

# Système intelligent de détection et de comptage d'objets en temps réel (ex : personnes, véhicules, produits en rayon)

## 1. Objectif du projet

Développer une application capable de **détecter, localiser et compter des objets d'une ou plusieurs catégories** en temps réel à partir d'un flux vidéo (ou de vidéos préenregistrées).

Le système peut être adapté à différents contextes :

- comptage de clients en magasin,
- suivi de véhicules sur la route,
- gestion de stocks visuels dans un entrepôt,
- surveillance intelligente.

## 2. Programmation

### 2.1. Semaine 1 : Cadrage du problème et choix des données

- Choix du contexte : sécurité, logistique, commerce, transport, etc.
- Définition des classes d'objets à détecter (personnes, voitures, boîtes, etc.).
- Sélection d'un dataset existant (COCO, Open Images, MOT, etc.) ou génération d'un petit dataset annoté.
- **Livrable : cahier des charges + choix du dataset + premiers exemples annotés.**

### 2.2. Semaine 2 : Analyse des données et prétraitement

- Exploration du dataset (dimensions, classes, qualité des annotations).
- Nettoyage éventuel, conversion vers format YOLO/COCO si besoin.
- Création d'un pipeline de traitement (chargement, affichage, annotation).
- **Livrable : notebook d'analyse + structure des données prêtes à l'emploi.**

### 2.3. Semaine 3 : Détection d'objets - version de base

- Implémentation d'un modèle de détection léger pré-entraîné (YOLOv5, YOLOv8, SSD, Faster R-CNN).
- Inférence sur des images fixes ou une courte vidéo.

- Première évaluation visuelle des résultats.
- **Livrable : script de détection avec résultats sur des exemples visuels.**

## 2.4. Semaine 4 : Détection en flux vidéo + comptage

- Application du modèle à des vidéos complètes ou flux webcam.
- Implémentation d'un système de comptage par classe.
- Détection d'entrées/sorties dans une zone définie (ligne virtuelle).
- **Livrable : démonstration de détection+comptage sur vidéo.**

## 2.5. Semaine 5 : Amélioration des performances et optimisation

- Tests sur plusieurs modèles (précision vs vitesse).
- Réduction des faux positifs / négatifs par filtrage ou ajustement de seuils.
- Suivi d'objets entre frames (tracking léger : SORT, DeepSORT).
- **Livrable : système plus robuste avec suivi + tableau comparatif des modèles.**

## 2.6. Semaine 6 : Interface utilisateur ou tableau de bord

- Interface graphique ou notebook Streamlit/Gradio : visualisation des vidéos avec détection en direct.
- Affichage en temps réel des comptages, statistiques, heatmaps (facultatif).
- Déploiement local ou cloud.
- **Livrable : démonstrateur avec interface conviviale.**

## 2.7. Semaine 7 : Tests, validation et documentation

- Scénarios de test (cas simples, cas complexes).
- Analyse des erreurs fréquentes.
- Documentation : manuel technique + mode d'emploi.
- **Livrable : rapport d'évaluation + dépôt Git bien documenté.**

## 2.8. Semaine 8 : Présentation et livraison finale

- Présentation orale avec démonstration en direct.
- Discussion sur les limites, pistes d'amélioration.
- Livraison du projet complet (code, vidéo, rapport, interface).
- **Livrable final : système intelligent de détection/comptage prêt à l'emploi.**

## 3. Outils et bibliothèques suggérés

- Python, OpenCV, PyTorch, Ultralytics YOLOv5/v8/v11, TensorFlow (optionnel).
- Streamlit ou Gradio pour l'interface.
- GitHub pour le code et le suivi.

## 4. Résultat attendu

Un **outil de détection et de comptage d'objets en temps réel**, avec interface, capable de fonctionner sur des vidéos réalistes ou en direct, documenté, livré et prêt à être présenté à un client.