

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC
À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR

MARIAM YOUNES

DÉTECTION DES MAUVAISES HERBES
EN AGRICULTURE

MARS 2025

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

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

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

REMERCIEMENTS

Tout d'abord, j'exprime mes plus sincères remerciements et ma gratitude envers Allah Tout-Puissant pour ses bénédictions divines, qui m'ont permis de mener à bien ce projet avec succès.

Je souhaite remercier les personnes suivantes pour leur aide dans la réalisation de ce projet. Je suis profondément reconnaissante à mon directeur de recherche, Monsieur Mhammed Mesfioui, professeur au Département de mathématiques et d'informatique de l'Université du Québec à Trois-Rivières. Sans son aide et son soutien tout au long de ce projet, cela n'aurait pas été possible. Sa patience infinie, ses conseils éclairés, son encouragement constant, sa supervision dynamique et bienveillante, ainsi que ses retours pertinents, ses précieux conseils, ainsi que sa lecture de nombreux brouillons et leurs corrections à chaque étape ont permis la réalisation de ce projet.

Je suis très reconnaissante envers tous mes professeurs qui m'ont guidée avec leurs précieux conseils pour terminer mes études. Je tiens également à exprimer ma gratitude envers toutes les personnes qui m'ont aidée à mener à bien ce projet.

Enfin, je tiens à exprimer avec respect ma profonde reconnaissance envers mes parents pour leur soutien inébranlable et leur patience tout au long de mon parcours.

Résumé

Ce projet explore la détection et la segmentation des mauvaises herbes dans divers environnements géographiques, en recourant à des techniques d'apprentissage profond pour développer une solution automatisée et performante. Deux modèles principaux ont été mis en œuvre : un modèle de classification basé sur ResNet-50, appliqué au dataset DeepWeeds pour identifier les mauvaises herbes, et un modèle de segmentation utilisant YOLO et Mask R-CNN pour localiser et délimiter précisément les mauvaises herbes dans leur environnement.

Afin de garantir l'adaptabilité aux différentes régions géographiques et aux variations de types de mauvaises herbes, un modèle incrémental a été développé. Ce modèle repose sur le affinement du modèle et l'apprentissage incrémental, permettant de généraliser le système sans nécessiter de réentraînement complet à chaque nouvelle donnée. De plus, une méthode d'auto-étiquetage a été intégrée pour accélérer et optimiser l'annotation des images.

Cette approche intégrée assure une détection fiable et une gestion efficace des mauvaises herbes dans des environnements variés, jetant ainsi les bases de systèmes automatisés avancés pour l'agriculture de précision.

Abstract

This project addresses weed detection and segmentation across diverse geographic regions using deep learning techniques to develop a robust and automated solution. Two main models were implemented : a classification model based on ResNet-50, applied to the DeepWeeds dataset to identify weeds, and a segmentation model utilizing YOLO in addition to Mask R-CNN to accurately locate and outline weeds in their environment.

To ensure adaptability to different geographic regions and varying weed types, an incremental model was developed. This model leverages fine-tuning and incremental learning, enabling the system to generalize without requiring complete retraining for each new data set. Additionally, an auto-labeling method was incorporated to accelerate and optimize image annotation.

This combined approach provides a reliable detection and efficient management of weeds in varied environments, laying the groundwork for advanced automated systems in precision agriculture.

Table des matières

| | |
|---|------------|
| Table des matières | vi |
| Liste de Figures | vii |
| Liste de Tableaux | ix |
| 1 Introduction générale | 1 |
| 1.1 Problématique | 1 |
| 1.2 Aperçu du projet | 2 |
| 1.3 Motivation | 2 |
| 1.4 Objectifs et méthodologie | 2 |
| 1.5 Résultats du projet | 3 |
| 1.6 Organisation du rapport | 4 |
| 2 Revue de la littérature | 5 |
| 3 Méthodologie | 11 |
| 3.1 Architecture CNN - Réseau de Neurones Convolutifs | 12 |
| 3.1.1 L'apprentissage des caractéristiques | 12 |
| 3.1.2 Classification | 15 |
| 3.2 YOLO (You Only Look Once) | 17 |
| 3.2.1 Caractéristiques principales : | 17 |
| 3.2.2 Schéma de YOLO : | 17 |
| 3.3 Mask R-CNN | 18 |
| 3.3.1 Caractéristiques principales : | 18 |
| 3.3.2 Schéma de Mask R-CNN : | 18 |
| 3.4 Environnement de développement et outils | 19 |
| 4 Détection avec ResNet50 sur deepWeeds | 21 |
| 4.1 Préparation des données | 21 |
| 4.1.1 Chargement et prétraitement des données | 21 |
| 4.1.2 Augmentation des données | 23 |
| 4.2 Modèle de classification | 25 |

| | | |
|----------|---|-----------|
| 4.2.1 | Architecture du modèle | 25 |
| 4.2.2 | Optimisation | 26 |
| 4.3 | Historique de l'entraînement | 27 |
| 4.3.1 | Analyse des résultats d'entraînement | 28 |
| 4.3.2 | Diminution de la perte et convergence | 29 |
| 4.3.3 | Interprétation des courbes de précision et de perte | 29 |
| 4.3.4 | Impact de l'optimisation avec SGD | 29 |
| 4.4 | Évaluation des performances | 29 |
| 4.4.1 | Précision globale | 29 |
| 4.4.2 | Rapport de classification | 29 |
| 4.4.3 | Matrice de confusion | 31 |
| 4.4.4 | Analyse des performances d'autres modèles de classification | 31 |
| 4.5 | Conclusion | 32 |
| 5 | Modèle incrémental | 33 |
| 5.1 | Datasets utilisés | 33 |
| 5.2 | Modèle initial | 34 |
| 5.2.1 | Configuration et entraînement du modèle | 34 |
| 5.2.2 | Sauvegarde et préparation à la généralisation | 35 |
| 5.3 | Application trained_model.pth | 37 |
| 5.3.1 | Extraction des échantillons | 37 |
| 5.3.2 | Prétraitement et fusion | 37 |
| 5.3.3 | Entraînement et fine-tuning sur le dataset enrichi | 37 |
| 5.4 | Application fine_tuned_model.pth | 39 |
| 5.4.1 | Résultats sur l'ensemble de test | 39 |
| 5.4.2 | Test de performance de fine_tuned_model_dt3.pth sur le premier et le deuxième dataset : DeepWeeds et Crop-and-weed-detection-data-with-bounding-boxes | 39 |
| 5.5 | Conclusion | 40 |
| 6 | Étiquetage automatique | 41 |
| 6.1 | Prétraitement des données | 42 |
| 6.2 | Entraînement initial sur un sous-ensemble annoté | 42 |
| 6.3 | Auto-étiquetage des données non annotées | 42 |
| 6.4 | Construction du dataset étendu | 43 |
| 6.5 | Analyse des résultats | 43 |
| 6.6 | Conclusion | 45 |
| 7 | Détection et segmentation avec YOLO et Mask R-CNN | 46 |
| 7.1 | YOLO-v5, v8 | 46 |
| 7.1.1 | Description du dataset | 46 |
| 7.1.2 | Préparation de data | 47 |

| | | |
|-------------------|---|-----------|
| 7.1.3 | Construction du modèle YOLOV | 48 |
| 7.1.4 | Résultat de YOLO (v5) | 48 |
| 7.1.5 | Interprétation générale YOLO(v5) | 49 |
| 7.1.6 | Temps d'inférence YOLO (v5) | 49 |
| 7.1.7 | Visualisation d'un résultat YOLO (v5) | 50 |
| 7.1.8 | Résultat de YOLO (v8) | 50 |
| 7.1.9 | Interprétation générale YOLO (v8) | 50 |
| 7.1.10 | Temps d'Inférence YOLO (v8) | 52 |
| 7.1.11 | Visualisation des résultats YOLO (v8) | 52 |
| 7.1.12 | Comparaison et conclusion | 52 |
| 7.2 | Mask R-CNN | 53 |
| 7.2.1 | Description du dataset | 53 |
| 7.2.2 | Installation du detectron2 | 54 |
| 7.2.3 | Préparation des données | 54 |
| 7.2.4 | Configuration du modèle | 54 |
| 7.2.5 | Entrainement du modèle | 55 |
| 7.2.6 | Évaluation du modèle | 55 |
| 7.2.7 | Résultats de Mask R-CNN | 55 |
| 7.2.8 | Interprétation des résultats | 56 |
| 7.2.9 | Visualisation des résultats | 56 |
| 7.2.10 | Conclusion | 58 |
| 8 | Conclusion générale | 59 |
| References | | 61 |

Table des figures

| | | |
|-----|--|----|
| 3.1 | Architecture de modèle CNN | 12 |
| 3.2 | Convolution avec un kernel de 2x2x1 et stride s = 1 | 13 |
| 3.3 | Processus de mise à plat. | 15 |
| 3.4 | Schéma de YOLO | 17 |
| 3.5 | Schéma de Mask R-CNN | 18 |
| 4.1 | Échantillons visuels des différentes espèces de mauvaises herbes du jeu de données DeepWeeds. | 22 |
| 4.2 | Répartition des classes des espèces de mauvaises herbes dans le jeu de données DeepWeeds. | 23 |
| 4.3 | Architecture de ResNet50. | 25 |
| 4.4 | Courbes de la perte (Loss) et de la précision (Accuracy). | 28 |
| 4.5 | Rapport de classification. | 30 |
| 4.6 | Matrice de confusion. | 31 |
| 5.1 | Répartition des deux classes des herbes dans le jeu de données DeepWeeds. | 35 |
| 5.2 | Courbes de la perte (Loss) et de la précision (Accuracy). | 36 |
| 5.3 | Matrice de confusion. | 36 |
| 6.1 | Modèle 1 :Entraînement et validation (précision, perte), Leraning rate schedule. | 44 |
| 6.2 | Modèle 2 :Entraînement et validation (précision, perte). | 44 |
| 7.1 | Exemples de différents types de mauvaises herbes dans le jeu de données crop-and-weed-detection-data-with-bounding-boxes | 47 |
| 7.2 | Visualisation du résultat sur l'image agri_0_1009.jpeg. | 48 |
| 7.3 | Courbe de précision et rappel. | 49 |
| 7.4 | Visualisation du résultat sur agri_0_3943.jpeg. | 50 |
| 7.5 | Courbe de rappel en fonction de la confiance et Courbe de précision en fonction de la confiance. | 51 |
| 7.6 | Matrice de confusion. | 51 |
| 7.7 | Visualisation du résultat sur l'image agri_0_3943.jpeg. | 53 |

| | | |
|------|---|----|
| 7.8 | Échantillon d'images de mauvaises herbes dans différentes conditions de luminosité et d'angles. | 54 |
| 7.9 | Localisation des mauvaises herbes par boîtes de délimitation et par masque, réalisée par le modèle généré.. | 57 |
| 7.10 | Localisation des instances de mauvaises herbes par masques et boîtes de délimitation. | 57 |

Liste des tableaux

| | | |
|-----|--|----|
| 4.1 | Espèces et leurs étiquettes correspondantes. | 22 |
| 4.2 | Performances des différents modèles de classification. | 31 |
| 5.1 | Précision par classe. | 38 |
| 7.1 | Résumé des performances par classe (YOLO v5) | 49 |
| 7.2 | Résumé des performances par classe (YOLO v8). | 50 |
| 7.3 | Performances en fonction des métriques pour les boîtes englobantes et la segmentation. | 55 |

Chapitre 1

Introduction générale

L'agriculture moderne fait face à de nombreux défis, dont la gestion efficace des mauvaises herbes est l'un des plus cruciaux. Les mauvaises herbes, en compétition directe avec les cultures pour les ressources telles que la lumière, l'eau et les nutriments, peuvent entraîner des pertes de rendement considérables. La gestion traditionnelle des mauvaises herbes repose en grande partie sur l'utilisation d'herbicides chimiques, ce qui présente des risques environnementaux, économiques et de santé, notamment la pollution des sols et la résistance accrue des mauvaises herbes. Ces pratiques soulèvent la nécessité de développer des alternatives plus durables et précises, notamment l'intégration de technologies avancées comme la vision par ordinateur et les techniques d'apprentissage automatique pour la détection automatique des mauvaises herbes sur le terrain.

Dans ce contexte, les systèmes de détection automatisée des mauvaises herbes, basés sur des images et l'intelligence artificielle, offrent une solution prometteuse pour réduire l'utilisation d'herbicides et améliorer la gestion des cultures. Cependant, la conception de ces systèmes présente des défis techniques importants, tels que la capacité à distinguer efficacement les mauvaises herbes des cultures dans des environnements complexes et variés. Cela nécessite l'exploitation d'algorithmes robustes et de données précises pour garantir une reconnaissance fiable dans des conditions réelles de culture.

1.1 Problématique

Le problème auquel nous faisons face est celui de la détection automatisée et précise des mauvaises herbes dans les champs agricoles, où les conditions d'éclairage, les textures des plantes, et la présence de divers éléments dans le sol complexifient cette tâche. L'objectif est d'identifier efficacement les mauvaises herbes afin de minimiser l'utilisation d'herbicides et d'améliorer la productivité agricole.

1.2 Aperçu du projet

Ce projet vise à développer un modèle basé sur des techniques de vision par ordinateur et d'apprentissage automatique pour la détection des mauvaises herbes dans des images de cultures. En utilisant des algorithmes de traitement d'image et de classification, notre système sera capable de distinguer les mauvaises herbes des cultures avec précision.

1.3 Motivation

La motivation principale de ce projet repose sur la nécessité de réduire l'impact environnemental et les risques pour la santé associés à l'utilisation excessive d'herbicides dans l'agriculture. Bien que ces produits chimiques soient couramment utilisés pour lutter contre les mauvaises herbes, leur usage intensif a des conséquences graves, non seulement sur l'environnement, mais aussi sur la santé humaine. Des études ont établi un lien entre l'exposition prolongée aux herbicides et des maladies graves, telles que certains cancers (lymphome non hodgkinien, cancer de la prostate, cancer du sein), des perturbations endocrinianes, et des troubles respiratoires.

La demande croissante pour des solutions alternatives aux herbicides conventionnels, qui minimisent ces effets néfastes, devient ainsi cruciale. Ce projet, en s'inscrivant dans le cadre de l'agriculture de précision, vise à fournir aux agriculteurs des outils innovants pour la détection automatique des mauvaises herbes. En améliorant l'efficacité de la gestion des mauvaises herbes grâce à des technologies basées sur la vision par ordinateur et l'intelligence artificielle, il devient possible de réduire significativement l'usage des herbicides. Cela contribue non seulement à protéger les écosystèmes agricoles, mais aussi à limiter l'exposition des travailleurs agricoles et des consommateurs à des substances toxiques.

L'objectif est donc de développer des méthodes plus intelligentes et respectueuses de l'environnement, qui permettront aux agriculteurs de cibler précisément les mauvaises herbes sans dépendre excessivement des produits chimiques, tout en optimisant la production agricole.

1.4 Objectifs et méthodologie

La méthodologie adoptée dans ce projet repose sur plusieurs étapes clés, élaborées pour répondre aux objectifs spécifiques de détection et de gestion des mauvaises herbes dans différents environnements agricoles.

Tout d'abord, nous procédons à l'acquisition et au prétraitement des données d'images de cultures provenant de divers environnements agricoles. Ces images sont ensuite soumises à un prétraitement rigoureux pour garantir leur qualité optimale pour l'analyse, incluant la normalisation, l'augmentation des données et la correction des imperfections.

Nous utilisons ensuite des modèles de classification basés sur des réseaux de neurones convolutifs (CNN), notamment ResNet-50, pour identifier les mauvaises herbes dans les

images. Ce modèle de classification est conçu pour distinguer avec précision les mauvaises herbes des cultures saines, en se basant sur les caractéristiques visuelles des images.

Pour améliorer la robustesse du modèle face aux variations des environnements et des types de mauvaises herbes, nous intégrons une approche d'adaptation incrémentale. Ce processus permet de mettre à jour le modèle au fur et à mesure qu'il reçoit de nouvelles données, sans nécessiter un réentraînement complet à chaque itération. L'adaptation incrémentale permet ainsi de maintenir la performance du modèle tout en le rendant flexible et capable de s'ajuster aux différences géographiques et aux variations dans les espèces de mauvaises herbes.

Une des étapes cruciales pour enrichir le dataset est l'intégration de l'auto-étiquetage. L'auto-étiquetage est une méthode semi-supervisée dans laquelle un modèle initial, formé sur un sous-ensemble d'images annotées, est utilisé pour prédire les étiquettes des images non annotées. Cette approche permet de constituer rapidement un dataset étendu et diversifié, tout en réduisant le besoin d'un étiquetage manuel coûteux et chronophage. Une fois que les nouvelles images sont étiquetées automatiquement, elles sont ajoutées à l'ensemble de données pour améliorer les performances du modèle.

Enfin, pour localiser et délimiter précisément les mauvaises herbes dans les images, nous adoptons une approche de segmentation d'images. Nous utilisons un modèle hybride combinant Mask R-CNN et YOLO pour effectuer une segmentation de haute précision. Mask R-CNN est utilisé pour détecter et segmenter les mauvaises herbes dans les images, tandis que YOLO (You Only Look Once) permet la détection en temps réel et la localisation des objets. Cette combinaison de modèles permet d'obtenir des résultats de segmentation détaillés et rapides, essentiels pour une gestion efficace des mauvaises herbes dans un contexte agricole automatisé.

1.5 Résultats du projet

Les résultats attendus ou potentiels de ce projet sont multiples et varient en fonction des objectifs atteints. L'un des principaux résultats serait le développement d'un système de détection et de segmentation des mauvaises herbes qui soit capable de fonctionner de manière efficace dans divers environnements agricoles, avec une précision élevée dans l'identification des mauvaises herbes par rapport aux cultures saines. Ce système pourrait être utilisé pour automatiser la gestion des mauvaises herbes, réduisant ainsi la dépendance aux méthodes manuelles et chimiques, et contribuant à une agriculture plus durable.

Un autre résultat clé est l'amélioration de la généralisation du modèle grâce à l'adaptation incrémentale, ce qui permettrait au système de s'adapter facilement à de nouveaux ensembles de données, comprenant des variations géographiques ou des types de mauvaises herbes non observées initialement. Cette capacité d'adaptation pourrait être cruciale pour déployer ce système dans différents contextes agricoles à travers le monde.

L'auto-étiquetage des images d'herbes permettrait de constituer un dataset étendu et diversifié, sans nécessiter de travail manuel coûteux. L'utilisation de cette méthode

semi-supervisée pourrait accélérer considérablement la création de bases de données pour l'entraînement de modèles de détection, particulièrement dans les régions où l'accès à des experts en identification des mauvaises herbes est limité.

Enfin, la segmentation précise des mauvaises herbes à l'aide des modèles Mask R-CNN et YOLO pourrait mener à des outils plus efficaces pour la gestion ciblée des mauvaises herbes, facilitant leur élimination dans des zones spécifiques sans affecter les cultures. Cette précision dans la segmentation pourrait également offrir de nouvelles possibilités pour des systèmes d'agriculture de précision qui optimisent l'utilisation des ressources et réduisent les impacts environnementaux.

1.6 Organisation du rapport

Ce rapport est structuré comme suit :

- Chapitre 1 : Introduction générale au problème de la détection des mauvaises herbes, ainsi que les motivations et objectifs du projet.
- Chapitre 2 : Revue de la littérature, couvrant les travaux existants et les avancées dans le domaine.
- Chapitre 3 : Méthodologies.
- Chapitre 4 : Classification avec ResNet50 sur DeepWeeds.
- Chapitre 5 : Modèle incrémental
- Chapitre 6 : Étiquetage automatique .
- Chapitre 7 : YOLO-Mask R-CNN.
- Chapitre 8 :Conclusion générale.

Chapitre 2

Revue de la littérature

Dans le contexte de l'agriculture moderne, la gestion des mauvaises herbes représente un défi crucial pour garantir des rendements optimaux tout en minimisant l'impact environnemental. La concurrence des mauvaises herbes pour les ressources vitales comme l'eau, la lumière et les nutriments affecte considérablement la productivité des cultures [1,2,3,4]. Traditionnellement, les méthodes de désherbage reposent principalement sur l'utilisation de produits chimiques, une approche coûteuse et souvent néfaste pour les écosystèmes et la santé humaine. Face à cette problématique, l'essor des technologies de vision par ordinateur et d'intelligence artificielle a ouvert la voie à des solutions plus durables et précises pour la détection et la gestion des mauvaises herbes. Ces outils permettent d'identifier automatiquement les mauvaises herbes dans les champs, facilitant ainsi des interventions ciblées et une réduction significative de l'utilisation d'herbicides.

La revue de la littérature qui suit se penche sur les principales approches et innovations récentes dans le domaine de la détection des mauvaises herbes à l'aide de méthodes automatisées. Nous explorerons les différents systèmes basés sur la vision artificielle, les techniques de segmentation d'images et les algorithmes d'apprentissage profond, qui permettent d'identifier et de différencier efficacement les mauvaises herbes des cultures. Ces avancées, déjà appliquées dans le cadre du phénotypage des plantes et de l'agriculture de précision, posent les bases d'une agriculture plus intelligente et respectueuse de l'environnement. L'analyse des travaux existants mettra en lumière les réussites, les défis actuels et les perspectives d'amélioration pour aboutir à des solutions robustes et généralisables de détection automatique des mauvaises herbes.

Depuis plus de 50 ans, la recherche sur la détection et la reconnaissance des mauvaises herbes a évolué de manière significative, avec des progrès remarquables dans l'intégration de nouvelles technologies [1,3,5,6,7,8]. L'article [9] examine les avancées et les limites des approches actuelles, en mettant un accent particulier sur l'utilisation des algorithmes d'apprentissage profond pour les systèmes agricoles à grande échelle.

Au début de cette recherche menée par les auteurs de cet article, l'identification des mauvaises herbes reposait principalement sur des méthodes de vision par ordinateur basées sur la couleur et la forme des plantes, souvent dans des environnements hautement

contrôlés. Ces premières techniques avaient des limites importantes, notamment en termes de précision et de capacité à gérer la variabilité des conditions environnementales. La montée en puissance des algorithmes d'apprentissage profond, et en particulier des réseaux de neurones convolutifs (CNN), a permis des avancées majeures, transformant le paysage de la reconnaissance des mauvaises herbes en offrant des possibilités d'application dans des environnements plus complexes et plus diversifiés. Ces nouvelles approches permettent désormais une identification précise des espèces de mauvaises herbes et des caractéristiques morphologiques des plantes à grande échelle. Les auteurs soulignent que l'utilisation de cadres d'apprentissage profond tels que Pytorch et Tensorflow a considérablement amélioré les taux de précision des algorithmes de reconnaissance des mauvaises herbes, atteignant respectivement 97 % et 96 %. Ces résultats témoignent de l'efficacité croissante de ces technologies dans les systèmes agricoles modernes. Cependant, pour maximiser leur potentiel, des efforts supplémentaires sont nécessaires pour créer des bases de données d'images à grande échelle, spécifiquement dédiées aux mauvaises herbes. Les auteurs comparent cette opportunité à l'effet qu'ImageNet a eu sur la recherche en apprentissage profond, ouvrant la voie à des développements algorithmiques significatifs.

Malgré ces avancées, plusieurs obstacles persistent. L'un des principaux défis réside dans la vitesse de calcul des algorithmes, qui doit être améliorée pour répondre aux besoins des systèmes de culture à grande échelle. De plus, la disponibilité de données spécifiques aux mauvaises herbes reste limitée, et les algorithmes actuels peinent à gérer la variabilité des conditions sur le terrain, telles que les différences de luminosité ou les occlusions causées par d'autres plantes ou objets. Ces limitations ralentissent l'adoption généralisée de ces technologies dans les grandes exploitations agricoles, où des solutions robustes et adaptées à diverses situations sont indispensables.

L'étude met également en avant des perspectives prometteuses pour l'intégration des algorithmes de reconnaissance des mauvaises herbes dans les stratégies de gestion des cultures. En permettant une application plus ciblée et plus précise des traitements de lutte contre les mauvaises herbes, ces algorithmes offrent des gains potentiels en termes de rendement des cultures et de qualité des récoltes. Les traitements peuvent ainsi être appliqués en fonction des risques spécifiques à chaque parcelle, réduisant l'utilisation excessive de produits chimiques et améliorant la durabilité des pratiques agricoles.

Toutefois, les auteurs soulignent que le développement de nouvelles architectures algorithmiques, spécialement conçues pour la reconnaissance des mauvaises herbes, est crucial pour surmonter les limitations actuelles. Ces efforts de recherche devraient inclure la création de bases de données d'images à grande échelle, capables de capturer la diversité des espèces de mauvaises herbes et des conditions environnementales rencontrées sur le terrain. L'avenir de cette discipline repose donc sur l'amélioration de la robustesse des algorithmes et sur l'extension de leurs capacités à des contextes plus variés.

En conclusion, les auteurs de [9] ont signalé que l'apprentissage profond a le potentiel de transformer radicalement la gestion des mauvaises herbes dans les systèmes agricoles à grande échelle. Cependant, pour tirer pleinement parti de ces technologies, les auteurs ont

signalé qu'il est nécessaire de surmonter plusieurs défis techniques, notamment en matière de rapidité de calcul, de disponibilité des données et de variabilité environnementale. L'étude appelle à une poursuite des efforts de recherche pour développer des algorithmes plus robustes et adaptés aux réalités du terrain, tout en favorisant une adoption plus large des solutions technologiques par l'industrie agricole.

Dans le contexte de l'agriculture moderne, les avancées technologiques ont permis de révolutionner les méthodes de suivi et d'analyse des cultures. L'article [10] explore cette transformation en proposant une plateforme basée sur la vision pour la phénotypage des plantes. Dans [10], les auteurs présentent le “Robust Trait Phenotyping Platform (RTPP)”, un outil web qui permet une surveillance non invasive et à faible coût des plantes en utilisant un ensemble d'imagerie spectrale à 4 canaux et des algorithmes d'apprentissage automatique.

L'un des principaux apports de [10] réside dans la capacité du RTPP à détecter, localiser et suivre la croissance des plantes, même dans des scénarios complexes où elles se chevauchent. Cette plateforme utilise une approche de segmentation basée sur la couleur pour surmonter les défis posés par les objets se touchant ou se chevauchant dans les images. Elle est capable d'estimer divers indices végétaux, d'évaluer des traits phénotypiques et de visualiser les tendances de croissance au fil du temps grâce à un tableau de bord dynamique. Ces fonctionnalités apportent une innovation notable dans le domaine de l'agriculture de précision, permettant une analyse plus fine et plus réactive des cultures.

La plateforme RTPP se distingue également par sa capacité à traiter de grandes quantités de données visuelles tout en maintenant une interface utilisateur simple et intuitive. Elle utilise des technologies telles que OpenCV [11] pour le traitement d'images et s'appuie sur des algorithmes de machine learning pour analyser les données recueillies. Cette combinaison permet non seulement de suivre quotidiennement la croissance des plantes, mais aussi de fournir des indicateurs phénotypiques clés, comme les indices de végétation, la teneur en chlorophylle et en azote.

Cependant, l'article met en lumière certaines limitations des systèmes existants, qui peinent à gérer des images où les plantes se chevauchent ou lorsque la position des objets dans l'image varie considérablement. Le RTPP, en revanche, propose une solution robuste à ces problèmes, tout en offrant des perspectives prometteuses pour l'avenir. Les auteurs suggèrent que l'amélioration de la plateforme pourrait passer par l'intégration d'algorithmes d'apprentissage automatique plus avancés, capables de gérer des scénarios encore plus complexes, tels que l'analyse de plantes à base de tige ou l'extraction de traits non linéaires à partir de données multispectrales.

Les résultats expérimentaux confirment la capacité du RTPP à offrir des estimations précises des traits phénotypiques, même dans des environnements agricoles complexes. Testée sur le terrain, cette plateforme a prouvé son efficacité pour détecter et suivre les plantes individuelles, tout en offrant une interface conviviale pour l'analyse des données. Cela rend le système particulièrement adapté aux applications agricoles de grande envergure, où la capacité à surveiller la croissance des plantes en temps réel et à analyser les

données visuelles de manière rapide et précise est essentielle.

Dans ce même cadre du travail présenté dans [10], le même modèle peut être également appliqué à la reconnaissance et à la gestion des mauvaises herbes dans les systèmes agricoles. En effet, la détection des mauvaises herbes repose sur des techniques de segmentation et de localisation similaires à celles utilisées par la plateforme RTPP pour identifier et suivre la croissance des plantes. L'utilisation d'imagerie multispectrale et d'algorithmes d'apprentissage automatique, comme ceux décrits dans [10], pourrait être transposée pour distinguer les mauvaises herbes des cultures, permettant ainsi une intervention ciblée en temps réel. De plus, la gestion des scénarios complexes, où les plantes ou mauvaises herbes se chevauchent ou sont proches les unes des autres, constitue un défi commun dans les deux domaines. L'approche proposée pour surmonter ces difficultés pourrait ainsi inspirer des solutions innovantes pour la détection automatique des mauvaises herbes, optimisant les pratiques de désherbage et réduisant l'utilisation de produits chimiques dans les cultures [10].

En conclusion, le RTPP est une plateforme robuste et efficace pour le phénotypage des plantes, adaptée à une large gamme d'applications en agriculture de précision, en sélection végétale et en recherche sur les plantes. Bien que des améliorations soient encore nécessaires, notamment pour gérer des traits phénotypiques plus complexes et pour rendre l'outil plus accessible à un public plus large, les résultats obtenus jusqu'à présent démontrent un potentiel énorme pour cette technologie dans les systèmes agricoles modernes.

Dans [12], les auteurs explorent l'application de l'apprentissage profond pour la détection des mauvaises herbes dans les champs de blé. Le contrôle des mauvaises herbes dans les cultures de blé est essentiel pour optimiser la production et assurer la sécurité alimentaire, en particulier dans les régions semi-arides. Les auteurs se sont concentrés sur trois espèces de mauvaises herbes spécifiques : la moutarde des champs, le chardon des champs et la dauphinelle fourchue, à travers toutes les phases de leur développement. L'étude s'est basée sur l'utilisation du réseau neuronal YOLOv5, un modèle de détection d'objets en temps réel connu pour sa rapidité et son efficacité. Plus précisément, elle a exploité YOLOv5s, une version allégée de YOLOv5, conçue pour être plus rapide et moins gourmande en ressources, tout en conservant des performances satisfaisantes, pour détecter les mauvaises herbes à différents stades de croissance. Le processus de collecte des données a été réalisé à l'aide d'un drone qui a capturé des images et des vidéos sur 185 champs de blé dans la région de Tokat, en Turquie, couvrant cinq périodes phénologiques pour chaque espèce. Les chercheurs ont étiqueté un total de 145 792 objets dans les images, constituant un vaste ensemble de données pour l'entraînement du modèle. Le réseau YOLOv5 s'est révélé particulièrement efficace pour détecter les mauvaises herbes, avec une précision allant de 0,45 à 0,96, selon l'espèce et le stade de développement. L'étude a notamment montré que les mauvaises herbes étaient détectées avec une précision de 0,86 à tous les stades, à l'exception du stade de fructification, où la performance était plus faible. Les résultats soulignent le potentiel de l'apprentissage profond pour améliorer la gestion des mauvaises

herbes dans les champs de blé. En effet, la détection précise des mauvaises herbes dès leur apparition permet d'optimiser les méthodes de lutte, réduisant ainsi la concurrence des mauvaises herbes pour les ressources et augmentant la productivité du blé. Pour l'espèce dauphinelle fourchue, par exemple, le modèle YOLOv5s a atteint des niveaux de précision compris entre 0,45 et 0,93, démontrant son efficacité dans des conditions de terrain réelles.

Les auteurs dans [12] suggèrent que des recherches futures devraient se concentrer sur l'évaluation du réseau YOLOv5 avec les versions plus récentes de ce modèle, ainsi que sur l'élargissement de son application à d'autres espèces de mauvaises herbes dans les champs de blé. De plus, ils proposent d'optimiser davantage le modèle pour améliorer la gestion des mauvaises herbes et offrir des recommandations plus précises aux agriculteurs. En conclusion de [12], les auteurs mettent en lumière les bénéfices de l'intégration de l'apprentissage profond pour la détection des mauvaises herbes dans l'agriculture. Grâce à l'utilisation de technologies comme les drones et les réseaux neuronaux, les agriculteurs peuvent désormais gérer de manière plus efficace les mauvaises herbes dans leurs cultures, améliorant ainsi la production de blé et contribuant à une agriculture plus durable et productive.

Dans l'article intitulé [13], les auteurs se concentrent sur la détection des mauvaises herbes dans les champs de betterave à sucre à l'aide de techniques d'apprentissage automatique. Cette étude compare deux méthodes de classification : les réseaux de neurones artificiels (ANN) et les machines à vecteurs de support (SVM), en utilisant des caractéristiques de forme extraites d'images.

Les images ont été prises au stade de croissance à quatre feuilles des plantes, et les caractéristiques de forme ont été extraites grâce à des descripteurs de Fourier et des invariants de moments. Le processus comprenait l'acquisition d'images, la segmentation, l'extraction de caractéristiques, puis la classification des plantes. Un total de 50 images a été utilisé, comprenant des betteraves sucrières ainsi que quatre espèces communes de mauvaises herbes présentes dans les champs de betteraves sucrières : l'amarante, le chénopode, la moutarde à feuilles de lièvre, et la moutarde rugueuse. Les données ont été traitées avec l'outil de traitement d'images de MATLAB.

Les résultats ont montré que la SVM surpassait l'ANN en termes de précision, atteignant une exactitude globale de 95% contre 92,92% pour l'ANN. En particulier, la SVM a correctement classé 93,33% des mauvaises herbes et 96,67% des plants de betterave, tandis que l'ANN a correctement classé 92,50% des mauvaises herbes et 93,33% des betteraves. Cette différence statistiquement significative dans l'identification des types de plantes et la discrimination entre cultures et mauvaises herbes favorise l'utilisation de la SVM dans ce contexte.

L'étude dans [13] conclut que les caractéristiques de forme sont efficaces pour la détection des mauvaises herbes, et que la SVM est une méthode de classification plus précise que l'ANN dans ce cas. Elle suggère que des recherches futures pourraient explorer d'autres algorithmes d'apprentissage automatique et des techniques d'extraction de caractéristiques supplémentaires, comme la couleur ou la texture, pour améliorer encore la

précision de la détection des mauvaises herbes. Ces résultats ont des applications pratiques dans le développement de robots de désherbage pour l'agriculture de précision, notamment dans la gestion automatisée des mauvaises herbes dans les champs de betteraves à sucre.

En conclusion, la littérature existante sur la détection des mauvaises herbes et le phénotypage des plantes met en lumière l'importance croissante des technologies basées sur la vision par ordinateur et l'intelligence artificielle pour relever les défis de l'agriculture moderne. Des plateformes innovantes ont été développées pour permettre une identification précise des mauvaises herbes et une gestion plus durable des cultures. Ces approches offrent des avantages significatifs, notamment la réduction de l'utilisation d'herbicides et une meilleure productivité agricole. Toutefois, malgré les progrès réalisés, certains défis persistent, notamment en ce qui concerne la robustesse des modèles dans des environnements complexes et la gestion de grandes quantités de données visuelles.

Les différentes méthodes analysées dans cette revue fournissent une base solide pour le développement de nouveaux outils, tels que les systèmes automatisés de détection des mauvaises herbes. Cependant, il apparaît clairement qu'une intégration plus poussée des techniques d'apprentissage profond et d'intelligence artificielle permettra de surmonter certains des obstacles identifiés et d'améliorer encore la précision et l'efficacité de ces systèmes.

Dans le prochain chapitre, nous expliquerons en détail les outils et les méthodologies utilisés pour concevoir et développer notre modèle de détection des mauvaises herbes. Un aperçu des choix technologiques, des architectures de modèles, ainsi que des ensembles de données utilisés dans cette étude sera présenté.

Chapitre 3

Méthodologie

Dans ce chapitre, nous détaillons la méthodologie et les outils utilisés pour concevoir et développer notre modèle de détection des mauvaises herbes. Un environnement de développement solide est essentiel pour implémenter et affiner les différents modèles présentés dans cette recherche.

Nous aborderons les choix technologiques effectués, en particulier l'utilisation des réseaux de neurones convolutifs (CNN) ainsi que des modèles de détection et de segmentation, tels que YOLO et Mask R-CNN. Les CNN sont particulièrement adaptés pour extraire des caractéristiques visuelles complexes des images, permettant de distinguer avec précision les mauvaises herbes des cultures saines. YOLO, avec ses capacités de détection en temps réel, nous permet d'identifier et de localiser rapidement les mauvaises herbes dans les images. Mask R-CNN, de son côté, est utilisé pour la segmentation détaillée des mauvaises herbes, offrant une délimitation précise des contours des mauvaises herbes, essentielle pour une analyse approfondie dans un contexte agricole.

Pour faciliter l'implémentation de modèles aussi complexes, nous avons utilisé Python comme principal langage de programmation, en nous appuyant sur des bibliothèques populaires telles que TensorFlow et Keras. Ces outils simplifient le développement et l'entraînement de modèles d'apprentissage automatique. Cette combinaison nous a permis de prototyper rapidement et d'expérimenter avec différentes architectures et hyperparamètres, en concentrant nos efforts sur l'optimisation des performances du modèle.

Ce chapitre met en avant l'importance des outils sélectionnés — CNN, YOLO, Mask R-CNN et Python — qui ont été essentiels pour le succès de notre système de détection des mauvaises herbes. La synergie de ces technologies permet d'offrir une approche complète pour relever les défis de la détection et de la segmentation des mauvaises herbes. Grâce à ces outils, nous pouvons améliorer la précision et l'efficacité du modèle, contribuant ainsi à renforcer l'efficacité globale de notre système de gestion des mauvaises herbes.

3.1 Architecture CNN - Réseau de Neurones Convolutifs

Pour l'extraction des caractéristiques des images, nous avons opté pour le modèle ResNet50V2 [14] , un réseau de neurones convolutifs (CNN) pré-entraîné sur ImageNet. Ce modèle se distingue par sa profondeur et l'utilisation de connexions résiduelles, qui permettent de mieux capter des caractéristiques complexes tout en atténuant les problèmes liés à la dégradation du gradient.

Dans un CNN, les données d'entrée traversent plusieurs couches conçues pour extraire progressivement des caractéristiques de plus en plus abstraites avant d'aboutir à la classification. Parmi ces couches, on retrouve la couche de convolution, qui applique des filtres pour détecter des motifs spécifiques, et la couche de pooling, qui réduit la dimensionnalité en sous-échantillonnant les données. Après plusieurs passages par ces couches, la sortie est aplatie et transmise à des couches entièrement connectées, responsables de la classification ou de la régression sur les caractéristiques extraites comme montré dans la figure 3.1 [15].

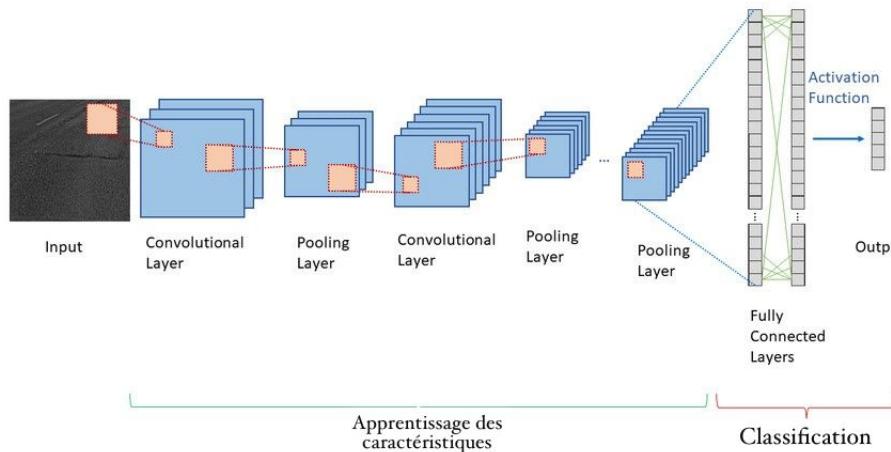


FIGURE 3.1 – Architecture de modèle CNN

3.1.1 L'apprentissage des caractéristiques

L'apprentissage des caractéristiques repose sur des techniques permettant d'extraire automatiquement des informations pertinentes à partir des images d'entrée. Dans un CNN, cette extraction se fait principalement à travers deux éléments clés : la couche de convolution, qui applique des filtres pour détecter des motifs spécifiques, et la couche de pooling, qui réduit la dimensionnalité des données tout en conservant les caractéristiques essentielles.

La couche de convolution

La couche de convolution joue un rôle clé dans l'extraction des caractéristiques en appliquant successivement une fonction de convolution et une fonction d'activation. Plusieurs couches de convolution peuvent être utilisées pour capturer différents niveaux de

caractéristiques.

Lors de l'opération de convolution, une fonction linéaire, appelée noyau ou filtre, est appliquée pour extraire des motifs spécifiques. Ce filtre se déplace sur l'image de gauche à droite, effectuant une multiplication élément par élément entre une région de l'image I et le noyau K , avant de sommer les résultats obtenus. Le stride (pas) définit la distance du déplacement du filtre à chaque étape, influençant ainsi la résolution des caractéristiques extraites.

$$F[i, j] = \langle I * K \rangle_{[i, j]} . \quad (3.1)$$

Voici un exemple dans lequel une image de dimension $4 \times 4 \times 1$ est convoluée avec un noyau de dimension $2 \times 2 \times 1$ et un stride =1, $s=1$ (voir figure 3.2)[16].

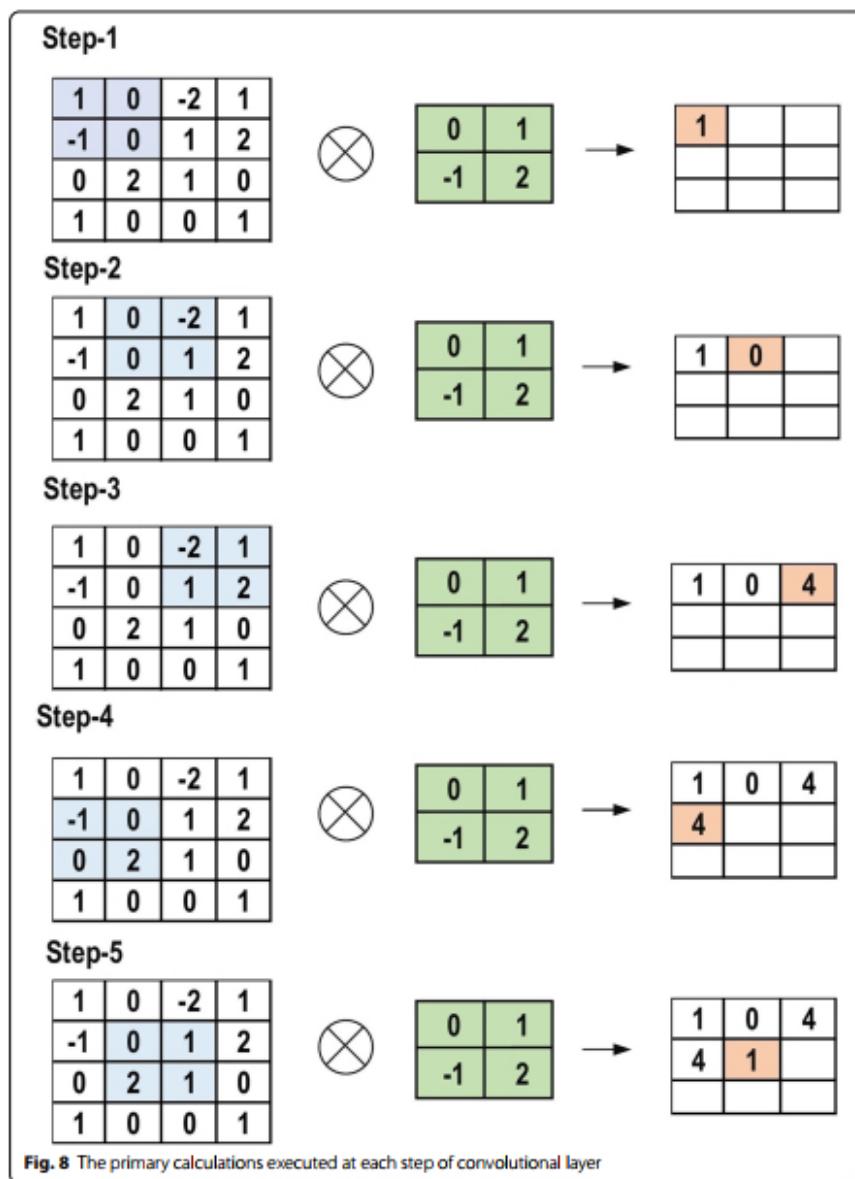


FIGURE 3.2 – Convolution avec un kernel de $2 \times 2 \times 1$ et stride $s = 1$

L'entrée (i,j)-ème de la carte des caractéristiques est donnée par la formule générale suivante dans le cas d'un seul canal :

$$F[i, j] = \langle I * K \rangle_{[i, j]} = \sum_x^{m_1} \sum_y^{m_2} K_{[x, y]} I_{[i-x, j-y]}. \quad (3.2)$$

où :

- X, Y : Coordonnées des éléments dans le noyau K lors du processus de convolution
- m_1, m_2 : Dimensions du noyau, où m_1 est la hauteur et m_2 est la largeur du noyau K
- $Stride(s)$: Le stride (pas) définit la taille du déplacement du noyau sur l'image

Ce processus est répété en appliquant plusieurs filtres, chacun capturant des caractéristiques spécifiques de l'image, comme les contours, les textures ou les variations de luminosité. Le nombre de filtres utilisés peut être supérieur à un, permettant d'extraire une diversité d'informations. Le stride (pas) joue un rôle clé en définissant la manière dont ces filtres se déplacent sur l'image, influençant ainsi la granularité des caractéristiques détectées.

- Le remplissage (ou padding).

Un inconvénient de cette procédure est que les filtres ont tendance à accorder plus d'importance au centre de l'image qu'à ses bords. Pour remédier à cela, on utilise le padding, une technique qui consiste à ajouter des valeurs autour de l'image avant d'appliquer la convolution. Par convention, le zéro padding est couramment utilisé, où des lignes et colonnes de zéros sont ajoutées autour du tenseur d'entrée, permettant ainsi de préserver les dimensions et de mieux capturer les caractéristiques situées aux extrémités de l'image.

- Fonction d'activation.

Les fonctions d'activation sont essentielles dans les réseaux de neurones, y compris les CNN, car elles introduisent une non-linéarité, permettant au modèle de capturer des relations complexes dans les données. Parmi les différentes fonctions d'activation, la plus couramment utilisée est la ReLU (Rectified Linear Unit), qui supprime les valeurs négatives en les remplaçant par zéro, améliorant ainsi la convergence du réseau et atténuant le problème du gradient évanescents (vanishing gradient). Ce phénomène survient lorsque les gradients deviennent extrêmement faibles au fil des couches, rendant l'apprentissage difficile, en particulier dans les réseaux profonds. En empêchant la propagation de gradients proches de zéro, ReLU contribue à un apprentissage plus efficace des paramètres du réseau.

$$R(x) = \max(0, x). \quad (3.3)$$

La couche de sous-échantillonnage

La couche de sous-échantillonnage réduit la taille spatiale des caractéristiques extraites par la convolution, permettant ainsi de conserver les informations les plus pertinentes tout en diminuant la complexité du modèle. Elle applique une fonction de pooling sur les données issues de la couche de convolution. Différentes méthodes de pooling existent,

notamment le sum pooling, le average pooling et le max pooling, ce dernier étant le plus utilisé, car il sélectionne les valeurs maximales dans une région donnée, mettant en avant les caractéristiques les plus importantes de l'image [16].

3.1.2 Classification

Ainsi, plusieurs couches cachées, comprenant des couches de convolution et de sous-échantillonnage, sont utilisées pour extraire les caractéristiques. Une fois cette extraction terminée, les résultats sont transformés en un vecteur unique (voir Figure 3.3)[17]. Ce vecteur est ensuite passé à travers une couche entièrement connectée, où la classification des données est effectuée en fonction des caractéristiques extraites.

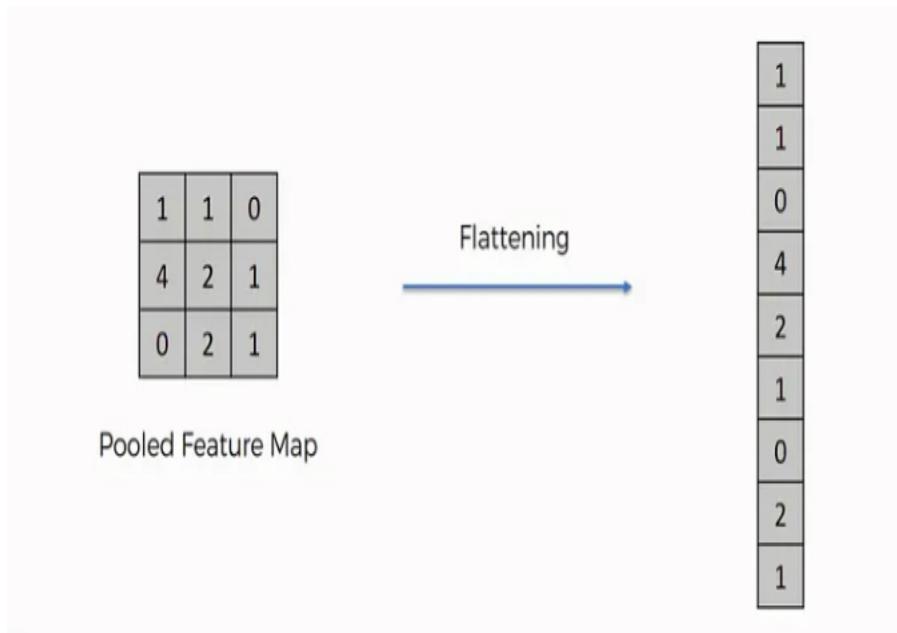


FIGURE 3.3 – Processus de mise à plat.

— La couche entièrement connectée

La couche entièrement connectée prend le vecteur aplati en entrée et le transforme en un autre vecteur, qui représente les sorties du modèle. Dans certains modèles d'apprentissage automatique, il est possible qu'une classe soit plus fréquente que l'autre, ce qui peut entraîner un déséquilibre. Pour résoudre ce problème, des poids équilibrés sont appliqués, combinés à la partie de pooling. Un terme de biais est également ajouté, suivi de l'application d'une fonction d'activation pour garantir une meilleure gestion de ce déséquilibre.

La description mathématique est la suivante :

$$X = \sum_{i=1}^n (w_i P_i + b'),$$

où :

- X est la sortie du neurone.
- w_i est le poids associé à l'entrée P_i .
- P_i est l'entrée du neurone, qui peut être un pixel, une valeur ou une caractéristique selon le contexte.
- b' est le biais du neurone, qui permet de décaler la fonction d'activation.
- L'indice i varie de 1 à n , où n est le nombre d'entrées du neurone.

Ensuite, la sortie X est passée à travers une fonction d'activation g , pour donner z :

$$z = g(X), \quad (3.4)$$

où :

- $g(x)$ est la fonction d'activation, qui peut être ReLU, sigmoïde, tanh, etc. Cette fonction prend la sortie X et la transforme de manière non linéaire.
- z est la sortie activée du neurone.

Domaines de définition :

- P_i est un élément du vecteur d'entrée, donc sa plage de variation dépend du type de données traitées (par exemple, des pixels d'image avec des valeurs dans $[0, 255]$ ou des caractéristiques normalisées entre $[0, 1]$).
- Les poids w_i sont des paramètres ajustés par l'apprentissage du réseau de neurones et peuvent être réels (souvent initialisés aléatoirement).
- Le biais b' est également un paramètre ajusté pendant l'apprentissage et peut être un réel.

Plage de variation des indices :

- Les indices i varient de 1 à n , où n est le nombre total d'entrées. Par exemple, dans un réseau de neurones avec 3 entrées, i prendra les valeurs 1, 2, 3.

Ici, g représente une fonction d'activation appliquée à la couche entièrement connectée. À chaque couche, des poids sont ajoutés aux sorties de la couche de pooling, et la fonction d'activation est ensuite appliquée. Plusieurs couches cachées sont utilisées dans le réseau, et la dernière couche applique une fonction d'activation qui permet d'effectuer la classification en calculant la probabilité associée à chaque classe, permettant ainsi de déterminer la classe la plus probable.

Le résumé est présenté ci-dessous (voir Figure 3.1) :

- L'image d'entrée est passée à travers une série de couches convolutionnelles qui extraient les caractéristiques en utilisant des filtres.
- La sortie de chaque couche est passée à travers une fonction d'activation ReLU (Rectified Linear Unit) pour introduire de la non-linéarité.
- Les couches de pooling sont utilisées pour réduire la dimensionnalité de la sortie et rendre le modèle plus efficace.
- Les couches convolutionnelles et de pooling peuvent être répétées plusieurs fois, en fonction de la complexité du problème.

- La sortie de la dernière couche convolutionnelle est aplatie et alimentée dans une couche entièrement connectée (FC Layer) qui effectue la classification ou la régression sur les caractéristiques extraites.
- La sortie de la couche FC est passée à travers une fonction d'activation telle que softmax ou régression logistique avec des fonctions de coût pour produire une étiquette de classe pour l'image d'entrée.

3.2 YOLO (You Only Look Once)

YOLOv5 est un modèle de détection d'objets basé sur une architecture de réseau convolutif entièrement optimisé pour la rapidité et l'efficacité. Il repose sur le principe de diviser une image en une grille et de prédire directement les classes et les emplacements des objets à partir des cellules de cette grille. YOLOv5 est particulièrement apprécié pour sa vitesse en temps réel et son précision robuste.

3.2.1 Caractéristiques principales :

- Détection rapide grâce à une seule passe à travers le réseau.
- Réduction de la latence, ce qui le rend adapté aux applications en temps réel.
- Prise en charge de l'optimisation multi-plateformes (CPU, GPU, edge devices).

3.2.2 Schéma de YOLO :

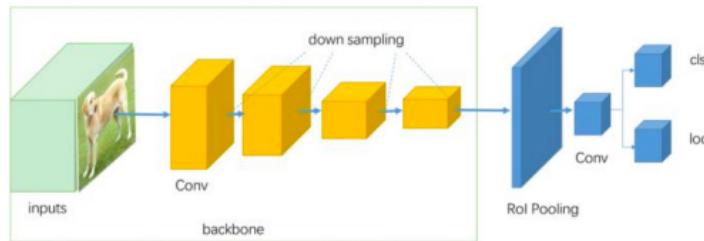


FIGURE 3.4 – Schéma de YOLO

Le schéma de YOLO, comme montré dans la figure 3.4 [18] , explique en détail les étapes de détection des objets à travers le réseau :

- Une image d'entrée passe par un backbone pour extraire des caractéristiques de bas niveau (textures, contours) et de haut niveau (objets).
- L'image est divisée en une grille (SxS) (par exemple, une grille 7x7). Chaque cellule de la grille est responsable de prédire les boîtes englobantes, score de confiance et les classes d'objets
- Ces caractéristiques sont envoyées à un "head" pour prédire les boîtes englobantes, les classes et les scores de confiance.

3.3 Mask R-CNN

Mask R-CNN [19] est une extension du modèle Faster R-CNN [20], ajoutant la capacité de segmentation sémantique pour fournir des masques d'objets en plus des boîtes englobantes. Ce modèle est utilisé dans des applications où une compréhension précise des contours des objets est nécessaire.

3.3.1 Caractéristiques principales :

- Détection d'objets avec des masques précis.
- Utilisation d'une architecture en deux étapes : détection des régions d'intérêt (RoI) suivie d'une segmentation fine.
- Approprié pour des tâches nécessitant à la fois des détections précises et une segmentation, comme la vision robotique ou l'analyse médicale.

3.3.2 Schéma de Mask R-CNN :

Mask R-CNN = Faster R-CNN + FCN on ROIs

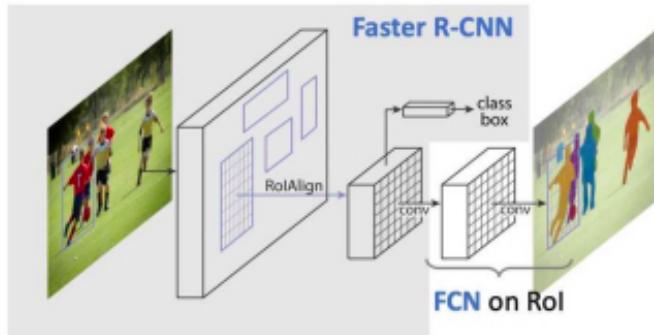


FIGURE 3.5 – Schéma de Mask R-CNN

Le schéma de Mask R-CNN, comme montré dans la figure Figure 3.5 [21], explique en détail les étapes de détection des objets et de segmentation d'instances à travers le réseau. Il montre comment le modèle génère des masques pour chaque objet détecté en plus des prédictions de boîtes englobantes et de classes, en utilisant un réseau de segmentation parallèle pour améliorer la précision des contours des objets :

- Une image d'entrée passe par un backbone (souvent ResNet ou ResNeXt)[14].
- Une Region Proposal Network (RPN) génère des propositions de régions d'intérêt(RoI) qui contiennent probablement des objets. Ces régions sont ensuite raffinées pour éliminer les faux positifs.
- Contrairement à Faster R-CNN, Mask R-CNN utilise RoIAlign au lieu de RoIPool. RoIAlign ajuste précisément les régions d'intérêt à la grille de caractéristiques,

améliorant ainsi la précision des masques générés.

- Chaque région d'intérêt est Classifiée et localisée par une boîte englobante qui délimite l'objet détecté
- Le modèle ajoute un troisième composant : une Fully Convolutional Network (FCN)
- appliquée aux régions d'intérêt. Cela permet de prédire des masques pixel-par-pixel pour chaque objet détecté.

3.4 Environnement de développement et outils

Dans cette section, nous mettrons en évidence l'importance des outils choisis — CNN, YOLO, Mask R-CNN, et Python — dans le développement réussi de notre modèle de détection des mauvaises herbes. La synergie de ces technologies joue un rôle clé pour aborder efficacement les défis liés à la détection des mauvaises herbes dans des environnements agricoles variés à travers des techniques avancées d'apprentissage automatique.

Dans ce travail, Python a été utilisé comme principal langage de programmation, offrant un accès à une vaste bibliothèque de ressources et de frameworks spécifiquement conçus pour l'analyse de données et l'apprentissage automatique, tels que TensorFlow et Keras. Ces bibliothèques fournissent des fonctions pré-construites et des outils qui simplifient le processus de développement, nous permettant ainsi de nous concentrer sur l'architecture du modèle et l'optimisation des performances. Le choix de Python a été essentiel pour faciliter le développement des modèles de détection des mauvaises herbes, permettant de tester et d'affiner les architectures plus efficacement.

Pour la détection et la segmentation des mauvaises herbes, nous avons employé des réseaux de neurones convolutifs (CNN) ainsi que les modèles Mask R-CNN et YOLO. Les CNN sont particulièrement efficaces pour l'extraction des caractéristiques visuelles des images, telles que la forme et la texture des plantes. Mask R-CNN, quant à lui, est utilisé pour générer des masques précis autour des mauvaises herbes, permettant une segmentation fine, tandis que YOLO permet une détection rapide et en temps réel des objets, ce qui est crucial pour les applications agricoles automatisées.

L'intégration de ces outils a permis de créer un système robuste et flexible, capable de s'adapter aux divers types d'environnements et de conditions dans lesquels les mauvaises herbes se trouvent.

Dans les chapitres suivants, nous détaillerons les étapes spécifiques de la détection des mauvaises herbes, en commençant par l'utilisation du modèle ResNet50 sur le jeu de données DeepWeeds. Ce jeu de données, qui contient des images d'herbes classées, constitue une base idéale pour entraîner et évaluer un modèle de classification. Nous commencerons par décrire les caractéristiques du jeu de données DeepWeeds, ainsi que la logique de sa sélection en tant que référence pour notre tâche de détection. Une fois les données acquises, nous aborderons les étapes de prétraitement, telles que la normalisation, l'augmentation des données et la préparation des images, qui permettent de maximiser les performances du modèle ResNet50.

Le modèle ResNet50, qui est un réseau de neurones convolutifs pré-entraîné, est particulièrement adapté pour l'extraction de caractéristiques visuelles de haute qualité. Nous l'utilisons ici pour classifier les images des mauvaises herbes avec une grande précision, en tirant parti de sa capacité à apprendre des représentations profondes des données tout en évitant les problèmes de dégradation du gradient grâce à ses connexions résiduelles. Nous détaillerons la mise en œuvre de ce modèle, son entraînement et l'évaluation de ses performances sur le jeu de données DeepWeeds, ainsi que les résultats obtenus.

Ensuite, nous explorerons la transition vers un modèle général de détection des mauvaises herbes, en intégrant des techniques d'adaptation incrémentale. Ce modèle est conçu pour s'adapter aux nouvelles données sans nécessiter un réentraînement complet. L'adaptation incrémentale est particulièrement utile pour gérer l'évolution des conditions agricoles et des types de mauvaises herbes au fil du temps. Nous expliquerons comment nous avons implémenté cette approche, permettant à notre système de s'améliorer continuellement au fur et à mesure qu'il reçoit de nouvelles images étiquetées.

Un autre aspect clé de notre méthodologie est l'intégration de l'auto-étiquetage, une approche semi-supervisée qui permet d'étiqueter automatiquement les données non annotées à l'aide d'un modèle préexistant. Cette technique permet d'enrichir rapidement notre jeu de données avec des annotations supplémentaires, réduisant ainsi le besoin de collecter manuellement des étiquettes pour chaque nouvelle image. Nous détaillerons comment nous avons mis en place l'auto-étiquetage pour améliorer l'efficacité de l'entraînement et augmenter la taille du jeu de données sans intervention humaine.

Enfin, nous aborderons l'utilisation combinée de YOLO et Mask R-CNN pour la détection et la segmentation des mauvaises herbes. YOLO est un modèle de détection en temps réel, idéal pour localiser rapidement les mauvaises herbes dans les images, tandis que Mask R-CNN est utilisé pour générer des masques précis autour de ces mauvaises herbes, permettant une segmentation fine et détaillée. Nous expliquerons comment ces deux modèles ont été intégrés pour produire des résultats de segmentation de haute précision, avec un focus sur leur complémentarité dans la détection et la délimitation des mauvaises herbes dans les champs agricoles.

Ainsi, à travers cette méthodologie intégrée, nous visons à développer un système robuste et évolutif de détection des mauvaises herbes, capable de s'adapter aux différentes conditions agricoles tout en garantissant une précision optimale.

Chapitre 4

Détection avec ResNet50 sur deepWeeds

Ce chapitre présente l'approche suivie pour développer un modèle de détection des mauvaises herbes en utilisant le jeu de données DeepWeeds. Le modèle, basé sur l'architecture ResNet50, a été finement ajusté pour classer les images de 9 espèces différentes de mauvaises herbes. La performance du modèle a été évaluée en termes de précision, de rappel, de score F1 et à travers une matrice de confusion, atteignant une précision globale de 94,57 %.

4.1 Préparation des données

4.1.1 Chargement et prétraitement des données

Le jeu de données DeepWeeds comprend 17 509 images étiquetées de 9 espèces de mauvaises herbes capturées en Australie [22]. Chaque image est associée à une classe dans un fichier CSV. Pour ce projet, nous avons réparti le jeu de données en ensembles d'entraînement (80 %) et de validation (20 %).

Avant d'entamer l'entraînement, il est essentiel d'analyser la distribution des classes d'espèces de mauvaises herbes pour détecter d'éventuels déséquilibres. Cela mène à observer s'il y a certaines classes dominées par une espèce commune et qui ont un nombre nettement plus élevé d'images que d'autres. Dans ce cas, la distribution déséquilibrée pourrait influer sur la performance du modèle en faveur des classes majoritaires. Ce déséquilibre doit donc être pris en compte dans l'entraînement et l'évaluation du modèle.

Dans le jeu de données DeepWeeds, chaque classe du jeu de données correspond à une espèce unique de mauvaise herbe, avec un label numérique allant de 0 à 8. Ces labels permettent d'identifier les espèces suivantes (voir Tableau 4.1) :

Ces labels servent d'identifiants pour les classes lors de l'entraînement et facilitent l'interprétation des prédictions du modèle.

TABLE 4.1 – Espèces et leurs étiquettes correspondantes.

| Species | Label |
|----------------|-------|
| Chinee apple | 0 |
| Lantana | 1 |
| Parkinsonia | 2 |
| Parthenium | 3 |
| Prickly acacia | 4 |
| Rubber vine | 5 |
| Siam weed | 6 |
| Snake weed | 7 |
| Negative | 8 |

Pour illustrer visuellement les différentes espèces de mauvaises herbes présentes dans le jeu de données DeepWeeds, nous avons créé une Figure 4.1, contenant un échantillon d'images pour chaque classe. Cette figure permet de mieux comprendre la diversité des espèces et les défis associés à leur classification automatique. Chaque espèce présente des caractéristiques visuelles distinctes, comme des différences de forme, de couleur et de texture, rendant leur identification complexe, notamment dans un environnement naturel.



FIGURE 4.1 – Échantillons visuels des différentes espèces de mauvaises herbes du jeu de données DeepWeeds.

La Figure 4.2 montre une répartition déséquilibrée des classes, où la catégorie "Negative" est largement dominante par rapport aux huit espèces de mauvaises herbes. Cependant, cette disparité s'explique par la nature du problème : la classe "Negative" regroupe des zones sans mauvaises herbes ou contenant des plantes non nuisibles, tandis que les huit autres classes représentent des espèces spécifiques de mauvaises herbes.

Si l'on considère ces huit espèces comme une seule catégorie, la distribution devient plus équilibrée entre les "Mauvaises herbes" et les "Non-mauvaises herbes". Cette structuration réduit le risque de biais lors de l'apprentissage et assure une représentation équitable des cas d'entraînement.

L'expérimentation a montré que la réduction de la classe "Other" entraînait une baisse des performances du modèle, contrairement à son maintien, qui permettait une meilleure généralisation. Cela s'explique par le fait que la classe "Other" contient une grande diver-

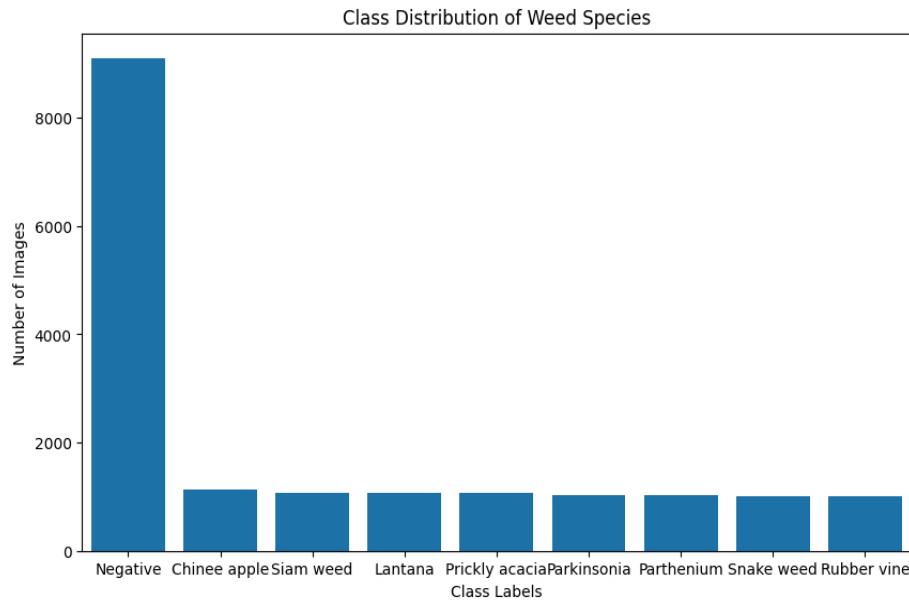


FIGURE 4.2 – Répartition des classes des espèces de mauvaises herbes dans le jeu de données DeepWeeds.

sité de plantes et de conditions environnementales, enrichissant ainsi l'apprentissage du modèle. En sous-échantillonnant cette classe, des informations précieuses sur les variations naturelles de l'arrière-plan et des espèces végétales étaient perdues, ce qui a réduit la capacité du modèle à discriminer efficacement les mauvaises herbes. Ainsi, afin de préserver la richesse des données et d'assurer une meilleure séparation entre les classes, nous avons opté pour l'utilisation de l'ensemble complet des images de "Other" tout en maintenant une répartition stratifiée des données pour éviter un déséquilibre excessif.

Ensuite, les images ont été redimensionnées à une taille de 224x224 pour s'adapter au modèle ResNet50 et normalisées afin d'optimiser l'entraînement. L'augmentation des données a également été appliquée aux images d'entraînement, avec des transformations comme les recadrages aléatoires et les retournements horizontaux.

4.1.2 Augmentation des données

L'augmentation des données est une étape essentielle en apprentissage profond pour améliorer la généralisation du modèle. Elle consiste à générer des versions légèrement modifiées des images originales afin d'enrichir le jeu de données sans avoir à collecter de nouvelles données. En introduisant des variations artificielles, l'augmentation aide à réduire les risques de surapprentissage (overfitting), ce qui permet au modèle de mieux se comporter sur des images inédites. Les techniques suivantes ont été utilisées :

Recadrage aléatoire (Random Cropping)

Cette technique consiste à découper une zone aléatoire de l'image d'origine pour créer une nouvelle version. Le recadrage permet de simuler des variations naturelles dans la com-

position de l'image, comme un changement d'angle ou un objet partiellement visible. Par exemple, certaines mauvaises herbes peuvent ne pas toujours être entièrement capturées dans le cadre, et le modèle doit apprendre à les reconnaître même avec des informations partielles. L'effet de cette approche est d'améliorer la capacité du modèle à identifier des objets incomplets ou partiellement masqués. Des recadrages couvrant une zone comprise entre 70 % et 100 % de l'image ont été appliqués.

Retournement horizontal (Horizontal Flipping)

Cette transformation consiste à inverser l'image de gauche à droite, ce qui permet de créer une diversité spatiale. Elle est particulièrement utile lorsque l'orientation de l'objet dans l'image n'a pas d'importance pour la classification, comme dans le cas des mauvaises herbes. L'effet de cette approche est de renforcer la robustesse du modèle en lui apprenant à reconnaître les mauvaises herbes quel que soit leur sens d'apparition. Par exemple, une image contenant la plante Siam Weed peut être retournée horizontalement, simulant ainsi un changement de perspective naturelle.

Rotation aléatoire (Random Rotation)

Les images peuvent être tournées d'un angle aléatoire, ce qui permet de simuler les variations naturelles d'orientation des plantes dans le champ, causées par le vent ou d'autres mouvements aléatoires. Cette technique améliore la capacité du modèle à détecter des plantes dans différentes orientations. Des rotations aléatoires de $\pm 30^\circ$ ont été appliquées aux images afin d'enrichir la diversité des données d'entraînement et de renforcer la capacité du modèle à reconnaître correctement les espèces, quel que soit leur angle d'apparition.

Zoom aléatoire (Random Zoom)

Le zoom aléatoire simule des variations de distance entre l'appareil photo et la plante en modifiant la taille des images, qui peuvent ainsi être agrandies ou réduites. Cette technique permet au modèle de reconnaître les mauvaises herbes sous différents niveaux de zoom, ce qui est crucial dans des scénarios réels où la distance de prise de vue peut varier. Un zoom aléatoire compris entre 90 % et 110 % de la taille originale a été appliqué, ce qui accroît la flexibilité du modèle et renforce sa robustesse face aux variations d'échelle.

Normalisation

La normalisation est une étape critique dans le prétraitement des données, car elle aligne les images sur les statistiques utilisées lors de l'entraînement du modèle ResNet50. Pour ce faire, les valeurs des pixels sont ramenées dans l'intervalle $[0, 1]$ en divisant chaque pixel par 255. Cette procédure garantit que les différentes caractéristiques des images, telles que la luminosité et le contraste, ne biaissent pas le processus d'entraînement.

L'effet de la normalisation est double : elle stabilise l'entraînement et accélère la convergence du modèle, ce qui permet d'atteindre des performances optimales plus rapidement. De plus, cette normalisation est cohérente avec les poids pré-entraînés de ResNet50, garantissant ainsi que les performances du modèle ne soient pas affectées par des écarts dans la distribution des pixels. En intégrant cette étape, on assure que le modèle puisse apprendre de manière efficace et précise, en se basant sur des représentations d'images standardisées.

En conclusion, l'utilisation combinée de ces différentes techniques d'augmentation permet de créer un jeu de données plus diversifié à partir de l'ensemble d'entraînement. En multipliant les scénarios potentiels dans lesquels une plante pourrait apparaître, nous réduisons le risque que le modèle se spécialise trop sur les images d'entraînement. L'augmentation des données garantit donc que le modèle soit capable de généraliser efficacement sur de nouvelles images, même dans des conditions légèrement différentes de celles du jeu de données d'origine.

4.2 Modèle de classification

4.2.1 Architecture du modèle

Le modèle que nous avons utilisé repose sur ResNet50, un réseau de neurones convolutionnel (CNN) profond et pré-entraîné sur ImageNet, un vaste ensemble de données contenant plus de 1,2 million d'images réparties sur 1 000 classes. ResNet50 est particulièrement efficace pour traiter des images complexes tout en réduisant le risque de surapprentissage grâce à sa conception par blocs résiduels. Ce choix se justifie par l'équilibre qu'il offre entre profondeur, performance et nombre de paramètres, rendant le modèle efficace et capable de s'adapter à une variété de tâches de classification.

L'architecture de ResNet50 comme le montre la figure Figure 4.3 [23] se compose de plusieurs couches de convolution, de normalisation par lots et de fonctions d'activation ReLU, qui permettent d'extraire des caractéristiques complexes des images. Cependant, afin de répondre aux exigences de notre projet, nous avons apporté certaines modifications à la structure de ce modèle, en particulier au niveau de la dernière couche qui a été remplacée par une couche dense adaptée au nombre de classes (9).

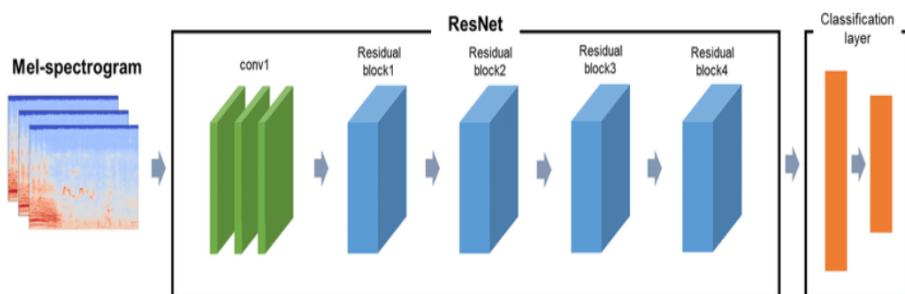


FIGURE 4.3 – Architecture de ResNet50.

- Couche d'entrée : Images RGB de taille 224x224.

- Couche convolutive : Blocs résiduels de ResNet50 pour l'extraction des caractéristiques.
- Couche de sortie : Une couche dense de 9 neurones avec une fonction d'activation softmax pour la classification multi-classes.

4.2.2 Optimisation

L'entraînement du modèle s'est déroulé sur 25 époques en utilisant l'optimiseur SGD (Stochastic Gradient Descent), une méthode classique et efficace pour l'apprentissage de réseaux de neurones profonds. Le choix de cet optimiseur vise à garantir une convergence stable et à minimiser la fonction de perte de manière progressive. Voici les différents aspects et paramètres d'optimisation adoptés :

SGD avec Momentum

L'optimiseur SGD inclut l'utilisation d'un momentum, qui permet d'accélérer l'apprentissage en orientant le modèle vers des gradients accumulés au fil des itérations. Cela aide à surmonter les oscillations, notamment dans des vallées de la fonction de coût, et à converger plus rapidement vers un minimum local. Le momentum aide également à réduire les risques de stagnation lorsque le modèle atteint un plateau.

La formule du gradient avec momentum est comme suit :

$$v_t = \beta v_{t-1} + (1 - \beta) \nabla L(\theta) \quad (4.1)$$

où :

- v_t est le gradient accumulé avec momentum.
- β est le facteur de momentum (généralement fixé à 0,9).
- $\nabla L(\theta)$ est le gradient de la perte par rapport aux poids θ , ajustés à chaque itération.

Taux d'apprentissage initial

Le taux d'apprentissage (learning rate) est fixé à 0,001 au début de l'entraînement, après plusieurs tests préliminaires pour identifier la valeur optimale. Ce paramètre est crucial pour contrôler la vitesse d'apprentissage du modèle : un taux trop élevé peut entraîner des mises à jour trop importantes des poids, ce qui risque de faire diverger le modèle et d'empêcher la convergence vers un minimum optimal. S'il est trop faible, la convergence devient lente, et l'entraînement peut se figer dans un minimum local.

Réduction progressive du taux d'apprentissage

Au fil des époques, le taux d'apprentissage est progressivement réduit à l'aide d'une technique d'ordonnanceur (scheduler). Cette approche permet d'ajuster dynamiquement le taux en fonction de la progression de l'entraînement, favorisant ainsi une optimisation fine en fin d'apprentissage. La stratégie adoptée consiste à réduire le taux d'apprentissage de x

% chaque fois que l'amélioration de la performance en validation stagnait pendant plusieurs époques consécutives. Après plusieurs tests, une réduction de 10 % a été appliquée dans ce cas. Cela permet de stabiliser la recherche du minimum optimal et d'éviter les fluctuations dans la perte.

Nombre d'époques

Le modèle a été entraîné pendant 25 époques, un choix qui résulte de plusieurs tests préliminaires pour trouver un nombre d'époques optimal. Après une série d'expérimentations avec différentes configurations d'époques, il a été observé que 25 époques offraient un bon compromis entre la convergence du modèle et la prévention du sur-apprentissage. Ce nombre d'époques a permis au modèle d'atteindre une performance stable sur l'ensemble de validation, tout en évitant une sur-optimisation des paramètres sur les données d'entraînement.

Chaque époque passe l'ensemble des données d'entraînement à travers le réseau, permettant au modèle de mettre à jour ses poids et de s'ajuster progressivement aux caractéristiques des images.

Fonction de perte

Pour ce modèle de classification multi-classes, nous avons utilisé la catégorial cross-entropy comme fonction de perte :

$$L = - \sum_{i=1}^n y_i \log(\hat{y}_i) \quad (4.2)$$

où y_i est la vraie classe, \hat{y}_i est la probabilité prédictive pour cette classe et n est le nombre total de classes possibles pour la classification. Cette fonction est bien adaptée aux problèmes de classification, car elle punit sévèrement les mauvaises prédictions, tout en maximisant la probabilité des classes correctes. Ces paramètres d'optimisation assurent un apprentissage fluide et robuste, garantissant que le modèle converge efficacement vers une solution optimale pour la classification des espèces de mauvaises herbes.

4.3 Historique de l'entraînement

L'historique de l'entraînement fournit une vue détaillée de la performance du modèle au fil des époques. Il permet d'observer l'évolution de la précision (accuracy) et de la fonction de perte (loss), tant sur les ensembles d'entraînement que de validation. La précision (accuracy) est une métrique qui mesure la proportion d'exemples correctement classés par rapport au nombre total d'exemples. Elle est définie comme suit :

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total d'exemples}}. \quad (4.3)$$

La fonction de perte (loss), quant à elle, mesure l'erreur entre les prédictions du modèle et les vraies valeurs cibles. Elle sert d'indicateur pour optimiser le modèle pendant l'entraînement. Une fonction de perte couramment utilisée dans les modèles de classification est l'entropie croisée, qui évalue la différence entre les distributions de probabilités prédites et réelles.

$$\text{Loss} = - \sum_{i=1}^n y_i \log(p_i) \quad (4.4)$$

où n représente le nombre total de classes, i est l'indice parcourant ces classes, y_i est une variable binaire indiquant si l'exemple appartient à la classe i (1 si oui, 0 sinon), et p_i est la probabilité prédite pour la classe i . Le terme $\log(p_i)$ permet de pénaliser les mauvaises prédictions en amplifiant la perte lorsque la probabilité associée à la classe correcte est faible.

Ces métriques sont essentielles pour évaluer l'efficacité du modèle et détecter d'éventuels problèmes, tels que le **surapprentissage (overfitting)** ou le **sous-apprentissage (underfitting)**. En visualisant ces courbes, nous avons pu ajuster les hyperparamètres et optimiser le comportement du modèle. L'optimalité a été déterminée en fonction de plusieurs critères : une faible fonction de perte et une précision élevée sur l'ensemble de validation, ainsi qu'une absence de divergences importantes entre les courbes d'entraînement et de validation, indiquant une bonne généralisation du modèle. Ces ajustements ont permis d'atteindre une performance optimale sur les données de test.

4.3.1 Analyse des résultats d'entraînement

L'évolution des courbes de la perte (Loss) et de la précision (Accuracy) à travers les 25 époques, comme le montrent la Figure 4.4, illustre une progression cohérente dans l'apprentissage du modèle. La perte d'entraînement diminue progressivement, accompagnée d'une amélioration de la précision. La perte de validation suit une tendance similaire, reflétant ainsi une généralisation efficace du modèle aux données non vues.

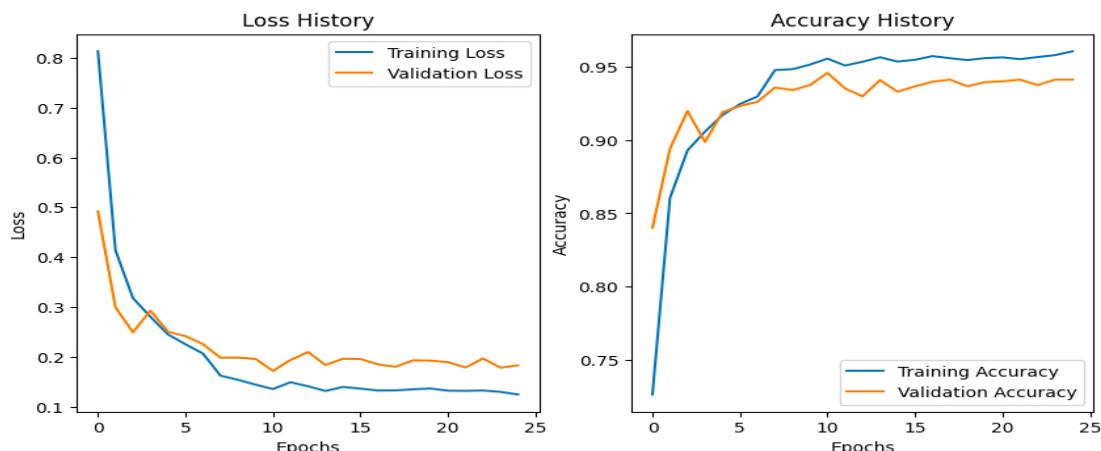


FIGURE 4.4 – Courbes de la perte (Loss) et de la précision (Accuracy).

4.3.2 Diminution de la perte et convergence

Lors des premières époques, la perte d'entraînement est relativement élevée (0.8131) avec une précision initiale de 72.64 %. Cependant, elle décroît rapidement pour atteindre 0.2065 à l'époque 6, avec une précision correspondante de 92.97 %. De manière parallèle, la perte de validation passe de 0.4920 à 0.2256, ce qui confirme la capacité du modèle à bien généraliser sur les données de validation.

À partir de l'époque 10, une stabilisation des pertes est observée, suggérant que le modèle atteint un plateau de convergence où les améliorations deviennent marginales. Les fluctuations de la perte de validation restent légères, oscillant entre 0.178 et 0.196 sur les dernières époques, ce qui montre une stabilité des performances.

4.3.3 Interprétation des courbes de précision et de perte

Les courbes montrent une cohérence entre la perte et la précision : lorsque la perte diminue, la précision augmente, indiquant que le modèle apprend efficacement sans signes de surajustement. Par exemple, à l'époque 24, la perte d'entraînement est de 0.1245 avec une précision de 96.05 %, tandis que la perte de validation est de 0.1831 avec une précision stable de 94.12 %.

4.3.4 Impact de l'optimisation avec SGD

L'utilisation de l'optimiseur SGD avec un taux d'apprentissage initial de 0,001 a permis une progression régulière. Le taux d'apprentissage décroissant a évité les oscillations autour du minimum global, garantissant une convergence correcte. Bien que la perte diminue plus lentement après l'époque 10, cela est attendu et montre une stabilisation normale des performances.

En conclusion, les courbes de perte et de précision confirment que le modèle a appris efficacement tout en maintenant une bonne généralisation sur les données de validation. Avec une précision finale de 94.12 % sur l'ensemble de validation et une perte stabilisée, le modèle démontre une robustesse dans la détection.

4.4 Évaluation des performances

4.4.1 Précision globale

Le modèle a atteint une précision globale de 94,57 % sur l'ensemble de validation, ce qui montre qu'il est capable de bien classer les images des différentes espèces de mauvaises herbes.

4.4.2 Rapport de classification

Le rapport de classification, comme affiché ci-dessous dans la figure 4.5, présente les scores de précision, rappel et F1-score pour chaque classe. La précision (precision) est la

proportion de prédictions positives correctes parmi toutes les prédictions positives faites par le modèle. Elle est définie comme suit :

$$\text{Précision} = \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux positifs}} \quad (4.5)$$

Le rappel (recall), également appelé sensibilité ou taux de vrais positifs, est la proportion de vrais positifs parmi toutes les instances positives réelles. Il est défini par :

$$\text{Rappel} = \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux négatifs}} \quad (4.6)$$

Le F1-score est une mesure harmonique de la précision et du rappel, offrant un compromis entre les deux. Il est particulièrement utile lorsque les classes sont déséquilibrées. Le F1-score est défini par :

$$\text{F1-score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \quad (4.7)$$

Le taux d'exactitude est défini comme la proportion de prédictions correctes parmi toutes les instances et est calculé avec une formule.

$$\text{Taux d'exactitude} = \frac{\text{Vrais positifs} + \text{Vrais négatifs}}{\text{Total des instances}} \quad (4.8)$$

Cela permet de mieux comprendre les performances du modèle pour chaque espèce de mauvaises herbes, en équilibrant l'importance de la précision et du rappel, surtout dans les cas où une classe est plus difficile à prédire que les autres.

| Overall Accuracy: 0.9457 | | | | |
|--------------------------|-----------|--------|----------|---------|
| | precision | recall | f1-score | support |
| 0 | 0.90 | 0.84 | 0.87 | 225 |
| 1 | 0.89 | 0.99 | 0.94 | 213 |
| 2 | 0.96 | 0.97 | 0.96 | 206 |
| 3 | 0.98 | 0.93 | 0.95 | 204 |
| 4 | 0.89 | 0.95 | 0.92 | 213 |
| 5 | 0.97 | 0.95 | 0.96 | 202 |
| 6 | 0.96 | 0.92 | 0.94 | 215 |
| 7 | 0.81 | 0.91 | 0.86 | 203 |
| 8 | 0.97 | 0.96 | 0.97 | 1821 |
| accuracy | | | | 0.95 |
| macro avg | | 0.93 | 0.93 | 3502 |
| weighted avg | | 0.95 | 0.95 | 3502 |

FIGURE 4.5 – Rapport de classification.

Les résultats montrent que le modèle a atteint des f1-scores élevés pour toutes les classes, avec des performances particulièrement bonnes pour les classes 2, 5, et 8.

4.4.3 Matrice de confusion

La matrice de confusion suivante (Figure 4.6) permet de visualiser les erreurs de classification. Elle montre comment le modèle classe les différentes espèces et les cas où des confusions entre classes apparaissent.

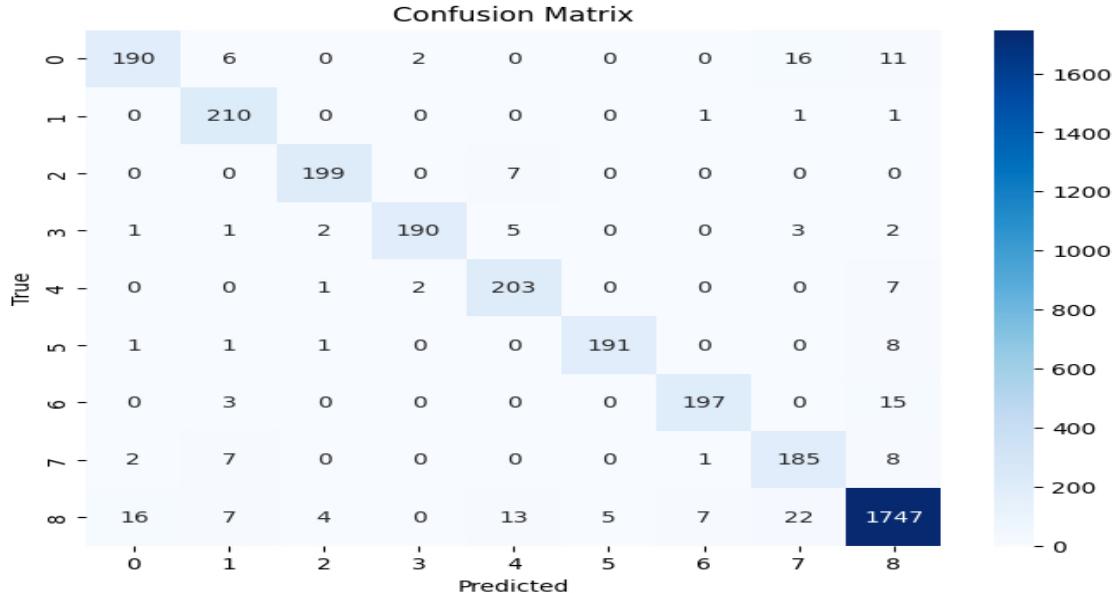


FIGURE 4.6 – Matrice de confusion.

Les images mal classées, appartenant aux classes 0 à 7 et identifiées comme des mauvaises herbes, présentent des valeurs d'intensité faibles par rapport à l'échantillon moyen.

4.4.4 Analyse des performances d'autres modèles de classification

Dans cette section, nous comparons les performances de plusieurs modèles de classification populaires, notamment ResNet50, ResNet18 [24]., DenseNet121 [25]., VGG [26], et MobileNet [27]. L'objectif est d'évaluer leur capacité à généraliser sur les données de test après l'entraînement. Les résultats sont présentés sous forme de taux d'exactitude pour les ensembles d'entraînement et de test, afin d'identifier les modèles les plus adaptés et d'analyser les éventuels problèmes de surapprentissage ou de sous-apprentissage.

TABLE 4.2 – Performances des différents modèles de classification.

| Modèle | Train Accuracy (%) | Test Accuracy (%) |
|-------------|--------------------|-------------------|
| ResNet50 | 96.05 | 94.57 |
| ResNet18 | 97.90 | 88.38 |
| DenseNet121 | 96.00 | 87.64 |
| VGG | 64.70 | 50.00 |
| MobileNet | 53.10 | 51.00 |

Les résultats montrent que ResNet50 offre un bon équilibre entre l'entraînement et le test, avec une précision de 96,05% et 94,57%, respectivement, suggérant une bonne

généralisation. ResNet18, bien que performant en entraînement (97,90%), présente une baisse notable sur le test (88,38%), indiquant un possible surapprentissage. DenseNet121 suit une tendance similaire avec des performances proches en entraînement (96,00%) mais une précision inférieure en test (87,64%). En revanche, VGG et MobileNet affichent des taux de précision de test faibles (50% et 51%), ce qui suggère que ces modèles sont moins adaptés aux données utilisées et requièrent des ajustements supplémentaires.

4.5 Conclusion

Dans ce chapitre, nous avons présenté le processus d'entraînement et d'évaluation d'un modèle de détection des mauvaises herbes à l'aide de ResNet50 sur le jeu de données DeepWeeds. Le modèle a obtenu une précision élevée de 94,57 % et des f1-scores élevés pour chaque espèce de mauvaises herbes. Ces résultats montrent que le modèle est bien adapté pour la classification d'espèces végétales dans des environnements complexes et pourrait être utilisé pour des applications automatisées dans le domaine de l'agriculture.

Chapitre 5

Modèle incrémental de détection des mauvaises herbes

La détection automatisée des mauvaises herbes est un enjeu majeur pour améliorer la gestion des cultures agricoles et optimiser l'usage de produits phytosanitaires. L'objectif de cette recherche est de développer un modèle de classification robuste et incrémental, capable de distinguer les mauvaises herbes des plantes cultivées sur plusieurs jeux de données.

Cette approche repose sur une simplification du problème initial en regroupant toutes les espèces indésirables sous une seule classe : mauvaises herbes (weed). Le modèle suit une logique incrémentale, où chaque version s'appuie sur les performances et connaissances des précédentes, améliorant ainsi sa précision et sa capacité de généralisation sur de nouveaux ensembles de données.

Le processus incrémental se déroule en plusieurs phases. Après un premier entraînement, le modèle est affiné sur de nouveaux jeux de données en intégrant à chaque étape une fraction des datasets antérieurs. Cette approche permet d'adapter le modèle à différents environnements tout en conservant les connaissances acquises. L'utilisation de cette méthodologie optimise également les ressources, puisque le modèle ajuste ses paramètres aux nouvelles données sans avoir à repartir de zéro.

Le modèle final a été entraîné et testé sur plusieurs ensembles de données en appliquant cette stratégie de fusion et d'adaptation. Chaque itération a abouti à une précision remarquable, et le modèle a démontré une performance optimale lorsqu'il a été appliqué à des données inédites, confirmant son efficacité et sa robustesse.

5.1 Description des autres jeux de données utilisés dans le modèle incrémental de détection des mauvaises herbes

Dans le cadre de cette étude, en plus du jeu de données DeepWeeds, deux autres jeux de données ont été intégrés pour enrichir l'entraînement et la validation du modèle. Le premier, crop-and-weed-detection-data-with-bounding-boxes [28], contient 1 300 images

de cultures de sésame et de différentes espèces de mauvaises herbes, chaque image étant étiquetée selon le format YOLO. Ce jeu de données a été constitué à partir de 589 images initiales, qui ont été soigneusement nettoyées pour éliminer les photos de mauvaise qualité, réduisant ainsi le total à 546 images utilisables. Les images ont ensuite été redimensionnées à une taille de 512 x 512 pixels pour faciliter le traitement et l'entraînement du modèle. Pour atteindre les 1 300 images, une technique d'augmentation des données a été appliquée, permettant ainsi de générer des variations à partir des images d'origine.

Le second jeu de données, weed-datasets [29], est issu d'une recherche antérieure et comprend des sous-ensembles spécifiques tels que le jeu de données des mauvaises herbes dans les cultures de maïs, de laitue et de radis. Les images du jeu de données de maïs ont été prises dans un champ de semis de maïs en utilisant un appareil photo Canon PowerShot, avec un total de 1 200 images de diverses mauvaises herbes. De même, des images de laitue ont été collectées dans une base de légumes en Chine, tandis que le jeu de données de radis a été constitué à partir d'images collectées dans des sources ouvertes, après avoir été recadrées pour obtenir 200 images de semis de radis et leurs mauvaises herbes associées.

5.2 Modèle initial de détection incrémentale des mauvaises herbes : trained_model.pth

Le modèle initial repose sur un réseau de neurones convolutionnel (CNN) et a été entraîné avec le dataset initial DeepWeeds, en ne tenant compte que de deux classes : mauvaises herbes (weed) et plantes non nuisibles (Negative). Cette simplification optimise l'entraînement et facilite la généralisation future du modèle à d'autres types de données et jeux de données qui ne contiennent pas nécessairement les mêmes classes que DeepWeeds.

Pour mieux comprendre la répartition des données, une représentation graphique a été générée pour visualiser le nombre d'images dans chaque classe. La figure Figure 5.1 présente un histogramme à deux barres : une pour les mauvaises herbes et l'autre pour les plantes non nuisibles. Cette visualisation permet de vérifier l'équilibre entre les deux classes, un élément crucial pour assurer des performances fiables du modèle. Un déséquilibre important pourrait biaiser l'entraînement en faveur de la classe dominante, réduisant ainsi la capacité du modèle à détecter avec précision les classes minoritaires. La représentation de la figure x1 a montré que le dataset utilisé est suffisamment équilibré, garantissant ainsi une apprentissage stable et une généralisation correcte sur de nouvelles données.

5.2.1 Configuration et entraînement du modèle

Le modèle a été entraîné sur 25 époques en utilisant l'optimiseur Stochastic Gradient Descent (SGD) avec un taux d'apprentissage de 0,01 et un paramètre de momentum de 0,9. La fonction de perte employée est la CrossEntropyLoss. 20% du jeu de données initial a été réservée pour la validation, tandis que 80% a servi pour les tests. L'entraînement

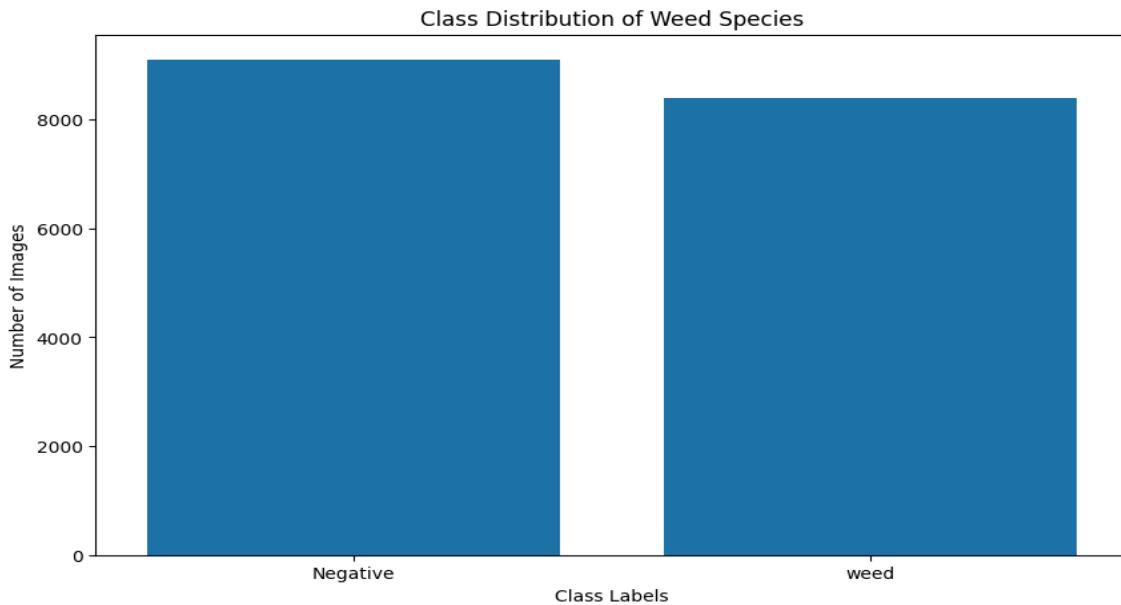


FIGURE 5.1 – Répartition des deux classes des herbes dans le jeu de données DeepWeeds.

a permis d'atteindre une précision globale de 95,17 % sur ce premier jeu de données, confirmant ainsi la capacité du modèle à distinguer efficacement les mauvaises herbes des plantes non nuisibles.

L'évolution des courbes de la perte (Loss) et de la précision (Accuracy) à travers les 25 époques, comme le montrent la Figure 5.2, illustre une progression cohérente dans l'apprentissage du modèle. La perte d'entraînement diminue progressivement, accompagnée d'une amélioration de la précision. La perte de validation suit une tendance similaire, reflétant ainsi une généralisation efficace du modèle aux données non vues mais on remarque une hausse brutale et légère de la perte et une diminution de la précision dans la dernière époque comme montré dans la Figure 5.2, ce qui indique le début d'un surapprentissage. Il est donc plus raisonnable d'arrêter l'entraînement à cette époque.

La matrice de confusion de la Figure 5.3 permet de visualiser les erreurs de classification. Elle montre comment le modèle classe les différentes espèces et les cas où des confusions entre classes apparaissent.

Les confusions les plus fréquentes concernent les mauvaises herbes et les plantes non nuisibles, où certaines images ont été mal classées. Cependant, ces erreurs restent limitées : 60 plantes non nuisibles (3.35% des bonnes plantes) ont été à tort identifiées comme mauvaises herbes, et 109 mauvaises herbes (6.35% des mauvaises plantes) ont été classées à tort comme plantes non nuisibles. Dans l'ensemble, le modèle montre une bonne fiabilité, notamment pour les catégories ayant un support important.

5.2.2 Sauvegarde et préparation à la généralisation

Une fois entraîné, le modèle a été sauvegardé sous le nom `trained_model.pth`. Cette version initiale sert de point de départ pour une utilisation ultérieure avec d'autres datasets.

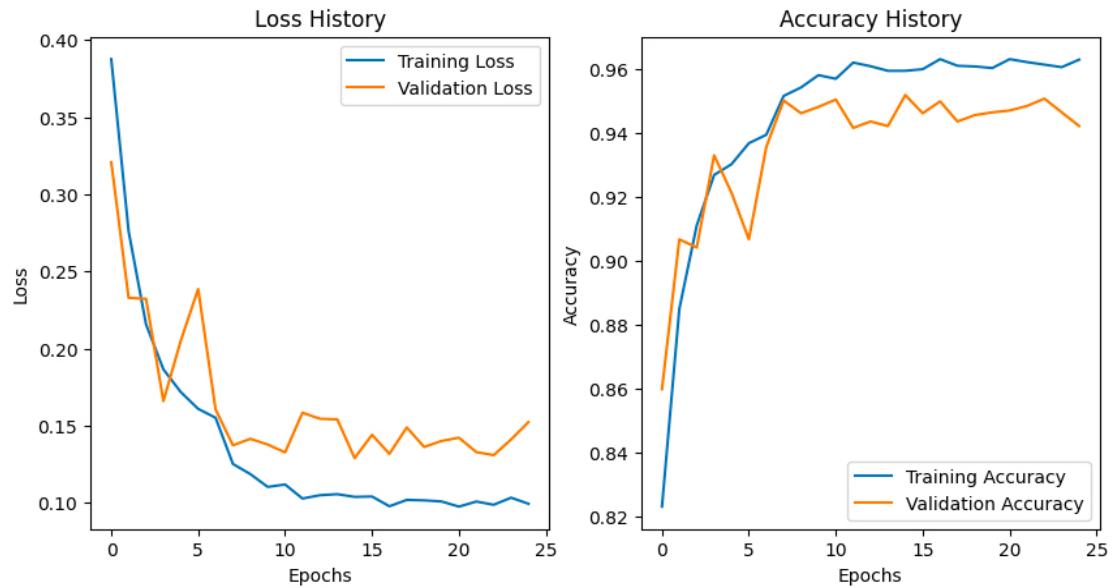


FIGURE 5.2 – Courbes de la perte (Loss) et de la précision (Accuracy).

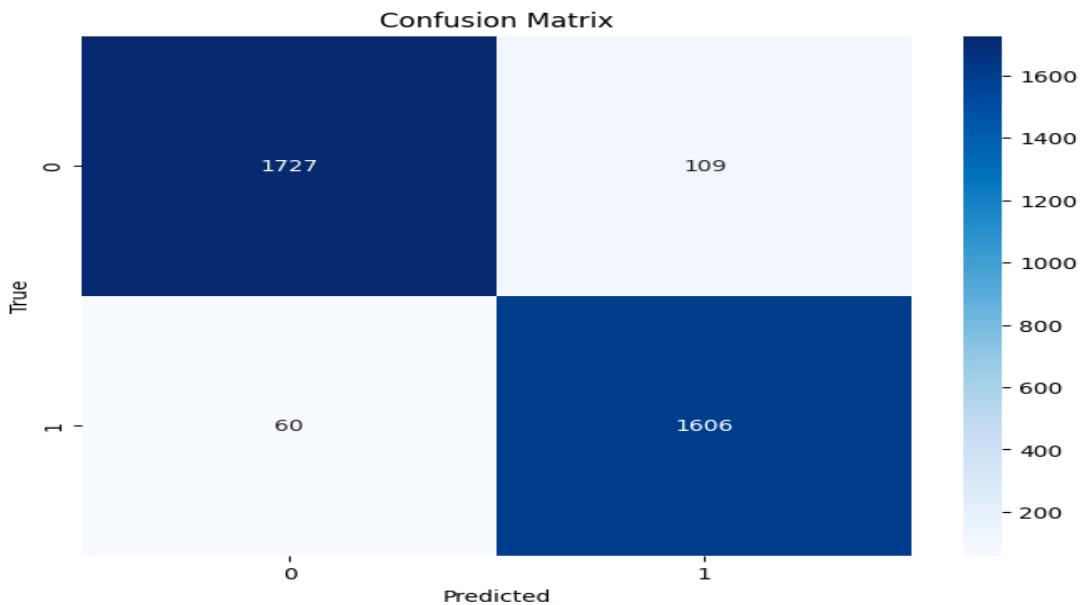


FIGURE 5.3 – Matrice de confusion.

L'objectif est de transférer les connaissances acquises et affiner le modèle afin qu'il puisse mieux s'adapter à des contextes différents. Cette approche de généralisation via le transfert d'apprentissage permet de réduire le temps d'entraînement et d'améliorer l'efficacité globale du modèle.

5.3 Application du modèle trained_model.pth avec un second dataset

Pour améliorer la performance du modèle trained_model.pth et renforcer sa capacité de généralisation, nous avons enrichi le dataset Crop-and-weed-detection-data-with-bounding-boxes avec 5 % d'échantillons provenant du dataset DeepWeeds. Le dataset DeepWeeds comporte 9 classes correspondant à différentes espèces de plantes et adventices, ce qui permet de garantir une grande diversité.

L'ajout de 5 % d'échantillons par classe permet de tirer parti des connaissances acquises par le modèle initial, tout en limitant le risque de surapprentissage. Cette fusion partielle vise également à améliorer la robustesse du modèle face à des cas non vus auparavant. Voici les étapes principales suivies :

5.3.1 Extraction des échantillons

5 % des échantillons de chaque classe de DeepWeeds ont été sélectionnés aléatoirement. Ces échantillons comprennent à la fois des adventices (mauvaises herbes : classe 1) et des plantes non adventices (classe : 0), garantissant ainsi une représentation équilibrée des différentes catégories présentes dans le dataset.

5.3.2 Prétraitement et fusion

Les échantillons extraits ont été transformés et normalisés de manière cohérente avec les données du dataset crop-and-weed-detection-data-with-bounding-boxes. Les deux datasets ont ensuite été fusionnés dans un fichier CSV unifié afin de faciliter l'entraînement du modèle.

5.3.3 Entraînement et fine-tuning sur le dataset enrichi

Pour tirer parti d'un modèle pré-entraîné et améliorer sa capacité à généraliser sur de nouvelles données, nous avons utilisé un modèle ResNet50 sauvegardé sous la forme trained_model.pth, initialement entraîné sur un autre dataset DeepWeeds. Ce modèle a été appliqué au dataset enrichi (combinaison de crop and weed detection avec 5 % d'échantillons du dataset DeepWeeds).

L'objectif était d'effectuer un fine-tuning en déverrouillant les dernières couches du modèle pour lui permettre d'apprendre les caractéristiques spécifiques à ce nouveau dataset enrichi, tout en conservant les représentations déjà acquises. Les étapes du Processus d'Entraînement sur le Dataset Enrichi sont comme suit :

- Chargement du modèle pré-entraîné :

Le modèle trained_model.pth (basé sur ResNet50) a été chargé avec les paramètres sauvegardés. Ce modèle a été configuré pour détecter deux classes : cultures (crop : 0) et mauvaises herbes (weed : 1).

- Déverrouillage des dernières couches :

Toutes les couches ont été gelées, à l'exception des 5 dernières, afin de ne pas perturber les poids déjà acquis sur le dataset initial tout en permettant l'adaptation à ce nouveau dataset enrichi.
- Chargement et préparation des données :

Le fichier crop_and_weed_data.csv, fusionné avec les échantillons de DeepWeeds, a été utilisé pour créer les ensembles d'entraînement et de test.
- Optimisation et entraînement du modèle :

L'entraînement du modèle a été réalisé sur 10 époques, avec une réduction progressive de la perte. Pour assurer une convergence stable, un fine-tuning a été effectué en utilisant l'optimiseur Adam, avec un taux d'apprentissage très faible de 0.000001, ce faible taux d'apprentissage a permis de modifier les poids du modèle de manière progressive, favorisant ainsi une convergence stable et réduisant le risque d'oscillations ou de divergence pendant l'entraînement. Cette approche a également contribué à préserver les connaissances acquises lors des phases initiales d'entraînement, tout en adaptant le modèle aux nouvelles données de manière efficace. Lors de nos expérimentations, l'utilisation de taux d'apprentissage plus élevés a conduit à des mises à jour trop abruptes des poids du modèle, compromettant la stabilité de l'apprentissage et augmentant le risque de convergence vers des solutions sous-optimales.
- Évaluation du modèle sur le dataset enrichi :

Le modèle a été évalué en mode test pour mesurer sa précision générale et par classe. Les résultats ont montré une précision globale remarquable égal à : 98.56%.

Les précisions par classe ont également été calculées pour analyser la performance du modèle pour chaque catégorie. Voici les résultats des précisions par classe :

| Classe | Précision (%) |
|----------|---------------|
| Classe 0 | 98.64% |
| Classe 1 | 98.45% |

TABLE 5.1 – Précision par classe.

Ces résultats montrent que le modèle est capable de classifier avec précision les images des différentes classes du dataset enrichi, avec des performances élevées pour les classes évaluées.

- Sauvegarde du modèle affiné :

L'évaluation du modèle sur le dataset enrichi a révélé une performance globale exceptionnelle, avec une précision générale de 98.56%. Il est important de noter que l'entraînement n'a été effectué que sur 10 époques, et le modèle a réussi à converger efficacement sans nécessiter un grand nombre d'époques. Cela peut s'expliquer par le fait qu'il a réutilisé son expérience acquise lors de l'entraînement sur le dataset DeepWeeds. De plus, les précisions par classe montrent que le modèle est

efficace dans la détection des différentes catégories, ce qui souligne la robustesse de l'approche adoptée.

Après le fine-tuning, le modèle a été sauvegardé sous le nom fine_tuned_model.pth, permettant ainsi de le réutiliser pour tester sa robustesse sur un nouveau dataset, weed-datasets, dans la suite de ce travail.

5.4 Application du modèle fine_tuned_model.pth avec un troisième dataset :Weed-datasets

Comme déjà expliqué avant, et après avoir effectué le fine-tuning du modèle ResNet50, nous avons sauvegardé le modèle sous le nom fine_tuned_model.pth afin de le réutiliser avec d'autres ensembles de données. Pour évaluer la robustesse et la capacité généralisante du modèle, nous l'avons appliqué à un nouveau dataset intitulé weed-datasets, qui contient des images de cultures et de mauvaises herbes collectées dans différents environnements agricoles.

De même, nous avons enrichi le dataset weed-datasets avec 5 % d'échantillons provenant du dataset résultante et qui est elle aussi enrichi obtenu dans la phase précédente, ce qui permet de garantir une grande diversité.

5.4.1 Résultats sur l'ensemble de test

Lors des tests finaux sur l'ensemble d'évaluation initial (utilisé après l'entraînement), le modèle a obtenu une précision globale de 93,03 %. La précision par classe a montré une excellente performance pour la différenciation entre cultures et mauvaises herbes :

- Précision de la classe 0 (Cultures) : 95,74 %.
- Précision de la classe 1 (Mauvaises herbes) : 92,39 %.

Ces résultats indiquent que le modèle est très bon pour détecter les bonnes et les mauvaises herbes. La performance reste très satisfaisante dans les deux cas, ce qui suggère que le modèle est bien adapté pour cette tâche de généralisation.

Après le fine-tuning, le modèle a été sauvegardé sous le nom fine_tuned_model_dt3.pth, permettant ainsi de le réutiliser sur de nouveaux jeux de données.

5.4.2 Test de performance de fine_tuned_model_dt3.pth sur le premier et le deuxième dataset : DeepWeeds et Crop-and-weed-detection-data-with-bounding-boxes

Afin d'évaluer la capacité de généralisation du modèle proposé, nous avons appliqué le modèle fine-tuned dt3.pth sur deux ensembles de données : DeepWeeds et Crop-and-weed-detection-data-with-bounding-boxes. Ce modèle a été entraîné de manière incrémentale en trois phases successives : il a d'abord été formé sur DeepWeeds, ensuite sur Crop-and-weed-detection-data-with-bounding-boxes, et enfin sur weed-datasets. L'objectif de cette

démarche était de simuler un apprentissage progressif et d'analyser les effets d'un transfert de connaissances au fil des jeux de données.

Lors de l'entraînement initial, le modèle avait atteint une précision de 95,17 % sur DeepWeeds et de 98,56 % sur Crop-and-weed-detection-data-with-bounding-boxes, témoignant d'une bonne capacité d'apprentissage. Toutefois, après le fine-tuning final, une réévaluation sur ces mêmes ensembles a révélé une dégradation de performance :

- DeepWeeds : Précision réduite à 80 %
- Crop-and-weed-detection-data-with-bounding-boxes : Précision de 94,20 %

Ces résultats suggèrent que, bien que le modèle conserve une capacité d'adaptation intéressante, des pertes de performance ont été observées, en particulier sur DeepWeeds. Cette baisse peut s'expliquer par des différences de distribution statistique entre les jeux de données utilisés ou encore par un effet de surapprentissage sur les données les plus récentes, entraînant une perte partielle des connaissances acquises antérieurement.

Afin d'approfondir cette analyse, une étude plus détaillée de l'impact des paramètres d'apprentissage s'impose. Des facteurs tels que le taux d'apprentissage, le nombre d'époques par phase, le choix de la stratégie de fine-tuning (congelée ou non), ou encore le rééquilibrage des classes, peuvent jouer un rôle majeur dans la stabilité des performances inter-données. Des expérimentations complémentaires avec ajustement de ces hyperparamètres pourraient permettre d'atténuer les pertes observées et d'optimiser la capacité du modèle à conserver des performances élevées tout au long de l'apprentissage incrémental.

Malgré ces fluctuations, les résultats obtenus confirment le potentiel du modèle pour des scénarios d'application réels, notamment grâce à sa robustesse sur Crop-and-weed-detection-data-with-bounding-boxes, avec une précision maintenue à 94,20 %. Cette approche progressive a permis de doter le modèle d'une meilleure aptitude à généraliser dans des contextes variés de gestion agricole, démontrant la pertinence de l'apprentissage incrémental dans le domaine de la détection de mauvaises herbes.

5.5 Conclusion

Ce chapitre a présenté un modèle de détection incrémentale des mauvaises herbes, visant à améliorer la gestion des cultures et l'utilisation des produits phytosanitaires. En regroupant toutes les mauvaises herbes sous une seule classe et en adoptant une approche incrémentale, le modèle a évolué en intégrant de nouveaux jeux de données, optimisant ainsi sa précision et sa capacité de généralisation. Cette approche a facilité la gestion du code et l'analyse des résultats, simplifiant ainsi le processus. Toutefois, un inconvénient réside dans la perte de distinction entre les différentes espèces de mauvaises herbes, ce qui peut être un facteur limitant selon les objectifs spécifiques de l'application. Après plusieurs itérations, le modèle a démontré une performance remarquable, confirmant son efficacité et sa robustesse pour la détection automatisée des mauvaises herbes.

Chapitre 6

Étiquetage automatique des mauvaises herbes

L'étiquetage manuel des données est une tâche coûteuse et chronophage, en particulier lorsqu'il s'agit de grands ensembles de données complexes, comme ceux contenant des images d'herbes aux caractéristiques variées. Afin de réduire la charge de l'étiquetage manuel, nous avons adopté une approche d'auto-étiquetage. L'objectif est d'entraîner un modèle sur un ensemble de données partiellement annoté, puis d'utiliser ce modèle pour étiqueter automatiquement des échantillons non labellisés. Cette méthode nous permet de constituer un dataset étendu et riche, tout en minimisant l'intervention coûteuse de l'être humain.

Les objectifs de ce travail qu'on a développé dans ce chapitre sont d'élaborer un modèle semi-supervisé capable d'auto-étiqueter les images d'herbes en identifiant correctement les différentes classes. Grâce à ce processus, nous visons à construire un ensemble de données riche et diversifié, qui servira à entraîner des modèles de classification plus performants dans des différentes zones géographiques. En parallèle, il est essentiel d'évaluer la fiabilité de l'étiquetage automatique du modèle proposé afin de mieux comprendre son impact sur la qualité du dataset final et de démontrer l'intérêt de cette méthode dans des scénarios d'utilisation concrets, notamment dans des zones géographiques qui ne disposent pas de datasets. Dans ces régions, la construction d'un dataset nécessite beaucoup de temps et d'efforts de la part des experts, dont la disponibilité est souvent limitée. De plus, le coût élevé de ce processus, combiné au temps nécessaire pour identifier correctement les différentes espèces, complique davantage la création de datasets fiables. Les mauvaises herbes, si elles ne sont pas contrôlées efficacement, peuvent causer des dégâts considérables, ce qui accentue l'urgence d'adopter des solutions automatisées pour optimiser la gestion et la classification des espèces végétales et de préparer des datasets spécifiques pour chaque région qui vont alimenter les modèles qu'on a développé tout au long de ce travail.

La méthodologie suivie dans ce chapitre pour développer ce modèle d'auto-étiquetage suit plusieurs étapes clés :

- Prétraitement des données

- Entraînement initial sur un sous-ensemble annoté
- Auto-étiquetage des données non annotées
- Construction du dataset étendu

6.1 Prétraitement des données

Le prétraitement des données constitue une étape essentielle pour assurer la qualité et la cohérence des images utilisées dans le modèle. Le dataset DeepWeeds a été divisé en deux sous-ensembles : un ensemble annoté et un ensemble non annoté. Après cette division initiale, 60 % des données annotées ont été extraites, tandis que 40 % ont été laissées non annotées, afin de permettre un processus d'auto-étiquetage. Les données annotées sélectionnées ont ensuite été subdivisées en deux parties : 80 % pour l'entraînement et 20 % pour la validation.

Les images ont été redimensionnées à une taille uniforme de 256×256 pixels, afin de garantir une compatibilité optimale avec le modèle. Par ailleurs, des techniques d'augmentation des données, telles que le flip horizontal et la rotation, ont été appliquées dans le but d'améliorer la robustesse et la capacité de généralisation du modèle. Enfin, la normalisation des pixels a été effectuée pour assurer une distribution homogène des valeurs d'entrée, ce qui permet de stabiliser l'entraînement du modèle.

6.2 Entraînement initial sur un sous-ensemble annoté

Dans le code, l'entraînement initial repose sur un modèle de classification basé sur ResNet50, un réseau de neurones pré-entraîné sur ImageNet, dont les couches supérieures ont été adaptées pour la classification des herbes. Le modèle comprend une couche de global pooling, suivie d'une couche dense de 1024 neurones avec une activation ReLU, et se termine par une couche de sortie avec une fonction softmax. L'optimiseur Adam a été utilisé avec un taux d'apprentissage initial de 0,001, réduit de 0,05 tous les 5 époques. La fonction de perte choisie est la categorical cross-entropy, adaptée à la classification multi-classes. Les images sont redimensionnées en 256x256 pixels, normalisées et traitées par lots de taille 32. Un sous-ensemble de données annotées (60 % du dataset) a été divisé en un ensemble d'entraînement (80 % de cet ensemble annoté, soit 48 % du dataset total) et un ensemble de validation (20 % de cet ensemble annoté, soit 12 % du dataset total). L'entraînement a été effectué sur 15 époques. Ce premier entraînement supervisé permet au modèle d'apprendre les principales caractéristiques des différentes classes, offrant ainsi une base solide pour la phase suivante de pseudo-étiquetage.

6.3 Auto-étiquetage des données non annotées

Une fois le modèle 1 entraîné sur le sous-ensemble annoté, il a été utilisé pour prédire les étiquettes des 40 % restants du dataset qui étaient initialement non annotés. Le proces-

sus d'auto-étiquetage repose sur l'exploitation des prédictions les plus fiables du modèle. Seules les prédictions avec une probabilité supérieure à 50 % ont été retenues pour l'auto-étiquetage, tandis que les prédictions en dessous de ce seuil ont été écartées afin d'éviter d'introduire des erreurs dans le dataset. Les échantillons ainsi auto-étiquetés ont été ajoutés progressivement au dataset annoté, augmentant considérablement le volume de données disponibles pour l'entraînement final.

6.4 Construction du dataset étendu

Après la phase d'auto-étiquetage et de validation, nous avons constitué un dataset enrichi contenant 60 % de données annotées manuellement par les experts dans le dataset original et 40% de données auto-étiquetées, correspondant aux échantillons ayant été retenus avec un seuil de confiance supérieur à 0,5. Des échantillons auto-étiquetés ont été écartés en raison d'un niveau de confiance insuffisant. Pour l'entraînement sur le dataset final (combinaison du dataset étiqueté manuellement et du dataset auto-étiqueté), nous avons utilisé comme ensemble de validation les données étiquetées manuellement (12 % du dataset total) et un ensemble de test comprenant 20 % des données auto-étiquetées. Cette répartition a été mise en place pour garantir une évaluation rigoureuse et efficace des modèles entraînés sur le dataset étendu.

6.5 Analyse des résultats

Les résultats d'entraînement affiché sur la Figure 6.1, montrent une amélioration continue de la précision pour les deux modèles. Pour Modèle 1, on observe une hausse rapide de la précision d'entraînement, passant de 58.10 % à la première époque à 95.90 % à la sixième, pour finalement atteindre 99.76 % à la quinzième époque. Toutefois, la précision de validation, bien qu'augmentant de manière significative entre la première (53.36 %) et la sixième époque (85.34 %), diminue légèrement à la fin (84.15 %). Cela peut indiquer une surcharge du modèle sur les données d'entraînement ou un ajustement insuffisant aux variations présentes dans les données manuellement étiquetées.

Pour le Modèle 2 (Le modèle issu de l'entraînement sur l'ensemble manuellement étiqueté et autoétiqueté), Figure 6.2, la performance initiale est déjà élevée avec une précision d'entraînement de 91.91 % et une précision de validation de 93.24 % à la première époque, suggérant que ce modèle capture rapidement les caractéristiques des classes dès le début de l'entraînement. À la dixième époque, la précision d'entraînement et de validation (99.35 % et 99.00 %, respectivement) indiquent un modèle performant et bien ajusté aux données de validation manuellement étiquetées, sans signes évidents de surentraînement.

Enfin, après l'étape d'auto-étiquetage, la précision de test finale (84.17 %) montre une performance satisfaisante mais légèrement inférieure à la précision sur le jeu de validation. Étant donné que les données de test sont issues de l'auto-étiquetage, cette baisse est raisonnable, car elles peuvent contenir des étiquettes moins précises que les données

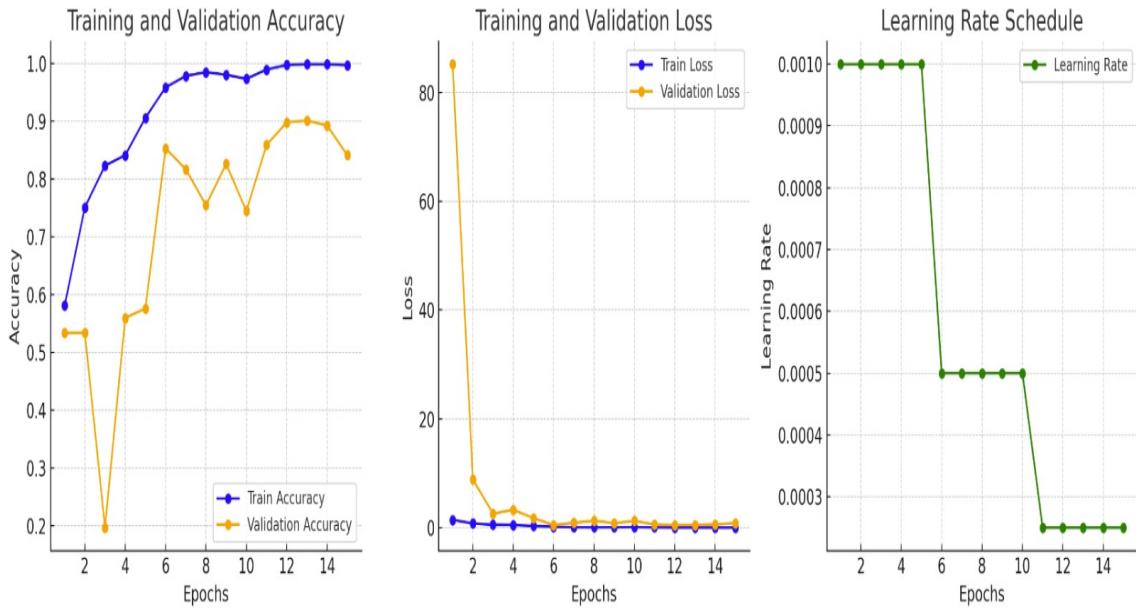


FIGURE 6.1 – Modèle 1 :Entraînement et validation (précision, perte), Leraning rate schedule.



FIGURE 6.2 – Modèle 2 :Entraînement et validation (précision, perte).

manuellement étiquetées, impactant légèrement la capacité du modèle à généraliser sur de nouvelles données. Les résultats montrent une progression impressionnante des modèles, surtout en termes de précision d'entraînement et d'évaluation, qui frôle presque les 100% pour les deux modèles après plusieurs époques.

6.6 Conclusion

Ce chapitre a introduit une approche d'auto-étiquetage automatisé des herbes pour surmonter les défis liés à l'étiquetage manuel des données, particulièrement pour les ensembles d'images complexes. En développant un modèle semi-supervisé capable d'auto-étiqueter les images, nous avons pu créer un dataset étendu et diversifié, essentiel pour améliorer les modèles de classification dans différentes zones géographiques. Cette méthode réduit les coûts et le temps associés à l'étiquetage manuel tout en garantissant la qualité du dataset final, en particulier pour les régions où les ressources humaines sont limitées.

Chapitre 7

Détection et segmentation avec YOLO et Mask R-CNN

Dans ce chapitre, nous analysons deux approches complémentaires pour la détection des mauvaises herbes :

L'optimisation de la détection avec les modèles YOLOv5[30] et YOLOv8[31], qui se concentrent sur la localisation rapide des mauvaises herbes à l'aide de cadres de délimitation. YOLOv5 est reconnu pour sa rapidité et son efficacité, tandis que YOLOv8 offre des améliorations en termes de précision et de polyvalence.

L'exploration de l'utilisation de Mask R-CNN pour la segmentation par instance, permettant une identification plus précise des différentes espèces de mauvaises herbes. Mask R-CNN est une extension de Faster R-CNN, ajoutant une branche pour la prédiction de masques d'objets en parallèle avec la reconnaissance des cadres de délimitation, ce qui permet une segmentation au niveau des pixels.

Cette combinaison de techniques vise à améliorer la précision de la détection et à offrir des perspectives d'optimisation pour les pratiques agricoles.

7.1 Optimisation de la détection des mauvaises herbes avec YOLO v5 et v8 par cadres de délimitation

7.1.1 Description du dataset

Nous rappelons, le jeu de données crop-and-weed-detection-data-with-bounding-boxes contient 1 300 images annotées de cultures de sésame et de différents types de mauvaises herbes, chaque image mesurant 512x512 pixels et étant annotée au format YOLO comme c'est affiché dans la Figure 7.1.

- Collecte de données : Au départ, 589 images de cultures et de mauvaises herbes ont été collectées.
- Nettoyage des données : Après avoir supprimé les images de mauvaise qualité, 546 images ont été conservées.

- Redimensionnement des images : Pour accélérer l'entraînement du modèle, les images initialement de taille 4000x3000 pixels ont été redimensionnées à 512x512 pixels.
- Augmentation des données : En utilisant des techniques comme l'ImageDataGenerator de Keras, le jeu de données est passé de 546 à 1 300 images.
- Étiquetage manuel : Des cadres de délimitation (bounding boxes) ont été dessinés manuellement autour des cultures et des mauvaises herbes dans chaque image pour créer des données annotées pour les modèles de détection.

Les annotations sont dans des fichiers .txt. Par exemple, pour l'image agri_0_1009.jpeg affichée dans Figure 7.2, les annotations sont stockées dans le fichier agri_0_1009.txt, qui contient : 1 0.608398 0.498047 0.541016 0.531250. Où :



FIGURE 7.1 – Exemples de différents types de mauvaises herbes dans le jeu de données crop-and-weed-detection-data-with-bounding-boxes

- 1 : Classe de l'objet détecté. Dans ce cas, "1" représente la classe de l'objet (par exemple, "mauvaise herbe" ou "culture"). Dans YOLO, chaque classe est représentée par un nombre entier unique.
- 0.608398 : Coordonnée x du centre de la bounding box, normalisée par la largeur de l'image. Elle est donnée comme une valeur entre 0 et 1.
- 0.498047 : Coordonnée y du centre de la bounding box, normalisée par la hauteur de l'image. Cette valeur est également comprise entre 0 et 1.
- 0.541016 : Largeur de la bounding box, normalisée par la largeur de l'image.
- 0.531250 : Hauteur de la bounding box, normalisée par la hauteur de l'image.

7.1.2 Préparation de data

- Définition des chemins : On définit les chemins pour les données d'origine et le répertoire de sortie où le jeu de données organisé sera stocké.
- Création de dossiers : Le script crée des répertoires pour les images et les annotations d'entraînement et de validation.
- Liste des fichiers : On collecte tous les fichiers image (avec l'extension .jpeg) et construit une liste des fichiers d'annotation correspondants (avec l'extension .txt).
- Mélange et division : Le jeu de données est mélangé et divisé en ensembles d'entraînement (80 %) et de validation (20 %).
- Déplacement des fichiers : Une fonction d'aide (move_files) est définie pour déplacer les images et les annotations vers leurs dossiers respectifs en fonction de la division.

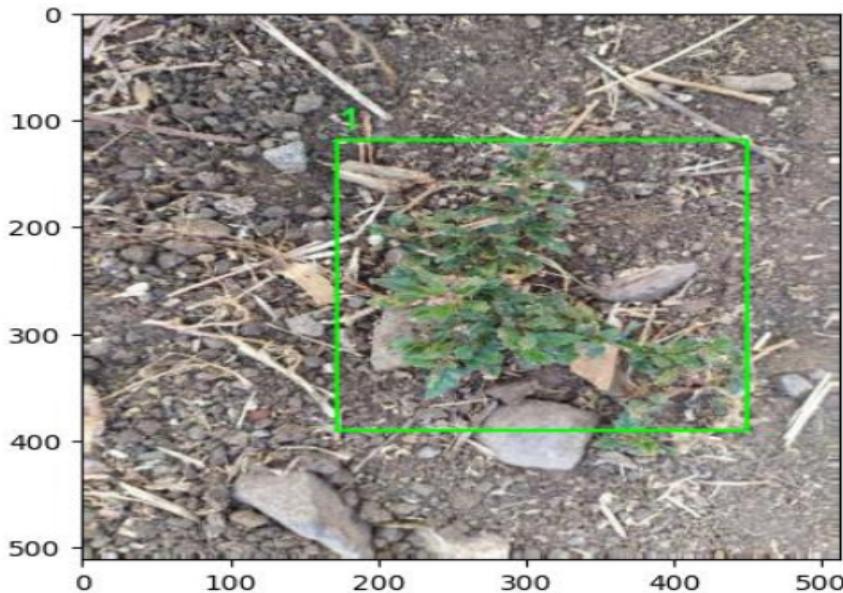


FIGURE 7.2 – Visualisation du résultat sur l'image agri_0_1009.jpeg.

- Création d'un fichier Yaml(agri_data.yaml) : On crée un fichier YAML contenant les chemins d'accès aux images d'entraînement et de validation, le nombre de classes(2) et les noms des classes ("weed" et "Crop")

7.1.3 Construction du modèle YOLOV

- Clonage du répertoire YOLOv5 ou v8 : On clone le dépôt YOLOv5 depuis Github, en permettant l'accès au code et aux ressources du modèle.
- Installation des dépendances : On installe les bibliothèques nécessaires spécifiées dans le fichier requirements.txt .
- Entraînement du modèle : On lance l'entraînement du modèle YOLOv5 avec des images de taille 640x640, un lot de 16 images, sur 50 époques, en utilisant le fichier de configuration agri_data.yaml et les poids pré-entraînés yolov5s.pt, même chose pour v8.
- Évaluation du modèle : On évalue la performance du modèle entraîné en utilisant le fichier de configuration et en mesurant des métriques de performance.

7.1.4 Résultat de YOLO (v5)

Après la phase d'entraînement et d'évaluation, comme affiché dans Table 7.1, le modèle montre de bonnes performances dans la détection des cultures et des mauvaises herbes, avec des points à améliorer notamment en matière de précision pour les mauvaises herbes et de performance sur des seuils d'IoU (Intersection sur Union) plus stricts, en mesurant le chevauchement entre les boîtes englobantes prédites par le modèle et les boîtes englobantes réelles

| Classe | Images | Instances | Précision (P) | Rappel (R) | mAP50 | mAP50-95 |
|--------|--------|-----------|---------------|------------|-------|----------|
| all | 260 | 395 | 0.783 | 0.857 | 0.865 | 0.569 |
| weed | 260 | 221 | 0.722 | 0.870 | 0.872 | 0.588 |
| Crop | 260 | 174 | 0.845 | 0.843 | 0.858 | 0.550 |

TABLE 7.1 – Résumé des performances par classe (YOLO v5)

7.1.5 Interprétation générale YOLO(v5)

Les résultats obtenus, affichés dans la Figure 7.3, montrent :

- 78.3% des objets détectés étaient correctement classés.
- 85.7% des objets réels ont été correctement détectés
- La moyenne de la précision à 50% d’Intersection over Union IoU de 0.865 pour toutes les classes indique une bonne performance du modèle dans la détection des objets.
- Une moyenne de la précision sur des seuils d’IoU de 0,50 à 0,95 de 0,569, indiquant une performance réduite du modèle à des seuils d’IoU plus stricts.

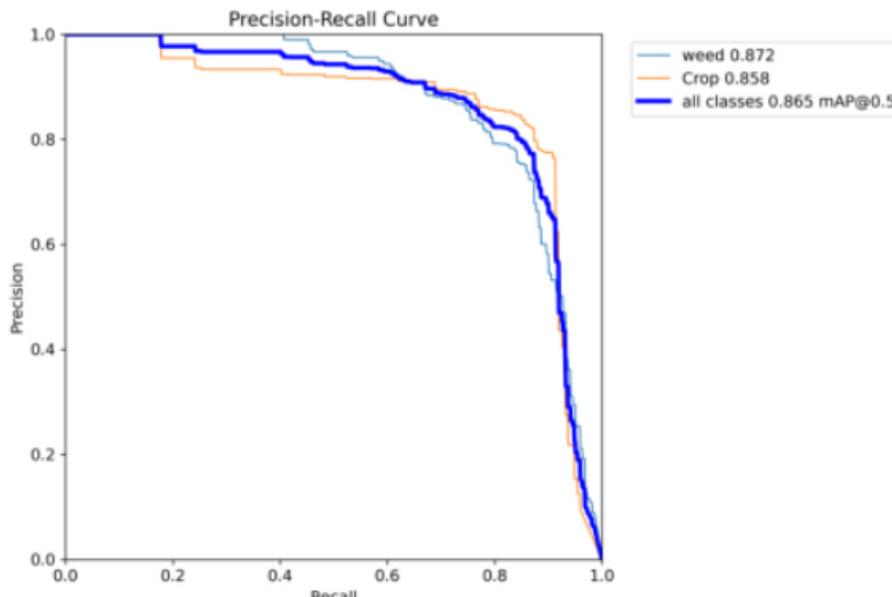


FIGURE 7.3 – Courbe de précision et rappel.

7.1.6 Temps d’inférence YOLO (v5)

- 0.2 ms pour le prétraitement, 6.3 ms pour l’inférence et 3.5 ms pour le Non-Maximum Suppression (NMS) par image ce qui indique que le modèle est rapide dans son traitement, ce qui est important pour les applications en temps réel.

7.1.7 Visualisation d'un résultat YOLO (v5)

La Figure 7.4 présente la visualisation du résultat obtenu avec YOLOv5 sur l'image "agri_0_3943.jpeg"

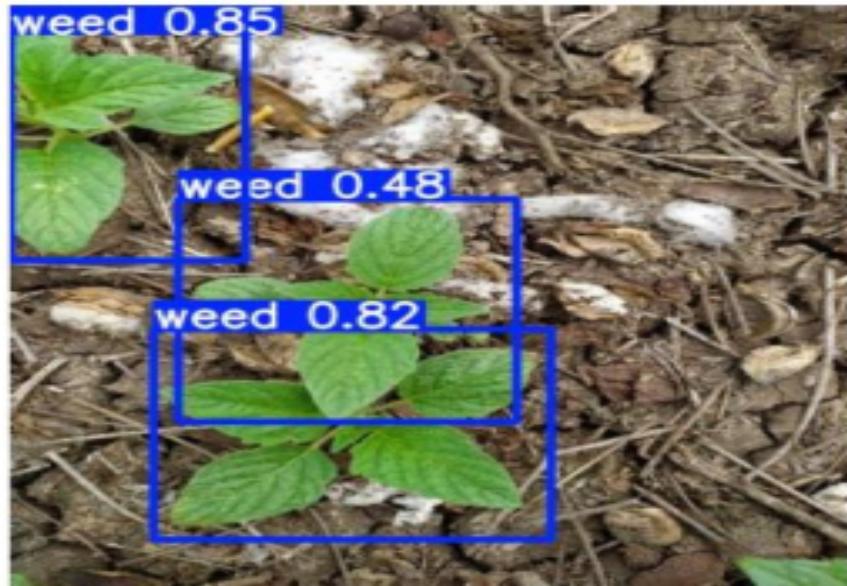


FIGURE 7.4 – Visualisation du résultat sur agri_0_3943.jpeg.

7.1.8 Résultat de YOLO (v8)

L'analyse des performances du modèle révèle des informations critiques sur la capacité du modèle à identifier avec succès les cultures tout en minimisant les fausses détections de mauvaises herbes. (Voir Tableau 7.2) YOLOv8 surpassé YOLOv5 dans la détection des mauvaises herbes grâce à des améliorations architecturales, notamment l'adoption de la détection sans ancrage, qui simplifie le modèle et améliore la précision de localisation des objets.

| Classe | Images | Instances | Précision (P) | Rappel (R) | mAP50 | mAP50-95 |
|----------------|--------|-----------|---------------|------------|-------|----------|
| Tout | 260 | 389 | 0.882 | 0.871 | 0.919 | 0.626 |
| Mauvaise Herbe | 137 | 245 | 0.826 | 0.812 | 0.892 | 0.643 |
| Culture | 123 | 144 | 0.938 | 0.931 | 0.947 | 0.609 |

TABLE 7.2 – Résumé des performances par classe (YOLO v8).

7.1.9 Interprétation générale YOLO (v8)

Les résultats obtenus, affichés dans la Figure 7.5 et du tableau 7.2, montrent :

- 88.2% des détections positives réalisées par le modèle sont effectivement correctes
- Le modèle a détecté 87.1% des instances réelles présentes dans les images ce qui montre que le modèle est performant pour capturer la majorité des objets cibles(cultures)

et mauvaises herbes)

- Le mAP50 l'indicateur de la performance générale du modèle, calculé à un seuil d'intersection sur union (IoU) de 0.5 est de 91.9%, montrant que le modèle a une forte capacité à détecter correctement les objets dans les images, ce qui est encourageant pour des applications pratiques.
- Le mAP50-95 de 62.6% reflète la performance du modèle à des seuils IoU plus stricts, et qui montre que le modèle fonctionne bien à un seuil IoU moins rigoureux, et qu'il rencontre des difficultés lorsqu'il s'agit de respecter des critères de précision plus exigeants.

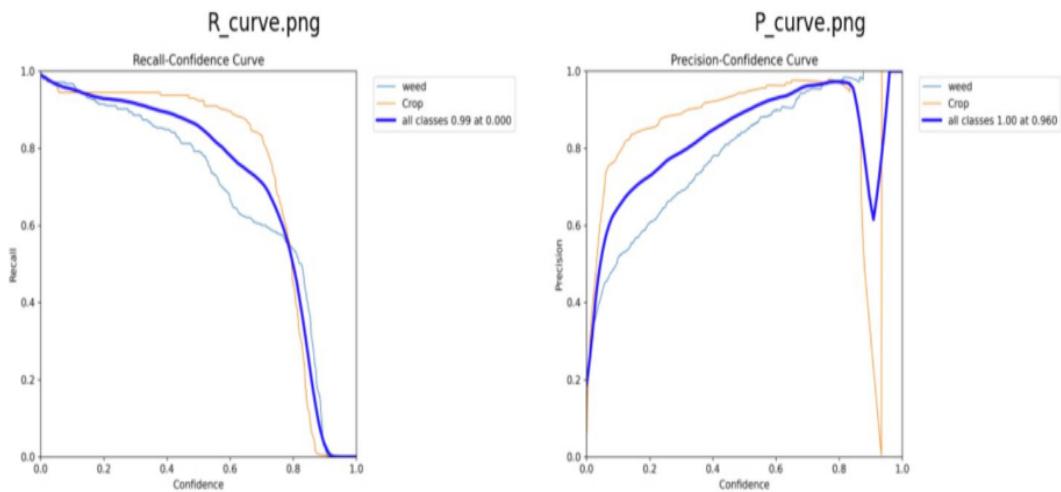


FIGURE 7.5 – Courbe de rappel en fonction de la confiance et Courbe de précision en fonction de la confiance.

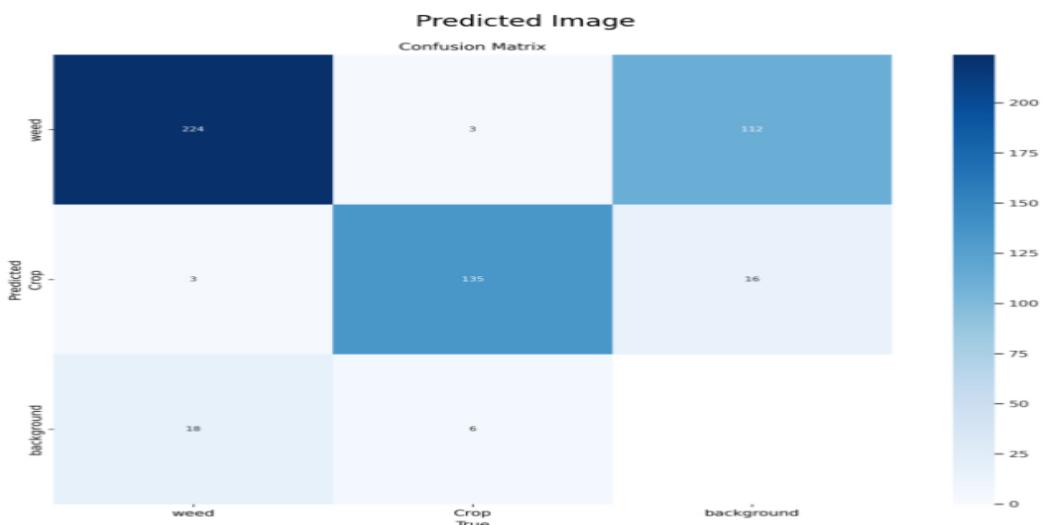


FIGURE 7.6 – Matrice de confusion.

7.1.10 Temps d'Inférence YOLO (v8)

Le prétraitement prend 0,5 ms par image, l'inférence 6,3 ms et le Non-Maximum Suppression (NMS) 3,5 ms par image, ce qui montre que le modèle peut être utilisé pour des applications en temps réel.

7.1.11 Visualisation des résultats YOLO (v8)

Les résultats obtenus avec YOLOv8 sur l'image agri_0_3943.jpeg sont affichés dans la Figure 7.4. Ainsi, comme le montre la matrice de confusion dans la figure 7.6, la plupart des images mal classées ne concernent pas une confusion entre les mauvaises herbes et les cultures, mais résultent de confusions entre l'arrière-plan et soit les mauvaises herbes (principalement), soit les cultures.

Les mauvaises herbes et les cultures présentent une grande diversité en termes de formes, tailles et textures. Cette variabilité peut rendre difficile pour le modèle de distinguer ces objets de l'arrière-plan, surtout si les caractéristiques visuelles de l'arrière-plan sont similaires à celles des objets cibles.

7.1.12 Comparaison et conclusion

YOLOv8 présente une amélioration notable en termes de précision et de rappel par rapport à YOLOv5 pour les deux classes, avec une précision de 88.2% pour le total des classes, contre 78.3% pour YOLOv5.

Ainsi YOLOv8 montre des résultats supérieurs à YOLOv5, surtout pour des seuils plus stricts (mAP@50-95), ce qui en fait une meilleure option pour des applications nécessitant une détection précise.

Et comme le montre la figure 7.7, la détection des mauvaises herbes avec YOLOv8 est très performante. Le modèle parvient à identifier et à délimiter précisément les instances de mauvaises herbes, même en termes de segmentation d'instances.

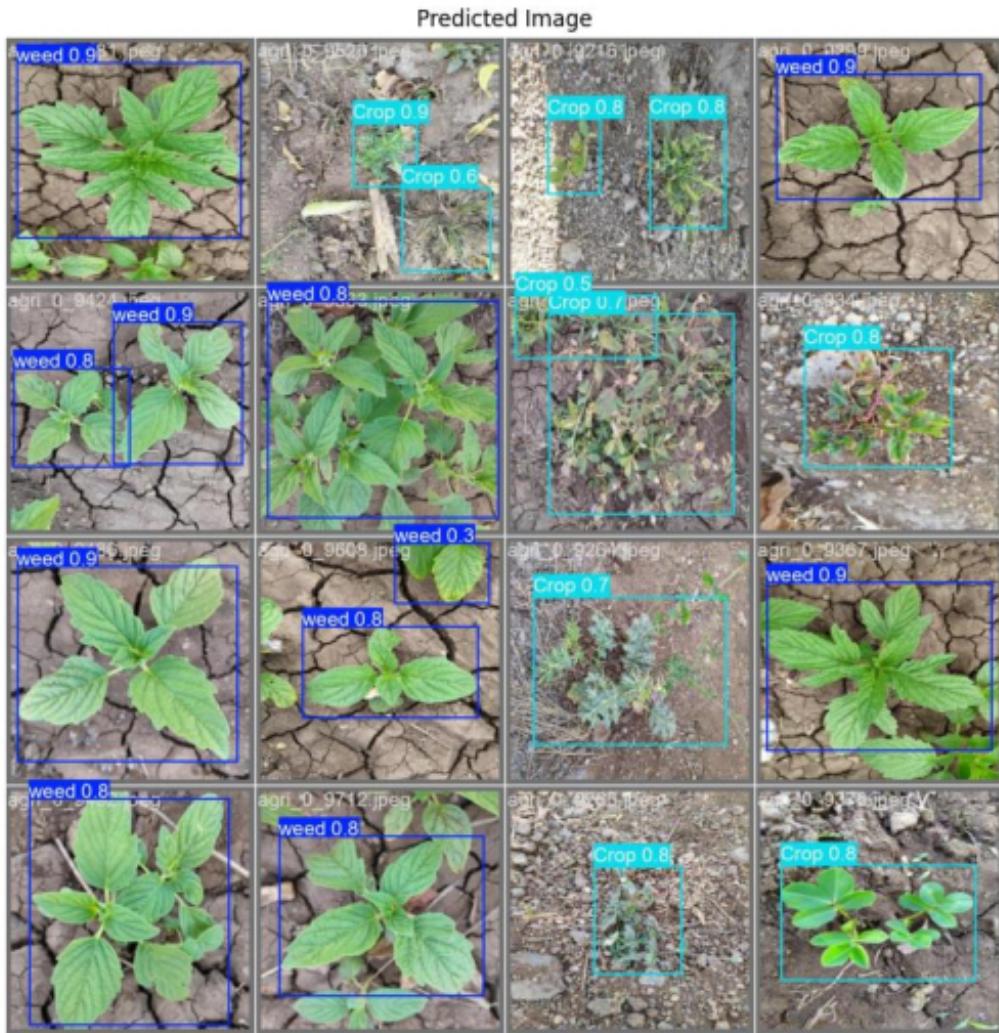


FIGURE 7.7 – Visualisation du résultat sur l'image agri_0_3943.jpeg.

7.2 Optimisation de la détection des mauvaises herbes avec Mask R-CNN : Approche de segmentation par instance et cadres de délimitation

7.2.1 Description du dataset

Le dataset CWF-788 est un ensemble d'images capturées dans des champs de chou-fleur fortement envahis par les mauvaises herbes, conçu pour entraîner et évaluer des méthodes de détection de cultures en milieu envahi. Ce dataset se compose de 788 images, accompagnées d'annotations pixelisées de haute qualité pour chaque image, permettant une segmentation précise des cultures et des mauvaises herbes.

Comme le montre la Figure 7.8, deux versions du dataset sont fournies avec des résolutions différentes :

- IMAGE400x300 : résolution de 400x300 pixels

— IMAGE512x384 : résolution de 512x384 pixels



FIGURE 7.8 – Échantillon d’images de mauvaises herbes dans différentes conditions de luminosité et d’angles.

7.2.2 Installation du detectron2

On installe Detectron2, une bibliothèque de deep learning développée par Facebook AI Research pour les tâches de vision par ordinateur, comme la détection d’objets, la segmentation d’images et la segmentation d’instance, et on importe les librairies nécessaires.

7.2.3 Préparation des données

- Construction de la fonction `load_image_dataset` : La tâche principale de cette fonction est de créer un jeu de données annoté pour la détection d’objets, en extrayant les informations de position et de segmentation des objets (Mauvaises herbes) à partir d’images et de masques correspondants. Les annotations sont structurées pour être compatibles avec le cadre de détection d’objets Detectron2. - Construction de la fonction `register_weed_datasets` : qui enregistre des ensembles de données personnalisés (train, validation et test) pour la détection d’objets (mauvaises herbes) dans Detectron2.

Elle associe chaque ensemble de données à des chemins d’images et de masques correspondants, tout en définissant une seule catégorie d’objets, ”weed” (mauvaise herbe).

7.2.4 Configuration du modèle

On construit la fonction `setup_cfg` qui définit les paramètres pour entraîner le modèle Detectron2 en ajustant les hyperparamètres et en utilisant un modèle de segmentation par instance, `mask_rcnn_R_50_FPN_3x`, depuis le model zoo. Avec une taille de batch de 2 images, un taux d’apprentissage de (`BASE_LR = 0.001`), et un nombre maximal d’itérations (`MAX_ITER = 1000`). Le modèle est configuré pour détecter une seule classe, ”weed”, et utilise un seuil de confiance de 0,5 pour les prédictions.

7.2.5 Entrainement du modèle

On initialise un DefaultTrainer avec les paramètres de cfg, on charge les poids sans reprendre un entraînement précédent (resume=False), puis on exécute l'entraînement avec trainer.train()

7.2.6 Évaluation du modèle

Dans cette étape, les performances du modèle entraîné sont évaluées en rechargeant les poids enregistrés et en configurant un prédicteur. Un évaluateur, COCOEvaluator, est ensuite utilisé pour évaluer les résultats sur l'ensemble de validation. Les données associées à cet ensemble sont extraites, puis la fonction inference_{on dataset test} appliquée pour obtenir les métriques de performance.

7.2.7 Résultats de Mask R-CNN

Le modèle montre d'excellentes performances dans la détection et la segmentation des mauvaises herbes comme le montre le tableau 7.3.

| Métrique | Boîtes Englobantes (bbox) | Segmentation (segm) |
|----------|---------------------------|---------------------|
| AP | 94.33% | 88.26% |
| AP50 | 100.0% | 100.0% |
| AP75 | 100.0% | 100.0% |
| APs | nan | nan |
| APm | 94.28% | 82.77% |
| APl | 94.55% | 89.57% |

TABLE 7.3 – Performances en fonction des métriques pour les boîtes englobantes et la segmentation.

Les performances du modèle Mask R-CNN ont été évaluées à travers diverses métriques, reflétant son efficacité dans la détection et la segmentation d'objets.

1. Précision Moyenne (AP) :
 - AP (Boîtes Englobantes) : Le modèle a atteint une précision moyenne de 94,13% pour la détection des boîtes englobantes, indiquant une détection précise des objets.
 - AP (Segmentation) : Pour la segmentation, la précision moyenne est de 88,26%, suggérant une performance légèrement inférieure dans la segmentation d'instances.

2. Précision à des Seuils d'IoU Spécifiques :
 - AP50 : Une précision de 100% à un seuil d'IoU de 50% démontre une détection fiable des objets avec des critères de correspondance modérés.
 - AP75 : Une précision de 100% à un seuil d'IoU de 75% reflète une capacité du modèle à identifier correctement les objets avec des critères de correspondance stricts.

3. Précision Moyenne par Taille d'Objet :

- APs (Petits Objets) : Le modèle a obtenu une précision de 94,28% pour les petits objets, indiquant une bonne performance dans la détection de ces objets.
- APm (Objets Moyens) : Avec une précision de 94,55%, le modèle est efficace pour les objets de taille moyenne.
- API (Grands Objets) : Une précision de 89,57% pour les grands objets reflète une performance légèrement inférieure, suggérant des opportunités d'amélioration.

4. Comparaison entre Détection par Boîtes Englobantes et Segmentation :

- APbbox : La précision de 94,13% pour les boîtes englobantes est supérieure à celle de la segmentation (88,26%), suggérant que le modèle est plus performant dans la détection de l'emplacement des objets que dans la segmentation précise au niveau du pixel.

7.2.8 Interprétation des résultats

Boîtes Englobantes (bbox)

Les résultats révèlent que le modèle affiche une performance exceptionnelle en détection des mauvaises herbes, atteignant une précision remarquable de 94,33%. Ce taux de précision élevé est particulièrement visible aux seuils d'IoU de 0.5 et 0.75, où il atteint une précision parfaite. Cette performance met en évidence la robustesse du modèle, notamment dans sa capacité à identifier les mauvaises herbes sur des objets de tailles variées. Les objets de grande taille bénéficient particulièrement de cette approche, ce qui confirme l'efficacité du modèle dans des environnements agricoles avec des mauvaises herbes dispersées.

Segmentation (segm)

Les résultats indiquent que, bien que la performance de la segmentation soit légèrement inférieure à celle de la détection par boîtes englobantes, elle reste néanmoins très satisfaisante. En effet, le modèle atteint une précision parfaite aux seuils d'IoU de 0.5 et 0.75, ce qui témoigne de sa robustesse dans des conditions de détection complexes.

Cependant, on observe que la détection des objets de taille moyenne présente une légère diminution de la précision par rapport à la détection réalisée par boîtes englobantes. En revanche, la capacité du modèle à détecter des objets de grande taille reste performante, soulignant ainsi l'efficacité du modèle même pour des objets plus imposants et mieux définis dans les images. Cette dynamique illustre bien les forces et les limites des deux méthodes de détection.

7.2.9 Visualisation des résultats

Après avoir visualisé les résultats des tests dans le fichier `coco_instances_results.json`. On a les images suivantes :(Voir Figure 7.9)

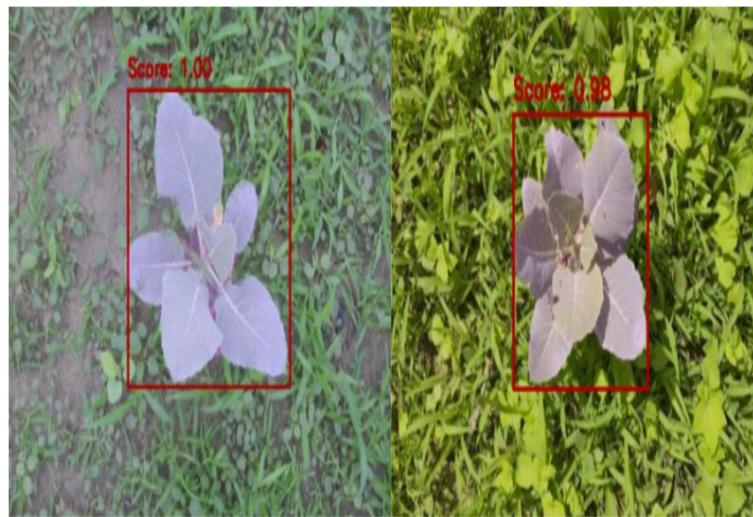


FIGURE 7.9 – Localisation des mauvaises herbes par boîtes de délimitation et par masque, réalisée par le modèle générée..

Pour tester la segmentation par instance, j'ai combiné deux images et effectué une prédiction. (Voir Figure 7.10)

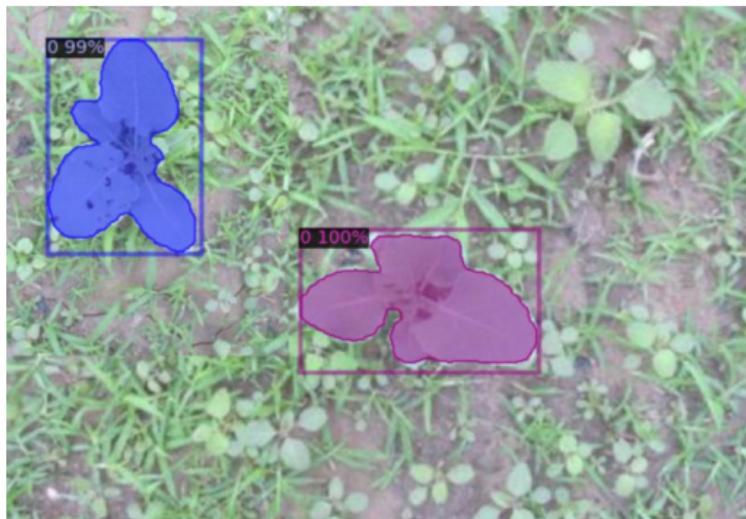


FIGURE 7.10 – Localisation des instances de mauvaises herbes par masques et boîtes de délimitation.

Les figures 7.9 et 7.10 illustrent l'efficacité de Mask R-CNN dans la détection et la localisation des mauvaises herbes. Le modèle identifie avec précision les objets en générant des boîtes de délimitation et en effectuant une segmentation par instance au niveau des pixels. Cette double approche permet de distinguer clairement chaque instance d'objet, même lorsque les objets se chevauchent ou présentent des formes complexes. De plus, la capacité de Mask R-CNN à effectuer une segmentation sémantique au niveau des pixels offre une compréhension détaillée de la structure et de la forme des objets détectés.

7.2.10 Conclusion

Les résultats obtenus démontrent l'efficacité des approches de détection des mauvaises herbes. Avec un AP de 94,33 % pour les boîtes englobantes et 88,26 % pour la segmentation, les modèles YOLO v5 et v8 montrent une excellente capacité à localiser et identifier les mauvaises herbes. Mask R-CNN, en particulier, excelle dans la segmentation par instance, permettant une différenciation précise entre les différentes espèces. Ces résultats soulignent l'importance d'une détection précise et rapide des mauvaises herbes pour optimiser les pratiques agricoles et minimiser l'utilisation de produits chimiques.

Chapitre 8

Conclusion générale

L'objectif principal de cette recherche était de proposer des solutions novatrices et robustes pour la détection et la gestion automatisée des mauvaises herbes, un enjeu crucial pour améliorer les pratiques agricoles tout en minimisant l'utilisation de produits phytosanitaires. Chaque chapitre a contribué à cet objectif global à travers des méthodologies spécifiques, des modèles d'apprentissage avancés et des approches stratégiques adaptées aux défis posés par ce domaine complexe.

Dans le premier modèle, nous avons présenté une solution performante pour la classification des mauvaises herbes en utilisant l'architecture ResNet50. Grâce à un entraînement soigné sur le dataset DeepWeeds, le modèle a atteint une précision remarquable de 94,57%, démontrant ainsi sa capacité à différencier neuf espèces distinctes de mauvaises herbes. Ces résultats soulignent le potentiel des modèles profonds pour résoudre des problèmes complexes d'identification visuelle.

Ensuite, dans le second modèle, nous avons développé une approche incrémentale visant à généraliser la détection des mauvaises herbes sur plusieurs ensembles de données. En regroupant les espèces indésirables sous une classe unique ("weed"), le modèle a été affiné de manière itérative pour s'adapter à différents environnements agricoles. Cette stratégie a permis d'optimiser l'utilisation des ressources tout en renforçant la robustesse et la précision du modèle face à des données inédites.

Dans le troisième modèle, nous avons proposé un système d'auto-étiquetage pour répondre à la problématique du coût et de la complexité de l'étiquetage manuel. À travers un modèle semi-supervisé, des ensembles de données non annotés ont pu être labellisés automatiquement, enrichissant ainsi la diversité et la qualité du dataset. Ce travail a ouvert la voie à la création de datasets spécifiques pour différentes zones géographiques, facilitant le déploiement de modèles adaptés à des régions sous-étudiées.

Enfin, dans le dernier modèle, nous avons exploré des approches avancées de détection et de segmentation en combinant les techniques de cadres de délimitation (YOLOv5 et YOLOv8) et de segmentation par instance (Mask R-CNN). Cette double approche a permis d'atteindre une précision accrue dans la localisation et l'identification des mauvaises herbes, renforçant ainsi l'applicabilité des modèles à des scénarios agricoles variés.

En résumé, cette étude a démontré l'efficacité des techniques d'apprentissage profond pour relever les défis de la détection des mauvaises herbes, allant de la classification initiale à des approches intégrées de segmentation. Elle a également souligné l'importance d'adopter des solutions automatisées, telles que l'auto-étiquetage, pour surmonter les limitations liées à la création de datasets. Ces travaux posent les bases pour des recherches futures, notamment dans l'optimisation de modèles plus légers, la généralisation à d'autres types de cultures et l'intégration d'approches temps réel pour une gestion agricole plus durable.

Références

- 1 F. Ahmed, H.A. Al-Mamun, A.H. Bari, E. Hossain, and P. Kwan, "Classification of crops and weeds from digital images : A support vector machine approach," *Crop Protection*, vol. 40, pp. 98–104, 2012.
- 2 M. Aitkenhead, I. Dalgetty, C. Mullins, A. McDonald, and N. Strachan, "Weed and crop discrimination using image analysis and artificial intelligence methods," *Computers and Electronics in Agriculture*, vol. 39, pp. 157–171, 2003.
- 3 A. Bakhshipour, A. Jafari, S.M. Nassiri, and D. Zare, "Weed segmentation using texture features extracted from wavelet sub-images," *Biosystems Engineering*, vol. 157, pp. 1–12, 2017.
- 4 M. Montalvo, J.M. Guerrero, J. Romeo, L. Emmi, M. Guijarro, and G. Pajares, "Automatic expert system for weeds/crops identification in images from maize fields," *Expert Systems with Applications*, vol. 40, pp. 75–82, 2013.
- 5 J.C. Neto, G.E. Meyer, D.D. Jones, and A.K. Samal, "Plant species identification using Elliptic Fourier leaf shape analysis," *Computers and Electronics in Agriculture*, vol. 50, pp. 121–134, 2006.
- 6 S. Pandey, R. Jain, M.A. Sayeed, and G. Shashikala, "Detection of weeds in a crop row using image processing," *Imperial Journal of Interdisciplinary Research*, vol. 2, 2016.
- 7 X.-E. Pantazi, D. Moshou, and C. Bravo, "Active learning system for weed species recognition based on hyperspectral sensing," *Biosystems Engineering*, vol. 146, pp. 193–202, 2016.
- 8 A. Perez, F. Lopez, J. Benlloch, and S. Christensen, "Colour and shape analysis techniques for weed detection in cereal fields," *Computers and Electronics in Agriculture*, vol. 25, pp. 197–212, 2000.
- 9 G. Coleman, A. Bender, K. Hu, and S.M. Sharpe, "Weed Detection to Weed Recognition : Reviewing 50 Years of Research to Identify Constraints and Opportunities for Large-Scale Cropping Systems," *Weed Technology*, vol. 36, no. 6, pp. 1–50, Nov. 2022. <https://doi.org/10.1017/wet.2022.84>.
- 10 A. Arunachalam and H. Andreasson, "Real-time Plant Phenomics under Robotic Farming Setup : A Vision-Based Platform for Complex Plant Phenotyping Tasks," *Computers & Electrical Engineering*, vol. 92, 107098, June 2021. <https://doi.org/10.1016/j.compeleceng.2021.107098>.
- 11 G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
- 12 M. Guzel, B. Turan, I. Kadioglu, A. Basturk, B. Sin, and A. Sadeghpour, "Deep

Learning for Image-Based Detection of Weeds from Emergence to Maturity in Wheat Fields," *Smart Agricultural Technology*, vol. 9, 100552, Dec. 2024. <https://doi.org/10.1016/j.smartag.2024.100552>.

13 A. Bakhshipour and A. Jafari, "Evaluation of Support Vector Machine and Artificial Neural Networks in Weed Detection Using Shape Features," *Computers and Electronics in Agriculture*, vol. 145, pp. 153–160, Feb. 2018. <https://doi.org/10.1016/j.compag.2017.12.032>.

14 K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 630-645, 2016. https://doi.org/10.1007/978-3-319-46493-0_38.

15 S. Khan, H. Rahmani, S.A.A. Shah, M. Bennamoun, "A Guide to Convolutional Neural Networks for Computer Vision". *Synthesis Lectures on Computer Vision*, 8(1), pp.1-207.2018. <https://doi.org/10.2200/s00822ed1v01y201712cov015>.

16 L. Alzubaidi *et al.*, "Review of Deep Learning : Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, vol. 8, article 53, 2021. <https://doi.org/10.1186/s40537-021-00444-8>.

17 M.S. Ali, "Flattening CNN Layers for Neural Network and Basic Concepts," Medium, Available online : <https://medium.com/@muhammadshoaibali/flattening-cnn-layers-for>

18 T. Diwan *et al.*, "Object detection using YOLO : challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, 2023. <https://doi.org/10.1007/s11042-022-13644-y>.

19 K. He, G. Gkioxari, P. Dollár and R. Girshick,"Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy. <https://doi.org/10.1109/ICCV.2017.322>.

20 W. Abdulla, 2017. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. *GitHub repository*. Retrieved from https://github.com/matterport/Mask_RCNN.

21 J.K. Zhang *et al.*, "Automatic colorectal cancer screening using deep learning in spatial light interference microscopy data," *Cells*, vol. 11, no. 4, article 716, 2022. <https://doi.org/10.3390/cells11040716>.

22 A.Olsen, D.Konovalov,B.Philippa, P. Ridd,, J.Wood, J.Johns, W.Banks, B. Giggenti, O.P.Kenny, J.Whinney,B. Calvert, M.Rahimi Azghadi,R.White,2019. Deep-Weeds : A Multiclass Weed Species Image Dataset for Deep Learning. *Scientific Reports*, vol. 9, article 38343. <https://doi.org/10.1038/s41598-018-38343-3>.

23 S. Shin, J. Kim, Y. Yu, S. Lee, et K. Lee, *Self-Supervised Transfer Learning from Natural Images for Sound Classification*2025 https://www.researchgate.net/figure/Overall-architecture-of-ResNet-50-that-inputs-the-mel-spectrogram-ResNet-infig2_350472892.

24 K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition,*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 770–778, 2016, <https://doi.org/10.1109/TPAMI.2016.90>.

25 G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp.694–1703, 2017 <https://doi.org/10.1109/CVPR.2017.24>.

26 K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition, *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, 2014, pp. 1–12 http://www.icml2014.org/wp-content/uploads/2014/06/ICML_2014_paper_479.pdf.

27 A. G. Howard et al., "MobileNets : Efficient Convolutional Neural Networks for Mobile Vision Applications, preprint arXiv :1704.04861, 2017. [En ligne]. <https://arxiv.org/abs/1704.04861>.

28 D. Steininger, A. Trondl, G. Croonen, J. Simon, and V. Widhalm,, 2023. The Cro-
pAndWeed Dataset : A Multi-Modal Learning Approach for Efficient Crop and
Weed Manipulation.*Proceedings of the IEEE/CVF Winter Conference on Appli-
cations of Computer Vision (WACV)*,pp. 3729-3738. <https://doi.org/10.1109/WACV56688.2023.00374>.

29 J. Motra, 2021. Weed Detection. Kaggle. Available at : <https://www.kaggle.com/datasets/jaidalmotra/weed-detection>.

30 G. Jocher, A. Chaurasia, and J. Qiu, YOLOv5 by Ultralytics, GitHub repository, 2020. [Online]. Available : <https://github.com/ultralytics/yolov5>.

31 D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with
YOLOv8," arXiv, 2023. Disponible sur : <https://arxiv.org/abs/2305.09972>.