Lab 2 Vision par Ordinateur

Raphaël Le Blanc , Yassine Serroukhe Idrissi

1 Question 1

1.1 Modèles de segmentation à Classes fixes :

Ils consistent en de l'apprentissage supervisé pour de la reconaissance d'objets. Ces modèles sont des réseaux de neurones entraînés sur un ensemble de classes d'objets finis, ce sont dans la grande majorité des cas des réseaux de neurones convolutionnels (CNN) . Ce sont des modèles qui nécessitent beaucoup de données pour s'entraîner afinn de pouvoir reconnaître les objets pour lesquels ils sont entraînés dans de nouvelles images .
Un point important est que ces modèles ne peuvent pas reconnaître des objets/animaux sur lesquels ils n'ont pas été entraînés. Ainsi, ils se généralisent mal pour la détection de tout type d'objets.

Quelques exemples sont : YOLO, Mask-R-CNN, U-net.

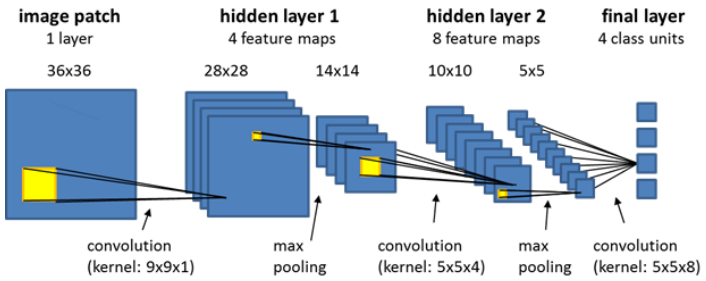

Figure 1: Convolutional Neural Network

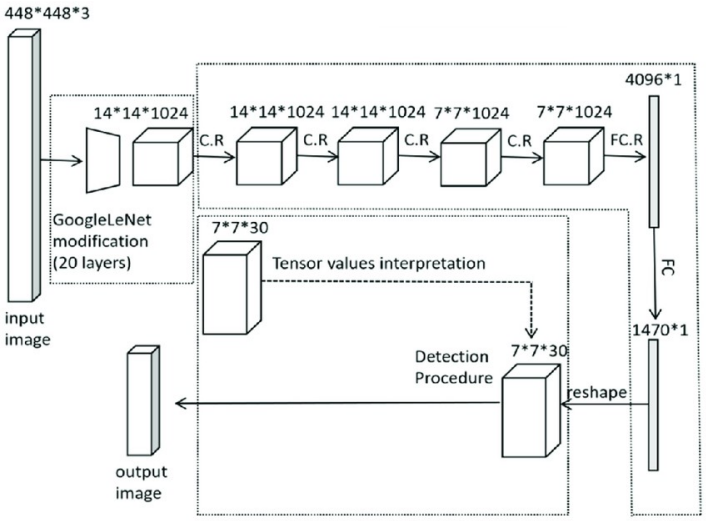

Figure 2: YOLO network

1.2 Modèles Zero-Shot

La segmentation zero-shot vise à segmenter des objets appartenant à des classes qui ne sont pas présentes dans les données d'entraînement. Cette approche s'appuie souvent sur des modèles de vision et de langage capables de généraliser à de nouvelles catégories en exploitant des descriptions textuelles ou des représentations visuelles riches.

Exemples : CLIP, SAM

Le terme "zéro-shot" signifie que le modèle peut traiter une nouvelle tâche sans entraînement spécifique, en généralisant à partir de ce qu'il a appris sur d'autres tâches.

Le modèle commence par extraire des caractéristiques d'une image via un backbone pré-entraîné :

- CNN
- ViT (Vision Transformer)

L'image est donc convertie en une représentation dense et riche, où chaque pixel ou patch contient des informations contextuelles.

Puis, si on prend l'exemple de CLIP, on fournit un texte au modèle qui va ensuite détecter dans l'image la partie correspondant au texte.

Dans le cas de CLIP, il y a 2 réseaux de neurones qui produisent respectivement un embedding pour les images et pour le texte dans le même espace de manière à ce que le mot "vache" par exemple soit proche de l'image d'une vache dans l'espace correspondant au résultat de l'embedding. Ainsi, grâce à un prompt de texte , on peut générer une image ou détecter l'objet du prompt dans l'image.

2 Question 2 : Cas d'utilisation et hypothèses

2.1 Détection d'objets rares

Pour la détection d'objets rares, nous faisons l'hypothèse que les modèles zero-shot sont plus adaptés car , pour de tels objets, il est peu probable que les modèles à classes fixes aient été entraînés sur de tels objets, soit par manque de données soit par manque d'intérêt. Aussi, les modèles à classes fixes ne seront pas capable d'identifier correctement l'objet dans l'image.

2.2 Objets camouflés

Ici, le défi serait de détecter un animal avec un décor qui lui ressemble, par la couleur et/ou la texture. Ici, on peut faire l'hypothèse que les méthodes zéro-shot performent mieux car les animaux camouflés peuvent êtres rares et donc non reconnus par les modèles zéro-shot. Cependant, si les modèles zéro-shot sont entraînés spécifiquement sur un corpus d'images d'animaux camouflés, ils pourraient mieux performer que les modèles zéro-shot.

2.3 Détection de personnes pour de la sécurité ou détection de voitures pour une voiture autonome

Pour de la conduite autonome, on s'attend à ce que les modèles à classes fixes performent mieux car dans ce cas d'utilisation, on ne cherche à détecter qu'un ensemble limité d'éléments (piétons, panneaux, voitures, camions...) ce qui permetrait d'avoir de meilleurs performances avec des modèles entraînés spécifiquement pour ces classes.

3 Question 3 : Description des expériences

3.1 Bases de données

Nous avons utilisé 2 bases de données disctinctes :

- baseline [1] : proposée dans le sujet de TP
- MoCA-mask [2] : trouvée sur SLT-Net proposé dans le discord du cours.

Pour ce qui est de MoCA-mask, nous avons utilisé les vidéos hedgehog et elephant pour tester notre 1ère et 2e hypothèse.

En effet, le modèle YOLO que nous utilisons n'as pas été entraîné pour détecter des hérissons, il est donc pertinent pour notre première hypothèse de tester les 2 modèles sur cet animal.
Pour ce qui est de la 2e hypothèse, l'elephant est pertinent ici car le modèle YOLO a été entraîné pour détecter des éléphants, ici c'est donc bien le camouflage qui va être un défi car si YOLO n'avais pas été entraîné sur des éléphants, la comparaison n'aurait eu aucun intérêt pour la 2e hypothèse.

3.2 Pré-traitement des données

Dans baseline, nous avons utilisé les vidéos "highway" et "pedestrians". Un peu de pré-traitement était nécessaire pour tester les modèles car dans les fichiers "groundtruth" certain masques de segmentation étaient erronés/inexploitables. Pour highway , les 470 premières images de la groundtruth et pour pedestrians, les 300 premières. Ces 2 vidéos ont pour objectif de tester la 3e hypothèse que nous avons énoncé. Ici, nous avons utilisé une vidéo de piétons et une vidéo de voitures pour tester les 2 modèles pour voir lequel des 2 est le plus adapté pour des applications de sécurité routière/voitures autonomes

3.3 Métrique de comparaison

La métrique que nous avons choisi d'utiliser pour comparer les 2 modèles et l'IoU (Intersection over Union) entre le masque obtenu grâce au modèle et la groundtruth fournier avec les vidéos.

Formule de L'IoU :

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

où :

- A est le masque ou la boîte prédite par le modèle,
- B est la ground truth,
- $|A \cap B|$ représente l'intersection des deux zones,
- $|A \cup B|$ représente leur union.

4 Question 4 :

4.1 CLIP

L'implémentation de CLIP est basée sur le notebook du cours de Vision par Ordinateur : Zero Shot Segmentation.ipynb [3]

Elle repose sur une fonction que nous avons codé nous même : get results() qui prend en entrée une liste d'images à laquelle est associée un prompt avec la liste de groundtruth correspondante et qui calcule l'IoU pour chaque image et renvoie la liste des IoU.

La fonction permet également d'afficher les masques de segmentation dans le cas ou l'IoU est mauvais pour comprendre pourquoi.

4.2 YOLO

L'implémentation présentée s'appuie sur le modèle YOLO de la bibliothèque ultralytics[4] pour réaliser la segmentation d'objets sur différentes catégories (piétons, véhicules, hérissons, éléphants).

Le code définit plusieurs fonctions dédiées que nous avons codé – par exemple, $\text{get}_r esults_y olo_p edestrian, get_r esults_y olo_c$

Ces fonctions réalisent les étapes suivantes :

- Exécution de l'inférence avec le modèle YOLO pour obtenir les masques et les classes détectées.
- Filtrage des détections en fonction de la classe ciblée (ex. "person" pour les piétons).
- Calcul de l'Intersection over Union (IoU) entre le masque prédit et le masque de référence.

- Affichage des masques de segmentation en cas d'IoU très faible ou très élevé pour faciliter l'analyse qualitative des résultats.

5 Question 5 : Résultats

5.1 CLIP

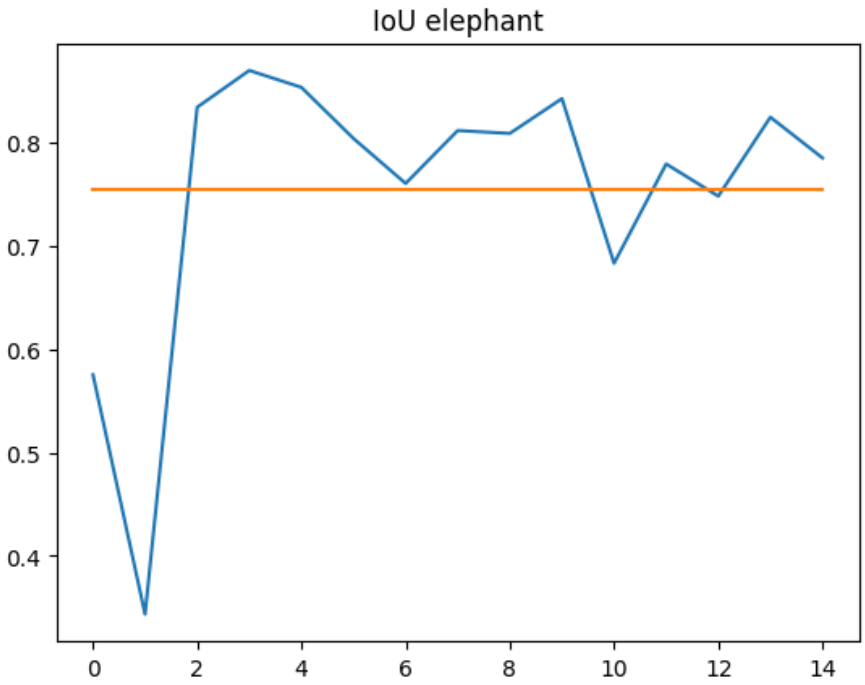

Figure 3: IoU par Image pour la vidéo de l'Elephant, la moyenne sur la vidéo est le trait orange

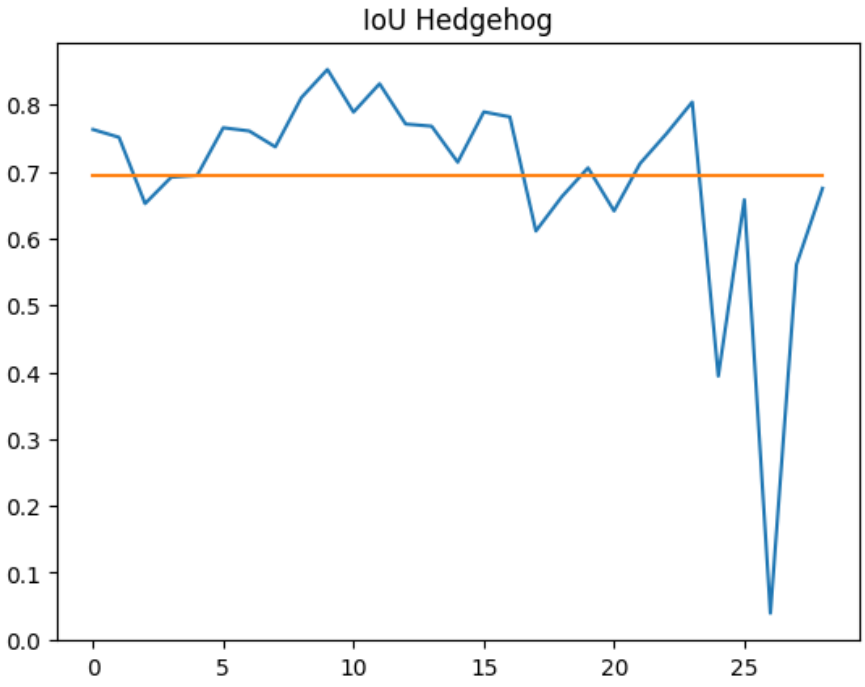

Figure 4: IoU par Image pour la vidéo du Hérisson, la moyenne sur la vidéo est le trait orange

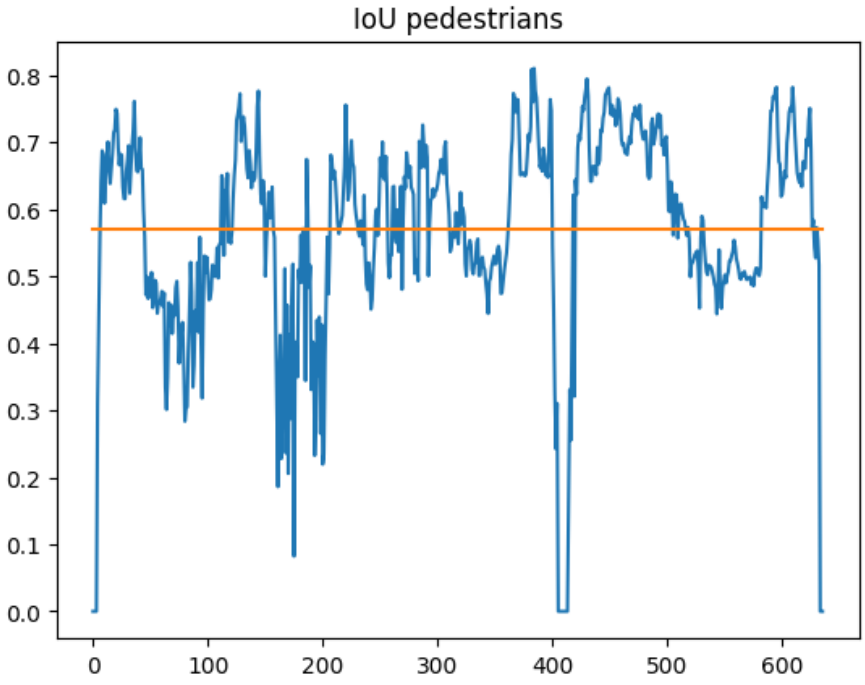

Figure 5: IoU par Image pour la vidéo des piétons, la moyenne sur la vidéo est le trait orange

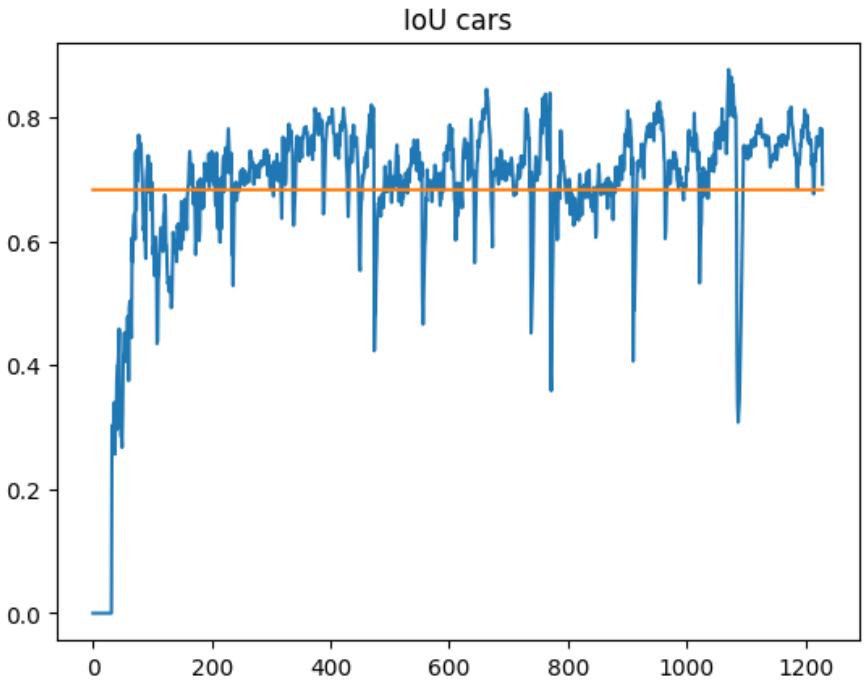

Figure 6: IoU par Image pour la vidéo de l'autoroute, la moyenne sur la vidéo est le trait orange

5.2 YOLO

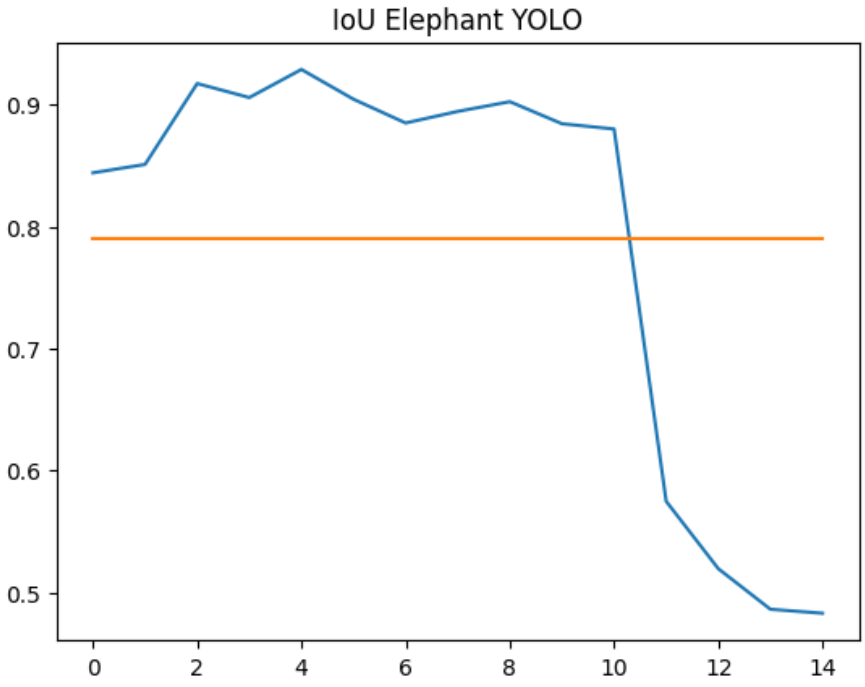

Figure 7: IoU par Image pour la vidéo de l'Elephant, la moyenne sur la vidéo est le trait orange avec YOLO

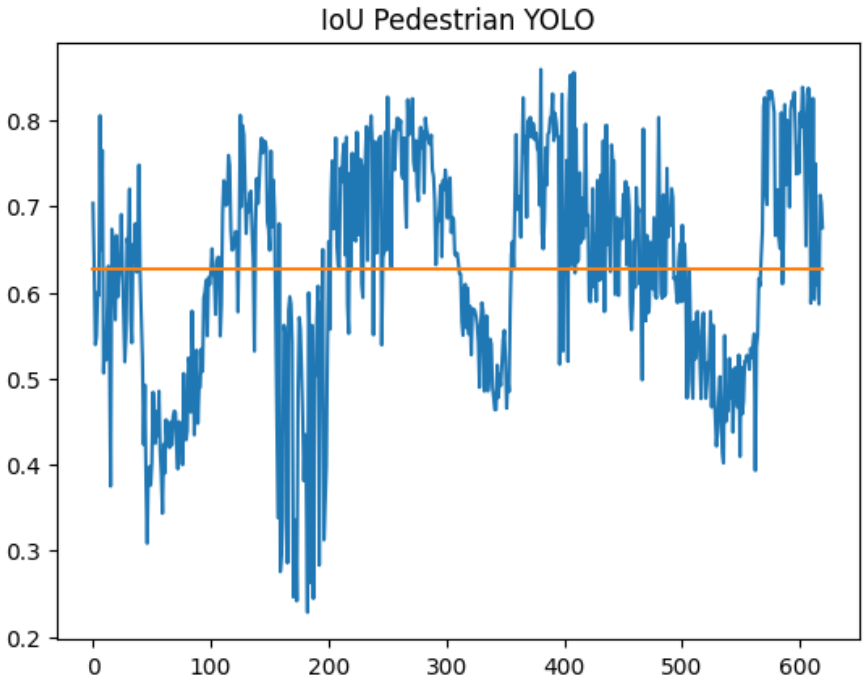

Figure 8: IoU par Image pour la vidéo des piétons, la moyenne sur la vidéo est le trait orange avec YOLO

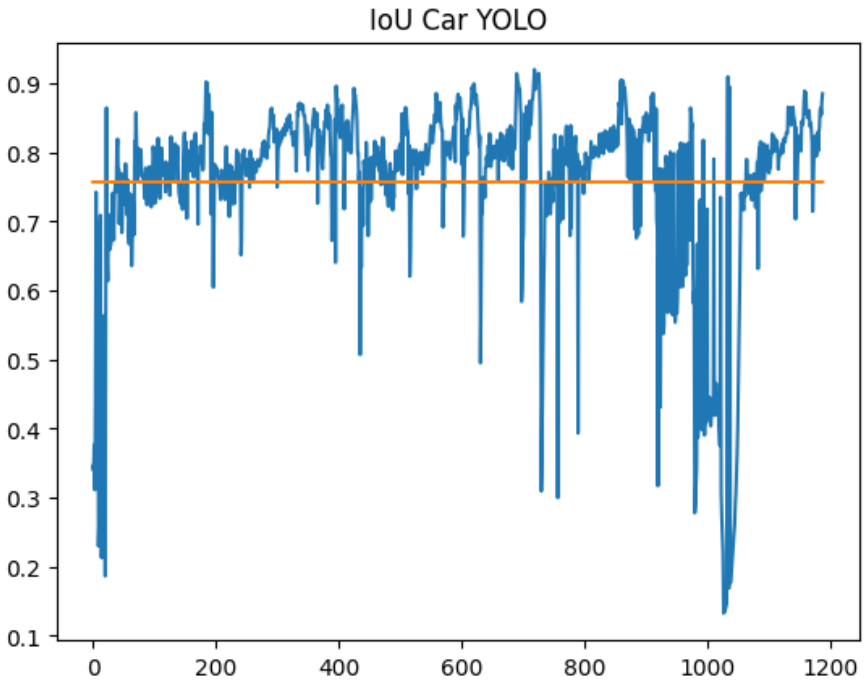

Figure 9: IoU par Image pour la vidéo de l'autoroute, la moyenne sur la vidéo est le trait orange ,avec YOLO

6 Question 6 : Analyse des résultats

6.1 1ère Hypothèse : Détection d'objets rares

Grâce aux résultats comparatifs sur la vidéo du hérisson, on peut vérifier l'hypothèse 1 puisque le modèle zéro-shot a plutôt bien détecté le hérisson tout au long de la vidéo tandis que le modèle à classes fixes qui n'as pas été entraîné avec une classe hérisson ne le détecte pas du tout.

Ainsi, on peut voir que dans le cas de détections d'objets peu courant, donc peu susceptibles d'apparaître dans les classes d'entraînnement de modèlles à classes fixes, les modèles zéro-shot sont bien meilleurs.

6.2 2e hypothèse : Détection d'objets camouflés

Pour ce qui est des objets camouflés, nous avons testé sur une vidéo d'éléphant car c'est un des animaux sur lequel le modèle YOLO que nous utilisons a été entraîné.

On peut voir que les 2 modèles détectent assez bien l'éléphant au début de la vidéo quand il n'est pas encore bien camouflé. Mais , vers la fin de la vidéo, l'éléphant n'est plus bien détecté par YOLO tandis que CLIP obtient toujours de résultats plutot bons (¿0.70 en IoU). Cependant, en regardant de plus près la fin de la vidéo, on s'aperçois qu'à la fin de la vidéo, une partie de l'éléphant est en fait cachée par un autre éléphant qui n'est pas inclu dans la groundtruth, or, YOLO détecte le 2e éléphant tandis que CLIP ne le détecte pas , ce qui explique l'écart de performance entre les 2 .

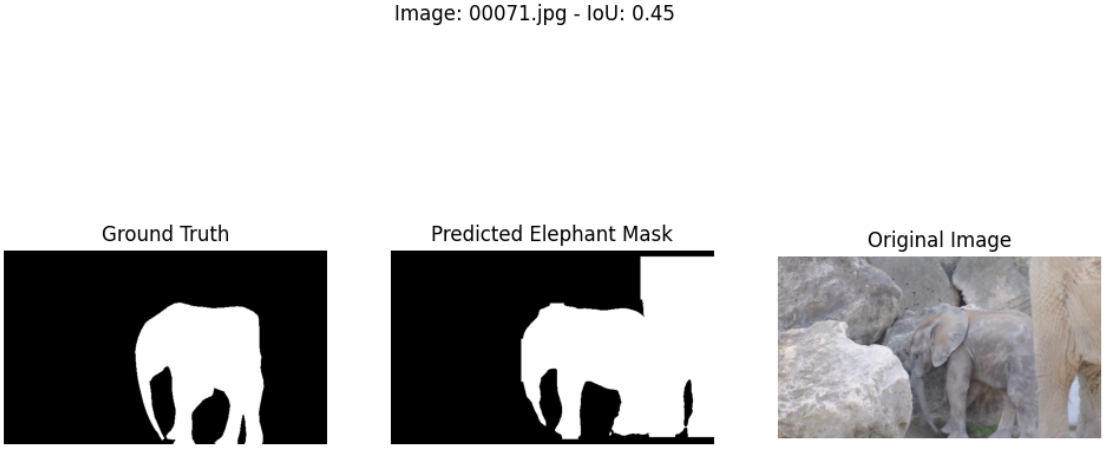

Figure 10: Exemple de masque YOLO avec les 2 éléphants

Aussi, on ne peut pas vraiment conclure pour savoir lequel des 2 modèles est le meilleur. En revanche, comme on l'a vu avec l'hypothèse 1, si les objets sont rares/ne font pas partie des classes d'entraînement de YOLO, alors zero-shot sera bien meilleur.

6.3 3e Hypothèse : Détection de voiture et de piétons pour voitures autonome et sécurité.

Ici, les résultats des 2 mesures sont relativement similaires, toutefois, le modèle à classes fixes semble être légèrement meilleur lorsqu'on regarde l'IoU moyen sur les 2 vidéos :

Vidéo	**YOLO**	**CLIP**
voitures	0.76	0.69
piétons	0.63	0.57

Table 1: Comparaison des performances des modèles sur deux vidéos (IoU moyen).

On a donc des résultats qui confirment notre hypothèse puisque le modèle YOLO a été entraîné pour détecter des personnes et des voitures.

A Annexe: sources informations supplémentaires

A.1 Sources

- Liste des classes du modèle YOLO utilisé
- [1] Base de donnée d'où proviennent les vidéos pour les piétons et les voitures
- [2] Base de données d'où proviennent les vidéos du hérisson et de l'éléphant
- [3] Code dont nous nous sommes inspirés pour CLIP
- [4] La bibliotheque que nous avons utiliser pour YOLO