PROJET FLOUKI : Floutage d'objet sur flux

Segura Lucas / Roussel Maxime

January 2024

Table des matières

PROJET FLOUKI : Floutage d'objet sur flux

Segura Lucas / Roussel Maxime

January 2024

Figure 1 – image flouter avec Flouki

1 Introduction

C'est dans le contexte du cours I122 traitement de signals que s'inscrit le projet "Flouki" qui fait reference au dieu de la malice Nordique (Loki) ,car c'est une solution de floutage vidéo basée sur l'intelligence artificielle a l'aide de la technologie YOLO développé par Ultralytics et la librairie Opencv proposé sur python. Flouki vise à flouter automatiquement les visages

et plaque d'immatriculations des vidéos et images, tout en maintenant une qualité visuelle optimale.

2 Prérequis

```
# Ultralytics requirements
Instalation: pip install -r requirements.txt

# Base ----------------------------------------
hydra-core>=1.2.0
matplotlib>=3.3
numpy>=1.18.5
opencv-python>=4.1.1
Pillow>=7.1.2
PyYAML>=5.3.1
requests>=2.23.0
scipy>=1.4.1
torch>=1.7.0
torchvision>=0.8.1
tqdm>=4.64.0

# Logging -------------------------------------
tensorboard>=2.4.1
# clearml
# comet

# Plotting ------------------------------------
pandas>=1.1.4
seaborn>=0.11.0

# Export --------------------------------------
# coremltools>=6.0  # CoreML export
# onnx>=1.12.0  # ONNX export
# onnx-simplifier>=0.4.1  # ONNX simplifier
# nvidia-pyindex  # TensorRT export
# nvidia-tensorrt  # TensorRT export
# scikit-learn==0.19.2  # CoreML quantization
# tensorflow>=2.4.1  # TF exports (-cpu, -aarch64, -macos)
# tensorflowjs>=3.9.0  # TF.js export
# openvino-dev  # OpenVINO export

# Extras --------------------------------------
ipython  # interactive notebook
psutil  # system utilization
thop>=0.1.1  # FLOPs computation
# albumentations>=1.0.3
# pycocotools>=2.0.6  # COCO mAP
```

```
    # roboflow

    # HUB ----------------------------------------
    GitPython>=3.1.24
```

3 Manuel d'utilisation

```
### Prérequis

- Python 3.x
- OpenCV (‘pip install opencv-python‘)
- Matplotlib (‘pip install matplotlib‘)
- Pillow (‘pip install pillow‘)
- Numpy (‘pip install numpy‘)
- ultralytics (‘pip install ultralytics‘)
- Nvidia Cuda
- PyTorch

- Ce projet a était travailler avec un systeme d'exploitation sous Window
    il est donc concu pour travailler avec ce genre d'environnement
    il se pourrais donc qu'il ne fonctionne pas avec les d'autres système.

### Utilisation

1. Clonez le dépôt :

   ‘‘‘bash
   git clone https://github.com/Maxilef/FLOUKI
   cd yolo-detection-objets-floutage
   ‘‘‘

2. Installez les dépendances requises :

   ‘‘‘bash
   pip install -r requirements.txt
   ‘‘‘
   il s'agit des dependences par defaut pour ultralitcs

3. Exécutez le script pour la détection et le floutage :

    par default il sagit de notre reseaux de neurones entrainner pour en utliser un au
     cela nessecite de modifier le code suivant
    model = YOLO(r".\runs\detect\train4\weights\best.pt") #votre reseau ici

    les resultats sont sauvegarder dans runs/save
```

- Pour le floutage sur une image :
  ```bash
  python detectme.py chemin/vers/votre/image.jpg
  ```

- Pour le floutage sur une vidéo :
  ```bash
  python detectme.py chemin/vers/votre/video.mp4
  ```

- Pour le floutage sur la webcam en direct :
  ```bash
  python detectme.py 0
  ```

4. Appuyez sur 'q' pour quitter le programme.

Configuration

- Vous pouvez ajuster le niveau de flou en modifiant la taille du noyau dans les appels

- Les images et vidéos de sortie avec les objets floutés seront enregistrées dans le re

N'hésitez pas à modifier le code selon vos besoins spécifiques !

5. Pour entrainer une dataset :
   ```bash
   python train_dataset.py
   ```
   il faudra alors dans le programme changer le chemin du dossier, de la ou se trouv

  ex :
  path_dataset = r"C:\Users\maxim\Desktop\yolov8\dataset\plate_and_face_detection.yo

4 Technologies Utilisées

4.1 YOLO : Detection objet

La technologie YOLO (You Only Look Once) développée par Ultralytics offre un aperçu captivant du domaine de la détection d'objets en vision par ordinateur. La détection d'objets constitue un pilier fondamental dans de nombreux domaines, allant de la surveillance vidéo à la conduite autonome en passant par la réalité augmentée. YOLO, en tant que système novateur, a été conçu pour résoudre efficacement ce défi en fournissant une méthode rapide et précise pour localiser et classifier divers objets au sein d'une image.C'est grace Yolo que nous avons pu detecter les objets a flouter dans les differents flux (image, vidéo).

La technologie YOLO utilise un réseau de neurones de type convolutionnel (CNN)2 pour effectuer la détection d'objets. Ces réseaux reposent sur des filtres de convolution (matrices numériques). Les filtres sont appliqués aux entrées avant que celles-ci ne soient transmises aux neurones. Ces réseaux de neurones sont utiles pour le traitement et la prévision d'images. YOLO utilise l'apprentissage supervisé (figure4) pour entraîner son modèle de détection d'objets. L'apprentissage supervisé est une approche dans laquelle le modèle est entraîné sur un ensemble de données annotées, où chaque exemple d'entraînement est associé à une étiquette (label) qui indique la classe de l'objet ainsi que sa localisation dans l'image.

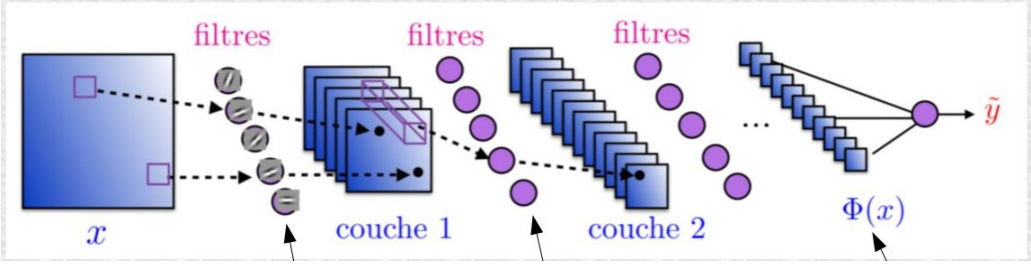

FIGURE 2 – Schema reseau de neuronne convolutionnel.

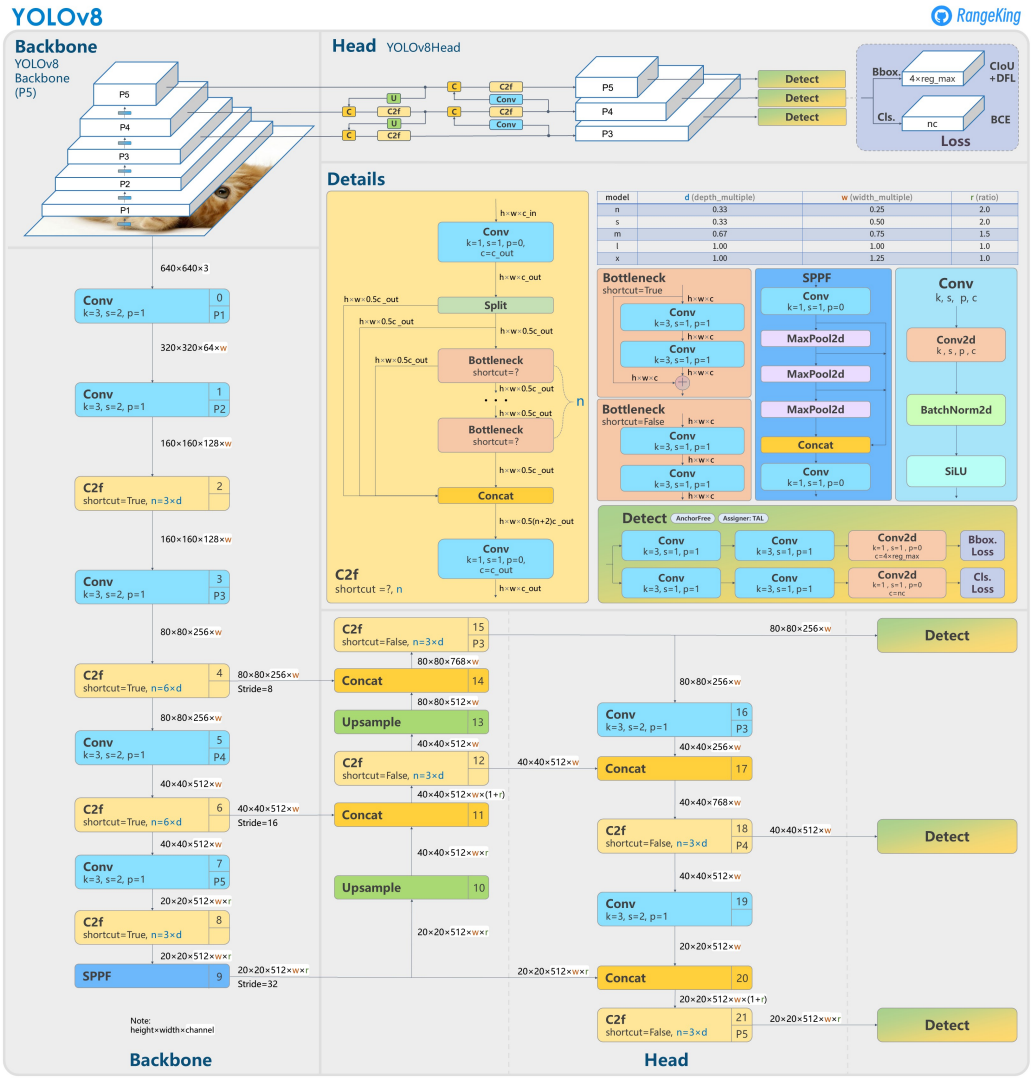

FIGURE 3 – Schema reseau de neuronne YOLOv8.

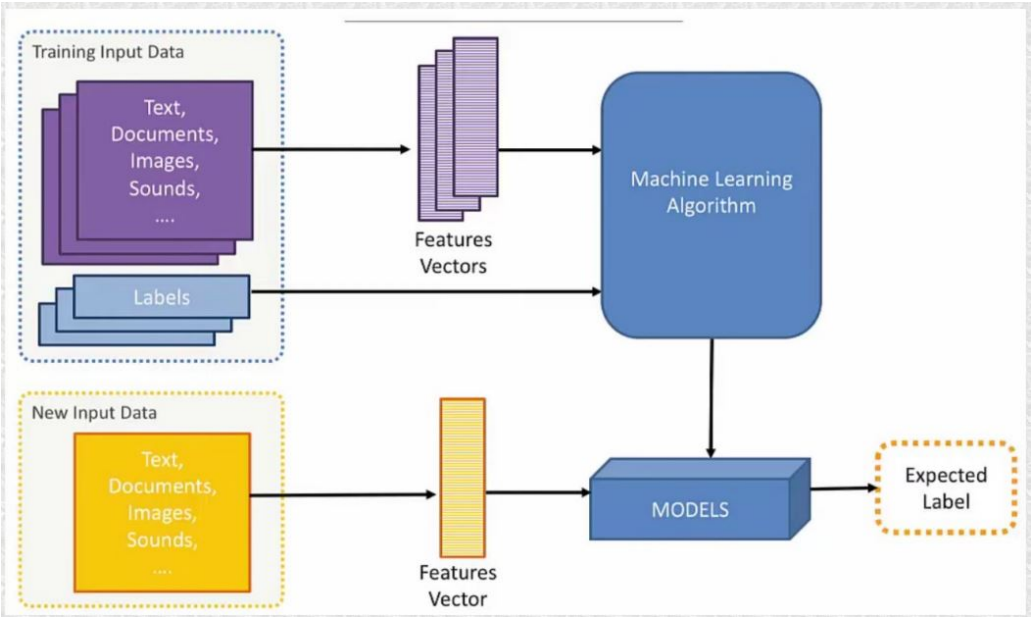

FIGURE 4 – Schema Apprentissage supervisé.

YOLOv8 propose une gamme de modèles pour la détection d'objets, du léger YOLOv8n au puissant YOLOv8x. Adaptés à différentes contraintes, ces modèles répondent aux besoins spécifiques, offrant des solutions pour des ressources limitées (YOLOv8n), un compromis entre compacité et performances (YOLOv8s), et une robustesse accrue pour des scénarios complexes (YOLOv8m). Les versions haut de gamme, YOLOv8l et YOLOv8x, sont conçues pour une qualité de détection maximale dans des tâches exigeantes.

Dans notre cas nous avons utilisé yolov8n car son reseaux de neurones à une dimention plus faible ce qui reduit le cout de l'apprentissage. De plus nous avions une configuration machine limité.

4.2 OpenCV : floutage vidéo

OpenCV (Open Source Computer Vision Library) est une bibliothèque open source spécialisée dans le traitement d'images et de vidéos. Elle offre une large gamme de fonctionnalités pour la vision par ordinateur, la reconnaissance de formes, la segmentation d'images, le suivi d'objets, et bien plus encore.

Dans le contexte du floutage d'images, nous avons utilisé OpenCV pour appliquer des opérations de flou à des images. Le floutage est une technique couramment utilisée pour réduire les détails et la netteté d'une image, souvent dans le but de protéger la confidentialité des informations ou de créer des effets visuels spécifiques.

Pour mettre en œuvre le floutage d'images, nous avons combiné les capacités d'OpenCV avec les prédictions générées par le modèle YOLO (You Only Look Once).

Nous avons utilisé OpenCV pour charger l'image, puis YOLO pour effectuer des prédictions de détection d'objets. Ensuite, nous avons appliqué un flou de Gauss aux boîtes englobantes correspondant aux objets détectés, réduisant ainsi les détails dans ces régions spécifiques de l'image.

Le flou de Gauss est une technique de floutage qui est couramment utilisée pour réduire les détails et la netteté d'une image de manière douce et progressive. Cette méthode utilise une fonction de distribution gaussienne pour calculer le poids attribué à chaque pixel de l'image, en fonction de sa distance par rapport au pixel central. Plus précisément, les pixels situés plus loin du centre reçoivent un poids plus faible, créant ainsi une transition douce entre les pixels adjacents.

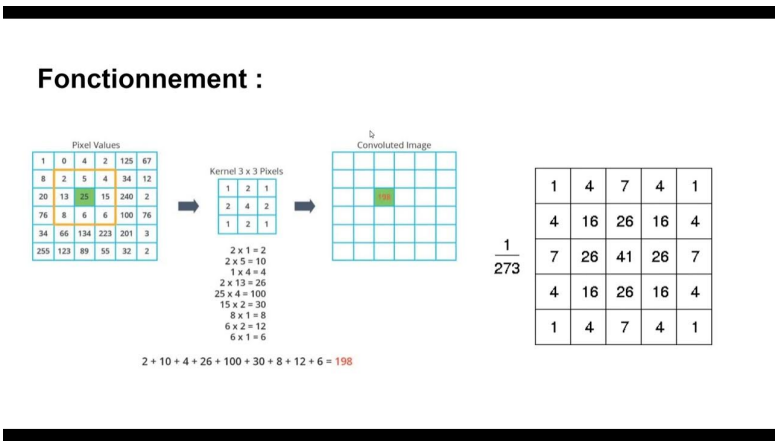

FIGURE 5 – flou de Gauss

5 Difficultés recontrées

Dans le cadre de notre projet, nous avons été confrontés à une problématique majeure lors de l'entraînement de notre IA, plus précisément en ce qui concerne l'utilisation optimale de notre unité de traitement graphique (GPU). Cette situation s'est avérée critique, étant donné que nous disposions de bases de données volumineuses, comprenant parfois plus de 1000 images labellisées. La lenteur des temps d'entraînement entre chaque test était devenue un obstacle majeur, compromettant la réalisation des objectifs dans le délai imparti. Car plus il y a d'images plus le temps d'entrainaînement devient long, c'est pourquoi nous avons choisie de prendre seulement 2 dataset (plaque immatriculation, et visage) pour entrainner notre IA. En effet, nous avons réalisé 300 entrainement et nous avons trouver notre meilleur reseaux au bout du 230 eme, ces entrainement en duré environ 5-6h.

Pour résoudre ce défi, nous avons pris la décision d'installer NVIDIA CUDA sur notre système d'exploitation. NVIDIA CUDA, une plateforme de calcul parallèle, a été la clé pour exploiter pleinement la puissance de calcul de notre GPU. En effet, cette technologie permet une utilisation efficace des capacités de calcul parallèle des GPU NVIDIA, offrant ainsi une accélération significative des tâches liées au calcul intensif, telles que l'entraînement de modèles d'intelligence artificielle.

Parallèlement, afin de garantir une compatibilité optimale avec CUDA, nous avons procédé à l'installation d'une version de PyTorch spécifiquement adaptée à cette configuration. PyTorch, en tant que framework d'apprentissage automatique, bénéficie d'une intégration fluide avec CUDA, permettant une exploitation efficace des ressources matérielles disponibles.

L'intégration réussie de NVIDIA CUDA et de PyTorch compatible avec CUDA a eu un impact significatif sur nos performances d'entraînement. Les temps d'attente entre chaque test ont été considérablement réduits, nous permettant de mener à bien notre projet dans le délai imparti. Cette solution a non seulement optimisé l'utilisation de notre matériel, mais a également renforcé la fiabilité et la rapidité de notre processus d'entraînement, ouvrant ainsi la voie à des itérations plus fréquentes et à une amélioration continue de notre modèle d'IA basé sur YOLO.

6 Conclusion

Ce projet d'intelligence artificielle dédié au floutage vidéo représente une fusion entre des avancées en traitement d'images et l'application de techniques d'apprentissage supervisé. En utilisant la techonologie YOLO qui intègre un modele CNN d'IA, nous avons pu grace a ces differentes technologies concevoir une solution pour la détection et le floutage en temps réel ou non d'objets tels que les visages et les plaques d'immatriculations au sein de flux vidéo (ou image).Cela nous a appris a faire des recherches sur un sujet tout nouveaux pour nous et de réalisé un projet interessant et complexe qui nous a fait evolué dans beaucoup dans la comprehension de l'Inteligence Artificielle et de l'apprentisage de celle-ci.