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

CR #4 - Interface, Structuration et Optimisation :

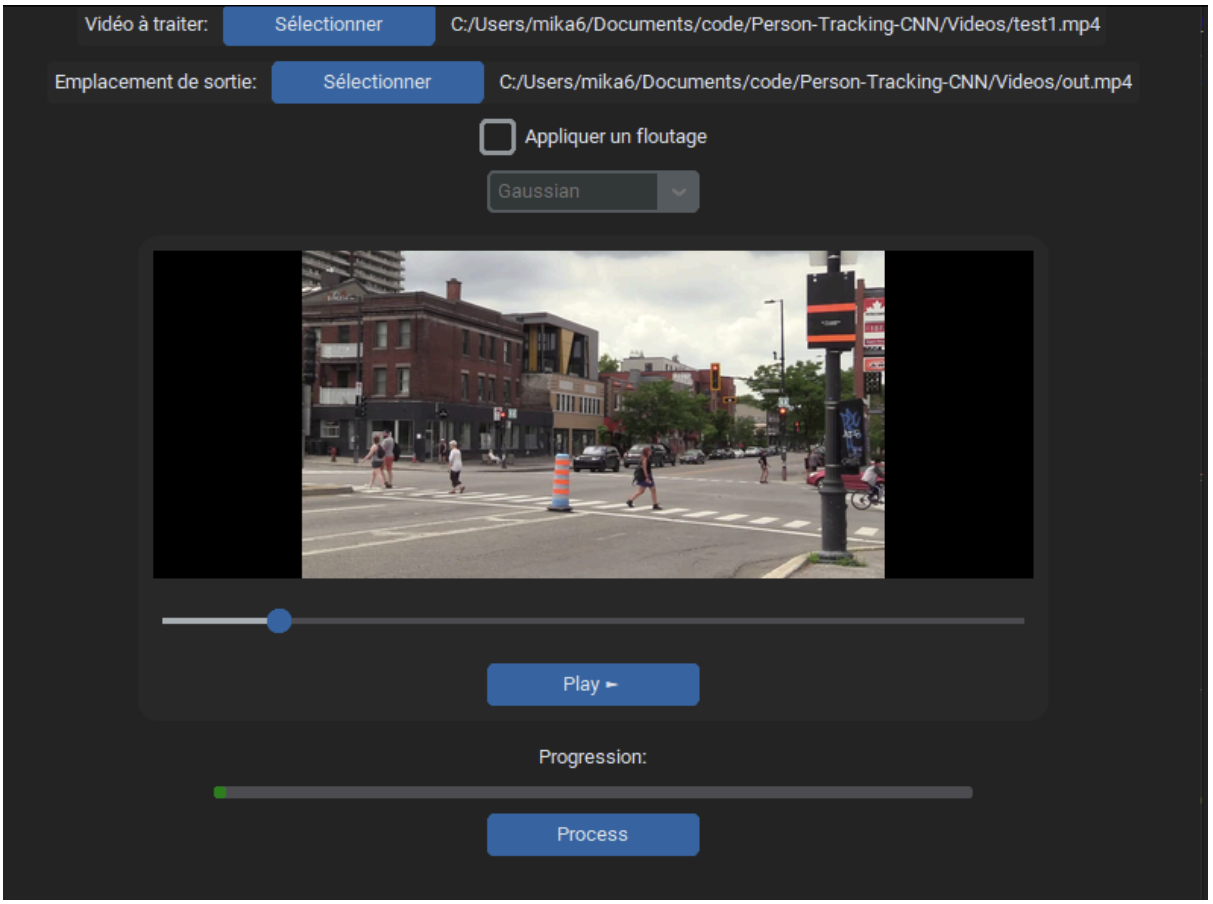
CANHOTO Mickaël,
FONTAINE Emmanuel
Master 2 Imagine

I - Introduction

Ce quatrième rapport se concentre sur l'interface utilisateur et quelques optimisations clés du projet, avec un accent particulier sur l'architecture de l'application, la structuration du code et les performances générales.

II - Interface

L’interface graphique à été construite avec la librairie Tkinter et CustomTKinter. Dans l’application nous pouvons choisir la vidéo, l’emplacement de la vidéo traité, l’anonymisation de la vidéo, la prévisualisation de la vidéo et une barre de progression.



Interface graphique

III - Structuration

Pour la structuration de notre programme, nous avons décomposé le projet en trois fichiers principaux afin de bien séparer les responsabilités et de garantir une meilleure organisation du code. Les fichiers sont :

- **`App.py`** : Ce fichier contient la logique de l'interface graphique. Il gère les interactions avec l'utilisateur, l'affichage des vidéos, les boutons, et les contrôles.
- **`VideoReader.py`** : Ce module est consacré aux fonctions de lecture et d’écriture de vidéos. Il s'occupe du traitement des flux vidéo, de l'acquisition des images, de la gestion des formats et de l'exportation des fichiers.
- **`Détection.py`** : Ce fichier contient le code pour l'utilisation du réseau de neurones convolutif (CNN) ainsi que les fonctionnalités d'anonymisation. Il traite les images extraites de la vidéo et applique le modèle de détection pour identifier les différentes

personnes. Une fois les détections effectuées, ce module applique les techniques d'anonymisation nécessaires (tel que le floutage, par exemple) pour garantir la confidentialité.

IV - Optimisation

1) Gestion mémoire

Le programme initial était très coûteux en termes de mémoire RAM. Pour éviter cela, nous lisons les frames au fur et à mesure plutôt que de les charger toutes en mémoire. Même principe pour l'écriture des frames, nous enregistrons chaque frame traitée directement.

2) Utilisation du GPU

La deuxième optimisation du programme est l'utilisation des composants graphiques (GPU) pour l'utilisation du CNN. Grâce à PyTorch, nous pouvons utiliser CUDA afin de réduire de manière considérable notre temps de traitement. Si CUDA n'est pas utilisable, le programme sera lancé sur le processeur (CPU).

Ces optimisations ont permis de réduire considérablement le temps de traitement d'une vidéo.

Pour une vidéo de 30 secondes en 1280x720 à 60 frames/secondes, nous sommes passés de 195 secondes à 60 (CPU : i5-9700, GPU : Nvidia 1070)

V - Conclusion

Pour conclure, cette semaine, nous nous sommes attardés sur l'interface utilisateur ainsi que quelques optimisations afin d'améliorer la vitesse de traitement.