



الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

جامعة عبد الحميد بن باديس – مستغانم

Abdel Hamid Ibn Badis University – Mostaganem

كلية العلوم والتكنولوجيا

Faculty of Sciences and Technology

قسم الهندسة الكهربائية

Département Génie électrique



– MEMOIRE DE FIN D'ETUDES –

Filière : Génie des électrique

Spécialité : électronique des systèmes embarquées

Thème :

**Solution Embarquée pour un système de comptage automatique sure
vidéo par intelligence artificielle**

Présenté par :

– Bendouba Abdessalem

Soutenu le 28 / 06 / 2026 devant le jury composé de :

Président	Bentoumi Mohammed	MCA
Examineur	Abderrahmane Abdelkader	MCB

Année universitaire :2025/2026

Remerciements

On remercie dieu le tout puissant de nous avoir donnée la santé et la volanté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de Mr MERAH et Mme Aarizou, on les remercie pour la qualité de leur encadrement exceptionnel, pour leur patience, leur rigueur et leur disponibilité durant notre préparation de ce mémoire

Nous exprimons également notre gratitude aux membres du jury pour l'honneur qu'ils nous font en acceptant d'évaluer ce travail, ainsi que pour leurs remarques constructives qui contreront a l'enrichir

Nos remerciements aussi s'adresse également à tous nos professeurs pour leur générosité et la grande passion dont ils ont su faire preuve malgré leurs charges académiques et professionnelles

Dédicace

Je dédie ce projet :

À ma chère mère,

Mon cher père,

Qui n'ont jamais cessé de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs

À ma sœur,

Et mon frère

Pour leurs soutiens moraux,

À mon cher grand père

À qui je souhaite une bonne santé

À mes chers amis,

Pour leur aide et soutiens dans les moments difficiles

À toute ma famille

ملخص

تأتي هذه المذكرة المندرجة ضمن إطار القرار الوزاري المشترك رقم 1275 (شهادة -مؤسسة ناشئة) تقدم نظامًا مبتكرًا ومستقلًا لتعداد الركاب في الوقت الفعلي لصالح مؤسسات استغلال وتشغيل شبكات الترامواي، خصوصًا بمستغانم. ونظرًا لأن طرق التعداد التقليدية المعتمدة على التحقق من التذاكر تتأثر بشدة بالتهرب الضريبي مما يؤدي إلى تخطيط تشغيلي غير دقيق، يقترح هذا المشروع بنية برمجية متكاملة ثلاثية الطبقات (طبقة الحافة، الخادم، ولوحة التحكم).

في طبقة الحافة (Edge Layer)، تقوم بطاقة حاسوبية مدمجة (Orange Pi 5) بتشغيل نموذج مخصص للرؤية الاصطناعية (YoloX-P2-nano) مخصص لتحديد الأهداف الصغيرة والدقيقة، مدعومًا بخوارزمية التتبع (OC-SORT) للتعامل مع الاكتظاظ واختفاء الهوية، وصياغة منطق حسابي فراغي ذكي (Gated Polygon Counter) لتجاوز عقبات الزوايا البصرية الصعبة والحركات العشوائية للمسافرين. كما تم تحسين الأداء الطاقوي والحراري للعتاد عبر آلة حالات ذكية وبنية ذاكرة مباشرة (Zero-Copy) تعتمد على تقنية (DMA-BUF).

تقوم طبقة الخادم بمركزة وتأمين البيانات داخل قاعدة بيانات (PostgreSQL)، وتتم إدارتها بواسطة بوابة برمجية (Nginx/FastAPI) تدعم اتصالات (HTTP/REST) و (WebSockets) لتوفير مراقبة آنية وتوجيه آمن لأوامر الصيانة والتحكم. تُثبت الدراسة الاقتصادية المنجزة الجدوى التجارية العالية لهذا النموذج القائم على بيع العتاد واشتراكات الصيانة المستمرة والترقية (TMA)، واعدة بقدرة ممتازة على التوسع والربحية.

- الكلمات المفتاحية : تعداد الركاب، الذكاء الاصطناعي، الحوسبة الطرفية.

Abstract

This Master's project, presents an innovative, non-intrusive, and real-time automated passenger counting system tailored for companies handling multiple tramway network, especially in Mostaganem. Since traditional ticketing-based counting methods are severely impacted by fare evasion, leading to inaccurate operational planning, we developed a highly optimized three-layer architecture (Edge, Server, Dashboard).

At the Edge layer, an embedded single-board computer (Orange Pi 5) executes a custom YoloX-P2-nano model optimized for micro-target detection, coupled with OC-SORT for robust tracking under heavy occlusion, and our spatial Gated Polygon Counter to handle chaotic crowd movements and awkward oblique camera angles. Hardware longevity and thermal stability are maximized through a state-triggered active-cooling cycle and a zero-copy memory architecture (DMA-BUF).

The server layer centralizes the processed flow data in a secure PostgreSQL database, managed by a dual HTTP/REST and WebSocket gateway (Nginx/FastAPI) to push real-time alerts and route remote whitelisted commands to the dashboard. The economic study validates the commercial viability of this on-premise model, showing rapid profitability driven by a hardware licensing and recurring maintenance (TMA) model.

- Keywords: Passenger Counting, Edge AI, Deep Learning, Zero-Copy.

Résumé

Ce projet de fin d'études Master, présente un système autonome et non intrusif de comptage automatique de passagers en temps réel, conçu spécifiquement pour les réseaux de tramway qui ont exploité par les sociétés spécialisées, spécialement en Mostaganem. Les méthodes traditionnelles de comptage basées sur la billettique étant fortement biaisées par la fraude, nous proposons une architecture modulaire à trois couches (Edge, Serveur, Dashboard).

Au niveau de la couche embarquée (Edge), un ordinateur monocarte (Orange Pi 5) exécute un modèle YoloX-P2-nano personnalisé pour la détection de micro-cibles, associé à l'algorithme OC-SORT pour le suivi sous occlusion et à notre logique spatiale Gated Polygon Counter pour surmonter les mouvements chaotiques et les angles de caméra obliques. La durabilité matérielle est maximisée par une machine à états limitant la surchauffe et par une architecture mémoire "Zero-Copy" (DMA-BUF).

La couche hôte centralise les flux au sein d'une base de données PostgreSQL sécurisée, administrée par une passerelle (Nginx/FastAPI) exploitant les protocoles HTTP/REST et WebSockets pour transmettre instantanément les alertes et les commandes de maintenance au tableau de bord. L'étude de faisabilité financière valide la viabilité commerciale de cette solution locale par un modèle d'affaires solide basé sur l'acquisition de matériel et des contrats récurrents de Tierce Maintenance Applicative (TMA).

- Mots-clés : Comptage de passagers, IA embarquée, Deep Learning, Zero-Copy

Sommaire

Remerciement	I
Dédicace	II
ملخص	III
Résumé	IV
Abstract	V
Sommaire	VI
Liste des tableaux	IX
Liste des figures	X
Liste des algorithmes	XI
Liste des abréviations	XII
Introduction Générale	1
Chapitre 1 : État de l'art et Contexte Technologique	2
Introduction	3
Section 1.1 : Évolution Historique du Transport et de la Surveillance	4
1.1.1 – Évolution technologique de la vidéosurveillance	4
1.1.2 – Historique et évolution de la surveillance dans les transports	4
Section 1.2 : Les Systèmes de Comptage Automatique par Vision Artificielle	6
1.2.1 – Approches traditionnelles (Traitement d'images classique)	6
1.2.2 – Approches modernes basées sur l'Apprentissage Profond (Deep Learning)	6
1.2.3 – Mécanismes de franchissement et logique de comptage	7
Section 1.3 : L'état de l'art du comptage avec la Vision Artificielle	8
1.3.1 – Le paradigme du "Tracking-by-Detection" TBD	8
Section 1.4 : Problématiques et Limites des Solutions Actuelles	9
1.4.1 – Défis liés à l'environnement d'exploitation	9
1.4.2 – Défis des solutions basées sur le Deep Learning (DL) en contexte industriel	10
Conclusion	11
Chapitre 2 : Entraînement du Modèle et Stratégies d'Intégration à l'Embarqué	12

<u>Introduction</u>	13
<u>Section 2.1</u> : L'énoncé du Problème, les Exigences du Projet et le Cahier des Charges	14
2.1.1 – Limites du Modèle de Comptage Traditionnel et Impact de la Fraude	14
2.1.2 – Limites opérationnelles du contrôle humain	14
2.1.3 – Défis logistiques d'une unité d'exploitation des tramway (ex. Mostaganem)	15
2.1.4 – Spécifications de la Solution de Vision Artificielle	15
<u>Section 2.2</u> : Pré-entraînement et Initialisation de l'Architecture	16
2.2.1 – L'architecture YoloX-P2-Nano	16
2.2.2 – Base de données source haute-fidélité (MOT16).....	16
2.2.3 – Transfert d'apprentissage et initialisation de la tête de détection P2.....	17
<u>Section 2.3</u> : Entraînement sur le Domaine Cible et Apprentissage Actif	18
2.3.1 – Constitution et labellisation du jeu de données local (Transport Urbain).....	18
2.3.2 – Base de données source haute-fidélité (MOT16).....	19
<u>Section 2.4</u> : Stratégies de Déploiement Physique et Logiciel	20
2.4.1 – Intégration in situ des nœuds d'inférence (Edge).....	20
2.4.2 – Déploiement de la couche d'orchestration et de restitution	21
<u>Conclusion</u>	22
Chapitre 3 : Conception et Réalisation du Système	23
<u>Introduction</u>	24
<u>Section 3.1</u> : Architecture Matérielle	25
3.1.1 – Unités de Calcul Embarquées (Edge Hardware)	25
3.1.2 – Infrastructure Serveur et Unités de Restitution	26
3.2.1.a – Unité de Calcul Centrale	27
3.2.1.b – Unité de Restitution	27
3.1.3 – Spécifications Optiques et Flux Vidéo	27
3.1.4 – Contraintes de site et intégration	28
<u>Section 3.2</u> : Architecture Logicielle	29
3.2.1 – La Couche Embarquée (The Edge Layer)	29
3.2.1.a – Étude Théorique et Logique Algorithmique	29

Suivit Multi-Objets	31
La Logique de Comptage	31
Optimisation de la Durée de Vie	34
Gestionnaire des Commandes	34
Persistance locale et résilience réseau	36
Architecture de gestion mémoire	36
3.2.1.2 – Environnement Logiciel et Dépendances	37
3.2.1.c – Réalisation et Optimisation.....	38
Environnement de développement	38
Optimisation du Modèle	38
Implémentation du Pipeline Zero-Copy (DMA-BUF)	39
3.2.2 – La Couche Hôte (The Server Layer)	39
3.2.2.a – Étude Théorique et Logique et des Flux de Données	39
Région d'interfaçage Edge-Serveur	40
Région de Stockage et d'Analyse Statistique et Control	40
Région d'interfaçage Serveur-Dashboard	41
3.2.2.b – Spécifications Techniques	44
Environnement Logiciel et Stack Backend.....	44
Système de Gestion de Bases de Données (SGBD)	44
Protocoles de Sécurité et Intégrité.....	44
3.2.2.c – Réalisation et Optimisation.....	45
Environnement de développement et Stack Backend	45
Implémentation de la sécurité et du "Handshake"	45
Gestion de la Persistance et Analyse Statistique	45
Orchestration des Commandes et Temps Réel (Région Dashboard)	46
3.2.3 – La Couche Tableau de Bord (The Dashboard Layer)	46
3.2.3.a – Architecture de l'Information et UX	46
Hiérarchisation et Arborescence de l'Information	46
Principes d'Ergonomie et Expérience Utilisateur.....	47

Gestion des Profils Utilisateurs (Logique A/O)	47
3.2.2.b – Spécifications et Environnement de Développement	48
Environnement de Développement	48
Stack Technologique Frontend	48
Protocoles de Communication et Temps Réel	48
Bibliothèques de Visualisation de Données	49
3.2.2.c – Conception de l'Interface et Développement	49
Prototypage et Maquettage	49
Architecture Logicielle du Frontend	50
Description des Interfaces Finales	50
Intégration et Communication avec le Backend	50
Prototypage et Maquettage	50
Section 3.3 : Résultats et validations	51
3.3.1 – Résultats et Validation (Dashboard Layer)	51
3.3.1.a – Métriques de Performance et Tests de Charge	51
3.3.1.b – Analyse des Performances	51
Conclusion	53
Chapitre 4 : Étude Économique du Produit	54
Introduction	55
Section 4.1 : Le Business Model Canvas (Plan d'Affaires)	56
4.1.1 – Segment de Clientèle	57
4.1.2 – Proposition de Valeur	57
4.1.3 – Canaux de Distribution	57
Section 4.2 : Plan d'Investissement Initial (CAPEX)	58
Section 4.3 : Charges d'Exploitation et Masse Salariale (OPEX)	59
4.3.1 – Masse Salariale du Personnel Permanent	59
4.3.2 – Charges Externes et Fournitures	59
Section 4.4 : Stratégie de Tarification et Modèle de Revenus	60
4.4.1 – Tarification de l'Équipement et de l'Installation (Vente Directe)	60

4.4.1.a – Calcul détaillé des coûts des composants Edge.....	61
4.4.2 – Octroi des Licences Logicielles (Software Licensing)	62
4.4.3 – Contrats de Tierce Maintenance Applicative (TMA)	62
<u>Section 4.5</u> : Projections Financières sur 5 Ans.....	63
4.5.1 – Tableau des Projections Financières et KPIs (HT)	63
<u>Conclusion</u>	64
Conclusion Générale	65
Référence Bibliographique	a

List Des Tableaux

Tableau 3.1.1 : Tableau des spécifications de l'Orange Pi 5.....	26
Tableau 3.1.2 : Tableau des spécifications de Testbench Server.....	27
Tableau 3.2.1 : Liste des Commandes	35
Tableau 3.3.1 Tableau des résultats de benchmark sur le Testbench Server.....	51
Tableau 4.1.1 Matrice du Business Model Canvas (BMC).....	61
Tableau 4.2 : Plan de financement et investissements de démarrage	58
Tableau 4.3.1 : Structure de la masse salariale (Année 01)	59
Tableau 4.3.2 : Détail des charges externes annuelles (Année 01).....	60
Tableau 4.4.1 : Coût unitaire détaillé des composants d'une unité Edge.....	61
Tableau 4.4.2 : Grille tarifaire unitaire du matériel et de l'installation (HT).....	62
Tableau 4.5.1 : Tableau de synthèse financière sur 5 ans.....	63

List Des Figures

Figure 3.I.1 : Diagramme de Communication entre les Couches.....	24
Figure 3.1.1 : La Carte Orange Pi 5	25
Figure 3.1.2 : Une illustration de la vision des camera	27
Figure 3.1.3 : Graphe de climat moyen en Mostaganem	28
Figure 3.2.1 : Diagramme de Machine d'états de système	34
Figure 3.2.2 : Schémas block de la couche Hôte.....	43
Figure 3.2.3 : Design Wireframe pour le tableau du bord de supervision	49
Figure 4.5.1 : Graphe de synthèse financière sur 5 ans.....	64

List Des Algorithmes

Algorithme 3.1.1.a.i : Gated Polygon Algorithm

List Des Abréviations

AAG : Auth and API Gateway	MBIG : MQTT Broker and IoT Gateway
ANADE : Agence Nationale d'Appui et de Développement de l'Entrepreneuriat	MOT : Multi Object Tracking
API : Application Programming Interface	MQTT : Message Queuing Telemetry Transport
ARM : Advanced RISC Machines	NPU : Neural Processing Unit
BMC : Business Model Canvas	OAD : Operator/Admin Database
CAPEX : Capital Expenditure	OC-SORT : Observation Centric - Simple Online and Realtime Tracking
CCTV : Closed-Circuit Television	OpenCV : Open Computer Vision
CDB : Central Database	OPEX : Operational Expenditure
CPU : Central Processing Unit	RCR : Remote Control Router
DAU : Data Analysis Unit	ReID : Re-identification
DL : Deep Learning	RoI : Region of Interest
DMA-BUF : Direct Memory Access Buffer	RTAP : Régie autonome des transports parisiens
EDLD : Edge Devices List Database	SCP : System Control Panel
EMA : Entreprise de métro d'Alger	SGBD : Système de Gestion de Base de Données
ETA : Entreprise de transport algérien par câbles	SLA : Service Level Agreement
ETP : Équivalent Temps Plein	SoC : System on Chip
ETUSA : Entreprise de transport urbain et suburbain d'Alger	SOTRAL : Société de gestion de la gare routière d'Alger
FPS : Frames Per Second (Trames par seconde)	SQL : Structured Query Language
GFLOPs : Giga Floating Point Operations Per Second	SSD : Single Shot MultiBox Detector
GPU : Graphical Processing Unit	TBD : Tracking By Detection
HT : Hors Taxe	TMA : Tierce Maintenance Applicative
JSON : JavaScript Object Notation	TMC : Tracker, Model, Counter
KPI : Key Performance Indicator	TOPs : Tera Operations Per Second

TRANSTEV : Groupe Terrestre de Voyageurs

TV : Television

USSR : Union of Soviet Socialist Republics (Union des républiques socialistes soviétiques)

UX : User Experience

VA : Vision artificielle

VPS : Virtual Private Server

VPU : Video Processing Unit

VTR : Video Tape Recorder

YOLO : You Only Look Once

Introduction Générale

Le secteur des transports urbains en Algérie traverse actuellement une phase de modernisation sans précédent. Portée par les directives gouvernementales en matière de transition numérique, cette dynamique vise à optimiser l'efficacité des services publics et à moderniser les infrastructures nationales. Au cœur de cette évolution, la gestion des flux de passagers s'impose comme un levier décisionnel stratégique.

Traditionnellement, la mesure de la fréquentation au sein des réseaux de transport public, à l'instar des lignes de tramway gérées par les sociétés d'exploitation des tramways, repose sur les systèmes de validation de titres de transport. Néanmoins, ce modèle fait face à une contrainte de terrain majeure : l'évasion tarifaire. Les données issues de la billettique s'avèrent de ce fait incomplètes et déconnectées des flux réels de voyageurs, faussant l'évaluation de la fréquentation et pénalisant la planification opérationnelle ainsi que les indicateurs de rentabilité économique.

Pour surmonter ces défis sans avoir recours à des capteurs intrusifs ou à une augmentation massive et économiquement non viable du personnel au sol, la vision artificielle s'impose comme la solution de référence. C'est dans ce contexte d'innovation et de transformation numérique que s'inscrit le présent projet de fin d'études. Pour l'obtention du diplôme Master, notre objectif est de concevoir, d'optimiser et de valider la faisabilité d'un système autonome et non intrusif de comptage automatique de passagers pour Les Sociétés d'exploitation des Tramways.

Pour concrétiser cette vision, notre travail est structuré en quatre chapitres principaux :

- Le Chapitre I dresse un état de l'art technologique de la vidéosurveillance dans les transports et analyse les différentes approches algorithmiques de traitement d'images et d'apprentissage profond pour le comptage de personnes.
- Le Chapitre II présente la théorie, qui tout notre technologies et implémentations on les base
- Le Chapitre III détaille l'architecture multiniveaux de notre solution (Edge, Serveur, Dashboard), ainsi que sa mise en œuvre matérielle et logicielle optimisée.
- Le Chapitre IV expose l'étude de faisabilité économique, le modèle d'affaires (Business Model Canvas) et les projections financières sur cinq ans de la startup issue de ce projet.

Chapitre 01 : État de l'art et Contexte Technologique

- **Introduction :**

Les Systèmes de surveillance automatisés basés sur la vision artificielle constituent aujourd'hui un pilier fondamental de la sécurité urbaine et de la gestion des infrastructures critiques. Au fil des dernière décennies¹, ces systèmes ont évolué d'un simple capteur d'image vers des outils d'analyse sophistiqués, capables de traiter des flux de données complexes avec une précision accrue.

C'est pour ces raisons que les sociétés de transport ont commencé à utiliser ces systèmes pour surveiller et compter les passagers à bord, facilitant ainsi la logistique et renforçant la sécurité des passagers, des moyens de transport utilisés et des gares.

De manière générale, on peut distinguer deux approches couramment utilisées pour le comptage des personnes : les méthodes traditionnelles utilisant les algorithmes et méthodologies de traitement d'image classique, et les nouvelle méthodes innovantes exploitant la complexité d'apprentissage profond (Deep Learning).

Dans le secteur du transport ferroviaire, l'enjeu réside dans l'optimisation des flux de passagers et la sécurisation des gares. Ce chapitre se propose d'analyser les avancées passées et actuelles de cette technologie et l'état de l'art, tout en soulignant les limites des solutions actuelles couramment utilisées face aux contraintes opérationnelles de l'entreprise.

¹ (Glinsky, 2000)

- **Section 1.1 – Évolution Historique du Transport et de la Surveillance :**

- ♦ 1.1.1 – Évolution technologique de la vidéosurveillance :

Le premier système de vidéosurveillance (CCTV) destiné au monitoring d'individus a été conçu en Union Soviétique en juin 1922 par l'ingénieur Lev Sergueïevitch Theremin (connu sous le nom de Léon Theremin). Ce dispositif, contrôlé manuellement, permettait de capturer le mouvement de civils à une distance oscillant entre 30,48 m et 48,77 m, avec une résolution d'image de 100 lignes. Présenté au dirigeant de l'URSS de l'époque, Joseph Staline, ce système de suivi, bien que rudimentaire et dépendant d'un opérateur humain, suscita une vive impression et fut rapidement classé secret d'État².

Vingt ans plus tard, en 1942, la société allemande Siemens développa un système de vidéosurveillance destiné aux essais du missile balistique V2. Conçu pour surveiller les lancements depuis une distance de sécurité et transmettre les flux visuels aux scientifiques³, ce système est historiquement considéré comme le premier dispositif de vidéosurveillance moderne. Néanmoins, dépourvu de capacité de stockage de l'information visuelle, il exigeait toujours une observation humaine continue et en temps réel.

En 1949, la société américaine Vericon commercialisa le premier système de vidéosurveillance civil. Contrairement aux technologies de l'époque, ce système utilisait une transmission filaire fermée, s'affranchissant ainsi de l'obligation d'obtenir une licence gouvernementale de radiodiffusion. Principalement développé pour l'industrie de la télévision, il trouva rapidement des applications dans le secteur industriel, la médecine et le domaine militaire. Ne disposant toujours pas de support d'enregistrement, il requérait une surveillance humaine ininterrompue. Ce n'est qu'au cours des années 1950, avec l'apparition des premiers magnétoscopes (VTR - Video Tape Recorders)⁴, que le problème du stockage de l'image fut résolu, apportant la dernière pièce technologique nécessaire à la constitution d'un système de vidéosurveillance moderne et autonome.

- ♦ 1.1.2 – Historique et évolution de la surveillance dans les transports :

En 1961, le gouvernement britannique était le premier à installer un système de vidéosurveillance (CCTV) dans une station de transport, à la station de métro Holborn à Londres⁵. Durant les années 1970, la popularisation des magnétoscopes (VCR) a accéléré

² (Glinsky, 2000)

³ (Dornberger, 1979)

⁴ (Phillips, 2024)

⁵ (Transport for London Corporate Archives, 2024)

l'adoption du CCTV dans les transports en facilitant l'archivage des données, permettant ainsi un monitoring continu sans surveillance humaine constante.

Le 18 novembre 1987, un incendie dramatique à la station de métro King's Cross à Londres entraîne le décès de 31 personnes. L'enquête publique qui s'ensuivit, dirigée par Desmond Fennell, mit en évidence des angles morts critiques empêchant les opérateurs de surveiller les escaliers mécaniques en bois et de coordonner les flux dans les halls de billetterie. Les recommandations de sécurité contraignantes issues de cette enquête⁶, ont transformé la vidéosurveillance d'une mesure de sécurité optionnelle en un composant vital de l'architecture de sécurité des usagers⁷.

Par ailleurs, la décennie 1980 a été marquée par une avancée technologique majeure avec l'invention du premier multiplexeur vidéo, permettant d'enregistrer simultanément les flux de plusieurs caméras sur une seule bande magnétique.⁸

Dans les années 1990, les autorités de transport ont entamé la transition de l'analogique vers le numérique. La démocratisation des enregistreurs vidéo numériques (DVR) a facilité le traitement et l'exploitation des données. C'est également durant cette période que la France a inauguré la ligne 14 du métro parisien, première ligne de métro à grand gabarit entièrement automatisée et sans conducteur.

À la suite des attentats de Madrid du 11 mars 2004 et de Londres du 7 juillet 2005, la sécurité des réseaux de transport a été reclassée comme un élément critique de l'infrastructure de défense nationale et de lutte contre le terrorisme.

Dans les années 2020, la surveillance dans les transports publics a évolué d'un outil d'enregistrement passif vers un capteur cognitif hautement actif. L'intégration de l'apprentissage profond (Deep Learning) et des réseaux de neurones convolutionnels (CNN), exécutés sur des accélérateurs matériels spécialisés, permet aujourd'hui une compréhension sémantique automatisée et en temps réel de l'environnement de transport.

⁶ (Fennell, 1988)

⁷ (London Transport, 1997)

⁸ (Wellens & Ergener, 1986)

- **Section 1.2 – Les Systèmes de Comptage Automatique par Vision Artificielle :**

Les systèmes de comptage automatique des passagers basés sur la vision artificielle représentent une évolution majeure dans la gestion opérationnelle des réseaux de transport public. Contrairement aux méthodes intrusives (les capteurs infrarouges ou tapis de pression) ou manuelles, l'analyse d'images exploitant la vision industrielle offre une mesure continue, non intrusive et hautement évolutive des flux de voyageurs, permettant d'optimiser l'exploitation des lignes en temps réel.

- ♦ 1.2.1 – Approches traditionnelles (Traitement d'images classique)

Historiquement, les premières solutions de vision industrielle appliquées au comptage reposaient sur des algorithmes de traitement d'images déterministes :

La soustraction d'arrière-plan (Background Subtraction)⁹ : Cette méthode modélise le fond statique de la scène pour isoler les pixels en mouvement représentant les passagers.

Le flot optique (Optical Flow) : Il analyse la trajectoire et l'intensité des vecteurs de déplacement des pixels entre deux trames successives pour en déduire la direction des flux.

Bien que ces approches traditionnelles présentent l'avantage d'une faible charge computationnelle, elles affichent des limites critiques en conditions réelles. Elles s'avèrent extrêmement sensibles aux variations de luminosité (reflets, ombres portées) et échouent lors de l'occlusion mutuelle des personnes au sein d'une foule dense, amalgamant souvent plusieurs individus en un seul bloc de pixels en mouvement.

- ♦ 1.2.2 – Approches modernes basées sur l'Apprentissage Profond (Deep Learning)

L'intégration des réseaux de neurones convolutionnels (CNN) a permis de surmonter ces verrous techniques en introduisant des modèles capables d'extraire des caractéristiques sémantiques robustes :

La détection de cibles : Les détecteurs de type Single-Shot (tels que YOLO¹⁰ ou SSD¹¹) localisent et classifient simultanément les passagers sous forme de boîtes englobantes (bounding boxes), garantissant une haute tolérance aux variations d'angles et de postures.

⁹ (Stauffer & Grimson, 1999)

¹⁰ (Redmon, Divvala, Girshick, & Farhadi, 2016)

¹¹ (Liu, Anguelov, Erhan, & Szegedy, 2016)

Le suivi multi-objets (Multi-Object Tracking - MOT¹²) : Pour modéliser les flux, l'association temporelle des détections est indispensable. Les algorithmes de suivi, comme SORT¹³, DeepSORT¹⁴, ByteTrack¹⁵ et OC-SORT¹⁶ associent les boîtes d'une image à l'autre en résolvant le problème d'affectation (algorithme Hongrois) et en estimant le mouvement futur (Filtre de Kalman). Les versions avancées comme OC-SORT minimisent les pertes de pistes (ID Switches) en cas de mouvements chaotiques ou d'occlusions temporaires.

L'estimation par cartes de densité (Density Estimation) : Conçue pour les scènes de saturation extrême où le suivi individuel échoue, cette méthode estime globalement le nombre de personnes par régression de la densité de la foule sur l'image entière.

- ♦ 1.2.3 – Mécanismes de franchissement et logique de comptage

Une fois les trajectoires individuelles modélisées par le tracker, l'évaluation quantitative des entrées et sorties s'appuie sur deux géométries de référence :

La ligne de franchissement virtuelle : C'est la méthode de référence standard. Elle incrémente un compteur directionnel dès que le centroïde d'un objet coupe une ligne numérique prédéfinie sur l'image.

L'analyse de zone d'intérêt (Region of Interest - ROI) : Cette approche délimite une surface polygonale virtuelle. Elle permet d'analyser non seulement le franchissement, mais également la présence, l'hésitation ou le temps d'attente des usagers à l'intérieur d'un espace défini, offrant ainsi une granularité analytique supérieure.

¹² (Anton, Laura, Stefan, & Konrad, 2016)

¹³ (Bewley, Ge, Ott, Ramos, & Upcroft, 2016)

¹⁴ (Nicolai, Alex, & Dietrich, 2017)

¹⁵ (Yifu, et al., 2022)

¹⁶ (Jinkun, Jiangmiao, Xinshuo, Rawal, & Kris, 2023)

- **Section 1.3 – L'état de l'art du comptage avec la Vision**

Artificielle :

L'état de l'art dans le domaine s'articule aujourd'hui autour de deux paradigmes dominants :

- ♦ 1.3.1 – Le paradigme du "Tracking-by-Detection" TBD

Il s'agit de l'approche la plus répandue et la plus performante pour les applications industrielles en temps réel. Elle sépare la détection d'objets (première étape) de l'association temporelle (deuxième étape) :

Détection d'objets en temps réel : Les têtes de détection ultra-rapides et légères de la famille YOLO (YOLOv8, YOLOv10) et les architectures basées sur les transformateurs comme RF-DETR¹⁷ offrent les meilleures performances en matière de compromis précision/latence.

Association et filtrage de mouvement :

- ♦ *Heuristiques et mouvement* : Des trackers comme ByteTrack¹⁸ et OC-SORT¹⁹ (Observation-Centric SORT) demeurent les références pour associer efficacement les trajectoires et corriger les occlusions complexes ou les mouvements non linéaires.
- ♦ *Modèles hybrides et probabilistes* : Les avancées récentes comme ConfMOT²⁰ (2025) réévaluent l'utilisation des filtres de Kalman en ajustant dynamiquement le bruit de mesure en fonction des scores de confiance (Confidence Scores) du détecteur, réduisant ainsi drastiquement les ruptures d'identité (ID Switches) en milieu dense.
- ♦ *Apprentissage d'association (Learned Association)* : Des architectures comme CAMELTrack²¹ (2025) ou MOTIP²² remplacent les règles d'association manuelles par des encodeurs temporels et des mécanismes d'attention, fusionnant l'apparence (ReID), la pose et le mouvement de manière dynamique²³.

¹⁷ (Isaac, Peter, Matvei, Deva, & Neehar, 2026)

¹⁸ (Yifu, et al., 2022)

¹⁹ (Jinkun, Jiangmiao, Xinshuo, Rawal, & Kris, 2023)

²⁰ (Yi, Li, & Zhang, 2025)

²¹ (Vladimir, Baptiste, Victor, Alexandre, & Christophe, 2025)

²² (Ruopeng, Ji, & Limin, 2025)

²³ (Vladimir, Baptiste, Victor, Alexandre, & Christophe, 2025)

- **Section 1.4 – Problématiques et Limites des Solutions Actuelles :**

Les avancées technologiques en matière de vision artificielle ont permis le développement de systèmes de comptage de passagers de plus en plus performants. Cependant, leur déploiement dans des environnements opérationnels réels, tels que les stations de tramway, révèle plusieurs limites intrinsèques aux approches actuelles, particulièrement lorsqu'elles sont confrontées aux contraintes spécifiques du terrain.

- ♦ 1.4.1 – Défis liés à l'environnement d'exploitation

Les systèmes de comptage basés sur la vision sont intrinsèquement dépendants des conditions de capture d'image. Les solutions déployées dans les réseaux de tramway sont sujettes à plusieurs problématiques majeures :

Variations d'éclairage et conditions météorologiques : Les changements de luminosité dus à l'alternance jour/nuit, aux tunnels, aux ombres portées ou aux conditions météorologiques (pluie, brouillard) dégradent la qualité des images, affectant directement la fiabilité des algorithmes de détection et de suivi.

Distorsion de perspective et angles de vue obliques : L'installation des caméras à des hauteurs spécifiques et avec des angles non parallèles à la scène (par exemple, une inclinaison vers le bas de 15-25°) induit une distorsion perspective. Les objets distants apparaissent réduits et peuvent être difficiles à distinguer, rendant les algorithmes de détection standards moins performants pour le comptage à longue distance.

Problématiques d'occlusion et de densité de foule : Dans les heures de pointe, la forte densité de passagers sur les quais et à l'intérieur des rames provoque des phénomènes d'occlusion. Les individus se masquent mutuellement, rendant difficile le suivi individuel continu et le comptage précis des entrées et sorties

Les algorithmes classiques de traitement d'images, tels que la soustraction d'arrière-plan ou le flot optique, montrent rapidement leurs limites face à ces conditions complexes. Ils peinent à maintenir la continuité des trajectoires lorsque les objets sont masqués, entraînant des erreurs de comptage significatives. De plus, les changements d'éclairage, le mouvement des ombres ou les perturbations de la scène (ex : objets non-humains en mouvement) peuvent générer des faux positifs ou négatifs.

- ♦ 1.4.2 – Défis des solutions basées sur le Deep Learning (DL) en contexte industriel

Bien que les modèles de Deep Learning aient révolutionné le domaine, leur application pose d'autres défis, notamment pour un déploiement sur des dispositifs embarqués aux ressources limitées :

Coût computationnel : Les modèles de détection et de suivi les plus performants (comme les architectures YOLO ou DeepSORT) sont très gourmands en ressources CPU/GPU/NPU. L'exécution en temps réel sur des plates-formes embarquées exige des compromis significatifs en termes de précision, de résolution ou de cadence d'images.

Décalage de domaine (Domain Shift) : Un modèle entraîné sur des datasets de haute qualité ne garantit pas une performance optimale lorsqu'il est déployé sur des flux vidéo de qualité inférieure (faible débit binaire, artefacts de compression, angles de vue dégradés). Il y a un risque de dégradation des performances, notamment pour la détection de petits objets ou dans des conditions de faible luminosité.

Gestion de la mémoire et de l'énergie : L'exécution continue de modèles d'IA complexes sur des dispositifs embarqués pose des problèmes de surchauffe et de consommation énergétique, nécessitant des stratégies de gestion de cycle de vie et d'optimisation poussées.

Complexité de déploiement et de maintenance : Les environnements industriels (stations de transport) présentent des contraintes d'intégration et de maintenance (accès, connectivité, conditions climatiques) qui doivent être prises en compte dès la conception.

Ces problématiques soulignent la nécessité d'une solution de comptage qui combine la puissance de l'IA avec une optimisation matérielle et logicielle poussée, capable de s'adapter aux conditions réelles d'exploitation des réseaux de tramways.

- **Conclusion :**

Ce premier chapitre a permis de dresser un état de l'art des technologies de vidéosurveillance et de comptage automatique dans le secteur des transports. Nous avons retracé l'évolution historique de ces systèmes, qui sont passés d'un enregistrement analogique passif à une analyse cognitive active en temps réel, largement portée par les progrès de l'apprentissage profond et des réseaux de neurones convolutionnels.

L'étude des différentes méthodologies de traitement d'images classiques et d'intelligence artificielle a mis en évidence le potentiel des architectures de détection et de suivi multi-objets. Cependant, la confrontation de ces technologies théoriques aux réalités physiques du terrain (occlusions mutuelles, distorsions de perspective liées aux angles de vue, contraintes énergétiques et computationnelles du matériel embarqué) révèle des verrous technologiques et des limites opérationnelles significatives qui restent à franchir.

Afin de concrétiser cette étude technologique et de concevoir un système adapté à un cadre industriel réel, il est indispensable de confronter ces problématiques aux exigences concrètes d'une infrastructure existante. Dans ce contexte, le chapitre suivant sera dédié à la Théorie qui est la base de notre implémentation de la solution.

Chapitre 02 : Entraînement du Modèle et Stratégies d'Intégration à l'Embarqué

- **Introduction :**

Ce chapitre est dédié à l'étude du cycle de vie de notre modèle de vision artificielle, retraçant son évolution d'un détecteur d'objets généraliste vers un système hautement spécialisé dans le comptage de passagers. Nous détaillerons l'ensemble du flux de travail (workflow) de l'apprentissage automatique, depuis la constitution et la labellisation du jeu de données jusqu'à l'entraînement itératif du modèle.

Enfin, nous aborderons le déploiement ciblé en environnement de transport urbain. Ce concept désigne l'adaptation et l'optimisation de l'algorithme face aux contraintes réelles du terrain : angles de prise de vue obliques, phénomènes d'occlusion mutuelle au sein des foules denses, dégradation du signal vidéo (compression réseau) et exécution sur des nœuds de calcul embarqués à ressources limitées.

- **Section 2.1 – l'énoncé du problème, les exigences du projet et le cahier des charges :**

L'analyse de l'exploitation d'un réseau du tramway (par exemple Mostaganem) a mis en évidence des problématiques structurelles et des besoins logistiques précis qui motivent le développement de notre solution.

- ♦ 2.1.1 – Limites du Modèle de Comptage Traditionnel et Impact de la Fraude :

Toutes les sociétés du transport s'efforcent d'optimiser la planification de sa flotte en mesurant l'affluence des voyageurs par station et par tranche horaire. Traditionnellement, cette mesure repose sur les données des validateurs de titres de transport installés à bord des rames. En théorie, chaque validation de ticket ou d'abonnement indique qu'un usager est monté à bord à une station et à un horaire précis.

Cependant, ce modèle d'estimation est lourdement faussé par le phénomène d'évasion tarifaire (la fraude). Les usagers voyageant sans titre de transport ne valident aucun billet. Par conséquent, les données statistiques qui servent à planifier la fréquence des rames sont incomplètes et déconnectées de la réalité des flux physiques de passagers. De plus, ce manque à gagner affecte directement la viabilité financière de l'entreprise, dont le modèle économique repose sur les recettes directes et les abonnements.

- ♦ 2.1.2 – Limites opérationnelles du contrôle humain :

Pour endiguer ce fléau, ces sociétés déploie des équipes d'agents de contrôle à bord des rames. Ces agents effectuent des vérifications aléatoires, verbalisent les contrevenants et tentent de corriger manuellement les estimations de flux. Néanmoins, cette méthode présente des limites opérationnelles évidentes de trois blocs fonctionnels interdépendants :

Les zones d'ombre temporelles et géographiques : Les agents ne peuvent pas couvrir l'intégralité du réseau en continu. De nombreux contrevenants effectuent de courts trajets entre deux stations non contrôlées sans s'acquitter de leur titre de transport.

La fraude par transfert de titre : Ce système manuel est contourné par le transfert de billets. Un usager sortant cède son ticket validé à un usager entrant sur le quai, lui permettant de présenter un titre en règle lors d'un contrôle ultérieur, faussant ainsi la traçabilité du flux.

♦ 2.1.3 – Défis logistiques d'une unité d'exploitation des tramway (ex. Mostaganem) :

L'unité opérationnelle fait face à des contraintes de transport locales particulièrement complexes liées à sa démographie étudiante et touristique :

La gestion des pics universitaires : Nécessité d'ajuster l'offre de transport aux heures d'entrée et de sortie des campus de Kharouba et de la Salamandre pour éviter la saturation des quais et des rames.

La saisonnalité touristique : Nécessité d'adapter la fréquence des rames en période estivale en raison de l'afflux massif de touristes se rendant vers la zone côtière de la Salamandre.

L'optimisation des charges : Lutter contre l'évasion tarifaire sans avoir recours à une augmentation massive du personnel de contrôle au sol, ce qui serait économiquement non viable.

♦ 2.1.4 – Spécifications de la Solution de Vision Artificielle

Pour surmonter ces défis sans impacter l'infrastructure physique des stations, les sociétés généralement s'orientent vers l'utilisation d'une technologie de comptage automatisée, indépendante du titre de transport, s'appuyant sur le réseau de caméras CCTV existant.

La solution proposée s'articule autour d'un pipeline technique structuré et performant:

Acquisition vidéo : Capture du flux des caméras fixes des gares (résolution minimale de 720p).

Détection d'objets : Identification et localisation en temps réel des personnes à l'aide de réseaux de neurones convolutionnel (modèles YOLO/SSD).

Suivi Multi-Objets (Tracking) : Association de trajectoires (tracklets) optimisée en C++/Python pour maintenir l'identité des passagers.

Comptage bidirectionnel : Analyse des déplacements à travers des zones géométriques virtuelles pour classifier les entrées, les sorties et les passages.

Anonymisation Embarquée : Intégration d'un algorithme de floutage automatique des visages dès l'acquisition vidéo afin de garantir le strict respect de la vie privée et de la confidentialité des données sur le quai.

Persistance des données : Stockage local horodaté par station avant synchronisation avec le serveur.

Précision globale : Supérieure ou égale à 95%

Latence de traitement : Moins de 100 ms par image

Taux d'erreur toléré : Inférieur à 5%

- **Section 2.2 – Pré-entraînement et Initialisation de l'Architecture :**

Pour optimiser le processus d'entraînement sous contraintes de temps, l'utilisation du transfert d'apprentissage (transfer learning) s'impose comme une stratégie incontournable. Cette approche consiste à démarrer l'apprentissage à partir de poids pré-entraînés sur des jeux de données généraux avant de spécialiser le réseau sur notre domaine d'application.

- ♦ 2.2.1 – L'architecture YoloX-P2-Nano

L'architecture de base YoloX : Développé en 2021, YoloX¹ a redéfini les standards de la détection d'objets en temps réel en surpassant la plupart des modèles YOLO contemporains sur les benchmarks de référence. Sa principale innovation réside dans l'introduction d'une tête de détection découplée (Decoupled Head), qui sépare les calculs de classification et de régression des boîtes englobantes, ce qui accélère considérablement la vitesse de convergence lors de l'entraînement. À l'instar des architectures conventionnelles de la famille YOLO, sa structure se décompose en trois blocs fondamentaux : la colonne vertébrale (Backbone), le goulot (Neck) et la tête de détection (Head)².

L'intégration de la tête P2 : La tête de détection P2 a été intégrée par nos soins afin de doter le modèle de la capacité de détecter des micro-cibles (motifs d'une taille minimale de 4x4 pixels). Alors que la tête P3 standard applique un sous-échantillonnage de l'image par un facteur 8 (stride 8), la tête P2 limite cette réduction à un facteur 4 (stride 4), préservant ainsi la résolution spatiale des objets lointains du quai.

Cependant, cette modification structurelle brise la compatibilité avec les fichiers de poids pré-entraînés standards (fournis pour l'architecture YoloX-Nano classique). L'ajout de la tête P2 modifie la dimension des tenseurs et la topologie du réseau, créant des couches orphelines (la tête P2 elle-même et ses connexions dans le neck) qui s'initialisent de manière totalement aléatoire, nécessitant une stratégie d'initialisation spécifique.

- ♦ 2.2.2 – Base de données source haute-fidélité (MOT16)

Le dataset MOT16³ (Multi-Object Tracking 2016) est une base de données de référence internationale dédiée à l'entraînement et à la validation des modèles de suivi multi-objets. Ce jeu de données se caractérise par une grande diversité de séquences vidéo capturant des scènes urbaines denses et encombrées, impliquant de nombreux piétons aux trajectoires linéaires et non linéaires.

¹ (Ge, Liu, Wang, Li, & Sun, 2021)

² (Redmon, Divvala, Girshick, & Farhadi, 2016)

³ (Anton, Laura, Stefan, & Konrad, 2016)

L'utilisation de MOT16 dans notre processus d'apprentissage s'explique par sa richesse en annotations de haute qualité et par sa représentativité des comportements de foule en mouvement sous différentes conditions d'éclairage et angles de caméra, ce qui en fait un étalon idéal pour pré-entraîner et stabiliser les algorithmes de suivi.

♦ 2.2.3 – Transfert d'apprentissage et initialisation de la tête de détection P2

Pour surmonter l'incompatibilité structurelle induite par la tête P2, nous avons mis en œuvre une stratégie de transfert d'apprentissage sélectif :

Transfert partiel des poids : Les poids issus du modèle pré-entraîné YoloX-Nano standard (entraîné sur COCO) ont été injectés uniquement dans les couches structurellement identiques (la colonne vertébrale et les têtes P3/P4/P5). Les couches propres à la tête P2 et les blocs de transition associés ont été temporairement laissés avec des poids initialisés de manière aléatoire.

Phase d'initialisation sur MOT16 : Afin de stabiliser ces couches orphelines et d'apprendre au réseau à extraire les caractéristiques des micro-cibles, un entraînement initial de plus de 100 époques a été exécuté sur le dataset MOT16.

Cette phase a permis d'ajuster les poids de la tête P2 de manière cohérente avec le reste du réseau, aboutissant à un modèle de base YoloX-P2-Nano généraliste, robuste et prêt pour la phase finale d'adaptation au domaine spécifique des quais de transport.

- **Section 2.3 – Entraînement sur le Domaine Cible et Apprentissage Actif :**

La création d'un jeu de données local, parfaitement représentatif des contraintes opérationnelles du réseau de transport, s'est déroulée selon un protocole rigoureux d'ingénierie de données en quatre étapes :

- ♦ 2.3.1 – Constitution et labellisation du jeu de données local (Transport Urbain):

La création d'un jeu de données local, parfaitement représentatif des contraintes opérationnelles du réseau de transport, s'est déroulée selon un protocole rigoureux d'ingénierie de données en quatre étapes :

Échantillonnage temporel (Frame Extraction) : À partir des séquences vidéo brutes fournies, un script Python sur mesure a été développé pour extraire les trames à intervalle régulier (une image sur deux). Cette étape élimine la redondance temporelle forte entre trames consécutives tout en conservant les variations dynamiques de mouvement.

Pré-annotation semi-automatique : Afin d'accélérer le processus de labellisation, nous avons utilisé le modèle basé sur les transformateurs RT-DETR⁴ (Real-Time DEtection TRansformer) pour générer un premier jeu de boîtes englobantes (bounding boxes). Cette approche tirant parti de l'état de l'art a permis d'obtenir une pré-segmentation rapide des passagers sur le quai.

Correction manuelle et contrôle qualité (LabelImg) : Chaque trame pré-annotée a été soumise à une vérification humaine minutieuse via l'outil open-source LabelImg. Les opérations de correction ont porté sur :

- ♦ La suppression des faux positifs (détections "hallucinées" sur le décor).
- ♦ L'ajout des passagers manqués, notamment les micro-cibles en fond de scène.
- ♦ La correction des boîtes superposées.
- ♦ L'ajustement précis des contours sur les silhouettes masquées par la foule.

Sélection stratégique et volume du dataset : Afin de maximiser la pertinence du modèle, nous avons privilégié la sélection de séquences capturant des moments de forte affluence et des mouvements complexes. Au terme de ce processus, un jeu de données

⁴ (Yian, et al., 2024)

de référence (Ground Truth) composé de 2 000 images haute-fidélité entièrement annotées a été constitué.

♦ 2.3.2 – Cycle d'apprentissage actif et optimisation de la précision :

L'entraînement initial du modèle YoloX-P2-Nano sur les 2 000 images de notre jeu de données de référence durant 200 époques a permis d'obtenir une précision de départ de 85 % mAP@50% (mean Average Precision à un seuil d'intersection de 0,5). Afin de franchir le seuil d'exigence requis sans avoir à annoter manuellement des milliers de trames supplémentaires de manière aléatoire, nous avons mis en œuvre une stratégie d'apprentissage actif (Active Learning) basée sur l'échantillonnage par incertitude.

Cette boucle d'optimisation semi-automatique a consisté à exécuter le modèle sur un large volume de flux vidéo non annotés pour identifier et extraire les trames présentant les plus grandes faiblesses algorithmiques. Le ciblage s'est appuyé sur deux métriques de sélection rigoureuses :

La zone d'incertitude de confiance (Uncertainty Sampling) : Nous avons traqué et isolé toutes les détections affichant des scores de confiance limites, compris strictement entre 0,3 et 0,5. Cette plage correspond précisément aux cas où le modèle hésite à valider la présence d'un passager.

Les ruptures de suivi temporel (Tracking Mismatches / Drops) : Nous avons ciblé les anomalies temporelles où l'algorithme de suivi (OC-SORT) perdait brutalement une boîte englobante pourtant stable sur les trames précédentes. Ces pertes de pistes signalent des échecs de détection causés par des occlusions complexes ou des variations de luminosité.

En combinant ces critères, nous avons extrait les 100 images les plus problématiques du flux opérationnel. Ces trames clés ont été entièrement relabellisées manuellement à partir de zéro, puis réinjectées dans l'ensemble d'entraînement.

Après un réentraînement de seulement 50 époques supplémentaires sur ce jeu de données enrichi, la précision globale du modèle a enregistré un bond significatif, s'établissant entre 94 % et 95 % mAP@50%. Ce résultat démontre l'efficacité de l'apprentissage actif, qui permet d'atteindre des performances industrielles de premier plan avec un modèle ultraléger en optimisant uniquement la qualité informative des trames d'apprentissage.

- **Section 2.4 – Stratégies de Déploiement Physique et Logiciel :**

Cette section détaille la transition opérationnelle de notre modèle d'apprentissage profond, depuis son environnement d'entraînement en laboratoire vers un système distribué et pleinement fonctionnel sur site. Nous y décrivons l'intégralité du pipeline de packaging, de distribution et de déploiement des nœuds d'inférence embarqués ainsi que de l'infrastructure serveur.

- ♦ 2.4.1 – Intégration in situ des nœuds d'inférence (Edge):

Préparation logicielle : Le déploiement débute par le flashage d'une image épurée du système d'exploitation Armbian sur une carte MicroSD haute performance de classe industrielle. Les dépendances système sont ensuite configurées, suivies par l'installation des bibliothèques d'exécution essentielles, notamment le runtime RKNN-RT (pour l'accélération matérielle sur le NPU), OpenCV (pour le pré-traitement des flux vidéo) et la bibliothèque cliente Mosquitto (pour les liaisons de communication MQTT).

Compilation du binaire : Afin de maximiser les performances de calcul, le code source développé en C++ est compilé en utilisant des techniques de compilation croisée (cross-compilation). Cette approche permet de générer le binaire natif ARM sur une station de travail de développement beaucoup plus puissante que l'ordinateur à carte unique (SBC) cible, réduisant les temps de compilation et garantissant l'intégration d'optimisations bas-niveau spécifiques au processeur RK3588.

Automatisation opérationnelle : Pour garantir la résilience et l'autonomie de l'unité Edge sans intervention humaine, un service système `systemd` sous Linux est configuré. Ce service assure le lancement automatique du processus principal de comptage dès la mise sous tension de la carte, et gère de manière autonome les redémarrages en cas de panne logicielle ou de coupure de courant accidentelle.

Intégration physique : L'unité Edge finale est montée de manière sécurisée à l'intérieur de l'armoire électrique de la station de transit. La liaison réseau avec les caméras de vidéosurveillance est établie par câble Ethernet (ou via une connexion Wi-Fi sécurisée si les contraintes physiques du site l'exigent). L'alimentation électrique est directement prélevée du réseau électrique de l'armoire technique via un convertisseur de tension régulé (AC-DC) afin de fournir un signal électrique stable, adapté aux exigences de la carte d'inférence.

♦ 2.4.2 – Déploiement de la couche d'orchestration (Serveur) et de restitution (Dashboard) :

Le déploiement de l'infrastructure centrale (serveur et interface utilisateur) repose sur des mécanismes de virtualisation légère et de routage de flux, garantissant l'interopérabilité, la portabilité et la facilité de maintenance de la solution :

Conteneurisation et isolation des services (Docker) : Afin de simplifier l'installation du système et de garantir sa portabilité sur n'importe quel système d'exploitation cible, nous avons utilisé Docker et Docker Compose. Cette approche permet de packager l'intégralité des services backend (la base de données PostgreSQL, le broker MQTT Mosquitto et le serveur FastAPI) dans des conteneurs isolés. L'ensemble de l'infrastructure d'orchestration peut ainsi être instancié de manière reproductible via une commande unique, éliminant les conflits potentiels liés aux dépendances système.

Configuration du reverse proxy Nginx : Le serveur web Nginx est configuré en tant que proxy inverse (reverse proxy) pour sécuriser et centraliser les flux de communication. Il intercepte les requêtes HTTP/REST (destinées à l'extraction de statistiques) ainsi que les poignées de main (handshakes) WebSockets (liaisons bidirectionnelles pour le routage de commandes temps réel), puis les redirige de manière sécurisée vers les ports internes des conteneurs appropriés.

Hébergement et restitution du Dashboard : Nginx est également exploité pour héberger et servir directement les fichiers statiques compilés du frontend de notre tableau de bord. Lors de la connexion d'un opérateur depuis son navigateur web, Nginx lui transmet l'interface applicative, initiant instantanément la liaison WebSocket avec le serveur pour la réception dynamique des données de flux.

- **Conclusion :**

En conclusion, ce deuxième chapitre a permis de détailler l'intégralité du cycle de vie de notre modèle de vision artificielle, depuis son initialisation théorique jusqu'à la définition de ses stratégies de déploiement. Grâce à un protocole d'apprentissage actif exploitant un jeu de données local, la précision de l'architecture YoloX-P2-Nano a été optimisée pour atteindre un taux de détection hautement performant de 94 % à 95 % mAP@50%, validant ainsi sa capacité à opérer en milieu urbain dense.

De plus, l'établissement des procédures de déploiement sur site (via Armbian et la compilation croisée C++) et d'orchestration centrale (via Docker et Nginx) garantit la portabilité et la viabilité technique de la solution sur l'ensemble de l'infrastructure de transit.

La phase d'apprentissage et de préparation opérationnelle étant désormais finalisée, il convient de concevoir la structure interne de notre système. Le chapitre suivant sera ainsi consacré à la conception détaillée et à la réalisation du dispositif, structuré de manière rigoureuse autour de deux axes fondamentaux : l'architecture matérielle et l'architecture logicielle.

Chapitre 03 : Conception et Réalisation du Système

• Introduction :

Le chapitre précédent a permis de définir la problématique et le cahier des charges, en matière de comptage automatisé sur chaque station, tout en soulignant les contraintes techniques et environnementales liées aux stations de tramway. Afin de répondre à ces exigences, le système a été conçu selon une architecture modulaire en trois couches interdépendantes. Cette approche permet de répartir les traitements sur des environnements distincts, optimisant ainsi chaque étape de la résolution du problème.

La Figure 3.I.1 présente la manière selon laquelle les différentes couches de ce système s'articulent entre elles. Le système se décompose comme suit :

La Couche Embarquée (The Edge Layer) : Dédiee à l'inférence par vision artificielle sur site, elle assure l'exécution des algorithmes et l'envoi des données en temps réel tout en minimisant la consommation énergétique.

La Couche Hôte (The Server Layer) : Véritable noyau du système, elle garantit la persistance et la sécurité des données, tout en assurant la liaison entre les unités de terrain et le centre de contrôle.

La Couche Tableau de Bord (The Dashboard Layer) : Cette interface est conçue pour la visualisation analytique des flux et le suivi de l'état du système, offrant une navigation intuitive pour les opérateurs.

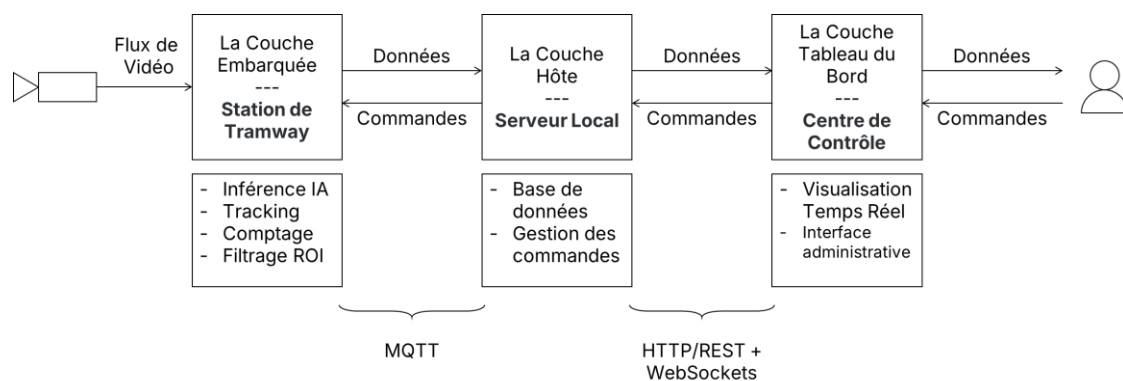


Figure 3.I.1 Diagramme de Communication entre les Couches

Ce chapitre détaille la mise en œuvre technique de chaque couche, en abordant successivement l'étude théorique, les spécifications matérielles et logicielles, ainsi que les optimisations appliquées pour garantir les performances globales du système.

• Section 3.1 – Architecture Matérielle :

♦ 3.1.1 – Unités de Calcul Embarquées (Edge Hardware) :

Le choix du matériel embarqué est dicté par la nécessité de concilier une haute puissance de traitement d'IA avec une consommation énergétique maîtrisée. Pour répondre aux exigences du bloc TMC, nous avons sélectionné l'ordinateur à carte unique (SBC) Orange Pi 5, doté de 4 Go de RAM LPDDR4X. (voir Figure 3.1.1). Les spécificités techniques de cette carte sont détaillées dans le tableau 3.1.1

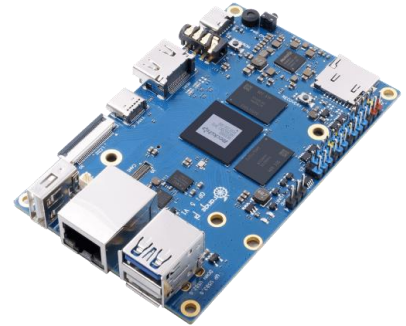


Figure 3.1.1 La Carte Orange Pi 5

Ce choix repose principalement sur les performances du SoC Rockchip RK3588, dont les caractéristiques techniques permettent une orchestration fluide des trois couches de traitement :

Puissance de calcul (CPU) : Son architecture Octa-core (4x Cortex-A76 et 4x Cortex-A55) assure une gestion rapide et sans latence de la logique de tracking (OC-SORT), du filtrage spatial (Gated Polygon) et de la communication avec le serveur.

Accélération de l'IA (NPU) : Le RK3588 intègre un NPU de 6 TOPs (structuré en trois cœurs de 2 TOPs chacun). Cette unité est capable d'exécuter deux instances du modèle YoloX en parallèle. Une analyse théorique de la charge d'inférence montre que :

$$@ 640 \times 640 \text{ PixelPerFrame} : 1.95 \text{ GFLOPs} \times 10 \text{ FPS} = 19.5 \text{ GFLOPs}$$

Projetée sur une résolution de 1080p, la charge atteint environ 555,3 GFLOPs (soit 0,555 TOPs). Ce résultat démontre que l'inférence n'occupe qu'une fraction de la capacité d'un seul cœur NPU ($0.555 \text{ TOPs} \ll 2 \text{ TOPs}$), garantissant une stabilité thermique et une marge de manœuvre pour des évolutions futures.

Décodage Vidéo (VPU) : Le moteur vidéo intégré supporte le décodage 4K à 60 FPS, permettant de traiter simultanément jusqu'à quatre flux 1080p, ce qui est idéal pour les configurations multi-caméras en station.

Enfin, bien que la plateforme soit optimisée pour une basse consommation, elle est alimentée par une source secteur stabilisée afin de prévenir toute fluctuation de tension sur site, garantissant ainsi la pérennité des composants électroniques.

Tableau 3.1.1 Tableau des spécifications de l'Orange Pi 5¹

SoC	Rockchip RK3588S (8nm LP process)
CPU	8-core (4xCortex-A76@2.4GHz; 4xCortex-A55@1.8GHz) 64-bit processor
GPU	Arm Mali-G610 MP4 “Odin” GPU
NPU	up to 6 TOPS, INT4/INT8/INT16 mixed operation
PMU	RK806-1
RAM	4GB/8GB/16GB (LPDDR4/4X)
Memory	QSPI Nor FLASH 16MB + MicroSD (TF) Card Slot + M.2 M-KEY Socket
USB	USB3.0 × 1 + USB2.0 × 2 +Type-C (USB3.1) ×1
Video Output	HDMI 2.1, up to 8K @60Hz + DP 1.4 (DisplayPort) + 2 * MIPI D-PHY TX 4Lane, up to 4K @60Hz
Camera	MIPI CSI 4Lane + 2 * MIPI D-PHY RX 4Lane
Ethernet	10/100/1000Mbps Ethernet
Expansion Port	For extending UART, PWM, I2C, SPI, CAN and GPIO interfaces.
Power Source	Support Type-C power supply, 5V @ 4A
FAN	5V FAN
Debugging	3 Pin debug serial port (UART)
Supported OS	Ubuntu, Debian, Android12
Dimension	62mm*100mm
Weight	46g

♦ 3.1.2 – Infrastructure Serveur et Unités de Restitution :

L'infrastructure matérielle hébergeant la couche hôte (serveur central) a été dimensionnée pour être hautement accessible, évitant l'achat de supercalculateurs industriels coûteux. Sur la base des tests de contrainte et de charge présentés dans la section de validation, nous établissons la configuration matérielle minimale recommandée pour garantir la stabilité opérationnelle du système :

¹ (Orange Pi, 2026)

– 3.1.2.a – Unité de Calcul Centrale (Serveur):

Le tableau suivant démontrera les spécifications de Testbench Server utilisées dans la création de la solution comme une plateforme cible

Tableau 3.1.2 Tableau des spécifications de Testbench Server

Composant	Spécification Technique
Système d'Exploitation	IDX GNU/Linux (Kernel 6.12.90+, x86_64)
Processeur (CPU)	Intel(R) Xeon(R) @ 2.20GHz (2 vCPUs)
Mémoire Vive (RAM)	8.0 Go LPDDR4 (4.4 Go disponibles lors du test)
Stockage	10 Go SSD (Interface haute performance)
Moteur d'exécution	Python 3.11.11 & Docker 27.5.1
Base de Données	PostgreSQL 15-alpine (Conteneurisé)
Broker MQTT	Eclipse Mosquitto 2.0 (Conteneurisé)

– 3.1.2.b – Unité de Restitution (Terminaux Clients):

Le tableau de bord d'administration étant une application Web légère accessible via navigateur, les terminaux clients (ordinateurs de bureau ou tablettes des opérateurs) ne requièrent aucune puissance de calcul dédiée. Tout appareil équipé d'un moteur de rendu web moderne (Chromium/Firefox) et d'une connexion réseau locale est qualifié pour afficher l'interface en temps réel.

♦ 3.1.3 – Spécifications Optiques et Flux Vidéo :

La performance de l'algorithme de vision dépend directement de la qualité du signal d'entrée. L'infrastructure actuelle de la station fournit deux flux vidéo via des caméras IP CCTV, dont les caractéristiques sont les suivantes :

Le Flux Vidéo Actuel : Deux flux 720p à 30 FPS avec un débit de 472 Kbps. Ce débit très faible induit des artefacts de compression, comme du bruit de quantification, rendant les cibles lointaines difficilement identifiables. Toutefois, les tests montrent des résultats probants sur le champ proche.

Projection Technique : Le système est conçu pour évoluer vers des flux 1080p à 30 FPS avec un débit propre, offrant une clarté optimale pour le header P2.

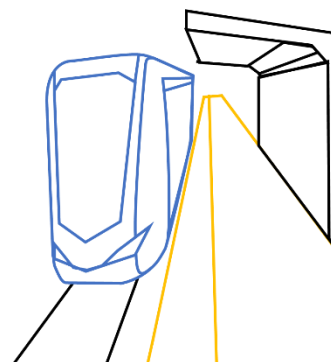


Figure 3.1.2 une illustration de la vision des caméras

Installation physique : Les caméras sont montées à une hauteur supérieure à 2 mètres, et à un angle incliné vers les portillons du tramway mais alignées parallèlement au sol, couvrant ainsi toute la longueur du quai.

Configuration et Codec : Les caméras utilisent le codec H.264 (AVC) avec un espace colorimétrique YCbCr et un sous-échantillonnage de la chrominance de 4:2:0.

♦ 3.1.4 – Contraintes de site et intégration :

L'unité embarquée doit opérer de manière fiable dans l'environnement exigeant des stations de Mostaganem, caractérisé par des fluctuations climatiques extrêmes.

Conditions Climatiques : Les températures moyennes varient de 9°C en hiver (avec des records à -5°C) à 31°C en été (avec des pics à 45°C). L'humidité relative élevée et les vents de sable (vents du Sud-Est)² imposent une protection contre la corrosion et l'infiltration de poussière fine³ (voir Figure 3.1.2.c.2)

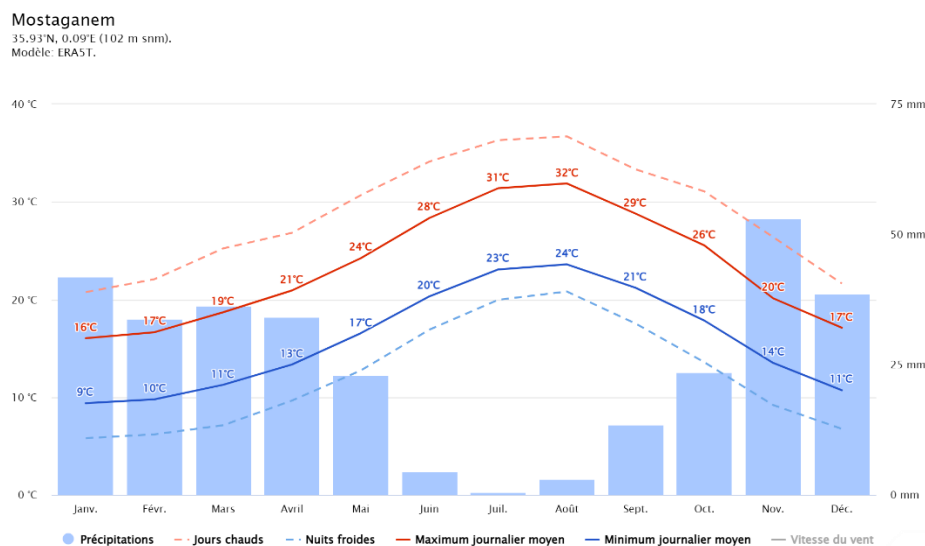


Figure 3.1.3 Le Graphe de température et précipitation moyens à Mostaganem³

Gestion Thermique et Boîtier : L'unité est logée dans un coffret métallique gris. Sous l'effet du rayonnement solaire direct, la température interne de ce boîtier peut atteindre 100°C par effet de serre.

Solutions de Résilience :

- *Refroidissement Actif :* Une solution de ventilation forcée est indispensable pour maintenir le SoC RK3588 en dessous de sa température de pont thermique.

² (KUHN Francis, 2020)

³ (Cedar Lake Ventures, s.d.)

- *Optimisation Logicielle* : La machine à états (détaillée en 3.1.1.d) joue ici un rôle crucial : en désactivant l'inférence lourde hors des périodes de passage des rames, elle limite l'auto-échauffement interne du processeur, préservant ainsi l'intégrité du matériel sur le long terme.
-

• Section 3.2 – Architecture Logicielle :

♦ 3.2.1 – La Couche Embarquée (The Edge Layer) :

– 3.2.1.a – Étude Théorique et Logique Algorithmique :

La couche embarquée repose sur une architecture logicielle modulaire conçue pour assurer la génération et la transmission de données en temps réel directement sur le site. Cette structure s'articule autour de trois blocs fonctionnels interdépendants :

- Le bloc TMC (Tracker, Model, Counter) : Responsable du traitement de la vision artificielle et de la logique de comptage.
- Le bloc de Processus Principal : Assurant l'orchestration système, la persistance locale des données et le formatage des flux sortants.
- Le bloc de Gestion des Commandes : Dédié à la réception et à l'exécution sécurisée des instructions distantes.

Ces modules interagissent de manière synchrone et asynchrone afin de résoudre les défis complexes liés au comptage de personnes en station. Pour garantir une conception robuste et optimisée, nous avons structuré la résolution du problème autour de sept piliers fondamentaux :

– Architecture du Modèle de Vision :

Le modèle de vision sélectionné pour ce projet est une version personnalisée de YoloX-Nano et optimisée par l'intégration d'une tête de détection P2.

Dans l'architecture des modèles YOLO, l'extraction des caractéristiques s'effectue via des têtes de détection opérant à différentes résolutions. Chaque tête traite l'image d'entrée avec un coefficient de réduction spécifique, appelé : *stride* (pas), ce qui lui permet de reconnaître des motifs (patterns) à des échelles variées. Cette approche multi-échelle résout un dilemme fondamental : les détails trop volumineux pourraient être confondus avec le bruit de fond par un modèle trop spécialisé dans le "petit", tandis que les objets de petite taille deviennent indétectables s'ils sont excessivement sous-échantillonnés.

Dans les versions standards de YOLO, on distingue généralement trois têtes de détection :

- ♦ *La tête P3 (stride 8)* : Elle réduit la résolution d'entrée par un facteur 8 (une zone de 8x8 pixels devient 1x1 dans la carte de caractéristiques). Elle est conçue pour détecter les petits objets, mais sa limite de sensibilité se situe autour de motifs de 8x8 pixels.
- ♦ *La tête P4 (stride 16)* : En appliquant un sous-échantillonnage supplémentaire à partir de P3, elle atteint un stride de 16. Elle est optimisée pour les objets de taille moyenne.
- ♦ *La tête P5 (stride 32)* : Avec un stride de 32, elle capture les caractéristiques globales et les objets volumineux. À ce niveau, les petits détails sont totalement lissés pour ne conserver que les formes sémantiques majeures, évitant ainsi que le modèle ne se laisse distraire par des micro-variations de l'arrière-plan.

Cette architecture permet au modèle de comprendre l'objet à travers plusieurs niveaux scalaires. Cependant, une problématique surgit lorsque la scène exige la détection de micro-cibles (détails inférieurs à 8x8 pixels). À ce stade, la tête P3 devient inopérante car l'information spatiale est déjà trop "homogénéisée" (fusionnée avec les pixels voisins) pour être distinguée.

Pour pallier cette limite, nous avons intégré une quatrième tête de détection, la tête P2. Avec un stride de 4, elle ne réduit l'image d'entrée que par un facteur 4. Cette résolution élevée préserve des détails extrêmement fins qui auraient normalement été perdus par P3. En conservant cette granularité spatiale, la tête P2 permet au modèle d'apprendre et d'identifier des motifs complexes sur des objets très petits ou très lointains, garantissant une précision de détection accrue dans les environnements denses ou à forte perspective.

La version standard de YoloX-Nano est conçue pour être une version légère et performante, affichant 0,91 million de paramètres et une charge de 1,08 GFLOPs pour une entrée de 640x640 pixels⁴. Cette efficacité en fait un candidat idéal pour un déploiement sur du matériel embarqué à ressources limitées.

En outre, le choix de YoloX est conforté par sa licence permissive Apache 2.0. Contrairement aux licences avec copyleft, elle autorise la modification et l'intégration du modèle au sein d'une solution propriétaire sans obligation de divulgation du code source. Cet aspect juridique est crucial pour garantir la protection de la propriété intellectuelle et la viabilité commerciale du projet.

⁴ (Ge, Liu, Wang, Li, & Sun, 2021)

Cependant, la configuration physique des caméras sur site induit une forte distorsion de perspective : les passagers situés à l'extrémité opposée du quai apparaissent comme des micro-cibles. Une tête de détection P2 a été utilisée pour préserver une résolution spatiale élevée et identifier des détails allant jusqu'à 4x4 pixels⁵.

Cette modification porte le modèle à 0,94 million de paramètres et 1,94 GFLOPs. Malgré ce surcoût computationnel, les performances restent compatibles avec les capacités du matériel cible. Cette architecture garantit une détection robuste des individus, même en champ lointain, assurant ainsi la fiabilité du comptage sur toute la longueur de la station.

De plus, le modèle ne traite pas l'image dans son intégralité, mais uniquement dans une zone d'intérêt (ROI) qui contient toutes les actions importantes et les personnes qui montent et descendent du tramway.

– Suivi Multi-Objet :

Le maintien de l'identité des passagers au fil des trames est assuré par l'algorithme de suivi OC-SORT (Observation-Centric SORT). Initialement, l'algorithme ByteTrack a été évalué en raison de son efficacité basée sur le Filtre de Kalman pour l'estimation d'état, et l'algorithme Hongrois pour l'association de données. Cependant, les contraintes opérationnelles du site — notamment la forte densité de foule, les mouvements erratiques des passagers et l'angle de vue oblique de la caméra — induisent des dynamiques non linéaires et des occlusions prolongées. Dans ces conditions, ByteTrack tend à perdre les trajectoires (tracklets), entraînant une multiplication des ruptures d'identités (ID Switches).

Le choix s'est donc porté sur OC-SORT, car il est spécifiquement conçu pour gérer les mouvements non linéaires et les interruptions de détection⁶. Grâce à sa logique centrée sur l'observation, OC-SORT permet une ré-identification (ReID) plus robuste des individus après une phase d'occlusion, garantissant ainsi la continuité du suivi et la fiabilité du comptage final, même dans un environnement chaotique.

– La Logique de Comptage

Les solutions de comptage par franchissement de ligne simple (Line-crossing) s'avèrent obsolètes dans l'environnement d'exploitation. La forte densité de foule et l'angle de vue oblique induisent deux problèmes majeurs : l'occlusion prolongée (un

⁵ (Xingkui, Shuchang, Xu, & Qi, 2021)

⁶ (Jinkun, Jiangmiao, Xinshuo, Rawal, & Kris, 2023)

individu disparaît dans la foule avant de franchir la ligne) et la détection tardive (un individu n'est identifié qu'après avoir franchi la zone théorique de la porte).

Pour pallier ces limites, nous avons substitué la ligne traditionnelle par une solution innovante : « le polygone à portes » ou « Gated Polygon ». Cette approche repose sur une machine à états spatiale qui analyse les trajectoires même en cas de perte temporaire du tracker. Le polygone se compose de segments standards et d'un ou plus de segments spécifiques appelés "Gate" (Porte). La logique de comptage (Entrée/Sortie/Passage) n'est validée qu'en fonction de la séquence d'interaction entre le centroïde de l'objet et ces segments spécifiques, permettant ainsi de filtrer les faux positifs et de gérer les changements de direction des usagers.

L'algorithme 3.1.1.A.I explique la fonctionnalité du Gated Polygon.

ALGORITHM 3.1.1.A.I : GATED POLYGON ALGORITHM

Entrées : Détections courantes (D), Gated Polygon (Z_{action}), Historique des trajectoires (H_{tracks})

Sorties : Compteurs globaux (**Total Entrées**, **Total Sorties**)

```

1   Mettre à jour le tracker avec les détections  $D$ 
2   Pour chaque cible active  $\underline{t}$  dans le tracker :
3       Extraire les coordonnées locales  $(x, y, w, h)$  et l'identifiant  $ID$ 
4        $x_{Global} \leftarrow x + X_{ROIstart}$  ;  $y_{Global} \leftarrow y + Y_{ROIstart}$ 
5        $C_{courant} \leftarrow \left( x_{Global} + \frac{w}{2} ; y_{Global} + \frac{h}{2} \right)$  /*Calculer le centroïde global */
6       Si  $ID$  Existe dans l'Historique  $H_{tracks}$  :
7           Récupérer le Centroïde Précédent  $C_{prec}$ 
8            $S \leftarrow$  l'état d'intersection entre  $[C_{prec}, C_{courant}]$  et  $Z_{action}$ 
9           Si  $ID$  est marqué comme "Entré depuis la porte" :
10              Si  $S = EXIT$  et franchissement de la ligne de porte détecté :
11                  Décrémenter Total Sorties (Annulation : l'utilisateur a fait demi-tour)
12              Fin Si
13          Sinon Si  $ID$  est marqué comme "Entré depuis la quai" :
14              Si  $S = EXIT$  et franchissement inverse détecté :
15                  Décrémenter Total Entré (Annulation : l'utilisateur a fait demi-tour)
16              Fin Si
17          Fin Si
18          Sinon (Nouvelle détection de mouvement) :
19              Si  $S = ENTER$  :
20                  Si franchissement de la ligne de porte :
21                      Incrémenter Total Entrées
22                      Marquer  $ID$  comme "Entré depuis la porte"
23                  Sinon :
24                      Incrémenter Total Sorties
25                      Marquer  $ID$  comme "Entré depuis la quai"
26                  Fin Si
27              Sinon :
28                  Si  $S = ENTER$  :
29                      Si franchissement de la ligne de porte : Incrémenter Total Sorties
30                      Sinon : Incrémenter Total Entrées
31                  Fin Si
32              Fin Si
33          Fin Si
34      Fin Si
35      Mettre à jour  $H_{tracks}$  avec  $C_{courant}$ 
36  Fin Pour

```

– Optimisation de la Durée de Vie

Afin de prolonger la durée de vie des unités embarquées et de minimiser la consommation énergétique, le système n'opère en mode de haute performance que lors de la présence effective d'une rame en station. Cette stratégie repose sur une machine à états finis pilotée par des signaux d'activation et de désactivation.

Le fonctionnement se décompose en Trois phases :

- ♦ *L'état Initial (Arrêt)* : tous les systèmes sont déchargés, et la carte est arrêtée.
- ♦ *L'état de veille (Idle)* : Le système maintient une activité minimale. Seuls les processus de diagnostic et de santé du matériel sont actifs, avec une fréquence de transmission réduite vers le serveur central. Cette phase permet de limiter l'échauffement thermique des composants critiques (CPU/VPU).
- ♦ *L'état actif (Active)* : À la réception du signal d'activation (arrivée du tramway), le système initialise le bloc TMC (Tracker, Model, Counter). Les modèles de vision sont chargés en mémoire et l'inférence démarre en temps réel.

Une fois le départ du tramway signalé (signal de désactivation), le système procède à la synchronisation finale des données de comptage avec le serveur hôte avant de libérer les ressources et de retourner à l'état de veille. La Figure 3.1.1.d.1 illustre le comportement et les transitions de cette machine à états.

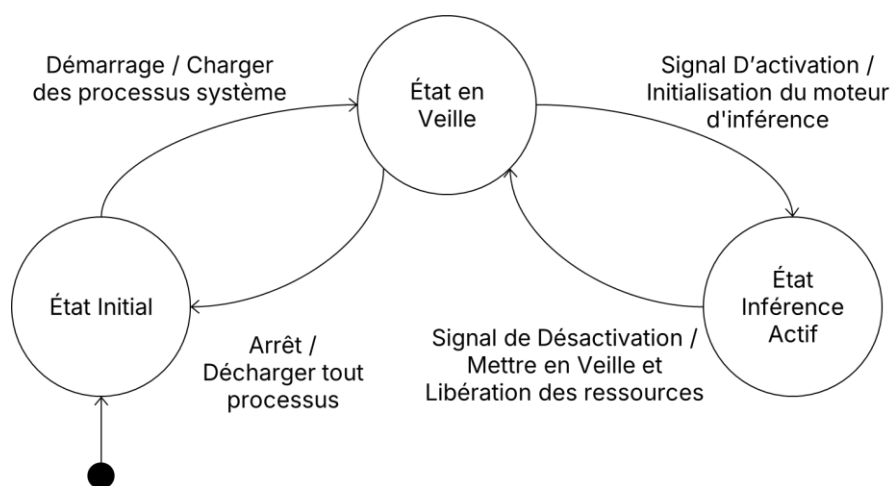


Figure 3.2.1 Diagramme de Machine d'états de système

– Gestionnaire des Commandes :

Afin de garantir une gestion centralisée et sécurisée des unités de terrain, nous avons implémenté un Routeur de Commandes Distantes. Ce module permet aux opérateurs, via le tableau de bord, d'interagir avec la couche embarquée de manière asynchrone. Par mesure de sécurité, l'exécution est restreinte à une "Liste Blanche" de

commandes prédéfinies, empêchant toute manipulation malveillante ou erreur critique du système.

Les commandes sont classées selon quatre catégories fonctionnelles :

Tableau 3.2.1 : la liste des commandes

Maintenance et Énergie	Reboot / DelayedReboot	Redémarrage immédiat ou différé de l'unité.
	TurnOff	Redémarrage immédiat ou différé de l'unité
	UpdateSystem / ResetUpdate	Mise à jour du logiciel ou retour à la version précédente en cas d'instabilité
Gestion Opérationnelle des États	SwitchActiveState / SwitchIdleState	Forçage manuel de la transition entre les états de veille et d'inférence.
	Ping	Test de latence et de connectivité entre l'unité et le serveur central
Gestion Diagnostique	ShowHealth / ShowPerformance	Analyse de l'état de santé (température, CPU) et des performances d'inférence (FPS).
	ShowDeviceInfo / ShowSystemInfo / ShowNetworkInfo	Extraction des métadonnées matérielles, logicielles et réseaux
	ShowLogs	Récupération des journaux d'erreurs pour le débogage à distance
	GetSnapshot	Capture d'une trame isolée pour vérifier l'alignement et la clarté du capteur optique
Configuration de système	ConfigureSystem / TestConfiguration	Modification des paramètres avec mécanisme de rollback automatique (si la

		nouvelle configuration n'est pas validée par l'opérateur, le système restaure l'état stable précédent).
	ResetSystem	Restauration des paramètres d'usine (Default Settings)

– Persistance locale et résilience réseau :

La fiabilité du comptage repose sur l'intégrité des données transmises. Afin de pallier les éventuelles instabilités ou coupures du réseau local de la station, nous avons implémenté un mécanisme de persistance locale agissant comme une mémoire tampon (buffer).

En cas de perte de connectivité avec le serveur central, le processus principal dévie automatiquement le flux de données (comptages et diagnostics) vers une base de données locale légère (SQLite). Chaque enregistrement est horodaté de manière précise afin de garantir une chronologie exacte des événements.

Dès que la connexion est rétablie (validée par le module Ping), le système initie une procédure de synchronisation asynchrone. Grâce à la structure optimisée et à la légèreté des données (format JSON), le rattrapage des données accumulées s'effectue de manière transparente sans interrompre le processus d'inférence en cours. Cette architecture garantit un système "sans perte" (lossless), essentiel pour la précision des statistiques logistique.

– Architecture de gestion mémoire :

Afin de garantir une exploitation totale des capacités du système et de respecter les contraintes strictes du temps réel, nous avons implémenté une architecture de gestion mémoire de type "Zero-Copy".

Dans une architecture conventionnelle, le processeur central (CPU) consacre une part importante de ses cycles à la duplication redondante des données entre les différents espaces mémoire (VPU vers RAM, puis RAM vers NPU). Ce processus, limité par la bande passante du bus mémoire, induit une latence critique et force les processus de traitement à attendre la fin du transfert, risquant ainsi de compromettre les objectifs de performance temporelle.

L'approche "Zero-Copy" pallie ce goulot d'étranglement en permettant à la donnée (la trame vidéo) de résider dans une section unique de la mémoire physique durant tout son cycle de vie. Au lieu de copier la donnée, chaque bloc fonctionnel (VPU, NPU, CPU) reçoit un pointeur d'accès direct. Cette gestion granulaire élimine les copies inutiles, libère les ressources du CPU pour la logique de contrôle et assure une fluidité maximale du flux de traitement, même lors de charges d'inférence intensives.

– **3.2.1.b – Environnement Logiciel et Dépendances :**

La base logicielle de la couche embarquée a été sélectionnée pour sa stabilité, sa légèreté et sa capacité à exploiter pleinement les ressources matérielles du SoC RK3588, tout en garantissant une sécurité juridique pour l'exploitation commerciale future :

Système d'Exploitation : Nous avons opté pour la distribution Armbian, basée sur Debian, en version "Minimal" (sans interface graphique). Ce choix est motivé par l'optimisation de cette distribution pour l'architecture ARM et sa stabilité éprouvée. L'absence d'interface utilisateur réduit l'empreinte mémoire et libère les ressources CPU pour les calculs critiques du bloc TMC.

Langage de développement : Afin de répondre aux exigences de traitement en temps réel, et d'implémenter l'architecture (Zero-copy), l'intégralité de la logique embarquée est développée en C/C++. Ce choix est indispensable pour l'implémentation de notre approche. Ce langage permet une gestion granulaire et direct de la mémoire vive et une exécution native sur le processeur par l'accès aux pointeurs et aux descripteurs de fichiers, éliminant ainsi les latences et les comportements par défaut qui imposent une copie des données lors de la conversion entre bibliothèques inhérentes aux langages interprétés. Ceci facilite également le partage de tampons mémoire entre le VPU, le CPU et le NPU sans duplication de données.

Bibliothèques, API et Outils critiques :

- RKNN-Toolkit2 : Cette API est, Indispensable pour l'exploitation du NPU de 6 TOPs. Elle permet la conversion et l'accélération du modèle YoloX via la quantification des poids en INT8.
- RKNN-RT : Le moteur d'inférence centrale, peut être utilisé pour l'accès direct à la mémoire, permettant au modèle YoloX de lire les données directement depuis le buffer de capture vidéo.

- SQLite : Moteur de base de données relationnelle (*Serverless*), choisi pour sa fiabilité dans la gestion de la persistance locale et sa conformité aux besoins de stockage de secours.
- Mosquitto (Client C++) : Client de communication MQTT sélectionné pour sa légèreté et sa robustesse. Ce choix est également dicté par sa licence permissive, garantissant une intégration sécurisée sans risques juridiques liés au copyleft, ce qui est crucial pour le modèle commercial du projet.
- DMA-BUF / libdrm : Sous-système de linux pour partager les mémoires tampons avec les pilotes des matérielles différentes.
- OpenCV (version optimisée) : Utilisée pour le pré-traitement des flux vidéo avant l'inférence.

– 3.2.1.c – Réalisation et Optimisation

La phase de réalisation logicielle a été guidée par un impératif de performance 'bas-niveau', visant à exploiter chaque unité de calcul du SoC RK3588. Cette section détaille le flux de travail adopté, de la quantification du modèle d'intelligence artificielle à l'implémentation d'un pipeline de données 'Zero-Copy', garantissant ainsi la fluidité du traitement malgré les contraintes de résolution et de débit vidéo

– Environnement de développement :

La réalisation logicielle s'est appuyée sur un écosystème hybride permettant de concilier puissance de calcul pour l'entraînement et légèreté pour le déploiement.

Langages : Utilisation de Python pour la phase de recherche et d'entraînement (Deep Learning), et de C++ pour l'implémentation finale sur l'unité embarquée afin de garantir des performances temps réel.

Infrastructure : L'entraînement a été déporté sur des environnements haute performance (Google Colab / VPS Ubuntu) pour accélérer les calculs GPU.

Chaîne d'outils (Toolchain) : Utilisation du compilateur g++ pour la génération des binaires ARM et de Git pour le suivi de version et la gestion collaborative du code.

– Optimisation du Modèle :

Le passage du modèle théorique à l'exécution matérielle sur le SoC RK3588 a nécessité un pipeline de conversion rigoureux

Exportation : Le modèle YoloX, initialement conçu sous PyTorch, a été exporté au format universel ONNX.

Quantification INT8 : Via l'outil RKNN-Toolkit2, nous avons procédé à une quantification post-entraînement (PTQ) en INT8. Cette étape convertit les poids décimaux en entiers, optimisant l'usage du NPU tout en minimisant la perte de précision grâce à un jeu de données de calibration.

Simulation : Avant le déploiement, le modèle a été validé via le simulateur RKNN pour vérifier la cohérence des sorties d'inférence.

– Implémentation du Pipeline Zero-Copy (DMA-BUF)

Pour atteindre les objectifs de fluidité (FPS) sans saturer le processeur, notre solution Zero-Copy a été implémentée utilisant l'API DMA-BUF du noyau Linux.

Traditionnellement, le CPU doit copier les données entre les différents tampons mémoire (Buffer) du système, ce qui gaspille des cycles de calcul et crée une latence. Notre implémentation permet de stocker la trame vidéo décodée dans une zone mémoire partagée. Le VPU (décodage), le NPU (IA) et le CPU (logique) accèdent à la donnée via des pointeurs (descripteurs de fichiers). En éliminant les copies redondantes, nous avons réduit drastiquement la charge CPU, permettant au système de traiter des flux 1080p en temps réel sans compromettre la stabilité thermique de l'Orange Pi 5.

♦ 3.2.2 – La Couche Hôte (The Server Layer) :

– **3.2.2.a – Étude Théorique et Logique et des Flux de Données :**

La couche hôte constitue et la "colonne vertébrale" (backbone) de l'ensemble du système. Elle assure la persistance des données collectées, le calcul des tendances statistiques, la surveillance de l'état de santé des unités embarquées et l'orchestration des commandes distantes. Pour garantir une sécurité maximale et une intégrité totale des flux, l'architecture serveur est structurée en trois régions fonctionnelles distinctes :

Région d'interfaçage Edge-Serveur : Ce segment est responsable de la gestion des flux bidirectionnels avec les unités de terrain. Il assure la réception des données de comptage et des diagnostics, ainsi que le routage sécurisé des commandes vers les dispositifs autorisés.

Région de Stockage et d'Analyse Statistique et Control : C'est le cœur de traitement du système. Cette zone gère les bases de données, analyse les tendances et déclenche des notifications d'événements significatifs en cas de défaillance critique ou d'anomalie détectée. En plus, elle gère les unités Edge et leurs configurations.

Région d'interfaçage Serveur-Dashboard : Ce segment gère la communication avec les interfaces utilisateurs. Il distribue les données traitées aux opérateurs et

administrateurs, tout en validant les permissions via une logique d'authentification stricte avant l'exécution de toute commande.

Cette segmentation permet d'isoler les flux de données et de protéger le noyau du système (la base de données) contre des requêtes non autorisées, tout en facilitant la maintenance et l'évolution de chaque bloc indépendamment :

– Région d'interfaçage Edge-Serveur :

Cette région constitue le point d'entrée unique pour les flux provenant du terrain. Elle repose sur deux blocs critiques assurant la sécurité et l'authentification des dispositifs.

MBIG (Mqtt Broker and Iot Gateway) : Ce bloc agit à la fois comme courtier de messages (Broker) et comme passerelle de sécurité. Tout flux entre la couche embarquée et le serveur transite par ce module. Lors de la première communication, le MBIG intercepte la requête et suspend le traitement afin de procéder à une validation d'identité. Grâce au formatage spécifique des métadonnées, le MBIG extrait l'identifiant unique (ID) de l'unité et le compare à la liste blanche des dispositifs autorisés. Si une correspondance est établie et que les permissions sont valides, la connexion est autorisée et un bail temporel est initié. Durant cette session, les messages suivants sont automatiquement approuvés pour minimiser la latence. À l'expiration du bail, le processus de validation est reconduit. Toute tentative provenant d'un ID non reconnu ou dépourvu de droits est immédiatement rejetée, protégeant ainsi le serveur contre les accès non autorisés.

EDLD (Edge Devices List Database) : L'EDLD est le référentiel central des identités et des privilèges du parc matériel. Cette base de données contient la liste exhaustive des unités autorisées ainsi que leur niveau d'accès spécifique (lecture de données, exécution de commandes critiques, etc.). Les administrateurs peuvent configurer ce bloc en temps réel (ajout, suppression ou modification de privilèges). L'EDLD sert de source de vérité pour le bloc MBIG, permettant une prise de décision granulaire sur l'autorisation ou le rejet des flux de données en provenance des stations.

– Région de Stockage et d'Analyse Statistique et Control :

Cette région représente le cœur opérationnel de la couche hôte. Elle assure la centralisation des flux, l'analyse statistique des tendances de fréquentation, ainsi que la gestion administrative globale des unités de terrain. Cette intelligence centrale est articulée autour de quatre blocs fonctionnels :

Base de Données Centrale (CDB) : La CDB assure la persistance de l'ensemble des informations système. Elle adopte une stratégie de stockage hybride selon la nature des données. Le premier type est "Données Temporelles" qui est un archivage permanent des entrées et sorties par station, permettant des analyses historiques à long terme. Le deuxième est "Fenêtre Glissante" qui est un stockage temporaire des diagnostics de santé, limité aux sept derniers jours. Cette approche optimise l'espace disque tout en offrant une visibilité suffisante pour la maintenance préventive.

Unité d'Analyse de Données DAU : Ce bloc transforme les données brutes en indicateurs décisionnels. Il traite les flux de comptage pour extraire des caractéristiques statistiques (moyennes, pics d'affluence) et temporelles (comparaisons jour/nuit ou semaine/weekend). Le DAU agit également comme un serveur de données, répondant aux requêtes de la couche tableau de bord pour l'affichage des graphiques analytiques.

Panneau de Contrôle Système (SCP-back) : Il s'agit du cerveau administratif du backend. Ce programme gère la configuration des bases de données et les paramètres distants des unités embarquées. Il intègre une logique de notification d'événements significatifs : en cas d'anomalie majeure (défaillance d'une carte, surchauffe), le SCP-Back alerte automatiquement les services de maintenance externes via les passerelles dédiées.

Routeur de Commandes Distantes RCR : Le RCR orchestre la transmission des instructions entre l'interface utilisateur et le terrain. Il reçoit les commandes du Dashboard, les redirige vers l'unité Edge concernée, intercepte la réponse de confirmation et la renvoie en temps réel à l'opérateur ayant initié l'action. Sa conception asynchrone garantit qu'aucune commande n'est perdue en cas de forte charge réseau.

– Région d'interfaçage Serveur-Dashboard :

Cette région gère l'ensemble des interactions entre la couche hôte et l'interface utilisateur. À l'instar de la région d'interfaçage avec les unités Edge, elle assure une fonction de pare-feu applicatif en filtrant chaque requête entrant dans le centre d'opérations. Elle est structurée autour de deux blocs complémentaires :

AAG (Auth and API Gateway) : La passerelle AAG constitue le point de passage obligatoire pour toute communication provenant du Dashboard. Elle assure une validation systématique des identités en interrogeant la base de données des privilèges. Contrairement au bloc MBIG qui utilise un protocole unique, l'AAG implémente une dualité de protocoles adaptée à la nature des échanges. Le premier

protocole est HTTP/REST, il est utilisé pour les requêtes de type Requête-Réponse, telles que la récupération des statistiques historiques (DAU) ou la modification des paramètres système (SCP-Back). Le deuxième protocole est WebSockets. Il est utilisé pour les flux bidirectionnels en temps réel. Ce protocole est indispensable pour la réception instantanée des notifications d'événements et pour le routage des commandes de contrôle à distance (RCR) sans latence de reconnexion.

OAD (Operators and Admins Database) : L'OAD est le référentiel central des identités et des droits d'accès des utilisateurs. Cette base de données répertorie chaque opérateur et administrateur avec un niveau de permission granulaire (lecture seule, contrôle total, contrôle limité, etc.). Les administrateurs système peuvent configurer ce bloc en temps réel pour révoquer ou accorder des accès. Considérée comme la Source de vérité pour la passerelle AAG, l'OAD permet de garantir qu'aucune action sensible ne soit exécutée par un utilisateur non autorisé ou dépourvu de privilèges adéquats.

La Figure 3.2.1.1 illustre un schémas block sui explique les relations des éléments de la couche hôte entre eux et les autres couches.

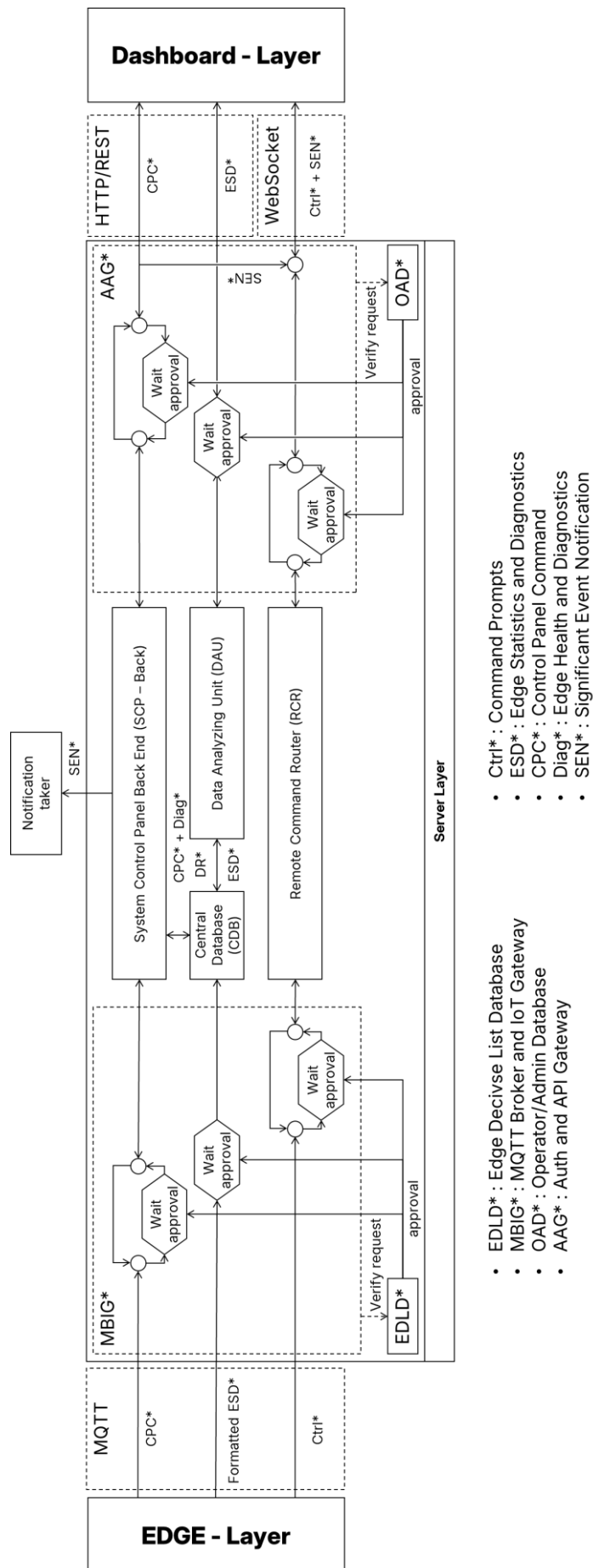


Figure 3.2.2 Schémas block de la couche Hôte

– **3.2.2.b – Spécifications Techniques**

La fiabilité de la couche hôte repose sur une infrastructure logicielle et matérielle capable de soutenir des flux de données critiques et persistants. Cette section détaille l'environnement technologique 'On-Premise' mis en œuvre, en mettant l'accent sur le choix des moteurs de bases de données et des protocoles de communication sécurisés garantissant une haute disponibilité du service.

– Environnement Logiciel et Stack Backend

Le socle logiciel repose sur des technologies de référence dans le domaine des systèmes distribués :

Système d'Exploitation : Linux (Distribution à déterminer selon l'infrastructure existante de l'exploitant).

Serveur Proxy / Gestionnaire de Flux : Nginx Utilisé comme un Reverse Proxy, Nginx orchestre les requêtes HTTP/REST et maintient la stabilité des connexions WebSockets pour les notifications en temps réel.

Broker MQTT : Eclipse Mosquitto, assure la réception asynchrone des données en provenance des stations.

Langage de Programmation : Python

– Système de Gestion de Bases de Données (SGBD) : PostgreSQL

Le choix s'est porté sur PostgreSQL pour la gestion de la CDB, l'EDLD et l'OAD. Ce SGBD relationnel est reconnu pour sa gestion avancée des types de données complexes et sa conformité ACID, garantissant qu'aucune donnée de comptage ne soit corrompue, même en cas de coupure de courant ou de forte charge transactionnelle.

– Protocoles de Sécurité et Intégrité

La sécurité de la couche hôte repose sur une approche de défense en profondeur :

Contrôle d'Accès : Mise en œuvre d'une gestion granulaire des permissions (RBAC) via les bases OAD et EDL

Filtrage des Flux : Validation systématique de la structure des paquets JSON entrants par les passerelles applicatives pour prévenir les injections SQL.

Isolation Réseau : Configuration des pare-feux pour ne laisser ouverts que les ports strictement nécessaires aux protocoles MQTT et WebSockets.

- **3.2.2.c– Réalisation et Optimisation**
- Environnement de développement et Stack Backend

La mise en œuvre de la couche hôte repose sur un écosystème logiciel favorisant l'asynchronisme et la robustesse des transactions :

Framework de développement : FastAPI (Python 3.10+). Nous avons sélectionné FastAPI pour sa performance supérieure et sa gestion native de l'asynchronisme (Async/Await). Ce choix est crucial pour le bloc AAG, permettant de maintenir des milliers de connexions WebSockets et requêtes REST simultanées sans bloquer le processeur.

Serveur de production : Uvicorn et Nginx. L'application est servie par Uvicorn et orchestrée par Nginx agissant en tant que "Reverse Proxy". Cette configuration permet de sécuriser l'accès au backend et de gérer efficacement les poignées de main (Handshakes) des WebSockets pour le bloc RCR.

Système de Gestion de Base de Données (SGBD) : PostgreSQL assure qu'aucune donnée de comptage ne soit perdue en cas d'interruption.

- Implémentation de la sécurité et du "Handshake"

La réalisation du bloc MBIG repose sur une logique d'interception des flux MQTT.

Validation d'identité : Chaque unité Edge est identifiée par un certificat et un ID unique. À la connexion, un script d'authentification interroge la base EDLD pour valider les droits d'accès.

Optimisation du Bail : Pour réduire la surcharge du serveur, nous avons implémenté un système de cache temporaire. Une fois validée, l'unité est "autorisée" pendant une durée définie, évitant ainsi des requêtes SQL redondantes à chaque message envoyé, ce qui maximise le débit de données en période d'affluence.

- Gestion de la Persistance et Analyse Statistique

L'optimisation de la base de données centrale (CDB) est un pilier de la viabilité à long terme du système :

Indexation Temporelle : Les tables de comptage sont indexées par Timestamp. Cela permet au bloc DAU de générer des graphiques historiques (comparaisons hebdomadaires/mensuelles) en quelques millisecondes, même après des années d'accumulation de données.

Maintenance Automatisée (Rolling Window) : Pour le stockage des diagnostics de santé, nous avons implémenté une procédure de nettoyage automatique. Tous les enregistrements de diagnostic de plus de 7 jours sont purgés, garantissant une taille de base de données stable et des performances constantes.

Traitement Asynchrone des Statistiques : Le calcul des tendances n'est pas effectué à la volée lors de la requête de l'utilisateur, mais pré-calculé de manière asynchrone par le DAU, réduisant ainsi la latence perçue sur le Dashboard.

– Orchestration des Commandes et Temps Réel (Région Dashboard)

Le bloc RCR est réalisé via une architecture orientée événements :

Passerelle WebSocket : Contrairement au REST, le WebSocket permet au serveur de "pousser" les notifications vers l'opérateur instantanément.

Fiabilité des Commandes : Chaque commande envoyée (Reboot, Update) est associée à un ID de transaction. Le RCR maintient l'état de la commande ("Envoyée", "En attente", "Confirmée") et gère les délais d'expiration (Timeouts) si l'unité Edge ne répond pas, assurant ainsi qu'un opérateur ne reste jamais dans l'incertitude quant à l'état d'une action.

♦ 3.2.3 – La Couche Tableau de Bord (The Dashboard Layer)

– 3.2.3.a – Architecture de l'Information et UX :

La conception de la couche tableau de bord ne se limite pas à une simple interface graphique ; elle repose sur une étude approfondie de l'ergonomie logicielle et de la hiérarchisation des données. L'objectif est de transformer les flux complexes provenant du serveur en informations décisionnelles exploitables par les opérateurs du réseau.

– Hiérarchisation et Arborescence de l'Information

Afin d'éviter la surcharge cognitive, l'interface est structurée selon une approche du global au particulier, organisée en quatre niveaux de lecture :

Vue Réseau (Dashboard Global) : Un résumé temps réel de l'état de l'ensemble des stations de Mostaganem. Des indicateurs visuels simples permettent d'identifier instantanément les anomalies ou les pics de fréquentation sur le réseau.

Vue Station (Détails Opérationnels) : En sélectionnant une station spécifique, l'opérateur accède aux données précises de comptage (Entrées/Sorties), aux

statistiques de la dernière heure et aux métadonnées de santé de l'unité Edge (Température, État CPU).

Console de Contrôle : Un espace dédié à l'émission des commandes aux périphériques de terrain. Cette zone est protégée par des mécanismes de confirmation pour prévenir toute action accidentelle.

Panneau de Contrôle Système : Cet espace est dédié à l'administration technique et au paramétrage global de la solution. Il offre une interface d'abstraction permettant de configurer les seuils d'alerte du processeur statistique, de gérer les répertoires d'identités (EDLD/OAD) et d'ajuster les variables de fonctionnement des unités Edge. Ce module centralise les fonctions de gestion du backend, permettant aux administrateurs de modifier la logique système sans intervention directe sur les fichiers de configuration brute.

– Principes d'Ergonomie et Expérience Utilisateur

Le design de l'interface suit les principes du Design Centré Utilisateur, avec une attention particulière portée sur les points suivants :

Sémantique des Couleurs : Utilisation d'un code couleur standardisé (Vert : Nominal, Orange : Attention, Rouge : Alerte critique/Surcharge). Ce système permet une réaction immédiate des opérateurs sans lecture approfondie des chiffres.

Réactivité en Temps Réel : Grâce à l'intégration des WebSockets, les indicateurs de comptage sont mis à jour dynamiquement sans rafraîchissement de la page, offrant une perception de réaction instantané entre l'événement sur le quai et son affichage au centre de contrôle.

Retour d'État (Feedback Loop) : Chaque interaction comme l'envoi d'une commande de reboot, génère un retour visuel immédiat (état "En cours", puis "Succès" ou "Échec"), garantissant que l'opérateur est toujours informé de l'issue de ses actions.

– Gestion des Profils Utilisateurs (Logique A/O)

L'architecture de l'information s'adapte dynamiquement selon le profil authentifié via le bloc OAD :

Profil Opérateur (O) : Accès prioritaire aux flux de comptage, aux alertes de fréquentation et aux commandes de maintenance de base.

Profil Administrateur (A) : Accès étendu incluant la gestion du parc matériel (EDLD), la configuration des seuils d'alerte, et l'audit des journaux système.

– **3.2.3.b – Spécifications et Environnement de Développement**

La réalisation de la couche tableau de bord nécessite un écosystème technologique capable de supporter des mises à jour fréquentes sans rafraîchissement de page, garantissant ainsi une fluidité indispensable au monitoring en temps réel.

– Environnement de Développement

Le développement de l'interface utilisateur a été effectué dans un environnement moderne favorisant l'itération rapide et le débogage :

Gestionnaire de paquets NPM, utilisés pour orchestrer les dépendances logicielles et assurer la reproductibilité de l'environnement de build.

Navigateurs cibles en optimisant pour les moteurs Chromium (Google Chrome, Microsoft Edge) et Firefox, assurant ainsi une compatibilité totale avec les standards WebSockets et CSS Grid.

– Stack Technologique Frontend

Pour atteindre notre objectif de créer une expérience d'utilisateur fonctionnelle, nous avons opté pour une architecture basée sur les composants :

Framework / Bibliothèque : React.js. Ces technologies permettent une mise à jour granulaire du DOM (Document Object Model), ce qui signifie que seul le chiffre du compteur change à l'écran, et non toute la page, réduisant ainsi la charge sur le processeur du client.

Styling et Layout: Tailwind CSS. Utilisé pour concevoir une interface légère, responsive et performante, tout en garantissant un temps de chargement minimal du Dashboard.

– Protocoles de Communication et Temps Réel

L'interfaçage avec la couche hôte (Backend) repose sur une dualité technologique

API WebSockets : Utilisée comme moteur principal pour la réception des flux de comptage en provenance du bloc MBIG et pour l'envoi des commandes vers le RCR. Ce protocole évite la surcharge liée aux requêtes HTTP répétitives.

Axios / Fetch API (REST) : Employés pour les appels asynchrones vers le bloc DAU lors de la récupération des données historiques et pour la configuration des paramètres via le SCP-Front

– Bibliothèques de Visualisation de Données

Afin de transformer les statistiques du SSDP en graphiques exploitables, nous avons intégré :

Chart.js : Ces bibliothèques permettent de générer des graphiques dynamiques et interactifs, facilitant l'identification visuelle des tendances de fréquentation (pics d'affluence, moyennes horaires).

– **3.2.3.c – Conception de l'Interface et Développement**

Cette phase transforme les exigences ergonomiques en une application fonctionnelle. Le développement a été guidé par un impératif de modularité, permettant d'isoler les composants d'affichage de la logique de communication avec le backend.

– Prototypage et Maquettage

Avant l'implémentation du code, nous avons réalisé des maquettes fonctionnelles (Wireframes). Cette étape a permis de valider l'agencement des éléments (disposition des flux vidéo, positionnement des alertes, menus de navigation ... etc.) sans la complexité du développement. L'objectif était de s'assurer que les informations critiques (le nombre de passagers et l'état du système) restent au centre du champ visuel de l'opérateur. La Figure 3.3.3.a.1 illustre un design initial (Wire-frame) de l'interface principale de tableau de bord

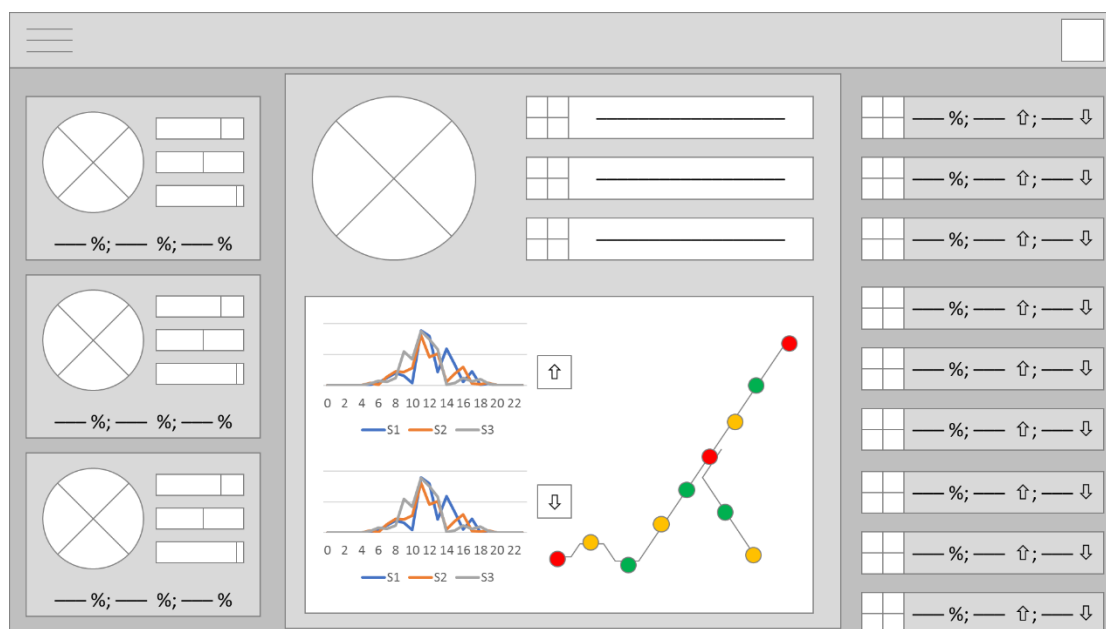


Figure 3.2.3 Design Wireframe pour le tableau de bord de supervision

– Architecture Logicielle du Frontend

L'application est structurée en composants réutilisables. Cette approche facilite la maintenance et l'évolution du système :

Gestion d'État (State Management) : Utilisation de React Context pour synchroniser les données de comptage entre les différents panneaux de l'interface.

Composants Temps Réel : Des "Listeners" WebSockets sont implémentés pour écouter les messages provenant du bloc AAG (Serveur). Chaque fois qu'une donnée arrive, le composant spécifique (ex : le graphique ou le compteur) se met à jour instantanément sans impacter le reste de l'interface.

– Description des Interfaces Finales

Le Tableau de Bord de Supervision : Présente une vue synthétique des flux d'entrées/sorties. Les données sont visualisées via des graphiques dynamiques permettant de détecter les anomalies de charge.

La Console de Commande (RCR-Interface) : Permet l'envoi des commandes de la "Liste Blanche". Un message de confirmation est intégré pour sécuriser les actions critiques (ex : Reboot, Update).

L'Espace de Configuration (SCP-Front) : Réservé aux administrateurs, cet écran permet d'éditer les bases OAD et EDLD, d'ajuster les seuils de détection et de modifier les paramètres réseau des unités Edge.

– Intégration et Communication avec le Backend

La communication est orchestrée via des requêtes structurées au format JSON.

Authentification : L'interface intègre un formulaire de connexion sécurisé communiquant avec la passerelle AAG. Un jeton d'accès (Token) est généré, définissant si l'utilisateur accède aux fonctionnalités "Opérateur" ou "Administrateur".

Flux de Données : Le Dashboard traite les messages entrants pour les transformer en indicateurs visuels (ex : si le tramway est dans une station ou non).

• Section 3.3 – Résultats et validations :

♦ 3.3.1 – Résultats et Validation (Dashboard Layer)

Afin de valider la stabilité, la sécurité et la capacité de montée en charge de la couche hôte, une série de tests de charge (Stress Tests) a été exécutée. L'objectif est de mesurer les limites de débit et de traitement de notre base de données PostgreSQL et de nos passerelles applicatives.

– 3.3.1.a – Métriques de Performance et Tests de Charge

Le benchmark s'est structuré autour de trois scénarios simulant une activité intensive sur le réseau :

- Scénario 1 : Écriture massive et simultanée de données de comptage.
- Scénario 2 : Lecture intensive de l'état d'une station par plusieurs opérateurs.
- Scénario 3 : Requêtes analytiques complexes (calculs temporels de l'algorithme DAU).

Tableau 3.3.1 Tableau des résultats de benchmark sur le Testbench Server

ID	Nature de l'Opération (Endpoint testé)	Volume Total (Requêtes)	Durée Globale (s)	Débit Moyen	Taux de Réussite
01	Écriture de données de comptage (Transactions SQL d'écriture de masse)	200 transactions	3,89 s	51,4 écritures / s	100% (200/200)
02	Lecture de l'état d'une station (Requêtes HTTP standard de lecture)	500 requêtes	4,45 s	112,4 lectures / s	100% (500/500)
03	Analyse SSDP Temporelle (WoW) (Calculs complexes de jointures PostgreSQL)	100 requêtes	29,72 s	3,4 calculs / s	100% (100/100)
04	Comparaison Inter-Stations (Calculs et corrélations de données)	100 requêtes	1,29 s	77,3 calculs / s	100% (100/100)

– 3.3.1.b – Analyse des Performances :

- Régularité des écritures (ID 01) : Le débit de 51,4 écritures/seconde garantit que le serveur peut traiter simultanément les données provenant de plusieurs dizaines de

stations de Mostaganem sans subir de ralentissement ni d'engorgement au niveau des transactions PostgreSQL.

- Optimisation des lectures (ID 02) : Avec 112,4 requêtes/seconde, la passerelle d'authentification (AAG) et le serveur Web gèrent de manière fluide les requêtes d'accès direct des opérateurs de la salle de contrôle.

- Comportement du processeur analytique DAU (ID 03 et 04) :

La baisse de débit à 3,4 calculs/seconde lors de la requête temporelle WoW (Week-over-Week) s'explique par la complexité mathématique des requêtes SQL de comparaison historique exécutées par le DAU.

Cette latence parfaitement documentée justifie l'utilisation d'une architecture asynchrone (FastAPI) pour empêcher que ces calculs lourds ne bloquent les autres processus opérationnels critiques.

- **Conclusion :**

Ce chapitre a permis de détailler l'architecture et la mise en œuvre technique de notre solution de comptage automatisé. En structurant le système en trois couches interdépendantes (l'Edge, le Serveur et le Dashboard) nous avons réussi à concilier les impératifs de performance temps réel avec les contraintes matérielles et environnementales strictes des stations de Mostaganem.

Les optimisations "bas-niveau", notamment l'architecture Zero-Copy et la quantification INT8 du modèle YoloX-P2, ont permis d'atteindre un débit d'inférence fluide sur le SoC RK3588, tout en préservant l'intégrité thermique du matériel via une machine à états intelligente. Parallèlement, la couche hôte garantit la souveraineté et la sécurité des données de l'exploitant grâce à une gestion granulaire des accès et une persistance résiliente sous PostgreSQL.

L'aboutissement de cette phase technique confirme la faisabilité opérationnelle du projet. Il convient désormais d'évaluer la viabilité commerciale et la structure économique de cette innovation, aspects qui feront l'objet de l'étude de marché et de la planification financière détaillées dans le chapitre suivant.

Chapitre 04 : Étude Économique et Plan d'Affaires

- **Introduction :**

Ce projet de Master a une potentielle d'être transformer d'une innovation technologique en un modèle commercial économiquement viable. Ce chapitre expose la stratégie commerciale, la structure des coûts et les projections financières à long terme pour le déploiement de notre solution de vision artificielle au sein des réseaux de transport public algériens.

• Section 4.1 – Le Business Model Canvas (Plan d'Affaires) :

Le Business Model Canvas (BMC) de la startup est structuré pour répondre aux exigences spécifiques du marché public et industriel des transports (B2B/B2G). Le Tableau 4.1.1. détaille la matrice BMC de notre projet

Tableau 4.1.1 Matrice du Business Model Canvas (BMC)

Partenaires Clés	Activités Clés	Propositions de Valeur	Relations Clients	Segments Clients
- EMA (Maître d'ouvrage) : Accord d'intégration réseau - Université de Mostaganem : Recherche & Développement. - Fournisseurs locaux (ex: Dzduino) : Pièces électroniques. - TRANSTEV : Validation de la roadmap d'expansion.	- Conception et optimisation de l'algorithme (YoloX-P2). - Assemblage et installation physique des unités Edge en gare. - Maintenance corrective et évolutive du système (TMA).	- Coût réduit : Utilisation du parc de caméras existantes. - Souveraineté des données : Anonymisation "at the edge" par floutage de visages. - Comptage réel (Anti-fraude) : Détection indépendante des tickets.	- Assistance personnalisée : Accord de niveau de service (SLA). - TMA (Tierce Maintenance Applicative) : Support technique exclusif. - Mises à jour de sécurité gratuites (premières années).	- Client Pilote : SETRAM Mostaganem (19 gares). - Moyen Terme : Réseaux SETRAM nationaux (Oran, Alger, etc.). - Long Terme : Réseaux ETUSA (Bus) et Métros d'Alger.
Ressources Clés		Canaux		
- Humaines : Ingénieur IA (Vous), installateurs. - Matérielles : Orange Pi 5 (Edge), Serveur hôte, outils de test. - Technologiques : Algorithme propriétaire de comptage.		- Vente directe aux directions de la SETRAM / TRANSTEV. - Site internet de promotion d'activité (Landing page). - Démonstrations en conditions réelles (Stations pilotes).		
Structure des Coûts		Flux de Revenus		
- CAPEX : Achat matériel (Orange Pi 5, Station de dev), frais de brevet INAPI. - OPEX : Salaire du personnel, loyer de l'atelier, frais de déplacement sur site.		- Vente Directe : Acquisition unitaire du matériel embarqué et frais d'installation. - Licence Logicielle : Droit d'utilisation du serveur local et du Dashboard. - Contrats TMA : Revenus récurrents pour le support et les mises à jour.		

♦ 4.1.1 – Segment de Clientèle (Customer Segments) :

On va cibler premièrement le marché algérien, notre stratégie de ciblage est divisée en trois phases selon un modèle de pénétration de marché :

Cible Primaire (Client Pilote - B2G) : L'unité opérationnelle de la SETRAM Mostaganem (19 stations). Ce choix permet de valider la solution sur un réseau local avant une expansion nationale.

Expansion Nationale (B2G/B2B) : L'ensemble des réseaux de tramways gérés par la SETRAM à travers l'Algérie (Alger, Oran, Constantine, Sidi Bel Abbès, Sétif, Ouargla).

Diversification future (B2B) : Les autres filiales du groupe TRANSTEV et de l'ETUSA (réseaux d'autobus, téléphériques) ainsi que l'infrastructure de métro (EMA).

Ces modules interagissent de manière synchrone et asynchrone afin de résoudre les défis complexes liés au comptage de personnes en station. Pour garantir une conception robuste et optimisée, nous avons structuré la résolution du problème autour de sept piliers fondamentaux :

♦ 4.1.2 – Proposition de Valeur (Value Propositions)

Notre solution apporte trois valeurs fondamentales à nos clients :

Optimisation des coûts : Utilisation intelligente des infrastructures de caméras CCTV existantes dans les gares, éliminant le besoin d'acheter de nouveaux capteurs coûteux.

Souveraineté et Confidentialité des données : Traitement de l'image s'effectuant localement sur le quai (Anonymisation "at the Edge"), évitant la transmission de données d'identité sensibles sur le réseau.

Précision Décisionnelle : Remplacement des statistiques théoriques (basées sur les tickets facilement contournables) par un taux des mesures réel, permettant une planification optimale des rames lors des pics d'affluence.

♦ 4.1.3 – Canaux de Distribution (Channels)

Vente Directe Institutionnelle : Négociation de contrats directement avec les directions opérationnelles des sociétés d'exploitation des moyennes du transport.

Démonstrations In Situ : Déploiement pilote sur les trois stations les plus fréquentées de Mostaganem pour prouver l'efficacité du système en conditions réelles.

• Section 4.2 – Plan d'Investissement Initial (CAPEX)

Pour lancer l'activité de la startup dans le cadre de l'arrêté 1275, un capital d'investissement initial est requis. Ce budget (CAPEX) couvre l'acquisition des équipements de développement, l'aménagement du premier atelier et la sécurisation de la propriété intellectuelle (INAPI). Les détails sont fournis dans le Tableau 4.2.

Tableau 4.2 : Plan de financement et investissements de démarrage (Année 01)

Catégorie	Libellé des Équipements	Qnt.	Coût Unitaire (DZD)	Coût Total (DZD)
Équipements de R&D	Station de calcul Deep Learning (Workstation RTX 3090/4090)	1	550 000 DA	550 000 DA
	Laptop de bureau (Administratif / Excel)	1	85 000 DA	85 000 DA
	Matériel électronique & Station de soudage thermique	1	45 000 DA	45 000 DA
Aménagement Atelier	Mobilier de bureau (Bureaux, chaises ergonomiques)	1	120 000 DA	120 000 DA
	Imprimante multifonction laser	1	35 000 DA	35 000 DA
	Climatiseur Split (12000 BTU) pour la salle des serveurs	1	75 000 DA	75 000 DA
Actifs Incorporels	Dépôt de brevet d'invention auprès de l'INAPI	1	50 000 DA	50 000 DA
	Hébergement web annuel & Nom de domaine (.dz / .com)	1	15 000 DA	15 000 DA
Fonds de R&D	Composants de rechange (Orange Pi 5, caméras de test)	1	150 000 DA	150 000 DA
Frais de Constitution	Enregistrement de la startup (CNRC & Notaire)	1	25 000 DA	25 000 DA
SOUS-TOTAL (1)	Total des Investissements Physiques et Incorporels			1150000 DA 7670 €

• Section 4.3 – Charges d'Exploitation et Masse Salariale (OPEX) :

Les charges d'exploitation (OPEX) représentent les coûts récurrents nécessaires au fonctionnement quotidien de la startup. Afin de maximiser la viabilité financière de l'entreprise durant sa première année, nous avons opté pour une structure de coûts extrêmement légère (Lean Startup), basée sur un personnel permanent réduit au minimum et l'utilisation de prestataires externes pour les phases d'installation physique.

♦ 4.3.1 – Masse Salariale du Personnel Permanent :

Durant la phase de lancement (Année 01), la gestion de la R&D, de l'administration et de l'intégration logicielle est assurée exclusivement par le fondateur. Les installations physiques sur site sont déléguées à des techniciens électriciens indépendants, rémunérés à la tâche, ce qui évite de lourdes charges salariales fixes. (Voir Tableau 4.3.1).

Tableau 4.3.1 : Structure de la masse salariale (Année 01)

Poste de Travail	Effectif (ETP)	Salaire Mensuel Brut (DZD)	Salaire Annuel Brut (DZD)	Charges Patronales (DZD)	Coût Total Annuel (DZD)	Coût Total Annuel (€)
Ingénieur Système & CEO (Fondateur)	1,00	60 000 DA	720 000 DA	Exonéré (Soutien ANADE)	720 000 DA	4 750 €
Technicien d'installation (Freelance)	-	Rémunération à la tâche (5000 DA / station)	Réparti dans les coûts de vente directes	-	0 DA (Fixe)	-
SOUS-TOTAL (02)	1,00				720 000 DA	4 750 €

♦ 4.3.2 – Charges Externes et Fournitures

Les charges externes englobent l'ensemble des dépenses liées à la location de l'atelier à Mostaganem, à la consommation d'énergie indispensable au fonctionnement de la station de calcul, et aux frais logistiques de déplacement pour la maintenance sur site. Les détails des charges externes annuelles de la première année de lancement sont présentés dans le tableau 4.3.2.

Tableau 4.3.2 : Détail des charges externes annuelles (Année 01)

Catégorie de Charge	Périodicité	Coût Unitaire (DZD)	Coût Annuel Total (DZD)	Coût Annuel Total €
Loyer de l'Atelier (Mostaganem)	Mensuel	25 000 DA	300 000 DA	1980 €
Électricité / Eau / Internet	Trimestriel	30 000 DA	120 000 DA	790 €
Frais de Déplacement / Logistique (Carburant)	Mensuel	15 000 DA	180 000 DA	1190 €
Honoraires Comptables & Juridiques	Annuel	50 000 DA	50 000 DA	330 €
Frais de Télécom & Communication	Mensuel	2 500 DA	30 000 DA	200 €
Assurance Multirisque Professionnelle	Annuel	25 000 DA	25 000 DA	170 €
SOUS-TOTAL (03)	Total OPEX Hors Salaires		705 000 DA	4660 €

Le total des charges opérationnelles (Masse Salariale + Charges Externes) pour assurer la première année d'activité s'élève à :

$$\text{Total OPEX} = \text{SousTotal (02)} + \text{SousTotal (03)}$$

$$\text{Total OPEX} = 720\,000\text{ DA} + 705\,000\text{ DA} = 1\,425\,000\text{ DA}$$

• Section 4.4 – Stratégie de Tarification et Modèle de Revenus :

Le marché de l'intégration technologique dans les transports publics exige une stratégie de tarification flexible et transparente. Notre modèle repose sur un équilibre entre l'acquisition initiale du matériel (CAPEX Client) et des contrats de services récurrents (OPEX Client), permettant de fidéliser le client tout en sécurisant des revenus stables.

♦ 4.4.1 – Tarification de l'Équipement et de l'Installation (Vente Directe)

La tarification unitaire d'un point de comptage (une station équipée) est calculée selon la formule stricte suivante, basée sur le coût des composants physiques et la main-d'œuvre de l'installateur indépendant, majorée d'une marge commerciale de 50 % :

$$\text{Prix Unitaire} = (\text{Coût des composants} + \text{Coût de la Main} - \text{d'œuvre}) \times 1,5$$

- 4.4.1.a – Calcul détaillé des coûts des composants Edge

Le coût des composants d'une unité Edge est un facteur clé dans la détermination du prix de vente. Ces composants, majoritairement importés, sont soumis aux cours du marché international et sont calculés sur la base de leur prix d'acquisition en euros, convertis au taux de change officiel (Tableau 4.4.1).

Tableau 4.4.1 : Coût unitaire détaillé des composants d'une unité Edge

Composant Principal	Spécification Technique	Coût Unitaire (EUR)	Coût Unitaire (DZD)
Orange Pi 5	SBC, 4 Go RAM LPDDR4X, SoC RK3588 (NPU 6 TOPs)	100 €	15 160 DA
Boîtier Industriel	IP66, aluminium/acier, dissipateur passif intégré (<i>Offert pour pilote</i>)	30 €	4 550 DA
Ventilateur Actif	Ventilateur 5V, faible consommation	10 €	1 520 DA
Adaptateur Secteur	5V/4A, prise type E ou F (<i>Probabilité de 50%, coût moyen</i>)	10 €	1 520 DA
Câblage Ethernet	Câble Cat6 blindé (5m), connecteurs RJ45	5 €	760 DA
Supports de Fixation	Équerres métalliques, visserie	5 €	760 DA
MicroSD Haute Vitesse	64 Go (pour OS et stockage local)	15 €	2 270 DA
TOTAL Coûts Composants (Pilote)	<i>(Hors boîtier IP66)</i>	145 €	21 990 DA
TOTAL Coûts Composants (Standard)	<i>(Avec boîtier IP66)</i>	175 €	26 540 DA

La grille tarifaire unitaire hors taxes du matériel et de l'installation est détaillée dans le Tableau 4.4.2.

Tableau 4.4.2 : Grille tarifaire unitaire du matériel et de l'installation (HT)

Type de Client	Coût Composants (DZD)	Main-d'œuvre (DZD)	Formule de Calcul	Prix de Vente (DZD)	Prix de Vente (EUR)
Nouveaux Clients (Standard)	26 450 DA	7 500 DA	$(26450 + 7\,500) \times 2$	67 900 DA	450 €

♦ 4.4.2 – Octroi des Licences Logicielles (Software Licensing)

Le logiciel central, détaillé dans chapitre 3, fait l'objet d'une licence d'exploitation par serveur local installé :

Licence Initiale : Tarifée à 100 € (soit 15 160 DZD) par serveur central. Mais Les mises à jour de sécurité critiques restent gratuites à vie.

Mises à Jour Majeures (Upgrades) : Les évolutions logicielles majeures (nouvelles fonctionnalités, optimisations d'interface) sont facturées à hauteur de 80 % du prix de la licence initiale, soit 80 € (environ 12 130 DZD).

♦ 4.4.3 – Contrats de Tierce Maintenance Applicative (TMA)

L'exclusivité de notre expertise sur le produit nous permet de proposer deux modèles de maintenance (TMA) après une période de garantie initiale de 6 mois offerte au clients :

- **Option A : L'Abonnement Annuel Récurrent**

Le client souscrit à un contrat de maintenance préventive et corrective annuel. Le tarif est indexé à hauteur de 25 % du Prix Unitaire du matériel installé par station et par an : **16 980 DA / station / an**

- **Option B : La Facturation à l'Incident (Pay-per-Incident)**

Si le client refuse l'abonnement, les interventions d'urgence sont facturées à l'acte à hauteur de 50 % du Prix Unitaire par problème résolu. Le calcul respecte la règle d'imbrication suivante :

- Un problème isolé sur le serveur central est facturé à un seul forfait (50 % du Prix Unitaire).
- Si le problème implique le serveur central et N unités Edge de terrain, la facturation est calculée uniquement sur le nombre d'unités Edge affectées :

$$\text{Coût de l'intervention} = N \times (50 \% \times \text{Prix Unitaire})$$

(Le dépannage du serveur est alors offert, car l'accès au serveur est indispensable pour résoudre le problème des cartes Edge).

• Section 4.5 – Projections Financières sur 5 Ans :

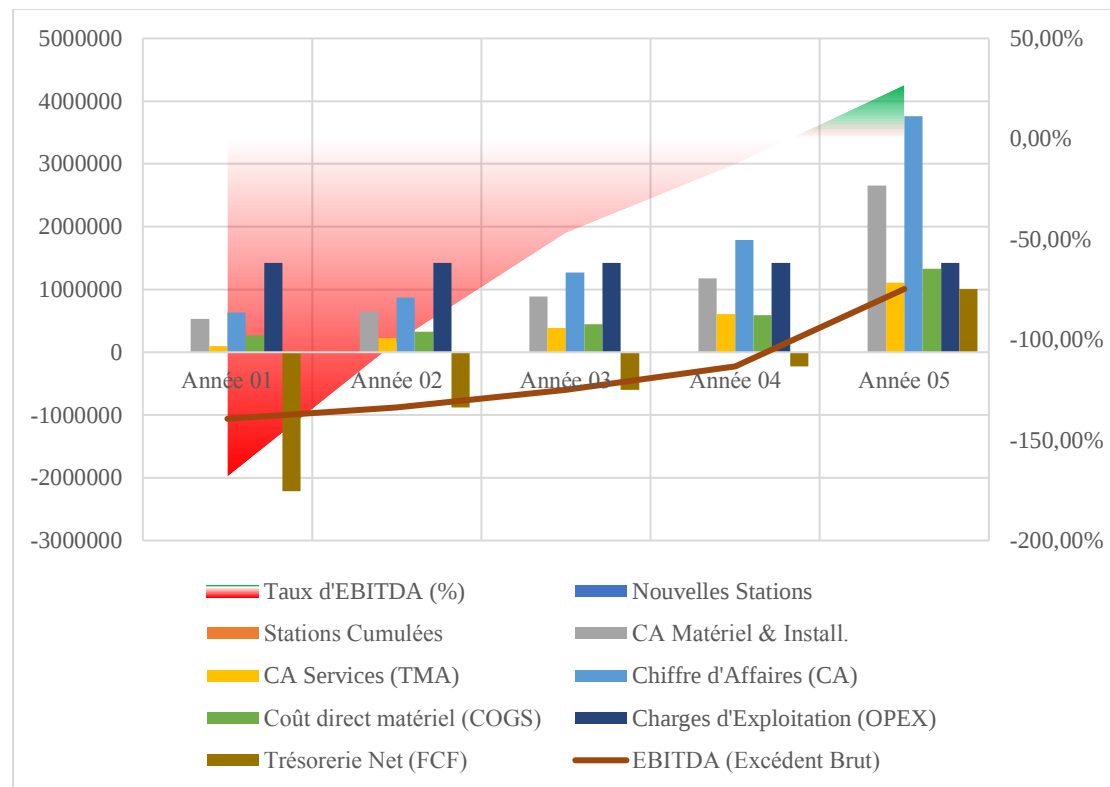
Pour valider la viabilité économique de la startup, nous présentons ici les prévisions financières sur cinq ans (de l'Année 01 à l'Année 05). Ces projections démontrent comment la récurrence des contrats de maintenance (TMA) stabilise le modèle économique, permettant une rentabilité dès la première année grâce à une tarification industrielle réaliste.

♦ 4.5.1 – Tableau des Projections Financières et KPIs (HT)

Les calculs ci-dessous intègrent l'augmentation progressive du parc de stations installées à travers le territoire national, en s'appuyant sur notre feuille de route commerciale. Ces données sont illustrées par le graphe de synthèse dans le Tableau 4.5.1.

Tableau 4.5.1 : Tableau de synthèse financière sur 5 ans

Indicateur Financier	Année 01	Année 02	Année 03	Année 04	Année 05
Nouvelles Stations	9	11	15	20	45
Stations Cumulées	9	20	35	55	100
CA Matériel & Install.	530 820 DA	648 780 DA	884 700 DA	1 179 600 DA	2 654 100 DA
CA Services (TMA)	99 540 DA	221 200 DA	387 100 DA	608 300 DA	1 106 000 DA
Chiffre d'Affaires (CA)	630 360 DA	869 980 DA	1 271 800 DA	1 787 900 DA	3 760 100 DA
Coût direct matériel (COGS)	265 410 DA	324 930 DA	442 350 DA	589 800 DA	1 327 050 DA
Charges d'Exploitation (OPEX)	1 425 000 DA	1 425 000 DA	1 425 000 DA	1 425 000 DA	1 425 000 DA
EBITDA (Excédent Brut)	-1 060 050 DA	-879 410 DA	-595 550 DA	-226 900 DA	1 008 050 DA
Taux d'EBITDA (%)	-168,17 %	-101,09 %	-46,82 %	-12,69 %	26,81 %
Trésorerie Net (FCF)	-2 210 050 DA	-879 410 DA	-595 550 DA	-226 900 DA	1 008 050 DA

Figure 4.5.1 : Graphe de synthèse financière sur 5 ans¹

• Conclusion :

L'étude économique et financière présentée dans ce chapitre confirme la viabilité du déploiement de notre solution de vision artificielle. En adoptant une structure de coûts allégée (Lean Startup) et en s'appuyant sur l'infrastructure de caméras existante des clients si possible, le projet minimise le capital d'investissement initial (CAPEX) tout en maximisant la valeur ajoutée technologique.

Les projections financières sur cinq ans prouvent que le modèle d'affaires, stabilisé par la récurrence des contrats de Tierce Maintenance Applicative (TMA), permet d'absorber le déficit de la première année et de générer des flux de trésorerie positifs à long terme. Cette solidité économique valide notre démarche d'entrepreneuriat industriel. Il convient désormais d'analyser rétrospectivement le cycle de développement du projet, d'en exposer les limites opérationnelles et d'en définir les perspectives d'amélioration futures au sein du conclusion.

¹ les valeurs de Taux d'EBITDA sont exprimés avec un axe secondaire au le droit de la figure

Conclusion Générale

Le travail de recherche et d'ingénierie logicielle présenté dans ce mémoire a permis de concevoir une solution complète de comptage automatique de passagers par vision artificielle. En structurant notre solution selon une architecture multiniveaux (l'Edge, le Serveur et le Dashboard), nous avons réussi à concilier les exigences de performance opérationnelle en temps réel de notre projet avec les dures réalités physiques et logistiques du terrain en Mostaganem.

I. Analyse Post-Mortem et Retour d'Expérience

L'ingénierie d'un tel système a été jalonnée de défis techniques majeurs, exigeant des arbitrages rigoureux :

- Le Pivot Algorithmique et les Limites d'Exécution :

La nécessité de protéger notre code pour le dépôt de brevet nous a conduits à délaissier le modèle YOLOv8 sous licence GPLv3 au profit de YoloX sous licence Apache 2.0. Ce pivot a exigé la création d'un fork personnalisé du dépôt original afin de corriger les incompatibilités d'un code source de 2022 vis-à-vis des dépendances logicielles récentes. Pour préserver la détection des passagers distants (micro-cibles) sur les images dégradées (faible débit à 472 Kbps) fournies par les caméras de la gare, l'implémentation de la tête de détection P2 a été indispensable. L'optimisation finale a été obtenue par une quantification INT8/FP16 via le compilateur RKNN et l'implémentation d'une gestion mémoire matérielle Zero-Copy (DMA-BUF) sous C++, éliminant la charge CPU liée aux duplications de tampons d'images.

- La Logique de Comptage Face au Terrain :

L'abandon d'une logique linéaire classique au profit du Gated Polygon a permis de résoudre le problème d'occlusion mutuelle dans les foules denses. De plus, notre machine à états est parvenue à réduire considérablement l'échauffement thermique des unités embarquées en gérant dynamiquement le cycle de service (veille) selon la présence ou l'absence d'une rame en station, un aspect crucial pour résister aux températures de 100°C pouvant être atteintes à l'intérieur des armoires électriques en plein été à Mostaganem.

II. Limites Identifiées et Arbitrages de Conception

Malgré des résultats d'émulation prometteurs, plusieurs limitations subsistent dans cette première version (V1) :

- Le Cas Limite du Départ de la Rame :

Une anomalie persiste lors de la désactivation du système : si un passager pénètre dans le polygone de détection au moment précis du départ de la rame, le gel du processus l'enregistre à tort comme une "Entrée". Bien que l'intégration d'un modèle de détection physique du tramway puisse résoudre ce problème, elle exigerait un surcoût computationnel difficilement supportable par le SoC RK3588.

- Fonctionnalités Exclues :

Le floutage des visages à la source a été exclu car jugé redondant ; les images n'étant jamais transmises en continu sur un réseau externe et les seules captures d'écrans de diagnostic transitant sous format chiffré Base64 sur le réseau LAN local de la société en question. De même, l'interface utilisateur (UI) du tableau de bord a été simplifiée pour privilégier la stabilité opérationnelle des WebSockets face à des contraintes de temps strictes.

III. Perspectives d'Avenir et Améliorations Futures

À l'avenir, le développement de cette solution s'orientera vers plusieurs axes d'amélioration technologique et commerciale :

- Optimisation Logicielle :

Nous envisageons la conception d'un modèle de détection léger développé entièrement à partir de zéro, ainsi que la compilation d'un système d'exploitation Linux embarqué personnalisé à faible empreinte mémoire via le projet YOCTO.

- Diversification du Produit :

Si ce projet était mis en œuvre sous forme d'entreprise, le prestataire pourrait répondre aux besoins de clients n'ayant pas l'infrastructure centralisée d'une grande société, par exemple SETRAM, nous prévoyons de décliner la solution sous trois "déclinaisons" : une version "Hors-ligne" stockant les données localement, une version "Monobloc" intégrant directement une caméra étanche, et une version "Grandes Charges" dotée d'un boîtier ultra-robuste avec absorption des chocs (pour une intégration future sur le matériel roulant des bus ou trains).

IV. Conclusion Opérationnelle

L'étude d'ingénierie et l'évaluation financière menées dans ce travail valident la viabilité globale de notre startup. En s'appuyant sur les caméras existantes et en adoptant une structure de coûts allégée, le projet limite le capital initial requis tout en offrant une rentabilité solide dès les premières années d'exploitation grâce à l'apport constant des contrats de maintenance applicative (TMA). Ce projet démontre qu'une synergie maîtrisée entre la recherche appliquée en vision artificielle et la stratégie d'entrepreneuriat industriel permet de répondre avec précision, sécurité et souveraineté aux défis de la mobilité urbaine de demain.

Bibliographie

- AGREEMENT SEEN SPUR TO COLOR TV. (1950). *New York Times*, 39.
Récupéré sur https://www.earlytelevision.org/pdf/nyt_8-16-50_agreement_is_seen_to_spur_color_tv.pdf
- Anton, M., Laura, L.-T. I., Stefan, R., & Konrad, S. (2016). *MOT16: A Benchmark for Multi-Object Tracking*. Récupéré sur arXiv: <https://arxiv.org/abs/1603.00831>
- Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 3464-3468). IEEE.
- Cedar Lake Ventures. (s.d.). *Average Summer Weather in Mostaganem, Algeria*. Consulté le Juin 20, 2026, sur WeatherSpark: <https://weatherspark.com/s/42391/1/Average-Summer-Weather-in-Mostaganem-Algeria>
- Computer History Museum. (2015, Novembre 15). *1956: ROTARY-HEAD DELIVERS HIGH-QUALITY VIDEO*. Récupéré sur Computer History Museum CHM: <https://www.computerhistory.org/storageengine/rotary-head-delivers-high-quality-video/>
- Dornberger, W. 1.-1. (1979). V-2. New York: Bantam Books.
- Fennell, D. (1988). *Investigation into the King's Cross Underground Fire*. London: Department of Transport.
- Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430*.
- Glinsky, A. (2000). *Theremin : ether music and espionage*. Urbana: University of Illinois Press.
- Isaac, R., Peter, R., Matvei, P., Deva, R., & Neehar, P. (2026). *RF-DETR: Neural Architecture Search for Real-Time Detection Transformers*. Récupéré sur arXiv: <https://arxiv.org/abs/2511.09554>

- Jinkun, C., Jiangmiao, P., Xinshuo, W., Rawal, K., & Kris, K. (2023, Mars 27). *Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking*. Récupéré sur arXiv: <https://arxiv.org/pdf/2203.14360>
- KUHN Francis. (2020, Mars 1). *Environnement projet tram Mostaganem*. Récupéré sur [masstransitpractices.com](https://masstransitpractices.com/content/uploads/2020/01/environnement-projet-tram-mostaganem.pdf):
[https://masstransitpractices.com/wp-content/uploads/2020/01/environnement-projet-tram-mostaganem.pdf](https://masstransitpractices.com/content/uploads/2020/01/environnement-projet-tram-mostaganem.pdf)
- Liu, W., Anguelov, D., Erhan, D., & Szegedy. (2016). SSD: Single Shot MultiBox Detector. *Computer Vision – ECCV 2016* (pp. 21-37). Springer International Publishing.
- london transport. (1997, Octobre 21). *Kings Cross Tragedy Means Safety First For London Underground*. Récupéré sur london transport: http://www.londontransport.co.uk/general/latest/kings_x.html
- Nicolai, W., Alex, B., & Dietrich, P. (2017, Mars 21). *Simple Online and Realtime Tracking with a Deep Association Metric*. Récupéré sur arXiv: <https://arxiv.org/abs/1703.07402>
- Orange Pi. (2026, Mars 03). *Orange Pi 5*. Récupéré sur Orange Pi: <http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5.html>
- Phillips, R. R. (2024, Mai 24). *First-Hand:Bing Crosby and the Recording Revolution*. Récupéré sur Engineering and Technology History Wiki: https://ethw.org/First-Hand:Bing_Crosby_and_the_Recording_Revolution
- Phillips, R. R. (2024, Mai 24). *First-Hand:Bing Crosby and the Recording Revolution*. Récupéré sur Engineering and Technology History Wiki: https://ethw.org/First-Hand:Bing_Crosby_and_the_Recording_Revolution#The_Desire_for_Video_Recording
- radio algerie. (2016, février 14). *Mise en place de 4 nouveaux groupes publics de transport*. Récupéré sur Radio Algérienne: <https://radioalgerie.dz/news/fr/article/20160214/68567.html>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE*

- Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779-788). IEEE.
- Ruopeng, G., Ji, Q., & Limin, W. (2025, Mars 25). *Multiple Object Tracking as ID Prediction*. Récupéré sur arXiv: <https://arxiv.org/abs/2403.16848>
- Stauffer, C., & Grimson, W. (1999). Adaptive Background Mixture Models for Real-Time Tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2246-2252). Fort Collins: IEEE Computer Society.
- Transport for London Corporate Archives. (2024). *Research Guide No 46: CCTV*. London: Transport for London. Récupéré sur <https://content.tfl.gov.uk/research-guide-46-cctv.pdf>
- Vladimir, S., Baptiste, S., Victor, J., Alexandre, A., & Christophe, D. V. (2025). *CAMELTrack: Context-Aware Multi-cue ExpLoitation for Online Multi-Object Tracking*. Récupéré sur arXiv: <https://arxiv.org/abs/2505.01257>
- Wellens, A. R., & Ergener, D. (1986, Septembre 01). An improved video multiplexer for behavioral research. *Behavior Research Methods, Instruments, & Computers*, 18, 480-481. doi:10.3758/BF03201417
- Xingkui, Z., Shuchang, L., Xu, W., & Qi, Z. (2021). *TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios*. Récupéré sur arXiv: <https://arxiv.org/pdf/2108.11539>
- Yi, K., Li, J., & Zhang, Y. (2025). Multi-Object Tracking with Confidence-Based Trajectory Prediction Scheme. *Sensors*. doi:10.3390/s25237221
- Yian, Z., Wenyu, L., Shangliang, X., Jinman, W., Guanzhong, W., Qingqing, D., . . . Jie, C. (2024, Avril 3). *DETRs Beat YOLOs on Real-time Object Detection*. Récupéré sur arXiv: <https://arxiv.org/abs/2304.08069>
- Yifu, Z., Peize, S., Yi, J., Dongdong, Y., Fucheng, W., Zehuan, Y., . . . Xinggang, W. (2022, Octobre 13). *ByteTrack: Multi-Object Tracking by Associating Every Detection Box*. Récupéré sur arXiv: <https://arxiv.org/pdf/2110.06864>