Initiation YOLO

Analyse de photographies de paysages urbains via la vision par ordinateur

Décrypter la Planification Territoriale. Échelles, Méthodes et Enjeux

Mardi 28 janvier 2025

Nicolas Lépy, Thomas Buhler

ThéMA
UMR 6049 Théoriser & Modéliser pour Aménager

GÉOGRAPHIE-CITÉS
UMR 8504

L’intelligence artificielle dans la vision par ordinateur

ThéMA UMR 6049 Théoriser & Modéliser pour Aménager

GÉOGRAPHIE-CITÉS UMR 8504

Initiation YOLO

openai/**whisper**

DeepL

Reconnaissance automatique de la parole

Vision par ordinateur

Data Science

INTELLIGENCE ARTIFICIELLE

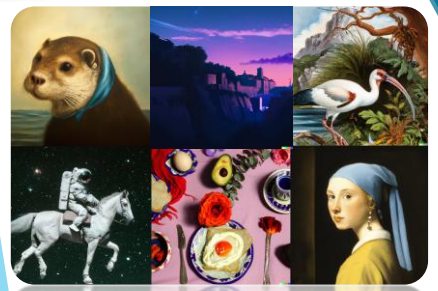
Robotique

IA générative

Logique floue

Systèmes experts

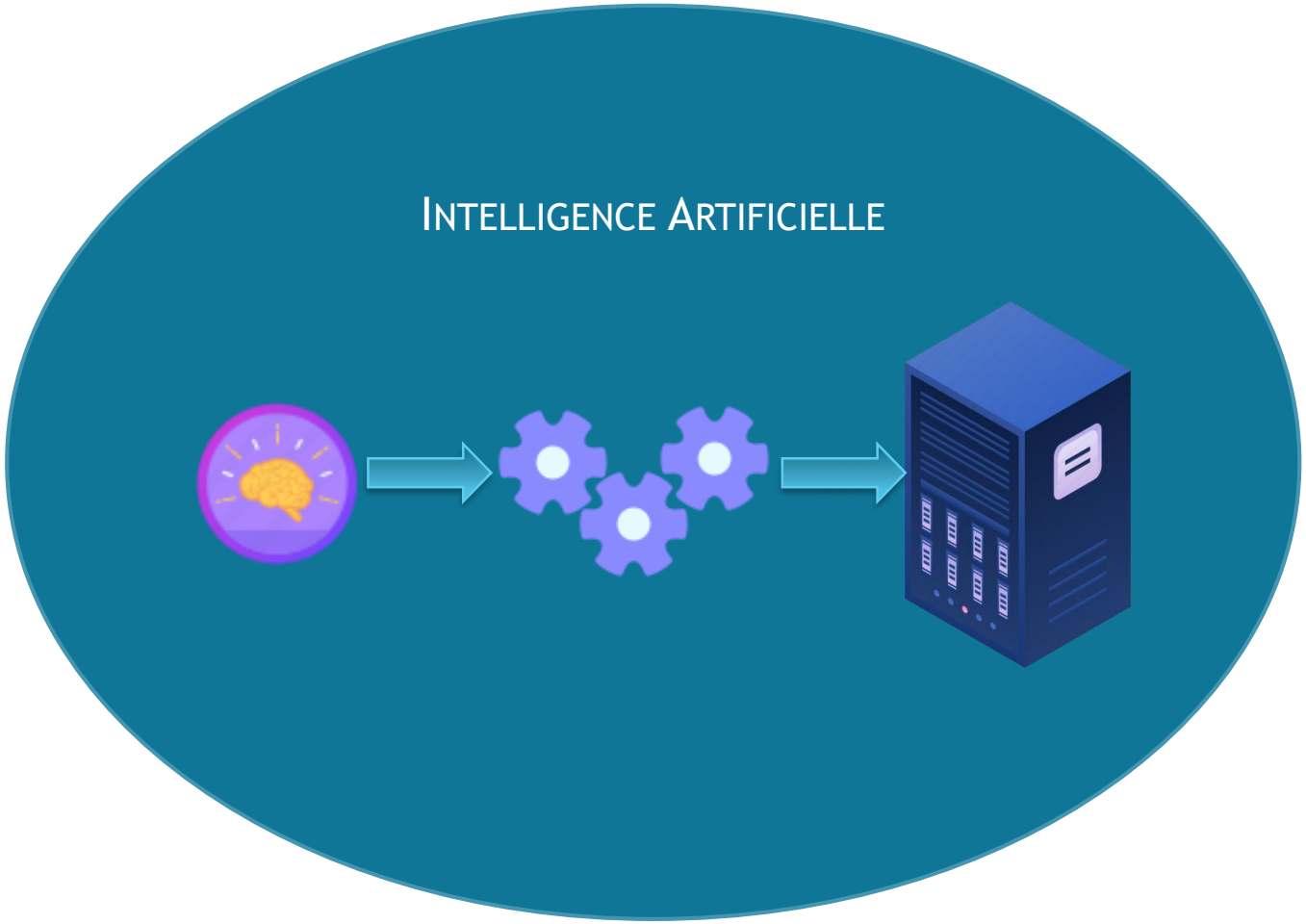
Systèmes multi-agents



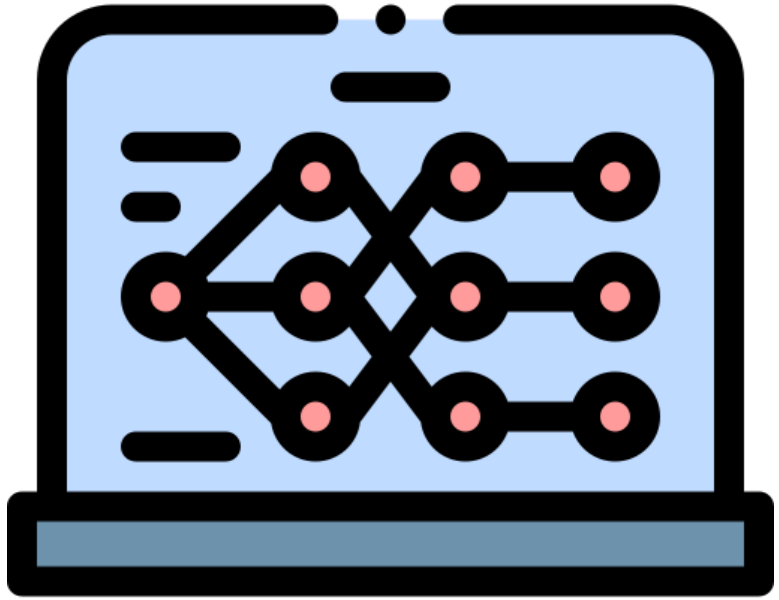
ThéMA UMR 6049 Théoriser & Modéliser pour Aménager

GÉOGRAPHIE-CITÉS UMR 8504

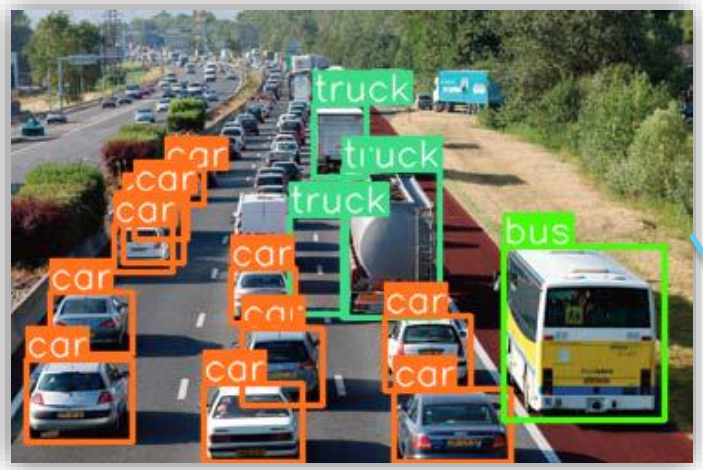
Démystifier l'IA

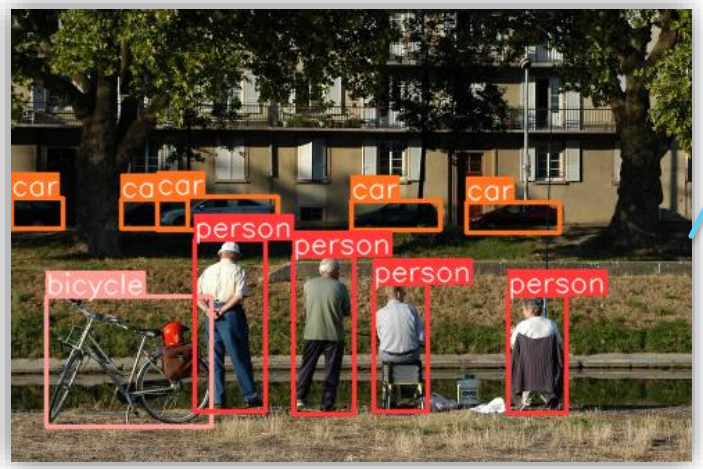


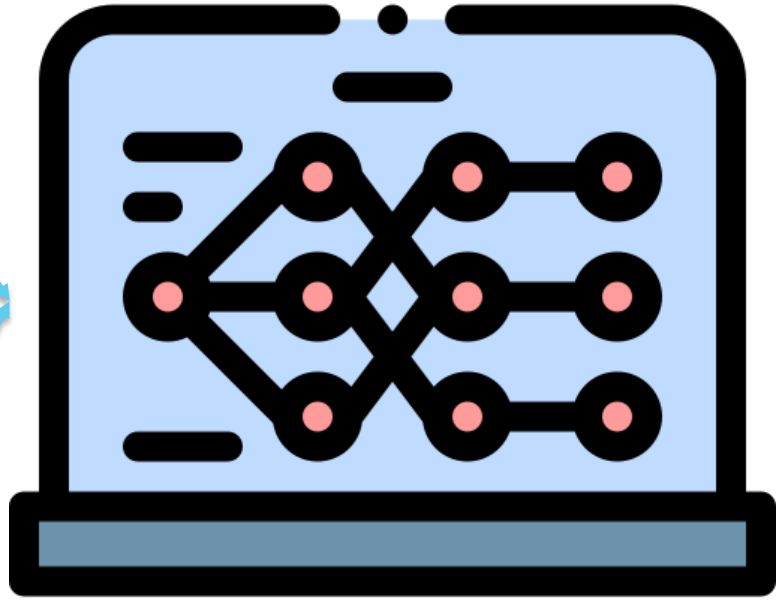
Démystifier l'IA



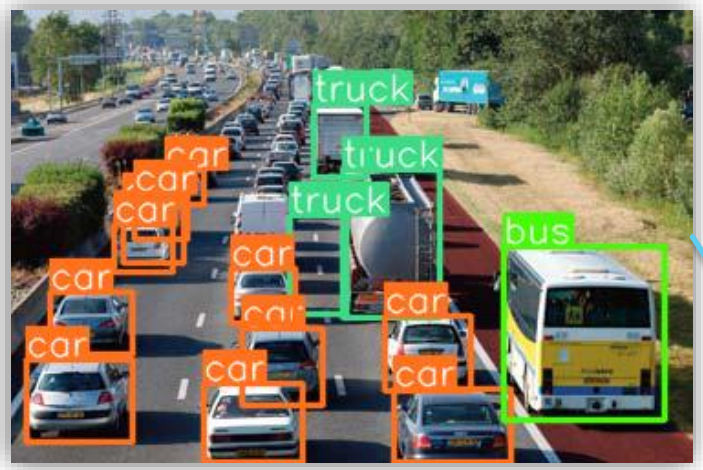
Démystifier l'IA

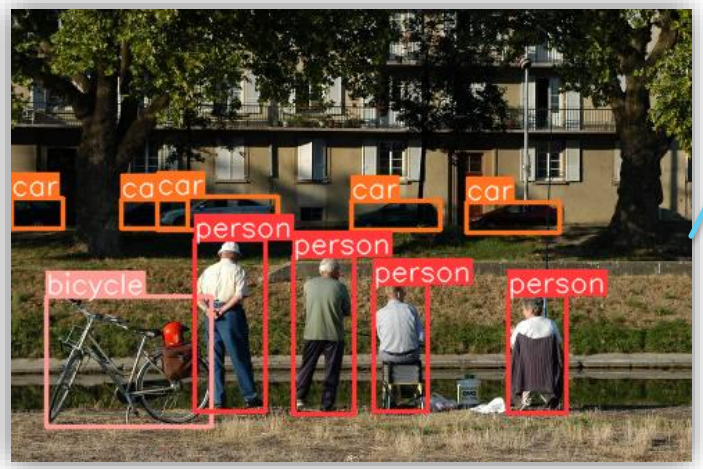


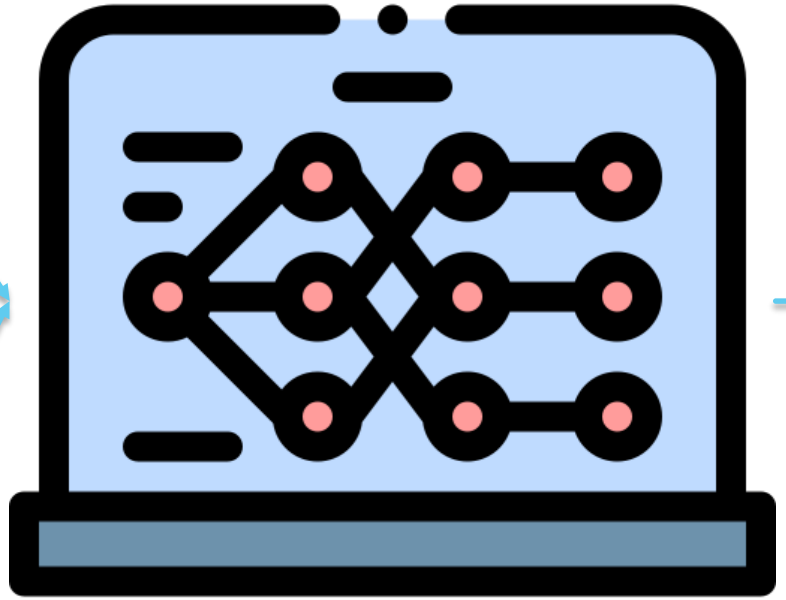




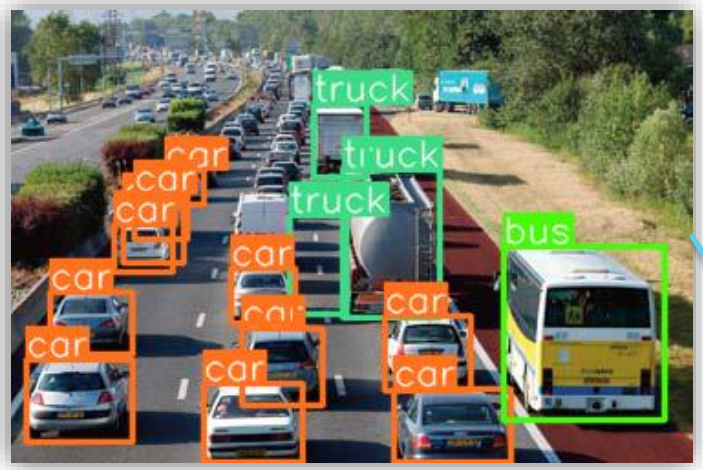
Démystifier l'IA

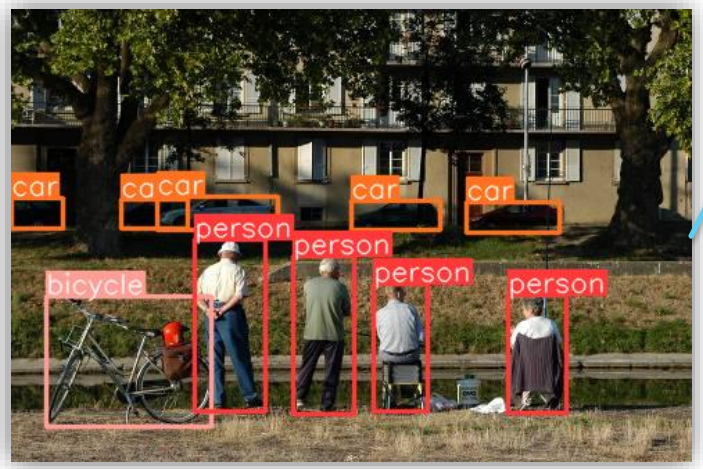


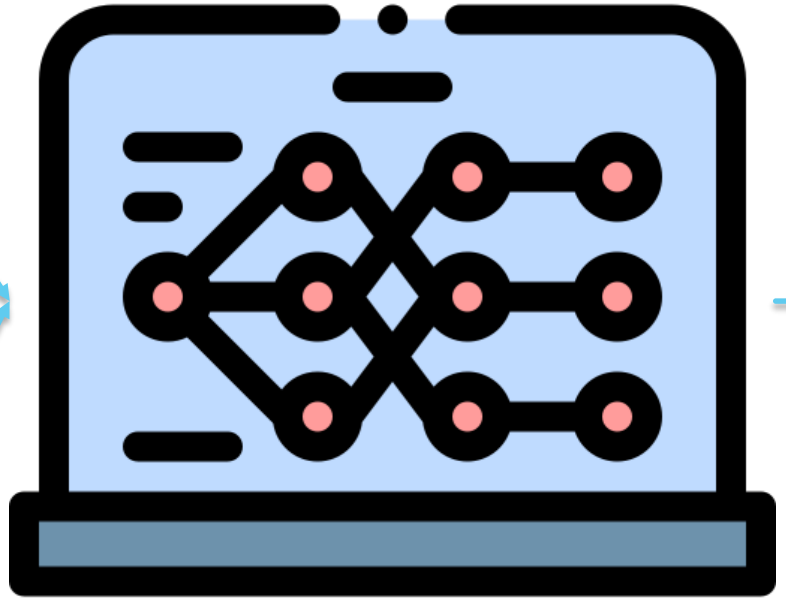


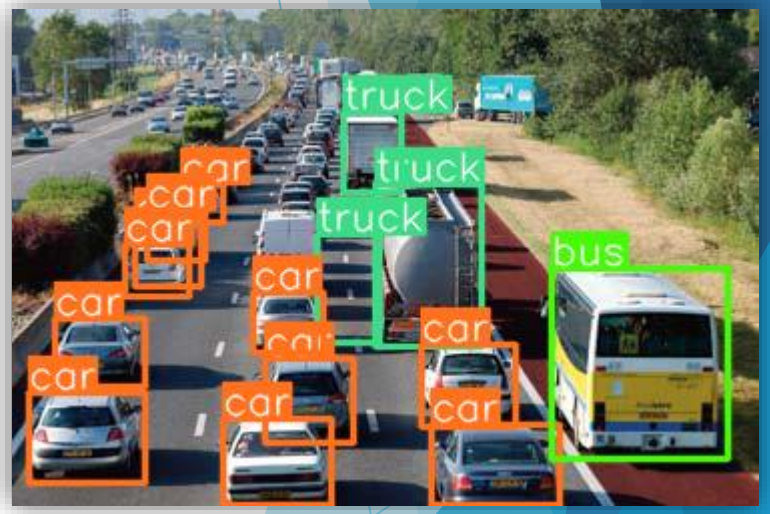


Démystifier l'IA

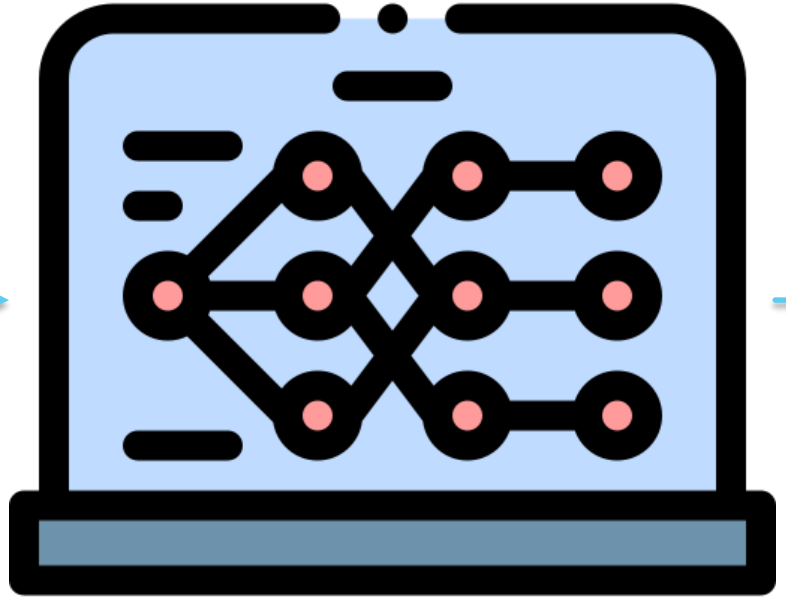


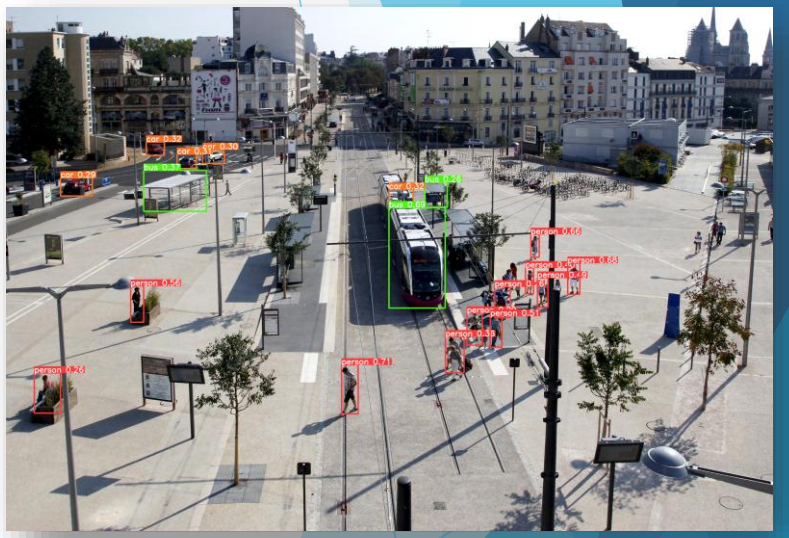






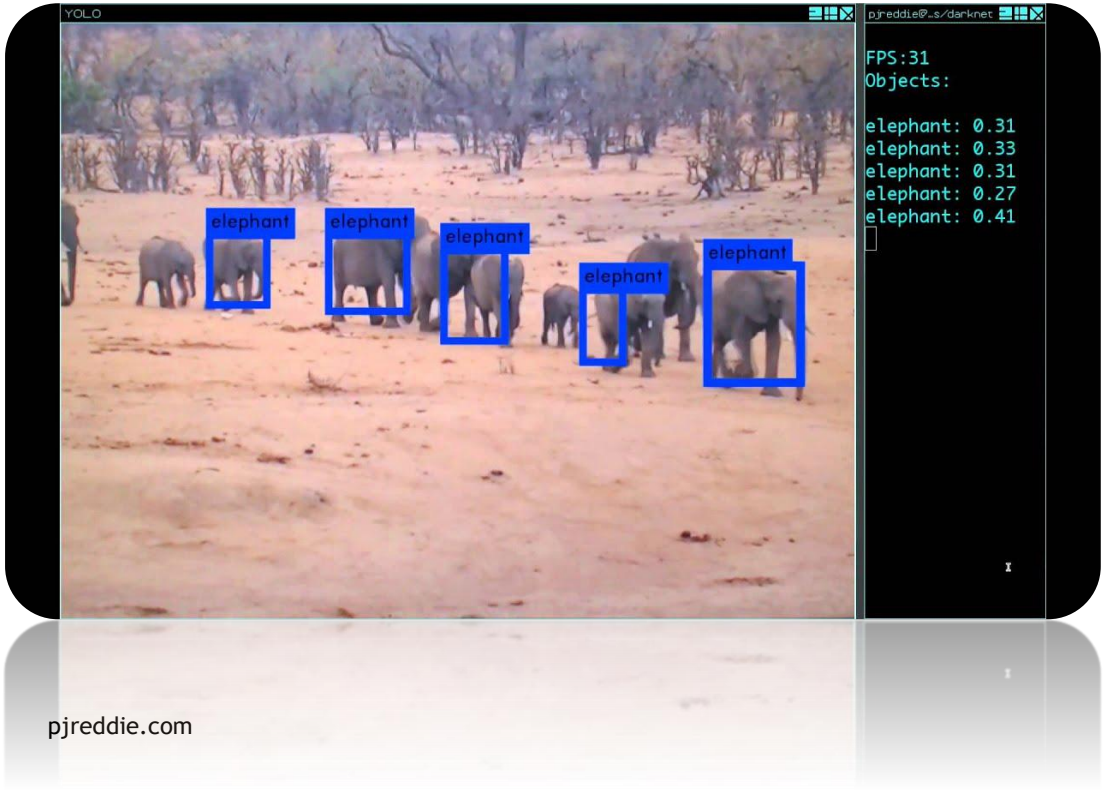
Démystifier l'IA



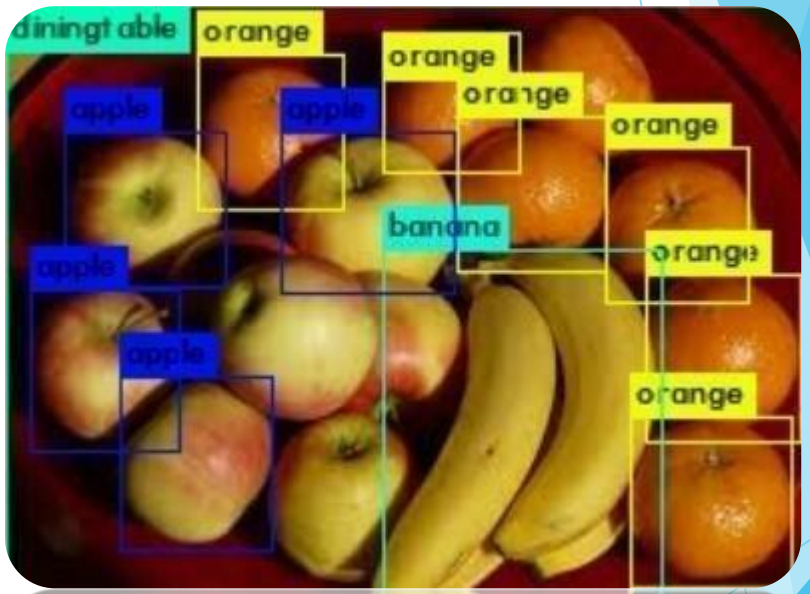


YOLO (You Only Look Once)

- Classification et localisation d'objets



pjreddie.com



A Review of Fruits Image Analysis Using Computer Vision and Deep Learning Techniques

YOLO (You Only Look Once)

- Vélo
- Voiture
- Moto
- Avion
- Bus
- Train
- Camion
- Bateau
- Feu de signalisation
- Borne d'incendie
- Panneau stop
- Parcmètre

YOLO (You Only Look Once)

- Qui a développé YOLO

Joseph Redmon

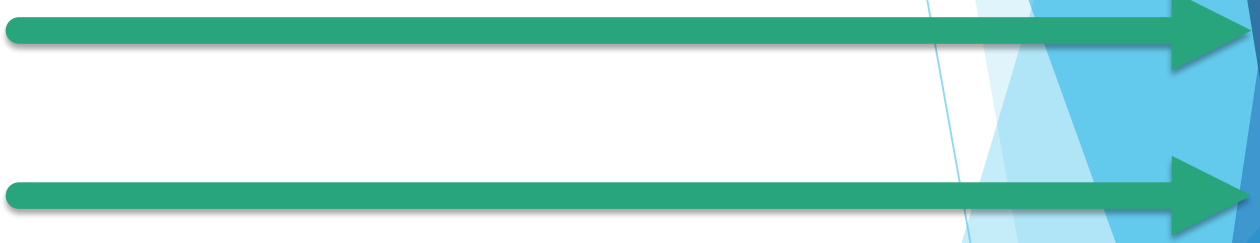
Université de Washington

2015

YOLOv4

2020

Licence Apache 2.0



YOLOv5

2020

Licence AGPL-3.0

Licence commerciale

Licence GPL-3.0

YOLOv7

Chien-Yao Wang, Alexey Bochkovskiy et al

2022

YOLOv11

2024

Limites de YOLO

- Quelques erreurs de classifications
- Petits objets ne sont pas détectés
- Des détections très générales
  - Pour les personnes détectées, pas d'information sur le sexe, l'âge…

Semantic Segmentation (semseg) CSAIL, MIT (2018)

- Segmentation d'images
  - Attribuer à chaque pixel de l'image une catégorie

[PDU Besançon 2015 p.30]

Semantic Understanding of Scenes through ADE20K Dataset. B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso and A. Torralba. International Journal on Computer Vision (IJCV), 2018.

Semantic Segmentation (semseg)

ANTHROPISÉ

VÉGÉTATION

POINT D'EAU

Comment accéder à ces modèles

- Le plus souvent, un modèle d'IA est un fichier avec une extension particulière

- On peut y accéder :

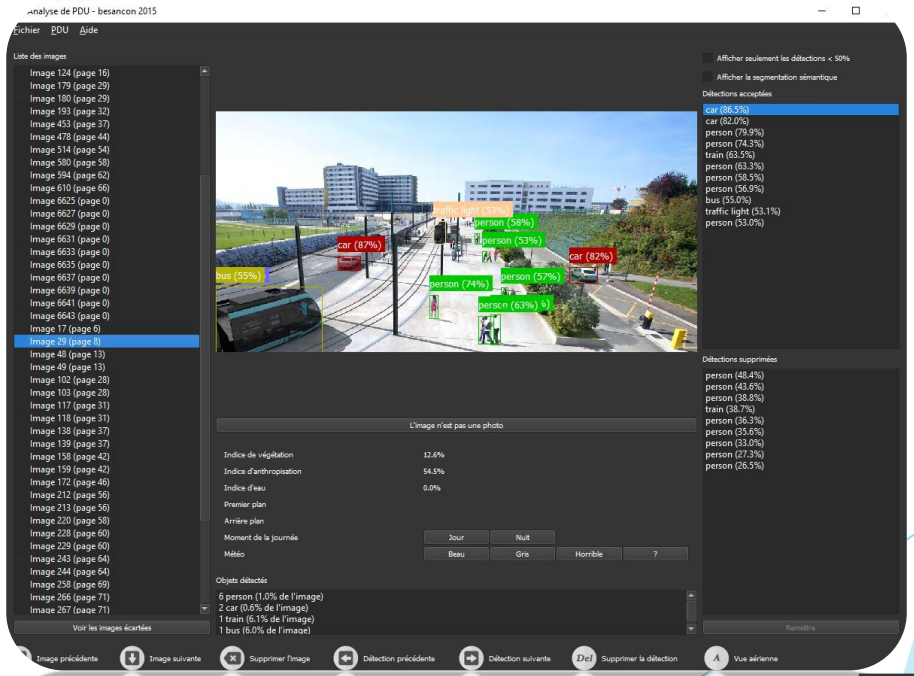
Par un langage de script

Langage Python

```python
from ultralytics import YOLO

model = YOLO(model="yolo11l.pt")
results = model("images\\image_1.png")
results[0].save(filename="export_1.jpg")
print(results[0].pandas())
```

Par une interface graphique

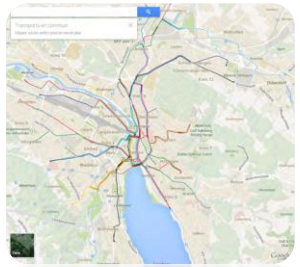


UMR 6049 Théoriser & Modéliser pour Aménager

GÉOGRAPHIE-CITÉS UMR 8504

Reconnaissance d’objets dans les PDU

- Développement d’une interface graphique
  - Faciliter l’utilisation des modèles
  - Extraction automatique des images et détection des photographies

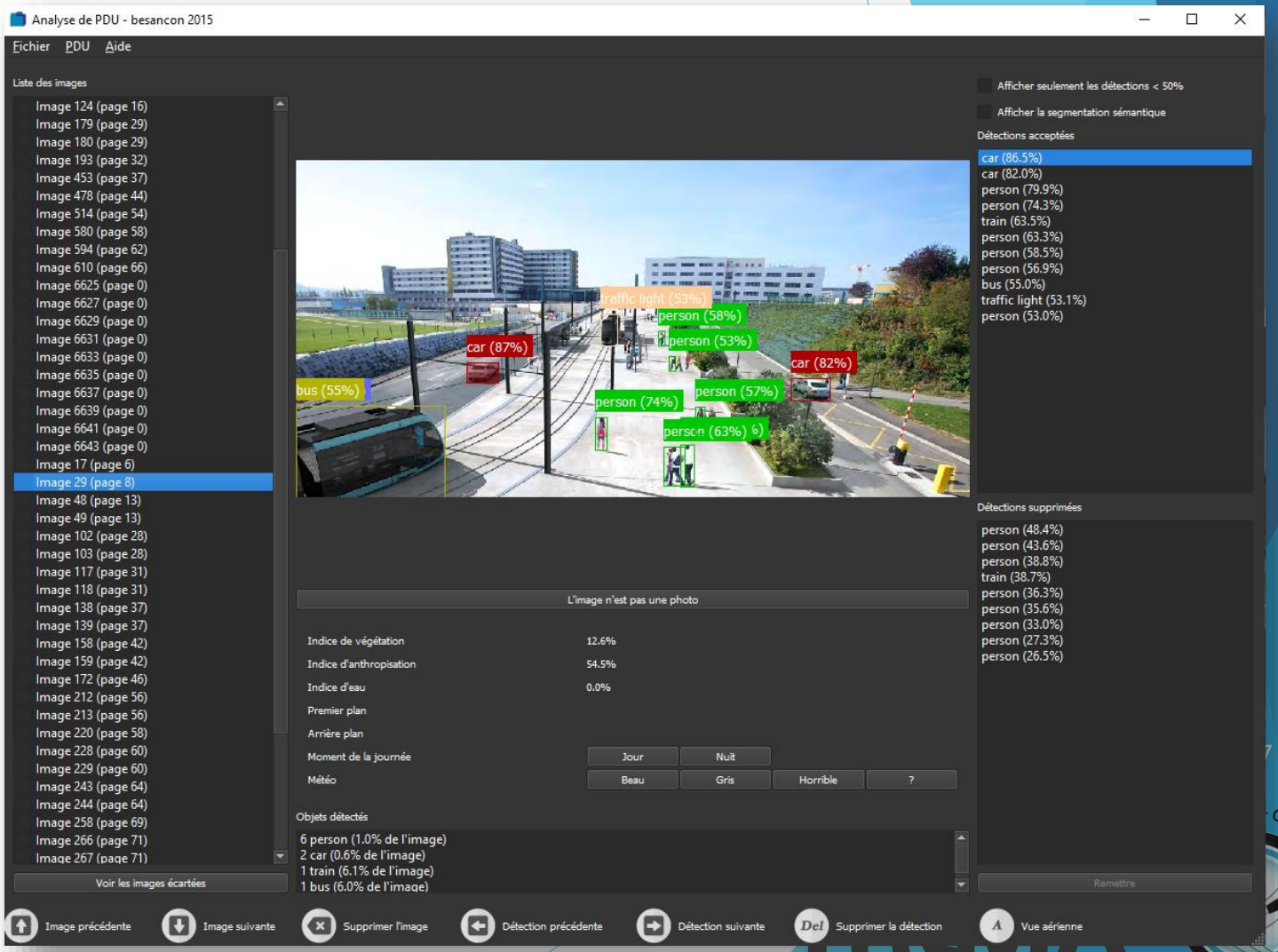


vs

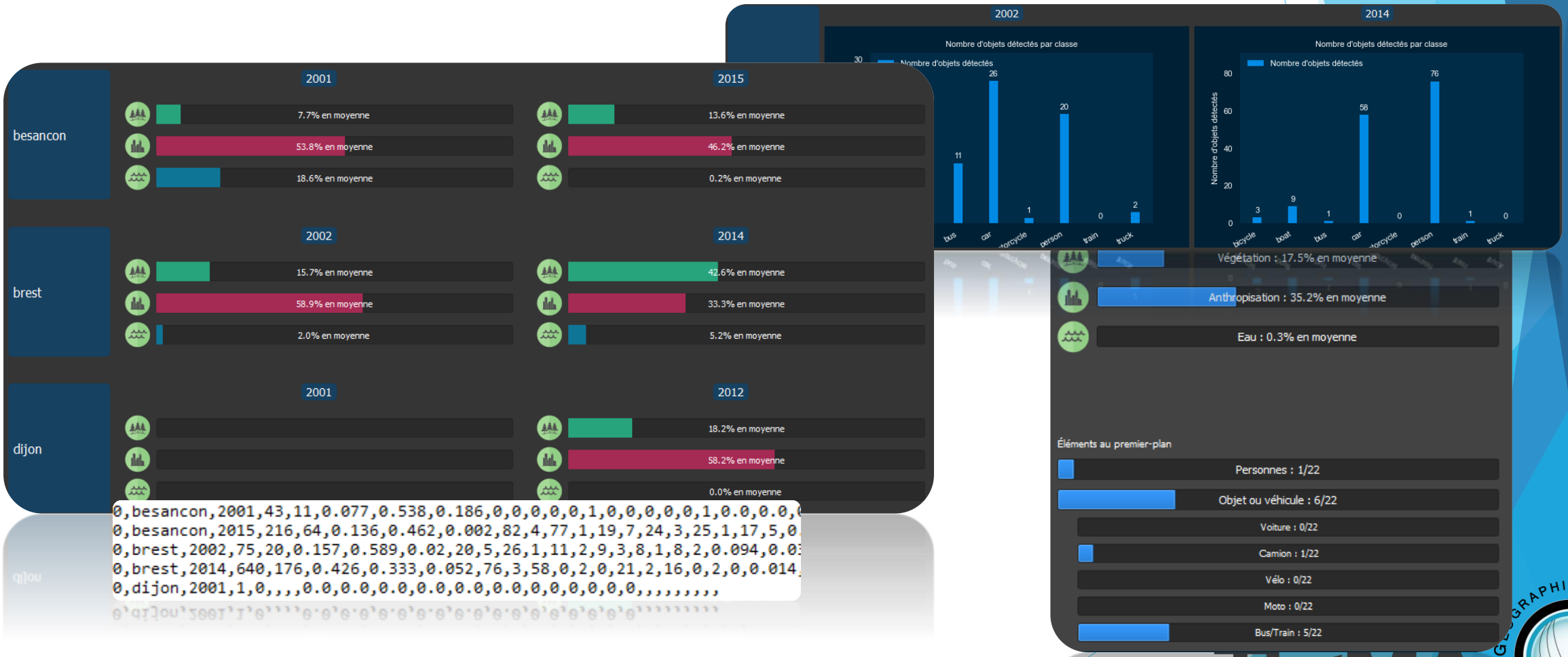
  - Métriques (végétation, …)

Végétation

Anthropisé

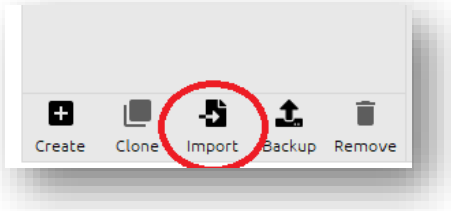


Reconnaissance d’objets dans les PDU

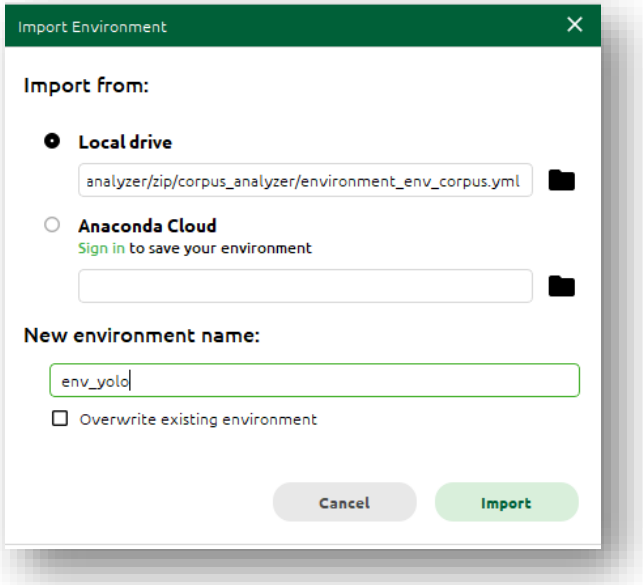


Reconnaissance d’objets dans les PDU

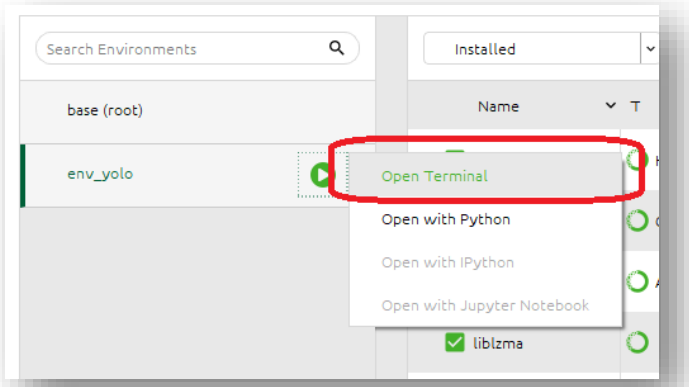
1.



2.



3.



4.

```
(env_yolo) C:\Users\nlepy>d:
(env_yolo) D:\>cd path/to/main.py
(env_yolo) D:\>python main.py
```

-> Uniquement si le logiciel est sur le disque D:

-> Se positionner sur le dossier qui contient le logiciel

-> Exécuter le logiciel

Accéder à YOLO par un langage de script

- Les modèles d'IA sont généralement disponibles sous forme de fichiers
- Des outils sont développés pour utiliser ces modèles sur n'importe quelle image
- Le plus souvent en langage **Python**, mais possible aussi sur **R** …

Accéder à YOLO par un langage de script

- Exécuter YOLO sur une image et exporter l'image du résultat

```python
from ultralytics import YOLO

# == Exécuter YOLO sur une image et exporter l'image du résultat ==

model = YOLO(model="yolo11l.pt")           # Charger le fichier yolo11l.pt
results = model("images\\image_1.png")     # Utiliser YOLO sur une photo sur le disque
results[0].show()                          # Afficher le résultat à l'écran
results[0].save(filename="export_1.jpg")   # Exporter le résultat sur le disque
```

Accéder à YOLO par un langage de script

- Exécuter YOLO sur une image et exporter le résultat au format texte

```
# == Exécuter YOLO sur une image et exporter le résultat au format texte ==

model = YOLO(model="yolo11l.pt")                          # Charger le fichier yolo11l.pt
results = model("images\\image_1.png")                    # Utiliser YOLO sur une photo sur le disque
print(results[0].boxes)                                   # Afficher le résultat sous forme textuelle à l'écran
results[0].save_txt(txt_file="export_1.txt", save_conf=True)   # Exporter la liste des objets détectés au format .TXT (plus brut)
results[0].to_df().to_csv("export_1.csv")                 # Exporter la liste des objets détectés au format .CSV (plus lisible)
```

Accéder à YOLO par un langage de script

- Exécuter YOLO sur toutes les images d'un dossier et exporter le résultat au format texte

```python
# == Exécuter YOLO sur toutes les images d'un dossier et exporter le résultat au format texte ==

model = YOLO(model="yolo11l.pt")            # Charger le fichier yolo11l.pt
results = model( source: "images", conf=0.5)   # On donne le chemin d'un dossier et non d'une image : toutes les images du dossier sont traitées par YOLO.
for result in results:                      # Boucle pour sauvegarder les résultats dans le fichier export.txt
    result.save_txt(txt_file="export.txt", save_conf=True)
```