# THÈSE DE DOCTORAT

Détection des Collemboles au microscope optique :
utilisation du deep learning pour une analyse automatisée

Présentée en vue de l'obtention du grade de docteur
en **Écologie et Biologie des Populations** de **L'Université Paul-Valéry Montpellier 3**

*Ecole Doctorale n°60 – Territoires, Temps Sociétés, Développement*

Présentée par **Théo Oriol**
Dirigée par **Jérôme CORTET**
Co-encadrée par **Jérôme PASQUET**

Soutenue prévue le **13 Décembre 2024** devant le jury composé de

| | |
|---|---|
| **M. Djamal MERAD,**<br>Professeur, Aix-Marseille Université | Rapporteur |
| **Mme Marie BEURTON-AIMAR,**<br>Maître de conférences, Université de Bordeaux | Rapporteure |
| **Mme Céline PELOSI,**<br>Directrice de recherche, UMR EMMAH Avignon | Présidente du jury |
| **M. Marc CHAUMONT,**<br>Maître de conférence, LIRMM | Examinateur |
| **M. Jérôme CORTET**<br>Professeur, Université Paul-Valéry Montpellier 3 | Directeur |
| **M. Jérôme PASQUET**<br>Maître de conférences, Université Paul-Valéry Montpellier 3 | Co-directeur |

# Remerciements

Ce travail de thèse marque l'aboutissement d'un long parcours de recherche, rendu possible grâce au soutien et à l'accompagnement de nombreuses personnes, que je souhaite remercier ici.

Tout d'abord, je tiens à exprimer ma plus profonde gratitude à mes directeurs de thèse, M. Jérôme Cortet et M. Jérôme Pasquet. Leur encadrement attentif, leurs précieux conseils et leur soutien tout au long de ces années ont été déterminants pour la réalisation de cette recherche. Leur expertise, leur disponibilité et leur engagement m'ont permis d'aborder ce projet avec sérénité et rigueur scientifique.

Je remercie également Nicolas Henon et Mathias Brand, qui m'ont apporté un soutien précieux tout au long de cette thèse. Leur expertise et leur aide ont été indispensables, et chacun de leurs conseils a contribué à enrichir ce travail.

Je tiens à exprimer toute ma reconnaissance à Sandra Bringay, Jérôme Azé et Jean-Marc Pouchoulon, sans qui je n'aurais jamais espéré arriver là où je suis aujourd'hui. Leur soutien indéfectible, leur encouragement et leur confiance m'ont permis de dépasser mes limites et de me projeter avec ambition dans ce parcours scientifique. C'est en grande partie grâce à eux que cette thèse a vu le jour, et je leur en suis profondément redevable.

Je souhaite également remercier l'équipe de l'ESA pour son accueil et son accompagnement tout au long de ce parcours. Les échanges constructifs, la bonne humeur partagée et le cadre de travail agréable ont été des éléments essentiels dans le bon déroulement de ces années de recherche.

Je souhaite exprimer toute ma reconnaissance aux membres du jury de ma thèse pour avoir accepté d'évaluer mes travaux.

Enfin, un grand merci à ma famille, mes amis, ainsi qu'à toutes les personnes qui m'ont soutenu, de près ou de loin, tout au long de cette aventure. Leur soutien, leur compréhension et leurs encouragements m'ont permis de rester motivé et de mener ce projet à son terme. À tous, merci infiniment.

Cette thèse est le fruit d'un effort collectif, et je vous en suis profondément reconnaissant.

# Sommaire


**1 État de l'art** **4**

1.1 Santé des sols . . . 4
1.1.1 Importance de la santé des sols . . . 4
1.1.2 Indicateur de la santé des sol . . . 5
1.1.3 Collemboles . . . 8
1.1.4 Les défis de l'utilisation des Collemboles comme indicateurs . . . 10
1.2 Aperçu machine learning et son évolution . . . 12
1.3 Concepts clés . . . 13
1.3.1 Apprentissage supervisé . . . 14
1.3.2 Apprentissage non supervisé . . . 14
1.4 Apprentissage profond . . . 14
1.4.1 Aperçu de l'apprentissage profond . . . 14
1.4.2 Perceptron . . . 15
1.4.3 Perceptron multi-couches . . . 17
1.4.4 Fonction d'activation . . . 19
1.4.4.1 Les fonctions d'activations classiques . . . 19
1.4.4.2 Les fonctions d'activations rectifiées . . . 20
1.4.4.3 Fonctions d'activation pour les sorties . . . 20
1.4.5 Rétropropagation . . . 21
1.4.6 Descente de gradient . . . 22
1.4.7 Fonction de perte . . . 23
1.4.7.1 Entraînement Supervisé . . . 24
1.4.7.2 Entraînement Non Supervisé . . . 24
1.4.8 Métrique d'évaluations . . . 25
1.4.9 Surapprentissage . . . 27
1.4.10 Réseau neuronal convolutif . . . 28
1.4.10.1 Histoire des CNN . . . 28
1.4.10.2 Architecture d'un CNN . . . 29
1.4.11 Augmentation de données . . . 33
1.4.12 Apprentissage par transfert . . . 34
1.4.13 Architectures importantes . . . 35
1.5 Détection d'objets . . . 37
1.5.1 Les premières méthodes de détection d'objets . . . 37

1.5.2 Les approches basés sur les réseaux neuronaux . . . 38
1.5.2.1 Détection en deux étapes . . . 38
1.5.2.2 Détection en une étapes . . . 44
1.5.3 Méthodes d'évaluations détection d'objet . . . 51
1.6 Détection d'objets dans l'écologie . . . 51
1.6.1 L'émergence du machine learning dans l'écologie . . . 51
1.6.2 L'apprentissage profond en écologie . . . 51

**2 Données et métriques 53**
2.1 Introduction . . . 53
2.2 Jeu de données . . . 53
2.3 Augmentation de données . . . 57
2.4 Métriques . . . 58

**3 Évaluation des méthodes de l'état de l'art 65**
3.1 Introduction . . . 65
3.2 Stratégie d'évaluation . . . 65
3.3 Choix des paramètres . . . 66
3.4 Résultats . . . 67
3.4.1 Résultats de l'état de l'art . . . 67
3.4.2 Matrice de confusion . . . 69
3.4.3 Courbe de Rappel-Précision . . . 71
3.5 Comparaison avec les experts . . . 73
3.6 Conclusion . . . 74

**4 Étude du biais 75**
4.1 Introduction . . . 75
4.2 Grad-CAM . . . 75
4.3 Changement de projet . . . 80
4.4 Impact du biais . . . 85
4.5 Conclusion . . . 87

**5 Active learning 89**
5.1 Introduction à l'active learning . . . 90
5.2 Méthodes proposées . . . 92
5.2.1 Jeu de données . . . 92
5.2.2 ALE . . . 94
5.2.3 Isolation Forest . . . 95
5.2.4 Modèle . . . 96
5.3 Protocole expérimental . . . 96

5.3.1 Expérience . . . . . 96
5.3.2 Évaluation . . . . . 98
5.3.3 Protocoles d'entraînement . . . . . 99
5.3.4 Résultats . . . . . 99
5.4 Conclusion . . . . . 107
5.5 Bilan . . . . . 109
5.6 Perspectives . . . . . 111

**Bibliographie** **113**

# Introduction

La santé des sols est un enjeu crucial pour la durabilité des écosystèmes, car elle influence directement la capacité des sols à réaliser des fonctions telles que la fertilité (Delgado-Baquerizo et al., 2017), la régulation du cycle de l'eau (Brussaard et al., 2007), la régulation biologique (Swift et al., 2004) et la dépollution (Lal, 2015) qui fournissent des services écosystémiques essentiels. La santé des sols est étroitement liée à la biodiversité du sol, qui joue un rôle fondamental dans le maintien de ces fonctions écologiques. Les sols, qui abriteraient 59 % des espèces décrites sur Terre (Anthony et al., 2023), sont des écosystèmes complexes où la biodiversité soutient la résilience des sols face aux perturbations environnementales. Il est donc nécessaire de protéger cette biodiversité pour préserver la santé des sols, ce qui nécessite un suivi efficace.

L'évaluation de la santé des sols reste complexe. L'une des principales difficultés réside dans le manque de mesures fiables, et de nombreuses recherches sont en cours pour définir des indicateurs simples et efficaces (Bünemann et al., 2018). Bien qu'il existe déjà plusieurs normes, y compris des normes ISO (par exemple, ISO 18721, ISO 11074), couvrant certains aspects de la santé des sols, leur adoption et application à grande échelle ne sont pas toujours systématiques. De plus, bien que la biodiversité des sols soit un facteur clé de leur santé, il existe encore un déficit de connaissances dans ce domaine, compliquant l'établissement d'indicateurs robustes (Lehmann et al., 2020). Bien que la diversité des organismes du sol soit reconnue, son utilisation pour évaluer la santé des sols reste limitée.

Les Collemboles (Lubbock, 1873) ou *springtails* en anglais, en raison de leur abondance et de leur diversité, sont considérés comme des indicateurs fiables de la santé des sols. Avec plusieurs milliers d'individus par mètre carré (Santorufo et al., 2014), ces petits arthropodes jouent un rôle clé dans des processus écologiques tels que le cycle des nutriments (Rusek, 1998) et l'agrégation des sols (Siddiky et al., 2012). Comme d'autres organismes du sol, ils sont sensibles aux changements dans les propriétés du sol, notamment l'humidité, la température, le pH et la disponibilité des nutriments (Rusek, 1998). Leur présence et leur diversité fournissent des indications précieuses sur la santé ou la dégradation du sol, en les rendant utiles pour surveiller l'impact des pratiques dans des contextes agricoles, forestiers ou urbains (Cortet et al., 1999 ; M. Fountain & Hopkin, 2004 ; Heisler & Kaiser, 1995). Toutefois, malgré leur potentiel, l'utilisation des Collemboles comme bioindicateurs écologiques à grande échelle est entravée par les difficultés liées à leur identification, un processus chronophage et nécessitant une expertise. De plus, le nombre limité de taxonomistes disponibles pour effectuer cette tâche contraste avec l'augmentation du nombre d'échantillons à analyser, ce qui complique la mise en œuvre

de ces indicateurs à grande échelle.

Pour surmonter ces obstacles, les avancées technologiques offrent une solution prometteuse. Le deep learning, une branche de l'intelligence artificielle, est devenu un outil puissant dans la recherche en écologie et en biodiversité. Ces dernières années, les modèles de deep learning se sont imposés comme une méthode de pointe en vision par ordinateur, offrant des performances remarquables dans diverses tâches. L'application de l'apprentissage automatique en écologie n'est pas nouvelle (Crisci et al., 2012) et a déjà été utilisée pour la modélisation écologique (Recknagel, 2001), l'étude du comportement animal (Arablouei et al., 2023) et l'identification des espèces (Wäldchen & Mäder, 2018). Le deep learning se distingue par sa capacité à traiter de grandes quantités de données et à apprendre directement à partir d'exemples, permettant ainsi d'extraire des caractéristiques importantes pour résoudre une tâche sans être explicitement programmé pour cela (LeCun et al., 2015), ce qui en fait un outil parfaitement adapté pour automatiser l'analyse de données écologiques complexes.

Les défis PlantCLEF (Wäldchen & Mäder, 2018), par exemple, présentent chaque année un ensemble de données volumineux et complexe utilisé pour évaluer et améliorer les modèles de deep learning. Les résultats obtenus montrent des performances de plus en plus élevées malgré la complexité croissante des tâches, confirmant ainsi la pertinence du deep learning pour automatiser et optimiser l'analyse de la biodiversité.

L'identification manuelle d'un grand nombre de spécimens est chronophage pour les experts. Automatiser cette tâche permettrait non seulement de gagner un temps précieux, mais aussi d'étendre la portée des études à plus grande échelle, contribuant ainsi à pallier la rareté des taxonomistes.

L'identification des espèces de Collemboles les plus fréquentes dans les sols pourrait être facilitée par le deep learning. Cela permettrait aux experts de concentrer leurs efforts sur les espèces plus rares ou plus difficiles à identifier. Des tentatives d'identification et d'analyse des microarthropodes à l'aide du deep learning (Kampichler et al., 2000 ; Sys et al., 2022) ont déjà été réalisées, bien que l'identification spécifique des Collemboles sur des lames de microscope à l'aide de cette technologie reste encore à explorer pleinement.

Dans un contexte où plusieurs programmes de recherche visent à appliquer ces indicateurs à des domaines pratiques pour les acteurs socio-professionnels, tels que les instituts techniques agricoles et les gestionnaires d'espaces verts, surmonter les défis liés à l'identification des espèces est crucial. Le deep learning, en tant qu'outil de pointe en vision par ordinateur, pourrait non seulement automatiser l'identification des espèces de Collemboles, mais aussi accélérer ce processus, le rendant plus accessible et applicable à grande échelle pour l'évaluation de la santé des sols.

Cette thèse se propose d'explorer deux hypothèses centrales. La première est qu'un modèle de deep learning peut être développé pour identifier les espèces de Collemboles aussi précisément qu'un expert humain à partir d'images de lames de microscope. La

deuxième hypothèse est que ce modèle sera généralisable, c'est-à-dire capable de maintenir des performances élevées lorsqu'il est appliqué à différents projets et contextes.

La thèse sera structurée en cinq chapitres principaux :

— **État de l'art en écologie et deep learning :** Ce chapitre offrira un aperçu de la santé des sols et des indicateurs écologiques, avec un focus particulier sur les Collemboles. Ensuite, il explorera les origines du deep learning, en se concentrant sur ses applications en classification d'images et en détection d'objets, afin de poser les bases théoriques et contextuelles nécessaires pour comprendre l'intégration de ces technologies dans le domaine.

— **Création du jeu de données et choix des outils :** Ce chapitre abordera la sélection des espèces de Collemboles pertinentes pour l'étude, en tenant compte de leur diversité et de leur représentativité. Les outils utilisés pour capturer les images, tels que les microscopes et les caméras haute résolution, seront décrits. Enfin, les métriques d'évaluation, adaptées à la classification des espèces, seront brièvement présentées pour mesurer la performance du modèle.

— **Benchmark, création et évaluation du modèle de deep learning :** Ce chapitre décrira le processus de création d'un modèle de deep learning spécifique pour l'identification des Collemboles. Il couvrira la conception des benchmarks, le développement de l'outil et l'évaluation des performances du modèle en le comparant aux résultats obtenus par des experts humains. Ce chapitre discutera également des outils utilisés et des méthodes d'évaluation mises en place pour garantir la robustesse du modèle.

— **Étude des biais de projets :** Ce chapitre analysera les biais potentiels qui pourraient affecter la généralisation du modèle développé. Cette section examinera comment ces biais peuvent émerger dans différents projet et contextes d'application. L'objectif sera de s'assurer que le modèle de deep learning est fiable et efficace, quelle que soit la diversité des données et des environnements auxquels il est appliqué.

— **Apprentissage actif :** Le dernier chapitre aborde l'utilisation de l'apprentissage actif pour améliorer la généralisation des modèles d'apprentissage profond face aux variations liées aux projets, notamment dans l'identification des Collemboles. Les biais de projet, causés par des différences dans les méthodes de collecte et de préparation des échantillons, compliquent l'identification automatique. Pour surmonter ces défis, nous avons conçu de nouvelles méthodes d'apprentissage actif qui permettent au modèle de mieux s'adapter à ces variations et d'intégrer de nouvelles espèces, améliorant ainsi sa robustesse et sa précision sur divers projets.

Chapitre 1

# État de l'art

## 1.1 Santé des sols

### 1.1.1 Importance de la santé des sols

La santé des sols est un élément fondamental pour la viabilité des écosystèmes terrestres et la durabilité de nombreuses activités humaines. Elle fait référence à la capacité du sol à fonctionner comme un système vivant et dynamique, en soutenant divers processus essentiels tels que la qualité de l'eau, de l'air et la santé des plantes et des animaux (Doran & Zeiss, 2000). Ces fonctions sont assurées de manière durable, permettant aux écosystèmes de s'adapter aux changements environnementaux.

En matière de régulation du cycle de l'eau, un sol sain joue un rôle essentiel en facilitant l'infiltration des précipitations. Ce processus réduit le ruissellement, l'érosion , tout en contribuant à la recharge des nappes phréatiques (Keesstra et al., 2012). En tant que réservoir naturel, il assure également une disponibilité en eau pour la végétation, réduisant ainsi l'impact des sécheresses et des inondations. De plus, les sols filtrent et décomposent les polluants, garantissant la qualité de l'eau souterraine et des cours d'eau, une fonction clé pour l'approvisionnement en eau potable et la santé des écosystèmes aquatiques (Keesstra et al., 2012).

Ils abritent souvent une biodiversité riche et variée, composée de milliards d'organismes, allant des micro-organismes comme les bactéries et champignons, aux macro-organismes tels que les vers de terre et les insectes (Barrios, 2007). Cette diversité biologique est cruciale pour des processus écologiques vitaux comme la décomposition de la matière organique et le cycle des nutriments. Les interactions entre ces organismes renforcent la fertilité et la structure du sol, améliorant ainsi sa capacité à stocker des nutriments et à retenir l'eau. En outre, la biodiversité dans le sol soutient la résilience des écosystèmes face aux perturbations et aux changements environnementaux, tout en contribuant à la suppression des maladies des plantes. Ces processus naturels maintiennent également des cycles biogéochimiques tels que ceux du carbone, de l'azote et du phosphore, essentiels à la fertilité des sols et à la productivité des écosystèmes terrestres (Crowther et al., 2019).

En tant que réservoirs de carbone organique, les sols sains contribuent à atténuer les effets du changement climatique. La matière organique stockée dans les sols contribue

à la séquestration du carbone, réduisant ainsi la quantité de dioxyde de carbone dans l'atmosphère (Lal, 2004). De plus, les sols régulent la libération d'autres gaz à effet de serre, tels que le méthane et les oxydes nitreux. Des pratiques de gestion durable, comme le labour réduit, la couverture végétale permanente et l'ajout de compost, augmentent la capacité des sols à capter le carbone, jouant ainsi un rôle important dans la lutte contre le réchauffement climatique (Lal, 2004).

Face aux perturbations, qu'elles soient naturelles (inondations, sécheresses) ou anthropiques (pratiques agricoles intensives, pollution), les sols sains démontrent une plus grande résilience. Leur structure stable, associée à une forte activité biologique et à une capacité de rétention d'eau, permet aux écosystèmes de mieux se rétablir après ces événements (Seybold et al., 1999). À l'inverse, les sols dégradés deviennent plus vulnérables à la compaction, à la perte de nutriments et à l'érosion, compromettant ainsi leur rôle de soutien aux écosystèmes environnants.

En conclusion, les sols sont au cœur de la durabilité des écosystèmes terrestres ainsi que des communautés humaines. En assurant la régulation du cycle de l'eau, en abritant une biodiversité riche, en séquestrant lz carbone et en offrant une résilience face aux perturbations, ils jouent un rôle essentiel dans la stabilité des écosystèmes, la régulation du climat et la sécurité alimentaire. Assurer leur gestion durable est primordial pour préserver ces fonctions vitales et garantir la durabilité à long terme des écosystèmes et des communautés humaines.

### 1.1.2 Indicateur de la santé des sol

Les indicateurs de la santé des sols permettent d'évaluer l'état et le fonctionnement des sols en se basant sur leur capacité à remplir les fonctions écologiques essentielles, telles que soutenir la productivité biologique, maintenir la qualité de l'environnement et promouvoir la santé des plantes et des animaux (Doran & Parkin, 1994). Selon la directive Sol de l'Union européenne, la stratégie de l'UE pour la protection des sols à l'horizon 2030 vise à obtenir un bon état de santé pour tous les sols d'ici 2050 et à faire de la protection, de l'utilisation durable et de la restauration des sols la norme. Cette stratégie propose une combinaison de mesures législatives et facultatives, incluant l'adoption d'indicateurs et de valeurs de référence pour la santé des sols, ainsi que des dispositions pour la surveillance et l'utilisation durable des sols. Un indicateur de sol idéal doit ainsi être facile à mesurer, applicable à différents environnements et fiable pour identifier les problèmes du sol (Schloter et al., 2003).

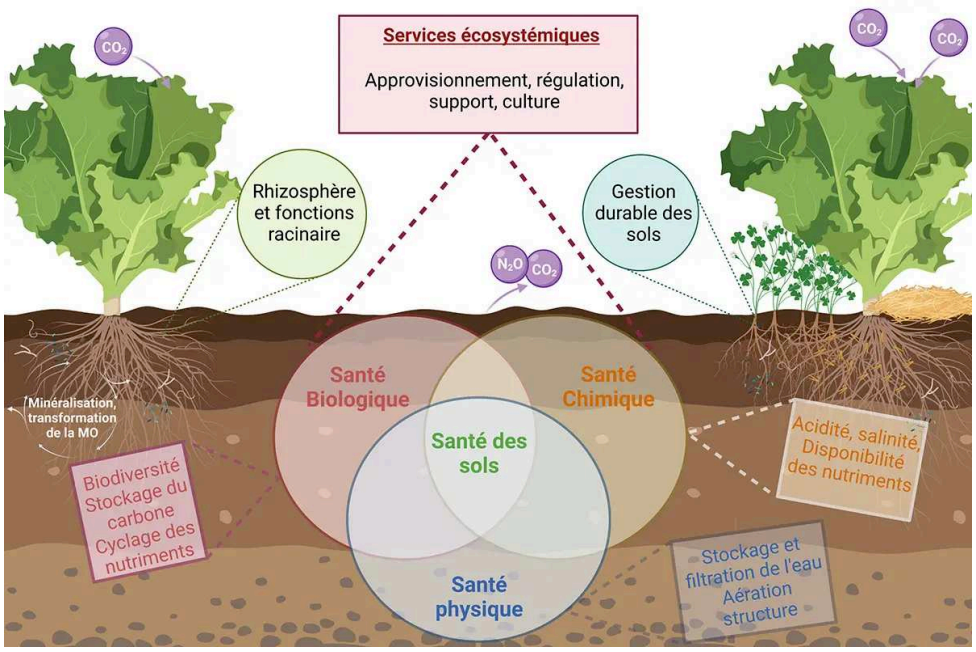


FIGURE 1 – Les différents indicateurs de la santé des sols les plus utilisés [1].

Selon (CARDOSO et al., 2013), les indicateurs de la santé du sol les plus utilisé sont (Figure 1) :

— Les indicateurs chimiques examinent les caractéristiques essentielles des sols qui influencent leur capacité à fournir des nutriments aux plantes et à retenir les éléments potentiellement nuisibles pour l'environnement et la croissance végétale. Parmi les principaux attributs chimiques utilisés pour évaluer la santé des sols, on retrouve le pH, la capacité d'échange cationique (CEC), la matière organique et les niveaux de nutriments.

— Les indicateurs biologiques permettent d'étudier les processus écologiques essentiels ainsi que leur dynamique au fil du temps, offrant ainsi une meilleure compréhension des interactions complexes entre les composantes biologiques, physiques et chimiques. Par exemple, le métabarcodage des bactéries, des champignons, des protistes et des animaux, l'abondance et la diversité des nématodes, des vers de terre et des collemboles ou encore la diversité bactérienne fondée sur l'ADN [2].

— Les indicateurs physiques incluent des évaluations généralement pratiques et réalisables sur le terrain. Parmi ces indicateurs, on trouve la texture, la densité apparente, la porosité et la stabilité des agrégats. Ces éléments sont étroitement liés à des processus hydrologiques tels que l'érosion, l'aération, le ruissellement, le taux d'infiltration et la capacité de rétention d'eau. Un sol est généralement jugé physiquement pauvre lorsqu'il présente de faibles taux d'infiltration, un ruissellement de surface élevé, une faible cohésion, une mauvaise aération et une densité racinaire

1. https ://rqrad.com/
2. https ://data.consilium.europa.eu/doc/document/ST-11299-2024-INIT/fr/pdf

limitée, ce qui complique également sa mécanisation.

Parmi les indicateurs moins utilisés, celui de la biodiversité se distingue. Longtemps sous-développé, cet indicateur fait aujourd'hui l'objet de recherches prometteuses. La biodiversité, ou diversité biologique, désigne la variété et la variabilité du monde vivant sous toutes ses formes. Elle est définie plus précisément dans (Hermitte, 1992) comme la variabilité des organismes vivants de toute origine, y compris, entre autres, les écosystèmes terrestres, marins et autres écosystèmes aquatiques et les complexes écologiques dont ils font partie. Elle constitue un élément fondamental pour évaluer la santé des écosystèmes terrestres. En effet, la richesse des organismes présents dans le sol contribue de manière significative à la santé des écosystèmes, car elle favorise des fonctions essentielles telles que la décomposition de la matière organique, la régulation des nutriments et la résilience face aux perturbations environnementales. La diversité biologique (Brussaard et al., 2007), qu'elle soit microbienne (Fierer, 2017 ; Jacoby et al., 2017), fongique (Frąc et al., 2018) ou faunistique (Bradford et al., 2002), joue un rôle central dans le maintien des processus écologiques et dans la stabilité des sols.

Les organismes du sol interagissent de manière complexe pour soutenir les fonctions essentielles des écosystèmes. Par exemple, les micro-organismes, tels que les bactéries et les champignons, sont responsables de la décomposition des matières organiques, transformant les débris en nutriments assimilables par les plantes. Cette décomposition est cruciale pour la fertilité du sol, car elle permet un cycle continu de matière organique et de nutriments. Parmi les exemples d'organismes jouant un rôle important dans ces processus, on peut citer les nématodes, qui se nourrissent de micro-organismes et régulent leurs populations, contribuant ainsi à la libération de nutriments. Les vers de terre, un autre exemple, jouent un rôle clé dans l'aération et la structuration des sols, améliorant la porosité, l'infiltration de l'eau et la circulation de l'air. Les diplopodes, en fragmentant la matière organique, accélèrent sa décomposition. Ces exemples montrent à quel point ces processus sont cruciaux pour le développement des racines des plantes et la productivité des écosystèmes.

L'évaluation de la biodiversité comme indicateur de la santé des sols repose sur la sensibilité des communautés biologiques aux changements environnementaux (Holt & Miller, 2011). De nombreuses études ont démontré que la diversité et l'abondance des organismes du sol varient en fonction de différents facteurs, tels que la qualité de la matière organique, la pollution des sols et les pratiques agricoles. Par exemple, les sols fortement perturbés, que ce soit par des activités agricoles intensives ou par la contamination chimique, tendent à présenter une diversité biologique réduite (Brussaard et al., 2007). Cette diminution de la biodiversité est souvent corrélée à une dégradation de la santé du sol, rendant la mesure de la biodiversité un indicateur précieux pour évaluer l'état des écosystèmes.

La recherche sur la biodiversité des sols a évolué au fil des années, et de nombreuses

études ont établi un lien fort entre la diversité biologique et la santé des sols.

(Bardgett & Van Der Putten, 2014) souligne l'importance de la biodiversité souterraine pour le fonctionnement des écosystèmes, tandis que (Noss, 1990) discute du rôle des bioindicateurs dans l'évaluation de la santé des sols, mettant en avant la nécessité de surveiller la diversité biologique pour assurer une gestion efficace de ce patrimoine commun.

En conclusion, la biodiversité des sols constitue un indicateur essentiel pour évaluer la santé des sols. Son rôle dans les processus écologiques, sa sensibilité aux perturbations environnementales et sa diversité en font un outil précieux pour les études de gestion et de conservation des sols. Il est impératif d'intégrer ces considérations dans les politiques environnementales afin de promouvoir des pratiques durables et de préserver la santé des écosystèmes terrestres. Une meilleure compréhension de la biodiversité des sols et de son impact sur leur santé est essentielle pour répondre aux défis environnementaux contemporains et garantir la durabilité des ressources naturelles.

### 1.1.3 Collemboles

Les Collemboles (Lubbock, 1873) sont un groupe d'arthropodes présents dans presque tous les types de sols à travers le monde. Le terme "Collembola" provient du grec kólla (« gomme, colle ») et embolê (« jet, insertion »), faisant référence au collophore, une structure située sur la partie ventrale de l'abdomen à laquelle on supposait une fonction d'adhésion au support. Avec plus de 8 700 espèces décrites (Bellinger et al., 2009), ils sont particulièrement abondants dans les sols, où leur densité peut atteindre plusieurs milliers d'individus par mètre carré (Cluzeau et al., 2012). Leur diversité et leur omniprésence dans les sols en font des composants essentiels des écosystèmes terrestres (Figure 2).



FIGURE 2 – Photo de Collemboles

Ils jouent un rôle crucial dans les processus écologiques du sol, notamment dans la décomposition de la matière organique et la régulation des communautés microbiennes (RUSEK, 1998). En se nourrissant de débris organiques, de champignons et de micro-organismes (GOBAT et al., 2010), ils participent activement à la fragmentation et à la minéralisation des résidus organiques, facilitant ainsi le recyclage des nutriments essentiels pour les plantes. Leur activité contribue également à améliorer la structure du sol et son aération. En creusant et en modifiant la structure du sol, ils favorisent l'infiltration de l'eau et la circulation de l'air, ce qui est bénéfique pour le développement des racines des plantes et la productivité des écosystèmes.

Leur sensibilité aux variations des conditions du sol (PONGE et al., 2003) en fait d'excellents bioindicateurs pour évaluer l'état écologique des sols. Leur diversité et leur abondance peuvent être affectées par différents facteurs environnementaux tels que la pollution chimique, la contamination par des traces métalliques, l'acidification des sols et les pratiques agricoles intensives. Par exemple, des études ont montré que les sols contaminés par des pesticides (M. T. FOUNTAIN et al., 2007 ; FRAMPTON, 1997 ; JOIMEL et al., 2022) ou par des éléments métalliques (M. FOUNTAIN & HOPKIN, 2004 ; GILLET & PONGE, 2003 ; RUSSELL & ALBERTI, 1998) présentent souvent une diversité réduite de Collemboles, signe d'une dégradation de la santé du sol.

L'utilisation des Collemboles comme bioindicateurs repose sur leur sensibilité aux polluants et leurs caractéristiques biodémographiques. Bien que leur rôle soit encore à approfondir, notamment en ce qui concerne les effets écotoxicologiques et leur bioaccumulation, ils sont considérés comme des indicateurs pertinents des effets des polluants sur les sols (CORTET et al., 1999). Une communauté riche et diversifiée de Collemboles peut refléter

un sol en bonne santé, tandis que leur sensibilité aux perturbations permet une détection rapide des impacts négatifs sur les écosystèmes.

De nombreuses études ont démontré leur efficacité en tant que bioindicateurs. Par exemple, (Bardgett & Van Der Putten, 2014) a mis en lumière l'importance de la biodiversité souterraine, incluant les Collemboles, pour le bon fonctionnement des écosystèmes. (Cortet et al., 1999) a également souligné la valeur des Collemboles dans l'évaluation de la santé des sols. En outre, (Ponge et al., 2003) a montré que la diversité des Collemboles reflète fidèlement l'état de santé général du sol.

En conclusion, les Collemboles sont des bioindicateurs essentiels pour évaluer la biodiversité et la santé des sols. Leur sensibilité aux perturbations environnementales, leur rôle écologique fondamental et leur présence dans une grande diversité de types de sols et de climats en font des outils précieux pour les chercheurs et les gestionnaires environnementaux. Leur intégration dans les programmes de surveillance des sols est cruciale pour mieux comprendre et préserver la santé des écosystèmes terrestres.

### 1.1.4 Les défis de l'utilisation des Collemboles comme indicateurs

Bien que les Collemboles soient des bioindicateurs précieux pour évaluer la biodiversité et la santé des sols, leur utilisation présente plusieurs défis, notamment la grande quantité de données à traiter et l'expertise nécessaire pour leur identification et analyse, rendant le processus long et complexe (jusqu'à plusieurs centaines d'heures par projet) . La collecte et la préparation des Collemboles pour l'analyse suivent un processus bien défini (suivant l'ISO 23611-2 :2024). Tout d'abord, des carottes de sol sont prélevées sur site, puis les différents spécimens présents sont extraits à l'aide d'un appareil appelé "Macfadyen", qui utilise un gradient de température pour les faire migrer dans un flacon rempli d'éthanol, assurant ainsi leur conservation. Une fois les spécimens collectés, les Collemboles sont séparés des autres taxons à l'aide d'une loupe binoculaire, décolorés avec un produit appelé "Marc-André 1" (Milano et al., 2018) puis montés sur lames avec un produit appelé "Marc-André 2" (Milano et al., 2018), permettant leur identification au microscope (Figure 3).

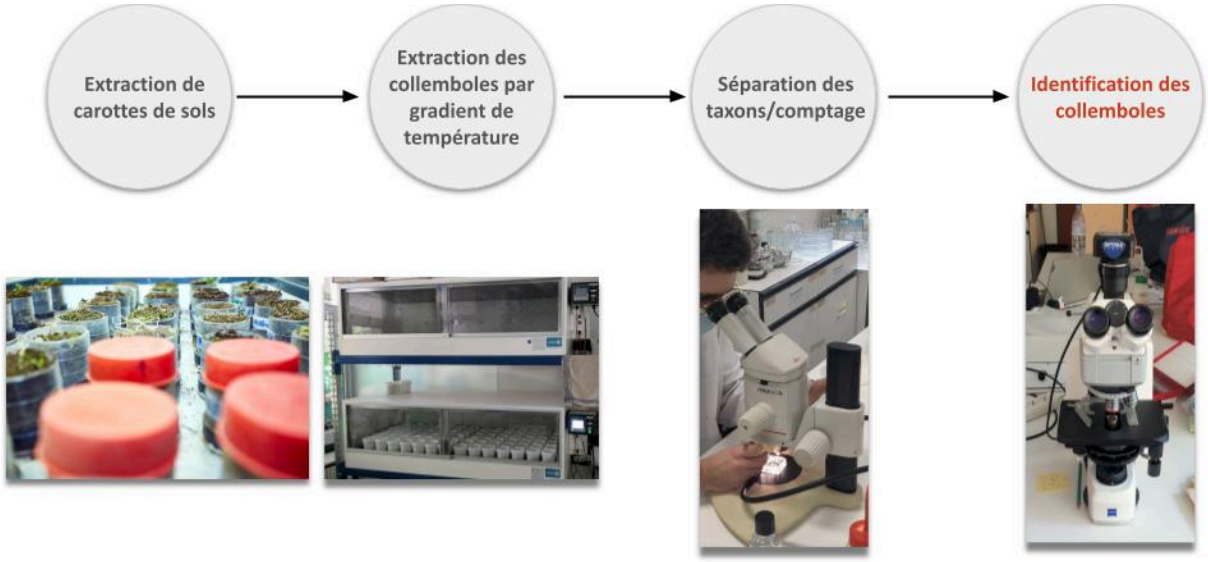


FIGURE 3 – Schéma expliquant le processus d'extraction et d'identification des Collemboles. Cette thèse s'inscrit dans la phase d'identification au microscope, qui est la partie la plus chronophage de ce processus. Une automatisation, même partielle, permettrait d'accélérer grandement l'utilisation des Collemboles comme bioindicateurs.

L'étude des Collemboles nécessite la collecte de données à partir de nombreux échantillons de sol (au moins 3 par site étudié), ce qui génère une quantité considérable d'informations à analyser. Chaque échantillon peut contenir des centaines, voire des milliers, de spécimens appartenant à une grande diversité d'espèces (en moyenne 14 par échantillon (CLUZEAU et al., 2012)) . La gestion de ces données massives implique le tri, le comptage et l'identification. Le traitement de ces volumes de données nécessite des méthodes d'analyse statistique pour interpréter les résultats de manière fiable (CASSAGNE et al., 2003).

La précision de l'utilisation des Collemboles comme bioindicateurs repose sur une identification précise des espèces. Cependant, la taxonomie des Collemboles est complexe et exige une expertise approfondie. Les différences morphologiques entre les espèces, notamment au niveau des soies, sont souvent subtiles et nécessitent l'utilisation de microscopes avec un fort grossissement pouvant aller jusqu'à $\times 630$ et de clés d'identification détaillées (HOPKIN, 1997). Les experts taxonomistes sont rares, et leur formation est longue, ce qui représente un défi supplémentaire pour les études à grande échelle. De plus, les variations morphologiques au sein des espèces en fonction des stades de développement ajoutent une couche supplémentaire de complexité à l'identification (DEHARVENG, 2004).

Pour répondre à ces contraintes, les développements récents dans le domaine des technologies de l'information, et plus particulièrement de l'intelligence artificielle et du deep learning, ont ouvert de nouvelles perspectives. Ces approches permettent d'envisager l'automatisation partielle ou totale de certaines étapes du processus d'identification des espèces, comme la reconnaissance des formes et la classification taxonomique. Le deep lear-

ning, en s'appuyant sur des réseaux neuronaux entraînés sur de vastes bases de données d'images, a déjà montré des résultats prometteurs dans le domaine de la reconnaissance d'images et pourrait jouer un rôle clé dans l'accélération et la fiabilisation de l'identification des Collemboles.

## 1.2 Aperçu machine learning et son évolution

Le machine learning est une branche de l'intelligence artificielle qui consiste à entraîner automatiquement des algorithmes à résoudre des problèmes sans instructions explicites. Son utilisation est particulièrement répandue dans le traitement du langage naturel (NLP) et l'analyse d'images (computer vision). Selon (Mohri et al., 2018) ,le machine learning peut être défini comme l'utilisation de l'expérience pour améliorer les performances ou effectuer des prédictions précises. La Figure 4 clarifie la relation entre l'intelligence artificielle, le machine learning et le deep learning.

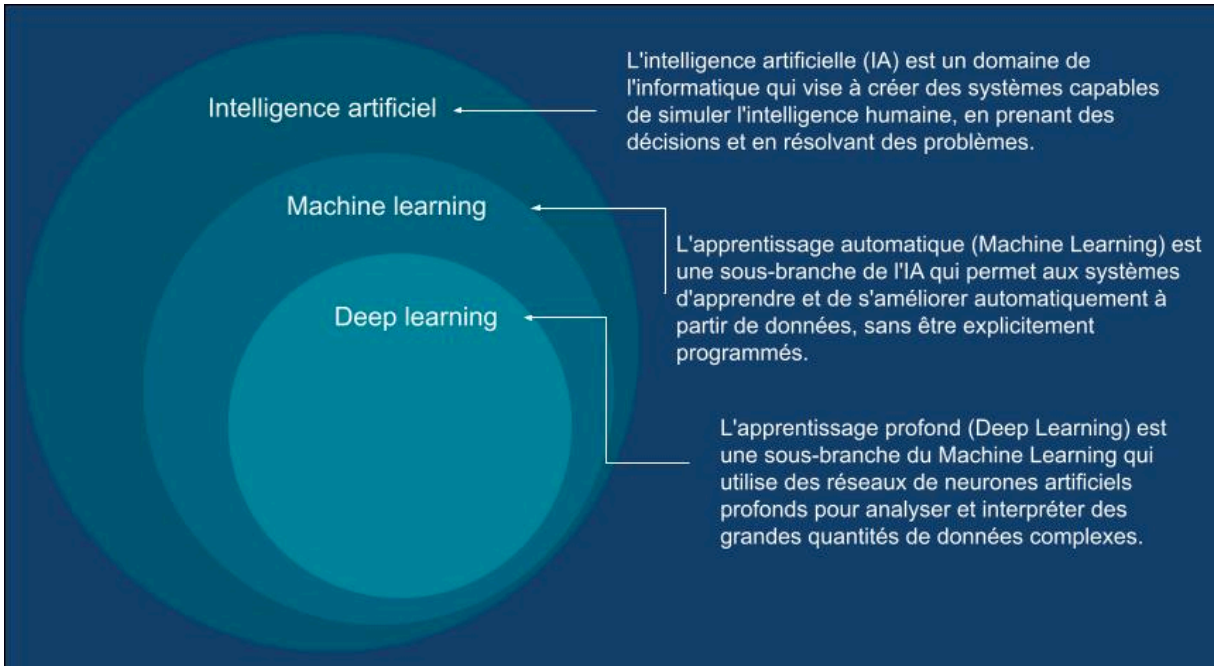


Figure 4 – Diagramme de Venn montrant la relation entre l'intelligence artificielle le machine learning et le deep learning.

Dans le contexte de l'analyse des Collemboles, abordée précédemment, les techniques de machine learning, et en particulier l'analyse d'images, offrent des solutions concrètes pour relever les défis liés à la gestion de grandes quantités de données et à l'identification précise des espèces. Le processus complexe et chronophage d'identification des Collemboles peut être automatisé en utilisant des algorithmes de classification qui apprennent à assigner chaque spécimen à une classe spécifique d'espèce, en fonction de ses caractéristiques visuelles. Cette approche est un exemple typique de la manière dont le machine

learning peut être appliqué pour accélérer le processus de recherche scientifique tout en maintenant un haut niveau de précision.

Parmi les nombreuses applications du machine learning, on retrouve certains des types de problèmes les plus courants, tels que :

- **La Classification** : Constitue une tâche dans laquelle un algorithme, disposant d'un ensemble de classes prédéfinies, doit assigner une classe spécifique à une entrée donnée. Ce problème peut être binaire, comme la détection de courriels indésirables (spam ou non), ou multi-classes, telle que la reconnaissance d'un chiffre à partir d'une image (LeCun et al., 1998).
- **La Régression** : Ce type de problème consiste à prédire une valeur pour une observation en fonction de valeurs d'entrée. Contrairement à la classification, il n'y a pas de classes fixes, et les prédictions peuvent prendre une gamme continue de valeurs. Par exemple, prédire la concentration d'azote dans un bassin fluvial en fonction des données environnantes (Chatterjee & Hadi, 2015).
- **Le Ranking** : Ce problème vise à ordonner des observations en fonction d'un critère spécifique, tel que le classement des pages web les plus pertinentes dans un moteur de recherche, ou recommander des produits basés sur les intérêts d'un utilisateur.
- **Le Clustering** : Il s'agit de regrouper des observations similaires dans des sous-groupes tout en maximisant la similarité intra-groupe et en minimisant la similarité inter-groupes. Par exemple, regrouper des individus dans les catégories "Diabétique" ou "Sain" en fonction de paramètres comme la réponse au glucose et la résistance à l'insuline (Fraley & Raftery, 1998).
- **La réduction de dimension** : Ce problème consiste à transformer les données de manière à réduire le nombre de variables tout en préservant un maximum d'informations pertinentes. Cette approche permet de faciliter l'analyse dans des contextes à haute dimension, tels que l'analyse d'images, la visualisation d'un problème, etc.

Le machine learning offre une approche puissante pour résoudre une large gamme de problèmes. Il permet d'automatiser des tâches complexes et de découvrir des modèles ainsi que des relations dans les données, souvent impossibles à détecter par des méthodes humaines en raison de leur haute dimensionnalité.

## 1.3 Concepts clés

Pendant la phase d'apprentissage d'un algorithme de machine learning, l'objectif est de paramétrer un modèle $f(x)$ afin qu'il puisse, à partir d'une observation, prédire une solution $y$. Il existe deux types de catégories d'apprentissages dans le machine learning. L'apprentissage supervisé et l'apprentissage non supervisé.

### 1.3.1 Apprentissage supervisé

Dans l'apprentissage supervisé pendant l'apprentissage du modèle f($x$), si l'observation que l'on fournit au modèle est associée à une annotation, alors on dit que l'apprentissage est supervisé. C'est un type d'apprentissage que l'on retrouve souvent dans la classification et qui nécessite d'annoter les observations. Les algorithmes d'apprentissage supervisé incluent, entre autres, les machines à vecteurs de support (SVM) (Cortes & Vapnik, 1995), la régression linéaire et les arbres de décision.

### 1.3.2 Apprentissage non supervisé

Dans l'apprentissage non supervisé, pendant l'apprentissage du modèle f($x$), si l'observation que l'on fournit au modèle n'est pas associée à une annotation, alors on dit que l'apprentissage est non supervisé. Ce type d'apprentissage est particulièrement utile pour des tâches telles que le clustering, où l'objectif est de découvrir des structures sous-jacentes dans les données. Il permet ainsi de dégager des représentations ou des caractéristiques importantes sans nécessiter d'étiquettes explicites. Les algorithmes d'apprentissage non supervisé, tels que l'analyse en composantes principales (ACP) (Maćkiewicz & Ratajczak, 1993), K-means et les réseaux de neurones auto-encodeurs.

## 1.4 Apprentissage profond

### 1.4.1 Aperçu de l'apprentissage profond

L'apprentissage profond, sous-catégorie du machine learning, en est très rapidement devenue l'état de l'art. Aussi connue sous le nom de deep learning, il utilise des réseaux de neurones artificiels inspirés par le système de fonctionnement des neurones biologiques afin de résoudre des tâches complexes. La recherche sur les réseaux de neurones artificiels a été lancée par Warren McCulloch et Walter Pitts dans leur article (McCulloch & Pitts, 1943), qui définissent le premier modèle mathématique pour un réseau de neurones artificiels. Ce travail sera ensuite suivi par celui de Donald Hebb dans (Hebb, 2005), qui proposera une explication sur comment ces neurones se mettent à jour. Le deep learning est devenu extrêmement populaire au début des années 2010 en raison de plusieurs facteurs convergents. L'augmentation massive des données disponibles, les avancées significatives en matériel informatique comme les GPU (graphics processing unit), et les nouvelles architectures de réseaux neuronaux ont permis des progrès rapides. Des outils open source comme TensorFlow et PyTorch ont rendu cette technologie plus accessible. Parmi les succès marquants du deep learning, on peut citer la victoire d'AlexNet (Krizhevsky et al., 2012) en 2012, un modèle de réseau neuronal qui a remporté la compétition ImageNet, Cette victoire a démontré la capacité des réseaux de neurones à surpasser les techniques

traditionnelles dans la reconnaissance d'images. Un autre exemple est celui d'AlphaGo (Silver et al., 2017), une intelligence artificielle développée par Google DeepMind, qui a battu un champion humain du jeu de Go en 2016. Ce succès a montré comment le deep learning pouvait résoudre des problèmes extrêmement complexes, même dans des domaines où l'intuition humaine joue un rôle central.

### 1.4.2 Perceptron

Le perceptron (Figure 5) est l'une des premières et des plus simples architectures de neurones artificiels, introduite par Frank Rosenblatt en 1958 (Rosenblatt, 1958). Il a jeté les bases des réseaux de neurones modernes en imitant un neurone biologique simple. En termes de machine learning, le perceptron est un classificateur linéaire qui décide à quelle classe appartient une entrée en effectuant une combinaison linéaire de ses caractéristiques suivie d'une fonction d'activation.

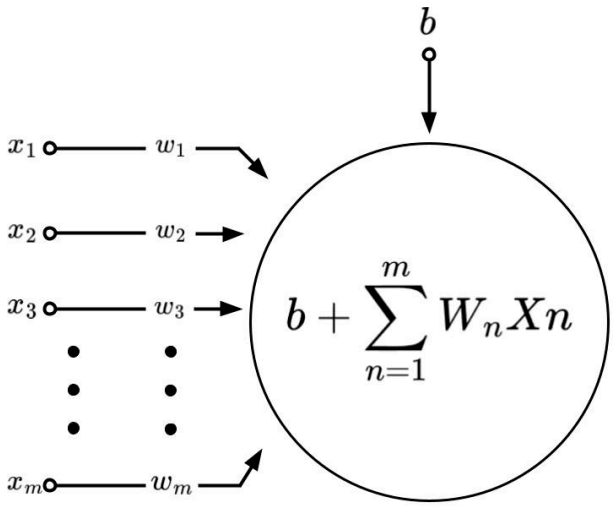


Figure 5 – Exemple de perceptron.

Un perceptron reçoit plusieurs entrées $x_1, x_2, \ldots, x_n$, chacune associée à un poids $w_1, w_2, \ldots, w_n$. Le modèle calcule une combinaison linéaire pondérée des entrées, à laquelle est ajoutée une valeur de biais $b$ :

$$s = b + \sum_{n=1}^{m} W_n X_n \tag{1.1}$$

Ensuite, une fonction d'activation est appliquée à cette combinaison pour produire la sortie. Dans le cas d'un perceptron simple, on utilise une fonction d'activation seuil (ou fonction de Heaviside), qui prend la forme suivante :

$$\hat{y} = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{si } z < 0 \end{cases} \tag{1.2}$$

Le perceptron peut être utilisé pour résoudre des problèmes de classification binaire. Par exemple, il peut classer des points dans un espace en deux catégories en traçant une hyperplan (une ligne en 2D, un plan en 3D, etc.) qui sépare les données. La fonction d'activation permet de déterminer de quel côté de l'hyperplan se trouve un point. L'apprentissage d'un perceptron consiste à ajuster les poids $w_i$ et le biais $b$ afin de minimiser l'erreur de classification sur un ensemble d'entraînement. Le processus d'apprentissage se fait via l'algorithme de rétropropagation, qui met à jour les poids en fonction de l'erreur entre la prédiction $\hat{y}$ et la vraie étiquette $y$. La règle de mise à jour est donnée par :

$$w_i \leftarrow w_i + \eta(y - \hat{y})x_i \tag{1.3}$$

$$b \leftarrow b + \eta(y - \hat{y}) \tag{1.4}$$

où $\eta$ est le taux d'apprentissage. L'algorithme continue à mettre à jour les poids et le biais jusqu'à ce que toutes les observations soient correctement classées ou qu'un nombre d'itérations maximum soit atteint.

Bien que le perceptron ait représenté une avancée importante, il présente des limitations. La principale est qu'il ne peut pas résoudre des problèmes qui ne sont pas linéairement séparables, comme le problème XOR.

**Le Problème XOR et ses Limites pour un Perceptron**

Le problème XOR (ou *exclusive OR*) est un exemple classique en machine learning qui met en évidence une limitation majeure du perceptron simple. Le XOR est une fonction logique à deux entrées binaires, $x_1$ et $x_2$, qui produit une sortie de 1 si les deux entrées sont différentes et une sortie de 0 si elles sont identiques :

$$\text{XOR}(x_1, x_2) = \begin{cases} 1 & \text{si } x_1 \neq x_2 \\ 0 & \text{si } x_1 = x_2 \end{cases}$$

**Table de vérité du XOR :**

| $x_1$ | $x_2$ | $\text{XOR}(x_1, x_2)$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Pourquoi un perceptron ne peut-il pas résoudre le problème XOR ?**
Le principal défi avec le problème XOR est qu'il n'est pas linéairement séparable. Cela signifie qu'il est impossible de tracer une ligne droite qui sépare les points correspondant à une sortie de 1 de ceux correspondant à une sortie de 0. Par exemple, les points $(0, 0)$ et $(1, 1)$ produisent une sortie de 0, tandis que $(0, 1)$ et $(1, 0)$ produisent une sortie de 1.

En raison de cette limitation, le perceptron simple a été progressivement remplacé par des modèles plus complexes, tels que les réseaux de perceptrons multicouches (MLP), qui sont capables de modéliser des relations non linéaires en utilisant plusieurs couches de neurones interconnectées.

### 1.4.3 Perceptron multi-couches

Le Perceptron Multi-Couches (MLP), ou Multilayer Perceptron (Figure 6), est une version plus avancée du perceptron simple, capable de modéliser des relations non linéaires grâce à ses multiples couches de neurones. Contrairement au perceptron simple, qui ne peut résoudre que des problèmes linéairement séparables, le MLP utilise des couches cachées pour apprendre des représentations plus complexes des données. Cela le rend particulièrement efficace pour les tâches de classification et de régression, où des modèles plus sophistiqués sont nécessaires.

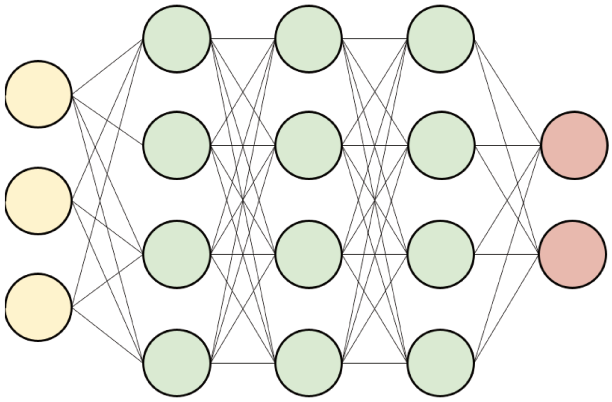

FIGURE 6 – Exemple de perceptron multi-couches.

Les Perceptrons Multi-Couches sont composé de trois types de couches :

— Une **couche d'entrée** qui reçoit les données d'entrée.

— Une ou plusieurs **couches cachées** qui transforment les données en passant par des neurones indépendants avec des fonctions d'activation non linéaires, permettant ainsi au réseau de modéliser des relations complexes.

— Une **couche de sortie**, qui génère les prédictions finales.

La sortie de chaque neurone $h_j$ dans une couche cachée est calculée à l'aide de la formule suivante :

$$h_j = \sigma\left(\sum_{i=1}^{n} w_{ij}x_i + b_j\right) \tag{1.5}$$

où :

— $w_{ij}$ est le poids associé à la connexion entre l'entrée $x_i$ et le neurone caché $h_j$

— $b_j$ est le biais du neurone $j$

— $\sigma$ est la fonction d'activation, par exemple la fonction sigmoïde $\sigma(z) = \frac{1}{1+e^{-z}}$ ou la fonction ReLU $\sigma(z) = \max(0, z)$

La fonction d'activation $\sigma$ introduit la non-linéarité dans le modèle, ce qui permet au MLP de modéliser des fonctions complexes.

L'apprentissage dans un perceptron multi-couches consiste à ajuster les poids et les biais du réseau pour minimiser une fonction de perte, qui mesure la différence entre les prédictions du modèle et les valeurs attendues (les étiquettes). Cet ajustement se fait via des méthodes d'optimisation, telles que la descente de gradient, qui guide la manière dont les poids doivent être mis à jour pour améliorer les performances du modèle. Cette approche permet de traiter les problèmes liés aux erreurs de prédiction.

Cette incapacité à résoudre des problèmes non linéaires a été surmontée avec l'introduction des systèmes multicouches, qui permettent d'apprendre des représentations plus abstraites des données grâce aux couches cachées. La rétropropagation du gradient de l'erreur dans ces systèmes multicouches a permis d'ajuster les poids de manière efficace à travers toutes les couches du réseau, rendant ainsi possible la résolution de problèmes

complexes tels que le XOR. La rétropropagation permet de transmettre l'erreur depuis la couche de sortie vers les couches précédentes en ajustant les poids en conséquence. Cette combinaison de descente de gradient et de rétropropagation est au cœur du processus d'apprentissage des perceptrons multi-couches, bien que ces concepts seront explorés en détail plus tard.

Le perceptron multi-couches présente plusieurs avantages : il permet l'apprentissage des relations non linéaires, grâce à ses couches cachées, et peut modéliser des relations complexes entre les variables d'entrée, ce qui le rend utile pour des tâches de classification et de régression où les simples séparations linéaires ne suffisent pas. De plus, il offre une grande flexibilité : en modifiant le nombre de couches et de neurones, il est possible d'adapter le perceptron multi-couches à différents types de problèmes et à des données variées. Cependant, il y a également des limites à considérer. Les perceptrons multi-couches peuvent être longs à entraîner, surtout pour des architectures profondes ou avec de grands ensembles de données. Le choix des hyperparamètres, tels que le taux d'apprentissage ou la taille des couches cachées, est crucial et peut fortement affecter les performances du modèle. Enfin, bien que la descente de gradient soit efficace, elle peut rencontrer des difficultés comme le problème du gradient qui disparaît, particulièrement dans les réseaux très profonds. Ces défis seront également abordés plus tard.

### 1.4.4 Fonction d'activation

Les fonctions d'activation ont pour objectif d'introduire une couche de non-linéarité entre les couches d'un modèle, permettant ainsi d'apprendre des patterns plus complexes. Il a été démontré qu'en utilisant la fonction d'activation sigmoïde, un modèle à seulement deux couches peut approximer n'importe quelle fonction (Cybenko, 1989). Nous allons ici présenter certaines des fonctions d'activation les plus courantes. On peut diviser ces fonctions d'activation en 3 grosses catégories :

#### 1.4.4.1 Les fonctions d'activations classiques

| **Fonction d'activation** | **Formule mathématique** |
|---|---|
| Sigmoid (ou logistique) | $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ |
| Tanh (Tangente hyperbolique) | $\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$ |

Table 1.1 – Fonctions d'activation classiques

Ces fonctions d'activation sont parmi les plus anciennes et ont été largement utilisées dans les premiers réseaux de neurones. Elles sont toutes deux lisses et différentiables, ce qui permet de calculer le gradient, une mesure qui indique la direction et l'ampleur de la variation la plus forte d'une fonction. Cependant, elles peuvent souffrir du problème de gradient qui disparaît, une situation où la valeur du gradient devient très petite, entravant

ainsi l'apprentissage du réseau en rendant les mises à jour des poids négligeables lors de la rétropropagation.

#### 1.4.4.2 Les fonctions d'activations rectifiées

| **Fonction d'activation** | **Formule mathématique** |
|---|---|
| ReLU (Rectified Linear Unit) | $\text{ReLU}(x) = \max(0, x)$ |
| Leaky ReLU | $\text{Leaky ReLU}(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha x & \text{sinon} \end{cases}$ où $\alpha$ est un petit coefficient positif |
| Parametric ReLU (PReLU) | $\text{PReLU}(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha x & \text{sinon} \end{cases}$ où $\alpha$ est un paramètre appris pendant l'entraînement |
| Exponential Linear Unit (ELU) | $\text{ELU}(x) = \begin{cases} x & \text{si } x \geq 0 \\ \alpha(e^x - 1) & \text{sinon} \end{cases}$ où $\alpha$ est un hyperparamètre qui contrôle la pente pour les valeurs négatives |

TABLE 1.2 – Tableau des fonctions d'activation rectifiées

Ces fonctions d'activation sont largement utilisées dans les architectures modernes de deep learning, en particulier dans les réseaux de neurones convolutionnels (CNN). Leur popularité vient de leur capacité à atténuer le problème du gradient qui disparaît, permettant ainsi la construction de réseaux de neurones beaucoup plus profonds. Cela favorise l'approximation de fonctions complexes en capturant et en modélisant des relations plus riches et plus sophistiquées au sein des données d'entrée.

#### 1.4.4.3 Fonctions d'activation pour les sorties

| **Fonction d'activation** | **Formule mathématique et explication** |
|---|---|
| Softmax | $\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$ Cela renormalise les valeurs $x_i$ en probabilités, assurant que la somme des probabilités soit égale à 1 pour un ensemble de $K$ classes. |
| Sigmoid | $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ Utilisée pour des sorties binaires ou dans des contextes multi-classes où une observation peut être associée à plusieurs classes. |

TABLE 1.3 – Tableau des fonctions d'activation pour les sorties

Ces fonctions sont couramment employées dans la couche de sortie des réseaux de neurones pour des tâches de classification. Softmax est spécifiquement utilisée pour les classifications multi-classes, tandis que la fonction Sigmoïde est préférée pour les tâches de classification binaire ou lorsque plusieurs classes peuvent être activées simultanément.

### 1.4.5 Rétropropagation

En apprentissage profond, la fonction de perte compare la sortie finale du réseau avec les annotations, ou vérité terrain, afin de calculer l'erreur. Cette erreur est directement reliée à la couche de sortie, qui utilise un algorithme d'optimisation pour ajuster immédiatement ses poids en fonction des informations fournies par cette erreur. La mise à jour des poids de la couche de sortie suit généralement la règle suivante :

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}} \tag{1.6}$$

où :

— $w_{ij}$ est le poids entre les neurones $i$ et $j$
— $\eta$ est le taux d'apprentissage
— $\frac{\partial \mathcal{L}}{\partial w_{ij}}$ est le gradient de la fonction de perte $\mathcal{L}$ par rapport au poids $w_{ij}$

Cependant, pour les couches cachées, la situation est plus complexe. Ces couches produisent des activations qui sont utilisées par les couches suivantes pour générer la sortie du réseau. Elles n'ont pas d'accès direct à l'information d'erreur finale. Pour ajuster leurs poids, il est nécessaire de déterminer comment l'erreur provenant de la couche de sortie se propage en arrière à travers le réseau.

C'est là qu'intervient la rétropropagation (ou backpropagation), qui permet de calculer les gradients intermédiaires en utilisant la règle de la chaîne. En propageant les erreurs de la couche de sortie jusqu'aux couches d'entrée, on peut calculer progressivement les contributions de chaque poids à l'erreur finale. Ce processus implique de calculer les dérivées partielles de la fonction de perte par rapport aux poids dans les couches cachées.

Grâce à la règle de la chaîne, le gradient d'une couche cachée peut être calculé comme suit :

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = \frac{\partial \mathcal{L}}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{ij}} \tag{1.7}$$

où $z_j$ représente l'activation du neurone $j$ avant l'application de la fonction d'activation.

Cette chaîne de dérivées se poursuit couche par couche, jusqu'à atteindre les premières couches du réseau. Ce calcul est complexe en raison de la nécessité de propager les erreurs sur plusieurs couches, mais il est essentiel pour ajuster les poids de manière efficace.

La rétropropagation permet de calculer les gradients nécessaires à la mise à jour des poids dans les couches cachées en propageant les erreurs à travers le réseau. Sans cette technique, il serait impossible d'ajuster les poids des couches cachées de manière optimale, car il manquerait les informations nécessaires pour comprendre leur contribution à l'erreur finale.

### 1.4.6 Descente de gradient

Les réseaux de neurones sont capables de résoudre des problèmes complexes en ajustant leurs paramètres pour minimiser une fonction de perte. Pour entraîner le modèle, un algorithme d'optimisation stochastique, appelé optimiseur, est utilisé en parallèle d'une fonction de perte qui guide l'ajustement des paramètres. La descente de gradient et ses variantes représentent l'état de l'art des optimiseurs utilisés en deep learning. Ces méthodes ajustent les paramètres de manière itérative en utilisant un sous-ensemble aléatoire des données (mini-batches), ce qui permet d'accélérer l'entraînement et d'éviter les minima locaux dans des environnements complexes. Voici les étapes principales de la descente de gradient.

**Algorithm 1** Descente de gradient

**Entrées** : Fonction de perte $\mathcal{L}$, Modèle $f$, Ensemble de données $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$
**Initialisation** : Pois du modèle $\mathbf{w}$, Taux d'apprentissage $\alpha$
**repeat**
    Calculer le gradient moyen : $\nabla_{\mathbf{w}} J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} \nabla_{\mathbf{w}} \mathcal{L}(f(\mathbf{x}^{(i)}; \mathbf{w}), y^{(i)})$
    Mettre à jour les poids : $\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} J(\mathbf{w})$

**until** Critère d'arrêt atteint (par exemple, nombre d'itérations)

La descente de gradient est efficace pour converger vers le minimum global ou local d'une fonction continue et différentiable. Cependant, elle peut être coûteuse en temps de calcul, surtout pour de grandes quantités de données, car l'algorithme est utilisé sur l'ensemble du jeu de données à la fois. Une amélioration de cet algorithme, appelée descente de gradient stochastique (SGD), a été créée pour résoudre ce problème.

**Algorithm 2** Descente de gradient stochastique (SGD)

**Entrées** : Fonction de perte $\mathcal{L}$, Modèle $f$, Ensemble de données $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$
**Initialisation** : Pois du modèle $\mathbf{w}$, Taux d'apprentissage $\alpha$
**repeat**
    Choisir aléatoirement un sous-ensemble $(\mathbf{x}^{(i)}, y^{(i)})$ de l'ensemble d'entraînement
    Calculer le gradient de la perte par rapport aux poids :
        $\nabla_{\mathbf{w}} J(\mathbf{w}) = \nabla_{\mathbf{w}} \mathcal{L}(f(\mathbf{x}^{(i)}; \mathbf{w}), y^{(i)})$
    Mettre à jour les poids du modèle : $\mathbf{w} \leftarrow \mathbf{w} - \alpha \nabla_{\mathbf{w}} J(\mathbf{w})$

**until** Critère d'arrêt atteint (par exemple, nombre d'itérations)

Contrairement à la descente de gradient classique, la descente de gradient stochastique partitionne le jeu de données en sous-ensembles aléatoires sur lesquels l'algorithme itère. Cela facilite le calcul du gradient et la mise à jour des poids. Il est important de noter que d'autres améliorations ont ensuite été effectuées, un exemple aujourd'hui très populaire est Adam (adaptive moment estimation) (Kingma & Ba, 2014).

**Algorithm 3** Adam (Adaptive Moment Estimation)

**Entrées** : Fonction de perte $\mathcal{L}$, Modèle $f$, Ensemble de données $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$
**Initialisation** : Pois du modèle $\mathbf{w}$, Taux d'apprentissage $\alpha$, Paramètres $\beta_1$, $\beta_2$, $\epsilon$
Initialise les moments $m_0 = 0$, $v_0 = 0$, étape $t = 0$
**repeat**
    Choix aléatoire d'un exemple $(\mathbf{x}^{(i)}, y^{(i)})$ dans l'ensemble d'entraînement
    $t \leftarrow t + 1$
    Calculer le gradient de la perte par rapport aux poids :
        $g_t = \nabla_{\mathbf{w}} \mathcal{L}(f(\mathbf{x}^{(i)}; \mathbf{w}), y^{(i)})$
    Mettre à jour les moments exponentiels du gradient :
        $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
        $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
    Calcul des moments biaisés :
        $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$
        $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$
    Mettre à jour les poids du modèle :
        $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$

**until** Critère d'arrêt atteint (par exemple, nombre d'itérations)

Adam est un algorithme qui combine la méthode des moments (moyenne des gradients) avec une adaptation du taux d'apprentissage pour entraîner efficacement des modèles d'apprentissage profond. Il offre une convergence rapide et robuste en ajustant automatiquement le taux d'apprentissage pour chaque paramètre du modèle. Les méthodes d'optimisation sont essentielles pour l'entraînement efficace des modèles d'apprentissage automatique, notamment dans le domaine de l'apprentissage profond. Chaque méthode, comme la descente de gradient et ses variantes comme SGD ou Adam, présente des avantages spécifiques en termes de vitesse de convergence et de capacité à atteindre un optimum. Le choix de la méthode dépend du problème spécifique, de la complexité du modèle, et de la taille et de la nature des données. Une bonne sélection et ajustement des méthodes d'optimisation sont cruciaux pour obtenir des modèles performants et robustes.

### 1.4.7 Fonction de perte

La fonction de perte guide l'optimiseur dans la mise à jour des paramètres du modèle afin d'optimiser ses performances vers des prédictions plus précises et plus adaptées aux

données qu'il traite. La nature d'une fonction de perte ainsi que son utilisation diffère en fonction de son utilisation.

#### 1.4.7.1 Entraînement Supervisé

Dans l'entraînement supervisé, où les données sont annotées, la fonction de perte mesure l'écart entre les prédictions du modèle et les valeurs cibles réelles. L'objectif est de minimiser cette fonction de perte afin d'optimiser les paramètres du modèle pour qu'il effectue des prédictions précises sur de nouvelles données. Certains exemples de fonctions de perte populaires utilisées en entraînement supervisé sont :

— **Erreur Quadratique Moyenne (MSE)** : Utilisée principalement dans les tâches de régression, la MSE calcule la moyenne des carrés des différences entre les prédictions et les valeurs réelles.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.8}$$

où $y$ sont les valeurs réelles, $\hat{y}$ sont les prédictions du modèle, et $n$ est le nombre de prédictions.

— **Entropie Croisée (Cross-Entropy)** : Utilisée pour les tâches de classification, l'entropie croisée mesure la divergence entre la distribution de probabilité prédite par le modèle et la distribution de probabilité réelle des annotations.

$$\text{CrossEntropy} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i), \quad \text{avec } y_i \in [0,1] \text{ et } \hat{y}_i \in [0,1] \tag{1.9}$$

où $y$ sont les valeurs réelles, $\hat{y}$ sont les prédictions du modèle, et $n$ est le nombre d'échantillons.

#### 1.4.7.2 Entraînement Non Supervisé

En revanche, l'apprentissage non supervisé ne repose pas sur des annotations ousur une vérité terrain. Les fonctions de perte en apprentissage non supervisé sont conçues pour capturer des caractéristiques telles que la similarité entre les données ou la densité dans l'espace des caractéristiques. Un exemple classique de méthode non supervisée est l'autoencodeur.

— L'**Autoencodeur** est un réseau de neurones qui apprend à comprimer les données d'entrée en une représentation de dimension inférieure (encodage), puis à les reconstruire (décodage). La fonction de perte associée à un autoencodeur mesure la différence entre les données d'entrée et leur reconstruction, souvent à l'aide de

l'erreur quadratique moyenne (MSE) :

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 \tag{1.10}$$

où $\mathbf{x}_i$ représente les données d'entrée et $\hat{\mathbf{x}}_i$ la reconstruction.

L'objectif de l'autoencodeur est de minimiser cette erreur, forçant le réseau à capturer les caractéristiques essentielles des données. Cependant, cette méthode présente des inconvénients notables. Un des principaux désavantages des autoencodeurs est leur tendance à sur-apprendre les spécificités des données d'entraînement, ce qui peut nuire à la généralisation sur des données nouvelles ou légèrement différentes. Ce phénomène, connu sous le nom de surapprentissage, signifie que le modèle pourrait simplement mémoriser les entrées au lieu de véritablement comprendre les structures sous-jacentes des données. De plus, la qualité de la représentation latente dépend fortement du choix de l'architecture du réseau et de ses hyperparamètres, ce qui peut rendre l'entraînement sensible à des ajustements minutieux.

Que l'entraînement soit supervisé ou non supervisé, l'objectif fondamental d'une fonction de perte reste de guider l'optimiseur en fournissant une estimation de l'erreur. Cette erreur est ensuite utilisée pour ajuster les paramètres du modèle dans le but d'améliorer ses performances. Le choix de la fonction de perte est donc crucial, car il doit être adapté à la tâche et au modèle utilisés. Un choix inapproprié peut entraîner des ajustements incorrects du modèle, ce qui pourrait nuire à sa capacité à résoudre efficacement le problème donné.

### 1.4.8 Métrique d'évaluations

Les métriques d'évaluation permettent d'évaluer les performances d'un modèle et son comportement par rapport aux données. Ce sont des outils indispensables qui nous offrent la possibilité d'analyser rigoureusement les performances d'un modèle. Le choix des métriques dépend du modèle utilisé et de son objectif spécifique. Voici quelques métriques d'évaluation et leurs utilisations courantes.

**Pour la classification :**

— **Précision (Precision)** : La proportion de prédictions positives correctes parmi toutes les prédictions positives faites par le modèle.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1.11}$$

où :

— $TP$ (True Positives) : le nombre de vrais positifs, c'est-à-dire le nombre de fois où le modèle a correctement prédit la classe positive.

— $FP$ (False Positives) : le nombre de faux positifs, c'est-à-dire le nombre de fois où le modèle a incorrectement prédit la classe positive.

— **Rappel (Recall)** : La proportion de vrais positifs correctement prédits parmi tous les vrais positifs présents dans les données.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1.12}$$

où

— $FN$ (False Negatives) : le nombre de faux négatifs, c'est-à-dire le nombre de fois où le modèle a incorrectement prédit la classe négative alors que l'exemple était positif.

— **F1-Score** : Une mesure combinée de la précision et du rappel, calculée comme la moyenne harmonique des deux.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1.13}$$

— **Exactitude (Accuracy)** :La proportion totale de prédictions correctes parmi toutes les prédictions faites par le modèle.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.14}$$

où

— $TN$ (True Negatives) : le nombre de vrais négatifs, c'est-à-dire le nombre de fois où le modèle a correctement prédit la classe négative.

**Pour la regression :**

— **Erreur Quadratique Moyenne (MSE)** : La moyenne des carrés des différences entre les valeurs prédites par le modèle et les valeurs réelles.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{1.15}$$

où $y$ sont les valeurs réelles et $\hat{y}$ sont les prédictions du modèle.

— **Coefficient de Détermination (R-squared)** : Une mesure statistique de la proportion de la variance dans la variable dépendante qui est prévisible à partir de la variable indépendante.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{1.16}$$

où $\bar{y}$ est la moyenne des valeurs réelles $y$.

Les métriques d'évaluation jouent un rôle essentiel dans l'apprentissage profond. Elles

permettent de quantifier les performances d'un modèle, d'analyser son comportement et de déterminer comment l'améliorer. Grâce à ces métriques, il est possible d'identifier les points faibles, les biais ou les insuffisances de performance du modèle.

### 1.4.9 Surapprentissage

En machine learning, l'objectif est de développer un modèle capable de généraliser à partir d'une base d'entraînement représentative. Toutefois, il arrive que le modèle apprenne à capturer non seulement les tendances générales, mais aussi le bruit et les fluctuations aléatoires des données d'entraînement, phénomène connu sous le nom de surapprentissage ou "overfitting". Cela entraîne une excellente performance sur les données d'entraînement, mais une mauvaise généralisation sur des données non vues.

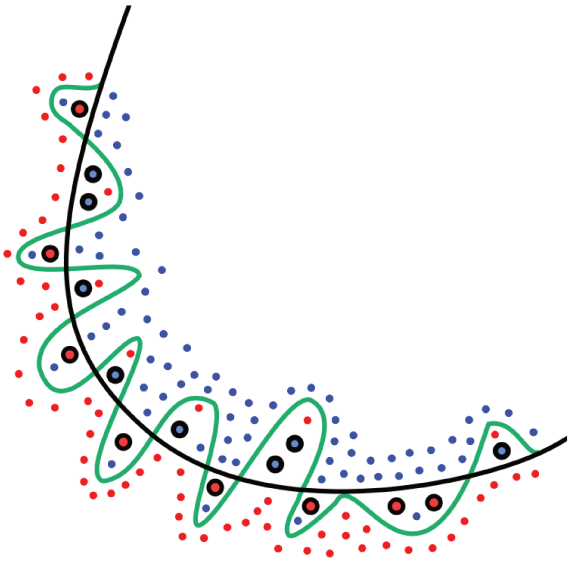

FIGURE 7 – Exemple de surapprentissage. La ligne verte représente un modèle qui a surappris et la ligne noire un modèle régularisé. Bien que la ligne verte suive au mieux les données d'apprentissage, elle est trop dépendante de ces données et est susceptible d'avoir un taux d'erreur plus élevé sur les nouvelles données non vues, illustrées par les points encerclé de noirs, par rapport à la ligne noire. [4]

Le surapprentissage (Figure 7) est particulièrement problématique en deep learning, où les réseaux de neurones profonds, dotés de millions de paramètres, ont une grande capacité à mémoriser les données d'entraînement. En raison de la haute dimensionnalité de ces modèles, l'overfitting est souvent inévitable. Toutefois, même si un modèle surapprend, il est possible de ralentir ce phénomène et d'atteindre une bonne généralisation en appliquant diverses stratégies.

Plusieurs causes de surapprentissage existent, notamment en deep learning :

4. https ://en.wikipedia.org/wiki/Overfitting

— **Un modèle trop complexe** : Les réseaux de neurones profonds avec un grand nombre de paramètres peuvent facilement mémoriser les données d'entraînement, en particulier si la taille et la diversité du jeu de données sont limitées.
— **Un jeu de données limité ou biaisé** : Si les données d'entraînement sont insuffisantes ou ne couvrent pas toute la variabilité nécessaire, le modèle risque d'apprendre des détails spécifiques aux données d'entraînement, au lieu de généraliser à de nouvelles situations.
— **Un manque de régularisation adéquate** : En deep learning, des techniques de régularisation comme le dropout ou encore les techniques d'augmentation de données sont essentielles pour contrer l'overfitting. Sans elles, les réseaux de neurones peuvent facilement devenir suradaptés aux données d'entraînement.

Dans le cas des réseaux profonds, le surapprentissage est souvent inévitable en raison de leur complexité. Cependant, il est possible de limiter ses effets et d'améliorer la généralisation en combinant des stratégies telles que l'utilisation de plus de données, la simplification du modèle et l'application de régularisations appropriées. Paradoxalement, même avec des modèles qui surapprennent, les techniques d'optimisation permettent parfois d'atteindre une généralisation acceptable, notamment en ralentissant l'apparition de l'overfitting.

En résumé, bien que le surapprentissage soit un défi constant en deep learning, il est possible de le mitiger pour obtenir des modèles robustes. Une combinaison stratégique de régularisation, de gestion des biais dans les données et d'ajustements de modèle permet de ralentir l'overfitting et d'améliorer les performances sur des données nouvelles.

### 1.4.10 Réseau neuronal convolutif

Les réseaux de neurones convolutifs (Convolutional Neural Networks, CNN) constituent une classe de modèles d'apprentissage profond particulièrement efficaces pour le traitement des données ayant une corrélation spatiale, telles que les images. Les CNN ont apporté des avancées significatives dans le domaine de la vision par ordinateur, notamment dans des tâches telles que la reconnaissance d'images et la détection d'objets. Cette section explore en détail l'architecture, le fonctionnement et les applications des CNN, ainsi que leurs avantages et limitations.

#### 1.4.10.1 Histoire des CNN

Inspirée par le cortex visuel des animaux, la conception des réseaux neuronaux convolutifs (CNN) fait suite à la découverte des mécanismes visuels des organismes vivants. Le néocognitron, mis en avant par (Fukushima, 1980), est considéré comme le premier réseau de neurones profond, et a introduit le concept de partage de poids pour les unités situées à différentes positions du réseau.

En 1989, Yann LeCun présente LeNet (LeCun et al., 1989), l'un des premiers réseaux neuronaux convolutifs. Quelques années plus tard, en 1998, il introduit LeNet-5 (LeCun et al., 1998), l'un des premiers CNN à succès, utilisé pour la reconnaissance de caractères manuscrits, notamment les chiffres du jeu de données MNIST. MNIST (Modified National Institute of Standards and Technology) est une base de données largement utilisée en machine learning, contenant 60 000 images de chiffres manuscrits pour l'entraînement et 10 000 pour les tests. Cette base de données a servi de référence pour tester les algorithmes de reconnaissance d'images, en raison de sa simplicité et de sa large adoption dans la communauté scientifique. Cette application a démontré la puissance des CNN pour les tâches de reconnaissance d'images et a posé les bases de leur architecture actuelle.

La popularité des CNN a pris un tournant majeur en 2012 avec la victoire d'AlexNet (Krizhevsky et al., 2012) au concours ImageNet Large Scale Visual Recognition Challenge (ILSVRC). AlexNet a surpassé les modèles existants en termes de précision grâce à une architecture de CNN plus profonde et à l'utilisation de GPU pour accélérer l'entraînement. Cette victoire a marqué un moment décisif dans l'adoption et le développement des CNN pour diverses applications de vision par ordinateur.

#### 1.4.10.2 Architecture d'un CNN

Les réseaux neuronaux convolutifs sont typiquement caractérisés par deux blocs : une "backbone" qui permet d'extraire les caractéristiques importantes des données d'entrée, et un bloc composé de couches entièrement connectées (Fully Connected Layers) qui a pour but de résoudre une tâche spécifique à partir des cartes de caractéristiques en sortie de la backbone. Une backbone est composée de plusieurs types de couches distinctes, chacune ayant un rôle particulier :

— Couches Convolutives (Convolutional Layers)
— Couches de Pooling (Pooling Layers)

**Couches convolutives**

Les couches convolutives appliquent des filtres de convolution aux données d'entrée pour extraire des caractéristiques locales telles que les bords, les textures et les motifs. Cela produit en sortie des cartes de caractéristiques. Les opérations convolutives (Figure 8) peuvent être décrites de la manière suivante :

$$\text{Output}(i,j) = \sum_{c=0}^{C-1} \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} \text{Input}_c(i+m, j+n) \cdot \text{Filter}_c(m,n) + \text{Bias} \tag{1.17}$$

où :

— $\text{Input}_c(i,j)$ est la valeur du pixel du channel $c$ de l'image d'entrée à la position $(i,j)$.

— $\text{Filter}_c(m, n)$ est la valeur du filtre du channel $c$ à la position $(m, n)$.
— Bias est le biais ajouté à la convolution.
— $k$ est la taille du filtre, par exemple $k = 3$ pour un filtre $3 \times 3$.
— $C$ est le nombre de channels de l'image d'entrée (par exemple, 3 pour une image RGB)

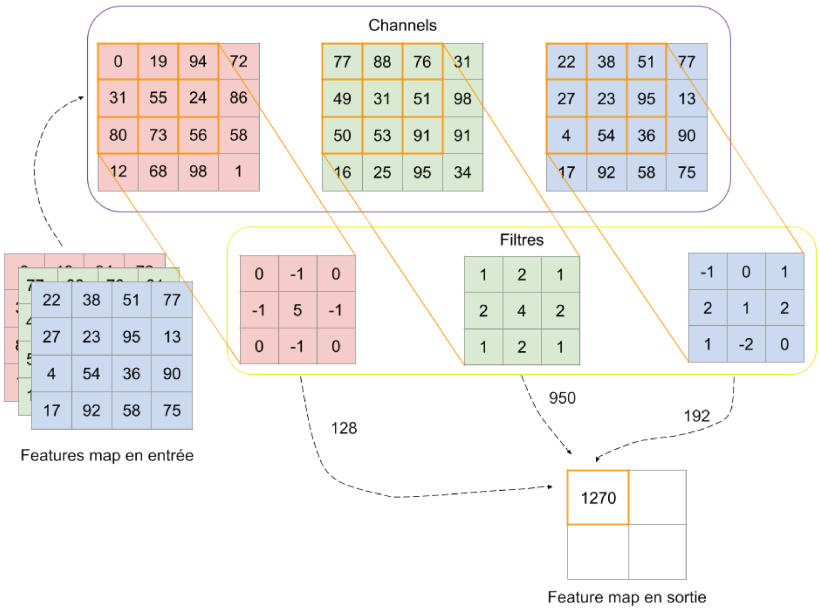


FIGURE 8 – Exemple d'opération convolutive.

Le principal avantage des réseaux convolutifs est l'utilisation de poids partagés à travers chaque pixel d'une carte de caractéristiques, ce qui réduit la complexité du modèle et la consommation de mémoire. Cela permet également d'améliorer les performances et de rendre le modèle invariant par translation. Cette invariance signifie que les caractéristiques détectées, telles que les bords et les motifs, resteront cohérentes, même si elles apparaissent à des endroits différents de l'image.

Les principaux paramètres des opérations de convolution sont la taille du filtre, le pas (stride) et le remplissage (padding), qui influencent directement la taille de la carte de caractéristiques en sortie :

— **Taille du Filtre (Kernel Size)** : Détermine la région locale de l'image sur laquelle la convolution est appliquée. Par exemple, un filtre de taille $3 \times 3$ signifie qu'il opère sur une région de 3 pixels par 3 pixels à chaque déplacement.
— **Pas (Stride)** : Contrôle le nombre de pixels par lesquels le filtre est déplacé sur l'image d'entrée. Un pas de 1 fait déplacer le filtre d'un pixel à chaque étape, tandis qu'un pas de 2 le fait avancer de deux pixels. Cela influence la taille de la carte de caractéristiques finale, selon la formule :

$$\text{Taille de la Sortie} = \frac{(\text{Taille de l'Entrée} - k)}{s} + 1 \tag{1.18}$$

— **Remplissage (Padding)** : Le remplissage consiste à ajouter des pixels autour des bords de l'image d'entrée. Il existe deux types principaux de padding :
  — **Padding valide (Valid Padding)** : Ne rajoute aucun pixel, ce qui peut réduire la taille de la carte de caractéristiques.
  — **Padding plein (Same Padding)** : Ajoute des pixels pour que la taille de la sortie soit identique à celle de l'entrée.

Cette approche rend les CNN bien adaptés pour la reconnaissance d'images, où la préservation des informations spatiales tout en réduisant la dimensionnalité est cruciale pour la performance du modèle.

### Couches de Pooling

Les couches de pooling (voir Figure 9) sont une composante essentielle des réseaux de neurones convolutifs (CNN) et jouent un rôle crucial dans la réduction de la dimensionnalité des cartes de caractéristiques tout en préservant les informations les plus importantes. Elles fonctionnent en appliquant une opération de pooling (généralement le max pooling ou l'average pooling) sur des sous-régions des cartes de caractéristiques produites par les couches convolutives. Leur objectif principal est de condenser l'information, de manière à simplifier le modèle sans perdre les caractéristiques critiques nécessaires pour effectuer des prédictions précises.

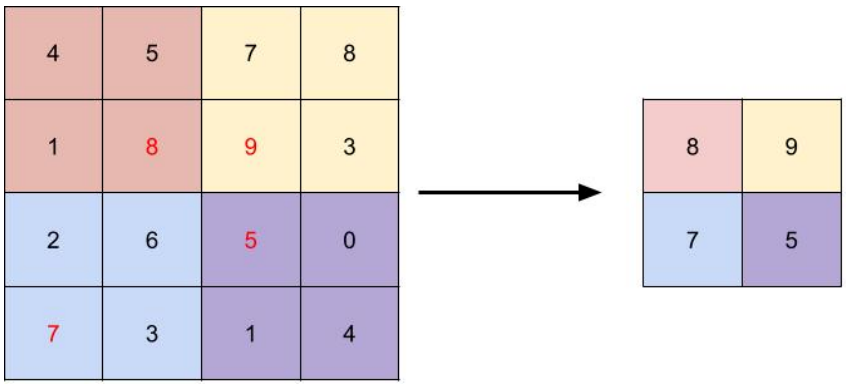


FIGURE 9 – Exemples de pooling en utilisant la fonction d'agrégation maximum.

Prenons l'exemple du max pooling. Cette opération parcourt les cartes de caractéristiques en utilisant une fenêtre de taille fixe (par exemple, $2\times2$) et, pour chaque sous-région, elle sélectionne la valeur maximale. Cela permet de ne retenir que l'information la plus saillante dans cette région, que ce soit un contour ou une texture. En conservant uniquement les valeurs maximales, le max pooling réduit la taille des cartes de caractéristiques de façon significative, ce qui diminue la charge computationnelle du modèle et facilite l'apprentissage de réseaux plus profonds. Cette réduction permet au modèle d'apprendre

des représentations plus abstraites des données tout en évitant une surcharge de calcul.

L'average pooling, quant à lui, prend la moyenne des valeurs dans chaque sous-région, ce qui fournit une approche différente de la condensation des informations. Plutôt que de se concentrer sur les éléments les plus dominants, il conserve une vue plus globale des caractéristiques présentes dans une région.

En plus de la réduction de la dimensionnalité, les couches de pooling introduisent des invariances importantes, en particulier l'invariance par translation. Cela signifie que si un objet dans l'image d'entrée se déplace légèrement, les valeurs extraites par le pooling resteront relativement stables, ce qui permet au modèle de mieux généraliser aux variations d'emplacement des objets. Cela améliore la robustesse du réseau aux petites transformations spatiales des données d'entrée.

De plus, en réduisant progressivement la taille des cartes de caractéristiques, les couches de pooling permettent aux réseaux convolutifs d'aller plus en profondeur. Chaque réduction de la taille des cartes de caractéristiques permet d'ajouter davantage de couches convolutives tout en maintenant un nombre raisonnable de paramètres. Cela signifie que le modèle peut capturer des caractéristiques de plus en plus complexes et abstraites à mesure que l'information passe à travers des couches plus profondes.

Les couches de pooling contribuent également à lutter contre le surapprentissage en réduisant la sensibilité du modèle aux variations mineures des données d'entraînement. En supprimant certains détails superflus tout en conservant les caractéristiques principales, ces couches améliorent la capacité du modèle à généraliser à de nouvelles données.

En résumé, les couches de pooling jouent un rôle fondamental dans la construction de réseaux de neurones convolutifs efficaces et robustes. Elles permettent de réduire la dimension des cartes de caractéristiques tout en préservant les informations essentielles, introduisent des invariances importantes comme l'invariance par translation, et facilitent la construction de réseaux plus profonds en limitant la complexité computationnelle. Ces couches permettent ainsi aux CNN de mieux généraliser, d'éviter le surapprentissage et de capturer des représentations de données de plus en plus abstraites et pertinentes.

#### Couches Entièrement Connectées

Les couches entièrement connectées, similaires aux perceptrons multicouches, sont composées de neurones où chacun est connecté à tous les neurones de la couche précédente et de la couche suivante. Elles sont généralement placées à la fin des réseaux convolutifs après les couches de convolution et de pooling. Le but des couches entièrement connectées est de recevoir les cartes de caractéristiques extraites par les couches convolutives et de les utiliser pour résoudre des tâches spécifiques, telles que la classification.

Dans une tâche de classification classique, les cartes de caractéristiques produites par les couches convolutives sont aplaties en un vecteur de taille fixe. Ce vecteur est ensuite passé à une ou plusieurs couches entièrement connectées, qui apprennent à combiner ces

caractéristiques pour effectuer des décisions finales. Une couche de sortie, souvent une couche softmax, transforme les activations de la dernière couche entièrement connectée en probabilités pour chaque classe.

Les couches entièrement connectées sont souvent utilisées parce qu'elles permettent d'effectuer des transformations non linéaires complexes sur les cartes de caractéristiques extraites. Elles sont particulièrement efficaces dans les tâches de classification où les relations entre les caractéristiques ne sont pas strictement spatiales mais nécessitent des combinaisons de haut niveau.

Cependant, cette approche a aussi des limites. Les couches entièrement connectées ajoutent un grand nombre de paramètres, augmentant ainsi le risque de surapprentissage et nécessitant plus de ressources pour l'entraînement. De plus, elles détruisent les informations spatiales capturées par les couches convolutives, car les données sont aplaties en un vecteur, perdant ainsi la relation spatiale originale entre les pixels.

Pour surmonter ces limitations, des réseaux entièrement convolutifs (Fully Convolutional Networks, FCN) ont été développés. Contrairement aux architectures classiques qui utilisent des couches entièrement connectées, les FCN remplacent ces couches par des couches convolutives, permettant ainsi de préserver les informations spatiales tout au long du réseau.

Dans un FCN, au lieu d'aplatir les cartes de caractéristiques à la fin du réseau, on utilise des convolutions finales pour réduire la dimensionnalité tout en maintenant la structure spatiale. Cette approche est particulièrement efficace pour des tâches comme la segmentation d'images, où chaque pixel de l'image doit être classé individuellement.

### 1.4.11 Augmentation de données

L'augmentation de données (Figure 10) est une technique essentielle pour entraîner les réseaux entièrement convolutifs de manière plus efficace. Elle consiste à appliquer des transformations aléatoires aux images d'entraînement, telles que des rotations, des recadrages ou des modifications d'échelle. Cela permet d'augmenter la diversité des exemples vus par le modèle, renforçant ainsi sa capacité de généralisation et améliorant sa précision sur les données de test, tout en réduisant le risque de surapprentissage.




FIGURE 10 – L'image montre des exemples d'augmentation de données appliqués à une image originale ("a"). Les augmentations incluent : un flip horizontal ("b") ; un flou gaussien ("c") ; et un crop aléatoire ("d") et une suppression de canal ("e"). Ces transformations augmentent la diversité des données d'entraînement, améliorant la généralisation et la robustesse du modèle.

L'avantage de cette technique lorsque l'on traite une faible quantité de données, est la réduction du risque de surapprentissage, puisque les modèles sont exposés à plus de variations, ils auront tendance à moins mémoriser l'ensemble de données. Cela devient essentiel quand les données pour entraîner un modèle deviennent limitées, ce qui est un problème courant dans les applications d'apprentissage profond. Un autre avantage est la robustesse qu'elle confère aux modèles. En appliquant des variations, elle rend les modèles plus fiables sur une plus large gamme de données d'entrée. Enfin, elle permet également de gagner du temps en réduisant la nécessité de collecter un ensemble de données plus important. Diverses transformations peuvent être appliquées pour créer de nouveaux échantillons. Dans le cas des images, des modifications simples peuvent être appliquées, telles que le retournement ou la rotation, et des modifications plus complexes, telles que la distorsion des couleurs ou le recadrage aléatoire. Utilisée judicieusement, la data augmentation contribue à la création de modèles plus performants et plus résistants au surapprentissage.

### 1.4.12 Apprentissage par transfert

Dans les réseaux convolutifs (CNN), une technique couramment utilisée pour améliorer la généralisation est l'apprentissage par transfert ou "transfer learning" (BOZINOVSKI & FULGOSI, 1976). Cette approche consiste à réutiliser un modèle pré-entraîné sur une tâche connexe comme point de départ pour une nouvelle tâche. Cela permet de profiter des connaissances déjà acquises lors de la première tâche pour améliorer les performances sur la seconde, tout en réduisant le besoin de données d'entraînement et en accélérant le processus d'apprentissage.

L'apprentissage par transfert est particulièrement efficace pour les CNN car il permet d'exploiter des modèles pré-entraînés sur de larges ensembles de données, comme ImageNet, pour des tâches plus spécifiques avec des données limitées. Cela améliore la généralisation, réduit le temps d'entraînement et nécessite moins de données spécifiques

à la tâche cible.

Les deux principales approches en apprentissage par transfert sont le fine-tuning et l'extraction de caractéristiques :

— **Fine-tuning** : Il consiste à reprendre un modèle pré-entraîné et à continuer son entraînement sur la nouvelle tâche. Typiquement, les premières couches du modèle, qui capturent des caractéristiques générales comme des contours et des textures, sont "gelées" (leurs poids ne sont pas mis à jour), et seules les dernières couches spécialisées sont ajustées. Toutefois, une autre approche consiste à ne pas geler les premières couches mais à les entraîner avec un taux d'apprentissage très faible, ce qui permet d'ajuster légèrement les poids tout en conservant les connaissances générales acquises précédemment.

— **Extraction de caractéristiques** : Cette méthode consiste à utiliser les couches d'un modèle pré-entraîné comme extracteurs de caractéristiques. On garde ces couches fixes et on ajoute de nouvelles couches pour la tâche cible.

Par exemple, pour classifier des images de fleurs, on peut utiliser un modèle pré-entraîné sur ImageNet comme VGG16 ou ResNet. On enlève la couche de sortie, puis on ajoute une nouvelle couche adaptée à la classification des fleurs. Ensuite, on entraîne ces nouvelles couches tout en gardant les premières couches gelées. Cela permet au modèle de réutiliser les caractéristiques générales apprises sur ImageNet, tout en s'adaptant à la nouvelle tâche de classification des fleurs.

L'apprentissage par transfert est particulièrement adapté aux tâches de vision par ordinateur, comme la classification d'images ou la détection d'objets. Par exemple, des modèles tels que VGG, ResNet, et Inception, pré-entraînés sur des ensembles de données massifs comme ImageNet, apprennent à reconnaître des caractéristiques visuelles générales comme les textures, les bords, et les formes. En transférant ces connaissances vers des tâches plus spécifiques (par exemple, la reconnaissance d'espèces de fleurs ou d'animaux), le modèle bénéficie de cette base solide, ce qui lui permet de généraliser plus efficacement et d'obtenir de meilleures performances avec moins de données.

L'apprentissage par transfert constitue ainsi un outil puissant pour les CNN, permettant de tirer parti des connaissances acquises sur de grandes bases de données pour adapter rapidement et efficacement un modèle à de nouvelles applications avec un minimum de ressources d'entraînement.

### 1.4.13 Architectures importantes

— **LeNet-5 (1998) :** LeNet-5 (LeCun et al., 1998), conçu par Yann LeCun, est l'un des premiers CNN. Il a introduit les couches de convolution et de pooling suivies de couches entièrement connectées, pionnier pour la reconnaissance de chiffres manuscrits. C'est l'architecture fondatrice des CNN modernes.

— **AlexNet (2012) :** AlexNet (Krizhevsky et al., 2012) par Krizhevsky et al. a popularisé les réseaux neuronaux profonds avec 5 couches convolutives et 3 entièrement connectées. Il a introduit ReLU pour accélérer l'entraînement et le dropout pour réduire le surapprentissage. Il a gagné la compétition ImageNet en 2012, initiant l'usage des GPU pour l'apprentissage en profondeur.

— **GoogLeNet/Inception (2014) :** GoogLeNet (Szegedy et al., 2015), aussi appelé Inception, a introduit les Inception modules, permettant l'extraction simultanée de multiples types de caractéristiques tout en réduisant le nombre de paramètres grâce à son efficacité en calcul.

— **VGG16 (2014) :** VGG16 (Simonyan & Zisserman, 2014) s'est démarqué avec l'utilisation de petites convolutions $3 \times 3$ empilées et une profondeur accrue, démontrant que des réseaux profonds et simples peuvent améliorer les performances de classification des images tout en étant faciles à généraliser.

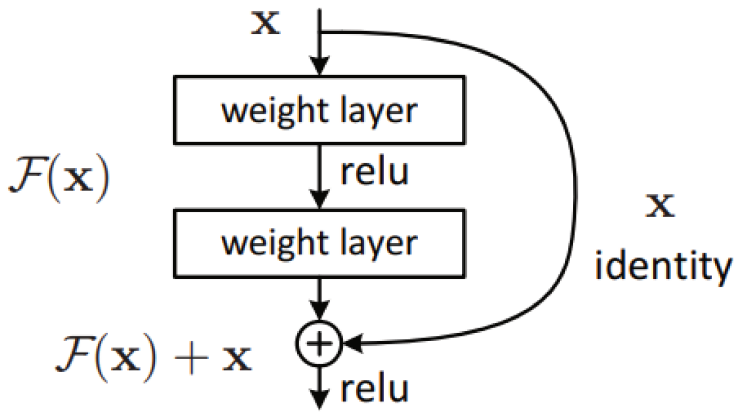


Figure 11 – Bloque résiduel (He et al., 2016)

— **ResNet (2015) :** ResNet (He et al., 2016) a introduit les blocs résiduels (Figure 11), résolvant le problème du gradient vanishing et permettant l'entraînement de réseaux extrêmement profonds (ResNet-50, ResNet-101). Cette innovation a permis d'améliorer les performances sur les grandes tâches de classification et de détection d'objets.

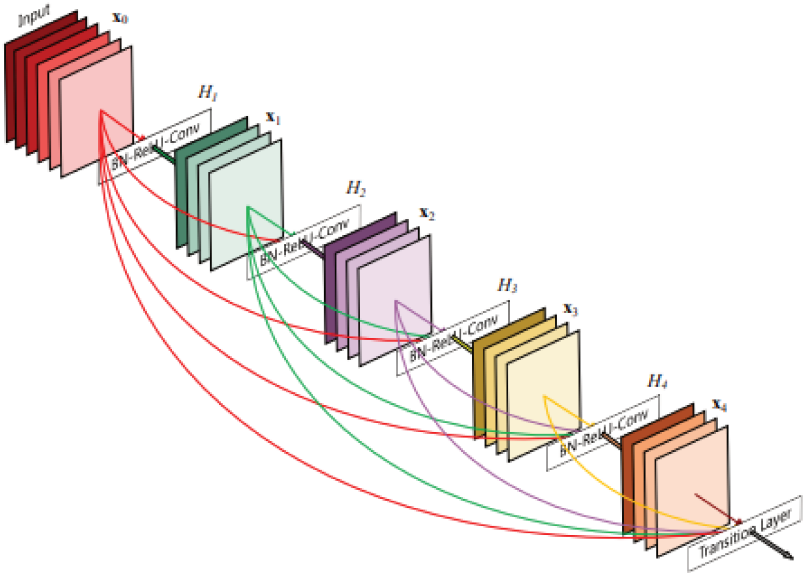


FIGURE 12 – Architecture de densenet (HUANG et al., 2017)

— **DenseNet (2017) :** DenseNet (HUANG et al., 2017) a proposé des connexions denses, où chaque couche reçoit les caractéristiques de toutes les couches précédentes (Figure 12). Cela a amélioré l'efficacité du flux de gradients et réduit le nombre de paramètres nécessaires, tout en améliorant les performances globales du réseau.

— **Vision Transformer (ViT, 2020) :** ViT (DOSOVITSKIY et al., 2020) a bouleversé les architectures CNN en prouvant que les transformeurs, originellement utilisés en traitement du langage naturel, peuvent surpasser les CNN sur les tâches de vision en traitant les images comme des séquences de patchs et en remplaçant les convolutions par des mécanismes d'attention.

## 1.5 Détection d'objets

La détection d'objets est une tâche fondamentale dans le domaine de la vision par ordinateur qui combine les aspects de la classification d'images et de la localisation d'objets. Contrairement à la classification d'images, qui assigne une étiquette à une image entière, la détection d'objets vise à identifier et localiser simultanément plusieurs objets d'intérêt dans une image. Cela implique de situer les objets détectés et de les classer en catégories pré-définies.

### 1.5.1 Les premières méthodes de détection d'objets

Avant l'avènement des techniques de deep learning, la détection d'objets reposait sur des méthodes comme SIFT (Scale-Invariant Feature Transform), HOG (Histogram of

Oriented Gradients) et les détecteurs de Viola-Jones. SIFT (Lowe, 1999), développé par David Lowe en 1999, identifie des points d'intérêt dans les images en utilisant des différences de gaussiennes pour être invariant aux transformations d'échelle, de rotation et aux changements d'éclairage. Un objet est reconnu dans une nouvelle image en comparant individuellement chaque caractéristique de la nouvelle image à cette base de données et en trouvant des caractéristiques candidates correspondantes sur la base de la distance euclidienne de leurs vecteurs de caractéristiques. HOG (Dalal & Triggs, 2005), proposé par Navneet Dalal et Bill Triggs en 2005, caractérise l'apparence et la forme des objets dans les images à travers des histogrammes de gradients orientés, ce qui le rend efficace pour des tâches comme la détection de piétons ou la détection d'humains. Le détecteur de Viola-Jones (Viola & Jones, 2001), développé en 2001 par Paul Viola et Michael Jones, a permis la détection d'objets en temps réel grâce à l'utilisation de caractéristiques de Haar et d'une structure de classifieurs en cascade, initialement appliquée à la détection de visages. Ces méthodes, basées sur des caractéristiques faites à la main et des techniques d'apprentissage automatique traditionnelles, offraient des solutions robustes et efficaces pour diverses tâches de vision par ordinateur. Bien que surpassées par le deep learning en termes de performance et de flexibilité, SIFT, HOG et les détecteurs de Viola-Jones restent des contributions fondamentales dans le domaine, comme références dans la recherche en vision par ordinateur. Certains algorithmes comme les détecteurs de Viola-Jones sont toujours utilisés dans certain cas très spécifiques où la puissance de calcul est très limité, car il est beaucoup plus simple d'utiliser cet algorithme qu'un modèle neuronal convolutif avec des millions de paramètres.

### 1.5.2 Les approches basés sur les réseaux neuronaux

Avec l'avènement des réseaux de neurones profonds, la détection d'objets a connu une transformation radicale. Les approches neuronales ont surpassé les méthodes traditionnelles basées sur des caractéristiques faites à la main, comme SIFT, HOG et les détecteurs de Viola-Jones, en termes de précision et de flexibilité. Les réseaux de neurones convolutifs (CNN) sont au cœur de cette révolution, en exploitant leur capacité à apprendre des représentations hiérarchiques et complexes directement à partir des données d'image. La détection d'objets a évolué pour inclure deux types principaux de détecteurs : les détecteurs en une étape et les détecteurs en deux étapes. Ces deux approches diffèrent principalement par leur architecture et leur manière de traiter la détection d'objets dans une image, offrant chacune des avantages spécifiques en termes de précision et de vitesse.

#### 1.5.2.1 Détection en deux étapes

Les détecteurs en deux étapes sont une classe de modèles de détection d'objets qui divisent le processus de détection en deux phases distinctes : la génération de proposi-

tions de régions et la classification des objets dans ces régions. Cette approche permet d'atteindre une précision élevée en affinant les propositions d'objets avant la classification finale. Les détecteurs en deux étapes les plus influents sont R-CNN (Region-based Convolutional Neural Networks) et ses variantes : Fast R-CNN, Faster R-CNN et Mask R-CNN.

R-CNN (Girshick et al., 2014) (Figure 13) est une architecture de détection d'objets qui a marqué une avancée significative en combinant des techniques de sélection de régions et des réseaux de neurones convolutionnels (CNN). L'architecture de R-CNN repose sur deux étapes principales : la génération de propositions de régions et la classification de ces régions.

**R-CNN (Region-based Convolutional Neural Networks)**

Le processus de détection d'objets avec R-CNN commence par la génération de propositions de régions potentielles dans l'image, appelées "region proposals". Cette étape est réalisée à l'aide de méthodes comme Selective Search (Uijlings et al., 2013), qui propose environ 2000 régions par image. Ces propositions sont des boîtes englobantes qui contiennent potentiellement des objets.

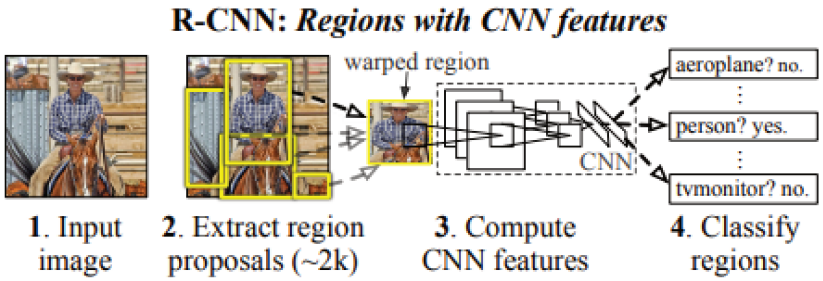


Figure 13 – Architecture R-CNN (Huang et al., 2017)

Chaque région proposée est ensuite redimensionnée à une taille fixe et passée à travers un réseau CNN pré-entraîné, tel que AlexNet ou VGG16, pour extraire des caractéristiques. Ces caractéristiques sont ensuite utilisées par des classificateurs SVM binaire (Support Vector Machine) (Cortes & Vapnik, 1995) chacun entraîné sur une classe. Pour affiner les boîtes englobantes prédites, R-CNN utilise un modèle de régression qui ajuste les coordonnées des boîtes englobantes initiales vers les coordonnées des boîtes englobantes réelles. Les transformations sont définies comme suit :

$$\begin{aligned} \hat{g}_x &= p_w d_x(\mathbf{p}) + p_x \\ \hat{g}_y &= p_h d_y(\mathbf{p}) + p_y \\ \hat{g}_w &= p_w \exp(d_w(\mathbf{p})) \\ \hat{g}_h &= p_h \exp(d_h(\mathbf{p})) \end{aligned} \tag{1.19}$$

où $\hat{g}_x, \hat{g}_y, \hat{g}_w, \hat{g}_h$ sont les coordonnées corrigées de la boîte englobante prédites, $p_x, p_y, p_w, p_h$ sont les coordonnées de la proposition initiale, et $d_x(\mathbf{p}), d_y(\mathbf{p}), d_w(\mathbf{p}), d_h(\mathbf{p})$ sont les décalages prédits par le modèle.

Les cibles de régression $t_x, t_y, t_w, t_h$ sont calculées comme suit :

$$\begin{aligned} t_x &= (g_x - p_x)/p_w \\ t_y &= (g_y - p_y)/p_h \\ t_w &= \log(g_w/p_w) \\ t_h &= \log(g_h/p_h) \end{aligned} \tag{1.20}$$

où $g_x, g_y, g_w, g_h$ sont les coordonnées de la boîte englobante réelle (ground truth).

La perte de régression $\mathcal{L}_{\text{reg}}$ est définie comme :

$$\mathcal{L}_{\text{reg}} = \sum_{i \in \{x,y,w,h\}} (t_i - d_i(\mathbf{p}))^2 + \lambda \|\mathbf{w}\|^2 \tag{1.21}$$

Cette fonction permet de mesurer la différence entre les coordonnées des boîtes englobantes prédites et réelles, tout en ajoutant une régularisation pour éviter le surapprentissage. Bien que R-CNN ait considérablement amélioré la précision de la détection d'objets par rapport aux méthodes précédentes, elle présente des limitations en termes de temps de calcul et de consommation de mémoire. Le traitement de chaque proposition de région de manière indépendante est coûteux en temps, et le besoin de redimensionner et de traiter chaque région par le CNN individuellement entraîne une consommation élevée de ressources. R-CNN a marqué une étape importante dans le domaine de la détection d'objets en combinant des techniques de génération de régions avec des réseaux de neurones convolutionnels. Bien qu'elle présente des limitations en termes d'efficacité, son approche novatrice a jeté les bases pour des architectures plus rapides et plus efficaces qui ont suivi.

**Fast R-CNN**

Fast R-CNN (Girshick, 2015) (Figure 14), a apporté plusieurs améliorations significatives par rapport à l'architecture originale de R-CNN. Ces améliorations visent à résoudre les limitations de R-CNN en termes de temps de calcul et de consommation de mémoire.

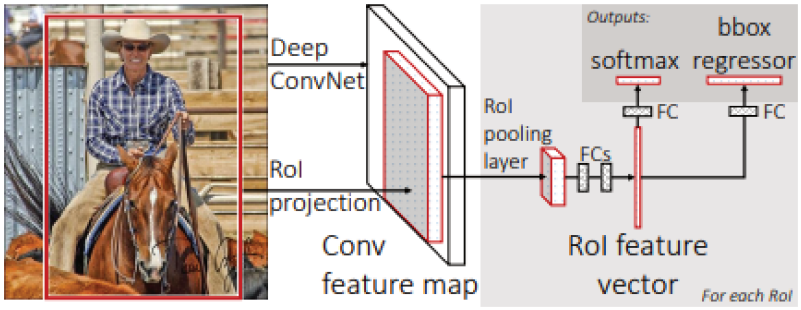


FIGURE 14 – Architecture Fast R-CNN (GIRSHICK, 2015)

Contrairement à R-CNN, où chaque région proposée est redimensionnée et passée individuellement à travers un réseau CNN, Fast R-CNN traite l'image entière en une seule passe à travers le réseau CNN. Les régions d'intérêt (RoIs) sont ensuite extraites des caractéristiques de la couche de convolution. Fast R-CNN introduit la couche RoI Pooling, qui extrait des caractéristiques de taille fixe pour chaque région d'intérêt à partir des cartes de caractéristiques produites par le CNN. Cette étape permet de conserver la taille des RoIs indépendamment de leur taille d'origine, facilitant le passage à travers des couches fully connected qui font la classification. Dans R-CNN, le réseau CNN est pré-entraîné sur une tâche de classification d'image, puis utilisé comme un extracteur de caractéristiques. Fast R-CNN unifie cette étape en permettant l'entraînement du CNN et du classificateur de région en une seule étape avec backpropagation. Cela améliore l'efficacité de l'entraînement et la performance du modèle. Fast R-CNN optimise simultanément la classification et la régression des boîtes englobantes. La fonction de perte totale est une combinaison de la perte de classification et de la perte de régression :

$$L = L_{\text{cls}} + \lambda L_{\text{reg}} \tag{1.22}$$

La perte de classification $L_{\text{cls}}$ est une perte de type softmax qui mesure la divergence entre la classe prédite par le modèle et la classe réelle de l'objet. Elle est définie comme suit :

$$L_{\text{cls}} = -\log p_u \tag{1.23}$$

où $p_u$ est la probabilité prédite pour la classe correcte $u$.

La perte de régression $L_{\text{reg}}$ est utilisée pour affiner les boîtes englobantes et minimise les erreurs entre les boîtes prédites et les boîtes réelles. Elle utilise la fonction de perte Huber (Smooth L1 Loss) (HUBER, 1992), qui est moins sensible aux outliers que la perte L2 :

$$L_{\text{reg}} = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L1}(t_i - d_i(\mathbf{p})) \tag{1.24}$$

En traitant l'image entière une seule fois à travers le réseau CNN et en appliquant la couche RoI Pooling, Fast R-CNN réduit considérablement le temps de calcul et la mémoire nécessaire par rapport à R-CNN, où chaque région proposée est traitée indépendamment. Fast R-CNN améliore l'efficacité et les performances de la détection d'objets en intégrant le traitement des régions d'intérêt dans un pipeline unique, en introduisant la couche RoI Pooling et en permettant un entraînement de bout en bout avec backpropagation. Ces innovations rendent le modèle plus rapide et moins gourmand en ressources par rapport à R-CNN, tout en offrant une meilleure précision de détection.

**Faster R-CNN**

Faster R-CNN (Ren et al., 2015) (Figure 15, apporte des améliorations significatives par rapport à Fast R-CNN. L'objectif principal de Faster R-CNN est de résoudre les inefficacités liées à la génération de propositions de régions en introduisant un réseau entièrement convolutionnel intégré pour cette tâche.

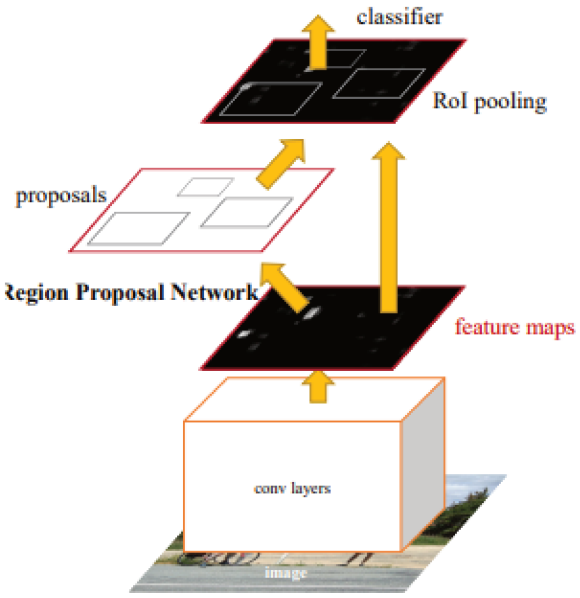


Figure 15 – Architecture Faster R-CNN (Ren et al., 2015)

L'une des principales innovations de Faster R-CNN est l'introduction du Region Proposal Network (RPN). Le RPN est un réseau de neurones entièrement convolutionnel qui partage les cartes de caractéristiques avec le réseau de détection d'objets. Il génère des propositions de régions en glissant une petite fenêtre sur les cartes de caractéristiques et en prédictant des boîtes englobantes pour chaque position. Cette approche permet de générer des propositions de régions beaucoup plus rapidement et de manière plus efficace que les méthodes de sélection de régions précédentes, telles que Selective Search. Les propositions de régions générées par le RPN sont ensuite passées à travers une couche RoI Pooling, comme dans Fast R-CNN, pour extraire des caractéristiques de taille fixe

pour chaque région d'intérêt. Ces caractéristiques sont ensuite utilisées par le réseau de détection d'objets pour classifier les objets et affiner les boîtes englobantes. La fonction de perte du RPN dans Faster R-CNN est définie comme suit :

$$\mathcal{L}_{\text{RPN}} = \frac{1}{N_{\text{cls}}} \sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{reg}}} \sum_i p_i^* \cdot \text{smooth}_{L1}(t_i - t_i^*) \tag{1.25}$$

où

— $p_i$ est la probabilité prédite pour la classe $i$.
— $p_i^*$ est la classe réelle.
— $t_i$ représente les coordonnées des boîtes englobantes prédites.
— $t_i^*$ représente les coordonnées réelles des boîtes englobantes
— $\mathcal{L}_{\text{cls}}$ est la perte de classification softmax.
— $\text{smooth}_{L1}$ est la perte Huber (Smooth L1 Loss) pour la régression des boîtes englobantes.
— $N_{\text{cls}}$ est le nombre de classes.
— $N_{\text{reg}}$ est le nombre de boîtes englobantes.

En intégrant le RPN directement dans le pipeline de détection d'objets, Faster R-CNN améliore considérablement l'efficacité et la rapidité de la génération de propositions de régions. Cela permet au modèle de traiter des images à une vitesse beaucoup plus élevée que les méthodes précédentes tout en maintenant une haute précision de détection. En résumé, Faster R-CNN améliore Fast R-CNN en intégrant un réseau de propositions de régions entièrement convolutionnel (RPN), permettant une génération de propositions plus rapide et plus efficace. Cette innovation, combinée à un entraînement conjoint du RPN et du réseau de détection d'objets, rend Faster R-CNN plus rapide et plus précis, tout en réduisant la complexité et la redondance des calculs.

### Mask R-CNN

Mask R-CNN (He et al., 2017) (Figure 16) est une extension de Faster R-CNN qui introduit la capacité de segmenter les objets détectés en plus de les localiser. Elle a apporté plusieurs améliorations significatives par rapport à Faster R-CNN.

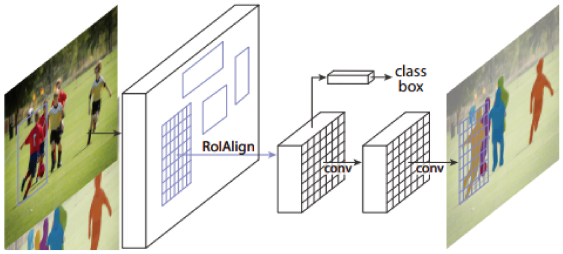


FIGURE 16 – Architecture Mask R-CNN (He et al., 2017)

Tout d'abord, Mask R-CNN conserve l'architecture de base de Faster R-CNN, y compris le Region Proposal Network (RPN) pour la génération de propositions de régions et le réseau de détection d'objets pour classifier et raffiner les boîtes englobantes. Ainsi, il bénéficie des améliorations de vitesse et de précision introduites par Faster R-CNN. La principale innovation de Mask R-CNN réside dans l'introduction d'un module de segmentation de masques. Ce module utilise les caractéristiques extraites par les propositions de régions pour prédire un masque binaire précis pour chaque région d'intérêt (RoI). Cela permet non seulement de localiser les objets dans une image mais aussi de segmenter précisément les régions correspondant à chaque instance d'objet. La perte de segmentation $\mathcal{L}_{\text{mask}}$ évalue la précision de la prédiction des masques de segmentation pour chaque région d'intérêt (RoI). Elle utilise une perte de log-vraisemblance négative (Negative Log Likelihood Loss) entre les masques prédits $m_i$ et les masques binaires réels $m_i^*$. Elle est calculée comme :

$$\mathcal{L}_{\text{mask}}(m_i, m_i^*) = -\sum_j m_i^* \log(m_i) + (1 - m_i^*) \log(1 - m_i) \tag{1.26}$$

Ici, $m_i$ représente la probabilité prédite pour chaque pixel d'appartenir à l'objet (masque prédit), et $m_i^*$ est le masque binaire réel pour l'objet dans la RoI.

La perte totale $\mathcal{L}$ de Mask R-CNN est une combinaison pondérée de la perte de classification et de la perte de segmentation :

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{mask}} \tag{1.27}$$

- $\mathcal{L}_{\text{cls}}$ : Perte de classification, qui évalue la précision des prédictions de classe pour chaque RoI.
- $\mathcal{L}_{\text{box}}$ : Perte de détection d'objets, qui mesure la précision des prédictions de chaque boîte englobante.
- $\mathcal{L}_{\text{mask}}$ : Perte de segmentation, qui mesure la précision des prédictions de masque pour chaque RoI.

En intégrant la perte de segmentation dans l'architecture de détection d'objets, Mask R-CNN permet non seulement de localiser et de classifier les objets, mais aussi de segmenter précisément les régions d'intérêt dans les images. Cette approche enrichit les capacités des modèles de vision par ordinateur en offrant une détection et une segmentation simultanées, ouvrant la voie à des applications avancées comme la segmentation sémantique et la compréhension fine des objets dans les scènes visuelles.

#### 1.5.2.2 Détection en une étapes

Les détecteurs en une étape sont une classe de modèles de détection d'objets qui combinent les phases de proposition de régions et de classification en une seule étape

de traitement. Cette approche permet une détection rapide et efficace, ce qui est particulièrement utile pour les applications nécessitant une réponse en temps réel. Les deux détecteurs en une étape les plus connus et les plus largement utilisés sont YOLO (You Only Look Once) et SSD (Single Shot MultiBox Detector).

**YOLO (You Only Look Once)**

YOLO (You Only Look Once) (Redmon et al., 2016) (Figure 17) est une approche de détection d'objets qui unifie la détection et la classification en une seule étape de traitement.

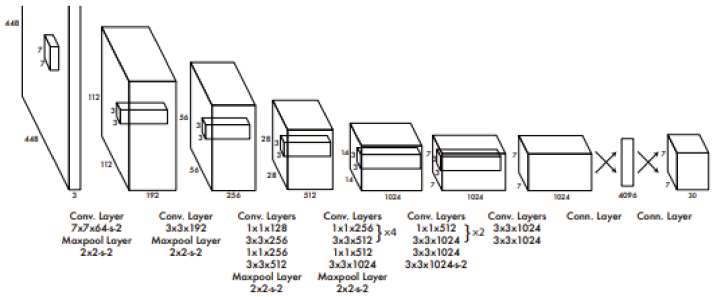


Figure 17 – Architecture Yolo (Redmon et al., 2016)

L'image d'entrée est divisée en une grille $S \times S$. Chaque cellule de la grille est responsable de prédire des boîtes englobantes et des probabilités de classe pour les objets dont le centre tombe dans cette cellule. Si nous avons une image d'entrée de taille $W \times H$, chaque cellule de la grille couvre une région de $\frac{W}{S} \times \frac{H}{S}$.

Pour chaque cellule de la grille, YOLO fait $B$ prédictions de boîtes englobantes. Chaque boîte englobante est définie par quatre valeurs : les coordonnées du centre $(x, y)$, la largeur $w$, et la hauteur $h$, ainsi qu'un score de confiance $C$. En plus de cela, chaque cellule prédit $P$ probabilités de classe. Ainsi, pour chaque cellule, les prédictions peuvent être résumées comme suit :

$$\mathbf{p}_i = \{x_i, y_i, w_i, h_i, C_i, \mathbf{c}_i\}$$

où $\mathbf{c}_i$ est un vecteur de probabilités de classe de longueur $P$.

Le score de confiance pour chaque boîte englobante $i$ est défini comme suit :

$$C_i = \Pr(\text{Objet présent}) \times \text{IOU}$$

où Pr(Objet présent) est la probabilité qu'un objet soit présent dans la boîte englobante, et IOU (Intersection over Union) est une mesure de la précision de la boîte englobante prédite par rapport à la boîte englobante réelle. L'IOU est calculée comme suit :

$$\text{IOU} = \frac{\text{Aire de l'intersection}}{\text{Aire de l'union}}$$

La fonction de perte de YOLO est conçue pour entraîner le réseau à minimiser les erreurs de prédiction des coordonnées des boîtes englobantes, des scores de confiance, et des probabilités de classe. La fonction de perte globale $\mathcal{L}$ est composée de plusieurs termes :

$$\begin{aligned}\mathcal{L} = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \tag{1.28}$$

où :

— $\mathbb{1}_{ij}^{\text{obj}}$ est un indicateur binaire valant 1 si l'objet $j$ est présent dans la cellule $i$, sinon 0.

— $\mathbb{1}_{ij}^{\text{noobj}}$ est un indicateur binaire valant 1 si aucun objet n'est présent dans la cellule $i$, sinon 0.

— $\lambda_{\text{coord}}$ et $\lambda_{\text{noobj}}$ sont des poids d'ajustement pour équilibrer les différents termes de la perte.

— $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$ sont les coordonnées réelles des boîtes englobantes.

— $\hat{C}_i$ est le score de confiance réel.

— $\hat{p}_i(c)$ est la probabilité réelle de la classe $c$.

En résumé, YOLO transforme la détection d'objets en un problème de régression unique, prédisant directement les coordonnées des boîtes englobantes et les probabilités de classe en une seule étape. Cette approche offre une grande rapidité, bien que la précision puisse être inférieure à celle des méthodes en deux étapes pour les objets de petite taille ou dans des scènes complexes.

**YOLOv3**

YOLOv3 (You Only Look Once version 3) (Redmon & Farhadi, 2018) (Figure 18) représente une série d'améliorations par rapport aux versions précédentes de YOLO, visant à augmenter la précision et la vitesse de la détection d'objets. Une des principales améliorations est l'utilisation de Darknet-53 comme backbone, une architecture de réseau convolutif plus profonde et plus puissante que Darknet-19 utilisé dans YOLOv2 (Redmon

& Farhadi, 2017).

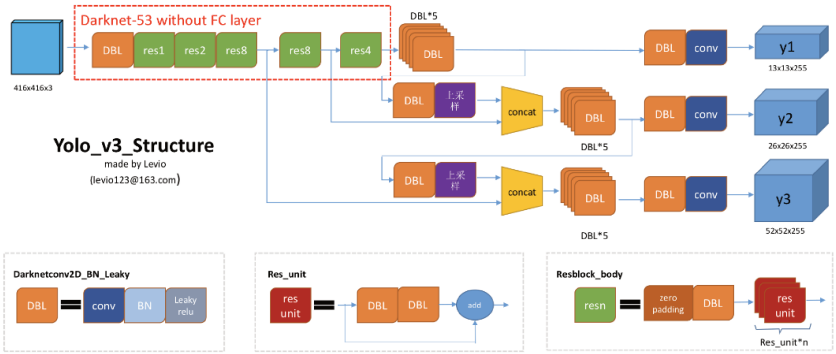


Figure 18 – Architecture Yolov3. Source de l'image [5]

Darknet-53 se compose de 53 couches convolutionnelles et utilise des connexions résiduelles (residual connections) inspirées de ResNet. Ces connexions résiduelles facilitent la propagation du gradient et améliorent la convergence du modèle, ce qui conduit à de meilleures performances sur des tâches complexes. YOLOv3 améliore également la détection des objets de différentes tailles en introduisant des prédictions à plusieurs échelles. Le réseau produit des prédictions à trois échelles différentes : large, moyenne et petite. Ces prédictions multi-échelles sont obtenues en utilisant des couches de convolution et des couches de upsampling (échantillonnage supérieur) pour fusionner les informations des caractéristiques de différentes résolutions, permettant ainsi de détecter efficacement les objets petits, moyens et grands. YOLOv3 utilise la fonction d'activation Leaky ReLU (Leaky ReLU$(x) = \max(0.1x, x)$) dans toutes ses couches convolutionnelles, sauf la dernière. Cela permet d'éviter le problème de mort des neurones (dead neurons) rencontré avec ReLU, tout en maintenant les avantages de la non-linéarité. Pour la prédiction des classes, YOLOv3 adopte une approche binaire en utilisant une logique de classification indépendante pour chaque classe. Plutôt que d'utiliser une seule couche softmax pour toutes les classes, YOLOv3 utilise des sigmoid pour chaque classe individuelle, ce qui est particulièrement avantageux pour les tâches de détection multi-label où un objet peut appartenir à plusieurs classes simultanément. Enfin, YOLOv3 bénéficie également d'améliorations dans les techniques d'entraînement et d'initialisation des poids. Le modèle est pré-entraîné sur ImageNet, permettant une meilleure généralisation et des performances accrues lorsqu'il est adapté à des tâches spécifiques de détection d'objets. Des techniques comme la normalisation par lot (batch normalization) sont également utilisées pour stabiliser et accélérer l'entraînement. En résumé, YOLOv3 introduit des améliorations significatives en termes de profondeur de l'architecture, de gestion multi-échelles, d'utilisation des ancrages, de fonctions d'activation, de méthodes de prédiction des classes et de techniques

5. https ://sheng-fang.github.io/2020-04-29-implement$_y$olo/

d'entraînement, permettant d'atteindre un équilibre optimal entre rapidité et précision pour les applications de détection d'objets en temps réel.

**YOLOv5**

YOLOv5 (You Only Look Once version 5) (Figure 19) représente une nouvelle étape dans l'évolution des modèles de détection d'objets, apportant des améliorations notables par rapport à ses prédécesseurs en termes de précision, de vitesse et de facilité d'utilisation. L'une des améliorations majeures de YOLOv5 est son architecture plus légère et plus optimisée. Contrairement à YOLOv3 qui utilise Darknet-53, YOLOv5 utilise une architecture de réseau convolutionnel développée spécifiquement pour maximiser l'efficacité et les performances sur les plateformes modernes de deep learning. Cela inclut des optimisations dans la structure du réseau, comme l'utilisation de couches CSP (Cross Stage Partial) qui réduisent le nombre de paramètres tout en maintenant ou en améliorant la précision.

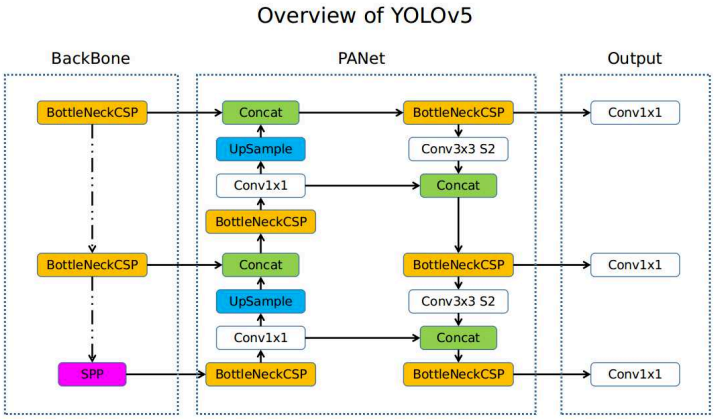


FIGURE 19 – Architecture Yolov5. Source de l'image [6]

YOLOv5 améliore également les prédictions à plusieurs échelles en affinant davantage la fusion des caractéristiques des différentes résolutions. Le réseau produit des prédictions à trois échelles différentes : petite, moyenne et grande, permettant de détecter efficacement des objets de diverses tailles avec une meilleure précision. Cette approche multi-échelle est renforcée par des améliorations dans les mécanismes d'upsampling et de downsampling, ce qui permet une meilleure intégration des informations contextuelles à différentes échelles. Un autre aspect clé de YOLOv5 est l'amélioration de l'utilisation des ancrages (anchors). YOLOv5 introduit un mécanisme d'auto-ajustement des ancrages (auto-anchor) qui permet au modèle de déterminer automatiquement les meilleures tailles d'ancrages en fonction des données d'entraînement. Cette optimisation des ancrages permet d'améliorer la précision de détection en ajustant les boîtes englobantes aux caractéristiques spécifiques

6. https ://github.com/ultralytics/yolov5/issues/280

des objets présents dans les données. En résumé, YOLOv5 introduit des améliorations significatives par rapport à ses prédécesseurs en optimisant l'architecture du réseau, en améliorant les mécanismes de prédiction multi-échelle, et en ajustant automatiquement les ancrages. Ces améliorations permettent à YOLOv5 d'atteindre un équilibre optimal entre rapidité et précision.

**Single Shot MultiBox Detector (SSD)**

Le Single Shot MultiBox Detector (SSD) (Liu et al., 2016) (Figure 20) est une architecture de détection d'objets conçue pour offrir à la fois une haute précision et une rapidité d'exécution, ce qui la rend particulièrement adaptée aux applications en temps réel.

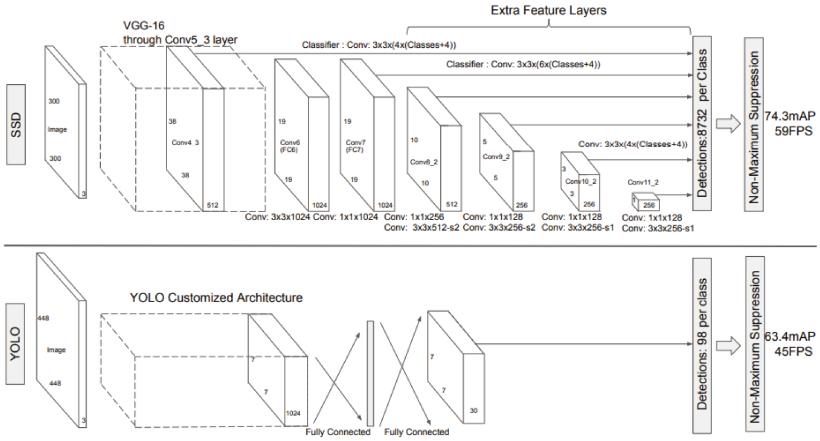


Figure 20 – Architecture ssd (Liu et al., 2016)

L'architecture de SSD repose sur un réseau de base, souvent un modèle pré-entraîné sur une tâche de classification d'image, comme VGG16. Sur ce réseau de base, SSD ajoute plusieurs couches convolutionnelles supplémentaires qui permettent de détecter des objets à différentes échelles. SSD est un des premier réseaux à utiliser des cartes de caractéristiques (feature maps) de différentes résolutions pour détecter des objets de tailles variées. Les cartes de caractéristiques de haute résolution sont utilisées pour détecter les petits objets, tandis que celles de basse résolution détectent les objets plus grands. Chaque cellule d'une carte de caractéristiques est associée à plusieurs "default boxes" (ou "prior boxes") de différentes tailles et proportions, qui servent de boîtes de référence pour la détection. Le processus de prédiction dans SSD est effectué en une seule étape grâce à l'utilisation de convolutions. Pour chaque cellule de la carte de caractéristiques, SSD prédit les décalages par rapport aux boîtes par défaut ainsi que les scores de classification pour chaque classe. En d'autres termes, chaque cellule de la carte de caractéristiques émet des prédictions pour la localisation des boîtes englobantes (bounding boxes) et les scores de confiance pour chaque classe d'objet. La fonction de perte de SSD combine deux éléments : la

localisation et la confiance.

$$\mathcal{L} = \frac{1}{N}(\mathcal{L}_{\text{cls}} + \alpha\mathcal{L}_{\text{loc}}) \tag{1.29}$$

où $N$ est le nombre de boîtes englobantes et $\alpha$ un poids de pondérations entre deux pertes.

La fonction de perte de localisation mesure la précision des prédictions des boîtes englobantes par rapport aux boîtes englobantes réelles. Elle utilise la fonction de perte Smooth L1 pour minimiser les erreurs de prédiction tout en étant robuste aux valeurs aberrantes. Pour chaque paire de boîtes englobantes correspondantes $(i, j)$, la perte est calculée comme suit :

$$\mathcal{L}_{\text{loc}} = \sum_{i,j} \sum_{m \in \{x,y,w,h\}} \mathbb{1}_{ij}^{\text{match}} \cdot L_1^{\text{smooth}}(d_m^i - t_m^j)^2 \tag{1.30}$$

où :

— $\mathbb{1}_{ij}^{\text{match}}$ est une fonction indicatrice qui vaut 1 si la boîte englobante prédite $i$ correspond à la boîte englobante réelle $j$, et 0 sinon

— Les variables $d_m^i$ représentent les prédictions de décalage (offsets) pour la boîte englobante $i$, tandis que $t_m^j$ sont les décalages cibles calculés par rapport à la boîte englobante réelle $j$.

La perte de classification mesure la précision des prédictions de classe pour chaque boîte englobante. Elle est calculée à l'aide de la perte de cross-entropie softmax pour les classes positives et négatives :

$$\mathcal{L}_{\text{cls}} = -\sum_{i \in \text{pos}} \mathbb{1}_{ij}^{k} \log(\hat{c}_i^k) - \sum_{i \in \text{neg}} \log(\hat{c}_i^0) \tag{1.31}$$

où :

— $\mathbb{1}_{ij}^{k}$ est une fonction indicatrice qui vaut 1 si la boîte englobante $i$ est associée à la classe $k$, et 0 sinon.

— $\hat{c}_i^k$ représente la probabilité prédite que la boîte englobante $i$ appartienne à la classe $k$, calculée par la fonction softmax.

SSD présente plusieurs avantages notables. En premier lieu, son architecture en une étape lui permet d'être extrêmement rapide, capable de traiter des images en temps réel. De plus, en utilisant des cartes de caractéristiques de différentes résolutions, SSD parvient à détecter des objets de tailles variées avec une grande précision. Cette capacité multi-échelle est cruciale pour les applications pratiques où les objets peuvent varier considérablement en taille. SSD est également flexible en ce qui concerne le choix du réseau de base. Bien que VGG16 soit souvent utilisé, des versions de SSD peuvent être construites sur des architectures plus légères ou plus puissantes selon les besoins en termes de précision et

de vitesse. Single Shot MultiBox Detector (SSD) est une méthode de détection d'objets efficace et rapide qui réalise la localisation et la classification en une seule étape. Grâce à son architecture et à l'utilisation de cartes de caractéristiques multi-échelles, SSD offre une solution équilibrée entre précision et vitesse, adaptée aux applications en temps réel où la détection rapide et précise est essentielle.

### 1.5.3 Méthodes d'évaluations détection d'objet

Les méthodes d'évaluation jouent un rôle crucial dans le développement et l'amélioration des modèles de détection d'objets. En utilisant des ensembles de données de test annotés avec des boîtes englobantes de vérité terrain, les performances des modèles peuvent être comparées de manière objective. Des défis de détection d'objets tels que PASCAL VOC, MS COCO et ImageNet offrent des benchmarks standards qui facilitent la comparaison des modèles et l'évaluation de leurs performances sur des tâches de détection variées. Une évaluation rigoureuse à l'aide de ces métriques permet de mesurer efficacement les capacités des modèles de détection d'objets, d'identifier les points faibles et d'orienter les améliorations futures. Ces mesures sont essentielles pour le développement continu de systèmes de détection d'objets plus précis et plus robustes dans des applications réelles.

## 1.6 Détection d'objets dans l'écologie

### 1.6.1 L'émergence du machine learning dans l'écologie

L'intégration de l'apprentissage automatique dans la recherche a marqué une avancée permettant l'analyse et l'interprétation de données écologiques complexes avec une efficacité et une précision accrues par rapport à l'expertise humaine. Les premières applications de l'apprentissage automatique en écologie se sont principalement concentrées sur la modélisation de la distribution des espèces et la prédiction de résultats en fonction des variables environnementales. Par exemple, les forêts aléatoires (Breiman, 2001) et les boosted regression trees (BRT) ont été utilisées pour prédire les habitats des espèces et évaluer les impacts du changement climatique sur la biodiversité (Cutler et al., 2007 ; Elith* et al., 2006). Ces premiers efforts ont démontré l'utilité des techniques d'apprentissage automatique pour traiter la nature multifacette des données écologiques, posant les bases de méthodes plus sophistiquées.

### 1.6.2 L'apprentissage profond en écologie

S'appuyant sur le succès des premières applications de l'apprentissage automatique, l'apprentissage profond a émergé comme une technologie transformative, excédant parti-

culièrement dans les tâches impliquant de grandes quantités de données d'images et de vidéos. L'une des premières applications notables de l'apprentissage profond en écologie concernait la détection d'objets pour l'identification des espèces. Les méthodes traditionnelles d'identification des espèces reposaient fortement sur l'observation manuelle et les connaissances d'experts, ce qui était à la fois chronophage et sujet à des erreurs. Les réseaux de neurones convolutifs profonds (CNN) ont révolutionné ce processus en automatisant la détection et la classification des espèces dans les images. Ils ont été largement utilisés pour automatiser la détection et la classification des espèces dans les images. Ces réseaux sont capables d'extraire et de reconnaître des caractéristiques complexes à partir des données visuelles, surpassant ainsi les méthodes traditionnelles en termes de précision et d'efficacité. Par exemple, les CNN ont été appliqués aux images de pièges photographiques pour identifier les espèces animales, réduisant considérablement le besoin d'interventions manuelles et améliorant l'efficacité des programmes de surveillance de la biodiversité (Norouzzadeh et al., 2018, 2021 ; Tabak et al., 2019). La détection d'objets à l'aide de l'apprentissage profond ne se limite pas à l'identification des espèces. Elle est également utilisée pour cartographier les habitats et surveiller les comportements des animaux. Les CNN permettent de segmenter les images en différentes catégories écologiques, comme les forêts, les zones humides, les zones urbaines et les terres agricoles, facilitant ainsi la planification de la conservation et la gestion des ressources (Filatov & Yar, 2022 ; Kislov et al., 2021 ; Kumar & Jayagopal, 2021). De plus, les algorithmes de détection et de suivi d'objets permettent de suivre les mouvements des animaux et d'étudier leurs interactions, offrant de nouvelles perspectives sur l'écologie comportementale (Weinstein, 2018). Ces avancées montrent comment l'apprentissage profond, et en particulier la détection d'objets, a transformé la recherche écologique en permettant de traiter et d'interpréter des volumes de données considérables avec une précision inégalée. Les innovations dans ce domaine ont non seulement amélioré l'efficacité des études écologiques, mais ont également permis d'obtenir des perspectives plus détaillées et plus fiables sur les dynamiques écologiques.

Chapitre 2

# Données et métriques

## 2.1 Introduction

Dans ce chapitre, nous présentons la création d'un jeu de données destiné à l'entraînement de modèles de deep learning pour la détection et l'identification des Collemboles. Face aux défis posés par cette tâche, le recours aux techniques de deep learning apparaît comme une solution prometteuse. Ces techniques, ayant déjà prouvé leur efficacité dans le traitement d'images, pourraient permettre l'automatisation de l'identification des Collemboles. Ce chapitre détaille la constitution d'un jeu de données annoté à partir d'observations de Collemboles au microscope, destiné à l'entraînement de modèles de deep learning. Il aborde également les différentes métriques d'évaluation ainsi que les techniques d'augmentation de données utilisées pour comparer les performances de différents modèles de deep learning, et leur applicabilité pratique pour les spécialistes du domaine.

## 2.2 Jeu de données

Pour développer des modèles de deep learning destinés à l'identification des espèces de Collemboles, il est crucial de disposer d'un jeu de données représentatif des défis auxquels les experts font face. Ce jeu de données doit inclure des images de Collemboles sous lame de microscope, accompagnées d'annotations précises indiquant non seulement l'espèce, mais aussi la position de chaque spécimen sur l'image, car il s'agit d'un problème de détection. Comme une telle base de données n'existait pas jusqu'à présent, nous avons entrepris d'en créer une (Oriol, Pasquet & Cortet, 2024a). À cette fin, des Collemboles montés sur lames ont été photographiés et annotés à l'aide d'un microscope, dans le cadre d'un stage de licence de 6 mois, à partir des collections de lames existantes et conservées au laboratoire, issues de divers projets déjà réalisés (Joimel et al., 2017).

La première étape a consisté à collecter et identifier différentes espèces de Collemboles. Ensuite, nous avons numérisé ce jeu de données afin de le rendre exploitable pour l'entraînement des modèles de deep learning. Chaque spécimen de Collembole a été photographié à l'aide d'un microscope Carl Zeiss Labscope A1 équipé d'une caméra Optika C-P6, avec un zoom x50 pour garantir une uniformité de grossissement. Le logiciel Proview a été utilisé pour numériser les images, appliquant automatiquement une balance des blancs et un

ajustement de l'exposition pour améliorer la clarté des clichés. Les annotations associent ainsi à chaque image la position exacte des Collemboles et leur espèce. Les spécimens ont été photographiés dans la position laissée par les experts sur la lame, sans modifications, ce qui implique une variabilité dans les postures et la décoloration, en fonction du temps d'exposition, variant de quelques mois à 20 ans.

Toutes les photos ont des dimensions uniformes de 3072 × 2048 pixels, correspondant à une zone de 3 × 2 mm. Au total, 1664 photos distinctes ont été prises, représentant 2195 Collemboles identifiés et annotés. Comme il s'agit d'un problème de détection d'objets, plusieurs Collemboles peuvent apparaître sur la même image, ce qui explique pourquoi le nombre total de Collemboles dépasse celui des photos.

Une fois les clichés réalisés, il a fallu numériser les annotations, c'est-à-dire indiquer la position de chaque Collembole sur l'image ainsi que son espèce. Pour cela, nous avons utilisé le logiciel LabelImg (Figure 21), qui permet de générer un fichier XML contenant, pour chaque image et chaque Collembole, toutes les informations d'annotation nécessaires.

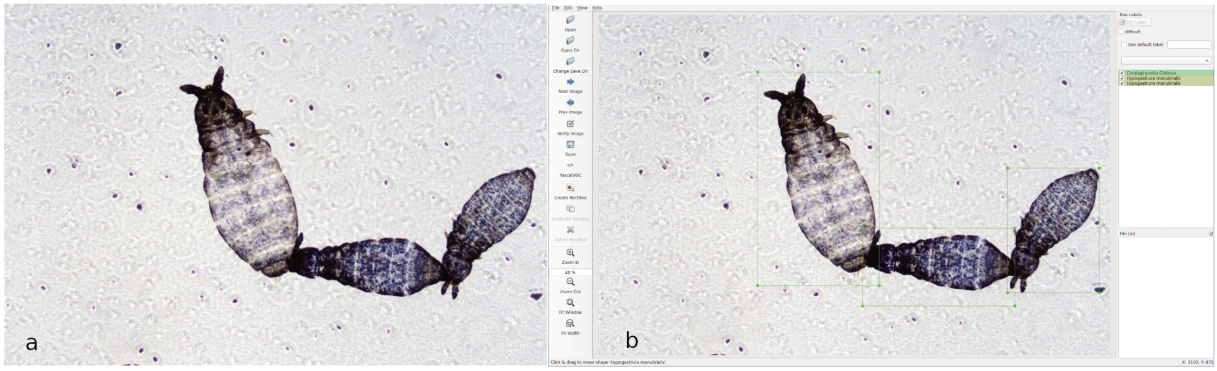


FIGURE 21 – a : Exemple d'image de Collemboles sous lame de microscope utilisé dans le jeu de données. b : Exemple d'annotation utilisant l'outil LabelImg : chaque boîte délimitée correspond aux coordonnées d'un collembole et est associée à un label (espèce). Ces informations sont ensuite enregistrées dans un fichier XML.

L'identification et la numérisation des Collemboles étant des processus très chronophages, un ensemble restreint d'espèces a été sélectionné parmi celles qui sont les plus représentées dans les sols soumis à une forte pression sélective, comme les sols agricoles. Neuf espèces ont été choisies pour créer la preuve de concept d'un outil capable d'identifier des espèces très communes de Collemboles : *Ceratophysella denticulata* (Bagnall, 1941) (CER-DEN), *Ceratophysella Gibbosa* (Bagnall, 1941) (CER-GIB), *Hemisotoma thermophila* (Bagnall, 1941) (HEM-THE), *Hypogastrura manubrialis* (Tullberg, 1869) (HYP-MAN), *Lepidocyrtus cyaneus* (Schille, 1908) (LEP-CYA), *Lepidocyrtus Lanuginosus* (Gmelin, 1788) (LEP-LAN), *Metaphorura affinis* (Börner, 1902) (MET-AFF), *Isotomiella minor* (Schäffer, 1896) (ISO-MIN), et *Parisotoma notabilis* (Schäffer, 1896) (PAR-NOT) (Figure 22).




FIGURE 22 – Exemple de photos pour chaque espèce / groupe d'espèces de Collemboles. On peut remarquer la forte variance dans l'apparence des Collemboles à la fois entre espèces et au sein d'une même espèce.

Ces espèces ont été sélectionnées pour illustrer la problématique liée à l'identification des Collemboles, notamment en raison de leur similarité inter-espèces et de leur variabilité intra-espèces. Par exemple, la similarité entre *Parisotoma notabilis* (Figure 22f) et *Hemisotoma thermophila* (Figure 22c) est frappante. Leur différenciation repose sur des caractéristiques subtiles, visibles uniquement à un fort grossissement (jusqu'à $\times 630$), telles que la position et la forme des soies sur les segments thoraciques ou abdominaux, qui varient d'une espèce à l'autre et sont essentielles pour les distinguer (Figure 23).

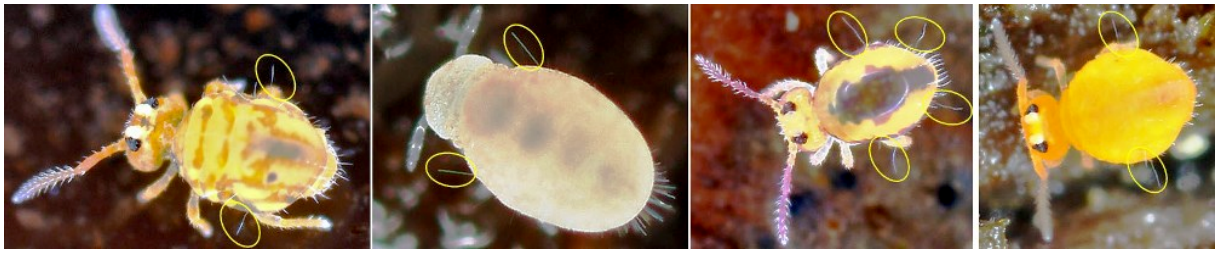

FIGURE 23 – Ces illustrations montrent l'implantation latérale des soies sur plusieurs spécimens de Collemboles [2].

En utilisant un zoom faible (×50), les Collemboles deviennent morphologiquement très similaires, ce qui est une contrainte imposée par la problématique. C'est également le cas de *Ceratophysella denticulata* (Figure 22a), *Ceratophysella Gibbosa* (Figure 22b) et *Hypogastrura manubrialis* (Figure 22e), ce qui rend ces deux groupes d'espèces difficiles à différencier. Pour des raisons pratiques, les espèces *Lepidocyrtus cyaneus* (Figure 22g) et *Lepidocyrtus Lanuginosus* (Figure 22h) ont été fusionnées en tant que LEP car la principale différence entre ces deux espèces est leur coloration, qui se dégrade avec le temps une fois qu'elles sont montées sur des lames, et comme il y a un faible nombre d'annotations pour les objets d'intérêts (entre 109 et 267), il a été décidé de les fusionner. *Ceratophysella denticulata* (Figure 22a) et *Ceratophysella Gibbosa* (Figure 22b) ont été fusionnées en une seule catégorie, désignée comme CER, en raison de la rareté des annotations pour *Ceratophysella Gibbosa* (Figure 22b) dans nos ensembles de données, ce qui permet de l'utiliser comme espèce d'intérêt. Il est également important de montrer des images d'autres Collemboles que ces 9 espèces à notre modèle d'apprentissage profond car, dans un scénario réaliste, il devra les différencier d'autres espèces inconnues. Pour ce faire, nous avons ajouté une nouvelle catégorie appelée "OTHER" ("Autres"), dans laquelle nous avons regroupé les Collemboles de multiples espèces non incluses dans les espèces d'intérêt. Pour limiter les biais, toutes les espèces de Collemboles ont été échantillonnées sur plusieurs projets, ce qui nous permet d'utiliser des lames de microscope récentes, les plus récentes provenant de projets de 2021, et des lames de microscope plus anciennes, les projets les plus anciens datant de 2003 (Tableau 2.1).

2. Source de l'image : https ://collemboles.fr/anatomie/12-soies-sensilles-des-collemboles.html

TABLE 2.1 – Nombre d'annotation par espèces par projets

| Projet | Date | Lieu | Publications | CER | HEM-THE | HYP-MAN | ISO-MIN | LEP | MET-AFF | PAR-NOT | OTHER | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BISES | 2021 | 200 sites en milieu urbain dans 4 villes de France (Nancy, Montpellier, Paris, Nantes) | Non publié | 93 | 93 | 0 | 61 | 121 | 23 | 154 | 255 | 800 |
| Bioindicateur 2 | 2009/2010 | 10 sites (terres cultivées, pâturages et forêts) dans plusieurs régions, France | (CORTET, 2012 ; PÉRÈS et al., 2011) | 83 | 1 | 0 | 4 | 8 | 0 | 54 | 181 | 331 |
| RMQS Biodiv | 2006/2007 | 97 sites (terres cultivées, pâturages et forêts) dans l'ensemble de la région Bretagne, France | (CLUZEAU et al., 2009, 2012 ; PONGE et al., 2013) | 1 | 16 | 0 | 5 | 8 | 0 | 6 | 0 | 36 |
| Thèse Celine Pernin | 2003 | Feu de forêt dans le massif des Maures, région de Provence, France | (PERNIN et al., 2006) | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Ecopic | 2018/2019 | Forêts le long d'un gradient altitudinal dans les Alpes françaises (Chamrousse, région Auvergne-Rhône-Alpes, France) | Non publié | 23 | 0 | 0 | 11 | 56 | 0 | 5 | 22 | 117 |
| Howécourt | 2003 | Friche industrielle à Homécourt, Grand-Est Region, France | Non publié | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 1 | 9 |
| NARBONS | 2003 | Champs de maïs à Montesquieu-Lauragais, Occitanie Region, France | (CORTET et al., 2007) | 0 | 0 | 0 | 11 | 6 | 0 | 0 | 0 | 17 |
| Thèse Benjamin Pey | 2007/2018 | Friche industrielle à Homécourt, Grand-Est Region, France | (JOIMEL et al., 2021) | 9 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 18 |
| Parcelles Lysimétriques | 2007 | Friche industrielle à Homécourt, Grand-Est Region, France | (CÉBRON et al., 2011 ; OUVRARD et al., 2011) | 0 | 30 | 0 | 0 | 16 | 0 | 0 | 6 | 52 |
| Pompey | 2007 | Forêt sur un ancien bassin de décantation, Frouard, Grand-Est, France | (HUOT et al., 2018) | 0 | 68 | 14 | 3 | 6 | 0 | 0 | 93 | 184 |
| TIDM URBA | 2021 | 83 sites dans la ville de Dijon, Bourgogne-Franch-Comté, France | Non publié | 77 | 0 | 95 | 53 | 32 | 106 | 48 | 162 | 573 |
| VADEBIO | 2010 | Terres cultivées à La Bouzule, Grand-Est Region, France | Non publié | 33 | 0 | 0 | 0 | 3 | 0 | 0 | 21 | 57 |
| **Total** | | | | 320 | 217 | 109 | 148 | 264 | 129 | 267 | 741 | 2195 |

## 2.3 Augmentation de données

Des techniques d'augmentation des données ont été appliquées pour améliorer la capacité de généralisation des modèles. Ces techniques incluent le recadrage aléatoire, la mosaïque, les distorsions de couleur (luminosité, contraste, saturation, teinte), le flou gaussien, l'échelle aléatoire, la rotation aléatoire et le retournement horizontal.

L'augmentation des données aide à simuler l'état des anciennes lames de microscope comme sur la Figure 24, ce qui améliore les résultats sur les projets utilisant ces lames. Les anciennes lames de microscope peuvent être en mauvais état en raison de l'usure, de la dégradation du matériel au fil du temps, du vieillissement des produits de fixation (Marc-André 2) ou de la qualité initiale des préparations. Ces lames peuvent présenter des artefacts, des variations de couleur, des flous, des rayures et d'autres formes de bruit qui compliquent l'identification visuelle des Collemboles. En appliquant des techniques d'augmentation de données qui imitent ces imperfections, comme l'ajout de flou gaussien, des distorsions de couleur et des recadrages aléatoires, les modèles peuvent apprendre à reconnaître les Collemboles même dans des conditions non idéales.

Cela est particulièrement bénéfique pour les projets anciens où les données disponibles sont limitées et de qualité variable. En entraînant les modèles sur des images augmentées,

qui simulent les conditions réelles des lames anciennes, les modèles deviennent mieux équipés pour traiter ces images lors de l'application pratique. Cela permet non seulement d'améliorer la précision de l'identification des Collemboles sur les lames de microscope anciennes, mais aussi de maximiser l'utilisation des données historiques, ce qui est essentiel pour des études longitudinales et des analyses de tendances sur de longues périodes.

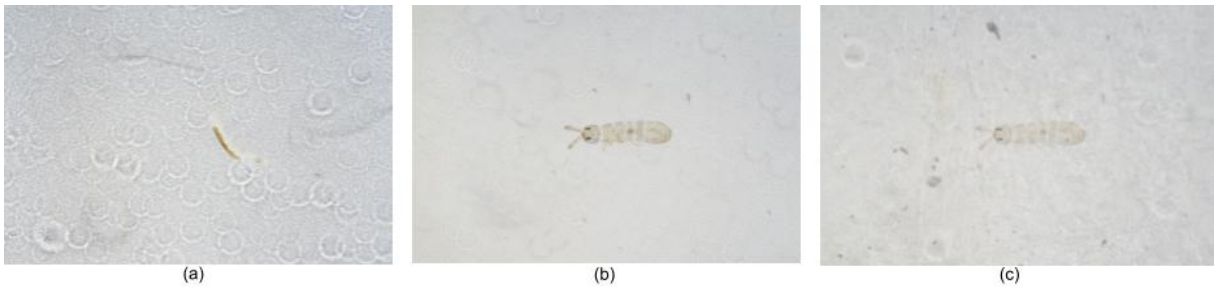


FIGURE 24 – L'image (a) représente une photo de lame de microscope d'un ancien projet, la lame est en mauvais état avec beaucoup de bruit sur l'image, et les caractéristiques des Collemboles ne sont pas toutes visibles. Grâce à l'augmentation des données, nous pouvons ajouter du bruit à l'image (b) pour créer l'image (c), ce qui permettra de créer davantage d'images de lames en mauvais état et d'améliorer les résultats du modèle sur les anciens projets.

## 2.4 Métriques

Nous allons évaluer les performances de modèles de détection d'objets en deep learning, qui, à partir d'une image, prédisent des boîtes englobantes (ou bounding boxes) pour des espèces sur lesquelles ils ont été entraînés. Pour évaluer la qualité de ces prédictions, il est nécessaire de les comparer aux boîtes issues de la vérité terrain. Afin de mieux comprendre les types d'erreurs commises par notre modèle, plusieurs métriques spécifiques peuvent être définies. Ces métriques permettent de quantifier la précision et la robustesse des prédictions, et sont essentielles pour comparer différents modèles et approches, tout en identifiant les domaines nécessitant des améliorations. Parmi les principales métriques utilisées, on trouve la précision moyenne (mAP), le rappel, la précision, le F1-score, l'Intersection over Union (IoU), et les courbes de Rappel-Précision.

**Intersection over Union (IoU)**

L'Intersection over Union (IoU) est une métrique d'évaluation utilisée pour mesurer la précision de la localisation des boîtes englobantes. Elle est définie comme le rapport entre l'intersection et l'union des zones de la boîte prédite par le modèle et de la boîte issue de la vérité terrain :

$$\text{IoU} = \frac{\text{Aire}(B_p \cap B_g)}{\text{Aire}(B_p \cup B_g)} \tag{2.1}$$

où :

— $B_p \cap B_g$ représente l'intersection des deux boîtes englobantes.

— $B_p \cup B_g$ représente l'union des deux boîtes englobantes.

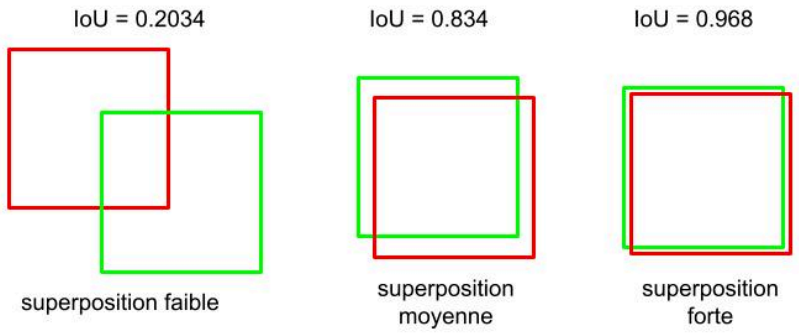


FIGURE 25 – Exemple d'utilisation de la métrique IoU. En rouge, une boîte correspondant aux coordonnées d'un objet A (prédiction) et en vert , celle d'un objet B (vérité terrain). L'IoU permet d'indiquer le niveau de superposition des deux objets, et ainsi de déterminer si la prédiction d'un modèle est proche de l'annotation faite par un expert.

Une IoU (Figure 25) de 0 signifie qu'il n'y a aucun chevauchement entre la boîte prédite et la vérité terrain, tandis qu'une IoU de 1 signifie que la boîte prédite correspond exactement à la vérité terrain. En pratique, une prédiction est souvent considérée comme correcte si l'IoU est supérieure à un certain seuil, généralement fixé à 0.5. Des seuils plus élevés peuvent être utilisés pour des applications nécessitant une plus grande précision. L'IoU est largement utilisée dans les tâches de détection d'objets pour diverses raisons. Elle permet d'évaluer la précision des boîtes prédites par rapport aux vérités terrain, ce qui est crucial pour la qualité des modèles de détection d'objets. Les métriques comme l'AP (Average Precision) et la mAP (mean Average Precision) utilisent souvent des seuils d'IoU pour déterminer les True Positives (TP), ce qui impacte directement les mesures de performance du modèle. Bien que l'IoU soit une métrique très utilisée, elle présente certaines limites. Pour des objets de petite taille, une petite erreur de localisation peut entraîner une grande variation de l'IoU, ce qui peut être injustement pénalisant. Pour des objets dont les boîtes englobantes sont presque adjacentes ou se chevauchent légèrement, l'IoU peut ne pas refléter correctement la qualité de la détection. En résumé, l'IoU est une

métrique fondamentale pour l'évaluation des modèles de détection d'objets, fournissant un moyen précis et rigoureux de quantification de la justesse de la localisation des objets. Toutefois, pour évaluer globalement les performances d'un modèle, il est également crucial de prendre en compte d'autres métriques complémentaires comme la précision, le rappel et le F1-score, qui permettent d'analyser plus en détail les types d'erreurs commises.

**Précision, Rappel et F1**

La précision et le rappel sont des métriques d'évaluations utilisées pour évaluer la performance des modèles de détection d'objets et de classification, en particulier lorsqu'il y a un déséquilibre entre les classes ou que la présence de faux positifs et de faux négatifs a des implications différentes. Elles offrent une vue complémentaire de la performance du modèle en termes de capacité à identifier correctement les objets et à minimiser les erreurs. La **précision** (Precision) mesure la proportion de prédictions positives correctes parmi l'ensemble des prédictions positives effectuées par le modèle. Elle est définie comme :

$$\text{Précision} = \frac{TP}{TP + FP} \tag{2.2}$$

Une haute précision indique que, lorsqu'un modèle prédit la présence d'un objet, il est souvent correct. Cependant, elle ne tient pas compte des objets manqués par le modèle (faux négatifs). Le **rappel** (Recall) mesure la proportion de prédictions positives correctes parmi l'ensemble des instances positives réelles dans les données. Il est défini comme :

$$\text{Rappel} = \frac{TP}{TP + FN} \tag{2.3}$$

Un rappel élevé indique que le modèle est capable de détecter la plupart des objets présents dans les données, mais il ne tient pas compte des fausses alertes (faux positifs). Il existe souvent un compromis entre précision et rappel. Un modèle peut avoir une haute précision mais un faible rappel s'il est très conservateur dans ses prédictions (seulement lorsqu'il est très sûr), ou un haut rappel mais une faible précision s'il prédit la présence d'objets de manière très libérale. Pour évaluer ce compromis, on utilise souvent la courbe Rappel-Précision (Precision-Recall Curve). Cette courbe trace la précision en fonction du rappel pour différents seuils de décision du modèle. Elle permet de visualiser comment ces deux métriques varient conjointement et d'identifier un équilibre approprié pour une application donnée. Pour combiner la précision et le rappel en une seule métrique, on utilise souvent le F1-score, qui est la moyenne harmonique de la précision et du rappel :

$$\text{F1-score} = 2 \cdot \frac{\text{Précision} \cdot \text{Rappel}}{\text{Précision} + \text{Rappel}} \tag{2.4}$$

Le F1-score est particulièrement utile lorsque l'on cherche à équilibrer à la fois la précision et le rappel. Pour obtenir une vision plus complète des performances du modèle à différents seuils de classification, l'analyse à l'aide de la courbe Rappel-Précision devient

indispensable, car elle illustre la relation entre ces deux métriques sur l'ensemble des prédictions.

### Courbe Rappel-Précision

La courbe de Rappel-Précision (Figure 26) est un outil graphique permettant de visualiser la relation entre la précision et le rappel à différents seuils de probabilité pour la détection. Elle est particulièrement utile dans les cas où les données sont déséquilibrées, c'est-à-dire lorsque les classes positives et négatives ne sont pas également représentées. Pour tracer une courbe de Rappel-Précision, on calcule la précision et le rappel pour chaque seuil possible et on trace la précision en fonction du rappel.

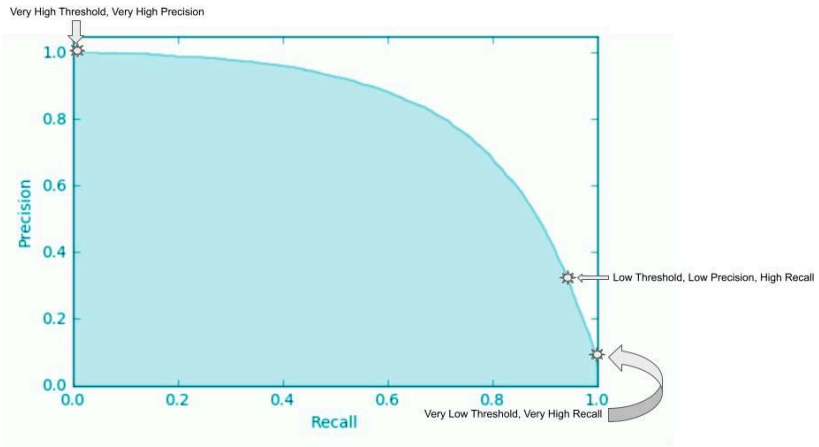


FIGURE 26 – Exemple de courbe de Rappel-Précision

La courbe de Rappel-Précision permet d'évaluer comment le modèle équilibre la précision et le rappel, et de choisir un seuil de décision optimal en fonction de l'application. Un modèle idéal produirait une courbe atteignant le coin supérieur droit du graphique, indiquant une précision et un rappel élevés. Pour quantifier cette courbe, l'Average Precision (AP) est souvent utilisée, résumant en une seule valeur la performance du modèle sur l'ensemble des seuils.

### Précision Moyenne (AP)

L'Average Precision (AP) est une métrique couramment utilisée pour évaluer la performance des modèles de détection d'objets et de classification. Elle combine les notions de précision et de rappel pour fournir une mesure unique de la performance du modèle à travers différents seuils de décision. Cette métrique est particulièrement utile pour comparer différents modèles ou différentes configurations du même modèle. Le calcul de l'AP repose sur la courbe Rappel-Précision (Precision-Recall Curve), qui trace la précision en fonction du rappel pour divers seuils de décision du modèle. Pour chaque prédiction du modèle, on peut définir un seuil de confiance. En variant ce seuil, on obtient différentes

valeurs de précision et de rappel. Par exemple, un seuil élevé de confiance pourrait augmenter la précision mais réduire le rappel, tandis qu'un seuil bas pourrait augmenter le rappel mais réduire la précision. Pour chaque seuil de décision, on calcule la précision et le rappel associés, générant ainsi une série de points représentant la relation entre rappel et précision à différents seuils. Une approche standard pour lisser la courbe consiste à utiliser l'interpolation, souvent en prenant la précision maximale pour tout rappel supérieur ou égal au rappel courant. L'AP est ensuite calculé comme l'aire sous la courbe Rappel-Précision interpolée. Cela peut se faire par sommation discrète sur des points de rappel échantillonnés :

$$AP = \sum_i (R_{i+1} - R_i) \cdot P_{\text{interp}}(R_{i+1}) \tag{2.5}$$

où $R_n$ représente le rappel à la $n$-ième seuil, $R_{n-1}$ est le rappel à l'étape précédente et $P_{\text{interp}}$ est la précision interpolé du rappel $R$ donné.

L'AP offre une mesure globale de la performance du modèle en tenant compte à la fois de la précision et du rappel à divers seuils de décision. Contrairement à une seule valeur de précision ou de rappel, l'AP considère l'équilibre entre ces deux métriques sur toute la gamme de seuils, offrant ainsi une image plus complète de la performance du modèle. Lorsque l'on évalue plusieurs classes d'objets, la moyenne de l'AP sur l'ensemble des classes, appelée mAP (mean Average Precision), devient une métrique clé pour comparer les performances globales du modèle.

**Précision Moyenne (mAP)**

La précision moyenne (mAP) est une extension de l'AP utilisée pour évaluer les performances globales des modèles de détection d'objets sur plusieurs classes. Elle est calculée en moyennant les APs sur toutes les classes. Cette métrique est particulièrement utile pour comparer les performances de différents modèles sur des ensembles de données complexes et multi-classes. Le mAP peut être formulée comme suit :

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{2.6}$$

où $N$ est le nombre total de classes et $\text{AP}_i$ est l'Average Precision pour la $i$-ème classe.

Le mAP est une mesure clé pour évaluer les modèles de détection d'objets car il prend en compte toutes les classes d'objets, fournissant ainsi une évaluation globale de la performance du modèle. En intégrant la précision et le rappel, le mAP évalue la capacité du modèle à trouver le bon équilibre entre ces deux mesures. Le mAP permet également de comparer facilement différents modèles ou configurations en offrant une mesure unique de performance. Le mAP est utilisé dans de nombreuses compétitions de détection d'objets, comme PASCAL VOC, MS COCO, et d'autres. Chaque compétition peut définir

des variantes du mAP, par exemple en utilisant différents seuils d'IoU ou en pondérant différemment les classes selon leur importance ou leur fréquence dans l'ensemble de données. Bien que le mAP fournisse une mesure précise et adaptée aux spécificités de chaque compétition, il est souvent complété par l'analyse des matrices de confusion pour offrir une vision plus détaillée des performances spécifiques à chaque classe. Bien que le mAP fournisse une évaluation globale utile, il est souvent complété par l'analyse des matrices de confusion, qui apportent des détails supplémentaires sur les performances spécifiques à chaque classe. Cette analyse complémentaire aide à identifier les points forts et les faiblesses du modèle, facilitant ainsi l'amélioration ciblée des performances de classification.

**Matrice de Confusion**

Dans les problèmes de classification multi-classes, une matrice de confusion (Figure 27) aide à évaluer la performance d'un modèle qui doit distinguer entre plusieurs catégories. Elle s'étend pour inclure toutes les interactions possibles entre les classes prédites et les classes réelles, offrant une vue complète de la capacité du modèle.

La matrice de confusion en classification multi-classes est organisée en lignes et colonnes. Chaque ligne correspond à une classe réelle et chaque colonne à une classe prédite par le modèle. Les cellules de la matrice indiquent le nombre de fois que le modèle prédit une classe pour une classe réelle donnée, avec les succès situés sur la diagonale principale et les erreurs réparties ailleurs.

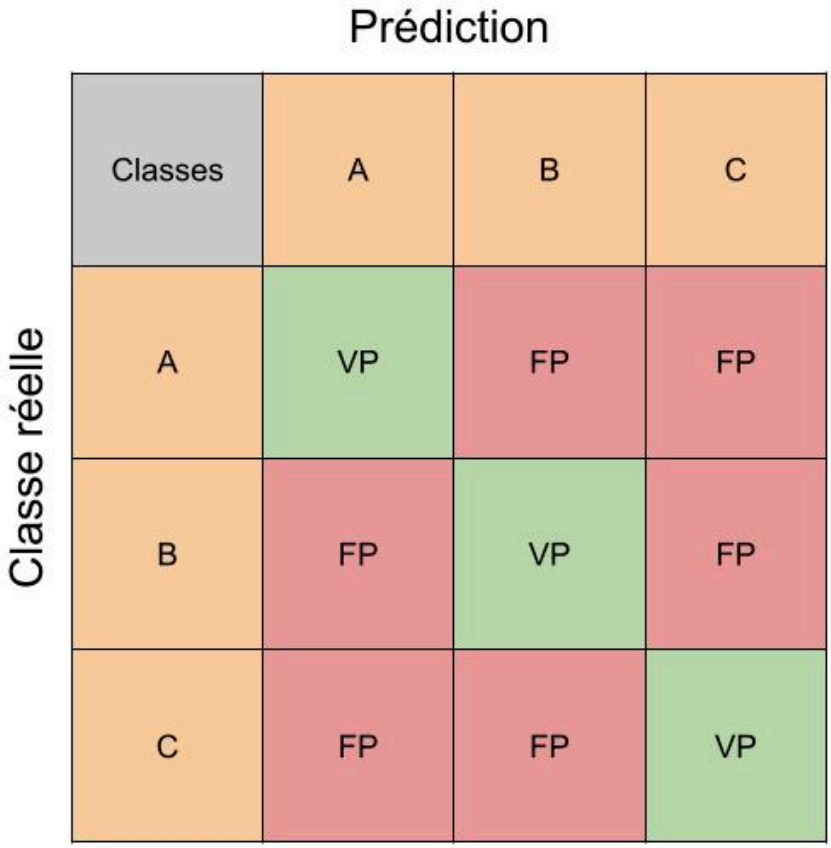


FIGURE 27 – Exemple de matrice de confusion pour les classes A, B, et C. La diagonale représente tous les vrais positifs et le reste indique les faux positifs.

Cette matrice permet d'identifier les classes que le modèle confond souvent, mettant en lumière des similitudes ou des ambiguïtés dans les données qui peuvent nécessiter une attention particulière. Elle montre également la sensibilité et la spécificité du modèle pour chaque classe, révélant les forces et les faiblesses du modèle dans la classification de chaque catégorie.

La matrice de confusion en contexte multi-classes offre une analyse détaillée qui est essentielle pour comprendre comment un modèle traite un ensemble de données avec de nombreuses catégories. Elle expose non seulement les erreurs totales mais aussi le type d'erreurs, ce qui est crucial pour affiner le modèle. De plus, elle aide à cibler des interventions spécifiques pour améliorer la performance du modèle, comme ajuster les caractéristiques ou augmenter les données pour des classes mal prédites.

Chapitre 3

# Évaluation des méthodes de l'état de l'art

## 3.1 Introduction

Ce chapitre propose une comparaison des techniques modernes de détection d'objets appliquées à l'identification automatique des collemboles. L'objectif est de démontrer l'efficacité de ces modèles et de déterminer les approches les plus performantes pour cette tâche complexe, dans le but de développer un outil destiné aux experts. Cet outil permettrait une surveillance plus rapide et à grande échelle de la santé des sols. Nous partons de l'hypothèse que les modèles de deep learning dédiés à la détection et à l'identification des collemboles sont capables de rivaliser avec l'expertise humaine dans ce domaine.

## 3.2 Stratégie d'évaluation

En raison de la faible quantité de données à notre disposition, une validation croisée a été utilisée pour garantir les résultats statistiques et s'assurer que le modèle n'était pas surajusté. La technique de validation croisée que nous avons utilisée fonctionne comme suit : K divisions (fold) (Figure 28) sont créées en mélangeant aléatoirement le jeu de données et en le divisant en un ensemble d'entraînement et un ensemble d'évaluation K fois, puis pour chaque division, un nouveau modèle est entraîné. Cela nous permet de former K modèles différents, chacun sur une version différente du jeu de données. Une fois tous les modèles entraînés, nous les évaluons et moyennons toutes leurs performances pour mieux comprendre la capacité du modèle sur ce jeu de données. La faible variance des résultats d'évaluation de chaque modèle indiquerait que les modèles sont moins susceptibles de surajuster les données, montrant ainsi qu'ils sont plus susceptibles de généraliser cette tâche sur des données non vues. À l'inverse, de grandes disparités dans ces scores pourraient signaler une dépendance excessive des modèles aux données d'entraînement, réduisant potentiellement leur efficacité sur des données non rencontrées auparavant. Dans notre cas, chaque modèle a utilisé une validation croisée de 5 divisions. Cela signifie que nous devions avoir suffisamment confiance dans la prédiction du modèle pour l'utiliser.

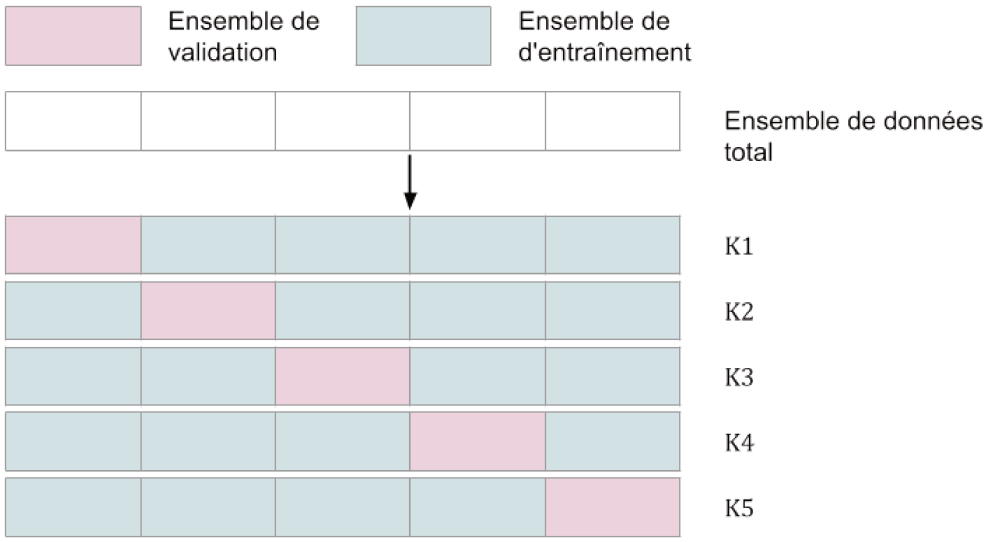


FIGURE 28 – Démonstration de l'utilisation d'un K-fold, on part d'un ensemble de données et on crée K version d'un jeu de données, dans notre cas 5, que l'on divise en ensemble d'entraînement et de validation.

## 3.3 Choix des paramètres

Pour assurer une standardisation dans la formation des modèles de deep learning destinés à l'identification des espèces de collemboles, tous ont été entraînés avec les mêmes paramètres, définis empiriquement. Ces paramètres incluent une taille de lot (batch size) de 8, un taux d'apprentissage initial de 0.01, et un taux de décroissance des poids de 0.005. L'optimiseur qui a été choisi est Adam, avec un paramètre $\beta_1$ fixé à 0.937, ce qui influence la moyenne mobile du gradient dans l'optimiseur. Le nombre d'époques sur lesquelles chaque modèle a été entraîné est de 800.

Cette approche uniforme dans la configuration des paramètres et dans la stratégie d'entraînement permet de garantir que tous les modèles sont évalués sur une base équitable et reflètent fidèlement leur capacité à généraliser à partir des données qu'ils n'ont pas encore vues, maximisant ainsi l'efficacité et la fiabilité de l'outil pour une utilisation pratique par les spécialistes.

## 3.4 Résultats

### 3.4.1 Résultats de l'état de l'art

TABLE 3.1 – Comparaison des différents modèles. Le meilleur mAP et le meilleur AP par espèce ont été écrits en gras.

| | mAP | "OTHER" AP | "CER" AP | "HEM-THE" AP | "HYP-MAN" AP | "LEP" AP | "PAR-NOT" AP | "ISO-MIN" AP | "MET-AFF" AP |
|---|---|---|---|---|---|---|---|---|---|
| Faster R-CNN (1280) | 0.656 | 0.632 | 0.83 | 0.587 | 0.718 | 0.652 | 0.525 | 0.417 | 0.887 |
| Yolov5n | 0.802 | 0.709 | 0.916 | 0.772 | 0.894 | 0.772 | 0.778 | 0.659 | 0.916 |
| Yolov5m6 | 0.869 | 0.779 | 0.926 | 0.841 | 0.943 | 0.837 | 0.885 | 0.775 | 0.966 |
| Yolov5l6 | 0.870 | 0.793 | 0.944 | 0.845 | 0.935 | 0.843 | 0.843 | 0.794 | 0.966 |
| Yolov5x6 | **0.894** | **0.803** | **0.947** | **0.891** | **0.948** | **0.855** | **0.910** | **0.807** | **0.991** |

L'objectif de cette étude était de comparer les performances des modèles de détection d'objets à l'état de l'art, incluant les modèles en une étape (YOLOv5) et les modèles en deux étapes (Faster R-CNN), pour la détection et l'identification des Collemboles sur des images de lames de microscope.

Le Tableau 3.1 présente et compare les résultats obtenus par chaque modèle en termes de mAP (mean Average Precision) et d'AP (Average Precision) pour chaque espèce de Collembole. Chaque mAP est la moyenne des résultats de chaque fold.

Les résultats montrent clairement la supériorité des modèles YOLOv5, en particulier YOLOv5x6, pour cette tâche. YOLOv5x6 a atteint un mAP de 0.894, surpassant toutes les autres versions de YOLOv5 ainsi que le modèle Faster R-CNN (1280), qui a obtenu un mAP de 0.656. Faster R-CNN présente des scores d'AP allant de 0.417 pour "ISO-MIN" à 0.887 pour "MET-AFF", avec des performances variables selon les espèces : par exemple, "CER" (0.830) et "MET-AFF" (0.887) montrent des scores élevés, tandis que "HEM-THE" (0.587) et "ISO-MIN" (0.417) présentent des scores plus faibles. Ces résultats confirment que YOLOv5, notamment dans ses versions plus grandes comme YOLOv5x6, est mieux adapté pour cette tâche par rapport à Faster R-CNN.

Les différentes versions de YOLOv5 ont montré des performances variées, avec une nette supériorité des modèles plus complexes et plus grands. YOLOv5n, la version la plus petite et la plus rapide, a obtenu un mAP de 0.802. Malgré sa taille réduite, ce modèle a présenté des performances raisonnables, avec des scores d'AP allant de 0.659 pour "ISO-MIN" à 0.916 pour "CER" et "MET-AFF". YOLOv5m6 et YOLOv5l6, avec des mAP respectifs de 0.869 et 0.870, offrent un bon équilibre entre précision et vitesse, avec des scores d'AP particulièrement élevés pour "MET-AFF" (0.966).

YOLOv5x6 est le modèle le plus performant avec un mAP de 0.894, atteignant un AP de 0.991 pour "MET-AFF" et maintenant des scores élevés pour toutes les autres catégories. YOLOv5x6 a donc démontré sa supériorité à la fois en moyenne et pour chaque espèce d'intérêt, surpassant toutes les autres versions et offrant une performance plus homogène.

Cependant, l'identification précise des Collemboles est complexifiée par plusieurs défis intrinsèques liés à leurs caractéristiques morphologiques uniques. La variabilité intra-

spécifique, les similitudes interspécifiques et les traits morphologiques nécessaires à l'identification ne sont parfois discernables qu'avec un agrandissement considérable. Ces aspects expliquent pourquoi certaines espèces sont plus difficiles à identifier, entraînant des performances variables selon les espèces testées. Par exemple, *Isotomiella minor*, qui a l'AP le plus faible dans le benchmark pour YOLOv5x6, est aussi l'une des espèces les moins annotées, suggérant une corrélation entre le nombre d'annotations et la performance. À l'inverse, *Hypogastrura manubrialis*, malgré son faible nombre d'annotations, figure parmi les espèces les mieux détectées, illustrant que la difficulté d'identification n'est pas toujours directement liée au nombre d'annotations.

Il est intéressant de noter que les espèces les plus faciles à identifier tendent, à l'exception de "CER", à être présentes dans un nombre plus restreint de projets. Par exemple, *Hypogastrura manubrialis* et *Metaphorura affinis* sont les deux espèces les mieux identifiées et leurs annotations ne proviennent que de deux projets. En revanche, *Isotomiella minor* et "LEP", qui sont parmi les plus difficiles à identifier, sont présentes sur 7 et 10 projets respectivement, introduisant une plus grande variabilité d'apparence due à la diversité des projets.

TABLE 3.2 – Ecart type de l'ensemble des résultats des fold du K-fold de chaque modèle

| | "OTHER" AP | "CER" AP | "HEM-THE" AP | "HYP-MAN" AP | "LEP" AP | "PAR-NOT" AP | "ISO-MIN" AP | "MET-AFF" AP |
|---|---|---|---|---|---|---|---|---|
| Faster R-CNN (1280) | 2.3% | 2.1% | 5.9% | 17.8% | 6.6% | 6% | 11.2% | 3% |
| Yolov5x6 | 2.4% | 1.5% | 3.3% | 0.8% | 4.7% | 2.9% | 5.9% | 0.4% |

Les résultats dans le Tableau 3.2 montrent les écarts types des performances des deux modèles, Faster R-CNN (1280) et YOLOv5x6, sur plusieurs classes. Pour Faster R-CNN (1280), les écarts types varient de 2.1% à 17.8%, avec la classe *Hypogastrura manubrialis* ayant l'écart type le plus élevé à 17.8%, tandis que "CER" a l'écart type le plus bas à 2.1%. Pour YOLOv5x6, les écarts types sont généralement plus faibles, allant de 0.4% à 5.9%, avec *Metaphorura affinis* ayant l'écart type le plus bas à 0.4% et *Isotomiella minor* le plus élevé à 5.9%. Ces écarts types indiquent une variabilité différente des performances pour chaque classe et modèle.

Lors de la comparaison entre YOLOv5x6 et Faster R-CNN, on observe une variance beaucoup plus élevée des résultats pour Faster R-CNN, notamment pour la détection des différentes espèces. Cette variance, visible dans les données du benchmark et l'évaluation par k-fold, pose un défi : l'ajout d'une nouvelle espèce rend plus difficile la prévision des performances que l'on pourrait espérer de son identification. En revanche, YOLOv5x6 présente une variance bien moindre, offrant des résultats plus fiables et améliorant la probabilité de bonnes performances lors de l'introduction de nouvelles espèces, à condition de disposer d'annotations suffisantes, conformément à l'objectif de cet outil.

En conclusion, ces résultats révèlent que YOLOv5x6, avec ses performances supérieures et sa variabilité réduite, est mieux adapté pour la détection des Collemboles que Faster R-CNN, particulièrement dans des contextes où les projets peuvent introduire une variabilité

significative dans l'apparence des espèces.

Afin de mieux comprendre les comportements de ces deux modèles vis-à-vis des différentes classes d'espèces, il est crucial d'analyser leurs matrices de confusion. Ces dernières permettent de visualiser les erreurs de classification et d'identifier les espèces mal détectées, tout en offrant des indications précises sur les forces et faiblesses des modèles en termes de généralisation et de robustesse face aux variations d'apparence des Collemboles.

### 3.4.2 Matrice de confusion

Pour approfondir cette analyse et identifier précisément les types d'erreurs, nous nous tournons vers les matrices de confusion associées à chaque modèle. Ces matrices fournissent une vue détaillée de la performance par classe, révélant les confusions spécifiques entre les classes ainsi que les réussites. En examinant les Figures 29 et 30, nous pouvons voir où chaque modèle excelle ou échoue, offrant des perspectives cruciales pour des ajustements futurs et une amélioration de la précision de classification.

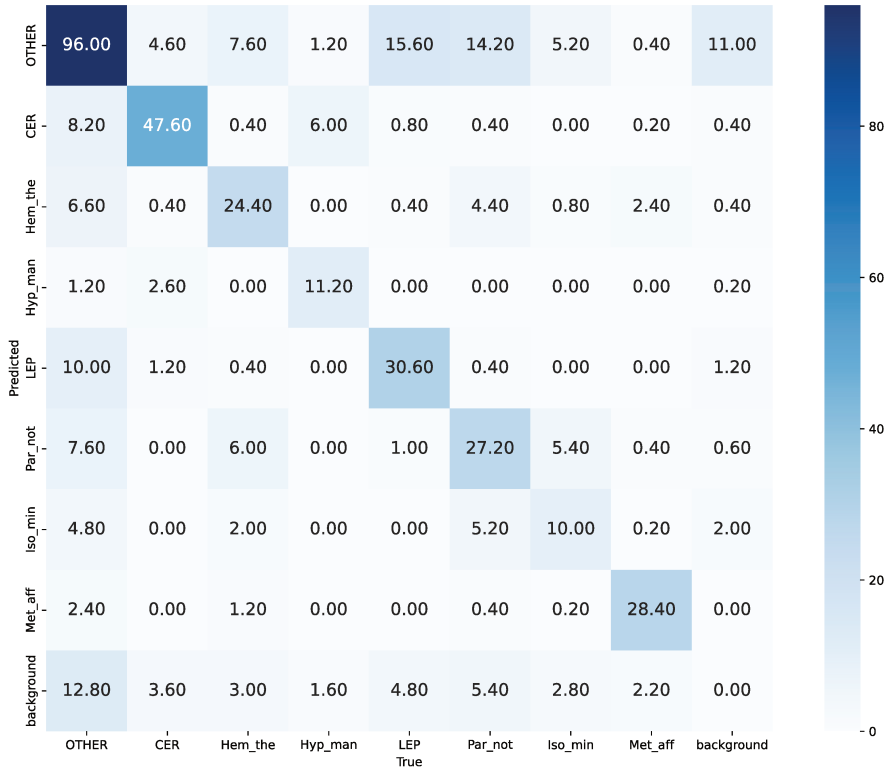


FIGURE 29 – Matrice de confusion de Faster R-CNN.

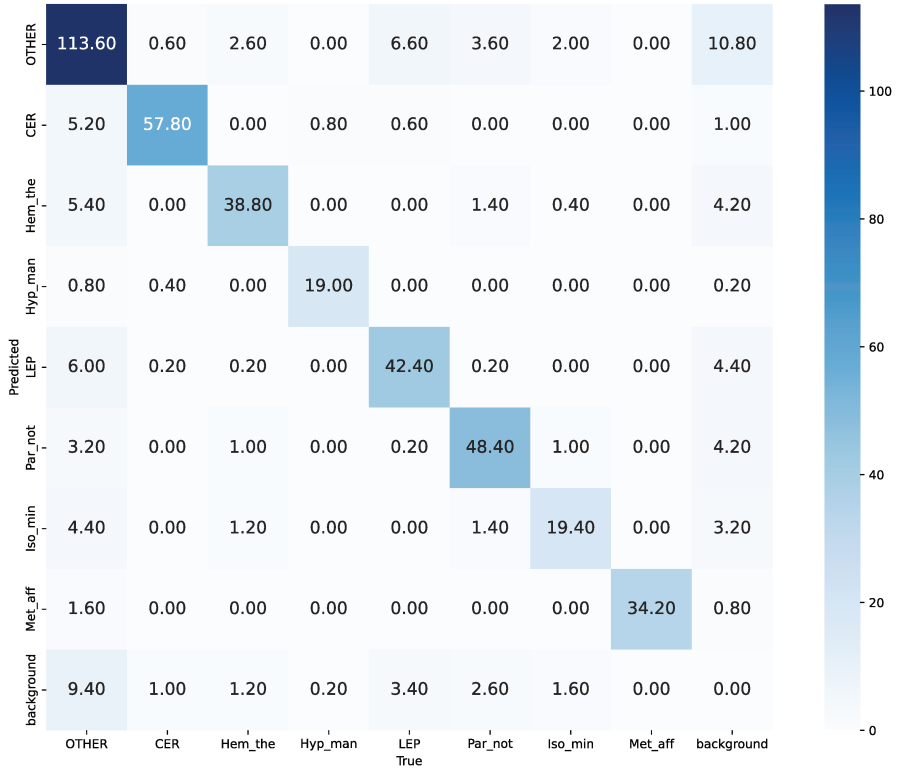


FIGURE 30 – Matrice de confusion de YOLOv5x6.

En analysant les erreurs faites par les modèles à travers ces matrices, on observe que YOLOv5x6 commet principalement trois types d'erreurs : la confusion des Collemboles avec l'arrière-plan ("background"), la confusion avec la catégorie "Autre" ("OTHER"), ainsi que la confusion entre certaines espèces d'intérêt, notamment *Isotomiella minor*, *Parisotoma notabilis*, et *Hemisotoma thermophila*. Par ailleurs, certains spécimens de la catégorie "OTHER" sont confondus avec les espèces d'intérêt, à l'exception notable de *Metaphorura affinis* et *Hypogastrura manubrialis*, qui sont mieux identifiés.

Faster R-CNN (1280) montre des erreurs similaires, mais de manière plus marquée. En plus des confusions observées avec YOLOv5x6, Faster R-CNN commet davantage d'erreurs en confondant *Hypogastrura manubrialis* avec "CER", ainsi que *Metaphorura affinis* avec *Hemisotoma thermophila*.

Ces erreurs peuvent en partie s'expliquer par la qualité variable des images. Par exemple, certaines vieilles lames provenant de projets plus anciens contiennent des Collemboles de qualité inférieure, compliquant leur détection et identification. De plus, la catégorie "OTHER" pose également des défis, car le modèle peut échouer à reconnaître un Collembole en raison de son état, du bruit sur l'image ou de l'incapacité à discerner les caractéristiques nécessaires à une identification correcte.

La dernière difficulté concerne les erreurs attendues dans les groupes de confusion. Il existe deux principaux groupes de confusion : le premier regroupe "CER" et *Hypogastrura manubrialis*, tandis que le second inclut *Parisotoma notabilis*, *Hemisotoma thermophila*,

et *Isotomiella minor*. Bien que YOLOv5x6 fasse globalement moins d'erreurs que Faster R-CNN dans toutes ces catégories, la réelle problématique réside dans les erreurs entre ces groupes de confusion. Si un Collembole est mal identifié comme appartenant à la catégorie "OTHER", il sera probablement ré-identifié par un expert. En revanche, une identification erronée dans les classes d'intérêt, comme celles dans les groupes de confusion, est plus problématique, car elle risque de ne jamais être corrigée, rendant ainsi le modèle beaucoup moins fiable.

Pour compléter notre compréhension des performances de classification et des erreurs spécifiques des modèles, les courbes de Rappel-Précision offrent une perspective supplémentaire. En visualisant ces courbes, nous pouvons analyser comment la précision et le rappel varient avec différents seuils de décision pour chaque modèle, ce qui aide à évaluer leur efficacité à discriminer entre les classes.

### 3.4.3 Courbe de Rappel-Précision

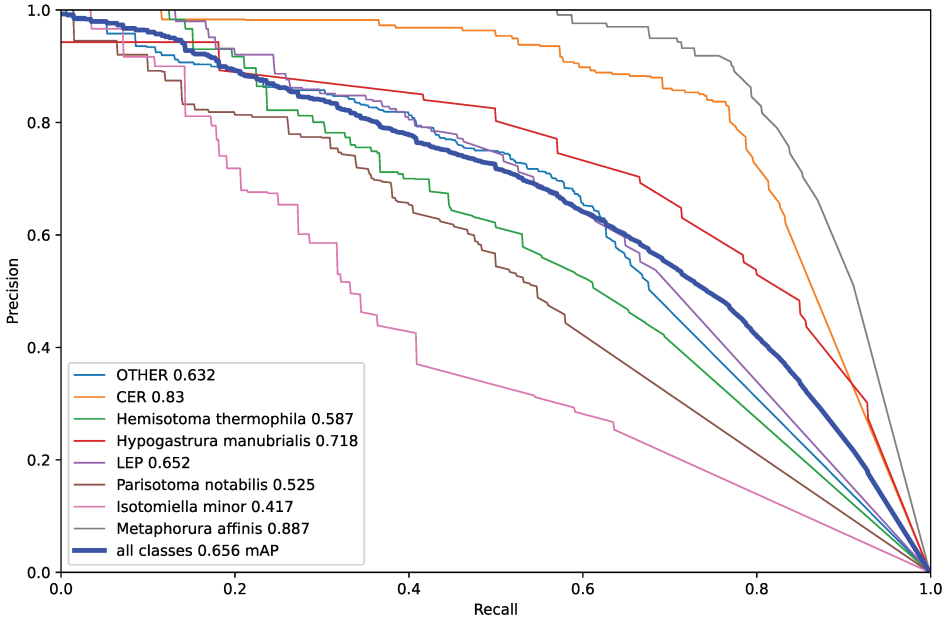


FIGURE 31 – Courbe de Rappel-Précision de Faster R-CNN, l'AP de chaque espèce est l'aire sous leur courbe, le mAP est la moyenne de tous les AP.

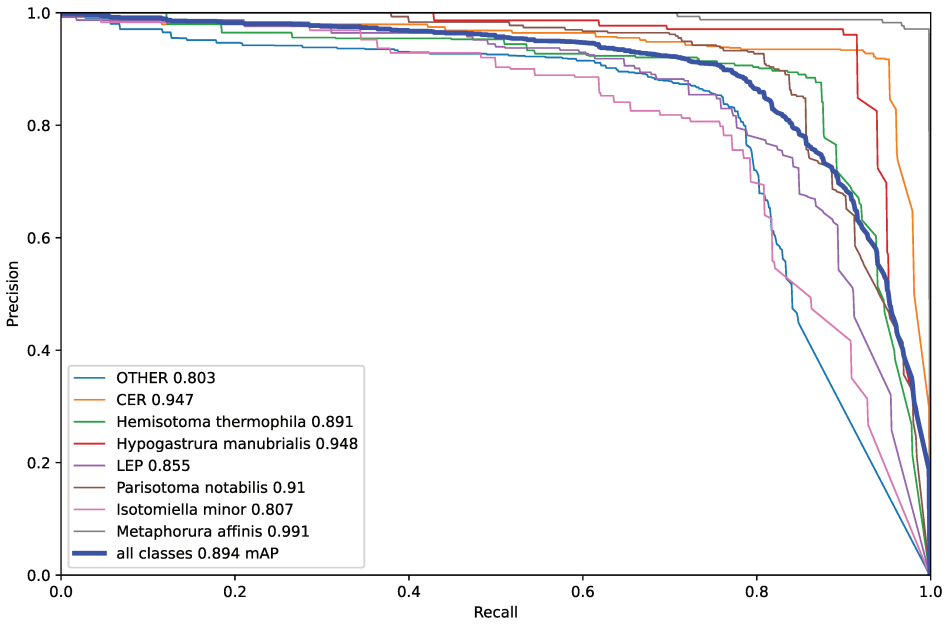


FIGURE 32 – Courbe de Rappel-Précision de YOLOv5x6, l'AP de chaque espèce est l'aire sous leur courbe, le mAP est la moyenne de tous les AP.

Les courbes de Rappel-Précision offrent une perspective supplémentaire sur les performances des modèles Faster R-CNN et YOLOv5x6. La performance inférieure et la plus grande variabilité observée dans la courbe de Rappel-Précision de Faster R-CNN (Figure 31) par rapport à celle de YOLOv5x6 (Figure 32) correspondent aux résultats anticipés suite à notre analyse des performances globales. Nous observons une variation plus marquée pour le modèle Faster R-CNN, ce qui peut indiquer une moins bonne généralisation ou une sensibilité accrue aux spécificités des données de test. Cette grande variance entre les classes suggère également une forte variance des résultats au sein même de chaque fold, compromettant ainsi la fiabilité des estimations de performance du modèle.

En analysant plus en détail les courbes de Rappel-Précision pour YOLOv5x6, on constate que ce modèle parvient à maintenir un bon équilibre entre précision et rappel. YOLOv5x6 conserve généralement une précision élevée, même avec un rappel relativement faible. Les espèces *Metaphorura affinis* et CER se démarquent en conservant une précision élevée à mesure que le rappel augmente, démontrant une robustesse notable dans diverses conditions de détection.

À l'inverse, la courbe de Rappel-Précision pour Faster R-CNN montre un déclin beaucoup plus rapide vers des valeurs quasi nulles pour la plupart des espèces. Les résultats de Faster R-CNN varient également beaucoup plus que ceux de YOLOv5x6 en fonction des changements de rappel, indiquant une fiabilité moindre des prédictions. Cela signifie que l'ajout d'une nouvelle espèce dans le cadre de la détection entraînerait une imprévisibilité accrue des performances de Faster R-CNN en termes de rappel, ce qui compromettrait la fiabilité et la sécurité du modèle pour une utilisation généralisée. En revanche, YO-

LOv5x6 présente une plus faible variabilité, offrant ainsi une fiabilité supérieure pour les prédictions, particulièrement dans le contexte de l'introduction de nouvelles espèces.

Ces analyses soulignent la supériorité de YOLOv5x6 par rapport à Faster R-CNN, notamment en ce qui concerne sa capacité à maintenir une précision plus stable et fiable à différents niveaux de rappel. Cette robustesse dans la détection en fait un candidat idéal pour une utilisation dans des applications où la variabilité des spécimens est courante. Cependant, pour mieux évaluer la capacité réelle de YOLOv5x6 à rivaliser avec des approches humaines, il est nécessaire de comparer directement ses performances avec celles des experts. Cette comparaison nous permettra de mesurer dans quelle mesure le modèle peut se substituer à l'expertise humaine pour l'identification des Collemboles, particulièrement dans les conditions de zoom standardisées que nous avons choisies.

## 3.5 Comparaison avec les experts

L'objectif de cette comparaison est de déterminer si le modèle YOLOv5x6 peut rivaliser avec l'expertise humaine en termes d'identification des Collemboles. Pour cela, nous avons demandé aux experts d'identifier les Collemboles sur les mêmes images utilisées par le modèle, c'est-à-dire des images prises avec un zoom x50. Ce niveau de zoom standardisé a été choisi pour garantir l'uniformité des données, bien que cela puisse limiter la capacité des experts à identifier précisément certaines espèces.

Les données ont été labellisées par des experts utilisant divers niveaux de zoom sur un microscope. Toutefois, pour cette comparaison, toutes les images ont été prises avec un zoom de x50 pour standardiser le processus. Ce niveau de zoom est insuffisant pour permettre une identification précise par les experts, empêchant ainsi toute comparaison directe des performances du modèle avec celles des experts dans des conditions optimales.

Le modèle Yolov5x6 a été choisi pour cette comparaison en raison de sa capacité à produire des résultats stables et précis, essentiels pour notre objectif de comparaison avec les évaluations humaines. Un seuil de précision de 95 % par espèce a été fixé pour garantir que les performances du modèle soient suffisamment élevées pour rivaliser avec les évaluations humaines dans des conditions idéales.

Voici le Tableau 3.3 qui détaille les précisions obtenues à différents niveaux de rappel, illustrant les performances du modèle Yolov5x6 :

TABLE 3.3 – Précision par espèce pour 20%, 50%, 80% and 95% rappel

| Espèce / Rappel | 20% | 50% | 80% | 95% |
|---|---|---|---|---|
| *OTHER* | 0.950 | 0.925 | 0.717 | 0.145 |
| *CER* | 0.979 | 0.968 | 0.935 | 0.917 |
| *Hemisotoma thermophila* | 0.964 | 0.952 | 0.903 | 0.448 |
| *Hypogastrura manubrialis* | 1 | 0.986 | 0.970 | 0.573 |
| *LEP* | 0.987 | 0.939 | 0.777 | 0.333 |
| *Parisotoma notabilis* | 1 | 0.983 | 0.927 | 0.472 |
| *Isotomiella minor* | 0.983 | 0.903 | 0.696 | 0.184 |
| *Metaphorura affinis* | 1 | 1 | 0.987 | 0.982 |

Les résultats montrent que bien que la majorité des espèces atteignent une précision supérieure à 95 % à un rappel de 50 % la précision diminue significativement à mesure que le rappel augmente. Cependant, *Metaphorura affinis* et CER maintiennent des précisions élevées même à un rappel de 95 % démontrant la robustesse du modèle dans ces cas spécifiques. Malheureusement, en raison des contraintes techniques liées au zoom insuffisant, les experts sont incapables de réaliser des identifications en raison d'un manque d'informations disponibles sur l'image. Malgré ces limitations, les performances de Yolov5x6 confirment sa capacité à atteindre des niveaux de précision élevés, conformes aux attentes pour une application pratique fiable.

## 3.6 Conclusion

Cette étude a comparé les performances des modèles de détection d'objets en une étape (YOLOv5) et en deux étapes (Faster R-CNN) pour identifier les Collemboles sur des images de lames de microscope. Les Collemboles, en tant que bio-indicateurs essentiels de la santé des sols, nécessitent une identification précise pour évaluer les effets des pratiques agricoles et de la pollution des sols. Les résultats ont montré que les modèles YOLOv5, en particulier YOLOv5x6, sont beaucoup plus sûr que Faster R-CNN. Les résultats de YOLOv5x6 varies beaucoup moins et sont cohérents au contraire des résultats de Faster R-CNN qui ont tendance à être plus confus, variants et incertains. Malgré cela Faster R-CNN réussit quand même partiellement à résoudre une tâche que les experts sont incapable d'accomplir, c'est à dire identifier des Collemboles en utilisant des images de Collemboles sur lames photographié avec un zoom faible de $\times 50$. En conclusion, YOLOv5x6 s'est avéré être le modèle le plus performant pour la détection et l'identification des Collemboles, démontrant son grand potentiel d'utilisation en écologie.

Chapitre 4

# Étude du biais

## 4.1 Introduction

Dans notre étude sur l'identification des collemboles, les versions du modèle YOLOv5, notamment YOLOv5x6, ont montré des performances impressionnantes, même à un faible grossissement (x50), un niveau de zoom qui empêche les experts humains d'identifier correctement les espèces en raison du manque de détails visibles. La capacité du modèle à surpasser les experts humains dans ces conditions spécifiques met en évidence son efficacité. Cependant, cette efficacité exceptionnelle soulève des questions importantes sur la capacité du modèle à généraliser au-delà des conditions expérimentales spécifiques.

Pour évaluer si le modèle YOLOv5x6 s'appuie sur les caractéristiques biologiques pertinentes des collemboles ou s'il est influencé par des éléments de bruit de fond propres aux conditions expérimentales, nous avons entrepris une série d'expérimentations. Ces tests sont essentiels pour garantir que, lors de l'identification d'un collembole, le modèle utilise effectivement ses traits distinctifs plutôt que des artefacts liés à l'image et son acquisition ou au contexte du projet.

Il est crucial d'exploiter des ensembles de données diversifiés et d'appliquer des méthodes rigoureuses d'évaluation et de validation. Cette approche est nécessaire pour s'assurer que le modèle apprend à partir des caractéristiques intrinsèques des espèces, et non des particularités des données d'entraînement. Si le modèle se base trop sur des caractéristiques non pertinentes, spécifiques à certains projets, sa capacité à réaliser des identifications fiables sur des ensembles de données non familiers sera significativement limitée.

Dans ce chapitre, nous émettons l'hypothèse que le modèle pourrait être biaisé par les conditions spécifiques du projet, et nous cherchons à comprendre la nature exacte de ce biais.

## 4.2 Grad-CAM

Pour tester l'hypothèse selon laquelle les modèles utilisent les caractéristiques des Collemboles pour faire une prédiction, nous avons appliqué la méthode Grad-CAM (Gradient-weighted Class Activation Mapping) sur le modèle Yolov5x6.

Le Grad-CAM (Selvaraju et al., 2017) permet de visualiser les zones de l'image sur lesquelles le modèle se focalise pour faire ses prédictions, en générant une carte de chaleur qui met en évidence les régions importantes pour la classification. Cette technique utilise les gradients des scores de classification à travers les couches convolutionnelles du modèle pour produire cette carte de chaleur, montrant ainsi quelles parties de l'image ont le plus contribué à la décision du modèle.

L'application de Grad-CAM sur Yolov5x6 présente une certaine complexité en raison de la structure spécifique du modèle. Contrairement à un modèle de classification classique, où nous avons uniquement un problème de classification, YOLOv5 détermine aussi une taille et position relative. Cela rend difficile la localisation précise des features maps responsables de l'identification de l'objet.

Dans un premier temps, nous avons appliqué le Grad-CAM sur la couche précédant la détection. Cependant, les cartes de chaleur produites montraient que le gradient se concentrait principalement sur le centre de l'ancre qui avait détecté l'objet, et non sur les caractéristiques spécifiques du Collembole (Figure 33).

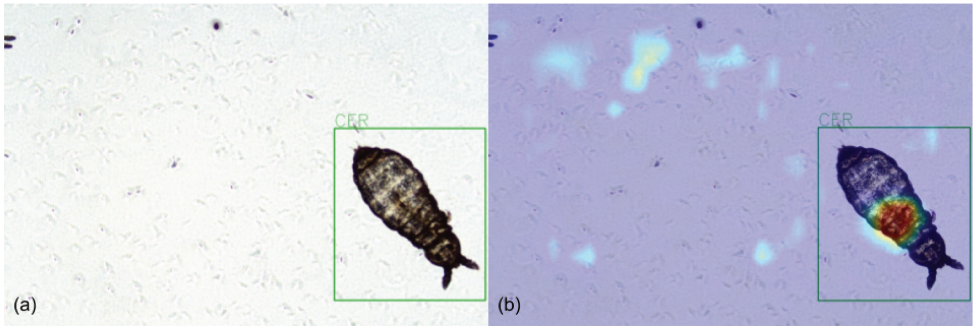


Figure 33 – Exemple de Grad-Cam appliqué sur la couche juste avant la détection (b), à partir de l'image originale sur laquelle le modèle a effectué une détection (a).

En remontant dans les couches du modèle et en observant les gradients sur des couches antérieures, nous avons remarqué que certaines couches se concentraient effectivement sur les Collemboles (Figure 34), mais il n'y avait pas une seule couche qui se distinguait comme étant responsable de l'identification. Il aurait été arbitraire de choisir une seule couche à analyser. De plus, la couche comportant des motifs intéressant change en fonction de l'image.

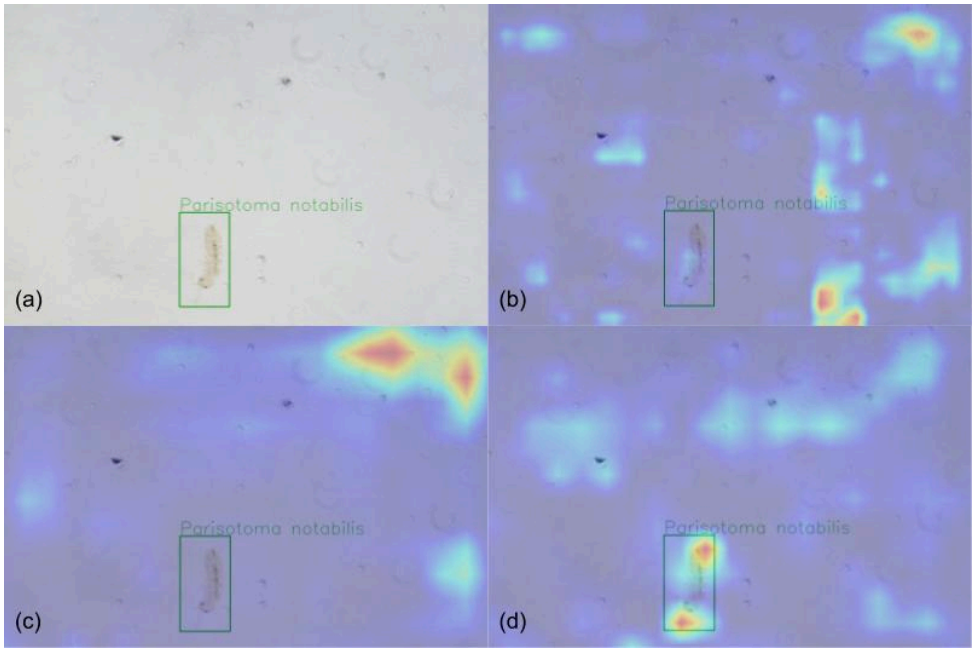


FIGURE 34 – Exemple de représentation de GRAD-CAM d'une image provenant de la couche 15 (b), de la couche 12 (c), ou de la couche 9 (d), à partir de l'image originale sur laquelle le modèle a effectué une détection (a)

Pour remédier à ce problème, nous avons sélectionné certaines couches spécifiques et fusionné leurs cartes Grad-CAM via une agrégation par moyenne afin d'obtenir une vue d'ensemble plus représentative de l'utilisation de l'image par le modèle lorsqu'il identifie un Collembole. Les couches 9, 16, et 20 ont été choisies pour cette fusion (Figure 35), car elles représentent des points d'insertion du gradient avant que celui-ci ne se concentre principalement sur la détection de l'objet.

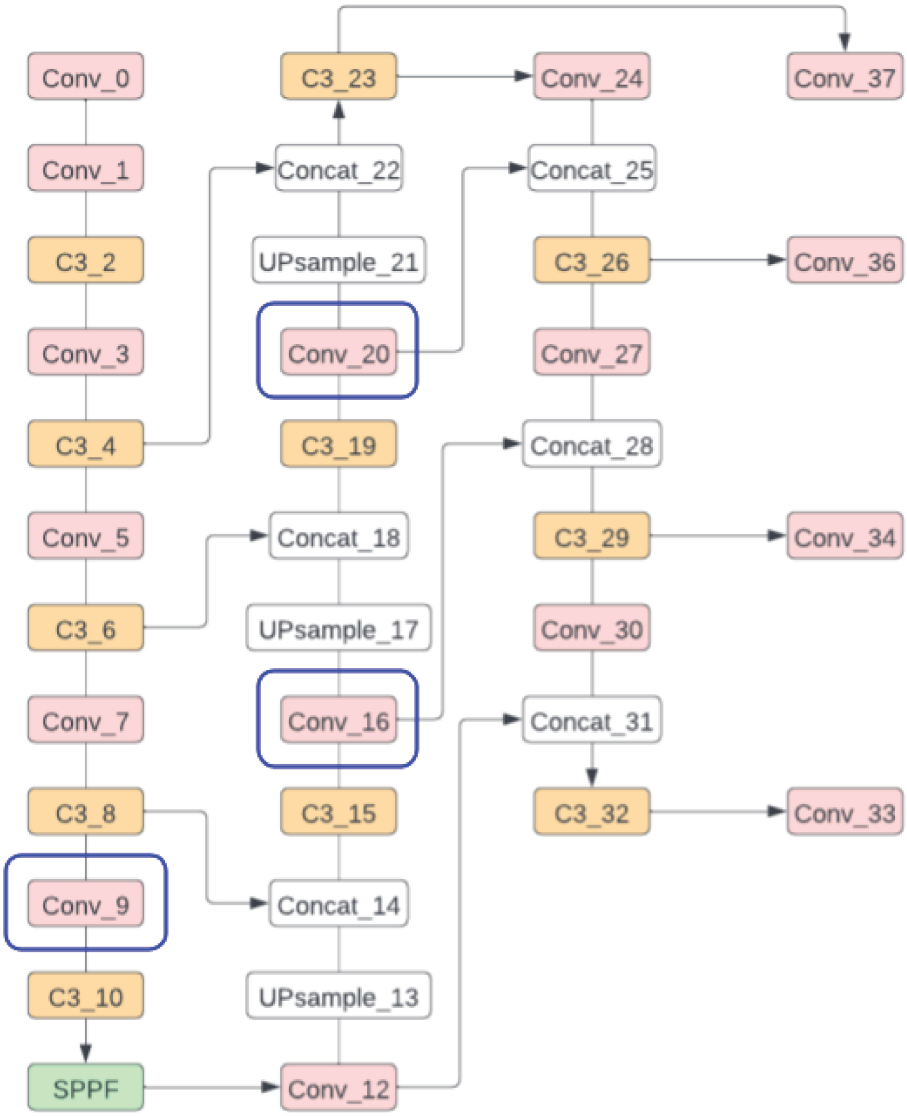


FIGURE 35 – Sur la figure, on peut voir l'architecture de Yolov5x6. Les couches entourées en bleu représentent celles qui ont été sélectionnées pour l'agrégation des Grad-CAM.

Il est important de noter que la fusion de plusieurs Grad-CAMs rend l'intensité du gradient moins significative, car il est difficile de déterminer l'importance relative des features provenant de différentes couches uniquement à partir de leur gradient respectif. Ce qui nous intéresse principalement dans cette fusion est de vérifier que le modèle focalise bien sur le Collembole lors de la prise de décision.

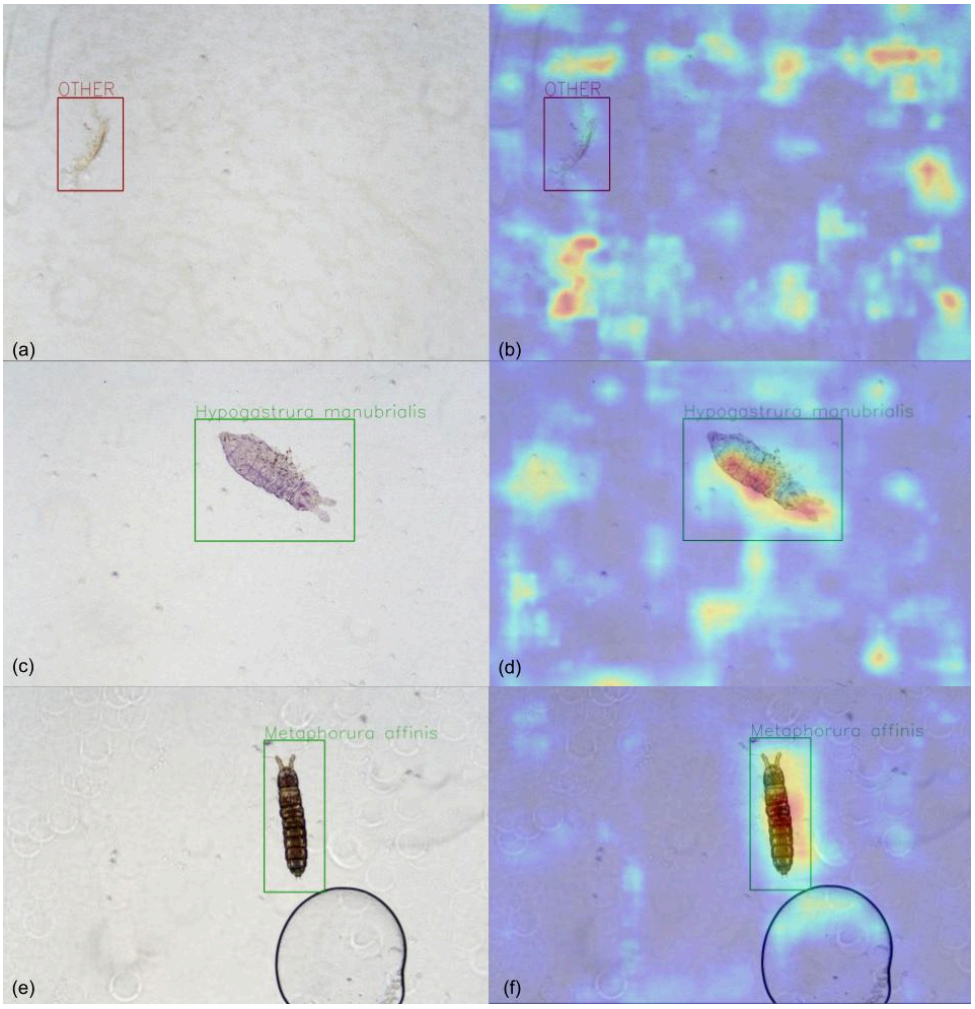


FIGURE 36 – Dans la première ligne, l'image (a) montre une mauvaise détection effectuée par le modèle, accompagnée à droite (b) de son Grad-CAM agrégé. Sur la deuxième ligne, l'image (c) illustre une détection correcte, avec à droite (d) le Grad-CAM agrégé correspondant. Enfin, sur la troisième ligne, l'image (e) présente une bonne détection, et à droite (f), le Grad-CAM agrégé associé.

Dans la Figure 36, nous présentons plusieurs exemples de fusion de Grad-CAM dans différents scénarios. Par exemple, pour le Collembole (a), où le modèle a effectué une mauvaise détection, on observe sur son Grad-CAM (b) que l'attention du modèle se porte principalement sur le fond de la lame, et non sur le Collembole.

À l'inverse, dans le cas d'une détection correcte, illustrée par le Collembole (c), on remarque sur le Grad-CAM (d) que l'attention du modèle est davantage concentrée sur le Collembole, bien que le fond de la lame soit encore observé, mais de façon moins marquée. Dans certains cas, le modèle ne prête pratiquement aucune attention au fond, comme le montre l'exemple du Collembole (e), correctement identifié, et son Grad-CAM (f), où l'attention est presque exclusivement portée sur le Collembole.

Ces observations soulèvent la question de savoir si le modèle ne s'appuie pas, en partie, sur le fond de la lame pour identifier les Collemboles, en particulier ceux plus difficiles

à reconnaître. Les Grad-CAM révèlent une attention variable accordée au fond de la lame, mais il reste difficile de déterminer précisément l'importance de cette attention dans chaque cas. Une analyse plus approfondie est donc nécessaire pour évaluer l'impact potentiel de ce biais sur les performances du modèle.

## 4.3 Changement de projet

Dans le prolongement de l'analyse des Grad-CAM, une deuxième approche a été mise en place pour étudier le comportement du modèle et son potentiel biais lié au fond des images. Cette méthode consistait à modifier l'arrière-plan des images de Collemboles déjà correctement identifiés et à observer la réaction du modèle face à ces changements. Plus précisément, nous avons extrait des spécimens de Collemboles de leurs images d'origine et les avons replacés sur des fonds provenant de différents projets où leurs espèces étaient absentes ou minoritaires (Figure 37). Pour cette analyse, nous avons sélectionné 10 spécimens par espèce d'intérêt (hors catégorie "AUTRE"), soit 70 spécimens au total, chaque spécimen ayant été au préalable correctement identifié par le modèle.

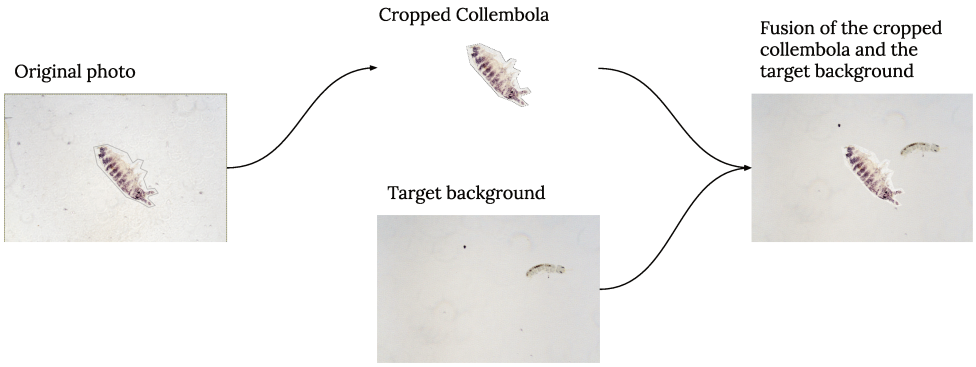


FIGURE 37 – Nous changeons le fond des Collemboles en les découpant de leur image d'origine et en les collant sur le fond cible provenant d'un projet différent.

L'objectif de cette expérience est de vérifier si les identifications restent correcte lorsque les Collemboles sont placés dans des projets où leur espèce est peu représentée ou absente. Cela permettra de déterminer si le fond de la lame constitue une source d'information utilisée par le modèle. En plaçant les Collemboles sur un fond qui ne fournit aucune information pertinente, nous pourrons observer si le modèle échoue à réaliser une réidentification correcte. Un tel échec indiquerait une dépendance du modèle au bruit de fond pour ses prédictions.

TABLE 4.1 – Nombre d'erreurs pour dix Collemboles par taxon lors du changement de l'arrière-plan du spécimen

| Taxon | Nombre d'erreurs sur 10 spécimens |
|---|---|
| *CER* | 1 |
| *Hemisotoma thermophila* | 0 |
| *Hypogastrura manubrialis* | 1 |
| *Isotomiella minor* | 3 |
| *LEP* | 1 |
| *Metaphorura affinis* | 1 |
| *Parisotoma notabilis* | 4 |
| Totale | 11 |

Après avoir appliqué le changement de fond et demandé au modèle de ré-identifier chacun des Collemboles précédemment correctement identifié, les résultats ont été compilés dans le Tableau 4.1. 11 erreurs on été relevées au total sur les 70 spécimens. Pour la plupart des espèces, les résultats de l'identification faite par le modèle change peu ou pas, le modèle semble être toujours capable d'identifier correctement l'espèce malgré le changement de fond. Cependant, deux espèces d'intérêt, *Parisotoma notabilis* et *Isotomiella minor*, ont obtenu de moins bons résultats lorsque le projet de fond était modifié. Ces deux espèces représentent 7 des 11 erreurs commises par le modèle, comme le montrent les résultats (Tableau 4.1). Il est probable que le modèle s'appuie sur le fond pour l'identification de ces deux espèces, car elles semblent plus difficiles à reconnaître sans le contexte du projet original. Toutefois, la majorité des détections pour les autres espèces d'intérêt reste correcte, ce qui suggère que l'influence du fond de la lame sur l'identification des Collemboles est relativement faible dans l'ensemble. *Isotomiella minor* étant l'espèces la plus difficile à identifier pour notre modèle (Tableau 3.1), on peut supposer que le modèle utilise ce biais pour compenser un manque de données. Le biais concernant *Parisotoma notabilis* est d'autant plus surprenant que cette espèce n'est généralement pas considérée comme difficile à identifier. Une hypothèse pourrait être que cette espèce est en réalité plus complexe à identifier qu'initialement supposé, et que le biais lié au fond a artificiellement amélioré les résultats du benchmark.

En utilisant à nouveau la technique du Grad-CAM avec une agrégation par moyenne sur les couches sélectionnées, on peut observer sur la Figure 38 deux exemple de Collembole de l'espèce CER correctement identifiés malgré un changement de fond de lame. Les Grad-CAM reste bien centré autour des Collemboles avant et après le changement de fond.

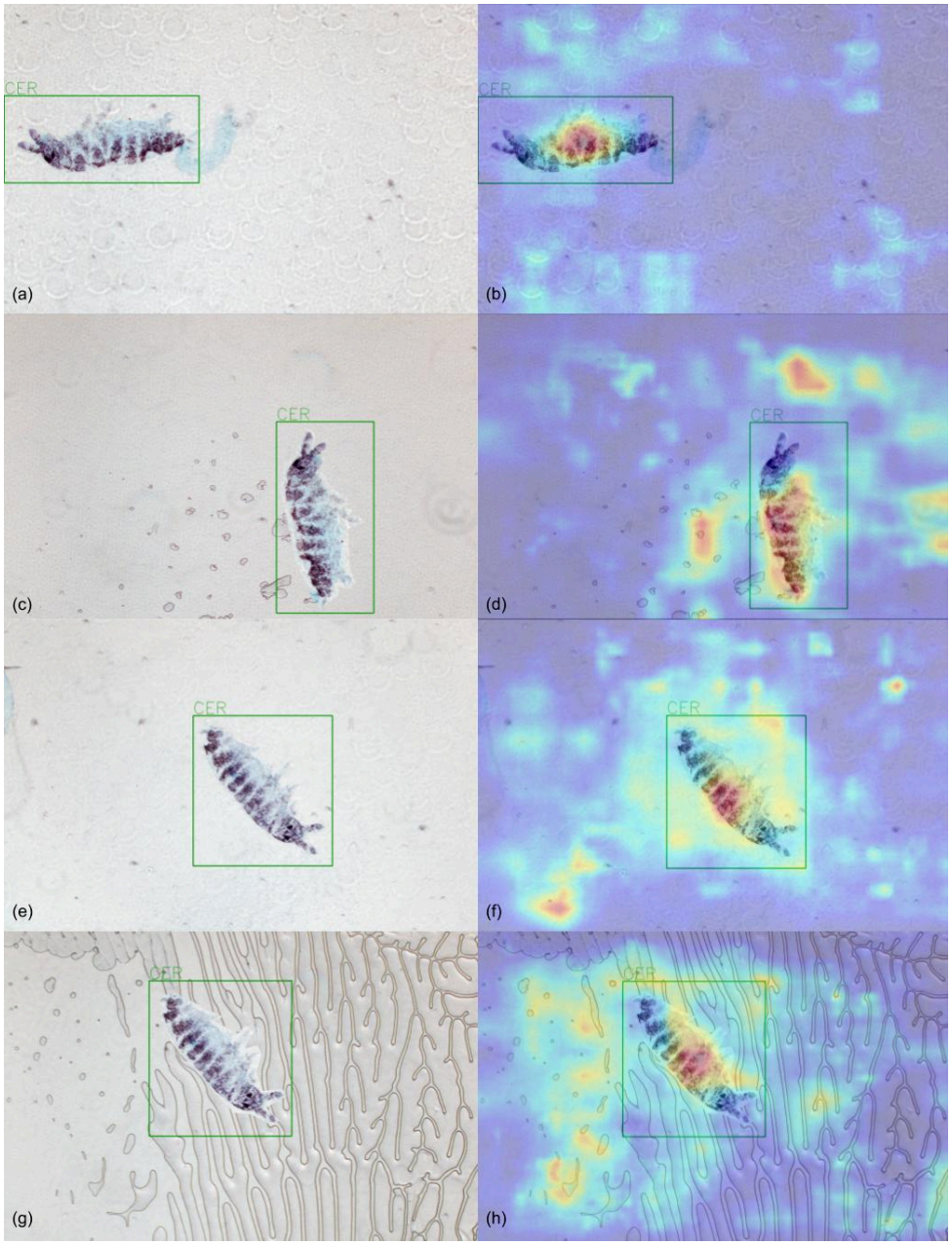


FIGURE 38 – Dans l'image (a), on voit l'image originale où le modèle a réalisé une détection correcte. À droite (b), on observe les Grad-CAM agrégées correspondantes. En dessous (c), l'image montre le même Collembole après le changement de fond, avec à droite (d) les Grad-CAM agrégées. Le même processus est répété pour les quatre images suivantes : d'abord, la deuxième image originale (e) et ses cartes Grad-CAM agrégées (f), suivies en dessous par le même Collembole après le changement de fond (g) avec, à droite (h), les Grad-CAM agrégées associées.

Dans l'image (a), on voit l'image originale où le modèle a réalisé une détection correcte. À droite (b), on observe les Grad-CAM agrégées correspondantes. En dessous (c), l'image montre le même Collembole après le changement de fond, avec à droite (d) les Grad-CAM agrégées. Le même processus est répété pour les quatre images suivantes : d'abord, la deuxième image originale (e) et ses cartes Grad-CAM agrégées (f), suivies en dessous

par le même Collembole après le changement de fond (g) avec, à droite (h), les Grad-CAM agrégées associées.

En revanche, sur la Figure 39, on constate un passage d'une bonne identification d'un Collemboles de l'espèce CER et d'un Collembole de l'espèce *Parisotoma notabilis* à une mauvaise identification, *Hemisotoma thermophila* et *Isotomiella minor* . Avant le changement de fond, l'attention est principalement concentré sur les Collembole, mais après le changement de fond, le modèle ne porte plus attention au même endroit sur le Collembole, voir ne le regarde plus.

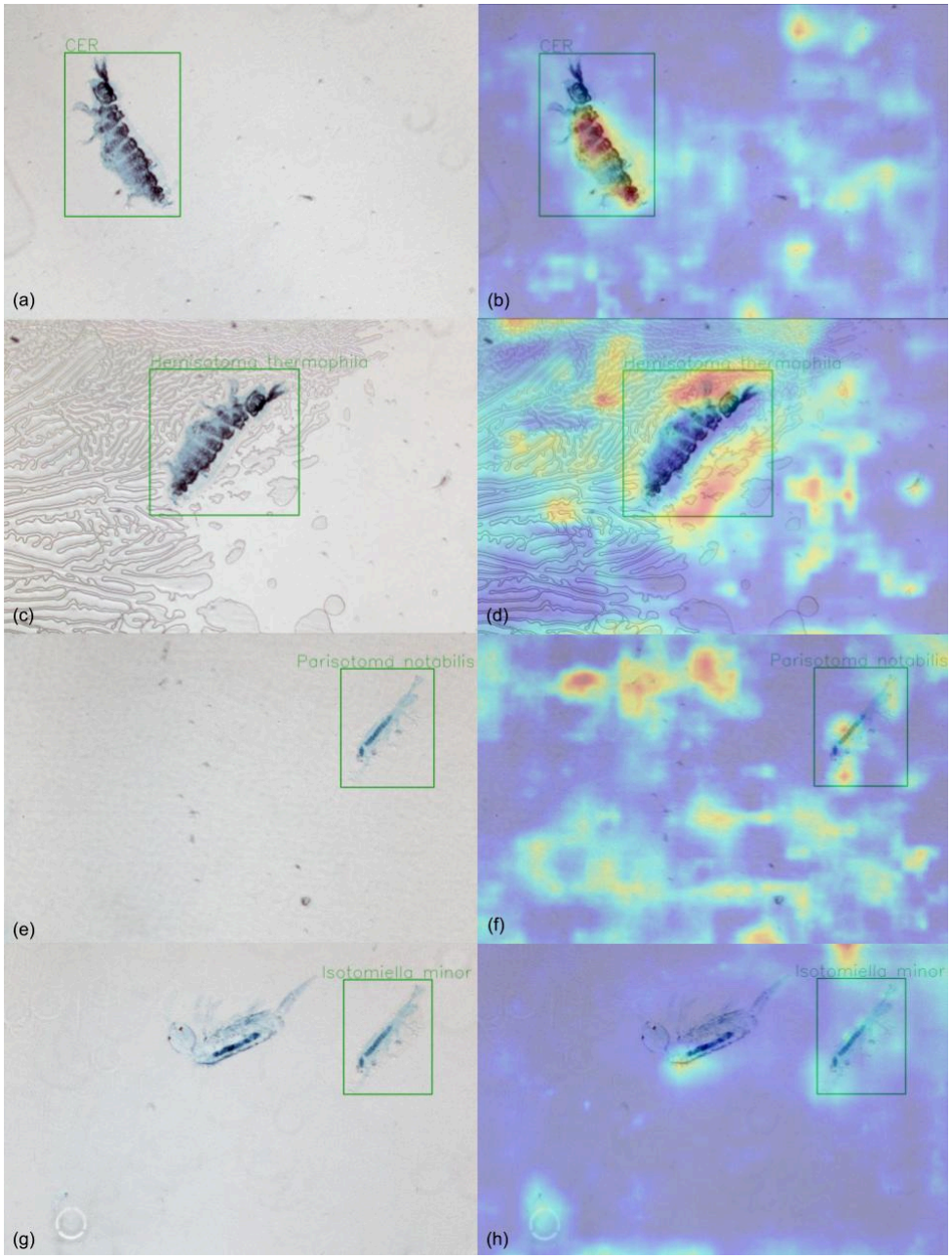


FIGURE 39 – Dans l'image (a), on voit l'image originale où le modèle a réalisé une détection incorrecte. À droite (b), on observe les Grad-CAM agrégées correspondantes. En dessous (c), l'image montre le même Collembole après le changement de fond, avec à droite (d) les Grad-CAM agrégées. Le même processus est répété pour les quatre images suivantes : d'abord, la deuxième image originale (e) et ses cartes Grad-CAM agrégées (f), suivies en dessous par le même Collembole après le changement de fond (g) avec, à droite (h), les Grad-CAM agrégées associées.

Ces résultats montrent que le réseau est sensible au type de fond, en particulier lorsque celui-ci est de mauvaise qualité (comme illustré dans la Figure 39). Ce type de fond est généralement associé à des projets plus anciens, ce qui peut introduire un biais affectant les espèces de Collemboles qu'ils contiennent. Ce phénomène est notamment observé avec le projet "Parcelles Lysimétriques", où l'impact du fond semble influencer les performances

du modèle. De plus, bien que ce biais spécifique ne touche pas fortement toutes les espèces, il est pertinent de se demander si d'autres biais, liés à l'apparence des Collemboles, pourraient affecter les performances du modèle. En effet, l'apparence des Collemboles peut varier selon les projets, notamment en fonction de la qualité des images ou des spécimens eux-mêmes. Ces variations entre projets peuvent introduire une variabilité supplémentaire dans les résultats du modèle, influençant potentiellement son efficacité à identifier certaines espèces de manière cohérente. Pour mieux comprendre l'impact de ce biais d'apparence, nous décidé d'approfondir l'impact de ce biais.

## 4.4 Impact du biais

Dans la continuité de notre analyse des biais, et après avoir exploré l'impact potentiel du fond sur les performances du modèle, nous avons étendu notre investigation au biais lié à l'apparence des Collemboles entre différents projets. Pour évaluer à quel point les modèles d'apprentissage profond sont affectés par un biais de projet, nous avons conçu une troisième expérience spécifiquement destinée à identifier ce type de biais.

Cette expérience consiste à comparer les performances d'un modèle entraîné sur 11 des 12 projets de Collemboles disponibles et évalué sur le 12ème projet, BISES, au benchmark global (Tableau 3.1) YOLOv5x6 benchmark. Ce dernier est l'agrégation des résultats de plusieurs modèles, chacun ayant été entraîné sur différentes parties (cross-validation) du jeu de données global. Cela permet de quantifier l'impact du biais de projet et d'apparence entre les différents jeux de données.

Le modèle YOLOv5x6 BISES a été entraîné sur tous les projets de Collemboles à l'exception de BISES, et ensuite évalué spécifiquement sur ce dernier projet. Cette approche garantit que le modèle n'a pas été exposé au projet BISES pendant l'entraînement, ce qui empêche l'utilisation d'informations spécifiques à ce projet pour identifier les Collemboles. Le projet BISES a été choisi en raison de sa taille et de la diversité de ses données, assurant ainsi une meilleure représentativité des résultats.

Il est également important de noter que le projet BISES ne contient aucune occurrence de l'espèce *Hypogastrura manubrialis*. Cette absence introduit une contrainte supplémentaire dans l'évaluation du modèle, car *Hypogastrura manubrialis* est présente dans plusieurs autres projets d'entraînement. De plus, comme aucun des projets de Collemboles disponibles ne contient l'ensemble des espèces présentes dans le jeu de données global, il est nécessaire de composer avec cette limitation pour évaluer correctement la capacité du modèle à généraliser ses prédictions.

Le modèle YOLOv5x6 benchmark, quant à lui, est issu d'une agrégation des résultats de plusieurs modèles entraînés sur différente cross de l'ensemble des projets. Cela permet d'obtenir une évaluation globale des performances des modèles sur l'ensemble des données disponibles. Les métriques utilisées pour évaluer les performances sont le mAP

(mean Average Precision) et l'AP (Average Precision) par espèce, afin de permettre une comparaison directe des résultats.

Les performances du modèle YOLOv5x6 BISES ont ensuite été comparées à celles du YOLOv5x6 benchmark. Les résultats sont résumés dans Tableau 4.2. La différence de performance entre les deux modèles, illustrée par la ligne "Perte de performance", révèle l'impact du biais de projet sur les capacités de généralisation du modèle.

TABLE 4.2 – Résultats de l'évaluation du modèle Yolov5x6 BISES de Yolov5x6 benchmark. La ligne "Perte de performance" fait référence à la différence de performance entre les deux modèles

| | mAP | "AUTRE" AP | "CER" AP | "Hem the" AP | "Iso min" AP | "LEP" AP | "Met aff" AP | "Par not" AP |
|---|---|---|---|---|---|---|---|---|
| Yolov5x6 BISES | 0.605 | 0.419 | 0.809 | 0.426 | 0.692 | 0.561 | 0.644 | 0.686 |
| Yolov5x6 benchmark | 0.854 | 0.798 | 0.933 | 0.808 | 0.743 | 0.923 | 0.922 | 0.849 |
| Perte de performance | -0.249 | -0.379 | -0.124 | -0.382 | -0.051 | -0.362 | -0.278 | -0.163 |

Le modèle YOLOv5x6 évalué sur BISES obtient un mAP global de 0.605, contre 0.854 pour le YOLOv5x6 benchmark, ce qui représente une perte de 0.249. Cette baisse indique que le modèle, bien qu'entraîné sur plusieurs projets, éprouve des difficultés à généraliser lorsqu'il est confronté aux spécificités du projet BISES, révélant un biais de projet important. Cela montre que les données spécifiques du projet BISES présentent des caractéristiques différentes qui impactent la performance du modèle.

Certaines classes montrent des baisses de performance significatives, révélant les difficultés du modèle à généraliser correctement pour certaines espèces entre les différents projets.

Les classes "Hem The" (Hem the), "LEP", "Met aff", et "AUTRE" présentent les pertes les plus importantes. La baisse la plus marquée concerne "Hem The", avec une perte de 0.382 (0.426 contre 0.808). Cela montre que cette espèce présente de grandes variations d'apparence ou de contexte entre les projets, ce qui complique la tâche du modèle. Les classes "LEP" et "Met aff" subissent des baisses respectives de 0.362 (0.561 contre 0.923) et 0.278 (0.644 contre 0.922), indiquant également des difficultés similaires de généralisation entre les projets pour ces espèces. De plus, la classe "AUTRE" affiche une perte de 0.379 (0.419 contre 0.798), reflétant la difficulté à identifier correctement des espèces plus rares ou moins représentées dans le projet BISES.

À l'inverse, certaines classes montrent des pertes de performance plus modérées. Par exemple, la classe "CER" présente une baisse de 0.124 (0.809 contre 0.933), une diminution certes significative, mais moins prononcée que dans d'autres classes. Cela peut s'expliquer par une meilleure cohérence de cette espèce entre les différents projets, bien que des variations contextuelles ou visuelles à BISES compliquent encore la tâche du modèle.

Enfin, la classe "ISO MIN" présente la perte de performance la plus faible avec 0.051 (0.692 contre 0.743 dans le benchmark (Tableau 3.1), suggérant que cette espèce est relativement homogène entre les projets. Cela permet au modèle de mieux généraliser pour cette classe par rapport aux autres.

Les résultats révèlent une perte de performance généralisée lorsque le modèle YOLOv5x6 est évalué sur le projet BISES par rapport au benchmark global. La diminution du mAP global de 0.249 montre que le modèle est moins performant lorsqu'il est confronté à des données spécifiques qu'il n'a pas vues pendant l'entraînement.

Les baisses les plus marquées sont observées pour les espèces telles que "Hem The", "LEP", "Met aff", et "AUTRE", qui souffrent toutes de pertes importantes, en partie à cause de la variabilité des conditions de collecte et des différences intra-projet. Ces classes montrent que le modèle éprouve des difficultés à s'adapter aux variations visuelles et contextuelles entre les différents projets, rendant la généralisation plus difficile pour ces espèces.

Cependant, il est intéressant de noter que la classe "ISO MIN" subit la plus faible perte de performance (0.051), bien que cette espèce soit celle qui a obtenu les moins bons résultats dans le benchmark global (Tableau 3.1), de plus il a été démontré que "ISO MIN" est sujette à un biais de projet, car le modèle tend à utiliser davantage le fond de l'image pour l'identifier par rapport aux autres espèces. Il est donc surprenant que la différence de résultats soit aussi faible. Une explication possible est que cette espèce est particulièrement difficile à identifier en raison de ses caractéristiques visuelles, ce qui conduit le modèle à utiliser le fond comme repère lorsque l'identification devient difficile. Malgré cela, "ISO MIN" semble présenter une cohérence inter-projet plus élevée que d'autres espèces, ce qui rend sa généralisation plus facile pour le modèle, même si son identification repose davantage sur le contexte que sur les caractéristiques spécifiques de l'espèce elle-même.

Il est important de noter que ce biais de projet est principalement lié aux changements d'apparence des Collemboles entre les différents projets, plutôt qu'aux différences de fond dans les images. Ces variations d'apparence sont dues à plusieurs facteurs, notamment l'âge des Collemboles et les traitements qu'ils ont subis dans chaque projet (comme les méthodes de collecte et de préparation des échantillons). Par conséquent, ces résultats ne sont pas surprenants et sont en grande partie attribuables aux altérations naturelles des spécimens en fonction des conditions spécifiques de chaque projet. Cette variabilité est un phénomène normal et met en lumière les défis auxquels les modèles de détection d'objets doivent faire face lorsqu'ils sont appliqués à des données provenant de sources diverses. Ces résultats soulignent la nécessité de prendre en compte les changements d'apparence des Collemboles liés aux projets lors de l'entraînement et de l'évaluation des modèles, afin d'améliorer leur capacité à généraliser à de nouveaux contextes.

## 4.5 Conclusion

Cette étude a mis en évidence l'existence d'un biais de projet dans le modèle Yolov5x6 utilisé pour l'identification des Collemboles. Bien que ce modèle ait montré des performances remarquables, surpassant même les capacités humaines dans certaines conditions,

il est clair que les variations entre les projets, telles que l'apparence des Collemboles, l'âge des spécimens, et les méthodes de traitement des échantillons, ont un impact significatif sur les performances du modèle.

Les résultats montrent que, malgré une robustesse apparente, le modèle rencontre des difficultés à généraliser ses prédictions lorsqu'il est confronté à des données provenant de projets non inclus dans son entraînement. La baisse de performance observée, en particulier sur le projet BISES, souligne l'importance de ce biais de projet.

Plutôt que de considérer ce biais comme un obstacle, il pourrait être intéressant d'explorer des moyens d'intégrer les variations liées à ce biais de projet directement dans le modèle. En intégrant ces variations dès la phase d'entraînement, le modèle pourrait être mieux préparé à gérer les spécificités de nouveaux projets et à fournir des identifications plus fiables et généralisables. Cette approche pourrait améliorer la capacité du modèle à s'adapter à des contextes variés, offrant ainsi une plus grande utilité dans des applications pratiques où les conditions et les spécimens peuvent varier considérablement.

Chapitre 5

# Active learning

Les modèles d'apprentissage profond ont émergé comme des outils prometteurs pour automatiser l'identification des Collemboles, permettant ainsi de surmonter les limitations des méthodes manuelles. Cependant, l'identification automatisée des Collemboles est sujette à un biais de projet.

Le biais de projet est dû à plusieurs facteurs. L'âge du projet influe grandement sur l'apparence et l'état des Collemboles, des altérations pouvant entraîner la détérioration ou l'éclatement des spécimens sur la lame. En effet, les techniques de montage sur lame sont en définitive conçues pour altérer l'apparence globale des Collemboles (couleurs notamment) pour accéder plus facilement à l'observation de critères uniquement visibles à très fort grossissement ($\times 630$), déterminants pour identifier l'espèce (par exemple la position des soies sur les différents segments du corps de l'animal). Par ailleurs, sur la lame, les produits de fixation utilisés (Marc André 2) peuvent continuer à agir au fil du temps et modifier l'apparence des Collemboles. Enfin, avant fixation sur lame, les animaux ont été collectés dans une solution d'alcool à 70%, et selon le temps de séjour passé dans les flacons (de quelques jours à plusieurs semaines), l'apparence des animaux peut évoluer. Ainsi, un modèle entraîné sur des images provenant d'un projet spécifique peut ne pas généraliser efficacement à des images provenant d'autres projets avec des conditions différentes.

Pour adresser ce problème, il est crucial de disposer de techniques qui permettent aux modèles d'apprentissage profond de mieux généraliser à travers ces variations en intégrant le biais venant de ces nouveaux projets. C'est là qu'intervient l'apprentissage actif (Active Learning), en permettant au modèle de solliciter des annotations supplémentaires pour les échantillons sur lesquels il est le plus incertain. En d'autres termes, au lieu d'annoter aléatoirement des échantillons, l'apprentissage actif cible spécifiquement les échantillons qui amélioreront le plus la performance du modèle. En intégrant de manière optimale des images issues d'un nouveau projet dans l'ensemble d'entraînement du modèle, on introduit également des motifs spécifiques à ce projet. Cela permettrait d'améliorer considérablement la précision du modèle lorsqu'il réalise des prédictions sur des images provenant de ce projet.

Cependant, les méthodes traditionnelles d'apprentissage actif, qui se basent uniquement sur l'incertitude des prédictions, peuvent ne pas être suffisantes pour surmonter le biais de projet. Ces méthodes peuvent négliger des aspects importants tels que le nombre de prédictions par image, ce qui peut être critique lorsqu'il s'agit d'ajouter de nouvelles

classes ou de détecter des spécimens complexes à différencier d'autres Collemboles. Par conséquent, une approche plus sophistiquée est nécessaire.

Ce chapitre introduit deux méthodes complémentaires d'apprentissage actif pour répondre à ces défis. La première, appelée "Active Learning Extension" (ALE), vise à améliorer la performance des modèles de détection d'objets en prenant en compte non seulement l'incertitude des prédictions, mais aussi le nombre de prédictions par image. En intégrant ces deux aspects, ALE permet de sélectionner des échantillons d'entraînement qui maximisent l'apprentissage du modèle de manière plus efficace, afin d'améliorer les résultats sur de nouveaux projets et d'ajouter des espèces au pool d'espèces détectées.

La deuxième méthode repose sur l'algorithme "Isolation Forest", une technique de détection d'anomalies. Isolation Forest identifie les anomalies dans les données en isolant les points rares ou atypiques, ce qui permet de cibler des spécimens de Collemboles non représentés ou mal représentés dans le jeu de données d'entraînement. Cette approche est particulièrement utile pour identifier des variations extrêmes dans les spécimens et surmonter le biais des projets.

Notre hypothèse est que, grâce à l'apprentissage actif, notamment avec l'intégration des méthodes ALE et Isolation Forest, les modèles pourront s'adapter plus efficacement aux variations introduites par de nouveaux projets et espèces. Cela permettrait d'améliorer considérablement la capacité des modèles à généraliser sur des jeux de données, tout en incorporant de nouvelles espèces dans le pool de classes détectées.

## 5.1 Introduction à l'active learning

L'apprentissage actif joue un rôle crucial dans l'optimisation des performances des modèles tout en minimisant le besoin de nouvelles annotations. Cette approche est particulièrement bénéfique lorsque l'annotation des données est coûteuse en temps et nécessite une expertise importante. Différents scénarios et stratégies d'apprentissage actif permettent de cibler efficacement les exemples les plus informatifs pour améliorer le modèle.

Au cœur de l'apprentissage actif se trouve le concept d'échantillonnage, qui consiste à sélectionner un sous-ensemble d'exemples à partir d'un ensemble plus large de données non étiquetées. L'échantillonnage vise à identifier les données qui auront l'impact le plus significatif sur l'amélioration du modèle tout en réduisant les coûts d'annotation. Il existe plusieurs types d'échantillonnage adaptés à divers contextes.

L'échantillonnage aléatoire consiste à sélectionner des exemples au hasard, sans prendre en compte l'informativité ou l'incertitude des données. Bien que cette méthode soit simple, elle n'est pas toujours la plus efficace, car elle ne se concentre pas sur les données qui pourraient apporter le plus de valeur au modèle. En revanche, l'échantillonnage actif se base sur l'analyse des prédictions du modèle pour sélectionner les exemples les plus incertains ou informatifs. Cela permet d'optimiser l'apprentissage en annotant uniquement les données

qui contribueront le plus à l'amélioration des performances.

L'échantillonnage sélectif est utilisé dans des scénarios de flux de données en temps réel, comme la surveillance vidéo, où des décisions rapides doivent être prises pour sélectionner les exemples à annoter parmi les données qui arrivent en continu. Il permet une gestion dynamique des données dans des contextes où la rapidité est essentielle. Quant à l'échantillonnage basé sur des pools, il est appliqué lorsque de grandes quantités de données non étiquetées sont disponibles. Le modèle sélectionne les exemples les plus incertains ou les plus informatifs à partir de ce pool, ce qui maximise l'efficacité de l'annotation en se concentrant sur les données les plus utiles.

Ces différentes méthodes d'échantillonnage sont soutenues par des stratégies de requête qui aident à identifier les exemples les plus pertinents à annoter. Parmi elles, deux métriques d'incertitude couramment utilisées sont le Least Confidence (LC) et l'entropie.

La stratégie Least Confidence (LC) sélectionne les exemples pour lesquels le modèle est le moins confiant dans sa prédiction, en se basant sur la différence entre les probabilités prédites les plus élevées. Elle est formulée de la manière suivante :

$$LC = 1 - (p_1 - p_2)$$

Où $p_1$ est la probabilité prédite la plus élevée et $p_2$ est la deuxième plus élevée. Une valeur LC plus élevée indique une plus grande incertitude, faisant de ces échantillons des candidats idéaux pour l'annotation.

L'entropie, quant à elle, mesure l'incertitude globale en tenant compte de la distribution des probabilités sur toutes les classes possibles. Une distribution de probabilité plus uniforme suggère une incertitude plus grande, tandis qu'une distribution plus concentrée indique une plus grande confiance dans la prédiction. L'entropie est calculée avec la formule suivante :

$$E = -\sum_{i=1}^{N} p_i \log(p_i)$$

où $p_i$ est la probabilité attribuée à la classe $i$ et $N$ est le nombre de classes possibles. Des valeurs d'entropie plus élevées suggèrent une plus grande incertitude, rendant ces échantillons particulièrement informatifs pour une annotation supplémentaire.

Dans le cas des modèles de détection d'objets tels que YOLO ou Faster R-CNN, ces métriques d'incertitude sont utilisées pour identifier les images où le modèle est le plus incertain, ce qui permet de cibler les exemples nécessitant une annotation supplémentaire. De plus, des techniques comme l'incertitude des boîtes englobantes (Bounding Box Uncertainty) peuvent être utilisées pour améliorer la précision dans des scénarios complexes, en affinant la détection des objets.

En résumé, l'apprentissage actif, combiné à des stratégies d'échantillonnage adaptées et à des métriques d'incertitude telles que le Least Confidence et l'entropie, permet d'améliorer les performances des modèles tout en réduisant la charge de travail liée à l'annotation. Cette approche rend les modèles plus robustes et mieux adaptés à des applications variées, notamment dans des contextes où les ressources en données annotées sont limitées.

## 5.2 Méthodes proposées

### 5.2.1 Jeu de données

Le jeu de données utilisé dans cette étude a été initialement développé pour évaluer la détection des Collemboles sur des images de microscope à l'aide de techniques d'apprentissage profond (Oriol, Pasquet & Cortet, 2024a). Il comprend des spécimens de Collemboles collectés à partir de 12 projets différents, ce qui permet de couvrir une large diversité de situations et de s'assurer que le modèle est exposé à une variété de conditions de traitements.

Le jeu de données utilisé pour faire de l'apprentissage actif comprend les mêmes espèces que précédemment, avec l'ajout de deux nouvelles espèces pour simuler l'apport de nouvelles classes : *Pseudosinella alba* (PSE-ALB) et *Sphaeridia pumilis* (SPH-PUM) (Figure 40).

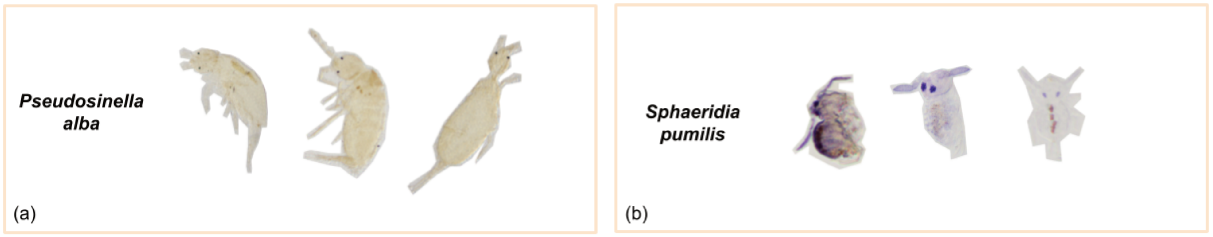


Figure 40 – Exemples de *Pseudosinella alba* (a) et *Sphaeridia pumilis* (b).

L'objectif étant de démontrer que les techniques d'apprentissage actif, spécifiquement ALE et Isolation Forest, sont capable d'intégrer le biais provenant de nouveaux projet, et d'intégrer de nouvelles espèces au pool d'espèces sélectionné pour être identifiées par le modèle, deux expérience sont proposées.

Table 5.1 – Distribution des annotations d'entraînement et la validation sur l'ensemble de données de base où on intègre deux nouvelles espèces via l'apprentissage actif.

| | AUTRE | CER | HEM THE | HYP MAN | ISO MIN | LEP | MET AFF | PAR NOT | PSE ALB | SPH PUM |
|---|---|---|---|---|---|---|---|---|---|---|
| Entraînement | 410 | 228 | 126 | 109 | 87 | 144 | 106 | 111 | 0 | 0 |
| Validation | 163 | 92 | 96 | 0 | 62 | 121 | 23 | 151 | 37 | 54 |

La première étudie l'évolution des performances de modèle quand on ajoute deux espèces grâce à l'apprentissage actif ; le jeu de données a été séparé de la manière suivante :

— **Ensemble d'entraînement** : Constitué de données provenant de 11 des 12 projets de Collemboles. Ces projets ont été utilisés pour entraîner le modèle de base avant l'utilisation de techniques d'active learning.
— **Ensemble de validation (BISES)** : BISES, le plus gros projet, a été réservé pour l'évaluation du modèle, simulant ainsi l'intégration d'un projet que le modèle n'a encore jamais vu. Il est important de noter que l'espèce *Hypogastrura manubrialis* n'est pas présente dans BISES, mais étant donné qu'aucun projet ne contient toutes les espèces d'intérêt à la fois, BISES a été choisi car il inclut le plus grand nombre d'espèces d'intérêt, le rendant ainsi particulièrement représentatif pour l'évaluation..

Pour simuler l'ajout d'espèces, toutes les instances des nouvelles classes (*Pseudosinella alba* et *Sphaeridia pumilis*) dans l'ensemble d'entraînement ont été masquées par un carré blanc (Figure 41) pour empêcher le modèle d'apprendre sur ces nouveaux taxons provenant d'un autre projet que BISES tout en préservant les annotations existantes sur les mêmes images. Cette technique assure que les nouvelles espèces ajoutées à travers l'apprentissage actif sont découvertes et intégrées de manière autonome par le modèle (Modèle de Base 1) .

Le Tableau 5.1 indique le nombre d'annotations par espèce dans le jeu de données de base où les nouvelles espèces sont cachées en entraînement.



FIGURE 41 – Exemple de Collembole obscurci à l'aide d'un carré blanc.

TABLE 5.2 – Distribution des annotations d'entraînement et la validation sur le jeu de données de base.

| | OTHER | CER | HEM THE | HYP MAN | ISO MIN | LEP | MET AFF | PAR NOT | PSE ALB | SPH PUM |
|---|---|---|---|---|---|---|---|---|---|---|
| Entraînement | 410 | 228 | 126 | 109 | 87 | 144 | 106 | 111 | 51 | 26 |
| Validation | 163 | 92 | 96 | 0 | 62 | 121 | 23 | 151 | 37 | 54 |

La deuxième expérience est la même mais sans l'intégration des deux nouvelles espèces via l'active learning elles ne sont pas cachées dans le jeu de données avec des carrés blancs. Cela permet de seulement étudier l'intégration du biais sur des espèces déjà connues par

le modèle (Modèle de Base 2) sans rajouter la complexité de devoir identifier de nouvelle espèces. Le projet sélectionné en évaluation est toujours BISES. Le Tableau 5.2 indique le nombre d'annotations par espèce dans le jeu de données de base ou les nouvelles espèces ne sont pas cachées en entraînement.

### 5.2.2 ALE

Notre premier algorithme, Active Learning Extension (ALE), propose une méthode d'apprentissage actif qui intègre à la fois l'incertitude des prédictions et le nombre total de prédictions effectuées sur chaque image. Ce cadre est particulièrement pertinent pour l'ajout de nouvelles classes, comme celles introduites dans le jeu de données, où les annotations initiales sont limitées.

**Sélection des Images**

ALE utilise une double approche pour sélectionner les images à annoter. D'une part, des métriques d'incertitude, comme la confiance minimale ou l'entropie, sont utilisées pour repérer les prédictions où le modèle est le plus incertain. D'autre part, ces scores d'incertitude sont agrégés (par somme, maximum ou moyenne), puis une fonction d'extension, ici la racine carrée, est appliquée pour pondérer le nombre total de prédictions. Cela augmente les chances de découvrir de nouvelles espèces ou objets.

La formule utilisée est la suivante :

$$ALE = A(S) \times \sqrt{N}$$

où $A(S)$ représente la fonction d'agrégation des scores d'incertitude, et $N$ est le nombre total de prédictions sur une image.

**Optimisation des Annotations**

ALE permet une sélection plus efficace des échantillons pour l'annotation en intégrant ces éléments. Cela maximise le gain d'informations du modèle avec un effort d'annotations minimal. L'objectif est d'accroître la diversité des exemples annotés tout en réduisant l'incertitude globale du modèle. De plus en maximisant le nombre d'annotations apporté au modèle, on maximise le nombre d'annotations provenant de nouvelle espèces. En optimisant le processus d'annotation et en ciblant les échantillons les plus informatifs, ALE offre un moyen de réduire le coût et le temps associés à la création de nouvelles annotations, tout en améliorant la capacité du modèle à identifier et intégrer de nouvelles espèces dans le jeu de données existant.

### 5.2.3 Isolation Forest

Le second algorithme d'apprentissage actif que nous proposons repose sur la méthode de l'Isolation Forest (Oriol, Pibre et al., 2024), une technique reconnue pour la détection des anomalies. L'Isolation Forest fonctionne en identifiant et en isolant efficacement les anomalies au sein d'un jeu de données, comme l'indique son nom.

Son principe repose sur une sélection aléatoire des caractéristiques et des valeurs de partition pour diviser les données en deux sous-ensembles. Ce processus est répété de manière récursive, créant ainsi une structure arborescente binaire appelée "isolation tree". L'objectif est de séparer les anomalies du reste des données avec un nombre minimal de divisions.

Les anomalies, qui représentent une petite fraction des données et se distinguent par leur rareté, sont isolées plus rapidement que les points normaux. En effet, elles nécessitent généralement moins de partitions, ce qui se traduit par des chemins plus courts dans l'arbre. La longueur du chemin, c'est-à-dire le nombre de divisions nécessaires pour isoler un point, est donc un indicateur clé : plus le chemin est court, plus la probabilité que le point soit une anomalie est élevée.

Le score d'anomalie de chaque point est calculé en fonction de la longueur moyenne du chemin qui a été nécessaire pour l'isoler. Un seuil, prédéfini ou défini par l'utilisateur, permet ensuite de déterminer quels points sont considérés comme des anomalies. Les points dont le score dépasse ce seuil sont classés comme outliers. Dans notre étude, nous avons utilisé les mêmes paramètres de seuil que ceux proposés dans les travaux initiaux sur l'Isolation Forest, afin d'assurer une plus grande robustesse du modèle. Toutefois, cette approche a également eu pour effet de prolonger le temps nécessaire à l'annotation.

Pour appliquer l'apprentissage actif (Figure 42), nous avons d'abord extrait les cartes de caractéristiques (feature maps) du réseau après l'étape de suréchantillonnage sur l'ensemble d'entraînement. Ces cartes de caractéristiques ont ensuite été soumises à l'algorithme Isolation Forest. Par la suite, nous avons appliqué un clustering sur les données de test, en récupérant celles identifiées comme anomalies par l'algorithme. Ces données ont ensuite été exploitées pour la phase d'apprentissage actif.

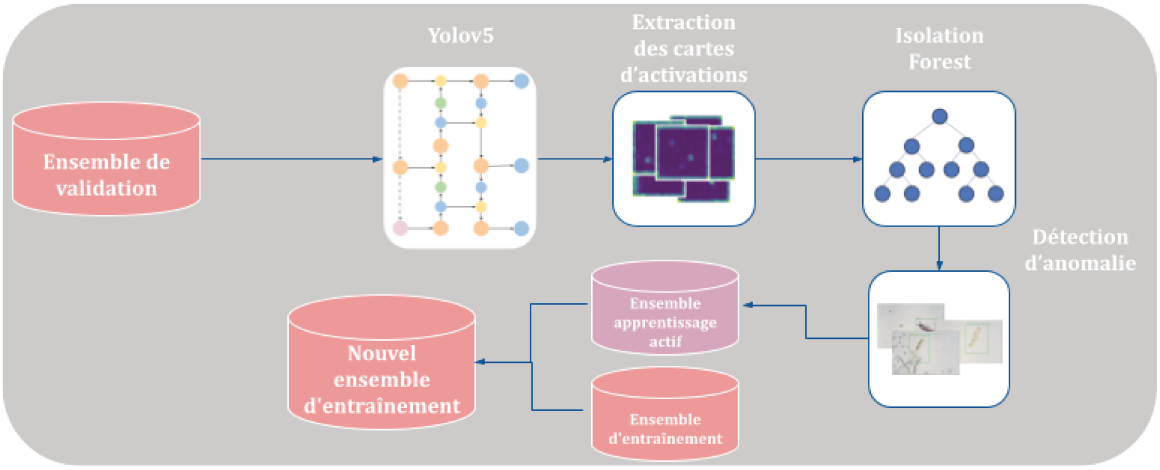


FIGURE 42 – Schéma de l'utilisation d'Isolation Forest comme technique d'apprentissage actif.

### 5.2.4 Modèle

Pour cette étude, nous avons utilisé le modèle YOLOv5x6, le même modèle utilisé dans le benchmark de détection d'objet (Tableau 3.1).

Par défaut, YOLOv5x6 utilise une couche d'activation sigmoïde pour la classification, ce qui est efficace pour générer des scores d'appartenance pour chaque classe. Cependant, l'utilisation de l'entropie comme mesure d'incertitude nécessite une distribution de probabilité qui somme à 1, ce qui n'est pas le cas avec la fonction sigmoïde. Pour surmonter cette limitation et permettre l'utilisation de l'entropie dans notre processus d'apprentissage actif, nous avons modifié le modèle YOLOv5x6 pour utiliser une couche d'activation softmax pour la classification. La fonction softmax convertit les scores de classe en une distribution de probabilité, ce qui est essentiel pour appliquer des techniques d'apprentissage actif basées sur l'incertitude, telles que celles mesurant l'entropie. Cette modification nous permet d'exploiter pleinement les capacités d'ALE (Active Learning Extension) et d'autres techniques d'apprentissage actif basées sur l'entropie, améliorant ainsi la capacité du modèle à identifier les incertitudes et à sélectionner des exemples pertinents pour l'apprentissage.

## 5.3 Protocole expérimental

### 5.3.1 Expérience

L'objectif principal de cette expérience était d'évaluer l'efficacité de nos modèles à travers l'utilisation de l'apprentissage actif, avec et sans ajout de nouvelles espèces, en se concentrant particulièrement sur l'approche Active Learning Extension (ALE) et l'algorithme Isolation Forest. Pour cela, nous avons mené deux expérimentations distinctes :

l'une impliquant l'ajout de deux nouvelles classes, et l'autre sans ajout de nouvelles classes, tout en suivant le même protocole d'entraînement.

**Création de Nouveaux Jeux de Données**

Chaque algorithme d'apprentissage actif, y compris ALE et Isolation Forest, a été utilisé pour créer de nouvelles versions du jeu de données d'origine, en sélectionnant des images spécifiques du projet BISES. Tous les algorithmes sélectionnent le même nombre d'images à chaque fois, garantissant une comparaison équitable des performances entre les différentes approches, à l'exception de l'Isolation Forest.

Les algorithmes et techniques utilisés pour faire la comparaison sont les suivants :

— **Isolation Forest** : est une méthode de détection d'anomalies qui sélectionne les échantillons en fonction de leur rareté dans les données.

— **ALE (Active Learning Extension)** : Cette approche prend en compte l'incertitude des prédictions ainsi que le nombre de prédictions par image.

— **État de l'art** : Les techniques de l'état de l'art ont également été utilisées pour générer des jeux de données distincts, afin de pouvoir comparer ALE à l'état de l'art.

— **Aléatoire** : 3 tirages aléatoires on été effectué afin de comparer l'état de l'art et ALE à l'aléatoire, seul le meilleur tirage aléatoire sera comparer aux autre méthodes.

**Finetuning des Modèles de Base**

Chaque modèle de base a été affiné (finetuné) en utilisant les nouveaux jeux de données créés par les algorithmes d'apprentissage actif. Ce processus a permis de produire plusieurs versions de chaque modèle, chacune ajustée selon la méthode de sélection spécifique. Après l'affinage, chaque nouveau modèle a été comparé au modèle de base d'origine et aux autres modèles issus des différentes techniques d'apprentissage actif.

— **Avec Ajout de Classes** : Pour cette expérimentation, le modèle de base comprenant toutes les classes a été utilisé. Cela a permis d'évaluer l'impact de l'ajout de nouvelles classes et de comparer l'efficacité d'ALE et d'Isolation Forest par rapport aux autres techniques d'apprentissage actif.

— **Sans Ajout de Classes** : Dans cette expérimentation, le modèle de base sans les deux nouvelles classes a été utilisé. Cette approche a permis d'évaluer l'effet de chaque méthode d'apprentissage actif sur l'optimisation du modèle sans l'introduction de nouvelles classes, simulant ainsi l'ajout de nouvelles espèces par l'apprentissage actif.

Les expérimentations ont été répétées trois fois, avec des sélections de 20, 50 et 100 images. Cette approche a permis de comprendre comment l'apprentissage actif, avec ou sans ajout de nouvelles classes, influence les performances des modèles à travers plusieurs

tailles de jeux de données.

Cependant, contrairement aux autres méthodes comme ALE, où l'on peut contrôler précisément le nombre d'échantillons sélectionnés, Isolation Forest sélectionne automatiquement les points considérés comme anomalies, sans possibilité de fixer le nombre exact d'échantillons à choisir.

Dans notre cas, Isolation Forest a sélectionné 45 échantillons pour le jeu de données avec ajout de classes et 55 pour le jeu de données sans ajout de classes. Pour assurer une comparaison équitable avec les autres méthodes d'apprentissage actif, nous avons décidé de ne comparer Isolation Forest seulement avec la version 50 échantillons des autres méthodes. Cela garantit que les résultats obtenus reflètent une comparaison cohérente et juste entre les différentes approches.

En combinant ALE et Isolation Forest, nous avons cherché à comprendre comment ces deux méthodes pouvaient améliorer les performances des modèles dans des contextes avec et sans ajout de nouvelles classes, tout en tenant compte des différences dans la sélection des échantillons.

### 5.3.2 Évaluation

Étant donné que les modèles ont été évalués sur des ensembles de validation différents, il n'était pas faisable de les comparer directement en utilisant leurs propres résultats de validation. Pour comparer tous les modèles de manière précise, il aurait été nécessaire d'utiliser uniquement les images communes à tous les ensembles de validation, en s'assurant qu'aucun modèle ne soit évalué sur les images sur lesquelles il a été entraîné. Toutefois, cette approche s'est avérée impraticable en raison du grand nombre de modèles et du nombre limité d'images communes disponibles. Pour surmonter ce défi, nous avons opté pour une méthode d'évaluation par paires de modèles. Voici les étapes de notre approche :

1. **Création de Nouveaux jeu de Données :**

   Pour chaque paire de modèles, nous avons créé un nouvel ensemble de données de validation contenant uniquement les images d'évaluation communes aux deux modèles de la paire. Cela garantit que chaque modèle est évalué uniquement sur les images qui n'ont pas été utilisées dans son entraînement.

2. **Évaluation par paires :**

   En utilisant ces jeux de données par paires, nous avons pu effectuer une comparaison directe entre chaque modèle. Cette méthode a permis de déterminer lequel des deux modèles performe mieux sur leur ensemble d'évaluation partagé

3. **Métrique d'évaluation :**

   La métrique utilisée pour ces comparaisons était la moyenne de précision (mAP). Le mAP est une mesure standard pour évaluer les performances des modèles de

détection d'objets, car elle tient compte à la fois de la précision et du rappel sur les différentes classes.

4. **Analyse des résultats :**
   — Comparaison Quantitative : En utilisant le mAP, nous avons obtenu une évaluation quantitative de la performance de chaque modèle par rapport à son homologue dans chaque paire.
   — identification des Modèles Supérieurs : Cette approche nous a permis d'identifier les modèles qui offraient des performances supérieures, mettant en évidence l'efficacité des techniques d'apprentissage actif utilisées.

   .

Cette méthode d'évaluation par paires a permis de contourner les limitations des ensembles de validation variés, assurant une comparaison juste et rigoureuse entre les modèles. En fin de compte, cette approche a offert une compréhension plus approfondie des performances des modèles dans des scénarios de détection d'objets complexes et variés.

### 5.3.3 Protocoles d'entraînement

Chaque modèle a été entraîné pendant 800 epochs avec les paramètres suivants : un taux d'apprentissage initial de 0.01, une décroissance du poids de 0.005, et l'optimiseur Adam avec un $\beta_1$ de 0.937. Les même techniques d'augmentation de données que le modèle du benchmark (Tableau 3.1) ont été appliquées pour améliorer la capacité de généralisation des modèles. Ces transformations ont permis d'augmenter artificiellement la taille du jeu de données, ce qui a contribué à améliorer la précision des modèles sur le jeu de test en les rendant plus adaptables à différents scénarios d'entrée.

### 5.3.4 Résultats

TABLE 5.3 – Comparaison des méthodes avec l'apprentissage actif de 20 images sur les jeux de données avec nouvelles classes.

| | ALE Entr Max | ALE Entr Somme | ALE LC Max | ALE LC Somme |
|---|---|---|---|---|
| Modèle d'origine | **12.80%** | **14.80%** | **16.10%** | **11.50%** |
| Rand | **4.42%±0.48** | **4.44%±2.05** | **6.78%±1.27** | **2.96%±0.38** |
| Entr max | **0.43%±0.26** | **0.73%±2.06** | **5.00%±1.68** | -0.73%±0.53 |
| Entr Somme | -1.77%±0.30 | -1.97%±2.17 | **0.57%±1.40** | -3.53%±0.50 |
| LC max | **0.17%±0.31** | **0.10%±2.30** | **2.87%±1.56** | -1.03%±0.81 |
| LC Somme | -0.43%±0.30 | -0.60%±2.12 | **2.00%±1.42** | -2.27%±0.50 |

TABLE 5.4 – Comparaison des méthodes avec l'apprentissage actif de 50 images sur les jeux de données avec nouvelles classes.

| | ALE Entr Max | ALE Entr Somme | ALE LC Max | ALE LC Somme | Isolation Forest |
|---|---|---|---|---|---|
| Modèle d'origine | **22.10%** | **24.20%** | **25.10%** | **25.20%** | **21.10%** |
| Rand | **5.16%±5.97** | **11.23%±4.66** | **7.44%±6.06** | **10.35%±5.05** | **4.73%±4.99** |
| Entr max | -0.65%±2.34 | **4.72%±1.42** | **2.08%±1.07** | **4.02%±1.07** | **1.05%±1.45** |
| Entr Somme | -3.45%±2.21 | **1.75%±1.58** | -1.20%±1.07 | **1.10%±1.19** | -4.07%±2.43 |
| LC max | -3.57%±2.22 | **2.10%±1.47** | -0.72%±1.03 | **1.45%±1.11** | -3.42%±1.23 |
| LC Somme | -2.85%±2.22 | **2.73%±1.51** | -0.40%±1.08 | **1.90%±1.21** | -2.47%±1.59 |

TABLE 5.5 – Comparaison des méthodes avec l'apprentissage actif de 100 images sur les jeux de données avec nouvelles classes.

| | ALE Entr Max | ALE Entr Somme | ALE LC Max | ALE LC Somme |
|---|---|---|---|---|
| Modèle d'origine | **32.60%** | **35.90%** | **33.20%** | **31.30%** |
| Rand | **5.22%±1.14** | **8.80%±0.54** | **6.32%±0.66** | **6.24%±1.42** |
| Entr max | **2.33%±0.66** | **4.83%±0.12** | **3.77%±1.07** | **2.90%±2.19** |
| Entr Somme | -2.60%±0.75 | **0.43%±0.13** | -1.43%±1.18 | -2.20%±1.70 |
| LC max | -1.87%±0.84 | **1.90%±0.15** | **0.20%±1.30** | **0.93%±1.20** |
| LC Somme | -1.60%±0.93 | **1.77%±0.17** | 0.00%±1.42 | **0.50%±0.70** |

Les résultats de l'expérimentation 1 présentés dans les Tableaux 5.3, 5.4 et 5.5 illustrent les comparaisons suivantes : la colonne de gauche énumère les techniques évaluées, incluant le modèle de base entraîné sur le jeu de données original, le meilleur modèles des tirages aléatoires, et divers modèles état de l'art combinant plusieurs métriques et techniques d'agrégation. Chaque modèle est désigné par une métrique d'évaluation suivie d'une technique d'agrégation. Par exemple, un modèle pourrait être désigné comme "Entr Somme" ou "ALE LC Max", indiquant respectivement l'utilisation de la métrique entropy avec l'agrégation "Somme" et l'approche ALE avec comme métrique LC (Least Confidence) une agrégation basée sur le maximum et l'extension racine carré.

Les colonnes suivantes affichent les résultats moyens ainsi que les variances obtenues après plusieurs répétitions pour chaque version de modèle d'apprentissage actif, afin de garantir une évaluation plus robuste. Les pourcentages présentés indiquent l'amélioration moyenne de la précision (mAP) par rapport aux autres techniques, tandis que les valeurs entre parenthèses montrent la variance des résultats. Cette méthodologie permet de capturer la stabilité et la robustesse de chaque approche.

Lorsque 20 images sont sélectionnées (Tableau 5.3), "ALE LC Max" montre les meilleures performances avec un score de 16.10%, ce qui le rend plus performant que les autres méthodes ALE comparé au modèle d'origine. "ALE Entr Somme" (14.80%) et "ALE Entr Max" (12.80%) montrent également des résultats solides. "ALE LC Somme", avec 11.50%, est la version la plus en retrait des méthodes ALE testées. "ALE LC Max" est le seul modèle qui performe mieux que tous les modèles de l'état de l'art et le tirage aléatoire. Les résultats du tirage de 20 images indique que même avec une petite sélection d'images, les performances des modèles augmente grandement.

En ce qui concerne la sélection de 50 images (Tableau 5.4), "ALE LC Somme" atteint un score de 25.20%. "ALE LC Max" (25.10%) et "ALE Entr Somme" (24.20%) suivent de près, suggérant que l'augmentation du nombre d'images a un effet significatif sur l'amélioration des résultats. Par contre, le modèle Isolation Forest, malgré une amélioration par rapport au modèle de base (21.10%), reste inférieur aux modèles ALE. Deux modèles font mieux que l'état de l'art et le tirage aléatoire, "ALE LC Somme" et "ALE Entr Somme".

Avec la sélection de 100 images (Tableau 5.5), "ALE Entr Somme" obtient les meilleures performances avec un score de 35.90%, surpassant "ALE LC Max" (33.20%) et "ALE LC Somme" (31.30%). Il est aussi une fois de plus, plus performant que tous les autre modèles de l'état de l'art et du tirage aléatoire.

Dans l'ensemble, les résultats montrent que l'ajout d'images provenant du projet BISES permet une amélioration significative des performances. L'inclusion de ces nouvelles images permet de mieux généraliser ce projet non vu pendant l'entraînement du modèle de base, ce qui explique pourquoi les modèles entraînés avec apprentissage actif surpassent systématiquement et fortement le modèle d'origine, lequel ne bénéficie pas de cette diversité supplémentaire.

Cependant, les résultats montrent que la performance de chaque version d'ALE varie selon le nombre d'images ajoutées. Par exemple, "ALE Entr Somme", bien qu'étant l'une des meilleurs verion d'ALE avec 50 images (Tableau 5.4), est la pire pour 20 (Tableau 5.3) et 100 images (Tableau 5.5). Il est aussi intéressant de noter que "ALE Entr Max", bien qu'affichant une amélioration progressive, se positionne systématiquement en retrait par rapport aux autres techniques ALE, quelle que soit la taille de l'échantillon. La technique d'agrégation maximum semble moins efficace en augmentant le nombre de tirage.

L'analyse montre également que la méthode ALE se distingue particulièrement dans le tableau des 50 tirages (Tableau 5.4), car elle est la seule où les deux versions utilisant l'agrégation par somme ("ALE Entr Somme" et "ALE LC Somme") surpassent toutes les autres techniques, y compris les méthodes d'état de l'art. Cela peut s'expliquer par le fait que le nombre d'images dans le projet BISES qui contiennent plus de trois annotations est de 62 (Tableau 5.6). Par conséquent, l'agrégation par "Somme" maximise son efficacité avec un tirage de 50 images (Tableau 5.4), permettant de capturer la diversité sans entraîner de redondance. En revanche, lorsque l'on passe à un tirage de 100 images, cette distinction s'estompe, car un plus grand nombre d'images devient commun entre les méthodes ALE et les techniques d'état de l'art, réduisant l'avantage de l'agrégation "Somme". Ces observations montrent qu'il est crucial de bien ajuster le nombre d'images lors de l'apprentissage actif, en particulier dans des contextes impliquant des projets non vus précédemment en entraînement, afin de tirer le meilleur parti des différentes approches et optimiser les performances du modèle.

On peut voire aussi qu'en comparant l'AP du modèle de base (Figure 43) avec notre meilleures version d'ALE 44, "ALE Entr Somme", sur leur jeu de données commun (lors de

la sélection de 50 images), nous observons que toutes les AP de "ALE Entr Somme" sont grandement améliorées par rapport au modèle de base. "ALE Entr Somme", entraîné avec seulement 3 annotations pour PSE ALB et 23 pour SPH PUM, comme indiqué dans le Tableau 5.8, montre déjà des résultats significatifs dans la détection des nouvelles espèces.

TABLE 5.6 – Nombre d'annotations par images dans le projet BISES.

| Nombre d'annotations par images | 1 | 1-3 | >=3 |
|---|---|---|---|
| Nombre d'images | 414 | 69 | 62 |

TABLE 5.7 – Nombre d'annotations par taxon ajoutées dans l'ensemble d'apprentissage de "ALE Entr Somme" et "Entr Somme" (la version tirage de 50 images) à partir de l'ensemble de validation.

| | AUTRE | CER | HEM THE | HYP MAN | ISO MIN | LEP | MET AFF | PAR NOT | PSE ALB | SPH PUM |
|---|---|---|---|---|---|---|---|---|---|---|
| ALE Entr Somme | 25 | 16 | 29 | 0 | 11 | 24 | 3 | 29 | 4 | 24 |
| Entr Somme | 25 | 13 | 25 | 0 | 11 | 16 | 3 | 27 | 3 | 19 |

Le Tableau 5.4 est le seul cas où une version de l'état de l'art, "LC Somme", surpasse une version d'ALE. Malgré cela, "ALE Entr Somme" et "ALE LC Max" montrent systématiquement des avantages de performance significatifs ou des résultats similaires comparés aux autres techniques. En comparant l'AP du modèle de base (Figure 43) avec notre meilleures version d'ALE 44, "ALE Entr Somme", sur leur jeu de données commun (lors de la sélection de 50 images), nous observons que toutes les AP de "ALE Entr Somme" sont grandement améliorées par rapport au modèle de base. "ALE Entr Somme", entraîné avec seulement 3 annotations pour PSE ALB et 23 pour SPH PUM, comme indiqué dans le Tableau 5.8, montre déjà des résultats significatifs dans la détection des nouvelles espèces.

TABLE 5.8 – Nombre d'annotations par taxon ajoutées dans l'ensemble d'apprentissage de "ALE Entr Somme" et "Entr Somme" (la version tirage de 50 images) à partir de l'ensemble de validation.

| | AUTRE | CER | HEM THE | HYP MAN | ISO MIN | LEP | MET AFF | PAR NOT | PSE ALB | SPH PUM |
|---|---|---|---|---|---|---|---|---|---|---|
| ALE Entr Somme | 25 | 16 | 29 | 0 | 11 | 24 | 3 | 29 | 4 | 24 |
| Entr Somme | 25 | 13 | 25 | 0 | 11 | 16 | 3 | 27 | 3 | 19 |

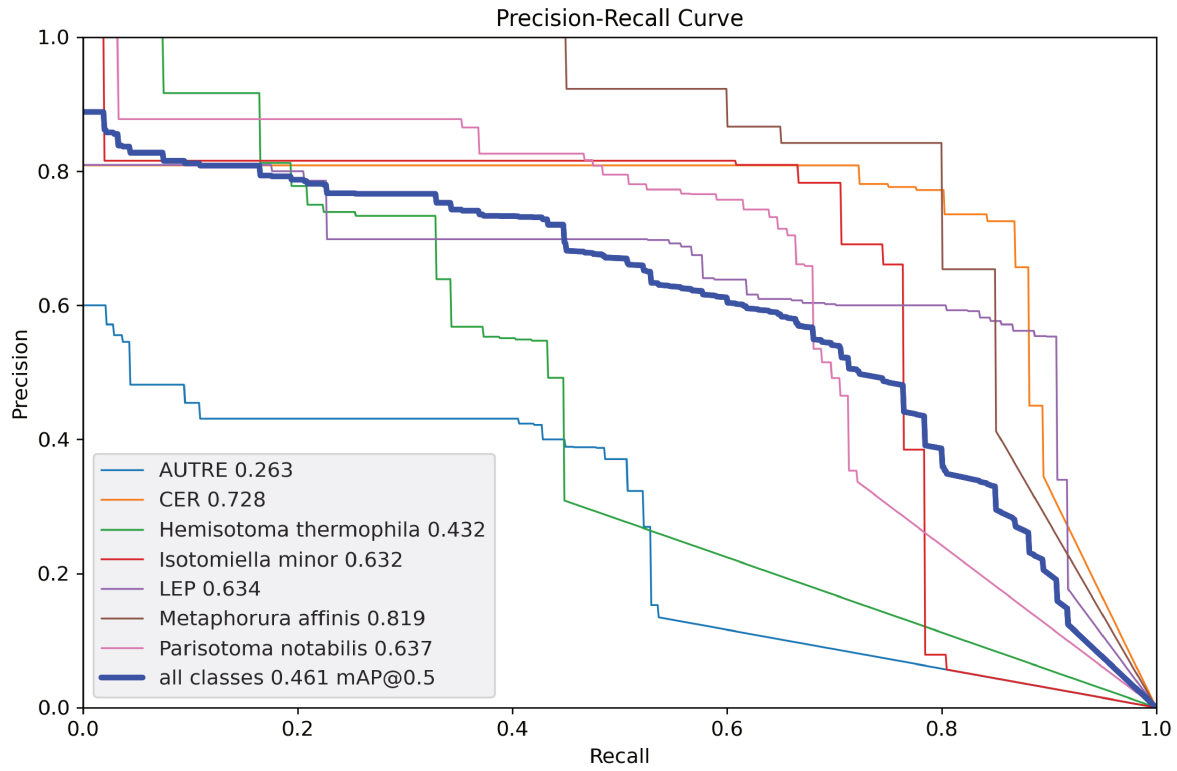


FIGURE 43 – L'AP et le mAP du modèle de base quand il est évalué sur son jeu de donnée commun avec "ALE Entr Somme" (la version tirage de 50 images) on peut y voir les résultats des 6 espèces qu'il est capable d'identifier ainsi que la catégorie "AUTRE".

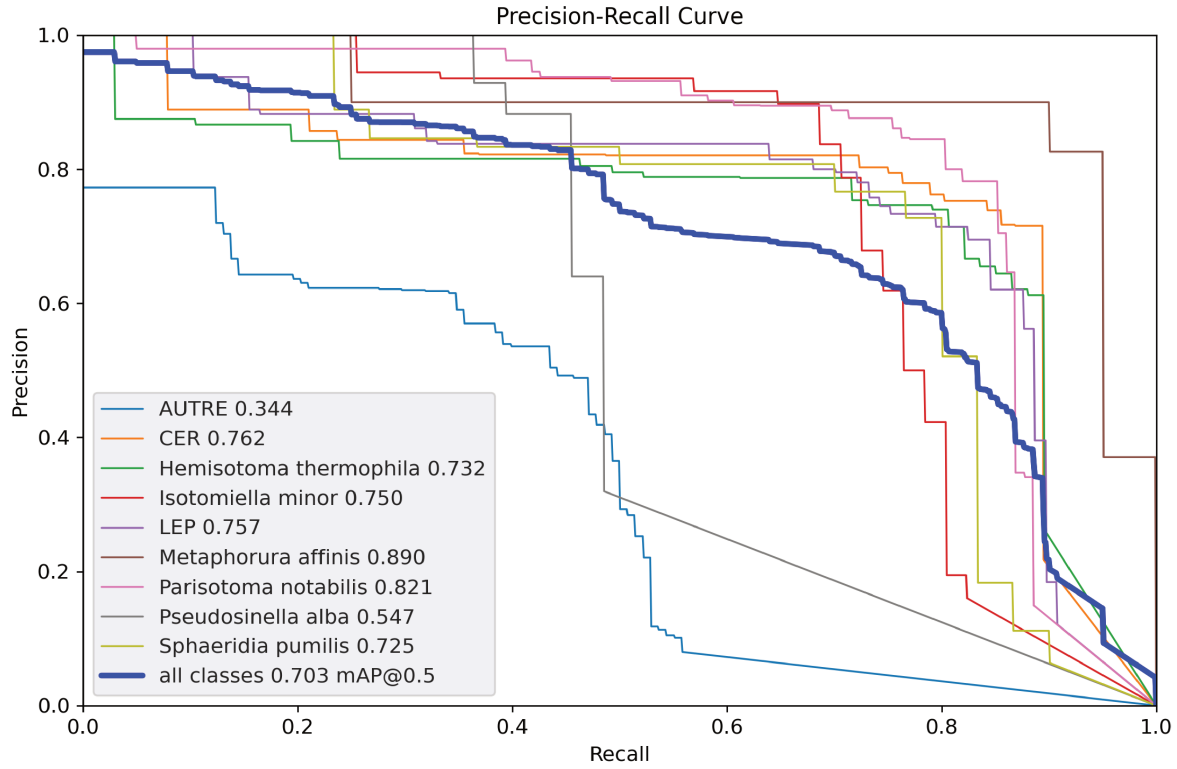


FIGURE 44 – L'AP et le mAP du modèle "ALE Entr Somme" (la version tirage de 50 images) quand il est évalué sur son jeu de donnée commun avec le modèle de base, on peut y voire une forte amélioration des performances, ainsi que l'ajout des deux nouvelles espèces.

TABLE 5.9 – Comparaison des méthodes avec l'apprentissage actif de 20 images sur les jeux de données sans nouvelles classes.

| | ALE Entr Max | ALE Entr Somme | ALE LC Max | ALE LC Somme |
|---|---|---|---|---|
| Modèle d'origine | **9.39%** | **9.10%** | **7.90%** | **9.59%** |
| Rand | **6.19%** | **4.50%** | **3.40%** | **5.59%** |
| Entr max | **2.19%** | **1.40%** | **0.50%** | **1.90%** |
| Entr Somme | **0.90%** | -0.50% | -1.19% | -0.10% |
| LC max | **2.69%** | **2.50%** | **1.80%** | **2.60%** |
| LC Somme | **2.49%** | **1.40%** | **1.00%** | **2.00%** |

TABLE 5.10 – Comparaison des méthodes avec l'apprentissage actif de 50 images sur les ensembles de données sans nouvelles classes.

| | ALE Entr Max | ALE Entr Somme | ALE LC Max | ALE LC Somme | Isolation Forest |
|---|---|---|---|---|---|
| Modèle d'origine | **13.60%** | **15.30%** | **14.00%** | **16.50%** | **9.49%** |
| Rand | **4.60%** | **7.10%** | **6.70%** | **8.10%** | -0.10% |
| Entr max | **1.50%** | **4.10%** | **2.30%** | **4.50%** | -1.49% |
| Entr Somme | **1.90%** | **4.90%** | **2.30%** | **4.10%** | -0.80% |
| LC max | **3.50%** | **6.10%** | **7.00%** | **7.10%** | **1.39%** |
| LC Somme | -1.80% | -1.30% | -0.60% | **1.60%** | -4.00% |

TABLE 5.11 – Comparaison des méthodes avec l'apprentissage actif de 100 images sur les jeux de données sans nouvelles classes.

| | ALE Entr Max | ALE Entr Somme | ALE LC Max | ALE LC Somme |
|---|---|---|---|---|
| Modèle d'origine | **17.20%** | **19.99%** | **17.60%** | **19.90%** |
| Rand | **5.80%** | **9.29%** | **5.00%** | **7.70%** |
| Entr max | -0.60% | **1.19%** | -1.20% | **1.00%** |
| Entr Somme | -0.80% | **1.70%** | -0.50% | **0.90%** |
| LC max | **1.30%** | **4.20%** | 0.60% | **3.60%** |
| LC Somme | **0.10%** | **3.09%** | -0.30% | **2.10%** |

Les résultats des tableaux 5.9, 5.10 et 5.11 illustrent les résultats de l'expérimentation 2.

Lorsque 20 images sont sélectionnées (Tableau 5.9), "ALE LC Somme" affiche les meilleures performances globales avec un score de 9.59 %, dépassant légèrement "ALE Entr Max" (9.39 %) et "ALE Entr Somme" (9.10 %). Par rapport aux autres méthodes d'apprentissage actif, "ALE LC Max" est en retrait avec 7.90 %. Toute les version de ALE performe mieux que l'état de l'art et le tirage aléatoire, à l'exception de "Entr Somme" qui performe légèrement mieux que "ALE Entr Somme", "ALE LC Max" et "ALE LC Somme". Seul le modèle "ALE Entr Max" performe mieux que tous les autres.

Lorsque 50 images sont sélectionnées (Tableau 5.10), "ALE LC Somme" enregistre une nette amélioration par rapport aux modèle d'origine, avec un score de 16.50 %. "ALE Entr Somme" suit avec 15.30 %. Les méthodes "ALE LC Max" et "ALE Entr Max" montrent également des performances respectables, avec 14.00 % et 13.60 %, respectivement. En revanche, l'algorithme "Isolation Forest" n'apporte qu'une amélioration limitée (9.49 %) et a de moins bonne performance que quasiment toute les méthode de l'état de l'art et y compris le tirage aléatoire, malgré l'apport de 5 images en plus, il est même plus proche des résultats des modèles ALE sur le tirage de 20 images alors qu'il en a plus du double. "ALE LC Somme" au contraire performe mieux que tous les modèles de l'état de l'art, suivi de près par "ALE LC Max" qui performe très légèrement moins bien que "LC Somme".

Lorsque 100 images sont sélectionnées (Tableau 5.11), "ALE Entr Somme" obtient les meilleures performances, atteignant 19.99 %, suivi de près par "ALE LC Somme" avec 19.90 % d'amélioration comparé au modèle d'origine. Ces deux méthodes surpassent largement le modèle d'origine et les autres méthodes d'état de l'art, tout en montrant une nette supériorité sur le tirage aléatoire (Rand). Les méthodes "ALE LC Max" et "ALE Entr Max" continuent de performer correctement, avec respectivement 17.60 % et 17.20 % mais on des résulats plus mitigé quand ils sont comparé à l'état de l'art.

Les résultats des trois tableaux montrent une nette supériorité des méthodes d'apprentissage actif, en particulier des approches "ALE Entr Somme" et "ALE LC Somme". Ces méthodes surpassent systématiquement le modèle d'origine et les méthodes de tirage aléatoire dans toutes les configurations (20, 50 et 100 images). Les méthodes d'agréga-

tion "Somme" démontrent une efficacité supérieure, en particulier à mesure que le nombre d'images sélectionnées augmente.

Les méthodes "ALE LC Max" et "ALE Entr Max" restent compétitives, bien qu'elles soient légèrement en retrait par rapport aux approches "Somme". Ces résultats confirment l'efficacité de l'apprentissage actif, en particulier lorsque des techniques d'agrégation efficaces sont utilisées pour maximiser la sélection d'images dans un cadre sans nouvelles classes.

Les modèles Isolation Forest quant à eux sont relativement moins efficace que les méthodes de l'état de l'art ou ALE et même le tirrage aléatoire, ce qui en fait une méthode peux efficace.

TABLE 5.12 – Tableau représentant l'évolution de la comparaison des résultats entre le modèle "ALE Entr Somme" et le modèle "Entr Somme" en fonction du nombre de tirages et de leurs images non communes sur le jeu de données avec nouvelles classes.

| **Nombre de tirage** | **Nombre d'image non communes** | **Comparaison des performances** |
|---|---|---|
| 20 | 3 | -1,97% |
| 50 | 6 | 1,75% |
| 10 | 4 | 0,43% |

TABLE 5.13 – Tableau représentant l'évolution de la comparaison des résultats entre le modèle "ALE Entr Somme" et le modèle "Entr Somme" en fonction du nombre de tirages et de leurs images non communes sur le jeu de données sans nouvelles classes.

| **Nombre de tirage** | **Nombre d'image non communes** | **Comparaison des performances** |
|---|---|---|
| 20 | 2 | -0.5% |
| 50 | 6 | 4.9% |
| 10 | 5 | 1.7% |

L'analyse des deux tableaux (Table 5.12 et Table 5.13) montre que l'évolution des performances entre "ALE Entr Somme" et "Entr Somme" dépend étroitement du nombre de tirages ainsi que du nombre d'images non communes entre les deux modèles, tant sur le jeu de données avec nouvelles classes que sans nouvelles classes. Pour 20 tirages, avec seulement 3 images non partagées sur le jeu de données avec nouvelles classes et 2 images non partagées sur le jeu de données sans nouvelles classes, "Entr Somme" dépasse "ALE Entr Somme" avec des différences de -1,97 et -0,5 %, respectivement. Cela montre que lorsque le nombre d'images non communes est faible, la stratégie de sélection d'ALE n'apporte pas encore un avantage suffisant par rapport à l'état de l'art.

Cependant, pour 50 tirages, où le nombre d'images non partagées atteint 6 dans les deux configurations, "ALE Entr Somme" se démarque de manière significative, avec une amélioration de 1,75 pour le jeu avec nouvelles classes et de 4,9 % pour le jeu sans nouvelles classes. C'est à ce niveau que "ALE Entr Somme" montre tout son potentiel, car ce seuil coïncide avec le nombre d'images ayant plus de 3 annotations (environ 62 images). La stratégie d'ALE, qui favorise les images avec le plus d'annotations, lui permet ainsi

d'exploiter pleinement la diversité des informations et de mieux généraliser les apprentissages.

Pour 100 tirages, bien qu'il y ait plus d'images communes, "ALE Entr Somme" parvient encore à faire légèrement mieux que "Entr Somme" avec des écarts positifs (0,43 pour le jeu de données avec nouvelles classes et 1,7 % pour le jeu sans nouvelles classes). Cela souligne que, même avec un plus grand nombre de tirages, "ALE Entr Somme" reste supérieur, mais de façon moins marquée. En conclusion, l'avantage clé d'"ALE Entr Somme" réside dans sa capacité à sélectionner les images les plus riches en annotations, comme c'est le cas avec 50 tirages, où la différence de performance est la plus importante.

L'intégration des méthodes d'apprentissage actif, notamment comme ALE et Isolation Forest montre des résultats encourageants pour l'amélioration des performances des modèles, spécifiquement dans des contextes où les données présentent un biais de projet prononcé. Les résultats confirment l'efficacité des méthodes ALE pour gérer les variations inter-projet des Collemboles, avec et sans ajout de nouvelles classes. Grâce à l'apprentissage actif, il est possible de sélectionner les images les plus informatives pour entraîner les modèles, atténuant ainsi les impacts des biais de projet et permettant une meilleure généralisation des modèles sur des jeux de données variés. L'intégration de ces techniques dans le cadre de la détection de nouvelles espèces pourrait ouvrir des perspectives prometteuses pour l'amélioration des systèmes de classification automatique des Collemboles.

## 5.4 Conclusion

Dans cette étude, nous avons utilisé un modèle de détection d'objets, YOLOv5x6, que nous avons initialement entraîné sur un jeu de données de Collemboles, provenant de différents projets. Ce modèle a ensuite été évalué sur un projet séparé pour tester sa capacité de généralisation. L'objectif principal était de comparer ce modèle de base avec le même modèle affiné (fine-tuned) sur des images sélectionnées issues du projet utilisé pour l'évaluation. Ces images ont été choisies via des techniques d'apprentissage actif, en utilisant des tirages aléatoires, plusieurs versions des méthodes de l'état de l'art et de nos propres techniques, Active Learning Extension (ALE) et Isolation Forest. Cette recherche démontre que l'apprentissage actif, et en particulier ALE, peut améliorer considérablement les performances du modèle YOLOv5x6 sur un nouveau projet qu'il n'a jamais vu. Grâce à l'apprentissage actif, nous avons pu sélectionner et annoter de manière optimale des images provenant de ce projet, y compris l'introduction de nouvelles espèces, ce qui a permis d'affiner le modèle avec des données additionnelles pertinentes. En appliquant l'apprentissage actif, ALE a prouvé son efficacité pour maximiser la pertinence des données d'entraînement supplémentaires, ce qui a conduit à une amélioration significative des performances de détection et d'identification des Collemboles dans de nouveaux contextes. Les résultats montrent que les méthodes d'apprentissage actif augmente significativement

les performances du modèles sur un projet non vu en entraînement et que ALE rivalise avec d'autres méthodes de l'état de l'art, en tirant pleinement parti des nouvelles données disponibles. Cette étude met en évidence le potentiel de l'apprentissage actif pour non seulement améliorer les modèles de détection d'objets sur des données inconnus, mais aussi pour faciliter l'incorporation de nouvelles informations taxonomiques grâce à une sélection d'images optimale. En optimisant l'efficacité de l'annotation de nouvelles images, ALE permet une extension flexible et puissante des capacités des modèles, ce qui est particulièrement utile dans les scénarios nécessitant une adaptation rapide à de nouvelles données, telles que l'intégration de nouvelles espèces dans l'identification du modèle.

# Conclusion et perspectives

## 5.5 Bilan

L'objectif principal de cette thèse était de développer des modèles d'apprentissage automatique pour l'identification automatisée des Collemboles à partir d'images de lames de microscope. Les Collemboles sont des micro-arthropodes du sol, peuvent être utilisés comme bio-indicateurs de la santé des sols. Leur présence, leur diversité, et leur sensibilité aux changements environnementaux en font des sujets d'étude privilégiés pour évaluer les effets des pratiques agricoles, des polluants, des variations climatiques sur les écosystèmes souterrains. Cependant, le travail d'identification des Collemboles est long et demande un niveau d'expertise élevé. Ces facteurs limitent la capacité des experts à effectuer des suivis écologiques à grande échelle. Automatiser l'identification des Collemboles avec des modèles de deep learning pourrait surmonter ces obstacles en rendant l'analyse plus rapide, plus objective et plus reproductible. Néanmoins, cette automatisation présente des défis importants liés au fait que les experts doivent analyser de minuscules détails morphologiques pour distinguer les différentes espèces, ce qui rend encore plus complexe en raison de la variabilité morphologique au sein d'une même espèce et de la similarité entre espèces.

Cette thèse s'est donc concentrée sur le développement de modèles robustes capables de faire face à cette complexité tout en maximisant la précision de leurs prédictions. Une grande attention a été portée sur l'étude des biais potentiels, qui pourraient altérer les performances des modèles et limiter leur capacité de généralisation. Pour répondre à cette problématique, plusieurs étapes de développement et d'évaluation ont été réalisées.

La création d'un jeu de données spécialisé pour l'identification des Collemboles a permis de combler un manque important dans les ressources existantes, car il n'existait pas de référentiel adapté aux spécificités de cette tâche. Ce jeu de données a intégré non seulement les similarités inter-espèces et variances intra-espèce, mais aussi des images présentant des variabilités liées à chaque projet. **Cette contribution a permis de poser les bases d'un benchmark cohérent pour l'évaluation des modèles de deep learning, tout en fournissant une ressource indispensable pour les futures recherches dans ce domaine**. Ce jeu de données a servi de support expérimental, offrant une structure permettant d'explorer comment les variations intra- et inter-projets influencent la performance des modèles.

L'évaluation approfondie des modèles de détection d'objets actuels, y compris des architectures classiques comme Faster R-CNN et des approches plus récentes comme les

versions de YOLOv5, a permis de démontrer les forces de chaque méthode sur les images de Collemboles, ainsi que leurs faiblesses face aux variations entre les projets. L'analyse, basée sur un protocole de validation croisée, a mis en évidence que certaines architectures sont plus sensibles aux différences de contexte, tandis que d'autres, comme YOLOv5x6, se sont montrées plus robustes. **Ces résultats ont jeté les bases d'une meilleure compréhension de la capacité de ces modèles à généraliser sur des jeux de données hétérogènes, identifiant ainsi les approches les plus prometteuses pour l'identification automatisée des Collemboles**.

L'analyse du biais de projet a permis de mettre en lumière le rôle prépondérant des variations contextuelles entre les différents ensembles de données. En utilisant des techniques comme le Grad-CAM et le changement de fond pour visualiser les zones d'attention du modèle, cette étude a démontré que les performances des modèles étaient souvent influencées par des artefacts contextuels, tels que le fond des images ou des variations d'apparence liées au projet. Cette compréhension approfondie du biais de projet a contribué à clarifier les sources potentielles de sur-apprentissage et de mauvaise généralisation, soulignant ainsi la nécessité de stratégies qui intègrent ces biais dans le processus d'entraînement. En identifiant les espèces et les contextes les plus sensibles à ces variations, **cette contribution a permis de mieux orienter les efforts pour améliorer la robustesse des modèles face aux biais liés aux spécificités des projets**.

Face à la présence confirmée du biais de projet, l'intégration de ce biais par l'apprentissage actif a grandement amélioré la robustesse du modèle. Plutôt que de chercher à réduire ces biais, l'approche a consisté à les intégrer de manière contrôlée par le biais de deux nouvelles méthodes d'apprentissage actif : l'Active Learning Extension (ALE) (Oriol, Pasquet & Cortet, 2024b) et l'Isolation Forest (Oriol, Pibre et al., 2024). ALE s'est avéré particulièrement performant, car il combine des mesures d'incertitude avec le nombre de prédictions par image, ciblant les échantillons complexes et informatifs pour optimiser l'apprentissage du modèle. Ce cadre a permis d'incorporer efficacement des variations entre les projets, rendant le modèle moins sensible aux biais de projet tout en maximisant son pouvoir de généralisation. En plus d'améliorer la précision sur des projets jamais vus pendant l'entraînement, ALE a également démontré sa capacité à introduire efficacement de nouvelles espèces dans le pool d'espèces détectées par le modèle, faisant de cette approche une extension dynamique et évolutive du système d'identification. L'introduction contrôlée de nouvelles espèces par l'apprentissage actif a transformé le modèle initial en un outil flexible, capable d'évoluer et de s'adapter continuellement aux nouvelles données. **En somme, cette contribution a ouvert la voie à l'intégration continue de nouveaux taxons tout en atténuant l'impact des biais de projet, faisant du modèle un outil plus robuste et complet pour l'identification automatique des Collemboles sur des jeux de données complexes et variés**.

## 5.6 Perspectives

Bien que les performances des modèles d'apprentissage automatique aient montré des résultats prometteurs dans l'identification des Collemboles, un aspect majeur reste peu exploré : l'explicabilité des décisions prises par ces algorithmes. Comprendre pourquoi le modèle fait certaines prédictions est essentiel, notamment dans des contextes écologiques ou biologiques où chaque décision doit être justifiable. Actuellement, les méthodes d'explicabilité, comme Grad-CAM, offrent un aperçu de la manière dont le modèle interprète les images, mais elles restent limitées dans le cadre de la détection d'objets. En effet, ces techniques ne fournissent qu'une visualisation partielle des zones d'intérêt, sans préciser les attributs spécifiques qui déterminent la classification.

Après avoir obtenu des résultats prometteurs avec Grad-CAM, il serait pertinent d'explorer des méthodes plus avancées pour affiner encore davantage l'interprétation des prédictions du modèle. L'utilisation de techniques telles que Layer-wise Relevance Propagation (LRP) (Binder et al., 2016) ou des mécanismes d'attention issus des transformers (Vaswani, 2017) pourrait non seulement améliorer la précision des visualisations, mais aussi permettre de mieux identifier les traits distinctifs sur lesquels le modèle s'appuie pour différencier les espèces. En poursuivant dans cette direction, l'incorporation d'explications textuelles basées sur ces visualisations ou de représentations intuitives des décisions du modèle faciliterait leur interprétation pour des experts non spécialisés en apprentissage profond, rendant ainsi ces outils plus accessibles et favorisant leur adoption dans des contextes réels.

En parallèle, un autre défi majeur réside dans l'intégration de nouvelles espèces provenant de projets non vus, un contexte où l'apprentissage actif est souvent utilisé pour réduire le nombre d'annotations nécessaires. Cependant, lorsque ces nouvelles espèces sont introduites, le modèle doit apprendre à les identifier avec un nombre très limité d'exemples. Dans ce contexte, les techniques basées sur la perte contrastive (contrastive loss) et l'apprentissage contrastif sont particulièrement pertinentes. Plutôt que de traiter chaque image de manière indépendante, ces méthodes permettent de maximiser l'utilisation des informations disponibles en comparant systématiquement les nouvelles images avec des images déjà connues, créant ainsi une multitude de combinaisons d'apprentissage.

L'apprentissage contrastif repose sur le principe de la comparaison d'images par paires ou triplets (Hoffer & Ailon, 2015 ; Koch et al., 2015). Plutôt que de traiter chaque image de façon indépendante, cette approche permet d'associer chaque exemple disponible à plusieurs autres images du jeu de données, formant ainsi un grand nombre de combinaisons à partir d'un nombre limité d'images. Par exemple, chaque nouvelle image d'une espèce peut être comparée à plusieurs images d'espèces déjà connues (pour maximiser la distance) ou à d'autres images de la même espèce (pour minimiser la distance), générant ainsi une multitude de relations à partir d'un faible nombre d'échantillons. Cette

stratégie étend efficacement l'utilisation de chaque annotation et multiplie les exemples que le modèle peut exploiter pour affiner sa compréhension des nouvelles classes.

En utilisant la perte contrastive, le modèle n'apprend pas seulement à classifier les images de façon traditionnelle, mais à positionner chaque nouvelle espèce dans un espace de représentation distinct, tout en prenant en compte sa relation avec les espèces déjà apprises. Cela permet de créer un signal d'apprentissage plus riche, car le modèle apprend simultanément à identifier ce qui rapproche les individus de la même classe et ce qui les distingue des autres classes. Ainsi, même si le nombre d'images annotées reste faible, l'apprentissage contrastif maximise le potentiel d'information apportée par chaque image, augmentant indirectement la taille de l'ensemble d'entraînement.

Ces deux approches complémentaires visent à améliorer l'efficacité et l'interprétabilité des modèles d'identification des Collemboles. D'un côté, l'exploration de méthodes explicatives plus sophistiquées permettrait d'obtenir des visualisations plus précises et des interprétations plus accessibles, essentielles pour justifier les prédictions dans des contextes écologiques. De l'autre, l'intégration de la perte contrastive dans le cadre de l'apprentissage actif maximise l'utilisation des données limitées, facilitant l'ajout de nouvelles espèces au modèle tout en améliorant sa robustesse sur des projets jamais vus. Ensemble, ces perspectives ouvrent la voie à des modèles à la fois plus transparents et plus adaptatifs, rendant leur utilisation plus pertinente pour des applications pratiques en écologie.

# Bibliographie


Anthony, M. A., Bender, S. F., & van der Heijden, M. G. (2023). Enumerating soil biodiversity. *Proceedings of the National Academy of Sciences*, *120*(33), e2304663120.

Arablouei, R., Wang, L., Currie, L., Yates, J., Alvarenga, F. A., & Bishop-Hurley, G. J. (2023). Animal behavior classification via deep learning on embedded systems. *Computers and Electronics in Agriculture*, *207*, 107707.

Bardgett, R. D., & Van Der Putten, W. H. (2014). Belowground biodiversity and ecosystem functioning. *Nature*, *515*(7528), 505-511.

Barrios, E. (2007). Soil biota, ecosystem services and land productivity. *Ecological economics*, *64*(2), 269-285.

Bellinger, P. F., Christiansen, H. E., & Janssens, F. (2009). Checklist of the Collembola of the World.

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., & Samek, W. (2016). Layer-wise relevance propagation for neural networks with local renormalization layers. *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, 63-71.

Bozinovski, S., & Fulgosi, A. (1976). The influence of pattern similarity and transfer learning upon training of a base perceptron b2. *Proceedings of symposium informatica*, *3*, 121-126.

Bradford, M., Jones, T. H., Bardgett, R. D., Black, H., Boag, B., Bonkowski, M., Cook, R., Eggers, T., Gange, A., Grayston, S., et al. (2002). Impacts of soil faunal community composition on model grassland ecosystems. *Science*, *298*(5593), 615-618.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

Brussaard, L., De Ruiter, P. C., & Brown, G. G. (2007). Soil biodiversity for agricultural sustainability. *Agriculture, ecosystems & environment*, *121*(3), 233-244.

Bünemann, E. K., Bongiorno, G., Bai, Z., Creamer, R. E., De Deyn, G., De Goede, R., Fleskens, L., Geissen, V., Kuyper, T. W., Mäder, P., et al. (2018). Soil quality–A critical review. *Soil biology and biochemistry*, *120*, 105-125.

Cardoso, E. J. B. N., Vasconcellos, R. L. F., Bini, D., Miyauchi, M. Y. H., Santos, C. A. d., Alves, P. R. L., Paula, A. M. d., Nakatani, A. S., Pereira, J. d. M., & Nogueira, M. A. (2013). Soil health: looking for suitable indicators. What should be considered to assess the effects of use and management on soil health? *Scientia Agricola*, *70*, 274-289.

Cassagne, N., Gers, C., & Gauquelin, T. (2003). Relationships between Collembola, soil chemistry and humus types in forest stands (France). *Biology and Fertility of Soils*, *37*, 355-361.

Cébron, A., Cortet, J., Criquet, S., Biaz, A., Calvert, V., Caupert, C., Pernin, C., & Leyval, C. (2011). Biological functioning of PAH-polluted and thermal desorption-treated soils assessed by fauna and microbial bioindicators. *Research in microbiology*, *162*(9), 896-907.

Chatterjee, S., & Hadi, A. S. (2015). *Regression analysis by example*. John Wiley & Sons.

Cluzeau, D., Guernion, M., Chaussod, R., Martin-Laurent, F., Villenave, C., Cortet, J., Ruiz-Camacho, N., Pernin, C., Mateille, T., Philippot, L., et al. (2012). Integration of biodiversity in soil quality monitoring: Baselines for microbial and soil fauna parameters for different land-use types. *European Journal of Soil Biology*, *49*, 63-72.

Cluzeau, D., Pérès, G., Guernion, M., Chaussod, R., Cortet, J., Fargette, M., Martin-Laurent, F., Mateille, T., Pernin, C., Ponge, J.-F., et al. (2009). Intégration de la biodiversité des sols dans les réseaux de surveillance de la qualité des sols: exemple du programme pilote à l'échelle régionale, le RMQS BioDiv. *Etude et gestion des sols*, *16*(3-4), 187-201.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273-297.

Cortet, J. (2012). Programme Bioindicateurs–Phase II. *Analyse de l'Indicateur Basé Sur Les Peuplements de Mésofaune du sol. Angers: ADEME*.

Cortet, J., Gomot-De Vauflery, A., Poinsot-Balaguer, N., Gomot, L., Texier, C., & Cluzeau, D. (1999). The use of invertebrate soil fauna in monitoring pollutant effects. *European Journal of Soil Biology*, *35*(3), 115-134.

Cortet, J., Griffiths, B. S., Bohanec, M., Demsar, D., Andersen, M. N., Caul, S., Birch, A. N., Pernin, C., Tabone, E., De Vaufleury, A., et al. (2007). Evaluation of effects of transgenic Bt maize on microarthropods in a European multi-site experiment. *Pedobiologia*, *51*(3), 207-218.

Crisci, C., Ghattas, B., & Perera, G. (2012). A review of supervised machine learning algorithms and their applications to ecological data. *Ecological Modelling*, *240*, 113-122.

Crowther, T. W., Van den Hoogen, J., Wan, J., Mayes, M. A., Keiser, A., Mo, L., Averill, C., & Maynard, D. S. (2019). The global soil community and its influence on biogeochemistry. *Science*, *365*(6455), eaav0550.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, *88*(11), 2783-2792.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303-314.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, *1*, 886-893.

Deharveng, L. (2004). Recent advances in Collembola systematics. *Pedobiologia*, *48*(5-6), 415-433.

Delgado-Baquerizo, M., Powell, J. R., Hamonts, K., Reith, F., Mele, P., Brown, M. V., Dennis, P. G., Ferrari, B. C., Fitzgerald, A., Young, A., et al. (2017). Circular linkages between soil biodiversity, fertility and plant productivity are limited to topsoil at the continental scale. *New Phytologist*, *215*(3), 1186-1196.

Doran, J. W., & Parkin, T. B. (1994). Defining and assessing soil quality. *Defining soil quality for a sustainable environment*, *35*, 1-21.

Doran, J. W., & Zeiss, M. R. (2000). Soil health and sustainability: managing the biotic component of soil quality. *Applied soil ecology*, *15*(1), 3-11.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Elith*, J., H. Graham*, C., P. Anderson, R., Dudík, M., Ferrier, S., Guisan, A., J. Hijmans, R., Huettmann, F., R. Leathwick, J., Lehmann, A., et al. (2006). Novel methods improve prediction of species' distributions from occurrence data. *Ecography*, *29*(2), 129-151.

Fierer, N. (2017). Embracing the unknown: disentangling the complexities of the soil microbiome. *Nature Reviews Microbiology*, *15*(10), 579-590.

Filatov, D., & Yar, G. N. A. H. (2022). Forest and water bodies segmentation through satellite images using u-net. *arXiv preprint arXiv:2207.11222*.

Fountain, M. T., Brown, V. K., Gange, A. C., Symondson, W. O., & Murray, P. J. (2007). The effects of the insecticide chlorpyrifos on spider and Collembola communities. *Pedobiologia*, *51*(2), 147-158.

Fountain, M., & Hopkin, S. (2004). A comparative study of the effects of metal contamination on Collembola in the field and in the laboratory. *Ecotoxicology*, *13*, 573-587.

Frąc, M., Hannula, S. E., Bełka, M., & Jędryczka, M. (2018). Fungal biodiversity and their role in soil health. *Frontiers in microbiology*, *9*, 707.

Fraley, C., & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The computer journal*, *41*(8), 578-588.

Frampton, G. K. (1997). The potential of Collembola as indicators of pesticide usage: evidence and methods from the UK arable ecosystem. *Pedobiologia*, *41*(1), 179-184.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, *36*(4), 193-202.

Gillet, S., & Ponge, J.-F. (2003). Changes in species assemblages and diets of Collembola along a gradient of metal pollution. *Applied soil ecology*, *22*(2), 127-138.

Girshick, R. (2015). Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, 1440-1448.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580-587.

Gobat, J.-M., Aragno, M., & Matthey, W. (2010). *Le sol vivant: bases de pédologie, biologie des sols* (T. 14). EPFL Press.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961-2969.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.

Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology press.

Heisler, C., & Kaiser, E.-A. (1995). Influence of agricultural traffic and crop management on Collembola and microbial biomass in arable soil. *Biology and fertility of soils*, *19*, 159-165.

Hermitte, M.-A. (1992). La convention sur la diversité biologique. *Annuaire français de droit international*, *38*(1), 844-870.

Hoffer, E., & Ailon, N. (2015). Deep metric learning using triplet network. *Similarity-based pattern recognition: third international workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, 84-92.

Holt, E., & Miller, S. (2011). Bioindicators: Using organisms to measure. *Nature*, *3*, 8-13.

Hopkin, S. P. (1997). *Biology of the springtails:(Insecta: Collembola)*. OUP Oxford.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700-4708.

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution* (p. 492-518). Springer.

Huot, H., Cortet, J., Watteau, F., Milano, V., Nahmani, J., Sirguey, C., Schwartz, C., & Morel, J.-L. (2018). Diversity and activity of soil fauna in an industrial settling pond managed by natural attenuation. *Applied Soil Ecology*, *132*, 34-44.

Jacoby, R., Peukert, M., Succurro, A., Koprivova, A., & Kopriva, S. (2017). The role of soil microorganisms in plant mineral nutrition—current knowledge and future directions. *Frontiers in plant science*, *8*, 1617.

Joimel, S., Chassain, J., Artru, M., & Faburé, J. (2022). Collembola are Among the Most Pesticide-Sensitive Soil Fauna Groups: A Meta-Analysis. *Environmental Toxicology and Chemistry*, *41*(10), 2333-2341.

Joimel, S., Schwartz, C., Bonfanti, J., Hedde, M., Krogh, P. H., Pérès, G., Pernin, C., Rakoto, A., Salmon, S., Santorufo, L., et al. (2021). Functional and taxonomic diversity of Collembola as complementary tools to assess land use effects on soils biodiversity. *Frontiers in Ecology and Evolution*, *9*, 630919.

Joimel, S., Schwartz, C., Hedde, M., Kiyota, S., Krogh, P. H., Nahmani, J., Peres, G., Vergnes, A., & Cortet, J. (2017). Urban and industrial land uses have a higher soil biological quality than expected from physicochemical quality. *Science of the Total Environment*, *584*, 614-621.

Kampichler, C., Džeroski, S., & Wieland, R. (2000). Application of machine learning techniques to the analysis of soil ecological data bases: relationships between habitat features and Collembolan community characteristics. *Soil Biology and Biochemistry*, *32*(2), 197-209.

Keesstra, S. D., Geissen, V., Mosse, K., Piiranen, S., Scudiero, E., Leistra, M., & van Schaik, L. (2012). Soil as a filter for groundwater quality. *Current Opinion in Environmental Sustainability*, *4*(5), 507-516.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kislov, D. E., Korznikov, K. A., Altman, J., Vozmishcheva, A. S., & Krestov, P. V. (2021). Extending deep learning approaches for forest disturbance segmentation on very high-resolution satellite images. *Remote Sensing in Ecology and Conservation*, *7*(3), 355-368.

Koch, G., Zemel, R., Salakhutdinov, R., et al. (2015). Siamese neural networks for one-shot image recognition. *ICML deep learning workshop*, *2*(1), 1-30.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kumar, S., & Jayagopal, P. (2021). Delineation of field boundary from multispectral satellite images through U-Net segmentation and template matching. *Ecological Informatics*, *64*, 101370.

Lal, R. (2004). Soil carbon sequestration to mitigate climate change. *Geoderma*, *123*(1-2), 1-22.

Lal, R. (2015). Restoring soil quality to mitigate soil degradation. *Sustainability*, *7*(5), 5875-5895.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Lehmann, J., Bossio, D. A., Kögel-Knabner, I., & Rillig, M. C. (2020). The concept and future prospects of soil health. *Nature Reviews Earth & Environment*, *1*(10), 544-553.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 21-37.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the seventh IEEE international conference on computer vision*, *2*, 1150-1157.

Lubbock, J. (1873). *Monograph of the Collembola and Thysanura* (T. 28). Ray Society.

Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). *Computers & Geosciences*, *19*(3), 303-342.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*, 115-133.

Milano, V., Maisto, G., Baldantoni, D., Bellino, A., Bernard, C., Croce, A., Dubs, F., Strumia, S., & Cortet, J. (2018). The effect of urban park landscapes on soil Collembola diversity: A Mediterranean case study. *Landscape and Urban Planning*, *180*, 135-147.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Norouzzadeh, M. S., Morris, D., Beery, S., Joshi, N., Jojic, N., & Clune, J. (2021). A deep active learning system for species identification and counting in camera trap images. *Methods in ecology and evolution*, *12*(1), 150-161.

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, *115*(25), E5716-E5725.

Noss, R. F. (1990). Indicators for monitoring biodiversity: a hierarchical approach. *Conservation biology*, *4*(4), 355-364.

Oriol, T., Pasquet, J., & Cortet, J. (2024a). Automatic identification of Collembola with deep learning techniques. *Ecological Informatics*, *81*, 102606.

Oriol, T., Pasquet, J., & Cortet, J. (2024b). ALE : active learning extension for object detection. *CORESA (COmpression et REprésentation des Signaux Audiovisuels)*.

Oriol, T., Pibre, L., Pasquet, J., & Mari, Z. (2024). Boxes to Segmentation to Boxes again for a better detection. *2023 Twelfth International Conference on Image Processing Theory, Tools and Applications (IPTA)*.

Ouvrard, S., Barnier, C., Bauda, P., Beguiristain, T., Biache, C., Bonnard, M., Caupert, C., Cebron, A., Cortet, J., Cotelle, S., et al. (2011). In situ assessment of phytotechnologies for multicontaminated soil management. *International Journal of Phytoremediation*, *13*(sup1), 245-263.

Pérès, G., Vandenbulcke, F., Guernion, M., Hedde, M., Beguiristain, T., Douay, F., Houot, S., Piron, D., Richard, A., Bispo, A., et al. (2011). Earthworm indicators as tools for soil monitoring, characterization and risk assessment. An example from the national Bioindicator programme (France). *Pedobiologia*, *54*, S77-S87.

Pernin, C., me Cortet, J. o. ^., Joffre, R., Le Petit, J., & Torre, F. (2006). Sewage sludge effects on mesofauna and cork oak (Quercus suber L.) leaves decomposition in a Mediterranean forest firebreak. *Journal of environmental quality*, *35*(6), 2283-2292.

Ponge, J.-F., Gillet, S., Dubs, F., Fedoroff, E., Haese, L., Sousa, J. P., & Lavelle, P. (2003). Collembolan communities as bioindicators of land use intensification. *Soil biology and biochemistry*, *35*(6), 813-826.

Ponge, J.-F., Pérès, G., Guernion, M., Ruiz-Camacho, N., Cortet, J., Pernin, C., Villenave, C., Chaussod, R., Martin-Laurent, F., Bispo, A., et al. (2013). The impact of agricultural practices on soil biota: a regional study. *Soil Biology and Biochemistry*, *67*, 271-284.

Recknagel, F. (2001). Applications of machine learning to ecological modelling. *Ecological modelling*, *146*(1-3), 303-310.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263-7271.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rusek, J. (1998). Biodiversity of Collembola and their functional role in the ecosystem. *Biodiversity & Conservation*, *7*, 1207-1219.

Russell, D., & Alberti, G. (1998). Effects of long-term, geogenic heavy metal contamination on soil organic matter and microarthropod communities, in particular Collembola. *Applied Soil Ecology*, *9*(1-3), 483-488.

Santorufo, L., Cortet, J., Arena, C., Goudon, R., Rakoto, A., Morel, J.-L., & Maisto, G. (2014). An assessment of the influence of the urban environment on collembolan communities in soils using taxonomy-and trait-based approaches. *Applied Soil Ecology*, *78*, 48-56.

Schloter, M., Dilly, O., & Munch, J. (2003). Indicators for evaluating soil quality. *Agriculture, Ecosystems & Environment*, *98*(1-3), 255-262.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE international conference on computer vision*, 618-626.

Seybold, C., Herrick, J. E., & Brejda, J. (1999). Soil resilience: a fundamental component of soil quality. *Soil science*, *164*(4), 224-234.

Siddiky, M. R. K., Kohler, J., Cosme, M., & Rillig, M. C. (2012). Soil biota effects on soil structure: interactions between arbuscular mycorrhizal fungal mycelium and collembola. *Soil Biology and Biochemistry*, *50*, 33-39.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, *550*(7676), 354-359.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Swift, M. J., Izac, A.-M., & Van Noordwijk, M. (2004). Biodiversity and ecosystem services in agricultural landscapes—are we asking the right questions? *Agriculture, ecosystems & environment*, *104*(1), 113-134.

Sys, S., Weissbach, S., Jakob, L., Gerber, S., & Schneider, C. (2022). CollembolAI, a macrophotography and computer vision workflow to digitize and characterize samples of soil invertebrate communities preserved in fluid. *Methods in Ecology and Evolution*, *13*(12), 2729-2742.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1-9.

Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J., VerCauteren, K. C., Snow, N. P., Halseth, J. M., Di Salvo, P. A., Lewis, J. S., White, M. D., et al. (2019). Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods in Ecology and Evolution*, *10*(4), 585-590.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*, 154-171.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, *1*, I-I.

Wäldchen, J., & Mäder, P. (2018). Machine learning for image based species identification. *Methods in Ecology and Evolution*, *9*(11), 2216-2225.

Weinstein, B. G. (2018). A computer vision for animal ecology. *Journal of Animal Ecology*, *87*(3), 533-545.