

# Modélisation et simulation en IA

**Détection des Places de Parking avec un Système de Réservation**

*Filière : Génie Industriel : Intelligence Artificielle et Data Science*

Soutenu par :

**Chegdati Chouaib && Bellmir Yahya**

**Devant le membre du jury : Mr. Masrour Tawfik**

# Plan

1

Introduction

2

Objectifs

3

SVM Model

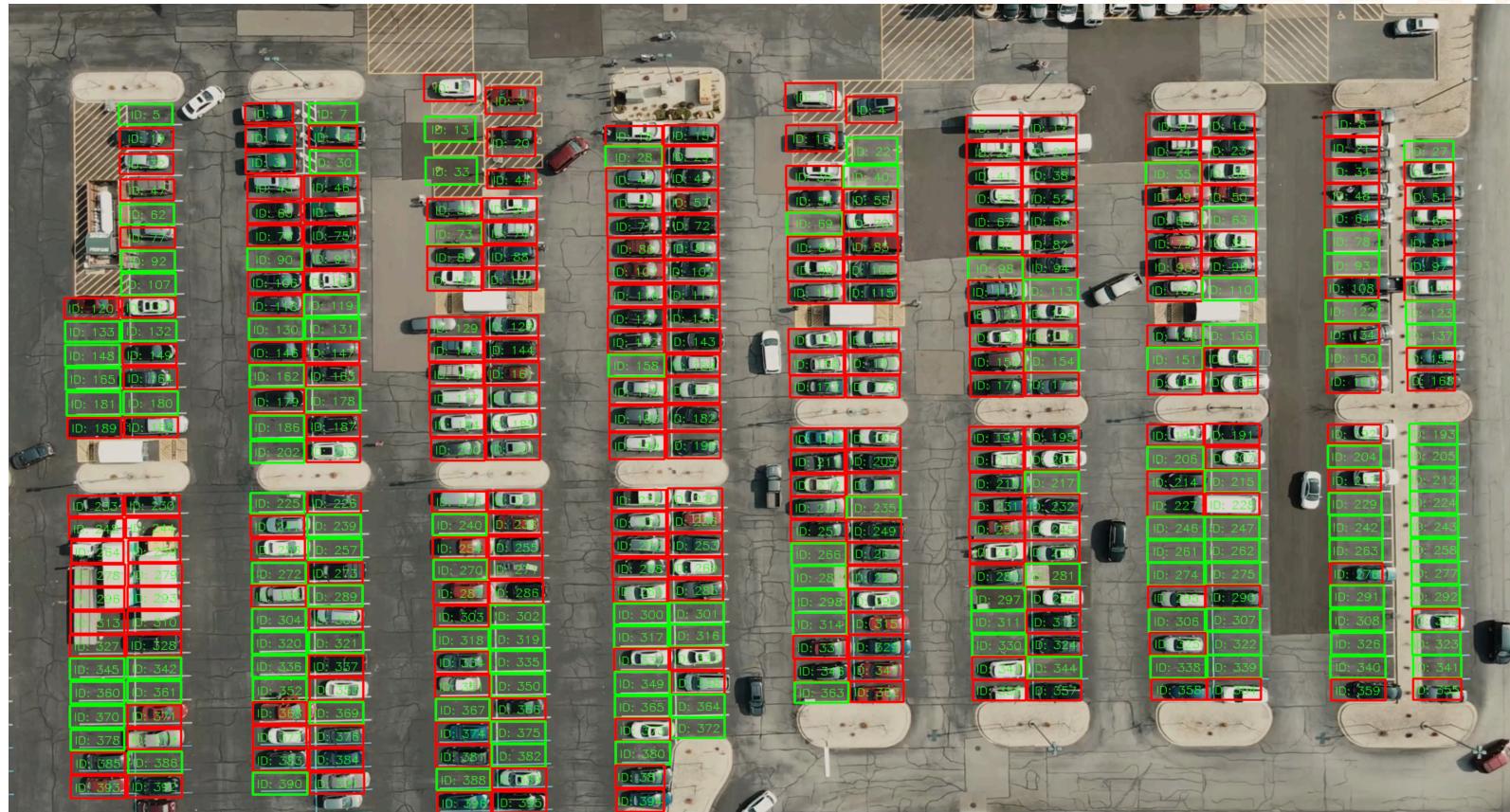
4

Yolov8 Model

5

Conclusion et perspectives

## Introduction

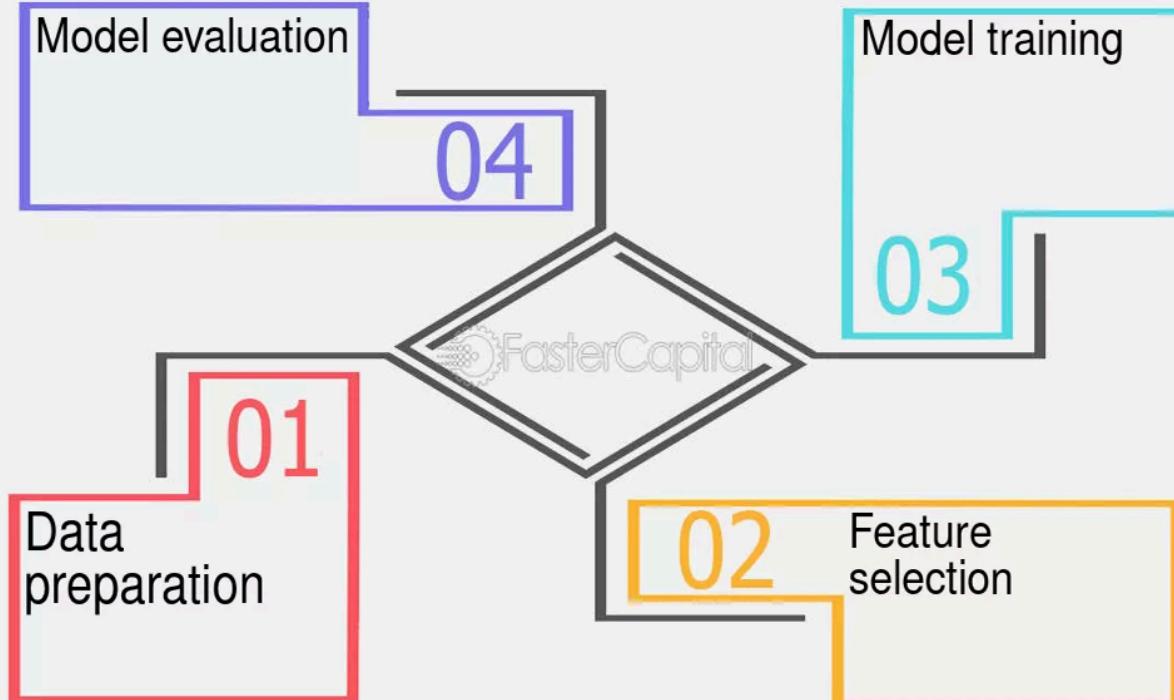


## Objectifs :

- Détection intelligente des places de parking (**SVM Model**)
- Système de détection d'objets pour vidéos et images variées (**YOLO**)
- Réservation des places de parking .
- Création d'une interface interactive .
- Évaluation des performances .

## SVM model :

### Training an SVM Model for Credit Risk Classification

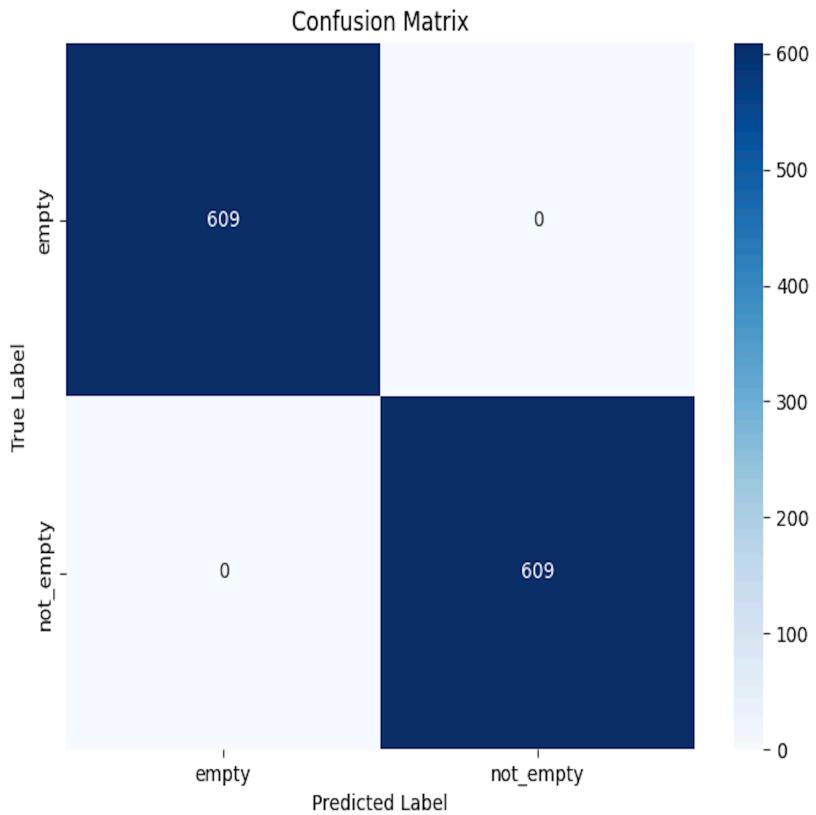


SVM model :

## ParkingSpaceRecognition.py :

- **Importer et prétraiter les images de parking** : pour les transformer en données exploitables par le modèle.
- **Construire et optimiser un modèle SVM** : pour la classification des places de parking.
- **Évaluer les performances** : du modèle avec une matrice de confusion.

## SVM model :



```
/Users/chouaibchegdaci/PycharmProjects/Parking_space_recognition/parking_space_recognition.py  
100.0% of samples were correctly classified
```

SVM model :

Util.py :

- Classification des places de parking (empty\_or\_not)
- Extraction des emplacements des places de parking (get\_parking\_spots\_bboxes)

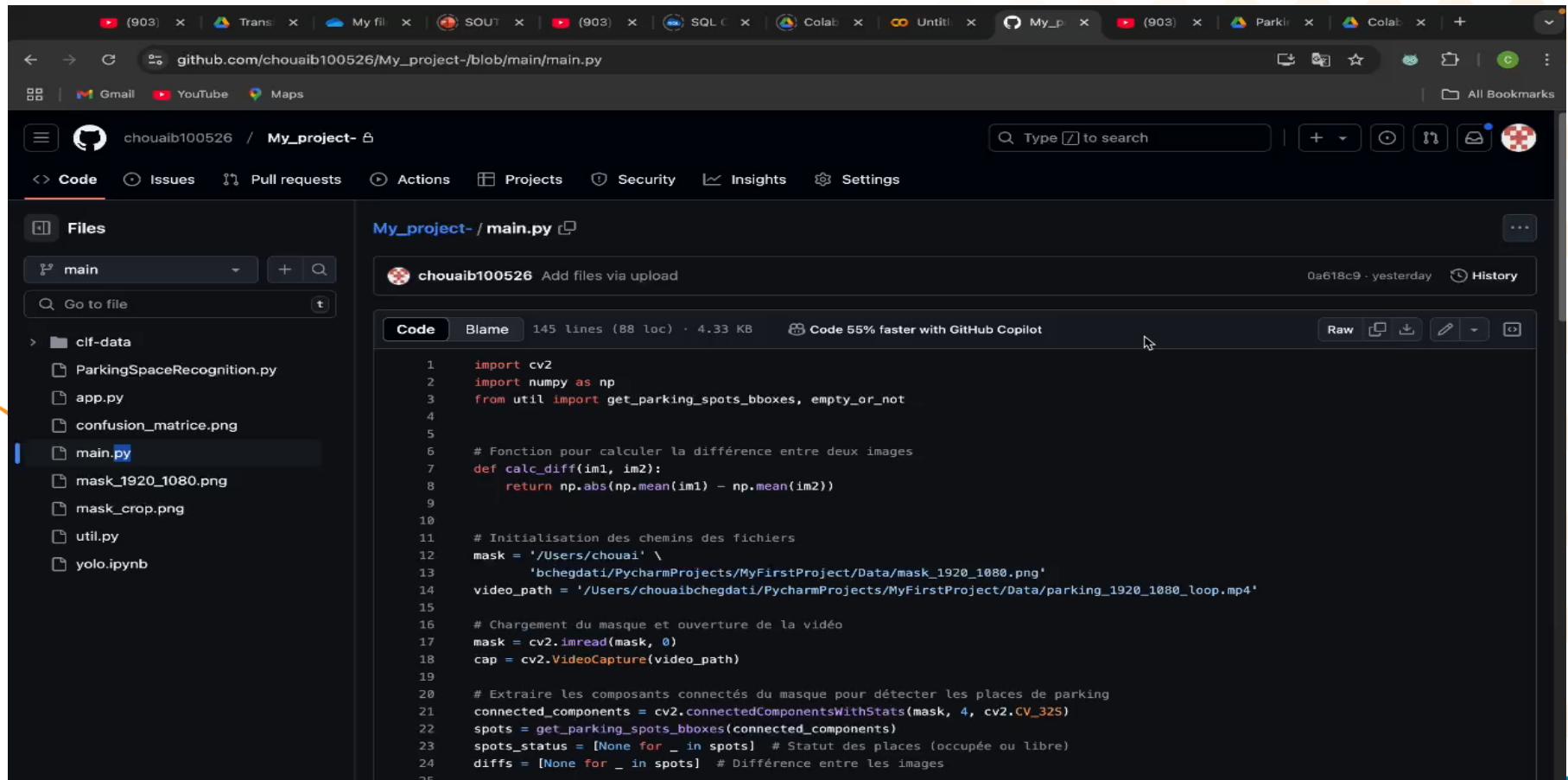
SVM model :

main.py :

- Détection des places de parking
- Suivi de l'occupation des places de parking
- Réservation des places de parking
- Affichage des résultats

## SVM model :

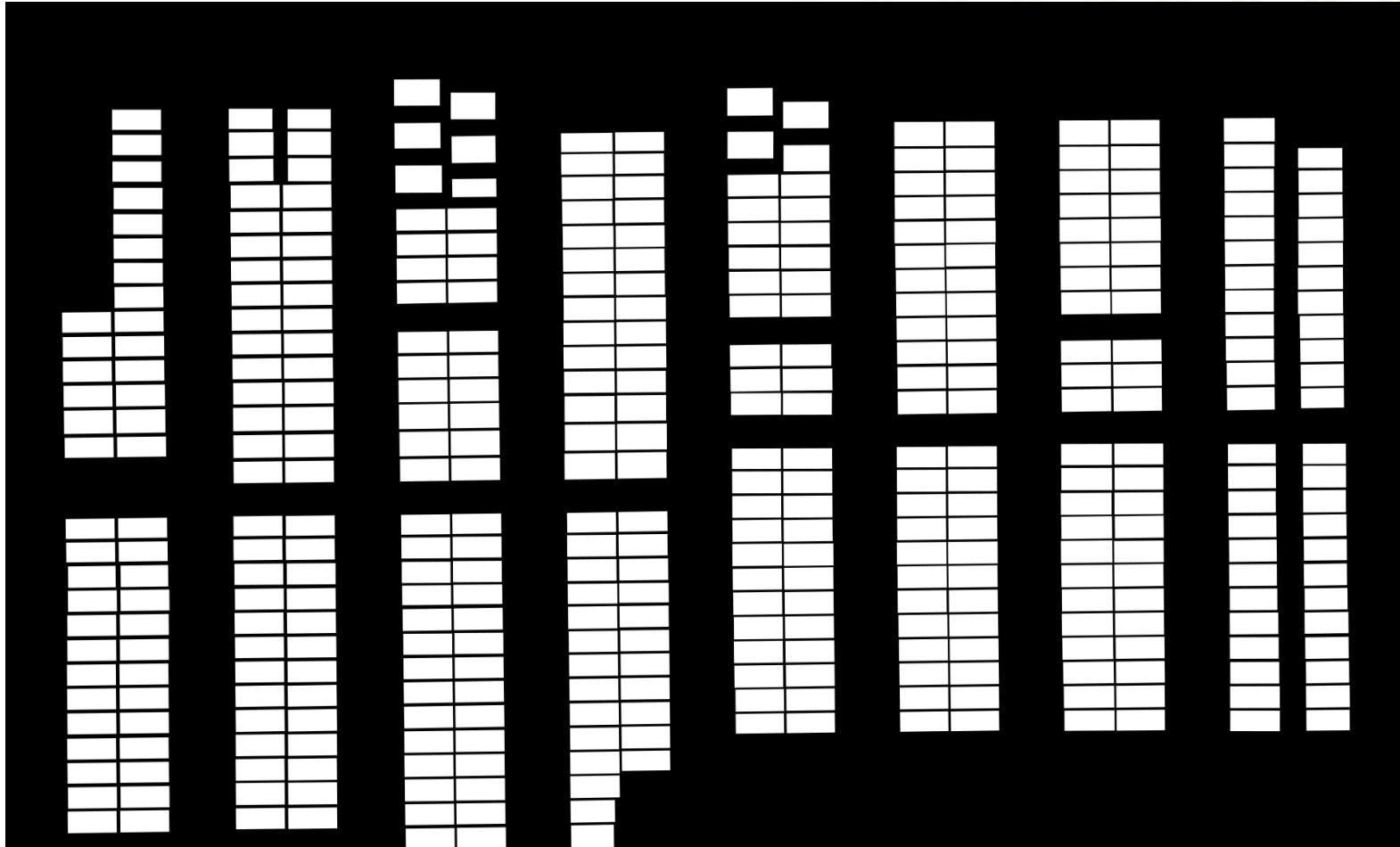
## App.py :



The screenshot shows a GitHub repository named "My\_project-". The "main.py" file is selected in the sidebar. The code in "main.py" is as follows:

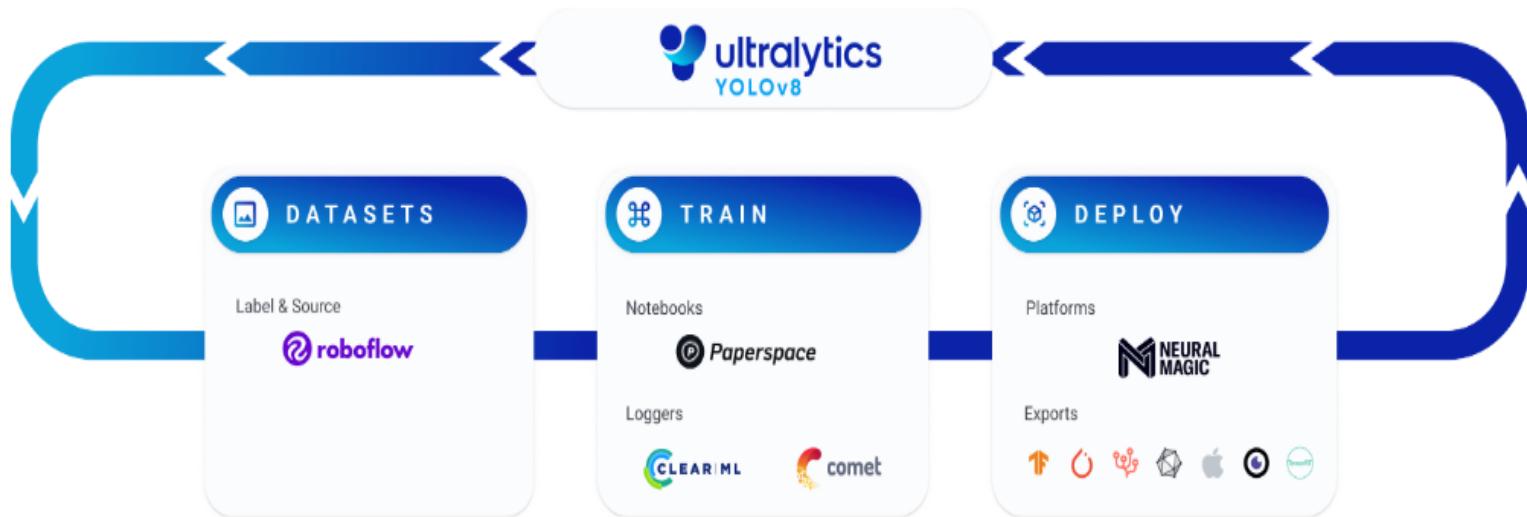
```
1 import cv2
2 import numpy as np
3 from util import get_parking_spots_bboxes, empty_or_not
4
5
6 # Fonction pour calculer la différence entre deux images
7 def calc_diff(im1, im2):
8     return np.abs(np.mean(im1) - np.mean(im2))
9
10
11 # Initialisation des chemins des fichiers
12 mask = '/Users/chouai' \
13     'bchegdatti/PycharmProjects/MyFirstProject/Data/mask_1920_1080.png'
14 video_path = '/Users/chouaibchegdatti/PycharmProjects/MyFirstProject/Data/parking_1920_1080_loop.mp4'
15
16 # Chargement du masque et ouverture de la vidéo
17 mask = cv2.imread(mask, 0)
18 cap = cv2.VideoCapture(video_path)
19
20 # Extraire les composants connectés du masque pour détecter les places de parking
21 connected_components = cv2.connectedComponentsWithStats(mask, 4, cv2.CV_32S)
22 spots = get_parking_spots_bboxes(connected_components)
23 spots_status = [None for _ in spots] # Statut des places (occupée ou libre)
24 diffs = [None for _ in spots] # Différence entre les images
25
```

## SVM model vs Yolo :



## Yolov8 :

You only look once (YOLO) est un système de détection d'objets en temps réel de pointe. Sur un Pascal Titan X, il traite les images à 30 FPS et affiche un mAP de 57,9% sur COCO test-dev. Elle est réputée pour sa rapidité et sa précision



Yolov8 :

## Collecte des données :

Pour de meilleurs résultats, nous avons utilisé un ensemble de données contenant [1733 images](#) déjà étiquetées à partir de [Roboflow](#) et nous avons étiqueté environ [150](#) images à partir d'un ensemble de données que nous avons trouvé sur [Kaggle](#).

**1662** dans [Train set](#)

**200** dans [Valid Set](#)

**21** dans [Test Set](#)



kaggle

Yolov8 :

## Préparation des données :

Les techniques de prétraitement des données que nous avons effectuées sont :

**Auto-Orient : Applied**

**Resize : Stretch to 640x640**



Les techniques d'augmentation des données que nous avons effectuées sont :

**Outputs per training example : 3**

**90° Rotate : Clockwise, Counter-Clockwise**

**Rotation : Between -15° and +15**

Yolov8 :

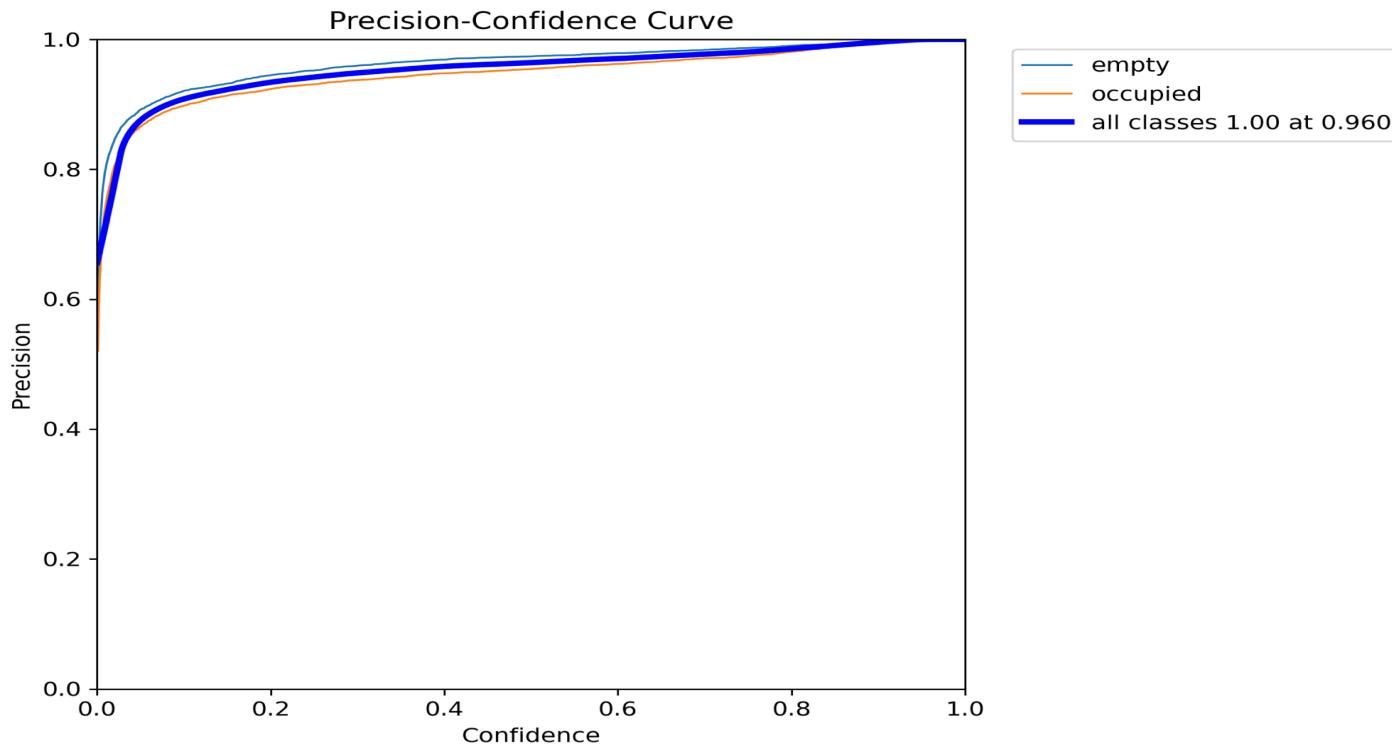
## Entrainement du modèle:

```
1 #import a pretrained YOLO model
2 import os
3 from ultralytics import YOLO
4 from IPython.display import display, Image
5 model = YOLO('yolov8n.pt')
6
7
8 ROOT_DIR = "/content/drive/MyDrive/Parking detection" # Define ROOT_DIR
9 results = model.train(data=os.path.join(ROOT_DIR, "data.yaml"), epochs
   =100) #train the model.
10 results = model.val() #evaluate model performance on the validation
   set
```



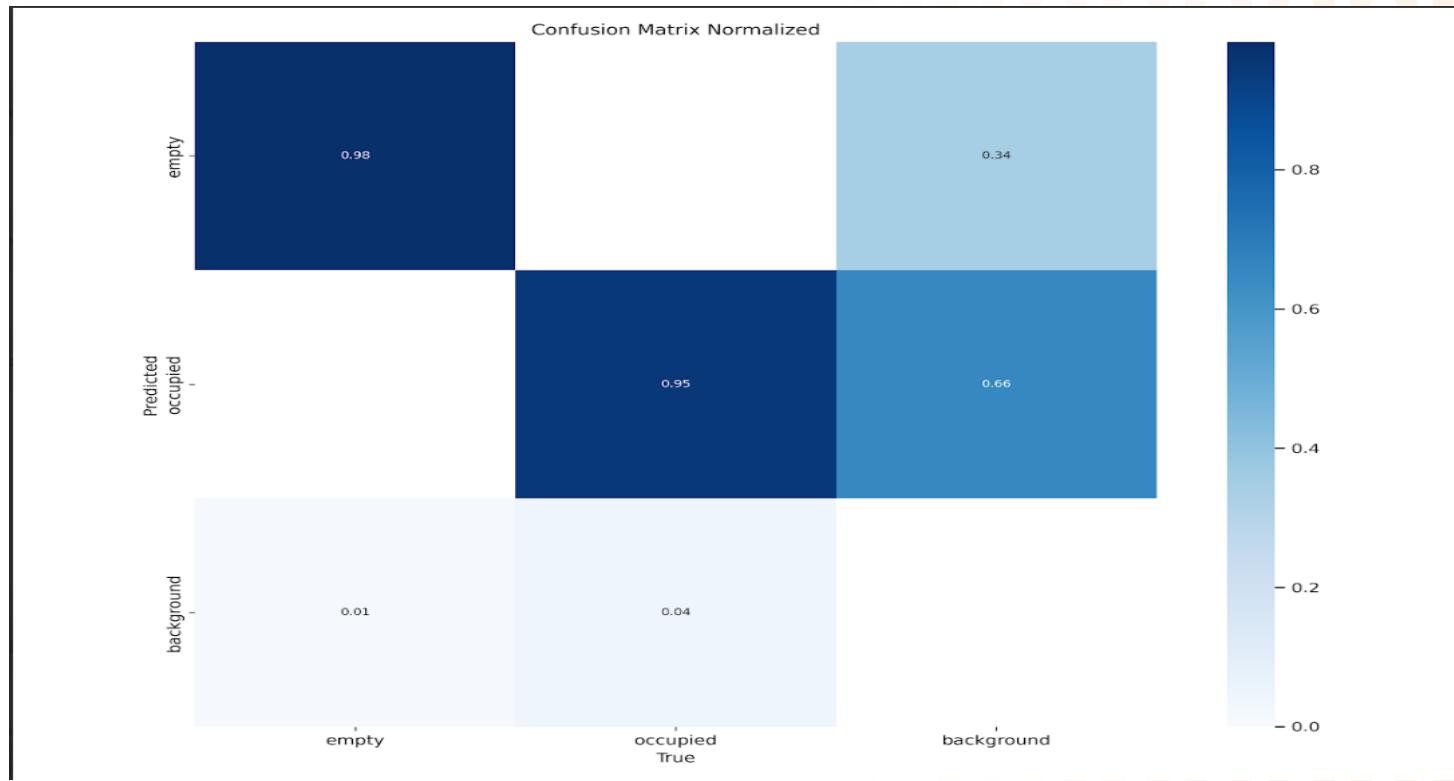
Yolov8 :

# Résultats



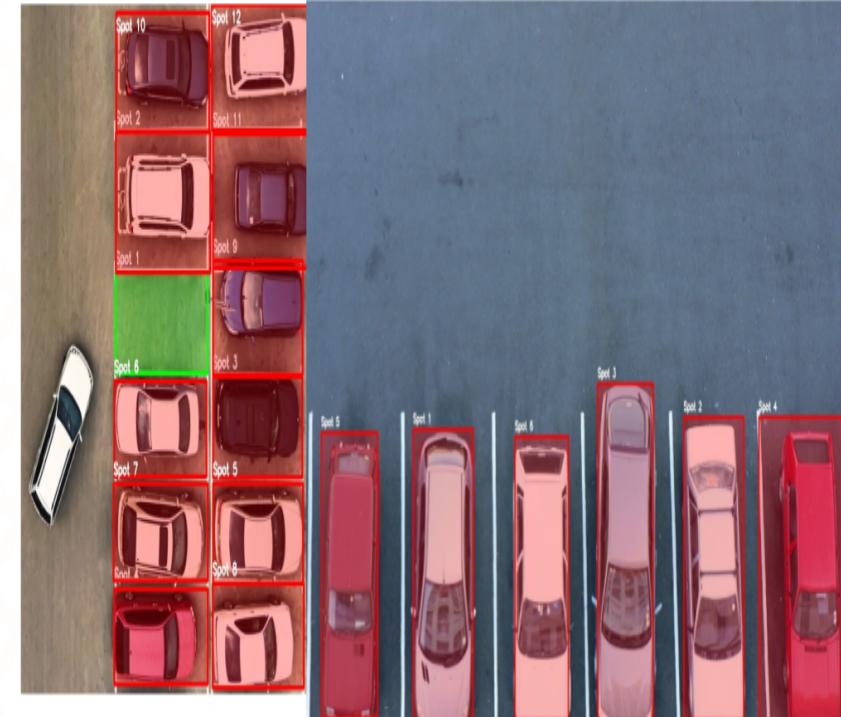
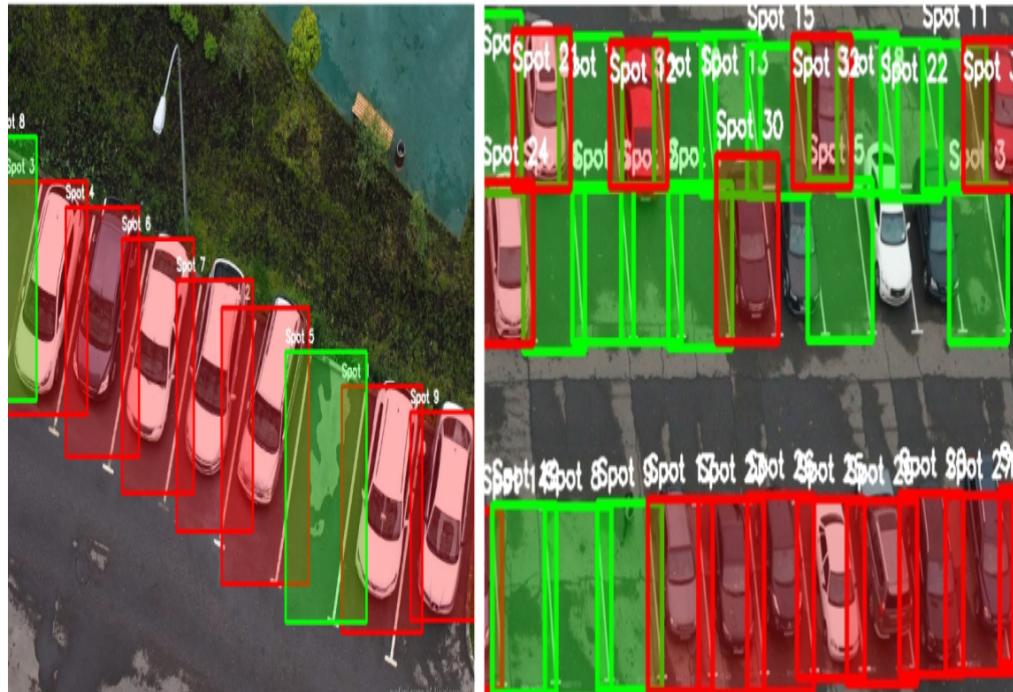
Yolov8 :

# Résultats



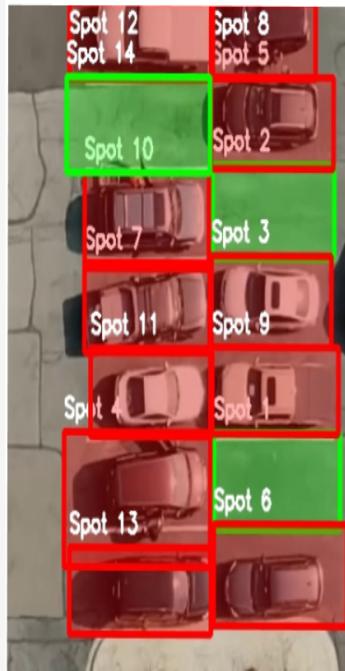
Yolov8 :

## Tests(Images dans Dataset)



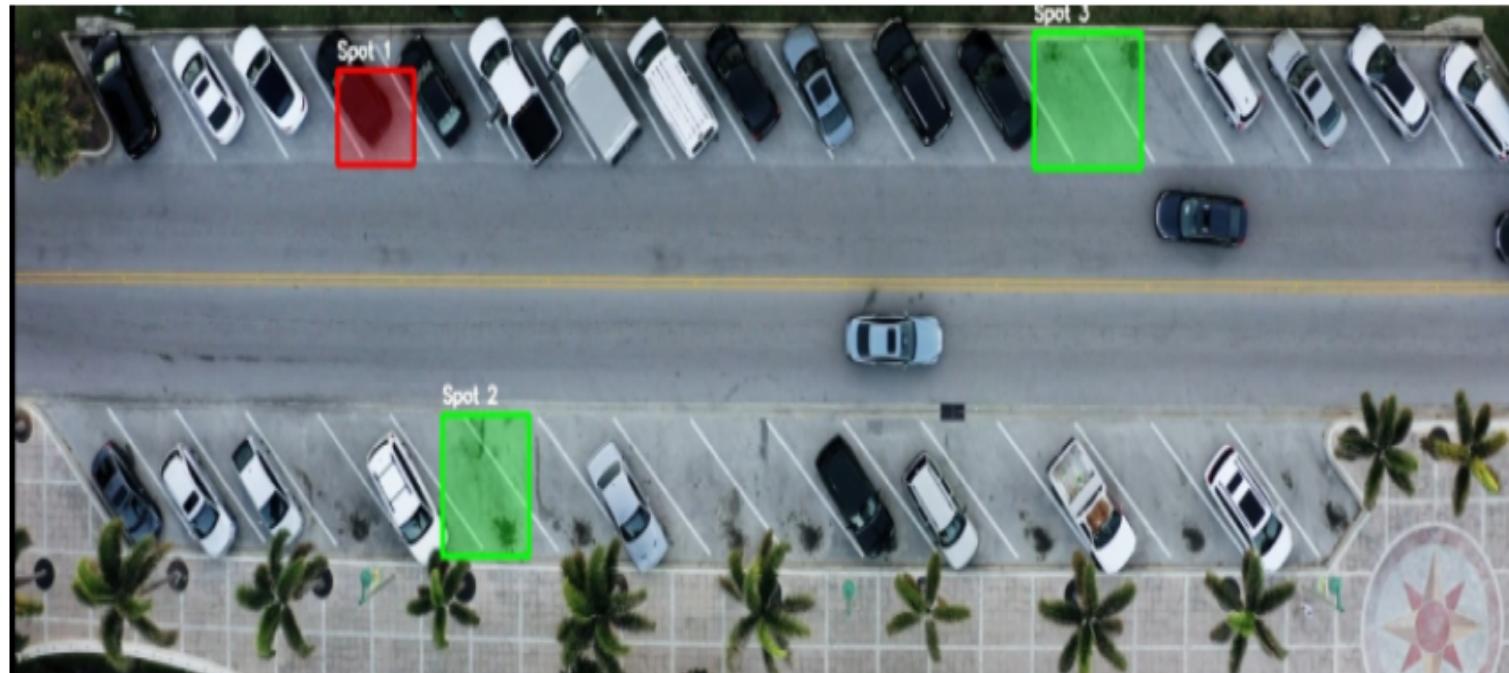
Yolov8 :

## Tests(Images hors Dataset)



Yolov8 :

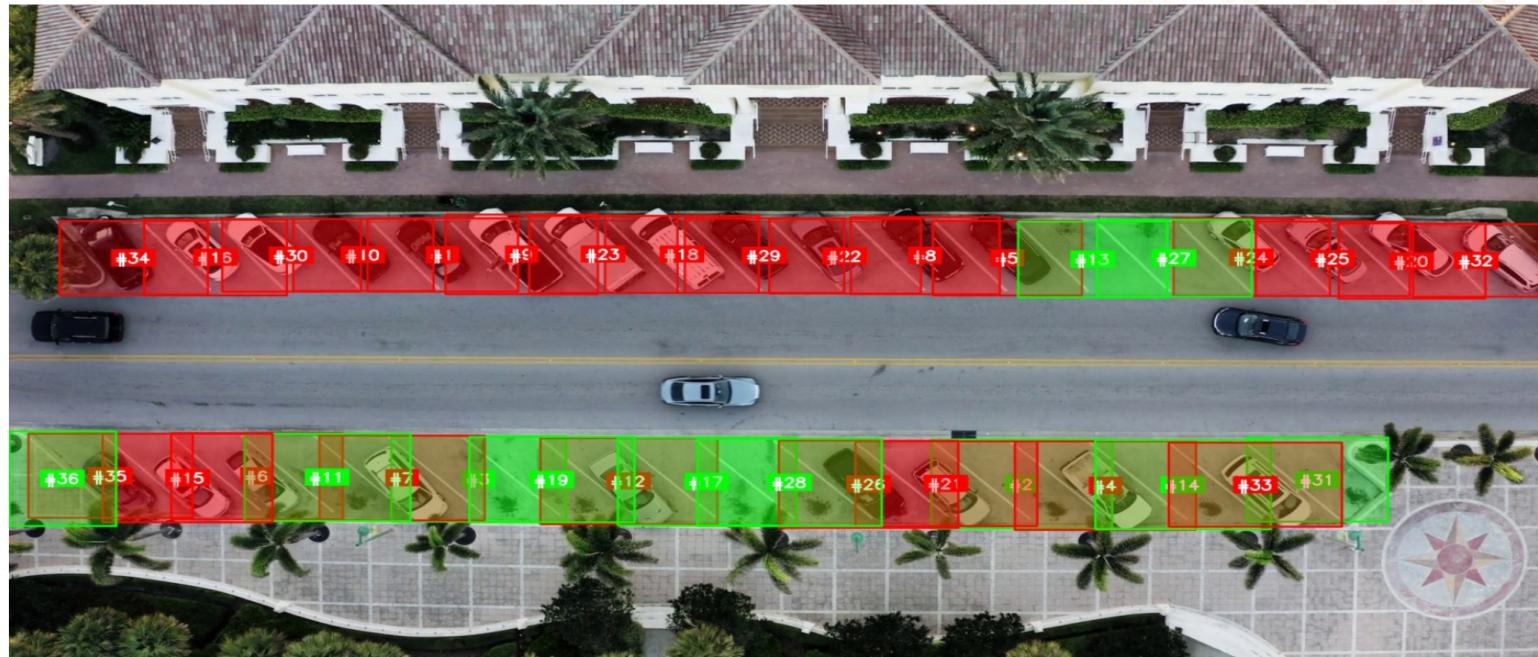
## Tests(Images hors Dataset)



Solution: Inclure ce parking dans notre Dataset et retrainer le modèle

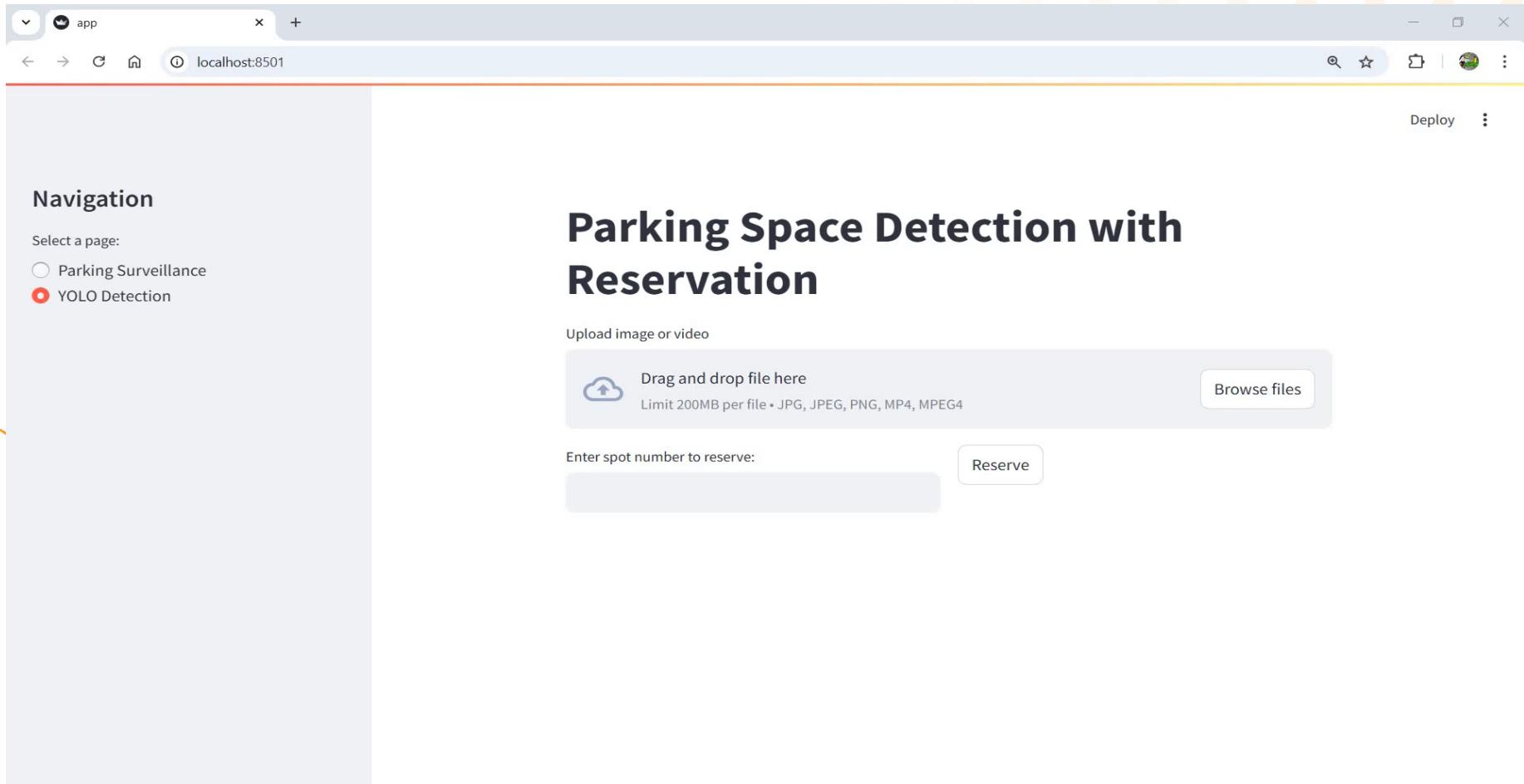
Yolov8 :

## Tests(Images hors Dataset)



Résultat après Solution

Yolov8 :



The screenshot shows a web application interface for parking space detection and reservation. The top navigation bar includes icons for back, forward, search, and refresh, along with a URL field showing "localhost:8501". On the right side of the header, there are "Deploy" and three-dot menu icons.

**Navigation**

Select a page:

- Parking Surveillance
- YOLO Detection

**Parking Space Detection with Reservation**

Upload image or video

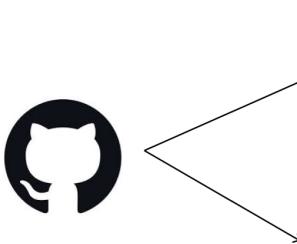
Drag and drop file here  
Limit 200MB per file • JPG, JPEG, PNG, MP4, MPEG4

Browse files

Enter spot number to reserve:

Reserve

## Déploiement :



Streamlit

Read the Docs

Yasouimo cd		618f20a · 2 days ago	46 Commits
	.devcontainer	Added Dev Container Folder	2 days ago
	.vscode	commit	last week
	Colab	commit	2 days ago
	Readthedocs	cd	2 days ago
	__pycache__	Forth commit	3 weeks ago
	clf-data	first commit	last month
	data	commit	2 days ago
	mask	commit	2 days ago
	models	commit	2 days ago
	src	Update yolo_page.py	2 days ago
	.readthedocs.yaml	commit	2 days ago
	README.md	Update README.md	2 days ago
	packages.txt	5th commit	3 weeks ago
	requirementProject.txt	commit	last week
	requirements.txt	ss	5 days ago

## Conclusion :

- **Atteinte des objectifs du projet :** Nous avons réussi à développer un système de reconnaissance des places de parking avec réservation, intégrant efficacement l'intelligence artificielle pour optimiser la gestion des places et l'expérience utilisateur.
- **Apprentissage des méthodes de simulation en IA et initiation :** Ce projet nous a permis d'acquérir une compréhension des méthodes de simulation en IA, de l'entraînement des modèles à l'optimisation des résultats, même en l'absence de données complètes.
- **Approfondissement des pratiques professionnelles :** Nous avons également appris les bonnes pratiques de travail en IA, notamment l'importance de la collecte de données, de l'entraînement des modèles, et du déploiement des systèmes pour garantir leur efficacité et leur scalabilité.

