



BLUR ME IF YOU
CAN

PROJET IMAGE HAI927I
DÉTECTION ET SUIVI DE PERSONNES DANS DES SÉQUENCES
D'IMAGES PAR CNN POUR LA PROTECTION DE LA VIE PRIVÉE.

13/12/2024

PLAN

- 01** CONTEXTE
- 02** ETAT DE L'ART
- 03** MÉTHODES
- 04** RÉSULTATS ET COMPARAISONS
- 05** DÉMONSTRATION
- 06** CONCLUSION

CONTEXTE



L'anonymat peut s'avérer essentiel lors de documentaires vidéos, il est donc important d'avoir des outils permettant de le garantir.



Nous allons utiliser des réseaux de neurones pour détecter les visages et y appliquer différentes méthodes d'anonymisation.



“Alain”, millionnaire “anonyme”



Extrait d'enquête d'action

ETAT DE L'ART

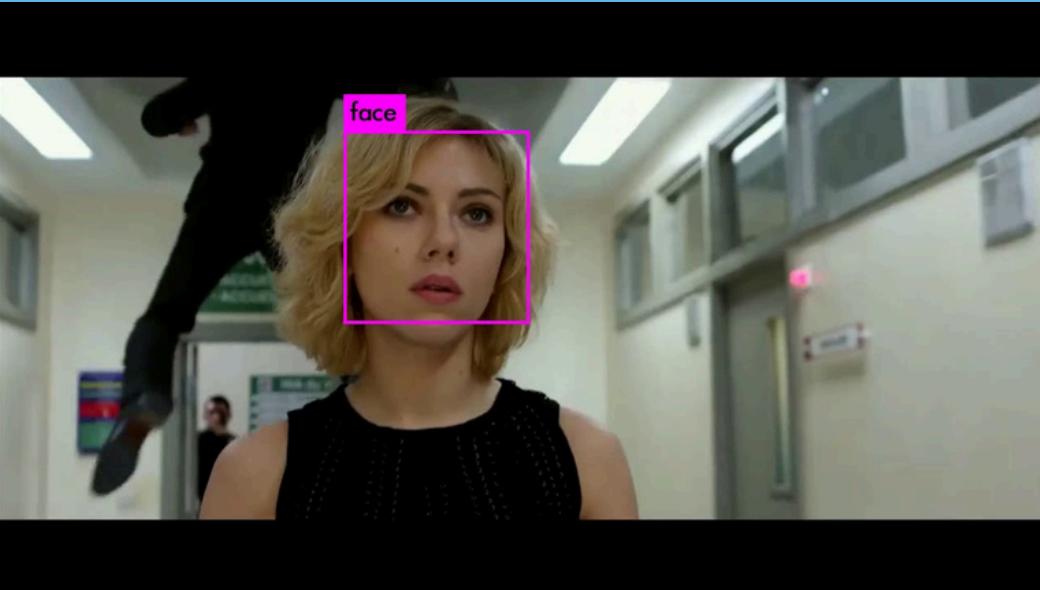
Pré-traitement

- Filtre médian
- Filtre gaussien
- Égalisation d'histogramme



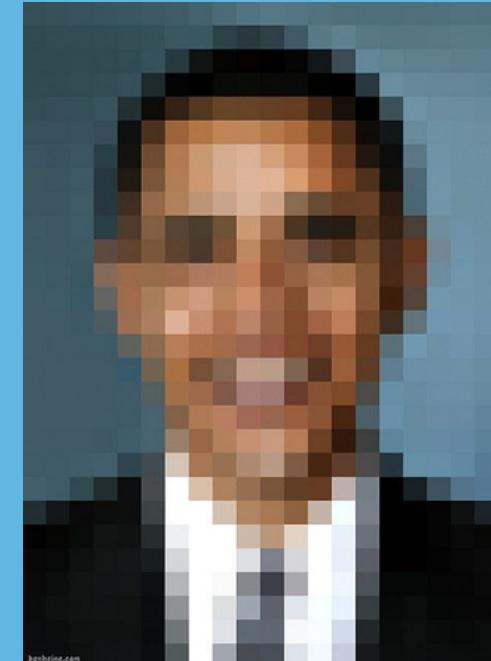
Suivi d'objet

Utilisation de réseaux de neurones convolutionnels (YOLO, RESNet).



Technique d'obscuration

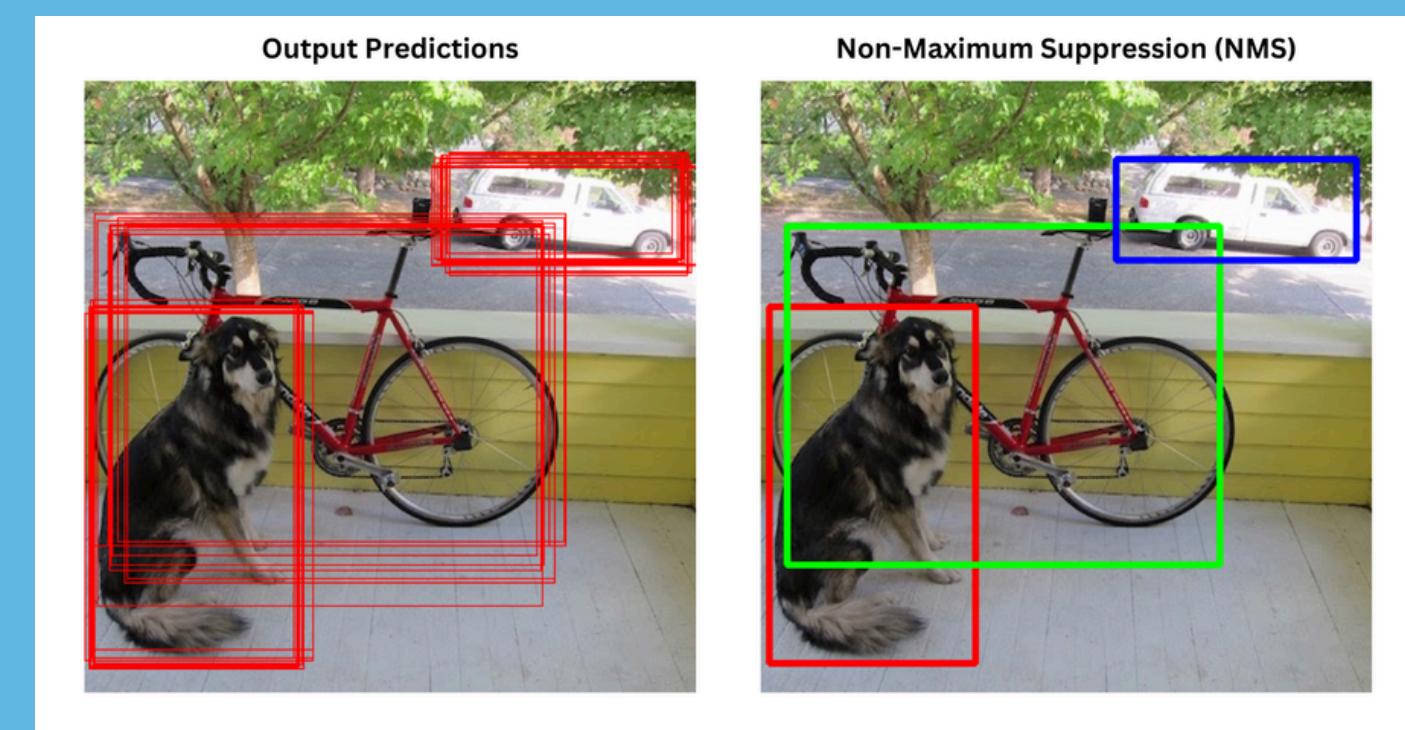
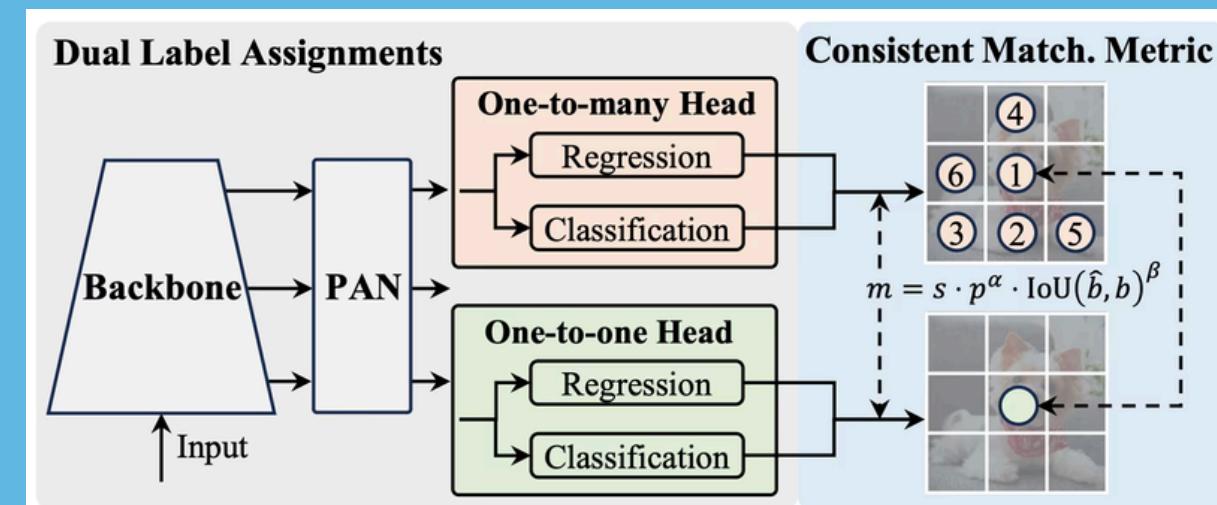
- Flou
- Pixellisation
- Mélange de pixels
- Chiffrement AES



ETAT DE L'ART

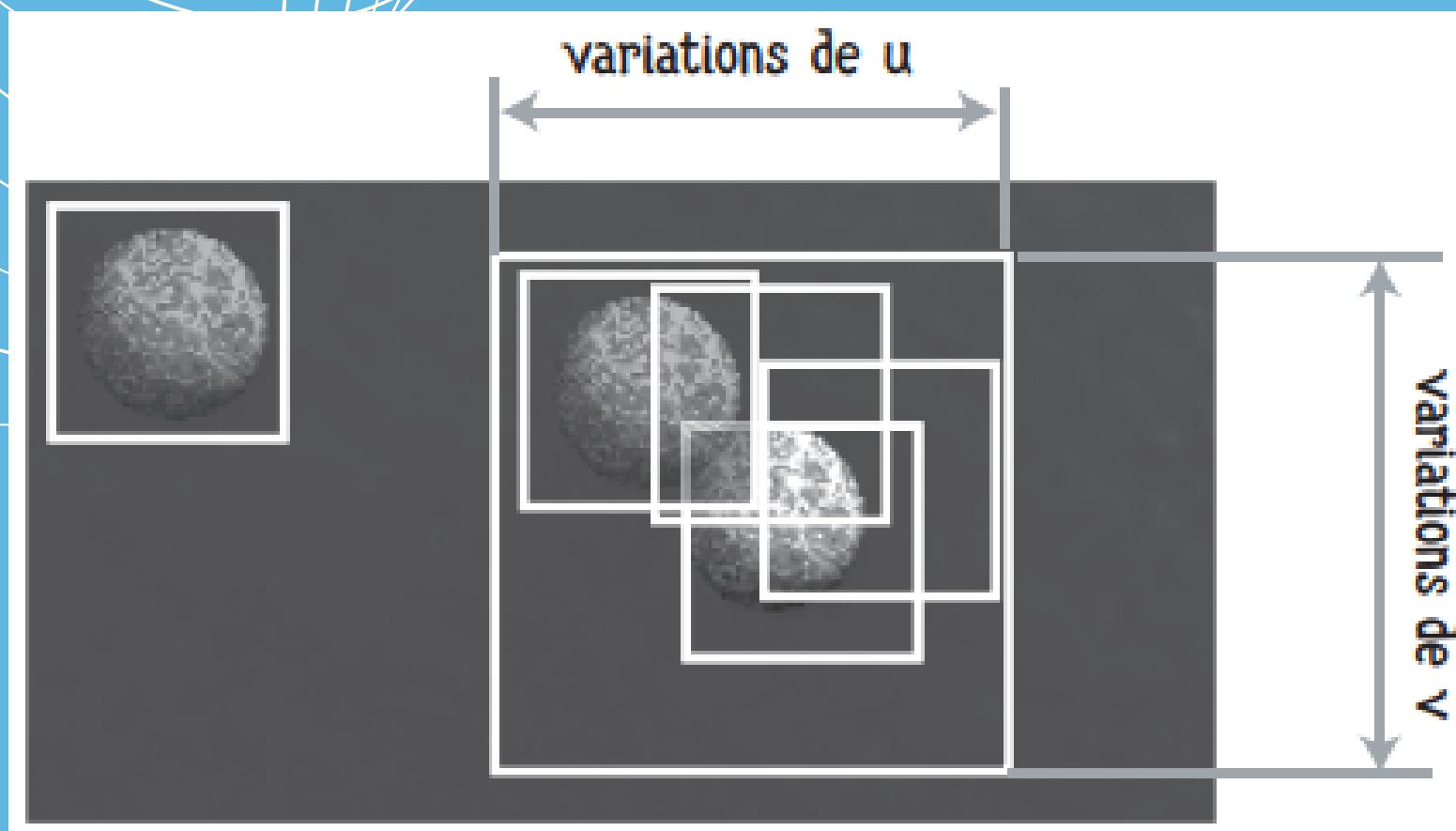
DÉTECTION PAR YOLOV10

- Extraction des caractéristiques de l'image
- Prédiction et classification.
- Pas de redondance dans les prédictions (NMS)
- Rapide et précis



ETAT DE L'ART

POURSUITE DE CIBLE



- Alternative à YOLO
- Définition de la cible
- Suivi de la cible par CSRT
(Discriminative Correlation Filter with Channel and Spatial Reliability Tracker)

NOS MÉTHODES

- Utilisation du modèle YOLO
- Reconnaissance sur chaque frame
- Numérotation des boîtes dans l'ordre

FULL YOLO

- Utilisation du modèle YOLO et de traqueurs
- Reconnaissance sur la première frame
- Poursuite de cible
- Détection de perte

HYBRIDE

RESULTS

Le modèle arrive très bien à détecter les visages à chaque frame, ce qui permet une bonne anonymisation



Frame originale



Frame avec détection par YOLO

RESULTATS

Le mélange des pixels fonctionne bien sur toutes les boîtes détectées + YOLO a du mal à reconnaître les visages anonymisés



Frame après mélange des pixels de chaque visage



Frame avec détection par YOLO
des visages mélangés

COMPARAISON

On observe bien que l'une des deux techniques est plus efficace que l'autre

C:\> Users > delvi > OneDrive > Bureau > Projet_Image_M2_BELOT_DELVIGNE > Code > src > interface.py

```
def nival vidan()\n\nbrian@LAPTOP-HCQT44PG:/mnt/c/Users/delvi/OneDrive/Bureau/Projet_Image_M2_BELOT_DELVIGNE/Code/src$ def nival vidan()\nbrian@LAPTOP-HCQT44PG:/mnt/c/Users/delvi/OneDrive/Bureau/Projet_Image_M2_BELOT_DELVIGNE/Code/src$ python3\n\n0: 480x640 2 faces, 364.9ms\nSpeed: 7.8ms preprocess, 364.9ms inference, 0.4ms postprocess per image at shape (1, 3, 480, 640)\nProcessing complete.\nTemps de calcul : 7.91471 secondes\n\n0: 480x640 2 faces, 398.5ms\nSpeed: 18.2ms preprocess, 398.5ms inference, 0.3ms postprocess per image at shape (1, 3, 480, 640)\n\n0: 480x640 2 faces, 416.6ms\nSpeed: 38.3ms preprocess, 416.6ms inference, 0.3ms postprocess per image at shape (1, 3, 480, 640)\n\n0: 480x640 2 faces, 296.8ms\nSpeed: 4.6ms preprocess, 296.8ms inference, 0.3ms postprocess per image at shape (1, 3, 480, 640)\n\n0: 480x640 2 faces, 348.9ms\nSpeed: 18.1ms preprocess, 348.9ms inference, 0.3ms postprocess per image at shape (1, 3, 480, 640)\n\n0: 480x640 2 faces, 240.1ms\nSpeed: 10.4ms preprocess, 240.1ms inference, 0.3ms postprocess per image at shape (1, 3, 480, 640)\n\n0: 480x640 2 faces, 307.7ms\nSpeed: 8.6ms preprocess, 307.7ms inference, 0.3ms postprocess per image at shape (1, 3, 480, 640)\n\n0: 480x640 2 faces, 266.1ms\nSpeed: 17.1ms preprocess, 266.1ms inference, 0.3ms postprocess per image at shape (1, 3, 480, 640)\n\nif similarity < float(seuil_Entry.get()):\n    print(f"tracker {i} a perdu la cible, tentative de réinitialisation...")\n    reset_tracker(processed_frame, i, tracker_histograms[i])\n    # Option : marquer la boîte ou réinitialiser le tracker\nelse :\n\nfrom Microsoft for the Python language?\nInstall Show Recommendations\n\nDuo Ln 468, Col 27 Spaces: 4 UTF-8 LF Python\n\nTaper ici pour rechercher
```

Mélange FULL YOLO

Mélange Hybride

COMPARAISON



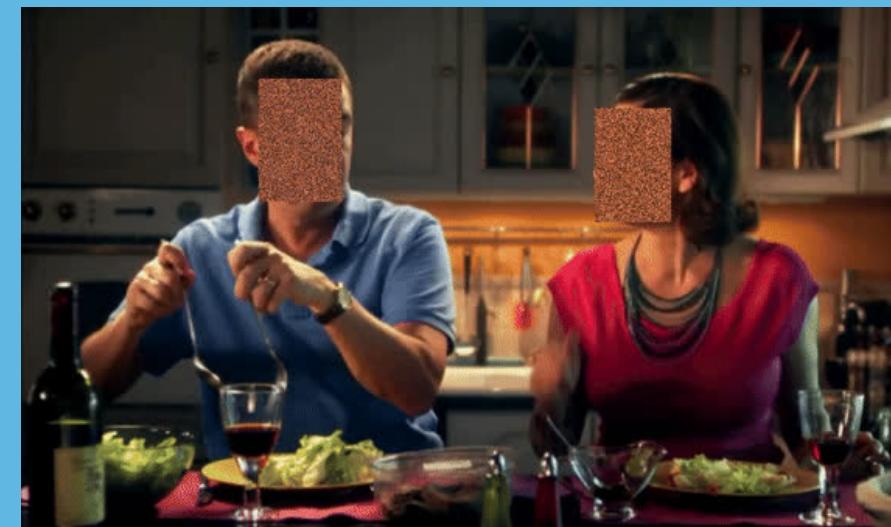
Vidéo originale



Vidéo flou sur 1 seul visage (YOLO)



Vidéo flou sur 1 seul visage
(poursuite de cible)



Vidéo mélange sur tous les visages

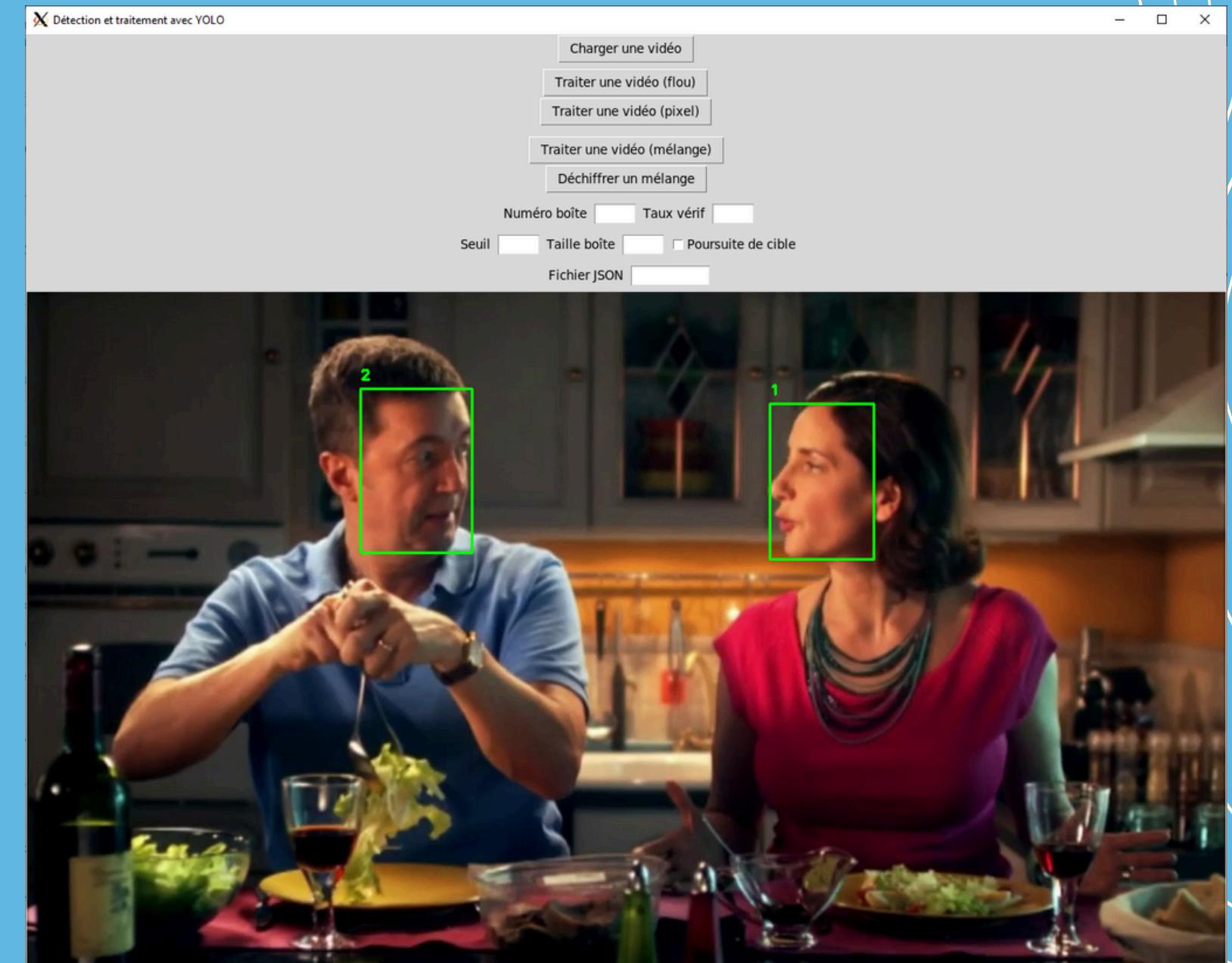


Vidéo déchiffrée

DÉMONSTRATION - INTERFACE

Fonctionnalités de l'interface :

- Chargement de vidéos (.mp4, .avi)
- 1ère détection par YOLO.
- Choisir une boîte précise ou faire tous les visages ainsi que leur taille.
- Choisir le type d'obscuration/Déchiffrer dans le cas d'un mélange.
- Utiliser la poursuite de cible ou non.
- Poursuite -> seuil et taux de vérification



DÉMONSTRATION - APPLI

CONCLUSION



Les limites

- Nécessité d'avoir les visages sur la première frame
- Superposition des visages pour le déchiffrement
- Gestion des changements de plan
- Disparition et réapparition des visages dans le cadre

CONCLUSION

En bref



- Deux méthodes permettant de reconnaître et de suivre les visages
- 3 méthodes d'anonymisation dont une réversible
- Une application simple pour obtenir un résultat en quelques clics

- Utilisation d'un modèle plus robuste que YOLO
- Chiffrement du fichier .json
- Amélioration de la détection hors champ

Perspectives d'amélioration



RÉFÉRENCES

- <https://docs.ultralytics.com/models/yolov10/>
- <https://github.com/AlexanderMelde/SPHAR-Dataset/tree/master>
- https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html
- https://docs.opencv.org/3.4/d2/da2/classcv_1_1TrackerCSRT.html