

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--TLEMCEEN--



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 : Génie industriel

Spécialité : Management industriel et logistique

Présenté par :

KACEM Hadjer Fatima Zahra
BERKANI Chaimaa Hibat Allah

Thème

**Simulation et Modélisation
Intelligente de la Santé des Abeilles
pour la Prévention des Maladies**

Soutenu publiquement, le 03/07/2024, devant le jury composé de :

M Akkacha BEKKADOUR	MCB	ESSA. Tlemcen	Président
Mme Amina OUHOUD	MCB	ESSA. Tlemcen	Directeur de mémoire
M Mustapha Anwar BRAHAMI	MCA	ESSA. Tlemcen	Directeur de mémoire
M Hichem MEGNAFI	MCA	ESSA. Tlemcen	Co- Directeur de mémoire
M Mohammed BENNEKROUF	MCA	ESSA. Tlemcen	Examineur
M Zoheir KARAOUZENE	MAA	ESSA. Tlemcen	Examineur
M Abdelghafour BENDJILLALI		CAWIT Mansourah. Tlemcen	Partenaire SE

Année universitaire : 2023 /2024

REMERCIEMENT :

Au nom de Dieu le Tout-Puissant, le Clément, le Miséricordieux, qui nous a incités à acquérir le savoir et les sciences, et nous a dotés de toutes les facultés et moyens pour y parvenir. Nous rendons grâce au Tout-Puissant pour la volonté, le courage, la patience et la force qu'il nous a donnés tout au long de ces longues années d'études.

Nous tenons à exprimer nos plus sincères remerciements et notre profond respect à notre encadrante Mme OUHOUD Amina, qui nous a soutenus et fait confiance tout au long de notre projet avec une grande patience. Merci pour vos encouragements dans les moments difficiles et pour être bien plus qu'une professeure, mais un guide exceptionnel. Grâce à votre expérience dans la recherche et l'enseignement, et à vos précieux conseils, nous avons pu découvrir le monde de la recherche scientifique dans le domaine du traitement d'image.

À notre deuxième encadrant, Mr. Mustapha Anwar BRAHAMI, nous vous remercions énormément pour votre patience, votre disponibilité et vos conseils.

À notre co-encadrant, Mr. Hicham MEGNAFI, qui nous a appris que rien n'est impossible, nous vous remercions pour la qualité exceptionnelle de votre encadrement.

À tous les enseignants du département Génie Industriel de l'École Supérieure en Sciences Appliquées de Tlemcen, pour tout le savoir qu'ils ont su nous transmettre durant notre cursus universitaire.

Tout mon respect et mon remerciement vont vers toute personne m'a aidé à réaliser ce travail, particulièrement Mr Ramz eddine Abderrazak ADJIM .

Aux membres du jury, pour l'intérêt qu'ils ont porté à notre projet en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, nous remercions tous ceux qui, de près ou de loin, ont contribué à l'élaboration de ce travail.

DÉDICACE :

Je dédie ce travail, Fruit de nombreuses années d'étude :

À moi-même, pour la persévérance et le dévouement investis tout au long de ce parcours. Grâce à ma détermination et à ma volonté inébranlables, j'ai surmonté des défis ardues et atteint mes objectifs. Bravo à moi-même pour mes accomplissements et ce parcours exceptionnel. Je suis immensément fier de ce que j'ai réalisé.

À la lumière de ma vie, la flamme de mon cœur, la source de ma force, ma joie et mon bonheur, ma chère maman B. DJEBRI, tu incarnes pour moi la beauté de la vie, la source de tendresse et l'exemple de dévouement. Tes prières et tes bénédictions m'ont donné le courage et la volonté de bien mener mes études. Tu es la femme la plus généreuse et courageuse du monde. Te dire je t'aime et merci est bien insuffisant, ma chère maman.

À l'homme de ma vie, mon modèle éternel, mon cher papa Habib KACEM, aucun mot ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eus pour toi. Je tiens à te remercier infiniment pour ton amour, ta compréhension et ta générosité. Ton soutien a été une lumière tout au long de mon parcours.

À mes chers frères Mourad et Mohammed, votre présence constante et votre encouragement indéfectible ont forgé la personne que je suis aujourd'hui. Merci d'être des frères formidables dont je suis si fier.

À ma famille bien-aimée KACEM et DJEBRI, mes oncles, mes tantes et mon grand-père Abdelkarim DJEBRI, merci pour votre soutien sans faille et votre amour constant. Vos encouragements et conseils précieux ont grandement contribué à ma réussite. Votre générosité et votre confiance en mon potentiel ont été d'une aide inestimable tout au long de mon parcours.

Sans oublier ma chère Amani BEKRI, que je considère comme une sœur. Votre aide précieuse et votre soutien constant ont été inestimables. Merci pour votre amitié et votre dévouement.

À mes chères amies, Bouchra BENSOUNA, Nessrine SELLA, Meriem BOUROUBA, Ikram KOUIDER et Ilhem BENOUIS, merci pour votre sincère amitié. Vous êtes des sœurs sur qui je peux compter. Je vous souhaite une vie pleine de santé et de bonheur. À tous ceux que j'aime, je dédie ce travail.

~.~♥Hadjer Fatima Zahra KACEM♥.~.~

DÉDICACE :

À chaque début correspond une fin, à chaque effort sincère correspond une belle récompense, et à chaque personne assidue correspond une part méritée. Avec chaque mot prononcé avec émotion, les mots restent au-delà de la description. C'est avec une grande fierté que je dédie mon travail :

À celui qui m'a appris que la vie est un combat et dont les armes sont le savoir et la connaissance, à celui qui ne m'a rien refusé, à celui qui a œuvré sans relâche pour mon confort et ma réussite, à l'épaule sur laquelle je pose mes fardeaux et à la main qui me reconforte à chaque instant, à mon cher et tendre, que j'aime autant que ce monde et plus encore, à la patrie à laquelle j'appartiens et à la terre qui m'abrite, à l'homme le plus grand et le plus précieux au monde, mon cher père

À la signification de l'amour et du dévouement, à mon sourire de la vie et ma raison d'exister, à la source de sécurité d'où je puise ma force, à la lumière de mes yeux, ma bonne étoile, ma victoire et ma fierté, à celle qui a été mon premier soutien pour réaliser mes ambitions, à celle qui a été mon refuge et ma main droite dans mes études, à celle par qui j'ai trouvé le chemin de ma vie et mon estime de soi, au cœur tendre, à celle dont les prières m'ont toujours entouré, à la précieuse de mon cœur, ma chère mère.

À la prunelle de mes yeux, ma fierté et ma joie, à celui qui a partagé avec moi les moments de bonheur et de tristesse, les succès et les échecs, à celui qui a toujours été le premier à me soutenir et à m'encourager, à mon compagnon de route, mon ami de tous les jours, mon amour et mon mari, à celui que je vois gravé à jamais au centre de mon cœur et de mon être, à mon soutien après Dieu, mon pilier inébranlable.

À mon binôme qui as partagé avec moi chaque étape de ce voyage, qui as été là pour chaque moment de doute et de succès, pour chaque rire et chaque défi. Ton soutien, ta patience et ton travail acharné ont été essentiels pour atteindre ce but. Ensemble, nous avons relevé chaque défi et appris énormément merci pour ta camaraderie, ton professionnalisme et ton amitié.

À tous qui m'ont toujours encouragé et à qui me souhaite de succès.

~.~♥Chaimaa Hibat Allah BERKANI♥.~.~

Résumé

Les techniques d'apprentissage profond, notamment les réseaux de neurones convolutifs (CNN) et YOLOV8, ont obtenu un grand succès dans le domaine de la vision par ordinateur, montrant une supériorité marquée sur les algorithmes traditionnels de détection d'objets. Ces techniques sont exceptionnellement rapides et précises en matière de reconnaissance d'objets et sont particulièrement adaptées aux applications de surveillance en temps réel.

Cette recherche vise à étudier et appliquer un algorithme de détection d'objets basé sur l'apprentissage profond pour diagnostiquer les maladies des abeilles. L'étude s'est concentrée sur l'utilisation de l'algorithme YOLOV8 comme méthode pratique, et les résultats ont montré des résultats très satisfaisants, ce qui confirme la supériorité exceptionnelle de l'algorithme YOLOV8 sur les méthodes traditionnelles telles que la méthode Haar dans ce contexte.

Mots clés: Détection d'objets, YOLOV8, deep learning, réseaux de neurones.

Abstract

Deep learning techniques, including convolutional neural networks (CNN) and YOLOV8, have achieved significant success in computer vision, demonstrating substantial superiority over traditional object detection algorithms. These techniques excel in speed and exceptional accuracy in object recognition, particularly suitable for real-time monitoring applications.

This research aims to study and apply deep learning-based object detection algorithms for diagnosing bee diseases. The study focused on using YOLOV8 as a practical method, yielding highly satisfactory results that underscore YOLOV8's clear superiority over traditional methods like the Haar technique in this context.

Key words: Object detection, YOLOV8, deep learning, neural networks.

ملخص

حققت تقنيات التعلم العميق، بما في ذلك الشبكات العصبية التلافيفية (CNN) وYOLOV8، من نجاح كبير في مجال الرؤية الحاسوبية، مظهرة تفوقاً ملحوظاً على الخوارزميات التقليدية لكشف الأشياء. تتميز هذه التقنيات بسرعة ودقة استثنائية في التعرف على الأشياء، وهي خصوصاً ملائمة في تطبيقات المراقبة في الوقت الحقيقي.

يهدف هذا البحث إلى دراسة وتطبيق خوارزمية الكشف عن الأشياء التي تعتمد على التعلم العميق لتشخيص أمراض النحل. تمحورت الدراسة حول استخدام خوارزمية YOLOV8 كطريقة عملية، وأظهرت النتائج تحقيق نتائج مرضية للغاية، مما يؤكد التفوق البارز لخوارزمية YOLOV8 على الطرق التقليدية مثل طريقة هار في هذا السياق.

الكلمات المفتاحية: كشف الأشياء، YOLOV8، التعلم العميق، الشبكات العصبية.

Table des matières

Table de figures	12
Liste des tableaux	12
Liste des abréviations	12
Introduction Générale	12
1 Chapitre 01 : Généralité sur l'Intelligence Artificielle	3
1.1 Définition de l'IA	5
1.1.1 Historique de l'Intelligence Artificielle	5
1.1.2 Machine Learning (ML)	6
1.1.3 Deep Learning (DL)	7
1.1.4 La différence entre l'apprentissage profond (DL) et l'apprentissage automatique (ML)	8
1.2 Réseaux de neurons	9
1.2.1 Neurones Biologiques	9
1.2.2 Neurones artificiel	10
1.2.3 Convolutional Neural Networks (CNN)	10
1.2.4 Les Réseaux de Neurones Récurrents (RNN)	12
1.3 YOLO (You Only Look Once)	13
1.3.1 L'importance de YOLO	14
1.3.2 L'architecture YOLO	15
1.3.3 Comparaison des versions de YOLO	20
2 Chapitre 02 : La santé des abeilles	25
2.1 Les maladies des abeilles	27

2.1.1	Le varroa destructor	27
2.1.2	La loque américaine	30
2.1.3	Le nosema ceranae	32
2.1.4	Sacbrood virus	34
2.2	Les causes des maladies des abeilles	36
2.2.1	Les pesticides	37
2.2.2	Le changement climatique	37
2.2.3	La perte d’habitat	37
2.2.4	Déforestation	37
2.2.5	Urbanisation	38
2.2.6	Utilisation de pesticides :	38
2.3	Les conséquences des maladies des abeilles	38
2.4	Lutte contre les maladies des abeilles	39
3	Chapitre 03 : Système de détection des maladies d’abeilles	41
3.1	Etat de l’art	42
3.2	Problématique	44
3.3	Solution proposée	44
3.4	Environnement de travail	45
3.4.1	Spécifications de l’ordinateur portable	45
3.4.2	Logiciels utilisés	45
3.4.3	Langage de programmation utilisé (Python)	46
3.5	Entraînement à l’aide du YOLOv8	46
3.5.1	Création de base de données	47
3.5.2	Entraînement	49
3.6	Résultats	52
3.6.1	Précision	52
3.6.2	Pertes pendant l’entraînement et la validation	53
3.6.3	Métriques de performance	54
3.7	Validation du modèle	55
3.7.1	Test du modèle YOLOv8	56
3.7.2	Via une application WEB	56
3.7.3	ESP32-CAM	58

3.7.4	Installer le logiciel requis	59
3.7.5	Configurer l'ESP32-CAM	59
3.7.6	Préparer l'environnement YOLOv8	60
3.7.7	Intégrer YOLOv8 avec l'ESP32-CAM	61
3.7.8	RESULTAT DU TEST	63
Conclusion Générale		68
A Première annexe		69
A.1	Importation Bibliothèque	69
A.2	Initialisation de l'Application Flask	69
A.3	Chargement du Modèle YOLO	70
A.4	Initialisation de la Capture Vidéo	70
A.5	Modèle HTML pour l'Interface Utilisateur	70
A.6	Boucle pour Capturer les Frames et Faire des Prédictionns	72
A.7	Génération du Flux Vidéo pour l'Interface Utilisateur	73
A.8	Routes Flask pour l'Interface Utilisateur et Contrôle Vidéo	74
A.9	Démarrage de l'Application Flask	75
B Deuxième annexe		76
B.1	Prototype	76

Table des figures

1.1	Frise chronologique avec les principales étapes dans l’histoire de l’IA	6
1.2	Relation entre IA et ML et DL	6
1.3	Structure du neurone biologique [Yandex].	10
1.4	Structure d’un neurone artificiel	10
1.5	Convolutional Neural Networks (CNN)	11
1.6	Réseaux de Neurones Récurrents (RNN)	12
1.7	illustration du structure de la méthode YOLO [22]	13
1.8	Comment YOLO gère les boîtes englobantes	14
1.9	Modèle Darknet-53.	15
1.10	Les attribues des cadres.	16
1.11	Les paramètres de prédiction.	17
1.12	Trois échelles de d’édiction.	19
1.13	Fonction ”Intersection over Union”.	19
1.14	Filtre non-max suppression.	20
1.15	YOLOv8 architecture.	23
2.1	Varroa	27
2.2	Varroa sur l’abeille (Apiculteur.ch)	27
2.3	Abeille ouvrière adulte infectée par le virus des ailes déformées	28
2.4	Loque Américain (Apistory.fr)	30
2.5	Nosema	32
2.6	Sacbrood (Beeaware.org)	35
2.7	les taux de prévalence des principales maladies des abeilles en Algérie (http://hdl.handle.net/123456789/	
3.1	Le varroa destructor chez les abeilles.	47
3.2	Annotation d’images sur Roboflow.	48

3.3	Exportation de base de données.	48
3.4	Division de la base de données.	49
3.5	Digramme des étapes d’entraînement et évaluation.	50
3.6	Code de téléchargement de base de données.	51
3.7	Téléchargement de base de données sur Colab.	51
3.8	Lancement d’entraînement.	52
3.9	Précision obtenue pour la classe Varroa	52
3.10	Graphs de précision.	54
3.11	Détection automatique des acariens Varroa sur les abeilles.	56
3.12	Interface de l’application web	57
3.13	Test de l’application développée	57
3.14	ESP32 CAM.	59
3.15	Adresse IP du ESP32-CAM.	60
3.16	Installation des bibliothèques de python.	60
3.17	Importation des Bibliothèques.	61
3.18	Définir les Constantes.	61
3.19	Charger le Modèle YOLO	61
3.20	Fonction pour Obtenir l’Image de l’ESP32-CAM.	62
3.21	boucle principale	62
3.22	Résultat du test.	63
3.23	Résultat de détection.	63
B.1	Ruche avec ESP32-Cam.	76

Liste des tableaux

- 1.1 Comparaison entre Machine Learning et Deep Learning. 9
- 1.2 Comparaison entre Neurone biologique et Neurone artificiel. 10

Liste des abréviations

AI	Intelligence Artificielle
ML	Machine Learning
DL	Deep Learning
GPU	Graphics Processing Unit
FNN	Feedforward Neural Network
CNN	Convolutional Neural Networks
RNN	Réseaux de Neurones Récurents
YOLO	You Only Look Once
GAN	Generative Adversarial Networks
NMS	la suppression non maximale
IOU	Intersection over Union
SSD	Single Shot Multi-Box Detector
BOF	Bag-of-Freebies
SPPF	Spatial Pyramid Pooling Fast
CLI	Interface en ligne de commande
IOT	Internet of Things
FAO	Food and Agriculture Organization of the United Nations
CTA	La Chute Totale des Acariens
AFB	American Foulbrood
UV	Ultraviolet
SBV	Sacbrood Bee Virus
PCR	Polymerase Chain Reaction

CCD Colony Collapse Disorder

IoT Internet of Things

HDD Hard Disk Drive

AMD Advanced Micro Devices

OpenCV Open Computer Vision

Introduction Générale

L'avènement de l'intelligence artificielle (IA) a inauguré une ère de transformation profonde dans divers secteurs, de la technologie à l'agriculture, en passant par la conservation de la biodiversité. Cette révolution technologique, fondée sur les progrès du machine learning et du deep learning, a permis des avancées significatives telles que la détection d'objets avec des systèmes comme YOLO (You Only Look Once). Parallèlement, les abeilles, en particulier l'abeille domestique (*Apis mellifera*), jouent un rôle critique en tant que principaux pollinisateurs, contribuant directement à la production alimentaire mondiale et à la préservation des écosystèmes naturels.

Notre mémoire se concentre sur l'exploration de ces deux domaines interconnectés à travers trois chapitres distincts. Le premier chapitre examine en profondeur l'évolution et l'impact de YOLO dans le domaine de l'IA, analysant ses applications variées et ses implications dans des secteurs technologiques et industriels spécifiques. Cette section mettra en lumière les différentes versions de YOLO, leurs avantages, leurs limitations ainsi que les perspectives futures de développement.

Le deuxième chapitre explore les défis pressants auxquels sont confrontées les populations d'abeilles, en mettant l'accent sur les maladies et les menaces environnementales qui compromettent leur santé et leur survie. Nous étudierons les facteurs contribuant à ces défis, les impacts sur les colonies d'abeilles, ainsi que les stratégies actuelles et émergentes de gestion et de conservation des ruches pour préserver leur rôle crucial dans la pollinisation et la biodiversité.

Enfin, le troisième chapitre se penche sur le potentiel de l'intelligence artificielle et des technologies de surveillance innovantes pour soutenir la santé des abeilles. Nous explorerons comment les capteurs intelligents, l'Internet des objets (IoT) et d'autres outils technologiques peuvent être intégrés pour surveiller en temps réel les colonies d'abeilles, détecter précocement les menaces telles que le *Varroa destructor*, et renforcer ainsi la sécurité alimentaire mondiale en assurant la continuité des services écosystémiques essentiels.

En résumé, cette mémoire vise à offrir une analyse approfondie et intégrée de l'impact de l'IA à travers YOLO, des défis critiques pour la santé des abeilles, et des solutions technologiques prometteuses pour préserver ces pollinisateurs essentiels et sécuriser notre approvisionnement alimentaire futur.

Chapitre 1

Chapitre 01 : Généralité sur l'Intelligence Artificielle

Introduction

L'intelligence artificielle (IA) a transformé de nombreux aspects de notre monde moderne, notamment à travers des technologies avancées telles que la détection d'objets. Parmi les méthodes les plus innovantes de détection d'objets, YOLO (You Only Look Once) se distingue par son efficacité et sa rapidité. Ce chapitre se propose d'explorer en profondeur YOLO, en retraçant son évolution et en analysant son impact sur le domaine de l'IA. Le chapitre explore le domaine de l'intelligence artificielle (IA) et se concentre particulièrement sur l'évolution de la détection d'objets à travers les différentes versions de YOLO (You Only Look Once). Pour comprendre pleinement YOLO et son impact, il est essentiel de retracer l'histoire de l'IA, en mettant en lumière des concepts tels que le Machine Learning (ML) et le Deep Learning (DL), ainsi que les différents types de réseaux de neurones. Ensuite, nous plongerons dans YOLO, en examinant son architecture, les différences entre ses différentes versions, ainsi que ses avantages et inconvénients. Enfin, nous discuterons des tendances futures de YOLO et de son rôle potentiel dans l'avenir de l'IA.

1.1 Définition de l'IA

L'Intelligence Artificielle (IA) vise à faire exécuter par une machine des tâches qui nécessitent habituellement l'intelligence humaine. Le terme "Intelligence Artificielle" a été introduit en 1956, bien que le terme "Informatique Heuristique" puisse être considéré comme une alternative plus appropriée. Contrairement à une machine intelligente ou un programme intelligent, l'IA se concentre sur l'étude des activités intellectuelles humaines pour lesquelles aucune méthode prédéfinie n'existe. En informatique, où le traitement de l'information est central, l'IA intervient lorsque ce traitement ne peut être résolu par une méthode simple, précise et algorithmique. Un algorithme se compose d'une séquence d'opérations clairement définies, exécutable sur un ordinateur contemporain, permettant d'atteindre une solution en un temps raisonnable, allant de quelques minutes à quelques heures, mais pas des siècles.[21]

1.1.1 Historique de l'Intelligence Artificielle

L'Intelligence Artificielle (IA) est officiellement née en 1956 lors d'une conférence à Dartmouth, aux États-Unis. Les recherches avaient en réalité commencé après la Seconde Guerre mondiale. Cette conférence historique a réuni des figures éminentes de l'IA telles que John McCarthy, Herbert Simon, Allen Newell, Claude Shannon et Marvin Minsky. Ces pionniers avaient pour ambition de créer des machines capables d'égaliser, voire de surpasser, l'intelligence humaine en exploitant les nouvelles avancées de l'informatique émergente.

En 1955, Newell et Simon ont développé un programme appelé "Logic theorist" qui a pu automatiquement démontrer une grande partie des théorèmes du traité de "Principia Mathematica" d'Alfred North Whitehead et Bertrand Russell. Ce programme a été considéré comme la première intelligence artificielle de l'histoire, marquant ainsi un jalon important. Par la suite, ils ont conçu le GPS ("General Problem Solver"), capable de résoudre divers types de problèmes spécifiés formellement. Dans les années suivantes, d'autres chercheurs ont exploré la traduction automatique, la robotique, la théorie des jeux, la vision, et d'autres domaines. Jusqu'au milieu des années 70, l'IA a connu une période très productive, mais a ensuite été confrontée à des défis majeurs. Les chercheurs ont réalisé que la promesse initiale de créer une intelligence générale équivalente à celle humaine n'avait pas été tenue. Cette période a été appelée le "premier hiver de l'IA". Cependant, avec l'avènement des systèmes experts et la résolution des problèmes d'apprentissage des perceptrons multi-couches, l'IA a connu un renouveau jusqu'à la fin des années 1990, où elle a de nouveau stagné pour des raisons similaires. Cette histoire mouvementée de l'IA illustre ses avancées, ses défis et ses périodes de stagnation, marquant ainsi son évolution à travers les décennies voir figure (1.1).[23]



FIGURE 1.1 – Frise chronologique avec les principales étapes dans l’histoire de l’IA

1.1.2 Machine Learning (ML)

L’objectif principal du Machine Learning (ML) réside dans la conception de programmes informatiques capables d’apprendre à partir de données. Ce processus d’inférence inductive, où les généralisations sont tirées à partir d’un ensemble d’instances observées, se différencie des premières approches de l’intelligence artificielle (IA) qui se concentraient principalement sur l’inférence déductive, la dérivation de théorèmes à partir d’axiomes. Bien que le ML soit souvent considéré comme un sous-domaine de l’IA comme il est illustré dans la figure (1.2) , il entrelace également diverses disciplines scientifiques telles que les statistiques, les sciences cognitives et la théorie de l’information.

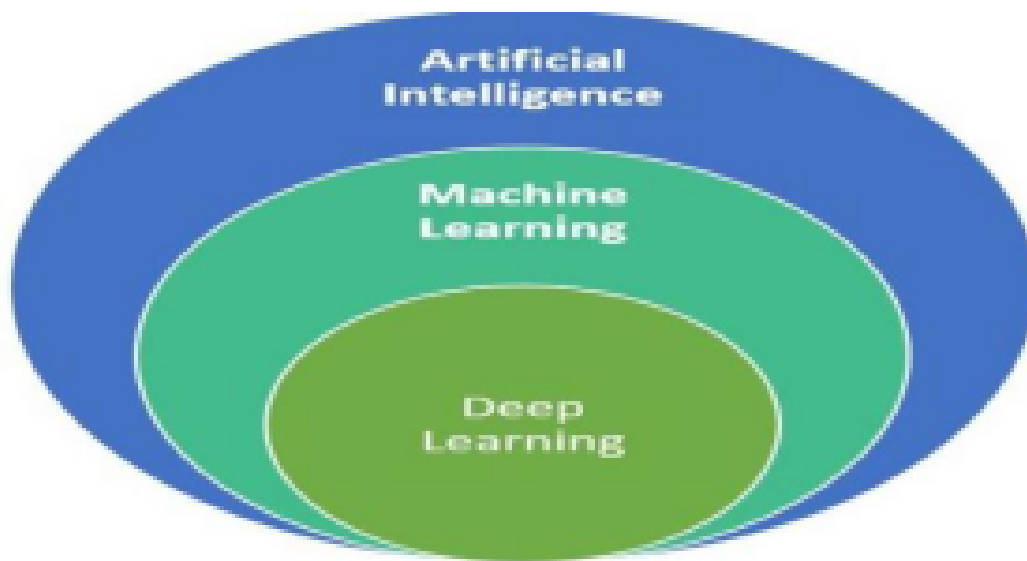


FIGURE 1.2 – Relation entre IA et ML et DL

Un domaine étroitement lié au ML est l'exploration de données, qui vise à découvrir des modèles nouveaux et intéressants à partir de vastes ensembles de données. Bien que le ML et l'exploration de données soient parfois confondus, on peut souligner que le ML se concentre davantage sur l'adaptabilité et l'utilisation opérationnelle, tandis que l'exploration de données s'attache à la gestion de grandes quantités de données et à la découverte de modèles jusque-là inconnus dans ces données. L'utilisation d'algorithmes de ML devient souvent un processus itératif d'essais et d'erreurs, où le choix de l'algorithme peut influencer considérablement les performances obtenues. Cette approche est efficace dans certains contextes, mais peut poser problème, notamment dans le domaine de la modélisation du langage et de la vision par ordinateur. Grâce aux progrès considérables en capacités théoriques et technologiques, le Deep Learning a émergé comme l'un des domaines scientifiques les plus dynamiques. Cette approche est largement utilisée dans des technologies telles que les véhicules à conduite autonome, la reconnaissance d'images sur les plateformes de médias sociaux et la traduction de textes entre différentes langues.[16]

1.1.3 Deep Learning (DL)

L'apprentissage en profondeur, une branche de l'apprentissage automatique, vise à enseigner aux ordinateurs à apprendre de l'expérience, de manière similaire à l'apprentissage humain. Contrairement aux méthodes traditionnelles qui utilisent des équations prédéterminées, les algorithmes d'apprentissage automatique, notamment ceux de l'apprentissage en profondeur, apprennent directement à partir de données. Cette approche est particulièrement efficace dans des domaines tels que la reconnaissance d'images, essentielle pour des applications telles que la reconnaissance faciale, la détection de mouvements, ainsi que pour des technologies avancées d'assistance à la conduite comme la conduite autonome, la détection de voies, et le stationnement automatique.

L'apprentissage en profondeur repose sur l'utilisation de réseaux de neurones pour apprendre des représentations utiles de caractéristiques à partir des données. Ces réseaux comportent plusieurs couches de traitement non linéaires, inspirées par les systèmes biologiques et fonctionnant en parallèle. Les modèles d'apprentissage en profondeur peuvent atteindre des niveaux de précision remarquables dans la classification des objets, parfois même dépassant les performances humaines. Ces modèles sont généralement entraînés sur de vastes ensembles de données étiquetées et utilisent des architectures de réseau de neurones avec de nombreuses couches, incluant souvent des couches convolutives. La formation de ces modèles est intensivement calculatoire, mais elle peut être accélérée en utilisant des GPU haute performance.

De nombreuses applications d'apprentissage en profondeur impliquent l'utilisation de millions de fichiers image. Pour cela, l'accès à un grand nombre de fichiers image est essentiel pour un apprentissage efficace. L'apprentissage par transfert est couramment utilisé dans ce domaine, permettant de réutiliser un réseau pré-entraîné comme point de départ pour une nouvelle tâche. Cette approche est plus rapide et plus facile que l'entraînement d'un nouveau réseau à partir de zéro, et elle peut être réalisée avec un nombre réduit d'images d'entraînement. L'avantage de l'apprentissage par transfert réside dans la capacité à transférer les caractéristiques apprises d'un réseau pré-entraîné à un nouveau problème, tout en réduisant le temps de formation et la taille de l'ensemble de données nécessaires.

L'extraction de fonctionnalités est également un aspect important de l'apprentissage en profondeur, permettant d'utiliser les capacités des réseaux pré-entraînés sans nécessiter une nouvelle formation. Cette approche, qui consiste à extraire des fonctionnalités apprises d'un réseau pré-entraîné pour former des classificateurs, peut être une méthode rapide pour tirer parti de l'apprentissage en profondeur dans de nouveaux contextes.[30]

1.1.4 La différence entre l'apprentissage profond (DL) et l'apprentissage automatique (ML)

L'apprentissage en profondeur constitue une catégorie de l'apprentissage automatique. Contrairement à l'approche traditionnelle de l'apprentissage automatique où les fonctions pertinentes d'une image sont extraites manuellement, l'apprentissage en profondeur permet de soumettre directement les images brutes à un réseau neuronal profond, qui apprend automatiquement les caractéristiques pertinentes. Pour obtenir des résultats optimaux, l'apprentissage en profondeur nécessite souvent des centaines de milliers, voire des millions d'images. De plus, cette méthode est très exigeante en termes de calcul et requiert l'utilisation de GPU haute performance le tableau (1.1) représente la différence entre apprentissage profond et apprentissage automatique .[30]

Machine Learning	Deep Learning
++ De bons résultats avec de petits ensembles de données	– Nécessite de très grands ensembles de données
++ Rapide pour former un modèle	– Intensément computationnel
– Besoin d’essayer différents fonctionnalités et classificateurs pour obtenir les meilleurs résultats	++ Apprend les fonctionnalités et les classificateurs automatiquement
– Plateaux de précision	++ La précision est illimitée

Tableau 1.1 Comparaison entre Machine Learning et Deep Learning.

1.2 Réseaux de neurons

Les neurones, sont les cellules fondamentales du système nerveux, jouent un rôle crucial dans le traitement de l’information et la transmission des signaux électriques.

1.2.1 Neurones Biologiques

Le cerveau humain est un réseau complexe de cellules nerveuses spécialisées appelées neurones, chargées de transmettre des informations entre elles à l’aide de signaux électriques et chimiques. Chaque neurone est constitué d’un corps cellulaire contenant le noyau et les organites nécessaires à son fonctionnement, de dendrites courtes et ramifiées qui reçoivent des signaux d’autres neurones ou de cellules sensorielles, et d’un axone long et fin qui transmet les signaux du corps cellulaire à d’autres neurones ou à des cellules cibles telles que les muscles ou les glandes comme c’est illustré dans la figure (1.3). Ces neurones interagissent au niveau des synapses, formées lorsque les dendrites d’un neurone se connectent à l’axone d’un autre neurone, créant ainsi un réseau neuronal sophistiqué. Lorsqu’un neurone reçoit un signal électrique via son axone, il l’intègre et le transmet à tous les neurones connectés via ses dendrites. [15]

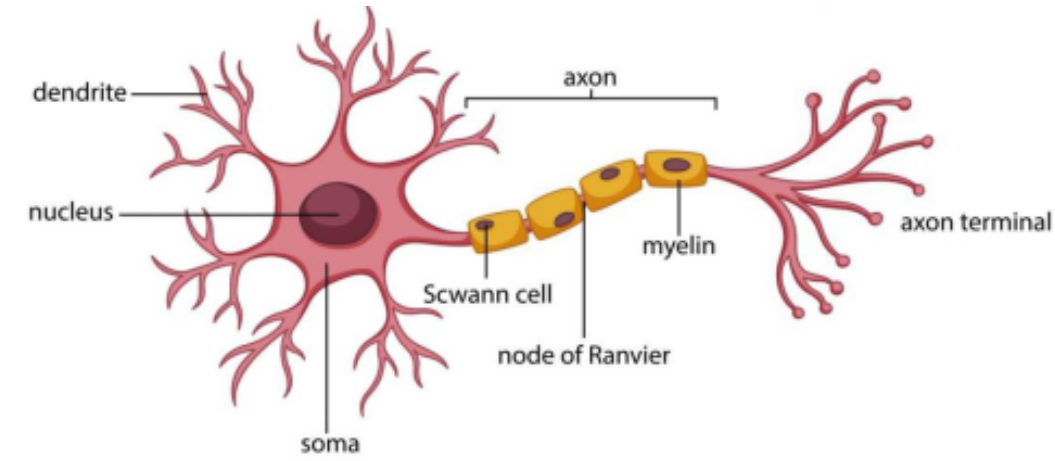


FIGURE 1.3 – Structure du neurone biologique [Yandex].

1.2.2 Neurones artificiel

La figure (1.4) montre Le neurone artificiel qui est également appelé perceptron, constitue l'élément fondamental des réseaux de neurones artificiels (artificial neural network) (ANN). Son but est de reproduire de manière simplifiée le fonctionnement des neurones biologiques du cerveau. Sa principale fonction est de traiter l'information en recevant des entrées et en produisant une sortie correspondante.[15] Le ta-

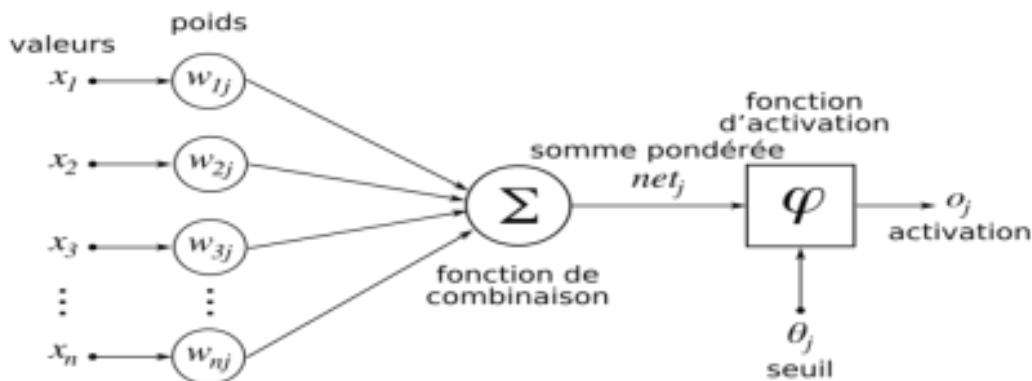


FIGURE 1.4 – Structure d'un neurone artificiel

bleau (1.2) met en évidence la comparaison possible entre le neurone biologique et les neurones artificiels.

1.2.3 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) sont une forme spécialisée de réseau de neurones artificiels. Ce qui les distingue des autres architectures, comme illustré dans la figure (1.5), c'est que chaque neurone dans une couche CNN n'est pas connecté à tous les autres neurones de la couche suivante. Cette caractéristique permet aux CNN d'exceller dans les tâches impliquant une notion de spatialité, ce qui en fait

Neurone biologique	Neurone artificiel
Synapses	Connections pondérées
Axone	Sortie
Dendrites	Entrées
Sommeteur	Fonction d'activation

Tableau 1.2 Comparaison entre Neurone biologique et Neurone artificiel.

un choix privilégié pour la reconnaissance de motifs, tels que les bords ou les segmentations d'images. [17]

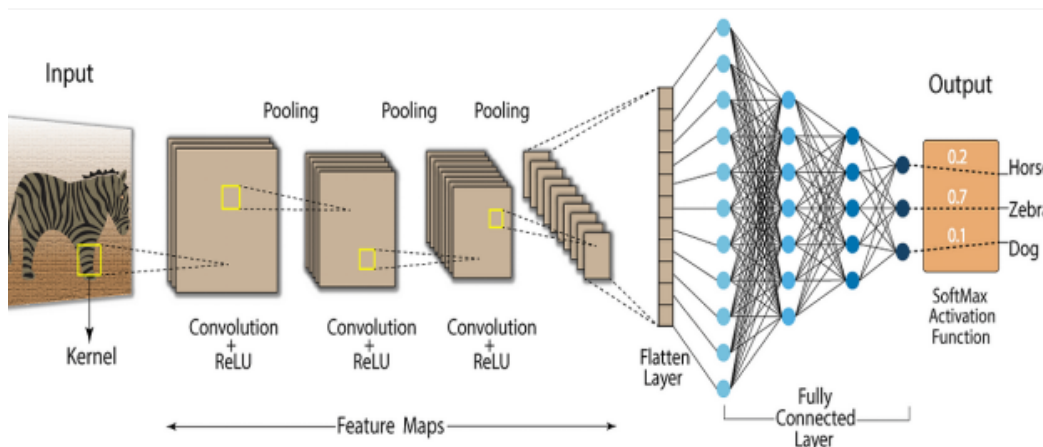


FIGURE 1.5 – Convolutional Neural Networks (CNN)

Dans les applications artistiques de l'intelligence artificielle, notamment avec les GAN (Generative Adversarial Networks), les CNN sont largement utilisés comme générateurs. Ils sont efficaces pour reconnaître des motifs spécifiques dans les données, tels que les bords (contours) ou les segmentations d'images, comme décrit dans des travaux antérieur.

Les CNN peuvent détecter la présence de motifs spécifiques dans une image à l'aide d'un processus appelé "filtering". Ce processus consiste à examiner chaque pixel de l'image pour déterminer s'il correspond à la caractéristique recherchée. Les résultats de cette opération sont représentés sous forme de "feature maps", qui sont ensuite soumis à des couches de convolutions successives pour rechercher des motifs à différentes échelles.

Pour réduire la taille des données et extraire les informations les plus pertinentes, les CNN utilisent des "pooling layers". Ces couches permettent de réduire le jeu de données en ne conservant que les valeurs maximales dans des fenêtres de la "feature map".

Enfin, les données sont normalisées et transformées en une liste pour être soumises à une "fully connected layer", qui génère le résultat final. Ce résultat est ensuite comparé aux données d'entrée, et les paramètres du réseau sont ajustés pour se rapprocher de l'input d'origine grâce à la rétro-propagation. Cette phase d'apprentissage permet au réseau de neurones de s'adapter aux caractéristiques spécifiques des données et d'améliorer sa performance dans la tâche assignée, que ce soit la détection de contours d'images ou la classification d'images.

1.2.4 Les Réseaux de Neurones Récurents (RNN)

Constituent un autre type important de réseau de neurones artificiels, inventés en 1925 par Lenzet Ising. Contrairement aux CNN qui se propagent uniquement dans un sens, les RNN présentent une caractéristique distincte : les couches cachées peuvent contenir des connexions récurrentes, comme illustré dans la figure (1.6). Cela signifie qu'une sortie peut être réutilisée comme entrée dans la couche cachée du RNN, permettant ainsi au réseau de maintenir une sorte de mémoire ou de temporalité dans ses calculs. [17]

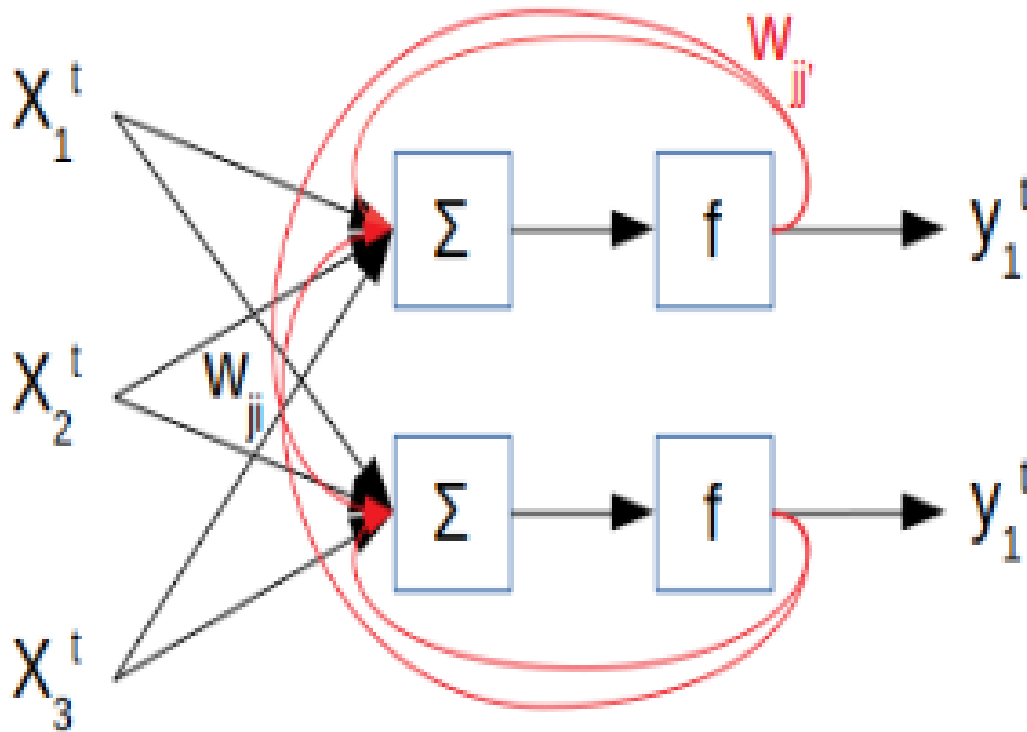


FIGURE 1.6 – Réseaux de Neurones Récurents (RNN)

Les RNN sont largement utilisés dans des domaines où la notion de séquence et de mémoire est cruciale, tels que la musique ou le traitement de texte. Cependant, malgré leur capacité à prendre en compte la récurrence dans les données, les RNN rencontrent souvent des problèmes de "vanishing gradient" et "exploding gradient", qui peuvent affecter la précision du réseau et rendre l'apprentissage moins efficace. Il existe plusieurs sous-types de RNN, dont les architectures peuvent être plus complexes, comprenant plusieurs couches de RNN et différentes directions de propagation. Bien que les RNN soient moins couramment utilisés dans la génération d'art visuel, ils sont parfois employés pour des tâches spécifiques, telles que la détermination de l'ordre des traits ou des coups de pinceaux dans le processus de dessin ou de peinture, visant à enseigner aux machines à imiter les comportements humains étape par étape.

1.3 YOLO (You Only Look Once)

Yolo a été introduit en 2015 par Joseph Redmon, suivi de deux versions ultérieures, YOLO9000 (YOLOv2) et YOLOv3, publiées en mai 2018. Cette approche révolutionnaire vise à prédire avec précision les boîtes englobantes tout en utilisant une technique de fenêtre glissante à convolution comme le montre la figure (1.7).^[11] L'algorithme YOLO adopte une approche novatrice en divisant l'image en une grille

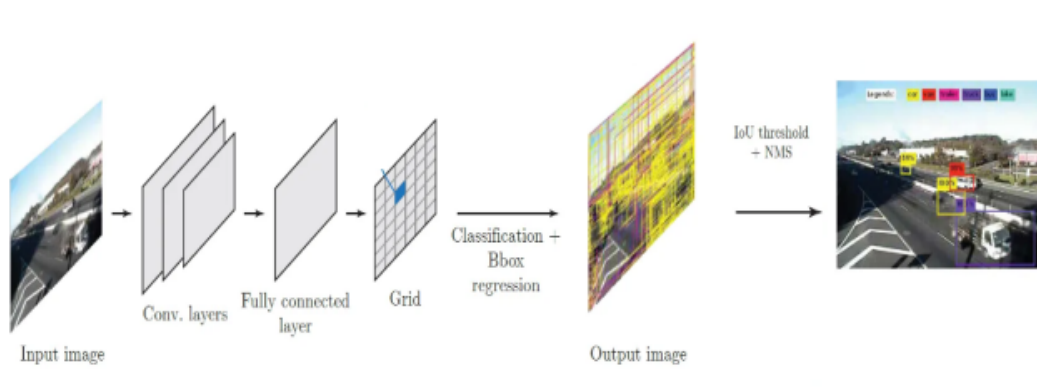


FIGURE 1.7 – illustration du structure de la méthode YOLO ^[22]

et en assignant à chaque cellule de la grille N boîtes englobantes. Pour chaque boîte englobante, le réseau génère une probabilité de classe et des valeurs de décalage pour localiser l'objet dans l'image. Les boîtes englobantes dont la probabilité de classe dépasse un seuil prédéfini sont sélectionnées pour identifier l'objet dans l'image comme le montre la figure (1.8).

L'un des principaux atouts de YOLO est sa vitesse exceptionnelle, capable de fonctionner à 45 images par seconde sur un Titan X, surpassant ainsi de loin d'autres algorithmes de détection d'objets. Cependant, YOLO rencontre des difficultés avec de petits objets dans l'image, comme un troupeau

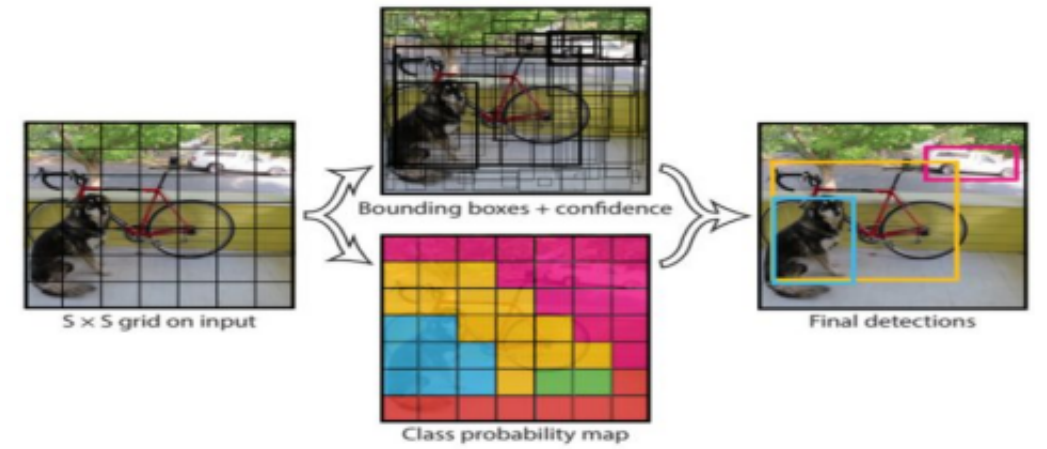


FIGURE 1.8 – Comment YOLO gère les boîtes englobantes

d’oiseaux, en raison des contraintes inhérentes à l’algorithme. Malgré ses limitations, YOLO présente l’avantage majeur d’être extrêmement rapide, car il n’exécute l’image qu’une seule fois sur le réseau neuronal convolutionnel (CNN), le rendant ainsi bien plus rapide que des méthodes concurrentes telles que Faster R-CNN, et permettant même une exécution en temps réel.

1.3.1 L’importance de YOLO

Réside dans sa capacité à surpasser les autres méthodes de détection d’objets, telles que les réseaux neuronaux régionaux (R-CNN) et ses variantes, qui se concentrent sur des régions spécifiques de l’image et entraînent chaque composant individuellement. Par exemple, le processus de classification des 2000 régions par image du R-CNN rend son exécution très lente, prenant jusqu’à 47 secondes par image test individuelle, ce qui le rend inadapté pour une mise en œuvre en temps réel.

Comparativement, le modèle YOLO se distingue par sa rapidité impressionnante, capable de traiter jusqu’à 45 images par seconde. Cette performance accrue est rendue possible par le fait que YOLO utilise un seul réseau neuronal pour exécuter toutes les étapes de la tâche de détection d’objets, ce qui le rend non seulement plus rapide, mais aussi plus facile à optimiser par rapport aux algorithmes précédents, tels que les R-CNN.[\[20\]](#)

1.3.2 L'architecture YOLO

YOLO est composée des éléments principaux suivants [20] :

1. Un réseau de neurones entièrement convolutif

- YOLO se distingue par l'utilisation exclusive de couches convolutives, faisant ainsi de lui un réseau entièrement convolutif (FCN).
- Dans la version YOLO v3, les concepteurs ont introduit une nouvelle architecture plus profonde pour l'extraction des caractéristiques, appelée Darknet-53.
- Cette architecture comporte 53 couches convolutionnelles, chacune suivie d'une normalisation par lots et d'une activation ReLU.
- Contrairement à d'autres architectures, YOLO n'utilise pas de couches de pooling, préférant une couche de convolution avec un stride de 2 pour le sous-échantillonnage des cartes de caractéristiques.
- Cette méthode permet de conserver les informations de bas niveau sans perte due à la mise en commun, comme le montre la figure (1.9)

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

FIGURE 1.9 – Modèle Darknet-53.

2. L'interprétation de la prédiction

- Entrée : Un lot d'images au format (m, H, W, C)
- m représente le nombre d'images
- H représente la hauteur de chaque image en pixels
- W correspond à la largeur de chaque image en pixels
- C indique le nombre de canaux de couleur par pixel, souvent 3 pour les images RGB (rouge, vert, bleu).
- Sortie : Une liste de boîtes englobantes accompagnées des classes détectées
- Chaque boîte englobante est représentée par 6 valeurs (pc, bx, by, bh, bw, c)
- pc : Probabilité de classe
- bx, by : Coordonnées du centre de la boîte
- bh, bw : Hauteur et largeur de la boîte
- YOLO utilise une couche convolutionnelle avec des convolutions 1×1 pour produire une carte de caractéristiques en sortie.
- Chaque cellule de cette carte prédit un nombre fixe de boîtes englobantes, par exemple, dans YOLO v3, chaque cellule peut prédire 3 boîtes englobantes.
- L'assignation de la prédiction à une boîte englobante spécifique se fait en fonction de la cellule de la carte des caractéristiques contenant le centre de l'objet comme le montre la figure (1.10).

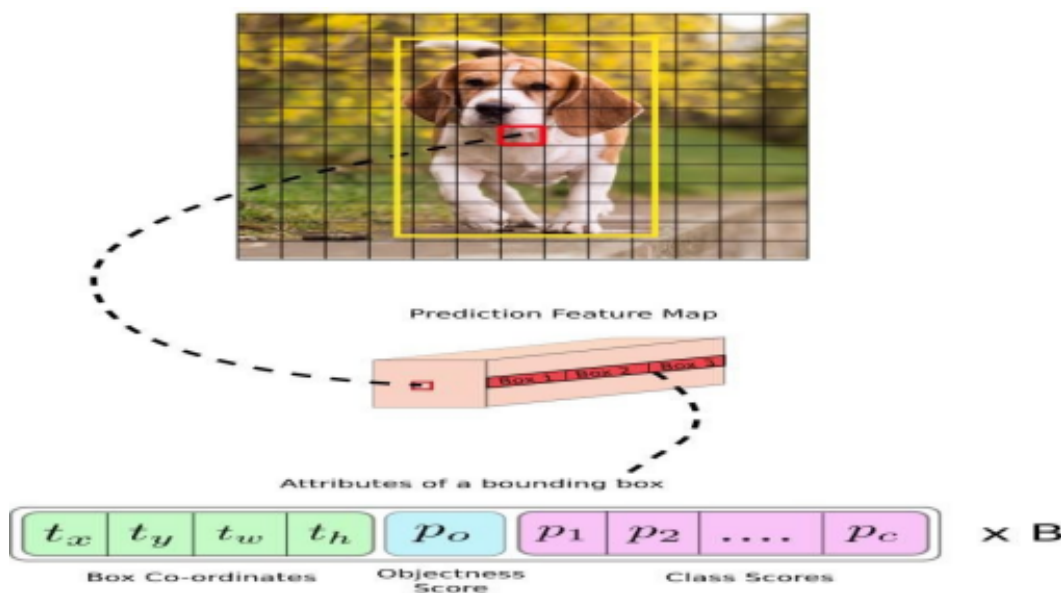


FIGURE 1.10 – Les attribues des cadres.

3. Boîtes d’ancrage et prévision

- Les détecteurs d’objets modernes adoptent une approche plus stable en prédisant des transformations log-espace ou des décalages par rapport à des boîtes de délimitation prédéfinies appelées ancres.
- YOLO v3 utilise trois ancres, permettant de prédire trois boîtes englobantes par cellule.
- Les ancres sont des boîtes englobantes précalculées à partir de l’ensemble de données COCO.
- Plutôt que de prédire directement les dimensions de la boîte, on prédit les décalages par rapport aux centres de ces ancres.
- Les coordonnées du centre de la boîte par rapport à l’emplacement de l’application du filtre sont prédites à l’aide d’une fonction sigmoïde.
- Cette approche assure une prédiction plus robuste des boîtes englobantes tout en minimisant les perturbations potentielles pendant l’entraînement.

4. Dimensions de la boîte de délimitation

- Les dimensions de la boîte englobante sont prédites en appliquant une transformation log-espace à la sortie, puis en la multipliant par une ancre.
- Les prédictions b_w et b_h sont normalisées par la hauteur et la largeur de l’image. Par exemple, si les prédictions b_x et b_y pour la boîte contenant le chien sont $(0,3, 0,8)$, alors la largeur et la hauteur réelles sur la carte des caractéristiques 13×13 sont $(13 \cdot 0,3, 13 \cdot 0,8)$.

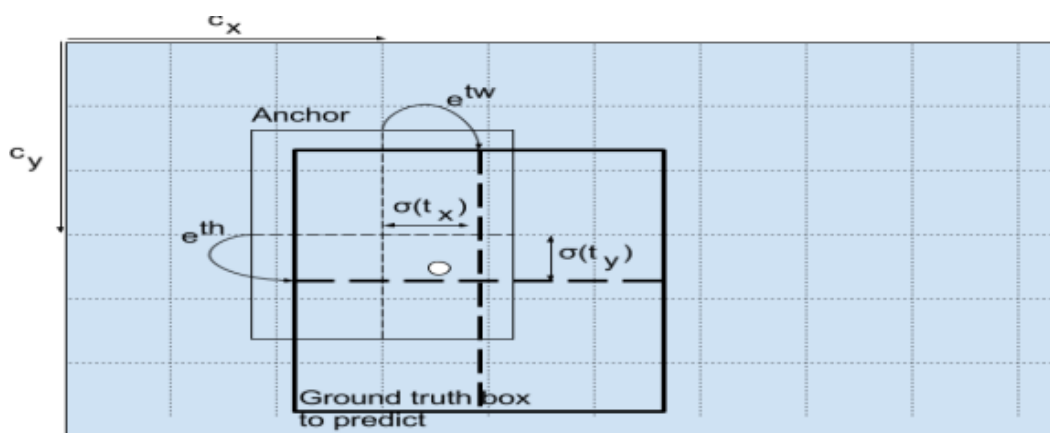


FIGURE 1.11 – Les paramètres de prédiction.

5. Score d'objectivité et confiance en classe

- Le score d'objectivité (confidence score) indique la probabilité qu'un objet soit présent dans une boîte englobante spécifique.
- Ce score devrait être élevé pour les boîtes englobantes qui contiennent effectivement un objet et bas pour celles qui n'en contiennent pas.
- La fonction de score d'objectivité passe également par une fonction sigmoïde, permettant d'être interprétée comme une probabilité.
- Les confidences de classe représentent les probabilités que l'objet détecté appartienne à une catégorie spécifique (chien, chat, voiture, etc.).
- Les versions ultérieures de YOLO utilisent une approche basée sur la fonction sigmoïde pour les confidences de classe.

6. Prédiction à différentes échelles

- YOLO v3 réalise les prédictions à trois échelles distinctes.
- La couche de détection est utilisée pour analyser des cartes de caractéristiques de tailles différentes, avec des pas de 32, 16 et 8 respectivement.
- Pour une entrée de 416 x 416, les détections sont effectuées sur des échelles de 13 x 13, 26 x 26 et 52 x 52.
- Le processus commence par le sous-échantillonnage de l'image d'entrée jusqu'à la première couche de détection.
- Les couches sont suréchantillonnées par un facteur de 2 et fusionnées avec les cartes de caractéristiques de la couche précédente de même taille.
- Une nouvelle détection est ensuite effectuée sur les cartes de caractéristiques avec un pas de 16, puis sur celles avec un pas de 8.
- À chaque échelle, chaque cellule prédit 3 boîtes de délimitation en utilisant 3 ancres spécifiques, totalisant ainsi 9 ancres utilisées.
- Cette approche permet d'améliorer la détection des petits objets, comme le montre la figure (1.12).

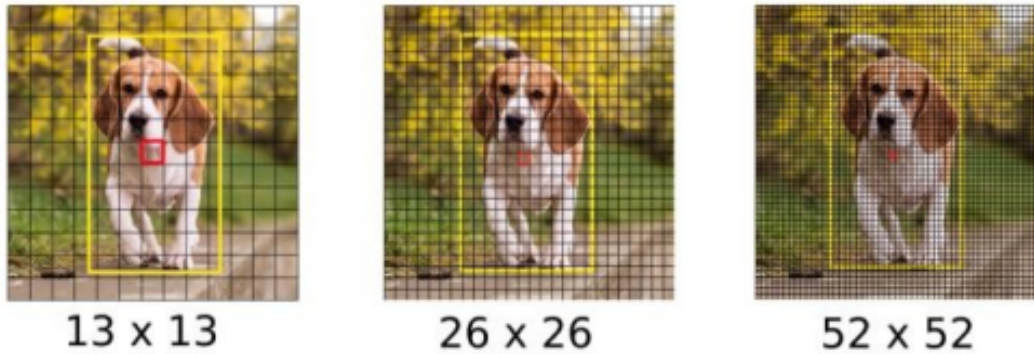


FIGURE 1.12 – Trois échelles de d'édiction.

7. Traitement des sorties (filtrage avec un seuil sur les scores de classe)

- YOLO génère un grand nombre de boîtes englobantes. Par exemple, pour une image de taille 416 x 416, cela peut aller jusqu'à 10647 boîtes englobantes.
- Pour réduire ce grand nombre de détections à une seule, plusieurs étapes de filtrage sont appliquées :
- Filtrage des boîtes en fonction de leur score d'objectivité : Les boîtes avec un score inférieur à un seuil prédéfini sont ignorées.
- Suppression non maximale (NMS) : Utilisée pour supprimer les détections multiples du même objet comme le montre la figure (1.14).
- Répétition des étapes de filtrage jusqu'à ce qu'il ne reste plus de boîtes à traiter.
- Pour calculer le chevauchement entre deux boîtes, on utilise la notion d'Intersection over Union (IoU).
- IoU mesure le rapport entre la zone de chevauchement et la zone totale couverte par les deux boîtes.
- Les boîtes qui se chevauchent plus que le seuil prédéfini de IoU sont supprimées, comme le montre la figure (1.13).

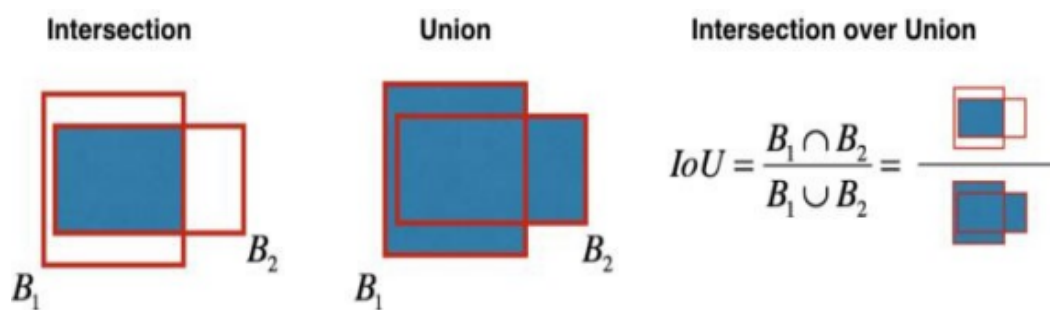


FIGURE 1.13 – Fonction "Intersection over Union".



FIGURE 1.14 – Filtre non-max suppression.

1.3.3 Comparaison des versions de YOLO

Les différentes versions de YOLO ont été développées pour améliorer la précision et les performances du réseau de détection d'objets.

1. YOLO V1

La première version de YOLO, introduite en 2015, a été formée sur le jeu de données ImageNet-1000. Malgré sa rapidité, YOLO V1 avait des difficultés à détecter avec précision les petits objets regroupés et à généraliser les objets dans des images de tailles variables, ce qui entraînait une mauvaise localisation des objets.[20]

2. YOLO V2

Sorti en 2016 sous le nom de YOLO9000, YOLO V2 a utilisé un réseau à 19 couches avec 11 autres couches spécifiques à la détection d'objets. Cette version a été conçue pour prendre en charge des techniques telles que le R-CNN plus rapide et le Single Shot multi-box Detector (SSD), offrant ainsi de meilleures performances en termes de détection d'objets.[20]

3. YOLO V3

YOLO V3 représente une amélioration progressive par rapport à YOLO V2. Avec une architecture composée de 106 couches, dont 53 couches formées sur Imagenet et 53 autres dédiées à la détection d'objets, YOLO V3 a considérablement amélioré la précision du réseau. Cependant, cela a entraîné une légère baisse de la vitesse, passant de 45 à 30 images par seconde.[20]

4. YOLOv4

Publié en avril 2020, représente une évolution majeure de l'architecture de détection d'objets YOLO. Conçu pour maintenir la vitesse d'inférence en temps réel tout en améliorant la précision,

YOLOv4 intègre des améliorations significatives telles que l'utilisation de Darknet-53 avec des connexions partielles entre les étages (CSPNet) et la fonction d'activation Mish. Il exploite également des techniques avancées de formation telles que Bag-of-Freebies (BoF) et l'auto-adversité pour renforcer la robustesse du modèle. YOLOv4 utilise des algorithmes génétiques pour optimiser les hyperparamètres, garantissant ainsi des performances exceptionnelles en termes de précision et de vitesse dans la détection d'objets en temps réel.[28]

5. YOLOv5

Une itération récente de l'architecture YOLO (You Only Look Once) pour la détection d'objets, a été publiée en 2020 par Glen Jocher d'Ultralytics. Cette version repose sur PyTorch au lieu de Darknet, le framework habituel. Une caractéristique notable de YOLOv5 est l'intégration de l'algorithme AutoAnchor, développé par Ultralytics, qui ajuste automatiquement les boîtes d'ancrage pendant le pré-entraînement pour s'adapter aux caractéristiques spécifiques du jeu de données et des paramètres d'entraînement. Cela améliore la précision de la détection en garantissant des boîtes d'ancrage bien adaptées aux objets à détecter. De plus, YOLOv5 incorpore des améliorations d'entraînement telles que l'utilisation de la perte CIoU et du Meilleur Rappel Possible dans un algorithme génétique pour optimiser les ancres. En résumé, YOLOv5 vise à fournir une meilleure précision et des performances de détection d'objets en temps réel grâce à des techniques d'entraînement avancées et à l'ajustement automatique des ancres.[28]

6. YOLOv6

YOLOv6 publié en septembre 2022 par le département Vision IA de Meituan, représente une avancée significative dans la détection d'objets. Ce modèle utilise une conception réseau efficace avec des blocs RepVGG ou CSPStackRep pour le backbone, une topologie PAN pour le cou, et une tête découplée efficace avec une stratégie hybride de canal. De plus, YOLOv6 introduit des techniques de quantification améliorées utilisant la quantification post-entraînement et la distillation par canal, ce qui se traduit par des détecteurs plus rapides et plus précis. Dans l'ensemble, YOLOv6 surpasse les modèles précédents, tels que YOLOv5, YOLOX et PP-YOLOE, en termes de précision et de vitesse.[28]

7. YOLOv7

Publié en juillet 2022 par les mêmes auteurs de YOLOv4 et YOLOR, représente une avancée majeure dans la détection d'objets. Ce modèle a surpassé tous les détecteurs d'objets connus en termes de vitesse et de précision, dans une plage allant de 5 FPS à 160 FPS. Comme YOLOv4, il a été entraîné uniquement avec l'ensemble de données MS COCO sans pré-entraînement des backbones. YOLOv7 a proposé quelques changements architecturaux et une série de méthodes bag-of-freebies, qui ont augmenté la précision sans affecter la vitesse d'inférence, impactant uniquement le temps d'entraînement.[28]

8. YOLOv8

Publié en janvier 2023 par Ultralytics, la société qui a développé YOLOv5, introduit cinq versions adaptées : YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) et YOLOv8x (extra-large). Cette nouvelle version prend en charge plusieurs tâches de vision telles que la détection d'objets, la segmentation, l'estimation de la pose, le suivi et la classification.[28]

(a) Architecture de YOLOv8

YOLOv8 utilise un backbone similaire à celui de YOLOv5 avec quelques modifications sur la couche CSPLayer, désormais appelée module C2f. Le module C2f (bottleneck partiel inter-étapes avec deux convolutions) combine des caractéristiques de haut niveau avec des informations contextuelles pour améliorer la précision de détection.[28]

YOLOv8 utilise un modèle sans ancrage avec une tête découplée pour traiter indépendamment les tâches d'objectivité, de classification et de régression. Cette conception permet à chaque branche de se concentrer sur sa tâche et améliore la précision globale du modèle. Dans la couche de sortie de YOLOv8, ils utilisent la fonction sigmoïde comme fonction d'activation pour le score d'objectivité, représentant la probabilité que la boîte englobante contienne un objet. Ils utilisent la fonction softmax pour les probabilités de classe, représentant les probabilités d'appartenance des objets à chaque classe possible. YOLOv8 utilise les fonctions de perte CIoU et DFL pour la perte de boîte englobante et l'entropie croisée binaire pour la perte de classification. Ces pertes ont amélioré les performances de détection d'objets, principalement lorsqu'il s'agit de petits objets.[28] L'architecture de YOLOv8 utilise un backbone CSPDarknet53 modifié. Le module C2f remplace la couche CSPLayer utilisée dans YOLOv5. Une couche SPPF (spatial pyramid pooling fast) accélère le calcul en regroupant les caractéristiques dans une carte de taille fixe. Chaque convolution est suivie d'une normalisation par lots et d'une activation SiLU. La tête est découplée pour traiter indépendamment les

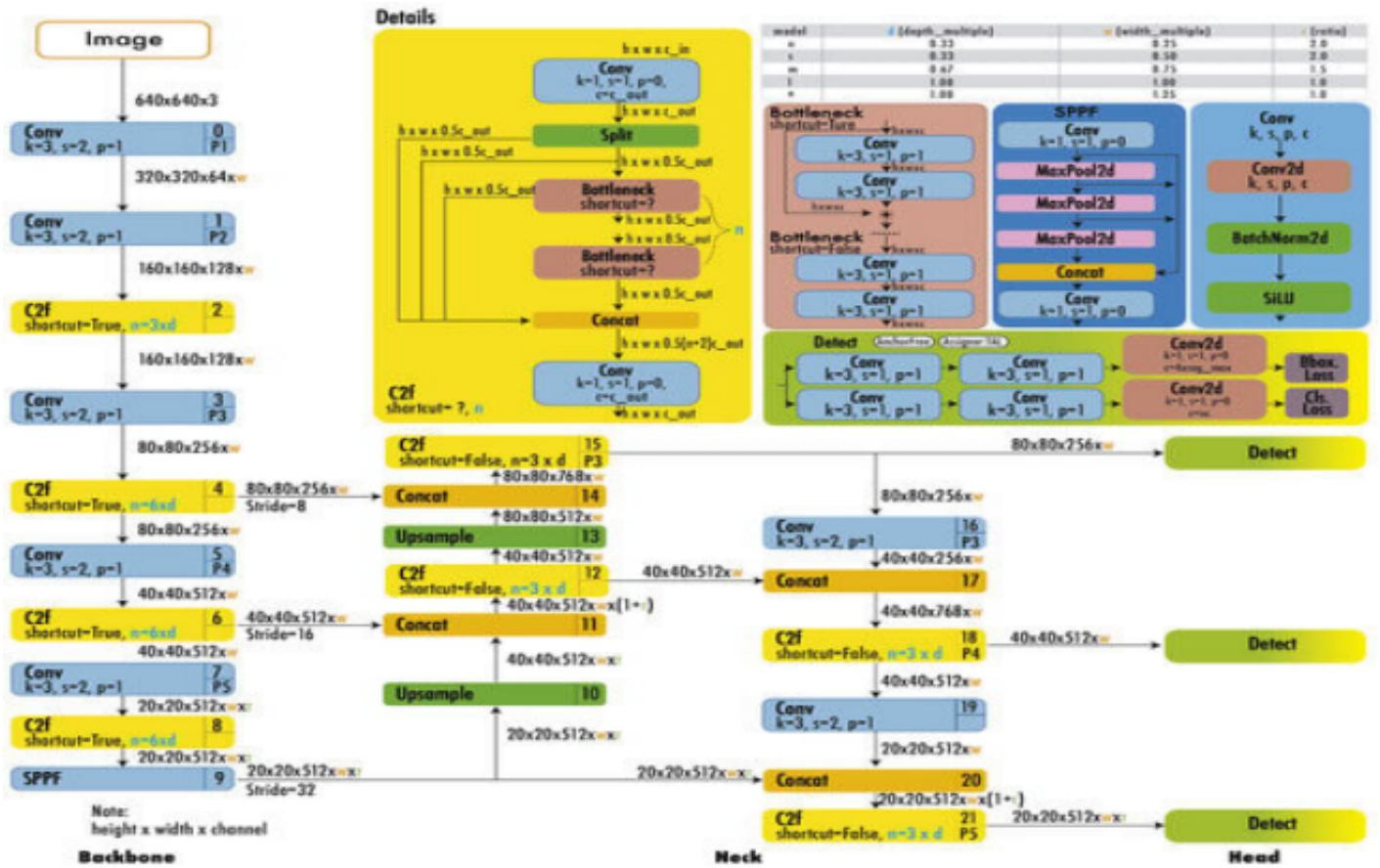


FIGURE 1.15 – YOLOv8 architecture.

tâches d'objectivité, de classification et de régression. Le diagramme est basé sur [28].

YOLOv8 propose également un modèle de segmentation sémantique appelé modèle YOLOv8-Seg. Le backbone est un extracteur de caractéristiques CSPDarknet53, suivi d'un module C2f au lieu de l'architecture de cou de YOLO traditionnelle. Le module C2f est suivi de deux têtes de segmentation, qui apprennent à prédire les masques de segmentation sémantique pour l'image d'entrée. Le modèle possède des têtes de détection similaires à YOLOv8, composées de cinq modules de détection et d'une couche de prédiction. Le modèle YOLOv8-Seg a obtenu des résultats de pointe sur différents benchmarks de détection d'objets et de segmentation sémantique tout en maintenant une grande vitesse et efficacité [28].

YOLOv8 peut être exécuté à partir de l'interface en ligne de commande (CLI), ou il peut également être installé en tant que package PIP. De plus, il est livré avec plusieurs intégrations pour l'étiquetage, l'entraînement et le déploiement.

Évalué sur le test-dev 2017 de l'ensemble de données MS COCO, YOLOv8x a atteint un AP de 53,9 % avec une taille d'image de 640 pixels (contre 50,7 % de YOLOv5 sur la même taille d'entrée) avec une vitesse de 280 FPS sur un NVIDIA A100 et TensorRT [28].

(b) Avantages et inconvénients de YOLO

YOLO offre une vitesse d'exécution significativement plus rapide que les autres algorithmes de détection d'objets, avec une cadence de 45 images par seconde. Cependant, l'algorithme YOLO présente des limitations lorsqu'il s'agit de détecter de petits objets dans une image, ce qui peut poser des problèmes dans certains scénarios, par exemple, la détection d'un groupe d'oiseaux.[\[20\]](#)

Conclusion

En conclusion, l'avènement de YOLO a marqué une avancée significative dans le domaine de la détection d'objets, offrant une approche novatrice et efficace pour localiser et classifier des objets dans des images en temps réel. À travers les différentes versions de YOLO, nous avons pu observer une amélioration constante de la précision et de la vitesse de détection, ouvrant ainsi la voie à un large éventail d'applications dans divers domaines, de la surveillance à la conduite autonome. Cependant, malgré ses avantages, YOLO présente également des inconvénients, tels que des difficultés à détecter de petits objets. Pour l'avenir, nous anticipons une évolution continue de YOLO, avec une intégration croissante de techniques de pointe et une adaptation à une gamme plus large de tâches et de plates-formes matérielles. En définitive, YOLO continuera de jouer un rôle crucial dans le développement de l'IA et de la vision par ordinateur.

Dans le prochain chapitre, nous plongerons dans l'étude des différentes maladies affectant les abeilles, en explorant leurs causes, leurs traitements potentiels, ainsi que les défis associés à leur surveillance et leur gestion.

Chapitre 2

Chapitre 02 : La santé des abeilles

Introduction

Les abeilles, en particulier l'abeille domestique (*Apis mellifera*), jouent un rôle fondamental dans l'écosystème en tant que principaux pollinisateurs. Elles sont responsables de la pollinisation de nombreuses plantes à fleurs, y compris une grande variété de cultures agricoles. Selon l'Organisation des Nations Unies pour l'alimentation et l'agriculture (FAO), environ 75% des cultures vivrières mondiales dépendent au moins en partie de la pollinisation par les insectes, et les abeilles en sont les principaux acteurs. Ce service écosystémique est essentiel non seulement pour la production alimentaire, mais aussi pour la biodiversité, car il permet la reproduction de nombreuses espèces végétales.

Cependant, les abeilles sont confrontées à de multiples menaces qui compromettent leur santé et leur survie. Parmi celles-ci, les maladies des abeilles jouent un rôle prépondérant. Ces maladies peuvent être causées par des bactéries, des virus, des champignons et des parasites, chacun ayant des impacts variés sur les colonies. Les apiculteurs, les chercheurs et les environnementalistes s'inquiètent de plus en plus des effets cumulés de ces maladies, aggravés par d'autres facteurs tels que l'utilisation des pesticides, la perte d'habitat et le changement climatique.

L'étude des maladies des abeilles est cruciale pour plusieurs raisons. Tout d'abord, comprendre les agents pathogènes et les conditions qui favorisent leur propagation peut aider à développer des stratégies de prévention et de traitement plus efficaces. Ensuite, une meilleure gestion des maladies des abeilles peut améliorer la santé des colonies, augmentant ainsi leur résilience face à d'autres stress environnementaux. Enfin, protéger les abeilles des maladies contribue directement à la sécurité alimentaire et à la conservation de la biodiversité.

Ce chapitre vise à fournir une vue d'ensemble détaillée des principales maladies affectant les abeilles. Il explorera les agents pathogènes spécifiques responsables de ces maladies, les symptômes observés dans les colonies infectées, les méthodes actuelles de prévention et de traitement, ainsi que les meilleures pratiques de gestion des ruches pour minimiser les risques d'infection. En comprenant mieux ces maladies et leurs impacts, les apiculteurs peuvent prendre des mesures plus proactives pour protéger leurs abeilles, assurant ainsi la pérennité des populations d'abeilles et la continuité de leurs services écosystémiques indispensables.

2.1 Les maladies des abeilles

2.1.1 Le varroa destructor

Le parasite *Varroa destructor* est omniprésent dans presque toutes les colonies d'abeilles, un parasite rouge foncé de forme ovale mesurant environ 1 mm sur 1,5 mm, affecte le couvain juste avant et après le scellement. Sa surface est couverte de poils fins et il possède huit pattes situées à l'avant de son corps. Il est actuellement impossible de l'éradiquer avec les méthodes et produits disponibles. Ce parasite infeste les larves de faux-bourçons et d'ouvrières juste avant l'operculation. La reproduction du varroa se déroule exclusivement dans le couvain operculé. Les acariens se nourrissent de l'hémolymphe des larves et des abeilles, entraînant une perte significative d'hémolymphe, un affaiblissement général, des malformations chez les larves et les jeunes abeilles, une susceptibilité accrue aux maladies et une propagation de germes infectieux (virus et bactéries).

Les colonies d'abeilles peuvent tolérer un certain niveau d'infestation sans problème. Cependant, si l'infestation dépasse un seuil critique, les symptômes de la varroase apparaissent et les colonies commencent à dépérir. Une colonie peut mourir en l'espace d'un à trois ans après la première infestation.[13]



FIGURE 2.1 – Varroa



FIGURE 2.2 – Varroa sur l'abeille (Apiculteur.ch)

1. Symptômes :

Les symptômes cliniques de la varroase affectent à la fois le couvain, les abeilles et la colonie dans son ensemble :

- Couvain irrégulier et clairsemé.
- Défauts de métamorphose et de pigmentation.
- Présence d'acariens dans le couvain et sur les abeilles.

- Jeunes abeilles et faux-bourçons déformés et sous-développés, notamment avec des abdomens raccourcis et des malformations des ailes.
- Développement lent de la colonie
- Rapport défavorable entre le nombre d'abeilles et le couvain
- Colonies désertées
- Symptômes d'infections secondaires
- Mort des colonies.[13]



FIGURE 2.3 – Abeille ouvrière adulte infectée par le virus des ailes déformées

2. Diagnostic

Les chercheurs et les apiculteurs utilisent diverses méthodes pour surveiller les varroas, parasites des abeilles mellifères adultes et du couvain. Voici quelques techniques courantes :

(a) Comptage par chute naturelle :

Cette méthode consiste à placer un plancher grillagé sous la ruche avec un plateau de comptage en dessous. Les varroas morts tombent sur le plateau (appelé lange). Il suffit de compter le nombre de varroas présents pour estimer l'infestation, souvent exprimée en nombre de varroas par jour.

(b) **Acaricide :**

La chute totale des acariens (CTA) peut être déterminée en appliquant un acaricide, efficace à plus de 95%. Cette technique est recommandée pour les colonies sans couvain, car les acaricides n'agissent pas à l'intérieur des cellules operculées. Bien qu'aucun traitement ne tue 100% des acariens, cette méthode est une référence pour évaluer l'efficacité des méthodes de détection de la varroase. Une chute artificielle des acariens peut également être provoquée en saupoudrant du sucre en poudre sur les rayons de la ruche ouverte, avec un effet similaire au lavage des abeilles.

(c) **Méthode du sucre :** Cette méthode simple et non destructive, appelée « sucre frappé », consiste à saupoudrer du sucre en poudre sur les abeilles vivantes. Le sucre obstrue les coussinets tarsiens des varroas, les faisant se détacher des abeilles. Cette technique ne nuit pas aux abeilles, même appliquée fréquemment.

(d) **Kits de lavage :** Pour un lavage à l'alcool, on prélève environ 300 abeilles (1/2 tasse) sur au moins trois cadres de la chambre à couvain, où les varroas sont les plus nombreux. Les abeilles sont collectées en inclinant le cadre à 30 degrés pour les faire tomber dans un contenant. Comme cette méthode tue les abeilles prélevées, il faut veiller à ne pas inclure la reine.[13]

3. **Traitement**

Les acaricides toxiques sont utilisés pour éliminer les acariens, mais ils peuvent également avoir des effets secondaires néfastes sur les abeilles et les colonies entières. Par conséquent, il est crucial d'appliquer ces acaricides à la dose efficace la plus faible possible pour minimiser l'accumulation de résidus chimiques et de leurs sous-produits dans les abeilles, le miel et la cire, ainsi que pour ralentir le développement de la résistance aux acaricides. Les traitements sont toujours effectués à la fin de la saison apicole, après la récolte. Actuellement, le traitement le plus simple, le plus efficace et le plus couramment utilisé est celui des lanières APIvar® à base d'amitrazé.[13]

4. **Prévention** Le parasitisme par les varroas peut souvent avoir des conséquences graves. Pour mener une lutte efficace contre cette menace, la première étape essentielle est le « dépistage » précoce afin de déterminer le moment optimal pour traiter et ainsi préserver ses ruches. Une surveillance « rigoureuse » des colonies est donc primordiale. L'administration des traitements doit être prudente pour minimiser les impacts négatifs sur les ruches. En cas de suspicion de la maladie, il est crucial de contacter immédiatement le vétérinaire officiel.[13]

2.1.2 La loque américaine

Les abeilles font face à une menace sérieuse avec la maladie bactérienne mortelle appelée loque américaine (AFB), causée par les larves du genre *Paenibacillus*, des bactéries qui forment des spores. Cette maladie affecte principalement le couvain, les larves âgées de moins de 24 heures étant particulièrement sensibles à l'infection. La loque américaine est considérée comme l'un des agents infectieux les plus préoccupants pour les abeilles domestiques. En raison de son impact socio-économique significatif, elle a des répercussions négatives importantes sur le commerce international.[13]



FIGURE 2.4 – Loque Américain (Apistory.fr)

1. Symptômes

- Peut voir la population de travailleurs de la colonie diminuer
- Peut avoir une colonie agitée et / ou agressive
- Odeur nauséabonde et pourrie (par rapport à la viande en décomposition ou au poulailler sulfureux)
- Motif de couvain irrégulier et / ou tacheté sur le cadre
- Bouchon scellé de cellules de couvain perforé, gras et / ou foncé
- Bouchage des cellules de couvain scellées coulées
- Humidité sur le couvercle du couvain scellé
- Larves de couleur marron café situées au bas de la cellule
- La larve collante au cordage reste au moins 2 cm de longueur lorsqu'elle est tirée hors de la cellule
- Larves de couleur brun café à noir durcies en «écailles» foncées, situées au bas des cellules

- difficile à éliminer et peut être fluorescent lorsqu'il est éclairé lors de l'utilisation d'une source de lumière UV
- Nymphes mortes de couleur marron café avec langue saillante.[11]

2. Cycle de vie et biologie du la loque :

La loque américaine passe par deux stades critiques dans son cycle infectieux. Initialement, les spores infectent les parties buccales des abeilles ouvrières et contaminent la nourriture destinée aux jeunes larves d'abeilles mellifères âgées de moins de deux jours. Ces spores se transforment alors en bactéries végétatives et se multiplient rapidement à l'intérieur des tissus des larves, générant des milliards de nouvelles spores.

Une fois que les alvéoles sont operculées, le couvain infecté meurt et se métamorphose en une substance gélatineuse nommée "écaille de loque américaine". Cette écaille s'attache fermement au fond de l'alvéole, devient progressivement noire, dure et adhère aux parois. Ce processus crée un potentiel élevé d'infection. Lorsque les ouvrières retirent ces écailles des rayons, elles contaminent leurs parties buccales, propageant ainsi l'infection à d'autres couvains en développement. Cela affaiblit la vigueur et la santé de la colonie, réduisant sa population et la rendant vulnérable aux attaques des colonies voisines. Les abeilles pillent alors les réserves de miel des colonies affaiblies, transportant également un grand nombre de spores infectieuses de la loque américaine vers leurs propres colonies saines. Cela entraîne une nouvelle contamination des couvains dans les nouvelles colonies, perpétuant le cycle de propagation de la maladie.[13]

3. **Traitement** Il existe plusieurs méthodes peuvent être employées en fonction de la maladie et de son stade de progression.

(a) **Destruction des colonies infectées** : Cette mesure est la plus efficace pour contrôler la propagation de la loque américaine et de son agent pathogène par dérive et pillage. Il est recommandé d'agir tôt le matin ou tard le soir lorsque toutes les abeilles sont dans la ruche. Après cette opération, il est crucial de nettoyer minutieusement les gants, le lève-cadres et l'enfumeur. Les ruches infectées doivent être fermées et les colonies touchées doivent être éliminées puis détruites par combustion.

(b) **Traitement antibiotique** :

Bien que des antibiotiques comme la tylosine, la lincomycine et l'oxytétracycline puissent atténuer les symptômes associés à la loque américaine, aucun traitement antibiotique n'est capable d'éliminer toutes les spores d'une ruche infectée. De plus, la présence persistante de résidus antibiotiques dans le miel a conduit de nombreux pays européens à interdire leur

utilisation, malgré leur usage courant aux États-Unis et au Canada. La résistance croissante de *Paenibacillus larvae* à ces antibiotiques est également préoccupante.

(c) **Transvasement :**

Pour les colonies non cliniquement atteintes par la loque américaine ou celles légèrement affectées mais capables de guérir, le transvasement peut être envisagé. Cette méthode implique la destruction de tous les rayons de la ruche, y compris ceux contenant du couvain, avec conservation uniquement des abeilles adultes. Tout le matériel en contact avec les colonies infectées doit être soigneusement nettoyé et désinfecté, idéalement par incinération à l'aide d'un chalumeau ou par nettoyage à chaud avec de la soude caustique (6%). En l'absence de ces possibilités, le matériel doit être détruit. Toutes ces opérations doivent être supervisées par les autorités sanitaires vétérinaires. Le miel des colonies malades peut contenir des spores, rendant son utilisation impossible pour le nourrissage des abeilles, bien qu'il reste sûr pour la consommation humaine en raison de la non-toxicité des spores pour l'homme. [13]

2.1.3 Le *Nosema ceranae*

La nosérose est une maladie des abeilles causée par les parasites *Nosema apis* ou *Nosema ceranae*, qui se multiplient dans les cellules intestinales de l'insecte. *Nosema apis* est une infection connue et étudiée depuis presque un siècle. En revanche, les effets de *Nosema ceranae* sur l'abeille européenne sont plus récents et moins documentés, initialement observés chez l'abeille asiatique. [13]



Normale



Malade

FIGURE 2.5 – *Nosema*

1. Symptômes :

- Traces de déjections sans forme, devant ou à l'intérieur de la ruche (à ne pas confondre avec les taches brun foncé, allongées, qui sont produites par les abeilles d'hiver atteintes de dysenterie)
- Abeilles disposées en « soleil », avec contact trophalactique (langue)
- Abeilles à l'abdomen gonflé
- Abeilles mortes devant la ruche
- Excitation normale au trou de vol (encombrement, houspillages)
- Abeilles incapables de voler, accrochées aux brins d'herbe
- Abeilles tremblantes, parfois avec les ailes en position écartée
- Colonie faible, plus ou moins dépeuplée.[13]

2. **Cycle de vie du nosema** : Le cycle biologique est complexe et peut varier de 7 à 10 jours en cas d'infection grave, selon les conditions environnementales. Il comporte deux phases distinctes :

(a) **Stade amiboïde** :

C'est la phase végétative et reproductrice du parasite, où il se divise activement à l'intérieur des cellules intestinales de l'abeille.

(b) **Stade de spore** :

Cette phase est passive et caractérisée par la résistance du parasite. Les spores jouent un rôle crucial dans la conservation et la propagation du parasite dans l'environnement extérieur.[11]

3. **Diagnostic** : Le diagnostic de la nosérose est complexe en raison de l'absence de symptômes caractéristiques spécifiques, mais plutôt d'un ensemble de symptômes qui peuvent se manifester de manière aléatoire chez toutes les abeilles adultes (reines, ouvrières et faux bourdons). On distingue deux types de diagnostic :

(a) **Diagnostic clinique** :

Il implique l'examen de l'intestin des abeilles. Habituellement, l'intestin des abeilles infectées est de couleur blanche, tandis que celui des abeilles saines est brun-rouge en raison de la présence de pollen. Pour effectuer cet examen, on coupe la tête des abeilles mortes et on tire sur l'abdomen pour mettre en évidence l'intestin. Cependant, le diagnostic clinique seul n'est pas suffisant et doit être confirmé par des méthodes de laboratoire.

(b) **Diagnostic en laboratoire** :

Il repose sur la détection des spores au microscope. Il est important de noter que la simple

présence de spores n'indique pas nécessairement la maladie. Pour un diagnostic précis de nosérose, il est nécessaire de compter les spores. Cela implique la dilution de 30 abdomens d'abeilles dans du sérum physiologique, suivie de l'examen au microscope entre lame et lamelle. D'autres critères peuvent également être nécessaires pour confirmer la présence de la maladie.[11]

4. **Traitement :**

Le seul traitement actuellement disponible pour la nosérose est l'antibiotique fumidil (fumagilline), qui doit être mélangé avec du sirop de sucre et administré à la colonie. Il est important de noter que la fumagilline n'a pas de Limite Maximale de Résidus (LMR), ce qui signifie qu'aucun résidu n'est toléré dans le miel. Actuellement, son utilisation est interdite en Europe en raison de cette restriction stricte. La fumagilline est extraite du champignon *Aspergillus fumigatus* et agit sur les différentes formes de multiplication de *Nosema*, bien qu'elle ne soit pas efficace contre les spores.

Des recherches sont en cours sur d'autres traitements naturels tels que le protofil à base de plantes et l'ApiHerb en Italie. De plus, il est possible de détruire les spores en chauffant l'équipement et les outils apicoles à une température d'au moins 60°C pendant 15 minutes. Les cadres peuvent être stérilisés en les chauffant à 49°C pendant 24 heures.[13]

2.1.4 **Sacbrood virus**

Le SBV (Sacbrood Bee Virus) est une maladie contagieuse affectant les abeilles mellifères, causée par un virus qui cible généralement le couvain operculé. Cette infection peut entraîner des pertes significatives de prénymphe, ce qui peut affaiblir considérablement la colonie. Les prénymphe atteintes par ce virus montrent un aspect distinctif en forme de sac. Bien que les abeilles adultes infectées ne présentent aucun symptôme (étant des porteurs sains), elles jouent un rôle de réservoir pour le virus.[13]

1. **Symptômes :**

- Larves desséchées de couleur marron à noir, aplaties en forme de barque, facilement détachables de leur alvéole.
- Larves ou nymphes sorties par les abeilles devant le trou de vol.
- Couleur anormale des larves (sous couvain ouvert ou operculé) allant de jaune à marron.
- Test de l'allumette négatif : la larve morte n'est pas filante.
- Couvain en mosaïque.
- Opercules aplatis ou affaissés.



FIGURE 2.6 – Sacbrood (Beeaware.org)

- Opercules percés et/ou déchirés (dans certains cas particuliers).[13]

2. Cycle biologique du SBV :

Nous pouvons distinguer trois stades dans cette maladie :

(a) **Étape initiale :**

Cette phase ressemble à d'autres affections du couvain. Les larves infectées perdent leur couleur « blanc nacré » et prennent une teinte légèrement altérée. On peut observer des larves droites, décolorées, dans des cellules désoperculées par les abeilles nettoyeuses.

(b) **Développement de la maladie :**

Les larves changent progressivement de couleur, passant du jaune au gris puis au brun (similaire à la loque américaine, mais sans adhérence, ni fil, ni odeur). Le virus se multiplie, faisant gonfler les larves et leur donnant l'apparence de « sacs » remplis de liquide, qui sont très contagieux.

(c) **Stade avancé :**

Les larves noircissent en commençant par les extrémités. Elles se dessèchent jusqu'à former une écaille en forme de barque, qui se détache facilement de l'alvéole. À ce stade, ces écailles ne sont plus contagieuses.[13]

3. Diagnostic :

Un prélèvement de couvain de 10 cm sur 10 contenant au moins 15 prénymphe atteintes peut être envoyé à un laboratoire agréé pour une analyse de particules virales par PCR. Cette analyse est à la charge de l'apiculteur. Il est essentiel de contacter le laboratoire référent de son département pour vérifier ses capacités (détention d'un agrément pour cette recherche), ainsi que les conditions

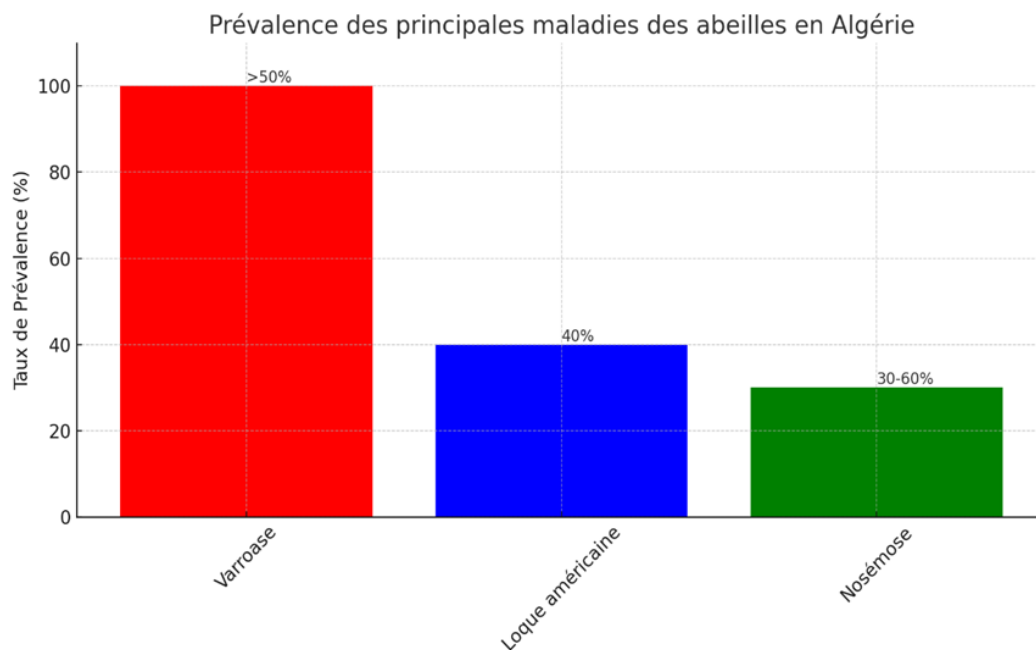


FIGURE 2.7 – les taux de prévalence des principales maladies des abeilles en Algérie (<http://hdl.handle.net/123456789/2197>)

d'expédition et le coût de l'analyse avant l'envoi.[13]

4. **Traitement :**

Cette maladie est généralement bien contrôlée par les abeilles nettoyeuses, avec des guérisons spontanées lorsque les conditions redeviennent favorables. Cependant, elle peut réapparaître en présence d'autres infections affaiblissant les colonies.

- Si la maladie est peu étendue (quelques cellules) : Contrôler la colonie pour détecter d'autres pathologies, nourrir avec du sirop 50/50, surveiller et éventuellement remplacer la reine.
- Si la maladie est étendue : Retirer les cadres de couvain atteints et les faire fondre ou les brûler, réunir les colonies faibles ou les supprimer.
- Si plusieurs cadres sont atteints : Transférer la colonie dans une ruche propre avec des cadres gaufrés (comme pour la loque américaine), désinfecter tout le matériel apicole, y compris les lève-cadres, à la flamme.[13]

2.2 Les causes des maladies des abeilles

Les maladies des abeilles peuvent avoir plusieurs causes, notamment :

2.2.1 Les pesticides

certaines pesticides peuvent être toxiques pour les abeilles et les affaiblir.

1. Types de pesticides :

Les insecticides, les herbicides et les fongicides peuvent tous être toxiques pour les abeilles.

2. Modes d'action :

Les pesticides peuvent tuer les abeilles par contact, par ingestion ou par inhalation.

3. Effets :

Les pesticides peuvent affaiblir les abeilles et les rendre plus sensibles aux maladies.

2.2.2 Le changement climatique

Le changement climatique influence les conditions environnementales, favorisant ainsi la propagation des maladies qui affectent les abeilles. Voici comment :

1. Augmentation de la température :

L'élévation des températures peut accélérer la reproduction des parasites comme le *Varroa destructor* et des agents pathogènes comme *Nosema* spp. Des températures plus chaudes prolongent également la période active de ces menaces, augmentant leur impact sur les colonies d'abeille.

2. Changements de précipitations :

Les changements de précipitations peuvent affecter la disponibilité de nourriture et d'eau pour les abeilles.

3. Événements météorologiques extrêmes :

Les événements météorologiques extrêmes, tels que les sécheresses et les inondations, peuvent tuer les abeilles et détruire leurs colonies.

2.2.3 La perte d'habitat

la perte d'habitat peut réduire la quantité de nourriture disponible pour les abeilles et les rendre plus sensibles aux maladies.

2.2.4 Déforestation

La déforestation détruit les habitats naturels des abeilles, éliminant de nombreuses plantes mellifères et réduisant les zones de nidification. Cela force les abeilles à parcourir de plus grandes distances pour trouver de la nourriture, augmentant leur stress et leur exposition aux pesticides..

2.2.5 Urbanisation

L'urbanisation réduit les terres disponibles pour les abeilles, remplaçant les espaces verts par des constructions et des infrastructures. Les zones urbaines offrent souvent moins de diversité florale et une plus grande concentration de polluants, ce qui impacte négativement la santé des abeilles.

2.2.6 Utilisation de pesticides :

L'utilisation intensive de pesticides dans l'agriculture et l'aménagement paysager tue non seulement les abeilles directement, mais détruit également les plantes mellifères dont elles dépendent. La réduction des sources de nourriture affaiblit les colonies et les rend plus sensibles aux infections et aux parasites.[1][2]

2.3 Les conséquences des maladies des abeilles

Les maladies des abeilles peuvent avoir plusieurs conséquences, notamment :

- **La réduction de la pollinisation** : la réduction du nombre d'abeilles peut entraîner une réduction de la pollinisation des cultures, ce qui peut avoir un impact négatif sur la production agricole.
- **La perte de biodiversité** : la pollinisation des abeilles est essentielle à la reproduction de nombreuses plantes sauvages. La réduction du nombre d'abeilles peut donc entraîner :
 - La disparition de certaines espèces végétales, ce qui peut avoir un impact négatif sur les écosystèmes.
 - La perturbation des chaînes alimentaires, car de nombreux animaux sauvages dépendent des plantes pollinisées par les abeilles pour se nourrir.
- **L'augmentation des prix des denrées alimentaires** : la réduction de la production agricole due à la diminution de la pollinisation peut entraîner :
 - Une hausse des prix des fruits et légumes frais.
 - Une pénurie de certains produits alimentaires.
 - Des problèmes de sécurité alimentaire, en particulier dans les pays en développement.[3]

2.4 Lutte contre les maladies des abeilles

Il existe un certain nombre de mesures qui peuvent être prises pour lutter contre les maladies des abeilles, notamment :

- **L'utilisation de pesticides non toxiques pour les abeilles :**

- Choisir des pesticides sélectifs qui ciblent les parasites et les agents pathogènes nuisibles sans nuire aux abeilles.
- Appliquer les pesticides aux moments où les abeilles ne sont pas actives, par exemple en fin de journée ou pendant la nuit.
- Planter des haies et des bandes fleuries autour des champs cultivés pour fournir aux abeilles une source de nourriture et un refuge contre les pesticides.

- **La mise en place de pratiques agricoles durables :**

- Pratiquer la rotation des cultures pour réduire la pression des parasites et des maladies.
- Maintenir la diversité végétale dans les paysages agricoles pour fournir aux abeilles une variété de sources de nourriture.
- Encourager la présence de fleurs sauvages et de plantes mellifères dans les paysages agricoles.

- **La plantation de fleurs mellifères :**

- planter des fleurs riches en nectar et en pollen à proximité des ruchers pour fournir aux abeilles une source de nourriture abondante et de qualité.
- Choisir des variétés de fleurs qui fleurissent à différents moments de l'année afin de fournir aux abeilles une source de nourriture continue.
- Planter des fleurs indigènes adaptées au climat local.

- **L'éducation des apiculteurs :**

- former les apiculteurs aux meilleures pratiques de gestion des maladies afin qu'ils puissent identifier les signes de maladie et prendre des mesures correctives le plus tôt possible.
- Fournir aux apiculteurs des informations sur les maladies des abeilles, les traitements disponibles et les moyens de prévenir la propagation des maladies.
- Encourager la collaboration entre les apiculteurs et les chercheurs pour développer de nouvelles méthodes de lutte contre les maladies des abeilles.^[3]

Conclusion

La santé des abeilles est cruciale pour la biodiversité et la sécurité alimentaire. En prenant des mesures pour combattre les maladies des abeilles, nous protégeons ces insectes pollinisateurs essentiels, garantissant ainsi une production alimentaire durable pour les générations futures.

Le chapitre suivant explore en détail l'utilisation de l'intelligence artificielle (IA) pour détecter la maladie du varroa qui est un parasite destructeur qui affaiblit les colonies d'abeilles et peut mener à leur effondrement. Grâce à des techniques avancées d'IA, il est possible de détecter la présence de varroa de manière plus rapide et précise, permettant ainsi une intervention plus efficace et ciblée. Cette approche innovante représente un espoir significatif pour améliorer la gestion de la santé des abeilles et renforcer nos efforts de conservation.

Chapitre 3

Chapitre 03 : Système de détection des maladies d'abeilles

Introduction

La santé des abeilles est un élément fondamental de la stabilité de notre écosystème et de notre sécurité alimentaire. Malheureusement, les colonies d'abeilles sont de plus en plus menacées par diverses maladies et parasites, tels que le *Varroa destructor*, le syndrome d'effondrement des colonies (CCD), et les infections virales et bactériennes. La détection précoce et précise de ces maladies est essentielle pour protéger les abeilles et assurer leur rôle crucial en tant que pollinisateurs. Traditionnellement, la surveillance des ruches repose sur des inspections manuelles, qui peuvent être laborieuses et manquer de précision. Cependant, avec les avancées technologiques, les smart ruches dotées de capteurs intelligents et de plateformes IoT offrent de nouvelles perspectives pour la détection et la gestion des maladies des abeilles. Ces systèmes permettent une surveillance en temps réel des conditions de la ruche et de la santé des colonies, aidant les apiculteurs à intervenir rapidement et efficacement pour préserver leurs populations d'abeilles. Dans ce chapitre, nous allons explorer en détail comment l'intelligence artificielle (IA) peut révolutionner la détection des maladies des abeilles en particulier la maladie Varroa. Nous commencerons par un aperçu des principes de base de l'IA, notamment l'apprentissage automatique et le traitement d'images, et comment ces technologies peuvent être appliquées à l'apiculture.

3.1 Etat de l'art

Les visites florales, certaines d'entre elles exclusivement [14]. Néanmoins, les colonies d'abeilles à travers le monde font face à de nombreux problèmes liés aux parasites tels que l'acarien *Varroa* (*Varroa destructor*), les maladies virales et le syndrome d'effondrement des colonies, qui, s'ils ne sont pas contrôlés, pourraient causer des perturbations dans les chaînes d'approvisionnement alimentaire et d'énormes pertes économiques, sans parler des autres dommages environnementaux connexes. Ce problème touche à la fois les pollinisateurs sauvages et élevés avec des pertes attendues de 3 % à 8 % de la production agricole en cas d'absence totale de pollinisateurs animaux.[6] Ces dernières années, les techniques modernes d'apprentissage automatique se sont révélées très efficaces pour traiter des données multidimensionnelles comportant une grande quantité d'informations et de dépendances. Ces techniques, souvent utilisées dans les applications de vision par ordinateur, ont permis des avancées dans des problèmes jusque-là difficiles à résoudre, tels que les tâches de classification, de reconnaissance ou d'inspection, et elles pourraient être appliquées avec succès aux applications d'inspection automatisée des abeilles. Par exemple, une vue d'ensemble complète des méthodes de comptage des abeilles développées au cours du siècle dernier est présentée dans[19],mais malgré la haute qualité de cet article, les méthodes basées sur l'apprentissage automatique et d'autres applications ne sont mentionnées que

brièvement. Dans le domaine de la détection des acariens Varroa, les chercheurs visent généralement à détecter la présence de ce parasite par des systèmes basés sur l'image. Étant donné que ce parasite cause de grandes pertes de colonies d'abeilles dans le monde entier, le développement et le déploiement de cette application pourraient également conduire à d'importantes économies économiques et de populations d'abeilles, à un traitement mieux planifié et à une inspection continue de l'état de santé des colonies d'abeilles. Il existe également des suggestions d'un système capable de détruire l'acarien Varroa directement sur le corps de l'abeille infestée [10]. Ce système mentionné est encore immature et n'est pas prêt pour le déploiement, mais des efforts futurs pourraient le rendre pratique. Cependant, un problème des systèmes basés sur la vision est que la plupart des acariens Varroa se trouvent sur le couvain des abeilles ou sont cachés entre les segments abdominaux de l'abeille et sont donc difficilement visibles. Par conséquent, une méthodologie d'estimation du niveau d'infestation, comme celle présentée dans [9], doit être développée et testée de manière approfondie pour permettre une meilleure utilisation de cette application.

Pour illustrer l'approche classique de la vision par ordinateur utilisée pour la détection des acariens Varroa,[27] présente deux configurations expérimentales. Dans la première configuration, les abeilles sont séparées par des tunnels étroits avec des arrière-plans contrastés, ce qui simplifie la segmentation et l'analyse. Le dispositif est équipé d'un éclairage artificiel pour créer des conditions d'enregistrement appropriées. La deuxième configuration capture les abeilles en plein air avec une caméra placée au-dessus de la surface mesurée, ce qui accélère le processus de mesure mais complique le traitement ultérieur. La phase de classification est conçue pour différencier les abeilles saines sans acariens Varroa des abeilles infectées. Ce travail compare divers ensembles de caractéristiques avec trois modèles de classification différents (Naive Bayes, SVM et Random Forest), atteignant des précisions et des scores F1 supérieurs à 0,8 pour certaines combinaisons. [26] D'autres approches utilisent des méthodes conventionnelles d'apprentissage automatique, telles que l'analyse des moments de Legendre-Fourier dans divers espaces colorimétriques pour détecter les abeilles infectées par les acariens Varroa [18]. Une étude plus avancée utilise des réseaux de neurones convolutionnels (CNN) pour la détection des acariens Varroa, comme présenté dans [25], où une approche de segmentation sémantique avec DeepLabV3 est utilisée pour séparer les abeilles saines des abeilles infectées par les acariens Varroa. En outre, des systèmes de détection d'objets basés sur des réseaux de neurones convolutionnels sont appliqués à la détection des acariens Varroa, offrant des mesures en temps réel et des analyses rapides grâce à des temps d'inférence courts, comme décrit dans [8] [31]

Ces approches montrent le potentiel des techniques de vision par ordinateur et d'apprentissage automatique pour améliorer la gestion sanitaire des colonies d'abeilles en surveillant efficacement les in-

festations par les acariens Varroa et en facilitant des interventions ciblées et précises pour préserver la santé des abeilles et la productivité des colonies.

3.2 Problématique

Les abeilles jouent un rôle vital dans la pollinisation et la préservation de la biodiversité, mais elles sont actuellement menacées par des maladies telles que le varroa destructor. La précision et la rapidité de la détection de cette maladie sont cruciales pour protéger les populations d'abeilles et maintenir leur contribution essentielle à l'écosystème. Cependant, les méthodes traditionnelles de détection présentent des limitations importantes en termes de coût, de durée et de nécessité d'une expertise spécialisée, ce qui entrave la mise en place rapide de mesures correctives et augmente le risque de pertes massives de colonies. En l'absence de détection précoce, les conséquences peuvent être graves : effondrement des colonies, diminution de la production apicole, impact négatif sur la pollinisation des cultures et propagation de la maladie à d'autres ruchers. Face à ces défis pressants, il est crucial d'améliorer la détection du varroa destructor tout en rendant ce processus plus efficace et accessible aux apiculteurs. Une solution prometteuse inclut l'intégration de systèmes de détection offrant une communication en temps réel avec les apiculteurs. Cette fonctionnalité permettrait aux apiculteurs de recevoir des alertes immédiates sur les niveaux d'infestation dans leurs ruchers, facilitant ainsi une intervention rapide et ciblée pour limiter la propagation de la maladie.

3.3 Solution proposée

Nous proposons de concevoir un système de détection basé sur l'intelligence artificielle. Cette solution exploite des méthodes sophistiquées d'apprentissage automatique et de traitement d'images afin de détecter efficacement les symptômes du varroa destructor. En permettant une identification rapide et précise de ce parasite, notre système facilitera une réaction immédiate et adaptée des apiculteurs. En conséquence, la gestion des colonies d'abeilles sera optimisée, réduisant ainsi les risques de pertes significatives.

3.4 Environnement de travail

3.4.1 Spécifications de l'ordinateur portable

Nous avons travaillé dans notre projet avec un ordinateur portable ayant les caractéristiques suivantes :

- **DESKTOP-S43E77N**
- **Processeur** : AMD Ryzen 3 3250U with Radeon Graphics 2.60 GHz
- **Mémoire vive** : 4 Go
- **Stockage** : 1T HDD

3.4.2 Logiciels utilisés

1. Roboflow

Roboflow est une plate-forme de vision par ordinateur qui permet aux utilisateurs de créer des modèles de vision par ordinateur plus rapidement et avec plus de précision grâce à la fourniture de meilleures techniques de collecte de données, de prétraitement et de formation de modèles. Roboflow permet aux utilisateurs de télécharger des ensembles de données personnalisés, de dessiner des annotations, de modifier les orientations des images, de redimensionner les images, de modifier le contraste des images et d'effectuer une augmentation des données. Il peut également être utilisé pour former des modèles.

Roboflow dispose également d'un outil de conversion d'annotations universel qui permet aux utilisateurs de télécharger et de convertir des annotations d'un format à un autre sans avoir à écrire de scripts de conversion pour les ensembles de données de détection d'objets [29].

2. Google Collab

Colaboratory, ou "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation. Nous pouvons effectuer les opérations suivantes à l'aide de Google Colab :

- Écrire et exécuter du code en Python
- Créer/Télécharger/Partager des blocs-notes
- Importer/enregistrer des blocs-notes depuis/vers Google Drive
- Importer/publier des blocs-notes depuis GitHub

- Importer des ensembles de données externes
- Intégrer PyTorch, TensorFlow, Keras, OpenCV
- Service Cloud gratuit avec GPU gratuit [4]

3.4.3 Langage de programmation utilisé (Python)

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [24]

1. Bibliothèques utilisées

(a) **PyTorch** :

PyTorch est une bibliothèque d'apprentissage automatique pour Python utilisée principalement pour le traitement du langage naturel. Le logiciel open-source a été développé par les équipes d'intelligence artificielle [5].

(b) **OPENCV** :

OpenCV (Open Computer Vision) est une bibliothèque graphique. Elle est spécialisée dans le traitement d'images, que ce soit pour de la photo ou de la vidéo. Sa première version est sortie en juin 2000. Elle est disponible sur la plupart des systèmes d'exploitation et existe pour les langages Python, C++ et Java [7].

3.5 Entraînement à l'aide du YOLOv8

Au cours de cette partie, notre objectif sera de créer une base de données, d'entraîner notre modèle YOLOv8, puis de réaliser des tests afin de recueillir des statistiques sur le fonctionnement de notre système.

3.5.1 Création de base de données

La création de la base de données passe par plusieurs étapes qui sont cruciales pour assurer la qualité et l'utilité des données collectées qui sont :

1. Collecte d'images pour l'ensemble de données

Pour établir une base de données destinée à la détection du varroa, nous avons commencé par collecter un ensemble d'images illustrant différents aspects de l'infestation par le varroa destructor chez les abeilles. Nous avons pris des photographies de ruches et d'abeilles montrant des signes visuels distinctifs de l'infestation, comme la présence des acariens sur les abeilles ou dans les cellules de la ruche comme le montre la figure (3.1).



FIGURE 3.1 – Le varroa destructor chez les abeilles.

2. Anotation d'images

L'annotation d'image consiste à marquer précisément les signes visuels de l'infestation sur les photographies que nous avons prises. Cela comprend la délimitation des acariens sur les abeilles et dans les cellules des ruches à l'aide de bounding boxes, ainsi que l'étiquetage des zones affectées pour identifier clairement les symptômes de l'infestation. Ces annotations sont cruciales pour entraîner les modèles d'apprentissage automatique à reconnaître et à classifier efficacement les cas de varroa, facilitant ainsi une détection précise et rapide de cette menace pour les colonies d'abeilles. Nous avons utilisé le logiciel Roboflow pour cette tâche, comme illustré par quelques images annotées dans la figure (3.2) . De plus, nous avons appliqué l'augmentation des données

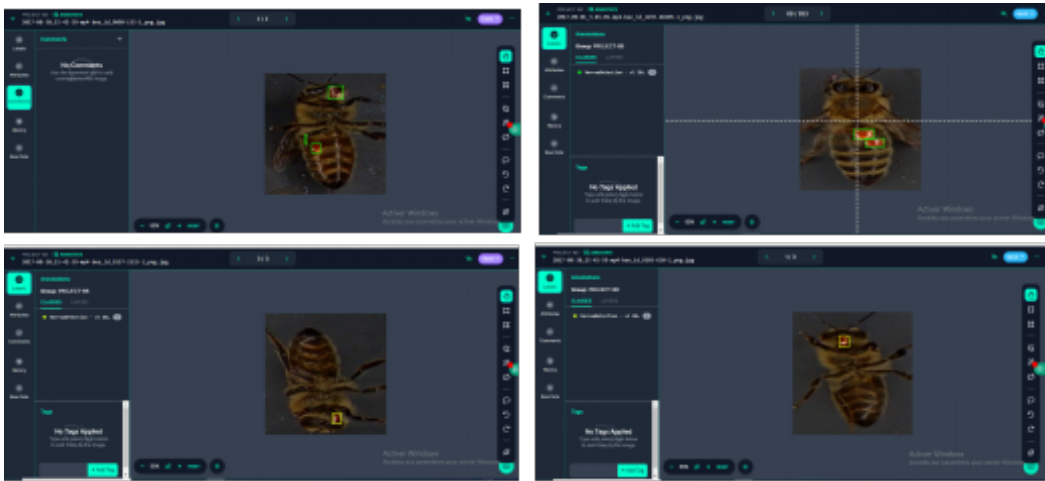


FIGURE 3.2 – Anotation d'images sur Roboflow.

(data augmentation) avec Roboflow, utilisant des techniques telles que le flip horizontal et vertical , Grâce à ces augmentations, nous avons réussi à augmenter le nombre d'images dans notre base de données de 910 images à 1820 images. Une fois l'augmentation terminée, nous avons exporté notre base de données au format YOLOv8 figure(3.3).

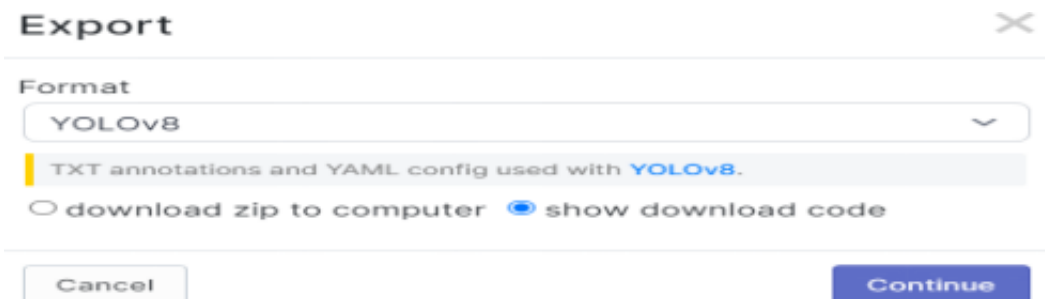


FIGURE 3.3 – Exportation de base de données.

3. Configuration de la base de données

Nous avons généré et divisé la base de données en 3 parties essentielles comme le montre la figure (3.4).

- 85% (1540 images) pour l'entraînement
- 10% (187 images) pour la validation
- 5% (93 images) pour le test

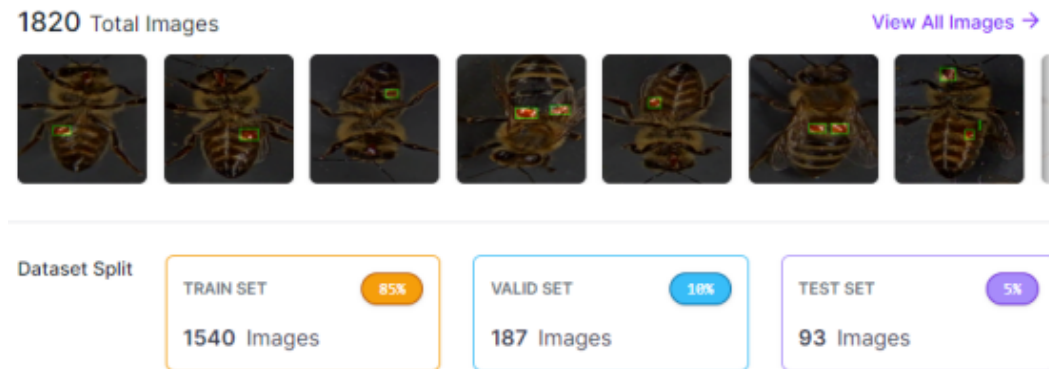


FIGURE 3.4 – Division de la base de données.

3.5.2 Entraînement

Dans cette section, nous aborderons les étapes d'entraînement ainsi que les résultats obtenus. Pour rendre le processus d'entraînement plus efficace et rapide, nous avons opté pour l'utilisation de Google Colab, en exploitant des ensembles de données et le référentiel YOLO V8 disponible sur GitHub. Le diagramme (3.5) ci-dessous illustre la séquence des étapes effectuées : La figure (3.5) illustre les différentes étapes à suivre pour entraîner et évaluer un modèle d'intelligence artificielle appelé YOLO v8 (You Only Look Once version 8). Voici une explication détaillée de chaque étape :

1. **Clone le dépôt yolo v8** : Cette première étape consiste à cloner (copier localement) le dépôt de code source contenant l'implémentation de l'algorithme YOLOv8 depuis un emplacement distant comme GitHub.
2. **Installer les dépendances** : Une fois le code source cloné, vous devez installer les dépendances logicielles nécessaires à l'exécution de YOLOv8, généralement en utilisant un gestionnaire de packages comme pip.
3. **Transférer la base de données** : À cette étape, vous devez fournir les données d'entraînement au modèle, généralement sous forme d'un ensemble d'images annotées contenant les objets à détecter voir la figure (3.6).

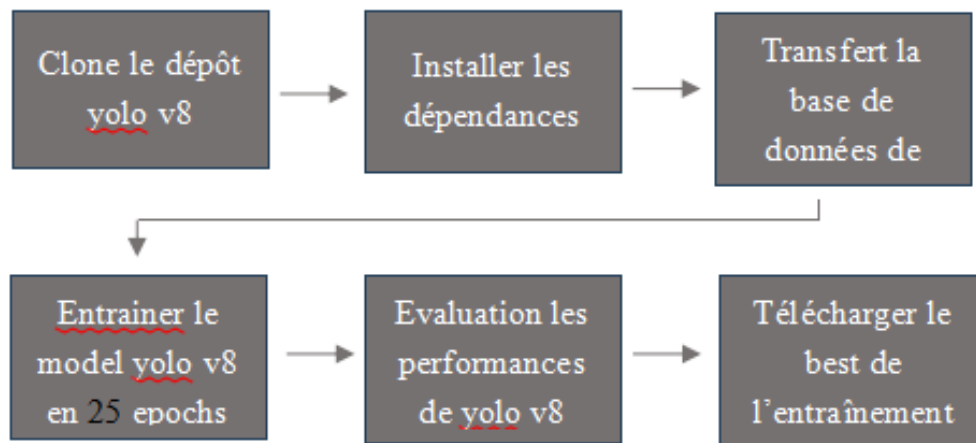


FIGURE 3.5 – Digramme des étapes d'entraînement et évaluation.

4. **Entraîner le modèle yolo v8** : Avec le code source, les dépendances installées et les données d'entraînement prêtes, vous pouvez lancer le processus d'entraînement du modèle YOLO v8 sur ces données, souvent sur 100 époques (25 cycles complets sur l'ensemble des données).
5. **Évaluation les performances de yolo v8** : Une fois l'entraînement terminé, vous pouvez évaluer les performances du modèle entraîné sur un ensemble de données de test distinct, en calculant des métriques comme la précision moyenne (mAP) sur les objets détectés.
6. **Télécharger le meilleur modèle** : Enfin, après avoir évalué les performances, vous pouvez télécharger les poids du meilleur modèle entraîné pour une utilisation future dans des applications de détection d'objets.

En résumé, ce diagramme décrit le pipeline complet pour cloner, configurer, entraîner, évaluer et déployer un modèle de détection d'objets YOLOv8 à l'aide de vos propres données d'entraînement.


```

$ yolo task=detect mode=train model=yolov8l.pt data=[dataset.location]/data.yaml epochs=25 imgsz=600 plots=True

--- /content
Downloading https://github.com/ultralytics/assets/releases/download/v8.0.9/yolov8l.pt to 'yolov8l.pt'...
100% 83.7M/83.7M [00:00<00:00, 267MB/s]
New https://docs.ultralytics.com/ available • update with 'pip install -u ultralytics'
ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8l.pt, data=[content/PROJECT-00-1/082A.yaml, epochs=25, patience=50, batchsz=16, imgsz=600, save=True, save_period=1, cache=False, dev
Downloading https://ultralytics.com/assets/serial.txt to '/root/.config/ultralytics/serial.txt'...
100% 755K/755K [00:00<00:00, 26.7MB/s]
2024-06-18 11:13:19.621894: I external/local_xla/xla/stream_executor/cuda/cuda_device.cc:924] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has
2024-06-18 11:13:19.522864: I external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:887] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has
2024-06-18 11:13:19.646812: I external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1135] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one
Overriding model.yaml rc=88 with rc=1

      from n  params  module  arguments
0         -1  3  1884  ultralytics.nn.modules.conv.Conv  [3, 84, 3, 2]
1         -1  3  73954 ultralytics.nn.modules.conv.Conv  [64, 128, 3, 2]
2         -1  3  279880 ultralytics.nn.modules.block.C2F  [128, 128, 3, True]
3         -1  3  295424 ultralytics.nn.modules.conv.Conv  [128, 256, 3, 2]
4         -1  8  2161248 ultralytics.nn.modules.block.C2F  [256, 256, 6, True]
5         -1  1  1160072 ultralytics.nn.modules.conv.Conv  [256, 512, 3, 2]
6         -1  8  8396800 ultralytics.nn.modules.block.C2F  [512, 512, 6, True]
7         -1  1  2360320 ultralytics.nn.modules.conv.Conv  [512, 512, 3, 2]
8         -1  3  4461548 ultralytics.nn.modules.block.C2F  [512, 512, 3, True]
9         -1  1  655594 ultralytics.nn.modules.block.SPPF  [512, 512, 5]
10        -1  1  0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
11        [-1, 6] 1  0  ultralytics.nn.modules.conv.Concat  [1]
12        -1  3  4723712 ultralytics.nn.modules.block.C2F  [1024, 512, 3]
13        -1  1  0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
14        [-1, 4] 1  0  ultralytics.nn.modules.conv.Concat  [1]
15        -1  3  1247744 ultralytics.nn.modules.block.C2F  [768, 256, 3]
16        -1  1  590336 ultralytics.nn.modules.conv.Conv  [256, 256, 3, 2]
17        [-1, 12] 1  0  ultralytics.nn.modules.conv.Concat  [1]
18        -1  3  8352640 ultralytics.nn.modules.block.C2F  [768, 512, 3]
19        -1  1  2360320 ultralytics.nn.modules.conv.Conv  [512, 512, 3, 2]

```

FIGURE 3.8 – Lancement d’entraînement.

3.6 Résultats

3.6.1 Précision

L’entraînement du modèle YOLOv8 a permis d’obtenir des résultats impressionnants en termes de précision et de temps d’exécution. Après 25 époques, le modèle a atteint une précision élevée, démontrant sa capacité à détecter et classifier efficacement les objets dans les images annotées. Chaque époque a pris environ 2.18 min à s’exécuter, ce qui signifie que l’entraînement complet sur 25 époques a duré environ 54 min . Cette efficacité temporelle est essentielle pour permettre des itérations rapides et des ajustements du modèle, facilitant ainsi l’optimisation continue des performances. Les graphiques générés tout au long du processus d’entraînement ont montré une amélioration progressive des métriques de performance, indiquant que le modèle a bien convergé vers une solution optimale.

```

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
25/25   15.6G    1.572    0.6875   1.951     9          800: 100% 97/97 [01:59<00:00, 1.23s/it]
      Class  Images  Instances  Box(P)   R      mAP50  mAP50-95): 100% 6/6 [00:05<00:00, 1.03it/s]
      all    187     233      0.859   0.897   0.921   0.461

25 epochs completed in 0.917 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 87.7MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 87.7MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 43607379 parameters, 0 gradients, 164.8 GFLOPs
      Class  Images  Instances  Box(P)   R      mAP50  mAP50-95): 100% 6/6 [00:10<00:00, 1.75s/it]
      all    187     233      0.859   0.887   0.921   0.461

Speed: 0.7ms preprocess, 24.6ms inference, 0.0ms loss, 8.3ms postprocess per image
Results saved to runs/detect/train2

```

FIGURE 3.9 – Précision obtenue pour la classe Varroa

3.6.2 Pertes pendant l'entraînement et la validation

D'après le graphe obtenu dans la figure (3.10) de précision, il est clair que notre niveau de précision maximal a été atteint après environ 23 epochs. De plus, le graphe de la perte a diminué en fonction du nombre des epochs jusqu'à sa valeur minimale aux environ de 10 epochs. Ces résultats d'entraînement montrent bien que les performances obtenues sont satisfaisantes en utilisant le modèle YOLOV8.

1. **Box Loss (train/box_loss et val/box_loss) :**

- (a) **Description :** Mesure l'erreur entre les boîtes englobantes prédites par le modèle et les boîtes englobantes réelles. Cela inclut souvent des composantes telles que la distance entre les centres des boîtes, la différence de taille et la forme.
- (b) **Observation :** La courbe de train/box_loss diminue de 1.9 à environ 1.2, et la courbe de val/box_loss diminue de 3.0 à environ 1.0. Cela indique que le modèle devient de plus en plus précis dans la localisation des objets au fur et à mesure de l'entraînement et maintient cette précision sur les données de validation.

2. **Classification Loss (train/cls_loss et val/cls_loss) :**

- (a) **Description :** Mesure l'erreur entre les classes prédites par le modèle et les classes réelles. Souvent calculée avec des fonctions de perte comme l'entropie croisée.
- (b) **Observation :** La courbe de train/cls_loss passe de 2.0 à 0.5, et celle de val/cls_loss chute de 4.0 à 0.2. Cela montre que le modèle améliore sa capacité à classer correctement les objets, tant pendant l'entraînement que sur les données de validation.

3. **Distribution Focal Loss (DFL) (train/df_loss et val/df_loss) :**

- (a) **Description :** Une forme de perte focalisée qui accorde plus d'importance aux exemples difficiles à classer. Elle est souvent utilisée pour améliorer la précision sur des classes déséquilibrées.
- (b) **Observation :** La courbe de train/df_loss diminue de 2.1 à 1.9, et celle de val/df_loss passe de 35.0 à presque 0. Ces diminutions indiquent que le modèle s'améliore dans la classification des exemples difficiles, aussi bien sur les données d'entraînement que sur les données de validation.

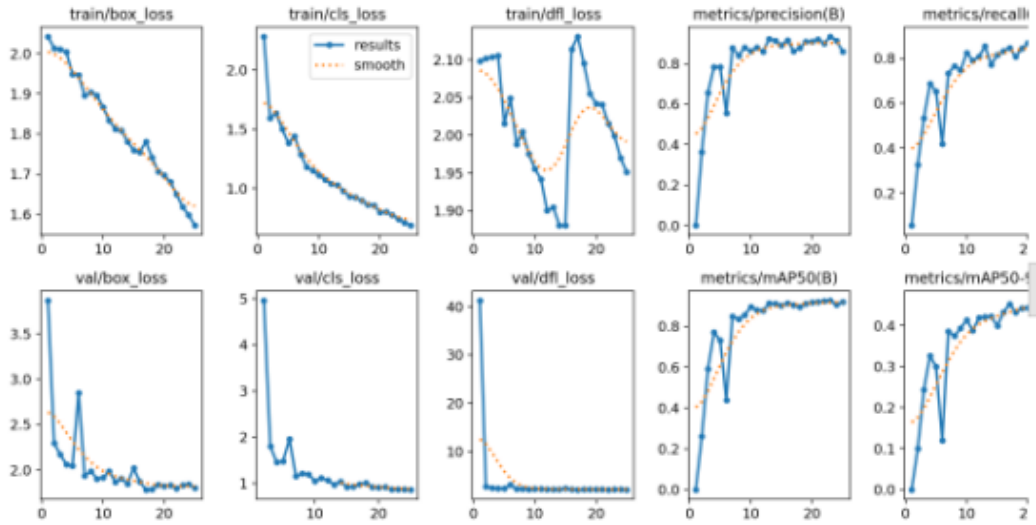


FIGURE 3.10 – Graphs de précision.

3.6.3 Métriques de performance

1. Precision (metrics/precision(B)) :

- (a) **Description** : La précision mesure la proportion de prédictions positives correctes par rapport à toutes les prédictions positives faites par le modèle. Formellement, $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$, où TP est le nombre de vrais positifs et FP est le nombre de faux positifs.
- (b) **Observation** : La courbe metrics/precision(B) atteint environ 0.8. Cela signifie que le modèle fait peu de fausses détections positives, montrant une bonne précision dans ses prédictions.

2. Recall (metrics/recall(B)) :

- (a) **Description** : Le rappel mesure la proportion de vrais positifs correctement identifiés par le modèle par rapport au nombre total de vrais positifs dans les données. Formellement, $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$, où FN est le nombre de faux négatifs.
- (b) **Observation** : La courbe metrics/recall(B) approche 0.85. Cela indique que le modèle détecte la majorité des objets présents dans les images, avec peu de faux négatifs.

3. mAP50 (metrics/mAP50(B)) :

- (a) **Description** : Le mAP (mean Average Precision) à un seuil d'IoU de 50% est la moyenne des précisions moyennes (AP) calculées pour chaque classe à un seuil d'Intersection over Union (IoU) de 50%.
- (b) **Observation** : La courbe metrics/mAP50(B) est proche de 0.85. Un mAP50 élevé indique que le modèle est performant en termes de précision et de rappel à ce seuil particulier.

4. mAP5095 (metrics/mAP5095(B)) :

- (a) **Description** : Le mAP moyen sur plusieurs seuils d'IoU, de 50% à 95%, calculé par incréments de 5%. C'est une mesure plus stricte qui évalue la performance du modèle à différents niveaux de précision.
- (b) **Observation** : La courbe metrics/mAP5095(B) atteint environ 0.55. Un mAP50-95 élevé indique une performance robuste et stable du modèle sur divers seuils de précision, montrant sa capacité à bien généraliser.

Ces observations montrent que le modèle s'améliore constamment au fil du temps, avec des pertes qui diminuent et des métriques de performance qui augmentent, indiquant une bonne capacité de généralisation et de précision.

3.7 Validation du modèle

Ces images (3.11) font partie d'un ensemble de données de validation pour un système de détection automatique des acariens Varroa sur les abeilles. Voici les points importants à noter :

- **Diversité des échantillons** : Les images montrent des abeilles dans différentes poses et angles, ce qui est crucial pour tester la robustesse du modèle de détection.
- **Détections multiples** : Plusieurs images présentent plus d'une boîte de détection, indiquant la capacité du système à identifier plusieurs acariens sur une seule abeille.
- **Précision des détections** : Les boîtes rouges semblent bien cibler les zones où les Varroa sont susceptibles de se trouver, notamment entre les segments abdominaux et sur le thorax des abeilles.
- **Variété des conditions** : Les images ont des éclairages et des arrière-plans différents, ce qui permet de tester le modèle dans diverses conditions.
- **Horodatage** : Chaque image est identifiée par un horodatage, suggérant qu'elles proviennent de séquences vidéo, ce qui pourrait être utile pour suivre l'évolution de l'infestation dans le temps.
- **Qualité variable** : Certaines images sont plus nettes que d'autres, ce qui est important pour évaluer les performances du modèle avec des données de qualité variable.



FIGURE 3.11 – Détection automatique des acariens Varroa sur les abeilles.

Cette validation vise à s’assurer que le modèle de détection des Varroa est précis et fiable dans diverses situations. Elle permet aux chercheurs ou aux développeurs d’ajuster et d’améliorer l’algorithme en fonction des résultats obtenus sur cet ensemble de données de test.

3.7.1 Test du modèle YOLOv8

3.7.2 Via une application WEB

Nous avons implémenté une application web utilisant Flask, un framework Python (voir annex 1) . Son but est de fournir une interface utilisateur pour la détection d’objets en temps réel à partir d’une caméra connectée à l’ordinateur. L’application démarre un serveur Flask qui affiche une page HTML personnalisée. Cette page contient un titre, des boutons pour démarrer et arrêter la vidéo, ainsi qu’une zone où le flux vidéo sera affiché. Lorsque l’utilisateur clique sur le bouton ”Démarrer”, une requête est envoyée au serveur pour commencer la capture vidéo. Un thread séparé gère la capture continue des images de la caméra. Chaque image capturée est ensuite traitée par le modèle YOLO pour détecter la présence d’objets, en l’occurrence des Varroa-Mites . Les résultats de détection sont encodés en JPEG et envoyés au navigateur pour affichage dans la zone vidéo . En cas d’erreur lors de la capture ou de la détection, des messages sont affichés dans la console du serveur Flask pour faciliter le débogage. Cette architecture permet une interaction en temps réel avec la caméra et l’affichage des résultats de détection d’objets directement dans l’interface web.

La figure (3.12) montre une interface web intitulée "PC Camera YOLOv8 BEE_Varroat Detection". Il s'agit d'un système de détection automatique utilisant un modèle d'intelligence artificielle YOLOv8 pour identifier les acariens Varroa sur les abeilles.

L'écran affiche une image capturée par une caméra, On peut voir une détection en cours, avec un cadre rouge autour d'un objet identifié comme "Varroa-Mite" (acarien Varroa) avec un score de confiance de 0,59 (59%) et 0,71 (71%) comme le montre La figure (3.13). Le test semble avoir réussi, car le système a identifié ce qu'il considère être un acarien Varroa dans l'image.

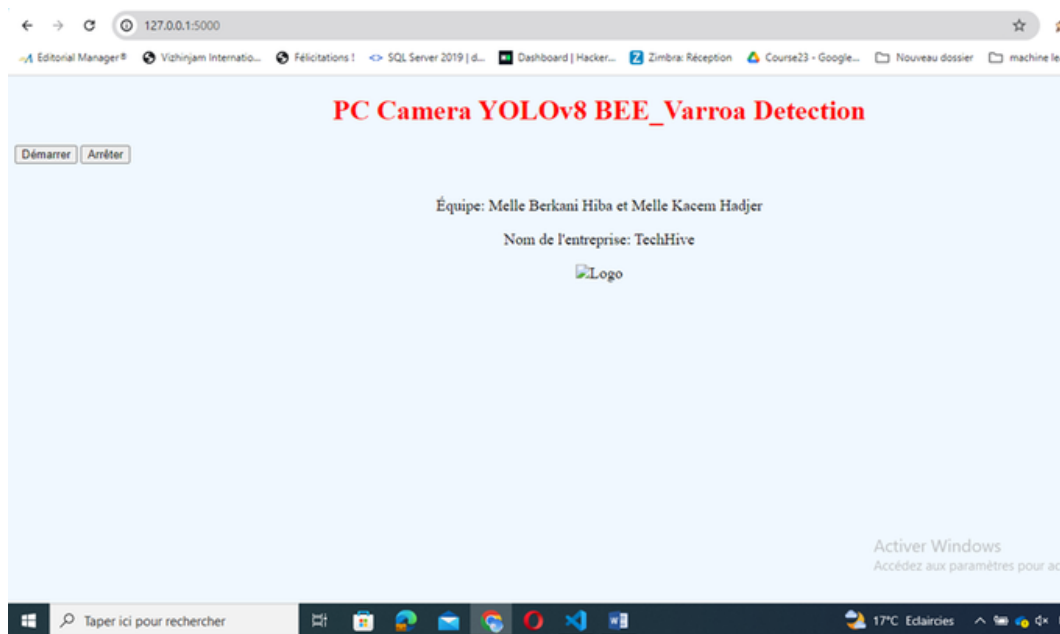


FIGURE 3.12 – Interface de l'application web

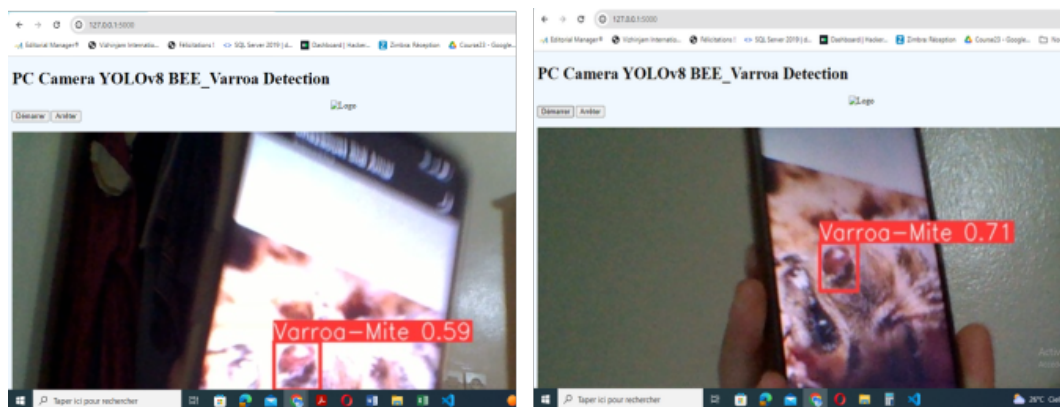


FIGURE 3.13 – Test de l'application développée

3.7.3 ESP32-CAM

L'ESP32-CAM est un module d'AI Thinker basé sur un ESP32 proposant une interface WiFi et Bluetooth associée à une caméra miniature couleur 2MP OV2640 comme le montre la figure (3.14), parmi ces caractéristiques :

- **Alimentation** : 3,3 ou 5 Vcc
- **Consommation** : 2000 mA maxi
- **Microcontrôleur** : ESP32 à 240 MHz
- **Mémoire RAM** : 520 Ko
- **Mémoire PSRAM** : 4 Mo
- **Compatibilité** : capteurs OV2640 et OV7670 (capteur OV2640 intégré)
- **Bluetooth** : compatible 4.2, EDR et BLE
- **WiFi** 802.11b/g/n/e/i (compatible WPA, WPA2, WPA2-Enterprise et WPS)
- **Interfaces disponibles** : UART, SPI, I2C, PWM, ADC et DAC
- **Formats vidéo supportés** : JPEG (OV2640)
- **Débit série** : 115200 bps
- **Lecteur de carte micro-SD** (4 Go maxi, carte non inclus)
- **Broches utilisées par le lecteur micro-SD** : 2, 4, 12, 13, 14 et 15
- **Sorties 3,3 Vcc**
- **Broches GPIO** : UART, SPI et I2C
- **Bouton reset**
- **Température de service** : -20 à 85 °C
- **Dimensions** : 27 x 40 x 4,5 mm
- **Poids** : 10 g [12]

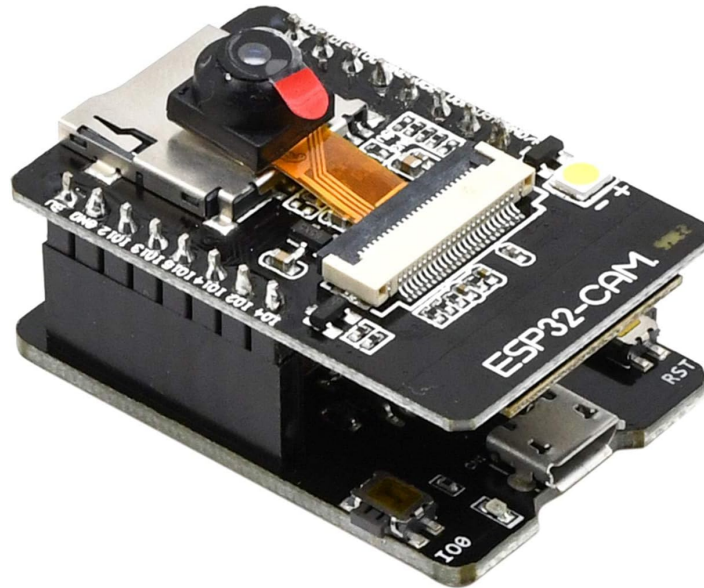


FIGURE 3.14 – ESP32 CAM.

3.7.4 Installer le logiciel requis

- Téléchargez et installez l'IDE Arduino depuis le site officiel.
- Ouvrez Arduino IDE.
- Allez dans Fichier >Préférences.
- Dans le champ "URL de gestionnaire de cartes supplémentaires", ajoutez cette URL : https://dl.espressif.com/dl/package_esp32_index.json.
- Allez dans Outils >Type de carte >Gestionnaire de cartes et recherchez "esp32".
- Installez le paquet "esp32" de Espressif Systems.

3.7.5 Configurer l'ESP32-CAM

- Connecter l'ESP32-CAM à l'ordinateur.
- Ouvrez Arduino IDE.
- Allez dans Fichier >Exemples >ESP32 >Camera >CameraWebServer.
- Modifiez les paramètres Wi-Fi dans le code (ssid et password).
- Sélectionnez la carte ESP32 Wrover Module sous Outils >Type de carte.
- Sélectionnez le bon port COM sous Outils >Port.
- Cliquez sur Téléverser pour flasher l'ESP32-CAM.
- Une fois le téléversement terminé, ouvrez le moniteur série (Baudrate 115200).

- Appuyez sur le bouton reset de l'ESP32-CAM.
- Notez l'adresse IP affichée dans le moniteur série comme illustré dans la figure (3.15).
- Accédez à cette adresse IP dans un navigateur web pour voir le flux vidéo.

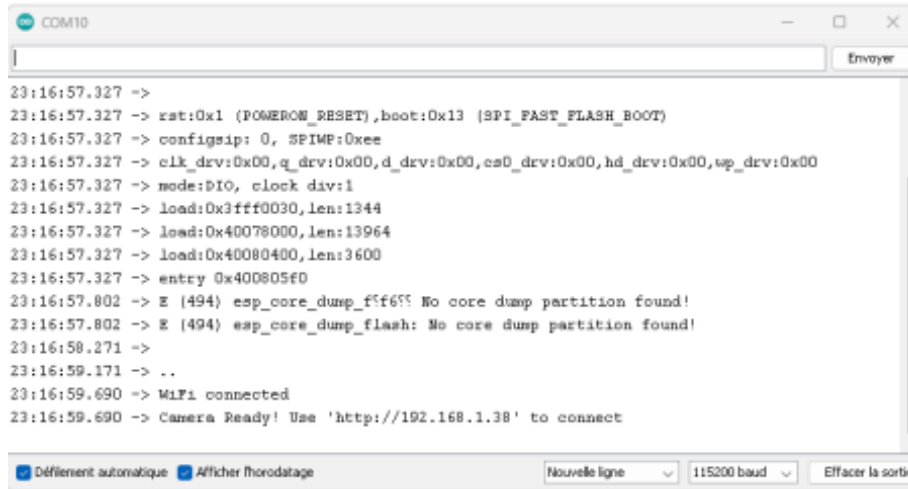


FIGURE 3.15 – Adresse IP du ESP32-CAM.

3.7.6 Préparer l'environnement YOLOv8

- Installer Python(version 3.8 ou supérieure).
- Ouvrez l'invite de commandes et installez les bibliothèques NumPy, OpenCV, ultralytics et cvlib en tapant pip install avant chaque bibliothèques comme illustré dans la figure (3.16)
- Téléchargez le modèle YOLOv8 depuis le dépôt officiel Ultralytics.
- Placez le modèle dans un répertoire.

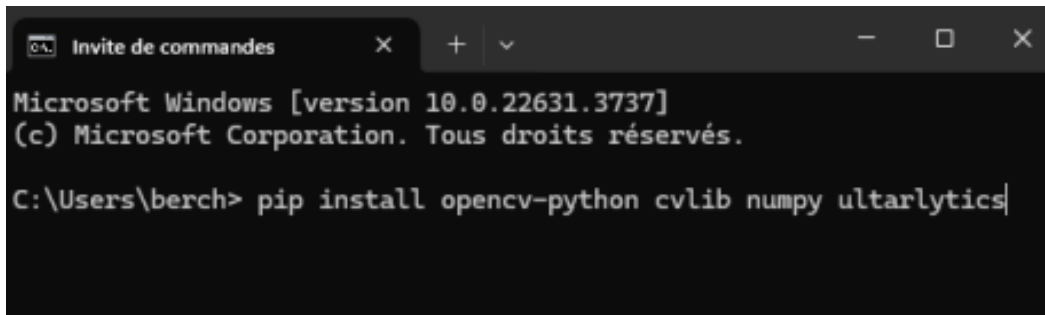


FIGURE 3.16 – Installation des bibliothèques de python.

```
from ultralytics import YOLO
import cv2
import requests
import numpy as np
```

FIGURE 3.17 – Importation des Bibliothèques.

3.7.7 Intégrer YOLOv8 avec l'ESP32-CAM

1. Importation des Bibliothèques

- 'Ultralytics' est utilisé pour charger et utiliser le modèle YOLO.
- 'cv2' est la bibliothèque OpenCV, utilisée pour le traitement d'images et l'affichage.
- 'Requests' est utilisé pour envoyer des requêtes HTTP pour obtenir l'image de l'ESP32-CAM.
- 'Numpy' est utilisé pour manipuler les tableaux de données d'image.

2. Définir les Constantes

```
# URL de l'ESP32-CAM (remplacez par l'URL correcte)
ESP32_CAM_URL = "http://192.168.1.38/capture"

# Chemin complet vers le modèle YOLO
model_path = "C:/Users/berch/Desktop/best.pt"
```

FIGURE 3.18 – Définir les Constantes.

- 'ESP32_CAM_URL' est l'URL de l'ESP32-CAM pour capturer une image.
- 'model_path' est le chemin complet vers le modèle YOLO pré-entraîné.

3. Charger le Modèle YOLO

```
# Charger le modèle YOLO
model = YOLO(model_path)
```

FIGURE 3.19 – Charger le Modèle YOLO

- ‘YOLO(model_path)’ charge le modèle YOLO à partir du chemin spécifié.

4. Définir une Fonction pour Obtenir l’Image de l’ESP32-CAM

```
def get_frame_from_esp32_cam(url):
    # Envoyer une requête GET pour obtenir l'image
    response = requests.get(url)
    if response.status_code == 200:
        # Convertir le contenu de la réponse en tableau numpy
        image_array = np.frombuffer(response.content, np.uint8)
        # Décoder l'image
        frame = cv2.imdecode(image_array, cv2.IMREAD_COLOR)
        return frame
    else:
        print("Erreur lors de la capture de l'image depuis l'ESP32-CAM")
        return None
```

FIGURE 3.20 – Fonction pour Obtenir l’Image de l’ESP32-CAM.

- ‘requests.get(url)’ envoie une requête GET à l’ESP32-CAM pour capturer une image.
- ‘np.frombuffer(response.content, np.uint8)’ convertit le contenu de la réponse en un tableau numpy.
- ‘cv2.imdecode(image_array, cv2.IMREAD_COLOR)’ décode le tableau numpy en une image.
- La fonction retourne l’image décodée ou ‘None’ en cas d’échec.

5. Boucle Principale

```
while True:
    # Obtenir l'image depuis l'ESP32-CAM
    frame = get_frame_from_esp32_cam(ESP32_CAM_URL)
    if frame is not None:
        # Prédire à partir de l'image avec affichage et confiance de 0.25
        results = model.predict(source=frame, show=True, conf=0.25)

        # Afficher le résultat
        print(results) # Debugging line to print the results object structure

        # Let's assume 'results' is a list of dictionaries or similar structures
        for result in results:
            print(result) # Debugging line to print each result in the results
            # Assuming result.imgs[0] should be replaced with the correct attrib
            # Since the exact structure is unknown, we need to inspect the result
            if hasattr(result, 'imgs') and len(result.imgs) > 0:
                cv2.imshow("YOLO Prediction", result.imgs[0])
            elif hasattr(result, 'image'):
                cv2.imshow("YOLO Prediction", result.image)
            else:
                print("No image attribute found in result")

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    else:
        print("Impossible de capturer l'image depuis l'ESP32-CAM")
```

FIGURE 3.21 – boucle principale

3.7.8 RESULTAT DU TEST

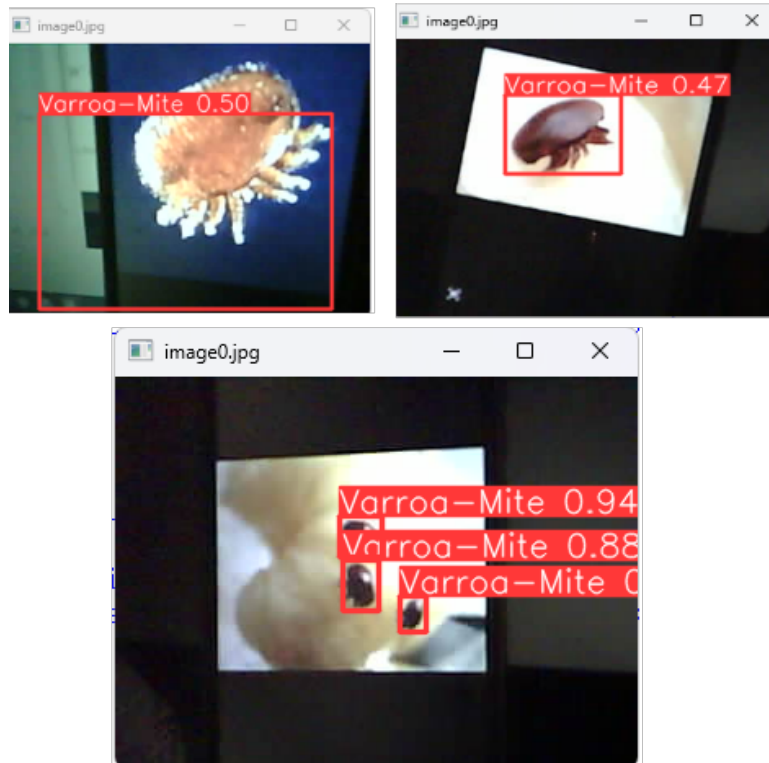


FIGURE 3.22 – Résultat du test.

```
No image attribute found in result
0: 608x800 1 Varroa-Mite, 1704.1ms
Speed: 15.6ms preprocess, 1704.1ms inference, 0.0ms postprocess per image at shape (1, 3, 608, 800)
Squeezed text (62 lines).
Squeezed text (62 lines).

No image attribute found in result
0: 608x800 3 Varroa-Mites, 1751.0ms
Speed: 15.6ms preprocess, 1751.0ms inference, 0.0ms postprocess per image at shape (1, 3, 608, 800)
Squeezed text (62 lines).
Squeezed text (62 lines).
```

FIGURE 3.23 – Résultat de détection.

Conclusion

La détection précoce et précise des maladies des abeilles, en particulier le varroa destructor, est cruciale pour la protection de nos colonies d'abeilles et, par extension, pour la sécurité alimentaire et la stabilité de notre écosystème. Les méthodes traditionnelles de surveillance des ruches, bien qu'essentielles, présentent des limitations notables en termes de coût, de temps et d'expertise requise. Les avancées technologiques, notamment l'intelligence artificielle et la vision par ordinateur, offrent des solutions prometteuses pour surmonter ces défis.

Dans ce chapitre, nous avons exploré l'état de l'art des systèmes de détection des maladies des abeilles et avons présenté une solution basée sur l'intelligence artificielle, exploitant les techniques de l'apprentissage automatique et du traitement d'images. Nous avons détaillé le processus de création d'une base de données d'images annotées, l'entraînement du modèle YOLOv8 et les résultats obtenus. Le modèle entraîné a montré une grande précision dans la détection du varroa destructor, ouvrant la voie à une surveillance plus efficace et rapide des ruches.

L'intégration de cette technologie dans les pratiques apicoles peut transformer la gestion sanitaire des colonies d'abeilles, en permettant des interventions rapides et ciblées. Cela contribuera non seulement à la préservation des populations d'abeilles, mais aussi à la protection de la biodiversité et à la stabilité des chaînes d'approvisionnement alimentaire.

Conclusion Générale

L'étude approfondie menée à travers les trois chapitres de cette mémoire révèle l'intersection cruciale entre l'intelligence artificielle (IA), la santé des abeilles et la sécurité alimentaire mondiale. Chaque chapitre explore des domaines distincts mais complémentaires, offrant des perspectives uniques sur les défis contemporains et les innovations technologiques qui façonnent notre monde moderne.

Le premier chapitre a examiné l'impact révolutionnaire de YOLO (You Only Look Once) dans le domaine de la détection d'objets. Nous avons constaté que YOLO représente une avancée significative grâce à sa capacité à localiser et classifier des objets en temps réel, malgré certaines limites liées à la détection d'objets de petite taille. L'avenir de YOLO semble prometteur avec des développements continus qui pourraient étendre son application à une gamme encore plus large de tâches et de plateformes matérielles.

Le deuxième chapitre s'est concentré sur la santé des abeilles, mettant en lumière les défis critiques auxquelles elles sont confrontées, tels que les maladies, les parasites et les impacts du changement climatique. En adoptant une approche intégrée, nous avons exploré comment des solutions innovantes basées sur l'IA peuvent contribuer à détecter et prévenir efficacement les maladies comme le varroa destructor, garantissant ainsi la durabilité des populations d'abeilles et la sécurité alimentaire mondiale.

Le troisième chapitre a introduit l'utilisation de l'Internet des Objets (IoT) et des technologies connectées pour améliorer la gestion des ruches et la surveillance de la santé des abeilles. En intégrant des capteurs intelligents et des systèmes de traitement d'images avancés, cette approche permet une surveillance en temps réel, facilitant des interventions rapides et ciblées pour préserver la santé des colonies d'abeilles. En conclusion, cette mémoire démontre que l'innovation technologique, notamment à travers l'IA et l'IoT, joue un rôle crucial dans la résolution des défis environnementaux et agricoles contemporains. En renforçant la capacité à détecter précocement les maladies des abeilles, en améliorant la gestion des ruches et en augmentant la résilience des écosystèmes agricoles, ces avancées contribuent directement à la sécurité alimentaire mondiale et à la préservation de la biodiversité.

Comme perspectives, il est essentiel de poursuivre les recherches et les développements technologiques dans ces domaines afin d'optimiser les systèmes existants et d'explorer de nouvelles applications potentielles. Nous travaillons actuellement au renforcement de la base de données, à l'extension des classes de maladies et à la mise en place d'alertes en cas de détection, tout en intégrant ces fonctionnalités dans une application. En intégrant les connaissances scientifiques, les innovations technologiques et les pratiques durables, nous pouvons aspirer à un avenir où les abeilles continuent de prospérer en tant que pollinisateurs essentiels, assurant ainsi la durabilité écologique et alimentaire pour les générations futures.

Bibliographie

- [1] <https://www.ofapidologie.org/les-causes>.
- [2] <https://mespremiereruches.com/maladies-abeilles-colonies/>.
- [3] <https://www.laboratoires-roig.com/blog/les-causes-et-consequences-de-la-disparition-des-abeilles-en-france-n28>.
- [4] What is google colab ? <https://blog.education-ecosystem.com/what-is-google-colab>, 2021. Consulté en juin 2024.
- [5] Deep AI. Pytorch. <https://deepai.org/machine-learning-glossary-and-terms/pytorch>, 2020. Consulté en juin 2024.
- [6] Marcelo A Aizen, Lucas A Garibaldi, Saul A Cunningham, and Alexandra M Klein. How much does agriculture depend on pollinators? lessons from long-term trends in crop production. *Annals of botany*, 103(9) :1579–1588, 2009.
- [7] Axopen. Open cv, c'est quoi ?, 2021. Consulté en juin 2024.
- [8] Simon Bilik, Lukas Kratochvila, Adam Ligocki, Ondrej Bostik, Tomas Zemcik, Matous Hybl, Karel Horak, and Ludek Zalud. Visual diagnosis of the varroa destructor parasitic mite in honeybees using object detector techniques. *Sensors*, 21(8) :2764, 2021.
- [9] Kim Bjerge, Carsten Eie Frigaard, Peter Høgh Mikkelsen, Thomas Holm Nielsen, Michael Misbih, and Per Kryger. A computer vision system to monitor the infestation level of varroa destructor in a honeybee colony. *Computers and Electronics in Agriculture*, 164 :104898, 2019.
- [10] Larissa Chazette, Matthias Becker, and Helena Szczerbicka. Basic algorithms for bee hive monitoring and laser-based mite control. In *2016 IEEE symposium series on computational intelligence (SSCI)*, pages 1–8. IEEE, 2016.
- [11] Trad Houssein Eddine. La détection d'objet avec opencv et deep learning. 2020.
- [12] Claude Emery. Esp32-cam. <http://emery.claude.free.fr/esp32-cam.html>. Consulté en juin 2024.

- [13] ZEHOUANE FETHI and ZOUHRI YUCEF. *LES MALADIES DES ABEILLES ET L'IMPOR- TANCE DU TRAITEMENT ALTERNATIF*. PhD thesis, Institut des sciences vétérinaires, 2020.
- [14] Kingston J. M. Albrecht M. Holway D. A. Kohn J. R. Hung, K. L. J. The worldwide importance of honey bees as pollinators in natural habitats. *Proceedings of the Royal Society B : Biological Sciences*, 285(1870) :20172140, 2018.
- [15] Hadjer LARIBI, Ouezna MELIANI, et al. *Sélection d'un Algorithme de Deep Learning Approprié pour la Maintenance Prédictive*. PhD thesis, 2023.
- [16] Agnieszka Lawrynowicz and Volker Tresp. Introducing machine learning. *Perspectives on Ontology Learning*, 12 :19, 2014.
- [17] Helena Li. Deep learning : Les algorithmes de génération d'art visuel selon les différentes visions de l'art.
- [18] Alicia Noriega-Escamilla, César J Camacho-Bello, Rosa M Ortega-Mendoza, José H Arroyo-Núñez, and Lucia Gutiérrez-Lazcano. Varroa destructor classification using legendre–fourier moments with different color spaces. *Journal of Imaging*, 9(7) :144, 2023.
- [19] Richard Odemer. Approaches, challenges and recent advances in automated bee counting devices : A review. *Annals of Applied Biology*, 180(1) :73–89, 2022.
- [20] KAABECHE OUSSAMA and AINOUZ TAHAR ABDURRAHMEN. Extraction des ilots par deep learning. Master's thesis, Université XYZ, 2020. Mémoire Master.
- [21] Dominique Pastre. L'intelligence artificielle definition-generalites-historique-domaines. *Intelligence artificielle, Université Paris*, 5, 2000.
- [22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [23] Nicolas P Rougier. Une brève histoire de l'intelligence artificielle. In *Pint of Science*, 2015.
- [24] L. Rédaction. Python. <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique>, 2020. Consulté en juin 2024.
- [25] Stefan Schurischuster and Martin Kampel. Image-based classification of honeybees. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2020.

- [26] Stefan Schurischuster, Beatriz Remeseiro, Petia Radeva, and Martin Kampel. A preliminary study of image analysis for parasite detection on honey bees. In *Image Analysis and Recognition : 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018, Proceedings 15*, pages 465–473. Springer, 2018.
- [27] Stefan Schurischuster, Sebastian Zambanini, Martin Kampel, and Benjamin Lamp. Sensor study for monitoring varroa mites on honey bees (*apis mellifera*). In *Proc. of Visual observation and analysis of Vertebrate And Insect Behavior Workshop (VAIB 2016)*, page 4, 2016.
- [28] Juan Terven and Diana Cordova-Esparza. A comprehensive review of yolo : From yolov1 to yolov8 and beyond. *arXiv preprint arXiv :2304.00501*, 2023.
- [29] Samuel Théophile. Roboflow : Conversion d’annotations pour la détection d’objets, septembre 2023. Consulté en juin 2024.
- [30] Hocine TOLGUI. Deep learning pour reconnaissance du visage. 2017.
- [31] George Voudiotis, Anna Moraiti, and Sotirios Kontogiannis. Deep learning beehive monitoring system for early detection of the varroa mite. *Signals*, 3(3) :506–523, 2022.

Annexe A

Première annexe

A.1 Importation Bibliothèque

```
from flask import Flask, Response, render_template_string
import cv2
from ultralytics import YOLO
import threading
```

- **Flask** : Framework web utilisé pour créer l'interface utilisateur et gérer les requêtes HTTP.
- **cv2 (OpenCV)** : Bibliothèque pour le traitement d'images et la capture vidéo.
- **YOLO (ultralytics)** : Framework d'apprentissage profond utilisé pour la détection d'objets.

A.2 Initialisation de l'Application Flask

```
app = Flask(__name__)
```

- Crée une instance de l'application Flask.

A.3 Chargement du Modèle YOLO

```
model_path = "C:/best (3) (1).pt"  
model = YOLO(model_path)
```

- Spécifie le chemin du modèle YOLO pré-entraîné et l'instancie avec `YOLO()`.

A.4 Initialisation de la Capture Vidéo

```
cap = cv2.VideoCapture(0)  
cap_opened = False  
frame_lock = threading.Lock()  
output_frame = None
```

- `cv2.VideoCapture(0)` : Ouvre le flux vidéo de la première caméra disponible.
- `cap_opened` : Variable pour suivre l'état de la caméra (ouverte ou fermée).
- `frame_lock` : Objet de verrouillage pour synchroniser l'accès aux frames vidéo.
- `output_frame` : Stocke la dernière frame traitée pour l'affichage.

A.5 Modèle HTML pour l'Interface Utilisateur

```
template = f"""  
<!DOCTYPE html>  
<html>  
<head>  
  <title>PC Camera YOLOv8 BEE_Varroa Detection</title>  
  <style>  
    body { background-color: #f0f8ff; } /* Couleur de fond personnalisée */  
    .logo {  
      width: 100px;  
      display: block;  
      margin-left: auto;
```

```

        margin-right: auto;
    }
</style>
<script type="text/javascript">
    function startVideo() {
        fetch('/start_video');
        document.getElementById('video_feed').src = "/video_feed";
    }
    function stopVideo() {
        fetch('/stop_video');
        document.getElementById('video_feed').src = "";
    }
</script>
</head>
<body>
    <h1 style="text-align: center; color: red; font-weight: bold;">PC Camera YOLOv8 BEE_Varro</h1>
    <p style="text-align: center;">Equipe: Melle Berkani Hiba et Melle Kacem Hadjer</p>
    
    <button onclick="startVideo()">Démarrer</button>
    <button onclick="stopVideo()">Arrêter</button>
    <br><br>
    <img id="video_feed" src="" width="100%">
</body>
</html>
"""

```

- **template** : Modèle HTML pour l'interface utilisateur. Il définit une page web avec un titre, un style CSS personnalisé, des boutons pour démarrer/arrêter la vidéo, et un élément `img` pour afficher le flux vidéo.

A.6 Boucle pour Capturer les Frames et Faire des Prédiction

```
def capture_frames():
    global cap_opened, output_frame, cap
    while True:
        with frame_lock:
            if not cap_opened:
                cap.release() # Libérer la ressource de la caméra
                continue
            if not cap.isOpened():
                cap = cv2.VideoCapture(0) # Réinitialiser la capture vidéo
            ret, frame = cap.read()
            if not ret:
                print("Erreur lors de la capture de la frame")
                continue

            # Prédire à partir de la frame avec YOLO
            try:
                results = model.predict(frame, conf_threshold=0.3, nms_threshold=0.4)
                if isinstance(results, list):
                    for result in results:
                        if hasattr(result, 'pred'):
                            detections = result.pred
                            if len(detections) > 0:
                                result_frame = result.render()
                                _, jpeg = cv2.imencode('.jpg', result_frame)
                                output_frame = jpeg.tobytes()
                                break
                        else:
                            print("Aucune détection trouvée.")
                    elif hasattr(results, 'pred'):
                        detections = results.pred
                        if len(detections) > 0:
```

```

        result_frame = results.render()
        _, jpeg = cv2.imencode('.jpg', result_frame)
        output_frame = jpeg.tobytes()
    else:
        print("Aucune détection trouvée.")
except Exception as e:
    print(f"Erreur lors de la prédiction avec YOLO : {e}")
    continue

```

- **capture_frames()** : Fonction exécutée dans un thread séparé pour capturer les frames vidéo, effectuer des prédictions avec YOLO, et mettre à jour `output_frame` avec les frames traitées.

A.7 Génération du Flux Vidéo pour l'Interface Utilisateur

```

def generate():
    global output_frame
    while True:
        with frame_lock:
            if output_frame is None:
                continue
            frame = output_frame
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

```

- **generate()** : Génère un flux d'images JPEG pour être affiché dans l'élément `img` de l'interface utilisateur.

A.8 Routes Flask pour l'Interface Utilisateur et Contrôle Vidéo

```
@app.route('/video_feed')
def video_feed():
    return Response(generate(),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/')
def index():
    return render_template_string(template)

@app.route('/start_video')
def start_video():
    global cap_opened
    cap_opened = True
    return "Video started"

@app.route('/stop_video')
def stop_video():
    global cap_opened
    cap_opened = False
    return "Video stopped"
```

- **@app.route('/video_feed')** : Route qui renvoie un flux d'images pour être affiché dans l'interface utilisateur.
- **@app.route('/')** : Route qui renvoie le modèle HTML (template) pour l'interface utilisateur.
- **@app.route('/start_video')** : Route pour démarrer la capture vidéo en mettant `cap_opened` à `True`. **@app.route('/stop_video')** : Route pour arrêter la capture vidéo en mettant `cap_opened` à `False`.

A.9 Démarrage de l'Application Flask

- `if __name__ == '__main__':`
 `threading.Thread(target=capture_frames, daemon=True).start()`
 `app.run(host='0.0.0.0', port=5000, debug=True)`
- `__name__ == '__main__'` : Condition qui vérifie si le script est exécuté directement.
- `threading.Thread(target=capture_frames, daemon=True).start()` : Démarrage d'un thread pour capturer les frames vidéo en arrière-plan.
- `app.run(...)` : Lancement de l'application Flask sur le serveur local.

Annexe B

Deuxième annexe

B.1 Prototype



FIGURE B.1 – Ruche avec ESP32-Cam.