REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
–T L E M C E N–



المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
- تلمسان -

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique

Présenté par : BORSALI Atika

Thème

Conception d'un modèle de gestion de stock pharmaceutique par reconnaissance visuelle basée sur les réseaux de neurones convolutifs

Soutenu publiquement, le 22/06/2026, devant le jury composé composé de :

Dr CHIALI Anisse	MCA	ESSA. Tlemcen	Président
Dr BEKKAOUI Mokhtaria	MCB	ESSA. Tlemcen	Directrice de mémoire
Dr FANDI Wahiba	MCB	ESSA. Tlemcen	Co-Directrice de mémoire
Dr KARAOUZENE Zoheir	MCB	ESSA. Tlemcen	Examinateur 1
Dr MOKHTARI Rida	MCA	ESSA. Tlemcen	Examinateur 2

Année universitaire : 2025/2026


بسم الله الرحمن الرحيم

Dédicaces

Je dédie ce mémoire

à mes chers parents,

à mon frère Nazim,

à toute ma famille.

Remerciements

Il reste difficile de citer tous ceux qui le méritent et j'espère n'oublier personne au moment de la finalisation de mon mémoire de fin d'études.

Je tiens avant tout à remercier Allah, le Tout-Puissant, qui m'a donné la force, la santé et la volonté de mener à bien ce travail.

Je remercie en particulier mes encadrantes Madame FANDI Wahiba et Madame BEKKAOUI Mokhtaria pour leur patience, leur disponibilité, ainsi que leurs judicieux conseils. Comme je voudrais adresser mes sincères remerciements et toute ma gratitude à Madame NEDJAR Imane, qui a analysé mes résultats et m'a prodigué des conseils d'une grande richesse, ainsi qu'à Monsieur M'HAMEDI Mohamed, dont l'aide m'a été précieuse pour identifier et corriger les erreurs liées au matériel.

Je tiens à exprimer ma profonde gratitude à la pharmacie MERZOUK, et tout particulièrement à son propriétaire Monsieur MERZOUK Azeddine, pour son accueil exceptionnel. Il m'a non seulement permis de prendre des photos des médicaments nécessaires à mon travail, mais il m'a également généreusement communiqué leurs données de ventes. Sa disponibilité, sa confiance et sa grande générosité m'ont été d'une aide précieuse. Qu'il trouve ici l'assurance de mes sincères remerciements.

Je remercie également le responsable du FABLAB Monsieur ADJIM Ramz-Eddine Abderazak pour son aide précieuse et son accompagnement tout au long de la réalisation de mon prototype. Il a su mettre à ma disposition les machines et les outils nécessaires, me guider dans les étapes techniques, et me prodiguer de précieux conseils chaque fois que j'étais confronté à une difficulté. Grâce à sa disponibilité et sa patience, j'ai pu concrétiser ce projet. Qu'il trouve ici l'expression de ma profonde gratitude.

Je remercie Monsieur CHIALI Anisse de m'avoir fait l'honneur de présider mon jury de mémoire.

Je remercie également Monsieur KARAOUZENE Zoheir et Monsieur MOKHTARI Rida qui ont bien voulu juger ce travail.

À mes chers parents, aucun mot ne saurait exprimer assez ma gratitude. Vous avez été mon refuge, ma force et mon exemple. Vos sacrifices, votre amour inconditionnel et vos encouragements constants m'ont permis d'arriver là où je suis. Je n'y serais jamais parvenu sans vous. Ce travail est le vôtre, il est l'aboutissement de tout ce que vous m'avez transmis. Merci pour tout, du plus profond de mon cœur. Et pour terminer, mon petit frère : tu es important pour moi.

Table des matières

Introduction générale **11**

1 État de l'art – Gestion de stock **13**

1.1 Introduction . . . 13
1.2 Définition du stock . . . 13
1.3 Types de stocks . . . 14
1.4 Fonctions et rôles des stocks . . . 14
1.4.1 Fonction de spéculation . . . 15
1.4.2 Fonction économique . . . 15
1.4.3 Fonction de sécurité . . . 15
1.4.4 Fonction de régulation . . . 15
1.4.5 Fonction technique . . . 15
1.5 Définition de la gestion des stocks . . . 15
1.6 Fonctions de la gestion des stocks . . . 16
1.6.1 Gestion des approvisionnements . . . 16
1.6.2 Contrôle des entrées et sorties . . . 16
1.6.3 Suivi des niveaux de stock . . . 16
1.6.4 Planification des commandes . . . 16
1.6.5 Prévision de la demande . . . 16
1.6.6 Optimisation des coûts de stockage . . . 16
1.7 Méthodes classiques de gestion des stocks . . . 17
1.7.1 La méthode EOQ (Quantité économique de commande) . . . 17
1.7.2 Le système de point de commande . . . 17
1.7.3 Les registres manuels et livres comptables . . . 17
1.8 Digitalisation de la gestion des stocks . . . 18
1.8.1 Les systèmes ERP (*Enterprise Resource Planning*) . . . 18
1.8.2 Les codes-barres et QR codes . . . 19
1.8.3 La technologie RFID (*Radio Frequency Identification*) . . . 19
1.8.4 Les systèmes IoT (capteurs connectés) . . . 20
1.8.5 Le système WMS (*Warehouse Management System*) . . . 21
1.9 Intelligence artificielle dans la gestion des stocks . . . 22
1.9.1 Machine learning . . . 22
1.9.2 Vision par ordinateur et détection d'objets . . . 22
1.9.3 Travaux existants . . . 23
1.10 Conclusion . . . 24

2 Intelligence artificielle et détection d'objets **25**
2.1 Introduction . . . 25
2.2 Intelligence artificielle . . . 25
2.3 Machine Learning . . . 26
2.3.1 Types d'apprentissage . . . 26
2.4 Deep Learning . . . 27
2.4.1 Fonctionnement des réseaux de neurones . . . 27
2.4.2 Réseaux de neurones convolutifs (CNN) . . . 28
2.4.3 Transfer Learning . . . 30
2.5 Vision par ordinateur . . . 31
2.5.1 Traitement d'image . . . 32
2.5.2 Extraction des caractéristiques . . . 32
2.5.3 Classification d'images . . . 32
2.5.4 Détection d'objets . . . 33
2.5.5 Segmentation d'images . . . 33
2.6 Détection d'objets . . . 33
2.6.1 Approches de détection . . . 34
2.6.2 Étapes du processus de détection . . . 34
2.6.3 Boîtes englobantes . . . 34
2.7 Modèles de détection d'objets utilisés . . . 35
2.7.1 Présentation de YOLO . . . 35
2.7.2 Architecture de YOLOv8 . . . 36
2.7.3 Architecture de YOLO26 . . . 38
2.7.4 RT-DETR (Real-Time Detection Transformer) . . . 38
2.8 Conclusion . . . 40

3 Modèle d'apprentissage entraîné pour la supervision du stock **41**
Introduction . . . 41
3.1 Constitution de la base de données . . . 41
3.1.1 Collecte des images . . . 41
3.1.2 Annotation des images . . . 42
3.1.3 Division et redimensionnement du dataset . . . 42
3.1.4 Augmentation des données . . . 43
3.2 Entraînement des modèles . . . 45
3.2.1 Approches de vision par ordinateur . . . 45
3.2.2 Environnement d'entraînement . . . 45
3.2.3 Architecture YOLO . . . 45
3.3 Métriques d'évaluation . . . 46
3.4 Résultats de YOLOv8 . . . 47

3.4.1 Paramètres d'entraînement 47
3.4.2 Résultats globaux 48
3.4.3 Analyse des courbes d'entraînement 49
3.4.4 Matrice de confusion 51
3.5 Résultats de YOLO26 52
3.5.1 Présentation du modèle 52
3.5.2 Résultats globaux 52
3.5.3 Analyse des courbes et des prédictions 52
3.5.4 Matrice de confusion 54
3.6 Résultats de RT-DETR 55
3.6.1 Présentation du modèle 55
3.6.2 Résultats globaux 56
3.6.3 Analyse 56
3.6.4 Matrice de confusion 57
3.7 Comparaison et choix du modèle final 59
3.7.1 Tableau comparatif 59
3.7.2 Analyse comparative 59
3.8 Conclusion 59

Conclusion générale **61**

Références bibliographiques **63**

Liste des figures

1.1 Fonctions et rôles des stocks. 14
1.2 Système ERP [37]. 18
1.3 Les codes-barres et QR codes [38]. 19
1.4 Système RFID [39]. 20
1.5 Système IoT [40]. 21
1.6 Warehouse Management System [41]. 21
1.7 Architecture globale de l'IA appliquée à la gestion des stocks. 23

2.1 Trois types de couches d'un réseau de neurones [4]. 28
2.2 Illustration d'un filtre dans un CNN. 29
2.3 Deux types de couches d'un CNN. 30
2.4 Exemple de max pooling ($4\times4 \rightarrow 2\times2$). 30
2.5 Étapes de la vision par ordinateur [8]. 32
2.6 Classification, détection et segmentation d'images. 33
2.7 Architecture de YOLOv8. 36
2.8 Fonctions de perte de YOLOv8 [14]. 38
2.9 Architecture de RT-DETR. 39

3.1 Collecte des images 42
3.2 Redimensionnement de l'ensemble des images 43
3.3 La base de données finale 44
3.4 Courbes de perte et de métriques de YOLOv8 49
3.5 Prédictions du modèle YOLOv8 sur les images de validation 50
3.6 Prédictions du modèle sur les images de validation 50
3.7 Matrice de confusion de YOLOv8 51
3.8 Courbes de perte (*box*, *cls*, *dfl*) – YOLO26 53
3.9 Prédictions du modèle YOLO26 sur les images de validation 53
3.10 Matrice de confusion de YOLO26 55
3.11 Courbes d'entraînement de RT-DETR 56
3.12 Prédictions du modèle RT-DETR sur les images de validation 57
3.13 Matrice de confusion de RT-DETR 58

Liste des tableaux

2.1 Les cinq variantes de YOLOv8 [15]. 38
2.2 Deux versions de RT-DETR. 40

3.1 Les transformations appliquées via Roboflow 44
3.2 Métriques d'évaluation 46
3.3 Métriques globales de YOLOv8 48
3.4 Détections correctes par classe – YOLOv8 51
3.5 Métriques globales de YOLO26 52
3.6 Détections correctes par classe – YOLO26 54
3.7 Métriques globales de RT-DETR 56
3.8 Détections correctes par classe – RT-DETR 58
3.9 Comparaison des trois modèles 59

Table des abréviations

Abréviation	Signification
AI / IA	Intelligence Artificielle
AIFI	Attention-based Intra-scale Feature Interaction
AMP	Automatic Mixed Precision
BCE	Binary Cross-Entropy
CCFM	Cross-Scale Feature Fusion Module
CIoU	Complete Intersection over Union
CNN	Convolutional Neural Network (Réseau de Neurones Convolutifs)
COCO	Common Objects in Context
CPU	Central Processing Unit
DFL	Distribution Focal Loss
EOQ	Economic Order Quantity (Quantité Économique de Commande)
ERP	Enterprise Resource Planning
FLOPs	Floating Point Operations per Second
FPN	Feature Pyramid Network
GPU	Graphics Processing Unit
IoT	Internet of Things (Internet des Objets)
IoU	Intersection over Union
mAP	Mean Average Precision
MuSGD	Momentum Stochastic Gradient Descent
NMS	Non-Maximum Suppression
PAN	Path Aggregation Network
PASCAL VOC	Pattern Analysis, Statistical Modelling and Computational Learning – Visual Object Classes
RFID	Radio Frequency Identification
RT-DETR	Real-Time Detection Transformer
STAL	Segment-Task Alignment Loss
SURF	Speeded-Up Robust Features
VRAM	Video Random Access Memory
WMS	Warehouse Management System
YOLO	You Only Look Once

Introduction générale

Dans le secteur pharmaceutique, l'indisponibilité des médicaments n'est pas juste un problème de stock mais c'est une question de santé publique. Une rupture de stock non détectée à temps peut avoir des conséquences bien plus sérieuses. Dans les cas plus critiques, en particulier pour les personnes atteintes de maladies chroniques ou sous traitement continu, une interruption même brève peut déstabiliser un équilibre thérapeutique fragile et aggraver leur état de santé, et très souvent, ce n'est pas que le médicament était vraiment absent mais c'est plutôt que personne ne s'est rendu compte qu'il n'en restait plus. Les avancées récentes en intelligence artificielle et en vision par ordinateur ouvrent une voie intéressante. L'idée de base est simple : plutôt qu'un employé parcoure les rayons et compte tout à la main, on installe une caméra qui fait tout ce travail à sa place. Derrière cette caméra, un algorithme analyse en permanence ce qu'il voit : d'abord il doit être capable de reconnaître chaque boîte de médicament avec son nom exact, puis évalue ce qui reste et signale automatiquement quand un produit est sur le point de manquer. Ce qui rend cette approche particulièrement intéressante, c'est qu'elle peut s'appliquer à tous les niveaux, d'une simple pharmacie de quartier jusqu'aux grands laboratoires et aux multinationales. Les outils existent, ils sont accessibles, et les résultats qu'ils donnent sont suffisamment fiables pour envisager une utilisation dans des contextes réels. C'est exactement ce que ce mémoire cherche à explorer, en tentant de construire un système capable de détecter et de reconnaître des boîtes de médicaments dans une pharmacie, pour assurer un suivi de stock automatique et continu. Ce qui nous a conduit à ce sujet, c'est d'abord notre passion pour l'intelligence artificielle et ce qu'elle est capable d'accomplir. C'est un domaine très puissant qui touche à tout : la médecine, l'industrie, le commerce, la sécurité. Il peut transformer des secteurs entiers, résoudre des problèmes concrets, automatiser des tâches qui semblaient jusqu'ici difficilement remplaçables. En voyant que la gestion de stock dans les pharmacies algériennes se faisait encore à la main pour la plupart, l'idée de ce travail s'est imposée d'elle-même : utiliser le Deep Learning pour régler un problème aussi concret et aussi proche de nous. La gestion des stocks a pas mal changé au fil des années. Au départ, on travaillait avec des formules comme la méthode EOQ (Economic Order Quantity) ou le point de commande, des calculs faits à la main pour savoir quand et combien commander. Avec le temps, des outils plus évolués sont apparus comme les logiciels ERP (Enterprise Resource Planning), les codes-barres, les puces RFID (Radio Frequency Identification), les capteurs IoT (Internet of Things). Mais malgré tout, toutes ces technologies ont besoin qu'une personne intervienne, que ce soit pour scanner, vérifier ou corriger, et ne permettent

pas de surveillance visuelle autonome. C'est là qu'intervient l'intelligence artificielle. Le Deep Learning en particulier donne aux machines la capacité de reconnaître des objets dans une image, exactement comme le fait l'œil humain. Derrière cela, il y a des réseaux de neurones convolutifs qui analysent l'image morceau par morceau et en tirent les informations les plus importantes et utiles. Appliquée à la gestion de stock, cette technologie permet de détecter automatiquement les produits présents dans un rayon, d'identifier leur nom et de localiser leur position précise dans l'image, le tout sans qu'aucune personne n'ait besoin d'intervenir. La question à laquelle ce mémoire essaie de répondre est donc la suivante : est-ce que les modèles de détection d'objets basés sur le Deep Learning peuvent réellement reconnaître chaque boîte de médicament et les distinguer les unes des autres, avec une précision suffisante pour concevoir une gestion de stock en temps réel ? Pour répondre à cette question, nous avons suivi une démarche expérimentale. Nous avons d'abord créé notre propre base de données à partir d'images prises dans une pharmacie ; elle couvre 90 références de médicaments commercialisés en Algérie. Ces images ont été annotées, traitées puis augmentées via la plateforme Roboflow. Ensuite, nous avons effectué un entraînement de trois modèles de détection d'objets : YOLO8, YOLO26 et RT-DETR sur Google Colab avec un GPU T4. Et enfin, nous avons comparé les résultats des trois modèles à travers des métriques comme la précision, le rappel et le mAP. L'objectif est de trouver le modèle qui offre le meilleur compromis entre précision et vitesse, pour pouvoir reconnaître et localiser les boîtes de médicaments. Ce travail présente la première étape, et la plus importante, vers un système de supervision de stock plus complet que nous comptons intégrer dans le projet d'ingéniorat. Ce mémoire s'organise autour de trois chapitres. Le premier décrit l'évolution de la gestion de stock, des méthodes classiques jusqu'à l'arrivée de l'intelligence artificielle et la vision par ordinateur. Le deuxième chapitre définit les bases théoriques concernant l'intelligence artificielle : nous avons expliqué ce qu'est le Machine Learning, le Deep Learning, les réseaux de neurones convolutifs, et nous avons détaillé les modèles de détection utilisés ainsi que leur architecture. Le troisième chapitre est entièrement consacré à la partie pratique, à savoir la construction de la base de données, l'entraînement des modèles, l'interprétation des résultats et la comparaison des performances. Nous terminerons par une conclusion générale qui synthétise les principaux résultats obtenus et ouvre sur les perspectives futures de ce travail.

Chapitre 1

État de l'art – Gestion de stock

1.1 Introduction

La gestion des stocks est l'un des piliers essentiels du bon fonctionnement d'une entreprise. Elle sert à encadrer les activités liées au stockage, à réduire les coûts, à optimiser la production et à assurer ainsi la continuité de l'activité. Dans le secteur pharmaceutique, elle constitue un élément stratégique qui garantit la disponibilité des médicaments, notamment ceux à forte rotation ou dont l'absence peut avoir des conséquences graves sur la santé des patients.

Dans ce chapitre, nous commençons par définir la notion de stock, ses différents types ainsi que ses fonctions au sein de l'entreprise. Nous expliquons ensuite ce qu'est la gestion des stocks et quelles sont ses missions. Puis, nous abordons les méthodes de gestion des stocks, la digitalisation de la gestion des stocks, ainsi que l'apparition de l'intelligence artificielle et de la vision par ordinateur. Nous citons également quelques travaux de recherche dans ce domaine et nous terminons par une conclusion.

1.2 Définition du stock

Un stock constitue l'ensemble de produits qui peuvent être de différentes natures : matières premières, produits semi-finis, produits finis, pièces de rechange, marchandises stockées dans des entrepôts ou magasins en attente d'être utilisés ou vendus.

Le stock est mis en place par l'ensemble des agents économiques, tels que les entreprises industrielles, les entreprises commerciales et les distributeurs, ainsi que d'autres acteurs comme les hôpitaux, les pharmacies, les supérettes et les centres commerciaux, etc.

Le stock n'est pas une option supplémentaire mais bien une nécessité absolue qui impacte le bon fonctionnement d'une entreprise. Parmi les avantages apportés par le stock :

— Il permet une adaptabilité aux variations de la demande des clients.

— Il assure la disponibilité des produits même en cas de pannes de machines et permet donc d'éviter les ruptures de stocks.

— Il permet aussi de profiter des périodes où la demande des clients est à son pic.

1.3 Types de stocks

Les produits qu'on stocke sont classés selon leur nature, leur rôle et leur position dans la chaîne de production, parmi lesquels :

— **Stock de matières premières** : il contient toutes les matières premières que l'entreprise compte utiliser pour la production.

— **Stock des produits en cours de fabrication** : désigne les articles qui sont déjà entrés dans le processus de production mais pas encore terminés. On les retrouve entre les différentes étapes de production.

— **Stock des produits finis** : ce sont les produits complètement fabriqués et prêts à être livrés au client.

— **Stock de sécurité** : c'est une réserve supplémentaire maintenue par l'entreprise pour gérer les imprévus et éviter les ruptures de stock.

— **Stock de marchandises** : il contient des produits achetés à l'extérieur par l'entreprise qu'elle compte revendre sans qu'aucune transformation ne soit effectuée.

— **Le stock saisonnier** : c'est un stock constitué à l'avance pour les périodes de forte demande liées à des évènements comme les fêtes, les saisons ou les promotions annuelles.

— **Stock de pièces de rechange et accessoires** : c'est l'ensemble des composants nécessaires à la maintenance. Il joue un rôle essentiel dans la continuité de la production et dans le bon fonctionnement des équipements.

1.4 Fonctions et rôles des stocks

Le stock remplit plusieurs fonctions essentielles au bon fonctionnement d'une entreprise.

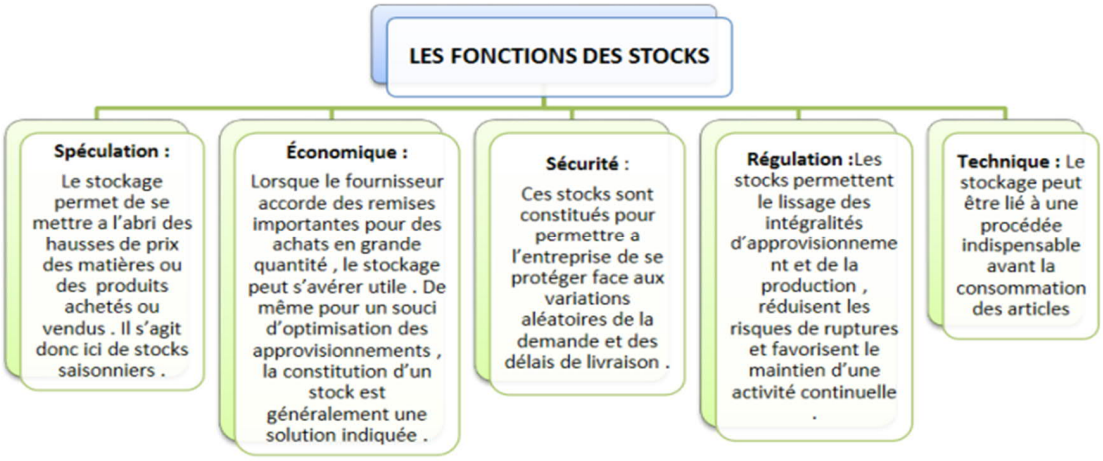


FIGURE 1.1 – Fonctions et rôles des stocks.

1.4.1 Fonction de spéculation

Le stock permet de protéger l'entreprise contre les hausses de prix des matières premières, et lorsque les prix sont bas, il aide l'entreprise à en acheter en grandes quantités. De cette façon, l'entreprise économise beaucoup d'argent et assure la continuité de son approvisionnement.

1.4.2 Fonction économique

Lorsqu'un fournisseur propose des remises pour des achats en grande quantité, si une entreprise dispose d'un stock, cela lui permettra de mieux gérer ses dépenses et de faire des économies sur les coûts de transport et de commande.

1.4.3 Fonction de sécurité

Les stocks de sécurité ont pour but de protéger l'entreprise contre les événements imprévus, par exemple les variations soudaines de la demande ou encore les retards de livraison des fournisseurs. Cette fonction garantit une bonne continuité de l'activité et assure la satisfaction des clients.

1.4.4 Fonction de régulation

Le stock régule les écarts et absorbe les décalages, que ce soit temporels ou quantitatifs, entre l'approvisionnement en matières premières, la production et la livraison aux clients. De cette façon, l'entreprise maintient une activité stable et diminue les perturbations ou les déséquilibres.

1.4.5 Fonction technique

Le stock est dans certaines entreprises obligatoire et présente une étape indispensable ; par exemple, certains produits nécessitent une période de repos ou de séchage avant d'être livrés ou entre les étapes de production, car cela impacte la qualité finale du produit [17].

1.5 Définition de la gestion des stocks

La gestion des stocks présente l'ensemble des activités qui permettent de prendre des décisions sur le moment et la quantité à réapprovisionner pour un produit. Son défi consiste à trouver un bon équilibre entre les coûts liés au stockage et la satisfaction des clients, autrement dit à éviter de commander à la fois une très faible quantité — car cela peut augmenter les risques de rupture de stock et une perturbation du système de

production entier — et une grande quantité, pour ne pas augmenter les coûts liés au stockage. La gestion de stock vise aussi à minimiser les coûts liés à la livraison.

1.6 Fonctions de la gestion des stocks

1.6.1 Gestion des approvisionnements

La gestion de stock permet de fournir une quantité suffisante de marchandises ou de matières premières pour répondre aux besoins des clients.

1.6.2 Contrôle des entrées et sorties

La gestion de stock permet de suivre et de contrôler chaque entrée en stock du fournisseur et chaque expédition des produits vers les clients, en enregistrant les quantités et les dates exactes.

1.6.3 Suivi des niveaux de stock

En gérant et en contrôlant les différentes catégories de stocks :

— **Le stock minimum** : c'est la quantité minimale dont doit disposer une entreprise pour assurer la continuité de la production.

— **Le stock de sécurité** : c'est un stock conservé en cas d'événement imprévu.

— **Le stock d'alerte** : c'est le niveau qui déclenche le besoin de passer une nouvelle commande.

— **Le stock maximum** : c'est la quantité de produits qu'il ne faut pas dépasser.

1.6.4 Planification des commandes

La gestion des stocks aide à identifier avec précision les articles qui se vendent le plus rapidement et ceux qui ne le sont pas, et cela permet de définir la bonne quantité à commander et quel seuil d'alerte ne pas atteindre.

1.6.5 Prévision de la demande

La gestion de stock permet d'estimer à l'avance la quantité de produits que l'entreprise doit acheter ; cela permet de réduire les retards de production, de calculer les dépenses futures et d'avoir une meilleure visibilité du déroulement de l'activité [18].

1.6.6 Optimisation des coûts de stockage

Elle permet d'optimiser les coûts de commande (frais administratifs, télécommunications, salaires) et les coûts de stockage (éclairage, loyer, assurances, transport) [19].

1.7 Méthodes classiques de gestion des stocks

Avant l'apparition des outils informatiques, la gestion des stocks s'appuyait sur plusieurs méthodes manuelles et classiques, parmi lesquelles :

1.7.1 La méthode EOQ (Quantité économique de commande)

L'EOQ (*Economic Order Quantity*) est une méthode qui aide à choisir la quantité optimale de produits à commander et la date à laquelle passer commande auprès des fournisseurs, afin de minimiser les coûts de stockage tout en évitant les quantités excessives ou insuffisantes. Ses principaux avantages sont l'élimination du gaspillage et l'optimisation des dépenses.

La quantité de commande est déterminée à l'aide de la formule du lot économique EOQ. C'est le modèle de dimensionnement de stocks le plus courant. Pour ce modèle, la consommation est supposée régulière et le délai de livraison est connu. Le stock moyen vaut la moitié de la quantité commandée ($Q/2$). Optimiser le stock revient alors à optimiser la quantité commandée, appelée quantité économique [20].

1.7.2 Le système de point de commande

Cette méthode est la plus répandue dans les entreprises. Son principe est le suivant : lorsque le niveau de stock atteint une valeur prédéfinie appelée « point de commande », une commande doit être déclenchée. Elle permet d'éviter les ruptures de stock et de garantir la disponibilité des produits pour répondre aux besoins des clients. Un point de commande bien défini assure un bon équilibre entre la satisfaction des clients et les dépenses liées au stock. En revanche, s'il est mal calculé, l'entreprise peut se retrouver en situation de surstock ou de sous-stock.

Cette politique de gestion des stocks requiert la définition de deux paramètres : le seuil (point de commande) et la quantité de réapprovisionnement. Dans le cas où la demande ou le délai d'approvisionnement serait aléatoire, cette quantité est généralement majorée à l'aide d'un délai de sécurité [21].

1.7.3 Les registres manuels et livres comptables

Ce sont des documents physiques utilisés pour enregistrer tous les mouvements et variations de stock ainsi que les informations propres à chaque article (numéro de code, désignation, numéro de lot). Cette méthode est lente et sujette aux erreurs humaines [22].

De nombreuses entreprises s'appuient encore sur ces méthodes traditionnelles, mais elles freinent considérablement leur développement. Voici leurs principales limites :

— Manque de visibilité sur la demande : fondées sur des prévisions simples et approximatives, elles peuvent conduire à une incapacité à couvrir les besoins des clients.

— Manque de communication et de coordination entre les étapes du processus : par exemple, le service de production peut manquer de matières premières sans que le service d'approvisionnement en soit informé, entraînant des blocages, des ruptures ou des surstocks.

— Risque élevé d'erreurs humaines lors de la saisie des données, du comptage des produits ou de l'interprétation des informations.

— Lenteur du processus.

— Absence de suivi en temps réel, donnant un niveau de stock inexact [23].

1.8 Digitalisation de la gestion des stocks

Les méthodes classiques de gestion de stock souffrent d'une inadaptabilité face aux changements de la demande et aux défis actuels du marché. C'est pourquoi il a été nécessaire de développer des technologies offrant une meilleure flexibilité et facilitant les tâches. Parmi ces technologies :

1.8.1 Les systèmes ERP (*Enterprise Resource Planning*)

Un système ERP est une plateforme qui centralise les informations et les données de chaque service. Il est utilisé par les entreprises pour gérer les finances, la comptabilité, les ressources humaines, la production, les stocks, etc. Il améliore l'organisation de l'entreprise et facilite la prise de décision [24]. Parmi les systèmes ERP existants, SAP ERP est l'un des plus répandus ; il gère les opérations d'une entreprise telles que les ventes, la production et les achats [25].

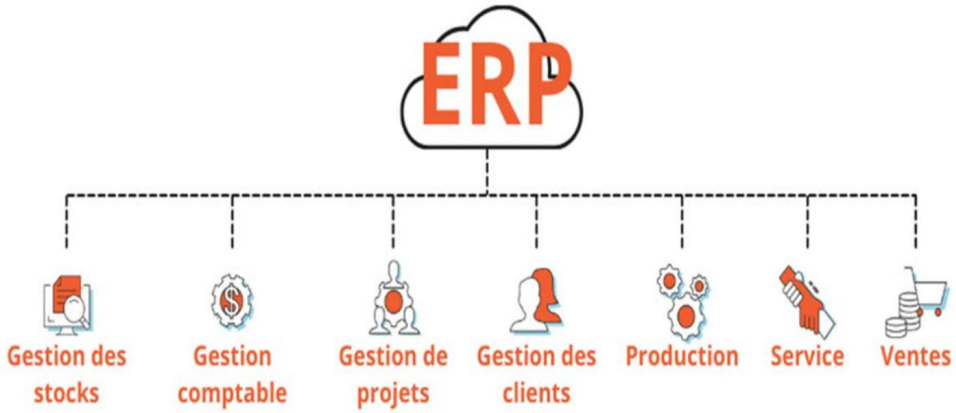


FIGURE 1.2 – Système ERP [37].

1.8.2 Les codes-barres et QR codes

Un code-barres est une combinaison de bandes noires verticales de largeur, de hauteur et d'espacement variables, permettant d'encoder de manière unique chaque produit. Largement utilisée en raison de sa simplicité et de sa fiabilité, cette méthode automatise la gestion des sorties de stock et simplifie l'inventaire, permettant de gagner un temps considérable [26].



FIGURE 1.3 – Les codes-barres et QR codes [38].

1.8.3 La technologie RFID (*Radio Frequency Identification*)

La RFID est une technologie qui utilise les ondes radio haute fréquence pour détecter la présence ou l'absence d'un produit doté d'une puce RFID et en lire le numéro d'identification. Très utilisée dans les secteurs industriel et commercial, elle offre plusieurs avantages :

— Suivi du stock en temps réel.

— Réduction des manutentions et des manœuvres.

— Identification facile et précise d'un lot de production.

— La puce RFID est très petite et peut se placer n'importe où.

— Elle assure la sécurité des données lors de leur transfert et peut se coupler à Internet via Wi-Fi.



FIGURE 1.4 – Système RFID [39].

1.8.4 Les systèmes IoT (capteurs connectés)

L'IoT (*Internet of Things*, ou Internet des objets) est un outil qui permet d'interconnecter des objets grâce aux technologies de l'information et de la communication. Appliqué à la gestion des stocks, il fait référence à un réseau contenant plusieurs appareils, capteurs et logiciels connectés via Internet, offrant une visibilité globale et précise du niveau de stock.

L'IoT présente plusieurs avantages : surveillance en temps réel du niveau de stock, génération de données liées à chaque produit et renforcement de la sécurité du magasin.

Bien que ces technologies aient apporté une meilleure traçabilité des produits et une automatisation partielle du processus, elles ne suffisent plus pour répondre aux exigences actuelles : intelligence artificielle, autonomie complète dans la prise de décision et modernisation des entreprises.



FIGURE 1.5 – Système IoT [40].

1.8.5 Le système WMS (*Warehouse Management System*)

Le WMS est un logiciel qui automatise la gestion des flux physiques dans un entrepôt, depuis la réception des marchandises jusqu'à leur expédition vers les clients. Il prend en charge l'ensemble des opérations : réception, rangement, suivi des stocks, réapprovisionnement, emballage et expédition. Le WMS s'appuie sur les technologies présentées précédemment — codes-barres, RFID — pour les intégrer dans un système de pilotage cohérent.

Comparé à l'ERP, le WMS propose des fonctionnalités plus complètes et plus spécialisées pour la gestion physique de l'entrepôt : gestion des emplacements, préparation de commandes, communication avec les terminaux mobiles de manutention et gestion des codes-barres.



FIGURE 1.6 – Warehouse Management System [41].

1.9 Intelligence artificielle dans la gestion des stocks

Les méthodes traditionnelles de gestion des stocks et les technologies de digitalisation citées précédemment nécessitent toujours l'intervention humaine, ce qui peut entraîner des inefficacités et des erreurs parfois importantes. L'intelligence artificielle (IA) et le machine learning appliqués à la gestion des stocks ont démontré leur capacité à rendre le système entièrement automatique et autonome, même dans la prise de décision, rendant l'optimisation du niveau de stock plus flexible et efficace.

L'un des rôles principaux de l'IA dans la gestion des stocks est d'analyser de grandes quantités de données en temps réel, permettant aux entreprises de surveiller en permanence les niveaux de stock.

1.9.1 Machine learning

Le machine learning est une branche de l'intelligence artificielle qui a transformé la gestion des stocks. Il permet d'éliminer considérablement la surveillance humaine, et son rôle principal réside dans l'analyse des données, l'optimisation et l'automatisation du processus [27].

Les techniques de machine learning permettent notamment de :

— Traiter de grandes quantités de données et analyser des informations complexes très rapidement, ce qui permet de prévoir avec une grande précision les demandes et les résultats futurs.

— Optimiser les niveaux de stock en tenant compte de la demande de chaque produit et de la capacité de l'entrepôt, afin d'éviter le surstock comme le sous-stock.

— Réduire les erreurs de saisie et les oublis que l'être humain peut commettre [28].

Le machine learning a joué un rôle extrêmement important dans l'amélioration de la gestion de stock puisqu'il a pu prévoir les demandes futures et optimiser les niveaux de stock, ce qui réduit considérablement les ruptures et les surstocks. L'intelligence artificielle ne se limite pas à cela ; elle est aussi capable de traiter les informations contenues dans des images et des vidéos, ce qui offre une automatisation plus complète des opérations de la gestion de stock. C'est là que la vision par ordinateur intervient.

1.9.2 Vision par ordinateur et détection d'objets

La vision par ordinateur est un domaine de l'intelligence artificielle qui permet aux machines d'interpréter le contenu de données visuelles telles que les images et les vidéos. Elle s'appuie sur des techniques comme le traitement des images, la reconnaissance des formes et l'analyse spatiale. La vision par ordinateur est appliquée dans plusieurs domaines, notamment l'industrie, l'imagerie médicale et l'agriculture [29]. Dans la gestion

des stocks, elle est utilisée pour l'identification des articles, la lecture des codes-barres, le suivi et la surveillance des stocks, la détection de la position de la marchandise et le comptage automatique des produits [30].

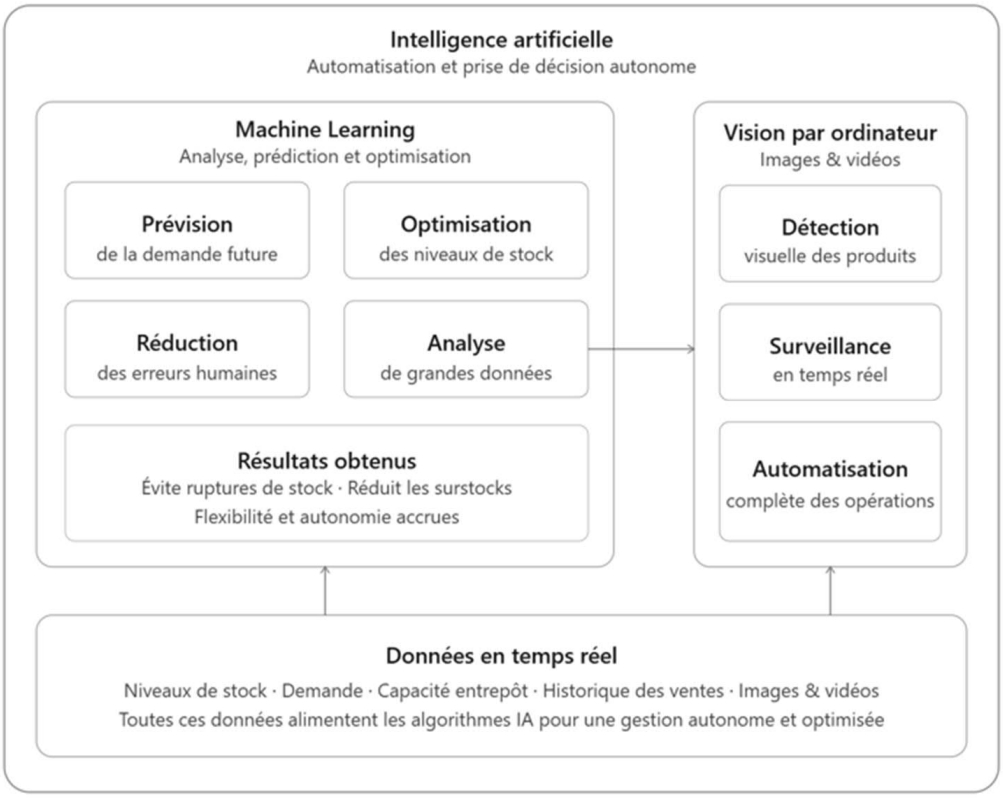


FIGURE 1.7 – Architecture globale de l'IA appliquée à la gestion des stocks.

1.9.3 Travaux existants

La détection automatique de produits par vision par ordinateur a fait l'objet de nombreux travaux de recherche au fil des années, avec une évolution notable des approches méthodologiques adoptées.

Moorthy et Behera (2015) ont utilisé l'algorithme SURF (*Speeded-Up Robust Features*), un algorithme qui détermine la quantité de produits disponibles dans le magasin ainsi que leurs positions. Il extrait des caractéristiques dans une image issue de la caméra — coins, bords et textures — puis les combine pour localiser les produits dans les rayons [31].

Dans la même période, Varol et al. (2015) ont proposé une approche basée sur le traitement d'images pour assurer le suivi des produits en rayon [32]. Avec l'essor du deep learning, les méthodes ont évolué vers des architectures plus complexes et plus performantes.

Higa et Iwamoto (2019) ont développé une méthode pour la surveillance des rayons d'un magasin. Ce système effectue trois étapes : d'abord, il détecte et identifie les

changements des rayons (par exemple l'absence d'un produit) en comparant l'image actuelle avec une image de référence. Ensuite, à l'aide d'un réseau de neurones convolutifs basé sur CaffeNet et CIFAR-10, le système peut donner quatre classifications de l'image selon le type de changement : produit retiré (diminution de stock) ou produit réapprovisionné (augmentation de stock), produit légèrement déplacé sans être retiré, ou fausse détection due à un changement d'éclairage. Enfin, à partir des résultats de la classification, le système met à jour l'inventaire en précisant le nombre exact de produits restants. Ce système peut aussi générer des alertes lorsqu'un produit atteint un seuil critique [33].

Sinha et al. (2022) ont exploité des architectures ResNet-18 et Faster-RCNN pour la détection et la reconnaissance de produits [34], tandis que Yücel et Ünsalan (2022) ont proposé une approche combinant deep learning et vision par ordinateur pour la surveillance des rayons [35].

Saqlain et al. (2022) ont proposé une méthode appelée Hyb-SMPC (*Hybrid approach for Shelf Monitoring and Planogram Compliance*). Ce système accomplit deux tâches principales : la première consiste à détecter et localiser les produits présents dans les rayons à l'aide de trois modèles de détection YOLO V4, YOLO V5 et YOLOR, ainsi qu'à compter les produits et repérer ceux dont le stock est sur le point de s'épuiser. La deuxième tâche consiste à vérifier si les produits sont correctement placés [36].

Notre travail s'inscrit dans cette dynamique en l'adaptant au contexte pharmaceutique algérien. Nous comparons trois modèles récents (YOLOv8, YOLO26 et RT-DETR) pour la détection automatique de médicaments à partir d'images réelles collectées en pharmacie, dans l'objectif d'un suivi de stock en temps réel.

1.10 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art de la gestion des stocks. Nous avons défini ce qu'est un stock, les types de stock et ses fonctions. Ensuite, nous avons vu une définition de la gestion des stocks et ses différentes fonctions. Nous avons également présenté les méthodes classiques de gestion des stocks ainsi que l'évolution de ces méthodes, passant de la digitalisation vers le machine learning et à la vision par ordinateur. Enfin, nous avons présenté quelques travaux existants en lien avec notre travail.

Dans le chapitre suivant, nous allons voir en détail l'intelligence artificielle et la vision par ordinateur, qui constituent le cœur de ce projet.

Chapitre 2

Intelligence artificielle et détection d'objets

2.1 Introduction

Depuis longtemps, la gestion des stocks a reposé sur une logique simple : des équipes entières dédiées à compter, vérifier les produits et corriger les erreurs. Cependant, ces tâches sont épuisantes et les erreurs humaines sont inévitables ; parfois les ruptures de stocks sont découvertes trop tard. Ce modèle a montré ses limites bien avant que la technologie ne propose une alternative sérieuse.

Aujourd'hui, les entrepôts, les usines et les grandes surfaces connaissent un changement profond, passant des simples codes-barres aux capteurs connectés, jusqu'à l'arrivée de la vision par ordinateur, une technologie capable d'analyser des images en temps réel pour surveiller les stocks automatiquement.

La vision par ordinateur, c'est avant tout la capacité d'une machine à voir et surtout à comprendre leur contenu, exactement comme le fait l'œil humain. Des caméras sont positionnées stratégiquement dans un espace de stockage pour capturer les images en continu. Ces images sont analysées en temps réel par des algorithmes capables de reconnaître un produit, d'estimer une quantité ; de plus, le système ne dort pas, ne se fatigue pas, ne rate aucun changement de stock. Il surveille en permanence et génère des alertes.

Ce chapitre présente les bases théoriques nécessaires à la compréhension de notre système. Nous commençons par expliquer ce qu'est l'intelligence artificielle et comment elle fonctionne, en passant par le Machine Learning, le Deep Learning et les réseaux de neurones convolutifs. Nous abordons ensuite la vision par ordinateur et plus particulièrement la détection d'objets, qui est le cœur de notre travail. Enfin, nous présentons les modèles utilisés YOLOv8, YOLO26 et RT-DETR en détaillant leur fonctionnement et leur architecture. Ce chapitre se termine par une conclusion.

2.2 Intelligence artificielle

Depuis toujours, les humains cherchent à comprendre comment ils pensent, raisonnent et prennent des décisions. L'intelligence artificielle part de ce même questionnement, non pas seulement pour comprendre l'intelligence, mais avec l'objectif de la reproduire dans une machine. Autrement dit, une machine intelligente doit être capable d'analyser son environnement et d'agir plus efficacement selon la situation.

L'IA n'est pas née du jour au lendemain. Les philosophes ont d'abord cherché à raisonner correctement. Les mathématiciens ont traduit ces idées en équations grâce à la logique et aux probabilités. Les neuroscientifiques ont essayé de comprendre les secrets du cerveau. Les psychologues ont observé comment les humains traitent l'information. Enfin, les informaticiens ont construit des algorithmes et des machines capables d'imiter le cerveau humain.

En 1956, lors d'un séminaire à Dartmouth, John McCarthy a officiellement donné le nom « intelligence artificielle » à ce domaine. L'idée de départ était simple : comprendre le fonctionnement de l'intelligence humaine et comment la reproduire dans une machine.

Aujourd'hui, l'IA est largement développée et ses résultats sont visibles dans tous les domaines. Des voitures roulent seules sur des millions de kilomètres sans accidents graves. Des outils traduisent plus de 100 langues. Des médecins sont accompagnés par des algorithmes qui détectent certaines maladies. Dans le monde du jeu, l'IA a même battu les meilleurs joueurs humains aux échecs.

Le Machine Learning et le Deep Learning ne sont pas des technologies à part : ils sont le cœur même de l'IA [1].

2.3 Machine Learning

Le Machine Learning est un sous-domaine de l'intelligence artificielle. Il donne la capacité à une machine d'apprendre de manière autonome à partir de données qu'on lui présente. Plus elle traite d'exemples, plus elle affine ses paramètres jusqu'à être capable de faire des prédictions sur des situations qu'elle n'a jamais rencontrées. Cela permet d'éviter de programmer chaque règle manuellement, surtout pour des situations complexes comme la reconnaissance des visages ou la compréhension de textes.

Le Machine Learning repose sur deux piliers essentiels :

— **Les données d'apprentissage :** ce sont des exemples fournis à la machine pour qu'elle puisse apprendre. Plus ces données sont nombreuses et diversifiées, meilleurs seront les résultats.

— **L'algorithme :** c'est lui qui analyse les données et construit un modèle capable de faire des prédictions. Un algorithme inadapté donnera de mauvais résultats, même si les données sont de bonne qualité [2].

2.3.1 Types d'apprentissage

Il n'y a pas une seule façon pour une machine d'apprendre ; selon le problème à résoudre et la nature des données, on distingue quatre types d'apprentissage :

— **L'apprentissage supervisé :** le plus utilisé. On fournit à la machine des exemples avec leurs réponses correctes ; elle s'entraîne sur ces exemples jusqu'à être capable de répondre à des situations inédites. C'est le type d'apprentissage retenu dans ce projet, car le système doit reconnaître les boîtes de médicaments une par une et être capable d'identifier leurs noms exacts.

— **L'apprentissage non supervisé :** la machine n'a pas de réponses fournies dans les exemples ; elle explore les données seule, compare les caractéristiques et cherche des ressemblances. Particulièrement utile quand les données sont volumineuses ou trop complexes pour être annotées manuellement.

— **L'apprentissage semi-supervisé :** compromis entre les deux précédents. Seule une partie des données est annotée ; la machine s'appuie sur les données étiquetées pour se guider et utilise les données non étiquetées pour affiner ses réponses.

— **L'apprentissage par renforcement :** la machine prend une décision ; si elle est correcte, elle reçoit une récompense ; si elle se trompe, elle reçoit une pénalité. Au fil des essais, elle apprend à prendre de meilleures décisions.

Cependant, le Machine Learning est insuffisant pour faire de la détection automatique des objets car il nécessite une intervention humaine lors de l'extraction des caractéristiques de l'image (couleur, texture, contours, etc.), les modèles sont très sensibles au changement d'éclairage, d'angles de vue ou d'arrière-plans, et ils ne peuvent pas détecter les objets partiellement cachés ou trop rapprochés. Par ailleurs, ces modèles tombent souvent dans le problème de surapprentissage où ils commencent à mémoriser les données d'entraînement au lieu de vraiment apprendre [3].

2.4 Deep Learning

Le Deep Learning est une branche du Machine Learning. Il utilise un réseau de neurones artificiels organisés en plusieurs couches successives, où chaque couche reçoit les données de la précédente et en tire des informations de plus en plus complexes. C'est ce travail progressif qui rend le Deep Learning puissant : chaque couche ajoutée lui permet de comprendre davantage.

2.4.1 Fonctionnement des réseaux de neurones

Un réseau de neurones artificiel est organisé en trois types de couches (voir Figure 2.1) :

— **La couche d'entrée :** reçoit les données brutes, comme les pixels d'une image.

— **Les couches cachées :** transforment les données couche après couche. Les premières couches détectent des détails simples (contours, textures) ; les suivantes

combinent ces caractéristiques pour identifier des parties de plus en plus complexes, jusqu'à reconnaître l'objet entier.

— **La couche de sortie :** produit le résultat final sous forme de prédiction.

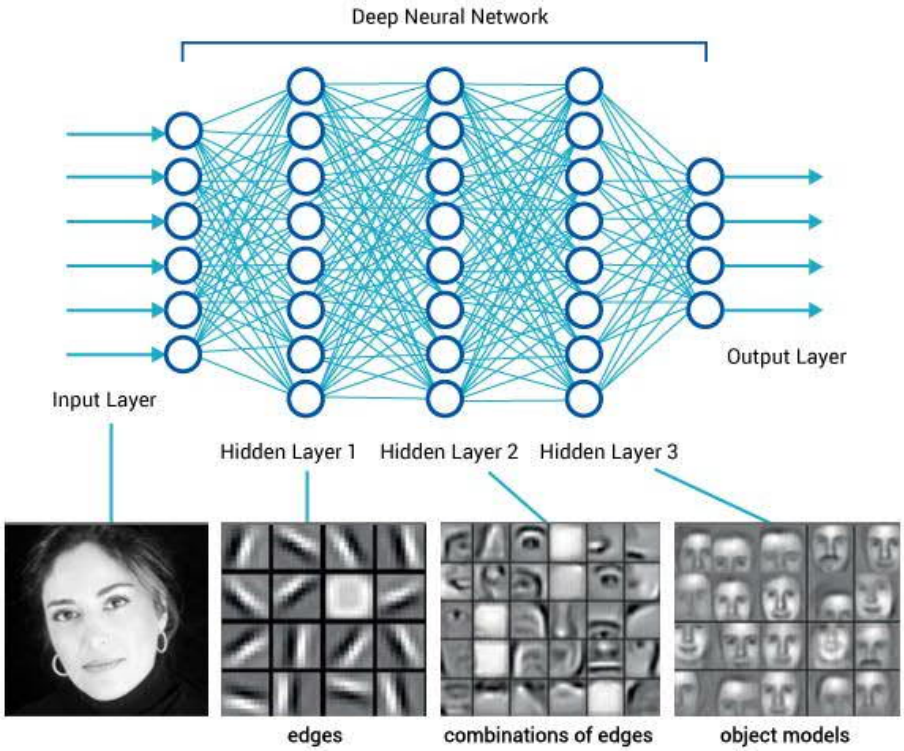


FIGURE 2.1 – Trois types de couches d'un réseau de neurones [4].

Le réseau compare ensuite sa prédiction avec la bonne réponse. S'il s'est trompé, il ajuste ses paramètres internes pour réduire l'erreur. Cette opération s'appelle la *rétropropagation* ; elle est répétée plusieurs fois jusqu'à ce que les prédictions deviennent précises [5].

2.4.2 Réseaux de neurones convolutifs (CNN)

Le Deep Learning est particulièrement efficace pour les données sous forme d'images, grâce au CNN (*Convolutional Neural Network*), un type de réseau de neurones conçu spécialement pour ce traitement.

Le CNN se base sur deux mécanismes clés :

— **Les connexions locales :** dans un réseau classique, chaque neurone est connecté à toute l'image. Dans un CNN, chaque neurone analyse uniquement une petite zone, ce qui réduit le nombre de calculs et rend le réseau beaucoup plus rapide, plus précis et plus facile à entraîner, car il se concentre sur les petits détails de l'image.

Avant d'aborder le deuxième mécanisme, expliquons la notion de filtre. Un filtre est un outil mathématique qui se présente sous forme d'une petite grille de nombres ; il se déplace sur toute l'image et effectue des calculs pour identifier si un motif précis est présent ou non (voir Figure 2.2).

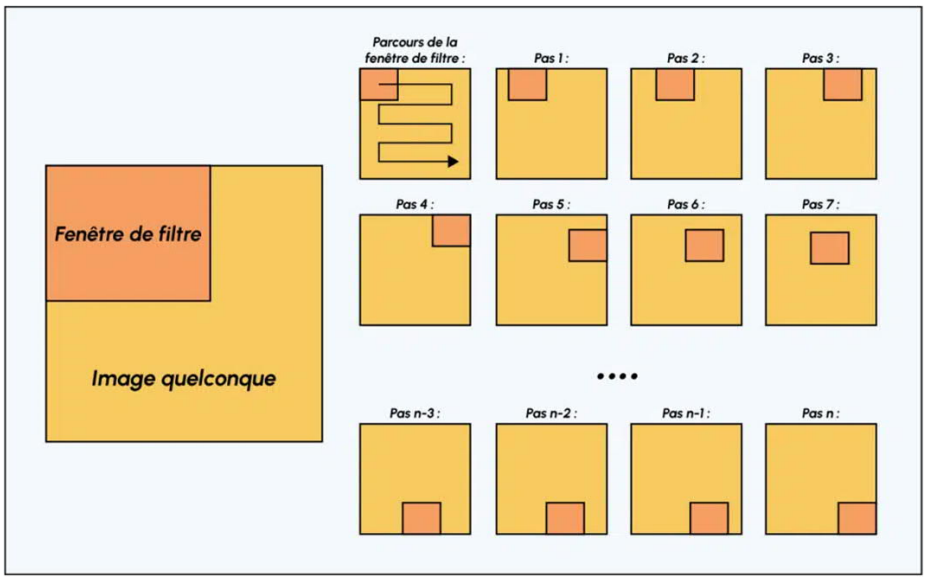


FIGURE 2.2 – Illustration d'un filtre dans un CNN.

— **Les poids partagés :** un même filtre passe sur toute l'image. Si le réseau apprend à détecter un bord à un endroit, il sera capable de le reconnaître automatiquement dans n'importe quelle partie de l'image.

Le CNN fonctionne en deux types de couches qui interviennent l'une après l'autre (voir Figure 2.3) :

— **Les couches de convolution :** servent à repérer les caractéristiques comme les contours, les textures et certaines formes.

— **Les couches de *pooling* :** permettent de simplifier les données en réduisant leur taille et gardent seulement les informations les plus importantes. Cela facilite le traitement et rend le réseau plus robuste face aux petits changements de position des objets dans l'image.

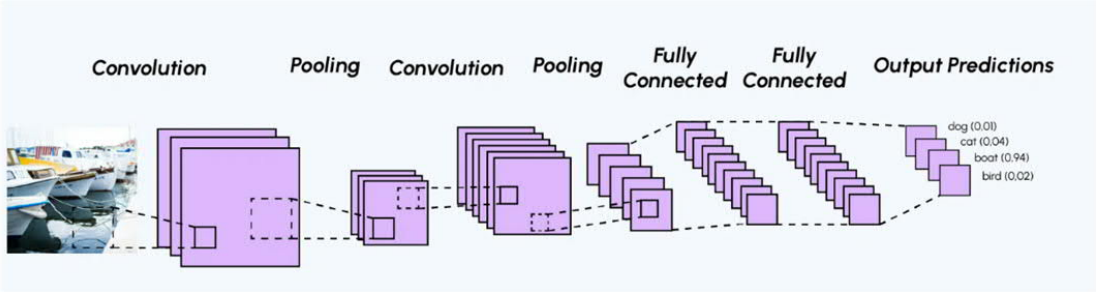


FIGURE 2.3 – Deux types de couches d'un CNN.

Comme le montre la figure 2.4, une image de 4×4 pixels est divisée en petites zones de 2×2 pixels, et pour chaque zone on ne retient que la valeur maximale (*max pooling*). On passe ainsi d'une image 4×4 à une image 2×2, deux fois plus petite, tout en conservant l'essentiel. Il existe deux autres types de pooling : le *min pooling*, où l'on garde la valeur la plus petite de chaque zone, et le *average pooling*, qui calcule la moyenne des valeurs de la zone.

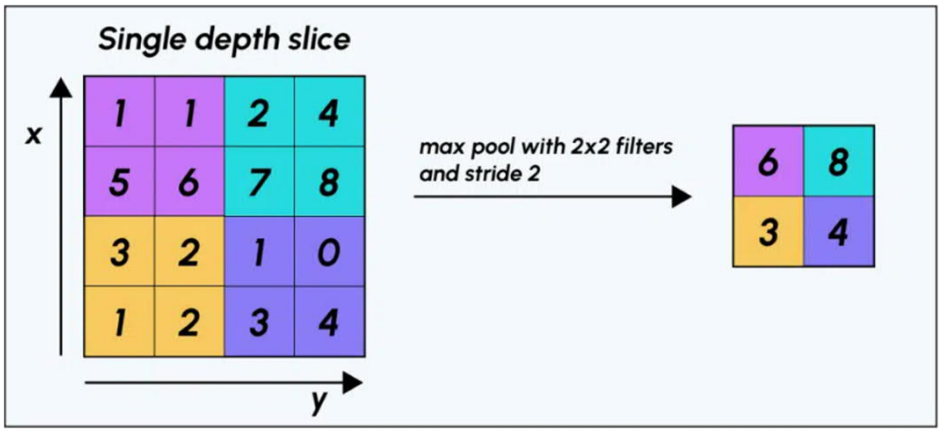


FIGURE 2.4 – Exemple de max pooling (4×4 → 2×2).

Grâce à ces deux mécanismes, le CNN est devenu la meilleure approche pour le traitement des images. En 2012, lors de la compétition ImageNet, un CNN a prouvé son efficacité en divisant par deux le taux d'erreur des meilleures méthodes précédentes. Mais entraîner un CNN à partir de zéro demande beaucoup de temps, de données et de puissance de calcul. C'est là qu'intervient le transfer learning [6].

2.4.3 Transfer Learning

L'apprentissage par transfert est une technique qui permet à un modèle de réutiliser ce qu'il a déjà appris sur une tâche précédente pour l'appliquer à une nouvelle tâche,

au lieu de repartir de zéro. Il est particulièrement utile quand on dispose de peu de données.

L'apprentissage par transfert se divise en trois grandes catégories :

— **Le transfert inductif :** la tâche cible est différente de la tâche source et le modèle s'appuie sur ce qu'il a appris pour améliorer ses résultats.

— **Le transfert transductif :** les deux tâches sont identiques, mais les domaines d'application sont différents.

— **Le transfert non supervisé :** le cas le plus rare et le plus complexe, car aucune donnée n'est étiquetée.

Pour transférer les connaissances acquises, il existe quatre approches :

— **Le transfert d'instances :** réutiliser certaines données de la source en leur attribuant des poids différents.

— **Le transfert de représentations :** chercher une représentation commune aux données des deux domaines.

— **Le transfert de paramètres :** partager les paramètres appris par le modèle source comme point de départ pour le nouveau modèle.

— **Le transfert de connaissances relationnelles :** transférer les liens qui existent entre les données du domaine source vers le domaine cible [7].

2.5 Vision par ordinateur

La vision par ordinateur est un domaine de l'intelligence artificielle qui cherche à donner aux machines la capacité de voir et de comprendre le contenu des images ou des vidéos, puis d'en extraire les informations utiles. Son objectif est de reproduire les capacités visuelles humaines pour des tâches comme la reconnaissance des visages, la détection d'objets ou la lecture de texte.

La vision par ordinateur repose sur plusieurs étapes qui travaillent ensemble pour permettre à une machine de comprendre une image (voir Figure 2.5) [9].

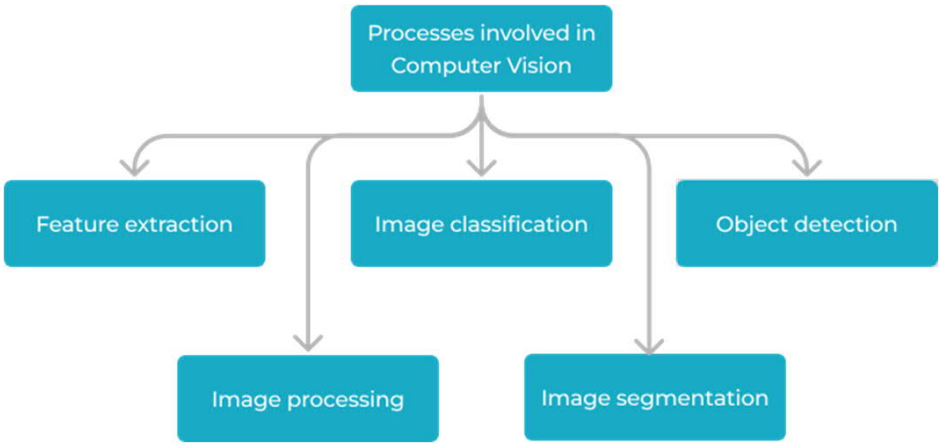


FIGURE 2.5 – Étapes de la vision par ordinateur [8].

2.5.1 Traitement d'image

Le traitement des images est une étape importante dans la vision par ordinateur, car les images ne sont pas toujours parfaites. Elles peuvent contenir plusieurs problèmes :

— **Le bruit :** petits points parasites qui apparaissent, surtout si l'image a été prise dans l'obscurité. On applique un filtre de lissage pour diminuer ces perturbations.

— **Les distorsions :** déformations causées par la lentille de la caméra ou l'angle de prise de vue, corrigées par des transformations géométriques.

— **La luminosité :** une image trop sombre ou trop claire rend l'analyse difficile ; on ajuste la luminosité et le contraste.

— **Le redimensionnement :** les modèles de Deep Learning n'acceptent que des images d'une taille précise.

Toutes ces opérations sont fondamentales : une image mal préparée donnera de mauvais résultats, peu importe la robustesse de l'algorithme utilisé.

2.5.2 Extraction des caractéristiques

Une fois l'image nettoyée, on extrait les informations clés : contours, textures, couleurs et formes. Plutôt que d'analyser l'image pixel par pixel, l'algorithme ne retient que l'essentiel, passant d'une image brute à des données plus simples et faciles à comprendre.

2.5.3 Classification d'images

C'est la tâche la plus simple : le système regarde une image dans sa globalité et lui attribue une seule étiquette. Pour l'entraînement, on lui fournit une base de données étiquetée, ou on peut regrouper les images de la même classe sans annoter.

2.5.4 Détection d'objets

La détection va plus loin que la classification : elle prédit plusieurs objets dans l'image et indique également où ils se trouvent, en les encadrant par un rectangle (boîte englobante).

2.5.5 Segmentation d'images

La segmentation est l'étape la plus précise. Elle ne se contente pas d'encadrer approximativement un objet comme le fait la détection ; elle trace les contours exacts pixel par pixel. Elle est utilisée dans des domaines où la connaissance des dimensions exactes d'un objet est indispensable, comme l'imagerie médicale pour identifier les tumeurs ou la robotique pour localiser précisément les objets à saisir [10]. La figure 2.6 illustre bien la différence entre les trois approches.

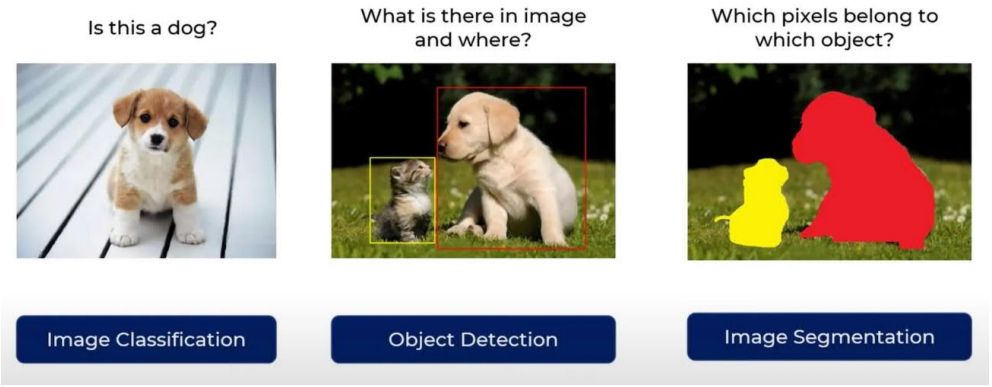


FIGURE 2.6 – Classification, détection et segmentation d'images.

2.6 Détection d'objets

La détection d'objets est la tâche la plus populaire dans la vision par ordinateur car elle offre un bon compromis entre précision et rapidité, ce qui la rend adaptée à une grande variété d'applications réelles.

Elle repose sur des algorithmes capables de repérer les objets avec grande précision. Le processus se déroule en trois étapes : le modèle analyse l'image et en extrait les caractéristiques importantes ; il les utilise ensuite pour repérer les zones susceptibles de contenir des objets ; enfin, il localise chaque objet en dessinant une boîte englobante autour.

On la retrouve aujourd'hui dans des domaines très variés : les voitures autonomes (repérer obstacles, piétons, panneaux), le domaine médical (détecter des anomalies sur une radio), l'industrie (surveiller les stocks et la qualité des produits), et les systèmes de sécurité (surveiller les espaces en temps réel) [11].

2.6.1 Approches de détection

La détection d'objets repose sur deux grandes approches :

— **Détection en une seule passe (*one-stage*) :** le modèle analyse toute l'image en un seul passage à travers le réseau de neurones. Il divise l'image en une grille et prédit simultanément les boîtes englobantes et les probabilités de classe pour tous les objets présents, ce qui le rend très rapide et idéal pour la surveillance en temps réel.

— **Détection en deux étapes (*two-stage*) :** le modèle commence par repérer les zones susceptibles de contenir des objets, puis les analyse une par une. Cette approche est plus lente mais plus utile et sûre lorsqu'on a besoin de plus de précision peu importe la vitesse [8].

2.6.2 Étapes du processus de détection

Le processus de détection d'objets se déroule en quatre étapes principales :

Étape 1 — Acquisition de l'image

Une caméra capture l'image. Sa qualité est déterminante : une image floue ou mal capturée donnera de mauvais résultats. La caméra doit être adaptée à l'environnement (classique, infrarouge pour la vision nocturne, ou spéciale pour conditions difficiles).

Étape 2 — Prétraitement

Les images sont redimensionnées à une taille standard et les valeurs des pixels sont normalisées dans une plage uniforme, rendant le modèle plus robuste face aux variations du monde réel.

Étape 3 — Algorithme de détection

Cœur du processus. Les détecteurs en deux étapes sont précis mais lents ; les détecteurs en une seule passe analysent toute l'image en un seul passage, ce qui les rend beaucoup plus rapides.

Étape 4 — Suppression des non-maximum (NMS)

Après la détection, le modèle peut produire plusieurs boîtes pour un même objet. La NMS supprime les boîtes peu fiables et ne conserve que la plus confiante.

2.6.3 Boîtes englobantes

Une boîte englobante est un rectangle tracé autour d'un objet dans une image, défini par un ensemble de coordonnées. Elle permet au modèle d'identifier la nature de l'objet et de le localiser précisément.

Ces coordonnées peuvent être exprimées selon trois formats différents :

— **XYXY :** format simple et direct. On note les coordonnées des deux coins opposés du rectangle (coin supérieur gauche et coin inférieur droit).

— **XYWH :** on indique le centre de l'objet (x, y), puis sa largeur W et sa hauteur H. C'est le format le plus utilisé lors de l'entraînement des modèles.

— **Coordonnées normalisées :** toutes les valeurs sont converties dans un intervalle entre 0 et 1, permettant au modèle de fonctionner quelle que soit la taille de l'image [12].

2.7 Modèles de détection d'objets utilisés

Parmi les nombreux modèles de détection d'objets existants, notre choix s'est porté sur YOLO et RT-DETR.

2.7.1 Présentation de YOLO

YOLO, abréviation de *You Only Look Once* (Tu ne regardes qu'une seule fois), est un modèle de détection d'objets en temps réel. Il a été introduit en 2015 par Joseph Redmon et Ali Farhadi à l'Université de Washington. Son principe repose sur une idée simple : traiter une image en une seule passe pour détecter tous les objets qu'elle contient.

Avant son apparition, les modèles comme Faster R-CNN analysaient une image en deux étapes distinctes : d'abord identifier des zones susceptibles de contenir un objet, puis les classifier une par une. Cette méthode offrait de bons résultats en termes de précision, mais elle était trop lente pour des situations qui demandent de la rapidité. YOLO a changé cette logique en réunissant ces deux étapes en une seule opération, ce qui lui permet de traiter plus de 30 images par seconde sur un GPU standard.

La version originale YOLOv1 a d'abord été pré-entraînée sur ImageNet, un dataset qui regroupe plus d'un million d'images appartenant à 1000 catégories différentes, afin d'apprendre à reconnaître les formes, les textures, les couleurs, etc. Elle a ensuite été affinée sur PASCAL VOC pour apprendre à localiser avec précision les objets dans une image.

À partir de YOLOv3, le dataset COCO (*Common Objects in Context*) est devenu la référence principale. Il contient plus de 330 000 images du quotidien annotées avec 80 catégories d'objets. Par rapport à PASCAL VOC qui ne couvrait que 20 classes, COCO est bien plus diversifié, ce qui permet aux modèles d'être plus performants dans des situations réelles.

Aujourd'hui, YOLO est l'un des algorithmes de détection d'objets les plus répandus, aussi bien dans la recherche que dans l'industrie, avec des applications qui vont de la conduite autonome à la surveillance vidéo, en passant par la robotique et l'imagerie

médicale.

Dans ce travail, nous avons opté pour YOLOv8 car c'est une version stable, bien documentée et largement utilisée. Nous avons également intégré YOLO26, la toute dernière version, pour voir concrètement ce qu'elle apporte de nouveau et la comparer à YOLOv8 [13].

2.7.2 Architecture de YOLOv8

L'architecture de YOLOv8 se décompose en quatre blocs, chacun ayant un rôle précis (voir Figure 2.7).

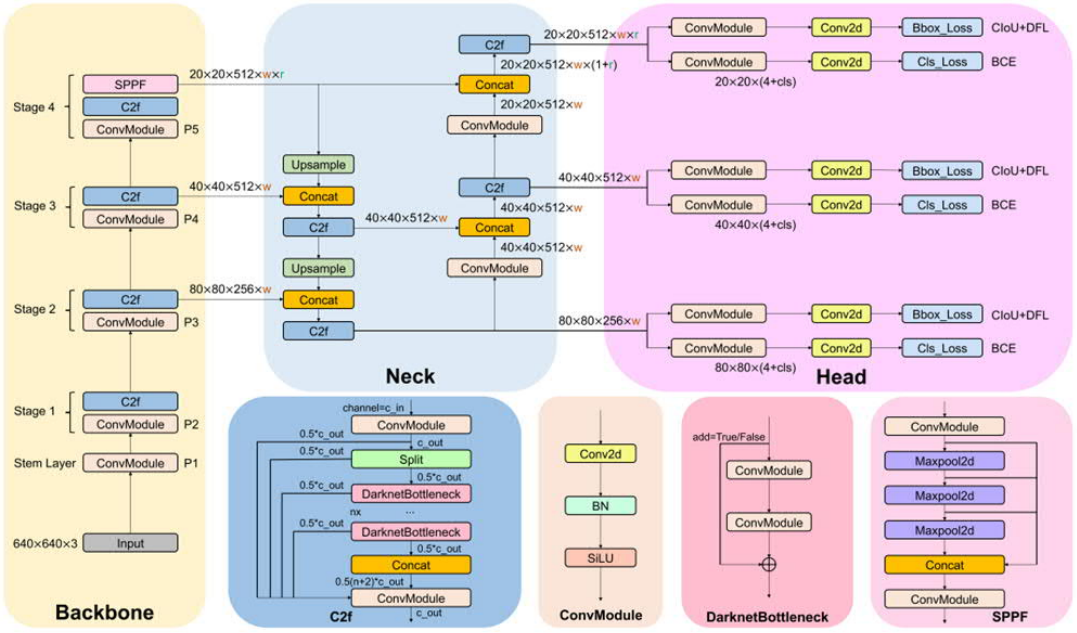


FIGURE 2.7 – Architecture de YOLOv8.

a) Backbone C'est le premier bloc du modèle. Il reçoit l'image en entrée au format 640×640×3 pixels et la fait passer par quatre stages successifs, produisant des versions de l'image à différentes résolutions : P1, P2, P3, P4 et P5. Les premiers niveaux gardent une haute résolution pour capter les petits détails ; les derniers voient l'image de manière plus globale.

Chaque stage utilise deux modules :

— **ConvModule :** enchaîne trois opérations : convolution (repérage des formes et textures), normalisation par lots (maintien des valeurs dans une plage stable) et activation SiLU (capacité d'apprendre des patterns complexes).

— **Module C2f :** l'information entre dans ConvModule qui la compresse, puis est divisée en deux parties. La première est mise de côté ; la deuxième passe par plusieurs

DarknetBottleneck. Les deux sorties sont ensuite réunies par un Concat, puis fusionnées par ConvModule. Ces opérations permettent de conserver les informations brutes tout en les enrichissant.

Au dernier stage, le module SPPF applique trois max-poolings successifs de taille 5×5, réduisant l'image en ne gardant que les valeurs les plus importantes dans chaque zone.

b) Neck Le Neck reçoit trois versions issues du Backbone : P3, P4 et P5. P5 (couches profondes) comprend bien la nature des objets mais a perdu leur localisation précise ; P3 (couches moins profondes) localise bien les objets mais ne connaît pas encore leur nature.

— **Le FPN :** remonte les informations de P5 vers P3 par *upsampling*.

— **Le PAN :** fait le chemin inverse, de P3 vers P5, pour récupérer la précision spatiale.

Cette combinaison produit trois cartes de résolutions 80×80, 40×40 et 20×20, transmises ensuite au Head.

c) Head Le Head produit les détections sur trois niveaux de résolution : 20×20 pour les grands objets, 40×40 pour les objets moyens et 80×80 pour les petits objets.

Pour chaque niveau, deux branches travaillent en parallèle :

— **Branche Bbox :** prédit les coordonnées de la boîte englobante en utilisant la fonction de perte CIoU + DFL.

— **Branche Cls :** prédit la classe de l'objet détecté en utilisant la fonction de perte BCE.

Remarque : les valeurs 20×20, 40×40 et 80×80 représentent le nombre de cellules dans lesquelles l'image est découpée, non des pixels. Pour une image de 640×640 découpée en 20×20, chaque zone couvre 32×32 pixels.

Une fois les détections produites par le Head, il arrive souvent que plusieurs boîtes se retrouvent autour du même objet. Pour nettoyer ce résultat, YOLOv8 fait appel à la NMS après la détection : parmi toutes les boîtes qui se chevauchent, elle ne garde que la plus confiante et élimine les autres.

d) Loss Ce bloc n'intervient que pendant la phase d'entraînement. Il mesure l'erreur entre la prédiction du modèle et la réponse exacte, à l'aide de deux fonctions : CIoU + DFL pour la précision de localisation et BCE pour la précision de classification. Plus l'erreur est grande, plus le modèle ajuste ses paramètres. C'est grâce à ce mécanisme de correction continue que le modèle apprend progressivement à détecter les objets avec précision.

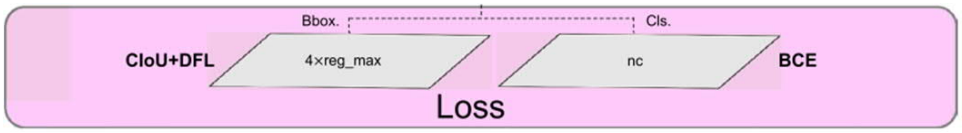


FIGURE 2.8 – Fonctions de perte de YOLOv8 [14].

Le tableau 2.1 présente les cinq variantes de YOLOv8, classées de la plus légère à la plus lourde, et leurs performances mesurées sur le dataset COCO. Le paramètre mAP indique la précision du modèle ; params représente la taille du modèle ; FLOPs indique le nombre de calculs pour analyser une image. Dans ce travail, la variante YOLOv8m a été retenue pour son bon équilibre entre précision et légèreté.

TABLE 2.1 – Les cinq variantes de YOLOv8 [15].

Modèle	**Taille (pixels)**	**mAP$_{50\text{-}95}$**	**Paramètres (M)**	**FLOPs (G)**
YOLOv8n	640	37,3	3,2	8,7
YOLOv8s	640	44,9	11,2	28,6
YOLOv8m	640	50,2	25,9	78,9
YOLOv8l	640	52,9	43,7	165,2
YOLOv8x	640	53,9	68,2	257,8

2.7.3 Architecture de YOLO26

YOLO26 partage la même structure que YOLOv8, avec ses quatre blocs Backbone, Neck, Head et Loss.

La différence principale se situe au niveau du bloc Loss : au lieu d'utiliser la fonction CIoU + DFL, YOLO26 utilise CIoU + ProgLoss + STAL, une fonction de perte plus performante, notamment sur les petits objets. YOLO26 introduit également un nouvel optimiseur d'entraînement appelé MuSGD, qui permet une convergence plus rapide. Ces changements font de YOLO26 un modèle jusqu'à 43% plus rapide sur CPU que YOLOv8, tout en étant plus précis [16].

2.7.4 RT-DETR (Real-Time Detection Transformer)

RT-DETR (*Real-Time Detection Transformer*) est un modèle de détection d'objets développé par Baidu en 2023. Ce qui le distingue, c'est qu'il est l'un des premiers détecteurs en temps réel à reposer sur l'architecture Transformer plutôt que sur des réseaux de neurones convolutifs classiques. De plus, il n'a pas besoin de la NMS en post-traitement, ce qui simplifie et accélère le processus de détection.

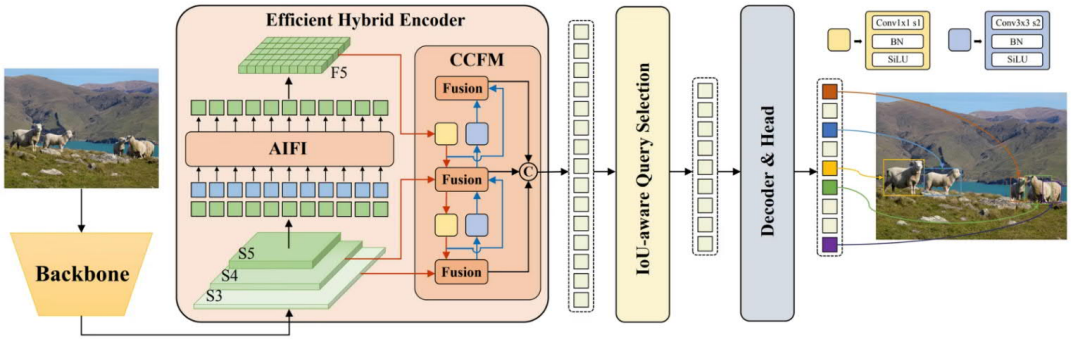


FIGURE 2.9 – Architecture de RT-DETR.

a) Backbone Le Backbone analyse l'image et extrait les informations visuelles à trois niveaux de profondeur : S3, S4 et S5. Les premiers niveaux retiennent les détails fins et la localisation précise, tandis que le niveau S5 voit l'image de manière plus globale mais perd en précision spatiale.

b) Efficient Hybrid Encoder C'est le cœur du modèle. Il reçoit S3, S4 et S5 et les traite en deux temps :

— **Module AIFI :** analyse uniquement S5 en utilisant le mécanisme d'attention des Transformers, permettant au modèle de regarder l'image dans sa globalité et de comprendre les relations entre différentes zones.

— **Module CCFM :** prend ensuite tous les niveaux S3, S4 et S5 et les fusionne progressivement via des opérations de Fusion. S3 apporte des informations sur la localisation exacte des objets, tandis que S5 apporte la nature des objets. En combinant les deux, le modèle obtient une représentation complète qui lui permet de reconnaître les objets et de les localiser avec précision quelle que soit leur taille.

c) IoU-aware Query Selection Ce module fait le tri parmi toutes les prédictions produites par l'encodeur, en se basant sur le score IoU (*Intersection over Union*) qui mesure le chevauchement entre une boîte prédite et un objet réel. Plus ce score est élevé, plus la prédiction a de chances d'être correcte. Cette opération permet au décodeur de travailler sur des prédictions déjà bien ciblées.

d) Decoder & Head C'est l'étape finale. Le décodeur reçoit les prédictions sélectionnées et les affine progressivement à travers plusieurs couches successives. À chaque couche, il ajuste les coordonnées des boîtes et les scores de confiance pour se rapprocher de plus en plus de la réalité.

RT-DETR offre la possibilité d'ajuster sa vitesse d'inférence en modifiant le nombre de couches du décodeur, sans avoir besoin de réentraîner le modèle. Concrètement,

moins il y a de couches, plus le modèle est rapide mais légèrement moins précis, et inversement. Cette flexibilité est très utile pour adapter le modèle à différentes contraintes matérielles sans repartir de zéro.

Il adopte également une approche sans ancres : au lieu de partir d'une liste de rectangles prédéfinis et de chercher les objets à l'intérieur, il repère directement les objets dans l'image et trace les boîtes autour d'eux.

Le modèle est disponible en deux versions présentées dans le tableau 2.2.

TABLE 2.2 – Deux versions de RT-DETR.

Type du modèle	**Poids pré-entraînés**	**Tâches supportées**
RT-DETR Large	rtdetr-l.pt	Détection d'objets
RT-DETR Extra-Large	rtdetr-x.pt	Détection d'objets

2.8 Conclusion

Ce chapitre a posé les bases théoriques sur lesquelles repose notre travail. Nous avons vu comment l'intelligence artificielle, à travers le Machine Learning et le Deep Learning, a progressivement donné aux machines la capacité de comprendre et d'analyser des images. Nous avons ensuite présenté la détection d'objets comme une tâche centrale de la vision par ordinateur, capable de localiser et d'identifier plusieurs objets simultanément dans une image en temps réel.

Les trois modèles retenus pour le système ont été présentés. YOLOv8 représente une base solide, stable et bien documentée. YOLO26, la version la plus récente, apporte des améliorations en termes de vitesse et de précision. RT-DETR, quant à lui, propose une approche différente en s'appuyant sur l'architecture Transformer plutôt que sur les réseaux convolutifs classiques.

Le chapitre suivant présentera la mise en œuvre pratique de ces modèles, depuis la constitution de la base de données jusqu'aux résultats obtenus.

Chapitre 3

Modèle d'apprentissage entraîné pour la supervision du stock

Introduction

Dans le cadre de ce projet de master portant sur la gestion de stock pharmaceutique, l'objectif est de développer un système capable de reconnaître les médicaments dans une pharmacie et de détecter toutes les boîtes présentes afin de surveiller le stock en temps réel. Ce chapitre constitue le cœur du travail expérimental. Pour mener à bien ce travail, nous avons collaboré avec la Pharmacie MERZOUK, située à Tlemcen, qui nous a autorisés à collecter les données nécessaires à la constitution de notre base de données.

Dans un premier temps, nous expliquerons comment nous avons conçu une base de données propre contenant des médicaments disponibles en Algérie, en décrivant les étapes de collecte et de préparation des images. Une fois la base de données prête, nous présenterons la phase d'entraînement. Nous avons choisi de tester trois modèles de détection d'objets — YOLOv8, YOLO26 et RT-DETR — afin de les comparer et de retenir celui qui offre le meilleur compromis entre précision et vitesse de détection.

3.1 Constitution de la base de données

3.1.1 Collecte des images

Pour entraîner les modèles, il est nécessaire de disposer d'un nombre suffisant d'images par classe tout en évitant une trop grande dispersion entre les classes. La collecte a été effectuée à la Pharmacie MERZOUK, une officine située à Tlemcen, qui nous a accordé l'autorisation de photographier son stock. La pharmacie dispose de plusieurs armoires de rangement comportant chacune de nombreuses étagères sur lesquelles les médicaments sont organisés par catégorie. Nous avons concentré notre collecte sur deux armoires représentatives, contenant une grande variété de boîtes de médicaments courants disponibles en Algérie. Afin de maintenir un équilibre entre le nombre de classes et la quantité d'images disponibles, nous avons limité notre dataset à 105 médicaments distincts, pour un total initial de 772 photographies.

Pour introduire de la diversité dans les données, les prises de vue ont été volontairement variées : photos rapprochées pour capturer les petits détails, vues d'ensemble

de l'armoire, angles droits et inclinés, différentes conditions d'éclairage (Figure 3.1).



FIGURE 3.1 – Collecte des images

3.1.2 Annotation des images

L'annotation est une étape indispensable pour toute tâche de détection d'objets. Elle consiste à encadrer manuellement chaque objet d'intérêt par un rectangle de délimitation (*bounding box*) et à lui attribuer l'étiquette correspondant à sa classe. Sans cette étape, le modèle ne dispose d'aucune information sur ce qu'il doit apprendre.

L'annotation a été réalisée avec la plateforme Roboflow, qui propose des outils intuitifs pour dessiner les boîtes englobantes et gérer les classes. Chaque boîte de médicament présente dans une image a été encadrée et associée au nom du médicament correspondant. Ce processus a été répété sur l'ensemble des images de la base de données.

3.1.3 Division et redimensionnement du dataset

À l'issue de l'annotation, Roboflow affiche un récapitulatif des classes et du nombre d'annotations par classe, puis permet de diviser automatiquement les images en trois ensembles. Nous avons retenu la répartition classique en Deep Learning :

— 80% pour l'entraînement,

— 10% pour la validation,

— 10% pour le test.

Par ailleurs, les modèles YOLO exigent que toutes les images aient une taille uniforme. Nous avons donc redimensionné l'ensemble des images à 640 × 640 pixels, en utilisant l'option *Fit (black edges)* de Roboflow. Les espaces vides générés par le redimensionnement sont comblés par des bordures noires (valeur de pixel = 0), ce qui permet au modèle d'identifier et d'ignorer facilement ces zones non informatives.

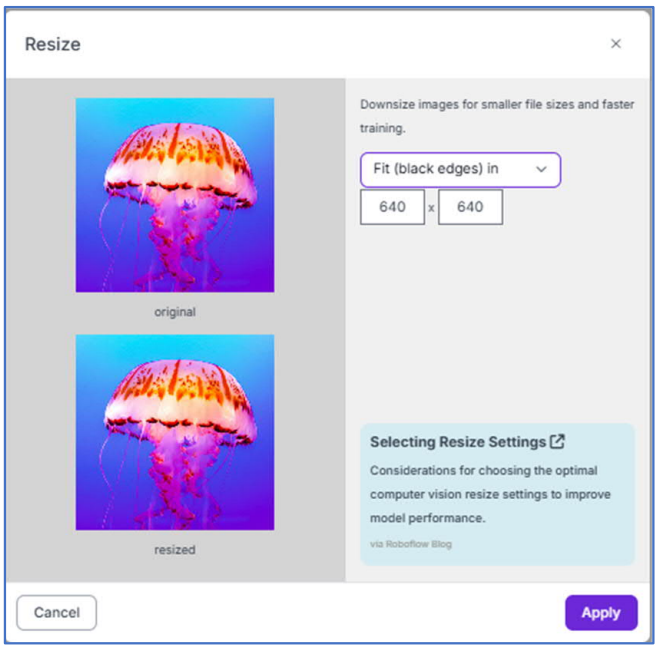


FIGURE 3.2 – Redimensionnement de l'ensemble des images

3.1.4 Augmentation des données

L'augmentation des données est une technique essentielle lorsque la taille du dataset est limitée. Elle consiste à générer de nouvelles images à partir des images existantes en leur appliquant des transformations contrôlées, ce qui permet d'enrichir la diversité des données et d'améliorer la robustesse du modèle face aux variations réelles.

Les transformations appliquées via Roboflow sont décrites dans le tableau 3.1.

TABLE 3.1 – Les transformations appliquées via Roboflow

Transformation	**Description**
Flip	Retournement horizontal et/ou vertical de l'image.
Rotation 90°	Rotation de l'image dans les quatre sens cardinaux.
Crop	Recadrage aléatoire d'une portion de l'image.
Rotation	Inclinaison légère entre -15ř et $+15$ř.
Shear	Déformation géométrique légère de l'image.
Hue	Modification de la teinte des couleurs.
Saturation	Variation de l'intensité des couleurs.
Brightness	Augmentation ou diminution de la luminosité.
Exposure	Variation de l'exposition lumineuse.
Noise	Ajout de bruit gaussien pour simuler une qualité d'image dégradée.

Après l'application de ces transformations, la base de données finale contient 2008 images au total (Figure 3.3), réparties comme suit :

— 1854 images pour l'entraînement,

— 78 images pour la validation,

— 76 images pour le test.

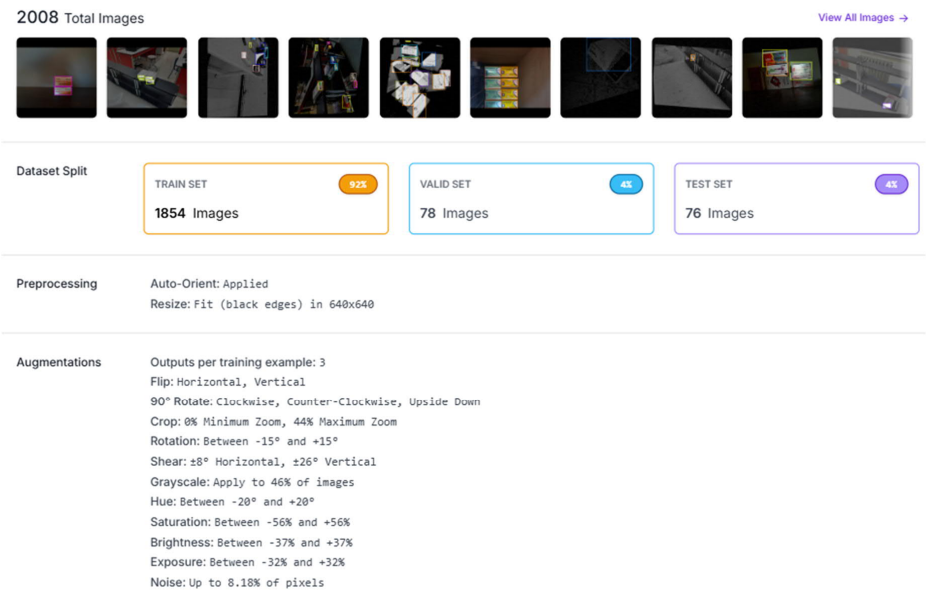


FIGURE 3.3 – La base de données finale

3.2 Entraînement des modèles

3.2.1 Approches de vision par ordinateur

L'entraînement est l'étape durant laquelle un modèle d'intelligence artificielle apprend à reconnaître et localiser les objets à partir d'un ensemble d'images annoté. L'entraînement par le Deep Learning permet au modèle d'extraire les caractéristiques visuelles importantes comme les formes, les contours ou les textures. Ce processus repose sur des réseaux de neurones qui analysent les images couche par couche, en améliorant progressivement leurs poids à chaque *epoch*, jusqu'à ce que les performances du modèle se stabilisent et n'évoluent plus [45].

Il existe plusieurs approches en vision par ordinateur, chacune adaptée à un besoin spécifique. Les plus couramment utilisées sont :

— **La classification d'images** : C'est la tâche la plus simple ; elle consiste à analyser toute l'image et à lui attribuer une seule étiquette [46].

— **La détection d'objet** : Elle consiste à identifier et localiser les objets grâce à des boîtes englobantes dans une image ou un flux vidéo [47].

— **La segmentation d'instance** : C'est une technique plus avancée que la détection d'objet ; elle ne s'arrête pas à la détection et la localisation des objets mais permet aussi de segmenter chaque objet détecté du reste de l'image. Cette approche est particulièrement utile dans des cas où la forme exacte de l'objet présente une information très importante et nécessaire [48].

Dans ce projet, l'objectif est de détecter et d'identifier toutes les boîtes de médicaments présentes dans le champ de vision de la caméra. Une image peut contenir plusieurs boîtes simultanément et la forme exacte des objets n'est pas une information nécessaire. La détection d'objets s'impose donc comme l'approche la mieux adaptée à notre cas d'usage.

3.2.2 Environnement d'entraînement

Les entraînements ont été réalisés sur Google Colab, un environnement de développement en ligne qui permet d'écrire et d'exécuter du code Python depuis un navigateur avec un accès gratuit à des GPU. Nous avons sélectionné le GPU Tesla T4 (16 Go de VRAM), suffisant pour entraîner des modèles de détection d'objets avec une taille d'image de 640×640 et un *batch size* de 16 [43].

3.2.3 Architecture YOLO

YOLO (*You Only Look Once*) est une famille de modèles de détection d'objets en temps réel, fondée sur les réseaux de neurones convolutifs (CNN). Contrairement aux

approches deux étapes qui proposent d'abord des régions candidates avant de les classifier, YOLO traite la détection comme un problème de régression unique : l'image est analysée en un seul passage, produisant simultanément les boîtes englobantes (position du centre, largeur, hauteur), les scores de confiance et les probabilités de classe [42].

Cette approche *one-stage* confère à YOLO une vitesse d'inférence très élevée, ce qui est particulièrement adapté aux applications temps réel comme notre système de surveillance de stock.

Les deux modèles YOLO utilisés — YOLOv8 et YOLO26 — sont des modèles pré-entraînés sur le dataset COCO (*Common Objects in Context*, 330000 images, 80 classes d'objets courants). Nous appliquons ensuite un *fine-tuning* (affinage) sur notre propre base de données pharmaceutique pour adapter les modèles à notre domaine spécifique.

Pour entraîner les modèles YOLO, la base de données doit respecter une structure bien précise. Roboflow offre la possibilité de la télécharger dans un format adapté à YOLOv8.

3.3 Métriques d'évaluation

Avant de présenter les résultats de chaque modèle, nous définissons les métriques utilisées pour évaluer leurs performances dans le tableau 3.2.

TABLE 3.2 – Métriques d'évaluation

Métrique	**Définition**
Précision (P)	Part des détections effectuées par le modèle qui sont réellement correctes. Une précision élevée signifie peu de faux positifs.
Rappel (R)	Part des objets réels que le modèle a réussi à détecter dans les images. Un rappel élevé signifie peu d'objets manqués.
mAP@50	*Mean Average Precision* avec un seuil de chevauchement (IoU) de 50% entre la boîte prédite et la boîte réelle. Métrique la plus couramment utilisée.
mAP@50-95	mAP calculée sur plusieurs seuils IoU de 50% à 95%. Évalue à la fois la détection et la précision de localisation des boîtes englobantes.
Vitesse d'inférence	Temps moyen (en ms) nécessaire au modèle pour analyser une image et produire ses détections.

3.4 Résultats de YOLOv8

3.4.1 Paramètres d'entraînement

Les principaux paramètres de cet entraînement sont les suivants :

— **Nombre d'époques** : Une époque correspond à un passage complet de toutes les images d'entraînement dans le modèle. Nous en avons fixé 200 pour laisser le temps au modèle d'apprendre correctement.

— **Patience** : La patience est un mécanisme d'arrêt anticipé. Si le modèle ne progresse pas pendant 50 epochs, l'entraînement s'arrête tout seul.

— **Taille des images** : Toutes les images ont été redimensionnées à 640×640 pixels, car les boîtes de médicaments occupent l'ensemble de l'image, y compris les bords. Réduire davantage risquerait de couper ou d'écraser certains médicaments.

— **Batch size** : 16, ce qui correspond au nombre d'images traitées simultanément.

— **Optimiseur** : L'optimiseur est l'algorithme qui ajuste les poids internes du modèle pour réduire ses erreurs. Nous avons choisi AdamW, une version améliorée du célèbre optimiseur Adam. Le taux d'apprentissage, c'est la taille des pas que fait le modèle pour se corriger, s'il est trop grand, le modèle risque de dépasser la bonne réponse et de ne jamais s'arrêter dessus, et si le pas est trop petit, il avance tellement lentement qu'on mettrait des heures à obtenir un bon résultat. Donc la valeur 0,001 est un bon compromis, et c'est celle recommandée pour AdamW.

— **Cos_lr** : activé (True), cela permet d'adapter le taux d'apprentissage en suivant une courbe cosinus, c'est-à-dire qu'on commence vite, puis on ralentit doucement vers la fin pour affiner les poids.

— **Augmentation des données** : activée (True). YOLO génère automatiquement des variations aléatoires comme par exemple la rotation, le changement de luminosité des images à chaque époque, ce qui améliore l'apprentissage.

— **Pré-entraînement** : nous avons utilisé les poids de YOLOv8 déjà entraînés sur COCO, un dataset qui contient plus de 330 000 images et 80 classes d'objets courants, puis nous avons continué l'entraînement sur notre propre base de données de médicaments. Cette étape s'appelle le fine tuning Les principaux paramètres de cet entraînement sont les suivants :

— **Nombre d'époques** : Une époque correspond à un passage complet de toutes les images d'entraînement dans le modèle. Nous en avons fixé 200 pour laisser le temps au modèle d'apprendre correctement.

— **Patience** : La patience est un mécanisme d'arrêt anticipé. Si le modèle ne progresse pas pendant 50 epochs, l'entraînement s'arrête tout seul.

— **Taille des images** : Toutes les images ont été redimensionnées à 640×640 pixels, car les boîtes de médicaments occupent l'ensemble de l'image, y compris les bords. Réduire davantage risquerait de couper ou d'écraser certains médicaments.

— **Batch size** : 16, ce qui correspond au nombre d'images traitées simultanément.

— **Optimiseur** : L'optimiseur est l'algorithme qui ajuste les poids internes du modèle pour réduire ses erreurs. Nous avons choisi AdamW, une version améliorée du célèbre optimiseur Adam. Le taux d'apprentissage, c'est la taille des pas que fait le modèle pour se corriger, s'il est trop grand, le modèle risque de dépasser la bonne réponse et de ne jamais s'arrêter dessus, et si le pas est trop petit, il avance tellement lentement qu'on mettrait des heures à obtenir un bon résultat. Donc la valeur 0,001 est un bon compromis, et c'est celle recommandée pour AdamW.

— **Cos_lr** : activé (True), cela permet d'adapter le taux d'apprentissage en suivant une courbe cosinus, c'est-à-dire qu'on commence vite, puis on ralentit doucement vers la fin pour affiner les poids.

— **Augmentation des données** : activée (True). YOLO génère automatiquement des variations aléatoires comme par exemple la rotation, le changement de luminosité des images à chaque époque, ce qui améliore l'apprentissage.

— **Pré-entraînement** : nous avons utilisé les poids de YOLOv8 déjà entraînés sur COCO, un dataset qui contient plus de 330000 images et 80 classes d'objets courants, puis nous avons continué l'entraînement sur notre propre base de données de médicaments. Cette étape s'appelle le fine tuning.

3.4.2 Résultats globaux

Les métriques globales obtenues à l'issue de l'entraînement de YOLOv8 sont présentées dans le tableau 3.3.

TABLE 3.3 – Métriques globales de YOLOv8

Métrique	**Valeur obtenue**
Précision (P)	0,796 (79,6%)
Rappel (R)	0,764 (76,4%)
mAP@50	0,816 (81,6%)
mAP@50-95	0,521 (52,1%)
Vitesse d'inférence	23,5 ms / image

On remarque que le mAP50 global est de 0,816 et le mAP50-95 est de 0,521 : le modèle détecte correctement 81,6% des objets toutes classes confondues, mais la performance chute à 52,1% lorsqu'on exige une localisation plus précise des boîtes

(entre 50 et 95 % de chevauchement). Cela signifie que le modèle détecte bien les objets mais les localise parfois de façon approximative.

Environ 79% des détections effectuées sont correctes et le modèle retrouve 77% des objets réels présents dans les images, soit un taux de manqués de 23%.

3.4.3 Analyse des courbes d'entraînement

Pendant l'entraînement, les valeurs suivantes sont affichées à chaque époque et représentées sous forme de courbes pour visualiser l'évolution des pertes et des métriques au fil des époques. Trois courbes de perte sont tracées :

— **Box loss** : mesure l'écart entre les boîtes englobantes prédites et les boîtes réelles.

— **Cls loss** : mesure les erreurs de classification de chaque objet détecté.

— **Dfl loss** : mesure la précision des dimensions des boîtes englobantes.

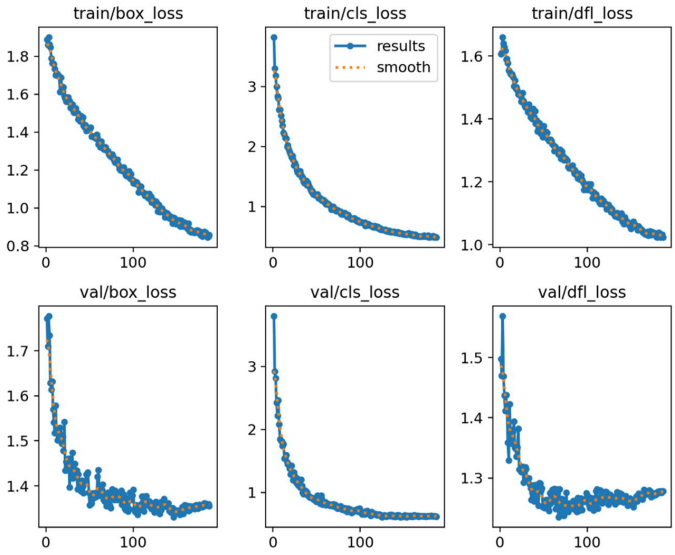


FIGURE 3.4 – Courbes de perte et de métriques de YOLOv8

Les trois courbes présentent une décroissance rapide durant les 50 premières époques, suivie d'une stabilisation progressive : le modèle a bien convergé sans signe de surapprentissage. Les courbes de métriques (précision, rappel, mAP) montrent une progression régulière jusqu'à la stabilisation.

Après chaque epochs, le modèle calcule les métriques de précision, rappel, le mAP50 et le mAP50-95, puis à la fin de l'entrainement il dessine les courbes d'évolution de ces métriques, cela permet de visualiser les performances durant tout le processus d'entrainement.

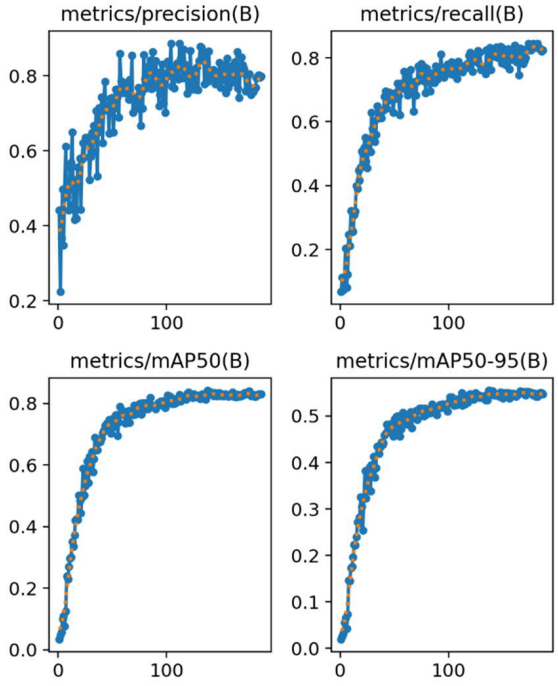


FIGURE 3.5 – Prédictions du modèle YOLOv8 sur les images de validation

La figure est générée à la fin de l'entrainement. Elle montre les prédictions du modèle sur les images de validation. Le modèle YOLOV8 entoure bien les boites, il a pu détecter un grand nombre de médicaments dans chaque image et même les score de confiance sont autour de 0.8. Mais il confond entre les médicaments qui se ressemblent.



FIGURE 3.6 – Prédictions du modèle sur les images de validation

3.4.4 Matrice de confusion

La matrice de confusion de YOLOv8 donne de bons résultats sur les 10 classes : chaque médicament est correctement identifié, comme présenté dans le tableau 3.4.

TABLE 3.4 – Détections correctes par classe – YOLOv8

Médicament	**Détections correctes**
Dolyc	46
Gripex	30
Paralgan	17
Panadol	17
Efferalgan	16
Vitamine C	12
Multi vitamine	11
Xydol 200	7
Panadol rouge	0
Ibuprofène 600	0

Toutes les cases hors diagonale affichent zéro, confirmant que YOLOv8 distingue parfaitement les médicaments les uns des autres sur cet ensemble de validation.

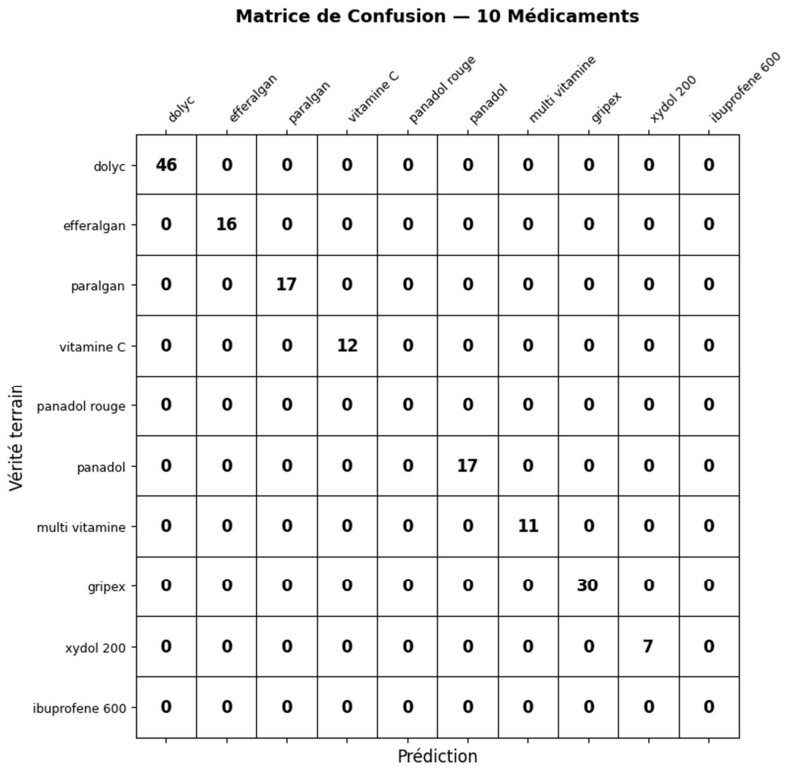


FIGURE 3.7 – Matrice de confusion de YOLOv8

3.5 Résultats de YOLO26

3.5.1 Présentation du modèle

YOLO26 est le modèle de détection d'objets le plus récent de la famille Ultralytics. Il a été conçu spécifiquement pour les appareils embarqués et les systèmes à ressources limitées, en offrant une meilleure efficacité computationnelle par rapport aux versions précédentes. Il partage les mêmes paramètres d'entraînement que YOLOv8 et utilise également le *fine-tuning* à partir des poids pré-entraînés sur COCO.

Pour l'entrainement, les mêmes paramètres sont utilisés que celle de yolov8.

3.5.2 Résultats globaux

Les métriques globales obtenues à l'issue de l'entraînement de YOLO26 sont présentées dans le tableau suivant :

TABLE 3.5 – Métriques globales de YOLO26

Métrique	**Valeur obtenue**
Précision (P)	0,766 (76,6%)
Rappel (R)	0,758 (75,8%)
mAP@50	0,808 (80,8%)
mAP@50-95	0,513 (51,3%)
Vitesse d'inférence	33,1 ms / image

Toutes les métriques sont bonnes et correctes. Le modèle détecte environ 75,8% des médicaments présents dans les images de validation et, parmi ces détections, 76,6% sont correctes. Le mAP50 confirme que le modèle reconnaît correctement la majorité des médicaments sur l'ensemble des classes (80,8%), mais les encadre parfois de manière approximative selon la métrique mAP50-95.

3.5.3 Analyse des courbes et des prédictions

Les courbes de perte (*box*, *cls*, *dfl*) de l'entraînement et de la validation diminuent rapidement lors des premières époques puis se stabilisent à des valeurs faibles, avec un faible écart entre les courbes d'entraînement et de validation — signe d'un bon apprentissage sans sur-apprentissage.

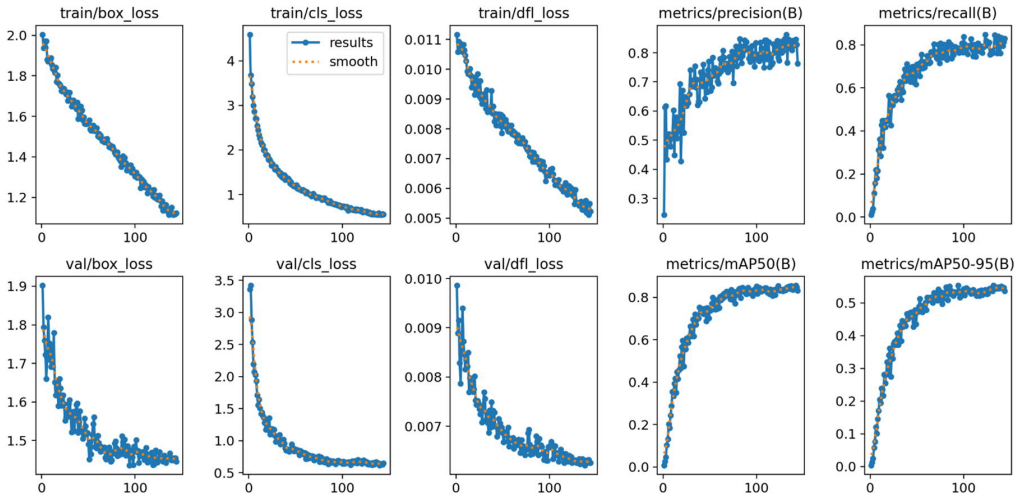


FIGURE 3.8 – Courbes de perte (*box*, *cls*, *dfl*) – YOLO26

Les images de prédiction montrent que YOLO26 est capable de détecter plusieurs médicaments simultanément, même lorsque les boîtes sont très proches les unes des autres. Certains scores de confiance atteignent 0,9, bien que des scores plus faibles soient observés sur des médicaments moins représentés dans le dataset.



FIGURE 3.9 – Prédictions du modèle YOLO26 sur les images de validation

3.5.4 Matrice de confusion

La matrice de confusion de YOLO26 montre de très bons résultats sur les 10 classes, avec toutes les cases hors diagonale à zéro, soit une classification sans aucune confusion entre classes. Les détections correctes par classe sont les suivantes :

TABLE 3.6 – Détections correctes par classe – YOLO26

Médicament	**Détections correctes**
Dolyc	46
Gripex	31
Paralgan	16
Efferalgan	16
Panadol	14
Vitamine C	14
Multi vitamine	12
Xydol 200	5
Panadol rouge	0
Ibuprofène 600	0

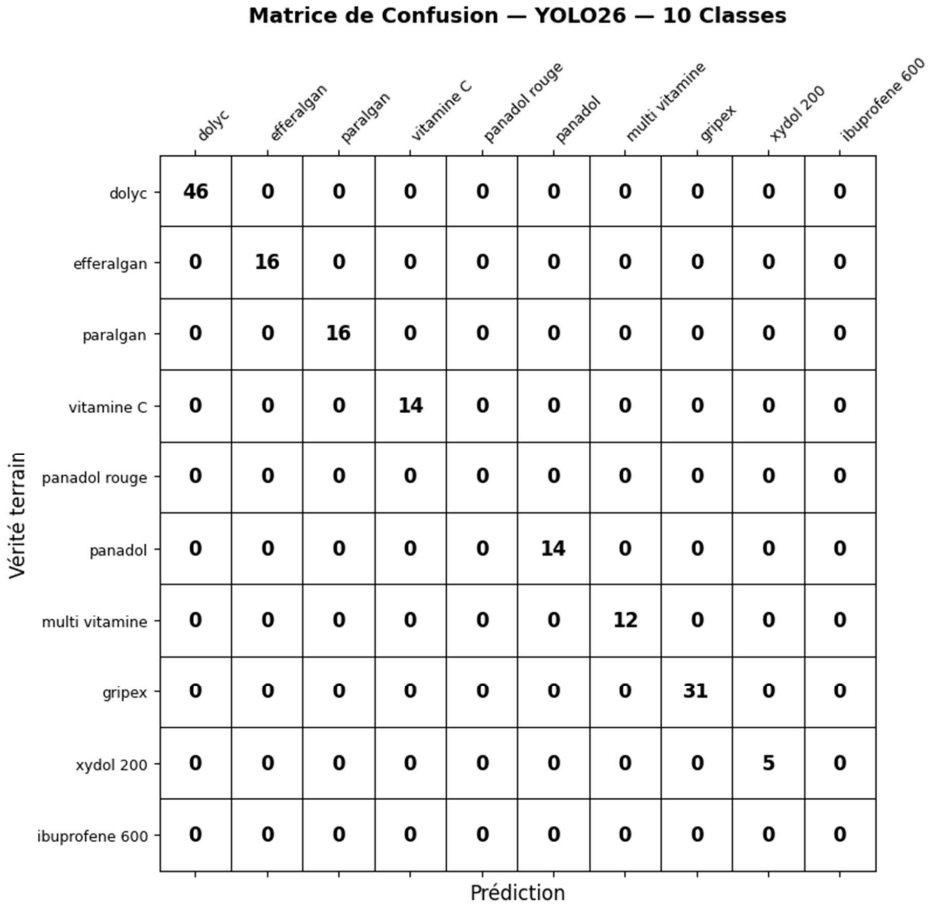


FIGURE 3.10 – Matrice de confusion de YOLO26

YOLO26 affiche un mAP50 légèrement supérieur à YOLOv8, qui reste cependant meilleur sur le mAP50-95 et le recall. La précision est presque similaire. Les courbes sont aussi très similaires. Ils détectent correctement plusieurs médicaments dans les images et les deux ont des scores élevés dans certains médicaments. Le YOLOv8 est légèrement plus rapide que le YOLO26, on remarque ça dans l'inférence qui est le temps que met un modèle pour analyser l'image et détecter les boites de médicaments. On conclusion on peut dire que le modèle YOLO26 est plus précis et performant mais un peu moins rapide que le modèle YOLOV8.

3.6 Résultats de RT-DETR

3.6.1 Présentation du modèle

RT-DETR (Real-Time Detection Transformer) est un modèle de détection d'objets développé par Baidu. Contrairement aux modèles YOLO qui reposent sur des CNN, RT-DETR exploite l'architecture Transformer initialement conçue pour le traitement du langage naturel adaptée à la vision par ordinateur. Cette architecture permet une compréhension globale du contexte de l'image grâce au mécanisme d'attention. Pour

l'entrainement les mêmes paramètres sont utilisés que celle de RT-DETR.

3.6.2 Résultats globaux

Les métriques globales obtenues à l'issue de l'entraînement de RT-DETR sont présentées dans le tableau suivant :

TABLE 3.7 – Métriques globales de RT-DETR

Métrique	**Valeur obtenue**
Précision (P)	0,676 (67,6%) en dessous du seuil de 0.70
Rappel (R)	0,717 (71,7%)
mAP@50	0,718 (71,8%)
mAP@50-95	0,476 (47,6%)
Vitesse d'inférence	43,1 ms / image (le plus lent des trois)

3.6.3 Analyse

Les courbes de perte descendent de manière simultanée pour l'entraînement et la validation, avec un faible écart entre elles, indiquant que le modèle apprend correctement sans mémoriser les données d'entraînement. Cependant, les scores de confiance sur les images de prédiction sont globalement plus bas que pour les modèles YOLO.

La précision en dessous de 0,70 indique qu'environ 32% des détections produites par RT-DETR sont des faux positifs. Ce résultat, combiné à une vitesse d'inférence de 43,1 ms, en fait le moins adapté à notre application temps réel.

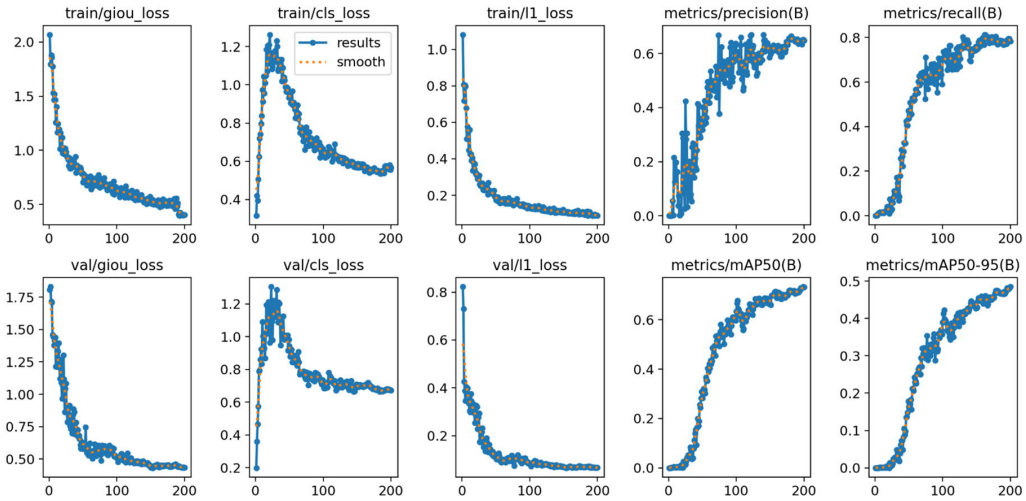


FIGURE 3.11 – Courbes d'entraînement de RT-DETR



FIGURE 3.12 – Prédictions du modèle RT-DETR sur les images de validation

3.6.4 Matrice de confusion

Malgré des métriques globales plus faibles, la matrice de confusion de RT-DETR montre une classification sans erreur : toutes les cases hors diagonale sont nulles. Les détections correctes par classe sont les suivantes :

TABLE 3.8 – Détections correctes par classe – RT-DETR

Médicament	**Détections correctes**
Dolyc	89
Efferalgan	32
Paralgan	20
Panadol	14
Gripex	13
Vitamine C	13
Xydol 200	12
Doliprane 200	12
Ibuprofène 600	12
Xydol 400	11

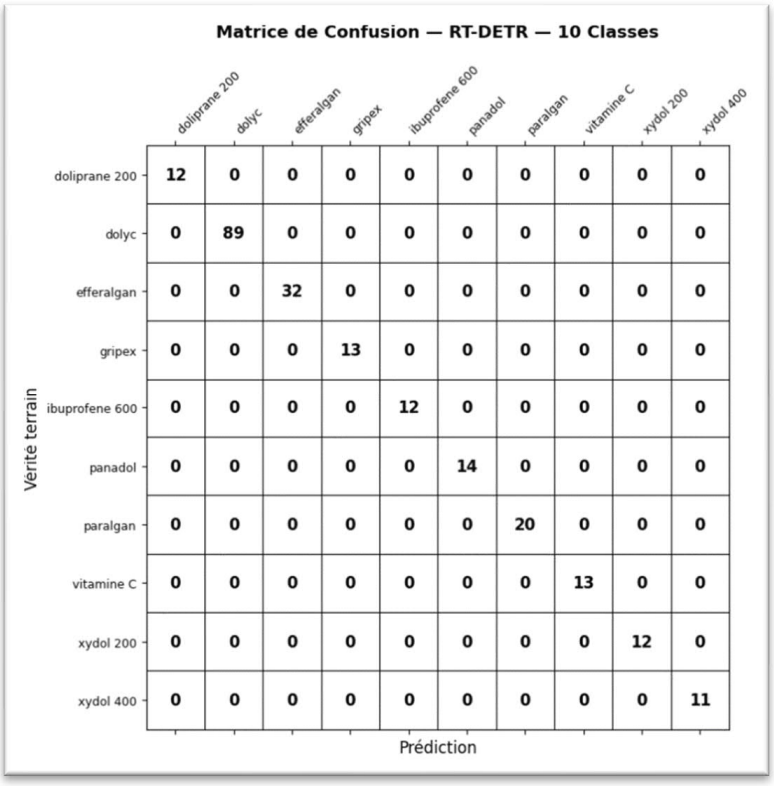


FIGURE 3.13 – Matrice de confusion de RT-DETR

3.7 Comparaison et choix du modèle final

3.7.1 Tableau comparatif

TABLE 3.9 – Comparaison des trois modèles

Métrique	**YOLOv8**	**YOLO26**	**RT-DETR**
mAP@50	0,817	0,808	0,718
mAP@50-95	0,522	0,514	0,476
Précision	0,796	0,766	0,676
Rappel	0,765	0,759	0,717
Vitesse d'inférence	23,5 ms	33,1 ms	43,1 ms
Confusion entre classes	Aucune	Aucune	Aucune

3.7.2 Analyse comparative

En regardant les chiffres, on voit que YOLOv8 domine sur tous les points. Son mAP50 de 0.817 est le plus élevé des trois, ce qui veut dire qu'il détecte correctement environ 82% des médicaments, et ça c'est un bon résultat. Le mAP50-95 à 0.522 montre aussi qu'il localise les boites avec une précision correcte, mieux que les deux autres modèles. En plus, il est le plus rapide avec une inférence de 23.5ms. Le modèle YOLO26 reste quand même très bon car ses métriques sont tres proche que celui de yolov2 et il ne confond aucune classe, ce qui est appréciable. RT-DETR, malgré son architecture avancée basée sur les Transformers, donne les résultats les plus faibles des trois. Sa précision est en dessous de 0.70, ce qui veut dire que presque un tiers de ses détections sont fausses. Il est aussi le plus lent avec 43.1ms d'inférence. Pour une application temps réel en pharmacie, ce modèle n'est pas le meilleur choix dans notre cas. Après avoir comparé les trois modèles, YOLOv8 s'impose comme le meilleur choix pour notre projet. Il offre les meilleures performances globales avec un mAP50 de 0.808, une bonne précision et un rappel satisfaisant.

3.8 Conclusion

Ce chapitre a présenté l'ensemble du pipeline expérimental, depuis la construction de la base de données jusqu'à la sélection du modèle final. La constitution du dataset a représenté le défi principal de ce travail : avec seulement 106 images initiales pour 90 classes, le risque de sous-apprentissage était élevé. La réduction du nombre de classes couplée à l'augmentation des données sur Roboflow a permis d'obtenir un dataset de 276 images suffisamment diversifiées pour garantir un apprentissage efficace. Parmi les

trois modèles évaluésYOLOv8, YOLO26 et RT-DETR, YOLOv8 s'est distingué par ses performances supérieures sur toutes les métriques de précision, sans sacrifier la vitesse d'inférence. Ces modèles sont tous de type one-stage, c'est-à-dire qu'ils produisent leurs détections en un seul passage à travers le réseau, ce qui les rend particulièrement adaptés aux exigences du temps réel. Le modèle YOLOv8 ainsi sélectionné sera intégré dans le projet d'ingénieur au sein d'un système complet de supervision de stock pharmaceutique, en lien avec le module Raspberry Pi, la base de données et le tableau de bord de visualisation.

Conclusion générale

Un médicament manquant au bon moment peut retarder un traitement, aggraver un état de santé, ou dans les cas les plus critiques, coûter la vie à un patient.

Dans le contexte pharmaceutique, la gestion des stocks est très importante, car une mauvaise gestion ou une rupture des médicaments peut affecter la vie d'un patient.

Dans ce mémoire, nous nous intéressons aux techniques de l'intelligence artificielle et de la vision par ordinateur pour la détection des médicaments en temps réel via des caméras, en vue de la mise en place d'un système intelligent de surveillance des stocks dans une pharmacie.

L'objectif est d'évaluer la capacité des modèles de détection d'objets YOLOv8, YOLO26 et RT-DETR à détecter avec précision différentes références de médicaments, afin de construire un modèle capable d'identifier et de suivre les médicaments à partir d'images.

Pour répondre à cet objectif, nous avons constitué une base de données originale à partir d'images réelles collectées dans une pharmacie située à Tlemcen. Cette base de données couvre 106 références de médicaments commercialisés en Algérie, avec 2008 images réparties sur ces 106 classes, bien que l'on note un déséquilibre assez notable : certains médicaments, comme le Dolyc, apparaissent 430 fois, tandis que d'autres, comme l'Aspégic 500 ou le Vitam adultes et enfants, ne dépassent pas 9 annotations.

Nous avons ensuite entraîné et comparé trois modèles : YOLOv8, YOLO26 et RT-DETR. Les résultats montrent que YOLOv8 est le plus performant des trois, avec un mAP50 de 81.6 %, une bonne précision et un bon rappel. Il détecte et identifie correctement la grande majorité des boîtes de médicaments, et ne confond pas les classes entre elles. YOLO26 ne commet pas non plus d'erreurs de classification, mais affiche une précision légèrement inférieure. Ce modèle a été conçu spécialement pour les dispositifs embarqués à puissance limitée ; cependant, dans notre cas, la différence de vitesse avec YOLOv8 est significative. RT-DETR fournit des résultats acceptables, mais reste en dessous des deux modèles YOLO sur la plupart des métriques. Au final, YOLOv8 s'impose comme le meilleur choix pour notre cas d'usage.

Ce travail démontre que la détection d'objets par apprentissage profond constitue une piste sérieuse et peut offrir une très bonne précision pour automatiser la surveillance des stocks pharmaceutiques. Les résultats sont encourageants, même si certaines classes restent moins représentées que d'autres. Rééquilibrer ces classes permettrait de rendre le modèle encore plus fiable.

Ce mémoire n'est qu'une première étape. La suite logique serait d'intégrer ce modèle dans un système complet de gestion des stocks, doté d'une interface affichant en temps

réel les quantités détectées et envoyant des alertes lorsqu'un médicament est sur le point de manquer. C'est précisément ce que nous envisageons de développer dans le cadre du projet d'ingéniorat. Plus largement, cette approche pourrait s'étendre bien au-delà des pharmacies, à tout environnement où le suivi visuel des stocks représente une contrainte quotidienne.

Ce mémoire n'est qu'une première étape. La suite logique de ce travail serait d'intégrer ce modèle dans un système complet de gestion de stock, avec une interface qui affiche en temps réel les quantités détectées et envoie des alertes quand un médicament est sur le point de manquer.

C'est exactement ce qui sera développé dans le cadre du projet d'ingéniorat. Plus largement, cette approche pourrait s'étendre bien au-delà des pharmacies, à n'importe quel environnement où le suivi visuel des stocks est une contrainte quotidienne.

Références bibliographiques

[1] Russell, S., & Norvig, P. (2021). *Intelligence artificielle* (L. Miclet, F. Popineau & C. Cadet, trad. ; 4e éd.). Pearson France.

[2] Azencott, C.-A. (2022). *Introduction au machine learning*. Dunod.

[3] Trigka, M., & Dritsas, E. (2025). A comprehensive survey of machine learning techniques and models for object detection. *Sensors*, 25(1), 214. https://doi.org/10.3390/s25010214

[4] Sachdeva, A. (2017, 6 mars). Deep learning for computer vision for the average person. *Medium*. https://medium.com/diaryofawannapreneur/deep-learning-for-computer-vision-for-the-average-person-861661d8aa61

[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[6] Jérémy Robert, « Convolutional Neural Network : Tout ce qu'il y a à savoir », Liora, 18 février 2026. https://liora.io/convolutional-neural-network

[7] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[8] Jobit. (2024, 8 février). The 5 fundamental processes in computer vision. *Random Walk*. https://randomwalk.ai/blog/the-5-fundamental-processes-in-computer-vision/

[9] Szeliski, R. (2022). *Computer Vision : Algorithms and Applications* (2nd ed.). Springer. https://szeliski.org/Book/

[10] Nikhil. (2023). Image classification vs image segmentation vs object detection. *Kaggle*. https://www.kaggle.com/discussions/getting-started/463662

[11] Rapid Innovation. (2024). Understanding object detection : Techniques, applications, and future trends. https://www.rapidinnovation.io/post/object-detection-in-2024-the-definitive-guide

[12] Ultralytics. (s.d.). Bounding box. *Ultralytics Glossary*. Récupéré le 23 mai 2026, de https://www.ultralytics.com/glossary/bounding-box

[13] Lightly AI. (s.d.). YOLO object detection explained : Models, tools, use cases. *Lightly AI Blog*. Récupéré le 23 mai 2026, de https://www.lightly.ai/blog/yolo

[14] Ju, R.-Y., & Cai, W. (2023). Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm. *Scientific Reports*, 13, 20077. https://doi.org/10.1038/s41598-023-47460-7

[15] Ultralytics. (2023). Explore Ultralytics YOLOv8. Récupéré le 23 mai 2026, de https://docs.ultralytics.com/models/yolov8

[16] Ultralytics. (2026). YOLOv8 vs YOLO26 : The evolution of Ultralytics real-time object detection. Récupéré le 23 mai 2026, de https://docs.ultralytics.com/compare/yolov8-vs-yolo26

[17] Souibaah, A. K. E. , « *Amélioration de performances de la chaine interne chez EVOLAB (Industrie Pharmaceutique)* », Mémoire de fin d'étude pour l'obtention du diplôme de Master en Management Industriel et Logistique, ESSA Tlemcen, 2020/2021.

[18] Essebbar, H. (2018, 22 juin). *Méthodes de Prévision de la Demande*. Scribd. https://www.scribd.com/document/382361495/Prevision-de-La-Demande

[19] Emy, I. (2020, 25 mai). *Coût de passation et de possession de stock*. Scribd. https://www.scribd.com/document/462975907/Le-cout-de-passation-et-de-possession

[20] Mecalux. (s.d.). *Quantité économique de commande (EOQ)*. Mecalux Blog. https://www.mecalux.fr/blog/quantite-economique-commande-eoq

[21] Rodríguez, L. (2025, 13 mars). *Point de commande : levier d'optimisation de la Supply Chain*. Slimstock. https://www.slimstock.com/fr/blog/point-de-commande/

[22] 4Gestion Academy. (2014). *Module : Gestion des stocks – Résumé théorique* [cours PDF, 48 pages]. https://4gestionacademy.com/wp-content/uploads/2014/03/Gestion-de-stocks.pdf

[23] Fastcoo Logistics. (2023, 20 avril). *Gestion traditionnelle des stocks : quels sont les défis et quelles sont les solutions ?* https://fastcoo.com/traditional-inventory-management-what-are-the-challenges-and-what-are-the-solutions/

[24] Oracle. (s.d.). *Qu'est-ce que l'ERP ?* Oracle Afrique. https://www.oracle.com/africa-fr/erp/what-is-erp/

[25] SAP. (s.d.). *What is SAP ERP ?* SAP France. https://www.sap.com/france/products/erp/what-is-sap-erp.html

[26] Dataveo. (s.d.). *À quoi sert le code-barres ?* https://dataveo.com/a-quoi-sert-code-barres.html

[27] Porter, J. (2025, May 8). *The role of machine learning in optimizing inventory management and reducing stockouts*. Stock IQ Technologies. https://stockiqtech.com/blog/machine-learning-optimizing-inventory-management/

[28] Singh, N., & Adhikari, D. (2023). AI in inventory management : Applications, challenges, and opportunities. *International Journal for Research in Applied Science and Engineering Technology*, *11*(11), 2049–2053. https://doi.org/10.22214/ijraset.2023.57010

[29] Lefebvre Dalloz Compétences. (s.d.). *La Vision par ordinateur (computer vision) expliquée*. Formation Lefebvre Dalloz. https://formation.lefebvre-dalloz.fr/dossier/ia-intelligence-artificielle/vision-par-ordinateur

[30] Giner, J., Katic, D., Kovacs, K., Glawar, R., & Sihn, W. (2023). A computer vision based approach to reduce system downtimes in an automated high-rack logistics warehouse. *Procedia CIRP*, *118*, 1078–1083. https://doi.org/10.1016/j.procir.2023.06.185

[31] Moorthy, R., Behera, S., Verma, S., Bhargave, S., & Ramanathan, P. (2015). Applying image processing for detecting on-shelf availability and product positioning in retail stores. In *Proceedings of the Third International Symposium on Women in Computing and Informatics* (pp. 451–457). ACM. https://doi.org/10.1145/2791405.2791533

[32] Varol, G., & Kuzu, R. S. (2015). Toward retail product recognition on grocery shelves. *Proceedings of SPIE*, *9443*, Article 944309. https://doi.org/10.1117/12.2179127

[33] Higa, K., & Iwamoto, K. (2019). Robust shelf monitoring using supervised learning for improving on-shelf availability in retail stores. *Sensors*, *19*(12), 2722. https://doi.org/10.3390/s19122722

[34] Sinha, A., Banerjee, S., & Chattopadhyay, P. (2022). An improved deep learning approach for product recognition on racks in retail stores. arXiv. https://arxiv.org/abs/2202.13081

[35] Yücel, M. E., & Ünsalan, C. (2022). Planogram compliance control via object detection, sequence alignment, and focused iterative search. arXiv. https://arxiv.org/abs/2212.01004

[36] Saqlain, M., Rubab, S., Khan, M. M., Ali, N., & Ali, S. (2022). Hybrid approach for shelf monitoring and planogram compliance (Hyb-SMPC) in retails using deep learning and computer vision. *Mathematical Problems in Engineering*, *2022*, Article 4916818. https://doi.org/10.1155/2022/4916818

[37] Succès Marketing. (s. d.). Qu'est-ce qu'un logiciel ERP ? Quelle est son utilité ? https ://succes-marketing.com/logiciel-erp/

[38] Swiss Life Asset Managers. (s.d.). Industrial and logistics properties : an ideal combination [Photographie]. Swiss Life Asset Managers. https://www.swisslife-am.com/en/home/hub/industrial_logistics_combination.html

[39] Magnific. (s.d.). Lecteur code-barres lit boîte stockage [Illustration vectorielle]. Magnific. https://www.magnific.com/fr/vecteurs-libre/lecteur-code-barres-lit-boite-stockage_3586145.html

[40] Newboot.io. (2024, 31 octobre). IoT & Logistique : Gestion des stocks en temps réel avec des capteurs IoT et la modélisation 3D [Article LinkedIn]. LinkedIn. https://fr.linkedin.com/pulse/iot-logistique-gestion-des-stocks-en-temps-réel-avec-capteurs-fygme

[41] Mecalux Software Solutions. (s.d.). Warehouse Management System - Easy WMS [Vidéo]. YouTube. https://www.youtube.com/watch?v=[ID_de_la_vidéo]

[42] Afifah, V., & Erniwati, S. (2026). YOLOv8 for object detection : A comprehensive review of advances, techniques, and applications. *International Journal of Advanced Computing and Informatics*, 2(1). https://doi.org/10.71129/ijaci.v2i1.pp53-61

[43] Saturn Cloud. (2023). Hardware specifications for Google Colaboratory. https://saturncloud.io/blog/whats-the-hardware-spec-for-google-colaboratory/

[44] Adsul, C. (2023). My YOLO models talking – here's what it's really saying. *Medium*. https://medium.com/@chinmayiadsul/my-yolo-models-talking-here-s-what-it-s-really-saying-85298b256e34

[45] Trigka, M., & Dritsas, E. (2025). A Comprehensive Survey of Machine Learning Techniques and Models for Object Detection. *Sensors*, 25(1), 214. https://doi.org/10.3390/s25010214

[46] Ultralytics. (2024). Image Classification – Ultralytics Docs. https://docs.ultralytics.com/tasks/classify

[47] Ultralytics. (2024). Object Detection – Ultralytics Docs. https://docs.ultralytics.com/tasks/detect

[48] Ultralytics. (2024). Instance Segmentation – Ultralytics Docs. https://docs.ultralytics.com/tasks/segment

Resumé

Dans ce mémoire, nous nous sommes intéressés aux techniques de l'intelligence artificielle et de la vision par ordinateur pour la détection des médicaments en temps réel. En vue de la mise en place d'un système intelligent de surveillance des stocks dans une pharmacie, nous avons évalué la capacité des trois modèles YOLOv8, YOLOv6 et RT-DETR à détecter avec précision différentes références de médicaments. Ces modèles sont entraînés et comparés sur des images réelles collectées dans une pharmacie à Tlemcen. Les résultats obtenus sont encourageants : les modèles testés parviennent à reconnaître et à localiser les boîtes de médicaments avec une bonne précision, sans confondre les références entre elles. YOLOv8 s'impose comme le meilleur compromis entre précision et rapidité, ouvrant la voie à un système de surveillance automatique du stock en temps réel.

Mots-clés : Intelligence artificielle, Vision par ordinateur, Détection de médicaments, Gestion des stocks, YOLOv8, YOLO26, RT-DETR, Pharmacie.

Abstract

In this thesis, we focused on artificial intelligence and computer vision techniques for real-time drug detection. With the aim of implementing an intelligent inventory monitoring system in a pharmacy, we evaluated the ability of three models — YOLOv8, YOLOv6, and RT-DETR — to accurately detect different medication references. These models were trained and compared on real images collected from a pharmacy in Tlemcen. The results obtained are encouraging : the tested models successfully recognize and localize medication boxes with good accuracy, without confusing references with one another. YOLOv8 stands out as the best trade-off between accuracy and speed, paving the way for an automatic real-time stock monitoring system.

Keywords : Artificial intelligence, Computer vision, Medicine détection, Inventory management, YOLOv8, YOLO26, RT-DETR, Pharmacy.

مـلـخص

في هذه المذكرة، اهتممنا بتقنيات الذكاء الاصطناعي والرؤية الحاسوبية من أجل الكشف عن الأدوية في الوقت الفعلي. بهدف تطوير نظام ذكي لمراقبة المخزون في صيدلية، قمنا بتقييم قدرة ثلاثة نماذج — YOLOv8 وYOLOv6 وRT-DETR — على الكشف الدقيق عن مختلف مراجع الأدوية. تم تدريب هذه النماذج ومقارنتها على صور حقيقية جُمعت من صيدلية في تلمسان. النتائج المتحصل عليها مشجعة: إذ تمكنت النماذج المختبَرة من التعرف على علب الأدوية وتحديد مواضعها بدقة جيدة، دون الخلط بين المراجع المختلفة. ويبرز YOLOv8 باعتباره أفضل توازن بين الدقة والسرعة، مما يفتح الطريق نحو نظام آلي لمراقبة المخزون في الوقت الفعلي.

كلمات مفتاحية : الذكاء الاصطناعي، الرؤية الحاسوبية، كشف الأدوية، إدارة المخزون، الصيدلية، RT-DETR، YOLO26، YOLOv8.