



**INF6804 - Vision par ordinateur**

**Hiver 2024**

## **TP2 – SEGMENTATION D’OBJETS VIDEO**

**Nicolas Dépelteau - 2083544**

**Sebastien Goll - 2231054**

**Soumis à : Guillaume-Alexandre Bilodeau**

**18 mars 2024**

# I Table des matières

<b>TP2 – SEGMENTATION D’OBJETS VIDEO</b> .....	1
1 Table des matières .....	2
2 Introduction .....	4
3 Présentation des deux méthodes (4 pts) .....	4
3.1 Soustraction d’arrière-plan .....	4
3.2 Segmentation d’instances .....	5
4 Hypothèses de performance pour des cas spécifiques (3 pts) .....	5
4.1 Présence d’occlusion .....	5
4.2 Adaptation au changement de luminosité et d’ombre .....	5
4.3 Présence de rotation .....	5
4.4 Présence de mouvement dans l’arrière-plan .....	5
5 Description des expériences, séquences de la base de données et critères d’évaluation (2 pts) .....	6
5.1 Présence d’occlusion .....	6
5.2 Adaptation au changement de luminosité et d’ombre .....	6
5.3 Présence de rotation .....	7
5.4 Présence de mouvement dans l’arrière-plan .....	7
6 Description des deux implémentations utilisées (2 pts) .....	8
6.1 Implémentation des expériences .....	8
6.2 Recherche de paramètre optimaux .....	8
6.2.1 Recherche de paramètre pour la méthode de soustraction d’arrière-plan .....	8
6.2.2 Recherche de paramètre pour la méthode de segmentation d’instance .....	8
6.3 Méthode de soustraction d’arrière-plan .....	8
6.4 Méthode de segmentation d’instance .....	9
7 Présentation des résultats de tests (3 pts) .....	9
7.1 Présence d’occlusion .....	9
7.2 Adaptation au changement de luminosité et d’ombre .....	10
7.3 Présence de rotation .....	11
7.4 Présence de mouvement dans l’arrière-plan .....	12
7.5 Métriques sommaires .....	12
8 Discussion des résultats et retour sur les hypothèses (3 pts) .....	13
8.1 Occlusion .....	13
8.2 Ombres .....	13
8.3 Rotation .....	14
8.4 Mouvement parasite .....	14
9 Conclusion .....	15
9.1 Avantages .....	15
9.1.1 Segmentation d’instance .....	15
9.1.2 Soustraction d’arrière-plan .....	15
9.2 Inconvénients .....	15

9.2.1	Segmentation d'instance.....	15
9.2.2	Soustraction d'arrière-plan.....	15
10	Références .....	16

## 2 Introduction

La détection de mouvement est une question importante en vision par ordinateur, de nombreuses applications requièrent la possibilité de trouver de détecté le mouvement, souvent à des fins de surveillance. L'état de l'art actuel en termes de vision par ordinateur et de manipulation d'images est principalement concentré autour des réseaux de neurones et de l'apprentissage profond. Les méthodes étudiées dans ce TP sont la méthode de soustraction d'arrière-plan ainsi que la méthode de segmentation d'objet à l'aide d'un réseau Mask R-CNN.

## 3 Présentation des deux méthodes (4 pts)

Nous allons étudier deux méthodes différentes afin de les comparer dans la tâche de segmentation d'objet en mouvement. Les deux méthodes utilisées sont les suivantes :

### 3.1 Soustraction d'arrière-plan

Cette méthode consiste à comparer chaque images/frame de la vidéo à un image de référence, représentant uniquement l'arrière-plan. Les pixels sont comparés un à un afin de calculer sa distance par rapport au pixel correspondant sur l'image de référence, si cette distance est trop importante, alors on juge que le pixel a changé, et donc que le pixel ne fait pas partie de l'arrière-plan. L'image de référence est une moyenne d'images de l'arrière-plan présent au préalable.

$$F(x, y) = |Image_i(x, y) - \mathbb{E}[Images_{référence}(x, y)]|$$

Si la fonction  $F$  est trop élevée, alors le pixel appartient à l'avant-plan sinon le pixel appartient à l'arrière-plan. Cette approche permet donc d'étiqueter chaque pixel de l'image actuelle comme appartenant à l'image d'arrière-plan ou au premier plan. Cette approche possède plusieurs avantages, tout d'abord, le calcul des distances et l'étiquetage des pixels est très rapide, permettant donc de faire tourner ce processus sur des appareils peu puissant en temps réel.

Toutefois, dû à la simplicité de la méthode, de nombreux inconvénient sont aussi existants, cette méthode est très sensible à l'image de l'arrière-plan initialement utilisée, si l'arrière-plan change au fil du temps (changement de luminosité, ombres qui changent, vent qui fait bouger des branches...) alors la soustraction d'arrière-plan devient beaucoup moins précise, et des étapes doivent être ajoutées afin de prendre en compte ces changements.

## 3.2 Segmentation d'instances

L'approche par segmentation d'instance utilise des réseaux de neurones afin d'extraire des *masks* de probabilité ainsi que leur classe associée. Les instances pouvant être détectées dépendent du réseau de neurone utilisé. On peut ensuite superposer le *mask* pour obtenir les pixels qui appartiennent à l'instance. Pour la détection de mouvement, cette approche donne les objets dans l'image. Ce n'est pas équivalent à la méthode de soustraction d'arrière-plan, néanmoins nous pouvons extraire les objets communs dans l'image et les utiliser en tant que détection.

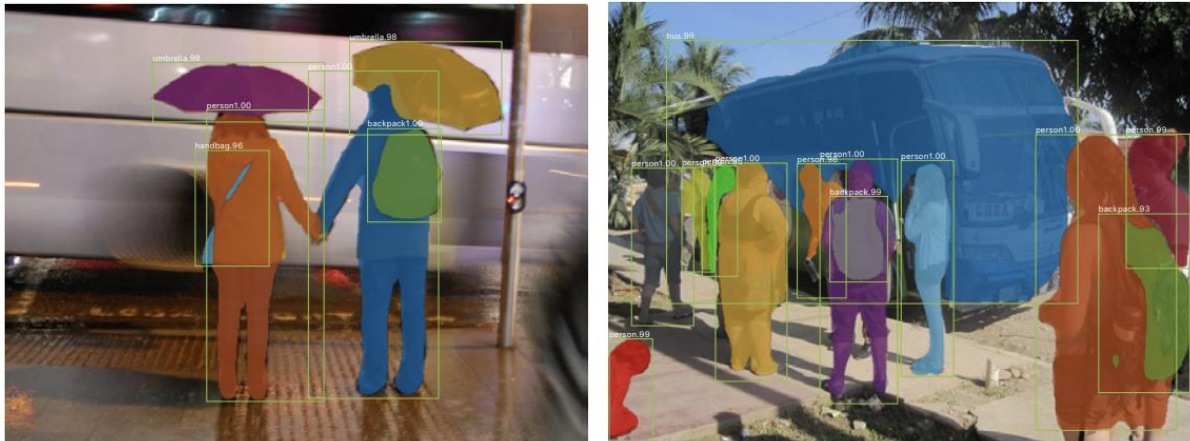


Figure 3.1 Exemple de segmentation d'instance (Girshick, 2018)

## 4 Hypothèses de performance pour des cas spécifiques (3 pts)

### 4.1 Présence d'occlusion

Pour les cas où l'objet est partiellement caché, nous croyons que la méthode de soustraction d'arrière-plan performera beaucoup mieux que celle par instance de segmentation. En effet, la méthode de soustraction d'arrière-plan ne prend pas compte du concept d'objet, mais seulement si les pixels sont assez différents de la référence, donc l'occlusion ne l'affecte pas. Toutefois, pour la segmentation d'objet, il est possible que le réseau ne détecte plus l'instance de l'objet.

### 4.2 Adaptation au changement de luminosité et d'ombre

Pour les cas où la luminosité au cours de la vidéo varie, par exemple une région devient ombragée. Nous croyons que la méthode de segmentation d'instance performera mieux puisque la méthode de soustraction d'arrière-plan prendra en compte l'ombre comme étant l'objet. En effet, s'il y a un trop grand changement de luminosité où s'il y a présence d'ombre, l'image de référence reste inchangée, donc la différence sera élevée et entraînera une mauvaise classification.

### 4.3 Présence de rotation

Pour les cas où l'objet aurait subi une rotation, nous croyons que la méthode de soustraction d'arrière-plan performera mieux que celle de segmentation d'instance puisqu'elle n'a pas le concept d'objet, ni d'orientation. En effet, la méthode de soustraction d'arrière-plan ne prend pas compte du concept d'objet, mais seulement si les pixels sont assez différents de la référence, donc une rotation ne l'affecte pas.

### 4.4 Présence de mouvement dans l'arrière-plan

Pour les cas où il y a du mouvement dans l'arrière-plan en soit, nous croyons que la méthode de segmentation d'instance performera mieux puisqu'elle détecte seulement les instances d'objet dans l'image. D'autre part, la méthode de soustraction d'arrière-plan aura beaucoup de difficulté à avoir une bonne image de

référence puisque la variance sera élevée sur la valeur des pixels qui participent au mouvement. Cette haute variance amène qu'une valeur de pixel totalement normal sera avec un grand écart par rapport à la moyenne, ce qui mènera à une fausse détection.

## 5 Description des expériences, séquences de la base de données et critères d'évaluation (2 pts)

Pour nos expériences, nous utilisons l'entièreté des vidéos. Chaque vidéo est traitée comme un jeu de données composées des images ainsi que leur prédiction parfaite associée. Nous utilisons les images de calibration pour entraîner nos modèles qui correspondent à chacune des deux méthodes étudiées. Pour chaque vidéo, nous effectuons la phase de calibration avec les images de calibrations, puis nous effectuons la prédiction sur le reste des images. Nous calculons également les métriques sur chaque prédiction. Nous avons également effectué une recherche d'hyperparamètre pour maximiser au préalable la performance de chaque méthode pour comparer meilleur à meilleur. Le critère d'évaluation du modèle est la métrique de score F1.

### 5.1 Présence d'occlusion

Pour l'occlusion, nous allons nous intéresser à la vidéo de l'autoroute, sur celle-ci, les objets d'intérêt (les véhicules roulant sur la route) apparaissent en haut de l'image et disparaissent en bas de celle-ci. Particulièrement lors de leur disparition, pendant quelques images, le véhicule va être « couper » en bas de l'image.



Figure 5.1: Véhicule occulter en sortant de l'image

Nous allons donc utiliser ces quelques images afin de comparer les deux méthodes dans ce cas spécifique. Nous nous attendons à observer que le modèle par segmentation d'instance ne détecte plus ce véhicule lors des dernières images. Au contraire, la méthode par soustraction d'arrière-plan ne devrait pas être perturbée par cette situation puisque cette méthode fonctionne uniquement au niveau du pixel.

### 5.2 Adaptation au changement de luminosité et d'ombre

La vidéo « PETS2006 » se déroule dans une gare avec une lumière artificielle très dirigée. Chaque personne passant devant la caméra projette une zone d'ombre très visible sur le sol, ceci nous intéresse tout particulièrement afin de voir comment les deux modèles réagissent à ce type de perturbations. Nous nous attendons à ce que la méthode par segmentation d'instance performe mieux que celle par soustraction d'arrière-plan. Ceci est dû au fait que la soustraction d'arrière-plan ne prend pas en compte le contexte du pixel qui est évalué, et seulement sa valeur par rapport à une valeur de référence. Lorsqu'une ombre passe sur le sol, celui-ci change de couleur/luminosité, et la soustraction d'arrière-plan devrait considérer ce pixel de la même façon que si quelque chose était passé devant la caméra à cet endroit.

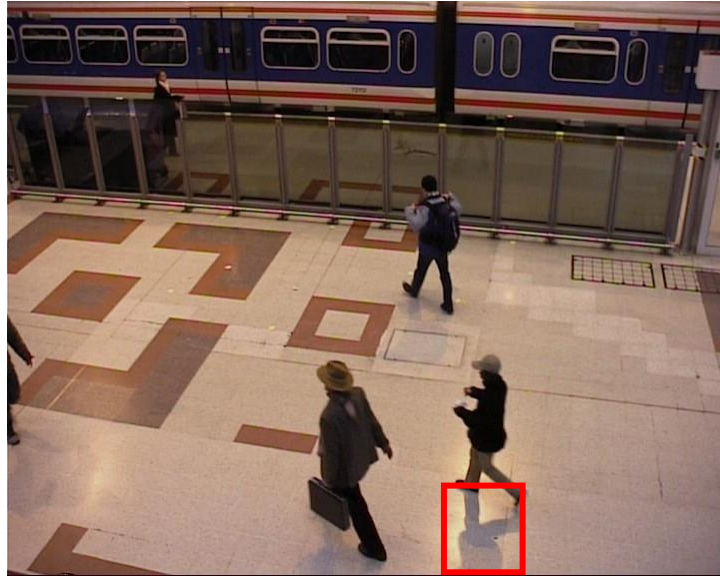


Figure 5.2: Ombre d'une personne passant devant la camera

### 5.3 Présence de rotation

Dans les images provenant, de la vidéo « office », on voit une personne manipulé un livre (l'ouvrir et le tourner). On peut alors regarder comment le model par segmentation d'instance détecte ce livre dans différente configuration. On pourrait s'attendre à des erreurs de ce modèle, mais cela dépend énormément de comment le réseau de neurones a été entraîné et de ses performances. La soustraction d'arrière-plan, elle, devrait être plus résiliente à ces situations.



Figure 5.3: Deux différente orientations/positions du livre

### 5.4 Présence de mouvement dans l'arrière-plan

Pour le mouvement dans l'arrière-plan nous nous intéressons à la vidéo de l'autoroute. Plus précisément à la région des arbres. Par exemple, comme illustré dans la Figure 5.4 Image contenant les arbres à côté de l'autoroute, le feuillage est en mouvement d'une image à l'autre à cause du vent. Cela affecte beaucoup la valeur des pixels représentant cette partie. Nous nous attendons à des artefacts pour la méthode de soustraction d'arrière-plan tandis que cette région sera ignorée par la segmentation d'instance.





Figure 5.4 Image contenant les arbres à côté de l'autoroute

## 6 Description des deux implémentations utilisées (2 pts)

### 6.1 Implémentation des expériences

Tout d'abord, nous avons utilisé la classe *Dataset* de *PyTorch* pour charger les images d'une vidéo ainsi que leur image de prédiction parfaite. Les images ont été téléverser dans un artéfact sur *weight and bias* (wandb). Notre expérience, qui est en soit une *run wandb*, crée notre méthode de segmentation qui respecte l'interface *SegmentationMethod*. Ensuite, pour chaque vidéo, on utilise les images de calibration du *Dataset* correspondant pour l'entraînement de la méthode de segmentation. Puis, on utilise le restant des images comme ensemble de test. La méthode de segmentation produit un *mask* pour chaque image de l'ensemble de test. On calcul les métriques, entre autres le score F1, pour chacune des prédictions. Finalement, on *log* les prédictions, les prédictions désirées et les métriques sur *weights and bias*.

### 6.2 Recherche de paramètre optimaux

Nous avons effectué une recherche de paramètre optimaux pour chacune des méthodes en utilisant *hydra* (Yadan, 2019) pour la gestion des configurations et des rapports (wandb) pour l'analyse des résultats.

#### 6.2.1 Recherche de paramètre pour la méthode de soustraction d'arrière-plan

- Le seuil de sensibilité (40, 50, 60, 70, 80, 90, 100)

#### 6.2.2 Recherche de paramètre pour la méthode de segmentation d'instance

- Le seuil de confiance sur la prédiction (0.2, 0.4, 0.5, 0.6, 0.8)
- Le seuil de confiance sur la prédiction du *mask* des pixels (0.2, 0.4, 0.5, 0.6, 0.8)

### 6.3 Méthode de soustraction d'arrière-plan

Nous avons implémenté au complet cette méthode à l'aide des méthodes de calcul sur les *Tensor* (PyTorch, s.d.). Nous prenons la moyenne des images de calibrations comme image de référence  $Image_{référence} = \mathbb{E}[Images_{calibration}]$ . Ensuite, nous effectuons la soustraction entre l'image de référence et l'image  $Diff = Image_i - Image_{référence}$ . À noter que nous utilisons les trois canaux rouge, vert et bleu dans nos expériences. En prenant cela en compte, nous avons adopter le calcul du *mask* comme suit :

$$Mask = \left[ k^2 < \sum_{canaux} (Image_i - \mathbb{E}[Images_{calibration}])^2 \right]$$

Nous utilisons une formulation qui correspond à si le vecteur rouge, vert et bleu se situe à l'intérieur d'une sphère autour du vecteur rouge, vert et bleu de référence. La valeur de k est un hyperparamètre



correspondant au rayon de sensibilité de la méthode. La valeur optimale que nous avons utilisée est de 55. La valeur a été déterminé à l'aide de [ce rapport wandb](#).

## 6.4 Méthode de segmentation d'instance

Nous avons utilisé un modèle entraîné suivant la formulation de Mask R-CNN (Girshick, 2018). Ce modèle vient directement de PyTorch (PyTorch, s.d.). Pour ce modèle, Il n'y a pas de phase de calibration. Nous devons également normaliser les images d'entrée pour qu'ils soient compatible avec le modèle. Le modèle nous donne prédiction de chaque instance dans l'images. Une prédiction est constituée d'une probabilité, d'une étiquette et d'une carte de chaleur correspondant à la probabilité que la  $i^{\text{ème}}$  pixel soit de la même classe que l'étiquette. On filtre alors toutes les prédictions pour obtenir seulement ceux avec les étiquettes des objets d'intérêt. De ceux qui reste, nous gardons ceux qui ont une probabilité supérieure à  $k$ . On ajoute alors les pixels de la carte de chaleur dont la probabilité est supérieure à  $j$ . Autrement dit, le *mask* en sortie correspond à la somme des pixels valide appartenant aux cartes de chaleur pour les prédictions qui ont une probabilité assez élevée et que l'étiquette soit dans ceux d'intérêt. Les valeurs optimales que nous avons utilisées pour  $k$  et  $j$  sont respectivement 0.8 et 0.5. Les valeurs optimales ont été déterminées à l'aide de [ce rapport wandb](#).

## 7 Présentation des résultats de tests (3 pts)

### 7.1 Présence d'occlusion

**Soustraction d'arrière-plan**



FI 0,79

**Segmentation d'instance**



FI 0.63

**Figure 7.1 Résultat pour le cas d'occlusion**

Nous pouvons observer que pour le cas d'occlusion, la méthode de segmentation d'instance n'arrive pas à identifier le véhicule en bas à gauche tandis que la méthode de soustraction d'arrière-plan le détecte sans soucis. La méthode de soustraction d'arrière-plan contient toutefois du bruit dans le *mask*. Pour ce cas d'utilisation, la meilleure méthode selon le score FI est celle de la soustraction d'arrière-plan.

## 7.2 Adaptation au changement de luminosité et d'ombre

**Soustraction d'arrière-plan**



FI 0,85

**Segmentation d'instance**



FI 0.93

**Figure 7.2 Résultat pour le cas d'ombre forte**

Nous pouvons observer que la méthode de segmentation d'instance ignore l'ombre au complet tandis que celle de soustraction d'arrière-plan inclut l'ombre dans son *mask*. La meilleure méthode selon le score FI est la méthode de segmentation d'instance.

### 7.3 Présence de rotation

**Soustraction d'arrière-plan**



FI 0,85

**Segmentation d'instance**



FI 0.90



FI 0.90



FI 0.95

**Figure 7.3 Résultats pour le cas de rotation**

Nous pouvons observer que pour le cas de la rotation, la méthode de segmentation d'instance n'arrive pas à identifier le livre avec une orientation horizontale. Toutefois, si nous considérons le score FI, la meilleure méthode est la méthode de segmentation d'instance. On peut également observer que le *mask* de la soustraction d'arrière-plan éprouve de la difficulté pour le blanc des pages du livre sur le blanc du tableau en arrière-plan.

## 7.4 Présence de mouvement dans l'arrière-plan

**Soustraction d'arrière-plan**



FI 0,81

**Segmentation d'instance**



FI 0.87

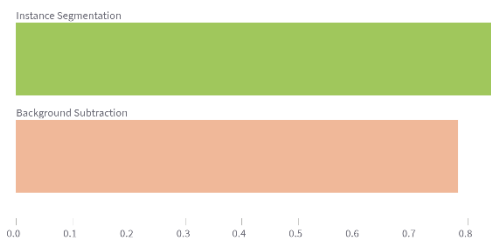
**Figure 7.4 Résultats pour le cas où il y a du mouvement dans l'arrière-plan**

Nous pouvons observer que le mouvement dans l'arrière-plan ajoute des portions indésirables au *mask* de la soustraction d'arrière-plan. Néanmoins, la méthode de segmentation d'instance ignore complètement ce mouvement. La meilleure méthode selon le score FI est la méthode de segmentation.

## 7.5 Métriques sommaires

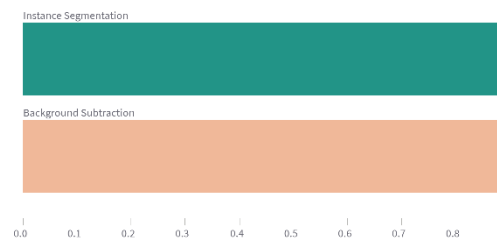
**Vidéo de l'autoroute**

Metrics/highway/F1 Score



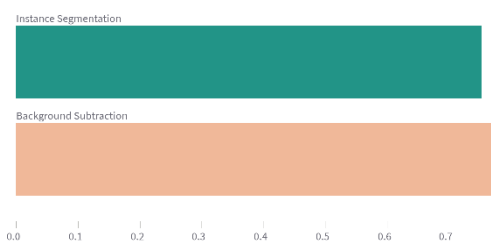
**Vidéo du bureau**

Metrics/office/F1 Score



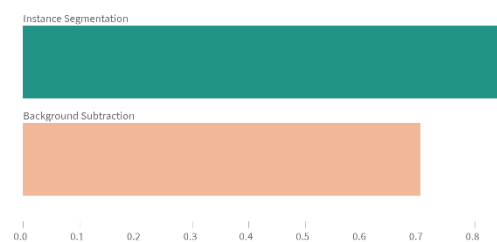
**Vidéo des piétons**

Metrics/pedestrians/F1 Score



**Vidéo de la station de train**

Metrics/PETS2006/F1 Score



**Figure 7.5 Score FI pour chaque vidéo**

En considérant seulement le score F1 pour chacun des images des quatre vidéos, on arrive avec le résultat que la méthode de segmentation d'instance est meilleure à classifié les pixels que la méthode de soustraction d'arrière-plan.

## 8 Discussion des résultats et retour sur les hypothèses (3 pts)

### 8.1 Occlusion

On remarque grâce aux images de la Figure 7.1 Résultat pour le cas d'occlusion que notre théorie était correcte, on remarque que pour la segmentation d'instance, la bout de voiture encore visible en bas de l'image n'est pas détectée alors qu'elle est détectée avec la soustraction d'arrière-plan.

On remarque aussi que les voitures se trouvant au loin, tout en haut de l'image, subissent le même sort, nous n'avions pas prévu ce résultat, mais cela peut montrer que la segmentation d'instance rencontre aussi des difficultés lorsque l'objet est trop petit.

La Figure 8.1 permet de voir la limite d'occlusion que le model par segmentation d'instance supporte, il s'agit en effet de la dernière image ou la voiture occulter est détectée. Cela montre que le model est quand même très résilient à ce type de scenario puisque seulement une petite partie de la voiture n'est visible sur cette image.



Figure 8.1 Résultat par segmentation d'instance montrant le cas limite d'occlusion

### 8.2 Ombres

Sur les images à la

Figure 7.2 Résultat pour le cas d'ombre forte, on voit comment les deux modèles réagissent dans une situation avec des ombres fortes. Comme prévu, le modèle par soustraction d'arrière-plan considère ces ombres comme une objet passant au premier plan, elles sont donc comme un objet au premier plan alors qu'elle ne devrait pas l'être.

Ceci montre les grosses limitations qu'il existe pour le model par soustraction d'arrière-plan, en effet, des changements de luminosité comme cela peuvent être causé par différentes raisons. Dans ce cas-ci, les ombres sont causées par la personne/l'objet passant au premier plan, dans ce cas, bien que la détection de l'ombre soit une erreur de détections, ce n'est pas trop grave puisqu'il y a quand même une zone d'intérêt proche, permettant au système de fonctionner correctement dans certains cas d'utilisation (ex : détection de mouvement globale).

Cependant, il existe des cas où l'ombre/le changement de luminosité peut être causé par d'autres choses qui ne permettraient pas le bon fonctionnement du système, ces autres cas peuvent être liés aux mouvements d'un objet hors champ (une personne passant derrière la caméra peut projeter une ombre) ou encore par de nombreux phénomènes météorologiques (changement de météo, saison, cycle jour/nuit...). Dans ces cas, le système ne fonctionnerait tout simplement pas car l'entière de l'image serait détectée en permanence.

Pour contrer certains de ces problèmes, il existe des techniques consistant à ne pas garder une image de référence fixe, mais de la faire évoluer au cours du temps en utilisant une moyenne glissante par exemple. Cependant, cette méthode ne peut pas résoudre toutes les situations décrites ci-dessus, dans le cas d'un changement brutal de luminosité, la moyenne glissante ne pourra pas empêcher le modèle de prédire des pixels faux-positifs.

Le modèle par segmentation d'instance quant à lui, ne détecte pas ces ombres, le rendant plus performant dans ce type de situation, cela dépend bien entendu du modèle utilisé pour détecter les instances. L'avantage de ces modèles repose sur la compréhension « sémantique » des instances qui sont détectées. En effet, les modèles détectant les instances peuvent « comprendre » que le fait qu'un objet (comme le sol) est à moitié plongé dans l'ombre ne change pas sa classe.

### 8.3 Rotation

Les images de la Figure 7.3 nous montrent comment les deux différents modèles réagissent dans le cas où un objet apparaît sous différents angles. Nous étudions cette situation via la détection d'un livre dans la vidéo « office », dans cette vidéo, le livre est manipulé et est visible dans différentes positions et orientations au cours de cette vidéo.

On peut observer que selon l'orientation de ce livre, le modèle par segmentation d'instance n'arrive pas à le détecter. Cela peut être expliqué par la difficulté qu'ont les réseaux de neurones convolutifs à couvrir toutes les possibles situations auxquels ils sont confrontés. Le modèle n'arrive visiblement pas à généraliser sa connaissance et sa capacité à reconnaître un livre dans cette situation, cela peut être résolu en utilisant un modèle plus puissant/plus spécialisé pour cette tâche précise.

Comme le modèle par soustraction d'arrière-plan ne se base pas sur des connaissances apprises en amont, il ne souffre pas de ce problème et est capable de détecter que les pixels à cet endroit précis ont changé sans pour autant savoir de quoi font-ils partie.

### 8.4 Mouvement parasite

Des mouvements parasites dans l'arrière-plan de la vidéo « highway » sont étudiés via les images de la Figure 7.4, dans cette vidéo, ce sont les feuilles et les branches de l'arbre visible en arrière-plan qui nous intéressent. Comme la vidéo est prise en extérieur, il est logique que ces parties d'arbre bougent légèrement à cause du vent, nous nous sommes donc intéressés à étudier comment chacun des modèles réagit à ce type de scénario.

Comme nous pouvions nous y attendre, le modèle par segmentation d'instance n'est pas du tout perturbé par ces légers changements de positions puisque ce modèle interprète l'image sémantiquement, et qu'un léger mouvement dans l'arrière-plan ne change pas son sens sémantique. L'arrière-plan est donc toujours détecté de la même manière et n'est perçu comme une zone d'intérêt.

Pour la soustraction d'arrière-plan en revanche, est très sensible aux légers changements de couleur/luminosité, le mouvement des branches produit donc des zones d'intérêt dues aux mouvements des arbres. Comme la zone en mouvement est une zone de couleur plutôt uniforme mais très texturée, les faux positifs détectés sont sous la forme de petits groupes de pixels puisque la couleur et la luminosité ne change pas énormément et que ces branches ne bougent pas beaucoup. On peut en revanche imaginer des cas où le mouvement serait beaucoup plus important et sur des objets beaucoup plus contrastés menant à une détection de mouvement beaucoup plus importante.

## 9 Conclusion

Nous avons mis en lumière via nos expériences que les deux méthodes étudiées sont viables, elle comporte toutes les deux des avantages certains, mais aussi des inconvénients notables.

### 9.1 Avantages

#### 9.1.1 Segmentation d'instance

- Peut fonctionner sur n'importe quelle image sans calibration
- Détecte directement les instances et donc directement les formes
- Processus plus puissant qui en général donne des zones d'intérêt plus « propres »
- Permet de savoir directement la classe de l'objet segmenté

#### 9.1.2 Soustraction d'arrière-plan

- Peut détecter un changement sans comprendre son sens sémantique
- Très sensible aux changements de l'image
- Processus plus simple donc plus « certain » et plus facile à vérifier et à mettre en œuvre
- Processus rapide

### 9.2 Inconvénients

#### 9.2.1 Segmentation d'instance

- Nécessite un model performant
- Ne peut détecter que ce que le modèle comprend
- Détecte aussi les instances immobiles en arrière-plan

#### 9.2.2 Soustraction d'arrière-plan

- Besoin d'une phase de calibration
- Sensible aux changements de luminosité et de couleur
- Les zones détectées sont souvent bruitées
- Ne détecte pas les parties de l'objet qui sont de la même couleur (ou proche) que l'arrière-plan

Le choix entre ces deux méthodes dépend donc principalement de la tâche à accomplir, selon les limitations de la tâche, de l'arrière-plan, de la position de la caméra, de l'environnement, etc., la méthode la plus efficace ne sera pas forcément la même.



## 10 Références

Girshick, K. H. (2018). Mask R-CNN. ArXiv. Retrieved from <https://arxiv.org/abs/1703.06870>

wandb. (n.d.). *wandb*. Retrieved from Github: <https://github.com/wandb/wandb>

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