

# Reconnaissance d'objets en vidéos: *Etat de l'art* et solutions Machine Learning

Groupe E4FI : *ILYAS GHANDAOUI*

ESIEE Paris — 9 novembre 2025

## Table des matières

# Résumé exécutif

La reconnaissance d'objets en temps réel dans les flux vidéo est un enjeu majeur pour de nombreux secteurs industriels, notamment la surveillance, la conduite autonome, le retail et la sécurité publique. Ce problème consiste à détecter et localiser rapidement des objets d'intérêt dans des séquences vidéo tout en garantissant précision et faible latence.

Ce rapport analyse les principales méthodes de pointe à fin 2025 pour aborder cette problématique, en se concentrant notamment sur des architectures issues du Machine Learning telles que YOLO (*You Only Look Once*), les Transformers adaptés à la vidéo, et les modèles DETR et leurs variantes récentes. Ces approches offrent un compromis entre rapidité d'exécution, précision de détection et robustesse aux conditions complexes (occlusions, petits objets, variations d'éclairage).

Les conclusions clés indiquent que YOLO est particulièrement adapté aux applications temps réel grâce à sa rapidité, tandis que les Transformers et DETR permettent une meilleure compréhension temporelle et une plus grande précision, au prix d'une complexité et de besoins en ressources supérieurs. L'intégration d'agrégation temporelle et les approches faiblement supervisées émergent pour améliorer la performance et réduire la dépendance aux annotations coûteuses.

En termes de faisabilité, l'entreprise peut déployer une solution efficace en combinant un modèle performant (type YOLO ou Transformer selon usage), des *datasets* publics adaptés (ImageNet VID, YouTube-VOS) et en s'engageant à assurer la conformité avec la réglementation RGPD au travers de mesures de protection des données robustes. Une implantation progressive, avec *proof-of-concept* sur des cas d'usage ciblés, suivie d'itérations d'amélioration, est recommandée pour maîtriser les coûts et les risques techniques.

## 1 Introduction et contexte

### 1.1 Mise en contexte du projet

La reconnaissance d'objets dans les vidéos est un défi crucial pour de nombreuses applications industrielles aujourd'hui. En surveillance, elle permet d'identifier en temps réel des intrusions, des comportements anormaux ou des objets abandonnés, contribuant ainsi à la prévention proactive des risques. Dans les véhicules autonomes, la détection dynamique des piétons, panneaux ou obstacles assure une navigation sécurisée et réactive. Le retail profite également de cette technologie pour analyser les comportements clients, optimiser l'agencement des rayons et limiter les pertes dues au vol.

Cette évolution est renforcée par la montée en puissance de la vision par ordinateur grâce à l'intelligence artificielle, qui transforme les systèmes traditionnels en plateformes intelligentes capables de décisions automatiques.

### 1.2 Objectifs du rapport

Ce rapport vise à :

- Explorer exhaustivement les méthodes de Machine Learning existantes et émergentes pour la reconnaissance d'objets vidéo ;
- Comparer leurs performances techniques (précision, vitesse, robustesse) et leur applicabilité aux cas d'usage industriels ;
- Identifier les défis techniques, biais potentiels dans les données et limites des approches actuelles ;
- Formuler des recommandations concrètes orientant le choix des technologies et des *datasets*, ainsi que les modalités de conformité vie privée.

### 1.3 Méthodologie de recherche

La recherche bibliographique s'est appuyée sur des bases spécialisées telles que Google Scholar, IEEE Xplore et arXiv pour accéder aux publications scientifiques récentes, ainsi que sur des ressources techniques industrielles actualisées en 2025 (blogs reconnus, documentations GitHub, rapports d'experts). Les méthodes ont été sélectionnées en fonction de leur popularité, leur performance démontrée sur des *benchmarks* standards (ImageNet VID, COCO), et leur pertinence pour les besoins temps réel ou haute précision.

L'approche adoptée est comparative, analysant tant les aspects quantitatifs (métriques, complexité) que qualitatifs (robustesse, biais, besoins en données), pour fournir une synthèse critique permettant une prise de décision éclairée par l'entreprise.

## 2 État de l'art des méthodes

### 2.1 Famille YOLO (You Only Look Once)

**Principe** YOLO reformule la détection d'objets en une régression unique effectuée en une seule passe. L'image est divisée en grille, chaque cellule prédit directement boîtes englobantes et classes, assurant une inférence ultra-rapide adaptée au temps réel.

#### Évolution

- YOLOv1–v3 : unification une étape, ancrés, multi-échelles, backbone Darknet-53.
- YOLOv4 : CSPNet, Mosaic/CutMix, entraînement SAT ; 43.5% mAP COCO.
- YOLOv5 : implémentation PyTorch, auto-hyperparamètres, export ONNX/TensorRT.
- YOLOv8 : tête *anchor-free*, tâches unifiées, Distribution Focal Loss.
- YOLOv10 : entraînement sans NMS via assignations doubles cohérentes ; latence réduite.
- YOLOv11 : blocs C3k2, SPPF ; meilleures performances à paramètres égaux, tâches étendues.

**Performances** En 2025, YOLOv11 couvre 39–55% mAP avec 1.5–11 ms par image sur GPU T4 ( $\approx$  88–200+ FPS), offrant un excellent compromis précision/vitesse sur vidéo temps réel.

## Avantages

- Vitesse exceptionnelle et latence faible.
- Précision compétitive face à des méthodes plus complexes.
- Moins de faux positifs grâce au contexte global.
- Polyvalence : détection, segmentation, classification, pose.
- Déploiement facilité (CPU/GPU/edge ; export vers ONNX/TensorRT/CoreML).

## Limites

- Petits objets/occlusions restent difficiles, surtout en variantes légères.
- Ressources GPU nécessaires pour les versions précises ( $1/x$ ).
- Sensible aux conditions extrêmes (éclairage, flou).
- Inévitable *trade-off* précision/vitesse selon la taille du modèle.

## Applications typiques

- Surveillance temps réel, conduite autonome, retail analytics, inspection industrielle.

**Type d'apprentissage** Supervision classique avec boîtes et labels ; pertes localisation/classe/confiance. Entraînement moderne : augmentation (Mosaic/MixUp), multi-échelle, assignation dynamique.

## 2.2 Architectures Transformer pour vidéo

**Contexte ViT→vidéo** ViT traite des patches comme tokens ; en vidéo, il faut modéliser simultanément dimensions spatiales et temporelles pour suivre objets et actions sur plusieurs frames.

**TGBFormer** Fusion Transformer + GraphFormer : dépendances globales et relations locales spatio-temporelles ; **86.5% mAP à 41 FPS** (ImageNet VID), viable pour haute précision avec contraintes temps réel modérées.

**VideoGLaMM** Alignement multimodal pixel-niveau : relie descriptions textuelles et localisations précises dans la vidéo via encodeurs visuels duals et décodeur spatio-temporel ; ouvre recherche/annotation par langue naturelle.

## ViViT et variantes

- Attention jointe spatio-temporelle (performante mais coûteuse).
- Encodeur factorisé (spatial puis temporel), plus efficace.
- Réduction temporelle/pooling central ; AMViT (mémoire adaptative) pour longues vidéos.

### Avantages

- Modélisation temporelle supérieure et compréhension contextuelle riche.
- Robustesse au mouvement rapide via agrégation multi-frame.
- Flexibilité multimodale (texte/audio).

### Limites

- Complexité quadratique, mémoire KV coûteuse.
- Besoins élevés en données, latence d'inférence.

**Type d'apprentissage** Principalement supervisé (pré-entraînement images/vidéos, fine-tuning). Forte émergence d'auto-supervision (prédiction futures, contrastif, réordonnement).

## 2.3 DETR et variants

**DETR original** *Set prediction* end-to-end : backbone CNN + encodage positionnel → Transformer ; *object queries* interrogent les features, têtes prédisent classe/boîte ; appariement bipartite (Hungarian) élimine le NMS. Convergence initiale lente.

**MI-DETR** Remplace la cascade par interrogations parallèles (*multi-time inquiries*) via têtes SA/CA/FFN indépendantes dont les sorties sont concaténées et projetées. Améliore extraction d'information et robustesse aux occlusions/variations.

## 2.4 Agrégation temporelle inter-frames

**YOLO + agrégation de features** Sélection/agrégation multi-échelle de features sur plusieurs frames : **92.9% AP50 à 30+ FPS**. Idéal quand la précision prime avec latence légère.

### Exploitation inter-frames

- Réduction du bruit et des faux positifs isolés.
- Complétion d'occlusions temporaires et stabilité des identités.
- Bounding boxes et classes plus précises ; suivi implicite.

**Avantages** Performance sur mouvements rapides ; robustesse aux conditions variables ; tracking simplifié.

**Limites** Coût mémoire (cache features), sensibilité aux changements brusques d'apparence, légère latence (2–5 frames).

## 2.5 Approches émergentes

**SNNs (MSD)** Détection bio-inspirée économique en énergie : **62.0% mAP** avec 7.8M paramètres et 6.43 mJ ; Spiking-YOLO approche 98% de Tiny YOLO avec consommation  $\sim 280\times$  inférieure.

**Faible supervision** DOtA (détection 3D multi-agents sans annotations), PointSR (supervision point-level), DVIN (vision-langage référentiel) réduisent massivement le coût d'annotation.

**Potentiel futur** Déploiement embarqué/edge (Loihi/TrueNorth), réduction des coûts d'annotation ( $\sim 50\text{--}90\%$ ), meilleure scalabilité vers nouveaux domaines.

## 2.6 Tableau comparatif synthétique

Méthode	mAP/FPS	Points forts	Limites	Cas d'usage
YOLO (v11)	39–55% 88–200+	/ Vitesse, simplicité, déploiement edge	Petits objets, GPU requis pour haute précision	Surveillance, embarqué, retail
Transformers vidéo	80–87% 30–40	/ Contexte global, temporalité, multimodal	Complexité, mémoire, latence	Analyse avancée, annotation, recherche vidéo
DETR / MI-DETR	43–55% 20–40	/ End-to-end, pas de NMS, robustesse	Convergence/latence, coût inférence	Scènes complexes, occlusions
YOLO + agrégation	AP50 30+	92.9 / Précision stabilisée multi-frame	Mémoire, latence légère	Haute précision très rapide temps réel
SNNs (MSD)	62% / très efficace	Énergie ultra-faible, puces neuromorphiques	Perf. inférieure, outillage limité	Embarqué batterie/IoT
Faible supervision	— / —	Moins d'annotations (50–90% gain)	Perf. dépend des signaux faibles	Nouvelles classes/domaines, prototypage

## 3 Datasets disponibles

### 3.1 ImageNet VID

**Description** Benchmark standard pour la détection d'objets en vidéo, dérivé d'ImageNet et étendu aux séquences avec annotations temporelles cohérentes.

#### Caractéristiques

- $\sim 4000$  séquences vidéo en train/val, 30 catégories d'objets (animaux, véhicules, objets du quotidien).

- Annotations en bounding boxes et labels de classe sur frames sélectionnées.
- Scènes naturelles : mouvements de caméra, changements d'échelle, occlusions, variations d'éclairage.

**Forces** Standard largement adopté ; annotations de haute qualité ; défis vidéo réalistes ; évalue la cohérence temporelle (ex. : TGBFormer 86.5% mAP à 41 FPS ; ClipVID 84.7% mAP à 39.3 FPS).

**Biais identifiés** Échelle limitée ; surreprésentation de certaines catégories ; séquences courtes ; résolutions variables.

**Utilisation recommandée** Benchmarking et validation par rapport à l'état de l'art avant application sur données spécifiques au domaine.

## 3.2 YouTube-VOS

**Description** Dataset à large échelle pour segmentation vidéo (spatio-temporel), clips YouTube diversifiés.

### Caractéristiques

- 4,453 clips (2018), 78 catégories ; version 2021 : 3,859 vidéos.
- Annotations pixel-level tous les 5 frames ( $\approx 6$  FPS), objets multiples par clip (jusqu'à 5).
- 133,886 annotations (2018) ; 232k annotations (2021) sur 8,171 instances uniques.

**Forces** Très grande échelle ; diversité réaliste ; 26 catégories de validation non vues (mesure de généralisation) ; masques précis.

**Limites** Clips courts (3–6 s) ; annotations *skip-frame* ; complexité très variable ; biais de popularité YouTube.

**Utilisation recommandée** Entraînement générique de détection/segmentation vidéo ; pré-entraînement avant fine-tuning ; tâches pixel-level (édition, AR).

## 3.3 COCO (Common Objects in Context)

**Description** Dataset d'images statiques massivement utilisé pour pré-entraîner les détecteurs avant adaptation vidéo.

### **Caractéristiques**

- 330k images (200k annotées), 80 catégories ;  $\sim 1.5M$  instances ( $\approx 47$  objets/image).
- Annotations multi-tâches : bounding boxes, masques instance, keypoints (250k+ personnes), segmentation *stuff*, 5 captions par image.
- Ensembles standardisés : Train2017 (118k), Val2017 (5k), Test2017 (20k).

**Usage pour la vidéo** Pré-entraînement des backbones (ResNet, EfficientNet, Transformers) ; performances COCO corrèlent les capacités vidéo (ex. : YOLOv11x 54.7% mAP, MI-DETR 52.4% mAP).

**Forces** Très grande échelle ; intégration facilitée (PyTorch/TensorFlow/Ultralytics) ; métriques rigoureuses ; scènes en contexte naturel.

**Limites** Pas de temporalité ; sous-représentation des très petits objets ( $<32 \times 32$ ) ; distribution de catégories déséquilibrée ; annotations statiques.

**Utilisation recommandée** Pré-entraînement incontournable pour tout détecteur ; fine-tuning sur petit dataset vidéo cible.

## **3.4 DAVIS (Densely Annotated VIdeo Segmentation)**

**Description** Benchmark haute qualité pour segmentation vidéo.

### **Caractéristiques**

- DAVIS 2016 : 50 séquences Full HD (1080p, 24 FPS) ; 2017 : 90 séquences, multi-objets.
- Annotations pixel-level exhaustives sur tous les frames ; attributs de défis annotés.
- Métriques : similitude région (J), précision contours (F), cohérence temporelle (T).

**Forces** Masques *pixel-perfect* ; qualité vidéo 1080p ; benchmark standard reconnu ; couverture systématique des défis.

**Limites** Échelle très limitée ; performances en voie de saturation ; coût d'annotation prohibitif ; focalisation segmentation binaire.

**Utilisation recommandée** Benchmark/validation pour segmentation très précise ; compléter par données domaine-spécifiques pour entraînement.

## **3.5 OD-VIRAT**

**Description** Benchmark large échelle pour détection en surveillance réaliste ; variantes Large (8.7M instances / 599,996 images) et Tiny (288,901 instances / 19,860 images).

## Caractéristiques

- 10 scènes de surveillance (chantiers, parkings, rues), caméras statiques en hauteur.
- Objets de petite échelle ; 5 catégories : Bike/Bicycle, Car, Carrying\_object, Person, Vehicle.
- Arrière-plans complexes ; sampling 0-frame-skip (Large) vs. 30-frame (Tiny).

**Forces** Conditions de surveillance authentiques ; échelle massive (Large) ; benchmarking spécialisé ; résolutions HD ( $1280 \times 720$ ,  $1920 \times 1080$  à 25–30 FPS).

**Limites** Domaine spécialisé (généralisation limitée) ; objets très petits difficiles ; seulement 5 catégories ; biais géographiques/temporalité.

**Utilisation recommandée** Entraîner/évaluer des détecteurs pour surveillance ; utiliser Tiny pour prototypage rapide et Large pour entraînement robuste ; adapter aux cas non-surveillance via fine-tuning.

## 3.6 Tableau comparatif datasets

Dataset	Taille	Annotations	Domaine	Biais principaux
ImageNet VID	~4k vidéos / 30 classes	BBoxes par frame	Détection vidéo générique	Échelle limitée ; sur représentation classes séquences courtes
YouTube-VOS	4,453 clips / 78 cat.	Masques pixel-level ( <i>skip-frame</i> )	Segmentation vidéo générale	Clips courts ; popularité YouTube ; variabilité forte
COCO	330k images / 80 cat.	BBoxes ; masques ; keypoints ; captions	Images statiques génériques	Pas de temporalité ; petits objets rares ; classes déséquilibrées
DAVIS	50–90 séquences (FHD)	Masques sur <b>tous</b> les frames	Segmentation vidéo précise	Échelle limitée ; saturation performances ; coût annotation
OD-VIRAT	8.7M inst. (Large)	BBoxes surveillance (5 cat.)	Surveillance réaliste	Petits objets ; domaine étroit ; biais géographique

## 4 Métriques de performance

L'évaluation rigoureuse des détecteurs d'objets vidéo repose sur des métriques complémentaires couvrant la localisation spatiale, la classification, et la vitesse (temps réel). Cette section synthétise les métriques standard, leurs interprétations et leur pertinence selon les cas d'usage.

## 4.1 Métriques de précision spatiale

**IoU (Intersection over Union)** Mesure le chevauchement entre la box prédite et la vérité terrain. *Définition :*

$$\text{IoU} = \frac{\text{area}(B_{\text{pred}} \cap B_{\text{gt}})}{\text{area}(B_{\text{pred}} \cup B_{\text{gt}})}$$

Pour la classification binaire (présence/absence sur pixel ou instance), on rencontre l'approximation suivante :

$$\text{IoU} = \frac{TP}{TP + FP + FN}$$

*Interprétation :*

- IoU = 0 : aucun chevauchement
- $0 < \text{IoU} < 0.5$  : localisation imprécise
- $\text{IoU} \geq 0.5$  : détection généralement considérée valide
- $\text{IoU} \geq 0.75$  : haute précision de localisation
- IoU = 1 : correspondance parfaite

*Pertinence* : métrique fondamentale de localisation, pénalise sous- et sur-détection. *Limits* : forte sensibilité aux petits objets ( $< 32 \times 32$ ), n'encode pas l'exactitude de la classe.

**Précision et Rappel** *Définitions :*

$$\text{Precision} = \frac{TP}{TP + FP} \quad ; \quad \text{Recall} = \frac{TP}{TP + FN}$$

*Trade-off* selon le seuil de confiance :

- Seuil élevé : précision  $\uparrow$ , rappel  $\downarrow$  (modèle conservateur)
- Seuil bas : précision  $\downarrow$ , rappel  $\uparrow$  (modèle permissif)

*Choix contextuel :*

- Surveillance : privilégier rappel élevé (éviter FN)
- Retail analytics : équilibre précision/rappel (F1)
- Conduite autonome : précision **et** rappel très élevés ( $> 0.98$ )

## 4.2 Métriques agrégées

**F1-Score** Moyenne harmonique précision–rappel :

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

Pénalise les modèles déséquilibrés, utile sur datasets avec classes majoritaires/minoritaires.

**Average Precision (AP) et Mean Average Precision (mAP)** *AP* résume la courbe précision–rappel (aire sous courbe). *mAP* : moyenne des AP sur toutes les classes.

- *mAP@0.50* (VOC) : seuil IoU unique à 0.50, indulgent sur la localisation.
- *mAP@0.50 :0.95* (COCO) : moyenne sur  $\text{IoU} \in \{0.50, \dots, 0.95\}$ , plus exigeant (souvent 15–20 pts *en dessous* de mAP@0.50 pour un même modèle).

*Pertinence* : standard de comparaison inter-modèles (COCO, ImageNet VID). *Limites* : masque les faiblesses sur classes rares, biais si objets « faciles » dominant, peu sensible à la consistance temporelle.

### 4.3 Métriques de vitesse

**FPS (Frames Per Second)** Nombre d’images traitées par seconde — critère clé pour le temps réel.

- < 15 FPS : trop lent (offline ou non critique)
- 15–25 FPS : acceptable si tolérance aux retards
- 25–30 FPS : minimum pour fluidité perceptuelle
- 30–60 FPS : idéal pour temps réel (surveillance, robotique)
- > 60 FPS : excellent, marge pour tracking/post-traitements

Exemple : TGBFormer atteint 86.5% mAP à 41 FPS (bon équilibre précision/vitesse).

**Latence** Délai capture→détection (prétraitement + inférence + post-traitement).

$$\text{Latence} = t_{\text{détection disponible}} - t_{\text{capture}}$$

*Relation* : *FPS* mesure le débit, *latence* la réactivité (pipeline : FPS élevé possible avec latence élevée). *Contraintes typiques* :

- **Conduite autonome** : < 50 ms    **Drone** : < 30 ms
- **Surveillance** : < 200 ms    **Analytics offline** : non critique

*Variabilité matériel* : GPU A100/V100  $\gg$  RTX 3060  $\gg$  CPU  $\gg$  Edge (Jetson/mobile). Optimisations courantes : quantification (FP32  $\rightarrow$  INT8), pruning, distillation, TensorRT/ONNX (gain 2–5× ; perte mAP 1–3 pts).

### 4.4 Choix des métriques selon cas d’usage

Le choix doit refléter coûts relatifs des erreurs et contraintes opérationnelles.

- **Surveillance sécurité** : rappel élevé, mAP@0.50, FPS  $\geq$  25.
- **Conduite autonome** : mAP@0.50 :0.95, FPS  $\geq$  30, latence < 50 ms.
- **Retail analytics** : précision, rappel, F1 élevé ( $\leq$  temps réel strict).
- **Inspection industrielle** : précision élevée, IoU/mAP@0.75 (localisation fine).
- **Analyse sportive** : mAP@0.50, FPS  $\geq$  30, F1 (fluidité prioritaire).
- **Santé/Médical** : précision très élevée, IoU, mAP@0.75 (offline acceptable).

Cas d'usage	Métriques prioritaires	priori-	Seuils typiques	Justification
Surveillance	Rappel, FPS	mAP@0.50,	Rappel > 0.95, FPS $\geq 25$	Éviter FN critiques, localisation modérée suffisante, temps réel requis.
Conduite autonome	mAP@0.50	:0.95, Latence	Latence < 50 ms, FPS $\geq 30$	Précision & réactivité maximales, erreurs très coûteuses.
Retail analytics	Précision, Rappel, F1	F1 > 0.85		Statistiques fiables, temps réel strict non critique (15–20 FPS).
Inspection industrielle	IoU, mAP@0.75, Précision		IoU/mAP élevés	Localisation fine pour action robotique, coûts FP/FN élevés.
Analyse sportive	mAP@0.50, FPS, F1		FPS $\geq 30$	Fluidité visuelle, corrections possibles en post-prod.
Santé/Médical	Précision, mAP@0.75	IoU,	Précision $\rightarrow$ max	Diagnostic fiable, minimiser FP/FN, offline acceptable.

## 4.5 Tableau récapitulatif : Métriques par cas d'usage

## 5 Types d'apprentissage

Cette partie présente les paradigmes d'apprentissage pertinents pour la détection d'objets en vidéo, leurs coûts et leurs compromis de performance.

### 5.1 Apprentissage supervisé

**Principe.** Données annotées (bounding boxes + classes) et optimisation d'une perte de localisation & classification.

**Stratégies d'annotation.** Annotation *dense* (toutes les frames) vs *skip-frame* (p.ex. 1 FPS sur une vidéo 30 FPS, coût réduit ~97%). Les frames intermédiaires sont apprises par propagation temporelle.

**Méthodes concernées.** YOLO (v1–v11), Transformers vidéo (p.ex. ViViT), DETR/MIDETR ; pré-entraînement sur COCO/ImageNet puis fine-tuning domaine.

**Avantages.** Précision maximale sur benchmarks ; performance prédictible si données suffisantes ; transfert efficace ; outillage mature (PyTorch, CVAT, Labelbox).

**Limites.** Coût d'annotation élevé ; biais humains ; généralisation limitée hors distribution ; dépendance au domaine (changement de classes/contexte  $\Rightarrow$  nouvelles annotations).

**Coût indicatif.** 1000 vidéos, 1 min, skip-frame 1 FPS  $\Rightarrow$  ~60k frames. Scénario modéré (4 objets/frame) : coût total ~\$23.6k (plateforme + main d'œuvre). Segmentation pixel-level  $\sim 3\times$  plus coûteuse.

## 5.2 Apprentissage non supervisé

**Principe.** Découverte de représentations à partir de signaux vidéo naturels (cohérence temporelle, continuité spatiale) sans labels.

**Exemples.** Cohérence par tracking ; segmentation auto-supervisée (p.ex. SOLV) ; random walks sur graphes ; apprentissage égocentrique.

**Avantages.** Zéro annotation ; exploitation de corpus massifs (web) ; découverte de patterns ; robustesse améliorée aux changements de domaine.

**Limites.** Précision inférieure au supervisé ; conception/entraînement plus complexes ; validation difficile sans ground truth ; sensibilité aux biais des données.

## 5.3 Approches hybrides

**Faible supervision.** Point-level (p.ex. PointSR), image-level (WSOD, MIL), pseudo-labels raffinés (p.ex. W2N), collaboration segmentation-détection (p.ex. SDCN). Réduction de coût de 80–95% vs bounding boxes.

**Auto-supervision.** Tâches de pré-texte (reconstruction masquée), adaptation de scène par auto-enseignement, cohérence multi-vues (p.ex. DOtA). Foundation models (SAM, CLIP) facilitent zero/few-shot.

**Avantages.** Coûts drastiquement réduits ; performances ~85–95% du supervisé ; meilleure scalabilité.

**Limites.** Maturité industrielle inégale ; léger gap de performance (5–15%) inacceptable en cas critique ; hyperparamètres supplémentaires.

## 5.4 Recommandation stratégique

**Court terme (0–6 mois).** Supervisé pour déploiement rapide : modèles pré-entraînés (YOLOv11, MI-DETR), fine-tuning sur 500–2000 vidéos, annotation skip-frame 1 FPS ; privilégier qualité d'annotation et pré-annotation (SAM) pour ~30–50% de gain.

**Moyen terme (6–18 mois).** Explorer faible supervision sur un sous-ensemble ; comparer vs supervisé ; si  $\geq 90\%$  de la performance, migrer progressivement (hybride : 10–20% supervisé complet + 80–90% faible).

**Long terme (18+ mois).** Auto-supervision & foundation models ; évaluer régulièrement zero/few-shot et arbitrer coût/performance.

**ROI simplifié.**

$$\text{ROI} = \frac{(V \times P) - C}{C},$$

où  $V$  est la valeur business,  $P$  la performance (normalisée),  $C$  le coût d'annotation. Une baisse de  $C$  de 80% pour  $\sim 10\%$  de perte de  $P$  améliore fortement le ROI pour des applications non critiques.

Application	Perf. min.	Supervision	Justification
Conduite autonome	>98% P/R	Supervisé complet	Criticité sécurité
Surveillance	>95% R	Supervisé + Faible	Événements critiques
Retail	>85% F1	Faiblement supervisé	Scalabilité prioritaire
Médical	>97% Précision	Supervisé complet	Réglementation stricte
Sport	>80% mAP	Faible/Auto	Volume élevé
Inspection qualité	>90% Précision	Hybride	Selon sévérité défauts

TABLE 1 – Matrice décisionnelle pour le choix du paradigme.

## 6 Vie privée et sécurité

Cette partie présente le cadre RGPD et les mesures techniques pour un déploiement à la fois efficace et conforme de la détection d'objets en vidéo.

### 6.1 Cadre réglementaire RGPD

**Surveillance vidéo = données personnelles.** Les flux vidéo capturent des éléments identifiants (visages, silhouettes). Tout traitement (collecte, conservation, analyse ML) est soumis au RGPD.

**Bases légales (Art. 6(1)).** Consentement (rarement applicable), Contrat (cas limités), Obligation légale (spécifique), Intérêts vitaux (urgence), Mission d'intérêt public (autorités), Intérêt légitime (le plus courant, sous conditions de nécessité et proportionnalité).

Base légale	Applicabilité	Remarques
Consentement	Rare	Difficulté de refus libre
Contrat	Limite	Nécessité stricte
Obligation légale	Spécifique	Secteurs réglementés
Intérêts vitaux	Urgence	Temporaire
Intérêt public	Autorités	Service public
Intérêt légitime	Fréquent	Nécessité + proportionnalité + information

TABLE 2 – Synthèse des bases légales applicables à la vidéosurveillance.

**Droits des personnes.** Information (signalétique), accès, rectification, effacement, opposition, limitation.

### 6.2 Évaluation d'impact (DPIA)

Obligatoire si risque élevé (Art. 35), typiquement pour une surveillance systématique à grande échelle en zone publique. Contenu : description du traitement, nécessité/proportionnalité, analyse des risques (probabilité & gravité), mesures d'atténuation (techniques & organisationnelles). Consultation de la CNIL si risque résiduel élevé (Art. 36).

## 6.3 Mesures techniques de protection

**Privacy by Design.** Intégrer la protection dès la conception : edge/federated learning, anonymisation par défaut, champs de vue limités, séparation précoce des identités.

**Anonymisation & pseudonymisation.** Privilégier l'anonymisation irréversible (flouage/pixellisation/masquage non réversible). Entraîner sur vidéos anonymisées ; des travaux (p. ex. E2PRIV) montrent des performances de détection quasi identiques tout en réduisant fortement les risques d'identification.

**Chiffrement.** En transit : TLS 1.2+, HTTPS/VPN. Au repos : AES-256, gestion des clés sécurisée (rotation, isolement).

**Contrôle d'accès.** Moindre privilège, MFA, journaux d'audit complets et inviolables.

**Durées de conservation.** 72 h usuelles pour sécurité générale ; jusqu'à 30 jours si justifié. Suppression automatique à l'échéance.

## 6.4 Technologies alternatives

**LiDAR.** Données 3D sans texture ni couleur, détection d'objets efficace avec exposition minimale des identités. Coût en baisse ; utile pour périmètres sensibles, comptage, intrusions.

## 6.5 Données sensibles & minimisation

Catégories Art. 9 (biométrie, santé, opinions, etc.) à éviter. Si inévitable : faire une base légale (Art. 6) *et* une exception (Art. 9(2)). Appliquer la minimisation (angles/cadrage, résolution suffisante, désactivation des fonctions non nécessaires, masquage permanent des zones non pertinentes).

## 6.6 Checklist de conformité

1. DPIA complétée, risque résiduel acceptable documenté.
2. Base légale identifiée (souvent intérêt légitime) et test de proportionnalité réalisé.
3. Privacy by Design implémenté, chiffrement et contrôle d'accès en place.
4. Transparence : signalétique conforme, notice vie privée, procédures d'exercice des droits.
5. Durées de conservation configurées (72 h à 30 j) avec suppression automatique.
6. Gouvernance : DPO consulté, formation des équipes, contrats sous-traitants conformes, plan de réponse aux incidents.

## 7 Défis et limites

### 7.1 Défis techniques

La reconnaissance d'objets en vidéo s'appuie sur des modèles confrontés à des scènes et des prises de vue très variées. Plusieurs défis techniques majeurs se dégagent :

- **Objets petits ou fortement occultés.** Les objets de très petite taille (moins de  $32 \times 32$  pixels) ou partiellement cachés posent un problème de résolution et de visibilité. Ils génèrent peu de pixels utiles pour l'extraction de caractères discriminants, réduisant la fiabilité de la détection, tandis que l'occultation partielle rend difficile la distinction de l'objet de l'arrière-plan.
- **Variations d'éclairage et conditions météo.** Les changements brusques d'éclairage (passage du soleil à l'ombre, contre-jour) et les conditions météorologiques extrêmes (pluie, brouillard, neige) altèrent la qualité d'image et dégradent les performances de modèles formés sur des images claires et stables. Ces variations requièrent des stratégies d'augmentation de données et des prétraitements dynamiques (normalisation adaptative).
- **Mouvement rapide et flou de mouvement.** Les objets se déplaçant rapidement, ou la caméra en mouvement, génèrent du flou cinétique qui dilue les contours et rend la localisation approximative. Les architectures à une seule passe (YOLO) sont particulièrement sensibles au flou, tandis que les transformers bénéficient de l'agrégation temporelle mais souffrent d'une latence accrue liée au traitement des séquences.
- **Arrière-plans complexes et scènes encombrées.** Dans des environnements urbains ou industriels chargés, les objets cibles peuvent se confondre avec des éléments de décor similaires (panneaux, machines, mobilier). Les modèles doivent distinguer les objets pertinents malgré de nombreux distracteurs et textures variées, ce qui nécessite des capacités de contextualisation globale avancées.
- **Compromis précision vs vitesse pour le temps réel.** Les applications de surveillance et de conduite autonome exigent à la fois une haute précision et une faible latence. Les CNN comme YOLO offrent une vitesse élevée ( $> 88$  FPS pour YOLOv11) mais peuvent sacrifier de la précision sur petits objets ou scènes complexes, tandis que les transformers vidéo (TGBFormer, ViViT) améliorent la précision multi-cadres au prix d'un débit réduit (25–40 FPS) et d'une forte consommation mémoire. Trouver l'équilibre adéquat reste un défi permanent.

### 7.2 Biais des datasets

Les datasets publics présentent des biais pouvant limiter la généralisabilité et l'équité des modèles :

- **Déséquilibre de classes.** Un nombre disproportionné d'images pour certaines catégories (personne, véhicule, chien) entraîne des modèles surspécialisés au détriment d'objets moins fréquents. Par exemple, ImageNet VID et COCO surreprésentent certaines races de chiens et types de véhicules, biaissant les performances selon la classe.

- **Biais géographiques et culturels.** La majorité des vidéos provient de pays industrialisés (États-Unis, Europe, Asie de l'Est), exposant peu les modèles à des architectures, vêtements, véhicules ou scènes d'autres régions. Les systèmes entraînés sur ces données peuvent mal détecter des objets ou comportements spécifiques à des environnements différents.
- **Biais temporels.** Beaucoup de datasets datent de plus de cinq ans. Les objets et scènes évoluent rapidement (design de véhicules, styles vestimentaires, nouveaux dispositifs urbains). Les modèles risquent de manquer de sensibilité aux objets récents ou aux évolutions d'infrastructure, affectant leur pertinence en production.
- **Impact sur la généralisation et l'équité.** Ces biais mènent à une généralisation limitée et à des inégalités de performance selon les contextes et les populations filmées. Un détecteur peut fonctionner parfaitement sur des scènes diurnes occidentales mais échouer sur des environnements nocturnes ou ruraux.
- **Stratégies de mitigation.** *(i) Augmentation de données* : simuler conditions d'éclairage, flou, perspective et objets rares pour enrichir le spectre d'exemples. *(ii) Ré-échantillonnage* : équilibrer les classes en sur-échantillonnant les catégories sous-représentées ou en sous-échantillonnant les classes dominantes. *(iii) Datasets diversifiés* : combiner plusieurs sources (ImageNet VID, YouTube-VOS, OD-VIRAT, datasets locaux) couvrant variétés géographiques, culturelles et temporelles. *(iv) Validation out-of-distribution* : tester la robustesse sur des vidéos hors domaine d'entraînement pour mesurer la généralisation et détecter les zones de faiblesse.

**Conclusion.** La prise en compte proactive de ces défis et biais est essentielle pour développer des solutions fiables, équitables et robustes dans des contextes réels variés.

## 8 Recommandations et faisabilité

### 8.1 Matrice décisionnelle par cas d'usage

Cas d'usage	Architecture recommandée	Datasets	Métriques prioritaires	Co
Surveillance temps réel	YOLOv11 (mode nano à média)	ImageNet VID, OD-VIRAT	Rappel élevé, mAP@0.50, FPS $\geq 25$	Int ob
Analyse retail	TGBFormer	YouTube-VOS, COCO	Précision, mAP@0.50 :0.95, F1-Score	Int ne
Conduite autonome	YOLO + agrégation de features	KITTI, BDD	mAP@0.50 :0.95, FPS $\geq 30$ , Latence < 50 ms	Ob sic

TABLE 3 – Matrice décisionnelle par cas d'usage

### 8.2 Roadmap d'implémentation

1. **Phase 1 (0–3 mois).** POC avec YOLOv11 sur ImageNet VID et OD-VIRAT en mode *skip-frame* ; évaluation des performances baseline (mAP, rappel, FPS) ; réalisation de

- la première DPIA pour cadrer les obligations RGPD.
2. **Phase 2 (3–6 mois).** Collecte et annotation interne de 500–1 000 vidéos spécifiques (*skip-frame* 1 FPS) ; *fine-tuning* du modèle sur données internes ; intégration progressive de mesures Privacy by Design (anonymisation, chiffrement).
  3. **Phase 3 (6–12 mois).** Déploiement en production sur environnement restreint (edge ou cloud) ; monitoring continu (tableaux de bord mAP/FPS) et audit RGPD ; ajustement des seuils et optimisation des pipelines d’inférence.
  4. **Phase 4 (12+ mois).** Optimisation des modèles (pruning, quantification) pour *edge deployment* ; exploration de l’apprentissage faiblement supervisé (PointSR) et auto-supervisé (DOtA) ; scalabilité vers de nouveaux sites et cas d’usage.

### 8.3 Estimation des ressources

**Humaines** 1–2 Data Engineers (collecte, annotation, pipelines données) ; 1 Machine Learning Engineer (fine-tuning, optimisation, déploiement) ; 1 expert conformité RGPD (DPIA, audits, procédures).

**Techniques** GPU NVIDIA A100/V100 (ou cluster cloud équivalent) pour entraînement et inférence ; stockage 5–10 TB pour datasets vidéo et temporaires ; infrastructure cloud ou on-premise Kubernetes pour scalabilité.

**Financières** Licences éventuelles pour datasets commerciaux (surveillance privée) ; coût annotation externe : 10<sup>~</sup>25/h annotateur ; dépenses d’infrastructure cloud : 5 000<sup>~</sup>10 000/mois ; formation et ateliers RGPD : 20 000<sup>~</sup>30 000 initiaux.

**Temporelles** 6–12 mois pour déploiement complet et stabilisation ; itérations trimestrielles pour revue performance et conformité.

### 8.4 Risques et mitigation

**Risque technique** Performances insuffisantes sur cas réels (petits objets, flou). *Mitigation* : validation POC sur données internes, ajustement d’architecture (agrégation).

**Risque conformité** Violations RGPD (surveillance illégale). *Mitigation* : DPIA rigoureuse, consultation DPO, mise en place Privacy by Design.

**Risque budget** Dépassement des coûts d’annotation et d’infrastructure. *Mitigation* : phases incrémentales avec KPI clairs, basculement partiel vers supervision faible.

**Risque adoption** Résistance des utilisateurs internes (sécurité, IT). *Mitigation* : ateliers de sensibilisation, documentation des bénéfices, support technique dédié.

## 8.5 Conclusion sur la faisabilité

Les technologies de détection d’objets vidéo sont désormais matures et accessibles, adaptées à divers cas d’usage. La combinaison d’architectures éprouvées (YOLOv11, TGBFormer) et de jeux de données standard facilite une intégration rapide. La conformité RGPD est réalisable via une DPIA initiale, des mesures de Privacy by Design et un suivi continu. Un déploiement phasé permet de maîtriser coûts et risques, avec un ROI positif dès la phase POC. **Recommandation** : lancer un projet pilote supervisé, puis migrer progressivement vers des approches hybrides pour étendre le système à grande échelle.

## 9 Conclusion

La reconnaissance d’objets dans les flux vidéo repose désormais sur un panorama riche de solutions ML, allant des détecteurs unifiés à une passe (YOLO) aux architectures Transformer vidéo, en passant par des approches hybrides combinant supervision faible et auto-supervision. En 2025, YOLOv11 demeure une référence pour les applications nécessitant une vitesse extrême, tandis que les Transformers vidéo (TGBFormer, ViViT) offrent une précision et une compréhension temporelle supérieures au prix d’une plus grande complexité. Les méthodes émergentes, telles que les *Spiking Neural Networks* et les stratégies faiblement supervisées (PointSR, DOTa), promettent de réduire significativement les coûts d’annotation et d’ouvrir de nouveaux cas d’usage embarqués.

Les jeux de données disponibles sont diversifiés et volumineux : ImageNet VID et YouTube-VOS couvrent des scènes variées avec annotations (boîtes et masques), COCO fournit une base d’images statiques dense, DAVIS sert de référence pour la segmentation au niveau pixel, et OD-VIRAT expose les défis de la surveillance réaliste. Les métriques standardisées (IoU, mAP@0.50 :0.95, F1-Score, FPS, latence) assurent une comparaison rigoureuse entre modèles.

Le cadre RGPD impose un impératif de conformité qui reste réalisable : choisir une base légale adaptée (intérêt légitime), conduire une DPIA, appliquer des principes de *Privacy by Design*, et mettre en œuvre chiffrement, anonymisation et procédures d’exercice des droits. La combinaison d’un déploiement phasé, d’une architecture technique robuste et d’un suivi juridique régulier assure la faisabilité du projet.

**Perspectives à moyen et long terme** L’évolution du domaine s’oriente vers :

- des approches hybrides optimisant conjointement coût et performance ;
- l’*edge computing* pour réduire la latence et préserver la vie privée ;
- des techniques de traitement temps réel toujours plus efficaces.

Ces perspectives ouvrent la voie à des systèmes de reconnaissance d’objets vidéo performants, scalables et respectueux des droits fondamentaux.

## 10 Sources et références

- <http://arxiv.org/pdf/1704.00675v2.pdf>

- <http://arxiv.org/pdf/2108.13141.pdf>
- <https://arxiv.org/abs/2503.13903>
- <https://arxiv.org/html/2407.19650v1>
- <https://arxiv.org/html/2504.13099v1>
- <https://arxiv.org/html/2507.12396v1>
- <https://arxiv.org/pdf/1809.00461.pdf>
- <https://arxiv.org/pdf/1809.03327.pdf>
- <https://aws.amazon.com/compare/the-difference-between-machine-learning-supervised-and-unsupervised-learning/>
- <https://blog.roboflow.com/object-detection-metrics/>
- <https://cloix-mendesgil.com/en/legal-insights/it-contracts-data-and-compliance/augmented-camera-gdpr-compliance/>
- <https://dataprivacymanager.net/video-surveillance-cctv-under-gdpr/>
- <https://docs.ultralytics.com/compare/>
- <https://docs.ultralytics.com/guides/yolo-performance-metrics/>
- <https://encord.com/blog/video-object-tracking-algorithms/>
- <https://encord.com/blog/yolo-object-detection-guide/>
- <https://github.com/ArpitaSatsangi/Real-time-Object-Detection-with-YOLO-using-OpenCV>
- <https://github.com/bo-miao/awsome-video-object-segmentation>
- <https://github.com/huanglianghua/video-detection-benchmark>
- <https://github.com/muhammadshiraz/YOLO-Real-Time-Object-Detection>
- <https://github.com/NVlabs/FasterViT>
- <https://github.com/rafaelpadilla/Object-Detection-Metrics>
- <https://github.com/roboflow/rf-detr>
- <https://github.com/xiaobai1217/Awesome-Video-Datasets>
- <https://hirevire.com/blog/best-gdpr-compliant-video-interview-software-tools>
- <https://hiringnet.com/object-detection-state-of-the-art-models-in-2025>
- <https://homepages.inf.ed.ac.uk/rbf/CAVIAR/PAPERS/WECCV.pdf>
- [https://image-net.org/static\\_files/files/Imagenet%202017%20VID.pdf](https://image-net.org/static_files/files/Imagenet%202017%20VID.pdf)
- <https://imerit.net/resources/blog/real-time-object-detection-using-yolo/>
- <https://labelyourdata.com/articles/object-detection-metrics>
- [https://openaccess.thecvf.com/content/CVPR2024/papers/Kowal\\_Understanding\\_Video\\_Transformers\\_via\\_Universal\\_Concept\\_Discovery\\_CVPR\\_2024\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2024/papers/Kowal_Understanding_Video_Transformers_via_Universal_Concept_Discovery_CVPR_2024_paper.pdf)
- [https://openaccess.thecvf.com/content/ICCV2023/papers/Deng\\_Identity-Consistent\\_Aggregation\\_for\\_Video\\_Object\\_Detection\\_ICCV\\_2023\\_paper.pdf](https://openaccess.thecvf.com/content/ICCV2023/papers/Deng_Identity-Consistent_Aggregation_for_Video_Object_Detection_ICCV_2023_paper.pdf)
- <https://pdfs.semanticscholar.org/2d33/cb991624a17a012b7702897230f9fe416d50.pdf>
- <https://pmc.ncbi.nlm.nih.gov/articles/PMC12233284/>
- <https://public.roboflow.com/object-detection>
- <https://stackoverflow.com/questions/44718287/where-can-i-find-imagenet-vid-dataset>
- <https://viso.ai/deep-learning/object-detection/>
- <https://viso.ai/deep-learning/supervised-vs-unsupervised-learning/>
- <https://www.cnil.fr/en/ai-and-gdpr-cnil-publishes-new-recommendations-support-responsible-ai>
- [https://www.cs.cmu.edu/~xiaolonw/papers/unsupervised\\_video.pdf](https://www.cs.cmu.edu/~xiaolonw/papers/unsupervised_video.pdf)

- <https://www.datacamp.com/blog/yolo-object-detection-explained>
- [https://www.edpb.europa.eu/sites/default/files/files/file1/edpb\\_guidelines\\_201903\\_video\\_devices.pdf](https://www.edpb.europa.eu/sites/default/files/files/file1/edpb_guidelines_201903_video_devices.pdf)
- [https://www.edpb.europa.eu/sites/default/files/files/file1/edpb\\_guidelines\\_201903\\_video\\_devices\\_en\\_0.pdf](https://www.edpb.europa.eu/sites/default/files/files/file1/edpb_guidelines_201903_video_devices_en_0.pdf)
- <https://www.hitechbpo.com/blog/top-object-detection-models.php>
- <https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning>
- <https://www.labellerr.com/blog/cvpr-2025-part-1/>
- [https://www.mobotix.com/sites/default/files/2023-10/mx\\_WP\\_Data\\_Protection\\_en\\_231025.pdf](https://www.mobotix.com/sites/default/files/2023-10/mx_WP_Data_Protection_en_231025.pdf)
- <https://www.optex-europe.com/about/blog/lidar-and-gdpr-a-privacy-first-approach-to>
- [https://www.reddit.com/r/computervision/comments/1jfmcmw/what\\_are\\_the\\_most\\_useful\\_and\\_stateoftheart\\_models/](https://www.reddit.com/r/computervision/comments/1jfmcmw/what_are_the_most_useful_and_stateoftheart_models/)
- [https://www.reddit.com/r/computervision/comments/1jydymw/is\\_yolo\\_still\\_the\\_stateofart\\_for\\_object\\_detection/](https://www.reddit.com/r/computervision/comments/1jydymw/is_yolo_still_the_stateofart_for_object_detection/)
- <https://www.resemble.ai/state-art-object-detection-models/>
- <https://www.sciencedirect.com/science/article/abs/pii/S0893608023000199>
- <https://www.scribd.com/document/851187720/2025-AEJ-Object-detection-in-real-time-v>
- <https://www.ultralytics.com/blog/exploring-the-best-computer-vision-datasets-in-2025/>
- <https://www.v7labs.com/blog/supervised-vs-unsupervised-learning>
- <https://www.v7labs.com/blog/vision-transformer-guide>
- <https://www.viam.com/post/guide-yolo-model-real-time-object-detection-with-example>

## 11 Détection du feu en milieu urbain : état de l'art et pipeline

Cette section propose une implémentation pragmatique et conforme à l'état de l'art pour entraîner un modèle de détection du feu (et de la fumée) dans des environnements urbains, avec un accent sur la robustesse en conditions réelles (nuit, pluie, trafic, éclairages parasites).

### 11.1 Problématique et définitions

- Objectifs : détecter `feu` et `fumée` en temps réel, limiter les faux positifs (feux de signalisation, phares, néons, reflets). - Classes recommandées : `fire`, `smoke` (optionnel : `flare`, `sparks` si le périmètre l'exige). - Contraintes : petites cibles, occlusions, variations d'éclairage, conditions météorologiques.

### 11.2 Modèles SOTA et choix pratiques

- **YOLOv8/YOLOv10/RT-DETR** : excellents compromis précision/latence. Pour l'urbain temps réel : `yolov8s` ou `yolov8m`. Pour précision maximale : `yolov8l/x` ou `rt-detr-l`.

- Alternatives : *DETR/DINO/EfficientDet*. Utiles pour recherche, mais moins simples à déployer qu'Ultralytics YOLO.

### 11.3 Données et annotation

- **Sources publiques** : Roboflow (*Fire/Smoke*), Kaggle (*Fire detection*), CAVIAR (pour négatifs urbains), vidéos YouTube sous licence appropriée.
- **Stratégie** : constituer un corpus *feu/fumée* + un corpus *négatif* (nuit, pluie, éclairages, chantiers) pour réduire les faux positifs. - **Guidelines d'annotation** : encadrer la flamme visible (`fire`) et les panaches (`smoke`). Éviter d'annoter les reflets lumineux non liés à un incendie.

### 11.4 Augmentations recommandées

- Photométriques : HSV, contraste, bruit, blur (simulateur de fumée/bruine légère).
- Géométriques : rotation  $\leq 10^\circ$ , translation  $\leq 10\%$ , scale 0.9–1.1.
- Compositions : `mosaic` et `mixup` avec parcimonie (stabilité > latence).
- Spécifiques domaine : overlays de fumée synthétique (alpha) pour robustesse au voile.

### 11.5 Configuration dataset (Ultralytics)

Créer le fichier PROJET/fire/fire.yaml :

```
path: C:/Users/ilyas/Documents/COURS/ESIEE_2526_MachineLearnig/PROJET/fire
train: images/train
val: images/val
test: images/test
names:
  0: fire
  1: smoke
```

Arborescence attendue (formats YOLO) : `images/*` et `labels/*` avec fichiers `.txt` (classe, `x_center`, `y_center`, `width`, `height`) normalisés.

### 11.6 Entraînement (Ultralytics YOLO)

Script `train_fire.py` (voir dossier PROJET/fire) :

```
from ultralytics import YOLO
import torch

model = YOLO('yolov8m.pt') # speed/accuracy trade-off
model.train(
    data='C:/Users/ilyas/Documents/COURS/ESIEE_2526_MachineLearnig/PROJET/fire/fire.yaml')
```

```

    epochs=100, batch=16, imgsz=640,
    device=0 if torch.cuda.is_available() else 'cpu',
    workers=4, patience=20,
    cos_lr=True, optimizer='AdamW', lr0=0.001, weight_decay=0.0005,
    amp=True, verbose=True
)

```

## 11.7 Inférence et seuils

Script infer\_fire.py :

```

from ultralytics import YOLO
model = YOLO('runs/detect/train_fire/weights/best.pt')
model.predict(source='0', conf=0.35, iou=0.5, imgsz=640, stream=True)
# Ajuster conf pour réduire faux positifs selon le contexte urbain

```

## 11.8 Évaluation

- Métriques : mAP@50-95, précision, rappel, F1.
- Jeux de test dédiés : nuit/pluie/fort trafic pour robustesse.
- *Hard negative mining* : itérer en ajoutant des négatifs déclenchés à tort (feux tricolores, phares, reflets).

## 11.9 Déploiement et contraintes

- Edge (CPU) : yolov8n/s + quantification INT8 (ONNX/TensorRT) si possible.
- GPU : yolov8m/l pour meilleure précision.
- Observabilité : journaliser alertes + extraits vidéo, revue humaine, boucle d'amélioration continue.

## 11.10 Risques et conformité

- Faux positifs (signalisations, reflets) ⇒ seuils adaptés, filtre par taille minimale de boîte.
- Confidentialité : respecter la réglementation (GDPR) pour l'usage de flux urbains et l'archivage.

## 11.11 Checklist rapide

- Dataset équilibré (feu/fumée/négatifs), validation séparée.
- Entraînement 80–100 epochs, early stopping, suivi des courbes.
- Seuils conf adaptés au contexte, tests terrain (jour/nuit/pluie).
- Déploiement avec journalisation et amélioration continue.