UNIVERSITÉ DU QUÉBEC À MONTRÉAL

DÉTECTION D'ESPÈCES D'ARBRES À L'AIDE D'OUTILS D'INTELLIGENCE ARTIFICIELLE ET RECOMMANDATION DE PLANTATION D'ARBRES URBAINS DANS LES ESPACES PRIVÉS

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

SAMY ASSOUANE

MAI 2026

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.12-2023). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je tiens tout d'abord à exprimer ma plus profonde gratitude et reconnaissance à ma directrice de recherche, la professeure Marie-Jean Meurs, dont l'accompagnement, la disponibilité et la patience ont été déterminants à l'aboutissement de ce travail de recherche. Sa vision éclairée dans le domaine de l'intelligente artificielle, sa rigueur scientifique et son exigence académique ont été pour moi une réelle source d'inspiration tout au long de mon parcours et m'ont permis d'acquérir de solides compétences. Je la remercie sincèrement pour son soutien constant à chacune des étapes qui ont permis la concrétisation de ce projet d'étude. Je remercie également toute l'équipe SylvCiT et plus largement tout les membres de la Chaire ArbrenVil, en particulier Christian Messier, Maxime Nicol, Joël Lefebvre et Étienne Comtois, pour l'environnement stimulant et bienveillant dans lequel j'ai eu la chance d'évoluer. Une pensée particulière va à Annick Saint-Denis, que je remercie chaleureusement pour son aide précieuse, sa gentillesse et ses encouragements qui ont grandement contribué à l'aboutissement de ce projet. Je désire aussi remercier mes amis, notamment Raouf grâce à qui j'ai eu l'opportunité de rejoindre ce projet de recherche, Abdelhadi, Dalil, Mehdi, Nail, Rafik et Reda, pour leur présence, leur appui et la motivation qu'ils m'ont toujours apportée. Je souhaite particulièrement adresser mes remerciements les plus sincères à ma femme, pour son soutien indéfectible, sa compréhension et sa présence à mes côtés durant les moments les plus exigeants de ce parcours. À ma sœur et à ma famille, qui m'ont toujours soutenu, je désire exprimer ma profonde gratitude pour leur bienveillance et la solidité de leurs liens. Enfin, je dédie ce mémoire à ma mère et à mon père, en témoignage de ma reconnaissance pour leurs sacrifices, leurs soutiens et les fortes valeurs qu'ils m'ont transmises tout au long de ma vie, et pour lesquels l'éducation a toujours été une priorité.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

ACRONYMES

API Application Programming Interface.

AOT Ahead-Of-Time.

C2PSA Cross Stage Partial with Spatial Attention.

C3K2 Cross Stage Partial with Kernel size 2.

CBR Case-Based Reasoning.

CNN Convolutional Neural Network.

CSRF Cross-Site Request Forgery.

CSP Cross Stage Partial Networks.

CSV Comma-Separated Values.

DD Degrés Décimaux.

DHP Diamètre à Hauteur de Poitrine.

DMM Degrés Minutes Décimales.

DMS Degrés Minutes Secondes.

FN Faux Négatifs.

FP Faux Positifs.

GBIF The Global Biodiversity Information Facility.

GPS Global Positioning System.

GPU Graphics Processing Unit.

HTTPS Hypertext Transfer Protocol Secure.

JSON JavaScript Object Notation.

MLE Maximum Likelihood Estimation.

MVVM Model-View-ViewModel.

NUMPY Numerical Python.

OOD Out-of-Distribution.

ORM Object-Relational Mapping.

RA réalité augmentée.

ReLU Rectified Linear Unit.

RGPD Règlement Général sur la Protection des Données.

RVB Rouge Vert Bleu.

SDK Software Development Kit.

SDM Species Distribution Models.

SGD Stochastic Gradient Descent.

SQL Structured Query Language.

SR Système de Recommandation.

TLS Transport Layer Security.

UQAM Université du Québec à Montréal.

VN Vrais Négatifs.

VP Vrais Positifs.

XSS Cross-Site Scripting.

YOLO You Only Look Once.

NOTATION

Nombres

M une matrice.

Unités

m un mètre.

km un kilomètre.

ppm une partie par million.

GB un gigaoctet (*gigabyte*).

GLOSSAIRE

Ajustement fin Méthode d'apprentissage automatique qui consiste à réentraîner un modèle, préentraîné sur un large corpus, sur un ensemble de données spécifique à une tâche cible. Cela se fait en ajustant tout ou une partie de ses paramètres.

Application dorsale Partie d'une application informatique qui s'exécute côté serveur. Elle permet d'effectuer les calculs et de gérer la logique métier, l'accès aux bases de données et la communication avec les interfaces de l'application frontale.

Apprentissage automatique Sous-domaine de l'intelligence artificielle qui développe des algorithmes capables d'apprendre des modèles à partir de données afin de réaliser automatiquement des tâches (prédire, classer, détecter des motifs) sans être concrètement programmés pour chaque tâche particulière.

Cadriciel Ensemble structuré de bibliothèques et d'outils qui fournit une architecture de base pour le développement d'applications.

Carte de caractéristiques Représentation intermédiaire qui est produite par les couches de convolution d'un réseau de neurones convolutif. Elle permet d'encoder la présence d'un motif particulier (bord, texture, forme, etc.) détecté sur l'image d'entrée.

Classe En classification par apprentissage automatique supervisé, une classe désigne une valeur possible de la variable cible auquel le modèle assigne chaque exemple. Ces classes sont aussi appelées cibles, labels ou catégories, et représente concrètement ce que le modèle doit prédire.

Diamètre à hauteur de poitrine Mesure standardisée du diamètre du tronc d'un arbre prise à 1,3 mètre du sol. Le DHP est un indicateur de la taille et de la maturité d'un arbre.

Diversité fonctionnelle Mesure de la variété des traits fonctionnels présents au sein d'un ensemble de végétaux. Une forêt urbaine avec une forte diversité fonctionnelle est plus résiliente aux perturbations et offre un éventail plus large de services écosystémiques.

Forêt urbaine Réseau ou système incluant toutes les surfaces boisées, les groupes d'arbres et les arbres individuels (publics et privés) situés en zone urbaine et périurbaine..

Groupe fonctionnel Regroupement d'espèces végétales partageant des traits fonctionnels communs qui leur confèrent des réponses adaptatives similaires faces à certaines conditions environnementales.

Hétérophyllie Phénomène botanique par lequel un même individu végétal produit des feuilles de formes, de tailles, de couleurs ou de structures différentes selon l'âge, la position ou les conditions environnementales.

Hyperparamètres Paramètres de configuration d'un algorithme d'apprentissage automatique qui sont prédéfinis avant l'entraînement du modèle et qui ne sont pas appris à partir des données.

Îlot de chaleur urbain Phénomène météorologique où les zones urbaines présentent des températures significativement plus élevées que les zones rurales environnantes. Il peut être causé par l'absorption de chaleur par des surfaces imperméables, la réduction de la végétation et des rejets thermiques.

Injections SQL Type de faille de sécurité informatique qui permet à un attaquant d'insérer des instructions en langage SQL dans un champ de saisie d'une application. Le but est de manipuler la base de données sans y être autorisé.

Intelligence artificielle Domaine de l'informatique qui conçoit des systèmes ou des algorithmes capables de réaliser des tâches simulant l'intelligence humaine (raisonnement logique, apprentissage, perception, compréhension du langage naturel), en s'appuyant sur des méthodes symboliques ou d'apprentissage automatique.

Modèle (en apprentissage automatique) Représentation mathématique apprise à partir de données d'entraînement, capable de produire des prédictions ou des décisions pour de nouvelles entrées.

Nombre à virgule flottante Représentation numérique informatique permettant d'exprimer des valeurs réelles avec une partie fractionnaire. Les poids d'un réseau de neurones sont généralement stockés sous forme de nombres à virgule flottante (float32 ou float16) ce qui influence la stabilité de l'entraînement et les performances du modèle.

Phyllosphère Surface externe des feuilles d'un végétal et l'ensemble de la communauté microbienne (bactéries, champignons, levures) qui colonise cette surface.

Réalité augmentée (RA) Technologie qui superpose des éléments numériques (images, objets 3D, textes) à la perception du monde réel en temps réel, à travers un écran d'appareil mobile ou des lunettes connectées.

Réseau de neurones Système de calcul (modèle mathématique) composé d'unités simples appelées neurones, organisés en couches et reliées par des connexions pondérées, qui transforment des entrées en sorties en ajustant ces poids lors d'un processus d'apprentissage supervisé ou non supervisé.

Ce sont des modèles non linéaires capables d'extraire des motifs et des relations complexes dans les données.

Service écosystémique Bénéfice direct ou indirect que les êtres humains tirent des écosystèmes naturels et des arbres, comme les services d'approvisionnement (nourriture, eau), de régulation (séquestration de carbone, régulation thermique, gestion des eaux pluviales) et culturels (valeur esthétique).

Stress hydrique État physiologique d'un végétal résultant d'un déficit en eau disponible dans le sol ou d'une transpiration excessive.

Surapprentissage Phénomène où un modèle d'apprentissage automatique apprend trop précisément les données d'entraînement, incluant leur bruit et leurs spécificités propres, au détriment de sa capacité à généraliser à de nouvelles données. Le modèle mémorise les exemples d'entraînement au lieu de capturer les régularités générales nécessaires pour bien prédire sur de nouvelles données.

Taux d'apprentissage Hyperparamètre fondamental de l'optimisation des réseaux de neurones qui détermine l'amplitude des ajustements apportés aux poids du modèle à chaque itération d'entraînement.

Transfert d'apprentissage Méthode d'apprentissage automatique qui consiste à réutiliser les connaissances acquises par un modèle sur une tâche source, pour améliorer l'apprentissage sur une tâche cible différente mais connexe.

Vision par ordinateur Domaine de l'intelligence artificielle qui développe des méthodes permettant aux ordinateurs d'interpréter et de comprendre des informations visuelles (images, vidéos) de manière automatique.

RÉSUMÉ

L'urbanisation croissante fragilise le couvert arboré, en particulier dans les espaces privés comme les cours intérieures et les jardins, alors qu'ils représentent un potentiel important mais souvent sous-valorisé et peu inventorié. En effet, les arbres privés contribuent grandement aux services écosystémiques, notamment à la biodiversité urbaine, à la lutte aux îlots de chaleur, à la gestion de l'eau et au bien-être des habitants. Le travail présenté s'intéresse à ces arbres privés, difficiles à inventorier et à gérer, et propose une approche pour aider les personnes citoyennes à planter de manière plus cohérente et durable.

Le projet repose sur une application mobile, qui combine vision par ordinateur et système de recommandation utilisant l'intelligence artificielle et la participation citoyenne. La première composante permet l'identification automatique d'espèces d'arbres à partir de photos de feuilles, en utilisant un modèle d'apprentissage profond entraîné par transfert d'apprentissage sur des milliers d'images nettoyées, pré-traitées et augmentées, des principales espèces présentes dans la région de Montréal. Une stratégie d'ajustement fin et d'optimisation du modèle est mise en place afin d'assurer une certaine robustesse et une efficacité suffisante. La seconde composante de l'application propose des recommandations d'espèces d'arbres à planter dans les espaces privés. Ces espèces sont adaptées aux contraintes de la zone de plantation et aux préférences des personnes utilisatrices, tout en cherchant à maximiser la diversité fonctionnelle et donc la résilience du couvert forestier. Notre outil est intégré comme module complémentaire de SylvCiT, une plateforme d'aide à la décision en code ouvert qui recommande des plantations visant à accroître la résilience de la forêt urbaine face aux changements globaux. Les recommandations de l'application proposée s'appuient sur les inventaires publics et sur la base de données collaborative des arbres privés qui sera alimentée au cours de l'utilisation de l'application.

Le projet vise à offrir un outil simple et opérationnel pour orienter les choix de plantation d'arbres en terrain privé, tout en construisant une base de données des arbres privés. Notre but est d'améliorer la résilience de la forêt urbaine, en maximisant la diversité fonctionnelle tout en respectant les contraintes et les préférences des personnes utilisatrices.

INTRODUCTION

L'urbanisation rapide et la densification des métropoles transforment profondément les paysages et les conditions de vie dans les villes. Cette expansion met à l'épreuve la capacité des villes à s'adapter aux changements environnementaux et à y répondre efficacement (Seto *et al.*, 2012). Dans ce contexte, la forêt urbaine contribue à ralentir ce phénomène en produisant des bénéfices mesurables pour les populations et les écosystèmes, en contribuant par exemple à la régulation du microclimat, au soutien de la biodiversité, et à la qualité de vie (Bolund et Hunhammar, 1999). Elle permet également d'améliorer la qualité de l'air en capturant de polluants atmosphériques, et en éliminant plusieurs contaminants (Nowak *et al.*, 2006). Aussi, il est prouvé que la forêt urbaine a un impact positif sur l'atténuation des îlots de chaleur urbains (Bowler *et al.*, 2010), d'autant plus que les épisodes de chaleur extrême et la dégradation de la qualité de l'air tendent à s'intensifier dans les régions urbaines. En effet, le couvert forestier urbain fournit un ensemble de services écosystémiques qui justifie sa protection et l'optimisation de sa gestion. Les espaces verts privés (jardins, cours intérieures, etc.) représentent une part importante de la forêt urbaine et contribuent à la diversité fonctionnelle de cette dernière (Hutt-Taylor et Ziter, 2022). Les études montrent que les jardins privés sont de véritables réservoirs de biodiversité, mais qu'il faut considérer leur organisation à l'échelle du voisinage et de la ville (Goddard *et al.*, 2010).

Cependant, ces arbres en espaces privés sont souvent sous-valorisés et très peu inventoriés malgré leur contribution aux services écosystémiques. En effet, ils sont rarement intégrés dans les outils classiques de planification, qui se concentrent surtout sur les inventaires du domaine public, fréquemment mis à jour. Leur inventaire exige des efforts importants, des permissions d'accès et une logistique coûteuse, contrairement aux arbres de rues et de parcs. Pour combler ce manque d'information et améliorer la connaissance des arbres situés sur les terrains privés, les approches participatives et la science citoyenne permettent de répertorier ces arbres tout en renforçant l'engagement des communautés (Bonney *et al.*, 2009). De plus, la science citoyenne est devenue un outil de recherche crédible à fort potentiel (Dickinson *et al.*, 2010).

La question de la disponibilité des données sur les arbres urbains en espaces privés n'est pas la seule difficulté, l'un des problèmes majeurs est l'identification fiable des espèces. Pour une personne non experte, reconnaître une espèce d'arbre à partir de critères visuels peut être complexe. Les progrès de la vision par ordinateur offrent une opportunité majeure, notamment grâce aux réseaux de neurones convolutifs entraînés sur de larges bases de données (LeCun *et al.*, 2015). Ces avancées permettent d'atteindre des

performances remarquables pour la classification d'images (Krizhevsky *et al.*, 2012). L'identification automatique d'espèces d'arbres à partir d'images de leurs feuilles a déjà été explorée et des systèmes pionniers ont montré la faisabilité et l'efficacité de telles applications (Kumar *et al.*, 2012).

Pour obtenir des résultats optimaux sur un cas spécifique, comme la reconnaissance d'espèces d'arbres, il est rarement préférable d'entraîner un modèle "à partir de zéro", en particulier lorsque les classes sont nombreuses et que les images à analyser ne sont pas standardisées. Le transfert d'apprentissage est un moyen pour réutiliser des représentations apprises sur de grands ensembles de données et donc d'améliorer les performances sur des tâches cibles, surtout lorsque les données annotées sont plus limitées. La littérature montre que les couches profondes d'un réseau de neurones deviennent progressivement plus spécifiques à la tâche d'origine, d'où la mise en place de stratégies d'ajustement fin (*fine-tuning*) permettant d'utiliser efficacement les caractéristiques apprises tout en spécialisant le modèle à la nouvelle tâche, particulièrement lorsqu'il y a une certaine proximité entre les domaines (source et cible) (Yosinski *et al.*, 2014; Lee *et al.*, 2022; Li et Zhang, 2021; Mehra *et al.*, 2024).

Pour une personne utilisatrice, identifier les espèces d'arbres déjà présentes sur son espace privé ne suffit pas pour l'aider à planter plus intelligemment un nouvel arbre. La plantation en terrains privés est une décision sous contraintes qui devrait idéalement contribuer à la résilience de la forêt urbaine en maximisant la diversité fonctionnelle. L'utilisation de systèmes de recommandation adaptés constituent un moyen efficace pour y parvenir. La littérature présente plusieurs approches (collaborative, basée sur le contenu ou hybride) en détaillant leurs limites (Adomavicius et Tuzhilin, 2005). Les travaux sur les systèmes hybrides ont notamment montré l'intérêt de combiner plusieurs types d'approches pour atténuer les faiblesses de chacun pris individuellement (Burke, 2002).

Plus concrètement, la problématique principale ciblée est :

— Quelles espèces d'arbre dois-je planter dans mon espace privé pour augmenter la résilience de la forêt urbaine, tout en tenant compte des mes contraintes et préférences ?

De cette problématique en découle deux autres, formulées comme suit :

— Comment pouvons-nous obtenir un inventaire fiable des arbres urbains en espaces privés, afin de pouvoir maximiser la diversité fonctionnelle ?

— Comment puis-je déterminer les espèces d'arbre déjà présentes dans mon espace privé ?

Le but de notre travail de recherche est donc de concevoir et de développer une solution permettant d'une part de détecter automatiquement une espèce d'arbre à partir de l'image de sa feuille afin d'alimenter, grâce à une application mobile citoyenne, une base de données des arbres urbains privés ; et d'autre part de recommander aux personnes utilisatrices des espèces d'arbre à planter pertinentes pour la maximisation de la diversité fonctionnelle ainsi que le respect des contraintes du lieu de plantation et des préférences des personnes utilisatrices.

Ce mémoire présente les différentes étapes et méthodologies utilisées lors de notre travail de recherche. Le chapitre 1 traite du contexte et des notions préliminaires relatives à la foresterie urbaine, aux services écosystémiques et aux moyens de différenciation des espèces d'arbre. Le chapitre 2 dresse un état de l'art des méthodes utilisées, à savoir, la vision par ordinateur, les systèmes de recommandation et les applications mobiles citoyennes. Le chapitre 3 détaille la composante de reconnaissance d'espèces d'arbre grâce à la vision par ordinateur, tandis que le chapitre 4 détaille la composante de recommandation. Le chapitre 5 décrit l'architecture et les choix technologiques de la solution applicative développée. Enfin, dans le chapitre 6, nous présentons les résultats obtenus, les limites et les perspectives d'évolution.

CHAPITRE 1
CONTEXTE ET NOTIONS PRÉLIMINAIRES

1.1 Foresterie urbaine

La foresterie urbaine, également connue sous le nom de sylviculture urbaine, représente la gestion, la planification et la valorisation des arbres et des espaces boisés en milieu urbain. Elle prend en compte la plantation, l'entretien, la protection et la conservation de ces derniers. Elle permet d'améliorer la qualité de vie des personnes citoyennes grâce aux nombreux bienfaits des arbres urbains sur les effets du changement climatique, la santé publique et la résilience urbaine (Frantzeskaki, 2019). En effet, les forêts urbaines procurent plusieurs services écosystémiques, tels que la régulation des microclimats, la réduction des îlots de chaleur, la gestion des eaux pluviales, l'amélioration de la qualité de l'air et le soutien à la biodiversité (Turner-Skoff et Cavender, 2019).

Contrairement à la foresterie classique, qui gère de vastes peuplements continus d'arbres hors des villes et dont les objectifs sont souvent reliés à la production, la foresterie urbaine s'intéresse au couvert arboré dans la ville pour les bénéfices de la population citoyenne. Elle s'applique donc à des espaces plus restreints comme les rues, les parcs, les trottoirs, mais aussi les jardins et les cours privés.

Dans le contexte d'une métropole comme Montréal, la forêt urbaine fait face à de nombreux défis comme les îlots de chaleur dans les secteurs denses, les sols compactés et pauvres en matière organique, le salage hivernal, les cycles gel-dégel, les espaces racinaires contraints, la proximité des bâtiments, et la présence fréquente de fils électriques aériens. La santé d'un arbre en milieu urbain dépend principalement de l'adéquation entre ses traits fonctionnels (ex. vitesse de croissance, tolérance au sel, à la sécheresse, à l'ombre, profondeur racinaire) et le site de plantation. La gestion des espaces de plantation urbains doit prendre en compte les conflits avec les infrastructures alentours, les coûts et les impacts sanitaires (Roman *et al.*, 2020). La foresterie urbaine se rapproche plus d'une ingénierie du vivant, où chaque plantation est une décision sous contraintes.

Même si l'intégration de technologies innovantes, telles que la télédétection à haute résolution et les outils d'intelligence artificielle, permet d'améliorer l'inventaire, le suivi et la gestion des arbres urbains, notamment pour la reconnaissance des espèces et l'optimisation de la diversité fonctionnelle (Neyns et Canters,

2022), la participation citoyenne est désormais reconnue comme essentielle pour assurer la durabilité et l'efficacité des projets de sylviculture urbaine (Ferreira *et al.*, 2020).

En somme, la foresterie urbaine représente un atout stratégique pour le développement durable des villes, en conciliant enjeux environnementaux, sociaux et économiques, tout en s'adaptant aux spécificités locales et aux attentes des habitants (Escobedo *et al.*, 2019). Les arbres urbains doivent être considérés comme une infrastructure verte qui favorise la résilience des villes face aux changements climatiques (Esperon-Rodriguez *et al.*, 2025).

1.2 Services écosystémiques des arbres en espaces privés urbains

Les arbres situés dans les milieux privés urbains, comme les jardins résidentiels, les cours intérieures et autres terrains privés, constituent un réservoir majeur de nature et de biodiversité en ville. Les travaux de Hutt-Taylor et Ziter (2022) montrent que dans la zone de Montréal, les arbres situés sur des terrains privés représentent environ 50 % de l'ensemble des arbres urbains. L'étude souligne également que les arbres privés apportent une part unique et importante de la diversité d'espèces et des types de services dans la forêt urbaine. En effet, 30 % du total des espèces ne sont pas présentes dans le domaine public. Ignorer ces arbres en espaces privés conduit inévitablement à une vision biaisée de la composition, de la diversité, de la structure et des bénéfices de la forêt urbaine (Hutt-Taylor et Ziter, 2022). Ils fournissent de nombreux services écosystémiques, c'est-à-dire des bénéfices tirés des écosystèmes, qui touchent à la fois l'environnement, la société et l'économie urbaine, contribuant ainsi à la santé et au bien-être des populations ainsi qu'à la résilience des villes (Bolund et Hunhammar, 1999).

1.2.1 Régulation des microclimats et atténuation des îlots de chaleur

Un îlot de chaleur urbain est une zone urbaine où la température est significativement plus élevée que dans les zones rurales environnantes. La différence de température provient essentiellement de l'accumulation de chaleur causée par les structures urbaines et l'activité humaine (Filho *et al.*, 2018; Mohajerani *et al.*, 2017). L'intensité d'un îlot de chaleur est généralement mesurée par l'écart de température entre le secteur le plus chaud de la ville et l'espace naturel (non urbain) qui l'entoure. Cela permet de constituer un indicateur simple et représentatif de l'impact thermique de la présence d'une zone urbaine (Martín-Vide *et al.*, 2015; Jabbar *et al.*, 2023). Les arbres en milieux urbains, y compris dans les espaces privés, participent grandement à la régulation des microclimats, qui désignent les conditions climatiques locales pouvant différer du climat

général environnant. Leur capacité à fournir de l'ombre et à évapotranspirer (le processus par lequel de l'eau est transférée vers l'air par transpiration des feuilles et évaporation de l'eau à leur surface) permet de réduire la température ambiante limitant ainsi les effets des îlots de chaleur urbains, notamment lors des grosses vagues de chaleur estivales (Pataki *et al.*, 2021). La présence d'un couvert arboré dense dans les quartiers résidentiels est directement associée à une diminution concrète de la température de surface et à une amélioration du confort thermique pour les habitants (Nguyen *et al.*, 2021; Nevers *et al.*, 2025).

1.2.2 Amélioration de la qualité de l'air

La pollution de l'air en milieu urbain est un problème majeur de santé publique touchant la majorité des grandes villes à travers le monde. Elle résulte principalement des activités humaines telles que le trafic routier, l'industrie et le chauffage. Le phénomène est aggravé par la densité urbaine et la configuration des villes (Piracha et Chaudhary, 2022; Karagulian *et al.*, 2015). En filtrant les particules fines et en absorbant divers polluants atmosphériques comme le dioxyde d'azote (NO_2), le dioxyde de soufre (SO_2) et l'ozone (O_3), les arbres contribuent ainsi à l'amélioration de la qualité de l'air urbain (Bolund et Hunhammar, 1999; Pataki *et al.*, 2021). Ce service écosystémique est particulièrement important dans les zones résidentielles où la proximité des sources de pollution peut affecter la santé des populations vulnérables (Nguyen *et al.*, 2021).

1.2.3 Stockage du carbone

Le taux de dioxyde de carbone (CO_2) dans les villes est nettement plus élevé que dans les zones rurales, principalement causé par la forte densité de population qui engendre inéluctablement une augmentation du trafic, du chauffage et des activités industrielles. On constate que la concentration de CO_2 peut être en moyenne de 66 ppm supérieure dans les villes par rapport à la campagne (George *et al.*, 2007) et que les valeurs mesurées dans les quartiers urbains varient souvent entre 440 et 510 ppm, avec des pics près des axes routiers et en hiver (lorsque le chauffage est utilisé) (Zhu *et al.*, 2021). En stockant du carbone dans leur biomasse, les arbres urbains contribuent à réduire localement ces concentrations, participant ainsi à la lutte contre le changement climatique. Même si l'impact global des arbres urbains sur le stockage du carbone est limité par rapport aux forêts naturelles, on constate que leur contribution reste significative à l'échelle locale, particulièrement dans certains quartiers où la densité d'arbres privés est élevée (Pataki *et al.*, 2021; Bolund et Hunhammar, 1999).

1.2.4 Gestion des eaux pluviales

Les sols urbains imperméables (béton, asphalte) limitent l'infiltration de l'eau et augmentent son ruissellement lors des précipitations, ce qui rend les zones urbaines particulièrement exposées aux risques d'inondation et d'érosion. Cette accumulation rapide d'eau favorise les crues soudaines, ce qui augmente d'une part les risques de dommages aux infrastructures et aux habitations, et d'autre part transporte des sédiments et polluants vers les cours d'eau naturels (Feng *et al.*, 2021). Ces phénomènes sont aggravés par l'urbanisation rapide et le changement climatique, en intensifiant les épisodes de pluies extrêmes (Cea et Costabile, 2022). Les arbres urbains jouent un rôle crucial dans la gestion des eaux pluviales. En effet, ils interceptent les précipitations, favorisent l'infiltration de l'eau dans le sol et réduisent le ruissellement, ce qui limite les risques d'inondation et d'érosion dans les zones urbaines (Bolund et Hunhammar, 1999; Evans *et al.*, 2022). Ce service est particulièrement précieux dans les espaces privés, où la perméabilité des sols est souvent supérieure à celle des espaces publics imperméabilisés.

1.2.5 Soutien à la biodiversité

La variété des espèces végétales, animales et microbiennes présentes dans les villes représente une importante diversité biologique. Grandement menacée par l'urbanisation, cette biodiversité urbaine joue un rôle clé pour la santé humaine, le bien-être et le fonctionnement des écosystèmes urbains. Les espaces privés riches en espèces végétales, offrent des habitats variés pour cette faune urbaine (oiseaux, insectes, petits mammifères) contribuant donc au soutien à la biodiversité en ville (Evans *et al.*, 2022; Paudel et States, 2023). Également, les arbres urbains agissent comme des réservoirs et des vecteurs essentiels pour la biodiversité microbienne en propageant des cellules bactériennes dans l'air, et grâce à la diversité bactérienne de la phyllosphère, c'est-à-dire l'ensemble des micro-organismes vivant à la surface des feuilles, qui augmente de manière significative avec l'intensité urbaine (Laforest-Lapointe *et al.*, 2017). Les arbres à fruits et les arbres à fleurs jouent un rôle bien plus large qu'une simple fonction esthétique. Ils fournissent des ressources alimentaires et des habitats à diverses espèces, notamment les insectes pollinisateurs et les oiseaux, favorisant ainsi la diversité de ces derniers (Hutt-Taylor et Ziter, 2022; Francini *et al.*, 2022).

1.2.6 Bénéfices sociaux et culturels

Les arbres en espaces privés procurent des bénéfices esthétiques, favorisent le bien-être psychologique de la population et la cohésion sociale (Wan *et al.*, 2021; Nguyen *et al.*, 2021). Leur présence est clairement

associée à une réduction du stress, à une amélioration de la santé mentale et à une diminution de certaines maladies chroniques (Kondo *et al.*, 2018; Nguyen *et al.*, 2021). La règle 3 - 30 - 300 est un indicateur utilisé pour évaluer l'accessibilité aux espaces verts urbains. Elle suggère que chaque personne citoyenne devrait pouvoir voir au moins 3 arbres depuis son domicile pour favoriser le bien-être mental, bénéficier d'un couvert forestier (canopée) de 30 % dans son quartier, et résider à moins de 300 mètres d'un parc (Robitaille et Douyon, 2025).

1.3 Différenciation des espèces d'arbres

La différenciation des espèces d'arbres repose sur la comparaison de nombreux traits biologiques. Parmi ces critères, la densité du bois et l'écorce jouent un rôle important, en effet sa texture, sa couleur, son épaisseur ou la présence de motifs particuliers permettent de distinguer de nombreuses espèces. Également, des caractéristiques comme la hauteur, le diamètre, la disposition des branches, la structure du tronc et de manière plus générale l'architecture, qui sont issues de stratégies d'adaptation à la compétition pour la lumière et à la stabilité mécanique, donnent des indices importants pour l'identification d'une espèce d'arbre particulière (Maynard *et al.*, 2022). On constate aussi que les fleurs et les fruits constituent des éléments de différenciation importants, car leur forme, leur couleur, leur période d'apparition et leur structure sont très souvent spécifiques à une espèce d'arbre particulière (Li *et al.*, 2021). Dans une moindre mesure, car moins accessibles, les racines permettent également une identification précise, car elles présentent aussi des variations en termes d'architecture et de densité propres à certaines espèces ou genres d'arbres (Vleminckx *et al.*, 2021). Les espèces d'arbres peuvent également s'identifier avec les bourgeons, ce qui est pratique en hiver (Schulz, 2025).

Enfin, les feuilles sont des organes facilement accessibles et riches en informations morphologiques et physiologiques, ce qui leur permet d'être au cœur de nombreuses approches de classification et d'identification d'espèces d'arbres (Cerutti *et al.*, 2013; Kumar *et al.*, 2012; Jeon et Rhee, 2017).

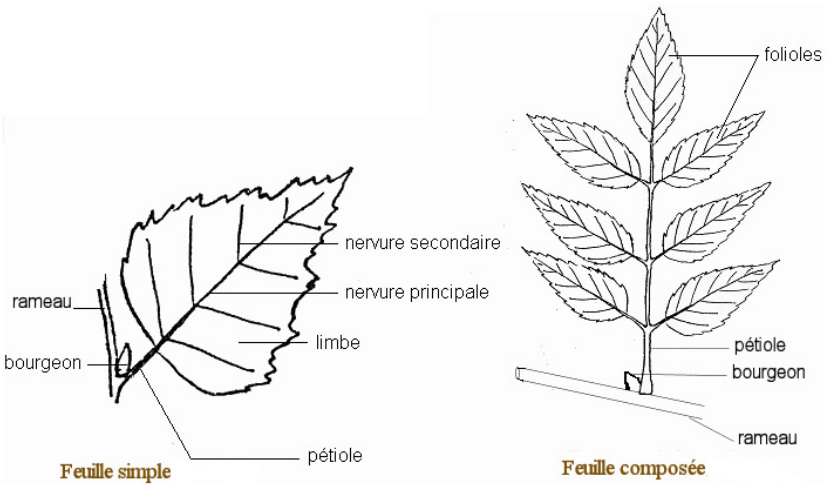


Figure 1.1 – Morphologie d'une feuille simple et composée (Zeghad, N, 2018).

La morphologie des feuilles d'arbres regroupe l'ensemble des leurs caractéristiques (Figure 1.1). Elle permet de décrire leur organisation et leur forme. Cela comprend l'architecture générale du limbe (feuille simple ou composée), sa forme (ovale, lancéolée, lobée, etc.), ses dimensions, ainsi que la nature de ses bords (entiers, dentés, crénelés, etc.) (Figures 1.2 et 1.3). Il existe aussi des critères structuraux comme le type de nervation, la disposition et l'angle des nervures secondaires, et l'aspect du pétiole (Hickey, 1973; Tsukaya, 2018). L'ensemble de ces caractères combinés, permet de différencier des espèces parfois très proches, c'est pour quoi l'analyse morphologique des feuilles est une méthode courante pour différencier les espèces d'arbres, notamment en vision par ordinateur (Kremer *et al.*, 2002; Viscosi et Cardini, 2011). D'autres caractéristiques peuvent s'évaluer davantage au toucher comme la texture et la pubescence.

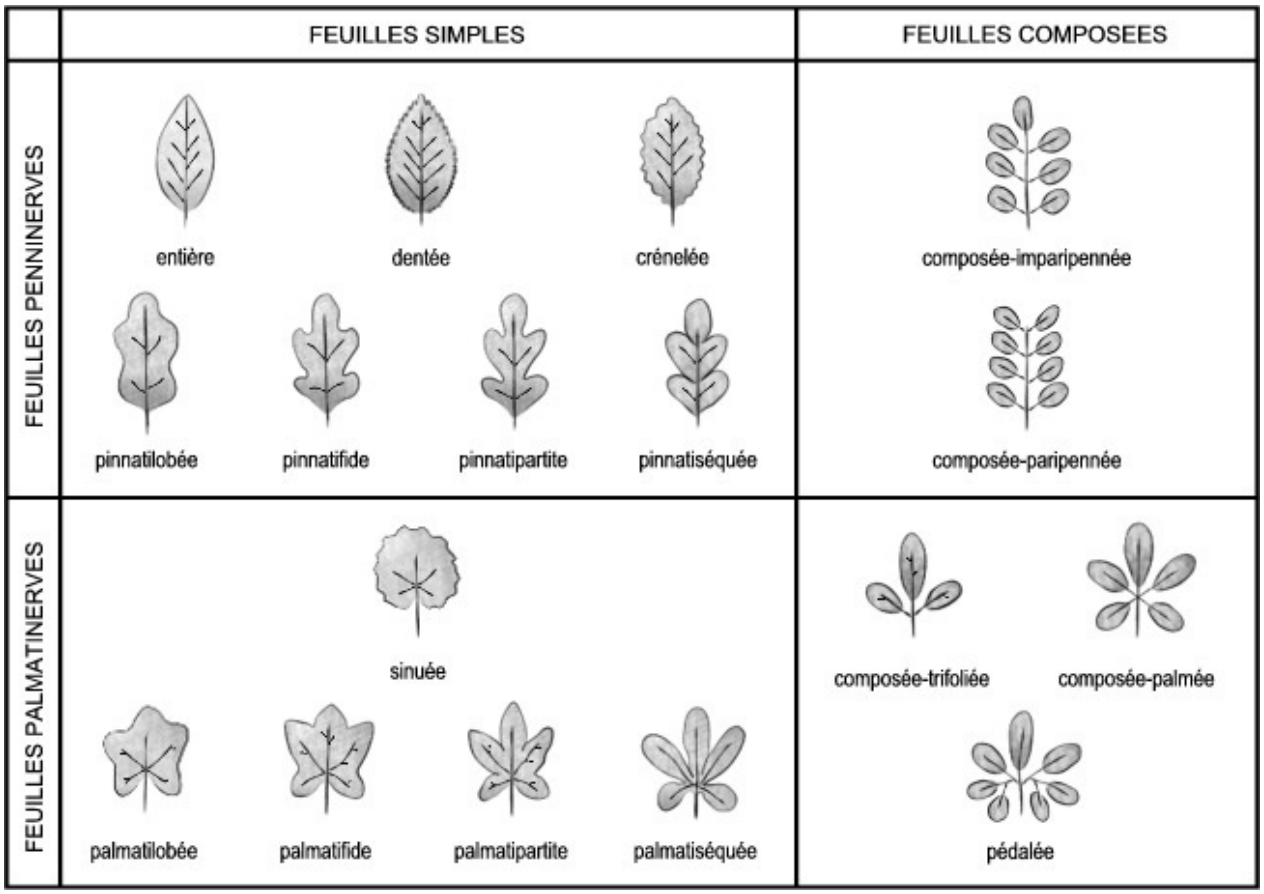


Figure 1.2 – Caractérisation des feuilles selon leur nervation et leur marge (Bouzid, 2016).

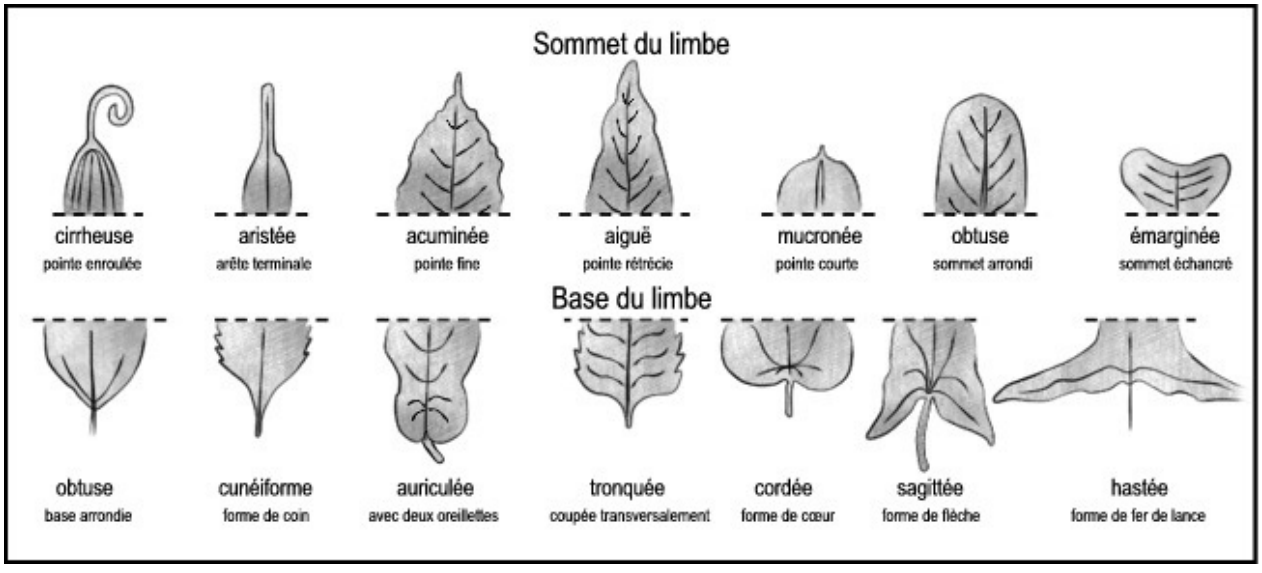


Figure 1.3 – Caractérisation des feuilles selon la forme du sommet et la base du limbe (Bouzid, 2016).

Certaines techniques de morphométrie géométrique, qui utilisent des points de repère précis, permettent de repérer précisément les variations de forme et d'identifier des différences subtiles entre espèces, même proches (Liu *et al.*, 2018; Viscosi et Cardini, 2011). La figure 1.4 montre comment certaines valeurs géométriques caractéristiques, comme les points B et T qui représentent respectivement la base et le sommet de la feuille, la largeur maximale relative de la feuille, la position du point p le long de l'axe BT où la largeur est maximale et les angles d'ouvertures α_B et α_T à la base et au sommet. Ces valeurs géométriques peuvent être utilisées comme descripteurs pour construire la représentation de la forme globale de la feuille (Cerutti *et al.*, 2013). De plus, la stabilité des caractères morphologiques foliaires à travers différentes populations et environnements a été démontrée, ce qui renforce l'utilité des feuilles et surtout leur fiabilité pour la discrimination des espèces (Kremer *et al.*, 2002).

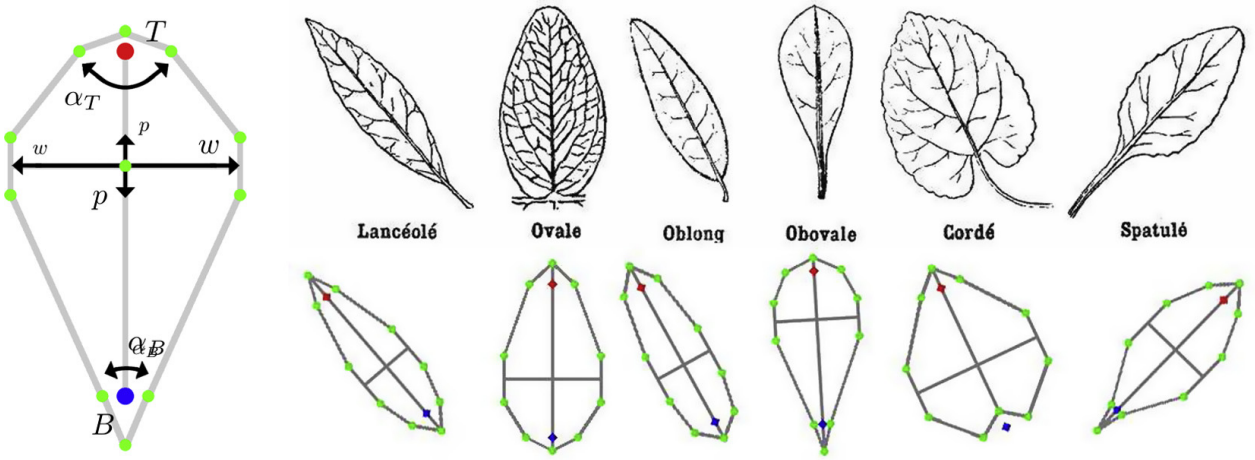


Figure 1.4 – Construction des modèles simples de feuilles avec la méthode de Cerutti *et al.* (2013).

CHAPITRE 2
ÉTAT DE L'ART

2.1 Vision par ordinateur pour l'identification d'espèces

La vision par ordinateur est un domaine de l'intelligence artificielle qui développe des méthodes permettant à un ordinateur de produire, en fonction de données visuelles d'entrée, des données exploitables en sortie. Cela passe par l'acquisition, le traitement et l'interprétation des données visuelles (images ou vidéos), afin d'extraire des informations complexes et pertinentes pour prendre des décisions automatisées. La vision par ordinateur a beaucoup évolué depuis les premiers algorithmes de traitement d'images vers des systèmes intégrant l'apprentissage automatique (en particulier l'apprentissage profond) pour devenir un sous-domaine important de l'intelligence artificielle (Voulodimos *et al.*, 2018; Manakitsa *et al.*, 2024; Bi *et al.*, 2022).

La vision par ordinateur a connu une transformation radicale avec l'introduction et l'évolution des réseaux de neurones convolutifs (*CNN*), qui sont devenus la technologie dominante pour de nombreuses tâches telles que la classification d'images, la détection d'objets et la segmentation d'image. Un CNN (figure 2.1) est un type de réseau de neurones artificiels conçu pour traiter des données structurées en grille, comme les images. Il se compose principalement de trois types de couches : convolutives, de mise en commun (*pooling*) et entièrement connectées (O'shea et Nash, 2015).

Les couches convolutives appliquent des filtres (ou noyaux), *i.e.* de petites matrices de coefficients appris par le réseau de dimension $m \times n$ (où généralement $m = n = 3$), sur l'image d'entrée pour extraire des caractéristiques locales, comme les bords, textures ou motifs. Chaque filtre glisse sur l'image et produit une carte de caractéristiques (*feature map*), et l'empilement de ces couches permet au réseau d'apprendre des représentations hiérarchiques, c'est-à-dire que les premières couches détectent des motifs élémentaires (comme les bords et les textures), alors que les couches plus profondes combinent ces motifs pour représenter des structures de plus en plus complexes. Afin de réduire le coût de calcul et de rendre ces représentations moins sensibles à de légers déplacements des motifs dans l'image, les CNN intègrent généralement des couches de mise en commun. Ces dernières ont pour rôle de réduire la taille spatiale des cartes de caractéristiques produites par les couches de convolution. Concrètement, elles divisent chaque carte en petites régions locales (généralement des blocs de 2×2) et remplacent chaque région par une

unique valeur synthétique. Les deux opérations couramment utilisées sont le *max pooling*, qui retient la valeur maximale de la région, et l'*average pooling*, qui calcule la moyenne des valeurs. Dans les deux cas, cela permet de réduire d'un facteur quatre le nombre de données à traiter. Cette réduction permet de conserver les informations essentielles et de rendre le modèle plus robuste aux petites translations. En effet, si un motif se déplace légèrement dans l'image, les activations correspondantes se déplacent elles aussi dans la carte de caractéristiques, mais restent dans la même région locale. La valeur synthétique extraite reste donc identique ou très proche, quelle que soit la position exacte du motif dans cette région. Enfin, les couches entièrement connectées interviennent en fin de réseau. Elles prennent comme entrée les caractéristiques apprises par les couches précédentes et les combinent afin de produire en sortie des prédictions, des scores de classes (puis des probabilités) pour la classification, ou une valeur continue pour la régression (O'shea et Nash, 2015).

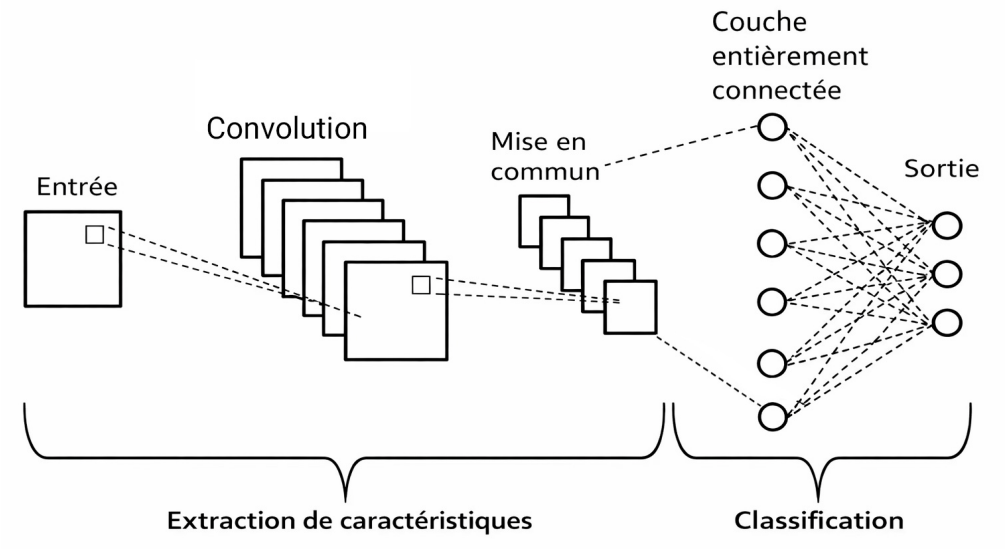


Figure 2.1 – Architecture générale d'un CNN (Fan et Truong, 2022).

L'entraînement d'un CNN consiste à ajuster itérativement les paramètres du réseau (les poids θ des filtres et des couches entièrement connectées) afin que les prédictions produites soient aussi proches que possible des valeurs attendues. Les prédictions se font par propagation avant, où l'entrée est passée successivement à travers les couches du réseau (Krizhevsky *et al.*, 2012; O'shea et Nash, 2015).

Pour les couches convolutives, l'opération fondamentale est la convolution entre l'entrée et un filtre, décrite comme suit :

$$(C * F)_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} C_{i+m,j+n} \cdot F_{m,n} \tag{2.1}$$

Où :

- C est la carte de caractéristiques d'entrée (ou l'image pour la première couche),
- F est le filtre convolutif de dimensions $M \times N$,
- $(C * F)_{i,j}$ est la valeur de la carte de caractéristiques de sortie à la position (i, j).

La sortie de chaque couche convolutive est ensuite passée à travers une fonction d'activation non linéaire, comme la fonction ReLU (*Rectified Linear Unit*) (Dubey *et al.*, 2022) :

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2.2}$$

La sortie vaut donc 0 si $x < 0$, et vaut x si $x \geq 0$ (figure 2.2). La non-linéarité permet au réseau de représenter des frontières de décision non linéaires et donc d'apprendre des représentations complexes. Sans cette non-linéarité, l'empilement des couches linéaires serait équivalent à une unique transformation linéaire (Dubey *et al.*, 2022).

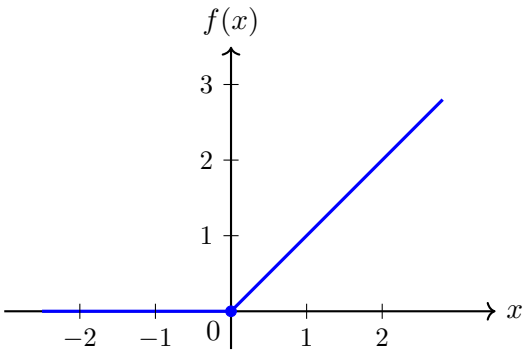


Figure 2.2 – Fonction ReLU.

Pour quantifier l'écart entre les prédictions du modèle et les valeurs réelles, on utilise une fonction de perte. Elle permet de calculer précisément la mesure que l'entraînement du modèle cherche à minimiser (Vapnik, 1991; Rumelhart *et al.*, 1986). Pour cela on calcule la fonction de perte globale (moyenne) L, définie comme suit :

$$L(\theta) = \left(\frac{1}{N}\right) \sum_{i=1}^{N} \ell\big(f(x_i;\theta), y_i\big) \tag{2.3}$$

Où :

- θ représente l'ensemble des paramètres du réseau,
- N est le nombre total d'exemples dans l'ensemble d'entraînement,
- x_i est l'entrée (l'image de l'exemple i),
- y_i est l'étiquette réelle (la classe de l'exemple i),
- $f(x_i;\theta)$ est la sortie du réseau pour l'entrée x_i avec les paramètres θ,
- ℓ est la fonction de perte individuelle pour un exemple qui mesure l'écart entre la prédiction $f(x_i;\theta)$ et la vraie étiquette y_i. En classification, on utilise généralement l'entropie croisée.

Pour savoir efficacement dans quel sens et de combien modifier les paramètres pour faire diminuer la perte, on calcule le gradient de la fonction de perte par rapport à chaque paramètre du réseau noté $\nabla_\theta L(\theta)$, c'est-à-dire un vecteur contenant toutes les dérivées partielles de la perte par rapport à chaque poids. Pour cela on utilise la rétropropagation, un algorithme introduit par Rumelhart *et al.* (1986), qui applique récursivement la règle de dérivation en chaîne (*chain rule*). L'erreur est ainsi propagée depuis la sortie vers les couches précédentes pour obtenir, pour chaque paramètre $W^{(l)}$ de chaque couche l, un terme $\frac{\partial L}{\partial W^{(l)}}$ qui indique comment ce filtre doit être modifié pour réduire la perte (Rumelhart *et al.*, 1986). Le gradient de la fonction de perte par rapport aux poids de la couche l est formulé comme suit :

$$\frac{\partial L}{\partial W^{(l)}} = \delta^{(l)} \cdot \left(a^{(l-1)}\right)^{\top} \tag{2.4}$$

Où :

- $\delta^{(l)}$ est le terme d'erreur de la couche l. Il représente la sensibilité de la perte aux variations de la pré-activation de la couche l, c'est-à-dire la sortie de la couche avant l'application de la fonction d'activation.
- $a^{(l-1)}$ représente le vecteur des activations (sorties) de la couche précédente.
- Le symbole $\top$ indique la transposition.

Enfin, une fois les gradients calculés par rétropropagation, on applique la méthode de descente de gradient pour ajuster les poids du réseau en les déplaçant petit à petit dans la direction qui fait diminuer la perte, donc dans celle opposée au gradient (Bottou, 2012). La mise à jour des paramètres se fait comme suit :

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t) \tag{2.5}$$

Où :

- θ_t sont les paramètres à l'itération t,
- η est le taux d'apprentissage (*learning rate*), un hyperparamètre contrôlant l'amplitude des mises à jour,
- $\nabla_\theta L(\theta_t)$ est le gradient de la fonction de perte par rapport aux paramètres.

Cependant, en pratique, le calcul du gradient sur l'ensemble des données d'entraînement est très coûteux en ressources. On utilise donc la descente de gradient stochastique (SGD), qui estime le gradient à partir d'un sous-ensemble aléatoire (*mini-batch*). Cette approximation introduit certes du bruit dans l'estimation du gradient, mais elle permet des mises à jour plus fréquentes, et aide souvent à échapper aux minima locaux, c'est-à-dire des solutions où la fonction de perte est plus basse juste localement, mais pas la plus basse possible (Bottou, 2012).

Les CNN sont restés relativement marginaux jusqu'au début des années 2010, principalement à cause du manque de puissance de calcul et de données annotées (Rawat et Wang, 2017). Le véritable tournant s'est produit avec *AlexNet* en 2012, qui a remporté le concours *ImageNet* (Deng *et al.*, 2009) en améliorant grandement les performances notamment grâce à une architecture plus profonde qui comporte davantage de couches que les architectures CNN antérieures, augmentant ainsi la capacité du modèle à construire des caractéristiques de plus en plus abstraites. Il introduit également l'utilisation du GPU et de techniques d'optimisation comme la régularisation par abandon aléatoire (*dropout*), qui consiste à désactiver aléatoirement pendant l'entraînement une fraction de neurones à chaque itération pour réduire le surapprentissage (*overfitting*), c'est-à-dire le phénomène où un modèle s'ajuste trop aux données d'entraînement réduisant sa capacité de généralisation sur de nouvelles données (Khan *et al.*, 2019). Ce succès a déclenché une vague d'innovations et d'adoption massive des CNN dans la communauté scientifique (Li *et al.*, 2020b; Zhao *et al.*, 2024).

Nous nous intéressons plus précisément à l'application des CNN pour la classification et l'identification d'espèces végétales par transfert d'apprentissage. Le transfert d'apprentissage consiste à réutiliser pour une

nouvelle tâche un modèle pré-entraîné sur un grand ensemble de données. Les poids des couches initiales du CNN qui apprennent des caractéristiques génériques (bords, textures, etc.) sont transférées sur une nouvelle tâche, puis on ajuste les couches finales sur le nouvel ensemble de données. Cela permet notamment de réduire le temps d'entraînement, d'améliorer la performance sur des petits ensembles de données, et d'éviter le surapprentissage (Li *et al.*, 2020b). Dans notre travail, on utilise le modèle pré-entraîné YOLO qui est reconnu pour être l'un des plus robustes pour la classification d'images (Situ *et al.*, 2023).

YOLO (*You Only Look Once*) est un algorithme de détection d'objets qui traduit la détection en un problème de régression unique, prédisant directement pour chaque objet présent dans l'image, les coordonnées de sa boîte englobante (c'est-à-dire le rectangle qui délimite et encadre l'objet), ainsi que la distribution de probabilités indiquant la classe à laquelle cet objet appartient. Il utilise un CNN pour extraire les caractéristiques de l'image, puis des couches entièrement connectées pour prédire les boîtes et les classes (Redmon *et al.*, 2015). YOLO est principalement conçu pour la détection d'objets, c'est-à-dire localiser et classer plusieurs objets dans une image en une seule passe. Cependant, il peut aussi être adapté à la classification d'images, notamment dans des contextes où la localisation et la classification sont liées, ce qui est particulièrement pratique dans notre cas où le but est de détecter une feuille d'arbre dans une image pour ensuite la classer en fonction de l'espèce d'arbre auquel elle appartient (Redmon *et al.*, 2015).

Aux cours de ces dernières années, les modèles de YOLO ont grandement évolué, notamment grâce à une puissance de calcul en perpétuelle ascension et à des données d'entraînement de plus en plus disponibles et de qualité. Cela a permis une forte amélioration de la précision, de la vitesse et la robustesse des détections (Jiang *et al.*, 2022).

L'identification d'espèces végétales, incluant donc les arbres, à partir de photographies de feuilles, passe de plus en plus par l'utilisation de systèmes basés sur des réseaux de neurones profonds. Les méthodes exploitent des CNN pour extraire des descripteurs morphologiques et texturaux, permettant de fournir une meilleure représentation des images de feuilles que les caractéristiques manuelles utilisées dans les systèmes classiques. Ces approches offrent une alternative apprenante de manière autonome, évolutive et plus facilement extensible à de nouvelles espèces (Barré *et al.*, 2017; Bisen, 2020; Tan *et al.*, 2020).

Les travaux de Kumar *et al.* (2012) présentent *Leafsnap*, la première application mobile grand public dédiée à l'identification automatique d'espèces d'arbres à partir de photographies de feuilles, en s'appuyant sur

des techniques avancées de vision par ordinateur et d'intelligence artificielle. L'objectif principal de leurs travaux est de permettre à toute personne utilisatrice de reconnaître facilement une espèce d'arbre en photographiant une feuille, en s'appuyant sur une base de données couvrant 184 espèces d'arbres du nord-est des États-Unis (Kumar *et al.*, 2012). Le système *Leafsnap* repose sur plusieurs étapes clés comme le filtrage des images pour éliminer automatiquement des photos qui ne contiennent pas de feuilles, la segmentation pour extraire précisément la feuille sur un fond neutre, facilitant l'analyse ultérieure, l'extraction de caractéristiques et enfin, la classification qui permet d'identifier l'espèce par comparaison des caractéristiques précédemment extraites. *Leafsnap* atteint des performances de pointe sur des images réelles issues du *Leafsnap dataset*, qui est à l'époque le plus grand ensemble de données de ce type. L'application a été téléchargée près d'un million de fois, ce qui témoigne de son utilité et de son accessibilité. Le système est capable de traiter des images prises dans des conditions variées, ce qui le rend pertinent pour une utilisation sur le terrain (Kumar *et al.*, 2012). Cependant, la solution présente des limites : les performances peuvent diminuer si la feuille n'est pas bien isolée ou si l'image est de mauvaise qualité. Le système est initialement limité aux espèces du nord-est des États-Unis, mais la méthodologie est généralisable (Kumar *et al.*, 2012).

Les travaux de recherche de Cerutti *et al.* (2013) publiés dans le journal *Computer Vision and Image Understanding*, présentent une méthode de vision par ordinateur pour identifier les espèces d'arbres à partir de photos de feuilles prises dans un environnement naturel complexe. Dans cette étude, les auteurs détaillent un système en plusieurs phases, comprenant une technique de segmentation (utilisant des contours actifs basés sur des polygones) et une représentation avec couleurs combinées. L'algorithme commence par extraire le contour de la feuille à partir de la photo, même si l'arrière-plan est complexe. Après la segmentation, le système analyse la forme de la feuille à l'aide de descripteurs géométriques avancés, inspirés de la botanique, notamment pour la forme globale, les bords et les caractéristiques locales comme la pointe et la base de la feuille. Ces descripteurs permettent une interprétation sémantique (c'est-à-dire des concepts botaniques compréhensibles) et offrent de meilleures performances que les méthodes statistiques classiques. L'objectif des travaux est le développement d'une application mobile, nommée *Folia*, permettant aux personnes utilisatrices d'identifier des espèces d'arbres de manière intuitive et éducative. Les résultats montrent que la méthode est efficace pour segmenter et classer les feuilles de 50 espèces d'arbres feuillus européens (Cerutti *et al.*, 2013).

L'étude de Zhao *et al.* (2015) décrit *ApLeaf*, une application mobile basée sur Android pour l'identification automatique de 126 espèces d'arbres à partir de photographies de leurs feuilles. Le système a été développé

pour offrir une alternative à l'application *Leafsnap*, présentée plus haut mais basée sur iOS, en se concentrant cette fois-ci sur le système d'exploitation Android. Le processus d'identification comprend également trois étapes principales : la segmentation de l'image de la feuille, l'extraction de caractéristiques comme les histogrammes de gradients orientés (qui décrivent les contours des objets en mesurant comment la luminosité varie selon différentes directions) et les caractéristiques de couleur, et enfin, l'identification de l'espèce via une mesure de distance obtenue avec l'intersection d'histogrammes. Testé sur la base de données *ImageCLEF 2012*, *ApLeaf* est conçu pour les écologistes, les botanistes amateurs et les éducateurs, en fournissant des résultats précis et en permettant l'identification sans connexion Internet (Zhao *et al.*, 2015).

Enfin, les travaux de Barré *et al.* (2017) décrivent la création et l'évaluation de *LeafNet*, un système de vision par ordinateur basé sur un CNN pour l'identification automatique de manière plus générale des espèces végétales à partir d'images de feuilles. L'objectif principal de cette étude est de démontrer que l'apprentissage automatique des caractéristiques par un CNN offre une meilleure représentation des caractéristiques des images de feuilles que les méthodes traditionnelles qui reposent sur des algorithmes élaborés manuellement. Les auteur.e.s ont comparé les performances de LeafNet sur des ensembles de données publics (*LeafSnap*, *Flavia* et *Foliage*) avec des systèmes personnalisés existants et ont pu démontrer que leur approche d'apprentissage profond les surpasse (Barré *et al.*, 2017).

En conclusion, nous avons vu que la vision par ordinateur et notamment les CNN peuvent être très efficaces pour la reconnaissance d'espèces d'arbres, en particulier avec les photos de leurs feuilles. Le transfert d'apprentissage permet de profiter de modèles robustes en les adaptant à des besoins spécifiques, comme démontré dans l'étude de Zhao *et al.* (2024).

2.2 Systèmes de recommandation pour la planification des plantations

Dans le contexte de la planification de plantation, les systèmes de recommandation (SR) sont des outils d'aide à la décision dans des environnements complexes (Lucchese *et al.*, 2021). Les SR ont beaucoup évolué, avec l'intégration de méthodes basées sur des algorithmes d'apprentissage profond, qui ont amélioré ainsi leur perfomances (Fan *et al.*, 2019; Gao *et al.*, 2022). Le fonctionnement d'un SR est décrit dans la littérature comme un processus en trois phases : la collecte d'informations, l'apprentissage et la prédiction / recommandation (Lucchese *et al.*, 2021).

La première phase, de collecte d'informations, a pour objectif de construire un profil ou un modèle de

la personne utilisatrice à partir de données pertinentes. La deuxième phase, d'apprentissage, utilise des algorithmes d'apprentissage automatique ou de fouille de données pour modéliser les préférences des personnes utilisatrices à partir des données collectées. Plusieurs techniques peuvent être utilisées comme les modèles linéaires, la factorisation de matrices ou encore les réseaux de neurones, pour apprendre des représentations latentes des personnes utilisatrices et des objets (c'est-à-dire des caractéristiques qui ne sont pas visible directement dans les données brutes), ou des règles de similarité (Lucchese *et al.*, 2021). La troisième phase, de prédiction / recommandation, produit soit une prédiction de score pour un objet donné, soit une liste d'objets recommandés.

Dans notre cas, la valorisation des services écosystémiques et de la biodiversité est centrale dans la sélection des espèces à planter. Dans la littérature il est démontré que la biodiversité favorise le fonctionnement des écosystèmes, surtout en conditions de changement climatique global (Hong *et al.*, 2021; Beillouin *et al.*, 2021). Les plantations monospécifiques sont beaucoup moins efficaces (Hua *et al.*, 2022).

Des SR comme les modèles de distribution des espèces (*Species Distribution Models*, *SDM*) et les systèmes d'aide à la décision sont des outils puissants pour recommander les espèces d'arbres à planter en ville, en tenant compte des critères écologiques, sociaux, économiques et de santé publique. Ils utilisent des données d'occurrence d'espèces, combinées entre autres à des variables environnementales, pour prédire les espèces qui seraient le plus à même de s'établir dans une zone donnée et en maximisant les services écosystémiques. Plusieurs articles de recherche, présentés ci-après, présentent des systèmes concrets mis en place.

L'étude de Werbin *et al.* (2020) présente un outil d'aide à la décision destiné à optimiser la plantation d'arbres pour atténuer la chaleur urbaine et améliorer l'équité sociale dans la ville de Boston. L'objectif principal de l'outil est de guider les fonctionnaires et les personnes résidentes pour leurs projets de plantations, d'une part en recommandant les régions prioritaires pour l'expansion de la canopée, et d'autre part en recommandant des espèces à planter, afin de contrer le phénomène d'îlot de chaleur urbain (Werbin *et al.*, 2020).

Le travail de Nyelele et Kroll (2021) propose un cadre de soutien à la décision multi-objectifs pour l'aménagement urbain qui vise à prioriser certains emplacements de plantation d'arbres en milieu urbain afin de maximiser les avantages des services écosystémiques. L'équipe utilise une méthodologie spatiale et des

techniques d'optimisation mathématique, telles que la programmation linéaire et réalisent une étude de cas dans le Bronx (*New York*) en explorant quatorze scénarios d'optimisation en considérant des objectifs tels que la réduction des polluants atmosphériques et de l'indice de chaleur, les coûts de plantation, et l'égalité de la couverture arborée. Les résultats de l'étude démontrent comment cette méthode peut guider les décideurs à opter pour des plantations d'arbres plus efficaces et plus justes. L'article souligne aussi la nécessité capitale de passer des approches subjectives à des approches systématiques pour les initiatives de plantation d'arbres en milieu urbain (Nyelele et Kroll, 2021).

2.3 Applications mobiles citoyennes

La littérature montre que les applications mobiles citoyennes représentent un moyen efficace pour augmenter la participation des personnes citoyennes à la gestion urbaine et à la gouvernance des services publics. En effet, la technologie mobile permet de rapprocher les administrations et les usagers, en fluidifiant la communication et en simplifiant la remontée d'informations (Ertiö, 2015; Allen *et al.*, 2020).

Ces applications couvrent de nombreux domaines, comme le signalement d'incidents urbains (trous dans la chaussée, mobilier urbain défectueux, etc.), la participation à la planification urbaine et la gestion environnementale (Brovelli *et al.*, 2016). Les études montrent que le succès de ces applications citoyennes dépend fortement de la perception de leur utilité par les personnes utilisatrices, de leur facilité d'utilisation et de la confiance envers les institutions qui les proposent (Hou *et al.*, 2020).

L'étude de Capponi *et al.* (2019) traite des systèmes de *Mobile Crowdsensing* (*MCS*), qui utilisent les smartphones des personnes citoyennes pour la collecte de données urbaines. Les auteurs exposent les défis cruciaux de ces systèmes, d'une part les préoccupations liées à la confidentialité des données et d'autre part la nécessité de développer des mécanismes efficaces pour encourager la participation des personnes utilisatrices et garantir la fiabilité des informations.

Également les recherches de Ertiö (2015) montrent que les applications mobiles citoyennes, sont d'une importance capitale pour le développement de la démocratie locale en planification urbaine, notamment en surmontant les limites des méthodes de participation traditionnelles. Dans l'étude trois aspects clés ont été mis en avant pour la catégorisation des applications mobiles citoyennes :

1. Le type de données collectées, qui différencie les applications centrées sur les personnes (qui visent à comprendre le comportement des personnes utilisatrices), et les applications centrées sur l'envi-

ronnement (qui visent à collecter des données environnementales) (Ertiö, 2015).

2. Le flux d'information, qui peut être soit un transfert unidirectionnel, soit un échange interactif avec éventuellement un dialogue entre les personnes citoyennes et les pouvoirs publics (Ertiö, 2015).
3. Et enfin, l'autonomisation des personnes citoyennes, qui peut être soit opérationnelle c'est-à-dire qu'elles ont le pouvoir d'influencer la manière dont une politique ou un service public est mis en œuvre concrètement ; soit stratégique où les personnes citoyennes ont la capacité de déterminer la politique ou le service lui-même, par exemple en définissant directement les objectifs à long terme (Ertiö, 2015).

Les travaux de Allen *et al*. (2020) qui traitent du rôle de la participation citoyenne en ligne dans l'amélioration des services urbains, montrent qu'une augmentation du niveau de participation des personnes citoyennes via des applications mobiles citoyennes améliore significativement la performance des services du secteur public.

Cependant, les application mobiles citoyennes sont confrontées à de nombreux défis, comme la qualité des données produites par les personnes utilisatrices (Bastos *et al*., 2022), ainsi que la sécurité et la confidentialité des informations collectées (Christin, 2016).

CHAPITRE 3
RECONNAISSANCE D'ESPÈCES

Dans ce chapitre nous allons décrire la première composante de l'application qui permet l'identification automatique des espèces d'arbres à partir de photos de feuilles. L'objectif de ce module de reconnaissance est de prendre en entrée une image de feuille capturée par la personne utilisatrice et de donner en sortie l'espèce de l'arbre en question. Dans le cadre de ce projet, nous avons entraîné un CNN sur les 150 espèces les plus abondantes dans la région de Montréal. Nous présenterons dans la partie 3.1 les différentes étapes de la constitution de l'ensemble de données nécessaire pour l'entraînement du modèle. Ensuite dans la partie 3.2 nous nous intéresserons au modèle de vision par ordinateur en lui-même, son entraînement et son optimisation. Enfin, nous traiterons de l'évaluation de ce dernier dans la partie 3.3. Cette fonctionnalité de reconnaissance constitue la première étape de la solution globale, les prédictions d'espèces serviront de paramètres d'entrée au module de recommandation qui proposera des espèces à planter dans les espaces privés.

3.1 Données d'entraînement

3.1.1 Récoltes des données

La constitution d'un ensemble de données riche et diversifié a été une étape cruciale dans notre travail. En effet, la qualité des données est essentielle pour l'entraînement d'un modèle d'apprentissage profond efficace.

Avant de récolter les données nécessaires à l'entraînement du modèle nous avons dû établir la liste des espèces d'arbres les plus répandues dans la région de Montréal. Pour cela nous nous sommes basés sur deux sources :

1. Les bases de données des arbres publics fournis par les villes de Montréal, Saint-Lambert, Repentigny, Boucherville, Varennes, Laval, Rosemère et Ville de Mont-Royal.
2. Les données de The Global Biodiversity Information Facility (2025) (GBIF), une infrastructure internationale de données qui met à disposition, en accès libre, des milliards d'observations sur la biodiversité mondiale (plantes, animaux, champignons, etc.).

Par la suite, nous avons procédé à une phase de nettoyage et d'enrichissement de cette base de données des espèces d'arbres présentes dans la région de Montréal. En effet, les noms des espèces d'arbres présents dans ces inventaires peuvent présenter certaines différences de notation. Il existe plusieurs synonymes de noms communs d'espèces d'arbres qui font référence à la même espèce, il a donc fallu uniformiser le nom des espèces en recherchant un identifiant unique, nous avons donc choisi pour cela le nom scientifique (nom binomial) de l'espèce.

Le nom scientifique d'une espèce d'arbre est une désignation universelle, généralement en latin, permettant d'identifier précisément une espèce végétale (d'arbres dans notre cas), indépendamment des variations du nom commun de chaque espèce (Welch, 1991). Il suit les règles de la nomenclature binomiale, comprenant le genre et l'espèce, parfois suivi du nom de l'auteur qui a découvert l'espèce. Il sert de référence stable pour relier entre elles les données de recherche, les inventaires et les bases de données (Winston, 2018).

Pour faire le lien entre chaque nom commun disponible dans notre base de données et les noms scientifiques, nous avons réalisé un script en langage *python* permettant de récupérer, à partir des données de *Catalogue Of Life*, la liste de tous les synonymes des noms des espèces d'arbres reliées à chaque nom scientifique. *Catalogue of Life* (*CoL*) (Bánki *et al.*, 2025) est un référentiel taxonomique mondial qui a pour objectif de fournir une liste unifiée et cohérente des espèces connues. Il s'appuie sur un réseau international de taxonomistes et de bases de données spécialisées. Il agrège donc des listes expertisées afin de fournir pour chaque taxon, entre autres, un nom accepté et des synonymes. C'est une ressource de référence pour harmoniser les noms d'espèces et limiter les ambiguïtés liées aux variations de nomenclatures.

Cette étape a abouti à l'élaboration d'une base de référence listant les noms scientifiques des espèces d'arbres présentes dans la région de Montréal avec leur synonymes (allant jusqu'à plusieurs dizaines pour certaines espèces)(tableau 3.1). Cela nous a permis d'associer à l'ensemble des arbres de l'inventaire leur nom scientifique correspondant.

Nom commun latin	**Nom français**	**Nom scientifique**	**Synonymes (liste non exhaustive)**
Acer platanoides	Érable de Norvège	Acer platanoides L.	Acer acutifolium St.-Lag., 1889 / Acer dasyphyllum Nicholson / Acer dobrudschae Pax, 1886
Acer saccharinum	Érable argenté	Acer saccharinum L.	Acer album Hort. / Acer album hort. ex G.Nicholson / Acer coccineum Wender. / Acer collinsonia Thunb., 1793
Gleditsia triacanthos	Févier épineux	Gleditsia triacanthos L.	Acacia americana Stokes / Acacia inermis Steud. / Acacia laevis Steud. / Acacia triacanthos Gron.
Fraxinus pennsylvanica	Frêne rouge	Fraxinus pennsylvanica Marshall	Calycomelia campestris (Britton) Nieuwl. & Lunell / Fraxinus juglandifolia var. aucubifolia H.Jaeger
Tilia cordata	Tilleul à petites feuilles	Tilia cordata Mill.	Tilia borbasiana Braun. / Tilia parvifolia Ehrh. / Tilia sylvestris Desf. ex Bonnier & Layens, 1894 / Tilia cordata Miller

Tableau 3.1 – Exemple des cinq espèces d'arbres les plus abondante dans la région de Montréal.

Les photographies de feuilles d'arbres annotées en fonction de l'espèce de l'arbre ont été principalement récoltées à partir des bases de données de GBIF (The Global Biodiversity Information Facility, 2025), provenant de musées, d'herbiers, de projets de science citoyenne, d'institutions de recherche et d'université. GBIF a une importance capitale pour la mise à disposition des données sur la biodiversité en permettant de récupérer de grands volumes d'images et de métadonnées géographiques ou taxonomiques (Edwards *et al.*, 2000).

De plus, le fait que GBIF dispose d'une API (*Application Programming Interface*, interface de programmation d'application) est un atout majeur pour récupérer de grandes quantités d'images, car cela permet d'automatiser entièrement le processus d'extraction des données. Plutôt que de devoir télécharger manuellement des ensembles de données via l'interface web, on peut interroger directement l'*Occurrence API* de GBIF (avec des filtres précis comme l'espèce, la zone géographique, le type de média, la période, etc.), et de procéder par la suite au téléchargement automatique de toutes les images retournées.

Lors du processus de récoltes des photos sur GBIF (GBIF Secretariat, 2023), nous avons filtré les occurrences selon plusieurs critères :

- La taxonomie, en spécifiant le nom scientifique faisant partie de la sélection des 150 espèces d'arbres considérées comme pertinentes pour la région de Montréal.
- La zone géographique, en priorisant les photos provenant du Québec ou dans des régions proches (quand les données disponibles le permettaient), afin de représenter au mieux les espèces avec leur spécificités propres à la région.
- Le type de photos, en restreignant la recherche aux images contenant des feuilles uniquement.

L'une des sources de données disponible sur GBIF et qui nous a été précieuse est *iNaturalist* (Matheson, 2014), une plateforme de science citoyenne à code source ouvert (*open source*) permettant aux personnes utilisatrices de documenter, d'identifier et de partager des observations de biodiversité à partir de photos, très utilisée pour la recherche et la conservation.

Nous avons ainsi pu obtenir un ensemble de données d'environ 15 000 images de feuilles d'arbres annotées, avec plus d'une centaine d'image pour certaines espèces.

3.1.2 Préparation et prétraitement

Avant de pourvoir utiliser les données récoltées pour entraîner de notre modèle de reconnaissance d'espèces, il a fallu préparer ces données en effectuant plusieurs prétraitements.

Dans un premier temps un filtrage manuel des images a été nécessaire. Les images floues ou de mauvaise qualité (avec une résolution insuffisante) ont été retirée de l'ensemble de donnée. Nous avons également sélectionné lors de cette étape les images où la feuille est l'élément principal, afin de ne pas perturber l'apprentissage et qu'il puisse se faire sur les caractéristiques propres des feuilles d'arbres et non sur l'environnement alentour réduisant ainsi les bruits de fond. Un traitement manuel a été aussi réalisé pour recadrer des images contenant des éléments pouvant induire en erreur le modèle, par exemple certaines images de GBIF sont des photos d'herbier et contiennent des étiquettes et des règles de mesure.

Dans un second temps un traitement automatique a été effectué sur les données pour normaliser les images. Tout d'abord, un CNN requière une entrée de taille fixe. Pour y parvenir plusieurs approches existent, comme le redimensionnement direct, le recadrage centré et le rembourrage. Le recadrage a été écarté car il risque de tronquer les régions les plus discriminantes de la feuille, et le rembourrage introduit des bordures artificielles susceptibles de perturber l'apprentissage. Le redimensionnement direct, bien qu'il ne conserve pas le ratio d'aspect original, a été retenu car la déformation géométrique introduite reste négligeable par rapport à la variabilité naturelle du jeu de données. Nous avons donc redimensionnées les images à une taille fixe de 320×320 pixels (figure 3.2) ce qui permet de garantir que toutes les entrées ont la même taille pour le réseau de neurones. Puis nous avons converti les images en tenseurs numériques (une structure de données multidimensionnelle qui stocke des valeurs numériques de manière organisée). Comme une image numérique en couleur est composée de trois canaux correspondant aux composantes Rouge, Vert et Bleu (modèle RVB), et chaque canal est une matrice 2D où les éléments représentent l'intensité de la couleur correspondante pour un pixel donné, les tenseurs sont donc de dimension 3 (une dimension pour chaque couleur). Le type utilisé est le *float32* (nombre à virgule flottante sur 32 bits), qui est le format numérique standard en apprentissage profond. Cette conversion des valeurs des pixels du type entier à un type flottant, permet le calcul des gradients et la mise à jour des poids lors de l'optimisation. Le passage à des tenseurs structurés et ordonnés (figure 3.1) de manière homogène permet aussi d'optimiser l'utilisation des GPU lors de l'entraînement du modèle (Pal et Sudeep, 2016).

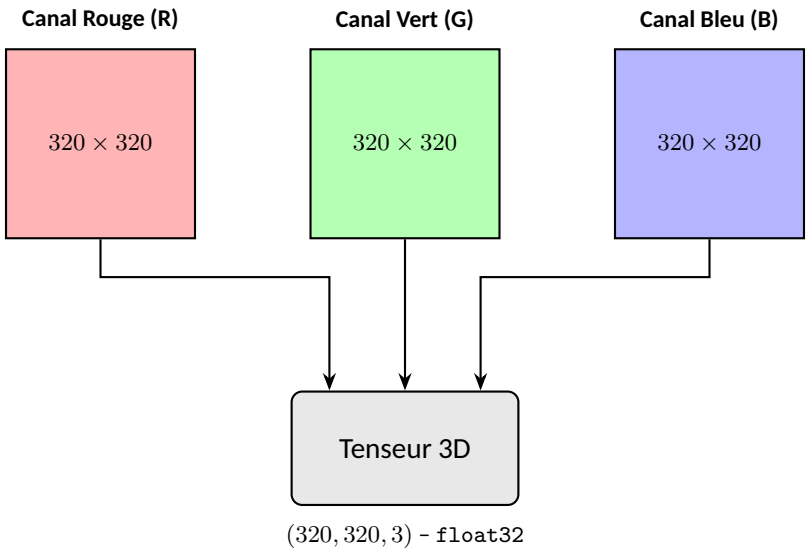


Figure 3.1 – Représentation d'une image couleur en tenseur tridimensionnel.

Figure 3.2 – Image originale.

Figure 3.3 – Image prétraitée.

Enfin, la dernière étape du prétraitement des données a été de les diviser en trois sous-ensembles distincts (tableau 3.2) :

— Un ensemble de données d'entraînement qui sert à ajuster les paramètres du réseau de neurones. Le CNN se base sur ces images pour apprendre à associer des motifs visuels (forme, nervures, texture

de la feuille, etc.) aux étiquettes d'espèces. Ce sous-ensemble représente 60 % de notre ensemble total de données.

— Un ensemble de données de validation qui joue un rôle de données de contrôle pendant l'entraînement. Il n'est pas utilisé pour mettre à jour les poids, mais pour évaluer les performances du modèle au fur et à mesure des époques, afin d'éviter notamment le surapprentissage. Ce sous-ensemble représente 20 % de notre ensemble total de données.
— Un ensemble de données de test qui sert strictement pour l'évaluation finale. Il n'intervient ni dans l'optimisation ni dans le choix des hyperparamètres. Il nous permet d'avoir une estimation concrète de la capacité de généralisation du modèle à de nouvelles images (qu'il n'a pas rencontré pendant l'entraînement). Ce sous-ensemble représente 20 % de notre ensemble total de données.

Ensemble	**Proportion**	**Nombre d'images**
Entraînement	60 %	8 500
Validation	20 %	2 834
Test	20 %	2 834
Total	100 %	14 168

Tableau 3.2 – Répartition des données selon les sous-ensembles.

3.1.3 Augmentation de données

Après le traitement et le nettoyage de l'ensemble de données décrits précédemment, il apparaît un déséquilibre du nombre d'images par espèce. En effet, après la sélection d'images pertinentes, certaines espèces n'avaient plus que quelques dizaines d'images pour les représenter.
Pour pallier ce problème de manque d'exemples dans certaines classes et de diversité dans les données récoltées, et afin d'améliorer la capacité de généralisation du modèle, nous avons appliqué des techniques d'augmentation de données.

L'augmentation de données est un ensemble de techniques qui permettent de créer artificiellement de nouveaux exemples d'apprentissage à partir des images existantes afin d'améliorer l'entraînement des CNN. C'est particulièrement utile lorsque la quantité de données disponibles est limitée comme dans notre cas, en augmentant la taille effective de l'ensemble de données d'entraînement sans avoir à collecter de nouvelles

images (Zheng *et al.*, 2020). Cela permet également de réduire le surapprentissage car le modèle voit des versions un peu différentes d'une même image, il apprend donc des motifs plus généraux. En élargissant la diversité des données, on améliore grandement la robustesse du modèle aux conditions réelles, c'est-à-dire avec des photos prises par des personnes utilisatrices ne suivant pas de règles précises (Alomar *et al.*, 2023).

Pour augmenter nos données trois types de transformations ont été appliquées :

- Des transformations géométriques comme des rotations, des retournements horizontaux et verticaux, des translations et des recadrages aléatoires pour permettre de rendre le modèle insensible à l'orientation de la feuille et aux petites variations de cadrage.
- Des transformations photométriques, en ajustant la luminosité, le contraste et la saturation, mais également en ajoutant du bruit gaussien. Cela permet de simuler des prises de photos par des personnes utilisatrices en conditions réelles.
- Des masquages partiels et aléatoires de petites régions de l'image pour simuler des occlusions partielles ou des défauts dans la feuille (déchirures, zones mangées par des insectes, etc.).

Nous avons limité ces transformations pour que les images restent biologiquement plausibles, c'est-à-dire qu'elles continuent de représenter des feuilles reconnaissables et réalistes. Des transformations trop agressives auraient pu produire des images artificielles qui ne correspondent à aucune espèce réelle, ce qui introduirait du bruit dans l'apprentissage réduisant ainsi les performances du modèle. Nous avons donc limité l'amplitude des transformations afin de préserver les caractéristiques discriminantes des feuilles. Les paramètres de ces transformations ont été définis de manière empirique en inspectant visuellement un échantillon d'images augmentées, en s'assurant qu'elles restaient cohérentes avec l'apparence naturelle des feuilles des espèces du jeu de données. Par exemple, nous avons limités les ajustements de la luminosité à des variations de $\pm$ 35 %, pour de simuler des conditions d'éclairage variables mais sans trop dévier des teintes caractéristiques de l'espèce, et les masquages aléatoires à $\pm$ 20 % de la surface de l'image, afin que les structures morphologiques essentielles à l'identification soient conservées.

Cette augmentation de données, qui a permis d'enrichir fortement notre ensemble de données, est effectuée automatiquement au moment du chargement des mini-lots lors de l'entraînement, ce qui permet de présenter au modèle des variations différentes à chaque époque d'entraînement (figure 3.4).

(a) Image originale

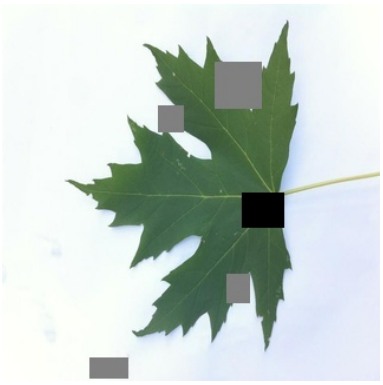
(b) Rotation légère (30°)

(c) Retournement horizontal

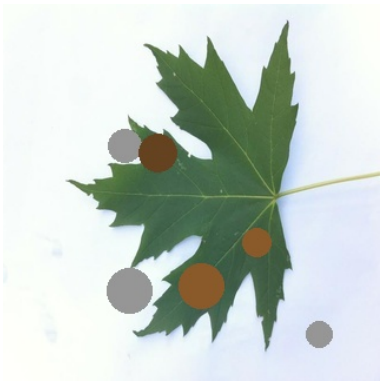
(d) Luminosité réduite

(e) Luminosité augmentée

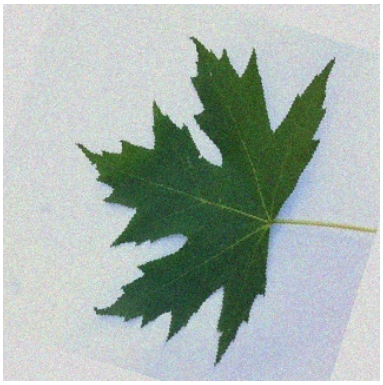
(f) Bruit gaussien



(g) Masquage rectangulaire



(h) Occlusion circulaire



(i) Transformations combinées

Figure 3.4 – Exemples d'augmentation de données appliquée à une image de feuille d'érable argenté.

3.2 Entraînement du modèle

3.2.1 Architecture et choix du modèle : Transfert d'apprentissage

Dans notre projet, nous avons fait le choix d'entraîner un modèle de CNN par transfert d'apprentissage, en nous appuyant sur un modèle pré-entraîné. Il est évidement possible d'entraîner un CNN à partir de zéro (*from scratch*), c'est-à-dire en initialisant tous les poids du réseau et en les apprenant uniquement à partir de notre ensemble de données d'images de feuilles d'arbres, mais en pratique cette approche présente plusieurs limites.
L'entraînement d'un CNN à partir de zéro nécessite généralement de grandes quantités d'images. Nous aurions eu besoin d'un ensemble de données beaucoup plus important, or le nombre d'images de feuilles d'arbres annotées disponibles et exploitables par espèce reste assez limité, et inégalement réparti. De plus, certaines espèces sont visuellement très proches et se différencient par des détails subtils, une puissance de calcul importante aurait donc été nécessaire pour atteindre des performances convenables.

Le transfert d'apprentissage nous a permis de contourner ces limitations en réutilisant les connaissances déjà apprises par le modèle pré-entraîné sur un grand ensemble de données génériques. Pendant cet entraînement, le réseau apprend des filtres convolutifs capables de détecter des motifs visuels généraux comme les bords, les textures, les motifs répétitifs, les structures géométriques, etc. (Yosinski *et al.*, 2014). Ces représentations de bas et moyen niveau sont ensuite réutilisées pour d'autres tâches, comme dans notre projet pour orienter le modèle à la classification d'images de feuilles d'arbres.
Les premières couches d'un CNN apprennent des caractéristiques génériques qui sont adaptées à de nombreux domaines de vision par ordinateur. Ce sont les couches profondes uniquement et le classifieur final qui capturent des informations spécifiques à la classification voulue (Yosinski *et al.*, 2014).

Le transfert d'apprentissage présente donc plusieurs avantages pour notre projet :

- Les besoins en données sont réduits, nous nous n'avons donc pas besoin de disposer d'une aussi grande quantité d'images de feuilles d'arbres que pour un entraînement à partir de zéro.
- La convergence est plus rapide vu que le modèle part d'un état déjà entraîné et non d'une initialisation aléatoire, ce qui accélère ainsi fortement l'entraînement.
- On obtient une meilleure généralisation car les représentations de bas niveau pré-apprises sont robustes aux variations de conditions (comme l'angle, l'éclairage, l'arrière-plan). Cet atout est crucial

dans notre projet, car on ne peut pas imposer aux personnes utilisatrices des conditions de prises de photos contraignantes.

— Des performances supérieures, car les descripteurs appris par les modèles pré-entraîné incluent déjà des représentations pertinentes pour les objets naturels et les textures organiques.

C'est pour toutes ces raisons que nous avons jugé plus pertinent d'adopter une approche par transfert d'apprentissage plutôt que d'entraîner un CNN à partir de zéro.

Notre choix s'est ensuite porté sur YOLO (*You Only Look Once*) (Jocher et Qiu, 2024), une famille de modèles d'intelligence artificielle en vision par ordinateur qui analyse une image en une seule passe pour en extraire une prédiction utile. Les modèles YOLO ont été entraînés pour la classification sur le sous-ensemble *ImageNet-1K* d'*ImageNet* (Deng *et al.*, 2009), une grande base de données d'images annotées conçue pour la recherche en vision par ordinateur, comprenant 1,2 million d'images réparties en 1 000 classes. Selon les variantes, il peut reconnaître la classe principale d'une image (classification) ou repérer et localiser plusieurs objets en indiquant leurs classes individuelles et leur position (détection et segmentation). Il est souvent utilisé car il combine de bonnes performances avec une exécution rapide et efficace, ce qui le rend pratique pour des applications en temps réel.

L'étape suivante a été de choisir parmi les versions que propose YOLO, celle qui répond à notre besoin en termes de performance, de rapidité et de consommation de ressources. Bien que YOLO soit historiquement associé à la détection d'objets en temps réel, certaines versions ont été adaptées pour la classification d'images. YOLO11, lancé en octobre 2024, représente l'évolution la plus récente de la famille des modèles YOLO (figure 3.5). Son efficacité est nettement supérieure à sa version précédente (YOLOv8) tout en réduisant le nombre de paramètres de 48 %. Une nouvelle architecture C3k2 optimisée et un mécanisme d'attention spatiale C2PSA, que nous détaillerons plus bas, ont été ajoutés à cette version (Khanam et Hussain, 2024).

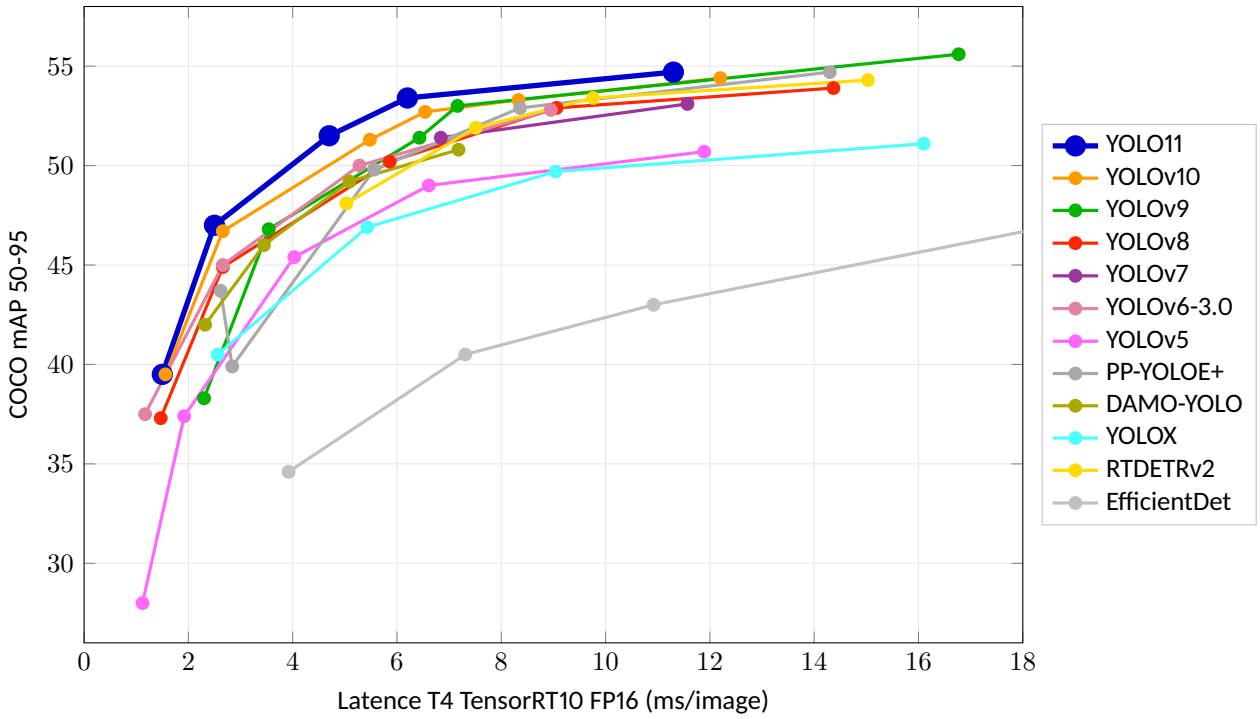


Figure 3.5 – Évaluation comparative (*benchmark*) des versions de YOLO (Ultralytics, b).

Où :

- **COCO** (*Common Objects in Context*) est le jeu de données de référence pour comparer les modèles de vision par ordinateur,
- **mAP 50-95** est la Précision Moyenne moyenne (*mean Average Precision*) calculée en faisant la moyenne sur plusieurs seuils d'intersection sur union (*IoU*, *Intersection over Union*). Plus concrètement lorsque le modèle détecte un objet, il dessine une boîte autour de lui. L'IoU mesure le degré de chevauchement entre la boîte prédite et la boîte réelle. Plutôt que de choisir arbitrairement un seul seuil, la mAP 50-95 calcule la précision moyenne sur dix seuils différents puis en fait la moyenne,
- **Latence T4 TensorRT10 FP16 (ms/image)** est la mesure de vitesse d'inférence, c'est-à-dire le temps que met le modèle pour traiter une image,
- **T4** est le GPU NVIDIA Tesla T4, une carte graphique utilisée comme référence matérielle standard,
- **TensorRT10** est le moteur d'optimisation d'inférence de NVIDIA (version 10), qui compile et optimise le modèle pour l'exécuter le plus rapidement possible sur GPU,
- **FP16** (*Float 16 bits*) indique que les poids du modèle sont représentés en demi-précision plutôt qu'en FP32, ce qui divise la mémoire utilisée et accélère le calcul avec une perte de précision négligeable.

YOLO11-cls (la version de classification d'image de YOLO11) propose cinq variantes de complexité croissante :

Variante	Paramètres (M)	Top-1 Acc.	Top-5 Acc.	Vitesse CPU (ms)
YOLO11n-cls	2,8	70,0 %	89,4 %	5,0 $\pm$ 0,3
YOLO11s-cls	6,7	75,4 %	92,7 %	7,9 $\pm$ 0,2
YOLO11m-cls	11,6	77,3 %	93,9 %	17,2 $\pm$ 0,4
YOLO11l-cls	14,1	78,3 %	94,3 %	23,2 $\pm$ 0,3
YOLO11x-cls	29,6	79,5 %	94,9 %	41,4 $\pm$ 0,9

Tableau 3.3 – Performances des variantes YOLO11-cls (Ultralytics, b).

Les métriques utilisées dans le tableau 3.3 pour comparer les performances des modèles sont définies comme suit :

Exactitude au rang 1 (*Accuracy Top-1*).
L'exactitude au rand 1 mesure la proportion d'échantillons pour lesquels la classe prédite avec la plus haute probabilité correspond à la classe réelle :

$$\mathrm{Accuracy}_{\mathrm{Top1}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left[\arg\max_{c} \left(p_c^{(i)}\right) = y_i\right] \tag{3.1}$$

Où :

- N est le nombre total d'échantillons,
- $p_c^{(i)}$ est la probabilité prédite pour la classe c sur l'échantillon i,
- y_i est la classe réelle,
- $\mathbb{1}[\cdot]$ est la fonction indicatrice (elle vaut 1 si la condition est vraie, sinon 0).

Cette métrique reflète la performance du modèle lorsque la personne utilisatrice ne doit recevoir qu'une unique prédiction. Par exemple, une exactitude au rang 1 de 80 % signifie que le modèle identifie correctement l'espèce dans 80 % des cas.

Exactitude au rang 5 (*Accuracy Top-5*).

L'exactitude au rang 5 mesure la proportion d'échantillons pour lesquels la classe réelle figure parmi les 5 classes prédites avec les plus hautes probabilités :

$$\mathrm{Accuracy}_{\mathrm{Top5}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\Big[y_i \in \mathrm{Top5}\Big(p^{(i)}\Big)\Big] \tag{3.2}$$

Où :

- N est le nombre total d'échantillons,
- y_i est la classe réelle,
- $p^{(i)}$ est le vecteur de probabilités prédit pour l'échantillon i,
- $\mathrm{Top5}\big(p^{(i)}\big)$ est l'ensemble des 5 classes ayant les plus grandes probabilités dans $p^{(i)}$,
- $\mathbb{1}[\cdot]$ est la fonction indicatrice (elle vaut 1 si la condition est vraie, sinon 0).

Cette métrique permet de quantifier les cas où le modèle hésite entre des espèces similaires mais inclut tout de même la bonne réponse dans les 5 premières propositions.

La vitesse CPU

La vitesse CPU (*ms*) correspond au temps d'inférence moyen (exprimé en millisecondes) nécessaire au modèle pour traiter une seule image sur un processeur central (CPU). Cette métrique est pertinente dans un contexte de déploiement sur des appareils ne disposant pas de GPU dédié.

Il est important de noter que bien que l'exactitude au rang 1 et au rang 5 soient les métriques natives rapportées par YOLO pour la classification, elles présentent des limites. Elles sont très sensibles au déséquilibre entre les classes, un modèle qui néglige des classes minoritaires peut tout de même afficher une exactitude globale élevée. Également l'exactitude au rang 5 perd toute pertinence lorsque le nombre de classes est faible, car la probabilité d'y figurer par chance devient trop importante. Ces métriques traitent toutes les erreurs de manière identique et ne renseignent pas sur la nature des confusions entre classes. Elles restent néanmoins utiles pour comparer plusieurs variantes de modèles YOLO, car elles sont calculées de manière identique pour tous les modèles, ce qui garantit une base de comparaison homogène et reproductible.

Après une analyse des caractéristiques de chacune des variantes, notre choix s'est porté sur la variante YOLO11m-cls (*medium*), pour les raisons suivantes :

- Sa capacité représentationnelle est adéquate à notre projet. En effet, avec 11,6 millions de para-

mètres, le modèle dispose d'une capacité suffisante pour distinguer les 150 classes, sans être surdimensionné.
— Le risque de surapprentissage est maîtrisé car un modèle plus volumineux (large ou extra-large) présenterait un risque accru de surapprentissage sur un ensemble de données de taille modéré comme le nôtre.
— Le temps d'entraînement est raisonnable (environ 30 % plus rapide que la variante large), ce qui nous a permis de réaliser davantage d'expérimentations et d'itérations.
— Son déploiement est flexible, notamment parce qu'il est suffisamment compact pour éventuellement une inférence directement sur mobile après optimisation.
— Son inférence est rapide, ce qui est un atout majeur pour une utilisation interactive où la personne utilisatrice doit avoir l'espèce de l'arbre prédite par le modèle quasi instantanément.

Le modèle YOLO11m-cls est composé de deux blocs (figure 3.6), un réseau dorsal (*backbone*) et une tête de classification (Ultralytics, a) :

Le réseau dorsal convolutif est la partie du réseau qui extrait les cartes de caractéristiques à partir de l'image d'entrée. Il est constitué d'une succession de couches de convolutions empilées et de modules spécifiques. Le nouveau module C3k2 (*Cross Stage Partial with Kernel size 2*) est un bloc fondamental de l'architecture YOLO11. Il est basé sur le principe des réseaux CSP (*Cross Stage Partial Networks*) de Wang *et al.* (2020), qui consiste à diviser les cartes de caractéristiques en deux branches, l'une traverse une série de couches convolutives 3×3 tandis que l'autre contourne ces transformations via une connexion résiduelle (c'est a dire un chemin direct qui permet aux données de sauter une ou plusieurs couches du réseau sans être modifiées). Les deux branches sont ensuite concaténées puis fusionnées par une convolution finale. Cette architecture permet de propager efficacement les gradients tout en réduisant la redondance des calculs, accélérant ainsi significativement le traitement tout en maintenant la capacité d'extraction de caractéristiques (*features*). (Khanam et Hussain, 2024).

Un module C2PSA (*Cross Stage Partial with Spatial Attention*) a également était introduit dans cette nouvelle version. C'est un mécanisme d'attention multi-têtes (*Multi-Head Self-Attention*), similaire à celui utilisé dans les *transformeurs* (Vaswani *et al.*, 2017). Cette architecture permet au modèle de se concentrer sur les régions les plus pertinentes de l'image en apprenant à reconnaître automatiquement, avec plusieurs "vues"

en parallèle, les régions spatiales discriminantes. L'entrée passe d'abord par une convolution, puis traverse plusieurs blocs d'attention qui calculent des poids d'importance pour chaque région spatiale de la carte de caractéristiques. Les régions jugées plus informatives reçoivent un poids plus élevé que les zones moins pertinentes. Le module C2PSA permet donc au modèle d'apprendre dynamiquement à pondérer les différentes régions de l'image selon leur importance et de focaliser son attention sur les zones discriminantes plutôt que de traiter uniformément toute l'image (Khanam et Hussain, 2024). Cette innovation est idéale dans notre cas car elle permet entre autres de distinguer plus efficacement des nervures de feuilles ou des patterns de dentelure entre espèces ayant des feuilles morphologiquement similaires. Ce bloc combine donc calculs convolutionnels et attentions spatiales pour capter efficacement les éléments importants dans l'image.

La tête de classification permet quant à elle d'effectuer la classification. Elle comprend une couche de mise en commun globale par moyenne (*Global Average Pooling*) qui est appliquée aux cartes de caractéristiques finales pour obtenir un vecteur de caractéristiques de dimension fixe, qui résume l'image entière. Ce vecteur est ensuite passé à travers une couche entièrement connectée (*Dense*), puis dans une dernière couche linéaire dont la dimension est égale au nombre d'espèces d'arbres à prédire pour produire un score pour chaque classe. Enfin, une normalisation via une fonction *softmax* transforme ces scores en une distribution de probabilité sur les espèces (Ultralytics, a).

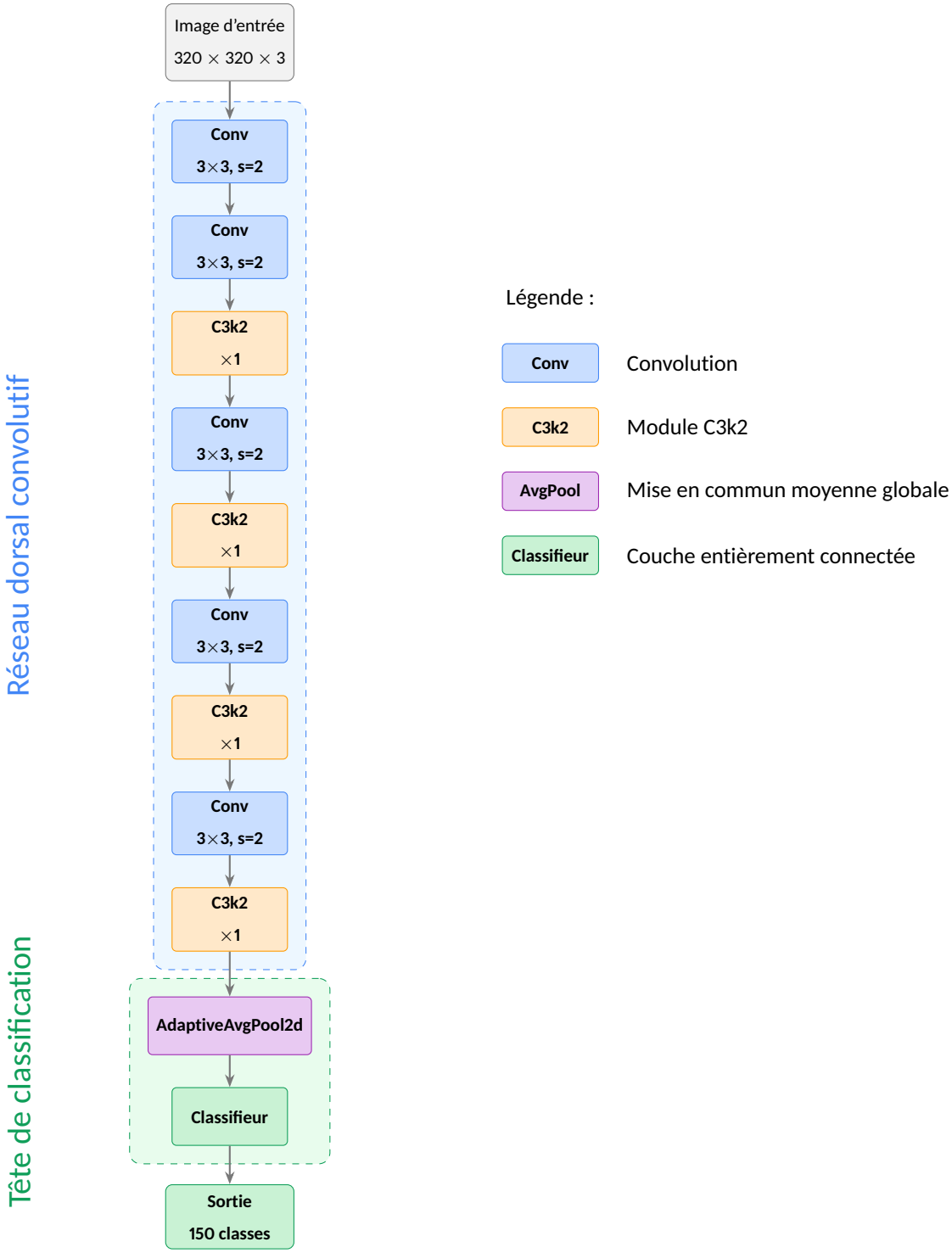


Figure 3.6 – Architecture de YOLO11m-cls (Ultralytics, b).

3.2.2 Stratégie d'ajustement fin

L'ajustement fin constitue l'étape centrale du transfert d'apprentissage. Le modèle YOLO11m-cls étant entrainé sur une large variété d'image et disposant d'une tête de classification configurée pour 1 000 classes, doit être adapté et optimisé pour effectuer des prédictions parmi les 150 espèces d'arbres de notre projet. Cet ajustement repose sur l'hypothèse fondamentale que les représentations apprises par un CNN sur une tâche source sont partiellement transférables à une tâche cible différente. Cela est possibles car les couches initiales d'un CNN apprennent des détecteurs de caractéristiques génériques qui sont universellement utiles pour la vision par ordinateur.

L'étude de Yosinski *et al*. (2014) montre que la transférabilité des caractéristiques apprises décroît avec la profondeur du réseau. Les couches proches de l'entrée encodent des primitives visuelles bas-niveau, tandis que les couches profondes capturent des concepts sémantiques de plus en plus spécifiques à la tâche d'origine.

L'efficacité du transfert d'apprentissage dépend de la similarité entre les domaines source et cible. Dans notre cas, *ImageNet-1K*, l'ensemble de données sur lequel a été entraîné le modèle YOLO11m-cls, contient de nombreuses catégories de plantes, de feuilles et d'objets naturels, ce qui établit une proximité sémantique favorable avec notre tâche de classification d'espèces d'arbres à partir de photo de leurs feuilles. En effet, le modèle pré-entraîné possède déjà des représentations pertinentes pour les textures organiques, les formes naturelles irrégulières, les variations de couleur en fonction des saisons et les motifs répétitifs.
Cette proximité des domaines source et cible, nous permet d'adopter une stratégie d'ajustement fin de l'ensemble du réseau, plutôt qu'un simple entraînement du classifieur final.

Notre stratégie d'ajustement fin s'articule en trois phases, permettant une adaptation progressive et contrôlée du modèle pré-entraîné vers notre tâche de classification des espèces d'arbres :

1. La première étape consiste à spécialiser la tête de classification en remplaçant tout d'abord la couche de sortie (la couche *Dense* finale de 1 000 neurones) par une nouvelle couche à 150 neurones, correspondant à nos 150 espèces d'arbres. Comme cette nouvelle couche n'a pas de poids pré-entraînés, elle est initialisée en utilisant l'initialisation de *Kaiming* (He *et al*., 2015). Cette méthode permet d'éviter que les activations et gradients explosent ou disparaissent au début de l'entraînement, de garantir une variance appropriée des activations dans le réseau profond et d'avoir une convergence

plus rapide et de meilleures performances qu'une initialisation naïve (He *et al.*, 2015).

Les poids du réseau dorsal (les blocs *C3k2* et le module *C2PSA*) sont figés et ne sont pas mis à jour par la rétropropagation. Ils représentent plus de 95 % des paramètres du modèle et encodent les connaissances transférables acquises lors de l'entraînement initial du modèle. Seule la tête de classification est entraînée.

Cette étape permet d'une part de commencer à stabiliser les gradients en évitant de perturber brutalement les représentations internes déjà apprises sur l'ensemble de données source et d'autre part de forcer la nouvelle tête de classification à apprendre à exploiter efficacement les caractéristiques déjà extraites par le réseau dorsal pour les projeter dans l'espace des espèces d'arbres.

2. La deuxième phase, le préchauffage (*warmup*) vise à stabiliser le processus d'optimisation avant l'entraînement intensif. Cette stabilisation est cruciale car la nouvelle tête de classification (de 150 neurones) peut produire initialement des gradients de grande magnitude qui pourraient détruire les représentations pré-apprises s'ils sont appliqués directement.

   Durant cette phase, le taux d'apprentissage croît linéairement de zéro jusqu'à sa valeur cible (0.0001). Cette progression graduelle permet aux gradients de se stabiliser avant d'atteindre leur amplitude nominale.

   Le préchauffage linéaire que nous utilisons est formalisé comme suit :

$$\eta(t) = \eta_{\min} + (\eta_0 - \eta_{\min}) \, \frac{t}{T_w} \, , \qquad 0 \leq t \leq T_w \tag{3.3}$$

   Où :

   — t est l'itération courante,
   — T_w est le nombre total d'itérations de la phase de préchauffage,
   — $\eta_{\min}$ est le taux d'apprentissage de départ,
   — η_0 est le taux d'apprentissage cible à atteindre à la fin du préchauffage.

   Le *momentum*, une technique d'optimisation qui accélère la convergence de la descente de gradient en accumulant une inertie basée sur les gradients passés, est également réduit afin de diminuer l'inertie de l'optimiseur et permettre ainsi une adaptation plus réactive aux gradients initiaux instables. En effet, comme il contrôle combien d'inertie est donné à l'optimisation, si on met un *momentum* très élevé dès le début, on peut accumuler une vitesse basée sur des gradients encore mal orientés en début d'entraînement, ce qui rends l'optimisation moins stable. Un *momentum* élevé avec des gradients bruités peut déstabiliser l'entraînement.

Cette phase de préchauffage se fait durant trois époques, une durée plus courte (1 ou 2 époques) ne permettrait pas une stabilisation suffisante des gradients de la nouvelle tête de classification et une durée plus longue (5 époques ou plus) retarderait inutilement l'apprentissage effectif sans bénéfice supplémentaire. Le préchauffage stabilise donc les gradients, évitant ainsi que des mises à jour trop agressives ne détruisent les représentations apprises par le modèle lors du pré-entraînement.

3. La troisième phase est l'ajustement fin principal, elle constitue l'entraînement à proprement dit du modèle. L'ensemble du réseau (le réseau dorsal et la tête de classification) est affiné pour optimiser la performance du modèle sur notre tâche de classification de feuilles d'arbres. Néanmoins, cela est fait avec un dégel progressif des couches profondes. Le but n'est pas d'ouvrir d'un coup tout le réseau de neurones, mais de ne libérer que les blocs les plus profonds dans un premier temps, ceux qui sont les plus proches de la sortie et donc les plus spécialisés pour la tâche source. Cela permet au modèle d'ajuster ses représentations de haut niveau aux spécificités des feuilles d'arbres, en limitant le risque d'oubli catastrophique des connaissances pré-apprises (*catastrophic forgetting*) (Kirkpatrick *et al.*, 2017), c'est-à-dire la perte des connaissances utiles apprises lors du pré-entraînement.

Notre stratégie d'ajustement fin du modèle YOLO11m-cls s'est donc articulé autour de la problématique essentielle de la conservation des connaissances acquises lors du pré-entraînement du modèle. Cela a pu se faire grâce à une adaptation progressive et une spécialisation contrôlée du modèle sur notre tâche de classification d'images de feuilles d'arbres.

3.2.3 Fonction de perte

La fonction de perte (*loss function*) est l'élément central de l'apprentissage profond supervisé. Elle donne l'écart entre les prédictions du modèle et les classes réelles, permettant ainsi de guider l'optimisation des paramètres du réseau. Notre modèle utilise la perte d'entropie croisée (*cross-entropy loss*) comme fonction de perte, c'est le choix standard pour une tâche de classification multi-classes. En effet, notre tâche consiste à classifier une image de feuille parmi des espèces d'arbres possibles. Il s'agit d'un problème de classification multi-classes mutuellement exclusives, c'est-à-dire que chaque image appartient à exactement une classe (une feuille ne peut pas provenir simultanément de deux espèces différentes d'arbres). Une modélisation probabiliste où le modèle produit une distribution de probabilités sur les 150 classes est donc intuitif. L'objectif de l'entraînement est d'aligner au maximum cette distribution prédite avec la distribution réelle.

Le choix de la fonction de perte pour la classification découle du principe de maximum de vraisemblance (*Maximum Likelihood Estimation*, *MLE*) (Pan et Fang, 2002). Ce principe statistique stipule que les paramètres optimaux du modèle sont ceux qui maximisent la probabilité d'observer les données d'entraînement.

La perte d'entropie croisée mesure donc la divergence entre la distribution de probabilités cible et la distribution prédite (la sortie de la fonction *softmax* du modèle), et est formalisé comme suit :

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \cdot \log(\hat{y}_{i,c}) \tag{3.4}$$

Où :

- N est le nombre d'exemples d'entraînement,
- C est le nombre de classes,
- $y_{i,c} \in \{0, 1\}$ est l'indicateur binaire valant 1 si l'exemple i appartient à la classe c, 0 sinon,
- $\hat{y}_{i,c} \in [0, 1]$ est la probabilité prédite que l'exemple i appartienne à la classe c.

La perte croît de manière asymétrique et non-linéaire, et pénalise fortement les prédictions confiantes mais incorrectes tout en récompensant modérément les prédictions correctes. Cette propriété est fondamentale pour l'apprentissage car elle force le modèle à éviter les erreurs graves plutôt qu'à simplement maximiser les bonnes prédictions.

3.2.4 Optimisation

L'optimisation d'un réseau de neurones profond constitue un défi majeur en apprentissage automatique. Nous avons mis en œuvre certaines techniques d'optimisation lors de l'entraînement du modèle YOLO11m-cls adaptées à notre contexte de classification des images de feuilles d'espèces d'arbres. Notamment le choix de l'algorithme d'optimisation, la configuration des hyperparamètres, les stratégies de régularisation et les mécanismes de contrôle de l'entraînement.

3.2.4.1 L'algorithme d'optimisation

L'entraînement d'un réseau de neurones consiste à minimiser la fonction de perte $L(\theta)$ par rapport aux paramètres θ du modèle. Cette opération s'effectue par descente de gradient stochastique, où les paramètres sont itérativement mis à jour dans la direction opposée au gradient. Il existe plusieurs algorithmes

d'optimisation pour améliorer la convergence, *SGD* (Bottou, 2012), *SGD + Momentum* (Liu *et al.*, 2020), *Adam* (Kingma, 2014), *AdamW* (Loshchilov et Hutter, 2017) ou encore *LAMB* (You *et al.*, 2019).
Nous avons choisi *AdamW* (*Adam with decoupled Weight decay*) comme algorithme d'optimisation pour notre modèle. En effet, l'étude de Loshchilov et Hutter (2017) démontre qu'*AdamW* surpasse systématiquement les autres algorithmes d'optimisation pour les tâches de transfert d'apprentissage. La correction de la décroissance des poids (*weight decay*, détaillé plus bas) est particulièrement importante lorsque l'on affine des poids pré-entraînés. Contrairement à l'algorithme d'optimisation par descente de gradient stochastique SGD (*Stochastic Gradient Descent*) par exemple, qui nécessite un réglage précis du taux d'apprentissage, *AdamW* tolère une plage plus large de valeurs initiales grâce à son mécanisme d'adaptation par paramètre. Également, l'estimation des *momentum* de premier ordre (la moyenne mobile des gradients indiquant la direction de descente) et de second ordre (la moyenne mobile des carrés des gradients mesurant leur variabilité) permet une convergence plus rapide, réduisant ainsi le nombre d'époques nécessaires, ce qui est un avantage significatif pour l'expérimentation (Loshchilov et Hutter, 2017).

Plus formellement, le fonctionnement de l'algorithme *AdamW* (Loshchilov et Hutter, 2017; Hao *et al.*, 2024) est décrit par la séquence d'opérations suivante, exécutée à chaque itération d'entraînement t :

1. Le calcul du gradient :

$$g_t = \nabla_{\theta_t} f(\theta_t) \tag{3.5}$$

2. La mise à jour de l'estimation du premier *momentum* :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.6}$$

3. La mise à jour de l'estimation du second *momentum* :

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(g_t \odot g_t) \tag{3.7}$$

4. La correction du biais :

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{3.8}$$

5. La mise à jour des paramètres avec décroissance des poids découplée (*decoupled weight decay*) :

$$\theta_{t+1} = \theta_t - \alpha \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda \theta_t \right) \tag{3.9}$$

Où :

- θ_t sont les paramètres du modèle à l'itération t,
- $f(\theta_t)$ est la fonction de perte,
- g_t est le gradient actuel,
- m_t est le *momentum* de premier ordre (la direction moyenne),
- v_t est le *momentum* de second ordre (la magnitude des gradients),
- $\hat{m}_t$ et $\hat{v}_t$ sont les *momentum* corrigés du biais,
- α est le taux d'apprentissage,
- β_1 est le taux de décroissance du premier *momentum*,
- β_2 est le taux de décroissance du second *momentum*,
- λ est le coefficient de décroissance des poids,
- ϵ est la constante de stabilité numérique pour éviter la division par zéro,
- $\odot$ est la multiplication élément par élément (le produit de Hadamard (1899)).

3.2.4.2 La configuration des hyperparamètres d'entraînement

Les hyperparamètres constituent les variables de configuration qui ne sont pas apprises par le modèle mais doivent être fixées avant l'entraînement. Leur choix influence significativement la convergence, la performance et la capacité de généralisation du modèle.

Stratégie de gestion du taux d'apprentissage.

Le taux d'apprentissage est l'hyperparamètre le plus critique de l'optimisation. Il contrôle l'amplitude des mises à jour des poids à chaque itération. Un taux trop élevé provoque des oscillations et peut empêcher la convergence, tandis qu'un taux trop faible ralentit excessivement l'entraînement et risque de piéger le modèle dans des minima locaux non optimaux.

Pour l'ajustement fin de notre modèle, nous utilisons un taux d'apprentissage initial de $\eta_0 = 0.0001$, soit 10 fois inférieur à la valeur typique pour un entraînement à partir de zéro (0.001). Cette réduction est motivée par le fait que nous souhaitons préserver les caractéristiques pré-apprises : des mises à jour trop agressives risqueraient de dégrader ces connaissances avant que le modèle n'apprenne à les exploiter pour notre tâche. De plus, un taux d'apprentissage plus faible favorise une convergence plus stable et meilleure la généralisation (Li *et al.*, 2020a).

Plutôt qu'un taux d'apprentissage constant ou décroissant par paliers, nous utilisons une planification par décroissance cosinus (*schedule cosine annealing*) qui fait décroître le taux d'apprentissage selon une courbe

cosinus :

$$\eta(t) = \eta_{\min} + \frac{1}{2}\left(\eta_{\max} - \eta_{\min}\right)\left(1 + \cos\left(\pi \frac{t}{T}\right)\right) \tag{3.10}$$

Où :

- t est l'indice de l'époque,
- $\eta(t)$ est le taux d'apprentissage à l'itération t,
- $\eta_{\max} = 0.0001$ est le taux d'apprentissage initial,
- $\eta_{\min} = 0.00001$ est le taux d'apprentissage final,
- T est la durée totale du cycle de décroissance, lorsque $t = T$ on arrive à la fin de la planification.

Lors de la phase d'exploration (en début d'entraînement), le modèle d'explore largement l'espace des solutions. Les grands pas permettent également d'échapper aux minima locaux peu profonds.
Puis lors de phase de transition (en milieu d'entraînement), la décroissance progressive permet une adaptation continue. Contrairement aux planifications par paliers (*schedules step decay*), il n'y a pas de discontinuité qui pourrait perturber la dynamique d'optimisation.
Enfin, lors de la phase d'affinage (en fin d'entraînement), le modèle affine précisément ses poids autour du minimum trouvé. Les petits pas permettent une convergence fine sans osciller autour de la solution.

Taille de lot et estimation du gradient.

La taille de lot (*batch size*) détermine le nombre d'échantillons utilisés pour estimer le gradient à chaque itération. Ce paramètre impacte la qualité de l'estimation du gradient. Un lot *batch* plus grand fournit une estimation plus stable et moins bruitée du gradient réel, car il moyenne sur davantage d'exemples. Tandis qu'un lot *batch* plus petit introduit du bruit dans les mises à jour, ce qui peut avoir un effet de régularisation bénéfique en empêchant la convergence vers des minimas trop spécialisés (moins généralisables) (Keskar *et al.*, 2016).
La taille de lot choisie est aussi liée aux ressources matérielles disponibles. En effet, la mémoire GPU requise croît linéairement avec la taille de lot. Et en ce qui concerne la vitesse d'entraînement, les GPUs modernes parallélisent efficacement le traitement des lots *batchs*, donc un lot *batch* plus grand peut être plus rapide par échantillon (Keskar *et al.*, 2016).
Pour notre projet nous avons retenu une taille de lot de 32 échantillons, qui représente un équilibre optimal entre mémoire GPU, stabilité du gradient et risque de surapprentissage.
Avec un lot *batch* de 32 et des images de 320×320 pixels, l'utilisation mémoire est d'environ 9 GB. Cette

taille permet également une représentation statistiquement significative des différentes classes à chaque itération, ce qui est particulièrement important pour un problème à 150 classes.

3.2.4.3 Stratégies de régularisation

La régularisation vise à prévenir le surapprentissage, le phénomène où le modèle mémorise les données d'entraînement au détriment de sa capacité à généraliser sur des données jamais vues. Nous avons combiné plusieurs techniques de régularisation :

Décroissance des poids.
La décroissance des poids (*weight decay*) est une technique de régularisation qui pénalise les poids de grande magnitude en ajoutant un terme de pénalité à la fonction de perte, encourageant ainsi le modèle à maintenir des poids de faible magnitude. Avec l'optimiseur que nous utilisons, AdamW, cette pénalité est appliquée directement aux poids (Loshchilov et Hutter, 2017), selon la formule suivante :

$$\theta_{t+1} = \theta_t \left(1 - \eta\lambda\right) - \eta\nabla L(\theta_t) \tag{3.11}$$

Où :

- $\lambda = 0.0005$ est notre coefficient de décroissance des poids.
- θ_t est le vecteur des poids du réseau à l'itération t,
- θ_{t+1} est le vecteur des poids du réseau aprés l'itération t,
- η est le taux d'apprentissage,
- $\nabla L(\theta_t)$ est le gradient de la perte par rapport aux paramètres, évalué en θ_t.

Cette valeur empêche d'une part les poids d'atteindre des valeurs extrêmes qui signaleraient un surapprentissage et d'autre part de préserver suffisamment de capacité pour discriminer 150 classes aux caractéristiques subtiles.

Arrêt anticipé.
L'arrêt anticipé (*early stopping*) constitue une forme de régularisation implicite basée sur le principe que le modèle commence par apprendre des patterns généralisables avant de mémoriser les spécificités de l'ensemble de données d'entraînement. Il consiste donc à arrêter l'entraînement au moment où la performance de validation cesse de s'améliorer, et de capturer ce point optimal.

Notre paramètre de patience est de 15 époques, cela signifie que l'entraînement continue pendant 15 époques après la dernière amélioration avant de conclure au surapprentissage. Ce seuil est suffisamment élevé pour tolérer les plateaux temporaires, fréquents avec la décroissance cosinus *cosine annealing*, tout en détectant efficacement le surapprentissage.
Lorsque l'arrêt anticipé est déclenché, on restaure automatiquement les poids correspondant à la meilleure performance observée sur la validation, plutôt que les poids finaux potentiellement sur-ajustés.

Augmentation de données.
L'augmentation de données (détaillée dans la section 3.2.3) constitue également une puissante technique de régularisation en augmentant artificiellement la diversité des exemples d'entraînement. Les transformations géométriques et colorimétriques appliquées, empêchent le modèle de mémoriser des caractéristiques non généralisables, comme une position exacte ou un éclairage spécifique.

Régularisation implicite par la taille de lot.
Avec une taille de lot de 32 (plutôt que 128 ou 256) le gradient calculé à chaque itération est plus approximatif, donc plus bruité. Cette variabilité agit comme une forme de régularisation naturelle, qui peut aider le modèle échapper aux minima pointus, favorisant sa convergence vers des minima plats associés à une meilleure généralisation (Keskar *et al.*, 2016).

3.2.4.4 *Momentum* et accélération de la convergence

Le *momentum* constitue une technique d'accélération qui accumule une inertie dans la direction des gradients passés, permettant une convergence plus rapide et plus stable.
De maniére générale, à chaque itération, plutôt que de suivre uniquement le gradient courant, le *momentum* maintient une moyenne mobile exponentielle des gradients passés :

$$v_t = \beta\, v_{t-1} + g_t \tag{3.12}$$

Où :

- v_t est le *momentum* à l'itération t,
- $\beta = 0.937$ est le coefficient de *momentum*,
- g_t est le gradient courant.

La mise à jour des paramètres utilise alors v_t plutôt que g_t directement. La valeur du *momentum* $\beta = 0.937$

(valeur par défaut) offre un bon équilibre entre une accélération significative dans les directions où le gradient est consistant (même signe sur plusieurs itérations), car la vélocité s'accumule et accélère la progression, et l'amortissement des oscillations dans les directions où le gradient varie, car les contributions positives et négatives se compensent partiellement, ce qui réduit les oscillations.
Avec cette valeur, la mémoire est modérée et l'influence d'un gradient passé décroît de moitié après environ 10 itérations ($0.937^{10} \approx 0.52$), offrant une mémoire suffisante sans introduire un retard excessif.
Durant les 3 époques de préchauffage, le *momentum* est réduit à $\beta = 0.7$ car les gradients initiaux, issus de la nouvelle tête de classification, sont bruités et potentiellement trompeurs. Un *momentum* élevé accumulerait et amplifierait ce bruit, ce qui peut conduire à des mises à jour trop agressives et dégrader les représentations du réseau dorsal pré-entraîné.

3.3 Évaluation

3.3.1 Méthodologie d'évaluation

L'évaluation d'un modèle d'apprentissage automatique est une étape fondamentale du processus de développement. Elle permet d'une part de quantifier les performances du modèle, et d'autre part d'identifier faiblesses afin d'orienter les améliorations futures.

L'évaluation d'un classifieur présente des défis spécifiques liés à la nature du problème, comme dans notre projet le grand nombre de classes, la similarité morphologique entre certaines espèces proches et la variabilité des feuilles pour une même espèces (âge de la feuille, conditions environnementales, saison). La méthodologie d'évaluation que nous avons mise en place fournit des métriques interprétables permettant d'agir sur l'entraînement du modèle pour en améliorer les performances.

Partitionnement de l'ensemble de données.
Le partitionnement de l'ensemble de données en sous-ensembles distincts est essentiel pour obtenir une estimation non biaisée des performances de généralisation du modèle. Nous avons adopté une division en trois sous-ensembles :

— L'ensemble d'entraînement, qui représente 60 % de l'ensemble de données total. Il est utilisé pour l'entraînement du modèle. L'augmentation de données (décrite à la section 3.2.3) est appliquée uniquement ici.

— L'ensemble de validation, qui représente 20 % de l'ensemble de données total. Utilisé pour le monitoring durant l'entraînement, la sélection des hyperparamètres, et le mécanisme d'arrêt anticipé. Ce sont précisément les métriques sur cet ensemble qui permettent de prendre les décisions d'optimisation.
— L'ensemble de test, qui représente les 20 % restant de l'ensemble de données total. Il est réservé exclusivement pour l'évaluation finale. Il n'est jamais utilisé durant le processus d'entraînement. Il fournit donc une estimation non biaisée de la performance de généralisation du modèle.

Le partitionnement des données est effectué de manière stratifiée, permettant ainsi de garantir que chaque espèce est représentée dans les mêmes proportions dans les trois ensembles. Cette stratification est très importante dans notre projet car certaines espèces sont sous-représentées. Sans stratification, avec un tirage aléatoire simple, certaines classes pourrait être complètement exclues de l'ensemble de validation ou de test.

Concrètement, pour chaque classe $c \in \{1, \ldots, 150\}$, si N_c représente le nombre total d'images de cette classe, alors :

— l'ensemble d'entraînement contient $\lfloor 0.6 \times N_c \rfloor$ images de la classe c,
— l'ensemble de validation contient $\lfloor 0.2 \times N_c \rfloor$ images de la classe c,
— l'ensemble de test contient les images restantes de la classe c.

Aussi les sous-ensembles obtenus doivent être strictement séparés tout au long du processus de d'entraînement afin d'éviter les fuites de données (*data leakage*) (Bernett *et al.*, 2024), qui surviennent lorsque des informations de l'ensemble de test fuitent vers le processus d'entraînement, conduisant à une surestimation des performances réelles.

Dans notre contexte, le *data leakage* pourrait survenir si plusieurs photographies d'une même feuille se retrouvaient dans les ensembles d'entraînement et de test. Le modèle pourrait alors reconnaître cette feuille spécifique plutôt que d'apprendre les caractéristiques générales de l'espèce (Bernett *et al.*, 2024). Pour réduire ce risque, nous avons évité quand autant que possible de récupérer des images d'une même espèce provenant d'un même contributeur lors de la session de collecte de données. Nous avons également procédé à une vérification automatisée des images, afin de vérifier qu'il n'y a pas de doublon dans notre ensemble de données. Toutefois, il convient de préciser que ces mesures réduisent le risque sans l'élimi-

ner totalement. Des photographies distinctes d'une même feuille physique pourraient subsister dans le jeu de données sans être détectables. Néanmoins, compte tenu de la diversité des sources de collecte et du nombre élevé d'images par espèce, la proportion de tels cas est faible.

L'évaluation d'un modèle de classification requiert un ensemble de métriques, permettant de capturer des aspects différents de la performance du modèle. Pour notre projet nous avons utilisé les métriques suivantes pour l'évaluation :

Perte de validation.
La perte (*loss*) de validation correspond à la valeur de l'entropie croisée (*cross-entropy*) (détaillée à la section 3.3.3) calculée sur l'ensemble de validation :

$$L_{\text{val}} = -\frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} \log\left(p_{y_i}^{(i)}\right) \tag{3.13}$$

La perte est une valeur réelle (continue) et permet de mesurer la qualité des probabilités prédites. Une perte plus basse signifie que le modèle produit des prédictions plus confiantes et mieux calibrées.

La comparaison entre la perte d'entraînement et la perte de validation au cours des époques permet de détecter le surapprentissage :

- Si la perte d'entraînement baisse et la perte de validation baisse, alors le modèle s'améliore sur l'entraînement et sur des données nouvelles. Ça veut dire qu'il apprend des motifs qui se généralisent, l'apprentissage est donc sain.
- Si la perte d'entraînement baisse et la perte de validation stagne, alors le modèle continue à être plus efficace sur les données d'entraînement, mais n'améliore plus ses performances sur les données nouvelles données (de validation). Cela peut signifier qu'il approche de sa limite de ce qu'il peut apprendre avec cette configuration des hyperparamètres, on peut éventuellement conclure que l'apprentissage est terminé.
- Si la perte d'entraînement baisse et la perte de validation augmente, alors le modèle devient meilleur sur les données d'entraînement mais pire sur les données de validation. Il apprend des détails spécifiques (du bruit propre au données d'entraînement). Le modèle mémorise au lieu d'apprendre des règles générales, il y a surapprentissage.

Matrice de confusion.

La matrice de confusion est une représentation tabulaire de dimension 150×150 où l'élément $M[i, j]$ représente le nombre d'échantillons de la classe réelle i qui ont été prédits comme appartenant à la classe j. La diagonale contient donc les prédictions correctes, tandis que tous les éléments en dehors de cette diagonale représentent les erreurs de classification. Dans notre cas la matrice complète est difficile à visualiser, à cause du nombre élevé de classes. Pour pallier ça, nous avons utilisés plusieurs stratégies d'analyse :

- La matrice normalisée où chaque ligne est divisée par le nombre total d'échantillons de la classe correspondante, pour nous permettre de comparer les taux d'erreur entre classes de tailles différentes.
- L'extraction des paires confondues, permettant d'identifier des couples $(classe_i, classe_j)$ avec $M[i, j]$ élevé. Ce qui est particulièrement intéressant dans notre cas pour déceler les espèces fréquemment confondues.
- L'agrégation taxonomique, qui consiste à regrouper les espèces par genre pour identifier si les confusions du modèle se font intra-genre ou inter-genre.

L'analyse de la matrice de confusion nous a permis d'orienter les améliorations de notre système. Notamment la collecte de données supplémentaires pour les espèces fréquemment confondues.

À partir de la matrice de confusion nous avons également calculé trois métriques pour chaque classe individuellement :

1. **La précision**, qui représente la proportion des prédictions de classe c qui sont effectivement correctes :

$$\text{Précision}_c = \frac{VP_c}{VP_c + FP_c} = \frac{M[c, c]}{\sum_i M[i, c]} \tag{3.14}$$

Où :

- VP_c (Vrais Positifs) est le nombre d'échantillons dont la vraie classe est c et prédits comme c,
- FP_c (Faux Positifs) est le nombre d'échantillons prédits comme c alors qu'ils appartiennent en réalité à une autre classe,
- M est la matrice de confusion de taille $K \times K$, où K est le nombre de classes,
- $M[c, c]$ est le nombre d'échantillons correctement classés pour la classe c, donc $M[c, c] = VP_c$,
- $\sum_i M[i, c]$ est le nombre total d'échantillons prédits comme c, donc $\sum_i M[i, c] = VP_c + FP_c$.

2. **Le rappel**, mesure pour une classe c, à quel point le modèle retrouve tous les vrais exemples de cette classe. Donc parmi tous les échantillons qui sont réellement de la classe c, quelle proportion le modèle a correctement prédits :

$$\text{Rappel}_c = \frac{VP_c}{VP_c + FN_c} = \frac{M[c,c]}{\sum_j M[c,j]} \tag{3.15}$$

Où :

- VP_c (Vrais Positifs) est le nombre d'échantillons dont la vraie classe est c et prédits comme c,
- FN_c (Faux Négatifs) est le nombre d'échantillons dont la vraie classe est c mais prédits comme une autre classe,
- M est la matrice de confusion de taille $K \times K$, où K est le nombre classes,
- $M[c,c]$ est le nombre d'échantillons correctement classés pour la classe c, donc $M[c,c] = VP_c$,
- $\sum_j M[c,j]$ est le nombre total d'échantillons qui sont réellement de la classe c, donc $\sum_j M[c,j] = VP_c + FN_c$.

3. **Le score F1**, représente quant à lui la moyenne harmonique de la précision et du rappel pour une classe c. Si l'une des deux métrique (la précision ou le rappel) est faible, le score F1 baisse fortement. Il pénalise donc les modèles déséquilibrés. Cette métrique est particulièrement pertinente pour notre projet car il donne une mesure plus réaliste quand les erreurs ne sont pas réparties uniformément. Le modèle peut être très bon globalement tout en étant mauvais sur certaines espèces proches :

$$\text{F1}_c = \frac{2\,\text{Précision}_c \times \text{Rappel}_c}{\text{Précision}_c + \text{Rappel}_c} \tag{3.16}$$

Ces métriques par classe révèlent les disparités de performance du modèle entre les différentes espèces.

Macro score F1.

Le macro score F1 est une métrique d'évaluation très utilisée en classification, particulièrement adaptée aux problèmes multi-classes sur des ensembles de données déséquilibrés. C'est la moyenne simple des scores F1 de chaque classe, calculée comme suit :

$$\text{Macro F1} = \frac{1}{N}\sum_{i=1}^{N} F1_i = \frac{1}{N}\sum_{i=1}^{N} \frac{2 \times P_i \times R_i}{P_i + R_i} \tag{3.17}$$

Où :

- N est le nombre total de classes,
- i est l'indice de la classe,
- $F1_i$ est le score F1 de la classe i,
- P_i est la précision de la classe i,
- R_i est le rappel de la classe i.

Ce choix est crucial dans notre contexte, car notre ensemble de données est déséquilibré avec certaines espèces surreprésentées. Comme le soulignent He et Garcia (2009) dans leur étude fondatrice sur l'apprentissage à partir de données déséquilibrées, qu'il est nécessaire d'utiliser des métriques qui prennent en compte équitablement les classes minoritaires. En effet, un classifieur peut atteindre des performances apparemment élevées en prédisant statistiquement plus fréquemment les classes majoritaires, tout en échouant complètement sur les classes minoritaires.

Les travaux de Sokolova et Lapalme (2009) montrent aussi que le macro-moyennage (une façon de calculer une métrique globale en moyennant sur les classes) est préférable lorsque l'objectif est d'évaluer la capacité du système à reconnaître l'ensemble des classes sans biais envers les plus représentées. Dans le cadre de notre étude de reconnaissance d'espèces d'arbres, il est important d'identifier correctement toutes les espèces, surtout qu'une espèce sous représentée dans notre ensemble de données n'est pas forcément une espèce peu abondante dans la forêt urbaine.

L'évaluation des performances du modèles se fait de deux temps :

Une évaluation périodique.
Durant l'entraînement, l'ensemble de validation est évalué à la fin de chaque époque. Cette évaluation périodique permet d'une part de monitorer la convergence en observant l'évolution des métriques, permettant ainsi de vérifier que l'entraînement progresse normalement et de déclencher l'arrêt anticipé en cas d'absence d'amélioration prolongée ; et d'autre part de détecter le surapprentissage en remarquant une différence notable entre les performances du modèle sur les données d'entraînement par rapport à celles sur les données de validation.
Le meilleur modèle est sélectionné en sauvegardant les poids correspondant à la meilleure performance sur les données de validation.

Une évaluation finale sur l'ensemble de test.

L'évaluation finale sur les données de test constitue la dernière étape du processus d'évaluation. Elle est effectuée une unique fois, après la sélection définitive du modèle (issu de l'évaluation périodique), pour obtenir une estimation non biaisée de la performance de généralisation du modèle.

Le principe fondamental de l'évaluation sur les données de test est qu'elle ne doit jamais influencer les décisions d'optimisation du modèle. En effet, si l'on évaluait répétitivement sur les données de test en ajustant les hyperparamètres en fonction des résultats, l'ensemble de test deviendrait un second ensemble de validation, et l'estimation des performances du modèle serait alors biaisée. Donc toutes les décisions d'architecture du modèle et de réglage des hyperparamètres ne sont prises que sur la base des performances du modèle sur les données de validation.

L'évaluation sur les données de test sert uniquement à avoir une vision non biaisée sur les performances finales du modèle.

3.3.2 Gestion de l'incertitude

Tout modèle de classification, aussi performant soit-il, produit inéluctablement des prédictions incertaines ou erronées. La gestion de cette incertitude est essentielle pour une utilisation responsable et fiable du modèle en conditions réelles. Nous avons donc mis en place des mécanismes de quantification de l'incertitude, ainsi que des stratégies de présentation des résultats aux personnes utilisatrices qui permettent de gérer cette incertitude.

Les probabilités obtenues via la fonction *softmax* est une mesure de confiance. La couche finale de notre modèle produit une distribution de probabilités $p = (p_1, p_2, \ldots, p_{150})$ sur les 150 classes. La probabilité maximale $p_{\max} = \max_c(p_c)$ constitue un indicateur de la confiance du modèle dans sa prédiction :

- Si $p_{\max}$ proche de 1, alors le modèle est très confiant dans sa prédiction. Une classe domine nettement les autres.
- Si $p_{\max}$ est compris entre 0.5 et 0.8, alors confiance est intermédiaire. Le modèle favorise une classe mais avec une certaine hésitation.
- Enfin, si $p_{\max}$ faible soit < 0.5, le modèle est incertain. La distribution est plate, plusieurs classes ont des probabilités assez proches.

Une mesure complémentaire de l'incertitude est l'entropie de la distribution prédite, décrite par la formule

suivante :

$$H(p) = -\sum_{c=1}^{150} p_c \times \log_2(p_c) \tag{3.18}$$

Elle quantifie le désordre de la distribution. Une entropie minimale, avec H proche de 0, indique une distribution orientée principalement sur une classe, l'incertitude est donc minimale. Alors qu'une entropie maximale, avec $H = \log_2(150)$, signifie que la distribution est uniforme sur les 150 classes, l'incertitude est maximale. L'entropie capture des nuances que la probabilité $p_{\max}$ ne révèle pas.

Le module d'attention spatiale C2PSA (décrit à la section 3.3.2) contribue lui aussi indirectement à la qualité de l'estimation de confiance. En effet, en focalisant l'extraction de caractéristiques sur les régions les plus importantes de l'image, il permet au modèle de produire des représentations plus discriminantes. Ces dernières conduisent à des distributions *softmax* mieux séparées, une haute probabilité pour la bonne classe, et de faibles probabilités pour les autres (Ouyang *et al.*, 2023).

À noter que la confiance *softmax* ne constitue pas une probabilité calibrée au sens strict, c'est-à-dire que la valeur sortie de la fonction *softmax* ne correspond pas forcément à une probabilité réelle, que le modèle a "raison". Les réseaux de neurones profonds présentent systématiquement une tendance à la sur-confiance (*overconfidence*), ils produisent des probabilités élevées même pour des prédictions incorrectes.

Pour avoir une calibration précise des probabilités nous avons appliqué une mise à l'échelle par température. La mise à l'échelle par température (*temperature scaling*) est la technique de calibration souvent la plus efficace (Balanya *et al.*, 2024). Elle consiste à diviser les logits par un paramètre de température $T > 0$ avant d'appliquer la fonction *softmax*, comme décrit dans la formule suivante :

$$p_c^{calibre} = \frac{\exp(z_c/T)}{\sum_j \exp(z_j/T)} \tag{3.19}$$

- Si $T > 1$, cela "adoucit" la distribution, réduisant la confiance et donc corrige la sur-confiance.
- Si $T < 1$, cela "durcit" la distribution, augmentant la confiance et donc corrige la sous-confiance.
- Si $T = 1$, alors aucune modification est effectuée, la sortie de la fonction *softmax* reste standard.

Le paramètre T est optimisé sur l'ensemble de validation en minimisant la log-vraisemblance négative (*negative log-likelihood*).

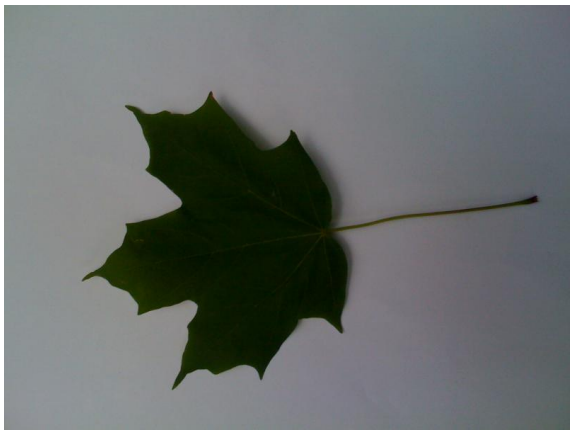
Dans notre contexte d'identification d'espèces d'arbre à partir de photos de feuilles, plutôt que de fournir une unique prédiction, la solution présente les K espèces les plus probables (*Top-K*) accompagnées de leurs probabilités respectives.
Avec l'approche *Top-K* l'aide à la décision est enrichie. La personne utilisatrice peut trancher entre les espèces proposées. Elle permet également de réduire le taux d'erreur perçu, même si la première prédiction (*Top-1*) est incorrecte, la bonne espèce figure souvent dans le *Top-5*. Enfin, elle offre une indication implicite de l'incertitude, la distribution des probabilités dans le *Top-K* informe visuellement la personne utilisatrice sur la confiance du modèle.

Le choix de $K = 5$ résulte d'un compromis et est largement documenté dans la littérature (Russakovsky *et al.*, 2015). Si K est trop petit, il y a un risque non négligeable de ne pas inclure la bonne espèce en cas de confusion. Au contraire, si K est trop grand, il y a un risque de diluer l'information utile. $K = 5$ est un standard couramment utilisé, et offre un bon équilibre entre information et lisibilité.

Sources d'incertitude identifiées.
L'analyse des erreurs de classification du modèle nous a permis de révéler plusieurs sources d'incertitude. La compréhension de ces sources permet d'interpréter les erreurs du modèle et de mettre en places des stratégies d'amélioration.

La première source d'incertitude est la similarité inter-espèces (figure 3.7). De nombreuses espèces ont des feuilles morphologiquement très proches. Cette similarité qui reflète leur parenté phylogénétique, constitue une source majeure de confusion pour le modèle. Cela dit, le module C2PSA de notre modèle aide partiellement à discriminer ces espèces en focalisant l'attention sur des détails subtils, mais certaines confusions restent irréductibles.

(a) Image de feuille d'érable de Norvège.



(b) Image de feuille d'érable à sucre.

Figure 3.7 – Exemple de similarité inter-espèce.

La seconde source d'incertitude est la variabilité intra-espèce. Au sein d'une même espèce, la morphologie des feuilles peut varier considérablement selon plusieurs paramètres (Chitwood et Sinha, 2016) :

- L'âge de la feuille : Les jeunes feuilles peuvent être significativement différentes que des feuilles matures en termes de taille, de forme et de coloration.
- La position sur l'arbre : Les feuilles intérieures plus à l'ombre et les feuilles périphérique plus exposées à la lumière présentent souvent des différences morphologiques.
- Les conditions environnementales : Le stress hydrique, le sol, et plus généralement le climat influencent la taille et la forme des feuilles.
- Les saisons : Les colorations printanières, estivales et automnales sont très variables, particulièrement dans la région de Montréal (notre zone d'étude) où les saisons sont très marquées.
- L'hétérophyllie : Certaines espèces d'arbres produisent naturellement des feuilles très différentes sur le même individu (Chitwood et Sinha, 2016).

L'augmentation de données aide le modèle à acquérir une certaine robustesse à cette variabilité, mais un ensemble de données représentatif de cette diversité intra-spécifique reste essentiel.

La troisième source d'incertitude que nous avons identifiée est la qualité de l'image d'entrée.
Les personnes utilisatrices peuvent prendre des photos de mauvaises qualités pouvant perturber les pré-

dictions du modèle. Plusieurs éléments entrent en jeux, comme les conditions d'éclairage, le flou, des occlusions ou encore un arrière-plan complexe.
L'augmentation de données (décrite à la section 3.2.3) permet d'améliorer la robustesse du modèle face à cette problématique. Toutefois, une image de très mauvaise qualité restera toujours difficile, voire impossible, à classifier correctement.
Également pour atténuer ce risque nous avons décidé d'afficher à la personne utilisatrice des règles simple de prise de vue sur l'interface de l'application :

- Prendre la photo en plein jour.
- Choisir une feuille entière dépourvue d'irrégularités.
- Placer la feuille sur sa main ouverte ou sur une surface uniforme.
- Vérifier que l'objectif de l'appareil n'est pas obstrué.
- Rester immobile lors de la prise de la photo.

La quatrième source d'incertitude est le cas des données hors distribution (*Out-of-Distribution*, OOD).
Cela survient lorsque l'image soumise au modèle représente une photo de feuille d'arbre dont l'espèce n'est pas présente dans l'ensemble de données d'entraînement. Le modèle est alors forcé d'assigner l'image à l'une des 150 classes connues, produisant inéluctablement une prédiction incorrecte.
L'une des solutions, que nous n'avons pas implémenté, serait de mettre en place un mécanisme de détection OOD, qui se baserait sur une probabilité maximale $p_{\max}$ anormalement basse, une distribution quasi-uniforme, ou encore une incohérence dans le *Top-5*, pour déduire que la feuille appartient à une espèce d'arbre inconnue pour le modèle.

Enfin, la gestion de l'incertitude est intégrée directement dans l'interface utilisateur de la solution. Lors de l'affichage des *Top-5* espèces prédites, on affiche systématiquement les probabilités associées. Également un message signale clairement les cas de faible confiance, invitant la personne utilisatrice à effectuer des vérifications. Un mécanisme de *feedback* et de confirmation a aussi été mis en place pour permettre aux personnes utilisatrices de signaler d'éventuelles erreurs. Ces éléments visent à établir une relation de confiance entre les personnes utilisatrices et le système.

CHAPITRE 4
RECOMMANDATION D'ESPÈCES D'ARBRES ET DE GROUPES FONCTIONNELS

Dans ce chapitre nous allons présenter le système de recommandation d'espèces d'arbres en milieu privé que nous avons développé. L'objectif est de proposer aux personnes utilisatrices des espèces d'arbres adaptées à leur contexte spécifique, tout en maximisant la diversité fonctionnelle de la forêt urbaine à l'échelle du quartier et de la ville (St-Denis *et al.*, 2024). La bonne gestion des arbres urbains implique une gestion des plantations individuelles, qui contribuent aussi à la résilience et aux services écosystémiques de l'ensemble du couvert forestier urbain (Goddard *et al.*, 2010; Cameron *et al.*, 2012).

Nous détaillerons dans la partie 4.1 les données utilisées lors du processus. Puis, nous verrons dans la partie 4.2 les critères de choix utilisés pour effectuer ces recommandations. Enfin, finirons par décrire le fonctionnement de l'algorithme de recommandation mis en place.

4.1 Données

La qualité et la richesse des données constituent le fondement de tout système de recommandation performant (Adomavicius et Tuzhilin, 2005). Dans notre projet, les données proviennent de deux sources distinctes, les inventaires municipaux d'arbres publics et les données d'arbres privés issus de la contribution citoyenne. Nous détaillerons donc les différentes sources de données exploitées par notre système de recommandation.

4.1.1 Inventaires d'arbres urbains publics, SylvCiT

Les inventaires d'arbres urbains publics constituent une ressource essentielle pour une planification stratégique à long terme et une gestion plus efficace de la forêt urbaine (Nowak *et al.*, 2008). Ces bases de données, maintenues par les municipalités, documentent les arbres présents sur le domaine public, incluant les arbres de rue et les arbres hors rue (les parcs et places publiques) (Ville de Montréal, 2020).

La ville de Montréal maintient un inventaire géoréférencé de plus de 330 000 arbres publics, accessible via le portail de données ouvertes municipales. Ces inventaires sont disponibles au format CSV (*Comma-separated values*) et sont continuellement mis à jour par les municipalités. Ces données disposent de nom-

breux paramètres, comme par exemple l'identification taxonomique de l'arbre (genre et espèce), le diamètre à hauteur de poitrine (*DHP*) et la localisation géographique (coordonnées *GPS* : latitude et longitude) (Ville de Montréal, 2020).

SylvCiT (St-Denis *et al.*, 2024) intègre déjà les inventaires des arbres publics des villes de Montréal, Saint-Lambert, Repentigny, Boucherville, Varennes, Laval, Rosemère et Mont-Royal. Certains récupérés sur les plateformes (comme pour la ville de Montréal) et d'autre partagés directement par les municipalités partenaires du projet.

Comme notre solution sera directement intégrée à SylvCiT, nous exploitons donc ces données déjà disponibles pour effectuer pour les recommandations. En effet, en connaissant la composition de la forêt urbaine publique environnante, nous pouvons suggérer des espèces qui complémentent le patrimoine arboré existant, maximisant ainsi la diversité fonctionnelle à l'échelle locale et permettant de dépasser la vision traditionnelle centrée sur le site de plantation individuel. Ces ensembles de données étant alimentés manuellement, ils présentent fatalement des erreurs, des incohérences et des anomalies. Notre SR étant sensible à la qualité et au formatage des données d'entrée, un nettoyage rigoureux et un prétraitement de ces données a donc été nécessaire. Le principe établi "*garbage in*, *garbage out*" illustre bien la nécessité d'avoir des données de qualité pour produire des recommandations pertinentes (Halevy *et al.*, 2009).

Nous avons donc appliqué plusieurs méthodes et techniques pour nettoyer, transformer et normaliser les différents inventaires d'arbres publics à notre disposition, détaillées ci-dessous :

La suppression des doublons

Les données d'inventaire d'arbres urbains peuvent contenir plusieurs enregistrements d'un même arbre, lié à des erreurs de saisie notamment lors des mises à jour des inventaires. Pour éliminer ces redondances nous avons effectué deux traitements. D'une part, en recherchant les doublons en se basant sur l'identifiant de l'arbre, même si ces identifiants sont censés être unique dans les bases de données. D'autre part, en mettant en place une détection des doublons qui se base sur un critère de proximité spatiale combiné à une correspondance taxonomique de l'essence de l'arbre. On considère que c'est le même arbre auquel deux enregistrements font référence, si la distance euclidienne entre ces deux arbres est inférieure à 2 mètres et si c'est la même espèce. Nous avons choisi la valeur de deux mètres, car c'est la précision minimale du système de géolocalisation utilisé par la ville de Montréal.

La gestion des erreurs de saisie

Les erreurs de saisie constituent la source principale de bruit dans les données d'inventaire. Pour les données taxonomiques, nous avons mis en place une uniformisation sur la base de la liste de référence des espèces d'arbres que nous avons détaillé dans la section 3.2.1. Les noms des espèces d'arbres ont été comparés aux listes des synonymes que nous avions constitué. Pour les noms d'espèce non reconnus nous utilisons un algorithme de correspondance approximative (*fuzzy matching*) qui calcule la distance de Levenshtein avec chaque élément de notre base de données de référence, c'est à dire une mesure de similarité entre deux chaînes de caractères, qui est définie comme le nombre minimal d'opérations élémentaires nécessaires pour transformer l'une en l'autre. L'espèce ayant la meilleure similarité est retenue, uniquement si cette dernière est supérieure à 80 %. En dessous de ce seuil, on considère que l'incertitude est trop élevée pour avoir que la donnée soit fiable.

Également, les données de géolocalisation avec des incohérences dans les valeurs ont été écartées. Tout d'abord une première vérification est faite sur le format et la valeur de la latitude et de la longitude. Celles qui ne sont ne respecte l'une des notations : degrés décimaux (*DD*), degrés + minutes décimales (*DMM*) ou degrés + minutes + secondes (*DMS*), et celles dont les valeurs ne sont pas dans le bon intervalle, sont supprimées. Par exemple, pour la notation en degrés décimaux, la latitude doit être comprise entre -90.0 et +90.0 et la longitude entre de -180.0 et +180.0. Puis, nous avons vérifié que les coordonnées de géolocalisation des arbres sont cohérentes avec les coordonnées de leur régions respectives, si ce n'est pas le cas, l'enregistrement est supprimé. Les données de localisation des polygones des régions d'où proviennent les inventaires, ont été obtenues à l'aide de l'outil *Geojson.io* (Mapbox, 2025), au format JSON (*JavaScript Object Notation*).

Le traitement des valeurs manquantes

En ce qui concerne les enregistrements avec des valeurs manquantes notre démarche dépend de quelle valeur était manquante. La suppression des enregistrements contenant des valeurs manquantes (l*istwise deletion*) est l'approche la plus simple, mais elle peut entraîner une perte d'information conséquente et introduire des biais. C'est pourquoi nous l'avons utilisé de manière sélective, uniquement pour les variables critiques ne pouvant être estimées de manière fiable. Comme les données qui sont essentielles au fonctionnement de notre algorithme sont le nom de l'espèce de l'arbre, les données de géolocalisation et le diamètre à hauteur de poitrine (*DHP*), seuls les enregistrements ayant des valeurs manquantes dans ces

données ont été écartés. Aucune méthode nous ne permettait d'estimer ces valeurs à partir des autres données disponibles.

L'harmonisation des données

Les structures des données des différents inventaires sont variables, il a donc fallu établir un modèle de données cible afin de les harmoniser, notamment pour les variables essentielles. Ce traitement a permis en particulier d'uniformiser le nom des espèces des arbres en prenant notre base de données de référence comme point modèle. Également nous avons vérifié que toutes les données de géolocalisation étaient bien au format de degrés décimaux (*DD*), le plus couramment utilisé et le plus simple pour les calculs. Pour les inventaires où ce n'est pas le cas une conversion automatique est effectué en parallèle de la vérification.

Enfin, après la récolte et le prétraitement des inventaires d'arbres, nous avons obtenu des données non bruitées garantissant la qualité et la fiabilité de notre système de recommandation. Le détail de la structure de l'ensemble de données est présenté dans le tableau 4.1 suivant :

Désignation	**Description**	**Valeur**
`id`	Identifiant unique de l'arbre dans l'inventaire	Entier
`nom_scientifique`	Nom scientifique de l'espèce	Texte
`genre`	Genre taxonomique de l'arbre	Texte
`essence_latin`	Nom de l'espèce en latin	Texte
`famille`	Famille botanique de l'arbre	Texte
`dhp`	Diamètre à hauteur de poitrine	cm
`latitude`	Coordonnée GPS Nord/Sud en degrés décimaux	[-90.0, +90.0]
`longitude`	Coordonnée GPS Est/Ouest en degrés décimaux	[-180.0, +180.0]
`municipalite`	Municipalité d'origine de l'enregistrement	Texte

Tableau 4.1 – Structure des données des inventaires d'arbres publics.

Bien que l'utilisation des inventaires d'arbres publics pour le système de recommandation constitue un point de départ solide et accessible, elle présente certaines limites lorsqu'il s'agit de produire des recommandations pour le domaine privé. En effet, la distribution spatiale des arbres publics ne reflète pas la diversité des arbres déjà présents dans les espaces privés, ce qui peut conduire à des recommandations qui, localement, réduisent en réalité la diversité fonctionnelle au lieu de l'augmenter. Cependant cette limitation sera

grandement réduite réduite au fur et à mesure avec l'enrichissement de la base de données des arbres en espaces privés.

4.1.2 Base de données des arbres en espaces privés provenant des contributions citoyennes

Les inventaires d'arbres urbains sont essentiellement disponibles pour les arbres du domaine public, causant une lacune importante dans le recensement des arbres sur les espaces privées. Or, les arbres en espaces privés représentent une proportion significative du couvert forestier urbain et jouent un rôle crucial dans la fourniture de services écosystémiques à l'échelle du quartier. La figure 4.1, représentant une image satellite d'un quartier de Montréal provenant de SylvCiT, permet de visualiser ce manque de données sur les arbres en espaces privés (rectangle rouges) par rapport aux arbres publics inventoriés (points de couleurs). Afin de combler cette lacune, les outils participatifs de science citoyenne occupe une place de plus en plus importante. La qualité des données collectées par les personnes citoyennes constitue une préoccupation légitime, des études comparatives entre les données collectées par des experts et celles collectées par des volontaires ont démontré une concordance d'environ 94 % pour l'identification au niveau du genre, et d'environ 80 % pour l'identification au niveau du genre (Roman *et al.*, 2017).

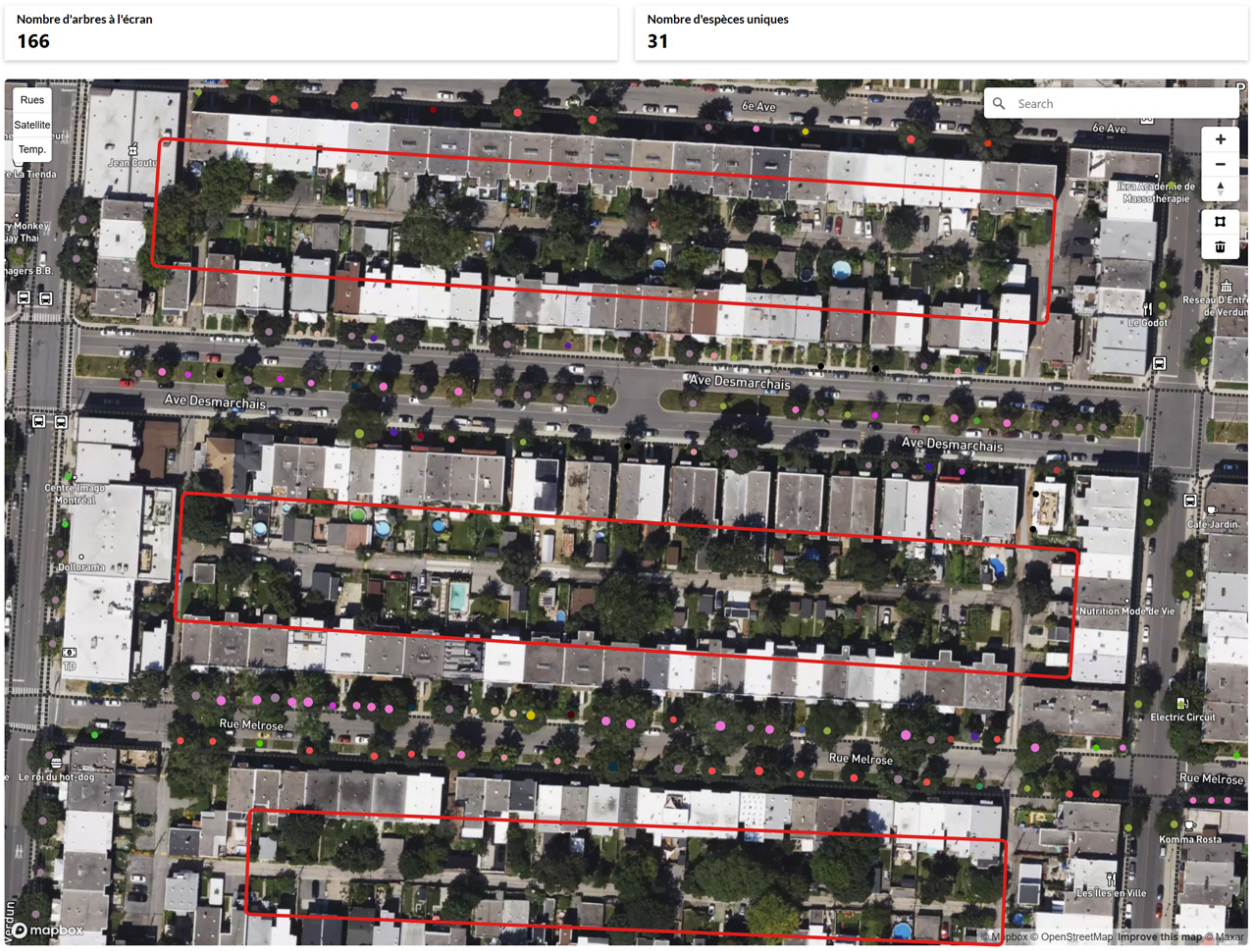


Figure 4.1 – Mise en évidence des arbres en espaces privés non inventoriés (St-Denis *et al.*, 2024).

Nous avons construit notre base de données à partir d'occurrences d'arbres récupérées via GBIF pour la région de Montréal, en utilisant principalement l'ensemble de données *iNaturalist* (*Research-grade Observations*) (iNaturalist contributors, 2025). Ce choix a été motivé par le fait que ces observations sont étiquetées "Qualité recherche", c'est-à-dire qu'elles ont fait l'objet d'une validation communautaire et contiennent des métadonnées utiles.

Avant d'intégrer ces données, nous avons appliqué une série de traitements identiques à ceux décrits dans la section précédente 4.1.1, essentiellement par rapport au nom de l'espèce de l'arbre. La suppression des doublons a été plus stricte que pour les inventaires d'arbres publiques. Afin de distinguer les arbres potentiellement situés en espaces privés de ceux déjà recensés en espaces publics, nous avons comparé ces observations aux inventaires d'arbres publics décrits précédemment, dans une logique de recoupement.

Toute observation située à moins de 2 mètres d'un arbre de l'inventaire public et correspondant à la même espèce d'arbre a été supprimée, car nous avons considéré que les deux enregistrements se rapportent au même arbre. Le seuil de 2 mètres tient compte de l'imprécision des données de géolocalisation sur téléphone.

Notre application mobile intègre aussi une fonctionnalité de contribution citoyenne, permettant aux personnes utilisatrices d'enrichir progressivement la base de données des arbres en espaces privés. En effet, lorsque chaque personne utilisatrice identifie un arbre existant dans son espace privé, via notre module de reconnaissance d'espèce d'arbre ou en renseignant directement ces informations si elles sont connues, il est automatiquement intégré à notre base de données. Les informations collectées à travers l'application mobile sont décrites comme suit :

- Le nom de l'espèce de l'arbre : (Champ obligatoire)
  - Identifié automatiquement par le modèle reconnaissance d'espèces d'arbres avec une photo de feuille.
  - Ou, s'il est connu, renseigné directement par la personne utilisatrice.
- Les données de géolocalisation, récupérées automatiquement soit lorsque la personne utilisatrice prends la photo, soit lors de la validation du formulaire. Un message informe la personne utilisatrice de se tenir le plus près possibles de l'arbre afin d'avoir des données les plus représentatives possible. (Champ obligatoire)
- L'année de plantation de l'arbre. (Champ facultatif)
- Le diamètre à hauteur de poitrine (DHP) de l'arbre. Une description explicative de la marche à suivre pour effectuer la mesure est affichée à la personne utilisatrice. Si la prise de la mesure est trop contraignante, l'application permet aussi d'estimer une classe de DHP. (Champ obligatoire)
- La taille estimée de l'arbre. (Champ facultatif)
- La condition de l'arbre. (Champ facultatif)
- Une photo de l'arbre. (Champ facultatif)

En ce qui concerne la qualité des données ainsi récoltées, le fait que l'identification des arbres présents sur l'espace privé de la personne utilisatrice joue un rôle crucial dans le processus de recommandation de l'espèce du nouvel arbre à planter, incitera la personne utilisatrice à une certaine rigueur lors de l'identification. De plus, les champs du formulaire d'ajout d'arbres sont composés uniquement de listes déroulantes,

qui nous permettent d'avoir des données de qualité. Un mécanisme de déduplication est automatiquement appliqué avant l'ajout d'un nouvel enregistrement. Chaque nouvelle contribution est comparée dans un premier temps aux inventaires d'arbres publics, et dans un second temps à la base de données d'arbre privés. Si une correspondance potentielle est identifiée (même espèce et avec une proximité de 2 mètres ou moins) avec un arbre déjà présent dans la base de données, le système effectue une fusion des enregistrements. Lors de la fusion des enregistrements, les valeurs numériques (l'année de plantation, la taille et le DHP) sont agrégées en calculant la moyenne arithmétique des valeurs disponibles, ce qui permet d'obtenir une estimation représentative. Pour les données de géolocalisation, nous retenons une position unique en calculant le point médian entre les deux coordonnées, en prenant la moyenne des latitudes et la moyenne des longitudes. Cette opération permet d'estimer la position de l'arbre par le centroïde des deux observations.

Pour deux observations O_1 et O_2 ayant respectivement les coordonnées (x_1, y_1) et (x_2, y_2), la position estimée de l'arbre P est calculée comme suit :

$$P = (x_m, y_m) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \tag{4.1}$$

Où :

- x_1 et x_2 sont les latitudes des deux observations,
- y_1 et y_2 sont les longitudes des deux observations,
- x_m est la latitude du point médian,
- y_m est la longitude du point médian.

L'intégration des données privées avec les inventaires d'arbres publiques nous permet de construire une image plus complète de la forêt urbaine. Cette combinaison est essentielle pour notre algorithme de recommandation, car la diversité fonctionnelle doit être évaluée sur l'ensemble des arbres présents.

4.1.3 Consolidation de la base de données de référence des espèces d'arbres

Les traits fonctionnels des arbres sont des caractéristiques biologiques qui influencent leurs réponses aux perturbations et leurs effets sur l'environnement (Garnier et Navas, 2012), notamment la production de services écosystémiques. La diversité fonctionnelle est la variété de ces traits fonctionnels au sein d'une communauté d'arbres, incluant les caractéristiques mesurables à l'échelle de l'individu affectant sa valeur sélective (fitness) et sa survie, tels que la taille des graines, la surface foliaire spécifique et la teneur en azote des feuilles (Violle *et al.*, 2007). Il a été donc important de compléter notre base de données d'espèces

d'arbres (détaillée à la section 3.2.1), afin que notre algorithme puisse intégrer la maximisation de la diversité fonctionnelle dans les recommandations. Pour cela nous nous somme basés sur les travaux de Belluau *et al.* (2021), qui a constitué une base de données de traits fonctionnels des arbres regroupés par groupes fonctionnels. Un groupe fonctionnel est un ensemble d'espèces qui sont regroupées selon la similitude des valeurs des traits fonctionnels. Cette base de données a été mise à jour en 2025. Le schéma ci-dessous (figure 4.2) décrit l'organisation de ces groupes fonctionnels :

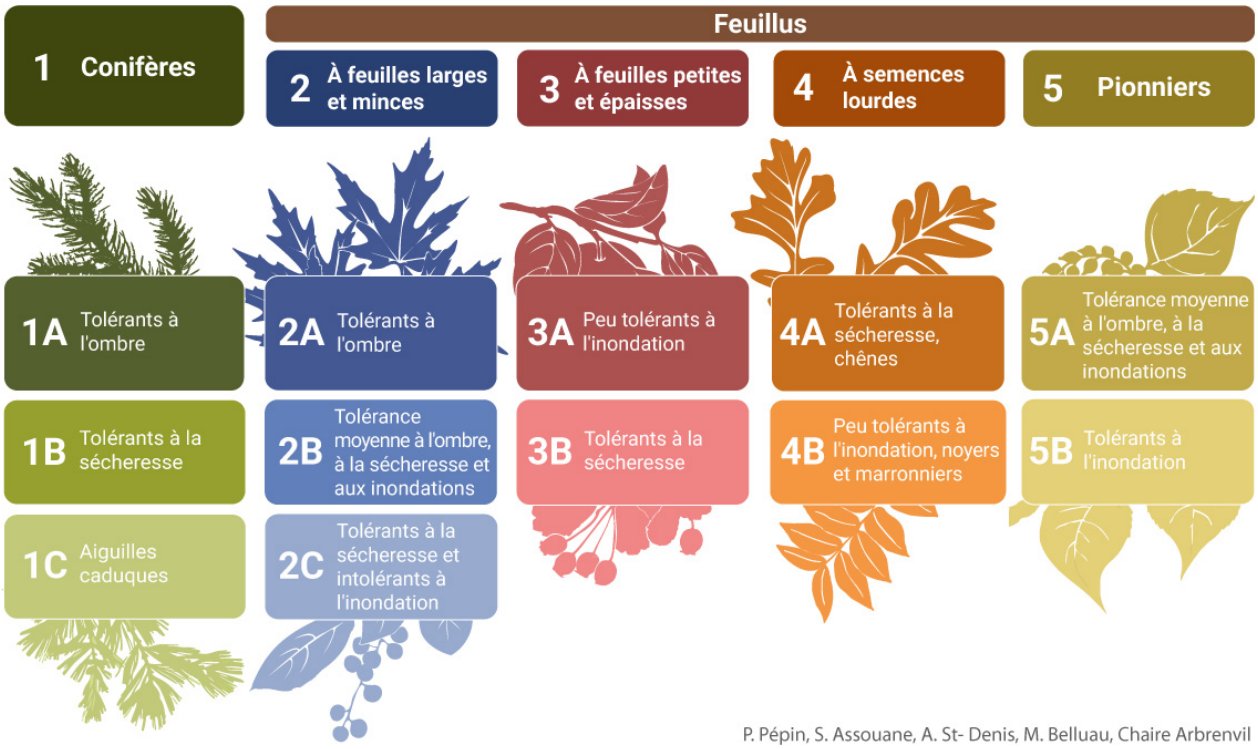


Figure 4.2 – Groupes fonctionnels (Belluau *et al.*, 2021).

Nous avons donc ajouté pour chaque espèce de notre base de données de référence le groupe fonctionnel auquel elle appartient.

Enfin, afin que notre algorithme de recommandation puisse prendre en compte les contraintes du site de plantation et les préférences des personnes utilisatrices, nous avons enrichie notre base de données avec certaines caractéristiques de chaque espèce que nous détaillerons dans la section 4.2. Ces données proviennent également de la base de données de Belluau *et al.* (2021).

Le tableau 4.2 suivant décrit la structure de l'ensemble de données des arbres en en espaces privés :

Déscription	**Description**	**Valeur**	**Statut**
`id`	Identifiant unique de l'arbre dans l'inventaire	Entier	Obligatoire
`nom_scientifique`	Nom scientifique de l'espèce	Texte	Obligatoire
`latitude`	Latitude GPS récupérée automatiquement à proximité de l'arbre.	[-90.0, +90.0]	Obligatoire
`longitude`	Longitude GPS récupérée automatiquement à proximité de l'arbre.	[-180.0, +180.0]	Obligatoire
`dhp`	Diamètre à hauteur de poitrine, mesuré ou estimé par classe	Réel (cm). ou {petit, moyen, grand}	Obligatoire
`annee_plantation`	Année de plantation	Entier (ex : 2015)	Facultatif
`taille`	Taille estimée de l'arbre.	{petit, moyen, grand}	Facultatif
`condition`	État de santé observé de l'arbre.	{excellent, bon, passable, mauvais}	Facultatif
`photo_arbre`	Lien URL de la photo de l'arbre.	Texte	Facultatif

Tableau 4.2 – Structure des données de l'inventaire des arbres privés

4.2 Critères de choix

La sélection d'espèces d'arbres pour les espaces urbains privés constitue un défi majeur impliquant de nombreux critères souvent conflictuels. Les démarches instinctives de sélection se concentre principalement sur des critères esthétiques et culturels, guidés par les préférences des personnes citoyennes, au détriment de considérations écologiques et fonctionnelles. Elles reposent rarement sur une évaluation des contraintes du site plantation et encore moins sur la nécessité de maximiser la résilience de la forêt urbaine.

Notre système de recommandation structure les critères de choix en trois catégories : les contraintes du site de plantation, les préférences des personnes utilisatrices et la diversité fonctionnelle. Cette structuration permet de maintenir une certaine flexibilité nécessaire pour s'adapter aux contextes variés des espaces privés.

Les contraintes du site de plantation

Nous prenons en compte les contraintes du site de plantation à travers les critères suivant :

- La hauteur maximale de l'arbre.
- La taille de la canopée de l'arbre.

Ils représentent les facteurs non négociables qui déterminent la faisabilité de la plantation d'une espèce donnée. L'espace disponible pour la plantation constitue le premier critère discriminant. Les arbres avec une grande hauteur à maturité ne conviennent pas aux petits jardins urbains ou aux cours intérieures notamment lorsqu'il y a des lignes électriques aériennes à proximité.

Les préférences des personnes utilisatrices.

En ce qui concerne les préférences des personnes utilisatrices, elles sont identifiées par les critères suivants :

- Espèce indigène ou ornementale.
- Production de fruits comestibles.
- Type de feuillage (persistant ou caduc).

Les critères de sélection d'arbres à planter varient significativement entre les préférences des personnes expertes et celles des personnes résidentes, ces dernières accordent une importance particulière aux nuisances comme la chute de feuilles et au critère ornementale.

La diversité fonctionnelle.

La maximisation de la diversité fonctionnelle est l'objectif central de notre système de recommandation. C'est précisément ce critère qui distingue notre approche des méthodes traditionnels de sélection d'arbres qui considèrent chaque plantation de manière isolée. Notre solution n'a pas pour seul objectif de proposer des espèces différentes, mais des espèces fonctionnellement complémentaires.

Également la règle 10 - 20 - 30 de *Santamour* suggère, qu'à l'échelle d'une forêt urbaine, de ne pas dépasser 10 % d'une espèce, 20 % d'un genre, 30 % d'une famille, afin de réduire le risque de pertes massives liées aux ravageurs et aux maladies. Cette règle est un critère de choix supplémentaire que nous considérons et qui favorise indirectement la maximisation de la diversité fonctionnelle.

Afin de pouvoir considérer la notion de diversité fonctionnelle, le critère de choix que nous devons considérer est la géolocalisation du site de plantation. C'est grâce à ces données que nous pouvons prendre en comptes les arbres alentours.

Enfin, pour prendre en compte ces critères de choix, nous collectons des informations sur l'espace disponible et les préférences des personnes utilisatrices via un formulaire lors du processus de recommandation :

- Une approximation de la surface de la zone de plantation.
- La présence de structures aériennes (des lignes électriques, des bâtiments adjacents, etc.).
  - Si oui, la hauteur disponible.
- La préférence pour les espèces ornementales.
- La préférence pour les espèces à fruits comestibles.
- La préférence pour les espèces à feuillage persistant ou à feuillage caduc.
- Les coordonnées du site de plantation sont prises automatiquement lors de la validation du formulaire. Un message informe la personne utilisatrice de se tenir au plus prés du site de plantation lors de cette étape.

4.3 Algorithme de recommandation

L'algorithme de recommandation que nous avons développé fait partie de la catégorie des systèmes de recommandation multicritères, où les recommandations se font sur plusieurs critères plutôt que sur une unique évaluation globale. Nous avons choisi cette approche, car elle est particulièrement adaptée à notre contexte de sélection d'arbres urbains, où la recommandation optimale dépend de plusieurs facteurs incluant des contraintes physiques, les préférences des personnes utilisatrices et des objectifs écologiques.

L'architecture de notre système de recommandation est une architecture en cascade qui comprend quatre phases :

1. Un filtrage par contraintes.
2. Un calcul des scores de diversité fonctionnelle.
3. Une agrégation multicritère.
4. Un classement.

Cette architecture permet une mise à jour indépendante de chaque composante et facilite l'intégration de critères supplémentaires.

La première phase de filtrage par contraintes élimine les espèces incompatibles avec les conditions du site de plantation. Cette phase implémente un système de règles qui considèrent la taille maximale l'arbre à maturité et la taille de la canopée. Nous éliminant directement tous les arbres qui ne correspondent pas aux conditions du site de plantation renseignées par la personne utilisatrice.

La deuxième phase calcule, pour chaque espèce candidate, des scores de contribution potentielle à la diversité fonctionnelle d'une part à l'échelle du quartier (dans un rayon de 500 m) et d'autre part à une échelle plus large (d'une municipalité par exemple, ce paramètre est ajustable). Ce calcul est réalisé selon l'approche développée par SylvCiT (St-Denis *et al.*, 2024).

La troisième phase effectue l'agrégation multicritères pour obtenir un score global pour chaque espèce candidate. Nous utilisons une fonction d'agrégation qui calcule le score global $S(i)$ pour l'espèce i comme une somme pondérée :

$$S(i) = w_1 \times P(i) + w_2 \times D_1(i) + w_3 \times D_2(i) \tag{4.2}$$

Où :

- $P(i)$ représente le score de correspondance aux préférences de la personne utilisatrice.
- $D_1(i)$ représente le score de contribution à la diversité fonctionnelle à l'échelle du quartier.
- $D_2(i)$ représente le score de contribution à la diversité fonctionnelle à l'échelle plus large.

Les poids w_1, w_2 et w_3 sont ajustables et permettent modifier simplement les priorités des recommandations. Par défaut, la pondération est équilibrée $(w_1 = w_2 = w_3 = 1/3)$.

La quatrième phase classe les espèces candidates. Les espèces sont ordonnées par score décroissant, et les 10 meilleures recommandations incluant des espèces candidates satisfaisant chacune des préférences de la personne utilisatrice. Concrètement si la personne utilisatrice avait stipulé qu'elle préférerait une espèce d'arbre produisant des fruits comestibles, et qu'aucune des 10 meilleures recommandations ne satisfait ce critère, alors la recommandation avec le plus bas score parmi les 10 est remplacée par l'espèce candidate qui satisfait ce critère et qui a le score le plus haut. Ce mécanisme permet de prendre systématiquement en compte les préférences de la personne utilisatrice tout en incitant à la maximisation de la diversité fonc-

tionnelle. Le classement final des recommandations est ensuite présenté à la personne utilisatrice.

En conclusion, notre algorithme de recommandation est un système hybride, qui combine plusieurs approches de recommandation, avec une architecture en cascade où les méthodes s'enchaînent séquentiellement. Il intègre les données d'inventaire public, les contributions citoyennes et les préférences des personne utilisatrices pour permettre de contribuer à la résilience et à la santé de la forêt urbaine tout en satisfaisant les personnes utilisatrices.

CHAPITRE 5

SYLVPRIV

Nous allons présenter dans ce chapitre l'application mobile SylvPriV que nous avons développée. Elle représente l'aboutissement de notre projet. L'application intègre les deux composantes principales, qui sont décrites dans les chapitres précédents : le module de reconnaissance d'espèces d'arbres (détaillé au chapitre 3) et le système de recommandation de groupes fonctionnels et d'espèces d'arbres (détaillé au chapitre 4). Nous expliquerons dans la partie 5.1 l'architecture logicielle et les choix technologiques que nous avons fait. La partie 5.2 détaillera les aspects de confidentialité et de sécurité des données, particulièrement sensible lorsqu'il y a collecte de données de géolocalisation. Enfin, la partie 5.3 décrira la stratégie de déploiement de la solution que nous avons adopté.

5.1 Architecture et technologies

5.1.1 Vue d'ensemble

L'architecture de la solution repose sur un modèle client-serveur conçu pour répondre à des exigences de performance de scalabilité et de maintenabilité. Cette architecture permet de séparer les responsabilités entre l'interface mobile et le traitement des données qui se fait côté serveur. Cela permet aussi une meilleure gestion des conflits et une grande flexibilité pour l'évolution future de la solution.

Elle s'articule autour de trois composantes interconnectées :

1. **L'application mobile**, qui est l'interface d'interaction avec les personnes utilisatrices. Elle permet la capture des images, l'affichage des résultats de reconnaissance et de recommandation d'espèces d'arbres, ainsi que la collecte des données contributives qui alimente notre base de données d'espèces d'arbres dans les espaces privés.
2. **L'application dorsale (*backend*)**, où le traitement des données est effectué. Il permet la communication avec l'application mobile et le module d'inférence. C'est à ce niveau, également, que les calculs du système de recommandation et la persistance des données dans la base de données sont effectués.
3. **Le module d'inférence**, qui héberge le modèle que nous avons entraîné pour la classification des

espèces d'arbres. Ce module est déployé sur le serveur et effectue les prédictions à partir des images transmises par l'application mobile.

Cette séparation des responsabilités suit le principe de conception "*Separation of Concerns*". Nous avons appliqué cette approche car cela permet à chaque composante d'évoluer indépendamment.

Le choix d'exécuter l'inférence du modèle de classification côté serveur plutôt que sur l'appareil mobile est motivé par plus facteurs. D'une part, car la mémoire et la puissance de calcul disponible sur des appareils mobiles d'entrée de gamme reste limitées. Et d'autre part, pour avoir la possibilité d'actualiser le modèle sans nécessiter une mise à jour de l'application mobile.

5.1.2 Application mobile

Le choix du cadriciel (*framework*) de développement mobile a des implications à long terme sur la maintenabilité de l'application, ses performances et son évolutivité. Après avoir réalisé une analyse comparative des principales options disponibles pour le développement d'une application (développement natif Android/iOS, React Native, Flutter, Xamarin), notre choix s'est porté sur Flutter.

Flutter est un kit de développement logiciel (SDK) à code source ouvert créé par Google et publié pour la première fois en 2018. Il avait été initialement conçu pour le développement d'applications mobiles Android et iOS, mais a depuis été étendu pour supporter également le web, les applications de bureau (Windows, macOS, Linux) et les systèmes embarqués. Depuis son lancement, Flutter a connu un rapide succès dans l'industrie, grâce à ses performances, des applications à fort volume d'utilisation comme Google Pay, Alibaba et eBay l'utilisent en environnement production.

Notre choix a été motivé par plusieurs raisons. Tout d'abord, Flutter permet de développer des applications pour Android et iOS à partir d'une base de code unique, écrite en Dart (un langage de programmation également développé par Google). Même si notre projet cible initialement le système d'exploitation Android, ce choix nous permet une extensibilité future vers iOS simple et sans une réécriture complète du code. Selon l'étude de Biørn-Hansen *et al.* (2020), les cadriciels de développement multiplateforme permettent une réduction significative du temps de développement par rapport au développement natif séparé.

Ensuite, contrairement à d'autre cadriciels multi-plateformes, comme React Native, qui utilise un pont Ja-

vaScript pour communiquer avec les composants natifs, Flutter compile directement le code Dart en code machine ARM natif via une compilation AOT (*Ahead-Of-Time*). Cela permet d'éliminer les ralentissements liés à ce pont et d'avoir ainsi d'excellentes performances.

Aussi, Flutter utilise son propre moteur de rendu graphique (Skia), ce qui garantit un visuel constant entre les différentes plateformes et versions d'Android. Cela permet d'assurer que l'interface utilisateur sera identique sur tous les appareils, indépendamment de la version d'Android ou du fabricant.

Enfin, l'écosystème de développement de Flutter est mature et dispose de packages bien maintenus pour les fonctionnalités essentielles, comme la capture d'images, la géolocalisation, les requêtes http, et la gestion d'état. La fonctionnalité de Hot Reload permet aussi de voir instantanément les modifications du code sans perdre l'état de l'application. Ce qui accélère considérablement le cycle de développement itératif, particulièrement précieux dans un contexte de recherche.

L'architecture interne de l'application mobile respecte le pattern MVVM (*Model-View-ViewModel*), qui est recommandé pour les applications de moyenne et grande taille. Cette structure sépare, comme pour l'architecture générale de la solution, l'application en trois composants principaux pour une meilleure organisation du code et une séparation des préoccupations (Separation of Concerns) :

- Le modèle (*Model*), qui représente les données de l'application et la logique métier. Il gère aussi l'accès aux sources de données locales ou distantes.
- La vue (*View*), qui est la couche *UI* (interface utilisateur) de l'application. Elle affiche les données et transmet les interactions de la personne utilisatrice au modèle de vue.
- Le modèle de vue (*ViewModel*), quant à lui, agit comme un intermédiaire entre le modèle et la vue. Il contient la logique de présentation, traite les actions de la personne utilisatrice, et expose les données du modèle d'une manière facilement utilisable par la vue.

5.1.3 Application dorsale

La technologie de la composante application dorsale de la solution a été imposé par le fait que la solution que nous avons développé doit s'intégrer avec la plateforme SylvCiT qui est basé sur le cadriciel Django. Cela dit, même si notre solution devait être indépendante, notre choix se serait également tourné vers ces technologies. En effet, Django est l'un des cadriciels les plus robuste et est largement utilisé dans l'industrie, mais aussi dans des contextes académiques.

Le langage utilisé est le Python, un langage de programmation interprété, créé par Guido van Rossum et publié en 1991. Il est polyvalent mais est particulièrement utilisé dans la recherche et l'apprentissage automatique. Il s'est imposé comme le langage de référence, grâce à un écosystème de bibliothèques scientifiques extrêmement riche incluant NumPy pour le calcul numérique, Pandas pour la manipulation de données (Raschka et Mirjalili, 2019).

Django possède de nombreux outils utiles pour le développement de projet comme le nôtre, comme un ORM (*Object-Relational Mapping*), un outil puissant qui permet de manipuler des données via des objets Python sans avoir à utiliser des requêtes SQL brutes. Il inclut aussi des protections contre les vulnérabilités web courantes comme la protection CSRF (*Cross-Site Request Forgery*), l'échappement automatique contre les attaques XSS (*Cross-Site Scripting*), la protection contre les injections SQL via l'ORM, et la gestion sécurisée des sessions. Ces mécanismes de sécurité réduisent significativement les risques d'attaques réussies de l'application.

Django existe depuis 2005 et bénéficie d'une communauté très active de développeurs. Cette maturité permet d'avoir une documentation exhaustive, un écosystème riche de bibliothèques, et une grande stabilité en environnement de production. Des géants de l'industrie comme Instagram, Pinterest et Mozilla utilisent Django pour des applications à grande échelle.

La communication entre l'application Flutter et l'application dorsale sous Django s'effectue via une API RESTful utilisant le protocole HTTPS. Une API RESTful (*Representational State Transfer*) est une interface de programmation qui permet la communication entre différentes applications, en utilisant des méthodes standardisées (GET, $POST$, PUT, $DELETE$) pour manipuler des données. Ce type d'architecture, défini par Fielding (2000), repose sur le principe que chaque requête contient toutes les informations nécessaires à son traitement. Cela permet d'uniformiser des interfaces, et de faciliter ainsi la communication entre systèmes hétérogènes (Fielding, 2000). Le format d'échange de données que nous utilisons est le JSON, il permet une compatibilité native avec les deux plateformes.

5.1.4 Base de données

Pour la persistance et l'indexation des données nous utilisons Apache Solr. C'est une plateforme de recherche à code source ouvert développée par la fondation Apache, basée sur la bibliothèque Apache Lucene. Elle dispose d'un moteur de recherche hautement performant qui est utilisé par des organisations

de grande envergure pour gérer des volumes de données massifs avec des temps de réponse relativement faibles.

L'utilisation de Solr dans notre solution permet une intégration sans conflit dans SylvCiT qui utilise déjà cet outil. Cela dit, Solr est tout de même l'une des solutions les plus adéquate pour notre projet. Premièrement, Solr offre un support natif des requêtes géospatiales grâce à son module spatial, permettant d'effectuer des recherches très efficaces par rayon ou par boîte englobante. Cette fonctionnalité est essentielle pour notre algorithme de recommandation qui doit identifier rapidement les arbres présents dans un rayon donné autour du site de plantation. Deuxièmement, l'architecture distribuée de Solr, permet une scalabilité horizontale adaptée à la croissance progressive de notre base de données d'arbres dans les espaces privés alimentée par les personnes citoyennes. Enfin, les performances de recherche de Solr, grâce à ses index inversés, permettent des temps de réponse de l'ordre de la milliseconde même sur de larges collections. Dans notre contexte, cela a un impact direct pour la recherche d'espèces par synonymes, et pour l'algorithme de recommandation lors du filtrage des espèces d'arbres par critères. Cette efficacité est d'autant plus importante que notre application est conçue pour une utilisation en temps réel.

5.2 Confidentialité et sécurité

5.2.1 Confidentialité et protection des données personnelles

La collecte de données lors des contributions citoyennes, notamment les données de géolocalisation, soulève des défis importants en matière de protection des données personnelles. Nous avons intégré dans notre solution des mécanismes de sécurité à plusieurs niveaux pour garantir la protection et la confidentialité des données des personnes utilisatrices. C'est mécanisme vont au-delà des exigences légales, en respectant la Loi 25 du Québec (anciennement projet de loi 64) et le Règlement Général sur la Protection des Données (RGPD) européen, qui sont les cadres réglementaires de référence.

Principe de minimisation des données.
Nous ne collectons que les données strictement nécessaires au bon fonctionnement de la solution. Pour la reconnaissance d'espèces d'arbres, seule l'image de la feuille est transmise au serveur et n'est pas conservée après l'inférence. Aussi le formulaire (détaillé à la section 4.2) nécessaire au module de recommandation d'espèces d'arbres à planter ne contient que les informations indispensables à la recommandation. Les coordonnées de géolocalisation sont collectées uniquement avec le consentement explicite de la personne

utilisatrice, qui peut choisir de réduire la précision des coordonnées (dans les paramètres de son téléphone mobile).

Anonymisation des données.
Nous avons fait le choix de ne demander aucune inscription sur l'application, les personnes utilisatrices n'ont ainsi pas à introduire leurs informations personnelles (nom, prénom, date de naissance, etc.). Les observations d'arbres sont donc stockées de manière totalement anonyme. Aucun élément ne permet de relier une observation à une personne utilisatrice spécifique, les observations sont simplement reliées à une position géographique.

Consentement éclairé.
Lors de la première utilisation de l'application, nous affichons une explication claire sur l'utilisation qui sera faite des données de géolocalisation. La personne utilisatrice doit donner son consentement explicite en autorisant l'application à accéder au module de géolocalisation de son téléphone mobile. Cette autorisation peut être révoquée à tout moment dans les paramètres.

5.2.2 Sécurité

Au delà de la confidentialité, la sécurité technique est essentielle pour éviter les accès non autorisés, les fuites et les compromissions des données. De nombreux travaux en sécurité informatique mettent en évidence plusieurs catégories de menaces, comme l'exploitation de permissions excessives, les attaques réseau (interception, injection), la fuites de données, et la mauvaise gestion des secrets applicatifs.

Dans notre solution nous avons mis en place plusieurs mécanismes de sécurité pour réduire au maximum les risques liés à ces menaces. L'application dorsale implémente avec Django plusieurs couches de protection. Tout d'abord la limitation du débit (*rate limiting*), qui vise à réguler le nombre de requêtes par adresse IP pour prévenir les attaques par déni de service et l'abus de l'API. La validation de toutes les données reçus avant le traitement, permet de prévenir l'injection de code malveillant. Également, la protection contre les injections SQL est garantie par l'utilisation systématique de l'ORM Django et les requêtes géospatiales sont paramétrées pour éviter toute injection via les coordonnées de géolocalisation. Enfin, nous avons mis en place une journalisation et un monitoring des accès à l'API et les erreurs sont journalisés pour permettre la détection d'activités suspectes. Les logs sont centralisés et analysés pour identifier les patterns d'abus potentiels.

La sécurité des communications entre les différentes composantes de notre solution, notamment entre l'application mobile et l'application dorsale, est aussi un élément important que nous avons pris en compte. Toutes les communications entre l'application mobile et le serveur sont chiffrées via le protocole de sécurité TLS 1.3 (*Transport Layer Security*). Aussi, afin de prévenir les attaques de type "*man-in-the-middle*", nous avons mis en place un système de certificats pour s'assurer que les requêtes proviennent bien de notre application mobile, empêchant ainsi un attaquant d'intercepter et de manipuler les communications. Ces tokens d'authentification sont stockés dans le stockage sécurisé de l'appareil mobile via la bibliothèque *flutter_secure_storage*, qui utilise le *Keystore* chiffré sur Android. Ce mécanisme de stockage sécurisé est protégé par les fonctionnalités de sécurité matérielle des appareils mobiles.

5.2.3 Confiance et comportement des personnes utilisatrices

La confiance des personnes utilisatrices en une application mobile est un élément essentiel de son succès. En effet, des études prouve que la perception par les personnes utilisatrices de la confidentialité et de la sécurité influence directement leur adoption, leur satisfaction et leur fidélité à l'utilisation d'une application (Choi *et al.*, 2021; Jozani *et al.*, 2020). L'étude de Balapour *et al.* (2020) montre que la perception de risque de confidentialité réduit la perception de sécurité, tandis que l'efficacité perçue de la politique de confidentialité renforce la perception de sécurité (Balapour *et al.*, 2020).

Également les préoccupations de confidentialité liées aux permissions demandées par une application mobile, telles que l'accès à la localisation ou à l'appareil photo, ont un poids plus important que d'autres facteurs, comme la performance, l'utilité perçue ou le design, dans l'adoption de cette application par les personnes utilisatrices. (Degirmenci, 2020; Furini *et al.*, 2020).

Cependant, on constate que même des personnes utilisatrices techniquement compétentes et conscientes des risques installent et utilisent des applications pour lesquelles elles développent des inquiétudes, lorsque les bénéfices perçus sont élevés, c'est le "*privacy paradox*" (Barth *et al.*, 2019; Pentina *et al.*, 2016). Les personnes utilisatrices ressentent souvent une fatigue de la confidentialité due à une multiplication des demandes de consentement, ce qui les pousse le plus souvent à accepter rapidement sans lire. Il a été donc crucial pour nous de concevoir des interfaces de consentement simples et compréhensibles (Tang *et al.*, 2020).

Enfin, les données agrégées sont utilisées uniquement à des fins de recherche et de planification en fores-

terie urbaine. Aucune donnée individuelle n'est partagée avec des tiers.

5.3 Déploiement

La stratégie de déploiement que nous avons choisi vise à assurer une meilleure portabilité et une automatisation du processus de déploiement. Pour cela nous avons opté pour Docker (Merkel *et al.*, 2014), une plateforme opensource qui fournit une virtualisation légère au niveau système d'exploitation et permet de conteneuriser une application (code source, dépendances et configuration). Une même machine physique ou virtuelle peut accueillir plusieurs conteneurs isolés.

L'application dorsale est donc déployée sur le serveur qui héberge actuellement SylvCiT dans une image Docker sans interférer avec le bon fonctionnement des autres applications déjà présente sur le serveur. En ce qui concerne le reste de la solution, l'application mobile Android sera mise à la disposition des utilisateurs sur le *Google Play*. Le processus de publication comprend la mise à disposition des métadonnées de l'application (description, captures d'écran et politique de confidentialité), la signature de l'APK avec une clé de publication sécurisée, et le passage en revue par l'équipe de modération de Google. Également, l'architecture multiplateforme de l'application mobile permettra ultérieurement, sans trop d'adaptation du code source, de publier également l'application sur l'*App Store* d'Apple pour les appareils iOS, élargissant ainsi la base de potentielles personnes utilisatrices.

Notre modèle de classification est quant à lui chargé sur le serveur, une seule fois au démarrage du conteneur et maintenu en mémoire pour minimiser la latence des inférences. Cela évite le coût de chargement du modèle à chaque requête, qui représenterait plusieurs secondes d'attente additionnelles pour les personnes utilisatrices.

Enfin, le code source de la solution complète est publié en code source ouvert sous la licence GPL-v3. Il est disponible dans le dépôt suivant : https://gitlab.labikb.ca/ikb-lab/data-science/sylvpriv.git. À noter que le projet a été lauréat de l'édition Hiver 2026 du Concours MixCité, et l'application fera l'objet d'une adaptation en vue d'un déploiement grand public fonctionnel, prévu pour septembre 2026.

CHAPITRE 6
RÉSULTATS ET DISCUSSIONS

Ce dernier chapitre présente les résultats obtenus à l'issu de notre travail, ainsi que les limites observées et les perspectives d'évolution envisageables. Dans la partie 6.1, nous analyserons les performances du modèle d'identification d'espèces d'arbres. La partie 6.2 sera consacrée à l'application mobile, où nous illustrons un cas d'utilisation concrets. Enfin, nous discuterons de manière critique dans la partie 6.3 des tenants et les aboutissants de notre projet.

6.1 Modèle d'identification d'espèces d'arbres

6.1.1 Résultats obtenus

L'évaluation de notre modèle de classification est une étape cruciale pour valider la pertinence de notre approche car l'algorithme de recommandation se base en partie sur les espèces d'arbres présentes sur le site de plantation.

Comme décrit à la section 3.4, l'évaluation finale a été réalisée sur l'ensemble de test, représentant $20\,\%$ des données, soit environ 2 800 images de feuilles réparties sur les 150 espèces d'arbres. Les métriques principales retenues sont la précision, le rappel, le score F1 et le macro score F1, cette dernière est particulièrement adaptée à notre problème à cause du nombre relativement élevé de classes et de la forte similarité morphologique entre certaines espèces (Kumar *et al.*, 2012).

Les performances du modèle sur l'ensemble de test sont résumées par trois métriques. La précision moyenne qui est de 65,7 %, ce indique que lorsque le modèle prédit une espèce donnée, cette prédiction est correcte dans environ 66 cas sur cent en moyenne sur l'ensemble des espèces, et cela indépendamment de leur fréquence d'apparition dans le jeu de données. Le rappel moyen quant à lui est de 81,6 %, ce signifie que le modèle parvient à identifier en moyenne correctement 81,6 % des images de chaque espèce, ce qui reflète sa capacité à ne pas manquer d'individus. Nous avons donc obtenu un macro score F1 de 72,8 %, qui constitue la synthèse la plus représentative des performances globales du modèle, en pondérant toutes les espèces quelle que soit leur fréquence dans le jeu de données, il pénalise les mauvaises performances sur les espèces rares autant que sur les espèces très représentées. Cela signifie que dans près de trois cas sur

quatre, le modèle identifie correctement l'espèce de l'arbre dès la première prédiction.

Ces performances sont comparables, à celles rapportées dans la littérature pour des tâches similaires de classification d'espèces végétales à partir d'images de feuilles. L'étude de Hart *et al.* (2023), qui compare de manière détaillée les résultats obtenus sur les 5 applications les plus utilisées pour cette tâche, rapporte qu'ils obtiennent en moyenne sur l'ensemble des applications, des performances comparables à notre modèle (tableau 6.1), entraîné spécifiquement sur les espèces présentes dans la région de Montréal.

Espèces	**Groupe fonctionnel**	**Précision**	**Rappel**	**Score F1**
tilia cordata	2B	0.8091	0.8342	0.8214
quercus coccinea	4A	0.8213	0.7587	0.7888
quercus palustris	4A	0.8228	0.7508	0.7852
acer pseudoplatanus	2A	0.7321	0.8400	0.7823
platanus occidentalis	5A	0.7717	0.7780	0.7748

Tableau 6.1 – Résultats des 5 meilleures espéces.

Nous avons également analysé les performances par groupe fonctionnel, ce qui nous a permis de révéler certaines disparités entre les groupes (tableau 6.2). Les feuillus pionnier avec une tolérance moyenne à l'ombre, sécheresse et inondations (groupe 5A) ont les meilleures performances avec un macro score F1 de 85 %, ce qui s'explique par la morphologie de leurs feuilles qui offre beaucoup de caractéristiques discriminantes visibles, comme la forme générale, le type de marge (dentée ou entière) et la nervation. Ces descripteurs sont particulièrement utilisés par notre réseau de neurones pour l'identification de l'espèce. À l'inverse, les feuilus à feuilles larges et minces, tolérants à la sécheresse et intolérants à l'inondation (groupe 2C) obtiennent les performances les plus basses avec un macro score F1 de 78 %.

Groupe fonctionnel	Précision	Rappel	Macro F1
1A : Conifères, tolérants à l'ombre	0.81	0.83	0.82
1B : Conifères, tolérants à la sécheresse	0.79	0.79	0.79
1C : Conifères, aiguilles caduques	0.79	0.84	0.81
2A : Feuilus à feuilles larges et minces, tolérants à l'ombre	0.82	0.82	0.82
2B : Feuilus à feuilles larges et minces, tolérance moyenne à l'ombre, sécheresse et inondations	0.82	0.83	0.83
2C : Feuilus à feuilles larges et minces, tolérants à la sécheresse et intolérants à l'inondation	0.77	0.78	0.78
3A : Feuilus à feuilles petites et épaisses, tolérants à l'inondation	0.79	0.81	0.80
3B : Feuilus à feuilles petites et épaisses, tolérants à la sécheresse	0.80	0.80	0.80
4A : Feuilus à semances lourdes, tolérants à la sécheresse, chênes	0.80	0.82	0.81
4B : Feuilus à semances lourdes, peu tolérants à l'inondation, noyers et marronniers	0.84	0.81	0.82
5A : Feuilus pionniers, tolérance moyenne à l'ombre, sécheresse et inondations	0.86	0.85	0.85
5B : Feuilus pionniers, tolérants à l'inondation	0.87	0.81	0.84

Tableau 6.2 – Résultats par groupes fonctionnels.

Afin d'aller plus loin dans l'étude des performances de notre modèle, nous avons comparé le Score F1 de chaque classe nous a permis d'identifier les espèces qui pose le plus problème au modèle. De plus, la matrice de confusion normalisée (figure 6.1) nous a permis de mieux visualiser ces tendances de confusion. On remarque que les erreurs de classification se produisent majoritairement au sein d'un même genre. Par exemple, les confusions les plus fréquentes concernent les différentes espèces d'érables (*Acer*), qui partagent des caractéristiques foliaires similaires. Parmi les paires d'espèces les plus fréquemment confondues, on trouve l'érable à sucre (*Acer saccharum*) et l'érable argenté (*Acer saccharinum*).

Figure 6.1 – Matrice de confusion par groupe fonctionnel.

L'analyse des courbes de perte d'entraînement et de validation au fil des époques confirme l'efficacité de la stratégie d'ajustement fin que nous avons mise en place. La convergence est atteinte après environ 56 époques, avec une stabilisation de la perte de validation autour de 0,85. Le petit écart entre la perte d'entraînement et la perte de validation à la fin de l'entraînement montre un léger surapprentissage, mais celui-ci reste modéré notamment grâce aux techniques de régularisation que nous avons mises en place (décrites dans la section 3.3.4). La phase de préchauffage de trois époques a joué un rôle crucial dans la stabilisation des gradients initiaux.

Enfin, l'un des résultats obtenus est le temps d'inférence moyen de 52 ms. La stratégie mise en place nous a permis d'obtenir un compromis intéressant entre qualité des prédictions et rapidité de calcul, ce qui est d'une importance capitale dans notre cas où la classification doit se faire en temps réel.

6.1.2 Limites et défis

Malgré des performances globalement satisfaisantes, notre modèle d'identification d'espèce d'arbre présente certaines limites.

Tout d'abord, la limite que nous avons identifiée est le déséquilibre de l'ensemble de données. Bien que nous ayons utilisé des techniques d'augmentation de données, certaines espèces restent sous-représentées dans notre ensemble de données d'entraînement. Ce déséquilibre est l'une des causes de l'hétérogénéité des performances selon les espèces que nous avons observées. Les espèces mieux représentées dans notre ensemble de données sont mieux identifiées par le modèle. Ce déséquilibre est aussi marqué par le fait que la majorité des photos de feuilles de notre ensemble de données proviennent d'observations faites durant la période estivale, lorsque les feuilles sont à leur stade de développement maximal et largement verdoyante. On peut s'attendre à ce que le modèle ait des performances moindres avec des photos de feuilles printanières (encore en croissance) et automnales (avec un changement de coloration). Cependant, les tests informels que nous avons effectués avec des images de feuilles automnales, montrent une légère baisse du macro score F1.

La seconde limite est la sensibilité du modèle aux conditions de prise de vue des photos des feuilles. Bien que l'augmentation de données simule diverses conditions d'éclairage et d'orientation, le modèle reste sensible aux images de mauvaise qualité. Les photos prises par les personnes utilisatrices peuvent être floues, sous-exposées ou avec un fond trop chargé et non uniforme, cela peut conduire à des prédictions incorrectes. Cette limitation est particulièrement problématique pour une application citoyenne où les conditions de capture des photos ne peuvent pas être contrôlées.

La limite qui est probablement la plus évidente est le cas des espèces hors distribution. En effet, le modèle ne dispose d'aucun moyen explicite pour détecter et gérer le cas des espèces non présentes dans son ensemble de données d'entraînement. Lorsqu'une personne utilisatrice soumet une photo d'une espèce non incluse parmi les 150 espèces, le modèle attribue tout de même une prédiction qui correspond à l'espèce connue dont la feuille lui ressemble morphologiquement le plus.

Enfin, les similarités intra-genre représentent une limite importante. Comme mentionné précédemment, les confusions entre espèces d'un même genre sont un défi majeur qui reflètent une réalité botanique. Dans certains cas, même des experts peuvent hésiter entre plusieurs espèces sans examiner d'autres caractéristiques comme l'écorce, les bourgeons ou encore les fruits de l'arbre. On constate également une variabilité intra-espèce, où les feuilles au sein d'une même espèce peuvent avoir d'importantes différences selon l'âge de la feuille, sa localisation dans l'arbre, la saison, et les conditions environnementales. Ces difficultés sont bien connues dans l'identification botanique à partir des feuilles, notamment pour les systèmes utilisant la vision par ordinateur (Kumar *et al.*, 2012).

6.1.3 Perspectives d'amélioration

Nous avons vu précédemment que notre modèle d'identification d'espèces d'arbres présente quelques limites qui peuvent être atténuées ou même totalement supprimées en réentraînant notre modèle. Dans la section 5.1.3, nous avions vu que l'inférence se fait côté serveur, il est donc assez simple de mettre à jour le modèle alors que la solution est en production, cela ne nécessite pas une mise à jour de l'application mobile. Plusieurs pistes d'amélioration peuvent être envisagées pour surmonter les limites et optimiser les performances du modèle.

Un enrichissement de l'ensemble de données d'entraînement permettrait d'améliorer directement la qualité de la classification. En effet, nous pourrions augmenter la quantité et la diversité des images auquel le modèle est confronté, particulièrement pour les espèces sous-représentées. Également, en ajoutant des images de feuilles prises durant différentes saisons. L'intégration des contributions citoyennes collectées via l'application permettra d'enrichir progressivement l'ensemble de données avec des images prises en conditions réelles. Cela nécessitera la mise en place d'un mécanisme de validation par des experts pour garantir la qualité des annotations avant leur intégration à l'ensemble de données d'entraînement.

Une évolution majeure serait de permettre l'identification à partir de plusieurs organes de l'arbre (les feuilles, l'écorce, les fleurs, les fruits et les bourgeons). Cette approche inspirée de celle utilisé pour Pl@ntNet (Goëau *et al.*, 2013), permettrait notamment d'éviter les confusions lors de fortes similarités intra-genre où les feuilles seules ne suffisent pas toujours à discriminer deux espèces. L'implémentation concrète pourrait se faire à travers un ensemble de modèles spécialisés dont les prédictions seraient fusionnées de manière pondérée.

Cependant, le problème des données hors distribution resterait entier. L'une des solutions serait de l'intégration d'un mécanisme de détection OOD (Out-of-Distribution), qui permettrait d'identifier les cas où l'image soumise ne correspond à aucune des 150 espèces connues par le modèle. Par exemple en analysant l'entropie de la distribution des sortie de la fonction *softmax*, ou en entraînant un détecteur d'anomalies sur les représentations du réseau dorsal. Une autre solution serait simplement d'augmenter progressivement le nombres d'espèces que le modèle est capable d'identifier, pour arriver finalement à l'ensemble des espèces présentent dans la région urbaine de Montréal.

Enfin, toutes les perspectives d'évolution citées peuvent être mises en place parallèlement et permettraient d'améliorer grandement les performances du modèle.

6.2 Cas d'utilisation

Nous allons voir un cas d'utilisation concret de la solution dans son ensemble. Nous prendrons pour exemple une personne utilisatrice se trouvant dans le quartier du Plateau de Montréal, désirant planter un nouvel arbre dans jardin, à l'emplacement indiqué sur l'image ci-dessous. Nous considérerons que la personne utilisatrice à au préalable téléchargé et installé l'application SylvPriV sur son téléphone.

La première partie du processus consiste à identifier les arbres déjà présents. Pour cela l'application commence par demander à la personne utilisatrice si des arbres sont déjà présents dans l'espace privé. Si c'est le cas et que la personne utilisatrice connaît précisément le nom de l'espèce de l'arbre, l'application demande simplement de se placer le plus près possible de l'arbre en question puis de sélectionner dans une liste déroulante avec recherche le nom de l'espèce. Si la personne utilisatrice ne connaît pas le nom de l'espèce de l'arbre, l'application la redirige vers le module d'identification d'espèce. Avant la prise de la photo, de rapides indictions sont affichées pour orienter la personne utilisatrice (se tenir au plus près de l'arbre, choisir une feuille non dégradée, prendre la photo en plein jour, placer la feuille sur une surface uniforme ou sur la main ouverte, etc.). Ces indications permettent d'avoir des coordonnées de géolocalisation les plus précises possibles et de fournir au modèle une image optimale afin de maximiser les chances d'avoir une classification correcte. En complément, quelques informations facultatives sur l'arbre sont demandées à la personne utilisatrice (l'année de plantation, son diamètre à hauteur de poitrine, une estimation de sa taille et une photo). Cela permet d'obtenir les informations susceptibles d'être utiles pour le système de recommandation, sans que cela ne soit contraignant pour la personne utilisatrice. Une fois la photo prise, elle est automatiquement redimensionnée et envoyé au serveur pour l'inférence. Le modèle va procéder à

l'identification et retourne à l'application mobile les 5 espèces ayant eu les plus fortes probabilités avec leur indice de confiance. La personne utilisatrice peut ensuite choisir à l'aide des images des espèces d'arbres des résultats, l'espèce qui correspond réellement à l'arbre présent. À l'issue de cette étape, l'arbre est ajouté à la base de données des arbres en espace privé, et est donc pris en compte par l'algorithme de recommandation.

La seconde partie du processus consiste à générer une recommandation d'espèces d'arbres adaptées à l'emplacement choisi par la personne utilisatrice, tout en prenant en compte les contraintes de l'espace privé et les préférences de la personne utilisatrice, et en gardant comme objectif de maximiser la diversité fonctionnelle. Dans un premier temps, l'application demande à la personne utilisatrice de se placer précisément à l'emplacement de plantation souhaité, afin de pouvoir récupérer les coordonnées de géolocalisation. Cette étape est importante, car elle permettra d'une part à l'algorithme de recommandation d'identifier les arbres environnants nécessaires pour la maximisation de la diversité fonctionnelle, et d'autre part pour pouvoir ajouter l'arbre une fois planté dans la base de données des espèces d'arbres privés. Une fois l'emplacement défini, l'application présente un formulaire court destiné à renseigner les contraintes du site de plantation. La personne utilisatrice indique la surface et la hauteur approximatives disponibles, ainsi que la présence potentielle de contraintes aériennes (fils). L'objectif est de filtrer en amont les espèces incompatibles, afin d'éviter de recommander des arbres qui deviendraient problématiques à moyen ou long terme.

Par la suite, la personne utilisatrice est invitée à exprimer ses préférences. Elle peut indiquer si elle souhaite un arbre fruitier, si elle privilégie une espèce ornementale, si elle préfère un arbre à feuillage persistant ou caduc. Ces informations ne sont pas obligatoires, mais elles permettent d'orienter les recommandations vers des espèces en accord avec les attentes réelles de la personne utilisatrice. Une fois ces informations renseignées, l'application transmet la requête au serveur. L'algorithme commence par récupérer l'ensemble des arbres pertinents dans un rayon défini autour du point de plantation (les arbres publics et les arbres privés), afin d'évaluer la diversité fonctionnelle existante, et d'identifier les groupes fonctionnels qui sont déjà fortement représentés et au contraire, lesquels sont sous-représentés. À l'issue du calcul de recommandation, l'application affiche les 10 meilleures espèces comme détaillé dans la section 4.3 classées. Pour chaque espèce proposée, une fiche explicative est accessible, qui présente le nom commun, le nom scientifique, une image, ainsi que des informations. L'application met également en évidence la justification de la recommandation, en indiquant le groupe fonctionnel de l'espèce, en quoi elle respecte les contraintes du site de plantation, et si elle correspond aux préférences de le personne utilisatrice.

Enfin, la personne utilisatrice peut sélectionner une espèce parmi les recommandations et automatiquement enregistré comme projet de plantation. Une fois la plantation effectuée, la personne utilisatrice peut revenir dans l'application pour valider que l'arbre a bien été planté. Cela permet d'enrichir progressivement la base de données des arbres en espaces privés, améliorant ainsi la qualité des recommandations futures dans le même secteur.

6.3 Discussion

6.3.1 Retour sur les résultats

En prenant du recul sur les résultats présentés précédemment, nous pouvons affirmer qu'à l'issue de notre travail, les objectifs principaux ont été atteints. La solution développée permet l'identification automatique d'espèces d'arbres à partir de photos de feuilles et la recommandation d'espèces à planter en fonction de contraintes et de préférences tout en maximisant la diversité fonctionnelle. Elle permet aussi de contribuer à l'enrichissement de la base de données des arbres en espaces privés.

Le modèle de classification atteint des performances satisfaisantes, dans la grande majorité des cas l'espèce correcte figure parmi les cinq propositions présentées à la personne utilisatrice. Cette qualité de la prédiction est amplement suffisante pour un outil destiné au grand public. En effet, l'objectif n'est pas de remplacer l'expertise d'un botaniste mais d'identifier de manière suffisamment fiable les espèces d'arbre urbains présents dans les espaces privés. L'utilisation qui est faite de la base de données constituée, nous permet d'accepter cette marge d'erreur.

Le système de recommandation que nous avons développé, permet quant à lui, de proposer des espèces d'arbre à planter en intégrant la notion de diversité fonctionnelle, contrairement aux guides de plantation traditionnels qui considèrent chaque arbre de manière isolé. Comme l'algorithme se base sur des filtres et des calculs précis, cela nous assure que les espèces recommandées correspondent à des groupes fonctionnels sous représentés, respectent les contraintes du site de plantation et incluent au moins une espèce qui correspond aux préférences de la personne utilisatrice.

L'originalité principale de notre travail réside dans le couplage entre reconnaissance automatique d'espèces et recommandation multicritère. À notre connaissance, aucune application existante ne propose un processus complet allant de l'identification des arbres existants à la recommandation d'espèces à planter en fonc-

tion de la diversité fonctionnelle locale (publique et privée). L'intégration à la plateforme SylvCiT constitue également un point important. En exploitant les données d'inventaires publics déjà agrégées par SylvCiT et en contribuant réciproquement à enrichir ces données avec les données d'arbres en espaces privés, notre projet participe à la construction d'une vision plus complète de la forêt urbaine.

6.3.2 Limites

Il convient toutefois de nuancer l'impact des limites énoncées précédemment à la section 6.1.2 sur le fonctionnement global de l'application. En effet, le système de recommandation se base principalement sur les groupes fonctionnels et non sur l'espèce précise de chaque arbre. L'analyse de la matrice de confusion nous a permis de remarquer que les erreurs de classification se produisent majoritairement entre les espèces taxonomiquement proches, qui appartiennent au même genre ou à la même famille. Ces espèces partagent généralement les même caractéristiques écophysiologiques, et sont donc souvent classées dans le même groupe fonctionnel selon la classification de Belluau *et al.* (2021). Par exemple, une confusion entre l'érable à sucre (*Acer saccharum*) et l'érable noir (*Acer nigrum*), qui appartiennent tout les deux au groupe fonctionnel $2A$ (feuillus à feuilles larges et mince, tolérants à l'ombre), n'affecte pas la pertinence des recommandations produites par le système de recommandation qui vise à maximiser la diversité fonctionnelle. Cette même logique s'applique aux espèces hors distribution. Lorsqu'une espèce absente de l'ensemble de données d'entraînement est soumise au modèle, il y'a une forte probabilité qu'il l'associe à une espèce partageant des caractéristiques foliaires comparables. Cette similarité morphologique est souvent liée à une proximité fonctionnelle, donc dans la majorité des cas, l'espèce prédite par le modèle appartiendra au même groupe fonctionnel que l'espèce réelle. De plus, le système de recommandation fonde son analyse sur une agrégation statistique de l'ensemble des arbres présents dans un rayon autour du lieu de plantation, ce qui permet une certaine tolérance aux erreurs ponctuelles. Quelques classifications erronées parmi les centaines d'arbres pris en compte dans le calcul de la diversité fonctionnelle, altèrent de manière négligeable la distribution des groupes fonctionnels observée et donc n'impacte pas les recommandations produites.

Au-delà des limites du modèle de classification, nous avons soulevé plusieurs limites plus générales qui méritent d'être soulignées concernant la solution dans son ensemble.

Le système de recommandation est dépendant des données d'inventaires publics, la pertinence des recommandations repose fondamentalement sur leur disponibilité et leur qualité. Or, bien que très utiles,

ces inventaires présentent quelques lacunes. Quand ces inventaires sont disponibles, il peut arriver que les mises à jour soient irrégulières et que le recensement soit incomplet (arbres récemment plantés ou abattus encore non enregistrés). Certaines municipalités ne disposent pas d'inventaires géoréférencés des arbres publics qui soit accessibles, ce qui limite l'utilisation pertinente de l'application aux zones répertoriées uniquement.

La limitation de la portée géographique de l'application va plus loin, car le modèle d'identification d'espèces d'arbres et la base de données de référence sont eux même spécifiquement conçus pour la région de Montréal. L'application n'est donc pas directement utilisable dans d'autres régions où la composition spécifique de la forêt urbaine diffère.

Une limite notable de l'application réside dans sa disponibilité exclusive sur le système d'exploitation Android. Or, au Canada, iOS détient environ 60 % de part de marché contre 40 % pour Android (la part des autres systèmes d'exploitation étant négligeable) (StatCounter GlobalStats, 2025a). Cette situation est opposée par rapport aux statistiques à l'échelle mondiale, où Android domine avec une part de marché mondiale d'environ 72 % contre 28 % pour iOS (StatCounter GlobalStats, 2025b). Cette limitation est d'autant plus significative que les personnes utilisatrices cibles de l'application, les propriétaires de maisons avec jardins ou cours, correspondent souvent à un segment socio-économique où la présence d'iOS est plus élevée. En ciblant uniquement Android, notre application se prive donc potentiellement de plus de la moitié des personnes utilisatrices cibles.

Le choix d'une approche participative pour notre projet, où les personnes citoyennes contribuent à la collecte de données, soulève une limite par rapport à la qualité des données obtenues. Les travaux de recherche de Bonney *et al*. (2014) sur la science citoyenne montrent que la qualité des données dépend fortement de la conception de l'interface de collecte et des mécanismes de validation. Nous avons tenté de minimiser les risques d'erreur en guidant les personnes utilisatrices (indications précises, conseils de prise de vue et formulaires à choix contraints) et en incluant dans l'application la fonctionnalité d'identification automatique d'espèces de d'arbres.

Enfin, un biais de participation est aussi à prendre en compte. Les personnes utilisatrices de l'application pourraient ne pas constituer pas un échantillon uniformément réparti dans la ville. Les contributions seront probablement concentrées dans certains quartiers où personnes citoyennes sont plus connectées et plus

sensibilisées aux enjeux environnementaux. Ce biais spatial nécessiterait, sur le long terme, des stratégies de recrutement de personnes utilisatrices ciblées pour équilibrer la couverture géographique.

6.3.3 Perspectives d'évolution

Au-delà des résultats obtenus, plusieurs perspectives d'évolution permettraient d'enrichir notre projet et d'accroître son impact sur la foresterie urbaine. Nous allons lister ci-dessous quelques-unes de ces pistes d'amélioration.

Tout d'abord, comme mentionné précédemment, la disponibilité exclusive de l'application sur Android constitue une limitation significative. Le développement d'une version iOS serait donc une priorité pour toucher le plus grand nombre de personnes utilisatrices. L'architecture multiplateformes choisie facilitera cette évolution, le code source étant facilement adaptables pour déployer une application sous iOS. Les principales adaptations se feront sur les spécificités des interfaces natives (comme les permissions, l'accès au module de géolocalisation du téléphone, et l'intégration au système). Une version web pourrait également être envisagée pour permettre un accès sans installation, particulièrement utile pour les utilisateurs occasionnels.

Le système de recommandation actuel propose des espèces à planter en fonction de la diversité fonctionnelle locale et de critères de base assez restreints. L'ajout de préférences supplémentaires serait par la suite une amélioration pertinente, qui permettrait une personnalisation accrue des recommandations. Les critères pourraient porter sur des préférences esthétiques (comme la couleur du feuillage et la floraison printanière) et sur des contraintes pratiques (comme la vitesse de croissance, allergénicité du pollen, et le niveau d'entretien requis). Cette personnalisation permettrait de renforcer la pertinence des recommandations par rapport aux besoins réels des personnes utilisatrices, augmentant ainsi la probabilité de concrétisation des plantations suggérées.

Dans la même logique, une évolution majeure serait d'intégrer directement les données d'Hydro-Québec sur le réseau de distribution électrique. Cela permettrait de prendre en compte la présence de lignes électriques aériennes dans les recommandations. Le système pourrait ainsi détecter automatiquement la proximité de câbles électriques à partir des coordonnées de géolocalisation de la personne utilisatrice et de filtrer ainsi les espèces recommandées en fonction de leur hauteur à maturité. Cette fonctionnalité réduirait les risques de conflits futurs entre l'arbre et les infrastructures électriques.

Dans une moindre mesure, l'implémentation d'un mode hors connexion serait envisageable. L'application, telle que développée actuellement, exige une connexion internet, ce qui peut constituer un frein dans les zones où la couverture réseau est faible ou inexistante. Les contributions citoyennes collectées hors ligne seraient automatiquement synchronisées dès le retour à une zone connectée. Cela dit, seule le module d'identification d'espèce d'arbre serait possible hors connexion, le système de recommandation devra impérativement consulter la base de données des arbres inventoriés pour prendre en compte la diversité fonctionnelle. Le réseau cellulaire étant de nos jours disponible dans la majorité des zones urbaines, cette amélioration ne semble pas être prioritaire.

En revanche, l'ajout de fonctionnalités de suivi des plantations recommandées, serait un gros atout pour l'application. Cela permettrait aux personnes utilisatrices d'enregistrer leurs plantations et de documenter leur croissance au fil du temps. Un guide d'entretien propre à l'espèce d'arbre plantée pourra aussi être fournis. Ces données sur le long terme offriraient une ressource précieuse pour la recherche sur la croissance des arbres en milieu urbain privé.

Pour encourager l'adoption de l'application et accroître son attractivité, il serait intéressant d'intégrer une fonctionnalité de réalité augmentée (RA) qui permettrait aux personnes utilisatrices de visualiser l'arbre recommandé directement dans son espace avant de procéder à sa plantation. En pointant la caméra de leur téléphone vers l'emplacement de plantation voulu, les personnes utilisatrices pourraient afficher un modèle tridimensionnel de l'arbre à son stade de croissance maximal. Cela leur permettrait d'évaluer son occupation de l'espace et son intégration dans le paysage existant. Cette prévisualisation renforcerait l'engagement des personnes utilisatrices en rendant l'expérience plus immersive. Nous avons réalisé un prototype très concluant de cette fonctionnalité pour une espèce d'arbre, mais la contrainte majeure est l'acquisition des modèles tridimensionnels de toutes les espèces d'arbres pouvant être recommandées par notre système.

Enfin, l'évolution intuitivement la plus intéressante, serait d'étendre la zone d'action de la solution. Bien que conçue initialement pour la région de Montréal, l'architecture modulaire que nous avons mise en place facilite l'adaptation à d'autres zones urbaines. D'une part, en intégrant dans notre base de données les inventaires des arbres publics d'autres régions ou d'autres villes. Et d'autre part, en adaptant le modèle de classification et le système de recommandations à de nouvelles espèces d'arbres. On peut également imaginer une collaboration avec des projets internationaux de foresterie urbaine. Cette extension à de nouveaux territoires est un travail sur le long termes, qui ne peut se faire sans la collaboration des municipalités.

CONCLUSION

Dans ce mémoire nous avons présenté la conception et le développement d'une solution informatique, composée d'une application mobile citoyenne et d'une application dorsale, pour accompagner les personnes citoyennes dans leurs décisions de plantation d'arbres en espaces privés à l'aide de vision par ordinateur et système de recommandation. Face aux défis de l'urbanisation croissante et la nécessité de renforcer la résilience des forêts urbaines, notre travail a cherché à combler l'absence d'outils permettant d'intégrer les arbres privés dans la stratégie de maximisation de la diversité fonctionnelle du couvert forestier urbain.

La problématique initiale se déclinait en trois questions : comment recommander des espèces d'arbres adaptées aux contraintes de la zone de plantation et aux préférences des personnes utilisatrices tout en maximisant la résilience de la forêt urbaine, comment obtenir un inventaire fiable des arbres en espaces privés, et comment permettre aux personnes utilisatrices non expertes d'identifier les espèces d'arbres déjà présentes sur leur terrain.

La première partie de notre travail a été le développement d'un modèle de classification d'espèces d'arbres à partir de photos de leurs feuilles. En exploitant, par transfert d'apprentissage, l'architecture du modèle YOLO11m-cls et en l'adaptant à notre contexte spécifique par une stratégie d'ajustement fin, nous avons obtenu des résultats satisfaisant sur les 150 espèces les plus répandues dans la région de Montréal. Ces performances sont suffisantes pour l'usage visé, d'autant plus que les erreurs de classification se produisent majoritairement entre les espèces d'un même groupe fonctionnel, limitant ainsi leur impact sur les recommandations.

La seconde partie concerne le système de recommandation multicritère que nous avons mis en place. Contrairement aux guides de plantation traditionnels qui considèrent chaque arbre de manière isolée, notre algorithme prend en compte la diversité fonctionnelle alentour en s'appuyant sur les inventaires d'arbres publics agrégés par SylvCiT et sur la base de données collaborative des arbres privés qui se constitue à travers l'utilisation de l'application. Cela permet de proposer des espèces qui maximisent la diversité fonctionnelle, tout en respectant les contraintes du site de plantation et les préférences des personnes utilisatrices.

La troisième partie est l'application mobile elle-même, qui permet aux personnes utilisatrices d'interagir avec les modules précédemment cités, et de contribuer à l'enrichissement de la base de données des arbres

privés. SylvPriV participe à la construction d'une vision plus complète de la forêt urbaine qui ne se limite pas aux arbres publics. Son intégration à la plateforme SylvCiT renforce cette idée en lui permettant également d'utiliser cette nouvelle source de données pour ces recommandations.

Néanmoins, plusieurs limites ont été identifiées, comme la restriction à 150 espèces d'arbres, la sensibilité du modèle d'identification aux conditions de prise de vue des photos, la dépendance aux inventaires publics disponibles et la portée géographique actuellement bornée à la grande région de Montréal. La disponibilité exclusive sur Android est également une contrainte importante, en privant l'application d'une base de personnes utilisatrices potentielles conséquente.

Cependant, nous avons émis de nombreuses perspectives d'évolution. L'enrichissement continu de la base de données des arbres privés grâce aux contributions citoyennes, l'extension de l'identification des espèces à d'autres organes de l'arbre, le développement d'une version iOS, l'intégration des données d'Hydro-Québec sur les lignes électriques aériennes et les caractéristiques des arbres, l'ajout de nouvelles espèces d'arbres et l'extension à d'autres régions urbaines sont les principales pistes pour accroître la portée de l'application.

Enfin, notre travail démontre que la combinaison d'outils d'intelligence artificielle et de la science citoyenne peut contribuer efficacement à optimiser la diversité fonctionnelle des arbres urbains. La forêt urbaine se construit également avec les plantations en espaces privés, guidées par des recommandations éclairées, et participe ainsi plus efficacement à la résilience des villes face aux changements globaux.

BIBLIOGRAPHIE

Adomavicius, G. et Tuzhilin, A. (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*(6), 734–749.

Allen, B., Tamindael, L. E., Bickerton, S. et Cho, W. (2020). Does citizen coproduction lead to better urban services in smart cities projects ? an empirical study on e-participation in a mobile big data platform. *Gov. Inf. Q.*, *37*. https://doi.org/10.1016/j.giq.2019.101412

Alomar, K., Aysel, H. I. et Cai, X. (2023). Data augmentation in classification and segmentation : A survey and new strategies. *Journal of Imaging*, *9*(2), 46.

Balanya, S. A., Maronas, J. et Ramos, D. (2024). Adaptive temperature scaling for robust calibration of deep neural networks. *Neural Computing and Applications*, *36*(14), 8073–8095.

Balapour, A., Nikkhah, H. et Sabherwal, R. (2020). Mobile application security : Role of perceived privacy as the predictor of security perceptions. *Int. J. Inf. Manag.*, *52*, 102063. https://doi.org/10.1016/j.ijinfomgt.2019.102063

Bánki, O., Roskov, Y., Döring, M., Ower, G., Hernández Robles, D. R., Plata Corredor, C. A. *et al.* (2025). Catalogue of life (2025-11-16 xr). Consulté le 16 décembre 2025, https://doi.org/10.48580/dgvbl

Barré, P., Stöver, B. C., Müller, K. et Steinhage, V. (2017). Leafnet : A computer vision system for automatic plant species identification. *Ecol. Informatics*, *40*, 50–56. https://doi.org/10.1016/j.ecoinf.2017.05.005

Barth, S., Jong, M., Junger, M., Hartel, P. et Roppelt, J. C. (2019). Putting the privacy paradox to the test : Online privacy and security behaviors among users with technical knowledge, privacy awareness, and financial resources. *Telematics Informatics*, *41*, 55–69. https://doi.org/10.1016/j.tele.2019.03.003

Bastos, D., Fernández-Caballero, A., Pereira, A. et Rocha, N. (2022). Smart city applications to promote citizen participation in city management and governance : A systematic review. *Informatics*, *9*, 89. https://doi.org/10.3390/informatics9040089

Beillouin, D., Ben-Ari, T., Malézieux, E., Seufert, V. et Makowski, D. (2021). Positive but variable effects of crop diversification on biodiversity and ecosystem services. *Global change biology*, *27*(19), 4697–4710.

Belluau, M., Bouchard, É., Déziel, M., Mordacq, O., Messier, C. et Paquette, A. (2021). Tree trait task force (3tf)—tree functional trait database.

Bernett, J., Blumenthal, D. B., Grimm, D. G., Haselbeck, F., Joeres, R., Kalinina, O. V. et List, M. (2024). Guiding questions to avoid data leakage in biological machine learning applications. *Nature Methods*, *21*(8), 1444–1453.

Bi, Y., Xue, B., Mesejo, P., Cagnoni, S. et Zhang, M. (2022). A survey on evolutionary computation for computer vision and image analysis : Past, present, and future trends. *IEEE Transactions on Evolutionary Computation*, *27*, 5–25. https://doi.org/10.1109/tevc.2022.3220747

Biørn-Hansen, A., Rieger, C., Grønli, T.-M., Majchrzak, T. A. et Ghinea, G. (2020). An empirical investigation of performance overhead in cross-platform mobile development frameworks. *Empirical Software Engineering*, *25*(4), 2997–3040.

Bisen, D. (2020). Deep convolutional neural network based plant species recognition through features of leaf. *Multimedia Tools and Applications*, *80*, 6443 – 6456. https://doi.org/10.1007/s11042-020-10038-w

Bolund, P. et Hunhammar, S. (1999). Ecosystem services in urban areas. *Ecological Economics*, *29*, 293–301. https://doi.org/10.1016/s0921-8009(99)00013-0

Bonney, R., Cooper, C. B., Dickinson, J., Kelling, S., Phillips, T., Rosenberg, K. V. et Shirk, J. (2009). Citizen science : a developing tool for expanding science knowledge and scientific literacy. *BioScience*, *59*(11), 977–984.

Bonney, R., Shirk, J. L., Phillips, T. B., Wiggins, A., Ballard, H. L., Miller-Rushing, A. J. et Parrish, J. K. (2014). Next steps for citizen science. *Science*, *343*(6178), 1436–1437.

Bottou, L. (2012). Stochastic gradient descent tricks. Dans *Neural networks : tricks of the trade : second edition* (p. 421–436). Springer.

Bouzid (2016). Cours de biologie végétale (1ère lmd) : Chapitre 4 — morphologie des organes végétaux. Notes de cours (PDF), Université Frères Mentouri Constantine 1. Consulté le 27 décembre 2025., https://fac.umc.edu.dz/snv/faculte/tc/17.pdf

Bowler, D. E., Buyung-Ali, L., Knight, T. M. et Pullin, A. S. (2010). Urban greening to cool towns and cities : A systematic review of the empirical evidence. *Landscape and urban planning*, *97*(3), 147–155.

Brovelli, M., Minghini, M. et Zamboni, G. (2016). Public participation in gis via mobile applications. *Isprs Journal of Photogrammetry and Remote Sensing*, *114*, 306–315. https://doi.org/10.1016/j.isprsjprs.2015.04.002

Burke, R. (2000). A case-based reasoning approach to collaborative filtering. Dans *European Workshop on Advances in Case-Based Reasoning*, (p. 370–379). Springer.

Burke, R. (2002). Hybrid recommender systems : Survey and experiments. *User modeling and user-adapted interaction*, *12*(4), 331–370.

Cameron, R. W., Blanuša, T., Taylor, J. E., Salisbury, A., Halstead, A. J., Henricot, B. et Thompson, K. (2012). The domestic garden–its contribution to urban green infrastructure. *Urban forestry & urban greening*, *11*(2), 129–137.

Capponi, A., Fiandrino, C., Kantarci, B., Foschini, L., Kliazovich, D. et Bouvry, P. (2019). A survey on mobile crowdsensing systems : Challenges, solutions, and opportunities. *IEEE Communications Surveys & Tutorials*, *21*, 2419–2465. https://doi.org/10.1109/comst.2019.2914030

Cea, L. et Costabile, P. (2022). Flood risk in urban areas : Modelling, management and adaptation to climate change. a review. *Hydrology*. https://doi.org/10.3390/hydrology9030050

Cerutti, G., Tougne, L., Mille, J., Vacavant, A. et Coquin, D. (2013). Understanding leaves in natural images - a model-based approach for tree species identification. *Comput. Vis. Image Underst.*, *117*, 1482–1501. https://doi.org/10.1016/j.cviu.2013.07.003

Chitwood, D. H. et Sinha, N. R. (2016). Evolutionary and environmental forces sculpting leaf development. *Current Biology*, *26*(7), R297–R306.

Choi, K., Wang, Y., Sparks, B. et Choi, S. (2021). Privacy or security : Does it matter for continued use intention of travel applications ? *Cornell Hospitality Quarterly*, *64*, 267 – 282. https://doi.org/10.1177/19389655211066834

Christin, D. (2016). Privacy in mobile participatory sensing. *Journal of Systems and Software*, *116*, 57–68. https://doi.org/10.1016/j.jss.2015.03.067

Degirmenci, K. (2020). Mobile users' information privacy concerns and the role of app permission requests. *Int. J. Inf. Manag.*, *50*, 261–272. https://doi.org/10.1016/j.ijinfomgt.2019.05.010

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. et Fei-Fei, L. (2009). Imagenet : A large-scale hierarchical image database. Dans *2009 IEEE conference on computer vision and pattern recognition*, (p. 248–255). Ieee.

Dickinson, J. L., Zuckerberg, B. et Bonter, D. N. (2010). Citizen science as an ecological research tool : challenges and benefits. *Annual review of ecology, evolution, and systematics*, *41*(1), 149–172.

Dubey, S. R., Singh, S. K. et Chaudhuri, B. B. (2022). Activation functions in deep learning : A comprehensive survey and benchmark. *Neurocomputing*, *503*, 92–108.

Edwards, J. L., Lane, M. A. et Nielsen, E. S. (2000). Interoperability of biodiversity databases : biodiversity information on every desktop. *Science*, *289*(5488), 2312–2314.

Ertiö, T. (2015). Participatory apps for urban planning—space for improvement. *Planning Practice & Research*, *30*, 303 – 321. https://doi.org/10.1080/02697459.2015.1052942

Escobedo, F. J., Giannico, V., Jim, C., Sanesi, G. et Lafortezza, R. (2019). Urban forests, ecosystem services, green infrastructure and nature-based solutions : Nexus or evolving metaphors ? *Urban Forestry & Urban Greening*, *37*, 3–12. Green Infrastructures : Nature Based Solutions for sustainable and resilient cities, https://doi.org/10.1016/j.ufug.2018.02.011

Esperon-Rodriguez, M., Arndt, S., Asibey, M. O., Augustinus, B. A., Bach, A., Ballinas, M., Barradas, V. L., Barton, D. N., Bauhus, J., Beaumont, L. *et al.* (2025). Urban forests as essential infrastructure for climate resilience and biodiversity : A call to policymakers. *Plants, people, planet*.

Evans, D., Falagán, N., Hardman, C., Kourmpetli, S., Liu, L., Mead, B. et Davies, J. A. C. (2022). Ecosystem service delivery by urban agriculture and green infrastructure – a systematic review. *Ecosystem Services*. https://doi.org/10.1016/j.ecoser.2022.101405

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, Y., Tang, J. et Yin, D. (2019). Graph neural networks for social recommendation. *The World Wide Web Conference*. https://doi.org/10.1145/3308558.3313488

Fan, X. et Truong, P. (2022). An introduction to convolutional neural network (cnn). https://medium.com/sfu-cspmp/an-introduction-to-convolutional-neural-network-cnn-207cdb53db97. Consulté le 27 décembre 2025.

Feng, B., Zhang, Y. et Bourke, R. (2021). Urbanization impacts on flood risks based on urban growth data and coupled flood models. *Natural Hazards*, *106*, 613 – 627. https://doi.org/10.1007/s11069-020-04480-0

Ferreira, V., Barreira, A., Loures, L., Antunes, D. et Panagopoulos, T. (2020). Stakeholders' engagement on nature-based solutions : A systematic literature review. *Sustainability*, *12*, 640. https://doi.org/10.3390/su12020640

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. University of California, Irvine.

Filho, W., Icaza, L. E., Neht, A., Kļaviņš, M. et Morgan, E. (2018). Coping with the impacts of urban heat islands. a literature based study on understanding urban heat vulnerability and the need for resilience in cities in a global climate change context. *Journal of Cleaner Production*, *171*, 1140–1149. https://doi.org/10.1016/j.jclepro.2017.10.086

Francini, A., Romano, D., Toscano, S. et Ferrante, A. (2022). The contribution of ornamental plants to urban ecosystem services. *Earth*, *3*(4), 1258–1274.

Frantzeskaki, N. (2019). Seven lessons for planning nature-based solutions in cities. *Environmental Science & Policy*, *93*, 101–111. https://doi.org/10.1016/j.envsci.2018.12.033

Furini, M., Mirri, S., Montangero, M. et Prandi, C. (2020). Privacy perception when using smartphone applications. *Mobile Networks and Applications*, *25*, 1055 – 1061. https://doi.org/10.1007/s11036-020-01529-z

Gao, C., Wang, X., He, X. et Li, Y. (2022). Graph neural networks for recommender system. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. https://doi.org/10.1145/3488560.3501396

Garnier, E. et Navas, M.-L. (2012). A trait-based approach to comparative functional plant ecology : concepts, methods and applications for agroecology. a review. *Agronomy for Sustainable Development*, *32*(2), 365–399.

GBIF Secretariat (2023). Gbif species page. https://www.gbif.org/species. GBIF Backbone Taxonomy. https://doi.org/10.15468/39omei. Consulté le 17 décembre 2025.

George, K., Ziska, L., Bunce, J. et Quebedeaux, B. (2007). Elevated atmospheric co2 concentration and temperature across an urban–rural transect. *Atmospheric Environment*, *41*, 7654–7665. https://doi.org/10.1016/j.atmosenv.2007.08.018

Goddard, M. A., Dougill, A. J. et Benton, T. G. (2010). Scaling up from gardens : biodiversity conservation in urban environments. *Trends in ecology & evolution*, *25*(2), 90–98.

Goëau, H., Bonnet, P., Joly, A., Bakić, V., Barbe, J., Yahiaoui, I., Selmi, S., Carré, J., Barthélémy, D., Boujemaa, N. *et al.* (2013). Pl@ ntnet mobile app. Dans *Proceedings of the 21st ACM international conference on Multimedia*, (p. 423–424).

Hadamard, J. (1899). Théorème sur les séries entières. *Acta Mathematica*, *22*(1), 55.

Halevy, A., Norvig, P. et Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE intelligent systems*, *24*(2), 8–12.

Hao, Y., Tang, Z., Tian, Y., Zhang, Y. et Zhou, Z. (2024). Adamw. Cornell University Computational Optimization Open Textbook – Optimization Wiki. Page web (Wiki). Dernière modification le 12 décembre 2024. Consulté le 25 décembre 2025., https://optimization.cbe.cornell.edu/index.php?title=AdamW&oldid=7088

Hart, A. G., Bosley, H., Hooper, C., Perry, J., Sellors-Moore, J., Moore, O. et Goodenough, A. E. (2023). Assessing the accuracy of free automated plant identification applications. *People and Nature*, *5*(3), 929–937.

He, H. et Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263–1284.

He, K., Zhang, X., Ren, S. et Sun, J. (2015). Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. Dans *Proceedings of the IEEE international conference on computer vision*, (p. 1026–1034).

Hickey, L. J. (1973). Classification of the architecture of dicotyledonous leaves. *American journal of botany*, *60*(1), 17–33.

Hong, C., Burney, J. A., Pongratz, J., Nabel, J. E., Mueller, N. D., Jackson, R. B. et Davis, S. J. (2021). Global and regional drivers of land-use emissions in 1961–2017. *Nature*, *589*(7843), 554–561.

Hou, J., Arpan, L. M., Wu, Y., Feiock, R., Ozguven, E. et Arghandeh, R. (2020). The road toward smart cities : A study of citizens' acceptance of mobile applications for city services. *Energies*. https://doi.org/10.3390/en13102496

Hua, F., Bruijnzeel, L. A., Meli, P., Martin, P. A., Zhang, J., Nakagawa, S., Miao, X., Wang, W., McEvoy, C., Peña-Arancibia, J. L. *et al.* (2022). The biodiversity and ecosystem service contributions and trade-offs of forest restoration approaches. *Science*, *376*(6595), 839–844.

Hutt-Taylor, K. et Ziter, C. D. (2022). Private trees contribute uniquely to urban forest diversity, structure and service-based traits. *Urban Forestry & Urban Greening*, *78*, 127760.

iNaturalist contributors (2025). inaturalist research-grade observations. Jeu de données d'occurrences (GBIF). iNaturalist.org. DOI : 10.15468/ab3s5x. Consulté via GBIF le 18 décembre 2025., https://doi.org/10.15468/ab3s5x

Jabbar, H. K., Hamoodi, M. N. et Al-hameedawi, A. (2023). Urban heat islands : a review of contributing factors, effects and data. *IOP Conference Series : Earth and Environmental Science*, *1129*. https://doi.org/10.1088/1755-1315/1129/1/012038

Jeon, W.-S. et Rhee, S. (2017). Plant leaf recognition using a convolution neural network. *Int. J. Fuzzy Log. Intell. Syst.*, *17*, 26–34. https://doi.org/10.5391/ijfis.2017.17.1.26

Jiang, P., Ergu, D., Liu, F., Cai, Y. et Ma, B. (2022). A review of yolo algorithm developments. *Procedia computer science*, *199*, 1066–1073.

Jocher, G. et Qiu, J. (2024). Ultralytics yolo11. https://github.com/ultralytics/ultralytics

Jozani, M. M., Ayaburi, E., Ko, M. S. et Choo, K.-K. R. (2020). Privacy concerns and benefits of engagement with social media-enabled apps : A privacy calculus perspective. *Comput. Hum. Behav.*, *107*, 106260. https://doi.org/10.1016/j.chb.2020.106260

Karagulian, F., Belis, C., Dora, C., Prüss-Ustün, A., Bonjour, S., Adair-Rohani, H. et Amann, M. (2015). Contributions to cities' ambient particulate matter (pm) : a systematic review of local source contributions at global level. *Atmospheric Environment*, *120*, 475–483. https://doi.org/10.1016/j.atmosenv.2015.08.087

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M. et Tang, P. T. P. (2016). On large-batch training for deep learning : Generalization gap and sharp minima. *arXiv preprint arXiv :1609.04836*.

Khan, A., Sohail, A., Zahoora, U. et Qureshi, A. S. (2019). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, *53*, 5455 – 5516. https://doi.org/10.1007/s10462-020-09825-6

Khanam, R. et Hussain, M. (2024). Yolov11 : An overview of the key architectural enhancements. *arXiv preprint arXiv :2410.17725*.

Kingma, D. P. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*. https://doi.org/10.48550/arXiv.1412.6980

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A. *et al.* (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, *114*(13), 3521–3526.

Kondo, M., Fluehr, J. M., McKeon, T. et Branas, C. (2018). Urban green space and its impact on human health. *International Journal of Environmental Research and Public Health*, *15*. https://doi.org/10.3390/ijerph15030445

Kremer, A., Dupouey, J., Deans, J., Cottrell, J., Csaikl, U., Finkeldey, R., Espinel, S., Jensen, J., Kleinschmit, J., van Dam, B., Ducousso, A., Forrest, I., de Heredia, U. L., Lowe, A., Tutkova, M., Munro, R., Steinhoff, S. et Badeau, V. (2002). Leaf morphological differentiation between quercus robur and quercus petraea is stable across western european mixed oak stands. *Annals of Forest Science*, *59*, 777–787. https://doi.org/10.1051/forest:2002065

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

Kumar, N., Belhumeur, P., Biswas, A., Jacobs, D. W., Kress, W., Lopez, I. C. et Soares, J. V. B. (2012). Leafsnap : A computer vision system for automatic plant species identification. Dans *Computer Vision – ECCV 2012*, (p. 502–516). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33709-3_36

Laforest-Lapointe, I., Messier, C. et Kembel, S. W. (2017). Tree leaf bacterial community structure and diversity differ along a gradient of urban intensity. *MSystems*, *2*(6), 10–1128.

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

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P. et Finn, C. (2022). Surgical fine-tuning improves adaptation to distribution shifts. *ArXiv*, *abs/2210.11466*. https://doi.org/10.48550/arxiv.2210.11466

Li, D. et Zhang, H. (2021). Improved regularization and robustness for fine-tuning in neural networks. *Advances in Neural Information Processing Systems*, *34*, 27249–27262.

Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R. et Soatto, S. (2020a). Rethinking the hyperparameters for fine-tuning. *arXiv preprint arXiv :2002.11770*.

Li, J., Chen, X., Niklas, K., Sun, J., Wang, Z., Zhong, Q., Hu, D. et Cheng, D. (2021). A whole-plant economics spectrum including bark functional traits for 59 subtropical woody plant species. *Journal of Ecology*, *110*, 248 – 261. https://doi.org/10.1111/1365-2745.13800

Li, Z., Liu, F., Yang, W., Peng, S. et Zhou, J. (2020b). A survey of convolutional neural networks : Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*, 6999–7019. https://doi.org/10.1109/tnnls.2021.3084827

Liu, Y., Gao, Y. et Yin, W. (2020). An improved analysis of stochastic gradient descent with momentum. *Advances in Neural Information Processing Systems*, *33*, 18261–18271.

Liu, Y., Li, Y., Song, J., Zhang, R., Yan, Y.-M., Wang, Y. et Du, F. (2018). Geometric morphometric analyses of leaf shapes in two sympatric chinese oaks : Quercus dentata thunberg and quercus aliena blume (fagaceae). *Annals of Forest Science*, *75*. https://doi.org/10.1007/s13595-018-0770-2

Loshchilov, I. et Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv :1711.05101*.

Lucchese, C., Muntean, C. I., Perego, R., Silvestri, F., Vahabi, H. et Venturini, R. (2021). Recommender systems. *Wirtschaftsinf.*, *39*, 401–404. https://doi.org/10.1007/978-1-4899-7687-1_964

Manakitsa, N., Maraslidis, G. S., Moysis, L. et Fragulis, G. (2024). A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision. *Technologies*. https://doi.org/10.3390/technologies12020015

Mapbox (2025). Geojson.io. Outil web. Consulté le 18 décembre 2025. Disponible : https://geojson.io/, https://geojson.io/

Martín-Vide, J., Sarricolea, P. et Moreno-García, M. (2015). On the definition of urban heat island intensity : the “rural” reference. *Frontiers in Earth Science*, *3*, 24. https://doi.org/10.3389/feart.2015.00024

Matheson, C. A. (2014). inaturalist. *Reference Reviews*, *28*(8), 36–38.

Maynard, D., Bialic-Murphy, L., Zohner, C., Averill, C., van den Hoogen, J., Ma, H., Mo, L., Smith, G. R., Acosta, A., Aubin, I., Berenguer, E., Boonman, C. C. F., Catford, J., Cerabolini, B., Dias, A. S., González-Melo, A., Hietz, P., Lusk, C., Mori, A. S., … Crowther, T. (2022). Global relationships in tree functional traits. *Nature Communications*, *13*. https://doi.org/10.1038/s41467-022-30888-2

Mehra, A., Zhang, Y. et Hamm, J. (2024). Understanding the transferability of representations via task-relatedness. *Advances in Neural Information Processing Systems*, *37*, 116513–116546.

Merkel, D. *et al.* (2014). Docker : lightweight linux containers for consistent development and deployment. *Linux j*, *239*(2), 2.

Mohajerani, A., Bakaric, J. et Jeffrey-Bailey, T. (2017). The urban heat island effect, its causes, and mitigation, with reference to the thermal properties of asphalt concrete. *Journal of environmental management*, *197*, 522–538. https://doi.org/10.1016/j.jenvman.2017.03.095

Nevers, C., Carmeliet, J., Strebel, D., Wood, S., Kubilay, A. et Derome, D. (2025). Understanding cooling potential of urban trees in a typical north america neighborhood. *Building and Environment*, *282*, 113303.

Neyns, R. et Canters, F. (2022). Mapping of urban vegetation with high-resolution remote sensing : A review. *Remote. Sens.*, *14*, 1031. https://doi.org/10.3390/rs14041031

Nguyen, P., Astell-Burt, T., Rahimi-Ardabili, H. et Feng, X. (2021). Green space quality and health : A systematic review. *International Journal of Environmental Research and Public Health*, *18*. https://doi.org/10.3390/ijerph182111028

Nowak, D., Crane, D., Stevens, J., Hoehn, R., Walton, J. et Bond, J. (2008). A ground-based method of assessing urban forest structure and ecosystem services. *Arboriculture & Urban Forestry*, *34*(6), 347–358.

Nowak, D. J., Crane, D. E. et Stevens, J. C. (2006). Air pollution removal by urban trees and shrubs in the united states. *Urban forestry & urban greening*, *4*(3-4), 115–123.

Nowak, D. J., Maco, S. et Binkley, M. (2018). i-tree : Global tools to assess tree benefits and risks to improve forest management. *Arboricultural Consultant. 51 (4) : 10-13.*, *51*(4), 10–13.

Nyelele, C. et Kroll, C. (2021). A multi-objective decision support framework to prioritize tree planting locations in urban areas. *Landscape and Urban Planning*, *214*, 104172. https://doi.org/10.1016/j.landurbplan.2021.104172

O'shea, K. et Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv :1511.08458*.

Ouyang, D., He, S., Zhan, J., Guo, H., Huang, Z., Luo, M. et Zhang, G.-L. (2023). Efficient multi-scale attention module with cross-spatial learning. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. https://doi.org/10.1109/icassp49357.2023.10096516

Pal, K. K. et Sudeep, K. (2016). Preprocessing for image classification by convolutional neural networks. Dans *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, (p. 1778–1781). IEEE.

Pan, J.-X. et Fang, K.-T. (2002). Maximum likelihood estimation. Dans *Growth curve models and statistical diagnostics* (p. 77–158). Springer.

Pataki, D., Alberti, M., Cadenasso, M., Felson, A., Mcdonnell, M., Pincetl, S., Pouyat, R., Setälä, H. et Whitlow, T. (2021). The benefits and limits of urban tree planting for environmental and human health. *Frontiers in Ecology and Evolution*, *9*. https://doi.org/10.3389/fevo.2021.603757

Paudel, S. et States, S. (2023). Urban green spaces and sustainability : exploring the ecosystem services and disservices of grassy lawns versus floral meadows. *Urban Forestry & Urban Greening*. https://doi.org/10.1016/j.ufug.2023.127932

Pentina, I., Zhang, L., Bata, H. et Chen, Y. (2016). Exploring privacy paradox in information-sensitive mobile app adoption : A cross-cultural comparison. *Comput. Hum. Behav.*, *65*, 409–419. https://doi.org/10.1016/j.chb.2016.09.005

Piracha, A. et Chaudhary, M. T. A. (2022). Urban air pollution, urban heat island and human health : A review of the literature. *Sustainability*. https://doi.org/10.3390/su14159234

Raschka, S. et Mirjalili, V. (2019). *Python machine learning : Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt publishing ltd.

Rawat, W. et Wang, Z. (2017). Deep convolutional neural networks for image classification : A comprehensive review. *Neural Computation*, *29*, 2352–2449. https://doi.org/10.1162/neco_a_00990

Redmon, J., Divvala, S., Girshick, R. B. et Farhadi, A. (2015). You only look once : Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/cvpr.2016.91

Robitaille, É. et Douyon, C. (2025). Using the 3-30-300 indicator to evaluate green space accessibility and inequalities : A case study of montreal, canada. *Geographies*, *5*(1), 6.

Roman, L. A., Conway, T. M., Eisenman, T., Koeser, A. K., Barona, C. O., Locke, D., Jenerette, G. D., Östberg, J. et Vogt, J. (2020). Beyond 'trees are good' : Disservices, management costs, and tradeoffs in urban forestry. *Ambio*, *50*, 615 – 630. https://doi.org/10.1007/s13280-020-01396-8

Roman, L. A., Scharenbroch, B. C., Östberg, J. P., Mueller, L. S., Henning, J. G., Koeser, A. K., Sanders, J. R., Betz, D. R. et Jordan, R. C. (2017). Data quality in citizen science urban tree inventories. *Urban Forestry & Urban Greening*, *22*, 124–135.

Rumelhart, D. E., Hinton, G. E. et Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. *et al.* (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), 211–252.

Sarwar, B., Karypis, G., Konstan, J. et Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Dans *Proceedings of the 10th international conference on World Wide Web*, (p. 285–295).

Schulz, B. (2025). Identification of trees & shrubs in winter using buds and twigs, second edition. isbn 9781842468340.

Seto, K. C., Güneralp, B. et Hutyra, L. R. (2012). Global forecasts of urban expansion to 2030 and direct impacts on biodiversity and carbon pools. *Proceedings of the National Academy of Sciences*, *109*(40), 16083–16088.

Situ, Z., Teng, S., Feng, W., Zhong, Q., Chen, G., Su, J. et Zhou, Q. (2023). A transfer learning-based yolo network for sewer defect detection in comparison to classic object detection methods. *Developments in the Built Environment*. https://doi.org/10.1016/j.dibe.2023.100191

Sokolova, M. et Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427–437.

St-Denis, A., Maure, F., Belbahar, R., Delagrange, S., Handa, I., Kneeshaw, D., Paquette, A., Nicol, M., Meurs, M. et Messier, C. (2024). An urban forest diversification software to improve resilience to global change. *Arboriculture & Urban Forestry (AUF)*, *50*(1), 76–91.

StatCounter GlobalStats (2025a). Mobile operating system market share in canada – december 2025. Données en ligne (StatCounter GlobalStats). Consulté le 25 décembre 2025., https://gs.statcounter.com/os-market-share/mobile/canada

StatCounter GlobalStats (2025b). Mobile operating system market share worldwide – december 2025. Données en ligne (StatCounter GlobalStats). Consulté le 25 décembre 2025., https://gs.statcounter.com/os-market-share/mobile/worldwide

Tan, J., Chang, S.-W., Abdul-kareem, S., Yap, H. et Yong, K. (2020). Deep learning for plant species classification using leaf vein morphometric. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *17*, 82–90. https://doi.org/10.1109/tcbb.2018.2848653

Tang, J., Akram, U. et Shi, W. (2020). Why people need privacy ? the role of privacy fatigue in app users' intention to disclose privacy : based on personality traits. *J. Enterp. Inf. Manag.*, *34*, 1097–1120. https://doi.org/10.1108/jeim-03-2020-0088

Thayyib, P. V., Mamilla, R., Khan, M., Fatima, H., Asim, M., Anwar, I., Shamsudheen, M. K. et Khan, M. A. (2023). State-of-the-art of artificial intelligence and big data analytics reviews in five different domains : A bibliometric summary. *Sustainability*. https://doi.org/10.3390/su15054026

The Global Biodiversity Information Facility (2025). What is gbif ? https://www.gbif.org/what-is-gbif. Consulté le 17 décembre 2025.

Tsukaya, H. (2018). Leaf shape diversity with an emphasis on leaf contour variation, developmental background, and adaptation. Dans *Seminars in cell & developmental biology*, Vol. 79, (p. 48–57). Elsevier.

Turner-Skoff, J. B. et Cavender, N. (2019). The benefits of trees for livable and sustainable communities. *PLANTS, PEOPLE, PLANET*, *1*(4), 323–335. https://doi.org/10.1002/ppp3.39

Ultralytics. Code source yolo11. Fichier de configuration YAML (YOLO11 classification), https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/models/11/yolo11-cls.yaml

Ultralytics. Documentation yolo11. https://docs.ultralytics.com/fr/models/yolo11/

Vapnik, V. (1991). Principles of risk minimization for learning theory. *Advances in neural information processing systems*, *4*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. et Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Ville de Montréal (2020). Arbres publics de la ville de montréal. Jeu de données, Données Québec. Consulté le 18 décembre 2025., https://www.donneesquebec.ca/recherche/dataset/vmtl-arbres

Violle, C., Navas, M.-L., Vile, D., Kazakou, E., Fortunel, C., Hummel, I. et Garnier, E. (2007). Let the concept of trait be functional ! *Oikos*, *116*(5), 882–892.

Viscosi, V. et Cardini, A. (2011). Leaf morphology, taxonomy and geometric morphometrics : A simplified protocol for beginners. *PLoS ONE*, *6*. https://doi.org/10.1371/journal.pone.0025630

Vleminckx, J., Fortunel, C., Valverde-Barrantes, O., Paine, C. E. T., Engel, J., Pétronelli, P., Dourdain, A., Guevara, J., Béroujon, S. et Baraloto, C. (2021). Resolving whole-plant economics from leaf, stem and root traits of 1467 amazonian tree species. *Oikos*. https://doi.org/10.1111/oik.08284

Voulodimos, A., Doulamis, N., Doulamis, A. et Protopapadakis, E. (2018). Deep learning for computer vision : A brief review. *Computational intelligence and neuroscience*, *2018*(1), 7068349.

Wan, C., Shen, G. et Choi, S. (2021). Underlying relationships between public urban green spaces and social cohesion : A systematic literature review. *City, culture and society*, *24*, 100383. https://doi.org/10.1016/j.ccs.2021.100383

Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W. et Yeh, I.-H. (2020). Cspnet : A new backbone that can enhance learning capability of cnn. Dans *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, (p. 390–391).

Welch, H. J. (1991). Classification and nomenclature. Dans *The Conifer Manual : Volume 1* (p. 44–61). Springer.

Werbin, Z. R., Heidari, L., Buckley, S., Brochu, P., Butler, L. J., Connolly, C., Bloemendaal, L. J. H., McCabe, T., Miller, T. et Hutyra, L. (2020). A tree-planting decision support tool for urban heat mitigation. *PLoS ONE*, *15*. https://doi.org/10.1371/journal.pone.0224959

Winston, J. E. (2018). Twenty-first century biological nomenclature—the enduring power of names. *Integrative and comparative biology*, *58*(6), 1122–1131.

Yosinski, J., Clune, J., Bengio, Y. et Lipson, H. (2014). How transferable are features in deep neural networks ? *Advances in neural information processing systems*, *27*.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K. et Hsieh, C.-J. (2019). Large batch optimization for deep learning : Training bert in 76 minutes. *arXiv preprint arXiv :1904.00962*.

Zeghad, N (2018). Chapitre iv : Morphologie des organes végétaux (partie i). Document de cours (PDF), Université Frères Mentouri Constantine 1. Consulté le 27 décembre 2025., https://fac.umc.edu.dz/snv/faculte/tc/2017/cours/Chapitre%20IV%20partie%20I.pdf

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M. et Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artif. Intell. Rev.*, *57*, 99. https://doi.org/10.1007/s10462-024-10721-6

Zhao, Z.-Q., Ma, L.-H., Cheung, Y., Wu, X., Tang, Y. et Chen, C.-L. P. (2015). Apleaf : An efficient android-based plant leaf identification system. *Neurocomputing*, *151*, 1112–1119. https://doi.org/10.1016/j.neucom.2014.02.077

Zheng, Q., Yang, M., Tian, X., Jiang, N. et Wang, D. (2020). A full stage data augmentation method in deep convolutional neural network for natural image classification. *Discrete Dynamics in Nature and Society*, *2020*(1), 4706576.

Zhu, X., Lu, K.-F., Peng, Z., di He, H. et Xu, S. (2021). Spatiotemporal variations of carbon dioxide (co2) at urban neighborhood scale : Characterization of distribution patterns and contributions of emission sources. *Sustainable Cities and Society*. https://doi.org/10.1016/j.scs.2021.103646