RE PUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

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

وزارة البحث العلمي والتعليم العالي

المدرسة العليا للعلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière: Automatique
Spécialité: Automatique

Présenté par : Kawther CHOUACHI et Aya KERROUMI

Thème

Détection et Classification des Maladies des Plantes de Culture par Vision Artificielle et Deep Learning.

Soutenu publiquement, le 18/06/2026 devant le jury composé de:

OUDJAMAA Farid	MCB	ESSA Tlemcen	Président
M'HAMEDI Mohammed	MCB	ESSA Tlemcen	Directeur de mémoire
BEKKAOUI Mokhtaria	MCB	ESSA Tlemcen	Co- Directeur de mémoire
NEDJAR Imane	MCB	ESSA Tlemcen	Examinateur 1
ARICHI Faycal	MCA	ESSA Tlemcen	Examinateur 2

Année universitaire : 2025/2026


بسم الله الرحمن الرحيم

Dédicaces

À mes très chers parents,

Qui ont toujours été mon refuge, ma force et ma plus grande bénédiction. Leur amour inconditionnel, leurs sacrifices silencieux et leurs prières constantes ont illuminé chacun de mes pas.

À ma mère,
Pour son amour infini, sa tendresse, sa patience et ses prières qui m'ont accompagnée tout au long de ce parcours.

À mon père,
Pour sa sagesse, ses sacrifices, son soutien indéfectible et les valeurs qu'il m'a transmises, qui constituent les fondations de tout ce que je suis aujourd'hui.

À ma sœur et à mes frères, Ikhlass, Mohammed et Naoufel
Pour leur affection, leur présence et les moments partagés, dans les joies comme dans les épreuves.

En tant que fille aînée,
Je dédie humblement ce travail à ma famille, avec gratitude et responsabilité, consciente que chacune de mes réussites porte également une part de vous.

À mes amies, Meriem, Sara, Nor et Lyna,
Pour leur amitié sincère, leur soutien précieux et les instants de partage qui ont rendu cette aventure plus douce et plus humaine.

À mon binôme, Aya,
Compagnonne de route dans cette aventure scientifique, pour sa collaboration, sa patience, son implication et son esprit de partage.

Enfin, à moi-même,
Pour la persévérance, le courage et la détermination qui m'ont permis d'aller jusqu'au bout de ce travail malgré les difficultés rencontrées.

Que ce mémoire soit le témoignage des efforts, du soutien et de l'amour qui ont accompagné ce parcours.

Avec une gratitude infinie et une tendresse sincère, alhamdullilah.

Kawther CHOUACHI

Dédicace

À mes très chers parents,
Vous êtes ma plus grande source de force et de motivation. Grâce à votre amour infini, vos sacrifices silencieux et votre soutien inconditionnel, j'ai pu avancer avec confiance et détermination sur ce long chemin. Vous avez tout donné sans jamais rien demander en retour, et c'est grâce à vous que je suis là aujourd'hui. Vous m'avez transmis les valeurs les plus précieuses de la vie : le travail, la patience, l'humilité et la dignité. Chaque ligne de ce mémoire porte votre empreinte, car ce travail est le fruit de vos efforts autant que des miens. Les mots ne sauront jamais exprimer à quel point je vous suis reconnaissante pour tout ce que vous avez fait et continuez de faire pour moi. Vous êtes mon refuge, ma fierté et ma raison de persévérer. Qu'Allah vous protège, vous garde en bonne santé, vous comble de bonheur et vous récompense au centuple pour tous vos sacrifices.

À mes sœurs, Samira et Khadidja,
Merci pour votre présence, votre affection et vos encouragements constants. Votre soutien m'a toujours donné la force de continuer et de croire en moi.

À mes frères, Dahou, Djilali et Yasser,
Je vous remercie pour votre bienveillance et votre soutien. Vos encouragements ont été précieux tout au long de mon parcours.

À ma chère nièce Firdaous,

Ta présence apporte beaucoup de joie et de bonheur dans ma vie. Ton sourire illumine mes journées et remplit mon cœur d'affection. Je te souhaite une vie pleine de santé, de réussite et de bonheur.

À mon binôme Kawther,
Merci pour ton soutien, ta collaboration et tous les moments de partage tout au long de ce projet. Je garderai un excellent souvenir de cette expérience et du travail réalisé ensemble.

À mes amies de la cité universitaire : Nour, Aya, Hiba, Maïssa, Lina, Ikram, Yousra, Souhila, Ahlem, Rabiaa, Loubna et Meriem,
ous avez été bien plus que de simples amies ; vous avez été une véritable famille loin de chez moi. Merci pour votre soutien, vos encouragements, les moments de joie et les souvenirs inoubliables partagés tout au long de ce parcours. Votre présence a rendu cette aventure plus belle et plus facile. Je vous souhaite tout le bonheur et la réussite que vous méritez.

Enfin, à moi-même,
Je me remercie pour tous les efforts que j'ai fournis durant ce parcours. Malgré les difficultés rencontrées, j'ai continué à travailler avec patience et détermination jusqu'à l'aboutissement de ce projet. Je suis fière du chemin parcouru et du travail accompli. Cette réussite est le résultat de ma persévérance, de ma volonté et de mon engagement.

Avec tout mon amour et ma profonde affection.

Aya KERROUMI

Remerciements

Avant toute chose, nous tenons à exprimer notre profonde gratitude à Dieu Tout-Puissant, pour nous avoir accordé la force, la patience et la persévérance nécessaires tout au long de ce travail. Sans Sa miséricorde et Ses bénédictions, l'accomplissement de ce mémoire n'aurait jamais été possible.

Nous exprimons également notre profonde reconnaissance aux membres du jury, à commencer par Monsieur le Président du jury, Dr. Oudjamaa Farid, pour l'honneur qu'il nous fait en acceptant de présider l'évaluation de ce travail. Nous adressons aussi nos sincères remerciements à Dr. Nadjar Imane et Dr. Arichi Faycal, examinateurs de ce mémoire, pour avoir accepté d'examiner notre travail. Leurs remarques, critiques et suggestions seront d'une grande valeur pour enrichir et améliorer ce mémoire.

Nous tenons également à remercier Dr. Ouhoud Amina, représentante du CATI, pour sa présence, son intérêt porté à notre projet et ses orientations précieuses. Ses remarques nous ont permis de mieux situer notre travail dans une perspective d'innovation, de valorisation et d'application concrète.

Nous remercions également Monsieur Bekkouche Omar, partenaire socio-économique invité, pour sa présence et l'intérêt porté à la présentation de ce travail.

Nous tenons à adresser nos plus sincères remerciements à notre encadrant, Dr. Mohamed M'hamedi, ainsi qu'à notre co-encadrante, Dr. Mokhtaria Bekkaoui, pour leur disponibilité, leurs conseils avisés et leur précieuse orientation tout au long de ce travail. Leur rigueur scientifique, leur bienveillance et leur accompagnement attentif nous ont permis d'avancer sereinement à chaque étape de ce projet et d'en améliorer considérablement la qualité. Leur pédagogie et leur professionnalisme ont été une véritable source de motivation et d'inspiration.

Nous adressons nos sincères remerciements à Monsieur Ramzo, responsable du Fablab de l'école, pour nous avoir généreusement ouvert les portes de cet espace et mis à notre disposition les équipements nécessaires à la réalisation de la partie pratique de ce mémoire. Sa disponibilité, sa confiance et son soutien technique ont été essentiels pour concrétiser les aspects expérimentaux de notre travail. Son accompagnement sur le terrain a été d'une grande valeur, tant sur le plan technique que humain.

Nous souhaitons également exprimer notre reconnaissance à Monsieur Messiredi et Monsieur Belarouci pour leurs remarques constructives, leur aide précieuse et leurs encouragements lors de l'élaboration de ce travail. Leur expertise et leur disponibilité nous ont grandement aidés à améliorer la qualité de notre mémoire.

Nous exprimons notre profonde gratitude à l'ensemble des enseignants de l'École Supérieure en Sciences Appliquées de Tlemcen pour la qualité de l'enseignement dispensé et leur accompagnement tout au long de notre parcours. Grâce à eux, nous avons pu acquérir les compétences nécessaires à la réalisation de ce projet et à notre avenir professionnel.

Enfin, nous adressons une pensée sincère et chaleureuse à toutes les personnes qui

nous ont accompagnés, conseillés ou encouragés, de près ou de loin, durant ces cinq années d'études au sein de l'École Supérieure en Sciences Appliquées de Tlemcen.

Résumé arabe

ملخص

تُعد زراعة الطماطم من القطاعات الزراعية المهمة في الجزائر نظراً لدورها في الأمن الغذائي والإنتاج الزراعي. إلا أن هذه الزراعة تواجه العديد من الأمراض الورقية التي تؤثر سلباً على المردود الزراعي وجودة المحصول. وفي هذا السياق، تبرز تقنيات الذكاء الاصطناعي، وخاصة التعلم العميق، كحلول واعدة للكشف الآلي عن الأمراض النباتية وتشخيصها.

تقدم هذه المذكرة دراسة مقارنة لعدة معماريات من التعلم العميق مخصصة لتصنيف وكشف أمراض أوراق الطماطم اعتماداً على قاعدة بيانات PlantVillage. وقد تم تقييم ثلاثة نماذج للتصنيف تعتمد على الشبكات العصبية الالتفافية، وهي VGG19 و MobileNetV2وEfficientNetB3، بهدف دراسة أدائها وقدرتها على التعميم. كما تم استخدام نموذج YOLOv8n للكشف عن الآفات المرضية وتحديد مواقعها على أوراق الطماطم.

أظهرت النتائج أن جميع النماذج المدروسة حققت أداءً متميزاً في التعرف على الأمراض النباتية. وقد تميز نموذج EfficientNetB3كأفضل نموذج للتصنيف بفضل توازنه بين الدقة والكفاءة الحسابية. أما MobileNetV2 فقد أثبت ملاءمته للتطبيقات المدمجة والأجهزة محدودة الموارد بسبب خفته وسرعة تنفيذه. ورغم أن VGG19 حقق نتائج جيدة، إلا أن تكلفته الحسابية المرتفعة واستقراره الأقل أثناء التدريب يحدان من قابليته للتطبيق العملي. وفي مجال الكشف، أظهر نموذج YOLOv8n قدرة عالية على تحديد وتموضع المناطق المصابة بدقة وفي الزمن الحقيقي، مما يجعله مناسباً للتطبيقات الزراعية الميدانية.

كما أبرزت الدراسة الأثر الإيجابي لتقنيات زيادة البيانات في تحسين متانة النماذج وقدرتها على التعميم. وتؤكد النتائج المتحصل عليها الإمكانات الكبيرة للتعلم العميق كأداة مساعدة لاتخاذ القرار في مراقبة صحة المحاصيل، وتمثل أساساً لتطوير أنظمة زراعية ذكية تدعم الزراعة الدقيقة في الجزائر.

الكلمات المفتاحية: الطماطم، الأمراض الورقية، التعلم العميق، الرؤية الحاسوبية، الشبكات العصبية الالتفافية، EfficientNetB3، MobileNetV2، YOLOv8n، الزراعة الدقيقة، الذكاء الاصطناعي .

Abstract

Tomato cultivation plays a strategic role in Algerian agriculture and food security. However, tomato crops are highly vulnerable to several foliar diseases that can significantly affect productivity and crop quality. In this context, artificial intelligence, particularly deep learning techniques, offers promising solutions for the automated diagnosis of plant diseases.

This thesis presents a comparative study of several deep learning architectures applied to the classification and detection of tomato leaf diseases using the PlantVillage dataset. Three convolutional neural network (CNN) models, namely VGG19, MobileNetV2, and EfficientNetB3, were evaluated under different training configurations to assess their classification performance and generalization capability. In addition, the YOLOv8n model was investigated for the detection and localization of disease symptoms on tomato leaves.

The obtained results demonstrate that all studied models achieve excellent performance in disease recognition tasks. EfficientNetB3 emerged as the most effective classification architecture, providing an outstanding balance between accuracy and computational efficiency. MobileNetV2 proved to be a suitable solution for resource-constrained environments due to its lightweight design and fast inference speed. Although VGG19 achieved satisfactory results, its high computational cost and lower training stability reduce its practical suitability. For disease detection, YOLOv8n showed remarkable capability in accurately identifying and localizing diseased regions in real time, making it highly suitable for field deployment.

The study also highlights the positive impact of data augmentation on model robustness and generalization. Overall, the findings confirm the potential of deep learning as an effective decision-support tool for crop health monitoring and provide a solid foundation for the development of intelligent precision agriculture systems in Algeria.

Keywords : Tomato, Leaf diseases, Deep Learning, Computer Vision, CNN, EfficientNetB3, MobileNetV2, YOLOv8n, Precision Agriculture, Artificial Intelligence.

Résumé

La culture de la tomate constitue un secteur stratégique pour l'agriculture algérienne, mais elle demeure fortement exposée à plusieurs maladies foliaires susceptibles de réduire significativement les rendements. Dans ce contexte, l'intelligence artificielle et plus particulièrement les techniques de deep learning offrent des solutions prometteuses pour l'automatisation du diagnostic phytopathologique.

Ce mémoire présente une étude comparative de plusieurs architectures de deep learning appliquées à la classification et à la détection des maladies des feuilles de tomate à partir du jeu de données PlantVillage. Trois modèles de classification basés sur les réseaux de neurones convolutionnels (VGG19, MobileNetV2 et EfficientNetB3) ont été évalués, avec et sans augmentation de données, afin d'analyser leur capacité de généralisation. Par ailleurs, le modèle YOLOv8n a été étudié pour la détection et la localisation des lésions foliaires.

Les résultats obtenus montrent que l'ensemble des modèles étudiés atteint un niveau de performance très élevé. EfficientNetB3 se distingue comme l'architecture de classification la plus performante grâce à son excellent équilibre entre précision et efficacité computationnelle. MobileNetV2 offre quant à lui un compromis particulièrement intéressant pour les applications embarquées en raison de sa légèreté et de sa rapidité d'exécution. Bien que VGG19 fournisse également de bonnes performances, son coût computationnel important et son entraînement plus instable limitent son intérêt pratique. Concernant la détection, YOLOv8n démontre une excellente capacité à identifier et localiser les maladies foliaires en temps réel, ce qui le rend particulièrement adapté aux applications agricoles de terrain.

Cette étude met également en évidence l'impact positif de l'augmentation de données sur la robustesse et la capacité de généralisation des modèles. Les résultats obtenus confirment ainsi le potentiel du deep learning comme outil d'aide à la décision pour la surveillance sanitaire des cultures et ouvrent des perspectives prometteuses pour le développement de systèmes intelligents destinés à l'agriculture de précision en Algérie.

Mots clés : Tomate, Maladies foliaires, Deep Learning, Vision par ordinateur, CNN, EfficientNetB3, MobileNetV2, YOLOv8n, Agriculture de précision, Intelligence artificielle.

Table des matières

Dédicaces ii

Dédicace iii

Remerciements v

Résumé arabe vii

Liste des Figures xv

Liste des Tableaux xvi

Liste des Abréviations xvii

Introduction Générale 1

Introduction Générale 1

1 Généralités sur l'Intelligence Artificielle et l'Apprentissage Profond 5

1.1 Introduction 5

1.2 Historique de l'Intelligence Artificielle 5

1.3 Sous-domaines de l'Intelligence Artificielle 6

1.4 Apprentissage automatique (Machine Learning) 7

1.5 Apprentissage profond (Deep Learning) 7

1.5.1 Réseaux de Neurones Artificiels (ANN) 8

1.5.2 Composants des Réseaux de Neurones Artificiels 8

1.5.3 Types de Réseaux de Neurones Artificiels 10

1.6 Réseaux de Neurones Convolutifs (CNN) 10

1.6.1 Architecture d'un CNN 11

1.6.2 Types de Réseaux de Neurones Convolutifs (CNN) 13

1.7 Préparation des données 13
1.8 Types de classification 14
1.9 Métriques d'évaluation 14
1.10 Évaluation et validation 15
1.10.1 Matrice de confusion 15
1.10.2 Métriques globales 15
1.10.3 Métriques par classe 16
1.10.3.1 Classification binaire : 16
1.10.3.2 Classification multiclasses : 16
1.11 YOLO et ses évolutions 17
1.11.1 Évolution des versions 17
1.11.2 Caractéristiques de YOLOv8n 17
1.11.3 Métriques d'évaluation 17
1.12 Conclusion 18

2 Méthodologie : Classification par Transfert d'Apprentissage et Détection par YOLOv8n 19
2.1 Introduction 19
2.2 Préparation des ensembles des données 19
2.2.1 Source des images : Dataset PlantVillage 19
2.2.2 Filtrage des classes Tomate 20
2.2.3 Nettoyage des images corrompues 20
2.2.4 Partitionnement 20
2.2.5 Prétraitement des données 20
2.2.6 Augmentation des données (train uniquement) 20
2.2.7 Normalisation 21
2.2.8 Redimensionnement 21
2.3 Contribution 1 : Modèles d'apprentissage de transfer 21
2.3.1 Modèle VGG19 (Visual Geometry Group 19) 23
2.3.2 Modèle MobileNetV2 23
2.3.3 Modèle EfficientNetB3 24
2.3.4 Stratégies d'adaptation des architectures CNN par apprentissage par transfert 25
2.4 Contribution 2 : YOLOv8n la détection et localisation 26

2.4.1 Architecture YOLOv8n 26
2.5 Entraînement des modèles 28
2.5.1 Optimiseur : AdamW 28
2.5.2 Callbacks : Model Checkpoint 28
2.5.3 Fonction de perte : Focal Loss 28
2.6 Environnement et outils de développement 28
2.6.1 Kaggle 28
2.6.2 Python 29
2.6.3 TensorFlow 29
2.6.4 Keras 30
2.6.5 OpenCV 30
2.6.6 Numpy 30
2.6.7 Pandas 31
2.7 Conclusion 31

3 Résultats, Discussion et Interprétation 32
3.1 Introduction 32
3.2 Résultats des modèles CNN de classification 32
3.2.1 VGG19 32
3.2.1.1 Convergence stabilité 32
3.2.1.2 Résultats sur le jeu de test 34
3.2.1.3 Interprétation 35
3.2.2 MobileNetV2 35
3.2.2.1 Convergence et Stabilité 35
3.2.2.2 Impact de l'augmentation 38
3.2.2.3 Résultats sur le jeu de test 38
3.2.2.4 Interprétation 40
3.2.3 EfficientNetB3 40
3.2.3.1 Convergence et Stabilité 40
3.2.3.2 Impact de l'augmentation 43
3.2.3.3 Résultats sur le jeu de test 43
3.2.3.4 Interprétation 45

3.2.4 Analyse comparative 45
3.3 Résultats et interprétation du modèle de détection YOLOv8n 45
3.4 Comparaison avec la littérature 47
3.4.1 Analyse des classes difficiles 48
3.4.2 Limites et approches 49
3.5 Conclusion 49

Conclusion Générale **50**

Annexe **51**

Table des figures

1 Évolution de la production de tomate en Algérie (2018–2024). 3
2 Évolution estimée de la consommation de tomate en Algérie. 3
3 Pertes de rendement causées par les principales maladies de la tomate. 4

1.1 Schéma de relation entre Machine Learning, Deep Learning et Artificial intelligence. 8
1.2 Fonction d'activation ReLu. 9
1.3 Fonction d'activation sigmoid. 9
1.4 Fonction d'activation softmax. 10
1.5 CNN générale. 11
1.6 Matrice de numéro 2. 11
1.7 Filtre parcout un image. 12
1.8 Résultat d'application le filtre «feature map». 12
1.9 Quelque types de Réseaux de Neurones Convolutifs (CNN). 13
1.10 Diagramme résumé des types de classification. 14
1.11 Matrice de confusion. 15

2.1 Pipeline de classification : Modèles d'apprentissage de transfer (PlantVillage). 22
2.2 Schéma d'architecture de modèle VGG19 [48]. 23
2.3 Schéma d'architecture de modèle MobileNetV2 [51]. 24
2.4 Schéma d'architecture de modèle EfficientNetB3 [53]. 25
2.5 Pipeline de classification et détection de YOLOv8n (PlantVillage). 27
2.6 Logo Kaggle [58]. 29
2.7 Logo Python [60]. 29
2.8 Logo TensorFlow [62]. 30
2.9 Logo keras [64]. 30
2.10 Logo OpenCV [65]. 30
2.11 Logo NumPy [66]. 31

3.1 Courbe de perte du modèle VGG19. 33
3.2 Courbe d'apprentissage du modèle VGG19. 33
3.3 Matrice de confusion du modèle VGG19. 34

3.4 Courbe d'apprentissage du modèle MobileNetV2 sans augmentation. 35
3.5 Courbe de perte du modèle MobileNetV2 sans augmentation. 36
3.6 Courbe d'apprentissage du MobileNetV2 avec augmentation. 37
3.7 Courbe de perte du MobileNetV2 avec augmentation. 37
3.8 Matrice de confusion du modèle MobileNetV2 sans augmentation. 38
3.9 Matrice de confusion du modèle MobileNetV2 avec augmentation. 39
3.10 Courbe d'apprentissage du modèle EfficientNetB3 sans augmentation. 41
3.11 Courbe de perte du modèle EfficientNetB3 sans augmentation. 41
3.12 Courbe d'apprentissage du modèle EfficientNetB3 avec augmentation. 42
3.13 Courbe de perte du modèle EfficientNetB3 avec augmentation. 42
3.14 Matrice du confusion de modèle EfficientNetB3 sans augmentation. 43
3.15 Matrice de confusion du modèle EfficientNetB3 avec augmentation. 44
3.16 Courbes des pertes et de métriques du modèle YOLOv8n entraînée. 46

Liste des tableaux

1 Liste des acronymes utilisés xvii

1.1 Performances YOLOv8 sur COCO. 18

2.1 Répartition des images par classe de maladie dans PlantVillage. 19

3.1 Rapport de classification de modèle VGG19. 34
3.2 Comparaison MobileNetV2 : avec vs sans augmentation. 38
3.3 Rapport de classification : MobileNetV2 sans augmentation. 39
3.4 Rapport de classification : MobileNetV2 avec augmentation. 40
3.5 Comparaison EfficientNetB3 : avec vs sans augmentation. 43
3.6 Rapport de classification : EfficientNetB3 sans augmentation. 44
3.7 Rapport de classification : EfficientNetB3 avec augmentation. 45
3.8 Synthèse comparative pour tous modèles. 45
3.9 Métriques globales de YOLOv8n : Validation vs Test 46
3.10 Average Precision (AP50) de YOLOv8n par classe. 47
3.11 Performance mAP de YOLOv8n par classe. 47
3.12 Comparaison avec quelques travaux de la littérature existante. 48

Liste des Abréviations

TABLE 1 – Liste des acronymes utilisés

Acronyme	**Signification**
CNN	Convolutional Neural Network
DL	Deep Learning
ESSAT	École Supérieure en Sciences Appliquées de Tlemcen
FAO	Food and Agriculture Organization
FNN	Feedforward Neural Network
IA	Intelligence Artificielle
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
ML	Machine Learning
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
VGG19	Visual Geometry Group, réseau de 19 couches convolutionnelles
YOLO	You Only Look Once

Introduction Générale

L'agriculture constitue l'un des bases fondamentaux de la sécurité alimentaire nationale et de l'économie algérienne. Dans ce secteur vital, la culture de la tomate (*Solanum lycopersicum*) occupe une place stratégique particulière : d'après les estimations, plus de 1,66 million de tonnes sur une superficie dépassant les 25 000 hectares en 2022, définissant l'Algérie comme le deuxième plus grand producteur de tomates dans la zone méditerranéenne, juste après l'Espagne, selon les données fournies par la FAO [1]. La tomate industrielle, qui représente 65% des surfaces agricoles du pays, a montré une progression notable de sa productivité, passant de près de 200 qx/ha de 2000 à 2009 à plus de 500 qx/ha de 2010 à 2017. autrement dit une augmentation de 136 % [2]. Sur le plan national, la production totale de légumes a augmenté de 7,6 à 7,9 millions de tonnes métriques de 2021 à 2022, exploitant une région de 307 000 hectares [3].

Cependant, malgré ces résultats, l'industrie de la tomate est fortement menacée aux maladies phytopathologiques. Selon la FAO, à l'échelle mondiale, les maladies et les organisms des plantes sont à l'origine d'une perte de 20 à 40 % de la production agricole chaque année, ce qui représente un coût économique de plus de 220 milliards de dollars USD par an [4]. En Algérie, une diminution de la production de tomates industrielles de 14,1% en 2022 a été remarquée, suite à une période de croissance continue de quatre ans [5].

Trois maladies foliaires majeures menacent les cultures de tomate en Algérie.

Le mildiou (Phytophthora infestans) est considéré comme la maladie la plus grave, pouvant détruire totalement la plante en quelques jours lorsqu'il n'est pas maîtrisé, et sévit particulièrement dans les zones de production du Nord durant les périodes humides du printemps et de l'automne **agrichem_mildiou_tomate**, [6]. Les pertes de rendement qu'il engendre peuvent atteindre 70 à 80 % [7], [8].

L'alternariose (Alternaria solani), fréquente dans les zones littorales algériennes, peut provoquer des pertes allant jusqu'à 80 % du rendement, comme l'a démontré une étude conduite à partir d'isolats prélevés dans plusieurs régions du pays [6], [9].

La moisissure des feuilles (Passalora fulva) provoque quant à elle une dégradation progressive du feuillage, réduisant la photosynthèse et la qualité commerciale des fruits [10].

Ces pathologies se développent particulièrement dans les conditions climatiques méditerranéennes et semi-humides qui caractérisent les principales zones de production légumière en Algérie, notamment les wilayas côtières et la plaine de la Mitidja, où les épisodes d'humidité élevée favorisent leur propagation rapide. L'ensemble de ces maladies représente ainsi une contrainte majeure pour la productivité et la stabilité de la filière tomate dans le pays [11].

La détection précoce et précise de ces maladies représente un enjeu agronomique et économique de premier ordre. Toutefois, les approches diagnostiques traditionnelles basées sur l'analyse visuelle réalisée par des spécialistes en agronomie souffrent de défauts majeurs : évaluation subjective, dépendance à un savoir-faire rare et coûteux, délais incomplètes à la vitesse de diffusion des pathogènes, et incapacité à garantir une inspection périodique à l'échelle d'une exploitation agricole. Ces contraintes sont particulièrement marquées dans le cadre algérien, où la grande majorité du secteur agricole est constituée de petites et moyennes fermes ayant un accès limité aux services phytosanitaires spécialisés [2].

Face à ces défis, les avancées récentes en vision artificielle et en apprentissage profond (deep

learning) offrent des perspectives prometteuses pour l'automatisation du diagnostic phytosanitaire. Les réseaux de neurones convolutifs (CNN) ont démontré des capacités remarquables dans l'analyse d'images agricoles. Mohanty et al. [12] ont été parmi les premiers à démontrer le potentiel des CNN profonds pour classifier les maladies de plantes avec une haute précision sur le dataset PlantVillage. Une revue systématique de 160 articles publiés entre 2020 et 2024 recense des modèles atteignant des précisions allant jusqu'à 99,83 % sur des tâches de classification foliaire [13].

Dans ce contexte, plusieurs structures basées sur le transfert d'apprentissage ont été spécifiquement ciblées pour notre attention. VGG19, une architecture profonde de 19 couches développée par Simonyan et Zisserman, est considérée comme une référence dans le domaine de l'extraction de caractéristiques visuelles complexes [13]. MobileNetV2, grâce à son architecture de blocs résiduels inversés et de convolutions séparables en profondeur qui convient bien aux environnements à ressources limitées, propose un équilibre performant entre précision et légèreté computationnelle [14].

EfficientNetB3, qui repose sur le principe de l'augmentation composée, offre la possibilité d'optimiser en même temps la profondeur, la largeur et la résolution du réseau, réalisant ainsi des performances exceptionnelles à un coût paramétrique réduit [13].

En parallèle, les architectures de détection d'objets en temps réel de la famille YOLO (You Only Look Once) permettent non seulement d'identifier la nature de la maladie, mais aussi de la localiser précisément sur l'image via des boîtes englobantes, ce qui permet de détecter plusieurs lésions en même temps sur une seule feuille. Des recherches récentes signalent des performances pouvant aller jusqu'à mAP@0,5 de 91,2 % pour la généralisation [15], et jusqu'à 98,56 % de précision pour la détection des maladies des tomates [16].

Dans cette approche, nous proposons d'exploiter quatre architectures d'apprentissage profond, à savoir VGG19, MobileNetV2, EfficientNetB3 et YOLOv8n, pour la détection et la classification des maladies foliaires de la tomate à partir du dataset PlantVillage.

Les objectifs spécifiques sont les suivants :

1. Étudier et comparer les architectures VGG19, MobileNetV2 et EfficientNetB3 pour la tâche de classification des maladies foliaires de la tomate sur le dataset "PlantVillage", en évaluant leurs performances respectives selon les métriques de précision, rappel, F1-score et matrice de confusion.
2. Entraîner et évaluer le modèle YOLOv8n pour la détection d'objets sur le dataset "PlantVillage Object Detection", en vue d'une localisation et d'une classification simultanées des pathologies via des boîtes englobantes.
3. Effectuer une analyse comparative des quatre architectures étudiées et les confronter aux résultats des travaux antérieurs, afin d'identifier le modèle le plus adapté à un déploiement embarqué en conditions réelles.

Ce mémoire est organisé en trois chapitres.

1. Le premier chapitre présente l'état de l'art : Historique de l'intelligence artificielle, fondements de l'apprentissage automatique et profond, architecture des réseaux de neurones convolutifs, méthodes de préparation des données et métriques d'évaluation.
2. Le deuxième chapitre décrit la méthodologie adoptée : Présentation des datasets utilisés, étapes de prétraitement et d'augmentation des données, description des quatre architectures étudiées, protocoles d'entraînement et choix des hyperparamètres, ainsi que les outils et l'environnement de développement.

3. Le troisième chapitre présente et discute les résultats obtenus : Analyse des courbes d'apprentissage, des matrices de confusion et des métriques pour chaque modèle, synthèse comparative, et confrontation avec les travaux antérieurs de la littérature.

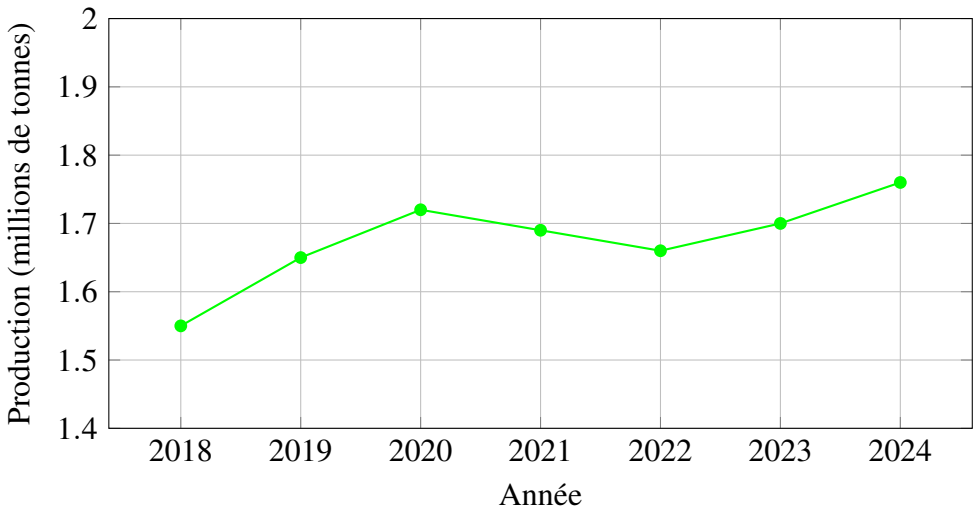


FIGURE 1 – Évolution de la production de tomate en Algérie (2018–2024).

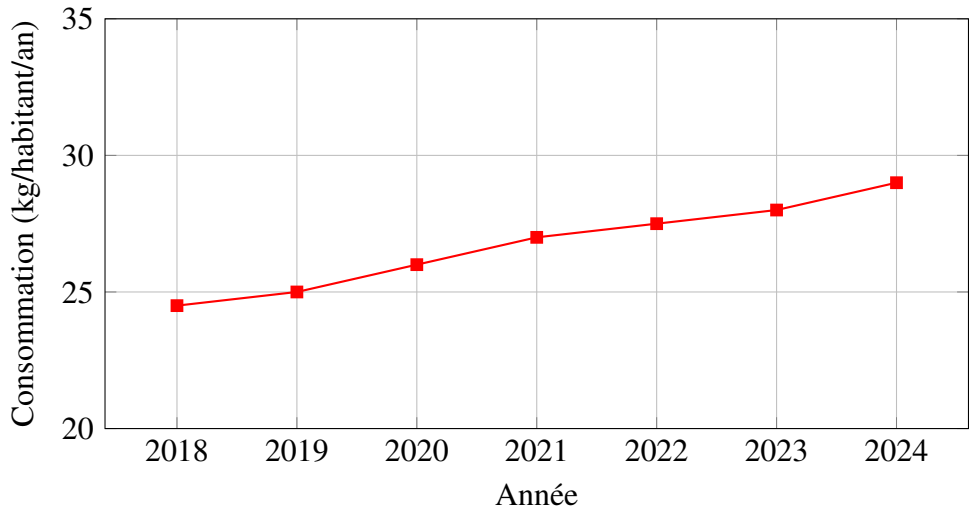


FIGURE 2 – Évolution estimée de la consommation de tomate en Algérie.

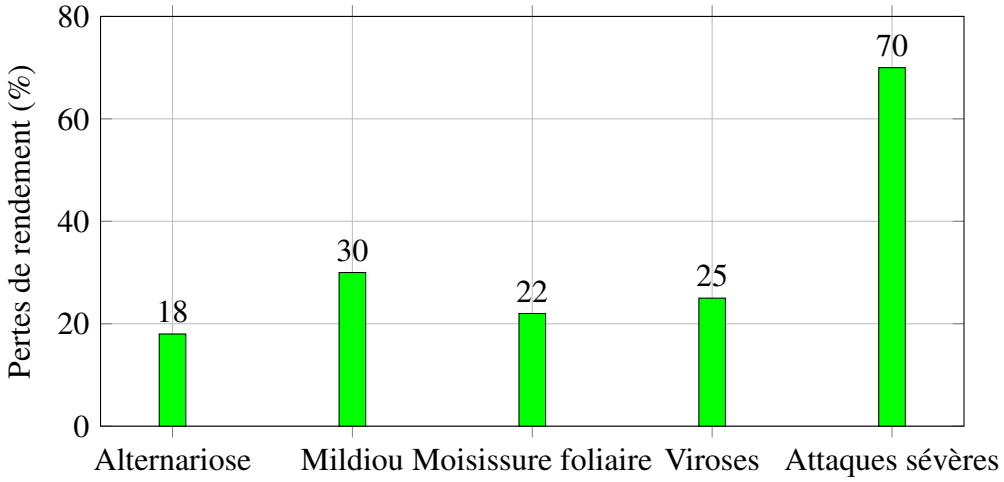


FIGURE 3 – Pertes de rendement causées par les principales maladies de la tomate.

Chapitre 1: Généralités sur l'Intelligence Artificielle et l'Apprentissage Profond

1.1 Introduction

L'agriculture mondiale est confrontée à des défis majeurs tels que le changement climatique et l'augmentation de la demande alimentaire, tandis que les maladies des plantes entraînent des pertes pouvant atteindre 40% des récoltes annuelles [17] [18].

Traditionnellement, la détection repose sur un examen visuel manuel effectué par des experts, une méthode fastidieuse et coûteuse, souvent sujette à des biais humains. C'est une méthode lente et souvent imprécise. L'intégration de l'intelligence artificielle (IA) a ouvert la voie à des systèmes automatisés capables de surveiller la santé des cultures.

1.2 Historique de l'Intelligence Artificielle

L'IA n'est pas une technologie toute nouvelle comme tout le monde le pense, l'IA est une technologie qui évolue depuis les 70 dernières années, des simples casse-têtes mathématiques aux réseaux neuronaux puissants d'aujourd'hui.

Le début remonte à 1943, lorsque Warren McCulloch et Walter Pitts ont développé le premier modèle mathématique d'un réseau de neurones [19]. En 1950, un mathématicien britannique, Alan Turing, a développé le test de Turing. Ce test, également connu sous le nom de « jeu de l'imitation », est conçu pour déterminer si une machine peut exhiber un comportement intelligent similaire de celui d'un humain. Le test implique un interrogateur qui pose des questions à la fois à un humain et à un ordinateur. Turing a été une figure importante dans les domaines des mathématiques, de la logique et du développement de l'informatique et de l'intelligence artificielle [20].

En 1952, Arthur Samuel développa l'un des premiers programmes capables d'apprendre à jouer ans intervention humaine aux dames de manière autonome. Qatre ans plus tard, en 1956, la première conférence officielle consacrée à l'intelligence artificielle a été organisée à l'invitation de John McCarthy, considéré comme le «père de l'IA». En 1959, Samuel introduisit le terme Machine Learning (apprentissage automatique) pour désigner la capacité des machines à acquérir des connaissances par l'expérience [19].

Toutefois, les années 1970 furent marquées par un ralentissement brutal, notamment à la suite du rapport « Lighthill » du gouvernement britannique, qui entraîna une réduction des financements et inaugura ce que l'on appelle « l'hiver de l'IA » [19].

L'apprentissage automatique connut un regain d'intérêt dans les années 1990 grâce au développement d'algorithmes tels que les Support Vector Machines (SVM) et aux progrès dans la reconnaissance vocale et visuelle. En 1997, l'ordinateur DeepBlue d'IBM réussit à vaincre le champion du monde d'échecs Garry Kasparov, événement marquant dans l'histoire de l'IA [19].

Le véritable tournant survint en 2006, lorsque Geoffrey Hinton et son équipe relancèrent l'étude des réseaux neuronaux profonds, ouvrant la voie à la révolution du deep learning. Au début des années 2010, la synergie entre le Big Data et les capacités de calcul intensif offertes par les GPU permit des

avancées spectaculaires dans la reconnaissance d'images, le traitement du langage naturel et les jeux stratégiques. L'exemple le plus emblématique reste la victoire d'AlphaGo de DeepMind contre un champion du monde de Go en 2016 [19].

Aujourd'hui, l'IA et ses déclinaisons : apprentissage automatique et apprentissage profond sont intégrées dans de nombreux secteurs tels que la médecine, la finance, les transports et le divertissement. Elles continuent de transformer profondément la société et nos modes de travail, confirmant leur rôle central dans l'innovation et le développement technologique.

1.3 Sous-domaines de l'Intelligence Artificielle

L'IA se compose de nombreuses sous-domaines, chacune utilisant différentes méthodes pour faire en sorte que la machine imite l'intelligence des êtres humains. Chaque sous-domaine offre des avantages et des perspectives uniques, en se concentrant sur différents aspects de son développement et de sa mise en œuvre. Les chercheurs et les praticiens peuvent améliorer les capacités des systèmes d'IA en se spécialisant dans ces domaines ou même en les combinant pour traiter des problèmes complexes de manière plus efficace et précise.

Il y a cinq sous-domaines principaux : l'apprentissage automatique, l'apprentissage profond, le traitement du langage naturel, la vision par ordinateur et la robotique.

- **Apprentissage automatique :** Le Machine Learning (ML) désigne un sous-ensemble la plus vaste de l'intelligence artificielle, il permet à un algorithme d'apprendre à partir de données sans être explicitement programmé pour chaque tâche [21] ; il construit un modèle capable de faire des prédictions ou de prendre des décisions (ex.détection de maladies à partir d'images). Les algorithmes ML sont choisis en fonction :
  - Modèles simples (régression logistique, SVM) : rapides à entraîner, adaptés à des jeux de données modestes.
  - Modèles complexes (CNN, ResNet, MobileNet) : nécessitent davantage de données et de calcul, mais offrent une meilleure capacité d'extraction de caractéristiques visuelles.
- **Apprentissage profond :** L'apprentissage profond est un sous-ensemble de l'apprentissage automatique, reposant sur des réseaux de neurones artificiels multicouches dont la conception s'inspire de la structure du cerveau humain. Les modèles d'apprentissage profond alimentent la plupart des systèmes d'intelligence artificielle de pointe actuels, de la vision par ordinateur à l'IA générative, en passant par les véhicules autonomes et la robotique [22].
- **Taitement du langage naturel :** Le traitement automatique du langage naturel (TALN) est un domaine qui a récemment suscité beaucoup d'intérêt pour la représentation et l'analyse computationnelle du langage humain. On a trouvé des applications dans beaucoup de domaines comme la traduction automatique, la détection de spams, l'extraction d'information, le résumé automatique, la médecine, les systèmes de questions-réponses [23].
- **Vision par ordinateur :** La vision par ordinateur vise à fournir aux machines la capacité de percevoir et d'interpréter les informations visuelles du monde. Ce domaine a évolué de simples techniques d'enregistrement de données vers des systèmes sophistiqués intégrant le traitement d'images numériques, la reconnaissance de formes, l'apprentissage automatique et l'infographie [24]. Cela inclut la détection d'objets, la segmentation d'images, la reconnaissance faciale et la compréhension des scènes, permettant aux systèmes d'IA de traiter et de répondre aux informations visuelles en temps réel [19].
- **Robotique :** Les robots sont largement utilisés dans des industries telles que la fabrication automobile pour effectuer des tâches simples et répétitives, et dans des industries où le travail doit être réalisé dans des environnements dangereux pour les humains [25]. En combinant l'in-

telligence artificielle avec les systèmes mécatroniques des robots, nous créons des machines capables d'interagir avec leur environnement et de le manipuler. Cela englobe la perception robotique, la planification des mouvements, les systèmes de contrôle et l'interaction homme-robot [26].

1.4 Apprentissage automatique (Machine Learning)

L'apprentissage automatique est un sous-domaine de l'IA, qui consiste à reproduire le comportement intelligent de l'être humain, à créer des modèles informatiques capables d'apprendre et de formuler des prévisions, afin de prendre des décisions de manière autonome à partir des données fournies. Ces modèles améliorent continuellement leur précision grâce aux données dont ils tirent des enseignements.

Arthur Samuel, à qui l'on doit le terme « apprentissage automatique », a utilisé cette expression en 1959 pour décrire « la capacité des ordinateurs à acquérir de nouvelles compétences sans programmation directe » [27]. Un algorithme d'apprentissage automatique, basé sur un modèle mathématique et à partir des ensembles de données disponibles, génère des prédictions ou des décisions spécifiques [28]. Les deux types majeurs d'apprentissage automatique sont l'apprentissage supervisé et l'apprentissage non supervisé.

— **l'apprentissage supervisé :** Le modèle d'apprentissage supervisé est un type d'apprentissage automatique où le modèle est formé sur un jeu de données étiquetées, ce qui signifie que chaque exemple d'entrée correspond à une sortie cible préétablie. L'objectif consiste à apprendre une fonction de correspondance (mapping) $f : X \rightarrow Y$ qui permettra de prédire la sortie Y pour toute nouvelle entrée X inconnue. Ce genre d'apprentissage comprend deux missions principales : la classification et la régression [29].
— **l'apprentissage non supervisé :** L'apprentissage non supervisé est un modèle d'entraînement où le système se forme sur des données qui ne sont pas annotées. En l'absence de cible prédéfinie, l'algorithme s'efforce de déceler lui-même des structures cachées, des motifs ou des regroupements naturels à l'intérieur des données. Les missions principales comprennent le partitionnement, la réduction de la dimensionnalité et l'identification des anomalies.

1.5 Apprentissage profond (Deep Learning)

Sous-domaine de l'apprentissage automatique qui utilise des réseaux neuronaux profonds à plusieurs couches (par exemple, les CNN). Cela permet au réseau de comprendre et de reproduire des comportements plus complexes et plus concrets [28], et en plus de extraire automatiquement des représentations hiérarchiques de caractéristiques directement à partir des données brutes [19].

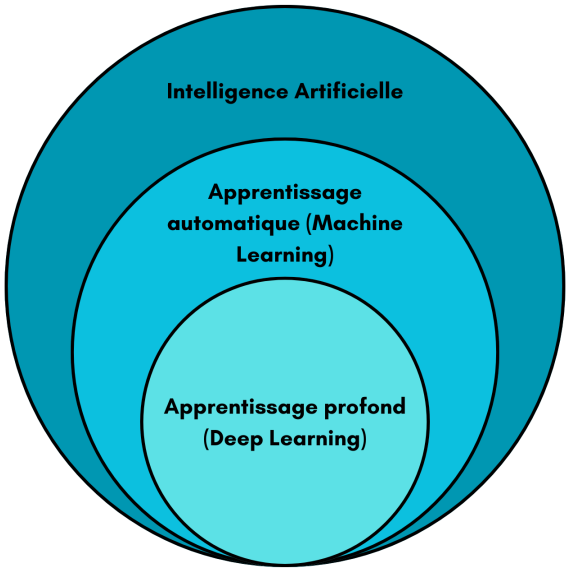


FIGURE 1.1 – Schéma de relation entre Machine Learning, Deep Learning et Artificial intelligence.

1.5.1 Réseaux de Neurones Artificiels (ANN)

Un modèle informatique basé sur la structure et les fonctions des réseaux neuronaux biologiques est connu sous le nom de réseau de neurones artificiels (RNA) [30]. S'inspirant du cerveau humain (Figure 6), ils se composent plusieurs perceptrons ou « neurones » qui traitent et transmettent l'information [28].

Ce sont des outils statistiques non linéaires de modélisation de données, utilisés pour décrire des interactions complexes entre les entrées et les sorties ou pour identifier des motifs.

Ces données traversant le réseau, où elles sont traitées par des couches successives de perceptrons, dont les interconnexions sont régies par des poids et des biais appris, jusqu'à atteindre la couche de sortie. Chaque couche comprend plusieurs neurones qui transforment les données entrantes [28].

1.5.2 Composants des Réseaux de Neurones Artificiels

— **Neurons :** L'unité basique d'un réseau neurones, connu aussi comme un perceptron [19]. Chaque neurone reçoit un signal d'entrée pour traiter à travers une calcule de somme pondérée de ses entrées, à laquelle s'ajout un biais. Puis applique une fonction d'activation.

$$a = \phi(\sum_i w_i x_i + b)$$

L'ensemble des neurones de même étape produit un couche.

— **Couche :** Une couche d'un réseau neuronal est constituée de neurones. Chaque neurone reçoit un vecteur contenant toutes les entrées, multiplie ces entrées par leurs poids respectifs, additionne les résultats pour obtenir une somme pondérée (calculée), puis ajoute cette somme pondérée à son propre biais. À la fin, chaque nœud « neurone » produit un scalaire. Tous les scalaires sont ensuite assemblés pour former un vecteur, qui devient l'entrée de la couche suivante (propagation vers l'avant) [31].

— **Poids et biais :** Les poids représentent la force des connexions entre les neurones. Ce sont

des paramètres a entraînées, ils sont initialisés de manière aléatoire et ajustés au cours de l'apprentissage afin de minimiser l'erreur de prédiction. Les biais, d'autre part, sont des valeurs constantes ajoutées à la somme pondérée des entrées. Ils permettent au modèle de mieux s'adapter aux données d'apprentissage en modifiant le seuil d'activation [19]. La représentation standard de poids est et de biais est [31].

— **Fonctions d'activation :** La fonction d'activation est une partie du réseau neuronal artificiel (RNA) qui convertit une entrée en une sortie donnée. Elle sert à activer et à désactiver les neurones [32] et introduire la non-linéaire dans le réseau, ce qui lui permet d'apprendre des relations complexes entre les entrées et les sorties [19]. Nombreuses fonctions d'activation sont utilisées dans les couches des réseaux neuronaux artificiels.

— Fonction ReLU

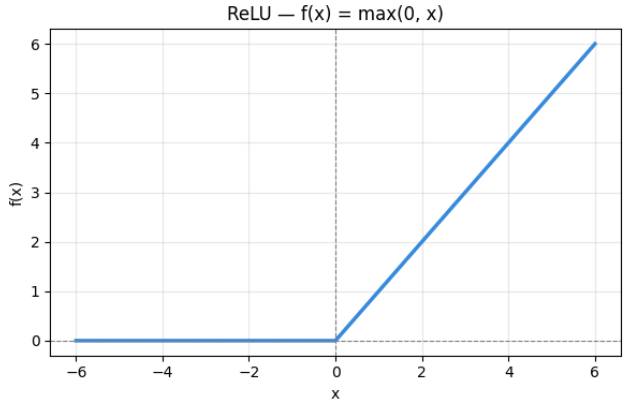


FIGURE 1.2 – Fonction d'activation ReLu.

— Fonction sigmoid

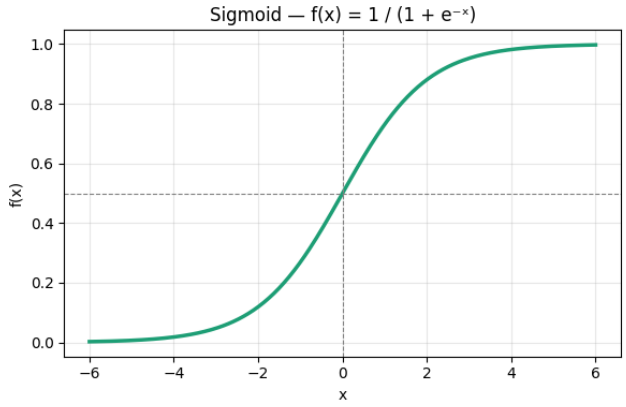


FIGURE 1.3 – Fonction d'activation sigmoid.

— Fonction softmax

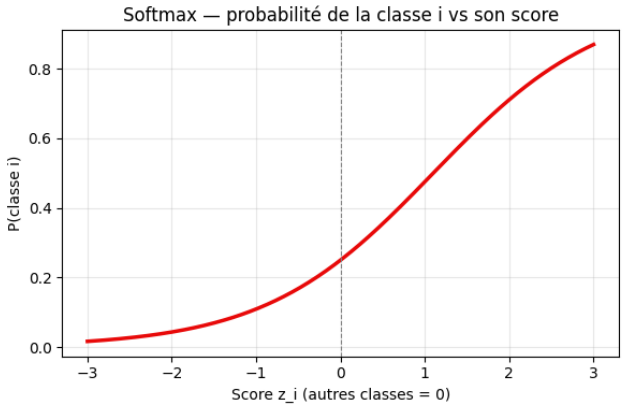


FIGURE 1.4 – Fonction d'activation softmax.

— **Fonction de perte :** Pour obtenir des résultats optimaux, il est important de réduire l'écart entre la prédiction du réseau et la valeur réelle, en calcule par exemple l'erreur quadratique moyenne [31].
— **Process d'optimization :** C'est un processus qui actualise les paramètres de modèle durant l'entraînement, afin de minimiser la fonction de perte [33]. Il s'agit généralement de la descente stochastique du gradient (ou de ses variantes) [19] (SGD), Adam est un autre populaire algorithme d'optimisation.
— **Données (Data) :** L'ensemble d'entrées. Ils peut contient par exemple des images, des information numériques.

1.5.3 Types de Réseaux de Neurones Artificiels

— **Réseau de Neurones à Propagation Avant (Feedforward Neural Network FNN) :** C'est l'architecture la plus simple et la plus fondamentale. Les informations circulent dans un seul sens, de la couche d'entrée vers la couche de sortie, sans cycle ni rétroaction. Il constitue la base des réseaux multicouches (MLP Multi-Layer Perceptron).
— **Réseau de Neurones Convolutifs (Convolutional Neural Network CNN) :** Conçu spécifiquement pour le traitement des données spatiales comme les images. Il exploite des couches de convolution pour extraire automatiquement des caractéristiques locales (bords, textures, formes), réduisant ainsi le nombre de paramètres par rapport à un réseau fully connected.
— **Réseau de Neurones Récurrents (Recurrent Neural Network RNN) :** Contrairement au FNN, le RNN possède des connexions cycliques qui lui permettent de conserver une mémoire des entrées précédentes. Il est particulièrement adapté aux données séquentielles et temporelles.

1.6 Réseaux de Neurones Convolutifs (CNN)

Un CNN est une classe de réseaux neuronaux profonds, il utilise des couches convolutives avec des champs réceptifs locaux et le partage des poids, pour apprendre des représentations hiérarchiques de caractéristiques à partir de données structurées en grille (images) [34]. Les CNN servent à identifier des images, à les regrouper en fonction de leur similarité et aider à identifier des images et à les étiqueter de manière appropriée [33]. Donc, un CNN est utilisé principalement dans le domaine de la vision par ordinateur et de l'imagerie. Un CNN comporte différentes couches, à savoir la couche

d'entrée, la couche de sortie et plusieurs couches cachées. Ces couches cachées d'un CNN se composent de couches entièrement connectées, de couches convolutives, d'une couche servant de fonction d'activation, de couches de normalisation et de couches de pooling et flatten [33].

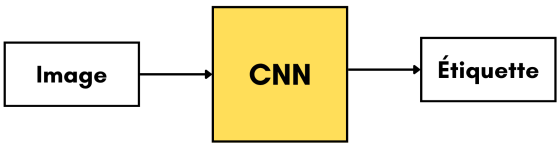


FIGURE 1.5 – CNN générale.

1.6.1 Architecture d'un CNN

Les principaux composants d'une architecture CNN sont les suivants :

— **Image d'entrée :** Le premier élément de l'architecture d'un CNN est l'image d'entrée ; pour que l'ordinateur comprenne cette image, elle doit être convertie sous la forme d'une matrice de zéros et de uns[33].

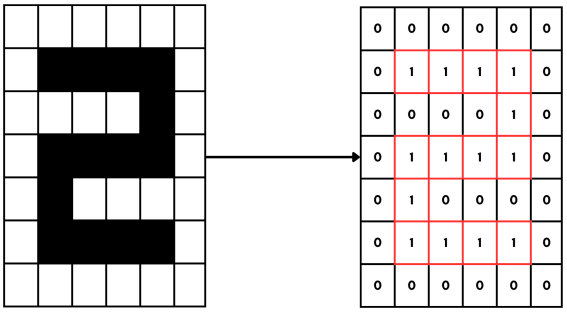


FIGURE 1.6 – Matrice de numéro 2.

— **Couche convolutive :** C'est l'élément constitutif central [35] où le processus commence, il comprise des filtres ou «kernels» de petit dimensions sert à extrait les caractéristiques d'un image d'entrée afin de classifier. Pendant la convolution, le filtre parcourt la hauteur et la largeur de l'image, effectuant à chaque position spatiale le produit scalaire entre ses éléments et les données correspondantes [36].

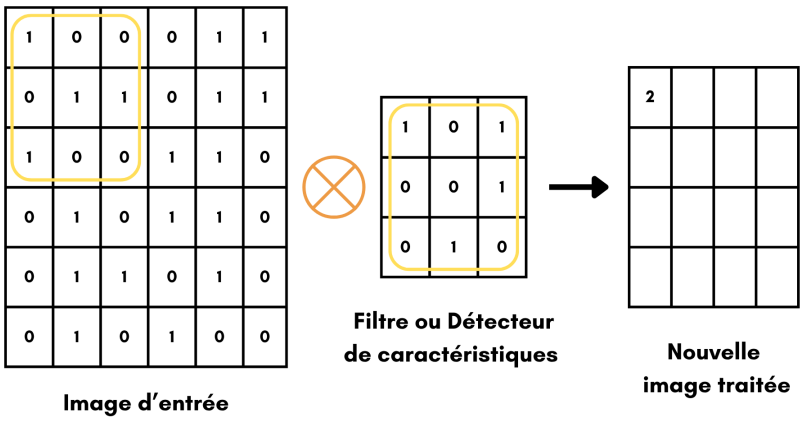


FIGURE 1.7 – Filtre parcout un image.

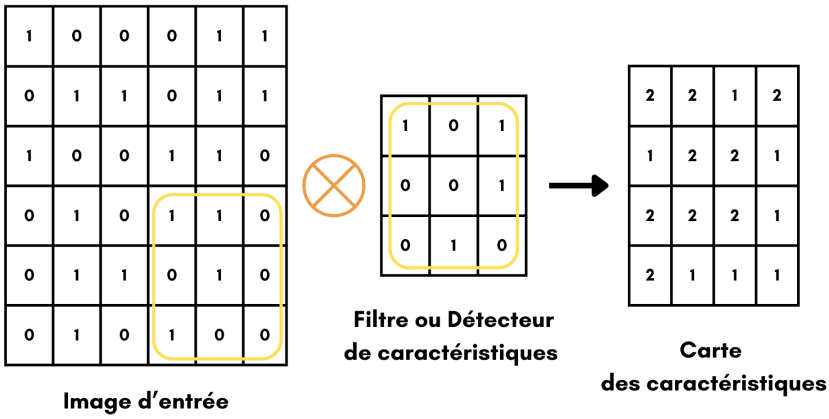


FIGURE 1.8 – Résultat d'application le filtre «feature map».

Le résultat s'appel «feature map».

— **Couche de pooling :** La couche de regroupement (également appelée sous-échantillonnage) réduit progressivement les dimensions des cartes de caractéristiques [36][33], tout en conservant les prospérités importantes. Cela permet d'ignorer les données non utiles. Types courants :
  — **Regroupement maximal :** Sélectionne la valeur maximale dans chaque région.
  — **Regroupement moyen :** Calcule la valeur moyenne.
  — **Regroupement global :** Calcule une valeur unique à partir de l'ensemble de la carte de caractéristiques. Le regroupement maximal est le plus courant. Une fenêtre matricielle 2x2 se déplace sur la carte de caractéristiques, sélectionnant la valeur la plus élevée dans chaque région.

— **Aplatissement (Flatten) :** Dans le contexte des réseaux de neurones convolutifs (CNN), l'aplatissement (Flatten) est une couche intermédiaire qui transforme un tenseur généré lors l'étape pooling à plusieurs dimensions en un vecteur à une dimension. Afin de pouvoir l'utiliser comme entrée dans un réseau neuronal artificiel [37] et connecter les couches convolutives aux couches fully connected (denses).

1.6.2 Types de Réseaux de Neurones Convolutifs (CNN)

La distinction entre les différents problèmes exigent le développement de modèles de réseaux de neurones convolutifs (CNN), chaque type étant adapté à des problèmes spécifiques et à des niveaux de complexité particuliers [36]. Différents types de modèles de CNN ont été proposés. La figure ci-dessous présente quelques-uns des principaux types de CNN en fonction de leur architecture.

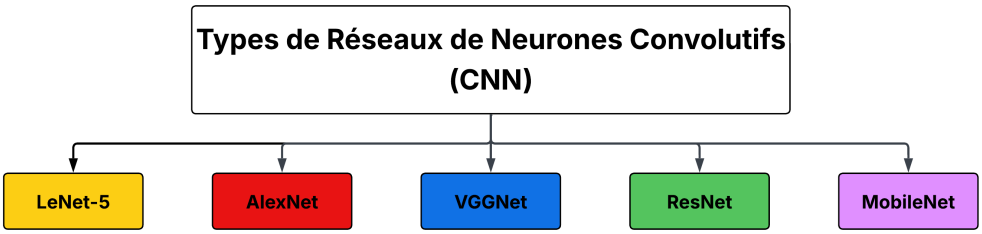


FIGURE 1.9 – Quelque types de Réseaux de Neurones Convolutifs (CNN).

1.7 Préparation des données

Chaque entrée à identifier ou prédire doit parcourt à travers ces étapes :

— **Proportion de répartition (train : validation : test) :** Les ratios de 70%-30% ou 80%-20% sont les plus fréquemment utilisés. Le set d'entraînement favorise l'apprentissage, le set de validation oriente l'ajustement des hyper-paramètres et le set de test offre l'évaluation définitive.

La décision est liée à la dimension du jeu de données : avec plus de 40,000 images (comme dans l'exemple de [38]), on peut opter pour une répartition (70%-15%-15%) afin de garantir un nombre significatif de données à tester tout en évitant le surapprentissage.

Si on traite un ensemble de données contenant des millions d'exemples et que l'objectif est d'entraîner un grand réseau neuronal profond, la séparation entre la validation et le test peut être effectuée selon des ratios de 98%-2% ou 99%-1% [33].

— **Normalisation :** Standardisation par canal : chaque canal (R, G, B) est centré et réduit à une moyenne approximative de 0 et un écart-type approximatif de 1 (par exemple, ImageNet : μ = [0.485, 0.456, 0.406], σ = [0.229, 0.224, 0.225]). Cette modification stabilise les gradients et accélère le processus.

— **Augmentation de données :** Le but de cet étape sert à augmenter la diversité du jeu d'entraînement et à enrichir artificiellement l'ensemble de données en créant des versions modifiées d'images existantes [19]. Cela contribue à réduire le surajustement, rendre le modèle robuste aux variations de pose, d'éclairage et aux artefacts du terrain. Cette approche constitue également une solution lorsqu'on travaille avec une base de données déséquilibrée.

Il existe de nombreuses techniques, parmi lesquelles :

— Transformations géométriques : retournement horizontal, rotation [0, 360°], translation, mise à l'échelle [0.9, 1.1], recadrage aléatoire.
— Transformations de couleur : variations de luminosité, contraste, saturation, teinte, ainsi que l'ajout de bruit gaussien.

1.8 Types de classification

Il existe quatre types de classification [39] :

— **Binaire :** C'est la plus courante ; chaque instance appartient à l'une des deux classes. Exemple : feuille sain vs malade.
— **Multi-classe :** Chaque instance est affectée à une seule classe parmi l classes, avec l>2. Exemple : classification d'une feuille en rouille, mildiou, tâche bactérienne ou sain.
— **Multi étiquette (multi label) :** Une instance peut appartenir simultanément à plusieurs classes. Exemple : une image contenant plusieurs maladies ou plusieurs espèces de parasites annotées en même temps.
— **Hiérarchique :** Les classes sont organisées en hiérarchie et l'instance est classée à différents niveaux de cette hiérarchie. Exemple : d'abord plante, tomate, tâche foliaire, tâche fongique.

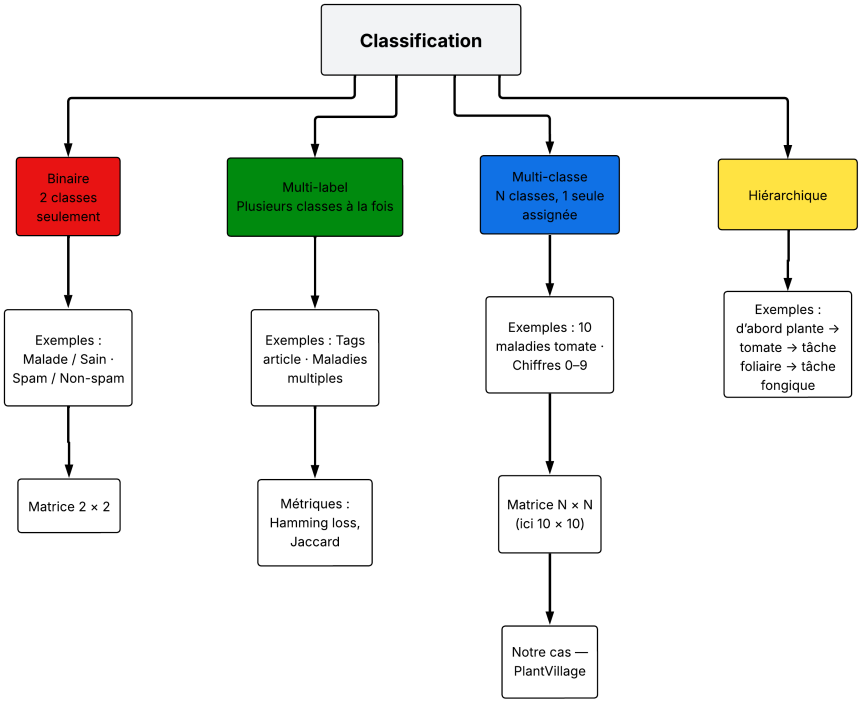


FIGURE 1.10 – Diagramme résumé des types de classification.

1.9 Métriques d'évaluation

L'évaluation de la performance d'un modèle repose sur l'utilisation de la matrice de confusion et de diverses métriques. Ce sont des outils clés qui offrent une visualisation de la capacité et de l'efficacité d'un classificateur à identifier correctement les étiquettes et à détecter les erreurs.

1.10 Évaluation et validation

Pour chaque classe, quatre indicateurs sont évalués, appelés "métriques d'évaluation" : la précision (qui représente le ratio de prédictions justes parmi les positives), le rappel (qui détermine la proportion de positifs réels identifiés), le F1-score (qui est une moyenne harmonique entre la précision et le rappel) ainsi que la précision globale.

Ces métriques sont calculées directement à partir de la matrice de confusion

1.10.1 Matrice de confusion

Représentation graphique qui synthétise le rendement du modèle pour chaque catégorie de maladies des tomates. Les lignes indiquent les classes réelles, tandis que les colonnes montrent les classes anticipées par le modèle. Les valeurs sur la diagonale représentent les classifications correctes, alors que les valeurs en dehors de la diagonale montrent les erreurs de classification. La matrice de confusion aide à repérer les catégories où le modèle éprouve le plus de difficultés. Dans le cadre de l'évaluation des modèles de classification, on distingue quatre types de résultats : Vrai Positif (VP), Vrai Négatif (VN), Faux Positif (FP), Faux Négatif (FN).

$$\text{Matrice de confusion} = \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \tag{1.1}$$

Où :

- **TP (True positif) :** Nombre d'instances qui sont initialement positifs et correctement être prédire positifs.
- **TN (True negatif) :** Nombre d'instances initialement négatives et correctement prédites comme négatives
- **FP (False positif) :** Nombre d'instances initialement négatives mais incorrectement classées comme positives
- **FN (False negatif) :** Nombre d'instances initialement positives mais incorrectement prédites comme négatives.

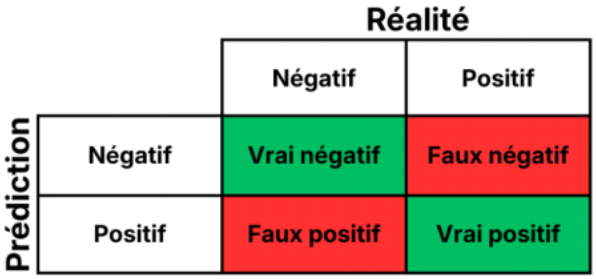


FIGURE 1.11 – Matrice de confusion.

1.10.2 Métriques globales

- **Accuracy : (Précision)** Il correspond du nombre de prédictions correctes divisé par le nombre total de prédictions. D'autre terme, c'est le pourcentage de prédictions correctes par rapport à l'ensemble des prédictions [19]. Dans le cas d'une classification binaire, cette métrique

mesure l'efficacité globale d'un classificateur ; pour le cas d'une classification multiclasses, elle mesure l'efficacité moyenne par classe d'un classificateur [39].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{1.2}$$

— **Perte (loss)**

$$\mathscr{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{1.3}$$

1.10.3 Métriques par classe

1.10.3.1 Classification binaire :

— **Précision :** Corrélation entre les étiquettes de données et les étiquettes positives attribuées par le classificateur, elle indique la fiabilité des prédictions positives [39].

$$\text{Précision} = \frac{TP}{TP+FP} \tag{1.4}$$

— **Rappel (Recall) :** Il évalue l'efficacité d'un classificateur à détecter tous les étiquettes positives [39] [19].

$$\text{Recall} = \frac{TP}{TP+FN} \tag{1.5}$$

— **F1 score :** Est la moyenne harmonique de la précision et du rappel, offrant une mesure équilibrée de ces deux indicateurs. Il correspond au la relations entre les étiquettes positives des données et celles attribuées par un classificateur [39].

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1.6}$$

— **Specificité :** Il mesure la capacité d'un classificateur à identifier efficacement les étiquettes réellement négatives parmi les étiquettes qui sont effectivement négatifs [19][39].

$$\text{Specificité} = \frac{TN}{TN+FP} \tag{1.7}$$

1.10.3.2 Classification multiclasses :

— Précision :

$$\text{Précision}_{\text{micro}} = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} (TP_i + FP_i)} \tag{1.8}$$

— Rappel (Recall) :

$$\text{Recall}_{\text{micro}} = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} (TP_i + FN_i)} \tag{1.9}$$

— F1 score :

$$F_{1,\text{micro}} = \frac{2 \cdot \text{Precision}_{\text{micro}} \cdot \text{Recall}_{\text{micro}}}{\text{Precision}_{\text{micro}} + \text{Recall}_{\text{micro}}} \tag{1.10}$$

— Specificité :

$$\text{Specificité} = \frac{TN}{TN+FP} \tag{1.11}$$

1.11 YOLO et ses évolutions

YOLO (You Only Look Once) est une famille de modèles de détection d'objets en temps réel, introduite par Redmon et al. en 2016 [40]. Contrairement aux approches classiques basées sur des propositions de régions, YOLO traite l'image en une seule passe et prédit directement les boîtes englobantes et les probabilités de classes. Cette approche unifiée permet une rapidité d'inférence remarquable (jusqu'à 45 fps pour YOLOv1, plus de 150 fps pour Fast YOLO) tout en maintenant une précision compétitive.

1.11.1 Évolution des versions

Depuis YOLOv1, plusieurs versions ont été publiées :
- **YOLOv1 (2016) :** première version, rapide mais moins précise.
- **YOLOv2 (YOLO9000) :** amélioration avec détection de 9000 classes.
- **YOLOv3 (2018) :** architecture Darknet-53, meilleure précision.
- **YOLOv4 (2020) :** optimisations pour GPU grand public.
- **YOLOv5 (2020, Ultralytics) :** version pratique, largement adoptée.
- **YOLOv7 (2022) :** amélioration des performances.
- **YOLOv8 (2023, Ultralytics) :** dernière génération, anchor free, multi tâches (détection, segmentation, classification, pose) [41].

1.11.2 Caractéristiques de YOLOv8n

YOLOv8 se distingue par :
- **Anchor free head** : Supprime la dépendance aux ancres, simplifie la détection.
- **Backbone optimisé** : Extraction de caractéristiques plus efficace.
- **Multi tâches** : Détection, segmentation, classification et estimation de pose.
- **Modèles disponibles** : n, s, m, l, x (taille croissante).

1.11.3 Métriques d'évaluation

Les performances des modèles YOLO sont évaluées par :

- **IoU (Intersection over Union) :** C'est un rapport entre l'aire d'intersection et l'aire d'union de la boîte prédite et de la boîte réelle.
- **AP (Average Precision) :** Mesure de performance correspondant à l'aire sous la courbe précision, rappel pour une classe donnée.
- **AP@50 :** Average Precision ou bien moyen de précision, calculée avec un seuil d'IoU de 0,5.
- **AP@75 :** Average Precision calculée avec un seuil d'IoU de 0,75.
- **mAP (Mean Average Precision) :** Elle présente une moyenne des valeurs d'AP obtenues sur l'ensemble des classes.
- **mAP@50 :** Moyenne des valeurs d'AP@50 sur toutes les classes.
- **mAP@50 :95 :** Moyenne des valeurs d'AP calculées pour des seuils d'IoU variant de 0,5 à 0,95, avec un incrément de 0,05, conformément au protocole d'évaluation COCO.

Exemple de performances sur le dataset COCO (YOLOv8) [41] :

TABLE 1.1 – Performances YOLOv8 sur COCO.

Modèle	**mAP@50–95**	**Params (M)**
YOLOv8n	37.3	3.2
YOLOv8s	44.9	11.2
YOLOv8m	50.2	25.9
YOLOv8l	52.9	43.7
YOLOv8x	53.9	68.2

1.12 Conclusion

Dans ce chapitre, nous avons présenté les bases théoriques nécessaires à la compréhension de notre travail. Nous avons débuté avec une revue historique de l'intelligence artificielle, avant de nous pencher sur ses principaux sous-domaines, notamment l'apprentissage automatique et l'apprentissage profond.

Par la suite, nous avons concentrés sur les réseaux de neurones, leurs éléments constitutifs et leurs diverses catégories, en accordant une attention spécifique aux réseaux de neurones convolutifs (CNN), lesquels représentent le noyau des systèmes contemporains de classification d'images. Enfin, nous avons abordé les aspects liés à la préparation des données, aux types de classification ainsi qu'aux métriques d'évaluation utilisées pour mesurer les performances d'un modèle.

L'ensemble de ces notions forme la base théorique sur lequel repose notre système de détection et de classification des maladies des plantes, qui sera détaillé dans les chapitres suivants.

Chapitre 2: Méthodologie : Classification par Transfert d'Apprentissage et Détection par YOLOv8n

2.1 Introduction

Dans ce chapitre, nous présentons la méthodologie mise en œuvre pour la détection et la classification des maladies des plantes de culture alimentaire en utilisant les techniques d'apprentissage profond. Notre étude se concentre spécifiquement sur l'évaluation des performances de plusieurs modèles de deep learning appliqués à un problème de classification d'images, en se basant sur la base de données « PlantVillage ».

Afin d'atteindre cet objectif, nous avons choisi d'utiliser différentes architectures de réseaux de neurones convolutifs, notamment VGG19, MobileNetV2, EfficientNetB3 et YOLOv8n reconnues pour leur efficacité dans les tâches de vision artificielle.

Les performances de ces modèles seront analysées et comparées à travers plusieurs métriques d'évaluation, telles que l'accuracy, la fonction de perte (loss), la précision (precision), le Rappel (recall), le F1-score, la matrice de confusion ainsi que d'autres indicateurs pertinents, afin de déterminer leur capacité à assurer une classification précise des maladies des plantes.

2.2 Préparation des ensembles des données

2.2.1 Source des images : Dataset PlantVillage

Dans le cadre de notre projet le jeu de données utilisé est PLantVillage reconue dans le domaine de la detection et la classification des maladies des plantes de culture alimentaire par vision artificielle Cette base de données contient une collection d'images attentivement sélectionnées, prises dans des conditions spécifiques pour l'entraînement de modèles d'intelligence artificielle. Le jeu de données PlantVillage comprend 38 classes de plantes avec 54 303 images de feuilles saines et malades réparties sur ces classes [42] [43]. Notre recherche se focalise uniquement sur les images de feuilles de tomate, comprenant 18 160 images réparties sur 10 classes distinctes :

TABLE 2.1 – Répartition des images par classe de maladie dans PlantVillage.

N°	Classe / Maladie	Nombre d'images
1	Bacterial Spot	2127
2	Early Blight	1000
3	Late Blight	1909
4	Leaf Mold	952
5	Septoria Leaf Spot	1771
6	Spider Mites (Two-spotted)	1676
7	Target Spot	1404
8	Tomato Mosaic Virus	373
9	Yellow Leaf Curl Virus	5357
10	Healthy (Sain)	1591

2.2.2 Filtrage des classes Tomate

Un script de filtrage vérifie automatiquement les répertoires train/ et val/ du dataset original PlantVillage, puis ne copie que les dossiers dont le nom contient la chaîne "Tomato". Les images issues de ces deux répertoires sont ensuite regroupées dans un dossier unique nommé tomato_combined/, avant d'être réorganisées selon notre propre méthode de division.

2.2.3 Nettoyage des images corrompues

Avant d'être traitée, chaque image est examinée une par une. Les fichiers dont l'extension est incorrecte (.jpg, .jpeg, .png, .bmp, .tiff) ou les images qui ne peuvent pas être lues sont supprimés automatiquement. Ce processus aide à prévenir les erreurs pendant l'entraînement du modèle.

Dans notre cas, le jeu de données (dataset) utilisé est constitué uniquement d'images au format (.jpg).

2.2.4 Partitionnement

Une étape essentielle dans la préparation des données consiste à diviser la base de données en ensembles distincts destinés aux objectifs d'entraînement, de validation et de test. La répartition des données.

Le paquet Scikit-learn permet d'utiliser parmi ces ensembles de données. Plus précisément, 60% des données sont destinées à l'entraînement, tandis que 20% sont réservés pour la validation et le dernier 20% pour [44].

2.2.5 Prétraitement des données

Bien que toutes les images présentes dans notre base de données soient de taille identique et sans bruit, qui faciliter la reconnaissance de la maladie pour le modèle, cependant, nous rencontrons un problème lorsque l'utilisateur télécharge des images depuis son téléphone. Il est possible que ces images aient une taille différente de celle du jeu de données ou contiennent du bruit, ce qui rend la reconnaissance de la maladie difficile. Pour résoudre ce problème, nous devons effectuer un prétraitement des images avant de les introduire dans le modèle [45]. Le prétraitement se réfère à une série d'opérations destinées à redimensionner et normaliser les images, ainsi qu'à améliorer leurs caractéristiques.

2.2.6 Augmentation des données (train uniquement)

Pour augmenter la classe minoritaire, l'augmentation de données est appliquée uniquement sur le jeu d'entraînement afin d'enrichir artificiellement la diversité des exemples fournis au modèle, sans recourir à de nouvelles acquisition d'images [44]. On dispose de plusieurs techniques d'augmentation de données, y compris l'application de transformations géométriques telles que le recadrage, cisaillement, la rotation, translations horizontale et verticale et autres sur les images constituant l'ensemble de données utilisé pour l'apprentissage [46].

2.2.7 Normalisation

Ajustez les valeurs de pixels des images pour qu'elles se situent dans une plage uniforme. Cela peut impliquer l'ajustement des valeurs de pixels à une plage spécifique (par exemple, 0-1) [45].

2.2.8 Redimensionnement

C'est l'ajustement des images à une taille précise (selon l'architecture) conforme aux exigences d'entrée du modèle. Cela garantit un équilibre dans les dimensions des images et évite les problèmes provenant de tailles d'images différentes [45].

2.3 Contribution 1 : Modèles d'apprentissage de transfer

La figure 2.1 démontre le processus adopté pour l'approche de classification via l'apprentissage par transfert. Ce processus se compose de plusieurs étapes successives :

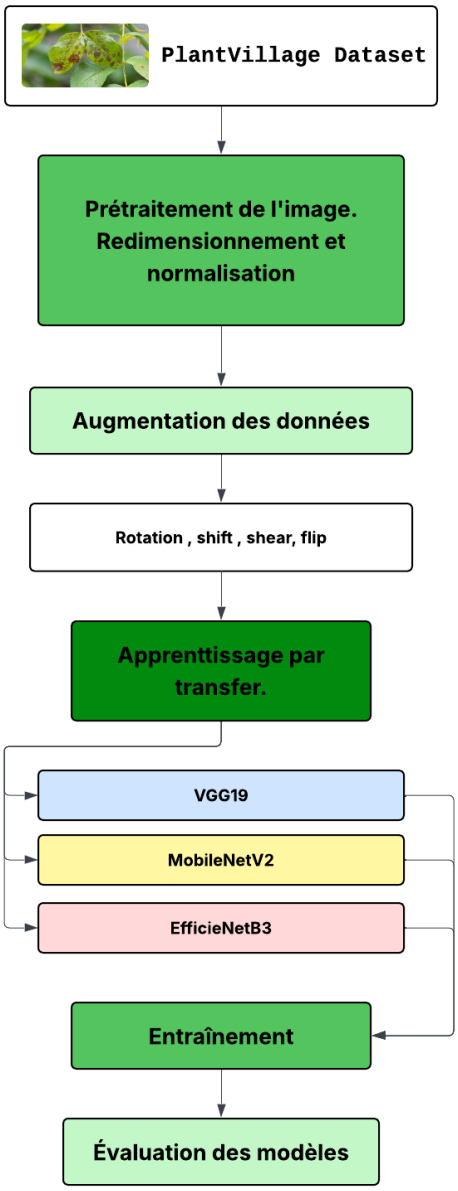


FIGURE 2.1 – Pipeline de classification : Modèles d'apprentissage de transfer (PlantVillage).

1. **Données d'entrée :** Le dataset utilisé est (PlantVillage), un ensemble d'images de feuilles de plantes étiquetées selon leur état de santé et leurs variétés.
2. **Prétraitement de l'image :** Chaque image est redimensionnée à une résolution fixe répondant aux exigences du modèle utilisé, puis normalisée pour réduire les valeurs des pixels.
3. **Augmentation des données :** Diverses transformations géométriques sont appliquées afin d'enrichir le jeu d'entraînement et d'améliorer la généralisation des modèles : rotation, translation, cisaillement (shear) et retournement horizontal (flip).
4. **Apprentissage par Transfer Learning (Fine-Tuning complet) :** Trois architectures CNN pré-entraînées sur ImageNet sont exploitées : VGG19, MobileNetV2 et EfficientNetB3. Avec l'intégralité de leurs couches extraites. Leurs couches convolutives extraient automatiquement les caractéristiques visuelles pertinentes (textures, formes, motifs), tandis qu'une nouvelle tête de classification est adaptée au nombre de classes de PlantVillage.
5. **Entraînement :** Les trois modèles sont entraînés sur les données augmentées en mode fine-tuning complet, en utilisant les poids pré-entraînés sur ImageNet comme point de départ.
6. **Évaluation des modèles :** Chaque architecture est évaluée à l'aide de la précision, du rappel

(recall), du F1-score et de la matrice de confusion.

2.3.1 Modèle VGG19 (Visual Geometry Group 19)

VGG19 est une version du modèle VGG qui se compose essentiellement de 19 couches (16 couches de convolution, 3 couches entièrement reliées, 5 couches MaxPool et 1 couche SoftMax). Voici l'architecture du VGG19 :

Ce réseau a reçu une image RGB de dimensions fixes (224x224) en entrée, ce qui implique que la matrice avait une structure de (224, 224, 3). La seule étape de prétraitement réalisée consiste à déduire la valeur RGB moyenne de chaque pixel, déterminée sur la totalité du jeu de données d'apprentissage. L'utilisation de noyaux de dimension (3x3) avec un pas de 1 pixel a permis de couvrir l'ensemble du concept d'image.

Un remplissage spatial a été employé afin de maintenir la résolution spatiale de l'image.

Un pooling maximal a été appliqué sur une fenêtre de 2x2 pixels avec un pas de 2. Ceci a été suivi par l'activation rectifiée linéaire (ReLu) pour introduire la non linéarité, ce qui permet une meilleure classification du modèle et une amélioration du temps de calcul, étant donné que les modèles antérieurs se servaient de fonctions tanh ou sigmoïde.

L'implémentation consistait en trois couches entièrement connectées, les deux premières ayant une dimension de 4096. Ensuite, une couche comportant 1000 canaux a été ajoutée pour la classification ILSVRC à 1000 classes, suivie d'une fonction softmax en tant que couche finale [47].

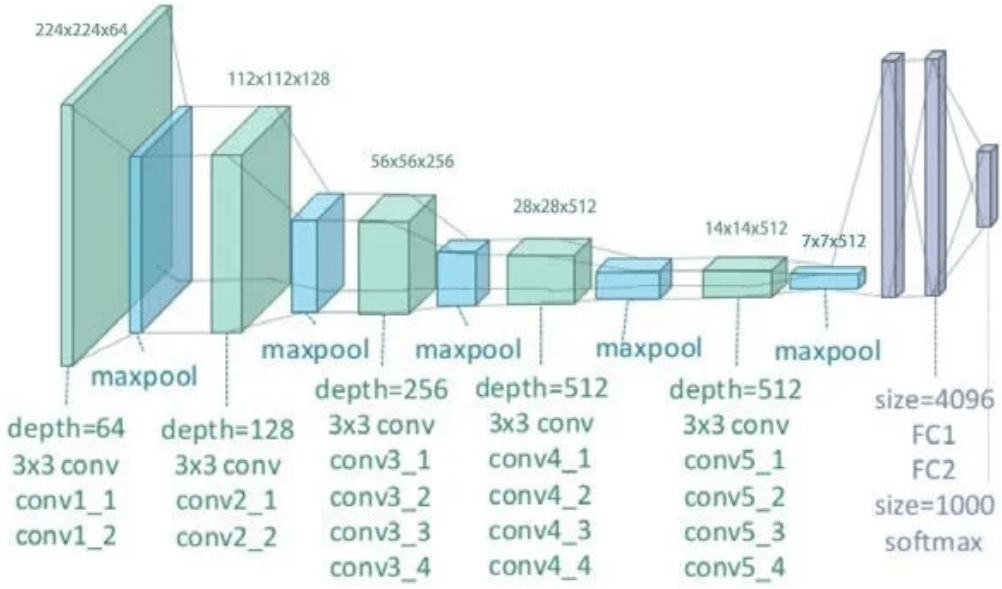


FIGURE 2.2 – Schéma d'architecture de modèle VGG19 [48].

2.3.2 Modèle MobileNetV2

MobileNetV2 est une architecture de réseau neuronal convolutif qui vise à offrir des performances optimales sur les appareils mobiles. Elle se fonde sur une structure résiduelle inversée où les connexions résiduelles se situent entre les couches de goulot d'étranglement. La couche d'expansion intermédiaire utilise des convolutions à profondeur légère pour filtrer les caractéristiques en tant que source de non-linéarité.

L'architecture de MobileNetV2 comprend la première couche entièrement convolutive dotée de 32 filtres, suivie de 19 couches résiduelles en forme de goulet d'étranglement [49]. Le MobileNet v2 est présenté comme une version améliorée du MobileNetV1, le rendant encore plus performant et efficace.

Dans le MobileNetV2 architecture, le bloc de convolution séparablement profond. Le nouveau bloc de convolution séparée en profondeur comprend trois couches de convolution. La première couche est une couche de convolution 1×1 qui sert à augmenter le nombre de canaux dans la carte des caractéristiques d'entrée avant qu'elle ne soit traitée par la couche de convolution séparée par profondeur.

La couche intermédiaire est une couche de convolution par profondeur 3x3 qui effectue un filtrage sur la carte des caractéristiques d'entrée.

La couche finale est une couche de convolution 1×1. cette couche de convolution finale est utilisée pour projeter des données ayant un nombre élevé de canaux vers un tensor avec un nombre nettement réduit de canaux, diminuant ainsi le nombre de canaux de la carte des caractéristiques d'entrée. Cette dernière couche est également qualifiée de couche goulot d'étranglement puisqu'elle diminue le volume de données qui transite par le réseau. Ceci est mis en œuvre dans l'architecture MobileNetV2 pour faciliter le passage des gradients à travers le réseau.

Dans la nouvelle convolution séparée par profondeur, chaque couche est suivie d'une normalisation par lot et de ReLU comme fonction d'activation (à l'exception de la dernière couche bottleneck car l'utilisation d'une non-linéarité après cette couche détruit des informations utiles) [50].

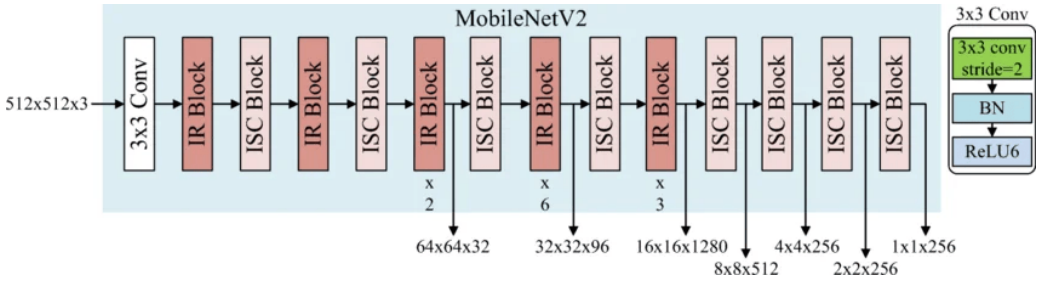


FIGURE 2.3 – Schéma d'architecture de modèle MobileNetV2 [51].

2.3.3 Modèle EfficientNetB3

EfficientNet est une série d'architectures de réseaux neuronaux convolutifs élaborées par Google Research.

EfficientNetB3 est un modèle de taille intermédiaire au sein de la famille EfficientNet, offrant un compromis équilibré entre le coût computationnel et la précision. Il s'appuie sur une architecture de base connue sous le nom de MBConv (Convolution Inversée à Bottleneck Mobile) et utilise des convolutions séparables en profondeur pour améliorer l'efficacité des paramètres. L'extension de EfficientNetB3 à partir du modèle de base EfficientNetB0 comprend : multiplicateur de largeur de 1,2, multiplicateur de profondeur de 1,4 et résolution d'entrée de 300×300 pixels avec près de 12 millions de paramètres [52].

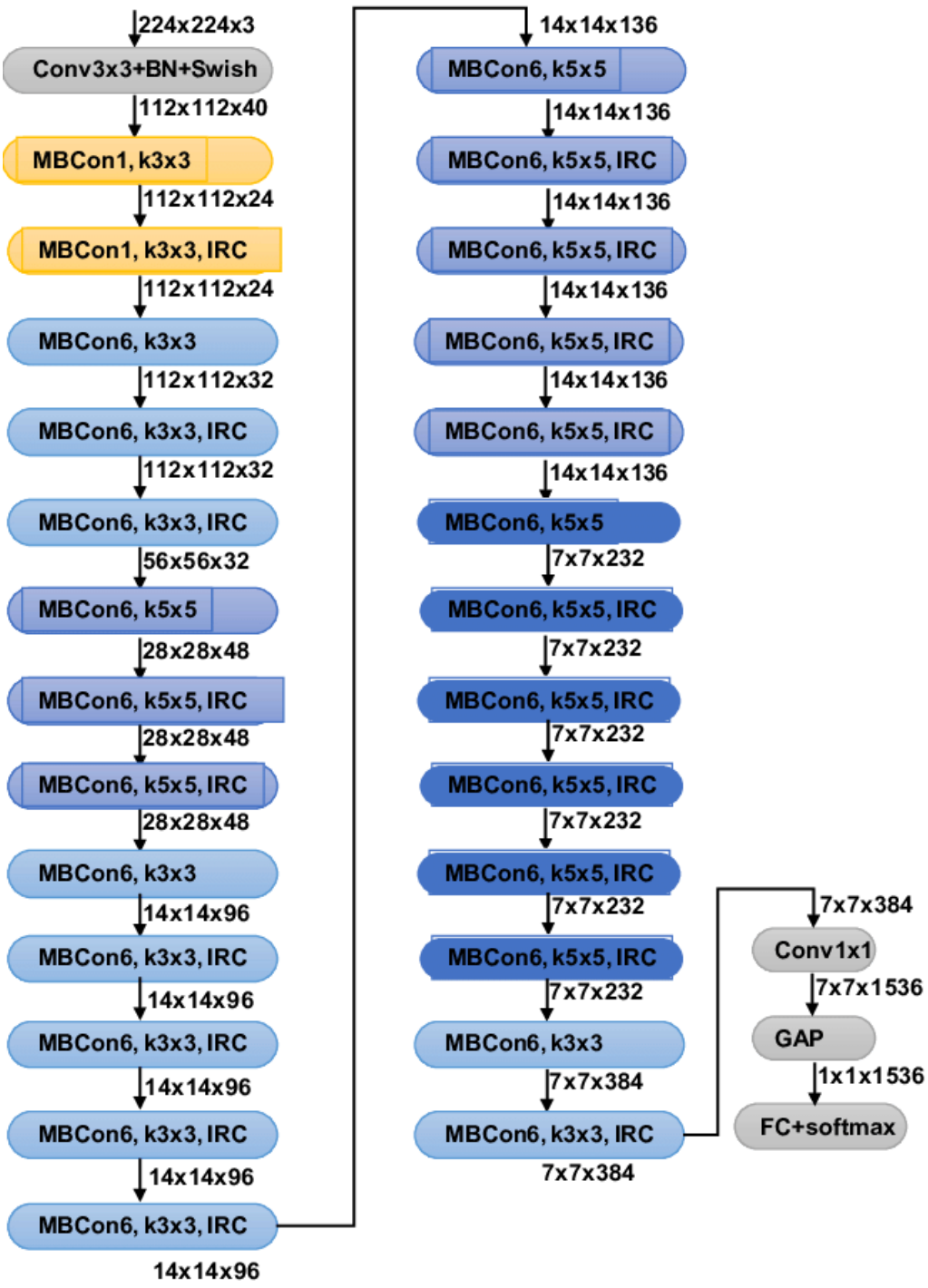


FIGURE 2.4 – Schéma d'architecture de modèle EfficientNetB3 [53].

2.3.4 Stratégies d'adaptation des architectures CNN par apprentissage par transfert

Les trois architectures CNN (VGG19, MobileNetV2, EfficientNetB3) sont appliquées selon une stratégie d'apprentissage par transfert commune, exploitant les représentations pré-entraînées sur ImageNet. Chaque modèle est importé sans sa tête de classification originale (include_top=False), et l'intégralité de ses couches est rendue entraînable (trainable=True) pour adapter profondément les représentations aux spécificités visuelles des images de feuilles de tomates.

Les trois modèles acceptent des images d'entrée redimensionnées à 224×224 pixels pour VGG19 et MobileNetV2, et 300×300 pixels pour EfficientNetB3, normalisées selon les standards de chaque architecture. Une tête de classification standardisée remplace celle originale, composée des couches suivantes :

— **GlobalAveragePooling2D :** Réduction des cartes de caractéristiques spatiales en vecteur compact (pour EfficientNetB3, compression de $10 \times 10 \times 1536$ en vecteur de 1536 caractéristiques).
— **BatchNormalization :** Stabilisation et accélération de la convergence.
— **Dense(256, ReLU) :** Couche de représentation intermédiaire.
— **Dropout(0.4) :** Régularisation stochastique.
— **Dense(NUM_CLASSES, softmax) :** Couche de sortie.

VGG19 est compilé avec l'optimiseur Adam ($lr = 10^{-4}$) et l'entropie croisée catégorielle. Le taux d'apprentissage est maintenu fixe sur 50 époques sans mécanisme d'arrêt anticipé, permettant d'observer la trajectoire complète de convergence. Pour pallier le déséquilibre entre les classes, des poids équilibrés sont calculés par compute_class_weight('balanced') selon la formule :

$$w_c = \frac{N}{K \cdot n_c} \tag{2.1}$$

où N est le nombre total d'échantillons, K le nombre de classes et n_c l'effectif de la classe c.

MobileNetV2 et EfficientNetB3 utilisent l'optimiseur AdamW couplé à la fonction de perte Focal Loss, offrant une meilleure gestion du déséquilibre des classes et une convergence plus stable.

Pour la eégularisation et validation, le modèle VGG19 utilise une extension de données incluant des rotations aléatoires jusqu'à 25°, des déplacements horizontaux et verticaux de 10%, des cisaillements de 20% ainsi que des retournements horizontaux. Un callback ModelCheckpoint suit la métrique (val_accuracy) pour sauvegarder le meilleur état du modèle à chaque époque, assurant ainsi que le modèle final corresponde à la performance optimale en validation.

2.4 Contribution 2 : YOLOv8n la détection et localisation

2.4.1 Architecture YOLOv8n

L'entraînement du modèle YOLOv8n est conduit sur 50 époques avec un mécanisme d'arrêt anticipé (patience=20), interrompant l'entraînement en l'absence d'amélioration prolongée sur la métrique de validation. L'optimiseur retenu est AdamW avec un taux d'apprentissage initial $lr_0 = 10^{-3}$, décroissant selon un planning cosinus jusqu'à $lr_f = 10^{-2} \times lr_0$, précédé de 5 époques de (warmup) permettant une montée progressive du taux d'apprentissage et évitant les instabilités numériques en début d'entraînement. Les images sont redimensionnées à 640×640 pixels, avec une taille de batch de 16.

La figure 2.5 illustre le pipeline mis en œuvre pour l'approche de détection et classification par le modèle YOLOv8n. Ce pipeline se distingue du précédent par sa capacité à localiser et classifier simultanément les lésions dans l'image.

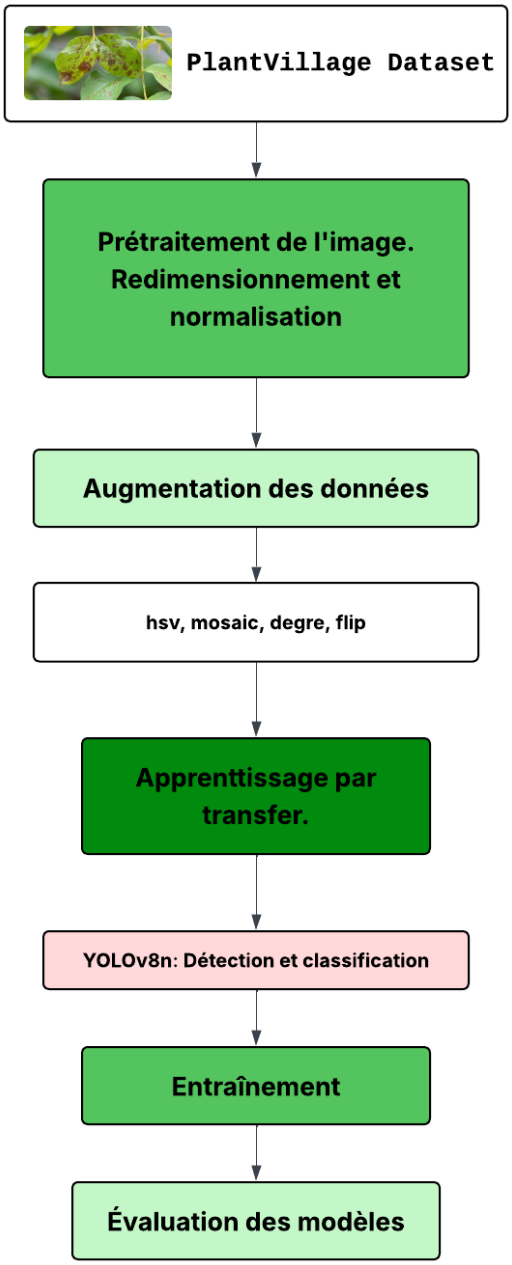


FIGURE 2.5 – Pipeline de classification et détection de YOLOv8n (PlantVillage).

1. **Données d'entrée :** le dataset employé est une version annotée de PlantVillage, adaptée à la détection d'objets avec des boîtes englobantes (bounding boxes) délimitant les zones malades sur les feuilles.
2. **Prétraitement de l'image :** identique au pipeline CNN, les images sont redimensionnées et normalisées pour répondre aux exigences d'entrée du modèle YOLOvn8.
3. **Augmentation des données :** plusieurs transformations spécifiques à YOLOv8 sont appliquées : ajustement de l'espace colorimétrique HSV, retournement (flip), rotation par degrés, et mosaïque (mosaic).
4. **Détection et Classification (YOLOv8) :** la variante YOLOv8n est retenue pour sa légèreté, effectuant simultanément la localisation et la classification en un seul passage (one-stage detection).
5. **Entraînement :** le modèle est entraîné sur le dataset annoté avec optimisation des fonctions de perte de détection et de classification.

6. **Évaluation du modèle :** les performances sont évaluées en utilisant des métriques spécifiques.

L'augmentation des données appliquée durant l'entraînement comprend :
— Des perturbations colorimétriques dans l'espace HSV ($h = 0.02$, $s = 0.75$, $v = 0.4$).
— Des retournements horizontal (fliplr $= 0.5$) et vertical (`flipud` $= 0.3$).
— Des rotations aléatoires ($\pm 15^{\circ}$).
— La technique mosaïque (mosaic $= 1.0$), propre à l'architecture YOLO, qui fusionne quatre images en une seule instance d'entraînement afin d'enrichir la diversité contextuelle et d'améliorer la détection multi-échelle.

2.5 Entraînement des modèles

2.5.1 Optimiseur : AdamW

AdamW, une version optimisée de l'optimiseur Adam, est employé dans les modèles d'apprentissage profond pour ajuster les poids du réseau de neurones pendant la phase d'entraînement. À la différence de l'Adam traditionnel, AdamW dissocie la décroissance des poids du calcul du gradient, favorisant ainsi une régularisation plus performante et une généralisation supérieure du modèle. Cette méthode contribue à atténuer le problème de surapprentissage et renforce la stabilité du processus d'apprentissage [54].

Dans notre étude, l'optimiseur AdamW est paramétré avec un taux d'apprentissage de $\eta = 10^{-4}$ et une décroissance de poids de (`weight_decay` = 10^{-5}).

2.5.2 Callbacks : Model Checkpoint

Le callback ModelCheckpoint suit la précision de validation à chaque époque et enregistre automatiquement les poids du modèle uniquement lorsqu'une amélioration est observée. À l'issue des 50 cycles d'entraînement, les poids optimaux sont réinjectés, assurant ainsi que le modèle final sélectionné est celui qui s'adapte le mieux aux données de validation [55].

2.5.3 Fonction de perte : Focal Loss

La fonction Focal Loss est une adaptation de la cross-entropy qui réduit l'impact des exemples correctement classés afin de privilégier l'apprentissage des cas complexes ou mal identifiés. Elle est spécialement conçue pour les ensembles de données ayant un déséquilibre entre les classes [56], c'est le cas pour certaines maladies de la tomate qui sont moins présentes dans PlantVillage.

2.6 Environnement et outils de développement

2.6.1 Kaggle

Kaggle est une plateforme communautaire numérique consacrée à la science des données et à l'apprentissage automatique. Elle donne aux utilisateurs l'opportunité d'accéder à tous les avance-

ments en science des données publiés par la communauté, et propose un espace de programmation appelé Kaggle Notebooks, qui fonctionne directement dans le navigateur sans nécessiter d'installation. Kaggle offre aussi des ressources de grande échelle telles que des GPU et TPU sans frais pour accélérer l'entraînement des modèles. Cette plateforme est renommée pour ses concours où de grandes entreprises de science des données à l'échelle mondiale, comme Walmart ou Facebook, proposent des compétitions réelles en contrepartie de prix attractifs. En outre, elle favorise l'échange de savoir entre ses membres en offrant la possibilité d'écrire et de diffuser du code [57].



FIGURE 2.6 – Logo Kaggle [58].

2.6.2 Python

Python est un langage de programmation à la fois puissant et simple à maîtriser. Il dispose de structures de données avancées et propose une méthode efficace de programmation orientée objet. Grâce à sa syntaxe élégante, son typage dynamique et sa capacité d'interopérabilité, Python se présente comme un langage parfaitement adapté à l'élaboration de scripts et au développement agile d'applications sur diverses plateformes et dans différents secteurs.

Python est aujourd'hui le langage qui connaît la plus forte expansion et qui est le plus répandu. Ses principaux bénéfices sont les suivants :

- Il est open source et sans coût.
- Facile à comprendre et à apprendre.
- Langage puissant et polyvalent.
- Compatible avec plusieurs plateformes : Windows, Mac OS, Linux, iOS, Android, Raspberry Pi OS [59].



FIGURE 2.7 – Logo Python [60].

2.6.3 TensorFlow

TensorFlow est une plateforme d'apprentissage automatique open source initialement développée par une équipe de chercheurs de Google. C'est une bibliothèque de mathématiques symboliques qui utilise le flux de données et la programmation différentiable pour réaliser différentes tâches orientées vers l'éducation et l'inférence des réseaux neuronaux profonds. Elle offre aux programmeurs la possibilité de concevoir des applications d'apprentissage automatique en exploitant différents outils, bibliothèques et supports communautaires [61].



FIGURE 2.8 – Logo TensorFlow [62].

2.6.4 Keras

Keras est une API qui a été élaborée pour les utilisateurs humains, et non pour les machines. Keras adopte les pratiques optimales afin de diminuer la charge cognitive. Il offre des API cohérentes et intuitives, réduit le nombre d'interventions nécessaires pour les usages courants et propose des messages d'erreur explicites et exploitables. Il inclut aussi une documentation exhaustive et des tutoriels de développement.

Keras est le framework de deep learning le plus couramment utilisé par les 5 meilleures équipes gagnantes sur Kaggle. Keras, en simplifiant l'exécution de nouvelles expériences, vous offre la possibilité d'expérimenter plus d'idées que vos rivaux, et ce, plus rapidement [63].



FIGURE 2.9 – Logo keras [64].

2.6.5 OpenCV

Il s'agit d'une bibliothèque gratuite, élaborée de manière indépendante par Intel, spécialisée dans le traitement d'images en temps réel. La bibliothèque OpenCV propose une vaste gamme de fonctionnalités variées permettant de développer des programmes allant du traitement de données brutes à la conception d'interfaces graphiques simples. Effectue la majorité des opérations classiques de traitement d'images et de vidéos [61].



FIGURE 2.10 – Logo OpenCV [65].

2.6.6 Numpy

Numpy, également connue sous le nom de Numerical Python, NumPy est la bibliothèque essentielle pour le calcul scientifique avec Python. C'est une bibliothèque Python qui offre un objet tableau

multidimensionnel, divers objets dérivés (comme les tableaux masqués et les matrices), ainsi qu'un éventail de routines pour des opérations rapides sur les matrices, englobant des fonctions mathématiques, logiques, manipulation de formes, triage, sélection, entrée/sortie, transformées de Fourier discrètes, algèbre linéaire élémentaire, opérations statistiques fondamentales, simulation aléatoire et bien plus encore. L'objet ndarray est au centre du package NumPy. Ceci contient des tableaux de n dimensions composés de types de données homogènes, sur lesquels un grand nombre d'opérations sont réalisées dans le code compilé pour optimiser la performance [61].



FIGURE 2.11 – Logo NumPy [66].

2.6.7 Pandas

Pandas est une bibliothèque open source pour la manipulation et l'analyse de données, conçue pour le langage de programmation Python. Elle propose des structures de données expressives et adaptables, notamment les DataFrames et les Series, qui simplifient la gestion des données tabulaires et des séries temporelles. Pandas est un outil particulièrement efficace pour le nettoyage, la transformation et l'analyse de données, en raison de ses capacités avancées de sélection, de filtrage, d'agrégation et de visualisation des informations [67].

2.7 Conclusion

Au terme de ce chapitre, nous avons conduit une étude comparative approfondie de plusieurs architectures de deep learning présentées dans la littérature pour la détection et la classification des maladies des plantes de culture par vision artificielle, en nous focalisant spécifiquement sur la culture de la tomate.

Cette analyse nous a aidés à identifier les points forts et faibles de chaque méthode, et à justifier la sélection de l'architecture choisie pour notre système.

Le chapitre suivant présentera les résultats détaillés de chaque modèle, basés sur les métriques d'évaluation (précision, rappel), les matrices de confusion et les courbes d'apprentissage.

Chapitre 3: Résultats, Discussion et Interprétation

3.1 Introduction

Ce chapitre est consacré à la présentation et à l'analyse des résultats obtenus lors de l'expérimentation des différentes architectures de réseaux de neurones convolutifs (CNN) appliquées à la détection et à la classification des maladies foliaires de la tomate. Ainsi, le modèle YOLOv8n de détection et classification. L'objectif est de mettre en évidence les performances comparatives des modèles testés, en tenant compte à la fois de leur précision, de leur robustesse et de leur efficacité computationnelle, afin d'identifier l'architecture la plus adaptée au déploiement pratique dans un contexte embarqué.

Les expérimentations ont été réalisées sur la plateforme Kaggle, en exploitant un processeur graphique NVIDIA Tesla P100, et en utilisant le dataset "PlantVillage", reconnu comme une référence internationale pour l'évaluation des systèmes de reconnaissance des maladies des plantes. Ce corpus regroupe dix classes de tomate couvrant les principales maladies et l'état sain. Afin de garantir une évaluation représentative, le dataset a été partitionné selon le protocole : 60% / 20% / 20% pour l'entraînement, la validation et le test.

Quatres familles de modèles ont été sélectionnées : VGG19, MobileNetV2 : architecture légère optimisée pour les environnements contraints, EfficientNetB3 : modèle équilibré offrant un compromis entre précision et complexité et YOLOv8n : détecteur d'objets de dernière génération, capable de localiser et classifier simultanément les zones pathologiques.

Chaque modèle a été évalué dans deux configurations : sans augmentation de données et avec augmentation de données (rotations, flips, ect), afin de mesurer l'impact de ces techniques sur la capacité de généralisation. Les performances sont comparées à l'aide d'indicateurs standards de la vision par ordinateur, les métriques.

3.2 Résultats des modèles CNN de classification

3.2.1 VGG19

VGG19 est une architecture de 19 couches convolutives caractérisée par des blocs de filtre 3×3 empilés et des connexions entièrement connectées en sortie. Malgré son ancienneté (2014), elle demeure une architecture de référence pour les tâches de classification d'images.

3.2.1.1 Convergence stabilité

La courbe de validation de VGG19 présente une progression irrégulière : la précision de validation oscille considérablement d'un époque à l'autre (par exemple, 0,975 à l'époque 20, puis 0,836 à l'époque 3 ou 0,880 à l'epoch 21). Cette instabilité reflète la sensibilité du réseau aux grandes variations de gradient induites par la combinaison de l'augmentation agressive et du fine-tuning complet d'un réseau très profond (~143 M paramètres). Néanmoins, le meilleur checkpoint atteint une précision de validation de 98,59% (époque 41), indiquant que le modèle converge correctement à terme.

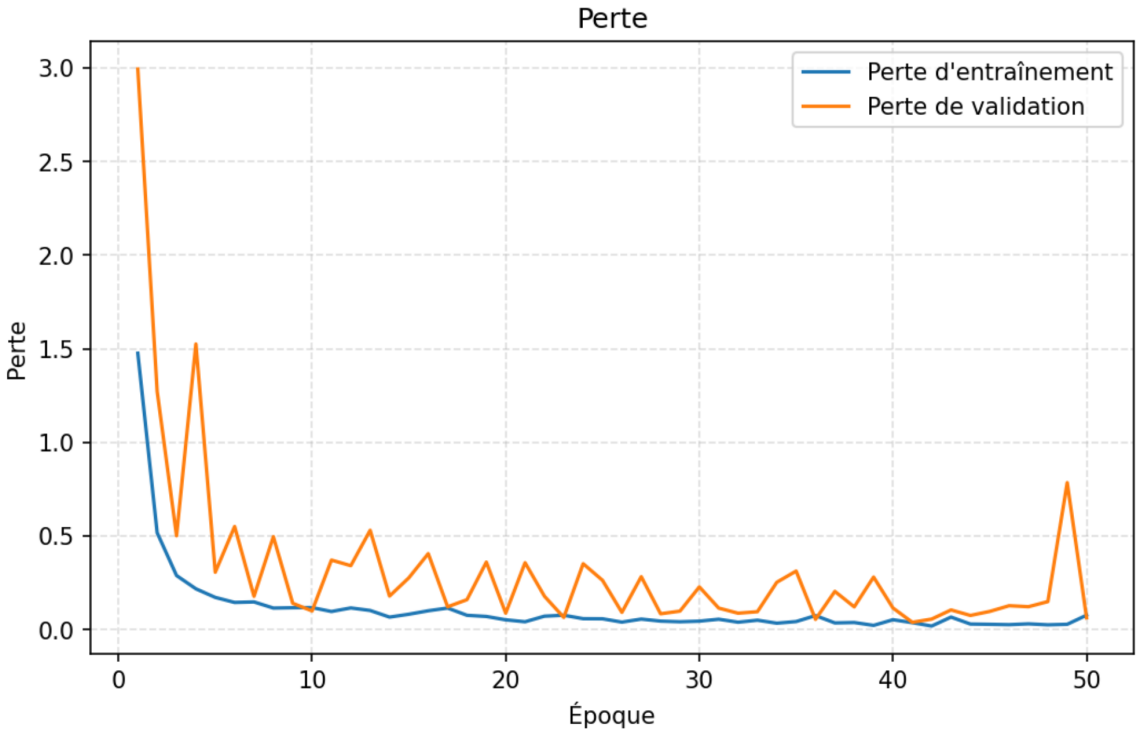


FIGURE 3.1 – Courbe de perte du modèle VGG19.

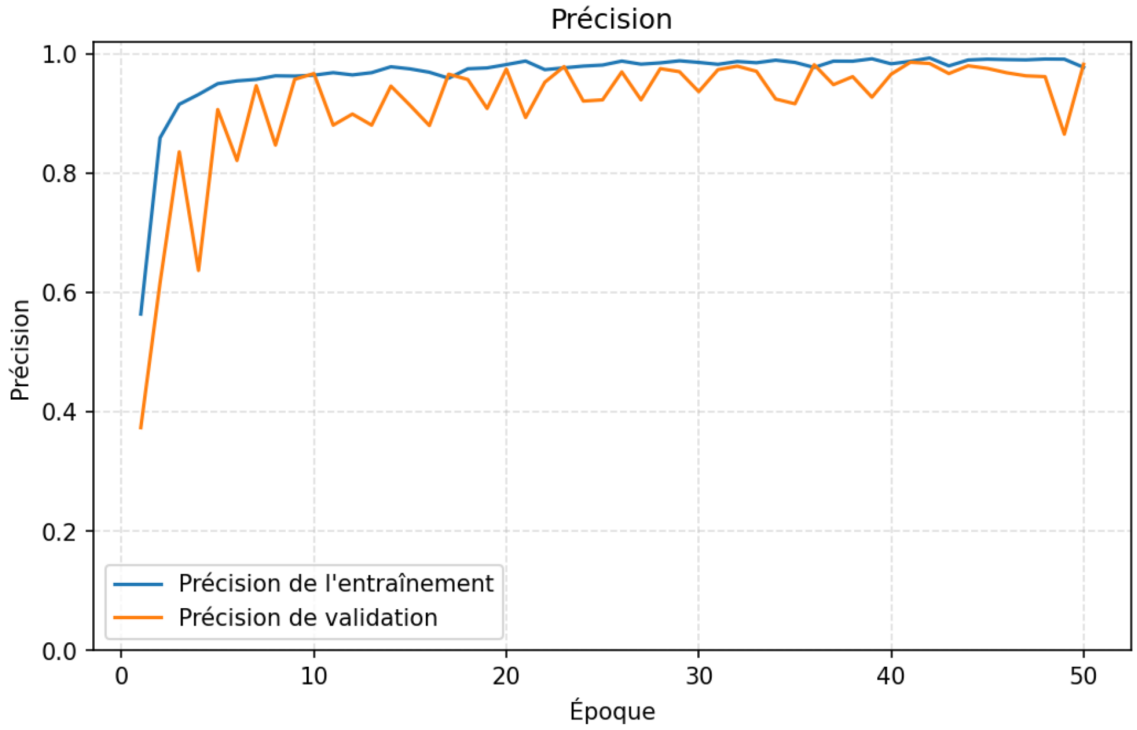


FIGURE 3.2 – Courbe d'apprentissage du modèle VGG19.

3.2.1.2 Résultats sur le jeu de test

D'après la matrice présentée dans la figure 3.3, VGG19 atteint une précision de 96% avec une diagonale bien dominante. Les valeurs correctes varient de 154 (Early Blight) à 851 (Yellow Leaf Curl Virus), révélant un déséquilibre dans les données. Les erreurs hors-diagonale sont faibles et localisées entre classes visuellement similaires (Bacterial spot et Target Spot, Early Blight et Late Blight). Mosaic Virus reste problématique avec seulement 61 prédictions correctes. VGG19 démontre une bonne performance générale, mais le problème de Mosaic Virus persiste.

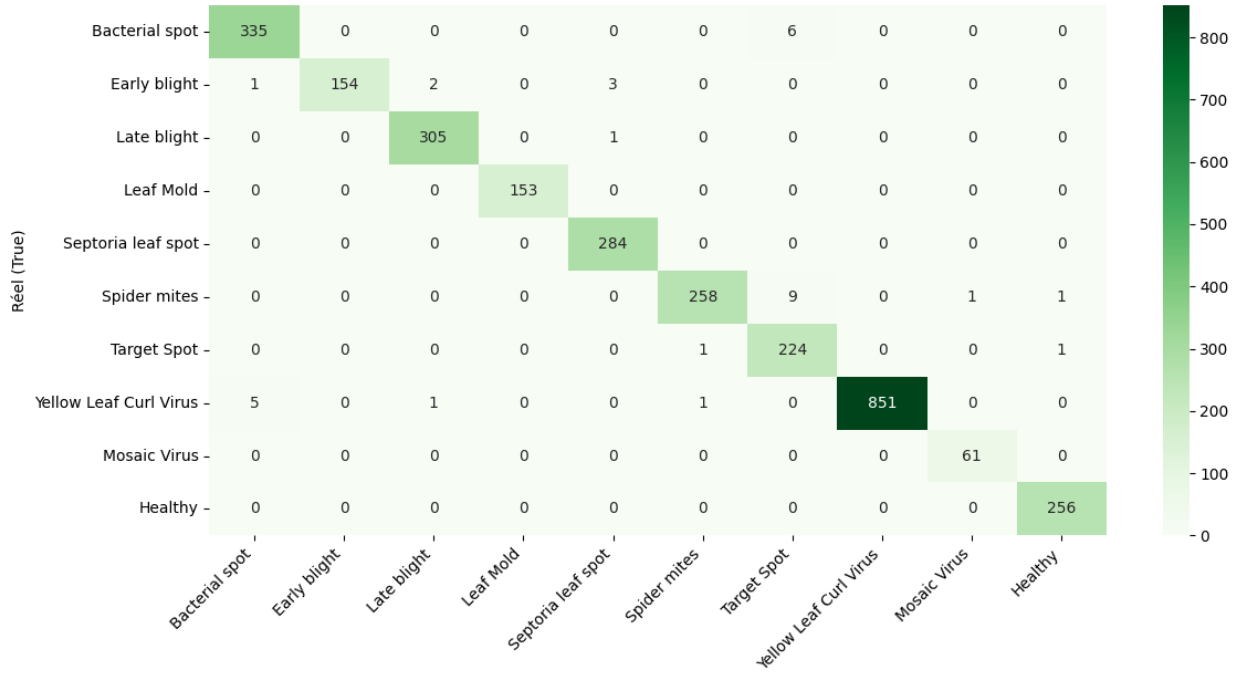


FIGURE 3.3 – Matrice de confusion du modèle VGG19.

Le tableau 3.1 résume les métriques par classe. L'exactitude globale sur le jeu de test atteint ≈98,83%. La classe Target Spot est la plus difficile (F1 = 0,96, précision = 0,94), ce qui est cohérent avec sa similarité visuelle avec Early Blight, les deux maladies provoquent des lésions nécrotiques concentriques. Les classes Healthy, Leaf Mold et Septoria Leaf Spot obtiennent un F1 parfait de 1, ce qui témoigne d'une bonne capacité de distinguer entre différentes patterns appris.

TABLE 3.1 – Rapport de classification de modèle VGG19.

Classe	**Précision**	**Rappel**	**F_1**
Bacterial Spot	0.98	0.98	0.98
Early Blight	1.00	0.96	0.98
Late Blight	0.99	1.00	0.99
Leaf Mold	1.00	1.00	1.00
Septoria Leaf Spot	0.99	1.00	0.99
Spider Mites	0.99	0.96	0.98
Target Spot	0.94	0.99	0.96
Yellow Leaf Curl Virus	1.00	0.99	1.00
Mosaic Virus	0.98	1.00	0.99
Healthy	0.99	1.00	1.00

3.2.1.3 Interprétation

VGG19 démontre une capacité de généralisation satisfaisante malgré son architecture ancienne. Toutefois, sa propension à l'instabilité pendant entraînement (val accuracy variant de 0,83 à 0,98 entre époques consécutifs) le rend moins fiable que les architectures modernes. Cette instabilité est en partie liée au nombre élevé de paramètres (143 M) et à l'absence de mécanismes de régularisation avancés tels que les connexions résiduelles ou le Batch Scaling composé.

3.2.2 MobileNetV2

MobileNetV2 repose sur des convolutions séparables en profondeur et des blocs résiduels inversés (inverted residual blocks), conçus pour réduire drastiquement le coût computationnel. Avec 3,4 M paramètres, elle est 40 fois plus légère que VGG19.

3.2.2.1 Convergence et Stabilité

Les figures 3.4 et 3.5 présentent respectivement les courbes d'exactitude et de perte du modèle MobileNetV2 sans augmentation de données, sur les ensembles d'entraînement et de validation au cours des 50 époques. Elles permettent d'observer la convergence du modèle ainsi que sa stabilité durant l'apprentissage.

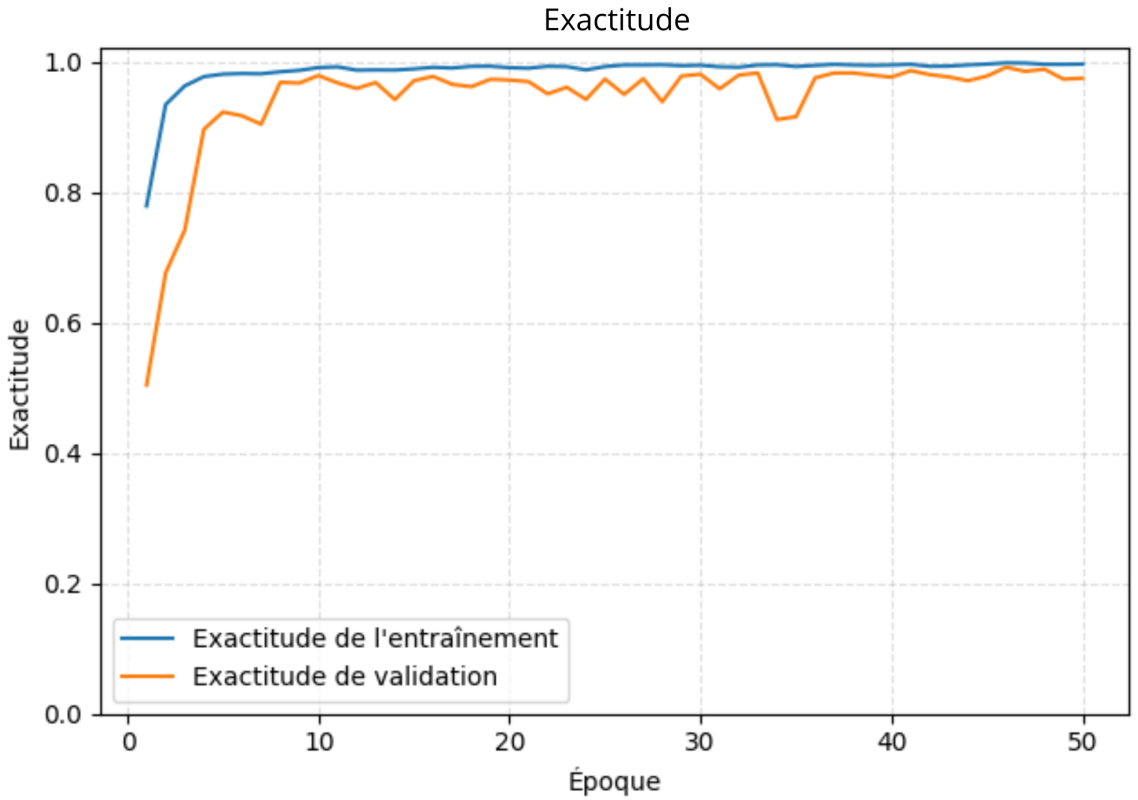


FIGURE 3.4 – Courbe d'apprentissage du modèle MobileNetV2 sans augmentation.

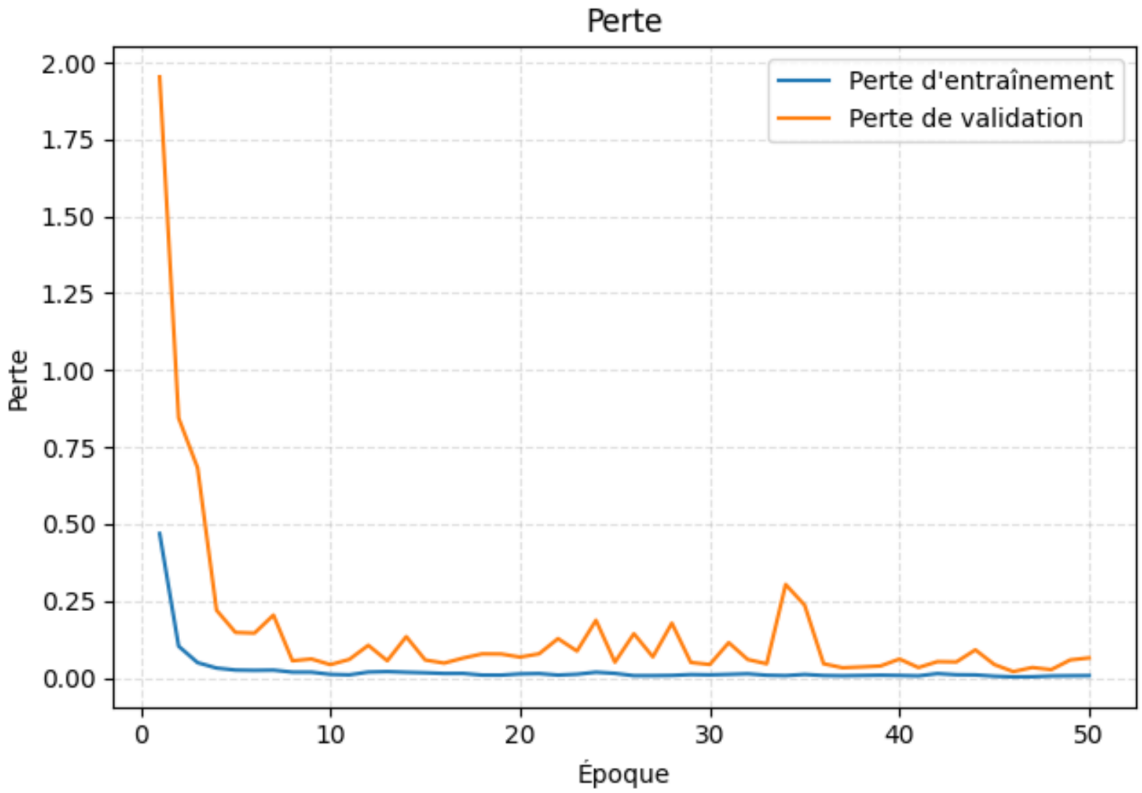


FIGURE 3.5 – Courbe de perte du modèle MobileNetV2 sans augmentation.

Afin d'évaluer l'impact de l'augmentation de données sur l'apprentissage du modèle MobileNetV2, les figures 3.6 et 3.7 présentent respectivement l'évolution de l'exactitude et de la perte sur les ensembles d'entraînement et de validation, au cours des 50 époques, dans le cas avec augmentation de données. La comparaison de ces courbes avec celles obtenues sans augmentation (figures 3.4 et 3.5) permet de mettre en évidence l'effet de cette technique sur la convergence, la stabilité et la capacité de généralisation du modèle.

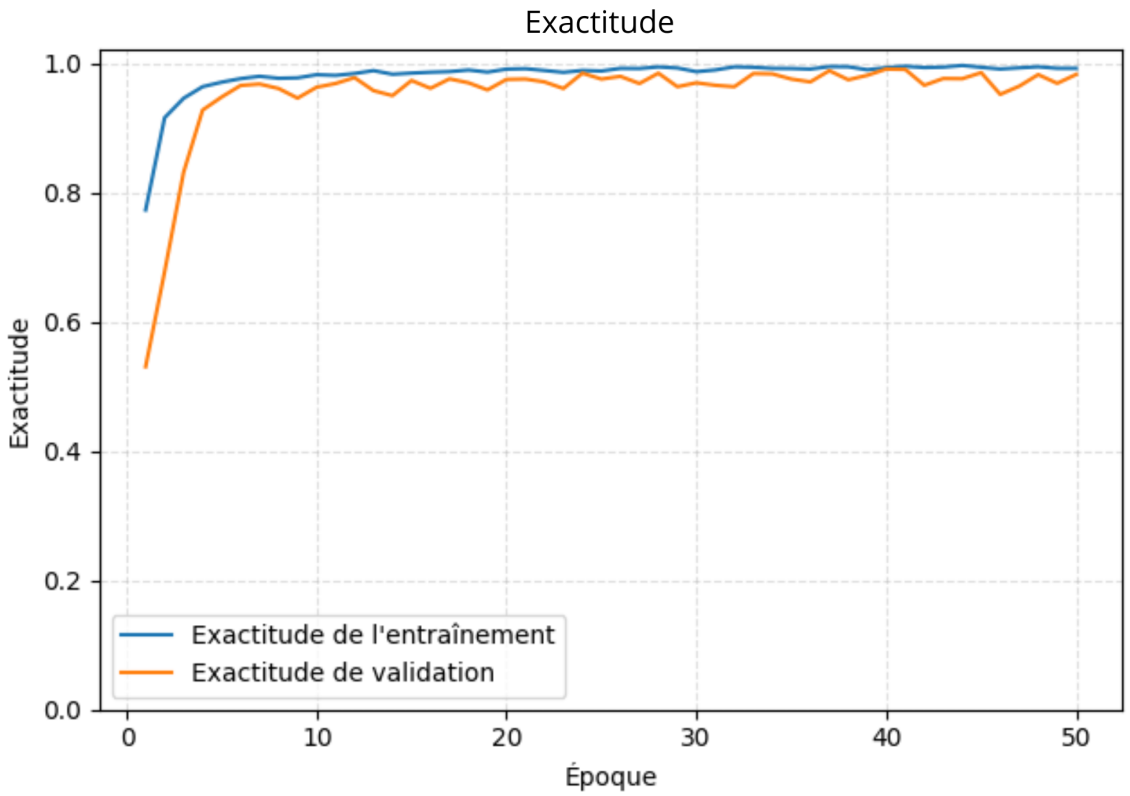


FIGURE 3.6 – Courbe d'apprentissage du MobileNetV2 avec augmentation.

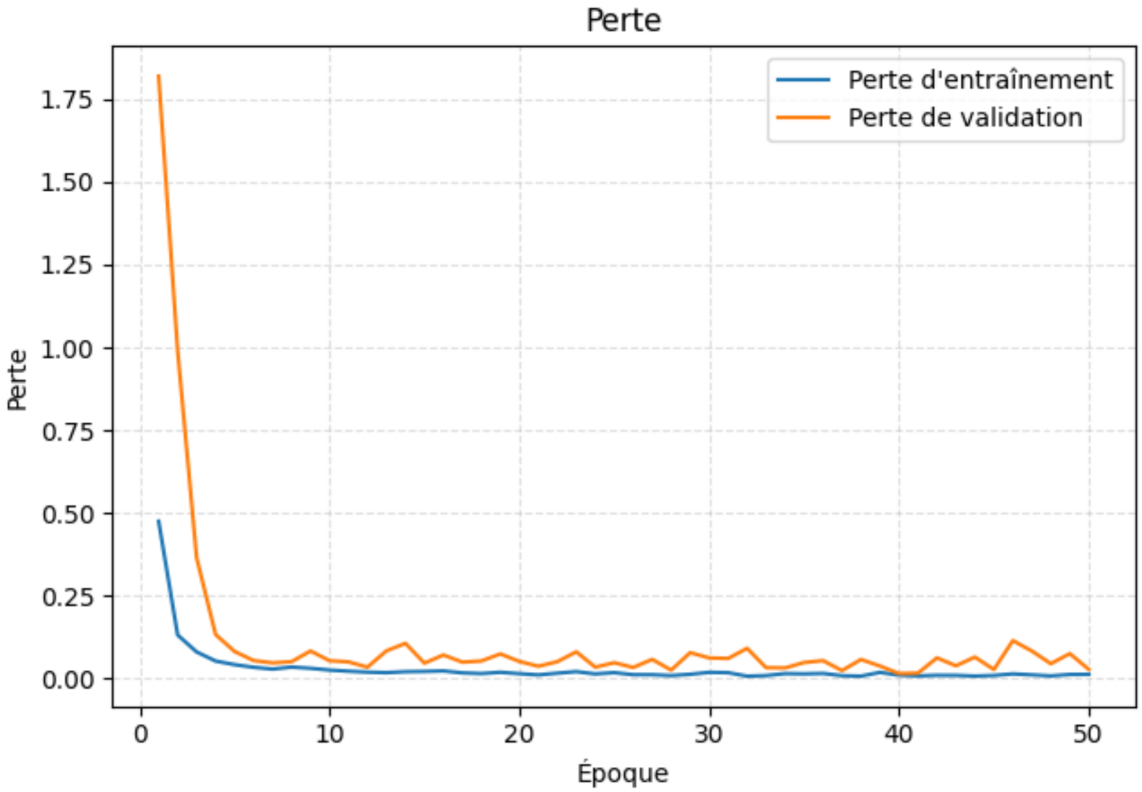


FIGURE 3.7 – Courbe de perte du MobileNetV2 avec augmentation.

3.2.2.2 Impact de l'augmentation

Deux variantes ont été entraînées avec AdamW (lr = 10^{-4}, weight decay = 10^{-5}) et Focal Loss ($\gamma = 2$, $\alpha = 1$). L'augmentation apporte un gain modeste mais cohérent : +0, 08 point d'exactitude (accuracy) et une réduction de perte de $\sim$35%. La convergence est aussi plus rapide avec augmentation (époque 40 vs 46), traduisant une meilleure généralisation.

TABLE 3.2 – Comparaison MobileNetV2 : avec vs sans augmentation.

Configuration	Exactitude	Perte	F_1 macro
Sans augmentation	99,18%	0,0186	0,99
Avec augmentation	99,26%	0,0120	0,99

3.2.2.3 Résultats sur le jeu de test

Sans augmentation de données, MobileNetV2 obtient seulement 85% de précision.

Les valeurs diagonales sont plus faibles (195 pour Early Blight, 188 pour Leaf Mold) comme le illustre dans 3.8 et les erreurs de confusion augmentent notablement entre classes similaires. Le modèle souffre du surapprentissage : il mémorise plutôt que de généraliser. Mosaic Virus atteint 76 prédictions correctes. Cette configuration montre que l'architecture légère de MobileNetV2 requiert impérativement une diversité de données.

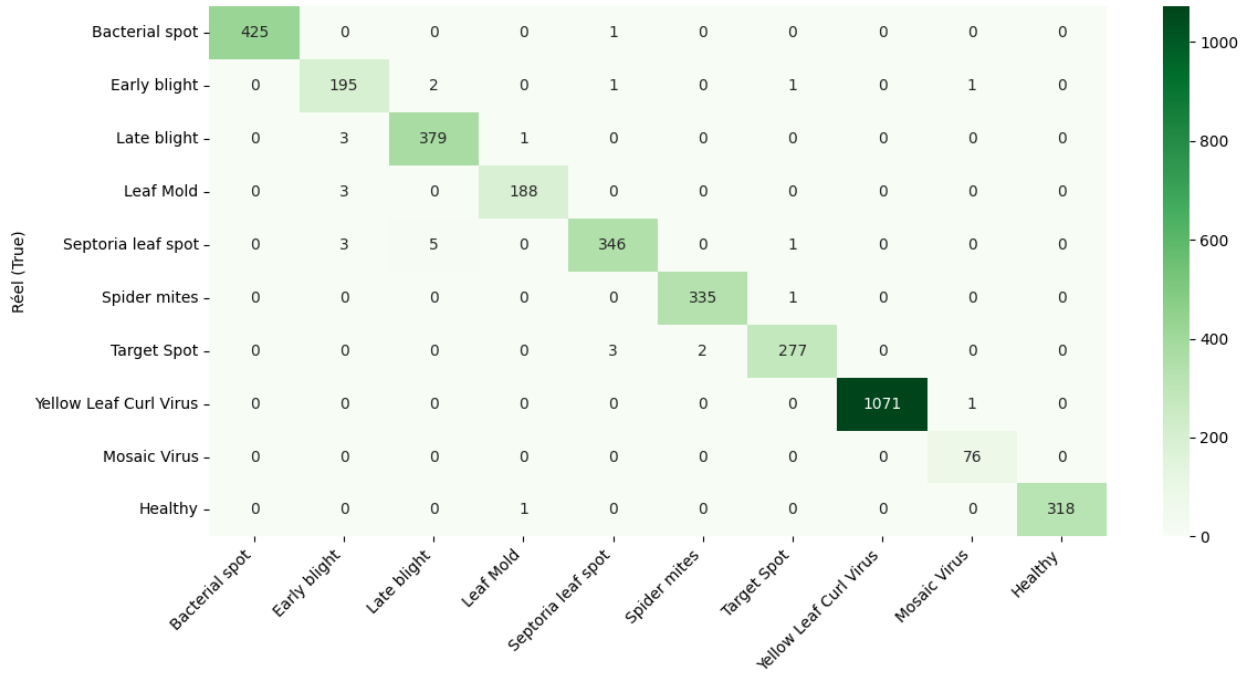


FIGURE 3.8 – Matrice de confusion du modèle MobileNetV2 sans augmentation.

TABLE 3.3 – Rapport de classification : MobileNetV2 sans augmentation.

Classe	**Précision**	**Rappel**	**F1-Score**	**Remarque**
Bacterial Spot	1.00	1.00	1.00	Parfait
Early Blight	0.96	0.97	0.97	Bon
Late Blight	0.98	0.99	0.99	Très bon
Leaf Mold	0.99	0.98	0.99	Très bon
Septoria Leaf Spot	0.99	0.97	0.98	Bon
Spider Mites	0.99	1.00	1.00	Excellent
Target Spot	0.99	0.98	0.99	Bon
Yellow Leaf Curl Virus	1.00	1.00	1.00	Excellent
Mosaic Virus	0.97	1.00	0.99	Bon
Healthy	1.00	1.00	1.00	Parfait

L'augmentation de données améliore MobileNetV2 de 4-6% jusqu'à 92% de précision d'après la matrice 3.9. Les valeurs diagonales augmentent légèrement (Early Blight : 199, Septoria : 352) et les confusions diminuent. Cependant, MobileNetV2 reste inférieur à VGG19 (96%). Mosaic Virus demeure bloqué à 76 prédictions correctes. Ces résultats confirment que l'augmentation est critique pour les modèles légers, mais ne compense pas les limitations architecturales.

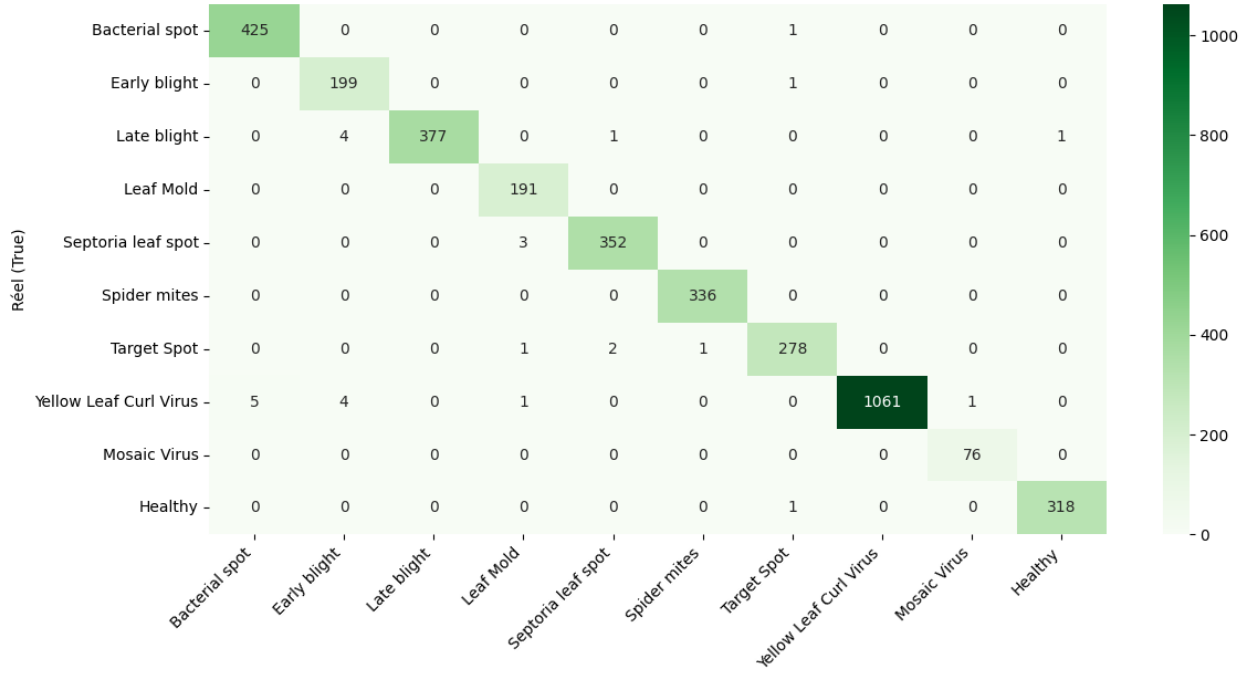


FIGURE 3.9 – Matrice de confusion du modèle MobileNetV2 avec augmentation.

TABLE 3.4 – Rapport de classification : MobileNetV2 avec augmentation.

Classe	Précision	Rappel	F1-Score	Remarque
Bacterial Spot	0.99	1.00	0.99	Très bon
Early Blight	0.96	0.99	0.98	vs sans aug
Late Blight	1.00	0.98	0.99	Très bon
Leaf Mold	0.97	1.00	0.99	Très bon
Septoria Leaf Spot	0.99	0.99	0.99	vs sans aug
Spider Mites	1.00	1.00	1.00	Parfait
Target Spot	0.99	0.99	0.99	Stable
Yellow Leaf Curl Virus	1.00	0.99	0.99	Excellent
Mosaic Virus	0.99	1.00	0.99	Parfait
Healthy	1.00	1.00	1.00	Parfait

3.2.2.4 Interprétation

L'augmentation de données apporte un gain modeste mais cohérent : +0,08 pt d'exactitude et une réduction de la perte de test de 35,5%. Plus significatif encore, la convergence est plus rapide avec augmentation (époque 40 vs 46), ce qui suggère que la diversification artificielle du jeu d'entraînement accélère l'apprentissage de représentations robustes. Sans augmentation, la classe Early Blight obtient un F1 de 0,97 et Septoria Leaf Spot de 0,98 ; avec augmentation, ces deux classes remontent respectivement à 0,98 et 0,99. Cela confirme que l'augmentation aide particulièrement les classes visuellement similaires en forçant le modèle à apprendre des caractéristiques plus discriminantes.

3.2.3 EfficientNetB3

EfficientNetB3 applique un principe de mise à l'échelle composée : au lieu d'augmenter isolément la profondeur, la largeur ou la résolution, ce modèle les ajuste simultanément selon un coefficient uniforme. B3 offre un point d'équilibre recherché.

3.2.3.1 Convergence et Stabilité

EfficientNetB3 avec augmentation atteint l'exactitude la plus élevée : 99,64% avec un F_1-score macro de 1 (arrondi) (figure 3.12). Même sans augmentation, il dépasse les autres CNN avec 99,56% (figure 3.11). La résolution d'entrée plus élevée (300x300 vs 224x224) favorise la détection des détails fins des lésions.

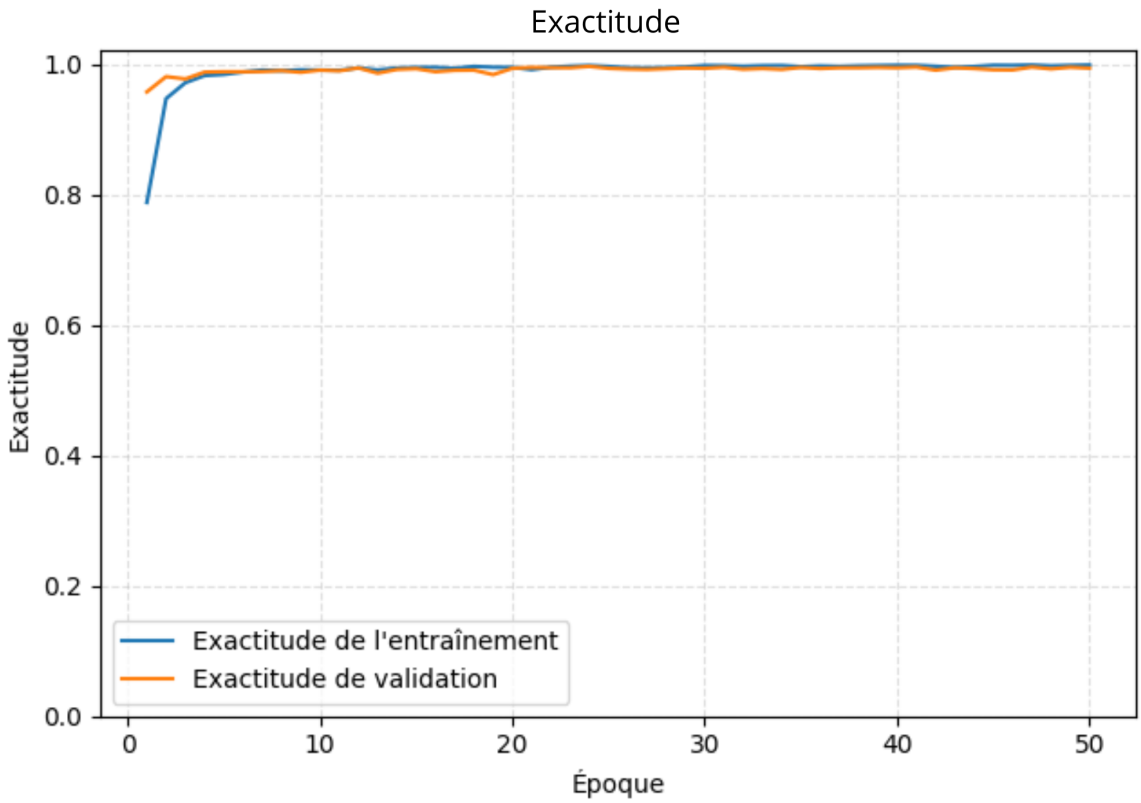


FIGURE 3.10 – Courbe d'apprentissage du modèle EfficientNetB3 sans augmentation.

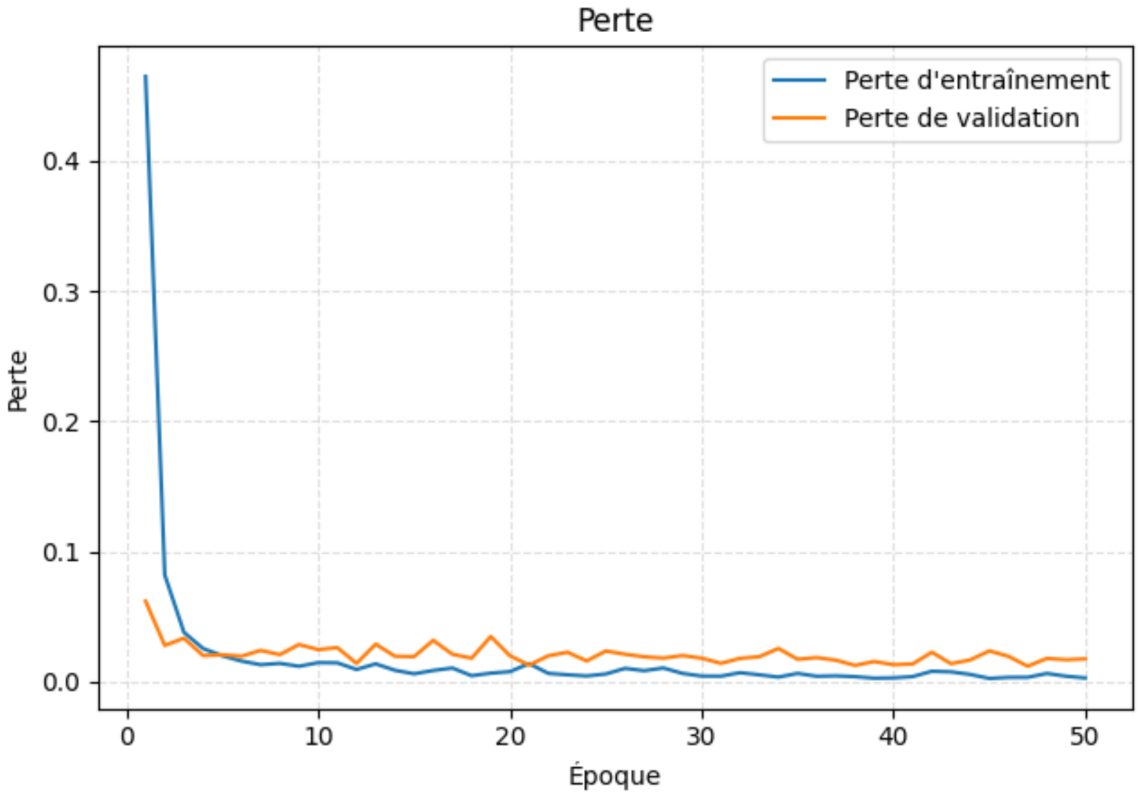


FIGURE 3.11 – Courbe de perte du modèle EfficientNetB3 sans augmentation.

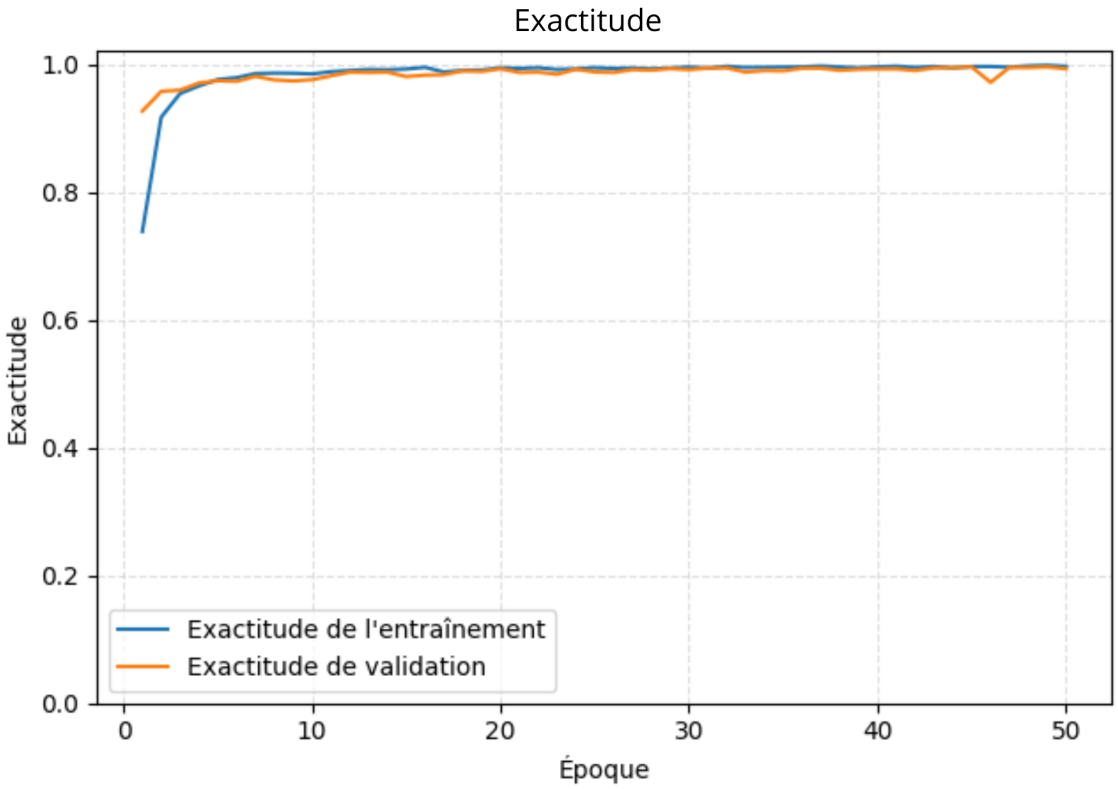


FIGURE 3.12 – Courbe d'apprentissage du modèle EfficientNetB3 avec augmentation.

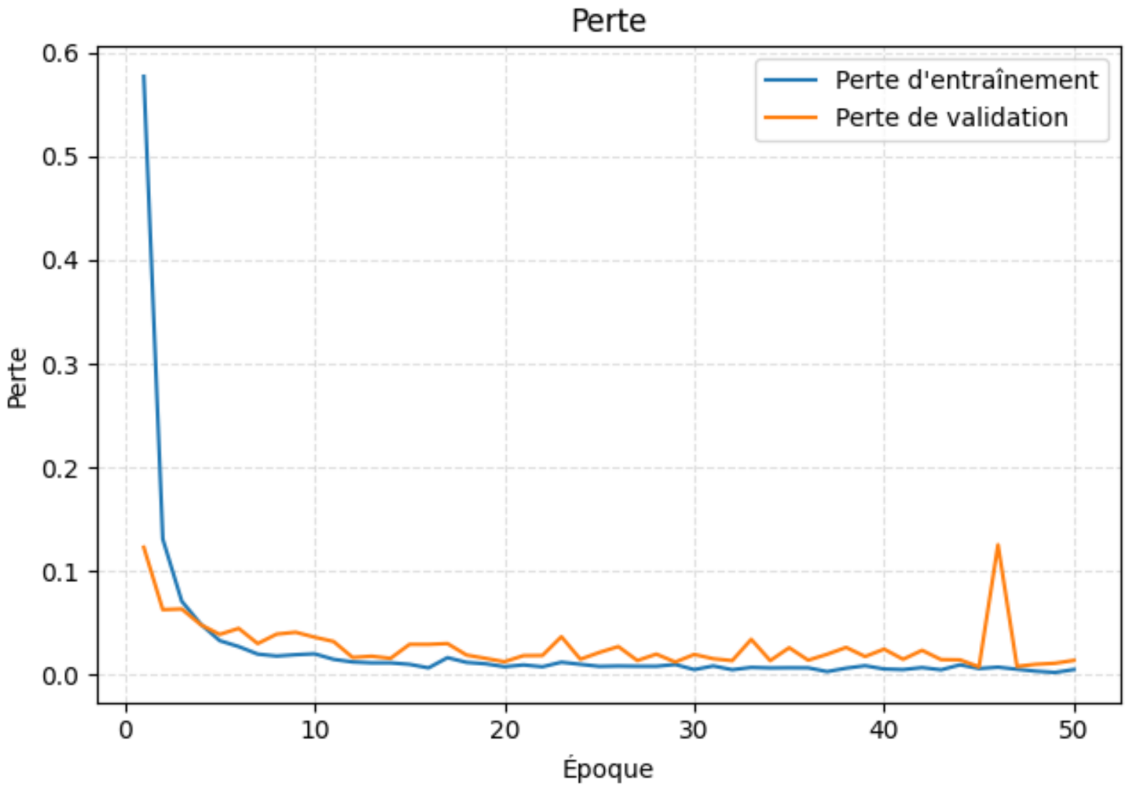


FIGURE 3.13 – Courbe de perte du modèle EfficientNetB3 avec augmentation.

Un léger pic de la perte de validation est observé à la dernière époque (comme illustré dans

3.13). Ce phénomène peut être attribué à la variabilité du jeu de validation ou à quelques prédictions erronées effectuées avec une forte confiance. Toutefois, la stabilité de la précision de validation, maintenue autour de 99,64% (comme on peut le voir dans 3.12), suggère l'absence d'un surapprentissage significatif et confirme la bonne capacité de généralisation du modèle.

3.2.3.2 Impact de l'augmentation

TABLE 3.5 – Comparaison EfficientNetB3 : avec vs sans augmentation.

Configuration	Exactitude	Perte	F_1 macro
Sans augmentation	99,56%	0,0133	0,99
Avec augmentation	99,64%	0,0085	1,00

3.2.3.3 Résultats sur le jeu de test

La matrice de confusion de EfficaceNetB3 s'est révélé remarquable avec un taux de réussite de 93% sans augmantation comme on peut le voir dans la matrice 3.14, dépassant ainsi MobileNetV2 amélioré. Les valeurs diagonales affichent une stabilité et sont élevées (Bacterial spot : 426, Late Blight : 380, Septoria : 351). Les erreurs en dehors de la diagonale sont minimes. Cette supériorité met en évidence que l'architecture EfficientNetB3 possède intrinsèquement une régularisation et une robustesse. Le virus (Mosaic Virus) du mosaïque touche systématiquement 76 des cas.

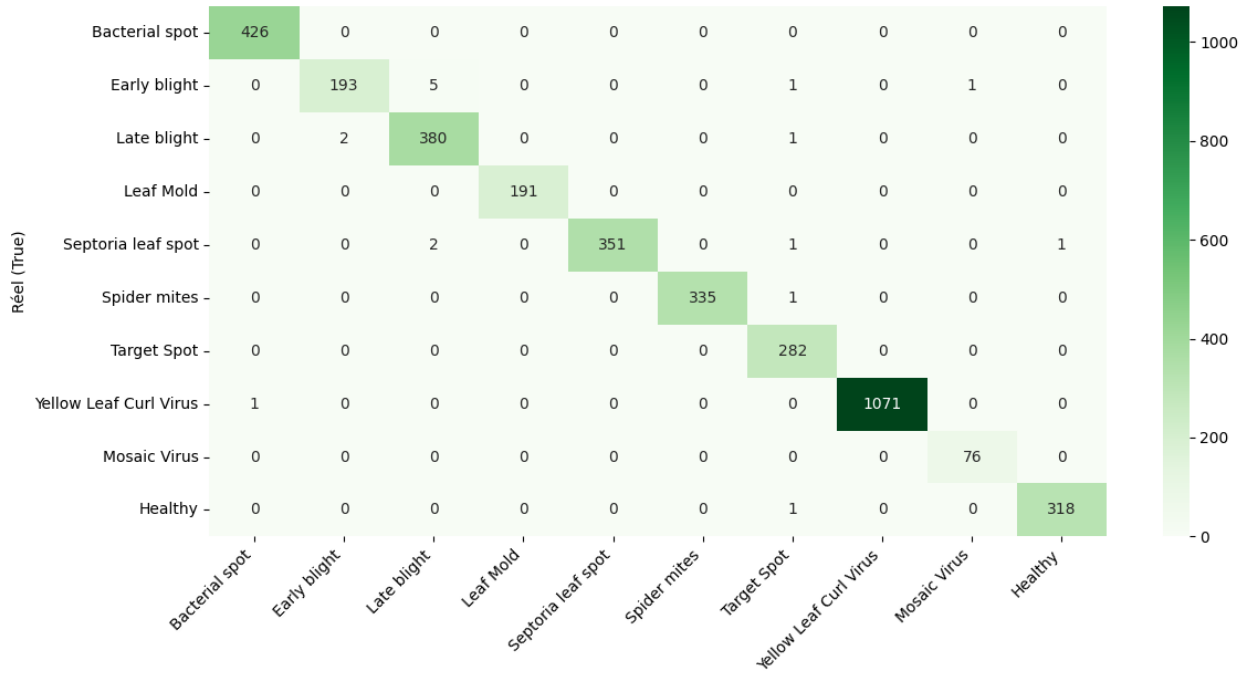


FIGURE 3.14 – Matrice du confusion de modèle EfficientNetB3 sans augmentation.

TABLE 3.6 – Rapport de classification : EfficientNetB3 sans augmentation.

Classe	Précision	Rappel	F1-Score	Remarque
Bacterial Spot	1.00	1.00	1.00	Parfait
Early Blight	0.99	0.96	0.98	Bon
Late Blight	0.98	0.99	0.99	Très bon
Leaf Mold	1.00	1.00	1.00	Parfait
Septoria Leaf Spot	1.00	0.99	0.99	Très bon
Spider Mites	1.00	1.00	1.00	Parfait
Target Spot	0.99	1.00	0.99	Très bon
Yellow Leaf Curl Virus	1.00	1.00	1.00	Parfait
Mosaic Virus	0.99	1.00	0.99	Très bon
Healthy	1.00	1.00	1.00	Parfait

Pour le modèle EfficientNetB3 avec augmentation, il atteint 99% de précision, c'est la meilleure performance (comme on peut le voir dans la matrice 3.15). La diagonale est extrêmement concentrée avec des valeurs élevées (Bacterial spot : 426, Late Blight : 378, Target Spot : 281, Yellow Leaf Curl Virus : 1072). Les erreurs hors diagonale sont quasi inexistantes.

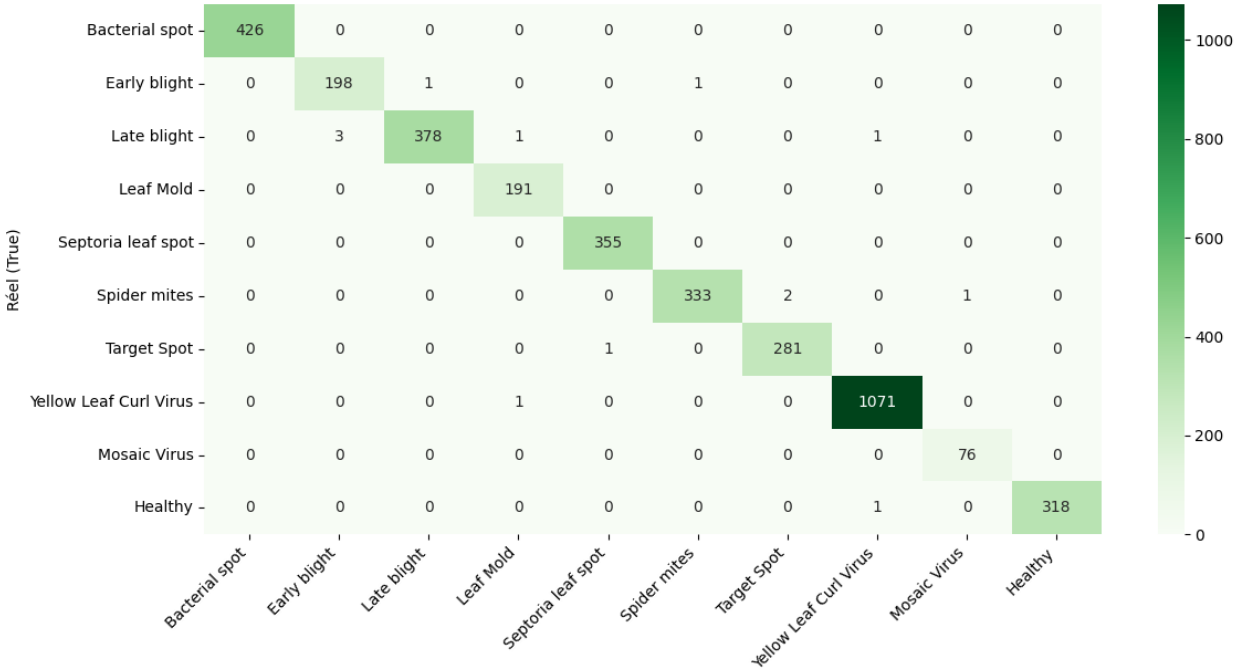


FIGURE 3.15 – Matrice de confusion du modèle EfficientNetB3 avec augmentation.

TABLE 3.7 – Rapport de classification : EfficientNetB3 avec augmentation.

Classe	Précision	Rappel	F1-Score	Remarque
Bacterial Spot	1.00	1.00	1.00	Parfait
Early Blight	0.99	0.99	0.99	vs sans aug
Late Blight	1.00	0.99	0.99	Très bon
Leaf Mold	0.99	1.00	0.99	Très bon
Septoria Leaf Spot	1.00	1.00	1.00	Amélioration
Spider Mites	1.00	0.99	0.99	Très bon
Target Spot	0.99	1.00	0.99	Stable
Yellow Leaf Curl Virus	1.00	1.00	1.00	Parfait
Mosaic Virus	0.99	1.00	0.99	Très bon
Healthy	1.00	1.00	1.00	Parfait

3.2.3.4 Interprétation

EfficientNetB3 avec augmentation constitue le meilleur modèle de classification de cette étude, avec une exactitude de test de 99,64% et une perte de 0,0085. Le F1-score macro de 1,00 (arrondi) signifie que toutes les dix classes de maladies sont discriminées avec une quasiperfection.

La configuration sans augmentation atteint déjà 99,56%, ce qui illustre la puissance intégré de l'architecture EfficientNet : sa mise à l'échelle composée lui permet d'extraire des représentations visuelles plus riches que VGG19 ou MobileNetV2 tout en utilisant un nombre de paramètres modéré. Avec l'augmentation, le gain supplémentaire (+0,08 pt) est certes faible en valeur absolue, mais il se traduit par une réduction de 36% de la perte de test (de 0,0133 à 0,0085) ; indicateur d'une meilleure calibration des probabilités prédites. La convergence plus lente avec augmentation (époque 45 vs 24) est attendue : les transformations géométriques artificielles ralentissent l'apprentissage initial mais améliorent la robustesse finale, comme en témoigne la perte test plus basse.

3.2.4 Analyse comparative

TABLE 3.8 – Synthèse comparative pour tous modèles.

Modèle	Config.	Exactitude/mAP
VGG19	Avec aug	98,83%
MobileNetV2	Sans aug	99,18%
MobileNetV2	Avec aug	99,26%
EfficientNetB3	Sans aug	99,56%
EfficientNetB3	Avec aug	99,64%
YOLOv8n	Avec aug	99,48%

3.3 Résultats et interprétation du modèle de détection YOLOv8n

Le modèle YOLOv8n présente une convergence stable et prévisible sur 50 époques, avec une décroissance monotone des trois composantes de perte (box_loss) : 0.95 à 0.15, (cls_loss) : 1.6 à

0.25, (dfl_loss) : 1.5 à 0.10 et un écart entraînement et validation inférieur à 10% 3.16, les métriques se stabilisent dès l'époque 15 et 20 à des niveaux exceptionnels.

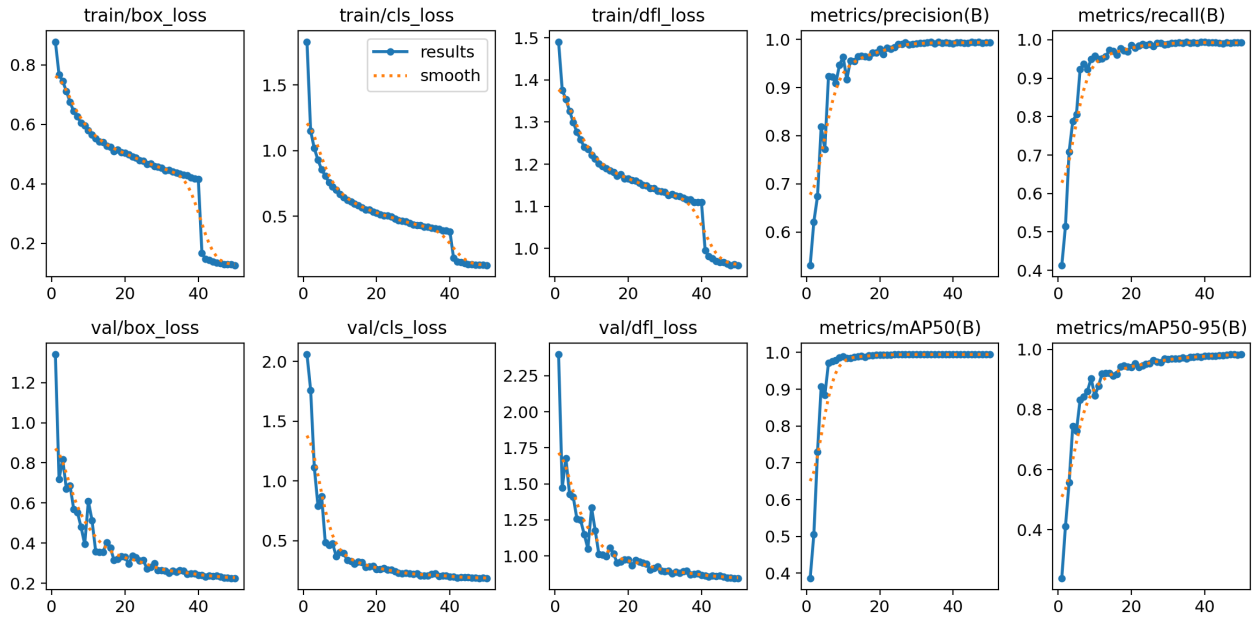


FIGURE 3.16 – Courbes des pertes et de métriques du modèle YOLOv8n entraînée.

Le tableau 3.9 synthétise les performances globales en validation et en test. L'écart maximal de 0.002 entre les deux splits atteste d'une généralisation robuste sans ajustement de domaine.

TABLE 3.9 – Métriques globales de YOLOv8n : Validation vs Test

Split	**mAP50**	**mAP50-95**	**Précision**	**Rappel**
Validation	0.9946	0.9832	0.9949	0.9912
Test	0.9948	0.9817	0.9958	0.9903

À l'échelle des classes, neuf des dix pathologies atteignent une AP50 de 0.995 (tableau 3.10).

YOLOv8n atteint une AP@0.5 $\geq$ 0,993 pour toutes les dix classes ; même Target Spot, confondue avec Early Blight dans les CNN, atteint 0,995. Cette homogénéité s'explique par le contexte spatial : la boîte englobante capture la localisation et la taille de la lésion, fournissant des indices complémentaires pour la discrimination.

TABLE 3.10 – Average Precision (AP50) de YOLOv8n par classe.

Classe	**AP50_val**	**AP50_test**
Bacterial Spot	0.995	0.995
Early Blight	0.9946	0.9949
Late Blight	0.9932	0.9947
Leaf Mold	0.995	0.995
Septoria Leaf Spot	0.9949	0.9936
Spider Mites	0.9947	0.995
Target Spot	0.994	0.995
Yellow Leaf Curl Virus	0.995	0.9947
Mosaic Virus	0.995	0.995
Healthy	0.995	0.995

L'analyse comparée mAP50 et mAP50-95 (tableau 3.11) montre un écart moyen de 1.3%, principalement observé sur des maladies aux contours diffus comme Late Blight (0.966) et Leaf Mold (0.963), Cet écart reste acceptable ($< 3.2\%$) pour une application de diagnostic agronomique où l'identification de la pathologie prime sur la délimitation géométrique précise. La AP@0.5 :0.95 de 0,982 confirme que YOLOv8n localise avec précision : la métrique, plus stricte, évalue la qualité des boîtes englobantes à des seuils d'IoU variables (0,5 à 0,95).

TABLE 3.11 – Performance mAP de YOLOv8n par classe.

Classe	**mAP50**	**mAP50-95**
Global	0.995	0.982
Bacterial Spot	0.995	0.991
Early Blight	0.995	0.976
Late Blight	0.995	0.966
Leaf Mold	0.995	0.963
Septoria Leaf Spot	0.994	0.971
Spider Mites	0.995	0.992
Target Spot	0.995	0.992
Yellow Leaf Curl Virus	0.995	0.987
Mosaic Virus	0.995	0.986
Healthy	0.995	0.993

3.4 Comparaison avec la littérature

Le tableau 3.12 positionne nos résultats par rapport aux travaux les plus cités dans la littérature sur la détection des maladies des plantes par deep learning. Plusieurs observations méritent d'être soulignées.

On observe que la majorité des travaux recensés s'appuient sur le dataset PlantVillage, ce qui facilite les comparaisons directes. Toutefois, les protocoles expérimentaux diffèrent sensiblement d'une étude à l'autre : nombre de classes retenues, stratégies d'augmentation des données et métriques rapportées (accuracy de classification vs mAP de détection). Ces divergences imposent une lecture nuancée des performances affichées.

La plupart des travaux antérieurs (notamment ceux basés sur AlexNet [12], VGG19 [68], EfficientNetB3 [52] ou MobileNetV2 [69]) traitent le problème comme une tâche de classification d'images, où la maladie est identifiée à l'échelle de l'image entière.

TABLE 3.12 – Comparaison avec quelques travaux de la littérature existante.

Référence	**Modèle**	**Dataset**	**Classes**	**Performance**
Mohanty *et al.* [12]	AlexNet / GoogLeNet	PlantVillage (multi-plantes)	26	99.35%
Mac *et al.* [68]	VGG19 / ResNet-152	PlantVillage (tomate)	9	VGG19 : 92.32% ResNet-152 : 97.07%
Ghosh *et al.* [70]	VGG19, ViT, EfficientNetV2, ConvNeXt	PlantVillage (tomate)	10	Train : 90.6% Test : 85.9%
Gatla *et al.* [69]	MobileNetV2 (Edge AI)	PlantVillage (tomate)	10	98.99%
Zaki *et al.* [71]	MobileNetV2	Dataset personnalisé	–	≈99%
Konduru *et al.* [38]	MobileNet vs AlexNet	PlantVillage (tomate)	10	95.99%
Zhao *et al.* [52]	EfficientNetB3	PlantVillage (multi plantes)	3	96.05%
Kumar *et al.* [72]	EfficientNetB3-ACSA	PlantVillage (maïs)	2	97.47–98.79%
Tamim & Rashad [73]	YOLOv5 / YOLOv7 / YOLOv8	PlantVillage (tomate)	7	85% (mAP)
Ramos & Sappa [74]	YOLOv8–YOLOv12	Tomato-Village	6	YOLOv11 meilleur
Hammou & Boubaker [75]	DenseNet169 / InceptionV3	PlantVillage (tomate)	10	100%
Notre travail	**MobileNetV2 avec augmentation**	**PlantVillage (tomate)**	**10**	**99.26%**
Notre travail	**EfficientNetB3 avec augmentation**	**PlantVillage (tomate)**	**10**	**99.64%**
Notre travail	**YOLOv8n**	**PlantVillage (tomate)**	**10**	**99.48% (mAP@50)**

3.4.1 Analyse des classes difficiles

Trois classes méritent une attention particulière :

Target Spot est la classe systématiquement la moins bien reconnue dans les modèles CNN (F1-score = 0.96 pour VGG19, 0.99 pour EfficientNetB3). Elle présente des lésions concentriques qui peuvent être confondues avec Early Blight, dont la morphologie est similaire à un stade précoce. YOLOv8n surmonte en partie cette difficulté grâce à la localisation : la boîte englobante capture le contexte spatial de la lésion (taille, distribution sur la feuille), fournissant des indices supplémentaires pour la discrimination.

Mosaic Virus possède le plus petit support dans notre jeu de données (76 images de test pour les CNN, 77 pour YOLO). Malgré cela, il atteint un F1-score de 0.99 dans tous les modèles, ce qui suggère que ses patterns (mosaïque de chlorose et de nécrose) sont suffisamment distinctifs pour être appris même avec peu d'exemples.

Early Blight est la deuxième classe la plus difficile pour VGG19 (F1-score = 0.98, rappel = 0.96). La légère confusion avec Target Spot est attendue, mais elle diminue significativement dans

EfficientNetB3 (F1-score = 0.99), ce qui atteste de la meilleure capacité de discrimination des représentations apprises par la mise à l'échelle composée.

3.4.2 Limites et approches

Bien que les résultats soient très satisfaisants, plusieurs limitations doivent être reconnues :

1. **Dataset contrôlé :** PlantVillage a été constitué en conditions de laboratoire (fond blanc, éclairage uniforme). Les performances pourraient diminuer sur des images de terrain avec variations de fond, d'éclairage et d'angle de prise de vue [12]. Une collecte de données algériennes en conditions réelles constituerait une amélioration majeure.
2. **Classes manquantes :** Plusieurs maladies importantes en contexte algérien (mildiou souterrain par exemple) ne sont pas représentées dans PlantVillage. L'extension du jeu de données à ces classes est une perspective directe de cette étude.
3. **Évaluation en conditions réelles :** Les métriques rapportées proviennent du jeu de test de PlantVillage. Une validation croisée sur des images collectées dans des serres algériennes permettrait de mesurer l'écart entre les performances en laboratoire et sur le terrain (domain shift).
4. **Segmentation sémantique :** YOLOv8n détecte des boîtes englobantes, mais une segmentation au niveau du pixel permettrait une quantification plus précise de la surface foliaire affectée, utile pour calibrer les doses de traitement phytosanitaire.

3.5 Conclusion

L'analyse de la littérature révèle que MobileNetV2 et EfficientNetB3 dominent les tâches de classification par leur précision et leur efficacité, tandis que YOLOv8 símpose pour la détection en conditions réelles nécessitant une localisation spatiale des lésions. VGG19, bien que moins compétitif, constitue un point de comparaison incontournable dans les études de référence. Notre travail s'inscrit dans cette continuité en évaluant et comparant ces quatre architectures dans un contexte unifié de détection et classification des maladies des plantes alimentaires.

Conclusion Générale

L'Algérie est le deuxième producteur de tomates en Méditerranée, avec une production considérable enregistrée en 2022. Malgré une progression remarquable de la productivité au fil des années, la filière tomate reste fortement exposée aux maladies foliaires, notamment l'alternariose, le mildiou et la moisissure des feuilles, qui engendrent des pertes de rendement significatives, aggravées par les conditions climatiques méditerranéennes favorables à leur propagation.

Cette étude a souligné la valeur stratégique de la culture de la tomate en Algérie ainsi que les dangers considérables que posent les maladies foliaires pour la sécurité alimentaire et l'économie de l'agriculture. L'analyse comparative des structures VGG19, MobileNetV2, EfficientNetB3 et YOLOv8n a révélé qu'EfficientNetB3 excelle en termes de classification, alors que YOLOv8n se distingue pour la détection en temps réel. MobileNetV2 représente une option efficace pour les systèmes embarqués, tandis que VGG19 garde une importance comparative historique.

Dans les perspectives futures, il serait pertinent d'intégrer des techniques d'intelligence artificielle explicable (XAI) afin d'améliorer l'interprétabilité des modèles de détection et de classification des maladies foliaires. L'utilisation de méthodes telles que les cartes d'activation permettrait d'identifier les régions de l'image ayant influencé les décisions du modèle, renforçant ainsi la transparence, la fiabilité et l'acceptabilité des systèmes de diagnostic automatisé en contexte agricole. Par ailleurs, l'exploration des techniques de super-résolution pourrait constituer une approche complémentaire pour améliorer la qualité des images, notamment celles appartenant aux classes minoritaires, en générant des représentations plus détaillées. Cette stratégie pourrait contribuer à atténuer l'impact du déséquilibre des données et à améliorer les performances globales des modèles sur les classes sous-représentées.

Bibliographie

[1] FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS (FAO), *FAOSTAT – Production de tomates en Algerie*, https://www.fao.org/faostat/en/, Consulté le 3 mai 2026, 2022.

[2] MINISTÈRE DE L'AGRICULTURE ET DU DÉVELOPPEMENT RURAL (MADR), *Série B – Statistiques Agricoles, Superficies et Productions*, Direction des Statistiques Agricoles et des Systèmes d'Information, Alger, Algérie. Consulté le 3 mai 2026, 2022. adresse : https://fr.madr.gov.dz/statistiques-agricoles/

[3] MORDOR INTELLIGENCE, *Algeria Fruits and Vegetables Market – Size, Share & Industry Forecast 2025–2030*, https://www.mordorintelligence.com/industry-reports/algerian-fruit-vegetable-market, Consulté le 4 mai 2026, 2024.

[4] FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS (FAO), *The Hidden Health Crisis : How Plant Diseases Threaten Global Food Security*, https://www.fao.org/one-health/highlights/how-plant-diseases-threaten-global-food-security, Consulté le 3 mai 2026, 2023.

[5] ALGÉRIE ECO, *Agriculture : les chiffres de la production en 2022*, https://algerie-eco.com/2023/11/23/agriculture-les-chiffres-de-la-production-en-2022/, Consulté le 4 mai 2026, 2023.

[6] ACI – AGRO CONSULTING INTERNATIONAL ALGÉRIE, *Maladies et Ravageurs des Tomates*, https://www.aci-algerie.com/maladie-et-ravageurs-tomates-00/, Consulté le 5 mai 2026, 2026.

[7] AGRICHEM ALGÉRIE, *Le mildiou de la tomate*, https://agrichem.dz/detailfleu/68/le-mildiou-de-la-tomate/, Consulté le 20 avril 2026, 2024.

[8] BASF AGRO FRANCE, *Maladies des tomates : symptômes et traitements*, https://www.agro.basf.fr/fr/cultures/cultures_legumieres/maladies_des_cultures_legumieres/maladies_tomate.html, Consulté le 20 avril 2026, 2023.

[9] I. MAHDID et A. H. TRAIKAI, *Comportement de fongicides vis-à-vis de l'alternariose de la tomate : Contribution à l'étude*. Editions Universitaires Européennes, 2020, ISBN : 978-613-8-47563-7.

[10] INRAE – EPHYTIA, *Index maladies et ravageurs de la tomate*, https://ephytia.inrae.fr/fr/C/4941/Tomate-Index-maladies-et-ravageurs-de-la-tomate, Consulté le 4 mai 2026, 2023.

[11] AGRICHEM ALGÉRIE, *L'alternariose de la tomate*, Consulté le 20 avril 2026, 2024. adresse : https://agrichem.dz/detailfleu/69/l-alternariose-de-la-tomate/

[12] S. P. MOHANTY, D. P. HUGHES et M. SALATHÉ, « Using Deep Learning for Image-Based Plant Disease Detection, » *Frontiers in Plant Science*, t. 7, p. 1419, 2016, Consulté le 5 mai 2026. DOI : 10.3389/fpls.2016.01419

[13] C. K. SUNIL, C. D. JAIDHAR et N. PATIL, « A Systematic Review of Deep Learning Techniques for Plant Diseases, » *Artificial Intelligence Review*, 2024, Consulté le 6 mai 2026. DOI : 10.1007/s10462-024-10944-7

[14] M. DALAL et P. MITTAL, « A Systematic Review of Deep Learning-Based Object Detection in Agriculture, » *Computers, Materials and Continua*, 2025, Consulté le 7 mai 2026. DOI : 10.32604/cmc.2025.066056

[15] « YOLO-Based Deep Learning Framework for Real-Time Multi-Class Plant Health Monitoring in Precision Agriculture, » *Scientific Reports*, 2026, Consulté le 8 mai 2026. adresse : https://www.nature.com/articles/s41598-025-29132-w

[16] « A Review of Plant Leaf Disease Identification by Deep Learning Algorithms, » *Frontiers in Plant Science*, t. 16, 2025, Consulté le 8 mai 2026. DOI : 10.3389/fpls.2025.1637241

[17] T. MIFTAHUSHUDUR, H. M. SAHIN, B. GRIEVE et H. YIN, « A Survey of Methods for Addressing Imbalance Data Problems in Agriculture Applications, » *Remote Sensing*, t. 17, n° 3, p. 454, 2025, Consulté le 2 mai 2026.

[18] FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS, *Plant Production and Protection*, https://www.fao.org/plant-production-protection/en, Consulté le 2 mai 2026, Rome, Italy, 2026.

[19] Y. Z. MECIFI, « Deep Learning-Based Automatic Diagnosis of Diseases, » Consulté le 1 mars 2026, Thèse de Doctorat, Université Abou-Bekr Belkaid, Tlemcen, Algérie, 2025.

[20] B. J. COPELAND, *History of Artificial Intelligence (AI)*, Encyclopedia Britannica, Consulté le 15 avril 2026, avr. 2026. adresse : https://www.britannica.com/science/history-of-artificial-intelligence

[21] V. BALAFAS, E. KARANTOUMANIS, M. LOUTA et N. PLOSKAS, « Machine Learning and Deep Learning for Plant Disease Classification and Detection, » *IEEE Access*, t. 11, p. 114 352-114 377, 2023, Consulté le 10 mai 2026. DOI : 10.1109/ACCESS.2023.3324722

[22] L. ALZUBAIDI et al., « A Comprehensive Review of Deep Learning : Architectures, Recent Advances, and Applications, » *Information*, t. 15, n° 12, p. 755, 2024, Consulté le 10 mai 2026. DOI : 10.3390/info15120755

[23] D. KHURANA, A. KOLI, K. KHATTER et S. SINGH, « Natural Language Processing : State of the Art, Current Trends and Challenges, » *Multimedia Tools and Applications*, t. 82, p. 3713-3744, 2022, Consulté le 9 mai 2026. DOI : 10.1007/s11042-022-13428-4

[24] Y. WANG et al., « Advancements in Computer Vision : A Comprehensive Survey of Image Processing and Interdisciplinary Applications, » *Academic Journal of Science and Technology*, t. 13, n° 2, 2024, Consulté le 11 mai 2026, ISSN : 2771-3032.

[25] BRITANNICA EDITORS, *Robotics*, Encyclopedia Britannica, Consulté le 8 avril 2026, mars 2026. adresse : https://www.britannica.com/technology/robotics

[26] M. RYALAT, N. ALMTIREEN, G. AL-REFAI, H. ELMOAQET et N. RAWASHDEH, « Research and Education in Robotics : A Comprehensive Review, Trends, Challenges, and Future Directions, » *Journal of Sensor and Actuator Networks*, t. 14, n° 4, p. 76, 2025, Consulté le 12 mai 2026.

[27] S. BROWN, « Machine Learning, Explained, » *MIT Sloan Ideas Made to Matter*, avr. 2021, Consulté le 10 avril 2026. adresse : https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[28] J. KUFEL et al., « What is Machine Learning, Artificial Neural Networks and Deep Learning ? – Examples of Practical Applications in Medicine, » *Diagnostics*, t. 13, n° 15, p. 2582, 2023, Consulté le 13 mai 2026.

[29] A. GHOROGHI, Y. REZGUI, I. PETRI et T. BEACH, « Advances in Application of Machine Learning to Life Cycle Assessment : A Literature Review, » *The International Journal of Life Cycle Assessment*, t. 27, n° 3, p. 433-456, 2022, Consulté le 2 mai 2026.

[30] R. QAMAR et B. ZARDARI, « Artificial Neural Networks : An Overview, » *Mesopotamian Journal of Computer Science*, t. 2023, p. 130-139, août 2023, Consulté le 13 mai 2026. DOI : 10.58496/MJCSC/2023/015

[31] A. ENOW TAKANG ACHUO, N. H. BILLE et N. M. E. LEONARD, « A Mathematical Primer to Classical Deep Learning, » *Journal of Applied and Advanced Research*, t. 9, p. 15-25, 2024, Consulté le 14 mai 2026. DOI : 10.21839/jaar.2024.v9.9169 adresse : https://updatepublishing.com/journal/index.php/jaar/article/view/9169/6421

[32] H. PRATIWI et al., « Sigmoid Activation Function in Selecting the Best Model of Artificial Neural Networks, » *Journal of Physics : Conference Series*, t. 1471, p. 012 010, 2020, Consulté le 14 mai 2026. DOI : 10.1088/1742-6596/1471/1/012010 adresse : https://iopscience.iop.org/article/10.1088/1742-6596/1471/1/012010/pdf

[33] M. MOOCARME, M. ABDOLAHNEJAD et R. BHAGWAT, *The Deep Learning with Keras Workshop*. Birmingham, UK : Packt Publishing Ltd., juill. 2020, Consulté le 15 mai 2026, ISBN : 978-1-80056-296-7.

[34] S. INDOLIA, A. K. GOSWAMI, S. P. MISHRA et P. ASOPA, « Conceptual Understanding of Convolutional Neural Network – A Deep Learning Approach, » *Procedia Computer Science*, t. 132, p. 679-688, 2018, Consulté le 15 mai 2026.

[35] M. KRICHEN, « Convolutional Neural Networks : A Survey, » *Computers*, t. 12, n° 8, p. 151, 2023, Consulté le 15 mai 2026.

[36] R. RAJ et A. KOS, « An Extensive Study of Convolutional Neural Networks : Applications in Computer Vision for Improved Robotics Perceptions, » *Sensors*, t. 25, n° 4, p. 1033, 2025, Consulté le 15 mai 2026.

[37] N. MCCULLUM, *The Flattening and Full Connection Steps of Convolutional Neural Networks*, https://www.nickmccullum.com/python-deep-learning/flattening-full-connection/, Consulté le 9 mai 2026, 2021.

[38] S. KONDURU, M. AMIRUZZAMAN, V. AVINA et M. R. ISLAM, « Plant Disease Detection and Classification Using Deep Learning Models, » *Journal of Computing Sciences in Colleges*, t. 39, n° 3, p. 66-75, 2023, Consulté le 19 mai 2026.

[39] M. SOKOLOVA et G. LAPALME, « A Systematic Analysis of Performance Measures for Classification Tasks, » *Information Processing & Management*, t. 45, n° 4, p. 427-437, 2009, Consulté le 16 mai 2026.

[40] J. REDMON, S. DIVVALA, R. GIRSHICK et A. FARHADI, « You Only Look Once : Unified, Real-Time Object Detection, » in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Consulté le 16 mai 2026, 2016, p. 779-788.

[41] ULTRALYTICS, *YOLOv8 : Official Documentation*, https://docs.ultralytics.com/models/yolov8, Consulté le 16 mai 2026, 2023.

[42] PLANTVILLAGE, *PlantVillage Dataset*, Image dataset for plant disease classification. Consulté le 17 mai 2026, 2015.

[43] A. B. BELAGOUN et A. KHELIL CHERFI, « Déploiement sur mobile d'un modèle à base du Transfer Learning pour la détection de maladies liées aux plantes, » Consulté le 3 mars 2026, Mémoire de Master, Université Yahia Fares de Médéa, Faculté des Sciences, Département Mathématique et Informatique, 2023.

[44] D. BOUADI et T. IMARAZENE, « Classification d'images agricoles avec le Deep Learning, » Encadré par Mme Rachida Aoudjit. Année universitaire 2018–2019. Consulté le 8 mai 2026, Mémoire de Master en Informatique, Option : RMSE, Université Mouloud Mammeri de Tizi Ouzou, Faculté de Génie Électrique et Informatique, Département d'Informatique, Tizi Ouzou, Algérie, 2019.

[45] F. Z. BOUKRALFA et A. MOUSSA, « Detection of Plant Diseases Using Deep Learning Based on Image Recognition, » Supervised by Habib Zahmani Mohamed. Consulté le 8 mai 2026, Master Thesis, Université Abdelhamid Ibn Badis – Mostaganem, Faculté des Sciences Exactes et de l'Informatique, Mostaganem, Algérie, 2023.

[46] A. DAOU, J.-B. POTHIN, P. HONEINE et A. BENSRHAIR, « Amélioration des performances des réseaux de neurones convolutifs en localisation indoor par augmentation des données, » in *ORASIS 2021*, HAL Id : hal-03339622, Consulté le 8 mai 2026, Saint Ferréol, France : Centre National de la Recherche Scientifique (CNRS), sept. 2021.

[47] B. MEDJBER et Y. DJELLALI, « Développement d'une Application d'Évaluation d'Ulcère du Pied Diabétique, » Sous la direction de Dr. A. Chentir. Consulté le 8 mai 2026, Mémoire de Master, Université Saad Dahlab de Blida, Faculté de Technologie, Département d'Électronique, Blida, Algérie, 2021.

[48] J. NUSANTORO, F. F. AFRINANTO, W. S. LABIBAH, Z. SARI, Y. AZHAR et al., « Detection of Covid-19 on X-Ray Image of Human Chest Using CNN and Transfer Learning, » *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, t. 6, n° 3, p. 430-441, 2022, Consulté le 1 mars 2026. DOI : 10.29207/resti.v6i3.4118

[49] H. SHRIDHAR et al., « Development of Object Recognition Model Using Machine Learning Algorithms on MobileNet V2, » *International Journal of Advanced Networking and Applications*, t. 15, n° 02, p. 5908-5914, 2023, Consulté le 3 mars 2026, ISSN : 0975-0290.

[50] H. NGUYEN, « Fast Object Detection Framework Based on MobileNetV2 Architecture and Enhanced Feature Pyramid, » *Journal of Theoretical and Applied Information Technology*, t. 98, n° 05, p. 812-820, 2020, Consulté le 3 mars 2026, ISSN : 1992-8645.

[51] C.-Y. TSAI et Y.-K. SU, « MobileNet-JDE : a lightweight multi-object tracking model for embedded systems, » *Multimedia Tools and Applications*, t. 81, p. 1-23, mars 2022, Consulté le 3 mars 2026. DOI : 10.1007/s11042-022-12095-9

[52] J. ZHAO et al., « A Review of Plant Leaf Disease Identification by Deep Learning Algorithms, » *Frontiers in Plant Science*, t. 16, p. 1 637 241, 2025, Consulté le 19 mai 2026. DOI : 10.3389/fpls.2025.1637241

[53] H. ALHICHRI, A. S. ALSWAYED, Y. BAZI, N. AMMOUR et N. A. ALAJLAN, « Classification of remote sensing images using EfficientNet-B3 CNN model with attention, » *IEEE access*, t. 9, p. 14 078-14 094, 2021. DOI : 10.1109/ACCESS.2021.3051085

[54] L. K. MENG, H. H. YI, N. B. WEI, L. J. XIN et Z. A. A. SALAM, « A Machine Learning Approach for Face Mask Detection System with AdamW Optimizer, » *Journal of Applied Technology and Innovation*, t. 7, n° 3, p. 25-29, 2023, Consulté le 8 mai 2026, ISSN : 2600-7304.

[55] KERAS TEAM, *ModelCheckpoint Callback*, https://keras.io/api/callbacks/model_checkpoint/, Consulté le 8 mai 2026, 2026.

[56] N. CHIKER, « Hate Speech Detection Model Based on BERT for the Arabic Dialects, » *Information Processing at the Digital Age Journal*, t. 27, n° 2, p. 32-39, 2023, Consulté le 8 mai 2026.

[57] B. BENSAFI, « Deep Learning et l'Aide au Diagnostic Médical (Application à la Classification d'image médicale), » fr, Consulté le 5 mai 2026, mém. de mast., 2023.

[58] WIKIPEDIA CONTRIBUTORS, *Kaggle — Wikipedia*, https://en.wikipedia.org/wiki/Kaggle, Consulté le 16 mai 2026, 2026.

[59] H. TALBI et N. KHALDI, « Étude comparative des méthodes de Deep Learning pour la classification et la segmentation de la rétinopathie diabétique, » Encadré par Mme Bougherira Hamida. Année universitaire 2021–2022. Consulté le 7 mai 2026, Mémoire de Master en Électronique, Spécialité : Électronique des Systèmes Embarqués, Université Saad Dahlab de Blida, Faculté de Technologie, Département d'Électronique, Blida, Algérie, 2022.

[60] P.-Y. GERLAT, *5 choses à savoir avant d'apprendre à programmer en Python*, Consulté le 16 mai 2026, oct. 2018. adresse : https://www.actuia.com/actualite/5-choses-a-savoir-sur-avant-de-commencer-a-programmer-en-python/

[61] H. MASSIOUN, « Détection et classification automatique des maladies des plantes à partir des images de feuilles, » Option : Système d'information avancée. Soutenu le 04 juillet 2024. Consulté le 6 mai 2026, Mémoire de Master Recherche en Informatique, Université A. Mira de Béjaïa, Béjaïa, Algérie, juill. 2024.

[62] CONTRIBUTEURS WIKIPÉDIA, *TensorFlow — Wikipédia*, https://fr.wikipedia.org/wiki/TensorFlow, Consulté le 16 mai 2026, 2026.

[63] K. BENALLAL et M. BOUATTOU, « Conception et implémentation d'un système de détection et classification des maladies foliaires, » Encadré par Dr M. Bouhedda et Mr H. Benyezza. Année universitaire 2020–2021. Consulté le 6 mai 2026, Mémoire de Master en Électronique, Spécialité : Électronique des Systèmes Embarqués, Université Dr. Yahia Fares de Médéa, Faculté de Technologie, Département du Génie Électrique, Médéa, Algérie, 2021.

[64] KERAS TEAM, *Keras RS Documentation*, https://keras.io/keras_rs/, Consulté le 16 mai 2026, 2026.

[65] ICON-ICONS.COM, *Logo OpenCV*, https://icon-icons.com/fr/icone/opencv-logo/1708883, Consulté le 16 mai 2026, 2026.

[66] WIKIMEDIA COMMONS, *NumPy logo (2020)*, https://fr.wikipedia.org/wiki/Fichier:NumPy_logo_2020.svg, Consulté le 16 mai 2026, 2026.

[67] A. AOUCI, *Classification and Data Analysis with Python Libraries*, Referenced for Pandas data manipulation and analysis. Consulté le 18 mai 2026, 2024.

[68] T. T. MAC, T.-D. NGUYEN, H.-K. DANG, D.-T. NGUYEN et X.-T. NGUYEN, « Intelligent Agricultural Robotic Detection System for Greenhouse Tomato Leaf Diseases Using Soft Computing Techniques and Deep Learning, » *Scientific Reports*, t. 14, n° 1, p. 23 887, 2024, Consulté le 18 mai 2026. DOI : 10.1038/s41598-024-75285-5

[69] A. GATLA et al., « Optimizing Edge AI for Tomato Leaf Disease Identification, » *Engineering, Technology & Applied Science Research*, t. 14, n° 4, p. 16 061-16 068, 2024, Consulté le 19 mai 2026. DOI : 10.48084/etasr.7802

[70] H. GHOSH et al., « Advanced Neural Network Architectures for Tomato Leaf Disease Diagnosis in Precision Agriculture, » *Discover Sustainability*, t. 6, n° 1, p. 312, 2025, Consulté le 18 mai 2026. DOI : 10.1007/s43621-025-01149-1

[71] S. Z. M. ZAKI, M. A. ZULKIFLEY, M. M. STOFA, N. A. M. KAMARI et N. A. MOHAMED, « Classification of Tomato Leaf Diseases Using MobileNet V2, » *IAES International Journal of Artificial Intelligence*, t. 9, n° 2, p. 290, 2020, Consulté le 19 mai 2026. DOI : 10.11591/ijai.v9.i2.pp290-296

[72] S. KUMAR et al., « Spatial Attention-Guided Pre-Trained Networks for Accurate Identification of Crop Diseases, » *Scientific Reports*, t. 15, n° 1, p. 23 213, 2025, Consulté le 19 mai 2026. DOI : 10.1038/s41598-025-08004-3

[73] S. K. TAMIM et M. RASHAD, *YOLO-Based Image Detection System for Early Detection of Tomato Plant Diseases*, Preprints.org, Consulté le 20 mai 2026, 2025. DOI : 10 . 20944 / preprints202506.0040.v1

[74] L. T. RAMOS et A. D. SAPPA, « A Comprehensive Analysis of YOLO Architectures for Tomato Leaf Disease Identification, » *Scientific Reports*, t. 15, n° 1, p. 26 890, 2025, Consulté le 20 mai 2026. DOI : 10.1038/s41598-025-11064-0

[75] D. R. HAMMOU et M. BOUBAKER, « Tomato Plant Disease Detection and Classification using Convolutional Neural Network Architectures Technologies, » in *Networking, Intelligent Systems and Security : Proceedings of NISS 2021*, Consulté le 20 mai 2026, Springer, 2021, p. 33-44. DOI : 10.1007/978-981-16-3637-0_3