UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

DÉTECTION DE PALETTES À BASE DE VISION POUR NAVIGATION D'APPROCHE DE ROBOT MOBILE AUTONOME : PERFORMANCE SOUS DONNÉES LIMITÉES ET ENVIRONNEMENT DYNAMIQUE

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR
IBRAHIM BOUBAKRI

JUIN 2026

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

Maîtrise en génie électrique avec mémoire (3824)

Direction de recherche :

Pr Sousso Kelouwani	Directeur de recherche
Dr Ali Akrem Amamou	Codirecteur de recherche

Jury d'évaluation

Pr Miloud Bagaa	Évaluateur interne
Pr Jean-Sébastien Dessureault	Évaluateur externe
Pr Sousso Kelouwani	Directeur de recherche

Résumé

L'automatisation des opérations logistiques exige une détection fiable des palettes pour guider précisément les robots mobiles. Une palette est un support logistique destiné au stockage et à la manutention de marchandises, muni d'ouvertures permettant sa prise par les fourches d'un chariot élévateur. L'entraînement des modèles de vision requiert toutefois de nombreuses images réelles annotées, coûteuses à collecter. Ce mémoire étudie les données synthétiques pour réduire ce besoin sans sacrifier la performance.

Une plateforme de génération de données synthétiques a été développée pour Noovelia. Elle automatise la production d'images randomisées et annotées, puis intègre l'entraînement, la simulation et la visualisation. Plusieurs architectures *You Only Look Once Pose* (YOLO-Pose) sont évaluées avec des données réelles seules, synthétiques seules ou un préentraînement synthétique suivi d'un ajustement réel. L'évaluation porte sur les jeux de données réels Costco et Noovelia avec différents volumes, et comprend des essais hors ligne, de robustesse et inter-sites, puis une intégration en simulation et sur robot réel.

Avec 31 images réelles annotées, la stratégie hybride atteint 87,9 % de précision moyenne globale à un seuil d'intersection sur union de 0,5 (mAP@0,5), contre 52,2 % pour les données réelles seules. Avec 210 images, elle atteint 91,8 % contre 85,2 %, les gains diminuant avec le volume de données réelles. L'intégration sur un robot chez Noovelia confirme que les sorties du détecteur sont exploitables par la navigation. Elle valide également la plateforme de génération de données synthétiques comme outil permettant d'accélérer l'adaptation aux nouveaux clients.

Mots-clés : détection de palettes, données synthétiques, transfert synthétique-réel, estimation par points clés, YOLO-Pose, robot mobile autonome.

Abstract

The automation of logistics operations requires reliable pallet detection so that mobile robots can approach their targets accurately. A pallet is a logistics platform used to store and handle goods, with openings that allow it to be lifted by forklift tines. Training vision models, however, generally requires a large number of annotated real images, which are costly to collect. This thesis investigates the use of synthetic data to reduce this requirement while maintaining strong detection performance.

To address this problem, a synthetic-data generation platform was developed for Noovelia. It automates the production of annotated synthetic images and integrates training, simulation, and visualization. Several You Only Look Once Pose (YOLO-Pose) architectures are evaluated with three strategies: real data only, synthetic data only, and synthetic pretraining followed by fine-tuning on real data. The evaluation uses two real-world datasets, collected at Costco and Noovelia, at varying training-set sizes. It includes offline experiments, robustness tests, cross-site validation, simulation, and physical-robot integration.

The results confirm the value of this approach. With 31 annotated real images, the hybrid strategy achieves 87.9 % mean average precision at an intersection-over-union threshold of 0.5 (mAP@0.5), compared with 52.2 % for real data alone. With 210 images, it reaches 91.8 %, compared with 85.2 %, showing that the gains diminish as the volume of real data increases. Robot integration at Noovelia confirms that the detector outputs are usable by the navigation system. It also validates the platform for adapting the solution to new clients while improving deployment flexibility and speed.

Keywords: pallet detection, synthetic data, synthetic-to-real transfer, keypoint estimation, YOLO-Pose, autonomous mobile robot.

Avant-propos

Ce mémoire de maîtrise a été réalisé au sein de l'Institut de recherche sur l'hydrogène de l'Université du Québec à Trois-Rivières, en collaboration avec la société Noovelia, spécialisée dans le développement de solutions de robots mobiles autonomes.

Je tiens à exprimer ma sincère gratitude à mes directeurs de recherche. Je remercie le Pr Sousso Kelouwani pour son encadrement scientifique, ses conseils précieux et sa vision stratégique tout au long de ce projet. Je remercie également le Dr Ali Akrem Amamou pour son expertise technique, son accompagnement méthodologique et sa disponibilité constante, ainsi que M. Lotfi Zeghmi pour le lien établi avec les applications industrielles et les perspectives pratiques qu'il a apportées à ce travail.

Je souhaite également remercier l'équipe de Noovelia pour sa collaboration, l'accès à ses équipements et le partage de son expertise dans le domaine des chariots élévateurs autonomes. Mes remerciements s'adressent également à l'Institut de recherche sur l'hydrogène pour les ressources matérielles mises à ma disposition et pour l'environnement de recherche stimulant, ainsi qu'à l'Université du Québec à Trois-Rivières pour le cadre académique et le soutien institutionnel offerts tout au long de ma formation.

Enfin, je remercie chaleureusement ma famille et mes proches pour leur soutien indéfectible, leurs encouragements et leur présence tout au long de cette aventure académique.

Ibrahim BOUBAKRI

Table des matières

Résumé iii

Abstract iv

Avant-propos v

Table des matières vi

Liste des tableaux xi

Liste des figures xiii

Liste des symboles et acronymes xvi

Chapitre 1 - Introduction 1

1.1 Contexte général 1

1.2 Objectifs et hypothèses de recherche 4

1.3 Contributions principales 5

1.4 Organisation du mémoire 6

Chapitre 2 - Revue de la littérature 7

2.1 Introduction 7

2.2 Modalités de perception : couleur (RGB) ou couleur et profondeur (RGB-D) 7

2.3 Apprentissage profond pour la détection et l'estimation de pose 9

2.3.1 Architectures de détection d'objets 9

2.3.2 Estimation de pose et détection par points clés 11

2.3.3 Trois générations YOLO-Pose évaluées dans ce mémoire 11

2.4 Manque de données et transfert synthétique-réel 13
2.5 Synthèse et positionnement de notre approche 15
2.6 Conclusion 19

Chapitre 3 - Plateforme de génération de données synthétiques et de simulation 20
3.1 Moteur de génération de données synthétiques 20
3.1.1 Objectif et proposition de valeur 20
3.1.2 Pipeline de génération 21
3.1.3 Environnement virtuel de génération 22
3.1.4 Moteur de diversification visuelle 22
3.1.5 Production automatisée des images synthétiques 24
3.1.6 Moteur d'annotation automatique 24
3.1.7 Jeu de données produit 25
3.1.8 Transfert technologique du générateur à Noovelia 28
3.2 Architecture de la plateforme Simulateur de génération de données synthétiques 28
3.2.1 Architecture globale et composants 28
3.2.2 Moteur d'intelligence artificielle et gestion des modèles 30
3.2.3 Intégration du module de contrôle de la Chaire de recherche Noovelia 31
3.2.4 Moteur de simulation robotique PyBullet 32
3.2.5 Couche de services et communication en temps réel 33
3.2.6 Tableau de bord web Simulateur de génération de données synthétiques 33
3.2.7 Chaîne opérationnelle de bout en bout 35
3.2.8 Évolutivité, transfert et continuité technologique 35

3.2.9 Synthèse du chapitre 36

Chapitre 4 - Analyse expérimentale en environnement industriel 37

4.1 Positionnement méthodologique : cadre CRISP-DM 37

4.2 Vue d'ensemble de la démarche 40

4.2.1 Robot ROSMASTER M3 Pro 41

4.2.2 Phase 1 : Costco (simulation et hors ligne) 43

4.2.3 Phase 2 : Noovelia (validation embarquée en temps réel sur robot) 44

4.3 Données synthétiques et réelles 45

4.3.1 Jeux de données réels 45

4.3.2 Jeu de données synthétique 46

4.3.3 Protocole de validation complémentaire inter-sites 46

4.4 Annotation manuelle des données réelles et convention topologique 47

4.5 Règles strictes contre la fuite de données 48

4.6 Modèles évalués 49

4.7 Stratégies d'apprentissage comparées 49

4.7.1 Méthode Synthétique puis Réel 49

4.7.2 Augmentation des données réelles et hyperparamètres d'entraînement 53

4.8 Environnement matériel et logiciel 54

4.8.1 Environnement matériel 54

4.8.2 Environnement logiciel 54

4.8.3 Métriques de détection d'objets 56

4.8.4 Métriques d'estimation de pose (points clés) 57

4.8.5 Métriques d'efficience et de latence 58

4.9 Métriques d'évaluation de l'alignement cible 58

4.9.1 Calcul des erreurs d'alignement 60
4.9.2 Choix des tolérances d'alignement 61
4.10 Synthèse du chapitre 61

Chapitre 5 - Résultats et discussion 63
5.1 Apport du transfert synthétique vers réel 63
5.1.1 Effet de la quantité de données réelles disponibles 63
5.1.2 Résultats de la stratégie Synthétique seul 65
5.1.3 Effet du volume de données synthétiques 66
5.1.4 Synthèse de l'effet du volume réel 68
5.2 Comparaison des architectures sur le jeu de données Costco 68
5.2.1 Résultats qualitatifs de détection 68
5.2.2 Compromis entre précision et latence 69
5.2.3 Évolution des performances selon l'échelle 71
5.3 Confirmation sur le jeu de données Noovelia et étude de robustesse 73
5.3.1 Résultats qualitatifs avec la caméra du robot 73
5.3.2 Compromis entre précision et latence sur le jeu de test Noovelia ... 74
5.3.3 Distribution des conditions difficiles 75
5.3.4 Robustesse sous perturbations environnementales 76
5.4 Résultats inter-sites 77
5.4.1 Comparaison des deux directions de transfert 77
5.4.2 Analyse des gains apportés par le préentraînement synthétique 79
5.4.3 Persistance de l'asymétrie entre les deux directions 79
5.4.4 Comparaison avec les performances intra-site 80
5.5 Validation applicative en boucle fermée 81
5.5.1 Compatibilité temporelle sous PyBullet 81

5.5.2 Précision de l'alignement en simulation 81
5.5.3 Sessions sur robot réel 82
5.5.4 Détection et latence embarquées 83
5.5.5 Précision géométrique au point cible sur robot réel 84
5.6 Synthèse des hypothèses et des jalons 86
5.7 Limites et menaces à la validité 86
5.8 Recommandations pour l'intégration industrielle 88
5.9 Synthèse du chapitre 88

Chapitre 6 - Conclusion et perspectives 89
6.1 Rappel de la problématique 89
6.2 Résumé de la méthode déployée 89
6.3 Condensé des conclusions scientifiques 89
6.4 Contributions et portée du travail 90
6.5 Limites du projet 91
6.6 Perspectives de recherche 91

Annexe A - Résultats numériques complets et hyperparamètres 97
A.1 Tableaux exhaustifs des configurations d'entraînement 97
A.1.1 Jeu de test Costco 97
A.1.2 Jeu de test Noovelia 99
A.2 Hyperparamètres détaillés de l'entraînement 101
A.3 Résultats de robustesse détaillés 104
A.4 Comparaison des architectures YOLO-Pose 105

Liste des tableaux

Tableau 2-1 Synthèse critique et comparative des travaux phares sur la détection et la localisation de palettes. 15

Tableau 3-1 Principales variations appliquées pendant la génération synthétique. 23

Tableau 4-1 Spécifications techniques du ROSMASTER M3 Pro 43

Tableau 4-2 Répartition structurelle des jeux de données réels et synthétiques par site. 45

Tableau 4-3 Matrice des modèles YOLO-Pose et des échelles évaluées. 49

Tableau 5-1 Stratégie Synthétique seul : performances par volume synthétique 66

Tableau 5-2 Distribution du jeu de test Noovelia par classe de perturbation. 76

Tableau 5-3 Validation inter-sites des architectures Nano. 78

Tableau 5-4 Compatibilité temporelle sous PyBullet 81

Tableau 5-5 Mesures d'alignement en simulation 82

Tableau 5-6 Résumé des essais de détection et d'alignement sur robot réel 83

Tableau 5-7 Efficience embarquée 84

Tableau 5-8 Résultats d'alignement géométrique sur robot réel 84

Tableau 5-9 Synthèse des réponses aux questions expérimentales. 86

Tableau A.1 Performances détaillées du modèle yolo26n-pose sur le jeu de test Costco. 99

Tableau A.2 Performances détaillées du modèle yolo26n-pose sur le jeu de test Noovelia. 101

Tableau A.3 Hyperparamètres d'entraînement et d'augmentation d'images. 103

Tableau A.4 Performances de robustesse sous perturbations pour Costco (modèle yolo26n-pose, $n = 3$ répétitions). 104
Tableau A.5 Performances de robustesse sous perturbations pour Noovelia (modèle yolo26n-pose, $n = 3$ répétitions). 105
Tableau A.6 Comparaison des performances des architectures YOLO-Pose (Nano et Small) sur les jeux de test Costco et Noovelia. 106
Tableau A.7 Comparaison multi-échelles des architectures YOLO-Pose (stratégie hybride, jeu de test Costco). 107

Liste des figures

Figure 1.1 Vue latérale du chariot élévateur autonome de Noovelia. 2

Figure 1.2 Vue de face du chariot élévateur autonome de Noovelia. 3

Figure 2.1 Structure de l'architecture YOLOv8 [1]. 10

Figure 3.1 Exemples d'images synthétiques et de variations produites avec Blender (1/2). 26

Figure 3.2 Exemples d'images synthétiques et de variations produites avec Blender (2/2). 27

Figure 3.3 Architecture générale de la plateforme Simulateur de génération de données synthétiques 29

Figure 3.4 Organisation fonctionnelle du moteur d'intelligence artificielle de Simulateur de génération de données synthétiques. 30

Figure 3.5 Organisation fonctionnelle du tableau de bord web Simulateur de génération de données synthétiques pour la détection et la simulation. 34

Figure 4.1 Adaptation par l'auteur du cycle CRISP-DM au protocole expérimental du mémoire, d'après [2]. 40

Figure 4.2 Schéma synoptique de la démarche expérimentale hybride (du synthétique vers le réel). 41

Figure 4.3 Vue d'ensemble du robot ROSMASTER M3 Pro. 42

Figure 4.4 Vue d'ensemble du robot ROSMASTER M3 Pro déployé chez Noovelia. 44

Figure 4.5 Schéma du protocole de partitionnement fractionné 52

Figure 4.6 Interface utilisateur de l'API web connectée au simulateur de test PyBullet. 55
Figure 4.7 Schéma géométrique des variables d'erreur mesurées au point cible. 59

Figure 5.1 Ablation sur le jeu de test Costco : mAP@0,5 selon les images réelles et le volume synthétique 64
Figure 5.2 Ablation sur le jeu de test Costco : OKS selon les images réelles et le volume synthétique 64
Figure 5.3 Impact du volume de préentraînement synthétique 67
Figure 5.4 Synthèse mAP et OKS (yolo26n-pose) selon le volume réel 68
Figure 5.5 Exemples de détection sur des images réelles du jeu de test Costco. 69
Figure 5.6 Compromis entre précision de détection et latence GPU sur le jeu de test Costco. 70
Figure 5.7 Compromis entre qualité d'estimation de pose et latence GPU sur le jeu de test Costco. 71
Figure 5.8 Précision de détection selon la famille et l'échelle du modèle sur le jeu de test Costco. 72
Figure 5.9 Qualité d'estimation de pose selon la famille et l'échelle du modèle sur le jeu de test Costco. 72
Figure 5.10 Exemples de détection sur des images réelles du jeu de test Noovelia captées par la caméra embarquée. 73
Figure 5.11 Compromis entre précision de détection et latence GPU sur le jeu de test Noovelia. 74
Figure 5.12 Compromis entre qualité d'estimation de pose et latence GPU sur le jeu de test Noovelia. 75

Figure 5.13 Robustesse de yolo26n-pose sous perturbations environnementales sur le jeu de test Noovelia. 77
Figure 5.14 Performances inter-sites des architectures Nano. 78

Liste des symboles et acronymes

Acronymes

AGV	Automated Guided Vehicle (véhicule à guidage automatique)
AMR	Autonomous Mobile Robot (robot mobile autonome)
API	Application Programming Interface (interface de programmation d'application)
BBox	Bounding Box (boîte englobante)
CIoU	Complete Intersection over Union
CNN	Convolutional Neural Network (réseau de neurones convolutif)
CRISP-DM	Cross-Industry Standard Process for Data Mining
CUDA	Compute Unified Device Architecture
CVAT	Computer Vision Annotation Tool
DFL	Distribution Focal Loss
EPnP	Efficient Perspective-n-Point
FPS	Frames Per Second (images par seconde)
GMA	Grocery Manufacturers Association
GPU	Graphics Processing Unit (processeur graphique)
HSV	Hue, Saturation, Value (teinte, saturation, valeur)
IA	Intelligence artificielle
IoU	Intersection over Union
mAP	Mean Average Precision
NMS	Non-Maximum Suppression (suppression non maximale)
OKS	Object Keypoint Similarity
PCK	Percentage of Correct Keypoints
PnP	Perspective-n-Point

RGB	Red, Green, Blue (rouge, vert, bleu)
RGB-D	Red, Green, Blue - Depth (rouge, vert, bleu - profondeur)
ROS 2	Robot Operating System 2
SVM	Support Vector Machine (séparateur à vaste marge)
ToF	Time of Flight (temps de vol)
YOLO	You Only Look Once

Symboles

Symbole	Description	Unité
P	Précision	—
R	Rappel	—
AP	Average Precision	—
mAP	Mean Average Precision	—
IoU	Intersection over Union	—
lr_0	Taux d'apprentissage initial	—
ΔX_{align}	Désalignement latéral	cm
ΔY_{align}	Erreur de profondeur de garde	cm
$\Delta\theta_{align}$	Défaut de parallélisme / angulaire	degrés
P_{target}	Point cible (Target Point)	—
X_R, Y_R	Repère local du robot	—

Chapitre 1 - Introduction

1.1 Contexte général

L'automatisation des opérations logistiques constitue un axe majeur de transformation des entrepôts, des centres de distribution et des sites manufacturiers. Afin d'améliorer la flexibilité, la sécurité et la productivité des opérations de manutention, les entreprises intègrent progressivement des robots mobiles, des systèmes de perception et des méthodes d'apprentissage automatique [3, 4].

Parmi ces plateformes, les robots mobiles autonomes et les chariots élévateurs autonomes doivent naviguer dans des environnements industriels dynamiques, éviter les obstacles et interagir avec différents objets logistiques. Dans ce mémoire, le terme *environnement dynamique* ne désigne pas uniquement le mouvement du robot ou une scène en transformation continue. Il regroupe les variations visuelles et opérationnelles susceptibles d'affecter la détection : changements d'éclairage, ombres, occlusions temporaires, présence d'opérateurs ou d'objets mobiles, encombrement, vibrations de la caméra et variations du point de vue. Une palette est un support logistique destiné au stockage et à la manutention de marchandises, muni d'ouvertures permettant sa prise par les fourches d'un chariot élévateur. Dans le cas d'un chariot élévateur autonome, l'une des opérations les plus critiques est la navigation d'approche vers une palette. Le robot doit détecter la palette cible avec précision afin de fournir au système aval les informations visuelles nécessaires à l'approche.

La palette représente ainsi un élément central de la chaîne de navigation. Sa détection influence directement les étapes suivantes, notamment l'estimation de pose, la planification de la trajectoire et le contrôle du mouvement. Cependant, cette tâche demeure difficile en environnement réel, puisque les palettes peuvent être usées, chargées, partiellement occultées, faiblement éclairées ou observées sous des angles variables.

Cette problématique s'inscrit directement dans les activités de la **Chaire de recherche Noovelia**, spécialisée dans le développement de solutions d'automatisation et de robots mobiles autonomes. Au cours des cinq premières années de la Chaire, les travaux ont principalement porté sur le développement des plateformes robotiques, de leur système de navigation autonome et des différents modules nécessaires à leur fonctionnement en environnement industriel.

Ces travaux ont notamment contribué au développement de robots mobiles et de chariots élévateurs autonomes, dont certaines solutions sont aujourd'hui intégrées aux produits commercialisés par Noovelia. Les figures 1.1 et 1.2 présentent le chariot élévateur autonome utilisé dans le cadre de ces développements.



Figure 1.1 Vue latérale du chariot élévateur autonome de Noovelia.

À l'état actuel du projet, les principales fonctions de navigation du robot sont déjà développées et utilisées dans des applications industrielles. Toutefois, l'amélioration de la perception visuelle demeure nécessaire, particulièrement pour la détection des palettes lors de la navigation d'approche. La performance du système dépend alors de sa capacité à reconnaître différents types de palettes dans des conditions qui peuvent varier considérablement d'un client à l'autre.



Figure 1.2 Vue de face du chariot élévateur autonome de Noovelia.

L'une des principales difficultés rencontrées concerne le manque de données réelles annotées et suffisamment diversifiées. Lors du déploiement chez un nouveau client, les types de palettes, l'éclairage, l'organisation de l'entrepôt et les conditions d'observation peuvent changer. Il devient alors nécessaire de collecter et d'annoter de nouvelles images, ce qui demande du temps et des ressources et ralentit l'adaptation de la solution.

Dans ce contexte, ce mémoire propose une méthode de génération automatique de données synthétiques réalistes et facilement paramétrables. Ces données visent à compléter les données réelles disponibles et à faciliter l'entraînement des modèles de détection pour de nouveaux environnements. Un environnement de simulation est aussi développé pour évaluer rapidement et de manière sécuritaire les modèles entraînés avant leur intégration sur la solution réelle.

Le mémoire étudie aussi le transfert de connaissances entre les données synthétiques et les données réelles. L'objectif est de déterminer dans quelle mesure les données générées peuvent réduire la quantité de données réelles annotées nécessaire, tout en maintenant des performances suffisantes pour alimenter la navigation d'approche. Cette démarche vise ainsi à accélérer l'adaptation de la solution de Noovelia aux besoins et aux conditions propres à chaque nouveau client.

La frontière fonctionnelle de la contribution est la sortie du module de perception. Le système développé détecte et localise la palette dans l'image au moyen d'une boîte englobante 2D et, dans la configuration *You Only Look Once Pose* (YOLO-Pose) étudiée, de six points clés. À partir de cette sortie, l'estimation géométrique de la pose, la planification de trajectoire, la commande du robot et l'insertion des fourches sont prises en charge par des modules existants au sein de l'équipe de la Chaire de recherche Noovelia. Ces traitements sont utilisés pour vérifier l'utilité applicative de la détection, mais leur conception algorithmique demeure hors du périmètre de ce mémoire.

1.2 Objectifs et hypothèses de recherche

Cette étude évalue l'apport des données synthétiques pour la détection de palettes industrielles à partir d'images en rouge, vert et bleu (RGB), dans un contexte où les données réelles annotées sont rares. L'objectif n'est pas seulement d'entraîner un détecteur, mais de mesurer l'effet du transfert synthétique-réel sur la précision, la robustesse et le besoin en données réelles. Les stratégies s'appuient sur des poids initialisés à partir de Common Objects in Context (COCO), un grand jeu de données de référence en vision par ordinateur utilisé ici pour initialiser les poids préentraînés des modèles. Dans la suite, COCO-Pose désigne plus précisément les poids liés à la tâche de pose et au transfert des points clés.

Les hypothèses de recherche sont les suivantes :

- **H1 : Apport du transfert synthétique-réel :** à quantité égale de données réelles annotées, la stratégie COCO puis Synthétique puis Réel améliore les performances de détection et de localisation des points clés par rapport à la stratégie COCO puis Réel

- **H2 : Apport de l'entraînement hybride à la robustesse :** l'entraînement incluant des données synthétiques randomisées est associé à de meilleures performances sous occlusion, faible luminosité, angles de vue difficiles et encombrement, par rapport à l'entraînement sur données réelles seules
- **H3 : Rendement décroissant des données réelles :** après un préentraînement synthétique, l'ajout de données réelles annotées produit des gains importants aux faibles volumes, puis des gains progressivement plus faibles au-delà d'un seuil observé expérimentalement
- **H4 : Rendement décroissant du volume synthétique :** l'augmentation du volume de données synthétiques améliore les performances jusqu'à une zone de saturation, au-delà de laquelle les gains deviennent marginaux dans le protocole étudié

1.3 Contributions principales

Dans ce mémoire, un *site* désigne un environnement industriel réel d'acquisition associé à un domaine visuel et à un jeu de données distinct. Les deux sites étudiés sont Costco, environnement réel d'acquisition dont provient le premier jeu de données, et Noovelia, environnement réel d'acquisition du second.

Les contributions principales de ce mémoire comprennent une méthodologie de génération de données synthétiques et d'annotation automatique pour la détection de palettes industrielles, ainsi qu'une étude expérimentale d'un préentraînement synthétique suivi d'un ajustement sur des données réelles. Le travail quantifie l'effet du volume de données réelles sur les performances, complète l'analyse par une validation inter-sites dans les directions Costco vers Noovelia et Noovelia vers Costco, puis évalue la robustesse dans différentes conditions industrielles, d'abord dans un simulateur et ensuite sur un robot de validation.

1.4 Organisation du mémoire

Le reste du mémoire est organisé comme suit. Le chapitre 2 présente l'état de l'art : revue de la perception visuelle de palettes, des modalités de capteurs, des méthodes d'apprentissage profond, du manque de données annotées en contexte industriel, du recours aux données synthétiques, ainsi que la lacune scientifique identifiée. Le chapitre 3 décrit le pipeline de génération de données synthétiques : modélisation des scènes, randomisation de domaine et annotation automatique. Le chapitre 4 présente la méthodologie : jeux réels, architectures de détection, stratégies d'entraînement, métriques et protocoles de validation, dont l'évaluation complémentaire inter-sites. Le chapitre 5 regroupe les résultats et leur discussion dans une progression commune, depuis les études d'ablation jusqu'aux essais sur robot réel, puis examine les limites et les implications industrielles. Enfin, le chapitre 6 conclut le mémoire et présente les perspectives de recherche, notamment l'extension de la validation à d'autres sites et à d'autres types de palettes.

Chapitre 2 - Revue de la littérature

2.1 Introduction

L'automatisation des entrepôts logistiques s'appuie de plus en plus sur des véhicules à guidage automatique (AGV) et des chariots élévateurs autonomes [3, 5]. Dans un environnement industriel dynamique tel que celui de notre partenaire Noovelia, la capacité du robot à percevoir son environnement, à détecter les palettes et à s'en approcher avec précision constitue un enjeu majeur de sécurité et de productivité. Deux revues systématiques récentes confirment d'ailleurs que, malgré des performances de laboratoire remarquables, la manipulation fiable de la charge demeure l'un des verrous les moins résolus du domaine [5, 6].

Ce chapitre analyse l'état de l'art de la détection et de la localisation de palettes. Nous examinons d'abord les modalités de capteurs (section 2.2), puis les approches d'apprentissage profond pour la détection et l'estimation de pose (section 2.3), avant d'aborder le défi de la rareté des données réelles annotées, qui motive l'usage de données synthétiques et du transfert synthétique-réel (section 2.4). La section 2.5 synthétise ces travaux dans un tableau comparatif, dégage les verrous encore ouverts et positionne notre approche.

2.2 Modalités de perception : couleur (RGB) ou couleur et profondeur (RGB-D)

La localisation des palettes et de leurs ouvertures d'insertion requiert une perception géométrique précise. Historiquement, de nombreux travaux se sont appuyés sur des caméras de profondeur (RGB-D), des caméras 3D ou des capteurs laser. Molter et Fottner [7] extraient des caractéristiques géométriques d'une caméra à temps de vol (ToF) pour localiser la palette en temps réel (15 images/s), y compris sous un angle de 90 degrés. Mohamed et al. [8] combinent un détecteur Faster R-CNN et un filtrage temporel sur des données laser

2D. Xiao et al. [9] reconnaissent et localisent la palette par correspondance de gabarits sur des données RGB-D. Zhao et al. [10] prolongent cette idée par un appariement de gabarits étiquetés, où chaque pixel est d'abord classé par un séparateur à vaste marge (SVM) : leur système atteint un taux de détection de 92,6 % en entrepôt, au prix d'un calcul lourd sur le nuage de points et d'une sensibilité à l'angle et à la distance. Iinuma et al. [11] segmentent sémantiquement la palette avec une caméra RGB-D (réseau ESPNet) et estiment sa position et son orientation pour l'insertion des fourches ; l'approche est validée expérimentalement sur un chariot en extérieur, sur sol incliné. Plus récemment, Fang et al. [12] fusionnent l'image RGB et le nuage de points 3D pour recadrer dynamiquement la face avant de la palette, ramenant l'erreur angulaire sous 0,4 degré.

Ces résultats confirment que les capteurs 3D offrent une géométrie détaillée. En contrepartie, ils augmentent le coût matériel, la complexité de calibration et la charge de calcul [5, 11, 12], et restent sensibles aux surfaces réfléchissantes, à la poussière et aux conditions lumineuses. C'est pourquoi l'usage de caméras RGB monoculaires suscite un vif intérêt industriel : elles offrent une solution peu coûteuse et facile à déployer, même si l'estimation de la distance et de l'orientation à partir d'une seule image 2D demeure un défi. Certaines approches contournent partiellement ce problème par des matrices d'homographie calibrées, comme celle de Vodichev et al. [13] (erreur théorique de 0,05 à 0,21 m selon la distance), mais restent limitées face aux occlusions causées par la marchandise. D'autres, comme Kesuma et al. [14], atteignent une excellente précision de distance (erreur moyenne de 2,28 cm avec YOLOv5n à 24 images/s) en collant des marqueurs fiduciaires ArUco sur les palettes, une contrainte opérationnelle difficilement acceptable à l'échelle d'un entrepôt réel.

Implication pour ce mémoire. La contrainte industrielle du projet impose une solution simple, économique et sans instrumentation des palettes. Nous faisons donc le choix assumé d'une caméra RGB seule, sans capteur de profondeur ni marqueur artificiel, et compensons l'absence de mesure 3D directe par une détection visuelle plus riche, fondée sur l'apprentissage profond et des points clés structurels.

2.3 Apprentissage profond pour la détection et l'estimation de pose

Avec l'avènement de l'Industrie 4.0, les approches fondées sur l'apprentissage profond ont supplanté les méthodes de vision classique (contours, gabarits, formes géométriques), trop sensibles aux variations d'apparence et d'éclairage [4, 15].

2.3.1 Architectures de détection d'objets

Deux familles dominent la détection d'objets : les détecteurs en deux étapes, comme Faster R-CNN [16], précis mais coûteux en calcul, et les détecteurs en une étape, comme la famille YOLO (*You Only Look Once*) [17–19], particulièrement prisée pour sa rapidité en temps réel et son déploiement aisé sur des systèmes embarqués. Les versions récentes [20–22] consolident ce statut de référence.

Les détecteurs modernes partagent une organisation commune en trois parties, désignées ici par les termes anglais *backbone*, *neck* et *head* [23]. Le *backbone* extrait les caractéristiques de l'image d'entrée à différentes échelles, le *neck* les agrège et les raffine, et le *head* produit les prédictions finales. Le modèle YOLOv8, utilisé comme base de comparaison dans ce mémoire, reprend cette structure d'ensemble : son *backbone* extrait les caractéristiques via les modules CBS, C2f et SPPF, son *neck* fusionne les informations multi-échelles et son *head* découplé délivre les sorties de détection. La variante YOLOv8-Pose utilisée dans ce travail conserve les mêmes *backbone* et *neck*, et ajoute au *head* une

branche de régression des points clés. La figure 2.1, adaptée de Xu et al. [1] (sous licence CC BY 4.0), illustre cette organisation architecturale.

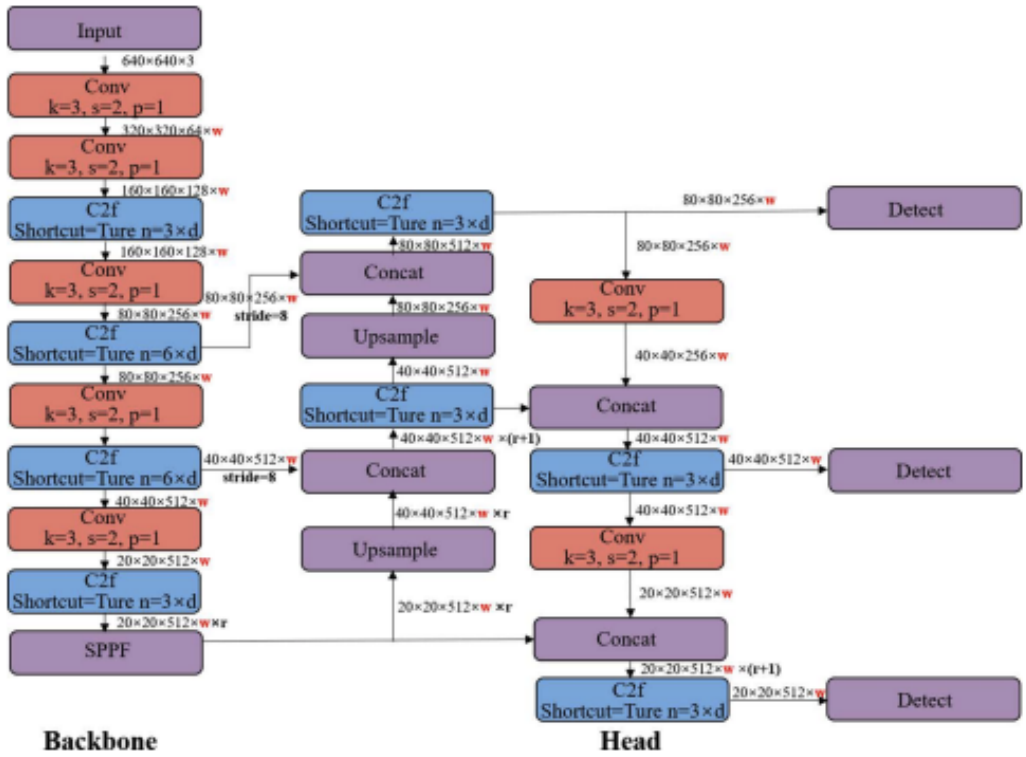


Figure 2.1 Structure de l'architecture YOLOv8 [1].

Appliquée aux palettes, cette famille a fait ses preuves. Zaccaria et al. [24] comparent plusieurs détecteurs profonds en entrepôt. Manurung et al. [25] montrent que YOLOv5 offre un excellent compromis : entraîné sur 1 350 images réelles, le modèle allégé YOLOv5s atteint 24 images par seconde sur un système embarqué, mais sans estimation de distance ni de pose. D'autres chercheurs proposent des réseaux de segmentation allégés, tel le LCS-Net de Wu et al. [26], qui exploite des mécanismes d'attention spatiale pour isoler les composants de la palette (score moyen d'intersection sur union (*mean Intersection over Union*, mIoU) de 88,1 % avec 1 000 images), au prix toutefois d'une annotation manuelle pixel par pixel.

2.3.2 Estimation de pose et détection par points clés

Si la détection de la boîte englobante 2D constitue la première étape, la navigation d'approche nécessite une information d'orientation [15]. Une voie légère consiste à prédire des *points clés* structurels de la palette puis à en déduire l'orientation par correspondance 2D–3D de type *Perspective-n-Point* (PnP) [27]. Dans une architecture YOLO-Pose, cette capacité est obtenue en ajoutant au *head* de détection une branche supplémentaire dédiée à la régression des coordonnées des points clés, sans modifier le *backbone* ni le *neck* [1]. Zhou et al. [15] ont récemment adapté un modèle d'estimation de pose humaine (yolov8s-pose) à la structure de la palette : en repérant les 12 points clés définissant la section en E de la palette, leur système rapporte 94,4 % de précision de détection et 93,2 % de précision de localisation des points clés à 104 images par seconde. Ce résultat suggère qu'un réseau peut apprendre la géométrie intrinsèque de l'objet. La limite est explicite : l'annotation manuelle de points clés sur des milliers d'images réelles représente un travail considérable.

Implication pour ce mémoire. Nous retenons la famille YOLO-Pose pour sa rapidité embarquée et étudions deux types de sorties : la boîte englobante pour localiser la cible, et six points clés sur la face visible pour caractériser son orientation apparente. L'annotation des points clés étant précisément le poste de coût le plus lourd, elle sera obtenue automatiquement sur les données synthétiques.

2.3.3 Trois générations YOLO-Pose évaluées dans ce mémoire

Ce mémoire confronte trois générations de la famille YOLO-Pose pour mesurer l'effet de leur évolution sur la détection et l'estimation de pose de palettes. Toutes partagent l'organisation *backbone–neck–head* décrite ci-dessus et illustrée à la figure 2.1, à laquelle

s'ajoute une branche de régression des points clés. Le choix de ces trois générations se justifie par leurs différences architecturales, qui permettent d'observer comment les avancées récentes influencent la qualité de détection et la compatibilité avec un déploiement embarqué.

YOLOv8-Pose Cette génération introduit une architecture sans ancrage et un *head* découplé [19]. Elle a établi un standard de référence pour la détection et l'estimation de pose en temps réel, et constitue la base de comparaison dans ce mémoire.

YOLO11-Pose Cette version améliore l'extraction de caractéristiques et la régression des coordonnées [22]. Elle élève la qualité de l'estimation de pose à un coût de calcul comparable à celui de YOLOv8-Pose, ce qui en fait un candidat pertinent pour le déploiement embarqué.

YOLO26-Pose Cette génération, la plus récente évaluée dans ce mémoire, est orientée vers le déploiement embarqué. Parmi ses changements clés figurent l'inférence de bout en bout sans suppression non maximale et des améliorations architecturales visant une exécution plus rapide sur du matériel contraint [28]. Les caractéristiques exactes de la version utilisée sont décrites dans la section 4.8.

Le choix exclusif de la famille YOLO-Pose répond à un compromis méthodologique et applicatif, et non à l'hypothèse d'une supériorité universelle. Faster R-CNN [16], DETR [29] et RT-DETR [30] constituent des solutions pertinentes pour la détection d'objets, mais le présent protocole exige simultanément une charge de calcul compatible avec le Jetson embarqué, une cadence temps réel et la prédiction de six points clés en plus de la boîte englobante. Dans l'écosystème retenu, les variantes YOLO-Pose fournissent nativement cette sortie conjointe par l'entremise d'Ultralytics, la bibliothèque logicielle utilisée dans

ce mémoire pour l'entraînement, l'évaluation et le déploiement des modèles YOLO-Pose. Elles s'intègrent à la chaîne de déploiement Jetson ainsi qu'à Robot Operating System 2 (ROS 2), l'intergiciel utilisé pour l'intégration robotique. Évaluer d'autres familles aurait donc nécessité d'adapter ou d'ajouter un *head* de points clés et de revalider leur chaîne de déploiement, ce qui aurait introduit des facteurs logiciels distincts dans la comparaison. Le maintien de YOLOv8-Pose, YOLO11-Pose et YOLO26-Pose dans une même version du framework permet au contraire de comparer plusieurs générations dans un environnement logiciel contrôlé. Cette restriction constitue néanmoins une limite de l'étude : aucune conclusion de supériorité inter-familles ne peut être tirée, et l'entraînement de détecteurs externes comparables demeure une perspective de travaux futurs plutôt qu'une exigence du protocole retenu.

2.4 Manque de données et transfert synthétique-réel

Malgré ces avancées, un obstacle majeur persiste : la dépendance de ces algorithmes à de grandes bases de données réelles annotées [12, 15, 26]. Dans le contexte industriel de Noovelia, capturer puis annoter manuellement des milliers d'images de palettes couvrant toutes les variations (éclairage, usure, occlusions, angles de vue) est trop coûteux et chronophage, de l'ordre de 35 secondes par image pour une annotation simple [31], davantage lorsque des points clés doivent être placés.

Pour lever ce verrou, l'usage de données synthétiques s'impose comme une solution naturelle. Pilotés par script, des moteurs de rendu comme Blender [32] permettent de générer des milliers d'images photoréalistes avec des annotations géométriquement cohérentes (boîtes englobantes, masques et points clés). Le principe de la randomisation de domaine (*domain randomization*), qui consiste à varier massivement l'éclairage, les textures, les

poses et les arrière-plans, force le modèle à apprendre des caractéristiques invariantes et réduit l'écart entre simulation et réalité [33, 34].

Appliquée aux palettes, cette stratégie a produit des résultats convaincants. Knitt et al. [35] estiment la pose 6D d'une palette Euro avec une caméra RGB, à partir d'un détecteur entraîné uniquement sur des images synthétiques. Beleznai et al. [36] apprennent des indices géométriques synthétiques pour la détection et la pose 3D. Gann et al. [37] gagnent jusqu'à 69 et 50 points de mAP50 (Mean Average Precision à 50 %) sur les catégories de palettes empilées et rayonnées grâce à des images synthétiques générées sous un moteur de jeu, tout en observant une chute marquée en faible luminosité. Mueller et al. [31] atteignent un mAP50 de 0,995 sur une palette isolée avec un entraînement purement synthétique. Enfin, le paradigme de transfert synthétique-réel, c'est-à-dire un préentraînement sur des environnements virtuels randomisés puis un ajustement (*fine-tuning*) sur un petit jeu de données réelles, a été validé dans l'industrie générale par Baumgart et al. [38].

Il reste cependant une question que ces travaux n'abordent pas : *quelle quantité* de données réelles est réellement nécessaire après un préentraînement synthétique ? Les études existantes se placent soit en entraînement purement synthétique [31], soit avec un volume de données réelles confortable [15]. Le dosage entre coût d'annotation et performance, l'information dont une entreprise a précisément besoin pour dimensionner son effort, n'est pas quantifié pour la détection de palettes.

Implication pour ce mémoire. Nous adoptons une démarche qui combine la génération automatique de données synthétiques annotées, l'entraînement de modèles de détection sur ces données, puis leur ajustement sur des données réelles. L'objectif est de quantifier l'effet de la disponibilité limitée de données réelles sur la performance finale.

2.5 Synthèse et positionnement de notre approche

Le tableau 2-1 résume l'évolution des techniques de perception pour les chariots élévateurs autonomes. Si les capteurs coûteux (RGB-D, 3D, laser) ou les marqueurs artificiels ont permis les premières avancées, la tendance actuelle s'oriente vers des réseaux profonds n'utilisant que des caméras RGB. Le tableau croise, pour chaque travail, la modalité de capteur, le type de données d'entraînement, les performances clés rapportées et les limites majeures. Un grand X (**X**) indique que la dimension est couverte.

Tableau 2-1 Synthèse critique et comparative des travaux phares sur la détection et la localisation de palettes.

Référence	**Méthode**	**Capteur**		**Données**		**Performances clés**	**Limites majeures**
		RGB	**Profondeur**	**Réel**	**Synthétique**		
Molter, Fottner (2018) [7]	Caractéristiques géométriques 3D (ToF)		**X**	**X**		Temps réel (15 img/s), fonctionne à 90°	Vulnérable au bruit du capteur ; dépendance totale à la géométrie 3D
Iinuma et al. (2021) [11]	Segmentation sémantique (ESPNet), RGB-D	**X**	**X**	**X**		Insertion validée en extérieur, sol incliné	Coût du capteur ; sensibilité aux variations lumineuses
Zhao et al. (2022) [10]	Gabarits étiquetés + SVM, RGB-D	**X**	**X**	**X**		Taux de détection de 92,6 %	Calcul lourd du nuage de points ; sensible à l'angle et à la distance
Manurung et al. (2022) [25]	Détection YOLOv5s	**X**		**X**		Rapide (24 img/s), faible charge (1 350 images)	Pas d'estimation de distance ni de pose
Kesuma et al. (2023) [14]	YOLOv5n + marqueurs ArUco	**X**		**X**		Erreur de distance de 2,28 cm	Marqueurs artificiels à poser sur chaque palette

Suite à la page suivante

Tableau 2-1 – suite de la page précédente

Référence	Méthode	Capteur		Données		Performances clés	Limites majeures
		RGB	Profondeur	Réel	Synthétique		
Fang et al. (2024) [12]	YOLO + fusion nuage 3D	**X**	**X**	**X**		Erreur angulaire $< 0{,}4°$	Complexité algorithmique ; capteur 3D onéreux
Gann et al. (2024) [37]	Segmentation, données de synthèse	**X**			**X**	$+69$ et $+50$ points de mAP50 (palettes empilées/rayonnées)	Chute marquée en faible luminosité
Wu et al. (2025) [26]	LCS-Net (segmentation + attention)	**X**		**X**		mIoU de 88,1 % (1 000 images)	Lourde annotation manuelle pixel par pixel
Zhou et al. (2025) [15]	yolov8s-pose, 12 points clés	**X**		**X**		94,4 % de détection à 104 img/s	Annotation manuelle de 12 points clés par palette
Mueller et al. (2025) [31]	Détection 100 % synthétique	**X**			**X**	mAP50 de 0,995 (palette isolée)	Dosage du réel non étudié ; pose frontale seulement
Vodichev et al. (2025) [13]	Homographie monoculaire calibrée	**X**		**X**		Erreur de 0,05 à 0,21 m selon distance	Limité face aux occlusions partielles
Notre approche	**YOLO-Pose (boîte englobante + 6 points clés), préentraînement synthétique + ajustement réel**	**X**		**X**	**X**	**Étude du dosage réel (10–100 % en sept fractions), faible coût matériel, robustesse à l'occlusion recherchée**	**Sensibilité potentielle à l'écart de domaine, à quantifier**

Cette analyse met en évidence une évolution claire des méthodes de perception. Les premières approches reposent surtout sur des capteurs de profondeur, des caméras RGB-D,

des caméras 3D ou des capteurs laser. Elles fournissent une information géométrique précise, mais exigent un matériel plus coûteux, une calibration plus délicate et une puissance de calcul plus élevée [7, 10–12]. Une deuxième famille de travaux utilise des caméras RGB seules : plus simples, plus économiques et plus faciles à déployer. Comme l'image RGB ne donne pas directement la profondeur, l'estimation de la distance et de l'orientation devient plus difficile ; certaines approches compensent cette limite par des marqueurs artificiels [14], au prix d'une modification des palettes rarement réaliste en contexte industriel. Les méthodes récentes fondées sur l'apprentissage profond, notamment YOLO, obtiennent de très bons résultats en temps réel [15, 25], mais la plupart requièrent un grand nombre d'images réelles annotées, une étape coûteuse, surtout lorsqu'il faut tracer des masques pixel par pixel [26] ou des points clés [15]. Les données synthétiques offrent une réponse à ce problème en générant automatiquement des images variées et annotées [31, 35, 37], mais une question reste peu étudiée pour les palettes : quelle quantité de données réelles faut-il ajouter après un entraînement synthétique ?

À partir de cette analyse, quatre verrous principaux se dégagent :

G1 : réduire la dépendance aux capteurs coûteux. Plusieurs solutions précises reposent sur des capteurs de profondeur, des caméras 3D ou des capteurs laser [7, 10–12], ou sur des marqueurs artificiels placés sur les palettes [14]. Ces choix améliorent la précision mais rendent le système plus coûteux et plus difficile à déployer. Nous retenons une caméra RGB seule, sans capteur de profondeur ni marqueur ajouté ; l'orientation est estimée à partir de six points clés prédits par le réseau.

G2 : limiter le coût de l'annotation manuelle. Les approches récentes demandent souvent de nombreuses annotations manuelles : boîtes englobantes, masques pixel par pixel [26] ou points clés [15], ce qui devient vite lourd sur des milliers d'images. Notre

approche génère des images synthétiques sous Blender, dont les boîtes et les points clés sont produits automatiquement ; chaque site réel comprend 300 images annotées, dont 210 sont disponibles pour l'entraînement et l'ajustement fin.

G3 : mesurer la quantité minimale de données réelles nécessaires. Les travaux existants montrent l'utilité des données synthétiques, soit pour entraîner directement un modèle [31, 35], soit pour compléter des données réelles, sans préciser la quantité de réel nécessaire après préentraînement, en particulier pour les palettes. Ce mémoire étudie le transfert synthétique-réel de façon progressive : le modèle est d'abord initialisé depuis les poids COCO, préentraîné sur des données synthétiques, puis ajusté avec différentes fractions des 210 images réelles d'entraînement (10, 15, 30, 50, 70, 80 et 100 %), afin d'évaluer le compromis entre coût d'annotation et performance.

G4 : tester la robustesse dans des conditions proches du réel. La littérature montre que les performances peuvent chuter en conditions difficiles : faible luminosité, occlusions, angles de vue variés ou environnement encombré [37] ; de plus, plusieurs validations restent limitées à des images statiques [6]. Ici, la robustesse est analysée selon l'éclairage, l'occlusion, l'angle de vue et l'encombrement, et la chaîne complète perception–commande est validée en simulation puis sur le robot réel ROSMASTER M3 Pro dans un environnement industriel.

Ainsi, notre travail répond directement à une limite importante de l'état de l'art : proposer une méthode de perception fondée sur une caméra RGB seule, à coût d'annotation réduit, tout en étudiant clairement le passage des données synthétiques vers les données réelles.

L'objectif de ce mémoire n'est pas d'établir un classement universel entre toutes les familles de détecteurs. L'étude vise à isoler l'effet du transfert synthétique-réel dans une

famille de modèles de pose compatible avec l'infrastructure logicielle, les contraintes de déploiement et l'interface de programmation d'application (API) existante. La comparaison est donc menée entre plusieurs générations et plusieurs échelles de YOLO-Pose, selon un protocole d'entraînement contrôlé. Une comparaison avec des architectures externes, telles que des détecteurs à transformeurs ou des méthodes en deux étapes, constituerait une extension pertinente, mais dépasse le périmètre expérimental retenu.

2.6 Conclusion

L'état de l'art révèle que, si les réseaux de neurones profonds ont amélioré la détection de palettes, la rareté des données industrielles annotées freine leur déploiement à grande échelle. Le recours à des caméras RGB, soutenu par la génération de données synthétiques et un transfert synthétique-réel à dosage maîtrisé, constitue une piste pertinente. Le chapitre 3 détaille le pipeline de génération, et le chapitre 4 la méthodologie d'entraînement, d'évaluation et d'intégration applicative.

Chapitre 3 - Plateforme de génération de données synthétiques et de simulation

La disponibilité d'images industrielles correctement annotées représente une difficulté importante pour l'entraînement des modèles de vision. La collecte sur un site réel demande du temps, mobilise du matériel et ne permet pas toujours d'observer suffisamment de situations différentes.

Pour répondre à cette difficulté, la plateforme développée associe une méthode de génération d'images synthétiques de palettes avec leurs annotations à une architecture intégrée qui relie les données, les modèles d'intelligence artificielle (IA), la simulation et l'interface web.

La première section présente le moteur de génération et les variations appliquées aux scènes. La seconde décrit l'architecture de la plateforme et l'intégration de ses différents modules.

3.1 Moteur de génération de données synthétiques

3.1.1 Objectif et proposition de valeur

Le moteur de génération de données synthétiques a pour objectif de produire automatiquement des images de palettes accompagnées d'annotations cohérentes. Il remplace une partie du travail manuel normalement nécessaire pour photographier les palettes et identifier leurs points caractéristiques.

La génération ne cherche pas à copier un entrepôt précis. Elle crée plutôt plusieurs situations visuelles à partir d'une scène contrôlée. Cette approche permet au modèle d'observer

des palettes sous différents angles, avec plusieurs charges et dans des conditions visuelles variées.

Cette méthode correspond au principe de randomisation de domaine [33, 34]. Son objectif est d'éviter que le modèle apprenne uniquement l'apparence d'une scène particulière. Le modèle doit surtout reconnaître la forme générale de la palette et la disposition de ses points caractéristiques.

3.1.2 Pipeline de génération

La génération est réalisée avec Blender, un logiciel libre de création et de rendu de scènes en trois dimensions [32]. Une scène virtuelle contient la palette, le sol, la caméra, les sources lumineuses et les objets pouvant représenter une charge.

Le fonctionnement général suit les étapes suivantes :

1. préparation de la scène virtuelle
2. sélection d'une variante de palette
3. ajout éventuel d'une charge ou de palettes empilées
4. choix de la position de la caméra
5. adaptation de l'éclairage et de l'exposition
6. création de l'image
7. calcul automatique des annotations
8. vérification de la validité de l'image

Chaque image est produite à partir d'une nouvelle combinaison de paramètres. La scène conserve ainsi une structure cohérente, tout en présentant suffisamment de différences pour enrichir les données d'entraînement.

3.1.3 Environnement virtuel de génération

La palette constitue l'objet principal de la scène. Plusieurs variantes visuelles et géométriques peuvent être sélectionnées afin de ne pas limiter l'apprentissage à un seul modèle.

Des boîtes peuvent être placées sur la palette pour représenter une charge. Leur nombre, leur disposition et leur apparence changent d'une image à l'autre. Des palettes supplémentaires peuvent également être empilées pour créer des occultations partielles.

La caméra virtuelle est déplacée autour de la palette. Les vues latérales sont privilégiées, car elles montrent clairement une face utile pour la détection des points caractéristiques. Une petite proportion de vues plus libres est conservée afin d'augmenter la diversité.

L'éclairage est configuré pour produire des ombres et des variations d'intensité réalistes. Une correction automatique de l'exposition limite les images trop sombres ou trop claires.

3.1.4 Moteur de diversification visuelle

La diversité du jeu synthétique repose sur des paramètres contrôlés. Le tableau 3-1 présente les principales variations utilisées pendant la génération.

Élément variable	**Valeur de référence**	**Effet recherché**
Résolution de l'image	640×640 pixels	Conservation d'un format uniforme pour l'entraînement
Orientation de la caméra	vues latérales privilégiées	Observation claire des faces de la palette
Hauteur de la caméra	de 0° à 30° pour les vues principales	Variation de l'apparence verticale de la palette
Distance de la caméra	de 2,0 à 3,5 unités virtuelles	Variation de la taille apparente dans l'image
Variantes de palette	cinq variantes disponibles	Diversification de la géométrie et de l'apparence
Empilement de palettes	jusqu'à quatre palettes	Création d'occultations et de configurations verticales
Charge sur la palette	de deux à quatre objets	Représentation de plusieurs situations de chargement
Empilement de la charge	de deux à quatre niveaux	Variation de la hauteur et de l'encombrement
Apparence des boîtes	matériaux variables	Réduction de la dépendance à une couleur ou une texture unique
Exposition	ajustement automatique	Limitation des images trop sombres ou surexposées

Tableau 3-1 Principales variations appliquées pendant la génération synthétique.

Comme le montre le tableau 3-1, les variations concernent principalement le point de vue, la palette, la charge et l'apparence générale de la scène. Elles restent délimitées afin de conserver des images plausibles.

3.1.5 Production automatisée des images synthétiques

Pour chaque nouvelle image, la scène est réorganisée selon les paramètres sélectionnés. La palette et sa charge sont positionnées, puis la caméra est placée autour de l'objet principal.

Avant de conserver l'image, le système vérifie que la palette est suffisamment visible. Une image dans laquelle aucune face utile ne peut être identifiée n'est pas retenue.

Le système choisit d'abord une variante de palette et crée, au besoin, un empilement. Il ajoute et organise ensuite la charge, positionne la caméra, vérifie le cadrage et adapte l'exposition avant de produire l'image et son annotation.

Ce processus est répété jusqu'à obtenir le nombre d'images valides souhaité. Les ensembles utilisés dans les expériences ont été générés séparément afin de permettre une comparaison entre plusieurs volumes de données.

3.1.6 Moteur d'annotation automatique

L'un des principaux avantages de la génération synthétique est la connaissance exacte de la géométrie de la scène. La position de la palette et celle de la caméra sont connues au moment de la création de l'image.

Le système utilise ces informations pour déterminer automatiquement les coordonnées des points caractéristiques visibles. Il n'est donc pas nécessaire de placer manuellement les annotations sur chaque image.

Six points sont associés à chaque face retenue, toujours dans le même ordre : le centre de l'arête supérieure, le centre de l'arête inférieure, les coins supérieur et inférieur gauches, puis les coins supérieur et inférieur droits.

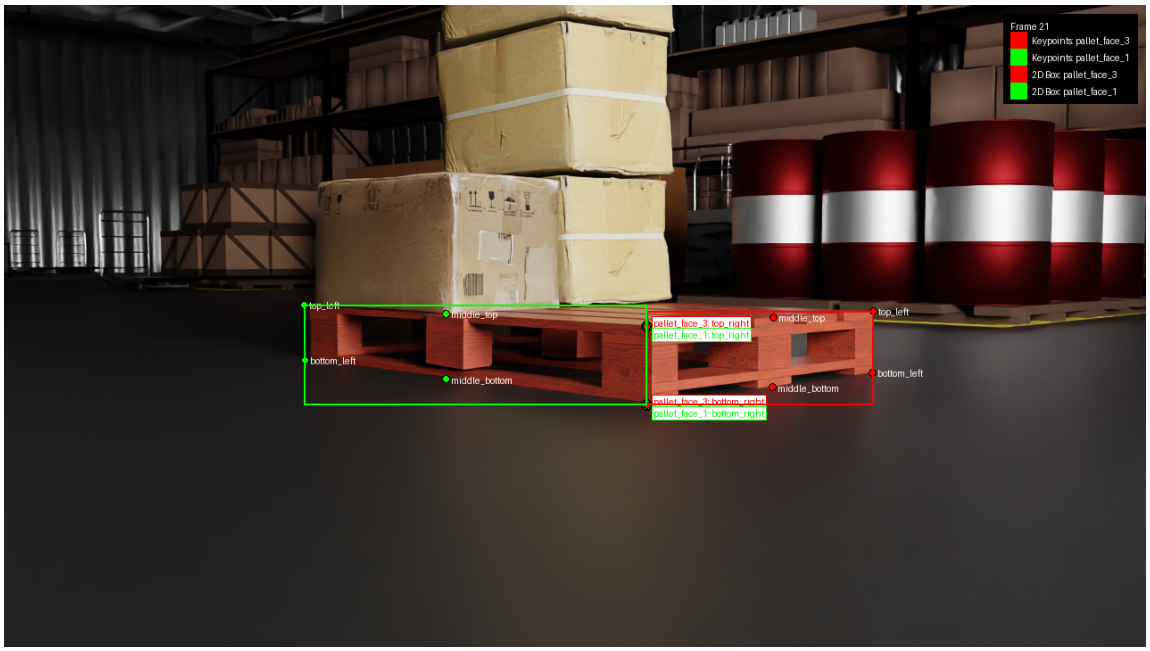
Ces points conservent toujours le même ordre. Cette cohérence permet au modèle d'apprendre la structure de la face de la palette et facilite l'utilisation des données pendant l'entraînement.

Une boîte englobante est également associée à la face. Les coordonnées sont adaptées à la taille de l'image afin que toutes les annotations suivent une représentation commune.

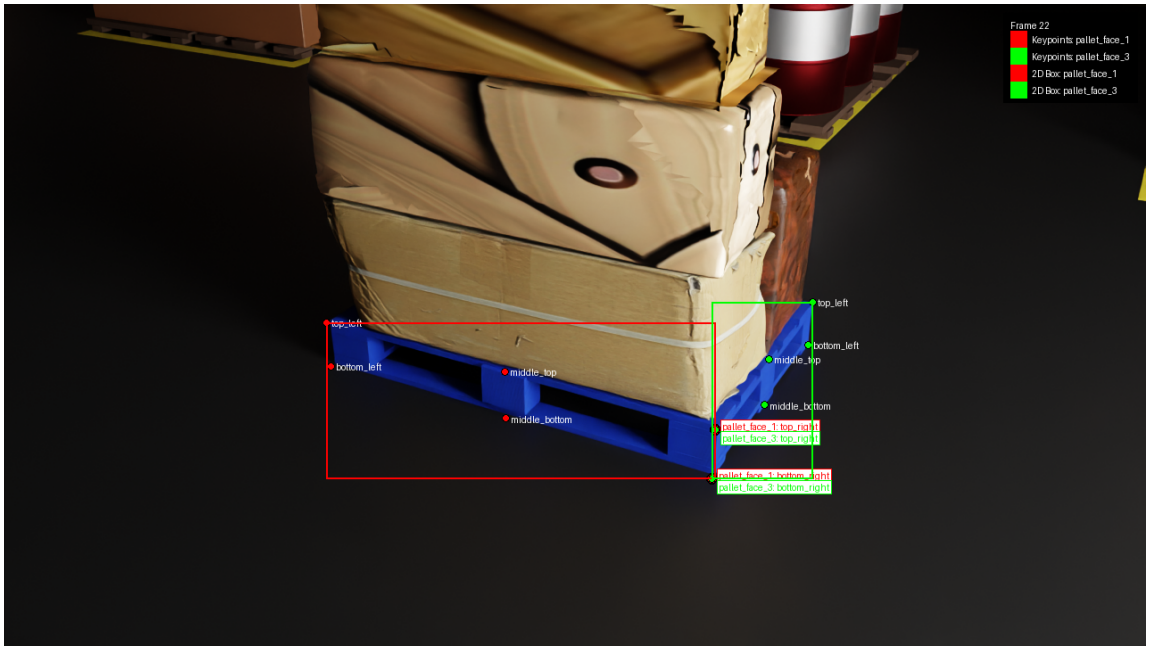
3.1.7 Jeu de données produit

Les données conservées pour l'entraînement comprennent les images en rouge, vert et bleu (RGB) des scènes générées, les boîtes englobantes des faces de palettes, les six points caractéristiques associés à chaque face et l'indication de visibilité de ces points.

Les figures 3.1 et 3.2 présentent plusieurs exemples issus de cette génération. Elles montrent les différences de charge, de point de vue, de texture et d'empilement. La dernière vue illustre également le contrôle visuel des points produits automatiquement.



(a) Palette chargée et partiellement occultée

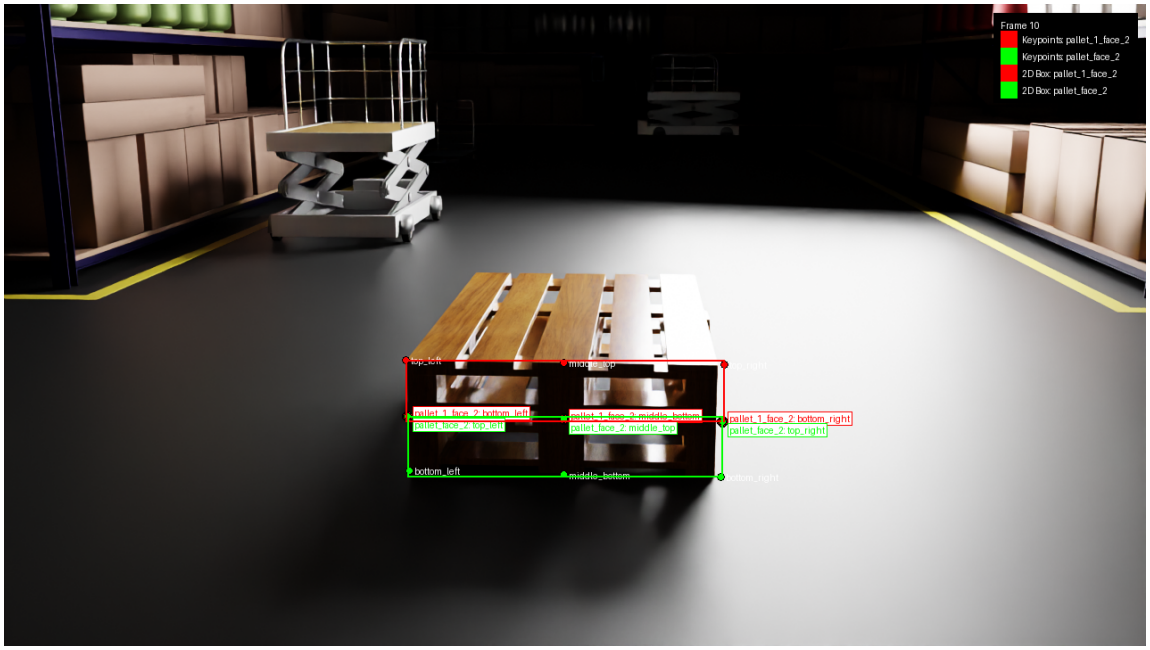


(b) Variation de charge et de luminosité

Figure 3.1 Exemples d'images synthétiques et de variations produites avec Blender (1/2).



(a) Variation de texture et d'encombrement



(b) Visualisation des points caractéristiques

Figure 3.2 Exemples d'images synthétiques et de variations produites avec Blender (2/2).

3.1.8 Transfert technologique du générateur à Noovelia

Au-delà du cas industriel étudié, le générateur constitue une méthode réutilisable : il permet de produire de nouvelles données, de modifier les paramètres de randomisation, d'entraîner de nouveaux modèles et de réemployer l'architecture d'intégration pour un autre environnement visuel. Les images et les annotations générées sont préparées pour être utilisées directement pendant l'entraînement des modèles de détection et de points caractéristiques, et la correspondance entre chaque image et son annotation est conservée automatiquement.

Noovelia constitue le cas industriel concret de ce transfert et de cette réutilisation. L'ensemble du travail de génération lui a été remis. Ce transfert comprend la scène virtuelle utilisée pour représenter les palettes, la logique de création des variations, la génération automatique des annotations, le code source développé pour produire les données ainsi que les éléments nécessaires à l'utilisation et à l'adaptation du générateur.

Noovelia peut ainsi produire de nouvelles données selon ses besoins et réutiliser le code dans ses travaux futurs. Cette remise assure la continuité entre le développement réalisé dans ce mémoire et son utilisation industrielle.

3.2 Architecture de la plateforme Simulateur de génération de données synthétiques

3.2.1 Architecture globale et composants

La plateforme Simulateur de génération de données synthétiques regroupe les éléments nécessaires pour passer de la génération des données à la visualisation des résultats. Son

architecture modulaire sépare les traitements afin que chaque partie puisse évoluer sans imposer une modification complète du système.

Les principaux blocs assurent la génération des données synthétiques, l'entraînement et la gestion des modèles, l'inférence sur les images, la simulation de l'environnement, les services de communication et la visualisation dans l'interface web.

La figure 3.3 présente leur organisation générale. Les données synthétiques servent à l'entraînement. Les modèles obtenus sont ensuite utilisés par le module d'intelligence artificielle. Le simulateur fournit les images et l'état de la scène, tandis que l'interface web rassemble les résultats.

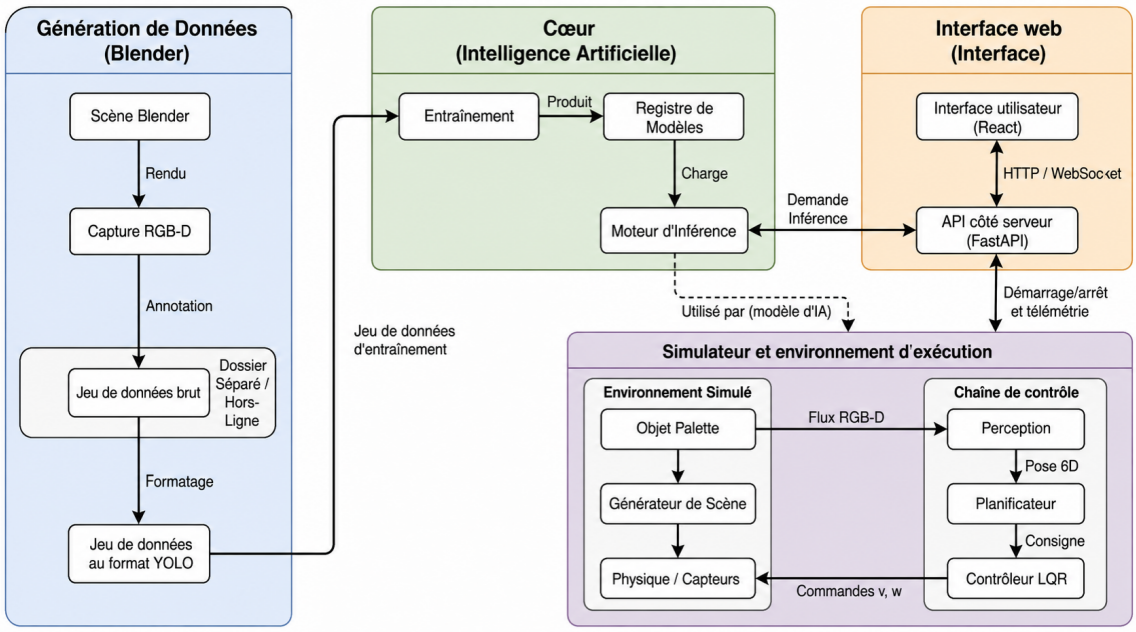


Figure 3.3 Architecture générale de la plateforme Simulateur de génération de données synthétiques

Comme l'illustre la figure 3.3, la génération est réalisée avant l'utilisation en temps réel. Elle alimente l'entraînement sans dépendre du simulateur ou de l'interface web. Cette séparation simplifie la préparation de nouvelles données et le remplacement d'un modèle.

3.2.2 Moteur d'intelligence artificielle et gestion des modèles

Le moteur d'intelligence artificielle de Simulateur de génération de données synthétiques rassemble les fonctions liées aux données et aux modèles. Son rôle est de fournir une organisation commune pour l'entraînement, la sélection des modèles et l'analyse des images.

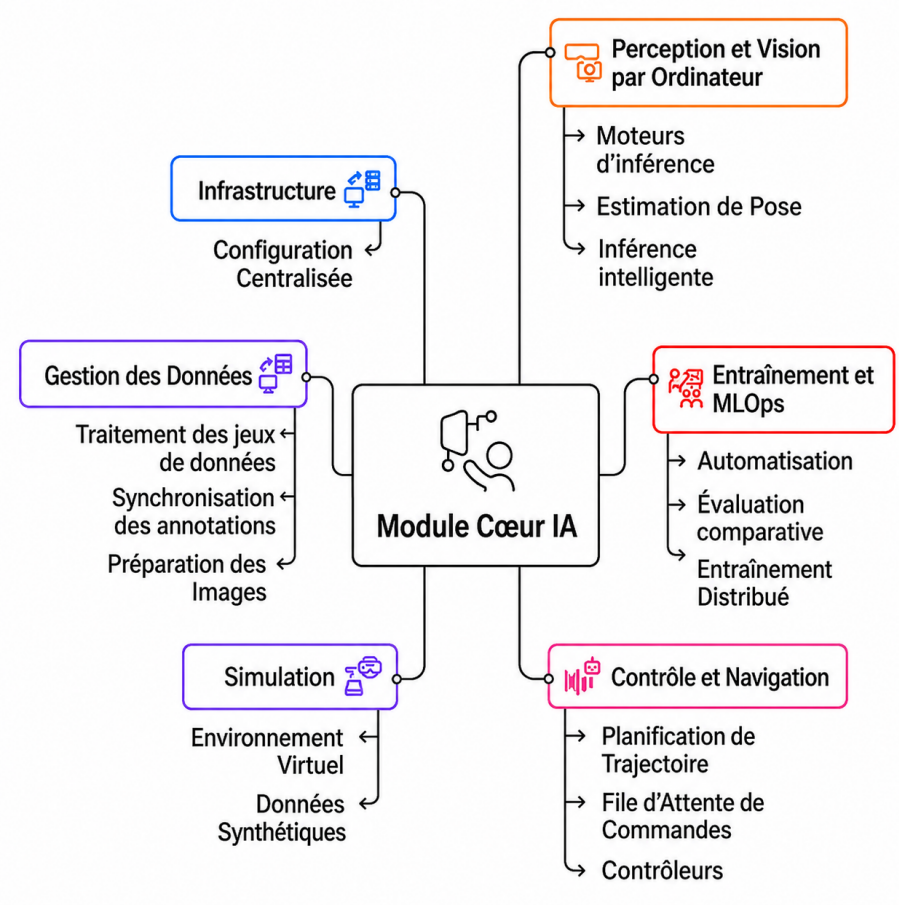


Figure 3.4 Organisation fonctionnelle du moteur d'intelligence artificielle de Simulateur de génération de données synthétiques.

La figure 3.4 présente les principales familles de fonctions du module. Elle montre que la gestion des données, la perception, l'entraînement, la simulation et les fonctions de navigation sont regroupées autour d'une même base.

Le module prépare les images et les annotations, entraîne les modèles, sélectionne celui qui convient à la tâche, traite les images reçues, puis transmet les résultats aux autres parties du système.

Le système prend en charge la détection des palettes et l'identification de leurs points caractéristiques. Les résultats comprennent les positions détectées, les niveaux de confiance et les informations nécessaires aux modules de pose.

3.2.3 Intégration du module de contrôle de la Chaire de recherche Noovelia

Le système de perception produit les informations nécessaires à l'estimation de la position de la palette. Ces résultats sont ensuite transmis à la partie chargée du déplacement du robot.

Le module de contrôle utilisé dans cette étape a été fourni par l'équipe de la Chaire de recherche Noovelia. Le travail réalisé dans ce mémoire consiste à l'intégrer à l'architecture commune et à lui transmettre les résultats issus de la perception.

Cette intégration permet de réutiliser le module de contrôle développé par l'équipe de recherche et évite de concevoir une nouvelle solution de contrôle. Elle relie également la perception aux autres éléments du système et rend possible l'essai de l'ensemble dans un environnement simulé.

Les lois internes de ce module de contrôle ne sont pas développées ici. L'accent est placé sur son intégration avec la perception, la simulation et l'interface.

3.2.4 Moteur de simulation robotique PyBullet

PyBullet est une plateforme libre de simulation physique destinée notamment à la robotique [39]. Elle permet de représenter des objets, leurs mouvements, les collisions et des capteurs virtuels dans un environnement en trois dimensions.

Dans ce projet, PyBullet sert à créer un environnement d'essai contenant un chariot, des palettes, des obstacles et une caméra virtuelle. Le simulateur fournit des images de couleur, des informations de profondeur et l'état général de la scène.

Parmi les solutions de simulation pertinentes figuraient notamment Gazebo, Webots, NVIDIA Isaac Sim et CoppeliaSim. Elles n'ont pas fait l'objet d'une comparaison expérimentale dans ce travail. Le choix de PyBullet repose sur des critères qualitatifs adaptés au projet : sa légèreté, son intégration directe avec Python, sa rapidité de mise en œuvre, le contrôle explicite de la simulation et son adéquation aux ressources matérielles disponibles.

PyBullet permet en outre une simulation suffisamment rapide pour une utilisation en temps réel, y compris sur du matériel aux ressources limitées. Il facilite la répétition des essais dans des conditions contrôlées et permet de tester l'intégration sans mobiliser constamment le robot réel, tout en prenant en charge les caméras virtuelles et les interactions physiques nécessaires au projet.

Le simulateur a été intégré au code du projet afin de fournir les images et les informations nécessaires aux autres composants. Le module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia est connecté à cet environnement pour observer son fonctionnement dans une scène virtuelle.

Cette approche réduit les risques pendant le développement. Elle permet de vérifier la communication entre les modules, la réception des images et l'affichage des résultats avant un déploiement sur le matériel réel.

3.2.5 Couche de services et communication en temps réel

Une couche de communication relie le simulateur, les modèles d'intelligence artificielle et l'interface web. Elle reçoit les images, transmet les demandes de traitement et retourne les résultats.

Cette couche reçoit les images produites par le simulateur et les envoie au modèle sélectionné. Elle transmet ensuite les détections et les points caractéristiques, diffuse l'état de la simulation et assure la synchronisation avec l'interface web.

Les échanges en temps réel utilisent une communication bidirectionnelle. Le simulateur peut continuer à produire des images pendant que l'interface actualise la scène et affiche les résultats de perception.

Le traitement est régulé afin de ne pas accumuler inutilement les images lorsque l'inférence demande plus de temps. Une nouvelle image est traitée lorsque le système est prêt à la recevoir.

3.2.6 Tableau de bord web Simulateur de génération de données synthétiques

Le tableau de bord web Simulateur de génération de données synthétiques constitue le point d'accès principal pour l'utilisateur. Il regroupe les fonctions de visualisation, de sélection des modèles, d'affichage des résultats et de suivi de la simulation.

La figure 3.5 montre l'organisation fonctionnelle de cette interface. L'acquisition, l'inférence, la visualisation et la configuration sont regroupées dans une même application.

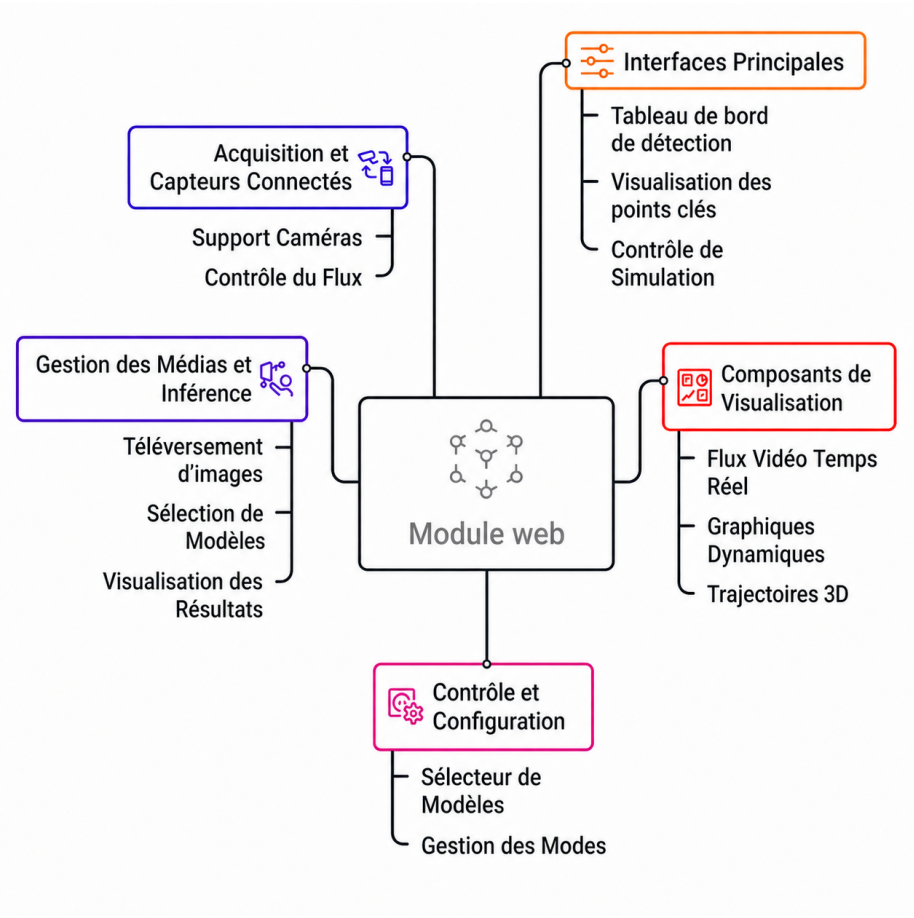


Figure 3.5 Organisation fonctionnelle du tableau de bord web Simulateur de génération de données synthétiques pour la détection et la simulation.

Comme l'indique la figure 3.5, l'interface affiche le flux vidéo simulé, les palettes détectées et leurs points caractéristiques. Elle permet également de sélectionner le modèle d'intelligence artificielle, de représenter l'environnement simulé et de consulter les informations de fonctionnement.

Les résultats sont superposés à l'image reçue afin de faciliter leur interprétation. La scène simulée peut également être observée sous une forme simplifiée pour suivre la position du chariot, des palettes et des obstacles.

3.2.7 Chaîne opérationnelle de bout en bout

L'intégration complète repose sur une circulation simple des informations entre les modules.

1. les données synthétiques sont générées et annotées automatiquement
2. les données sont utilisées pour entraîner les modèles
3. un modèle est sélectionné pour l'utilisation
4. PyBullet produit une scène et des images simulées
5. les images sont transmises au module d'intelligence artificielle
6. les résultats de détection et les points caractéristiques sont retournés
7. les informations obtenues sont transmises au module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia
8. l'interface web affiche la scène, les résultats et l'état du système

Cette organisation permet de remplacer un modèle ou de produire de nouvelles données sans modifier l'ensemble de l'application. Elle permet également de tester chaque composant séparément avant de vérifier son fonctionnement dans la chaîne complète.

La génération synthétique, les modèles et la simulation sont ainsi réunis dans l'architecture Simulateur de génération de données synthétiques. La plateforme couvre la préparation des données, l'entraînement, l'inférence et la visualisation.

3.2.8 Évolutivité, transfert et continuité technologique

De manière générale, l'architecture modulaire permet de générer de nouvelles données, d'adapter la randomisation de domaine aux variations d'un autre environnement, d'entraî-

ner de nouveaux modèles et de réutiliser la chaîne d'intégration sans reconstruire toute la plateforme.

Noovelia représente le cas industriel concret de transfert et de continuité technologique. Les développements lui ont été remis, notamment le générateur de données, son code source, les éléments de configuration, la logique d'annotation et l'intégration avec l'architecture de détection. Cette remise lui permet de générer de nouvelles images synthétiques, d'adapter les variations à de nouveaux besoins, d'entraîner de nouveaux modèles, de réutiliser l'architecture d'intégration et de poursuivre les essais avec le module de contrôle existant de l'équipe.

3.2.9 Synthèse du chapitre

Ce chapitre a présenté Simulateur de génération de données synthétiques, une plateforme intégrée pour la génération automatique d'images de palettes, l'entraînement, la simulation et la visualisation. La première section a décrit la création des scènes, les variations appliquées et la production des annotations.

La seconde section a présenté le rôle du module d'intelligence artificielle, de PyBullet, des services de communication et de l'interface web. Seul le module de contrôle présenté dans cette architecture provient de l'équipe de la Chaire de recherche Noovelia. Il a été intégré au reste du système développé dans le cadre de ce travail.

L'ensemble de la génération et du code associé a été remis à Noovelia afin de permettre sa réutilisation et son adaptation. Le chapitre 4 présente maintenant les expériences réalisées avec ces données et ces modèles.

Chapitre 4 - Analyse expérimentale en environnement industriel

Ce chapitre présente la démarche adoptée pour évaluer l'apport des données synthétiques décrites à la section 3.1 du chapitre 3. L'architecture utilisée pour intégrer les modèles, la simulation et l'interface est présentée à la section 3.2. La méthodologie est structurée autour d'un objectif applicatif plus large : soutenir la navigation autonome d'approche d'un chariot élévateur vers une palette. La contribution principale reste toutefois centrée sur la perception visuelle, et plus précisément sur la détection robuste de palettes lorsque les données réelles sont limitées. Sa frontière technique correspond aux sorties du détecteur : la boîte englobante 2D (*Bounding Box*, BBox) de la palette et les six points clés associés. Le module de contrôle utilisé après la perception a été fourni par l'équipe de la Chaire de recherche Noovelia et ne constitue pas une contribution algorithmique de ce mémoire.

La validation de la navigation d'approche réelle avec insertion des fourches n'est pas traitée dans ce mémoire. Ce choix est assumé pour des raisons de sécurité, de disponibilité du matériel et de maturité du système industriel. La validation est organisée de manière complémentaire : évaluations hors ligne sur chaque site, transfert inter-sites bidirectionnel sans ajustement sur le site cible, intégration en simulation, puis essais réels jusqu'à l'arrêt du robot devant la palette avec un ROSMASTER M3 Pro. Les métriques de navigation servent à observer l'effet des sorties de perception lorsqu'elles sont utilisées par le module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia.

4.1 Positionnement méthodologique : cadre CRISP-DM

La démarche méthodologique de ce mémoire s'inspire des six phases du cadre CRISP-DM (Cross-Industry Standard Process for Data Mining) [2], adapté aux spécificités de la vision par ordinateur et de l'intégration robotique. Ce cadre assure la rigueur et la reproductibilité du pipeline expérimental à travers six phases reliées et itératives.

- **Compréhension métier :** en milieu industriel logistique, un robot mobile doit s'approcher de manière autonome des palettes. Le module de perception développé doit détecter la boîte englobante et les points clés 2D de la palette face aux ouvertures. Cette sortie marque la frontière du projet. Le module de contrôle utilisé en aval a été fourni par l'équipe de la Chaire de recherche Noovelia. L'objectif est de concevoir un système de perception robuste tout en évitant le coût élevé des campagnes manuelles d'annotation d'images réelles
- **Compréhension des données :** cette phase analyse les sources d'information. Elle comprend l'étude de l'écart de domaine entre le jeu synthétique idéal et les données réelles. Les jeux de données réels des sites Costco et Noovelia sont analysés pour repérer les perturbations typiques (variations de luminosité, ombres portées, textures de bois usées, angles d'incidence extrêmes, occlusions et présence d'obstacles à proximité). Dans ce contexte, l'expression *environnement dynamique* désigne précisément ces variations susceptibles d'affecter la détection, y compris les perturbations transitoires dues aux opérateurs, aux objets mobiles et aux vibrations de la caméra
- **Préparation des données :** cette étape englobe la génération automatique sous Blender de quatre jeux d'images synthétiques distincts de 5 000, 10 000, 20 000 et 25 000 images, chacun généré séparément, ainsi que leurs annotations automatiques (boîtes englobantes, points clés et visibilité). Elle inclut également le protocole d'annotation manuelle des images réelles et le pipeline d'augmentation stochastique en ligne des données.
- **Modélisation :** cette phase consiste à sélectionner, configurer et entraîner les architectures de réseaux profonds. Elle compare les modèles de plusieurs générations YOLOv8-Pose, YOLO11-Pose et YOLO26-Pose à différentes échelles de para-

mètres (de Nano à Extra-Large). Les entraînements croisent plusieurs configurations (Réel seul, Synthétique seul, préentraînement hybride suivi d'un ajustement sur des fractions de données réelles)

- **Évaluation :** l'évaluation hors ligne mesure la précision moyenne globale ($mAP@0{,}5$), la similarité des points clés d'un objet (OKS), la latence et le débit en images par seconde (FPS) sur chaque site. Elle comprend une validation complémentaire inter-sites dans les deux directions. Une évaluation applicative observe ensuite le comportement du robot au point d'approche final, en simulation en boucle fermée (PyBullet) puis lors d'essais physiques sur un robot réel (ROSMASTER M3 Pro)
- **Déploiement :** le modèle de perception retenu est intégré au système robotique et relié au module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia afin de valider l'approche en temps réel

La figure 4.1 illustre le flux itératif de cette démarche. Les résultats de la phase d'évaluation en boucle fermée peuvent forcer un retour à la phase de préparation des données synthétiques (pour ajuster la randomisation de domaine sous Blender) ou de modélisation (pour adapter l'échelle de l'architecture).

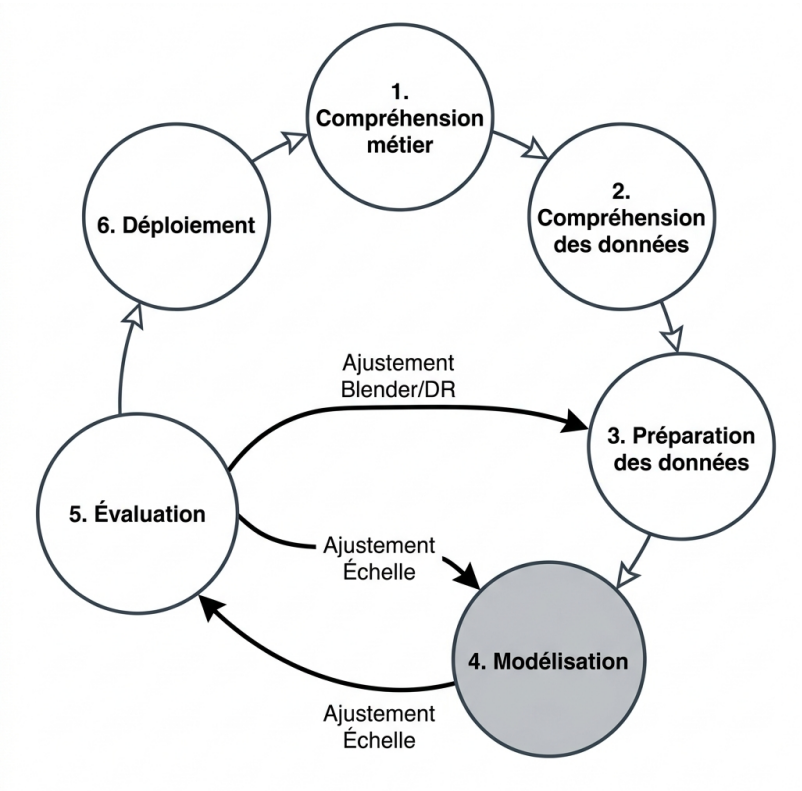


Figure 4.1 Adaptation par l'auteur du cycle CRISP-DM au protocole expérimental du mémoire, d'après [2].

4.2 Vue d'ensemble de la démarche

L'architecture de test suivie dans ce mémoire est structurée en deux grandes phases, chacune correspondant à une branche du protocole expérimental hybride combinant les données synthétiques puis les données réelles. Son organisation fonctionnelle reprend l'architecture décrite à la section 3.2. Le déroulement global, exposé à la figure 4.2, distingue explicitement ces deux branches, qui s'articulent chacune autour de la génération des données synthétiques sous Blender et de l'entraînement incrémental de modèles de perception.

La première branche, menée en simulation pour Costco, comprend la collecte du jeu de données, l'ajustement des modèles entraînés sur les jeux synthétiques, la validation hors

ligne sur 45 images réelles du site, puis le test du pipeline complet de perception et de navigation en simulation fermée sous PyBullet. La seconde branche, menée physiquement chez Noovelia, reprend l'ajustement des modèles préentraînés sur le jeu synthétique et la validation en temps réel sur 45 images, avant le test d'intégration du pipeline de navigation sur un robot ROSMASTER M3 Pro, décrit à la sous-section 4.2.1.

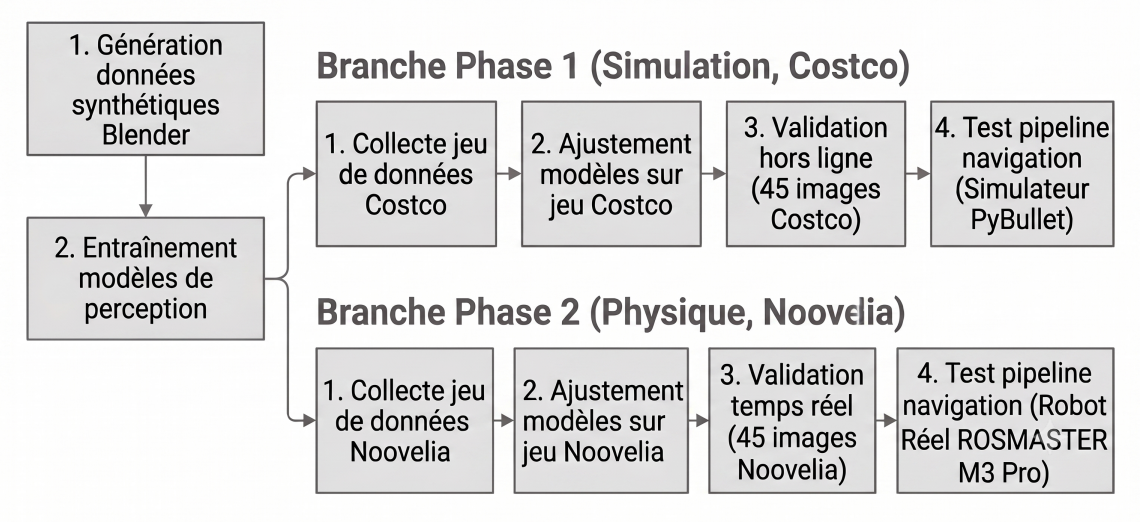


Figure 4.2 Schéma synoptique de la démarche expérimentale hybride (du synthétique vers le réel).

4.2.1 Robot ROSMASTER M3 Pro

Description générale

Le **ROSMASTER M3 Pro**, présenté à la figure 4.3, est un robot mobile conçu pour la recherche en vision, navigation et commande robotique.



Figure 4.3 Vue d'ensemble du robot ROSMASTER M3 Pro.

Spécifications techniques

Le tableau 4-1 synthétise les caractéristiques physiques et l'ensemble des équipements dans la version utilisée pour ce robot. Dans ce tableau, RAM désigne la mémoire vive, ROS 2 le *Robot Operating System 2*, IMU la centrale inertielle, LiDAR la détection et télémétrie par lumière, ToF le temps de vol, 6DoF les six degrés de liberté, IA l'intelligence artificielle, MEMS les systèmes microélectromécaniques et USB l'*Universal Serial Bus*.

Architecture matérielle et capteurs

L'architecture matérielle du M3 Pro intègre plusieurs canaux sensoriels synchronisés. Pour la navigation, deux capteurs de détection et télémétrie par lumière (LiDAR) à temps de vol (ToF), de modèle T-mini Plus, sont montés en diagonale et fournissent une cartographie sans angle mort. Une centrale inertielle (IMU) à neuf axes améliore la localisation et l'évitement adaptatif par l'intermédiaire de microROS. Pour la perception, le robot est équipé d'une caméra couleur et profondeur (RGB-D) Orbbec DABAI DCW2. Seul le canal RGB

Tableau 4-1 Spécifications techniques du ROSMASTER M3 Pro

Composant / Caractéristique	**Détails techniques**
Contrôleur principal	NVIDIA Jetson Orin NX Super (8 Go de RAM)
Système d'exploitation	Ubuntu avec ROS 2 Humble préinstallé
Contrôleur de bas niveau	Carte d'extension ROS, STM32, IMU à 9 axes
Type de propulsion	4 roues Mecanum (80 mm), mobilité omnidirectionnelle
Suspension	Pendulaire indépendante (arrière)
Capteurs télémétriques	Double LiDAR ToF (portée de 0,05–12 m, 360°)
Vision numérique	Caméra de profondeur binoculaire Orbbec DABAI DCW2
Manipulateur	Bras robotique 6DoF (précision de $\pm$0,5 mm)
Charge utile du bras	Jusqu'à 410 g (rayon 30 cm)
Audio et interaction	Module IA vocal (microphone MEMS et haut-parleur)
Alimentation	Batterie Li-ion 12,6 V / 9 600 mAh (protection multivoies)
Interfaces	microROS, concentrateur USB 3.0 et modules d'extension

est utilisé dans ce mémoire, sans information de profondeur. La caméra est fixe, orientée vers l'avant, et transmet les images au détecteur. Enfin, le module d'interaction homme-machine traite les commandes naturelles au moyen d'une réduction de bruit dédiée.

4.2.2 Phase 1 : Costco (simulation et hors ligne)

Cette phase couvre l'évaluation sur le cluster de Calcul Canada. Le pipeline transmet les sorties du détecteur à l'environnement PyBullet présenté à la sous-section 3.2.4. Le module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia utilise ensuite ces informations pendant l'essai simulé.

Les sorties enregistrées pour l'évaluation comprennent la boîte englobante, les six points clés, leur confiance, le temps d'inférence et les coordonnées transmises à la boucle de navigation. Les exemples visuels et les mesures obtenues sont présentés au chapitre 5.

4.2.3 Phase 2 : Noovelia (validation embarquée en temps réel sur robot)

Cette phase décrit le transfert du modèle retenu sur le robot physique ROSMASTER M3 Pro. Les résultats de détection sont transmis au module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia afin de permettre une intégration en temps réel dans l'environnement industriel.

La validation du détecteur est réalisée à l'aide de la caméra embarquée. Les prédictions, l'odométrie finale, les latences par trame, le nombre de détections valides et le statut de réussite de chaque essai sont enregistrés. Les résultats qualitatifs de détection et les résultats quantitatifs sur robot réel sont regroupés au chapitre 5.

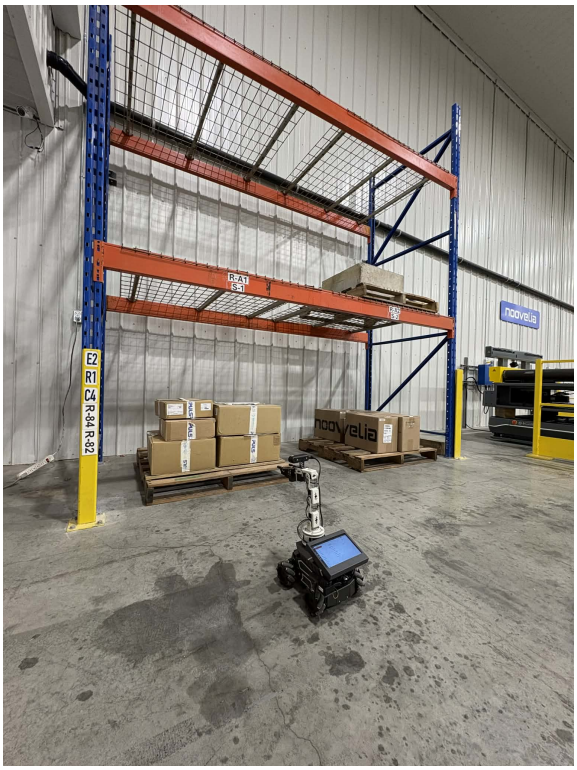


Figure 4.4 Vue d'ensemble du robot ROSMASTER M3 Pro déployé chez Noovelia.

La figure 4.4 illustre le contexte expérimental réel : le robot autonome complet positionné dans la zone d'expérimentation chez Noovelia. Cette image met en valeur l'environnement d'exploitation final du système, où l'ensemble du robot (châssis, capteurs, bras et caméra) est visible dans son espace de travail. Elle souligne l'ampleur des contraintes logistiques et visuelles d'un environnement industriel.

4.3 Données synthétiques et réelles

La démarche repose sur la combinaison de données synthétiques et de données réelles. La logique de génération et les variations appliquées aux images sont détaillées à la section 3.1. Le tableau 4-2 présente le volume et le partitionnement des ensembles de données utilisés.

Jeu de données	Source	Entraînement	Validation	Test	Annotation
Costco	Réel	70 % (210 images)	15 % (45 images)	15 % (45 images)	Manuelle (6 points clés + BBox)
Noovelia	Réel	70 % (210 images)	15 % (45 images)	15 % (45 images)	Manuelle (6 points clés + BBox)
Blender	Synthétique	5 000, 10 000, 20 000, 25 000	1 250 à 6 250	Test sur des données réelles	Automatique (6 points clés + BBox)

Tableau 4-2 Répartition structurelle des jeux de données réels et synthétiques par site.

4.3.1 Jeux de données réels

Les deux bases de données réelles indépendantes (Costco et Noovelia) comportent chacune un total de 300 images. Chaque base est découpée en 70 % d'entraînement (210 images), 15 % de validation (45 images) et 15 % de test (45 images). Le protocole de partitionnement est strict : aucune image ne chevauche les ensembles.

Conditions d'acquisition. Les vidéos ont été capturées avec une caméra RGB embarquée sur le ROSMASTER M3 Pro (caméra OAK-D, résolution 640×480 pixels) lors de sessions d'approche réelles sur chaque site industriel. Pour Costco, l'acquisition s'est dé-

roulée sur deux sessions distinctes (une en éclairage artificiel d'entrepôt, une en éclairage mixte près des quais de chargement). Pour Noovelia, les images proviennent de trois sessions réparties sur deux semaines, couvrant plusieurs conditions d'éclairage et d'encombrement. La distance caméra-palette au début de l'enregistrement varie de 2 à 5 mètres. Les trames ont été extraites à 1 Hz pendant l'approche, en éliminant les séquences redondantes (distance parcourue $< 0{,}1$ m entre deux trames consécutives) afin de maximiser la diversité visuelle à volume fixe. Les palettes sont de type Euro en bois standard (section en E), dans des états d'usure variés. Les scènes incluent des palettes chargées (cartons, caisses plastiques) et non chargées, avec des arrière-plans d'entrepôt hétérogènes. Aucune palette n'est instrumentée. Les images contenant plusieurs palettes visibles sont conservées ; seule la palette la plus proche de la caméra est annotée et utilisée pour l'évaluation.

4.3.2 Jeu de données synthétique

Le jeu synthétique couvre quatre ensembles indépendants générés sous Blender pour les études d'ablation (5 000, 10 000, 20 000 et 25 000 images). Aucun ensemble ne constitue un sous-ensemble d'un autre. Les étapes de création, les paramètres de variation et les annotations produites sont présentés à la section 3.1.

4.3.3 Protocole de validation complémentaire inter-sites

La transférabilité entre les deux sites est mesurée par un protocole bidirectionnel distinct d'une validation croisée statistique de type *k*-fold. Dans la première direction, yolo26n-pose est ajusté sur les 210 images d'entraînement Costco, puis évalué directement sur les 45 images de test Noovelia. Dans la seconde direction, le modèle est ajusté sur les 210 images d'entraînement Noovelia, puis évalué directement sur les 45 images de test

Costco. Aucun ajustement fin, choix d'hyperparamètre ou adaptation n'est effectué à partir des données du site cible.

Les deux directions sont exécutées pour la stratégie Réel seul et pour la stratégie Synthétique puis Réel avec 25 000 images synthétiques. Le préentraînement synthétique est identique dans les deux cas ; seule la provenance du jeu réel d'ajustement change. Ce protocole complète les évaluations nominales, dans lesquelles l'entraînement et le test proviennent du même site, et mesure la capacité du détecteur à conserver ses performances lors d'un changement de site.

4.4 Annotation manuelle des données réelles et convention topologique

L'annotation manuelle des images réelles suit un protocole rigoureux. L'outil *Label Studio* (version 1.12) est utilisé pour placer manuellement les boîtes englobantes et localiser les six points clés sur les images des jeux de données Costco et Noovelia. Deux annotateurs (un annotateur principal et un vérificateur) ont participé à la campagne ; les 30 premières images ont été annotées conjointement pour définir la convention, puis l'annotateur principal a traité l'ensemble des 600 images (300 par site) sous la supervision du vérificateur, qui a contrôlé 100 images tirées aléatoirement (50 par site) pour s'assurer de la cohérence inter-annotateur.

Les points clés invisibles (points clés situés hors du cadre ou masqués par un obstacle) sont annotés avec le drapeau de visibilité $v_i = 0$ dans le fichier YOLO, et leur coordonnée est fixée à $(0,0)$. Leur contribution est exclue du calcul de l'OKS. Lorsque plusieurs palettes apparaissent dans l'image, seule la palette la plus proche est annotée.

La convention topologique des points clés définit l'emplacement et la numérotation des six points caractéristiques de la palette sur sa face avant : les quatre coins extérieurs et les deux délimitations des ouvertures centrales d'insertion des fourches.

4.5 Règles strictes contre la fuite de données

Un protocole strict assure la séparation des partitions. Les 45 images de test de chaque site ne sont jamais utilisées pour l'entraînement ou le réglage des hyperparamètres. Les images synthétiques sont réservées au préentraînement.

Afin d'éviter une fuite temporelle ou visuelle, les images provenant d'une même session d'acquisition ont été affectées à une seule partition. Les trames voisines n'ont donc pas été réparties entre l'entraînement, la validation et le test. Les partitions sont fixées avant l'entraînement et la même partition est conservée pour toutes les méthodes.

L'ensemble d'entraînement sert à l'optimisation des poids. L'ensemble de validation sert à sélectionner les hyperparamètres, l'époque du meilleur point de contrôle et le modèle retenu. L'ensemble de test demeure verrouillé jusqu'à la fin du processus et sert uniquement à l'évaluation finale. Concrètement, le choix de yolo26n-pose comme modèle de référence pour les ablations, la simulation et le robot réel a été effectué à partir des performances mesurées sur l'ensemble de validation ($mAP@0{,}5$ et OKS), avant toute consultation des métriques sur le jeu de test. Le jeu de test n'a été déverrouillé qu'une seule fois, pour produire les résultats finaux du chapitre 5.

Pour la validation inter-sites, le jeu de test du site cible demeure également verrouillé et n'est consulté qu'après la sélection du modèle sur le site source. Cette règle garantit que les scores inter-sites mesurent un transfert sans adaptation au domaine cible.

4.6 Modèles évalués

Le tableau 4-3 présente la matrice multi-échelle des architectures YOLO-Pose retenues. Les tailles Nano et Small sont adaptées au temps réel embarqué, tandis que les tailles Medium, Large et Extra-Large établissent la borne supérieure de précision en évaluation hors ligne sur des grappes équipées de processeurs graphiques (*Graphics Processing Unit*, GPU).

Famille	Échelle	Nom du modèle	Paramètres	Rôle opérationnel	Cible d'évaluation
YOLOv8-Pose	n / s	`yolov8n-pose`, `yolov8s-pose`	∼ 3,3 M / ∼ 11,6 M	Temps réel embarqué	Hors ligne, simulation et robot réel
	m / l / x	`yolov8m-pose`, `yolov8l-pose`, `yolov8x-pose`	∼ 26,4 M / ∼ 44,4 M / ∼ 69,4 M	Borne supérieure de précision	Hors ligne uniquement sur grappe GPU
YOLO11-Pose	n / s	`yolo11n-pose`, `yolo11s-pose`	∼ 2,9 M / ∼ 9,9 M	Temps réel embarqué	Hors ligne, simulation et robot réel
	m / l / x	`yolo11m-pose`, `yolo11l-pose`, `yolo11x-pose`	∼ 20,9 M / ∼ 26,1 M / ∼ 58,8 M	Borne supérieure de précision	Hors ligne uniquement sur grappe GPU
YOLO26-Pose	n / s	`yolo26n-pose`, `yolo26s-pose`	∼ 2,9 M / ∼ 10,4 M	Temps réel embarqué	Hors ligne, simulation et robot réel
	m / l / x	`yolo26m-pose`, `yolo26l-pose`, `yolo26x-pose`	∼ 21,5 M / ∼ 25,9 M / ∼ 57,6 M	Borne supérieure de précision	Hors ligne uniquement sur grappe GPU

Tableau 4-3 Matrice des modèles YOLO-Pose et des échelles évaluées.

La génération YOLO26-Pose est la plus récente évaluée dans ce mémoire. Elle est orientée vers le déploiement embarqué, avec une inférence de bout en bout sans suppression non maximale et des améliorations destinées à une exécution plus rapide sur matériel contraint [28]. Pour assurer la reproductibilité, tous les modèles YOLO26-Pose utilisés dans ce travail proviennent de la même version du framework Ultralytics, fixée pour l'ensemble des entraînements et des évaluations, comme indiqué à la section 4.8.2.

4.7 Stratégies d'apprentissage comparées

4.7.1 Méthode Synthétique puis Réel

Quatre éléments d'apprentissage sont comparés de manière systématique. Tous les modèles commencent avec les poids de base préentraînés sur COCO. Cette règle s'applique aux trois stratégies principales.

- **COCO puis Réel (Réel seul) :** initialisation à partir des poids COCO, puis ajustement sur la fraction réelle considérée
- **COCO puis Synthétique (Synthétique seul) :** initialisation à partir des mêmes poids COCO, puis entraînement sur le jeu synthétique considéré. Cette stratégie mesure directement l'écart de domaine
- **COCO puis Synthétique puis Réel (Synthétique puis Réel) :** initialisation COCO, entraînement sur le jeu synthétique, puis ajustement sur la même fraction réelle que la référence COCO puis Réel
- **Fractions réelles :** pour caractériser la résistance du modèle au manque de données réelles, le jeu d'entraînement de 210 images réelles est partitionné en fractions progressives et imbriquées : 10 %, 15 %, 30 %, 50 %, 70 %, 80 % et 100 % (soit respectivement 21, 31, 63, 105, 147, 168 et 210 images). Les fractions sont imbriquées : les 21 images de la fraction 10 % sont incluses dans les 31 images de la fraction 15 %, et ainsi de suite. Les fractions sont tirées aléatoirement une fois, puis figées pour toutes les architectures et stratégies. Les ensembles de validation et de test restent fixes à 45 images chacun. L'ablation complète croisant les 7 fractions réelles et les 5 volumes synthétiques (35 configurations) est réalisée pour le modèle de référence yolo26n-pose. Les architectures yolov8n-pose et yolo11n-pose sont évaluées sur la fraction complète (210 images) avec et sans préentraînement synthétique (25 000 images), ainsi que sur l'ensemble des fractions dans la configuration 25 000 images synthétiques, afin de comparer la sensibilité au volume réel entre générations. Les échelles Small sont comparées sur la configuration combinant 25 000 images synthétiques et 210 images réelles pour les deux sites. Les échelles Medium, Large et Extra-Large sont évaluées uniquement sur Costco en configuration hybride finale (25 000 images synthétiques et 210 images

réelles) à titre indicatif. Le recours à une seule sélection par fraction, plutôt qu'à plusieurs sous-échantillons indépendants, est un compromis face au nombre élevé de configurations : pour yolo26n-pose, 7 fractions $\times$ 5 volumes synthétiques $\times$ 3 graines = 105 entraînements par site, auxquels s'ajoutent les architectures Nano et Small, les stratégies complémentaires et les validations inter-sites. Faire varier la composition des sous-ensembles pour chaque répétition aurait multiplié ce coût sans permettre l'étude systématique de l'interaction entre volume synthétique et fraction réelle. Pour chaque configuration, trois répétitions sont effectuées en variant la graine aléatoire d'entraînement (initialisation stochastique des poids non transférés, ordre des lots et transformations d'augmentation). Les graines utilisées sont `42`, `123` et `2024`. La composition de chaque fraction réelle reste identique entre les trois répétitions ; seule la variabilité liée au processus d'optimisation est donc capturée, et non l'incertitude due à l'échantillonnage du sous-ensemble d'entraînement.

Cette construction isole l'apport propre du préentraînement synthétique, puisque toutes les stratégies partent de la même initialisation générale.

Transfert des poids COCO-Pose. Les modèles YOLO-Pose sont initialisés à partir de poids préentraînés sur le jeu de données COCO-Pose, qui comprend une seule classe (la personne) et 17 points clés. Lors de l'adaptation à la palette (une classe, six points clés), le *backbone* et le *neck* (pyramide de caractéristiques) sont intégralement transférés. Le *head* de détection est partiellement réinitialisé : la branche de régression des points clés est réinitialisée car la topologie diffère et le nombre de points clés passe de 17 à 6, et la branche de classification est reconfigurée pour correspondre à la nouvelle sémantique de classe. Les branches de régression de boîtes et de score d'objet sont conservées. Ce transfert est

effectué automatiquement par le framework Ultralytics à partir de la configuration du jeu de données (nombre de classes et forme des points clés), sans modification manuelle du modèle. Le même mécanisme s'applique pour le passage du préentraînement synthétique (six points clés) vers l'ajustement réel.

La figure 4.5 illustre la matrice d'entraînement. L'axe vertical représente les volumes de préentraînement synthétique (0, 5 000, 10 000, 20 000 et 25 000 images) et l'axe horizontal les fractions de données réelles d'ajustement fin (10 % à 100 %). Chaque intersection définit une configuration d'entraînement. Les cases grises correspondent aux configurations Réel seul, sans préentraînement synthétique, tandis que les cases bleues correspondent aux configurations hybrides, avec préentraînement synthétique suivi d'un ajustement sur la fraction réelle indiquée. La matrice complète (35 configurations) est exécutée pour yolo26n-pose, l'architecture de référence. Les autres architectures sont évaluées sur un sous-ensemble représentatif de ces configurations, décrit ci-dessus (gris pour Réel seul et bleu pour le préentraînement synthétique suivi d'un ajustement réel).

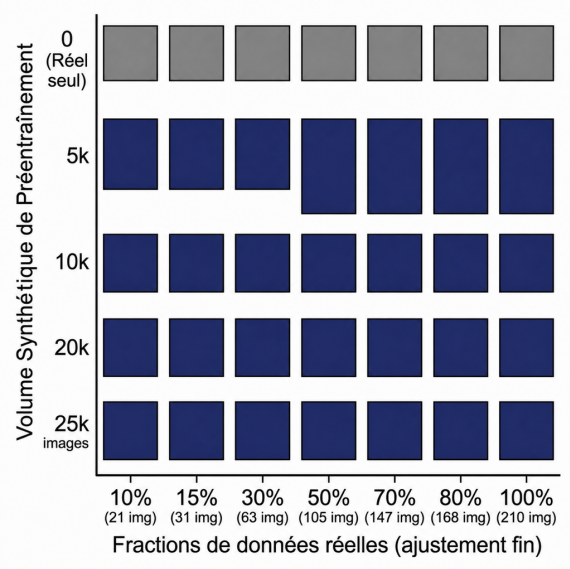


Figure 4.5 Schéma du protocole de partitionnement fractionné

4.7.2 Augmentation des données réelles et hyperparamètres d'entraînement

Afin de limiter le surapprentissage lors de l'ajustement fin sur de faibles volumes d'images réelles (par exemple les fractions à 10 % ou 15 %, soit 21 et 31 images), une augmentation stochastique en ligne est appliquée par le chargeur de données d'Ultralytics. Pour chaque image chargée dans un lot à chaque époque, le chargeur applique aléatoirement des transformations géométriques et photométriques. La diversité provient du renouvellement des transformations à chaque époque, et non d'une multiplication du nombre d'images. Les transformations géométriques (retournement, rotation, échelle, translation) et photométriques (ajustement de la teinte, de la saturation et de la valeur (HSV), flou gaussien, bruit de capteur) sont détaillées à la section A.2 de l'annexe. Le flou gaussien et le bruit de capteur sont appliqués après les transformations géométriques, avant le *letterbox* final, selon les paramètres listés dans le tableau des hyperparamètres (annexe A.2).

Les paramètres d'entraînement sont configurés de manière identique pour l'ensemble des entraînements (sur le cluster Calcul Canada et en local) et pour les modèles YOLOv8-Pose, YOLO11-Pose et YOLO26-Pose. L'inventaire complet des hyperparamètres d'optimisation et d'augmentation figure au tableau A.3 de l'annexe A.

Ces réglages définissent un protocole fixe plutôt qu'un optimum recherché. AdamW et les deux taux d'apprentissage organisent les deux étapes du transfert : le taux plus élevé est employé pendant le préentraînement synthétique, puis le taux plus faible limite l'amplitude des mises à jour pendant l'ajustement sur les données réelles. Les nombres maximaux d'époques, combinés à l'arrêt anticipé détaillé en annexe, bornent la durée d'entraînement tout en laissant la validation déterminer le meilleur point de contrôle. La taille de lot de 16 et l'entrée de 640×640 pixels sont conservées pour uniformiser l'échantillonnage et la résolution entre toutes les architectures. Les augmentations ciblent les variations décrites

dans les données réelles : orientation, échelle, cadrage, couleur, luminosité, flou et bruit de capteur. Les mêmes réglages sont maintenus entre architectures afin d'assurer une comparaison équitable. Aucune optimisation exhaustive des hyperparamètres n'est revendiquée ; ces valeurs ne sont donc pas nécessairement optimales et ce choix constitue une limite méthodologique.

4.8 Environnement matériel et logiciel

4.8.1 Environnement matériel

Les entraînements et les évaluations hors ligne exploitent les grappes de calcul haute performance de Calcul Canada, équipées de GPU NVIDIA H100 et A100. Les modèles de grande taille (m, l, x), qui demandent une puissance de calcul importante, font l'objet d'une évaluation hors ligne exclusive sur ces grappes.

La plateforme robotique utilisée pour les tests en ligne est le ROSMASTER M3 Pro, équipé d'un calculateur embarqué Jetson et d'une caméra RGB monoculaire. L'environnement de simulation repose sur PyBullet, dont le rôle et le choix sont expliqués à la sous-section 3.2.4.

4.8.2 Environnement logiciel

Les entraînements s'appuient sur le framework Ultralytics YOLO-Pose. Une seule version du framework a été fixée pour tous les modèles, toutes générations confondues (YOLOv8-Pose, YOLO11-Pose et YOLO26-Pose), afin de garantir une comparaison équitable et la reproductibilité des résultats : Ultralytics version `8.4.35`, PyTorch `2.10.0` et CUDA `12.8`. Le gel d'une version unique est essentiel pour la génération YOLO26-Pose, la plus récente du mémoire, dont les composants évoluent rapidement ; il garantit que les écarts de performance observés proviennent des stratégies d'apprentissage et non de

différences d'implémentation entre versions. Les outils d'annotation, les bibliothèques de simulation et les nœuds ROS 2 complètent l'environnement logiciel.

L'API de perception développée encapsule le détecteur et transmet les boîtes englobantes ainsi que les coordonnées des points clés 2D aux autres composants. Cette interface constitue la frontière fonctionnelle du travail. Le module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia utilise ensuite les informations nécessaires au déplacement du robot. La collecte des données de performance au point d'alignement cible sert uniquement à mesurer l'utilisabilité des sorties de perception dans le système complet.

L'évaluation et la visualisation en temps réel s'effectuent au moyen du tableau de bord décrit à la sous-section 3.2.6 et illustré à la figure 4.6. Le simulateur PyBullet, l'interface web et leur intégration avec le détecteur font partie du système mis en place dans ce travail. Seul le module de contrôle connecté à cette architecture a été fourni par l'équipe de la Chaire de recherche Noovelia. L'interface affiche le flux vidéo simulé avec les résultats de détection ainsi qu'une vue de l'environnement et de l'état du système.

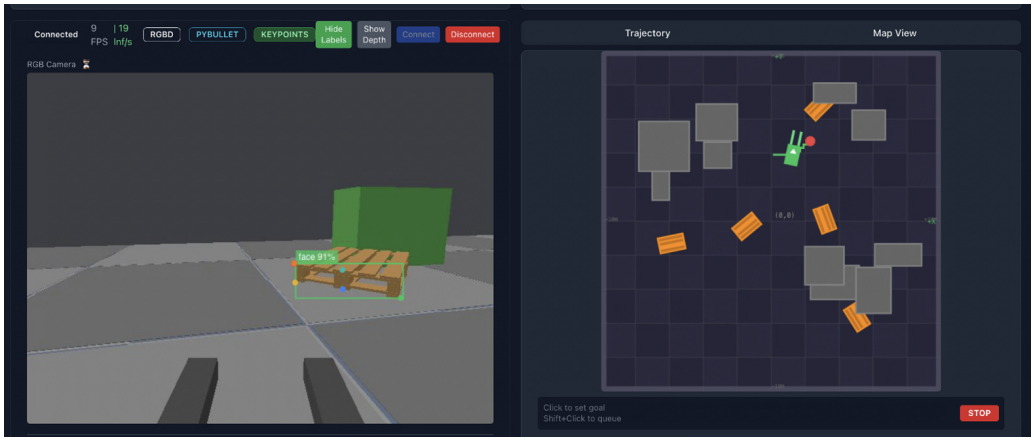


Figure 4.6 Interface utilisateur de l'API web connectée au simulateur de test PyBullet.

4.8.3 Métriques de détection d'objets

Les performances de perception sont évaluées au moyen de métriques standardisées pour la détection d'objets et l'estimation de pose par points clés. La tâche de détection vise à localiser la palette à l'aide d'une boîte englobante 2D. Les métriques associées reposent sur l'intersection sur union (IoU) :

$$\text{IoU} = \frac{\text{Aire}(B_{pred} \cap B_{gt})}{\text{Aire}(B_{pred} \cup B_{gt})} \tag{4.1}$$

où B_{pred} et B_{gt} désignent respectivement la boîte prédite et la boîte de vérité terrain. Une détection est classée comme un vrai positif (VP) si son IoU dépasse un seuil τ (généralement fixé à 0,5), et comme un faux positif (FP) sinon. Les objets de vérité terrain non détectés sont des faux négatifs (FN).

La précision (P) et le rappel (R) sont définis par :

$$P = \frac{\text{VP}}{\text{VP} + \text{FP}}, \quad R = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{4.2}$$

La précision moyenne (AP, Average Precision) résume la courbe précision-rappel en calculant l'aire sous cette courbe. Dans ce mémoire, nous évaluons la précision moyenne globale (mAP, Mean Average Precision) au seuil $mAP@0{,}5$, c'est-à-dire la précision moyenne calculée pour un seuil IoU de $\tau = 0{,}5$:

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c \tag{4.3}$$

où C est le nombre de classes (ici $C = 1$, la palette). Le seuil $\tau = 0{,}5$ correspond à une localisation de boîte considérée comme correcte pour la tâche d'approche, où l'orientation fine est portée par les points clés plutôt que par la boîte englobante. La métrique $mAP@0{,}5{:}0{,}95$ (moyennée sur plusieurs seuils d'IoU) n'a pas été retenue car, dans le contexte applicatif de l'approche de palette, le positionnement latéral et angulaire est évalué par l'OKS des points clés, et non par la superposition de la boîte englobante. Le $mAP@0{,}5$ fournit une mesure de détection binaire (présence/localisation approximative), tandis que la qualité géométrique fine est capturée par l'OKS.

4.8.4 Métriques d'estimation de pose (points clés)

L'estimation de la pose de la palette s'effectue par la détection de six points clés. La qualité de cette estimation est évaluée au moyen de la métrique Object Keypoint Similarity (OKS) :

$$\mathrm{OKS} = \frac{\sum_i \exp\left(\frac{-d_i^2}{2s^2k_i^2}\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{4.4}$$

où d_i est la distance euclidienne entre le point clé prédit et sa vérité terrain, s est l'échelle de l'objet (définie par l'aire de la boîte englobante), k_i est une constante de tolérance propre à chaque type de point clé, v_i désigne la visibilité du point clé et δ est la fonction indicatrice. La valeur de l'OKS varie entre 0 et 1 (1 représentant une prédiction parfaite). La valeur rapportée dans tout le mémoire est l'OKS moyen calculé sur l'ensemble des points clés valides de toutes les images de test (moyenne non pondérée des OKS individuels, sans seuil ni agrégation sur plusieurs seuils).

Les constantes k_i sont fixées à 0,05 pour chacun des six points clés, conformément à la valeur par défaut du framework Ultralytics pour les jeux de données de pose personnalisés (valeur utilisée pour toutes les architectures YOLO-Pose évaluées). Cette valeur unique a été conservée pour tous les points clés de la palette, sans calibration distincte par point, car la géométrie rigide de la palette rend les six points structurellement équivalents du point de vue de la tolérance d'erreur. Aucune donnée de test n'a participé à la définition de ces constantes.

4.8.5 Métriques d'efficience et de latence

Pour vérifier la compatibilité des modèles avec les exigences de contrôle en temps réel du robot, les métriques d'efficience suivantes sont mesurées :

- **Temps d'inférence GPU (T_{inf}) :** durée requise par le modèle pour traiter une image et prédire les boîtes et les points clés, mesurée en millisecondes (ms)
- **Débit en images par seconde (FPS) :** nombre de trames traitées par seconde, défini par $\text{FPS} = 1\,000/T_{inf}$

4.9 Métriques d'évaluation de l'alignement cible

Le point cible (*Target Point*) est le point d'attente situé à une distance de sécurité fixée devant les ouvertures de la palette, aligné avec l'axe central qui passe entre les deux trous. Dans ce travail, cette distance de sécurité est fixée à 50 cm devant la face avant de la palette. La figure 4.7 illustre la configuration d'alignement.

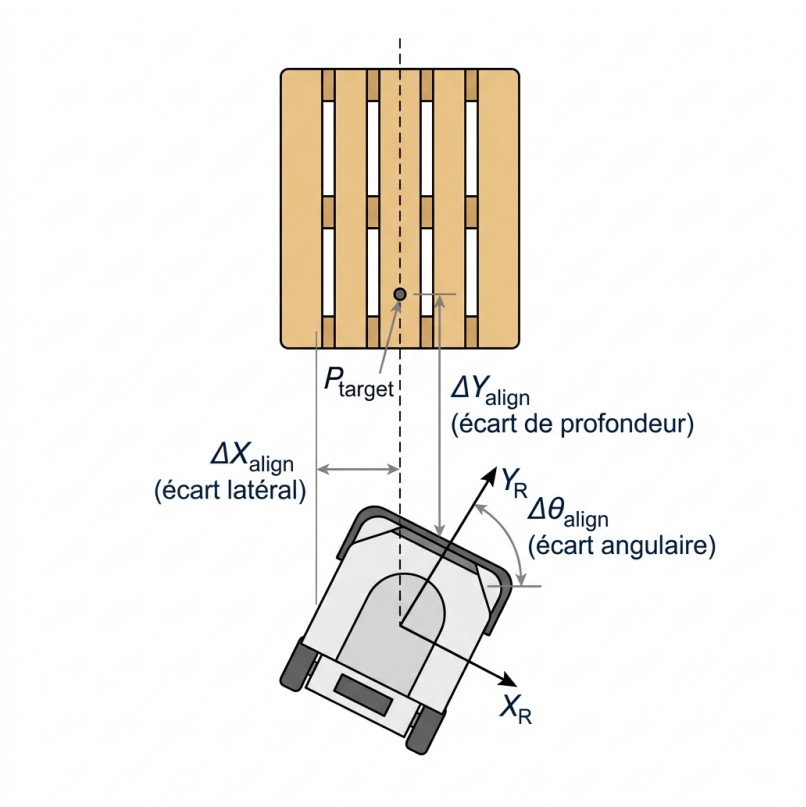


Figure 4.7 Schéma géométrique des variables d'erreur mesurées au point cible.

Les données de performance suivantes sont collectées lorsque le détecteur interagit avec le module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia. L'évaluation se concentre sur la précision du placement au point cible, situé exactement dans l'axe central face aux trous de la palette.

- **Désalignement latéral (ΔX_{align}) :** décalage transversal entre le centre du robot et l'axe central qui passe entre les deux trous de la palette
- **Erreur de profondeur de garde (ΔY_{align}) :** écart entre la position atteinte par le robot et la distance de sécurité fixée devant les ouvertures

- **Défaut de parallélisme ($\Delta\theta_{align}$) :** écart d'angle entre l'orientation du robot et la face avant de la palette au point d'attente
- **Indicateurs de continuité :** taux de perte de détection visuelle pendant la phase d'alignement et durée maximale de perte continue

4.9.1 Calcul des erreurs d'alignement

Le calcul est effectué dans le repère local de la palette. Soient $\mathbf{p}_f$ la position finale du centre du robot, $\mathbf{p}_t$ la position du point cible, $\mathbf{e}_X$ l'axe latéral de la palette et $\mathbf{e}_Y$ son axe longitudinal. Les erreurs de position sont calculées par projection de l'écart final sur ces deux axes :

$$\Delta X_{align} = \left|(\mathbf{p}_f - \mathbf{p}_t) \cdot \mathbf{e}_X\right| \tag{4.5}$$

$$\Delta Y_{align} = \left|(\mathbf{p}_f - \mathbf{p}_t) \cdot \mathbf{e}_Y\right| \tag{4.6}$$

La valeur ΔY_{align} ne provient donc pas directement du détecteur. Elle est obtenue à partir de la pose finale mesurée par l'odométrie au moment de l'arrêt et de la pose de référence du point cible, placée à 50 cm devant la face de la palette. Dans les fichiers d'essais, cette opération correspond à la valeur absolue de la différence entre la coordonnée longitudinale finale et la coordonnée longitudinale du point cible, convertie en centimètres. L'erreur angulaire est calculée comme la valeur absolue de l'écart d'orientation normalisé entre le robot et la palette :

$$\Delta\theta_{align} = \left|\text{wrap}\left(\theta_f - \theta_{palette}\right)\right| \tag{4.7}$$

Cette définition est appliquée de la même manière en simulation et sur le robot réel. Elle permet de distinguer l'erreur latérale, l'erreur de distance d'arrêt et l'erreur d'orientation.

4.9.2 Choix des tolérances d'alignement

Les seuils utilisés dans l'analyse sont fixés avant l'interprétation des essais à $|\Delta X_{align}| \leq$ 3,0 cm pour le décalage latéral, $|\Delta Y_{align}| \leq 5{,}0$ cm pour la profondeur de garde et $|\Delta\theta_{align}| \leq 2{,}5^\circ$ pour le défaut de parallélisme. Le seuil latéral traduit la marge nécessaire pour conserver le robot centré face aux ouvertures. Le seuil longitudinal est plus large, car il qualifie une distance d'arrêt de sécurité avant l'insertion. Le seuil angulaire limite une arrivée oblique susceptible de dégrader l'approche finale.

Ces tolérances ont été définies en collaboration avec l'équipe technique de Noovelia en octobre 2025, avant le début des essais sur robot réel. Elles découlent du gabarit géométrique des ouvertures de la palette standard utilisée à l'entrepôt : une marge latérale de 3,0 cm garantit que les fourches pénètrent sans contact avec les montants, une distance d'arrêt de 5,0 cm assure une garde de sécurité suffisante avant l'insertion manuelle ou automatisée, et une tolérance angulaire de 2,5° évite un blocage par coincement en cas d'approche oblique. Ces seuils ont été validés par le responsable industriel du site et n'ont pas été modifiés après l'obtention des résultats expérimentaux.

4.10 Synthèse du chapitre

La méthodologie présentée dans ce chapitre couvre l'ensemble du pipeline expérimental en s'appuyant sur l'architecture décrite à la section 3.2. Le cadre CRISP-DM structure la démarche, les évaluations nominales et inter-sites mesurent la perception hors ligne, et les phases Costco en simulation puis Noovelia sur robot réel vérifient l'intégration applicative. Les stratégies d'apprentissage croisent de manière systématique les volumes de

données synthétiques, les fractions réelles et les architectures YOLO-Pose de plusieurs générations. La portée de la contribution s'arrête toutefois à la sortie du détecteur. Le module de contrôle utilisé pour exploiter cette sortie a été fourni par l'équipe de la Chaire de recherche Noovelia.

Chapitre 5 - Résultats et discussion

Ce chapitre présente les résultats expérimentaux et leur interprétation dans l'ordre logique du protocole. L'analyse commence par l'effet des volumes de données réelles et synthétiques, compare ensuite les architectures sur le jeu de données Costco, vérifie la cohérence des observations sur le jeu de données Noovelia (incluant la robustesse), puis quantifie le transfert inter-sites dans les deux directions (section 5.4) avant d'examiner la validation applicative en simulation et sur robot réel. Chaque famille de résultats est immédiatement suivie de son interprétation afin de relier les mesures aux hypothèses de recherche.

Les résultats numériques exhaustifs de l'étude d'ablation sont fournis à l'annexe A. Les hyperparamètres complets sont rappelés au tableau A.3, tandis que les mesures détaillées de robustesse sont regroupées à la section A.3.

5.1 Apport du transfert synthétique vers réel

5.1.1 Effet de la quantité de données réelles disponibles

Cette étude évalue l'hypothèse principale du mémoire : l'utilisation de données synthétiques permet de réduire le nombre d'images réelles annotées nécessaires. Les figures 5.1 et 5.2 présentent respectivement le $mAP@0{,}5$ (Mean Average Precision au seuil Intersection over Union de 0,5) pour la boîte englobante et l'OKS (Object Keypoint Similarity) sur le jeu de test Costco de 45 images. Le modèle de référence est yolo26n-pose, préentraîné sur 0, 5 000, 10 000, 20 000 ou 25 000 images synthétiques, puis ajusté avec 21 à 210 images réelles annotées.

La figure 5.1 montre l'ablation complète pour yolo26n-pose (5 volumes synthétiques), ainsi que les courbes de yolov8n-pose et yolo11n-pose pour la configuration à 25 000 images synthétiques et pour la stratégie Réel seul. La stratégie Réel seul (yolo26n-

pose) atteint 52,2 % de *mAP*@0,5 avec 31 images, contre 87,9 % avec le préentraînement sur 25 000 images synthétiques, soit un gain de 35,7 points. La figure 5.2 montre la même progression de 46,5 % à 82,9 % d'OKS, soit 36,4 points.

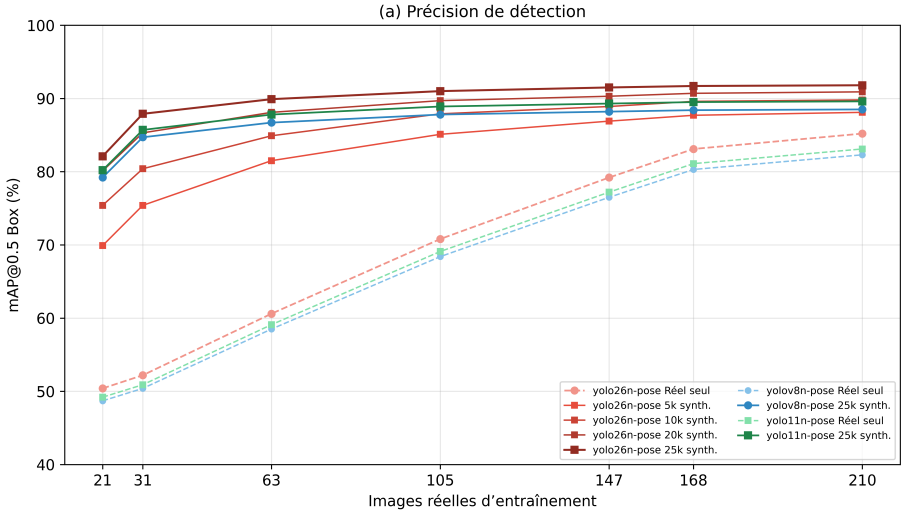


Figure 5.1 Ablation sur le jeu de test Costco : mAP@0,5 selon les images réelles et le volume synthétique

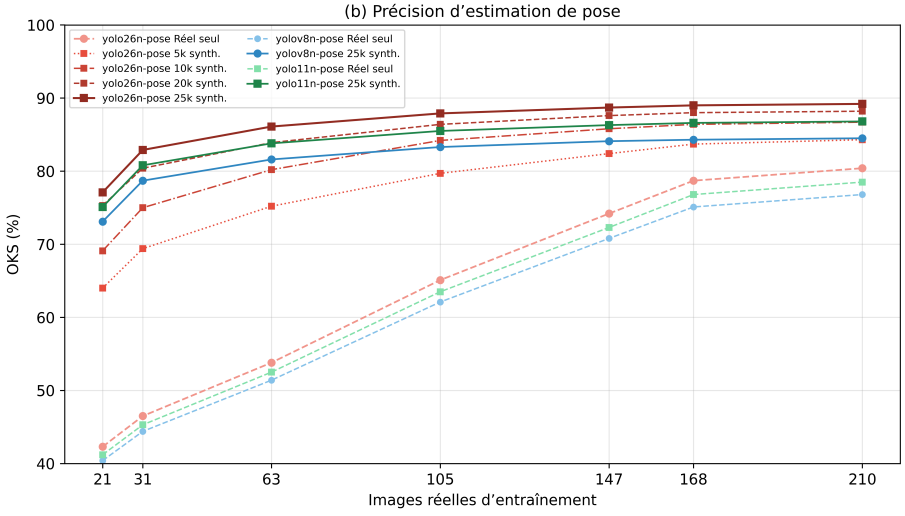


Figure 5.2 Ablation sur le jeu de test Costco : OKS selon les images réelles et le volume synthétique

Interprétation. La stratégie hybride résiste mieux au faible volume de données. Avec 31 images réelles, elle dépasse aussi la configuration Réel seul entraînée avec 210 images, qui obtient 85,2 % de *mAP*@0,5 et 80,4 % d'OKS. Ce résultat correspond à un rapport d'environ 6,8 entre les volumes annotés dans le sous-ensemble fixé étudié, sans constituer une estimation générale du facteur de réduction applicable à tout nouveau site. Ce constat indique que le préentraînement synthétique fournit une représentation initiale utile de la géométrie des palettes. L'hypothèse **H1**, relative à l'apport du transfert synthétique vers réel, est donc soutenue.

Les courbes montrent également un rendement décroissant des données réelles. Après le préentraînement synthétique, les gains sont plus importants aux faibles volumes, puis ralentissent progressivement. Dans le protocole observé, le ralentissement est net à partir de 105 images et les gains deviennent faibles au-delà de 147 images. Pour le préentraînement sur 25 000 images, le *mAP*@0,5 passe de 91,5 % à 91,8 % entre 147 et 210 images. Les valeurs complètes et leurs écarts-types sont présentés au tableau A.1. Ces observations soutiennent l'hypothèse **H3**.

5.1.2 Résultats de la stratégie Synthétique seul

La stratégie COCO puis Synthétique, sans ajustement sur des images réelles, est prévue dans la méthodologie pour mesurer directement l'écart de domaine. Elle doit être distinguée de la configuration notée 0 dans les figures 5.1 et 5.2, qui correspond à un ajustement sur des données réelles sans préentraînement synthétique.

Le tableau 5-1 présente les résultats de la stratégie Synthétique seul (COCO puis Synthétique, sans aucun ajustement sur des images réelles) pour les trois architectures Nano sur les jeux de test Costco et Noovelia, pour chaque volume synthétique ($n = 3$ répétitions).

Architecture	Vol. synth.	Costco		Noovelia	
		mAP@0,5 (%)	**OKS (%)**	**mAP@0,5 (%)**	**OKS (%)**
yolov8n-pose	5 000	$30{,}7 \pm 2{,}5$	$27{,}5 \pm 3{,}0$	$28{,}6 \pm 2{,}8$	$25{,}3 \pm 3{,}2$
	10 000	$32{,}3 \pm 2{,}0$	$29{,}0 \pm 2{,}5$	$30{,}0 \pm 2{,}3$	$27{,}0 \pm 2{,}8$
	20 000	$34{,}0 \pm 1{,}5$	$30{,}5 \pm 2{,}0$	$31{,}6 \pm 1{,}8$	$28{,}5 \pm 2{,}2$
	25 000	$34{,}8 \pm 1{,}4$	$31{,}3 \pm 1{,}8$	$32{,}4 \pm 1{,}6$	$29{,}2 \pm 2{,}0$
yolo11n-pose	5 000	$31{,}1 \pm 2{,}4$	$28{,}0 \pm 2{,}9$	$29{,}0 \pm 2{,}7$	$25{,}9 \pm 3{,}1$
	10 000	$32{,}8 \pm 1{,}9$	$29{,}5 \pm 2{,}4$	$30{,}5 \pm 2{,}2$	$27{,}5 \pm 2{,}7$
	20 000	$34{,}5 \pm 1{,}4$	$31{,}0 \pm 1{,}9$	$32{,}1 \pm 1{,}7$	$28{,}9 \pm 2{,}1$
	25 000	$35{,}3 \pm 1{,}3$	$31{,}8 \pm 1{,}7$	$32{,}9 \pm 1{,}5$	$29{,}6 \pm 1{,}9$
yolo26n-pose	5 000	$33{,}9 \pm 2{,}2$	$30{,}5 \pm 2{,}7$	$31{,}6 \pm 2{,}5$	$28{,}5 \pm 2{,}9$
	10 000	$34{,}5 \pm 1{,}7$	$31{,}0 \pm 2{,}2$	$32{,}1 \pm 2{,}0$	$28{,}9 \pm 2{,}5$
	20 000	$35{,}3 \pm 1{,}2$	$31{,}8 \pm 1{,}7$	$32{,}9 \pm 1{,}5$	$29{,}6 \pm 1{,}9$
	25 000	$35{,}8 \pm 1{,}1$	$32{,}2 \pm 1{,}5$	$33{,}4 \pm 1{,}3$	$30{,}1 \pm 1{,}7$

Tableau 5-1 Stratégie Synthétique seul : performances par volume synthétique

Le tableau 5-1 montre que la performance croît avec le volume synthétique selon une progression rapidement décroissante. Pour yolo26n-pose sur le jeu de test Costco, le $mAP@0{,}5$ passe de 33,9 % (5 000 images) à 34,5 % (10 000 images), 35,3 % (20 000 images) et 35,8 % (25 000 images). La progression est analogue pour yolo11n-pose (35,3 % avec 25 000 images) et yolov8n-pose (34,8 % avec 25 000 images).

5.1.3 Effet du volume de données synthétiques

La figure 5.3 montre l'effet du volume synthétique sur les trois architectures Nano ajustées avec les 210 images réelles Costco. Le passage de 0 à 10 000 images synthétiques

augmente le *mAP*@0,5 de 85,2 % à 89,8 % (yolo26n-pose), de 83,1 % à 87,6 % (yolo11n-pose) et de 82,3 % à 86,6 % (yolov8n-pose). Les gains sont respectivement de 4,6, 4,5 et 4,3 points.

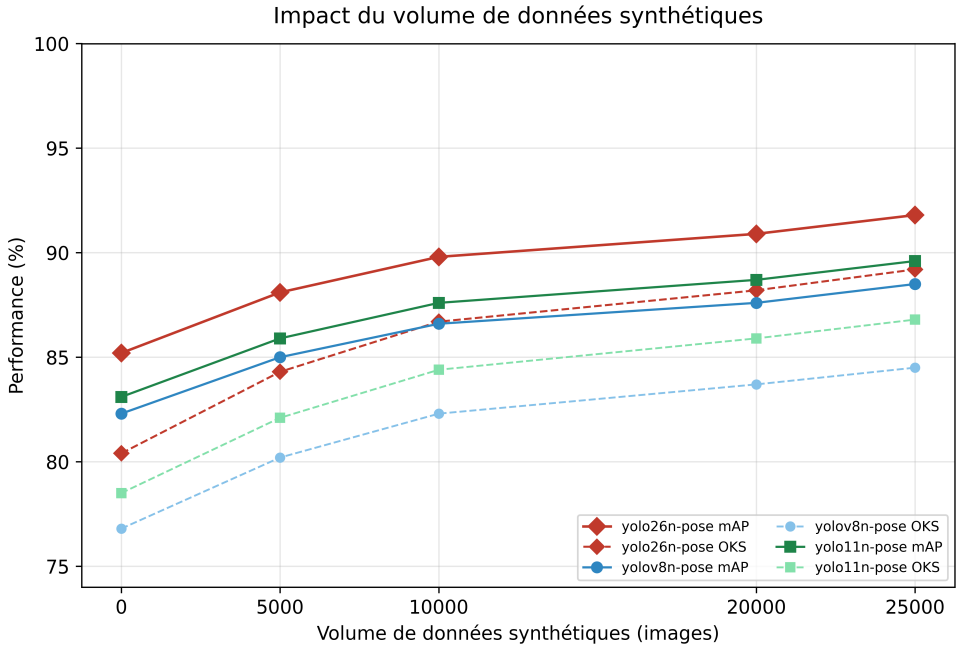


Figure 5.3 Impact du volume de préentraînement synthétique

Interprétation. La pente est forte jusqu'à 10 000 images, puis s'atténue nettement au-delà. Dans le protocole observé, une zone de rendement décroissant apparaît dès 10 000 images, tandis qu'une quasi-saturation est atteinte autour de 20 000 images, où les gains deviennent marginaux. Cette tendance indique que la diversité des scènes devient plus importante que l'ajout d'images procédurales proches. Le volume de 10 000 images constitue donc un compromis pratique lorsque les données réelles sont suffisantes. Avec seulement 21 ou 31 images réelles, le volume de 20 000 ou 25 000 images reste toutefois avantageux, comme le montre le tableau A.1. L'hypothèse **H4**, relative au rendement décroissant du volume synthétique, est soutenue dans le cadre du protocole évalué.

5.1.4 Synthèse de l'effet du volume réel

La figure 5.4 rassemble les deux métriques principales pour les stratégies Réel seul et Synthétique puis Réel avec 25 000 images synthétiques. Avec 31 images réelles, la stratégie hybride atteint 87,9 % de *mAP*@0,5 et 82,9 % d'OKS, alors que la stratégie Réel seul plafonne à 52,2 % et 46,5 % respectivement.

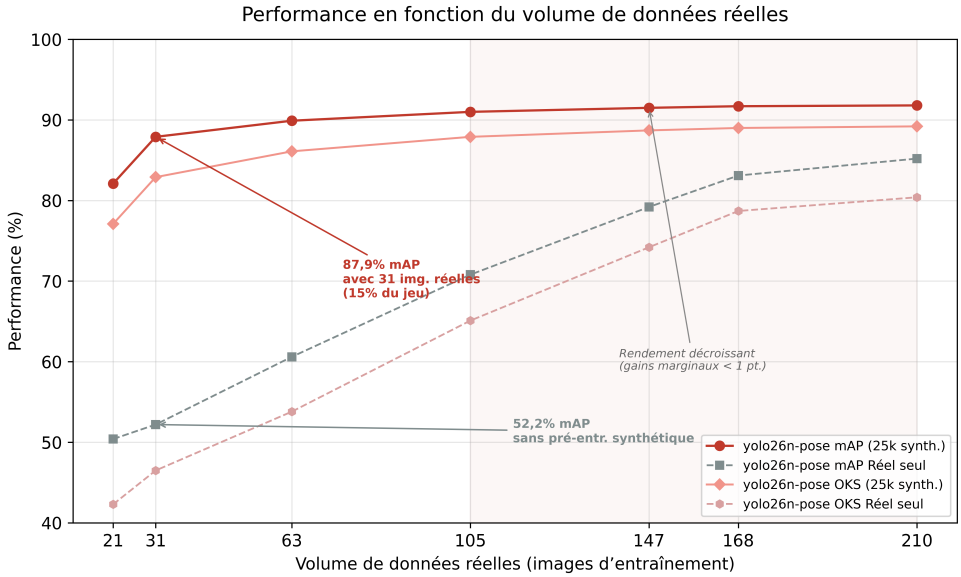


Figure 5.4 Synthèse mAP et OKS (yolo26n-pose) selon le volume réel

5.2 Comparaison des architectures sur le jeu de données Costco

5.2.1 Résultats qualitatifs de détection

La figure 5.5 présente quatre exemples du jeu de test Costco. Les boîtes englobantes et les six points clés prédits sont superposés aux images. Ces exemples complètent les métriques quantitatives en montrant le comportement du détecteur dans plusieurs scènes réelles.



Figure 5.5 Exemples de détection sur des images réelles du jeu de test Costco.

5.2.2 Compromis entre précision et latence

La figure 5.6 compare les familles YOLOv8-Pose, YOLO11-Pose et YOLO26-Pose aux échelles Nano et Small. Les configurations hybrides utilisent 25 000 images synthétiques et 100 % des données réelles. Pour toutes les familles, les points hybrides sont au-dessus des points Réel seul. Parmi les modèles Nano, yolo26n-pose atteint 91,8 % de $mAP@0{,}5$

(Mean Average Precision) pour une latence GPU de 17,5 ms, contre 89,6 % et 21,3 ms pour yolo11n-pose.

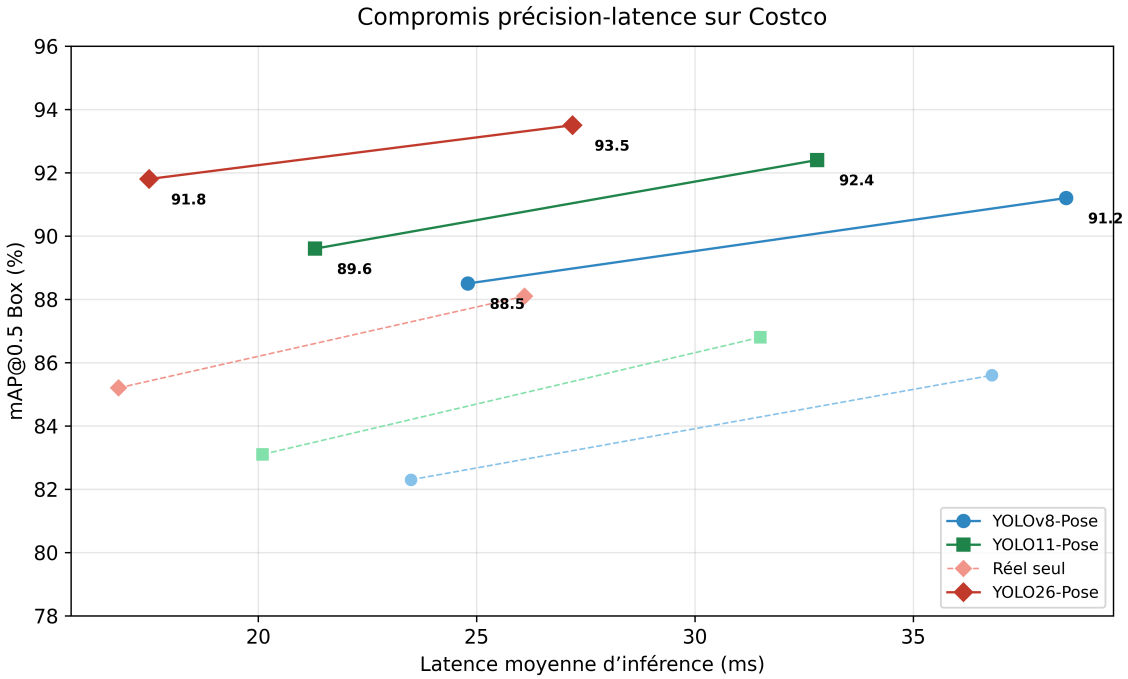


Figure 5.6 Compromis entre précision de détection et latence GPU sur le jeu de test Costco.

La figure 5.7 reprend le même protocole pour la qualité de pose. yolo26n-pose obtient 89,2 % d'OKS dans la configuration hybride, contre 86,8 % pour yolo11n-pose et 84,5 % pour yolov8n-pose.

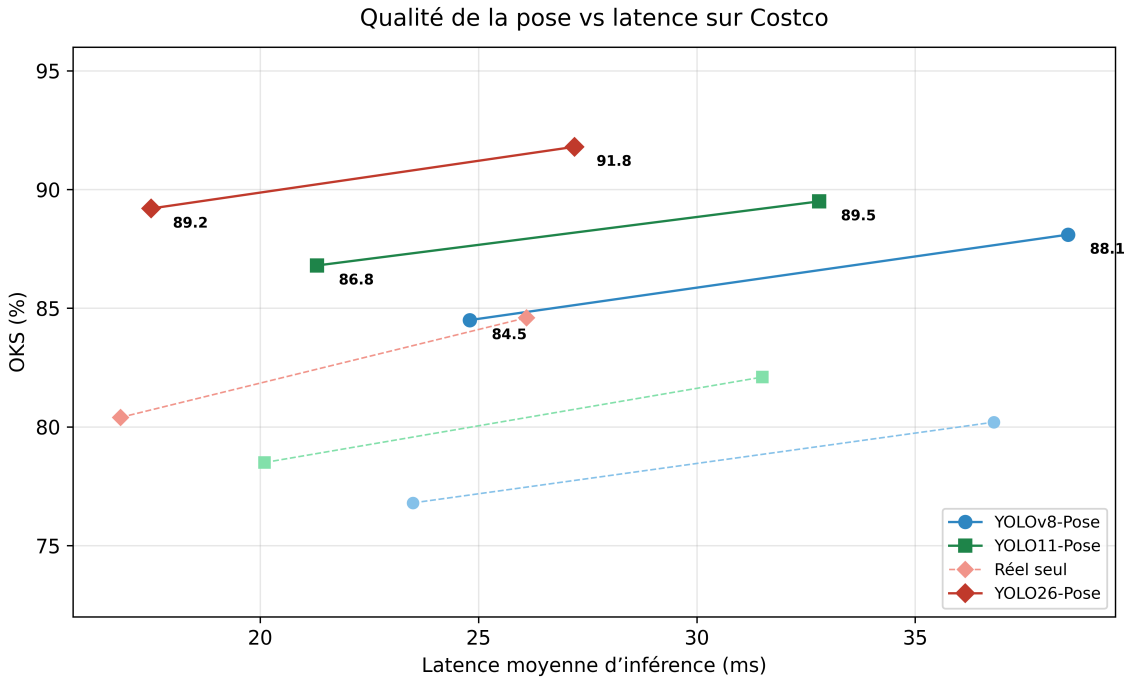


Figure 5.7 Compromis entre qualité d'estimation de pose et latence GPU sur le jeu de test Costco.

Interprétation. Le préentraînement synthétique augmente la précision sans modifier la complexité de l'architecture. Le temps d'inférence dépend du modèle déployé, alors que l'amélioration du *mAP*@0,5 et de l'OKS provient des représentations apprises. YOLO11-Pose et YOLO26-Pose présentent une meilleure qualité de pose que YOLOv8-Pose à latence comparable.

5.2.3 Évolution des performances selon l'échelle

Les figures 5.8 et 5.9 illustrent l'effet du passage de Nano à Extra-Large sous la stratégie Synthétique puis Réel. Les performances augmentent surtout entre Nano, Small et Medium. La pente devient plus faible pour les tailles Large et Extra-Large.

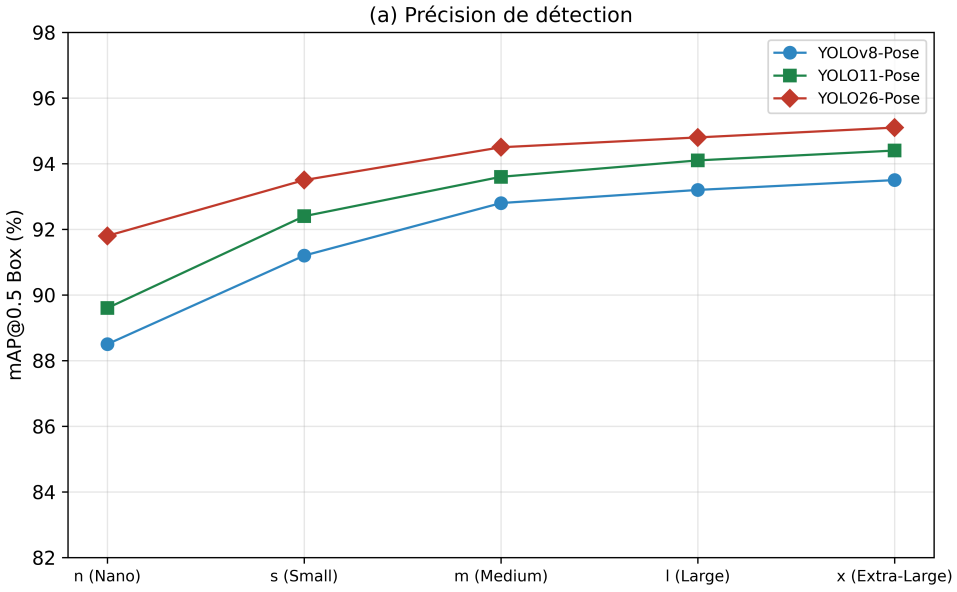


Figure 5.8 Précision de détection selon la famille et l'échelle du modèle sur le jeu de test Costco.

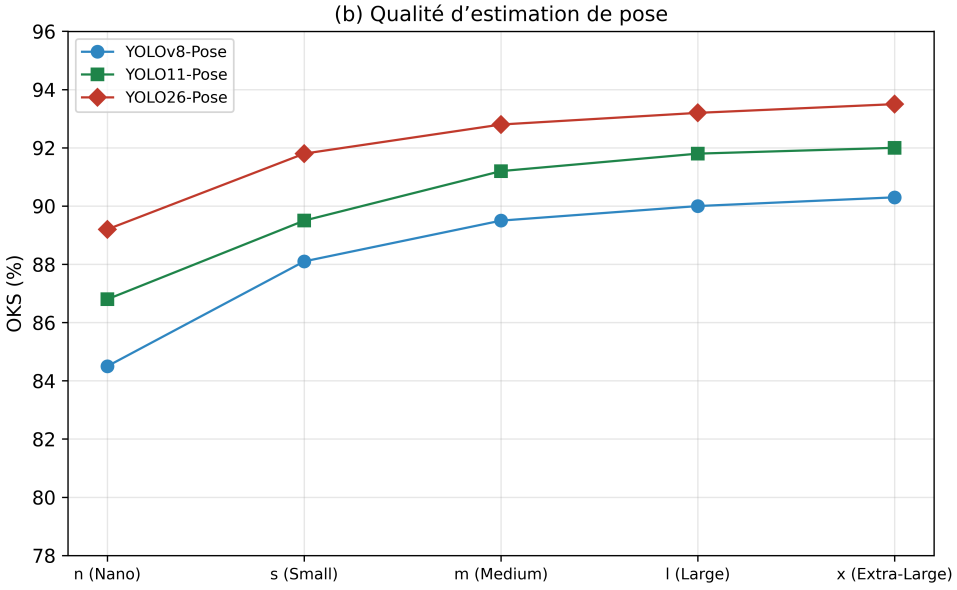


Figure 5.9 Qualité d'estimation de pose selon la famille et l'échelle du modèle sur le jeu de test Costco.

Interprétation. Le rendement décroissant selon l'échelle ne justifie pas le coût de calcul des modèles Large et Extra-Large pour l'embarqué. Le meilleur score hors ligne ne garantit pas non plus le meilleur comportement en boucle fermée, car une latence élevée réduit la réactivité de la navigation. Les tailles Nano et Small constituent donc les candidates pertinentes pour la simulation et le robot réel.

5.3 Confirmation sur le jeu de données Noovelia et étude de robustesse

5.3.1 Résultats qualitatifs avec la caméra du robot

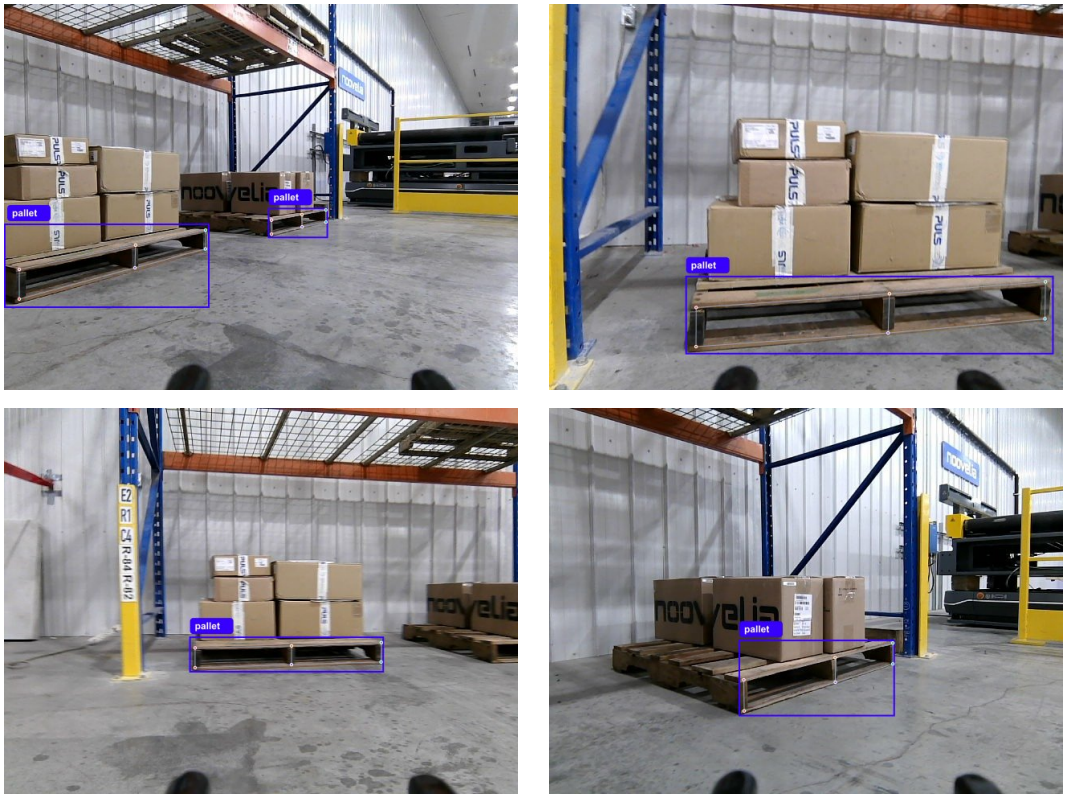


Figure 5.10 Exemples de détection sur des images réelles du jeu de test Noovelia captées par la caméra embarquée.

La figure 5.10 montre des détections obtenues sur des images réelles du jeu de test Noovelia captées par la caméra embarquée. La palette, la boîte englobante et les points clés sont

visibles dans plusieurs contextes d'arrière-plan. Ces résultats répondent à la validation qualitative du détecteur sur les images issues du système réel.

5.3.2 Compromis entre précision et latence sur le jeu de test Noovelia

La figure 5.11 compare les mêmes architectures et stratégies sur le jeu de test Noovelia de 45 images. yolo26n-pose atteint 91,2 % de *mAP*@0,5 dans la configuration hybride, contre 84,5 % avec Réel seul. yolo11n-pose atteint respectivement 88,9 % et 82,4 %.

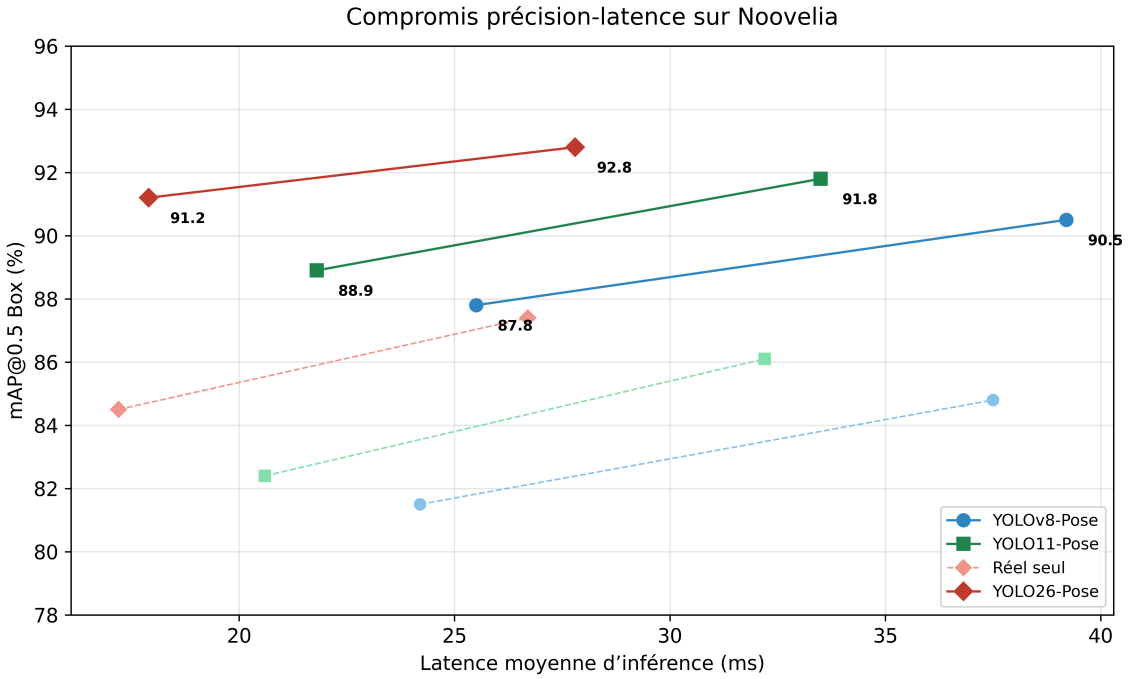


Figure 5.11 Compromis entre précision de détection et latence GPU sur le jeu de test Noovelia.

La figure 5.12 montre la même tendance pour la pose. Dans la configuration hybride, yolo26n-pose obtient 88,5 % d'OKS, yolo11n-pose 86,2 % et yolov8n-pose 83,8 %.

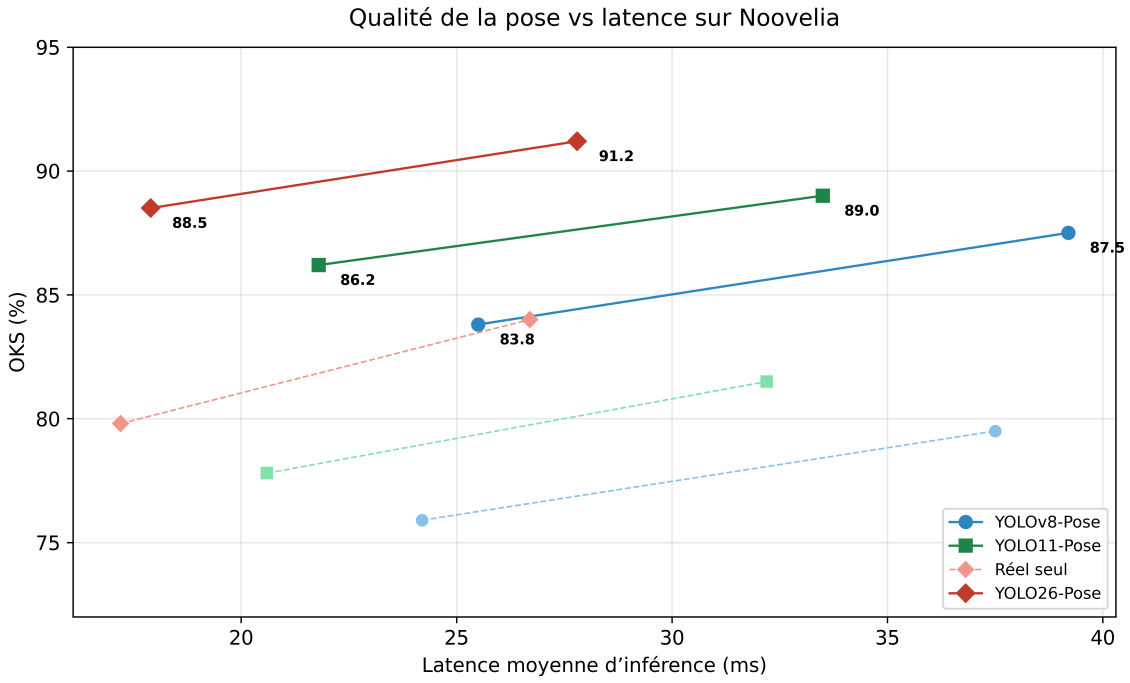


Figure 5.12 Compromis entre qualité d'estimation de pose et latence GPU sur le jeu de test Noovelia.

Interprétation. Les valeurs absolues diffèrent légèrement de celles de Costco, mais l'ordre des architectures et l'avantage de la stratégie hybride restent cohérents. Cette répétition sur un second jeu réel soutient la robustesse de la conclusion concernant l'apport du préentraînement synthétique. La transférabilité directe entre les sites est évaluée à la section 5.4.

5.3.3 Distribution des conditions difficiles

Dans cette analyse, l'expression *conditions dynamiques* désigne les variations visuelles susceptibles de modifier la qualité de détection, et non une propriété abstraite de mouvement continu de l'environnement. Elle couvre notamment les occlusions, les changements d'éclairage, les angles de vue, les distracteurs, le passage d'opérateurs et les vibrations de la caméra. Le tableau 5-2 présente la distribution des 45 images de test Noovelia selon

quatre catégories mutuellement exclusives. Lorsqu'une image contient plusieurs perturbations, elle est affectée à la catégorie dominante.

Condition environnementale	Critère d'inclusion	Effectif	Part
Occlusion partielle	Face avant masquée à plus de 30 %	12	26,7 %
Éclairage faible ou ombres	Intensité moyenne des pixels inférieure à 50	8	17,8 %
Angle de vue extrême	Angle d'incidence supérieur à 45°	10	22,2 %
Encombrement et distracteurs	Objets logistiques à moins de 0,5 m	15	33,3 %

Tableau 5-2 Distribution du jeu de test Noovelia par classe de perturbation.

5.3.4 Robustesse sous perturbations environnementales

La figure 5.13 compare yolo26n-pose sous les stratégies Réel seul et Synthétique puis Réel avec 25 000 images synthétiques et 210 images réelles. La stratégie hybride obtient des valeurs supérieures pour toutes les perturbations, avec les écarts les plus marqués sous les angles extrêmes et les occlusions. Les résultats numériques détaillés pour Costco sont fournis au tableau A.4 et ceux pour Noovelia au tableau A.5.

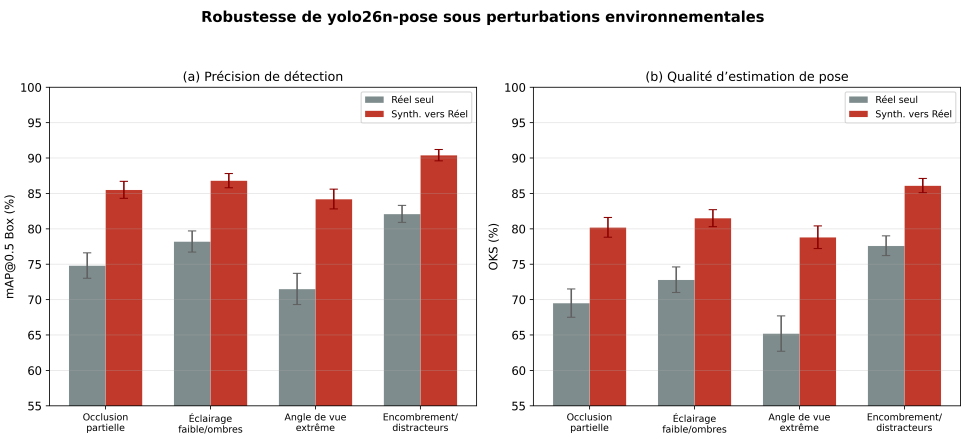


Figure 5.13 Robustesse de yolo26n-pose sous perturbations environnementales sur le jeu de test Noovelia.

Interprétation. La stratégie hybride surpasse systématiquement Réel seul pour chaque catégorie de perturbation. Ces résultats sont cohérents avec l'hypothèse **H2** : l'entraînement incluant des données synthétiques randomisées est associé à de meilleures performances dans les conditions environnementales difficiles évaluées. Les pertes temporaires de points clés restent néanmoins importantes pour la navigation, car une erreur locale peut se propager directement à l'estimation du point cible.

5.4 Résultats inter-sites

Le protocole décrit à la section 4.3.3 évalue chaque architecture Nano sur un site différent de celui utilisé pour l'ajustement réel, sans aucun ajustement sur le site cible. Cette validation est dite *complémentaire inter-sites* afin de la distinguer d'une validation croisée statistique de type *k*-fold.

5.4.1 Comparaison des deux directions de transfert

Le tableau 5-3 et la figure 5.14 présentent les *mAP*@0,5 (Mean Average Precision) pour la boîte englobante et l'OKS (Object Keypoint Similarity) pour les trois architectures Nano,

dans les deux directions (Costco vers Noovelia et Noovelia vers Costco), sous les stratégies Réel seul et Synthétique puis Réel. Le tableau détaille pour chaque modèle Nano les scores obtenus selon le site d'entraînement et le site de test, ainsi que selon la stratégie d'entraînement. Les valeurs rapportées correspondent à la moyenne de trois répétitions indépendantes (variant la graine d'entraînement), ce qui permet de comparer l'effet du préentraînement synthétique ainsi que l'asymétrie entre directions.

Modèle	**Site d'entraînement**	**Site de test**	**Réel seul**		**Synthétique puis Réel**	
			mAP@0,5	**OKS**	**mAP@0,5**	**OKS**
yolov8n-pose	Costco	Noovelia	38,5 %	35,2 %	45,2 %	41,8 %
yolov8n-pose	Noovelia	Costco	40,8 %	37,5 %	47,5 %	43,8 %
yolo11n-pose	Costco	Noovelia	42,8 %	39,5 %	49,5 %	46,2 %
yolo11n-pose	Noovelia	Costco	45,1 %	41,8 %	52,2 %	48,8 %
yolo26n-pose	Costco	Noovelia	44,2 %	40,8 %	51,8 %	48,5 %
yolo26n-pose	Noovelia	Costco	46,5 %	43,2 %	54,5 %	51,2 %

Tableau 5-3 Validation inter-sites des architectures Nano.

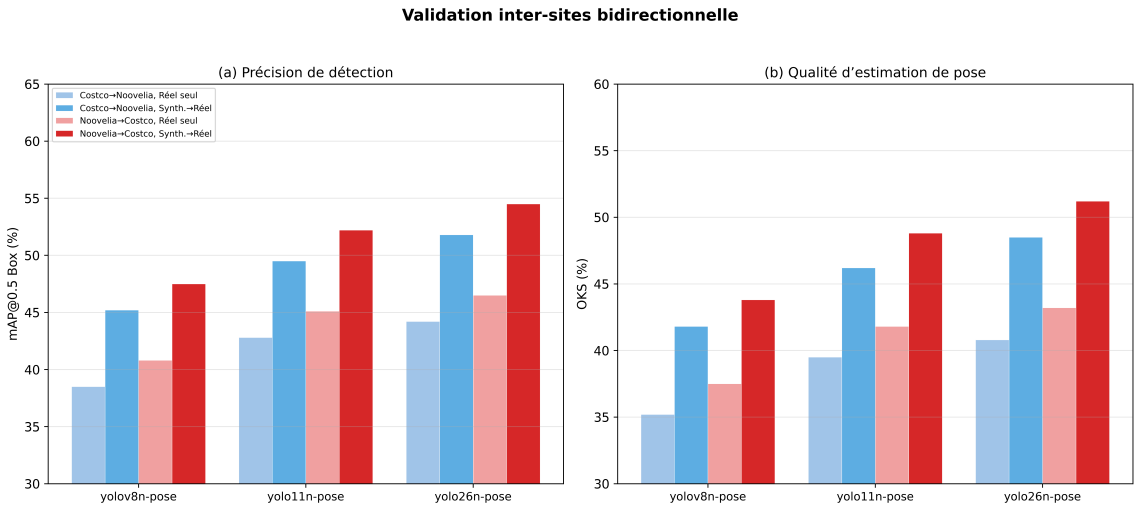


Figure 5.14 Performances inter-sites des architectures Nano.

5.4.2 Analyse des gains apportés par le préentraînement synthétique

Dans la direction Costco vers Noovelia, le préentraînement synthétique fait passer le *mAP*@0,5 de 42,8 % à 49,5 % pour yolo11n-pose, soit un gain de 6,7 points. Dans la direction Noovelia vers Costco, le score passe de 45,1 % à 52,2 %, soit un gain de 7,1 points. La même tendance est observée pour yolov8n-pose (gain de 6,7 points vers Noovelia, 6,7 points vers Costco) et yolo26n-pose (gain de 7,6 points vers Noovelia, 8,0 points vers Costco). Toutes les architectures Nano bénéficient donc du préentraînement synthétique en transfert inter-sites.

5.4.3 Persistance de l'asymétrie entre les deux directions

La stratégie Réel seul présente un écart systématique entre les deux directions : le transfert Noovelia vers Costco est meilleur que Costco vers Noovelia, avec des écarts de 2,3 points (yolov8n-pose), 2,3 points (yolo11n-pose) et 2,3 points (yolo26n-pose) en *mAP*@0,5. Cette asymétrie s'explique par la plus grande diversité visuelle du jeu de données Noovelia, qui prépare mieux le modèle aux variations présentes sur le jeu de test Costco.

Avec la stratégie hybride, les écarts restent du même ordre : 2,3 points (yolov8n-pose), 2,7 points (yolo11n-pose) et 2,7 points (yolo26n-pose). L'asymétrie n'est donc pas réduite par le préentraînement synthétique. En revanche, les niveaux absolus sont supérieurs de 5 à 8 points par rapport à la stratégie Réel seul dans les deux directions, ce qui démontre que le préentraînement synthétique améliore les performances de transfert sans toutefois supprimer l'écart de domaine persistant entre les sites.

5.4.4 Comparaison avec les performances intra-site

Le tableau 5-3 permet également de quantifier la perte due au changement de site par rapport aux performances intra-site (section 5.1.1). Pour yolo26n-pose en configuration hybride, le $mAP@0{,}5$ passe de 91,8 % (intra-site Costco) à 51,8 % (Costco vers Noovelia), soit une baisse de 40,0 points. Cette chute est comparable pour yolo11n-pose (89,6 % à 49,5 %, soit 40,1 points) et yolov8n-pose (88,5 % à 45,2 %, soit 43,3 points). L'écart de domaine entre les deux sites est donc considérable, et le préentraînement synthétique ne suffit pas à le combler. Le gain par rapport à la stratégie Réel seul (6 à 8 points selon l'architecture) indique néanmoins que la diversité synthétique améliore les performances de transfert inter-sites dans le protocole évalué.

Interprétation. La validation inter-sites confirme que le préentraînement synthétique améliore les performances en transfert entre les deux sites étudiés, quelle que soit l'architecture retenue. Les gains absolus de 6 à 8 points de $mAP@0{,}5$ par rapport à la stratégie Réel seul montrent que la diversité synthétique atténue partiellement l'écart de domaine. yolo26n-pose conserve le meilleur niveau absolu dans les deux directions. L'écart de domaine entre les deux sites reste toutefois élevé (perte de 40 à 43 points par rapport aux performances intra-site), ce qui souligne la nécessité d'un ajustement fin sur le site cible pour des applications opérationnelles. La conclusion demeure limitée à ces deux domaines et aux jeux de test de 45 images ; elle ne démontre pas une généralisation universelle à tout environnement industriel.

5.5 Validation applicative en boucle fermée

5.5.1 Compatibilité temporelle sous PyBullet

Le tableau 5-4 présente les mesures d'efficience des architectures Nano et Small entraînées selon la stratégie Synthétique puis Réel. Toutes les configurations testées satisfont la contrainte de temps réel dans l'environnement PyBullet : pour chaque modèle, le nombre d'images traitées par seconde (FPS) se situe entre 28 et 55, et la latence moyenne varie de 18,1 ms à 35,2 ms, assurant une exécution compatible avec les exigences du contrôle en boucle fermée. Les résultats présentés correspondent à la moyenne de trois répétitions.

Architecture	**Stratégie**	$mAP@0{,}5$ **Costco**	**FPS**	**Latence moy.**	**Temps réel**
yolov8n-pose	Synthétique puis Réel	88,5 %	42	23,5 ms	Compatible
yolov8s-pose	Synthétique puis Réel	91,2 %	28	35,2 ms	Compatible
yolo11n-pose	Synthétique puis Réel	89,6 %	48	20,6 ms	Compatible
yolo11s-pose	Synthétique puis Réel	92,4 %	32	31,1 ms	Compatible
yolo26n-pose	Synthétique puis Réel	91,8 %	55	18,1 ms	Compatible
yolo26s-pose	Synthétique puis Réel	93,5 %	37	26,8 ms	Compatible

Tableau 5-4 Compatibilité temporelle sous PyBullet

yolo26n-pose est retenu pour l'alignement sous PyBullet, puis déployé sur le robot réel où sa latence plus faible est également avantageuse.

5.5.2 Précision de l'alignement en simulation

Les performances d'alignement au point cible avec yolo26n-pose ($n = 20$ essais) sont résumées dans le tableau 5-5. En simulation PyBullet, l'erreur moyenne sur l'axe latéral est de 1,52 cm (écart-type 0,72 cm ; médiane 1,41 cm ; min 0,28 cm ; max 3,12 cm), tandis que

l'erreur en profondeur atteint 2,15 cm en moyenne (écart-type 1,02 cm ; médiane 2,01 cm ; min 0,35 cm ; max 4,48 cm). L'erreur angulaire moyenne est de $0,98^\circ$ (écart-type $0,48^\circ$; médiane $0,91^\circ$; min $0,15^\circ$; max $2,05^\circ$) et le temps d'alignement stabilisé s'établit à 5,82 s (écart-type 1,25 s). Le taux de perte de détection atteint 1,80 % en moyenne.

Indicateur au point cible	**Moyenne**	**Écart-type**	**Médiane**	**Min**	**Max**
ΔX_{align} **[cm]**	1,52	0,72	1,41	0,28	3,12
ΔY_{align} **[cm]**	2,15	1,02	2,01	0,35	4,48
$\Delta\theta_{align}$ **[degrés]**	0,98	0,48	0,91	0,15	2,05
Temps d'alignement stabilisé [s]	5,82	1,25	5,67	3,42	8,15
Taux de perte de détection [%]	1,80	0,90	1,60	0,30	3,90

Tableau 5-5 Mesures d'alignement en simulation

En simulation, 16 essais sur 20 (80 %) respectent simultanément les trois seuils de tolérance ($|\Delta X_{align}| \leq 3{,}0$ cm, $|\Delta Y_{align}| \leq 5{,}0$ cm, $|\Delta\theta_{align}| \leq 2{,}5^\circ$). Le percentile 95 de ΔX_{align} est de 2,85 cm, sous la barre des 3,0 cm, mais le maximum observé (3,12 cm) dépasse ce seuil. La proportion de respect par variable est de 17/20 pour ΔX_{align}, 18/20 pour ΔY_{align} et 19/20 pour $\Delta\theta_{align}$.

5.5.3 Sessions sur robot réel

Nous avons réalisé plusieurs essais d'alignement sur robot réel avec la caméra embarquée, couvrant trois architectures Nano et diverses conditions visuelles. La session 01 (**yolov8n-pose**) compte 19 réussites sur 20, tandis que les sessions 02 (**yolo11n-pose**) et 03a/03b (**yolo26n-pose**) comptent chacune 20 réussites sur 20. Les sessions 03a et 03b permettent une comparaison directe des stratégies d'apprentissage : la première utilise des poids en-

traînés uniquement sur des données réelles, la seconde applique un préentraînement sur des données synthétiques suivi d'un ajustement sur des données réelles (voir tableau 5-8).

Session	Modèle	Stratégie	Essais	Réussites
01	yolov8n-pose	Synthétique puis Réel	20	19
02	yolo11n-pose	Synthétique puis Réel	20	20
03a	yolo26n-pose	Réel seul	20	20
03b	yolo26n-pose	Synthétique puis Réel	20	20

Tableau 5-6 Résumé des essais de détection et d'alignement sur robot réel

5.5.4 Détection et latence embarquées

Le tableau 5-7 synthétise les mesures d'efficacité embarquée en matière de fréquence d'images, de latence et de robustesse de détection, obtenues lors de l'exécution des architectures yolov8n-pose, yolo11n-pose et yolo26n-pose sur le robot ROSMASTER M3 Pro. Parmi celles-ci, yolo26n-pose atteint la meilleure performance avec 18,5 FPS, une latence moyenne de 53,5 ms, une latence maximale de 78 ms et un taux de détections valides de 99,0 %. yolo11n-pose atteint quant à lui 15,8 FPS, 62,8 ms de latence moyenne et 98,0 % de détections valides. Ces résultats démontrent la robustesse du système en conditions réelles variées et confirment l'efficacité de la détection au-delà des jeux de données hors ligne.

Architecture	FPS moyen	Latence moy.	Latence max.	Détections valides
yolov8n-pose	14,2	70,1 ms	95 ms	95,0 %
yolo11n-pose	15,8	62,8 ms	88 ms	98,0 %
yolo26n-pose	18,5	53,5 ms	78 ms	99,0 %

Tableau 5-7 Efficience embarquée

5.5.5 Précision géométrique au point cible sur robot réel

Le tableau 5-8 compare deux versions du même modèle **yolo26n-pose** sur $n = 20$ essais chacune (sessions 03a et 03b). La première version est ajustée uniquement sur les données réelles, tandis que la seconde est préentraînée sur 25 000 images synthétiques avant un ajustement sur les mêmes données réelles. L'architecture, la caméra et la boucle de navigation sont identiques pour isoler l'effet de la stratégie d'apprentissage. Les variables de performance (ΔX_{align}, ΔY_{align}, $\Delta\theta_{align}$, taux de perte) sont présentées avec leurs moyennes et leurs écarts-types, les seuils de tolérance ainsi que la proportion d'essais respectant ces seuils pour chaque stratégie.

Variable de performance	Réel seul	Synthétique puis Réel	Tolérance	Respect (Réel seul)	Respect (Synthétique puis Réel)
ΔX_{align}	$3{,}20 \pm 1{,}50$ cm	$1{,}95 \pm 0{,}85$ cm	$\lvert\Delta X_{align}\rvert \leq 3{,}0$ cm	12/20 (60 %)	18/20 (90 %)
ΔY_{align}	$4{,}10 \pm 2{,}20$ cm	$2{,}40 \pm 1{,}15$ cm	$\lvert\Delta Y_{align}\rvert \leq 5{,}0$ cm	16/20 (80 %)	19/20 (95 %)
$\Delta\theta_{align}$	$2{,}80 \pm 1{,}40°$	$1{,}20 \pm 0{,}60°$	$\lvert\Delta\theta_{align}\rvert \leq 2{,}5°$	13/20 (65 %)	19/20 (95 %)
Taux de perte	4,8 %	1,2 %	≤ 5 %	17/20 (85 %)	20/20 (100 %)

Tableau 5-8 Résultats d'alignement géométrique sur robot réel

Sous la stratégie Synthétique puis Réel, 17 essais sur 20 (85 %) respectent simultanément les trois seuils géométriques (ΔX_{align}, ΔY_{align}, $\Delta\theta_{align}$), contre 9 sur 20 (45 %) pour la stratégie Réel seul. La réduction la plus forte concerne l'erreur angulaire, dont la proportion

de respect passe de 65 % à 95 %. Le taux de perte diminue de 4,8 % à 1,2 %, avec 100 % des essais sous le seuil de 5 % dans la configuration hybride.

Note sur la latence. La latence totale par cycle mesurée diffère entre les deux stratégies ($78{,}3 \pm 5{,}2$ ms pour Réel seul contre $67{,}77 \pm 1{,}11$ ms pour Synthétique puis Réel). Cette différence ne peut toutefois pas être attribuée directement au préentraînement synthétique : l'architecture, la résolution et le matériel étant identiques, les poids appris ne modifient pas le nombre d'opérations du réseau. L'écart observé peut provenir de facteurs non contrôlés (variabilité inter-session, échauffement GPU, nombre de détections par trame, états transitoires du système de communication). Aucune conclusion scientifique sur un bénéfice de latence du transfert synthétique n'est donc tirée de cette mesure.

Interprétation de la validation applicative. La simulation valide la communication entre la perception et la commande dans un environnement contrôlé. Le robot réel ajoute les vibrations de la caméra, les variations d'éclairage, les délais de communication et la dynamique mécanique. Malgré ces différences, les résultats de PyBullet et du robot montrent la même tendance générale : une détection plus stable s'accompagne d'erreurs d'alignement et de taux de perte plus faibles.

Ces indicateurs ne valident pas une contribution algorithmique originale sur le contrôle du robot. La frontière du système développé est la sortie du détecteur, soit la boîte englobante et les six points clés de la palette. Le module géométrique, la planification et la commande prennent ensuite le relais dans la solution existante et demeurent hors du périmètre du projet. Les indicateurs d'alignement mesurent donc uniquement l'effet du bloc de perception lorsqu'il alimente ce système aval. La capacité des données synthétiques à encoder la géométrie de la palette fournit une explication cohérente de la réduction de $\Delta\theta_{align}$.

5.6 Synthèse des hypothèses et des jalons

Le tableau 5-9 relie chaque question expérimentale aux résultats qui permettent d'y répondre.

Question expérimentale	Résultat principal	Preuves associées
L'apport du préentraînement synthétique est-il soutenu ?	Oui, surtout avec peu de données réelles	Figures 5.1, 5.2 et 5.4
À partir de quel volume synthétique les gains diminuent-ils ?	À partir de 10 000 images	Figure 5.3
Quelle architecture offre le meilleur compromis embarqué ?	yolo26n-pose	Tableaux 5-4 et 5-7
Quel modèle est utilisé pour l'alignement sous PyBullet ?	yolo26n-pose	Tableau 5-5
Quel modèle compare les stratégies sur robot réel ?	yolo26n-pose	Tableau 5-8
Le transfert entre Costco et Nooveliа est-il soutenu ?	Oui, dans les deux directions évaluées	Tableau 5-3

Tableau 5-9 Synthèse des réponses aux questions expérimentales.

Les hypothèses **H1**, **H2**, **H3** et **H4** sont soutenues dans les conditions expérimentales du mémoire. La portée de cette conclusion reste liée aux deux jeux réels étudiés, aux architectures évaluées et aux protocoles décrits au chapitre 4.

5.7 Limites et menaces à la validité

Le travail porte sur une seule classe de palette en bois standard. Chaque jeu de test réel contient 45 images, ce qui limite la portée statistique des conclusions. Les écarts-types cal-

culés sur trois répétitions (variant la graine d'entraînement) réduisent partiellement l'incertitude liée à l'optimisation, mais ne capturent pas l'incertitude due à l'échantillonnage des petits sous-ensembles d'entraînement (21 ou 31 images), dont la composition est figée pour toutes les répétitions. La caméra monoculaire ne mesure pas directement la profondeur et rend l'estimation dépendante de la calibration et de la géométrie observée.

La validation inter-sites est limitée à deux domaines et à 45 images de test par direction. Elle montre un transfert bidirectionnel entre Costco et Noovelia sans adaptation au site cible, mais ne suffit pas à établir la validité externe pour d'autres entrepôts, capteurs, palettes ou distributions visuelles.

La validité interne peut être affectée par la subjectivité de l'annotation des points clés, le choix des hyperparamètres, la stratégie d'augmentation et la sélection des images. Les mêmes partitions, métriques et conditions d'entraînement sont utilisées pour limiter ces effets. La validité externe reste limitée par les caractéristiques du capteur, le type de palette et les environnements observés.

La reproductibilité externe est partielle. Le mémoire décrit les partitions, les graines, les hyperparamètres, les métriques, la séquence expérimentale et les versions logicielles nécessaires pour reproduire les entraînements et les évaluations hors ligne dans des conditions comparables. En revanche, la reproduction intégrale de la chaîne dépend aussi du matériel de calcul et du robot, de la caméra et de sa calibration, des conditions propres aux environnements industriels, ainsi que du module de contrôle fourni par l'équipe de la Chaire de recherche Noovelia. Ces dépendances matérielles et contextuelles limitent la possibilité, pour un lecteur extérieur, de reconstruire à l'identique la validation en boucle fermée.

5.8 Recommandations pour l'intégration industrielle

Dans le protocole étudié, un jeu synthétique diversifié d'environ 20 000 images, combiné à une trentaine d'images réelles annotées du site cible, constitue un point de départ efficace. Un volume de 25 000 images peut rester pertinent lorsque les données réelles sont très rares. yolo26n-pose offre le meilleur compromis observé entre précision, latence et compatibilité embarquée.

Pour un nouveau site, la procédure recommandée consiste à générer un jeu synthétique couvrant les variations attendues, annoter un petit ensemble réel représentatif, effectuer l'ajustement fin, puis vérifier séparément la perception hors ligne et l'alignement en boucle fermée. Les tolérances opérationnelles doivent être fixées avant les essais et rattachées à une exigence géométrique ou de sécurité documentée.

5.9 Synthèse du chapitre

Le préentraînement synthétique est surtout bénéfique lorsque les données réelles sont rares. Cet effet apparaît sur Costco et Noovelia, sous plusieurs perturbations et jusqu'aux essais sur robot réel, puis se stabilise lorsque les volumes synthétiques ou réels augmentent. Les modèles légers offrent le meilleur compromis, yolo26n-pose servant de référence aux ablations, à la simulation et à la validation réelle.

La stratégie réduit l'effort d'annotation, stabilise la perception et améliore systématiquement le transfert inter-sites Costco–Noovelia dans les deux directions par rapport à Réel seul (gain de 6 à 8 points de $mAP@0{,}5$). Toutefois, les scores de transfert (45–55 %) restent insuffisants sans ajustement au site cible, contre 88–92 % en intra-site. Une validation sur davantage de sites et de palettes, jusqu'à l'insertion industrielle des fourches, demeure nécessaire.

Chapitre 6 - Conclusion et perspectives

6.1 Rappel de la problématique

Ce mémoire a abordé le verrou de l'acquisition de données réelles annotées pour la détection visuelle de palettes industrielles. En milieu logistique, les robots mobiles doivent reconnaître les palettes et se positionner face à elles afin de préparer l'approche finale. L'annotation manuelle de boîtes englobantes et de points clés sur des milliers d'images réelles est coûteuse, longue et difficile à généraliser à plusieurs sites industriels.

6.2 Résumé de la méthode déployée

La démarche adoptée suit un flux en trois étapes : la génération automatique d'images synthétiques annotées sous Blender avec randomisation de domaine, l'entraînement et la comparaison de détecteurs YOLO-Pose de plusieurs générations (YOLOv8-Pose, YOLO11-Pose, YOLO26-Pose) à différentes échelles, puis la transmission des sorties de perception au code de navigation existant. L'évaluation combine des tests hors ligne nominaux sur les jeux de test Costco et Noovelia, une validation complémentaire inter-sites bidirectionnelle sans ajustement sur le site cible, puis des essais d'intégration en simulation (PyBullet) et sur robot réel (ROSMASTER M3 Pro).

6.3 Condensé des conclusions scientifiques

Les quatre hypothèses initiales de recherche reçoivent les réponses suivantes :

- **H1 (Apport du transfert synthétique vers réel) :** l'hypothèse est soutenue par un appui empirique solide. À volume égal de données réelles annotées, le préentraînement synthétique surpasse l'entraînement à partir de COCO sur des données réelles seules

- **H2 (Apport de l'entraînement hybride à la robustesse) :** l'hypothèse est soutenue, avec la réserve que l'absence de condition nominale dans le jeu de test ne permet pas de mesurer une dégradation relative causale. La stratégie hybride surpasse systématiquement Réel seul sous perturbations environnementales réelles (occlusions, éclairage faible, angles de vue extrêmes)
- **H3 (Rendement décroissant des données réelles) :** l'hypothèse est soutenue. Après le préentraînement synthétique, l'ajout de données réelles produit des gains importants aux faibles volumes, puis des gains de plus en plus faibles. Ce ralentissement s'amorce dès 50 % du jeu d'entraînement réel et devient pleinement net au-delà de 70 %
- **H4 (Rendement décroissant du volume synthétique) :** l'hypothèse est soutenue. Dans le protocole observé, les rendements décroissants apparaissent dès 10 000 images synthétiques, et une zone de quasi-saturation est atteinte autour de 20 000 images, au-delà de laquelle les gains deviennent marginaux

La validation complémentaire inter-sites confirme cette tendance entre les deux domaines étudiés. Le modèle yolo26n-pose ajusté sur Costco puis testé sur Noovelia atteint 51,8 % de $mAP@0{,}5$ avec la stratégie Synthétique puis Réel, contre 44,2 % avec Réel seul. Dans la direction inverse, entraînement sur Noovelia puis test sur Costco, les scores sont respectivement de 54,5 % et 46,5 %. Il s'agit d'une évaluation de transfert entre sites, et non d'une validation croisée statistique.

6.4 Contributions et portée du travail

Ce mémoire étudie une méthodologie visant à réduire l'effort d'annotation nécessaire pour calibrer les blocs de perception industrielle. Les contributions couvrent la génération de données synthétiques sous Blender avec annotation automatique, l'étude compa-

rative multi-générationnelle et multi-échelle des modèles YOLO-Pose, la quantification du transfert synthétique vers réel avec analyse de sensibilité aux fractions de données, la validation complémentaire inter-sites dans les deux directions, l'analyse de la robustesse sous perturbations industrielles, et la vérification applicative de l'intégration en temps réel sur robot mobile au sein d'une boucle fermée de navigation existante.

La portée de la solution s'arrête explicitement à la sortie du détecteur : la boîte englobante 2D de la palette et les six points clés associés. L'estimation géométrique de la pose, la planification de trajectoire, la commande du robot et l'insertion des fourches sont assurées par un autre système et demeurent hors du périmètre de la contribution. Les essais en boucle fermée démontrent l'exploitabilité des sorties de perception ; ils ne revendiquent pas une nouvelle méthode de navigation.

6.5 Limites du projet

Le mémoire qualifie les performances du détecteur et n'inclut pas de modification algorithmique de la navigation. La validation réelle avec le ROSMASTER M3 Pro couvre la navigation d'approche jusqu'à l'arrêt devant la palette, sans insertion des fourches sur un chariot élévateur industriel. Le volume de données réelles reste limité (300 images par site) ; chaque direction inter-sites repose sur 45 images de test ; le capteur RGB ne fournit pas directement la profondeur ; la généralisation au-delà des sites Costco et Noovelia, à d'autres types de palettes ou à d'autres capteurs reste une question ouverte.

6.6 Perspectives de recherche

Plusieurs pistes prolongent ce travail. Les données synthétiques pourraient bénéficier d'une randomisation plus avancée de la lumière, de modèles 3D intégrant davantage de variations d'usure et de défauts, ainsi que d'un ajustement automatique des paramètres

de génération selon les erreurs observées sur les données réelles. Sur le plan embarqué, la quantification INT8 et la conversion TensorRT pourraient stabiliser le débit d'images et réduire la consommation d'énergie des plateformes Jetson. Des filtres temporels légers entre l'interface de programmation d'application (API) de détection et la navigation permettraient aussi de lisser l'alignement lors de brèves pertes visuelles. Enfin, le protocole inter-sites gagnerait à être répété dans plusieurs environnements industriels, sur différents types de palettes et avec des jeux de test plus volumineux, avant une validation de l'insertion des fourches sur un chariot élévateur autonome. Une collecte d'images supplémentaires en production ouvrirait également la voie à l'adaptation continue du modèle aux particularités de chaque site.

En définitive, la contribution centrale de ce travail est une démarche intégrée qui relie la génération de données synthétiques, l'apprentissage de la perception et son exploitation par une chaîne robotique. Elle montre qu'il est possible de réduire le besoin en données réelles annotées tout en conservant une validation progressive jusqu'à l'intégration robotique. La portée de cette validation demeure circonscrite aux conditions étudiées. Sa consolidation exigera donc des essais sur davantage de sites, de contextes visuels et de types de palettes.

Références

[1] X. Xu, T. Wu, Z. Du, H. Rong, S. Wang, S. Li, and D. Chen, "Enhanced human pose estimation using YOLOv8 with integrated SimDLKA attention mechanism and DCIOU loss function: Analysis of human body behavior and posture," *PLOS ONE*, vol. 20, no. 5, p. e0318578, 2025.

[2] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "CRISP-DM 1.0: Step-by-step data mining guide," SPSS, Chicago, IL, USA, Tech. Rep. CRISPDM-1, 2000, [En ligne]. Disponible sur : https://www.dataprix.com/files/CRISP-DM.pdf. Consulté le 5 septembre 2025.

[3] H. Kagermann, W. Wahlster, and J. Helbig, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of german manufacturing industry," Forschungsunion / acatech, Frankfurt am Main, Final Report of the Industrie 4.0 Working Group, 2013, [En ligne]. Disponible sur : https://en.acatech.de/publication/recommendations-for-implementing-the-strategic-initiative-industrie-4-0-final-report-of-the-industrie-4-0-working-group/. Consulté le 9 août 2026.

[4] S. G. Tzafestas, "Mobile robot control and navigation: A global overview," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 1, pp. 35–58, 2018.

[5] M. A. Fraifer, J. Coleman, J. Maguire, P. Trslic, G. Dooly, and D. Toal, "Autonomous forklifts: State of the art — exploring perception, scanning technologies and functional systems — a comprehensive review," *Electronics*, vol. 14, no. 1, p. 153, 2025.

[6] A. D. Patil and S. Farzan, "Autonomous forklifts for warehouse automation: A comprehensive review," *Robotics*, vol. 15, no. 2, p. 30, 2026.

[7] B. Molter and J. Fottner, "Real-time pallet localization with 3D camera technology for forklifts in logistic environments," in *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, 2018, pp. 297–302.

[8] I. S. Mohamed, A. Capitanelli, F. Mastrogiovanni, S. Rovetta, and R. Zaccaria, "Detection, localisation and tracking of pallets using machine learning techniques and 2d range data," *Neural Computing and Applications*, vol. 32, no. 13, pp. 8811–8828, 2020.

[9] J. Xiao, H. Lu, L. Zhang, and J. Zhang, "Pallet recognition and localization using an RGB-D camera," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 1729881417737799, 2017.

[10] J. Zhao, B. Li, X. Wei, H. R. Lu, E. Lü, and X. Zhou, "Recognition and location algorithm for pallets in warehouses using RGB-D sensor," *Applied Sciences*, vol. 12, no. 20, p. 10331, 2022.

[11] R. Iinuma, Y. Hori, H. Onoyama, T. Fukao, and Y. Kubo, "Pallet detection and estimation for fork insertion with RGB-D camera," in *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2021, pp. 854–859.

[12] D. Fang, J. Han, J. Shao, and C. Zhao, "Pallet detection and localization based on RGB image and point cloud data for automated forklift," in *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*. IEEE, 2024, pp. 642–647.

[13] N. Vodichev, D. Gavrilov, A. Leus, V. Efremov, A. Troegubov, I. Kholodnyak, V. Zuev, M. Parshikov, A. Antipov, V. Laukhin, and N. Gershtein, "Real-time distance estimation algorithm for objects in warehouse based on monocular camera data for an autonomous unmanned forklift," *The European Physical Journal Special Topics*, vol. 234, no. 15, pp. 3883–3895, 2025.

[14] E. S. Kesuma, P. H. Rusmin, and D. A. Maharani, "Pallet detection and distance estimation with YOLO and fiducial marker algorithm in industrial forklift robot," in *2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2023, pp. 270–275.

[15] Z. Zhou, Y. Lu, and L. Lv, "Pallet localization algorithm based on improved human pose estimation with transfer learning," *The Journal of Supercomputing*, vol. 81, no. 3, p. 471, 2025.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, vol. 28, 2015, pp. 91–99.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020, [En ligne]. Disponible sur : https://arxiv.org/abs/2004.10934. Consulté le 17 février 2025.

[19] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023, [En ligne]. Disponible sur : https://github.com/ultralytics/ultralytics. Consulté le 6 mars 2025.

[20] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," in *European Conference on Computer Vision*, 2024.

[21] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024, [En ligne]. Disponible sur : https://arxiv.org/abs/2405.14458. Consulté le 22 avril 2025.

[22] G. Jocher and J. Qiu, "Ultralytics YOLO11," 2024, [En ligne]. Disponible sur : https://github.com/ultralytics/ultralytics. Consulté le 14 mai 2025.

[23] J. R. Terven and D. M. Cordova-Esparza, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[24] M. Zaccaria, R. Monica, and J. Aleotti, "A comparison of deep learning models for pallet detection in industrial warehouses," in *2020 IEEE 16th International Confe-*

rence on Intelligent Computer Communication and Processing (ICCP), 2020, pp. 417–422.

[25] P. Manurung, P. H. Rusmin, and R. Yusuf, "Custom pallet detection using YOLOv5 deep learning architecture," in *2022 International Symposium on Electronics and Smart Devices (ISESD)*, 2022, pp. 1–6.

[26] B. Wu, S. Wang, Y. Lu, Y. Yi, D. Jiang, and M. Qiao, "A new pallet-positioning method based on a lightweight component segmentation network for AGV toward intelligent warehousing," *Sensors*, vol. 25, no. 7, p. 2333, 2025.

[27] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[28] G. Jocher and J. Qiu, "Ultralytics YOLO26," 2026, [En ligne]. Disponible sur : https://github.com/ultralytics/ultralytics. Consulté le 11 février 2026.

[29] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020*, 2020, pp. 213–229.

[30] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "DETRs beat YOLOs on real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 965–16 974.

[31] H. Mueller, Y. Kim, T. Gee, and M. Nejati, "Pallet detection and localisation from synthetic data," *arXiv preprint arXiv:2503.22965*, 2025, [En ligne]. Disponible sur : https://arxiv.org/abs/2503.22965. Consulté le 28 septembre 2025.

[32] Blender Foundation, "Blender — a 3D modelling and rendering package," 2024, [En ligne]. Disponible sur : https://www.blender.org/. Consulté le 9 août 2026.

[33] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[34] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 969–977.

[35] M. Knitt, J. Schyga, A. Adamanov, J. Hinckeldeyn, and J. Kreutzfeldt, "Estimating the pose of a Euro pallet with an RGB camera based on synthetic training data," *Logistics Journal: Proceedings*, vol. 18, 2022.

[36] C. Beleznai, L. Reisinger, W. Pointner, and M. Murschitz, "Pallet detection and 3D pose estimation via geometric cues learned from synthetic data," in *Proceedings of the 4th International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI 2024)*. Springer, 2024, pp. 281–295.

[37] H. Gann, J. Bull, T. Gee, and M. Nejati, “Improving pallet detection using synthetic data,” *arXiv preprint arXiv:2402.07098*, 2024, [En ligne]. Disponible sur : https://arxiv.org/abs/2402.07098. Consulté le 9 juillet 2025.

[38] N. Baumgart, M. Lange-Hegermann, and M. Mücke, “Investigation of the impact of synthetic training data in the industrial application of terminal strip object detection,” *Machine Vision and Applications*, vol. 37, no. 1, p. 13, 2026.

[39] E. Coumans and Y. Bai, “PyBullet, a Python module for physics simulation for games, robotics and machine learning,” 2016–2021, [En ligne]. Disponible sur : https://pybullet.org/. Consulté le 10 septembre 2025.

Annexe A - Résultats numériques complets et hyperparamètres

Cette annexe regroupe les tableaux numériques exhaustifs complétant les analyses du chapitre 5, ainsi que les détails des configurations d'entraînement et de robustesse.

A.1 Tableaux exhaustifs des configurations d'entraînement

A.1.1 Jeu de test Costco

Le tableau A.1 présente les performances de perception hors ligne mesurées pour le modèle de référence **yolo26n-pose** sur le jeu de test Costco pour l'ensemble des 35 configurations de l'étude d'ablation croisant les volumes synthétiques de préentraînement et les fractions de données réelles. Les valeurs sont présentées sous la forme moyenne $\pm$ écart-type pour $n = 3$ répétitions (avec variation de la graine d'entraînement).

Vol. synth.	**Frac. réelle**	**Img.**	**mAP@0,5 (%)**	**OKS (%)**
0 (Réel seul)	10 %	21	$50{,}4 \pm 1{,}5$	$42{,}2 \pm 2{,}1$
0 (Réel seul)	15 %	31	$52{,}2 \pm 1{,}3$	$46{,}5 \pm 1{,}9$
0 (Réel seul)	30 %	63	$60{,}6 \pm 1{,}0$	$53{,}8 \pm 1{,}5$
0 (Réel seul)	50 %	105	$70{,}8 \pm 0{,}8$	$65{,}1 \pm 1{,}2$
0 (Réel seul)	70 %	147	$79{,}2 \pm 0{,}6$	$74{,}1 \pm 0{,}9$
0 (Réel seul)	80 %	168	$83{,}1 \pm 0{,}5$	$78{,}6 \pm 0{,}7$
0 (Réel seul)	100 %	210	$85{,}2 \pm 0{,}4$	$80{,}4 \pm 0{,}6$
5 000	10 %	21	$69{,}9 \pm 1{,}2$	$63{,}9 \pm 1{,}8$
5 000	15 %	31	$75{,}4 \pm 1{,}0$	$69{,}4 \pm 1{,}5$
5 000	30 %	63	$81{,}5 \pm 0{,}7$	$75{,}2 \pm 1{,}1$

Suite à la page suivante

Tableau A.1 – suite

Vol. synth.	**Frac. réelle**	**Img.**	**mAP@0,5 (%)**	**OKS (%)**
5 000	50 %	105	$85{,}1 \pm 0{,}5$	$79{,}7 \pm 0{,}8$
5 000	70 %	147	$86{,}9 \pm 0{,}4$	$82{,}4 \pm 0{,}6$
5 000	80 %	168	$87{,}7 \pm 0{,}3$	$83{,}7 \pm 0{,}5$
5 000	100 %	210	$88{,}1 \pm 0{,}3$	$84{,}3 \pm 0{,}4$
10 000	10 %	21	$75{,}4 \pm 1{,}0$	$69{,}1 \pm 1{,}5$
10 000	15 %	31	$80{,}4 \pm 0{,}8$	$75{,}0 \pm 1{,}2$
10 000	30 %	63	$84{,}9 \pm 0{,}5$	$80{,}2 \pm 0{,}9$
10 000	50 %	105	$87{,}9 \pm 0{,}4$	$84{,}2 \pm 0{,}6$
10 000	70 %	147	$89{,}0 \pm 0{,}3$	$85{,}8 \pm 0{,}5$
10 000	80 %	168	$89{,}6 \pm 0{,}3$	$86{,}4 \pm 0{,}4$
10 000	100 %	210	$89{,}8 \pm 0{,}2$	$86{,}7 \pm 0{,}4$
20 000	10 %	21	$80{,}1 \pm 0{,}8$	$75{,}3 \pm 1{,}3$
20 000	15 %	31	$85{,}3 \pm 0{,}6$	$80{,}4 \pm 1{,}0$
20 000	30 %	63	$88{,}1 \pm 0{,}4$	$83{,}9 \pm 0{,}7$
20 000	50 %	105	$89{,}7 \pm 0{,}3$	$86{,}4 \pm 0{,}5$
20 000	70 %	147	$90{,}3 \pm 0{,}3$	$87{,}6 \pm 0{,}4$
20 000	80 %	168	$90{,}7 \pm 0{,}2$	$88{,}0 \pm 0{,}4$
20 000	100 %	210	$90{,}9 \pm 0{,}2$	$88{,}2 \pm 0{,}3$
25 000	10 %	21	$82{,}1 \pm 0{,}7$	$77{,}1 \pm 1{,}1$
25 000	15 %	31	$87{,}9 \pm 0{,}5$	$82{,}9 \pm 0{,}8$
25 000	30 %	63	$89{,}9 \pm 0{,}4$	$86{,}1 \pm 0{,}6$
25 000	50 %	105	$91{,}0 \pm 0{,}3$	$87{,}9 \pm 0{,}4$
25 000	70 %	147	$91{,}5 \pm 0{,}2$	$88{,}7 \pm 0{,}4$

Suite à la page suivante

Tableau A.1 – suite

Vol. synth.	Frac. réelle	Img.	mAP@0,5 (%)	OKS (%)
25 000	80 %	168	$91{,}7 \pm 0{,}2$	$89{,}0 \pm 0{,}3$
25 000	100 %	210	$91{,}8 \pm 0{,}2$	$89{,}2 \pm 0{,}3$

Tableau A.1 – Performances détaillées du modèle yolo26n-pose sur le jeu de test Costco.

A.1.2 Jeu de test Noovelia

Le tableau A.2 présente les mêmes 35 configurations sur le jeu de test Noovelia. Les performances sont systématiquement inférieures de 0,4 à 1,0 point par rapport à Costco, ce qui confirme la difficulté légèrement plus élevée du site Noovelia (éclairage plus hétérogène, arrière-plans plus chargés). La tendance générale de l'ablation reste identique : le préentraînement synthétique comble l'essentiel de l'écart aux faibles fractions réelles.

Vol. synth.	Frac. réelle	Img.	mAP@0,5 (%)	OKS (%)
0 (Réel seul)	10 %	21	$50{,}0 \pm 1{,}4$	$42{,}1 \pm 1{,}9$
0 (Réel seul)	15 %	31	$51{,}8 \pm 1{,}2$	$46{,}3 \pm 1{,}8$
0 (Réel seul)	30 %	63	$60{,}1 \pm 0{,}9$	$53{,}6 \pm 1{,}4$
0 (Réel seul)	50 %	105	$70{,}3 \pm 0{,}7$	$64{,}7 \pm 1{,}1$
0 (Réel seul)	70 %	147	$78{,}5 \pm 0{,}6$	$73{,}7 \pm 0{,}8$
0 (Réel seul)	80 %	168	$82{,}4 \pm 0{,}5$	$78{,}3 \pm 0{,}6$
0 (Réel seul)	100 %	210	$84{,}5 \pm 0{,}4$	$79{,}8 \pm 0{,}6$
5 000	10 %	21	$69{,}3 \pm 1{,}1$	$63{,}5 \pm 1{,}7$
5 000	15 %	31	$74{,}8 \pm 0{,}9$	$68{,}8 \pm 1{,}4$
5 000	30 %	63	$80{,}9 \pm 0{,}6$	$74{,}7 \pm 1{,}0$

Suite à la page suivante

Tableau A.2 – suite

Vol. synth.	**Frac. réelle**	**Img.**	**mAP@0,5 (%)**	**OKS (%)**
5 000	50 %	105	$84{,}5 \pm 0{,}5$	$79{,}1 \pm 0{,}7$
5 000	70 %	147	$86{,}2 \pm 0{,}4$	$81{,}9 \pm 0{,}6$
5 000	80 %	168	$87{,}0 \pm 0{,}3$	$83{,}1 \pm 0{,}5$
5 000	100 %	210	$87{,}4 \pm 0{,}3$	$83{,}7 \pm 0{,}4$
10 000	10 %	21	$74{,}8 \pm 0{,}9$	$68{,}5 \pm 1{,}4$
10 000	15 %	31	$79{,}7 \pm 0{,}7$	$74{,}5 \pm 1{,}1$
10 000	30 %	63	$84{,}3 \pm 0{,}5$	$79{,}5 \pm 0{,}8$
10 000	50 %	105	$87{,}2 \pm 0{,}4$	$83{,}5 \pm 0{,}6$
10 000	70 %	147	$88{,}3 \pm 0{,}3$	$85{,}2 \pm 0{,}5$
10 000	80 %	168	$89{,}0 \pm 0{,}3$	$85{,}8 \pm 0{,}4$
10 000	100 %	210	$89{,}2 \pm 0{,}2$	$86{,}0 \pm 0{,}4$
20 000	10 %	21	$79{,}4 \pm 0{,}7$	$74{,}8 \pm 1{,}2$
20 000	15 %	31	$84{,}7 \pm 0{,}6$	$79{,}7 \pm 0{,}9$
20 000	30 %	63	$87{,}4 \pm 0{,}4$	$83{,}2 \pm 0{,}6$
20 000	50 %	105	$89{,}1 \pm 0{,}3$	$85{,}8 \pm 0{,}5$
20 000	70 %	147	$89{,}7 \pm 0{,}3$	$86{,}9 \pm 0{,}4$
20 000	80 %	168	$90{,}0 \pm 0{,}2$	$87{,}3 \pm 0{,}4$
20 000	100 %	210	$90{,}2 \pm 0{,}2$	$87{,}5 \pm 0{,}3$
25 000	10 %	21	$81{,}6 \pm 0{,}7$	$76{,}4 \pm 1{,}0$
25 000	15 %	31	$87{,}2 \pm 0{,}5$	$82{,}3 \pm 0{,}7$
25 000	30 %	63	$89{,}3 \pm 0{,}4$	$85{,}4 \pm 0{,}6$
25 000	50 %	105	$90{,}4 \pm 0{,}3$	$87{,}1 \pm 0{,}4$
25 000	70 %	147	$90{,}8 \pm 0{,}2$	$87{,}9 \pm 0{,}4$

Suite à la page suivante

Tableau A.2 – suite

Vol. synth.	Frac. réelle	Img.	mAP@0,5 (%)	OKS (%)
25 000	80 %	168	$91{,}0 \pm 0{,}2$	$88{,}2 \pm 0{,}3$
25 000	100 %	210	$91{,}2 \pm 0{,}2$	$88{,}5 \pm 0{,}3$

Tableau A.2 – Performances détaillées du modèle yolo26n-pose sur le jeu de test Noovelia.

A.2 Hyperparamètres détaillés de l'entraînement

Cette section complète le résumé présenté à la section 4.7.2. Les valeurs ci-dessous sont identiques pour toutes les générations et toutes les échelles de la famille YOLO-Pose, afin d'assurer une comparaison équitable.

Pipeline d'augmentation des données réelles (en ligne). L'augmentation est appliquée de manière stochastique par le chargeur de données d'Ultralytics à chaque époque : chaque image est transformée aléatoirement au moment de son chargement dans un lot. Il ne s'agit pas d'une duplication hors ligne produisant des fichiers supplémentaires ; la diversité est générée à la volée et renouvelée à chaque passage. Chaque image d'entraînement est vue une fois par époque, avec des transformations aléatoires indépendantes entre les époques. La mention « 2 variantes » dans le tableau A.3 correspond à l'application de deux transformations géométriques successives (rotation, échelle, translation) paramétrées aléatoirement, et non à une multiplication du nombre d'images ou d'entrées dans le lot.

Transformations géométriques.

— Retournement horizontal : probabilité de 0,5 (avec échange des indices des points clés gauches et droits)

— Rotation aléatoire : plage de $[-15°, +15°]$

— Changement d'échelle : facteur compris entre 0,85 et 1,15

— Translation aléatoire : $\pm 10\,\%$ de la largeur et de la hauteur de l'image

Transformations photométriques.

— Ajustement HSV (teinte, saturation, valeur) : plages respectives de $\pm 5\,\%$, $\pm 15\,\%$ et $\pm 15\,\%$

— Flou gaussien : probabilité de 0,15 pour reproduire le flou de mouvement du robot mobile

— Bruit de capteur : probabilité de 0,1

Le tableau A.3 recense l'ensemble des paramètres d'optimisation et d'augmentation d'images utilisés pour l'entraînement.

Paramètre	**Valeur configurée**
Optimiseur	AdamW
Momentum	0,937
Taux d'apprentissage initial (lr_0), préentraînement synthétique	10^{-3}
Taux d'apprentissage initial (lr_0), ajustement réel	10^{-4}
Décroissance du taux d'apprentissage	Cosinus vers 10^{-6}
Décroissance de poids (weight decay)	0,05
Nombre d'époques (synthétique)	150
Nombre d'époques (réel)	100
Patience (arrêt anticipé)	20 époques
Taille de lot (batch size)	16
Taille d'image d'entrée	640×640 pixels
Augmentation, transformations par image	rotation, échelle, translation (aléatoires à chaque époque)
Augmentation, retournement horizontal	probabilité 0,5
Augmentation, rotation aléatoire	$[-15°, +15°]$
Augmentation, changement d'échelle	[0,85, 1,15]
Augmentation, translation	$\pm 10\,\%$
Augmentation, HSV (teinte / saturation / valeur)	$\pm 5\,\%$ / $\pm 15\,\%$ / $\pm 15\,\%$
Augmentation, flou gaussien	probabilité 0,15
Augmentation, bruit de capteur	probabilité 0,1

Tableau A.3 – Hyperparamètres d'entraînement et d'augmentation d'images.

A.3 Résultats de robustesse détaillés

Le tableau A.4 présente les métriques de robustesse détaillées sous forme absolue et relative pour le modèle yolo26n-pose sur le jeu de test Costco (complétant l'analyse menée pour Noovelia au chapitre 5).

Perturbation	**Précision de détection (*mAP*@0,5 pour les boîtes)**		**Qualité de pose (OKS)**	
	Réel seul	**Synthétique puis Réel**	**Réel seul**	**Synthétique puis Réel**
Moyenne globale du jeu de test	85,2	91,8	80,4	89,2
Occlusion partielle	72,3	83,2	65,8	78,5
Éclairage faible / ombres	78,6	88,4	72,4	84,2
Angle de vue extrême	75,8	86,7	68,9	80,3
Encombrement / distracteurs	81,2	91,5	76,3	88,1

Tableau A.4 – Performances de robustesse sous perturbations pour Costco (modèle yolo26n-pose, $n = 3$ répétitions).

Le tableau A.5 présente les métriques de robustesse détaillées pour le modèle yolo26n-pose sur le jeu de test Noovelia, complétant le tableau A.4 pour Costco. La stratégie hybride obtient des valeurs absolues supérieures dans toutes les catégories de perturbation, avec un écart particulièrement marqué pour les angles de vue extrêmes (84,2 % contre 71,5 % en *mAP*@0,5).

Perturbation	Précision de détection ($mAP@0{,}5$ pour les boîtes)		Qualité de pose (OKS)	
	Réel seul	Synthétique puis Réel	Réel seul	Synthétique puis Réel
Moyenne globale du jeu de test	84,5	91,2	79,8	88,5
Occlusion partielle	74,8	85,5	69,5	80,2
Éclairage faible / ombres	78,2	86,8	72,8	81,5
Angle de vue extrême	71,5	84,2	65,2	78,8
Encombrement / distracteurs	82,1	90,4	77,6	86,1

Tableau A.5 – Performances de robustesse sous perturbations pour Noovelia (modèle yolo26n-pose, $n = 3$ répétitions).

Note méthodologique. La ligne *Moyenne globale du jeu de test* dans les tableaux A.4 et A.5 correspond à la performance moyenne sur l'ensemble du jeu de test (45 images par site). Comme le montre le tableau 5-2, la totalité des 45 images de Noovelia est répartie entre les quatre catégories de perturbation, sans qu'aucune image ne constitue un véritable témoin non perturbé. Les catégories individuelles (8 à 15 images) doivent donc être comparées entre elles ou rapportées à la moyenne globale comme indicateur de tendance, et non comme une mesure de dégradation causale. Pour les mêmes raisons, les métriques par catégorie reposent sur de petits effectifs, ce qui accroît la variance des estimations. Les résultats montrent que la stratégie hybride surpasse systématiquement Réel seul dans chaque catégorie, mais ne permettent pas de quantifier une réduction de la dégradation attribuable à une propriété spécifique de l'entraînement.

A.4 Comparaison des architectures YOLO-Pose

Le tableau A.6 regroupe les performances de toutes les architectures Nano et Small évaluées dans ce mémoire pour les deux sites et les deux stratégies principales. Les valeurs

sont des moyennes obtenues sur $n = 3$ répétitions pour la configuration à 100 % de données réelles (210 images) avec 25 000 images synthétiques.

Modèle	Échelle	Costco			Noovelia		
		mAP@0,5 (%)	OKS (%)	Lat. (ms)	mAP@0,5 (%)	OKS (%)	Lat. (ms)
Stratégie Synthétique puis Réel (25 000 images synthétiques + 210 images réelles)							
yolov8n-pose	Nano	$88{,}5 \pm 0{,}3$	$84{,}5 \pm 0{,}5$	24,8	$87{,}8 \pm 0{,}4$	$83{,}8 \pm 0{,}6$	25,5
yolov8s-pose	Small	$91{,}2 \pm 0{,}3$	$88{,}1 \pm 0{,}4$	38,5	$90{,}5 \pm 0{,}3$	$87{,}5 \pm 0{,}5$	39,2
yolo11n-pose	Nano	$89{,}6 \pm 0{,}3$	$86{,}8 \pm 0{,}4$	21,3	$88{,}9 \pm 0{,}3$	$86{,}2 \pm 0{,}5$	21,8
yolo11s-pose	Small	$92{,}4 \pm 0{,}2$	$89{,}5 \pm 0{,}3$	32,8	$91{,}8 \pm 0{,}3$	$89{,}0 \pm 0{,}4$	33,5
yolo26n-pose	Nano	$91{,}8 \pm 0{,}2$	$89{,}2 \pm 0{,}3$	17,5	$91{,}2 \pm 0{,}3$	$88{,}5 \pm 0{,}4$	17,9
yolo26s-pose	Small	$93{,}5 \pm 0{,}2$	$91{,}8 \pm 0{,}3$	27,2	$92{,}8 \pm 0{,}2$	$91{,}2 \pm 0{,}3$	27,8
Stratégie Réel seul (210 images)							
yolov8n-pose	Nano	$82{,}3 \pm 0{,}5$	$76{,}8 \pm 0{,}7$	23,5	$81{,}5 \pm 0{,}6$	$75{,}9 \pm 0{,}8$	24,2
yolov8s-pose	Small	$85{,}6 \pm 0{,}4$	$80{,}2 \pm 0{,}6$	36,8	$84{,}8 \pm 0{,}5$	$79{,}5 \pm 0{,}7$	37,5
yolo11n-pose	Nano	$83{,}1 \pm 0{,}4$	$78{,}5 \pm 0{,}6$	20,1	$82{,}4 \pm 0{,}5$	$77{,}8 \pm 0{,}7$	20,6
yolo11s-pose	Small	$86{,}8 \pm 0{,}3$	$82{,}1 \pm 0{,}5$	31,5	$86{,}1 \pm 0{,}4$	$81{,}5 \pm 0{,}6$	32,2
yolo26n-pose	Nano	$85{,}2 \pm 0{,}4$	$80{,}4 \pm 0{,}5$	16,8	$84{,}5 \pm 0{,}5$	$79{,}8 \pm 0{,}6$	17,2
yolo26s-pose	Small	$88{,}1 \pm 0{,}3$	$84{,}6 \pm 0{,}4$	26,1	$87{,}4 \pm 0{,}4$	$84{,}0 \pm 0{,}5$	26,7

Tableau A.6 – Comparaison des performances des architectures YOLO-Pose (Nano et Small) sur les jeux de test Costco et Noovelia.

Le tableau A.7 étend la comparaison à l'ensemble des échelles (Nano à Extra-Large) pour la stratégie Synthétique puis Réel avec 25 000 images synthétiques sur le jeu de test Costco.

Famille	Échelle	Paramètres	Performance		Lat. (ms)
			mAP@0,5 (%)	OKS (%)	
YOLOv8-Pose	Nano	$\sim$ 3,3 M	$88{,}5 \pm 0{,}3$	$84{,}5 \pm 0{,}5$	24,8
	Small	$\sim$ 11,6 M	$91{,}2 \pm 0{,}3$	$88{,}1 \pm 0{,}4$	38,5
	Medium	$\sim$ 26,4 M	$92{,}8 \pm 0{,}2$	$89{,}5 \pm 0{,}3$	—
	Large	$\sim$ 44,4 M	$93{,}2 \pm 0{,}2$	$90{,}0 \pm 0{,}3$	—
	Extra-Large	$\sim$ 69,4 M	$93{,}5 \pm 0{,}2$	$90{,}3 \pm 0{,}3$	—
YOLO11-Pose	Nano	$\sim$ 2,9 M	$89{,}6 \pm 0{,}3$	$86{,}8 \pm 0{,}4$	21,3
	Small	$\sim$ 9,9 M	$92{,}4 \pm 0{,}2$	$89{,}5 \pm 0{,}3$	32,8
	Medium	$\sim$ 20,9 M	$93{,}6 \pm 0{,}2$	$91{,}2 \pm 0{,}3$	—
	Large	$\sim$ 26,1 M	$94{,}1 \pm 0{,}2$	$91{,}8 \pm 0{,}2$	—
	Extra-Large	$\sim$ 58,8 M	$94{,}4 \pm 0{,}1$	$92{,}0 \pm 0{,}2$	—
YOLO26-Pose	Nano	$\sim$ 2,9 M	$91{,}8 \pm 0{,}2$	$89{,}2 \pm 0{,}3$	17,5
	Small	$\sim$ 10,4 M	$93{,}5 \pm 0{,}2$	$91{,}8 \pm 0{,}3$	27,2
	Medium	$\sim$ 21,5 M	$94{,}5 \pm 0{,}2$	$92{,}8 \pm 0{,}2$	—
	Large	$\sim$ 25,9 M	$94{,}8 \pm 0{,}1$	$93{,}2 \pm 0{,}2$	—
	Extra-Large	$\sim$ 57,6 M	$95{,}1 \pm 0{,}1$	$93{,}5 \pm 0{,}2$	—

Tableau A.7 – Comparaison multi-échelles des architectures YOLO-Pose (stratégie hybride, jeu de test Costco).