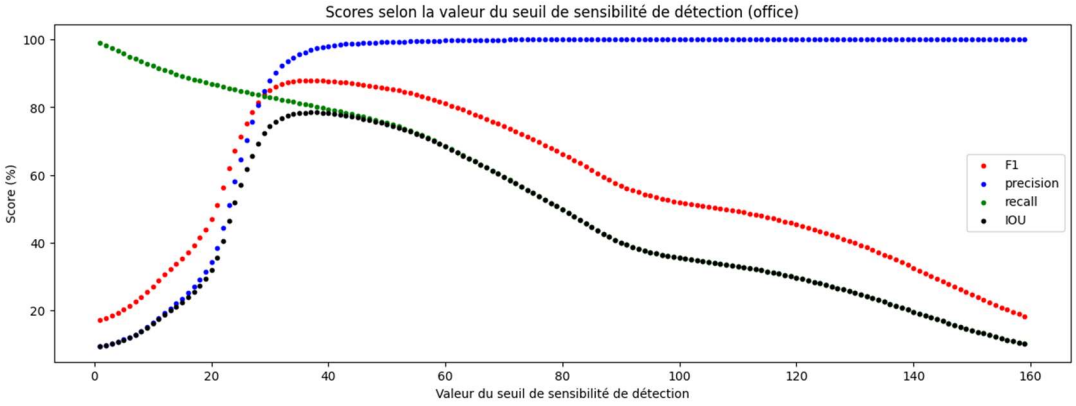

Figure 4: Scores obtenus par soustraction d'arrière-plan sur les données Office en fonction du seuil de sensibilité de détection

1.3 Soustraction d'arrière-plan sur Pedestrians

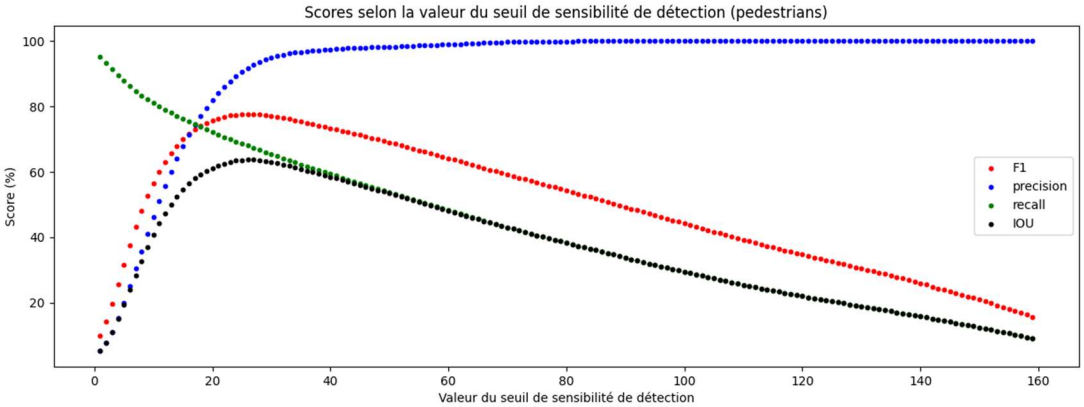

Figure 5: Scores obtenus par soustraction d'arrière-plan sur les données Pedestrians en fonction du seuil de sensibilité de détection

1.4 Soustraction d'arrière-plan sur PETS2006

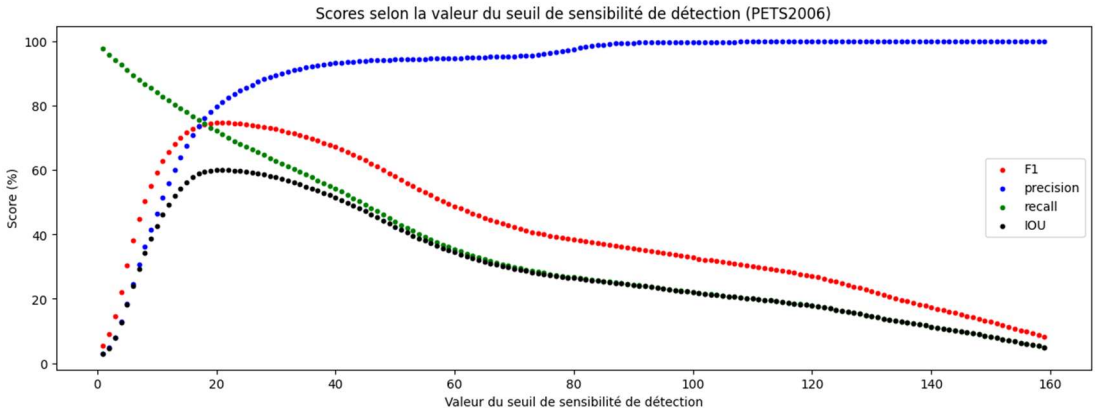

Figure 6: Scores obtenus par soustraction d'arrière-plan sur les données PETS2006 en fonction du seuil de sensibilité de détection

2.1 Segmentation d'instance sur Highway

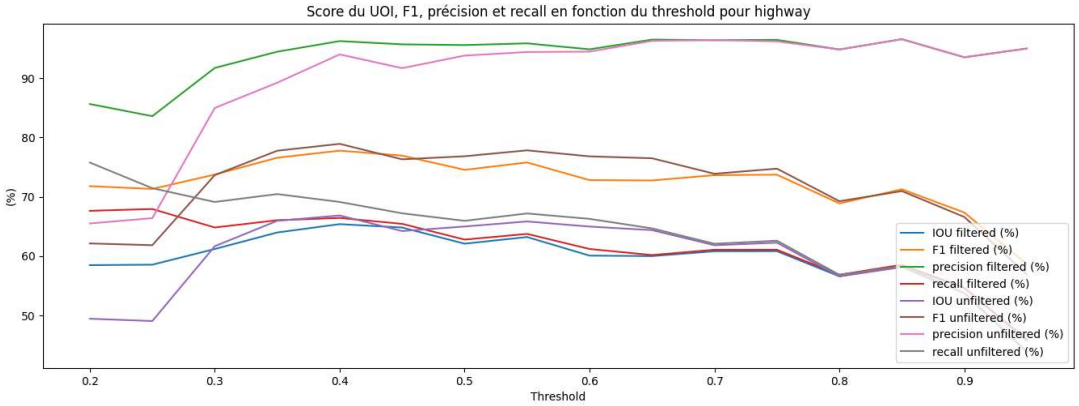

Figure 7. Scores obtenus par segmentation d'instances avec et sans filtrage sur les données Highway en fonction du threshold

2.2 Segmentation d'instance sur Office

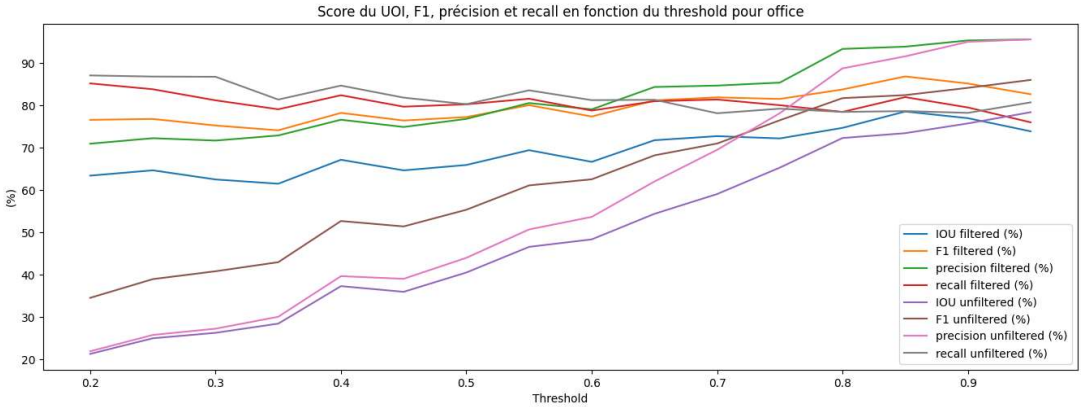

Figure 8. Scores obtenus par segmentation d'instances avec et sans filtrage pour les objets d'intérêt sur les données Office en fonction du threshold

2.3 Segmentation d'instance sur Pedestrians

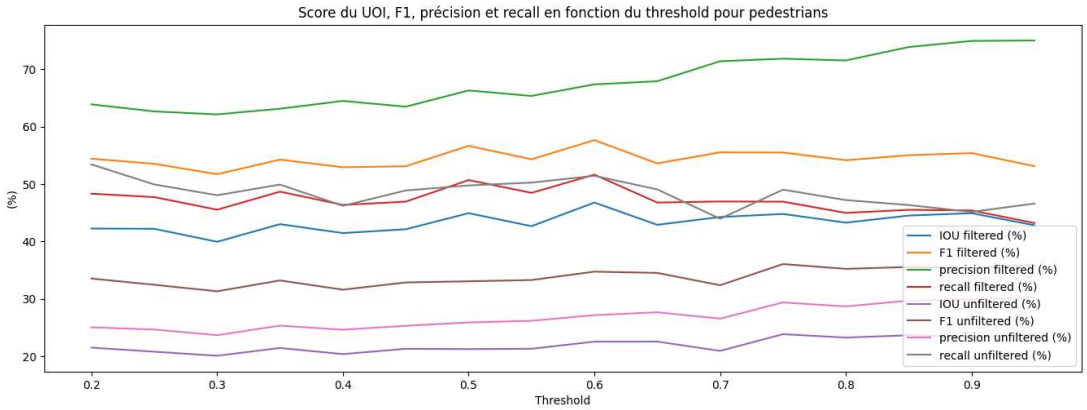

Figure 9. Scores obtenus par segmentation d'instances avec et sans filtrage pour les objets d'intérêt sur les données Pedestrians en fonction du threshold.

2.4 Segmentation d'instance sur PETS2006

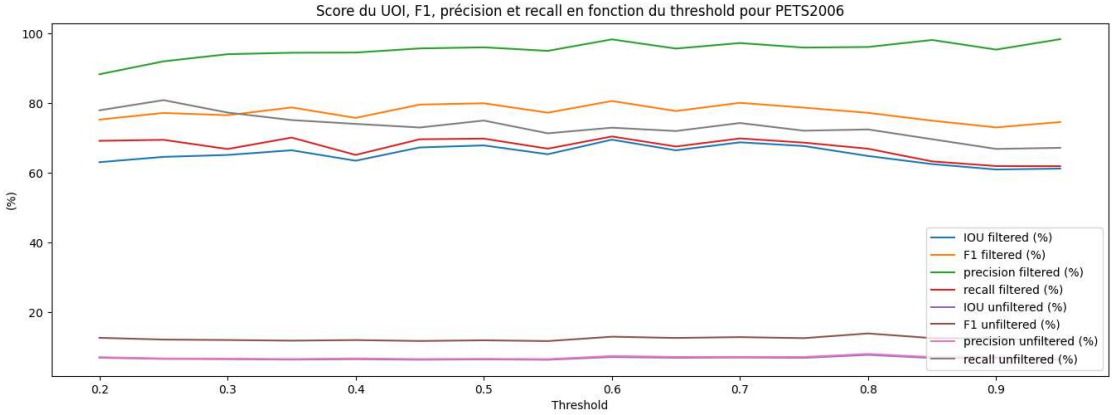

Figure 10. Scores obtenus par segmentation d'instances avec et sans filtrage pour les objets d'intérêt sur les données PETS2006 en fonction du threshold.

3.1 Résultat pour le cas d'utilisation 1

<u>Soustraction d'arrière-plan :</u>

Seuil de détection n=16 pour pedestrians et n=42 pour highway.

Figure 11: Effet des objets partiellement occultés sur la méthode de soustraction d'arrière-plan sur l'image pedestrians_000428.

Figure 12: Effet des objets partiellement occultés sur la méthode de soustraction d'arrière-plan sur l'image highway_001565

Segmentation d'instances (threshold de 0.60 sur pedestrians et 0.4 sur highway) :

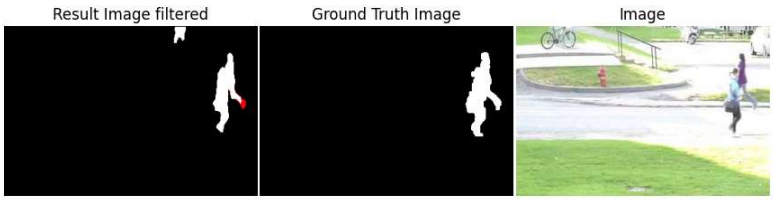

Figure 13. Effet des objets partiellement occultés sur la méthode de segmentation d'instance sur l'image pedestrians_000428.

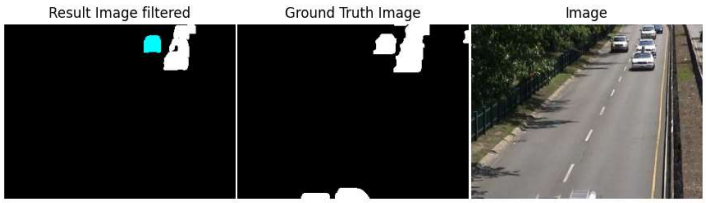

Figure 14. Effet des objets partiellement occultés sur la méthode de segmentation d'instance sur l'image highway_001565.

Tableau 1: Scores obtenus pour l'expérience du cas d'utilisation 1

Méthode	Soustraction d'arrière-plan		Segmentation d'instance	
Image	pedestrians_000428	highway_001565	pedestrians_000428	highway_001565
F1 score (%)	81.0	66.0	67.8	70.8
UOI score (%)	68.1	49.2	52.7	54.9

3.2 Résultat pour le cas d'utilisation 2

Soustraction d'arrière-plan :

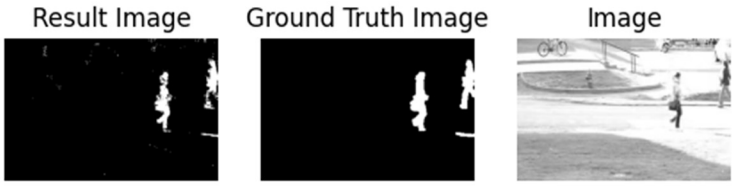

Figure 15: Impact de la luminosité sur la méthode de soustraction d'arrière-plan sur l'image pedestrians_000412

Figure 16: Impact de la luminosité sur la méthode de soustraction d'arrière-plan sur l'image pedestrians_000499

Segmentation d'instances (threshold de 0.60, meilleure performance sur cette vidéo):

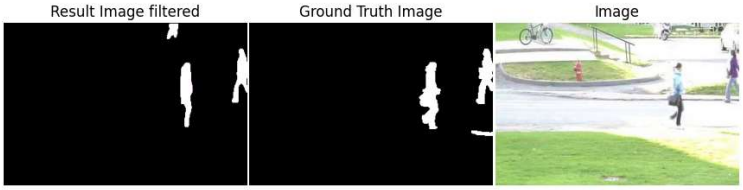

Figure 17. Impact de la luminosité sur la méthode de segmentation d'instance sur l'image pedestrians_000412

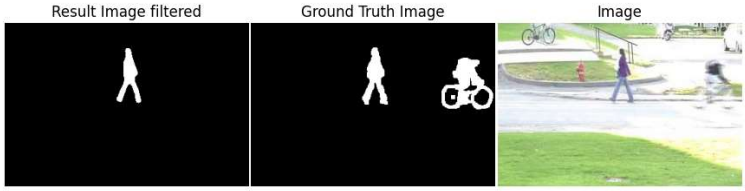

Figure 18. Impact de la luminosité sur la méthode de segmentation d'instance sur l'image pedestrians_000499

Tableau 2: Scores obtenus pour l'expérience du cas d'utilisation 2

Méthode	Soustraction d'arrière-plan		Segmentation d'instance	
Image	pedestrians_000412	pedestrians_000499	pedestrians_000412	pedestrians_000499
F1 score (%)	79.4	69.3	65.1	35.6
UOI score (%)	65.8	53.0	49.1	22.0

3.3 Résultat pour le cas d'utilisation 3

Cette figure est un exemple de la problématique rencontrée avec la méthode de segmentation d'instance avec une valeur de threshold plus faible (0.40).

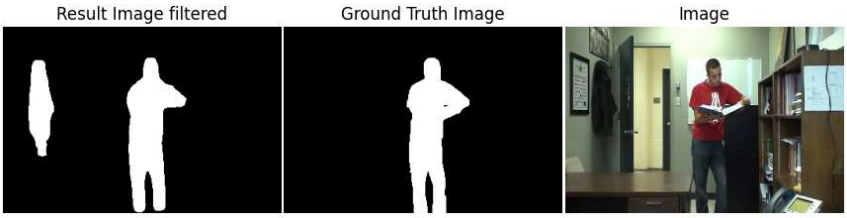

Figure 19. Fausse détection d'objet avec la méthode de segmentation d'instance sur l'image office_001565 pour un threshold de 0.40.

Discussion des résultats et retour sur les hypothèses

Analyse du « ground truth »

En analysant les images ground truth, il est possible de remarquer que certains objets qui devraient être identifiés ne le sont pas. Par exemple, dans l'image 000489 de la base de données Pedestrians, il est possible de remarquer que la bicyclette en mouvement est identifiée alors que la bicyclette immobile dans le haut de l'image n'est pas identifiée. Cette observation met en évidence un certain biais du ground truth qui favoriserait la méthode de soustraction puisqu'elle n'identifie pas les objets immobiles.

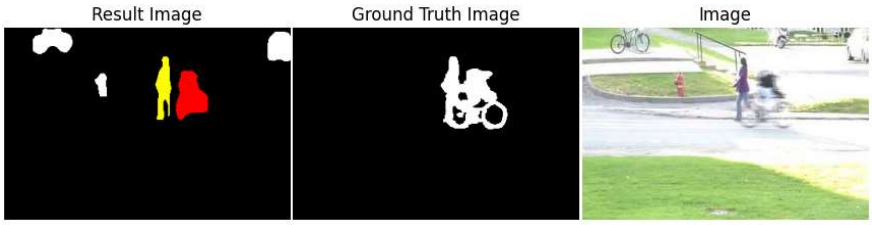

Figure 20. Analyse du groundtruth et comparaison à la segmentation d'instance sur l'image « in000489 ».

Cas d'utilisation 1:

Les Figure 11 et Figure 12 montrent que la méthode de soustraction d'arrière-plan permet de bien détecter des objets partiellement occultés, avec des scores F1 de 81.0 et 66.0%. Les scores sont négativement affectés par la présence de faux positifs ainsi que de faux négatifs éparpillés sur l'image. Pour la méthode de segmentation d'instances, on peut voir sur la Figure 13 que les deux piétons qui se croisent sont identifiés avec un score F1 de 67.8%. Sur la Figure 14, les voitures qui se suivent de près sont également identifiés correctement avec un score F1 de 70.8%. Les scores sont négativement affectés par de faux positifs et négatifs sous la forme d'objets entiers qui ne sont pas correctement identifiés, comme la partie de voiture tout au bas de l'image à la Figure 14. Cependant, les deux méthodes semblent autant bien performer pour les objets partiellement occultés. L'hypothèse initiale était que la méthode de soustraction d'arrière-plan serait plus performante, car la méthode de segmentation d'instances risquerait de combiner deux objets ensemble et les associer à un objet ne faisant pas partie des objets d'intérêt. Cette hypothèse est en partie vérifiée, car on peut remarquer que les deux piétons qui se croisent ou même les voitures qui se suivent de près ne sont pas identifiés comme des objets d'intérêt. Par contre, il faudrait tester sur un plus grand échantillon d'images avec des objets occultés afin de déterminer si les performances d'une méthode se distinguent de l'autre, car les scores obtenus au Tableau 1 ne penchent pas en faveur d'une des méthodes.sont trop similaires.

Cas d'utilisation 2:

On peut voir sur les Figure 15 et Figure 16 que la luminosité élevée sur les images de pedestrians nuit à la détection des objets d'intérêts pour la méthode de soustraction. En effet, la détection des piétons et du cycliste manque beaucoup de netteté (présence de faux négatifs) au niveau de la zone de l'image en saturation. Ainsi, les objets d'intérêts tendent suffisamment vers le blanc pour passer sous le seuil de détection. Les scores F1 obtenus sont 79.4 et 69.3 %. Pour la méthode de segmentation d'instances, la Figure 17 montre que le piéton qui passe dans la zone saturée de l'image est correctement identifié. Cependant, le cycliste qui passe dans cette même zone n'est pas du tout identifié sur la Figure 18. Les scores F1 obtenus sont de 65.1 et 35.6%. L'hypothèse initiale suggérant que la méthode de segmentation d'instances performerait mieux est invérifiée. En effet, même si la méthode de soustraction d'arrière-plan a de la difficulté à bien détecter l'ensemble des objets d'intérêts à cause de la saturation de l'image, la méthode de segmentation d'instances n'a pas réussi à détecter un objet en entier, ce qui nuit beaucoup à sa fiabilité. Il est important de noter que l'échantillon utilisé est trop faible (uniquement 2 images), et que le cycliste pourrait être correctement identifié par la méthode de segmentation en peaufinant l'ajustement des paramètres. Ainsi, sachant que la méthode de soustraction a une lacune au niveau des zones saturés, une amélioration du test pourrait permettre de montrer que la méthode de segmentation s'avère plus efficace.

Cas d'utilisation 3 :

Les résultats moyens générés pour chaque méthode sur la vidéo « office » sont utilisés pour évaluer leur performance quant à la fausse détection d'objets. Pour la méthode de segmentation d'instances, il est possible d'observer à la Figure 8 que les performances augmentent avec la fiabilité de la détection d'objets du modèle. Tel que montré à la Figure 19, le manteau est identifié comme un humain, mais avec une faible certitude (threshold de 0.40). Ainsi, lorsque le seuil d'acception devient plus sévère, les performances se rapprochent de celles de la méthode de soustraction d'arrière-plan. En effet, les deux méthodes obtiennent des résultats comparables avec des paramètres optimisés, soit 75% pour la soustraction d'arrière-plan et 80% pour la segmentation d'instances. Il n'est donc pas possible de déterminer si une méthode est significativement plus performante qu'une autre. Pour améliorer ce test et obtenir une analyse plus concluante, il faudrait utiliser une autre banque de données qui contient davantage d'objets similaires qui risqueraient d'être incorrectement identifiés par la méthode de segmentation.

Conclusion

Pour conclure, les 3 cas d'utilisation spécifique ont permis d'identifier des défauts pour chaque méthode, mais ces expériences ne sont pas suffisantes pour déterminer laquelle est ultimement meilleure. Même si la méthode de soustraction d'arrière-plan surperforme le modèle de segmentation, il est important de réaliser qu'elle est beaucoup moins flexible dans le contexte d'utilisation. En effet, celle-ci nécessite une image de référence, alors que le Mask_RCNN permet d'identifier de nombreux types d'objets dans des contextes qu'il n'a jamais vus, en plus d'ajouter une étiquette sur le type de l'objet. Puisque les performances atteintes sont souvent similaires, cela laisse croire qu'un modèle convolutif spécifiquement entrainé pour reconnaitre le « groundtruth » serait davantage performant que les deux méthodes analysées.

Bibliographie

[1] G.-A. Bilodeau, « Extracting regions of interest, INF6804 Vision par ordinateur, Chap. 3 ».

[2] K. He, G. Gkioxari, P. Dollar, et R. Girshick, « Mask R-CNN », *2017 IEEE International Conference on Computer Vision (ICCV)*, p. 2980-2988, 2017.

[3] « INF6804/TemporalAvgBGS.ipynb at master · gabilodeau/INF6804 », GitHub. Consulté le: 17 mars 2024. [En ligne]. Disponible à: https://github.com/gabilodeau/INF6804/blob/master/TemporalAvgBGS.ipynb

[4] Lamghari, Sofiane, « Mask_RCNN », INF6804 Vision par ordinateur, Consulté le : 17 mars 2024. [En ligne]. Disponible à: https://colab.research.google.com/github/gabilodeau/INF6804/blob/master/Mask_RCNN.ipynb