
Détection et suivi de personnes dans des séquences d'images par CNN pour la protection de la vie privée.

CR #3 - Développement :

*CANHOTO Mickaël,
FONTAINE Emmanuel
Master 2 Imagine*



Détection et suivi de personnes	1
dans des séquences d'images par CNN pour	1
la protection de la vie privée.	1
CR #3 - Développement :	1
I - Introduction	1
II - Choix du datasets	1
III - Lecture Vidéo	2
IV - Implémentation du CNN avec YOLO	2
V - Conclusion	3

I - Introduction

Ce troisième rapport se concentre maintenant sur le choix du(des) dataset(s), ainsi qu'une première approche d'implémentation du CNN en utilisant le modèle YOLO. Nous effectuerons un premier test sur une vidéo de caméra de surveillance.

II - Choix du datasets

Pour l'entraînement de notre CNN, nous avons trouvé 4 datasets qui s'adapte à nos besoins:

- [COCO \(Common Objects in Context\)](#) : Le dataset COCO, développé par l'équipe de Microsoft, est un ensemble de données volumineux et diversifié contenant plus de 330 000 images annotées. Parmi les annotations, on trouve des détections de personnes dans diverses situations et contextes.
- [Penn-Fudan Pedestrian Dataset](#) : Ce dataset est spécifique à la détection de piétons dans un environnement urbain et se compose de 170 images annotées. Chaque image comporte des masques de segmentation permettant de délimiter précisément les piétons.
- [PnPLO](#) : est un dataset trouvé sur le site Kaggle nous permettant d'entraîner notre CNN.
- [WiderPerson](#) : Ce dataset est conçu pour la détection de personnes dans des environnements variés et présente une grande diversité de contextes, d'échelles et d'angles de vue. Il comprend plus de 13 000 images et des annotations détaillées.

III - Lecture Vidéo

Pour lire et traiter les vidéos, nous avons implémenté une classe [VideoReader](#) (dans [VideoReader.py](#)). Cette classe extrait les images d'une vidéo et les met à disposition pour le traitement image par image. Voici les principales fonctions de cette classe :

- **extract** : Charge toutes les frames de la vidéo.
- **get_frame(index)** : Récupère une frame (image) spécifique.
- **get_frame_count** et **get_fps** : Fournissent le nombre de frames et la fréquence d'images par seconde.

IV - Implémentation du CNN avec YOLO

Pour la détection des personnes, nous utilisons le modèle YOLO avec une version pré-entraînée, spécifiquement [yolo11n.pt](#). YOLO (You Only Look Once) est un modèle de détection en temps réel performant et bien adapté pour ce projet.



Détection de personnes

On remarque que le réseau neuronal fonctionne bien même s'il reste encore de nombreuses choses à améliorer.

V - Conclusion

En conclusion, nous avons un début d'implémentation fonctionnelle qui faudra ensuite améliorer. Par le suivi de détection avec ByteTrack, le prétraitement de la vidéo, l'ajout d'une interface en finalement l'anonymat des personnes.