SEGMENTATION SÉMANTIQUE PAR RÉSEAUX NEURONAUX DES ESPÈCES INDÉSIRABLES DANS LA CULTURE DU BLEUET SAUVAGE

Mémoire présenté comme exigence partielle
de la maîtrise en technologie de l'information

Par Jean-François Grenier

Octobre 2022

REMERCIEMENTS

Merci à mon directeur, M. Daniel Lemire, pour son appui et sa disponibilité tout au long de mes travaux de maîtrise. Sa rigueur, son réalisme et sa passion ont grandement contribué à ce mémoire.

Merci à lui également de m'avoir convaincu de communiquer sur ces travaux en dehors de la rédaction de ce mémoire. Cette ouverture sur la communauté a apporté des idées très importantes à la suite de mes propres travaux et je l'espère, a pu aider d'autres à relever leurs défis.

Ma compagne de vie m'a toujours inspiré par sa détermination et sa motivation. Merci Rubis d'avoir été un modèle et un support tout au long de mon parcours.

Merci également à tous mes collègues qui ont tricoté autour de mes absences afin de me permettre d'accomplir ce travail, particulièrement lors de ces années très spéciales que furent 2020 et 2021.

RÉSUMÉ

L'agriculture est souvent au coeur des discussions environnementales. L'industrialisation de la production alimentaire passe en grande partie par des produits chimiques qui ont des impacts en dehors des champs. L'application sélective de ces produits est souvent difficile et il ne serait pas rentable d'effectuer cette application manuellement. L'intelligence artificielle a beaucoup progressé dans le traitement d'images au cours des dernières années, laissant entrevoir la possibilité d'accomplir cette application sélective en temps réel lors du passage dans les champs.

L'entraînement de ces réseaux neuronaux nécessite un grand volume de données réelles, données qui n'existent pas sans de grands efforts d'acquisition et de classification préalable. Dans le cas du bleuet sauvage, comme plusieurs cultures régionales, il n'existe pas de données facilement accessibles sur lesquelles travailler. Il faut donc trouver des approches différentes.

À l'aide d'un moteur graphique habituellement utilisé pour des jeux vidéos (Unreal Engine), nous avons créé des environnements virtuels qui ont permis l'entraînement de réseaux neuronaux sous Tensorflow/Keras. Cette approche a permis de réduire considérablement l'effort d'acquisition et de classification de données réelles. Le travail s'est entièrement déroulé sur un ordinateur de bureau, sans nécessiter l'accès à des équipements spécialisés ou de grandes ressources financières.

L'entraînement sur les données virtuelles seulement a généré une précision de 67%, ce qui n'est pas très impressionnant considérant que nous avions une classe majoritaire à 72% dans notre jeu de données. Le réalisme des images et sa correspondance au cas réel pourraient clairement être améliorés. Cependant, l'entraînement incluant une petite quantité de données réelles a augmenté la précision à 88%. Nous ne sommes toutefois pas encore au niveau observé dans d'autres travaux de recherches similaires (95+%). Les données virtuelles ont possiblement un plafond de performance maximale inférieur aux données réelles. Pour un usage cartographique, cette précision combinée à l'imagerie par drone permettrait déjà de créer des cartes utiles de classification des surfaces en cultures.

L'obtention de tels résultats suite à nos efforts indique que l'approche pourrait mener à un système d'application en temps réel. Le réalisme des simulations étant relié à la qualité de l'entraînement, le recours à des spécialistes en design graphique augmenterait possiblement la précision du réseau obtenu.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

SIGLES ET DÉFINITIONS

RN - Réseau neuronal / Réseau de neurones

Un réseau de neurones artificiels, ou réseau neuronal artificiel, est un système d'apprentissage machine inspiré du fonctionnement des neurones biologiques sans toutefois les copier entièrement. Chaque neurone du réseau applique un seuil mathématique simple et la mise en groupe de plusieurs de ces neurones permet l'apparition de capacités émergentes, comme la classification de données ou d'images. À la manière des neurones biologiques, le fonctionnement de ces réseaux reste cependant opaque à l'observateur, le fonctionnement de cette capacité n'étant pas exprimé dans le réseau sous une forme compréhensible.

JD - Jeu de données

Un jeu de données (en anglais « dataset » ou « data set ») est un ensemble de valeurs (données, images, séquences audios, etc...) où chaque valeur est associée à une variable (ou attribut) et à une observation. Par exemple, dans ce mémoire, les jeux de données sont composés d'images accompagnées d'une classification de chaque pixel la composant.

RNC - Réseaux neuronal convolutif

Les réseaux neuronaux convolutifs s'inspirent du fonctionnement du cortex visuel pour pré-traiter l'information visuelle en balayant des zones au lieu d'y aller pixel par pixel. Cette approche réduit l'importance de la position des éléments et permet de considérablement réduire le nombre de paramètres du réseau pour

une performance similaire. Cette méthode est particulièrement efficace et est devenue la base de la majorité des réseaux traitant de l'imagerie. Ils ont, comme les réseaux neuronaux conventionnels, la capacité d'extraire automatiquement des caractéristiques propres à l'application.

Matrice de confusion

En apprentissage automatique supervisé, la matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée. La cellule ligne L, colonne C contient le nombre d'éléments de la classe réelle L qui ont été estimés comme appartenant à la classe C1.

Elle montre rapidement si un système parvient à classifier correctement le jeu de données et d'identifier individuellement les classes mal apprises par le réseau [1].

1. https ://fr.wikipedia.org/wiki/Matrice_de_confusion

INTRODUCTION

Au cours des dernières décennies, les bonds prodigieux en productivité agricole ont eu lieu en grande partie grâce à des méthodes incompatibles avec le développement durable. Un des principaux éléments de cette situation est l'utilisation massive de pesticides.

En réponse à ces questions environnementales mais aussi par souci économique, le domaine de « l'agriculture de précision » se développe. Il s'agit d'un amalgame de processus, outils et technologies qui permettent d'exploiter de façon plus rentable et plus saine les terres agricoles. L'acquisition d'information, le traitement des données et l'optimisation des processus à l'aide de ces connaissances sont les moteurs de cette nouvelle gestion.

Dans ces travaux nous traiterons d'un point spécifique de l'agriculture de précision, soit le traitement du contenu d'images. Notre environnement d'essai sera l'industrie du bleuet sauvage, une culture régionale du Nord-Est de l'Amérique.

Le traitement d'images via la segmentation sémantique

Le traitement d'information visuelle est un exemple de données peu structurées dans lequel les approches classiques de l'apprentissage machine ont beaucoup de difficultés. L'apprentissage profond basé sur les réseaux neuronaux s'avère excellent dans ces situations et a bouleversé le domaine dans la dernière décennie.

La segmentation sémantique est l'assignation d'une classe à chaque pixel d'une image afin de cerner précisément les éléments qui la composent. Ce procédé diffère

de la détection d'objets qui se limite à identifier la position approximative des éléments dans l'image. Bien que plus difficile, ce niveau de précision a ses avantages. Par exemple, lors de l'application sélective d'herbicide, la proximité des plantes est un enjeu important. Le simple rectangle autour d'une cible probable n'est plus suffisant car il peut mener à une application sur les mauvaises espèces.

Les petites cultures et l'accès aux données d'entraînement

Le problème de l'apprentissage profond réside cependant dans son besoin d'un énorme volume de données d'entraînement pour arriver à des résultats utilisables. L'acquisition d'un grand volume de ces images nécessite beaucoup d'effort et il est logique que les grandes cultures soient considérées prioritaires dans l'esprit des chercheurs et des industriels. Plusieurs industries régionales intéressent moins les chercheurs et les grandes industries par leur petite taille. Le grand défi de ces cultures est donc la disponibilité des données d'entraînement.

Alternatives aux grands jeux d'images réelles

Il est crucial d'explorer les alternatives à ce long et coûteux processus d'acquisition d'images réelles si l'on veut appliquer l'apprentissage profond à ces plus petites cultures. Nous aborderons donc la création d'images de synthèse, le transfert de domaines lors de l'entraînement du réseau ainsi que l'acquisition et la maximisation d'un petit jeu de données réelles.

Pour tester de notre méthodologie, nous procéderons d'abord à des essais sur des grands jeux d'images de la conduite automobile, ceux-ci étant facilement accessibles. Une fois la méthode validée, nous l'appliquerons aux données d'entraînement de notre cas spécifique. Finalement, nous ferons l'analyse de nos résultats afin de vérifier si notre approche a permis de segmenter des images avec une performance suffisante pour envisager une utilisation dans des cas réels.

CHAPITRE I

PROBLÉMATIQUE

La rédaction de ce mémoire s'effectue dans un contexte où la recherche en segmentation par RN est en pleine effervescence. Il est certain que les expérimentations et réseaux présentés dans ce document ne représentent déjà plus les meilleures pratiques.

Néanmoins, les principes de génération de données, d'entraînement de RN et d'application de ces technologies à des cas réels demeureront valides. L'objectif de ces travaux est davantage l'exploration de ces nouvelles capacités plutôt que la détermination quantitative de la méthode le plus efficace. Changer d'architecture de RN peut se faire avec peu d'impact sur le processus d'entraînement et de prédiction.

1.1 DÉFINITIONS ET OBJET DE RECHERCHE

Deux éléments sont à définir afin de cerner le sujet de ce mémoire. Les RN ne sont qu'une structure d'algorithme, le processus d'entraînement et ses données sont tout aussi importants.

Premièrement, les RN et l'intelligence artificielle sont à la mode et souvent considérés comme révolutionnaires. On oublie que ces idées sont vieilles de plusieurs

décennies déjà et que d'autres formes d'apprentissage machine existent et performent déjà à exécuter des tâches similaires.

Malgré le battage médiatique autour de ces termes et la force qu'on leur associe parfois à tort, ils représentent une technologie très utile qui changera bien des domaines. La quantité de données et de matériel informatique nécessaire pour entraîner des RN de grande envergure les rend encore difficiles d'utilisation. Cependant, le traitement d'images est un cas d'information non-structurée dans lequel les RN dépassent les méthodes d'apprentissage machine plus classiques.

La segmentation est une sous-catégorie des RN de traitement d'image. Dans ces RN, on attend comme résultat une carte des classes de même dimension que l'image entrée dans le RN. Cette carte des classes présente les probabilités calculées pour chaque pixel. Par exemple, sur une image de 512x512 pixels et un RN identifiant 8 classes, le résultat aura la taille de 512x512x8 (plus de 2 millions de valeurs). En obtenant N valeurs pour chaque pixel de l'image, on peut comprendre la complexité exponentielle que représente la segmentation en comparaison avec la classification.

Deuxièmement, peu importe l'architecture du RN, sa performance est dépendante de la qualité de son entraînement. La complexité de création de ces données est du même niveau que la complexité des résultats attendus du RN. Par exemple, si on considère un classificateur binaire, il suffit de pouvoir classer les données d'entraînement dans des groupes A/B. Ce type de situation se produit lorsqu'on tente de déterminer, par exemple, si un client va accepter ou non une promotion selon son historique et ceux de client jugés similaires. Dans un tel cas, il suffit de récupérer les données d'une promotion précédente et assumer que la nouvelle sera similaire.

Dans la segmentation d'image, le résultat attendu est la classification de chaque pixel d'une image. On peut donc considérer la complexité théorique maximale du résultat comme étant le nombre de pixel de l'image multiplié par le nombre de classes. Créer manuellement ce type de données est long et par conséquent, coûteux. Heureusement, il existe des outils permettant d'accélérer ce travail manuel en classant des zones d'images plutôt que chaque pixel. Un tel outil sera d'ailleurs utilisé pour nos travaux.

Un frein majeur à l'application de ces RN est donc l'effort que représente la création de ce JD. Certains domaines sont suffisamment populaires (la conduite automobile est le principal exemple) pour avoir des JD d'images segmentées disponibles publiquement. En dehors de ces rares cas, il est impossible d'appliquer la segmentation par RN sans avoir une stratégie de création de son propre JD.

L'APPLICATION PRATIQUE DES RÉSEAUX NEURONAUX DE SEGMENTATION D'IMAGE

L'agriculture représente un grand défi à l'application de ces technologies pour plusieurs raisons :

— L'impossibilité de contrôler complètement les paramètres lors de la prise de photo en extérieur, comme il serait possible de faire sur une ligne d'assemblage en usine, rend peu performantes les méthodes classiques qui performent mieux dans des environnements constants.

— L'énorme variabilité entre les exploitations, les équipements, les plantes, les saisons et les champs créent des images qui peuvent être très différentes bien que celles-ci représentent « la même chose » aux yeux d'un humain.

— Il existe peu de données existantes qui pourraient être utilisées pour l'entraînement. Celles que nous avons identifiés sont cataloguées à la fin du

chapitre II et ont été publiées majoritairement après le début de nos travaux en 2017 (elles auraient été utiles à nos travaux si elles avaient été disponibles avant). Ce point tend cependant à changer avec l'agriculture de précision et l'arrivée de nouvelles méthodes de culture basées sur les données, particulièrement avec l'usage de drones.

Ces environnements peu structurés et variables représentent des cas pour lesquels les RN sont supérieurs aux méthodes classiques. Cependant, la création de JD dans ce milieu s'avère difficile sans un investissement en temps et en ressources, rendant ces méthodes peu attirantes pour les petites organisations et ce, malgré leur potentiel.

1.2 ÉNONCÉ DE LA PROBLÉMATIQUE

La segmentation d'images par RN est récente et en mouvement constant. Il est difficile de la comparer objectivement aux méthodes traditionnelles d'apprentissage machine en dehors de quelques JD publics sur lesquels la majorité de la recherche s'effectue. Il est donc possible que cette performance ne puisse pas se transférer facilement à des cas différents qui n'ont pas ces JD massifs ou dont les caractéristiques sont trop différentes.

Cette difficulté d'entraînement des RN en dehors de ces JD se résout par la présence de données historiques ou de la classification manuelle, méthodes difficiles dans des scénarios de segmentation, et encore plus lorsqu'on parle d'industries n'ayant pas l'habitude de la gestion des données.

L'univers de l'intelligence artificielle évolue encore beaucoup par heuristique et avancés à tâtons. Les règles ne sont pas toujours claires et l'expérimentation demeure souvent la meilleure façon de valider une idée. Il n'existe aucune façon de

confirmer que la technologie peut s'appliquer à notre cas sans l'essayer, bien que cela semble plausible.

1.3 QUESTIONS ET HYPOTHÈSES DE RECHERCHE

Nous tenterons de répondre à la question : Comment entraîner des RN de segmentation d'images lorsqu'il est trop coûteux et complexe de segmenter manuellement un JD d'images d'une taille suffisante ?

Pour discuter de cette question, nous émettons l'hypothèse suivante : comme cela a déjà été réalisé avec succès pour des cas de conduite automobile (Ros *et al.*, 2016) (Richter *et al.*, 2016), des images majoritairement virtuelles permettront d'entraîner des RN de segmentation d'images à un niveau suffisant pour permettre leur utilisation dans des cas réels d'agriculture.

Pour ce faire, nous poursuivons les objectifs suivants :

- Analyser la documentation scientifique en lien avec notre objet de recherche
- Explorer les méthodes d'entraînement existantes
- Tenter une méthode d'entraînement permettant l'utilisation en agriculture
  - La méthode d'entraînement doit être suffisamment simple pour permettre l'adaptation à des cas d'usages multiples.
- Implémenter cette méthode et évaluer la performance des RN obtenus.

1.4 ESQUISSE MÉTHODOLOGIQUE

Notre méthodologie portera sur trois éléments principaux soit la génération d'images virtuelles, les RN de segmentation d'images et le transfert d'apprentissage.

- Tout d'abord, la création d'images de synthèse, nous permettant de créer des données d'entraînement facilement.

— Ensuite, construire un RN de segmentation d'image et l'entraîner sur ces données virtuelles.
— Finalement, le transfert du RN vers les images réelles.

GÉNÉRATION D'IMAGES DE SYNTHÈSE

Les premières expérimentations ont été effectuées en superposant ensemble des images de différentes espèces de plantes et en générant automatiquement le masque des classes. Le rapprochement des histogrammes a permis de réduire les écarts apparents entre les différentes composantes des images. Les résultats de cette méthode sont variables et ne ressemblent pas vraiment à des images réelles (Figure 1.1). Il serait possible d'améliorer la qualité, mais la décision fut prise d'aller vers une autre approche, soit l'utilisation des technologies issues de l'univers des jeux vidéos.

UNREAL ENGINE 4

Puisque plusieurs engins 3D existent et sont facilement accessibles, il n'aurait pas été pertinent de construire de zéro une telle technologie pour nos besoins. Divers engins furent étudiés afin d'en trouver un qui se prête à notre situation.

Unreal Engine 4 [1] fût choisi principalement pour l'énorme communauté l'entourant, permettant d'apprendre à l'utiliser rapidement et d'avoir un grand nombre d'exemples et de matériel pouvant être utilisés. D'autres outils comme Unity [2],

1. https ://www.unrealengine.com/en-US/industry/training-simulation

2. https ://unity.com/fr/solutions

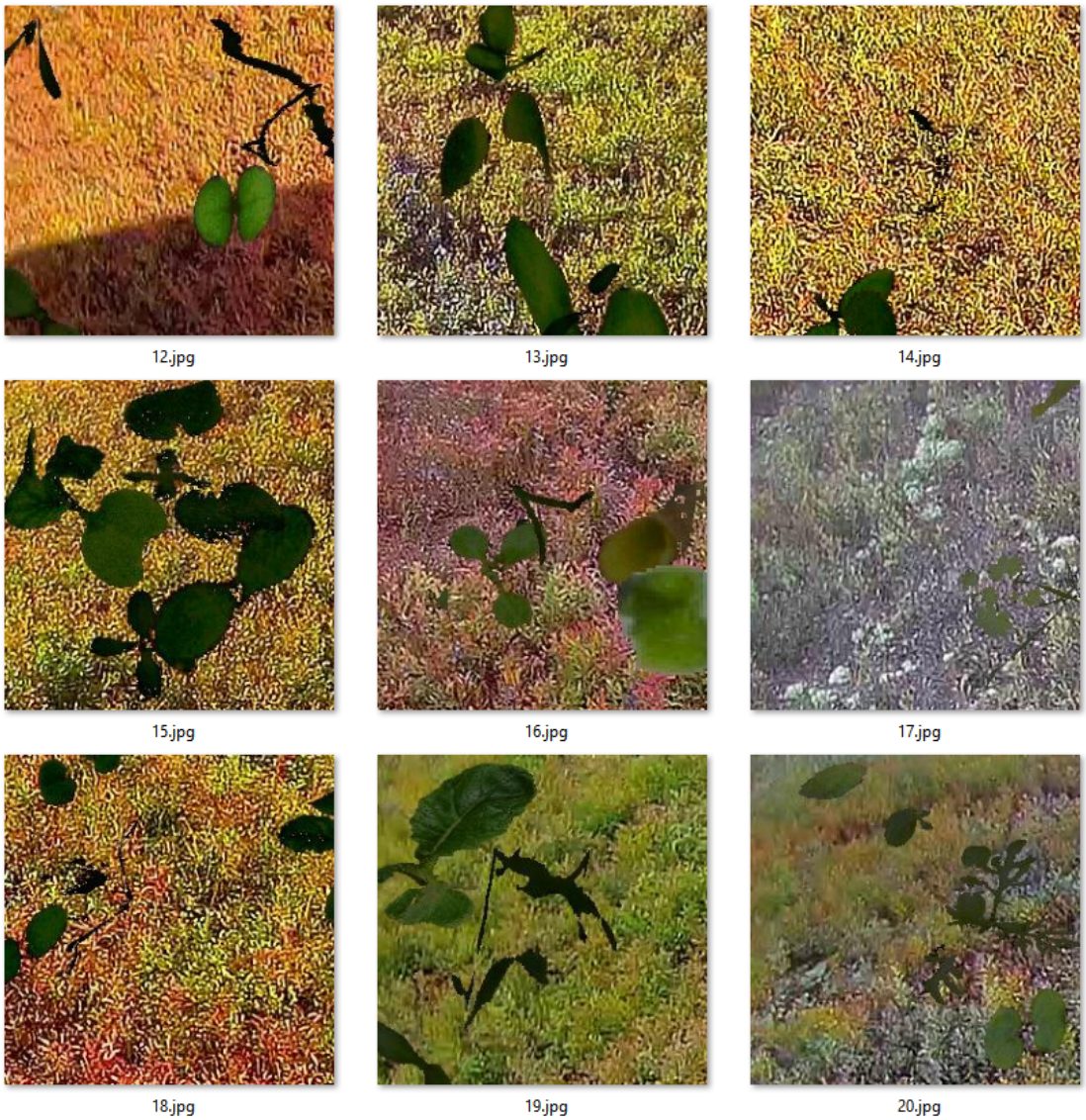

Figure 1.1 Exemples de superpositions d'images

Godot [3], Lumberyard [4] ou Blender [5] ont été comparés et auraient pu accomplir des résultats similaires.

RÉSEAUX NEURONAUX DE SEGMENTATION D'IMAGE

Pour ces travaux, nous cherchions une mise en oeuvre existante d'un RN qui serait simple d'utilisation afin de se concentrer sur l'entraînement et les données plutôt que sur la programmation. Nous avons favorisé Tensorflow et Keras, puisque ces derniers s'exécutent facilement sur divers environnements matériels/logiciels et que l'accessibilité de la documentation facilite la résolution de problème.

RESEAU SEGNET CONSTRUIT AVEC KERAS

Les premières expérimentations furent effectuées avec une architecture Segnet et la librairie Keras [6]. Il s'agit d'une librairie en code source ouvert écrit en Python [7] qui permet d'interagir avec des plateformes d'apprentissage profond comme Tensorflow [8], Theano [9], Microsoft Cognitive Toolkit [10] et MXNet [11]. Elle est conçue

---

3. https ://godotengine.org/
4. https ://aws.amazon.com/fr/lumberyard/
5. https ://www.blender.org/
6. https ://keras.io/
7. https ://www.python.org/
8. https ://www.tensorflow.org/
9. http ://deeplearning.net/software/theano/
10. https ://docs.microsoft.com/en-us/cognitive-toolkit/
11. https ://mxnet.apache.org/

comme une interface permettant d'expérimenter plus rapidement et intuitivement avec les RN en évitant une grande partie de la complexité des plateformes. La mise en œuvre de SegNet utilisée pour ces travaux provient de Github (Kamikawa, 2017) et peut être téléchargée à partir du site web.

Celle-ci met en oeuvre les couches convolution et déconvolution par indices, qui sont des composantes améliorant grandement la performance du RN. Les autres mises en œuvre sont souvent faites avec des couches de convolution/déconvolution classiques ne permettant pas les mêmes résultats. Il s'agit donc d'un RN conforme au RN Segnet original (Badrinarayanan *et al.*, 2017).

Deux autres options furent étudiées, soient Mask R-CNN en Keras par MatterPort [12] et Deeplab V3+ en Tensorflow [13]. Elles ont été rejetées en raison de leur niveau de complexité.

TRANSFERT D'APPRENTISSAGE

La gestion des différences entre JD dans le contexte de l'apprentissage machine se nomme l'adaptation de domaine. Il existe toujours un certain décalage entre les JD. Ainsi, si on accomplit l'entraînement sur l'un et qu'on utilise le RN sur un autre sans tenir compte de ces décalages on risque d'obtenir de mauvais résultats. Ces problèmes sont courants lors de l'utilisation de JD synthétiques ayant à tout coup des différences avec le monde réel. Par exemple, un JD d'entraînement contenant uniquement des fleurs bleues ne permettra pas de reconnaître les fleurs jaunes d'une autre série d'images, la propriété « couleur bleue » étant trop fortement lié à la probabilité d'être une fleur.

---

12. https ://github.com/matterport/Mask_RCNN

13. https ://github.com/leimao/DeepLab-V3

MÉLANGE DE JEUX DE DONNÉES

Il existe deux méthodes pour minimiser cette forme de surapprentissage. La méthode la plus simple est d'utiliser des données d'entraînement issues de plusieurs scénarios réels. Bien que notre cas ne permette pas d'utiliser uniquement ces données, il est possible de mélanger de diverses façons à l'entraînement une petite quantité d'images segmentées issues du monde réel et réduire l'écart d'apprentissage avec les données virtuelles. Ce mélange se fait soit en intégrant complètement toutes les données pour l'entraînement, soit en effectuant quelques époques d'entraînement sur les données réelles après avoir atteint le plateau de performance sur les données synthétiques. Cette méthode simple sera testée dans nos travaux.

ADAPTATION PAR ÉLIMINATION NON-SUPERVISÉE DES DIFFÉRENCES ENTRE DOMAINES

Les méthodes non-supervisées cherchent à réduire les différences entre domaines sans intervention humaine. La méthode retenue consiste à ajouter une branche au RN ayant pour fonction de discriminer de quel JD provient l'image d'entraînement (Ganin et Lempitsky, 2015). Comme on peut le voir à la Figure 1.2, la branche rose a été ajoutée pour accomplir cette fonction de discrimination. S'il est possible de déterminer le JD à partir des propriétés extraites par la première partie du RN, c'est que les propriétés ne sont pas uniformément distribuées entre les JD.

La rétropropagation du gradient est le processus d'apprentissage selon lequel on établit la marge d'erreur entre la prédiction du réseau et la bonne réponse puis que l'on ajuste les poids (valeurs) des neurones. Cette méthode inverse le gradient, ce qui permet d'utiliser la rétropropagation pour pénaliser l'extraction de propriétés distinguant les JD au lieu de l'encourager. Le RN est donc forcé d'utiliser des propriétés communes à tous JD.

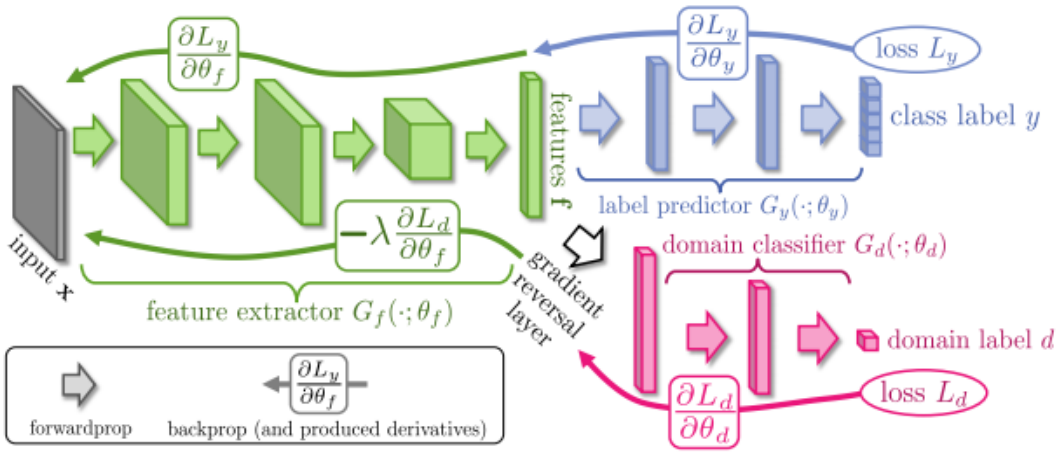

Figure 1.2 Représentation d'architecture d'un réseau incluant une branche d'adaptation des domaines (Ganin et Lempitsky, 2015)

Cette approche est simple tant qu'il nous est possible de développer la couche d'inversion du gradient. Il existe de multiples méthodes d'adaptation non-supervisée qui sont plus complexes à implémenter et qui n'ont pas été retenues pour cette raison. Une approche étudiée pour ces travaux utilisait la régulation des données pour atteindre l'adaptation entre les domaines (Zellinger *et al.*, 2019). Le travail requis pour son développement dépasse le cadre de ce mémoire.

ADAPTATION PAR MÉTHODES SUPERVISÉES OU SEMI-SUPERVISÉES DES DIFFÉRENCES ENTRE DOMAINES

Ces méthodes n'ont pas été intégrées dans ce travail par leur dépendance au travail humain pour faire l'adaptation entre les JD, nous éloignant de notre objectif de simplicité. Il a été décidé de limiter le travail d'adaptation dans le cadre de ce mémoire à une méthode qui a démontré de bons résultats et qui est facile à intégrer.

CHAPITRE II

CONSIDÉRATIONS THÉORIQUES

Cette revue sera divisée en deux sections. Premièrement, nous couvrirons les développements des dernières années en vision numérique basés sur les RN. Bien que l'utilisation des RN date de plusieurs décennies déjà, ce n'est que depuis très récemment que nous avons à la fois le matériel et les techniques d'apprentissage capables de les exploiter en dehors d'un cadre théorique.

Deuxièmement, nous aborderons l'utilisation de la vision numérique en agriculture. Bien que l'utilisation des RN en soit à ses débuts, l'idée d'utiliser de l'imagerie numérique n'est pas nouvelle pour autant dans le domaine. D'autres méthodes ont été exploitées et continuent de l'être.

Le perceptron, une architecture simple de neurones a vu le jour en 1957 par le travail de Frank Rosenblatt (White et Rosenblatt, 1963). La base de l'entraînement de RN, le gradient stochastique, date également de cette époque (Robbins et Monro, 1951). Cependant, la capacité de calcul requise pour entraîner de gros RN n'était pas vraiment disponible avant 2010. Cette incapacité à les exploiter a limité ces outils à la théorie mathématique pendant longtemps.

Les fonctions convolutionnelles furent le déclic qui a permis aux RN de développer considérablement leur efficacité en traitement de l'imagerie. De façon similaire au

cortex visuel, elles permettent de « dé-spatialiser » le contenu d'une image en étant beaucoup plus souples quant à l'emplacement de l'information (LeCun *et al.*, 1998). Cette façon de procéder s'est avérée très efficace sur la reconnaissance de caractères dès la fin des années 1990.

2.1 RÉSEAUX DE CLASSIFICATION

Ces RN qui déterminent la présence d'une ou plusieurs classe sur une image ont servi de précurseurs aux RN de segmentation. Ces architectures ne servent pas à déterminer la position ou le contour d'un objet mais plutôt à déterminer s'il apparaît dans l'image. Certains de ces RN peuvent d'ailleurs servir d'éléments structurels à un RN de segmentation car il est possible de récupérer cette capacité de détection. Ces deux approches demeurent intimement liées dans leur évolution, les avancées dans l'une pouvant souvent servir dans l'autre.

ALEXNET (2012)

Cette architecture (Krizhevsky *et al.*, 2012) est considérée comme la base des RNC modernes. Elle remporta le « 2012 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) » avec un taux d'erreur de 15.4%, son plus proche compétiteur étant à 26.2%. Il s'agit d'un design plutôt simple (en comparaison avec ce qui se fait aujourd'hui) débutant par quelques couches convolutionnelles suivies de couches denses conventionnelles, la dernière couche représentant chacune des classes que le RN est capable d'identifier.

Ce bond prodigieux de la capacité de classification d'un algorithme n'est pas passé inaperçu et cette technologie devint rapidement un des sujets les plus étudiés de la vision par ordinateur. Son architecture et sa méthode d'entraînement sont encore les bases sur lesquelles les algorithmes d'aujourd'hui sont construits.

ZF NET (2013)

ZF NET (Zeiler et Fergus, 2014) fut le gagnant de la compétition de 2013 avec un résultat de 11.6%. Il s'agissait d'une évolution d'AlexNet. Notre compréhension de l'aspect convolutionnel des RNC prend ses origines dans les études faites sur ce RN. Les auteurs ont conçu une méthode pour « inverser » les convolutions et qui permettait d'observer les propriétés des images que le RN avait détectées et utilisées pour générer sa prédiction. Ces travaux de compréhension sur la mécanique interne des RN ont servi de base aux évolutions suivantes.

VGG NET (2014)

Cette architecture (Simonyan et Zisserman, 2015) diffère de ses prédécesseurs dans sa structure. Ses composantes sont plus simples en utilisant des convolutions de 3x3 pixels au lieu de 11x11 (AlexNet) et 7x7 (ZF Net). Elle est cependant beaucoup plus profonde (elle contient plus de couches). Cette notion de filtres simples sur plusieurs niveaux représente hiérarchiquement les propriétés d'une image et constitua le début des architectures très profondes en comparaison avec les premiers RN.

GOOGLENET (2015)

GoogLeNet est une des premières architectures profondes comprenant plus de 100 couches (Szegedy *et al.*, 2016). Elle gagna la compétition de 2014 avec un taux d'erreur de 6.7% (ce qui représente presque trois fois moins d'erreurs que le gagnant de 2012). Ses auteurs se sont intéressés à son architecture au lieu de simplement « ajouter des couches » comme ce fut souvent le cas dans les années précédentes. L'augmentation du nombre de couche apporte comme problème d'aug-

menter considérablement le nombre de paramètres à entraîner, transformant le temps d'entraînement de quelques semaines à quelques années rapidement. L'utilisation de « modules » étant eux-mêmes de petits RN un à la suite de l'autre s'avéra très efficace. En effet, GoogLeNet atteint ses résultats en utilisant 12 fois moins de paramètres qu'AlexNet, pour une performance pourtant beaucoup plus efficace.

RESNET (2015)

Resnet (He *et al.*, 2016) gagne en 2015 avec un taux d'erreur de 3.6%, plaçant ainsi les RNC au-dessus des humains pour la première fois. Ce RN très profond (152 couches) utilise le concept de « bloc résiduel » pour améliorer le temps d'entraînement (d'où son nom). Cette technique permet de contenir les effets du nombre de couches. Il semble cependant que nous arrivons à une limite de rendement décroissant. Les auteurs ont tenté des RN jusqu'à 1202 couches, mais la performance allait en diminuant, probablement en raison du surapprentissage (phénomène par lequel un RN contient tellement de paramètres qu'il devient capable d'apprendre « par cœur » les données d'entraînement au lieu de développer des généralités qui permettent de classifier de nouvelles images).

DENSENET (2015)

DenseNet est une évolution de ResNet qui permet d'atteindre des résultats supérieurs en utilisant moins de paramètres (Huang *et al.*, 2017). Cette architecture utilise une approche similaire à ResNet mais propose de concaténer ensemble les résultats de plusieurs couches plutôt que de les additionner. Ces travaux confirment la tendance vers les assemblages différents de neurones plutôt que la simple augmentation de couches.

2.2 RÉSEAUX DE DÉTECTION

Les algorithmes couverts jusqu'ici détectent les objets sans les positionner et supportent parfois mal d'avoir plusieurs classes dans la même image. Ce type de classificateur est limité lors de l'utilisation en vision numérique qui présente rarement des cas bien contrôlés au niveau des données entrantes. Les réseaux de détection sont mieux conçus pour gérer la présence de plusieurs éléments dans une même image et indiquent en plus leur emplacement.

R-CNN (2013), FAST R-CNN & FASTER R-CNN (2015)

R-CNN représente le premier essai à combiner la classification et la division en région de l'image entrante, permettant ainsi de distinguer plusieurs éléments dans une même image et de donner leur position approximative (Girshick *et al.*, 2014).

Cette première version était cependant très lente (avec des délais pouvant approcher la minute pour analyser une seule image) et demeura donc un outil théorique, étant incapable d'atteindre les vitesses nécessaires à faire de l'analyse d'image en temps quasi-réel comme le pouvait Viola-Jones, le principal algorithme de détection et de localisation dans les images à l'époque. Ce dernier pouvait atteindre facilement plusieurs dizaines d'analyses par secondes sur du matériel courant (Viola et Jones, 2004) sans avoir à utiliser de RN.

YOLO (2015)

Le RN « You Only Look Once » (Redmon *et al.*, 2016) traite différemment le problème. Au lieu de prendre un classificateur et d'y ajouter la localisation, YOLO traite la détection comme un problème de régression et tente ensuite de classifier

les éléments repérés dans l'image. Cette approche est beaucoup plus rapide au coût d'une diminution de la précision sur la localisation.

SSD (2015)

« Single Shot Detector » est une autre approche qui combine ensemble la classification et la localisation, ce qui résulte en une approche simple, rapide et assez précise par rapport aux alternatives (Liu *et al.*, 2016). L'objectif de cette architecture était un peu différente de ce qui s'était fait à ce jour en priorisant la vitesse à la précision. C'est l'un des premiers réseaux capables d'être utilisés dans un contexte en temps réel sans requérir une énorme puissance de calcul. Au lieu de chercher à établir des propositions d'objet et de les soumettre à un classificateur qui doit classer ces propositions, le traitement est intégré en seul processus, et ce, d'une façon qui évite les pertes de précision.

NASNET (2017)

Ce RN à la fine pointe de la recherche est issu du concept « AutoML » (Zoph *et al.*, 2018), soit de faire de la structure même du RN un paramètre pouvant être modifié et optimisé durant l'entraînement. Les résultats sont prometteurs et NASNet est déjà un RN offrant des performances comparables ou supérieures aux RN conçus par les experts tout en accaparant moins de ressources de calcul. Cette approche fonctionne en classification et en localisation.

2.3 RÉSEAUX DE SEGMENTATION

Les réseaux de cette section vont une étape plus loin que la détection. Ils associent une probabilité de classe à chaque pixel d'une image. Cela permet d'y voir

les éléments détectés par l'algorithme, mais également de voir avec une grande précision la position de ces derniers, ce que les RN de détection ne font pas. Ces derniers se limitent habituellement à fournir une boite rectangulaire autour de la zone la plus probable de contenir l'objet détecté. Il s'agit donc d'excellents outils pour faire de la cartographie de haute précision.

Ces RN sont similaires à ceux de classification et de détection, mais ils ajoutent la capacité de « générer » une image segmentée à partir de l'image d'origine au lieu de donner simplement une classe ou des coordonnées rectangulaires de détection. Ce domaine est en pleine évolution et nous nous limiterons à 4 exemples pour cette analyse, bien qu'il en existe déjà plusieurs dizaines.

U-NET (2015)

Ce RN fut conçu pour cartographier des images médicales (Ronneberger *et al.*, 2015). Les auteurs se sont penchés sur la création des images annotées, travail très long lorsqu'il s'agit d'identifier chaque pixel. Les résultats obtenus furent supérieurs à ses compétiteurs à la compétition « IEEE International Symposium Biomedical Imaging » de 2015 et la vitesse considérée acceptable pour un usage réel.

SEGNET (2015)

Ces travaux cherchent à segmenter des images d'environnement, particulièrement des images capturées à partir d'un véhicule et des photos d'espaces intérieurs (Badrinarayanan *et al.*, 2017). Cette capacité à reconnaître l'environnement est un sujet d'actualité en navigation robotique. Les auteurs ont cherché à réduire la capacité de calcul et de mémoire nécessaires à l'algorithme dans l'optique de

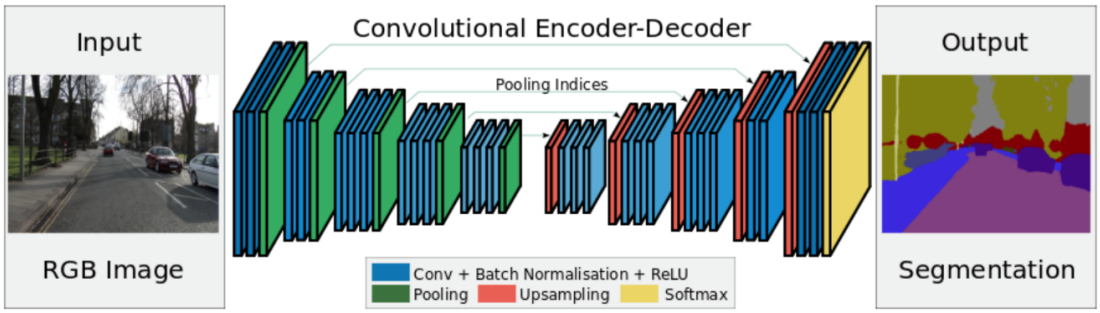

Figure 2.1 Architecture haut-niveau d'un réseau Segnet (Badrinarayanan *et al.*, 2017)

pouvoir utiliser ce réseau dans des systèmes embarqués soumis à des contraintes de puissance de calcul et de consommation d'énergie.

DEEPLAB (2016)

DeepLab (Chen *et al.*, 2018) serait présentement le RN obtenant les meilleures performances en segmentation. Son architecture diffère des autres implémentations en utilisant le concept de convolutions à trous inspiré du traitement de signal qui permet de restaurer avec une plus grande fidélité le masque de segmentation relié à une image. Ce type de convolution espace les pixels observés afin de couvrir une zone plus grande sans plus d'effort de calcul. Par exemple, une zone d'observation de 5x5 pixels dans laquelle on laisse tomber les colonnes et rangées paires résultera en seulement 3x3 pixels à traiter.

MASK R-CNN (2017)

Avant de parler de ce réseau, il est pertinent de décrire la différence entre la segmentation sémantique et la segmentation d'instance. La segmentation sémantique

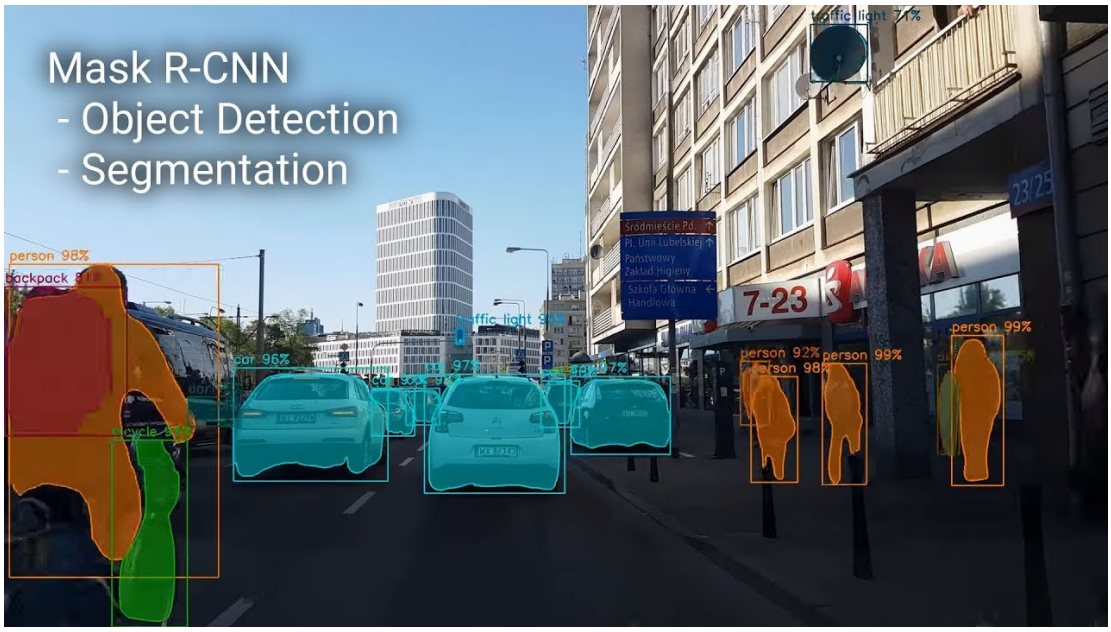

Figure 2.2 Exemple de prédiction d'un réseau MASK R-CNN (He *et al.*, 2017)

chercher à décrire de quelle classe sont les pixels d'une image, sans être capable toutefois de faire la différence entre plusieurs éléments d'une même classe dans une image. La segmentation par instance ajoute cette couche d'information supplémentaire aux prédictions, c'est-à-dire pouvoir déterminer les frontières entre plusieurs éléments de la même classe.

Ce RN, basé sur R-CNN, permet de faire ces deux types de segmentation (He *et al.*, 2017). Par exemple, il est capable de distinguer qu'il y a plusieurs voitures en file sur une image et les considère comme des instances différentes au lieu de traiter toute la zone comme un segment de classe voiture. On peut voir un exemple d'une telle prédiction à la figure 2.2. Bien qu'il y ait une zone au centre de l'image ayant la notion sémantique de voiture, le classificateur a également identifié des instances distinctes de voitures. Son architecture utilise une base de classifica-

teur/détecteur et y ajoute la capacité de générer un masque d'emplacement à chaque instance repérée.

2.4 VISION NUMÉRIQUE ET IDENTIFICATION DES PLANTES

Des méthodes de traitement de l'image autres que les RN existent en agriculture et sont déjà en utilisation par certains équipements novateurs. Ces méthodes ont de bons résultats quand il est possible d'établir une structure dans l'information à traiter, ce qui s'avère possible dans certaines situations de culture.

SPECTRE INFRAROUGE

Les plantes, par la chlorophylle, ont la propriété de réfléchir le spectre infrarouge proche (NIR) en grande partie, facilitant leur distinction contre un fond inorganique (le sol). Il existe sur le marché des équipements basés sur ce principe qui réduisent les quantités de pesticides appliqués en ne couvrant que les plantes. Par exemple, le Weed-IT Quadro [1] utilise cette propriété pour identifier les plantes et s'opère la nuit pour contrôler complètement l'éclairage et simplifier le traitement des images.

Cette méthode a cependant comme défaut de ne pas pouvoir distinguer les plantes entre elles. Bien que des travaux sur le sujet se font depuis plusieurs décennies (Shropshire, 1989), les résultats n'ont pas mené à des applications commerciales. L'infrarouge est une excellente façon de déterminer la présence d'une plante, mais qui ne permet pas d'efficacement différencier les plantes entre elles dans des environnements non-structurés.

---

1. https ://www.weed-it.com/weedit-quadro

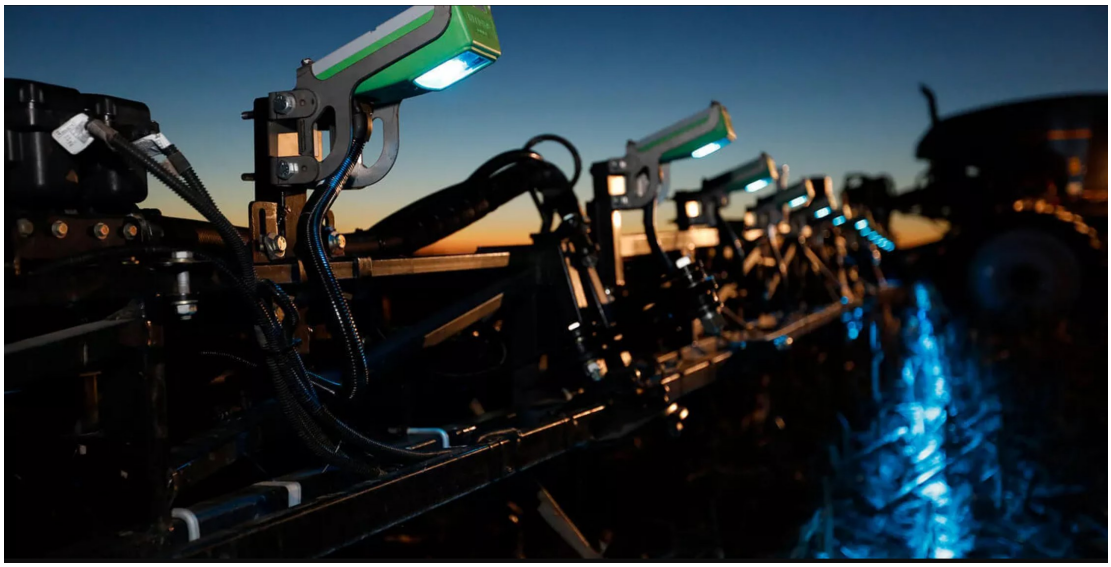

Figure 2.3 Appareil Weed-It Quadro photographié de nuit permettant de voir le contrôle de l'éclairage pour les capteurs. (WEED-IT Precision Spraying, 2021)

ANALYSE DES FORMES ET DES COULEURS

La détection des mauvaises herbes a été tentée en utilisant les méthodes plus « classiques » de vision numérique. Ces algorithmes cherchent à identifier les plantes par des propriétés extraites sur plusieurs facteurs qui sont ensuite soumises à des classificateurs statistiques pour séparer les espèces botaniques (Pérez *et al.*, 2000).

Comme bien d'autres cas de vision numérique, les méthodes statistiques et mathématiques précédant les RNC ont beaucoup de difficulté à s'adapter à la grande variabilité des images issues du monde réel, rendant leur utilisation peu performante en dehors de tests contrôlés.

Cela ne signifie pas que ces techniques soient pour autant inutilisables. Des entreprises telles que Garford utilisent ces méthodes dans des produits commerciaux. Comme cela peut être constaté dans la Figure 2.4, l'environnement de culture dans

Figure 2.4 Robocrop Inrow de Garford en opération (Garford Farm Machinery Ltd, 2021)

lequel ces machines sont utilisées possède l'avantage de la « structure » physique. Les plants sont en rang, ont une distance similaire entre eux et les différences de couleurs importantes entre les plantes et le sol facilite le travail de détection.

Cet environnement invariant permet d'utiliser des techniques plus simples que les RN et d'obtenir des résultats convenables. Le Robocrop Inrow [2] utilise ses caméras à l'avant pour détecter la position des plants de cultures et voir s'il y a des mauvaises herbes. Si ces mauvaises herbes sont détectées, un bras mécanique pivote autour de la plante à conserver pour éliminer les plantes indésirables, le tout sans usage d'herbicide.

GÉNÉRATION AUTOMATIQUE DE DONNÉES D'ENTRAINEMENT

Dans ces travaux, les auteurs ont vérifié l'hypothèse qu'il est possible d'utiliser des données virtuelles pour entraîner un RN segNet (Di Cicco *et al.*, 2017). En s'inspirant d'images réelles, ils ont généré des scènes par ordinateur pour simuler des champs sans devoir faire d'annotation manuelle.

Ils ont obtenu des performances très proches de l'utilisation d'images réelles en entraînant les RN sur ces données virtuelles. Dans tous les cas, il demeure possible de mixer les données virtuelles et réelles, ce qui améliore les résultats. Il s'agit d'une piste prometteuse pour entraîner des RN en minimisant l'effort d'annotation.

POSITIONNEMENT INVARIABLE DES PLANTES

Dans ces travaux, les chercheurs tentent de remplacer le recours à des outils de positionnement GPS de haute précision par la reconnaissance des plantes comme « marqueurs géographiques » permettant de se repérer d'un passage à l'autre dans

2. https ://garford.com/fr/bineuse-robocrop-inrow/

un champ (Kraemer *et al.*, 2017). Le taux d'erreur obtenu sur le positionnement est inférieur à 20mm et leur système s'est avéré capable de se positionner malgré les changements au sol et la croissance des plantes. Ces résultats s'avèrent prometteurs quant à la capacité de généralisation et d'adaptation des RN neuronaux.

2.5 UTILISATION DES RÉSEAUX RNC DE SEGMENTATION EN AGRICULTURE

Les succès des RN de ces dernières années a relancé les travaux de recherche et d'application de la vision numérique dans plusieurs domaines dont l'agriculture. Nous avons recensé 4 documents particulièrement intéressants pour nos recherches.

WEEDNET, IDENTIFICATION AÉRIENNE

Cette recherche (Sa *et al.*, 2018) utilise dans un cas pratique le RN SegNet. Les données proviennent d'imagerie aérienne acquise via l'utilisation de drones et de caméras. L'utilisation du spectre infrarouge a permis de segmenter les plantes du sol via l'indice NDVI [3]. Un champ de test a été séparé en trois sections (plantes comestibles, mauvaises herbes et mixte). Ce contrôle sur les données entrantes a permis de limiter le travail d'annotation manuelle.

L'expérimentation a permis d'obtenir un score F1 (mesure combinant la précision et le rappel variant de 0 à 1) prometteur de 0.81. Le RN pouvait être utilisé sur un Jetson TX2 (ordinateur miniature) à 1.8 images par seconde, permettant ainsi l'utilisation dans des équipements embarqués.

---

3. https ://en.wikipedia.org/wiki/Normalized_difference_vegetation_index

IDENTIFICATION AU NIVEAU DU SOL

Dans cette recherche, l'acquisition d'images a été effectuée directement au sol par l'utilisation d'un robot se déplaçant dans le champ agricole. Les images obtenues sont d'une meilleure qualité que ce qui peut être obtenu via un drone, étant capturées de beaucoup plus près. Les auteurs ont créé leur propre RN fortement inspiré d'autres modèles (comme segNet) (Milioto *et al.*, 2018). Cela a permis l'obtention d'un RN simple mais aussi moins propice au surapprentissage, ce qui est un risque lorsqu'on utilise un RN massif pour séparer un petit nombre de classes.

Ils ont cependant fourni à leur RN des images à 14 canaux. Les 3 couleurs (R, G, B) ainsi que 11 transformations (Sobel, Laplace, etc. . .) utilisées dans la segmentation des plantes lors de l'utilisation d'algorithmes statistiques basés sur l'extraction de propriétés. Cela a permis d'améliorer les performances du petit RN en lui évitant de devoir apprendre et appliquer des transformations équivalentes.

Leur système fonctionnait à 5 images par secondes sur un TX2. Les différents tests ont obtenu une précision entre 80% et 95%. Malgré les différentes méthodologies, ce résultat démontre ce qui peut être attendu en termes de perte de précision entre des données aériennes et des données prises directement au sol.

DÉTECTION ET SUIVI DE MALADIE PAR IMAGERIE AÉRIENNE

La bactérie *Xylella fastidiosa* est une bactérie dangereuse qui cause de sérieux dommages en agriculture. Elle est présente principalement dans les Amériques mais s'étend en Europe et en Asie et peut s'attaquer à plus de 350 espèces végétales. Cela en fait une menace mondiale dont la détection est essentielle pour être en mesure de la contenir et l'éradiquer.

Dans cet article (Zarco-Tejada *et al.*, 2018), des lectures d'imagerie, de spectrométrie et d'imagerie thermique ont été mises en commun afin d'identifier les infections avant que des symptômes ne soient visibles sur les oliviers. Les prédictions ont été vérifiées sur le terrain et une précision supérieure à 80% fut confirmée. Les altérations que la bactérie cause aux arbres causent des symptômes invisibles à l'oeil mais qui s'avèrent détectables si on traite rapidement l'énorme de volume de données que génèrent les senseurs.

Ces travaux utilisent des méthodes plus classiques de segmentation basées sur les seuils, suffisantes lorsqu'il s'agit seulement de distinguer les arbres du sol. Par la suite, les mesures obtenues pour chaque arbre étaient mise en commun pour établir le risque et/ou la présence d'infection. Des situations moins structurées, par exemple tenter le même genre de segmentation dans des scènes naturelles complexes, aurait été plus difficiles à accomplir avec ces méthodes et auraient bénéficié des capacités des RN.

Les RN ne sont pas forcément meilleurs par rapport aux méthodes classiques, surtout dans des situations bien structurées dans lesquelles il est possible de segmenter notre objectif par rapport à une scène globale. L'utilisation de capteurs spécialisés mettant en évidence de l'information invisible sur le spectre visible demeurent une excellente alternative à entraîner des RN capables d'extraire les mêmes informations à partir d'images RGB provenant de caméras.

AGRICULTURE-VISION : A LARGE AERIAL IMAGE DATABASE FOR AGRICULTURAL PATTERN ANALYSIS

L'équipe de ce projet s'est attaqué au problème de la disponibilité de JD d'imagerie agricole, une situation qui fut évidente lors des recherches de ce mémoire. Ils ont assemblé plus de 94 000 images et annotations reliées. Ils ont également utilisé

leur propres données pour entraîner des RN de segmentation et ont obtenu des résultats dans un intervalle de 40-50% (Tik Chiu, Mang ; Xu, Xingqian ; Wei, Yunchao ; Huang, Zilong ; Schwing, Alexander G. ; Brunner, Robert ; Khachatrian, Hrant ; Karapetyan, Hovnatan ; Dozier, Ivan ; Rose, Greg ; Wilson, David ; Tudor, Adrian P. ; Hovakimyan et S., 2020). Cela démontre qu'il y a encore beaucoup de travail à faire dans le domaine agricole.

Les auteurs comptent continuer à ajouter de l'information à ce JD et souhaitent en faire une référence dans le domaine de la recherche, particulièrement en segmentation, qui est souvent beaucoup plus utile en cartographie que les autres techniques visuelles comme la détection d'objets ou la classification de sujets uniques.

DEEPWEEDS - A MULTICLASS WEED SPECIES IMAGE DATASET FOR DEEP LEARNING

Des chercheurs se sont attaqués à un problème similaire à celui dont nous traitons dans ce mémoire, soit d'identifier des espèces végétales jugées nuisibles dans un milieu semi-naturel qui n'est pas aussi structuré que des espèces plantés par l'homme comme dans l'exemple précédent. Leur JD d'images de plus de 17 000 exemples couvre 8 espèces communes de l'Australie (Olsen *et al.*, 2019).

Leurs résultats de classification en temps réel avec des résultats autour de 95% confirme la possibilité d'identifier des mauvaises herbes directement lors du désherbage. Le type de caméra et les conditions d'utilisation ont été choisies dans cette optique et constitueraient une bonne base à l'élaboration d'un prototype commercial.

Note : Agriculture-Vision et DeepWeeds ont été publiés en cours de travaux de maîtrise. Si ceux-ci avaient été disponibles avant, ils auraient été utilisés plutôt que

les JD d'images basées sur la conduite automobile et les environnements urbains dans nos premières expérimentations.

QUANTIFICATION DES BLEUETS PAR IMAGERIE

L'auteur cherchait à repérer les bleuets dans un jeu d'images, mais surtout à les identifier individuellement. Identifier les fruits un par un permettrait d'en évaluer le mûrissement et déterminer s'il est prêt à être récolté (Arellano et Tapia Farias, 2019). Le réseau Mask R-CNN a été utilisé, son avantage étant de permettre la segmentation des classes mais aussi des instances de cette classe dans l'image (et donc d'en connaître la quantité). Une fois entraîné sur son JD créé manuellement (lui non plus n'ayant pas accès à des JD existants), il arrive à une performance suffisante (autour de 75%) pour compter le nombre de baies présentes dans une image.

D'autres chercheurs ont réalisé des travaux similaires avec d'autres fruits en arbustes et arrivent à des résultats qui indiquent qu'on pourrait éventuellement effectuer une cueillette robotisée basée sur ce principe. Il est cependant incertain qu'une telle technologie pourrait être utilisée avec le bleuet sauvage, celui-ci étant au ras du sol et non dans dans des arbustes comme les bleuets cultivés.

DÉTECTION D'ESPÈCES INDÉSIRABLES DANS LA CULTURE DU BLEUET SAUVAGE

L'auteur s'est attaqué au même problème que nous avec une méthode similaire. Il cherchait à identifier la fétuque et la petite oseille, deux espèces indésirables présentes dans les cultures de bleuets sauvage (Hennessy *et al.*, 2020). Ces travaux ont été effectués dans les provinces atlantiques, qui ont des cultures de bleuets sauvages similaires à ce qui se fait au Québec. Un RNC de détection de type

YOLO a été entraîné à reconnaître ces deux espèces du point de vue d'un épandeur d'herbicide. Une précision autour de 96% pour les deux espèces permettrait son utilisation en cas réel.

Du matériel performant a permis l'entraînement et l'inférence était possible à une vitesse permettant le temps réel. L'objectif était d'identifier si dans une image il se trouvait une ou plus instances de la plante. Cette classification est plus élémentaire que nos travaux, mais serait suffisante pour l'application de certains herbicides, confirmant la validité de la démarche.

RECUEIL DE RÉSULTATS DES RÉSEAUX NEURONAUX EN AGRICULTURE

Cette compilation de plusieurs recherches d'application des RN en agriculture (Yang et Sun, 2019) relève quelques faits intéressants. Tout d'abord, on constate une explosion dans la quantité de travaux sur le sujet dans les dernières années, point que nous avons nous-mêmes constaté durant la rédaction de ce mémoire. Le traitement d'images utilisant des RNC de diverses architectures semble être le nouveau couteau suisse de la recherche, cherchant à l'appliquer à tous les défis de données non-structurées qui avaient échappé à l'automatisation jusqu'à maintenant. On constate à la lecture de leur synthèse que des précisions de l'ordre du 85%+ sont généralisées, et que les cas de 95%+ sont courant. Il nous semble donc logique de supposer que notre technologie actuelle permet de viser 95%+ comme précision, tant qu'une capacité d'entraînement existe pour le cas d'usage que l'on cherche à résoudre.

2.6 DONNÉES D'ENTRAÎNEMENT

Dans le cas d'un accès limité à des données d'entraînement, il faut des stratégies de génération d'images de remplacement. Deux grandes approches existent, soit d'utiliser des éléments d'images existantes ou alors les simuler complètement.

RÉSEAUX GAN ET DÉTECTION DE POLYPE EN IMAGERIE MÉDICALE

Les auteurs de ces travaux (Thambawita *et al.*, 2021) ont utilisé des réseaux antagonistes génératifs afin de créer des données d'entraînement pour leur réseau de segmentation. Ce type de réseau génératif permet de générer des données en utilisant deux éléments, soit un générateur et un comparateur. Le générateur simule des images tandis que le comparateur cherche à faire la différence entre les images simulées et les images réelles. L'approche permet d'obtenir un générateur d'image capable de remplacer la disponibilité d'un grand jeu de données réelles. Il ont utilisé cette capacité pour insérer des polypes dans des images saines de façon convaincante, permettant ainsi d'entraîner un réseau a identifier ces excroissances anormales des tissus dans d'autres images médicales. Le domaine médical a lui aussi le problème des jeux de données limités, et la protection de la vie privée rend encore plus complexe l'acquisition et l'utilisation de telles images. L'utilisation de ces images simulées a donné des résultats d'entraînement comparables à l'utilisation de grands jeux d'images réelles, et a donné un résultat supérieur à l'utilisation d'un petit jeu d'images réelles. La méthode semble donc tout à fait valable pour produire des données d'entraînement.

Il s'agit d'une méthode beaucoup plus élaborée que ce qui a été tenté dans ces travaux. Nous n'avons pas considéré cette option car ce type de réseaux n'était pas encore très répandu et accessible aux début de nos travaux, la génération via des environnements simulés étant plus simple à mettre dans place. Il serait inté-

ressant d'appliquer ce type d'approche en agriculture afin de voir si des résultats comparables sont possibles.

UTILISATION EXCLUSIVE D'IMAGES DE SYNTHÈSE EN SEGMENTATION AUTOMOBILE

Ces travaux (Johnson-Roberson *et al.*, 2017) avaient pour but de comparer l'utilisation exclusive d'images simulées aux images réelles dans un cas de conduite automobile, soit d'identifier les autres voitures. L'engin 3D du jeu Grand Theft Auto V a été modifié pour manipuler et extraire directement l'information des scènes visuelles au lieu d'avoir uniquement une capture d'écran. Cela a permis d'avoir plusieurs conditions météorologiques et d'obtenir directement la segmentation des images à partir du jeu, évitant le besoin de les identifier manuellement. Il ont ainsi généré 200 000 images et ont pu les utiliser en comparaison à des images réelles.

Les auteurs ont obtenu une performance supérieure en utilisant les images virtuelles plutôt que les images réelles. Ils notent cependant que leurs tests se passent sur un échantillon réduit de possibilités et n'est peut-être pas représentatif de tous les cas. En effet, les deux jeux d'images réelles, Cityscapes (Cordts *et al.*, 2016) et KITTI (Geiger *et al.*, 2013), contiennent tous deux des images provenant d'Allemagne dans des conditions météorologiques et d'éclairage similaires. Malgré tout, l'utilisation d'images simulées semble être une approche d'entraînement valide et ce, même pour un réseau devant opérer sur des images réelles.

2.7 ANALYSE CRITIQUE

Au début de nos travaux (2017), les jeux de données d'images réelles agricoles étaient petits ou inexistants. Cette réalité justifiait une approche de synthèse cher-

chant à pallier à ce manque de données du domaine sur lequel nous travaillons. Les approches de synthèse, bien que moins performantes que les images réelles, ont l'énorme avantage de pouvoir être générées à faible coût en réutilisant des technologies existantes. Ces approches se sont avérées efficace dans d'autres domaines et rien ne laisse supposer qu'elles ne fonctionneraient pas en agriculture.

Plusieurs études sur l'analyse d'image par apprentissage profond en agriculture (et dans une multitudes d'autres domaines) ont été publiées dans les dernières années, soit après le début de nos travaux. Si le projet était débuté aujourd'hui, l'approche aurait plutôt été de tester ces nouvelles solutions au lieu de tenter d'utiliser des approches de transfert ou de synthèse. Cette absence de données d'entraînement pour un domaine aussi important que l'agriculture a été remarquée (et corrigée en partie) par plusieurs chercheurs.

CHAPITRE III

MÉTHODOLOGIE

Les premiers essais en génération d'images synthétiques furent réalisés en superposant des images de plantes sur des images réelles de champs de bleuets. Cette méthode simple avait comme objectif de tester la faisabilité des images synthétiques pour entraîner un RN en segmentation. Les résultats obtenus sont peu réalistes et ne pourraient pas servir à faire de la prédiction en terrain réel. La Figure 3.1 présente quelques exemples de ces images.

L'arrière-plan des images virtuelles provient d'images réelles de champs de bleuets sauvages capturées à la récolte 2017 grâce à des caméras installées sur des tracteurs de récolte. Ces images ont été sélectionnées manuellement pour s'assurer qu'elles contiennent uniquement des plants de bleuets sauvages. Environ 450 images ont été produites, incluant une grande variété de couleurs, d'éclairage, de types de sol et d'allure des plants.

L'avant-plan provient quant à lui d'une collection d'images déjà segmentées de travaux portant sur l'identification des pousses de plantes de 12 espèces (Giselsson *et al.*, 2017). Ce JD contient plus de 5500 images. Ces images ont l'avantage d'être déjà segmentées.

Des problèmes furent rencontrés pendant ces essais. La segmentation des images était basée sur l'utilisation de la couleur noire et non d'un masque, laissant des trous dans les feuilles lors du retrait de la couleur. La similitude des couleurs entre les arrière et avant plans était très variable et les tentatives de rapprochement ont engendré des résultats médiocres. Il fut décidé d'abandonner cette méthode et d'explorer plutôt la génération d'images à partir d'un environnement simulé.

La méthode n'est cependant pas à éliminer complètement. Comme on l'a constaté dans la revue de la littérature, il existe maintenant des RN capables d'apprendre la génération d'images synthétiques. D'autres (Chen et Kae, 2019) se spécialisent même à exécuter la composition d'image afin de la rendre la plus réaliste possible. Ces outils donneraient certainement de meilleurs résultats que ce que nous avons observé avec notre méthode.

3.1 GÉNÉRATION PAR L'UTILISATION D'UN MOTEUR 3D

La génération d'images synthétiques pour la vision numérique semble regagner en popularité dans l'entraînement des RN. Ceux-ci requièrent d'énormes quantités de données pour être performants et il est rarement possible d'avoir un accès facile à autant d'images. La segmentation apporte un défi supplémentaire à l'annotation puisse qu'il faut générer les cartes de classes pour chaque pixel. Il s'agit d'un défi parfait pour un environnement virtuel à partir duquel peuvent être générées des quantités quasi infinies d'images et les cartes de classes de chacune d'elles.

L'utilisation des outils du jeu vidéo évite un énorme travail de développement. Trois engins ont été testés pour cette tâche, soit : Unreal, Unity et Lumberyard. Le choix d'utiliser Unreal est principalement dû à la taille de sa communauté de développeurs, permettant de trouver facilement réponses aux questions d'un débu-

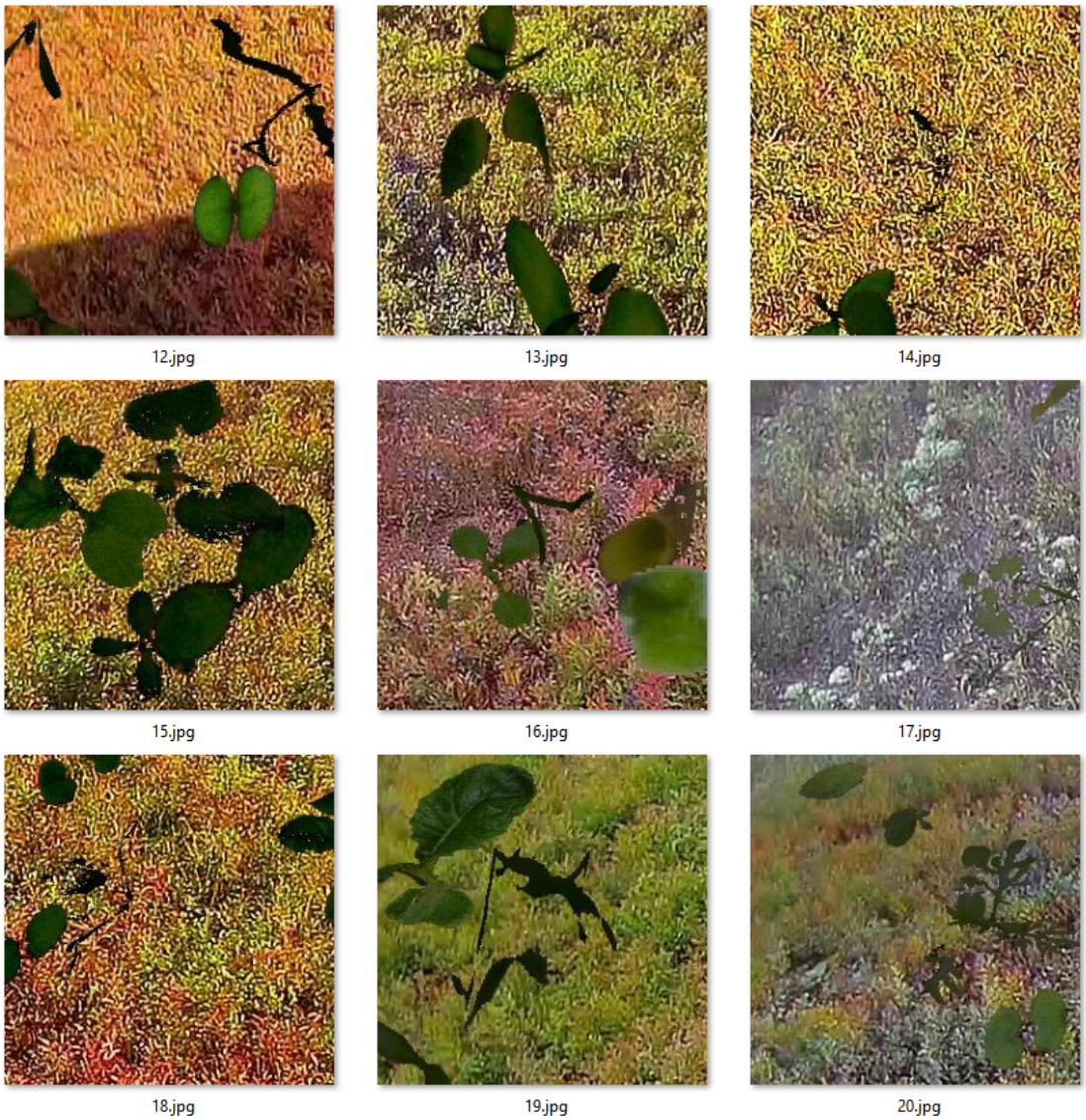

Figure 3.1 Exemple d'images synthétiques générées par superposition

tant. Ces moteurs 3D auraient tous été capables d'effectuer les tâches nécessaires mais avec une courbe d'apprentissage plus difficile qu'Unreal.

Le critère ayant influencé le choix d'utiliser un engin issu des jeux vidéo plutôt qu'un outil cinématographique est la rapidité. Les jeux vidéo sont conçus pour être utilisé en temps réel et le temps de rendu d'une image est minime (plusieurs dizaines d'images par secondes). Les outils cinématographiques ont des capacités de rendu supérieures mais un temps de génération d'image considérablement plus long. Il est courant de rencontrer des temps de génération de plusieurs minutes par image avec ces outils. La puissance de calcul disponible pour ce mémoire étant limitée, la capacité de faire des itérations rapides s'avère plus importante que la hausse de qualité qui aurait pu être obtenue avec un processus différent.

La principale différence entre les deux systèmes réside dans l'utilisation du « ray tracing » ou de « shaders ». Le ray tracing consiste à suivre le parcours de chaque « rayon » de lumière des sources lumineuses jusqu'à la caméra, permettant ainsi une reproduction très réaliste d'une scène au coût d'une énorme quantité de calculs. Les shaders sont quant à eux des descripteurs de surfaces, utilisant différents concepts pour décrire l'apparence attendue d'un certain matériel sous un certain éclairage. Les approximations issues de shaders sont moins réalistes mais beaucoup plus rapides. Bien que des travaux soit en cours actuellement pour permettre l'utilisation du ray tracing en temps réel, il s'agit de prototypes de l'industrie [1] et non de méthodes facilement disponibles. Nous ne croyons pas que ces différences de rendu peuvent affecter significativement notre entraînement, mais il serait pertinent de valider cette hypothèse dans des travaux subséquents.

---

1. Epic Games Demonstrates Real-Time Ray Tracing in Unreal Engine 4 with ILMxLAB and NVIDIA, 21 Mars 2018, https ://www.unrealengine.com/en-US/blog/epic-games-demonstrates-real-time-ray-tracing-in-unreal-engine-4-with-ilmxlab-and-nvidia

Une amélioration de la capacité prédictive pourrait justifier le coût des ressources de rendus en ray tracing.

3.2 CRÉATION D'UN ENVIRONNEMENT VIRTUEL

Pour notre cas d'usage nous n'avons pas besoin de l'entièreté des capacités d'un moteur 3D destiné au jeu vidéo, seulement la création d'un environnement dans lequel il nous est possible de déplacer une caméra. Les modèles et textures constituent l'aspect demandant le plus de travail. Heureusement, l'utilisation d'engins communs donne accès à un grand nombre d'éléments disponibles de manière gratuite ou commerciale. Des éléments communs, comme par exemple les arbres, sont faciles à trouver et à intégrer. Les éléments plus particuliers (comme le bleuet sauvage) ne sont pas vraiment disponibles et doivent être créés ou adaptés à partir d'éléments déjà existants. Comme on peut le constater à la Figure 3.2, la qualité de ce qui est disponible s'approche du photoréalisme [2] et permet d'éviter un travail de design considérable.

Pour créer un maximum de variations tout en évitant les difficultés de la génération complètement automatisée, nous procédons par tuiles. Cette méthode permet de créer des scènes contrôlées afin d'éviter les scénarios impossibles dans des cas réels. Il s'agit d'élaborer une série de zones préassemblées qui sont ensuite placées aléatoirement à la manière d'un échiquier. Pour chaque position, nous sélectionnons une tuile et une orientation aléatoirement [3].

Afin de créer les cartes de classes des images synthétiques, une fonction de transformation du rendu fut ajoutée au moteur 3D. Cette transformation retire les

---

2. Tiré de : https ://www.unrealengine.com/marketplace/profile/MAWI%20United%20GmbH

3. Blueprint disponible au : https ://blueprintue.com/blueprint/g4akk0wv/

Figure 3.2 Exemple d'éléments commerciaux disponibles pour Unreal Engine

effets d'éclairage et transforme les textures des modèles en couleurs unies. Chaque pixel de ces images de classes possède donc une couleur identifiant la classe qu'il représente [4].

3.3 EXTRACTION DE L'INFORMATION DE L'ENVIRONNEMENT VIRTUEL

L'extraction des images s'effectue en deux phases. Premièrement, la caméra est déplacée de façon aléatoire. Ensuite, deux captures d'écran sont prises : la vue synthétique et la carte des classes associée. Ces opérations sont répétées un certain nombre de fois avant de générer une nouvelle carte et de recommencer [5].

---

4. Blueprint disponible au : https ://blueprintue.com/blueprint/51f-wyuf/

5. Blueprint disponible au : https ://blueprintue.com/blueprint/ah8ql9p_/

Figure 3.3 Exemple de tuile de génération procédurale

Figure 3.4 Exemple d'image saisie dans l'environnement virtuel

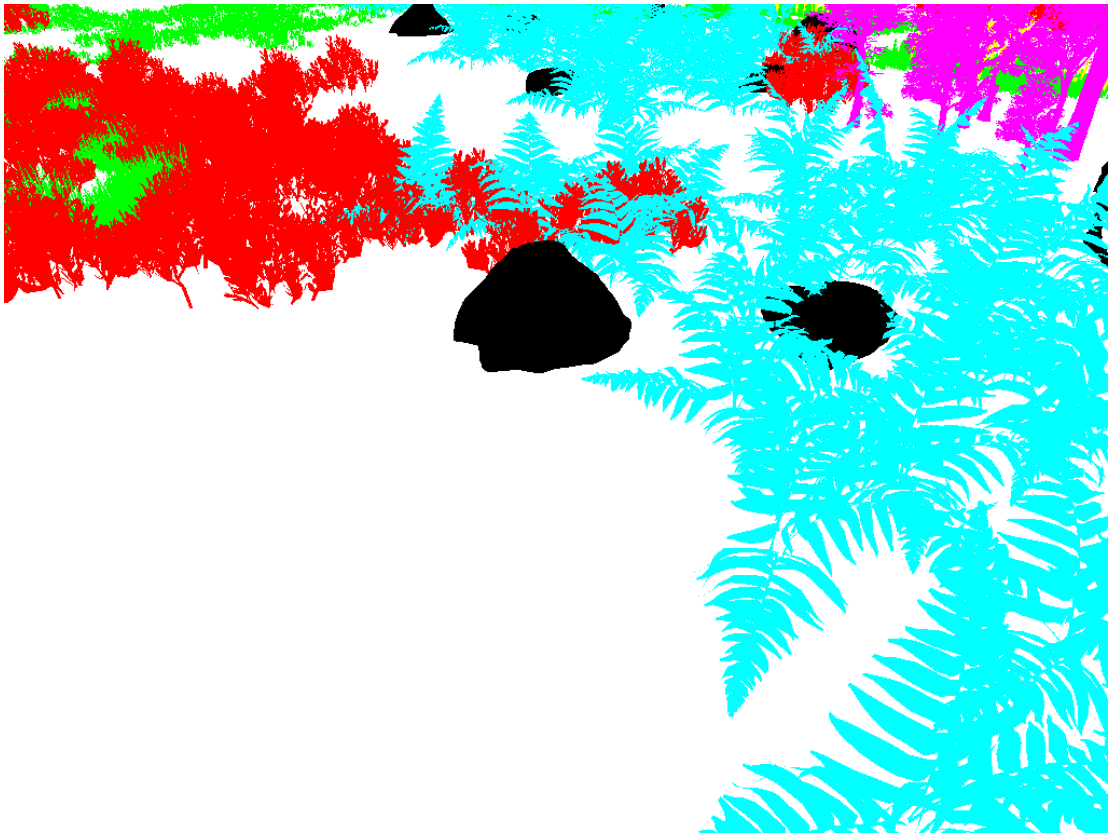

Figure 3.5 Exemple de carte des classes saisie dans l'environnement virtuel

La Figure 3.4 montre un exemple de la capture d'image synthétique et la Figure 3.5 montre la carte des classes correspondante à cette image. Il est possible de générer environ une image par seconde avec cette méthode. Ainsi, un JD constitué de plusieurs dizaines de milliers d'images peut être généré en une nuit, permettant des itérations rapides lors de modifications à l'environnement virtuel.

UTILISATION D'ÉLÉMENTS COMMERCIAUX UNREAL ENGINE

Il a ultimement été décidé d'acheter des éléments visuels commerciaux afin de faciliter le travail de génération et la qualité des images de synthèse, puisque le réalisme des images de synthèse a un impact sur la qualité de l'entraînement. Deux environnements préconçus ont été choisis pour leur réalisme et leur facilité d'utilisation. Les deux proviennent de la compagnie NatureManufacture [6]. La figure 3.4 montre un exemple tiré de ces éléments visuels à partir desquels le processus de génération d'image et de cartes de classes a été effectué pour l'entraînement final.

3.4 ACQUISITION D'IMAGES RÉELLES

Une première saisie d'imagerie a été effectué durant la récolte 2017. Des caméras de type « dashcams » ont été montées sur deux tracteurs de récolte afin de capturer en vidéo ce qui se trouve devant le tracteur. Les caméras (Viofo A119S) capturent des images de 2 mégapixels à une vitesse de 5 images par seconde et sont équipées de GPS afin d'intégrer des données de positionnement. Environ 260G d'images furent saisies durant cette période.

La Figure 3.7 présente un exemple des données capturées à la récolte. Le projet original de ce mémoire visait à automatiser certaines étapes de la récolte et

---

6. Site web : https ://naturemanufacture.com/

Figure 3.6 Capture d'écran de l'environnement Meadow par NatureManufacture

Figure 3.7 Exemple d'image capturée en « dashcam »

l'emphase était donc sur le tracteur et l'accessoire de récolte plutôt que le champ lui-même et cette saisie était faite en conséquence. Ces images sont de basse résolution par rapport à ce qui a été effectué en 2018. Elles ont cependant permis les tests préliminaires au début de 2018 avant qu'il soit possible d'effectuer de nouvelles saisies pendant l'été.

La saisie d'image de 2018 a été influencée par l'expérience de 2017 et les changements d'orientation du mémoire qui nécessite davantage de données de meilleure qualité sur les champs eux-mêmes. Un drone DJI Phantom 4 fut acquis pour permettre ces saisies. Cet appareil permet de capturer des images allant jusqu'à 12 mégapixels et possède l'avantage de pouvoir capturer des images d'un champ sans le perturber comme un véhicule au sol le ferait. Ce dernier point s'avère très important pour la saisie en dehors de la période de récolte car la circulation de véhicules dans un champ doit être évitée.

Figure 3.8 Drone utilisé pour la saisie d'images terrain

Les cartes de classes des images réelles ont dû être réalisées manuellement. Plusieurs outils d'annotation d'images fonctionnent en définissant une boite rectangulaire autour d'un ou plusieurs sujets dans une image. Bien que des images annotées selon ce principe auraient fonctionné pour notre entraînement, considérant la petite taille de notre JD, nous voulions que ces images soient très bien segmentées afin de générer le meilleur résultat possible. Par conséquent, il faut donc utiliser un outil capable de générer des annotations pixel par pixel. Nous obtenons ainsi des annotations sous la même forme que les JD d'images virtuelles et qui peuvent être entrées directement dans le RN sans traitement supplémentaire.

L'outil se nomme PixelAnnotationTool (Breheret, 2017) et est disponible librement sur GitHub. Il utilise une approche semi-automatique basée sur l'algorithme de lignes de partage des eaux (Wikipédia, 2020) de la librairie de traitement d'images OpenCV. L'utilisateur place des marqueurs de classe sur des sections de l'image et l'algorithme tente ensuite de catégoriser tous les pixels de l'image en fonction de ces marqueurs. L'utilisateur peut par la suite placer de nouveaux marqueurs pour indiquer les erreurs et le processus se répète jusqu'à ce que la cartographie de classe soit adéquate. Cette méthode sauve un temps considérable en comparaison avec le processus complètement manuel qui aurait été fait avec un outil comme Photoshop ou GIMP. Construire les classes d'une image prend quelques minutes avec cet outil et permet de le faire à un niveau de précision difficile à reproduire sans assistance.

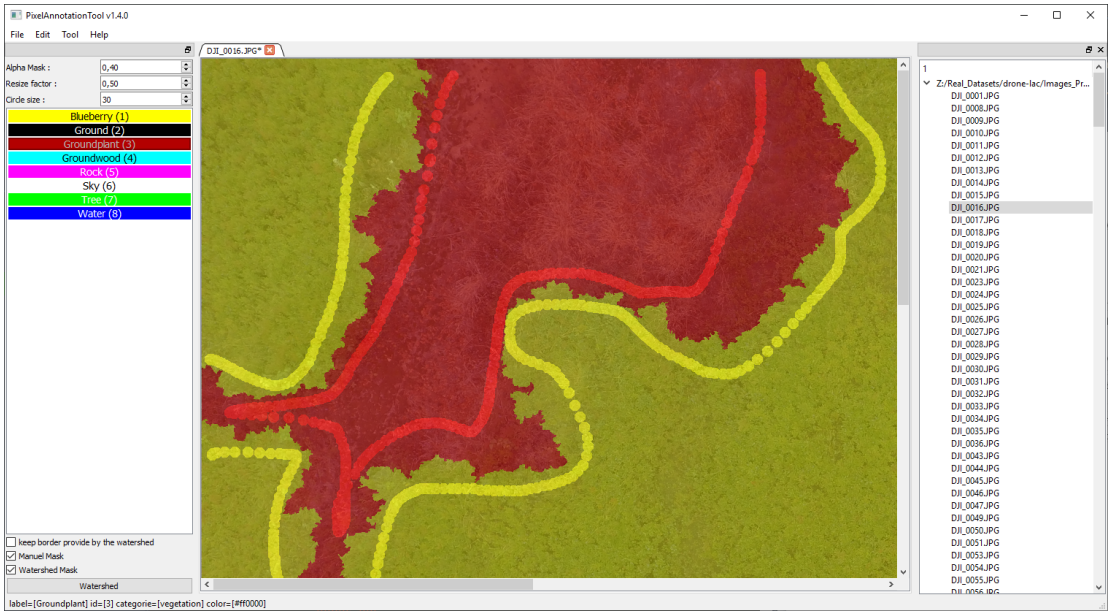

Figure 3.9 Capture d'écran de l'interface de PixelAnnotationTool

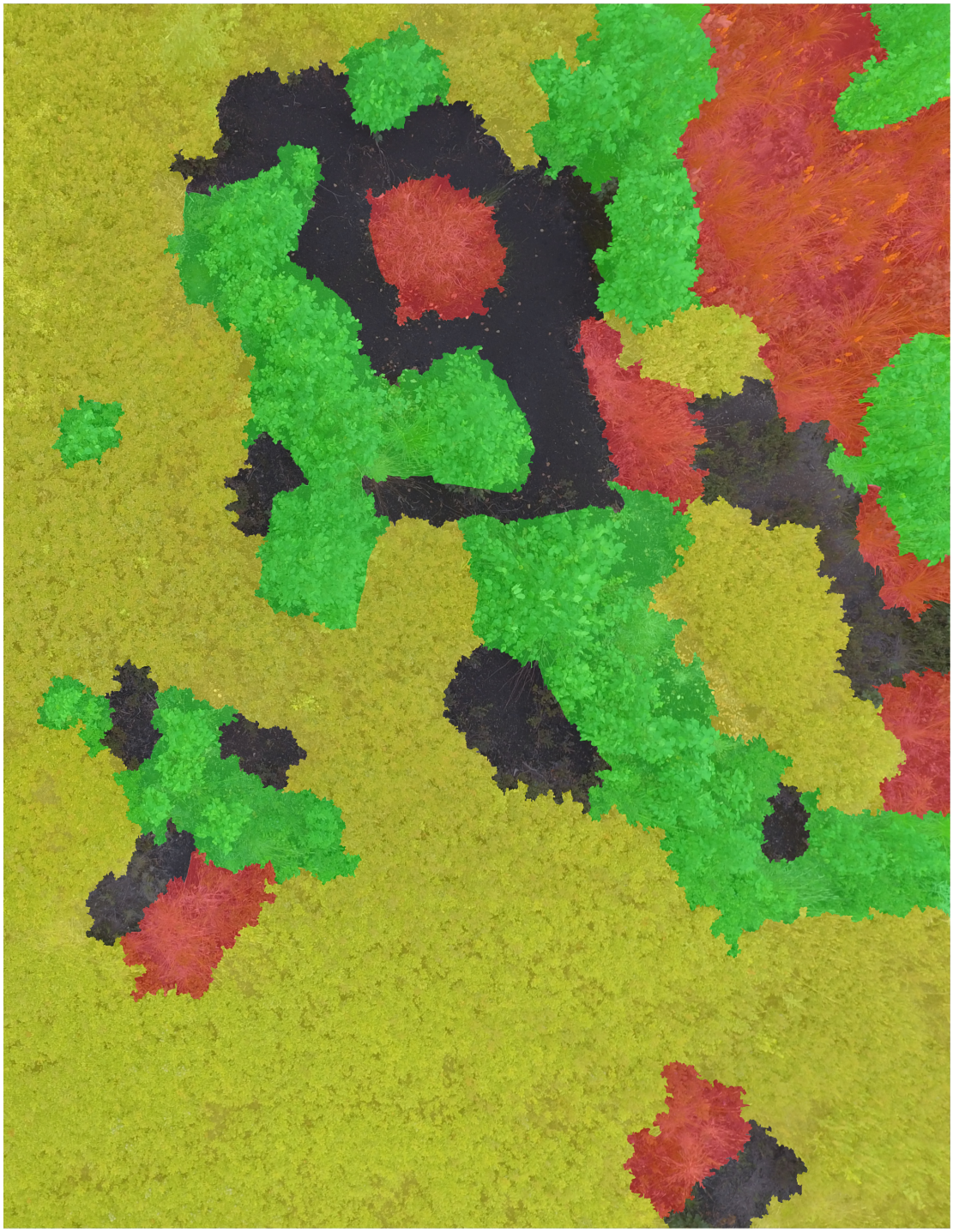

Figure 3.10 Exemple d'une carte de classe générée avec PixelAnnotationTool.

CHAPITRE IV

VALIDATION DE LA MÉTHODOLOGIE AVEC DES JEUX DE DONNÉES EXISTANTS

Avant d'aller plus loin dans les travaux de création de JD synthétiques et le travail d'acquisition de données réelles sur le terrain, il est important de valider la méthodologie sur des données existantes. Les résultats de cette validation permettront de confirmer la démarche et possiblement de modifier certains éléments.

Nous avons recueilli 3 JD (2 synthétiques et 1 réel) issus du domaine de la conduite automobile automatisée. Ce domaine est en plein essor et les études sur le sujet sont nombreuses. Les défis sont similaires, à savoir d'entraîner des RN devant identifier les objets perçus par les caméras dans des situations très diverses. Plusieurs travaux universitaires sur le sujet s'accompagnent de données et nous pouvons les soumettre au même processus d'entraînement et de validation que ce que nous comptons faire avec nos données agricoles.

Pour ces tests, nous ferons une classification binaire (route et le reste des images) afin de simplifier l'expérimentation. L'entraînement est plus simple et permet donc d'expérimenter plus rapidement.

4.1 JEU DE DONNÉES SYNTHÉTIQUES #1 THE SYNTHIA DATASET : A LARGE COLLECTION OF SYNTHETIC IMAGES FOR SEMANTIC SEGMENTATION OF URBAN SCENES

Ce JD est peu photoréaliste et a l'aspect d'un vieux jeu vidéo. Son utilisation nous permet d'avoir une base minimale qui permettra d'évaluer si un JD synthétique sommaire permet d'avoir une influence sur l'entraînement.

La création de ce JD s'est faite d'une façon similaire à celle utilisée dans nos travaux. Les auteurs ont utilisé le moteur de jeu vidéo Unity afin de créer une ville virtuelle. Il devient ensuite simple de générer les images et la segmentation de ces dernières puisque l'on contrôle complètement l'engin de simulation. Une voiture virtuelle se déplace dans la ville et capture les images lors de son parcours (Ros *et al.*, 2016).

L'équipe l'ayant mis en place a obtenu des résultats intéressants dans les cas d'entraînement sur les données synthétiques seules et sur la combinaison de données synthétiques et réelles (90%+ d'exactitude sur la classe route). Ces résultats laissent croire que les données synthétiques n'ont pas besoin d'être photoréalistes pour avoir une influence notable sur l'entraînement.

4.2 JEU DE DONNÉES SYNTHÉTIQUES #2 PLAYING FOR DATA : GROUND TRUTH FROM COMPUTER GAMES (GTA5)

Le deuxième JD a été créé à partir du jeu vidéo Grand Theft Auto 5, qui se déroule dans des environnements urbains et souvent au volant d'un véhicule. La technologique utilisée dans ce jeu est supérieure à Synthia mais n'est pas ce qu'on peut considérer comme moderne.

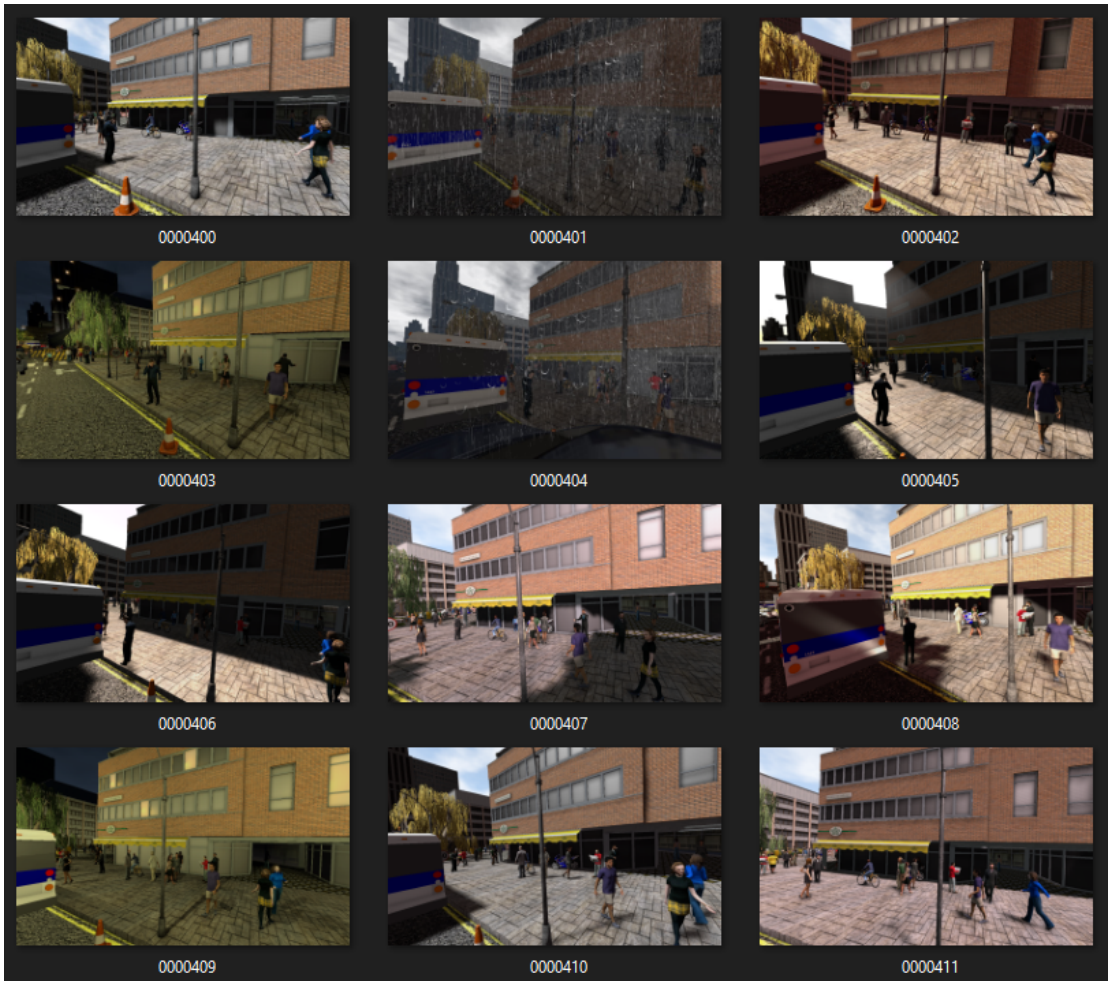

Figure 4.1 Exemples des images du JD SYNTHIA

La procédure pour générer les données diffère de celle utilisée pour Synthia. Puisqu'il s'agit d'un logiciel commercial qui ne pouvait être modifié pour générer les cartes de segmentation des images, les auteurs ont plutôt conçu un outil d'aide à l'assignation manuelle des catégories, permettant de réduire le temps consacré à cette étape (Richter *et al.*, 2016).

Ces images représentent ce qu'il est possible d'obtenir en consacrant des efforts plus importants à l'élaboration de la simulation sans pour autant tomber dans la technologie dernier cri. Nous avons donc un comparatif permettant d'évaluer l'impact du photoréalisme des images synthétiques. Dans ce cas aussi, les auteurs ont constaté une amélioration de la performance des RN entraînés sur des données réelles lorsqu'elles sont augmentées de ces données synthétiques. Cependant ils ont utilisé les données synthétiques comme complément aux données réelles et non comme source principale d'entraînement, ce qui diffère de notre cas.

4.3 JEU DE DONNÉES RÉELLES THE MAPILLARY VISTAS DATASET FOR SEMANTIC UNDERSTANDING OF STREET SCENES

Ce JD est le plus imposant en termes de quantité et qualité d'image dans ceux disponibles publiquement pour la recherche (plusieurs groupes privés ont sans doute des collections beaucoup plus complètes). Des images urbaines ont été saisies un peu partout dans le monde dans de nombreuses conditions et la segmentation faite sur les images est très bien exécutée (Neuhold *et al.*, 2017).

Ces images représentent le meilleur scénario possible, soit de disposer d'une grande quantité de données réelles variées. Elles serviront à établir la performance de segmentation maximale que peut atteindre notre RN et par conséquent servira de base de comparaison à toutes les autres expérimentations. Plus un entraînement s'approche du résultat obtenu en entraînant directement sur ces données,

Figure 4.2 Exemples des images du JD GTA5

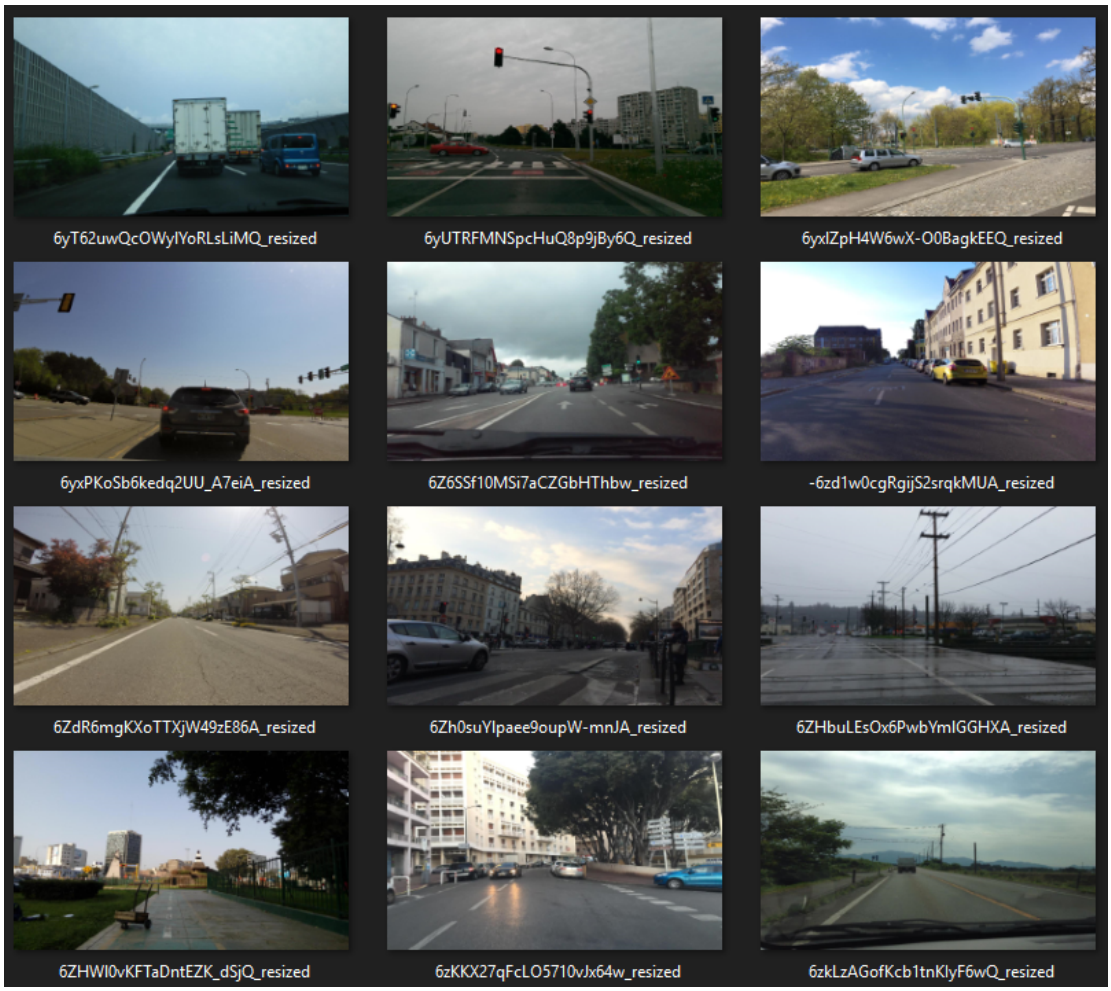

Figure 4.3 Exemples des images du JD Mapillary

plus on peut considérer la méthode d'entraînement performante. Nous pigerons également dans ces images lorsque nous voulons ajouter des données réelles aux entraînements.

4.4 EXPÉRIMENTATION

- Réseau
  - Architecture : Segnet

    - Entrée : 384x384 pixels, RGB
    - Sortie : 384x384 pixels, contenant chacun une valeur entre 0 et 1 représentant la probabilité d'être un pixel représentant une route
- Entrainement
    - Nombre d'images par lot : 4
    - Nombre de lots par époque : 250 (1000 images par époque)
    - Nombre d'époques : Variable (voir section résultats pour les détails)
    - Optimisateur : Adadelta
- Évaluation
    - JD d'images : Mapillary
    - Nombre d'images par lot : 30
    - Nombre de lots : 50 (1500 images par validation)
- Type de modifications appliquées aux images
    - Inversion horizontale et verticale
    - Ajout de flou de mouvement
    - Ajout de bruit à l'image (pixels blancs et noirs aléatoire)

4.5 RÉSULTATS

Les images de test du JD réel utilisé contiennent 22% de pixel représentant des routes. Un algorithme prédisant que tous les pixels ne sont pas de la route obtiendrait donc une précision de 78%. Ce type de situation est courant dans les cas réels, il est rare que les données observées aient une réparation parfaite entre chaque classe. Notre RN doit donc obtenir une performance significativement supérieure à ce seuil de 78% pour considérer qu'il a appris à distinguer les sections d'images.

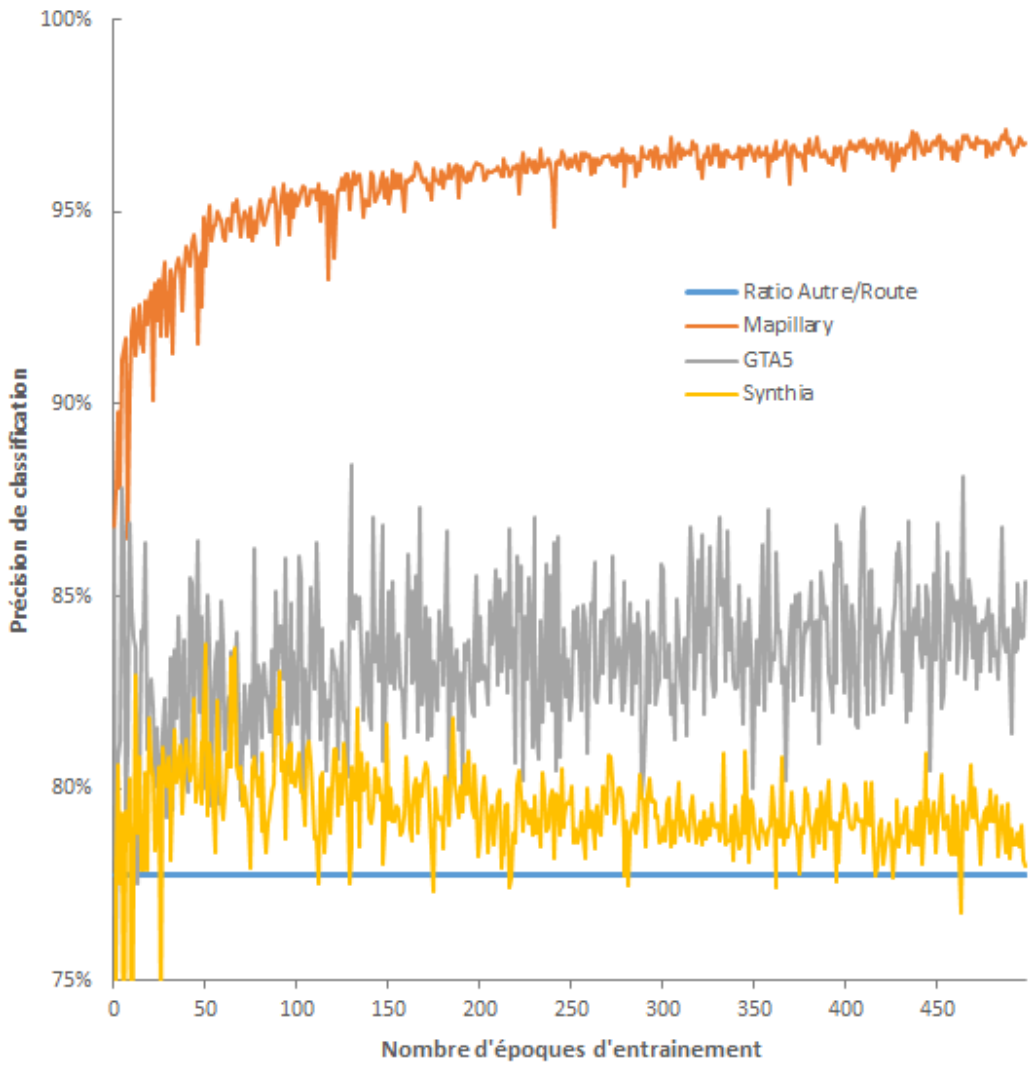

Figure 4.4 Résultat d'entraînement sur les JD d'images complets

PERFORMANCE DE BASE DES JEUX D'IMAGES UTILISÉS

Les images de l'ensemble Mapillary ont entraîné les meilleurs résultats par une marge très claire par rapport aux images synthétiques. On peut également constater que la précision continue d'augmenter même après 500 époques : un entraînement plus long aurait donc probablement engendré une précision encore supérieure. Les JD d'images synthétiques ont plafonné beaucoup plus rapidement.

Le JD d'images Synthia a donné une performance très proche du ratio route / non-route des pixels. On peut aussi voir une baisse de la performance avec l'augmentation du nombre d'époques, démontrant un surapprentissage des caractéristiques du JD d'images virtuelles qui dégrade la performance sur les vraies images.

Le JD d'images de meilleure qualité visuelle (GTA5) donne une performance supérieure, confirmant que le photoréalisme améliore la performance du RN lorsqu'on l'évalue sur des images réelles. Il semble donc important de générer des images de synthèse les plus réalistes possible. Pour cette raison, nous allons utiliser le JD d'images basées sur GTA5 pour les tests suivants en le supposant représentatif du type d'image synthétiques qu'il serait possible de générer pour d'autres situations.

PERFORMANCE DE LA MÉTHODE D'ADAPTATION DE DOMAINE

Ce test ne s'avère pas concluant sur l'utilité de l'adaptation de domaine. Les résultats sont essentiellement les mêmes avec et sans la branche ajoutée au RN. Il est possible que cette stratégie n'ajoute rien à notre cas ou que des questions d'usage empêche ses avantages de créer une différence observable, par exemple une configuration trop agressive des paramètres.

Avec les 2 tests précédent, on constate qu'il y a peu d'intérêt à se rendre à 500 époques. Même sur l'entraînement basé sur le JD d'image réelles complet, l'amélio-

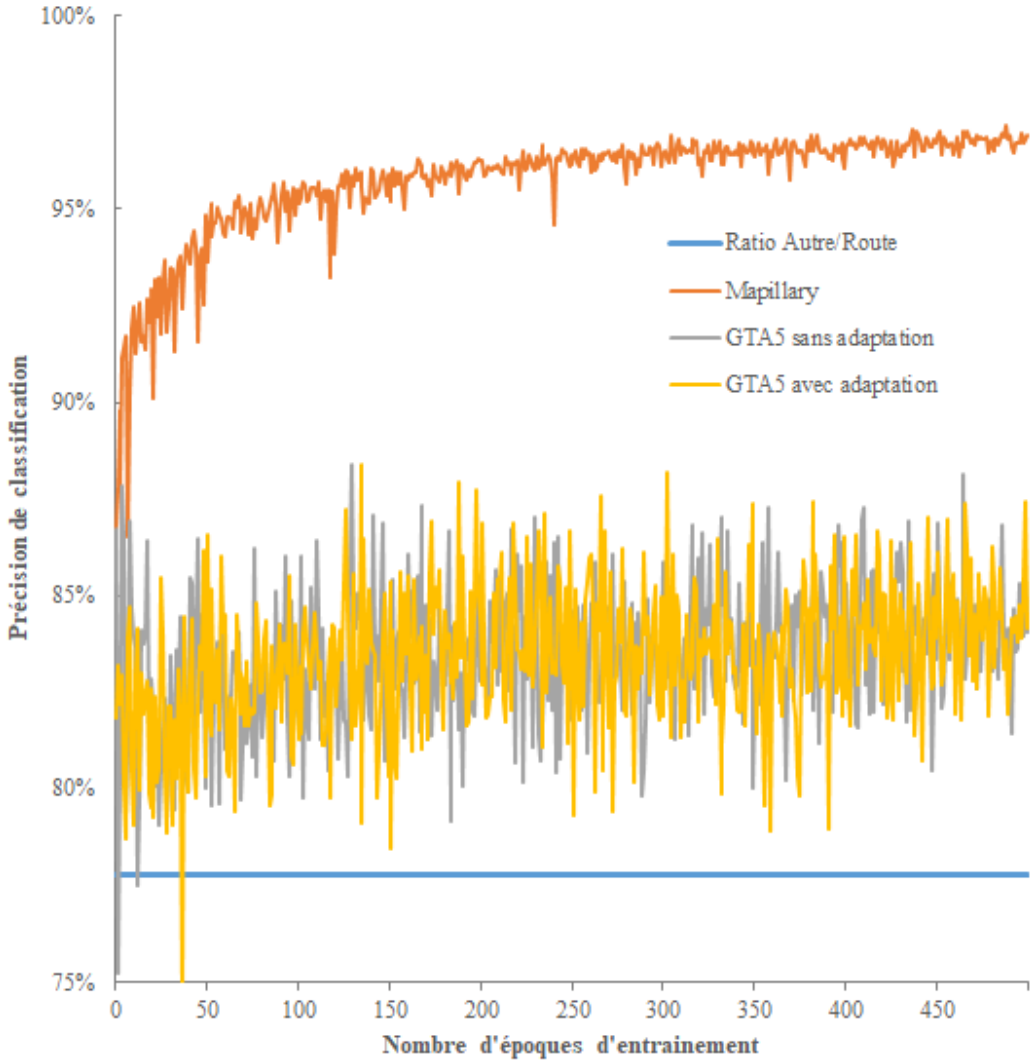

Figure 4.5 Résultat d'entraînement avec adaptation de domaine

ration obtenue au-delà de 200 époques est faible. Sur les entraînements avec images synthétiques, on atteint un plateau après quelques dizaines d'époques seulement. Nous allons donc réduire le nombre d'époque à 200 pour les tests suivants.

PERFORMANCE SUR L'ENTRAÎNEMENT UTILISANT DE PETITS JEUX DE DONNÉES RÉELLES

Une autre stratégie pouvant être utilisée dans un cas comme le nôtre est l'entraînement sur une petite quantité de données. Ce type d'entraînement donnera un RN qui est moins performant à généraliser sur des contextes différents des images d'entraînement. Cependant, identifier manuellement quelques dizaines d'images est rapide et plus simple que de développer des environnements virtuels.

Pour ce test nous avons utilisé des JD de 100 et 10 images issues de Mapillary. On peut constater deux points intéressants.

— L'entraînement sur GTA5 n'a pas réussi à surpasser l'entraînement sur le JD de 10 images. Il semble donc très pertinent d'inclure dans les entraînements des images réelles, même en très petite quantité, car elles ont une grande influence sur le résultat.

— On constate le rendement décroissant de l'augmentation de la quantité d'images sur la performance. La différence entre 10, 100, et 15000+ (JD complet) images va en s'amenuisant. Dans un contexte de conduite automobile automatisée ces différences sont cruciales. Par contre, dans un cas ou l'impact d'une mauvaise classification a des conséquences moins grandes, il pourrait être réaliste d'utiliser des RN entraînés sur quelques dizaines/centaines d'images et obtenir une performance suffisante pour des cas d'usage moins critiques.

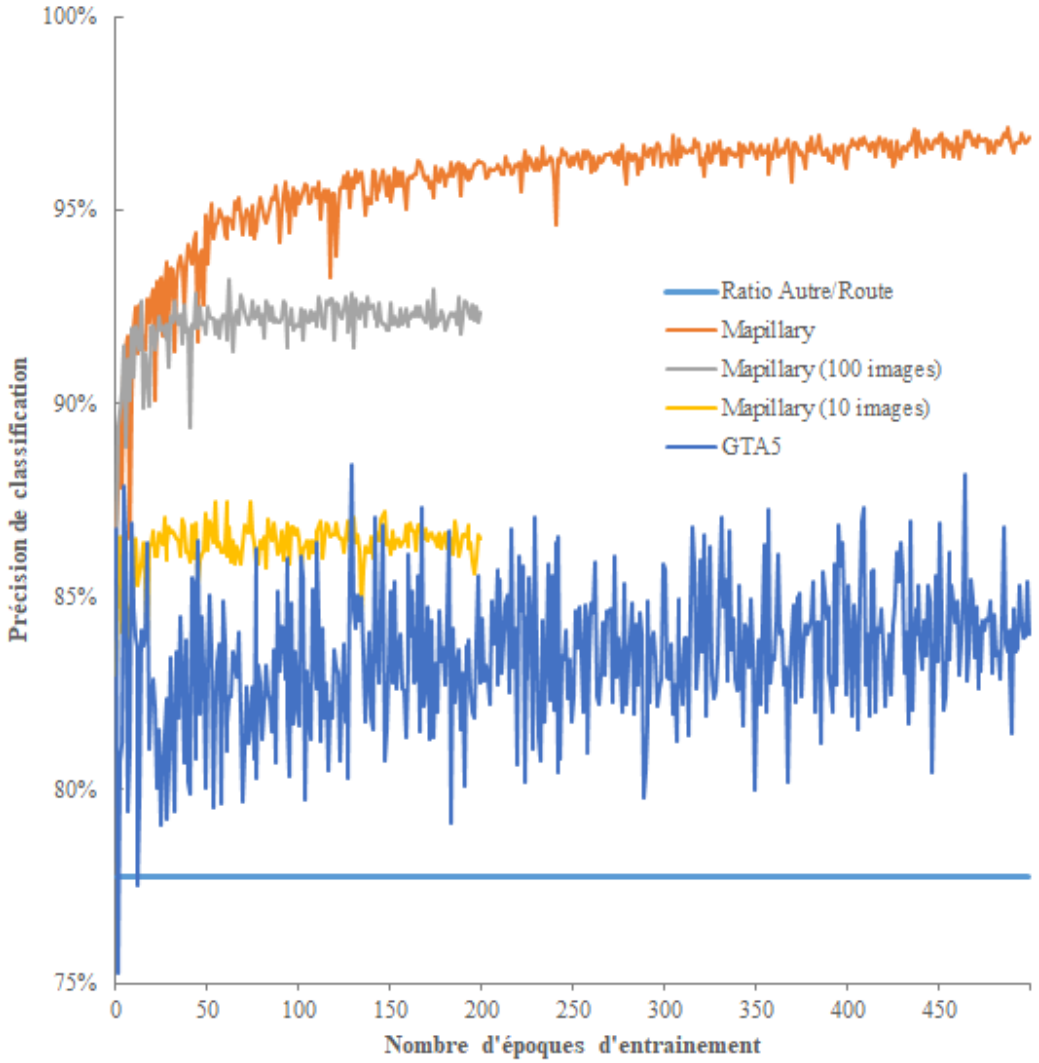

Figure 4.6 Résultat d'entraînement avec de petits JD réel

PERFORMANCE DE L'AJOUT D'IMAGES RÉELLES AUX DONNÉES SYNTHÉTIQUES DE FAÇON GLOBALE

L'ajout de 100 images au JD d'images GTA5 a engendré une différence perceptible. On constate que le RN apprend beaucoup plus lentement que s'il avait seulement les 100 images réelles, mais que la tendance est à la hausse et surpasse l'utilisation du JD d'images GTA5. On est cependant encore loin de la performance atteinte en utilisation les 100 images réelles après 200 époques alors qu'on a atteint le plateau en moins de 100 époques sur les 100 images réelles.

L'énorme déséquilibre (25 000 contre 100) entre la quantité d'images virtuelles et d'images réelles nuit à l'entraînement. Il est peu efficace pour l'algorithme d'optimiser pour les 100 images, considérant à quel point elles sont rares. Une stratégie permettant de compenser le débalancement doit donc être mise en place pour forcer l'entraînement à accorder davantage d'importance aux images réelles malgré leur nombre réduit. Une telle stratégie devrait augmenter la vitesse d'apprentissage et idéalement améliorer la performance de segmentation.

PERFORMANCE DE L'ALTERNANCE PAR ÉPOQUE DES IMAGES RÉELLES ET SYNTHÉTIQUES

Pour ce test, nous avons fait alterner les époques entre les images synthétiques et 10 images réelles. On remarque deux éléments sur ces essais :

— Nous avons obtenu un résultat supérieur à l'usage d'un seul des JD, confirmant l'utilité d'intégrer des images synthétiques à l'entraînement. Cette intégration a permis d'obtenir une performance similaire à l'entraînement utilisant 100 images réelles à l'aide de seulement 10 de ces images.

— La performance à la fin des époques variait considérablement, selon la nature (réelle ou synthétique) des images du JD utilisé dans la dernière

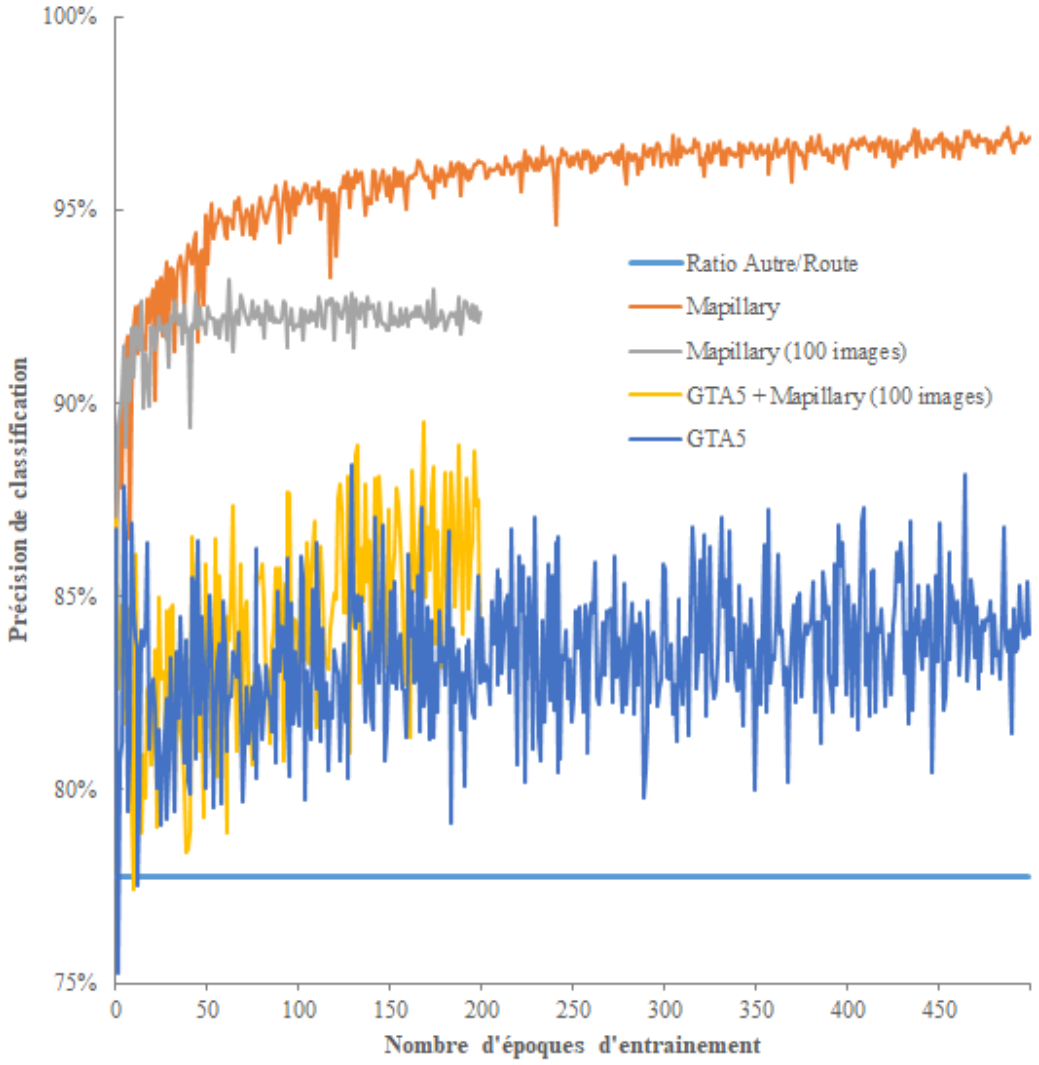

Figure 4.7 Résultat d'entraînement avec des images réelles ajoutées globalement aux données synthétiques

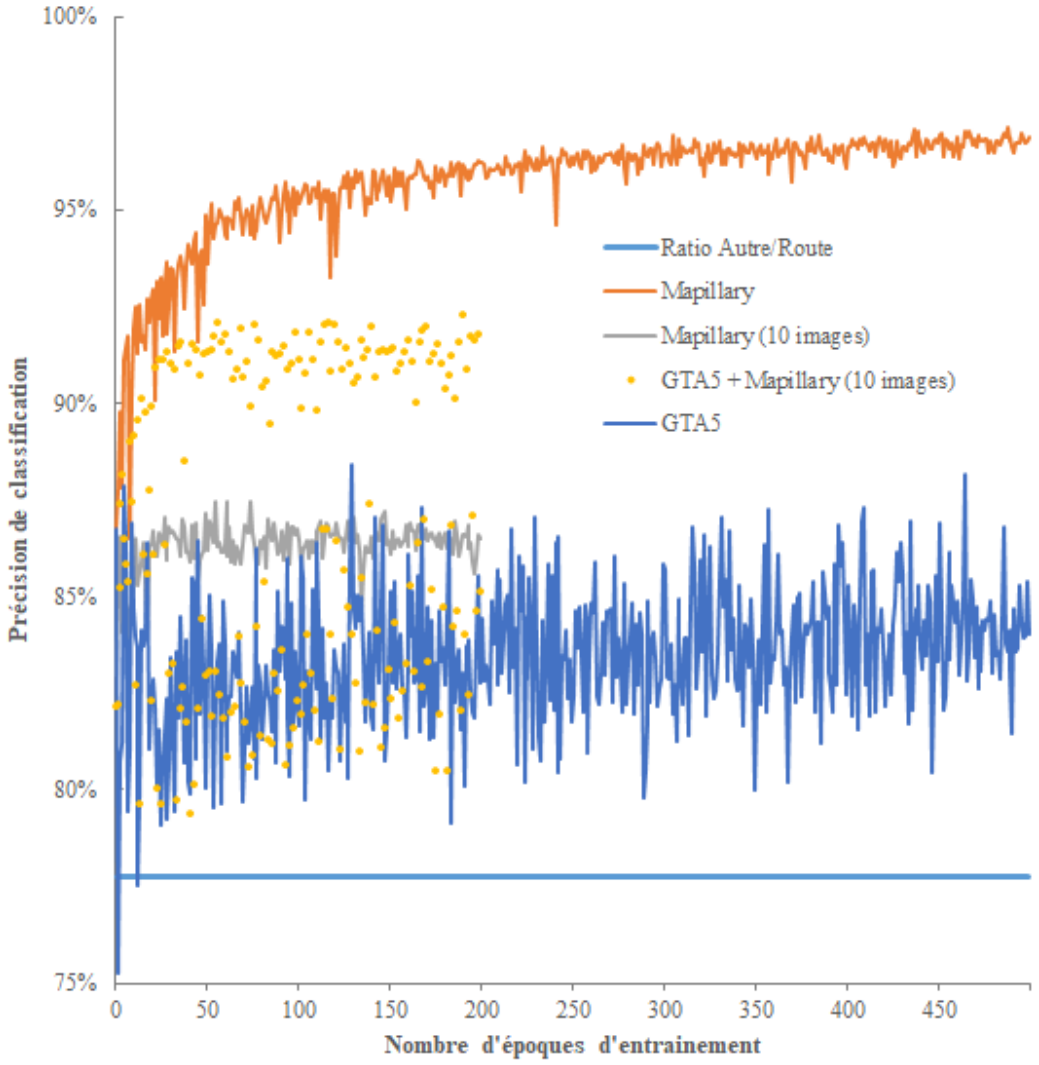

Figure 4.8 Résultat d'entraînement en alternance par époque des images réelles et synthétiques

époque. Cette énorme variation pourrait être due à la méthode utilisée pour intégrer les 2 JD d'images et pourrait être amoindrie en utilisant d'autres stratégies à cette fin.

PERFORMANCE DE L'ALTERNANCE ALÉATOIRE DES IMAGES RÉELLES ET SYNTHÉTIQUES

Pour réduire la variance des résultats de l'alternance par époque, nous avons cette fois utilisé une alternance aléatoire sur chaque lot, donnant une chance équivalente à l'utilisation d'images synthétiques ou d'images réelles. Notre hypothèse étant que cette méthode éviterait la grande variance des résultats en évitant l'utilisation de gros blocs continus d'images d'un même JD.

Nous avons réduit la variance des résultats, mais nous avons également réduit la performance de la segmentation, il semble que de compléter l'entraînement par une passe en continue sur des images réelles donne un meilleur résultat qu'une alternance perpétuelle. Nous voulions réduire la variance en assumant qu'un entraînement sans ces grandes variations serait en mesure de donner une meilleure performance, ce qui s'avère faux dans notre contexte.

PERFORMANCE DE L'ALTERNANCE ALÉATOIRE DES IMAGES RÉELLES ET SYNTHÉTIQUES SUIVI D'UN ENTRAÎNEMENT SUR IMAGES RÉELLES

Pour cet essai nous sommes parti du meilleur résultat obtenu lors de l'alternance par époque (époque 191) et nous avons poursuivi l'entraînement en utilisant uniquement le JD de 10 images réelles. Ces entraînements supplémentaires n'ont pas apporté d'amélioration sur la performance du RN : la performance demeure similaire à celle obtenue lors de l'entraînement sur 100 images. Il n'y a pas de diminution de la performance non plus, ce qui aurait été possible considérant que

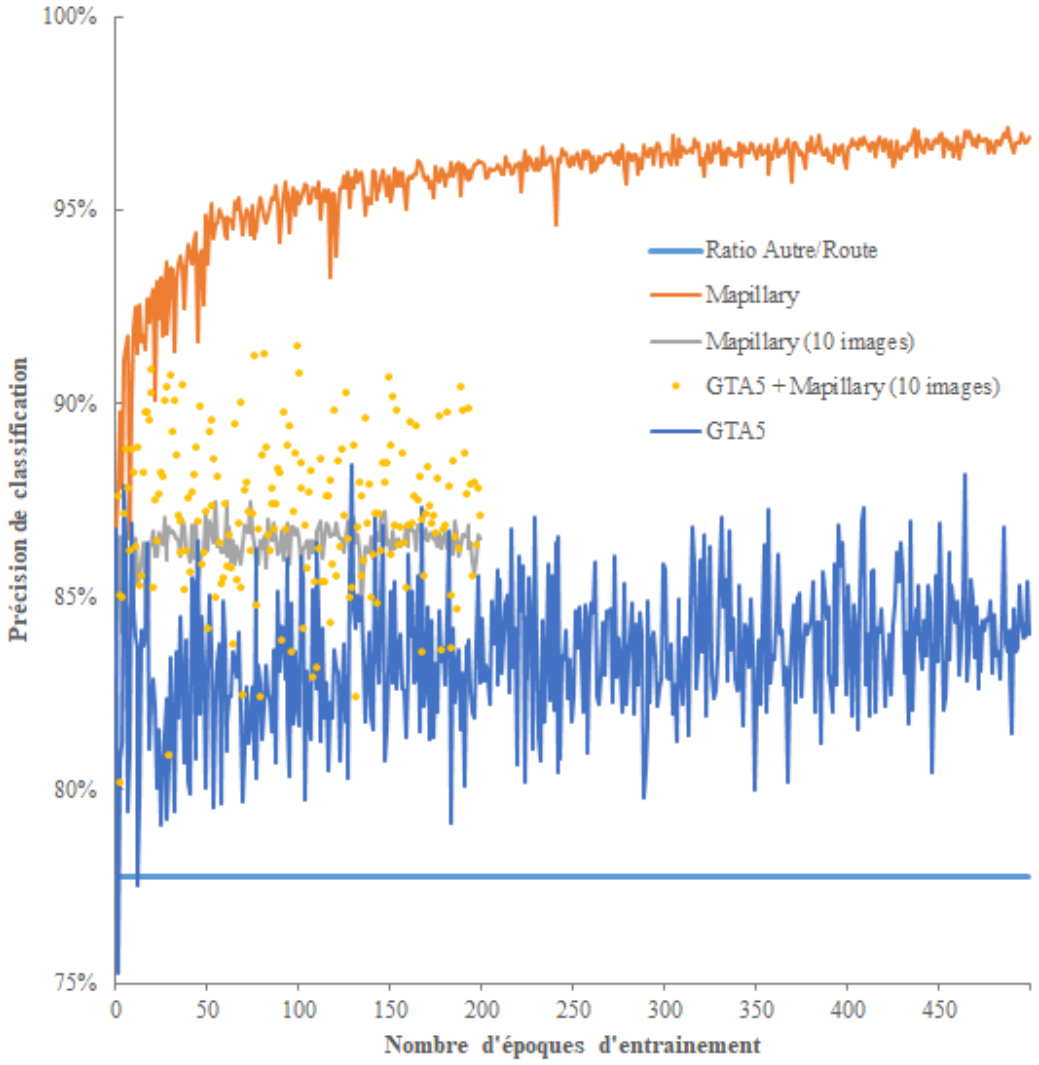

Figure 4.9 Résultat d'entraînement en alternance aléatoire des images réelles et synthétiques

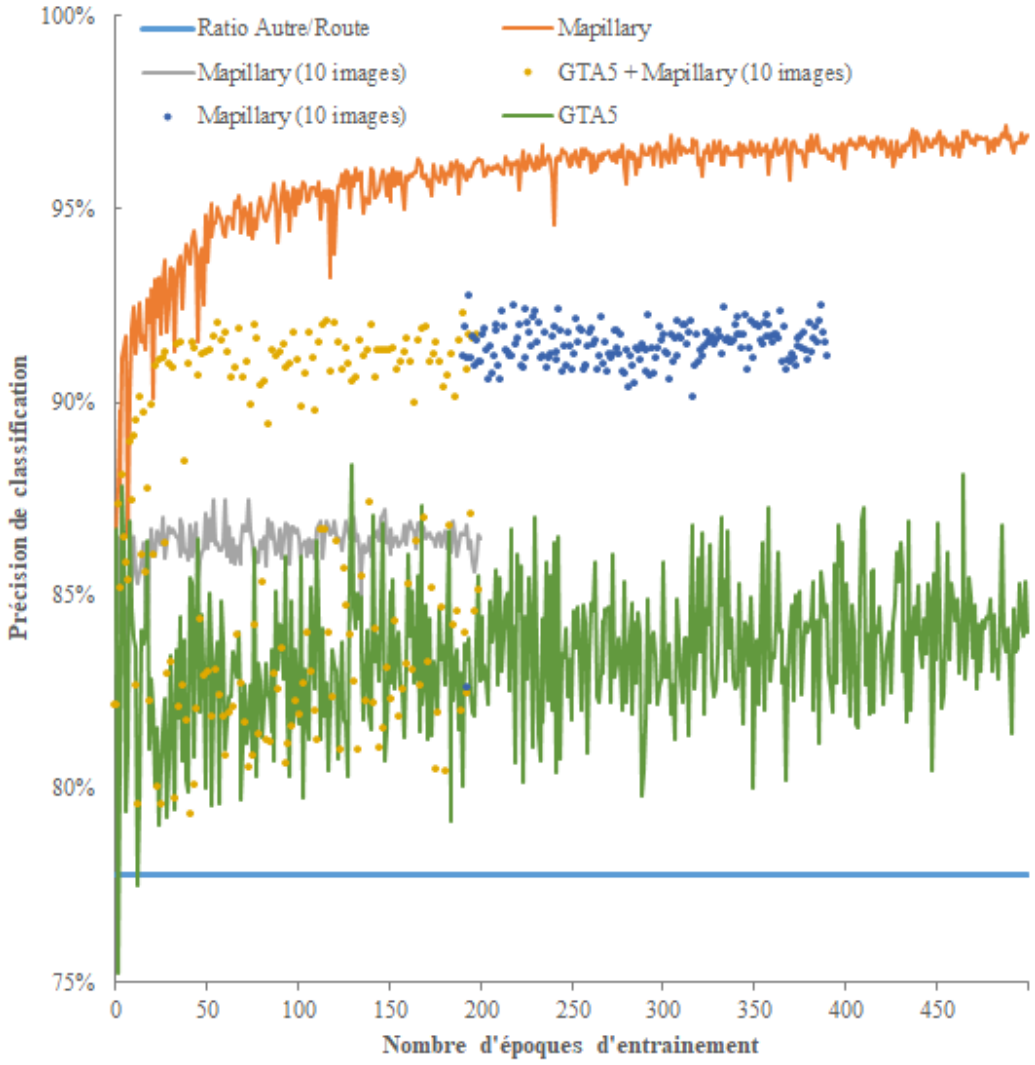

Figure 4.10 Résultat d'entraînement en alternance aléatoire des images réelles et synthétiques suivi d'images réelles

l'entraînement sur les 10 images réelles seulement avait une performance inférieure à ce qui a été observé dans cet essai.

4.6 APPRENTISSAGES

- Lors de nos tests, il s'est avéré impossible de dépasser les environs de 92% en se limitant à l'usage de 10 images réelles.
- Nous avons cependant eu un résultat similaire en utilisant 100 images réelles ou 10 images réelles + un JD d'image synthétiques.
- Mélanger les images réelles directement au JD d'images synthétiques apporte peu d'amélioration. Il semble que le ratio n'est pas suffisamment élevé pour influencer l'entraînement. Il est possible d'améliorer la performance en modifiant l'ordre et la quantité des images de chaque JD.
- Le JD virtuel peu réaliste (Synthia) s'est avéré inutile, puisqu'il est possible d'obtenir un résultat équivalent avec un entraînement sur 10 images réelles, ce qui serait très rapide à accomplir en situation réelle.
- L'adaptation de domaine n'a pas apporté de différence notable. Est-ce dû à l'inefficacité de la méthode ou à d'autres facteurs ? Il est décidé d'abandonner cette méthode considérant sa difficulté d'intégration et les résultats obtenus lors de la validation.

Les mauvais résultats de notre RN Segnet sur le JD Synthia, en comparaison aux expérimentations similaires de ses auteurs, laisse croire que notre RN n'est pas très performant sur sa capacité de généralisation. Il semble que nous ayons atteint les limites de cette architecture élémentaire et qu'il faudrait explorer d'autres architectures plus modernes pour la suite.

Ces tests ne permettent pas d'affirmer avec certitude la supériorité d'une méthode ou d'un RN par rapport à un autre. Nous avons cependant constaté l'utilité des

données synthétiques et comment il s'avère possible d'obtenir une performance permettant certains usages en utilisant un nombre réduit d'images réelles.

Certains facteurs peuvent être optimisés à la suite de ces expérimentations :

— L'architecture du RN : des architectures plus performantes que Segnet sont disponibles et seront essayées.
— Le ''data augmentation" : nous avons fait du traitement très élémentaire sur les images pour l'entraînement (rotation, miroir, ajout de flou, ajout de bruit). Un meilleur pré-traitement pourrait améliorer la capacité de généralisation.
— Favoriser le photoréalisme dans la génération des images de synthèse considérant l'impact observé.

CHAPITRE V

APPLICATION DE LA MÉTHODOLOGIE SUR DES DONNÉES AGRICOLES SYNTHÉTIQUES ET RÉELLES

Maintenant que nous avons des données synthétiques, des images agricoles réelles et une première série d'expérimentations sur un autre cas d'utilisation, nous sommes en mesure d'appliquer ces apprentissages sur notre objectif principal.

Pour ces expérimentations, nous considérons 8 classes présentées à la table 5.1. Ce nombre de classes a été choisi pour sa précision suffisante à l'usage et sa facilité de représentation dans un mode RGB (approche binaire sur chaque canal) qui facilite la création et le traitement des images. Pour le traitement par le RN, les couleurs sont converties sur une seule dimension (contenant un entier compris entre 0 et 7) représentant la classe de chaque pixel.

5.1 JEU DE DONNÉES SYNTHÉTIQUES : UNREAL ENGINE ET ENVIRONNEMENT MEADOW DE NATUREMANUFACTURE

Le JD qui a été produit à partir de l'environnement virtuel contient plus de 48 000 images prises à des hauteurs représentant le point de vue que pourrait avoir un drone. Chacune de ces images est accompagnée de la carte des classes générée automatiquement. La résolution de 1920 par 1080 pixels a été choisie car elle

R	G	B	Couleur	ID	Classe
1	1	1	Noir	0	Ciel
0	0	0	Blanc	1	Sol
1	1	0	Jaune	2	Plants de bleuets
0	1	1	Cyan	3	Débris au sol
1	0	1	Rose	4	Roches
1	0	0	Rouge	5	Plantes au sol (sauf bleuets)
0	1	0	Vert	6	Arbres
0	0	1	Bleu	7	Eau

Tableau 5.1 Classes et couleurs correspondantes

ID	Classe	Proportion des images
0	Ciel	12.3%
1	Sol	7.4%
2	Plants de bleuets	0%
3	Débris au sol	0.5%
4	Roches	2.8%
5	Plantes au sol (sauf bleuets)	30.7%
6	Arbres	45.8%
7	Eau	0.4%

Tableau 5.2 Répartition des classes dans les images virtuelles

permet d'avoir suffisamment de détails sur les images tout en maintenant la taille du JD à une taille utilisable sur l'équipement disponible à l'entraînement.

Ces images contiennent des exemples de toute les classes à l'exception des plants de bleuets, qui ne sont pas disponibles sur le marché. Un travail d'infographie pourrait pallier à cette situation mais dépasse la portée de ce mémoire.

Figure 5.1 Exemple d'image du JD virtuel Meadow

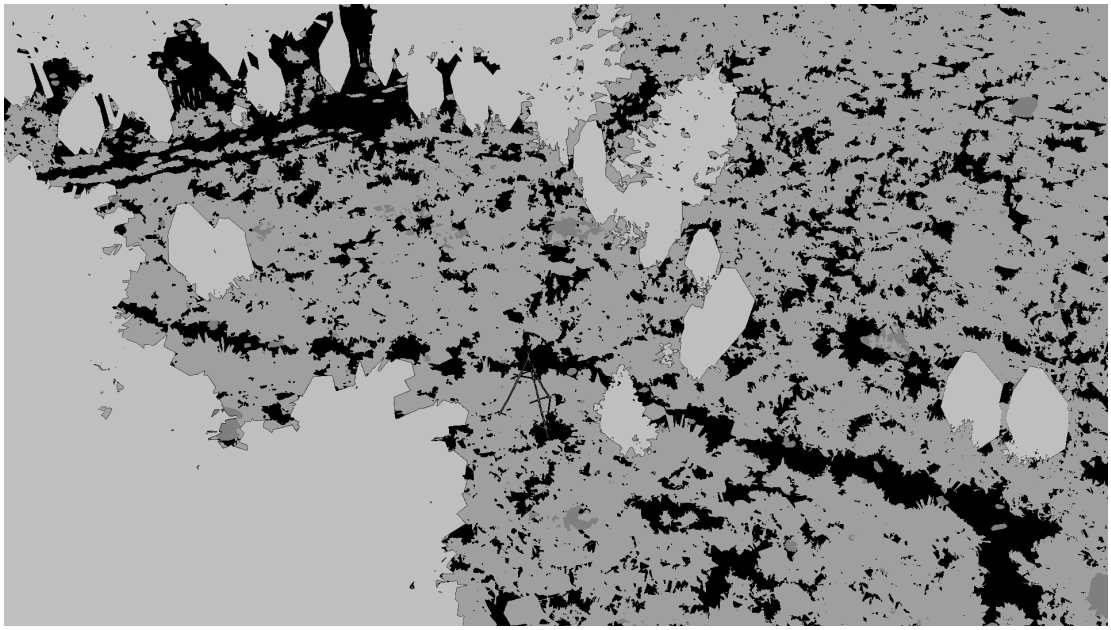

Figure 5.2 Exemple de carte de classe du JD virtuel Meadow

PARTITIONNEMENT DES IMAGES

- Total : 48 683 Images
- Entraînement et validation : 44 683 Images (92%)
  - L'allocation des images entre les deux partitions s'effectue dynamiquement lors de l'entraînement.
- Test - 4000 Images (8%)
  - Elles ont été retirées aléatoirement du jeu d'images total, ces images ne sont utilisées que pour les tests d'évaluation.

5.2 JEU DE DONNÉES RÉELLES : CAPTURE D'IMAGES EN DRONE ET GÉNÉRATION MANUELLE DES CARTES DE CLASSES

Le JD d'images réelles provient d'une sortie en drone au dessus d'un champ de bleuets sauvages contenant une variété d'autres plantes et une couverture inégale du sol par les plants de bleuets. Ce champ a plus de diversité dans le contenu de ses images et se prête bien à notre contexte d'expérimentation. La résolution native de 4000 par 3000 pixels a été conservée. 65 images ont été cartographiées de façon semi-manuelle : 50 servant à l'entraînement et 15 à la validation. Ces images ne contiennent pas toutes les classes non plus (l'eau y est absente).

Note : Les différences de résolutions entre les deux JD n'ont pas de conséquence car les zones choisies aléatoirement sur les images sont toujours ramenées à la taille d'entrée du RN.

Le calcul de proportions des classes a été fait en analysant 30 000 échantillons issus du générateur de données d'entraînements, afin de tenir compte de tout biais que pourrait avoir ce système. Les données réelles ne contiennent pas une proportion significatives de certaines classes, il est donc probable qu'un RN n'arrive pas à développer de capacité prédictive par rapport à ces classes. Nous sommes égale-

Figure 5.3 Exemple d'image du JD réel

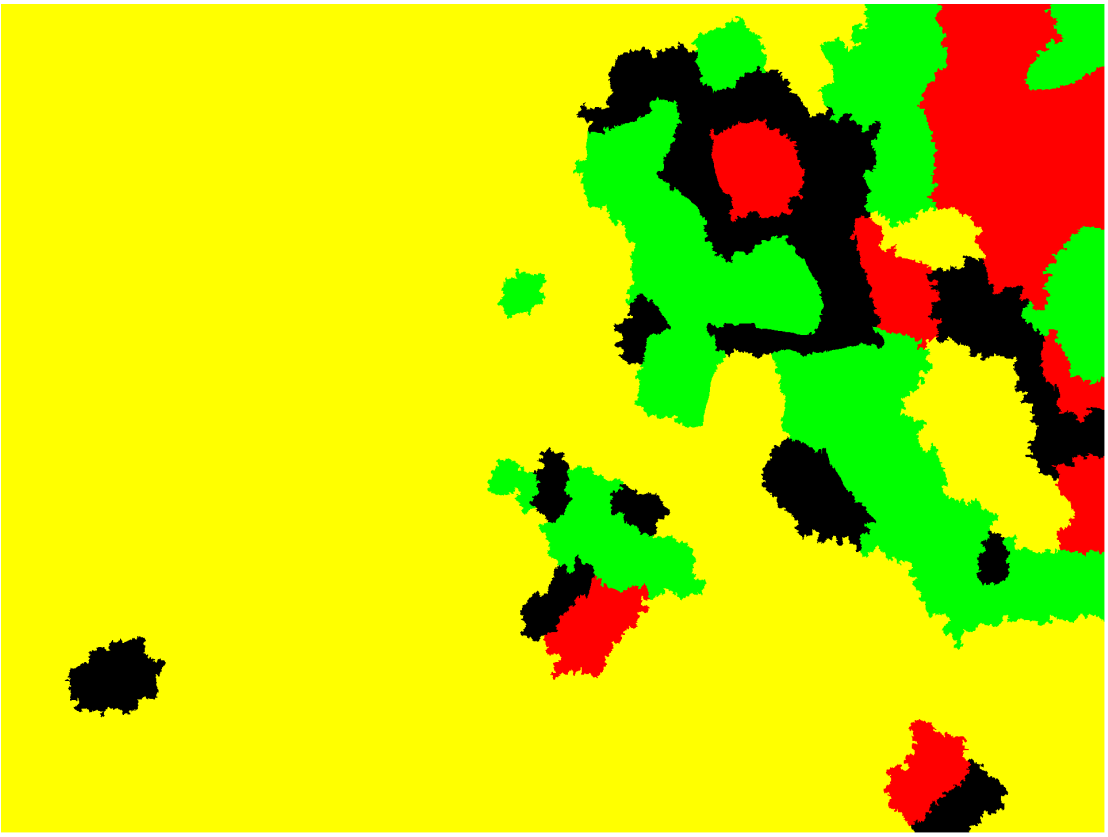

Figure 5.4 Exemple de carte de classe du JD réel

ID	Classe	Proportion des images
0	Ciel	0%
1	Sol	2.5%
2	Plants de bleuets	74%
3	Débris au sol	>0%
4	Roches	0%
5	Plantes au sol (sauf bleuets)	23%
6	Arbres	0.6%
7	Eau	>0%

Tableau 5.3 Répartition des classes dans les images réelles

ment en mesure d'affirmer qu'un RN doit avoir un succès supérieur à la classe majoritaire (environ 74%) pour avoir une utilité minimale.

PARTITIONNEMENT DES IMAGES

- Total : 65 Images
- Entraînement et validation : 50 Images (77%)
  - L'allocation des images entre les deux partitions s'effectue dynamiquement lors de l'entraînement.
- Test - 15 Images (23%)
  - Elles ont été retirées manuellement afin d'assurer des images variées. Considérant la petite taille du JD, il aurait été risqué de choisir aléatoirement ces images et d'obtenir des images trop similaires. Ces images ne sont utilisées que pour les tests d'évaluation.

5.3 NOUVEAU RÉSEAU : U-NET ET TENSORFLOW 2.0

Le RN utilisé pour ces expérimentations diffère de celui précédemment utilisé. Premièrement, l'architecture Segnet est plutôt élémentaire et passer à une construction plus moderne a fait ses preuves dans des expérimentations comparatives. Deuxièmement, le cadriciel utilisé (Tensorflow/Keras) est passé à sa version 2.0 et le code utilisé précédemment est incompatible. Plusieurs améliorations (notamment au niveau de la gestion des JD) disponibles avec cette nouvelle version auraient requis beaucoup de modifications au code initial. Le concept de l'adaptation de domaine n'ayant pas donné de résultats tangibles durant les premiers essais, il n'a pas été réimplémenté.

Pour sa construction, la partie d'encodage et les poids déjà entraînés du RN MobileNetV2 sont utilisés. Le principal avantage de cette réutilisation est d'accélérer l'entraînement en évitant de partir de zéro sur la capacité du RN à distinguer les caractéristiques des images. Tout nos essais partiront de cette base afin de s'assurer qu'elle n'influence pas les résultats finaux.

Le RN et sa mécanique d'ingestion des données sont basés sur le travail de Yann Leguilly (Leguilly, 2019) dans une application sur le JD ADE20K, lui-même inspiré de la documentation de Tensorflow. Il peut être consulté librement sur GitHub [1].

Le pré-traitement des images a été refait et certaines méthodes n'ont pas été récupérées. Il a été conservé d'inverser les images ainsi que de les recadrer/zoomer de façon aléatoire. Ces techniques ont été plus facile à mettre en place avec la nouvelle architecture tout en permettant de maximiser le nombres de permutations possibles à partir du nombres restreint d'images réelles.

---

1. https ://github.com/JeffOnGithub/tf2-unet

5.4 EXPÉRIMENTATION

- Réseau
  - Architecture : U-Net
  - Entrée : 448 x 448 pixels, RGB (3 dimensions)
  - Sortie : 448 x 448 pixels, 8 dimensions (une par classe)
- Entraînement
  - Nombre d'images par lot : 28
  - Nombre de lots par époque : 100
  - Nombre d'époques : 50
  - Optimisateur : RMSprop
- Évaluation
  - Nombre d'images : 15
  - Nombre d'images par lot : 28
  - Nombre de lots : 10
- Type de modifications appliqués aux images
  - Inversion horizontale et verticale
  - Recadrage et zoom aléatoire

5.5 RÉSULTATS

Trois métriques seront utilisées dans cette section :

Précision - Cette métrique est la proportion des items pertinents parmi l'ensemble des items proposés (Wikipédia, 2022). Elle note la véracité des prédictions de façon booléenne et peut s'exprimer en nombre de pixels (chaque pixel étant une prédiction en lui-même) ou alors en pourcentage faisait la moyenne des prédictions. Elle a l'avantage d'être simple et intuitive mais manque de subtilité. Elle

ne tient pas compte de la confiance du réseau en sa prédiction ni des déséquilibres dans les données.

Perte d'entropie croisée (crossentropy loss) - Cette métrique exprime la véracité des prédictions mais également le niveau de confiance qu'a le réseau en celles-ci sur un gradient entre 0 et 1. Cette métrique est plus nuancée que la précision et sa classification booléenne (Fortuner *et al.*, 2022). Elle a été retenue pour l'entraînement de notre réseau [2] car elle permet de doser l'effet de la rétro-propagation (forte ou faible, en fonction du degré de confiance que le réseau avait en sa prédiction). Elle a le désavantage d'être peu intuitive et ne tient pas compte des débalancements de classe.

IoU (Intersection Over Union) - Cette métrique (aussi sur un gradient entre 0 et 1) permet d'évaluer la performance d'un réseau de segmentation dans un cas de déséquilibre des classes comme le nôtre. Elle tient compte de la proportion des classes en pénalisant plus ou moins fortement le score du réseau selon la proportion de chaque classe (Tiu, 2022) (une classe majoritaire est plus facile à prédire qu'une classe minoritaire). Elle est moins intuitive mais a l'avantage d'évaluer la réelle performance de segmentation d'un réseau et non sa capacité à apprendre la (les) classes majoritaire(s) seulement. Pour cette raison elle sera la métrique principale pour évaluer la performance du réseau.

PERFORMANCE SUITE À L'ENTRAÎNEMENT SUR LES IMAGES SYNTHÉTIQUES

Tout d'abord, nous entraînons le RN sur les images de synthèse et conservons l'époque qui a donné les meilleurs résultats. Considérant qu'il n'y a pas de plants

---

2. La fonction Keras ayant été utilisée : `https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy`

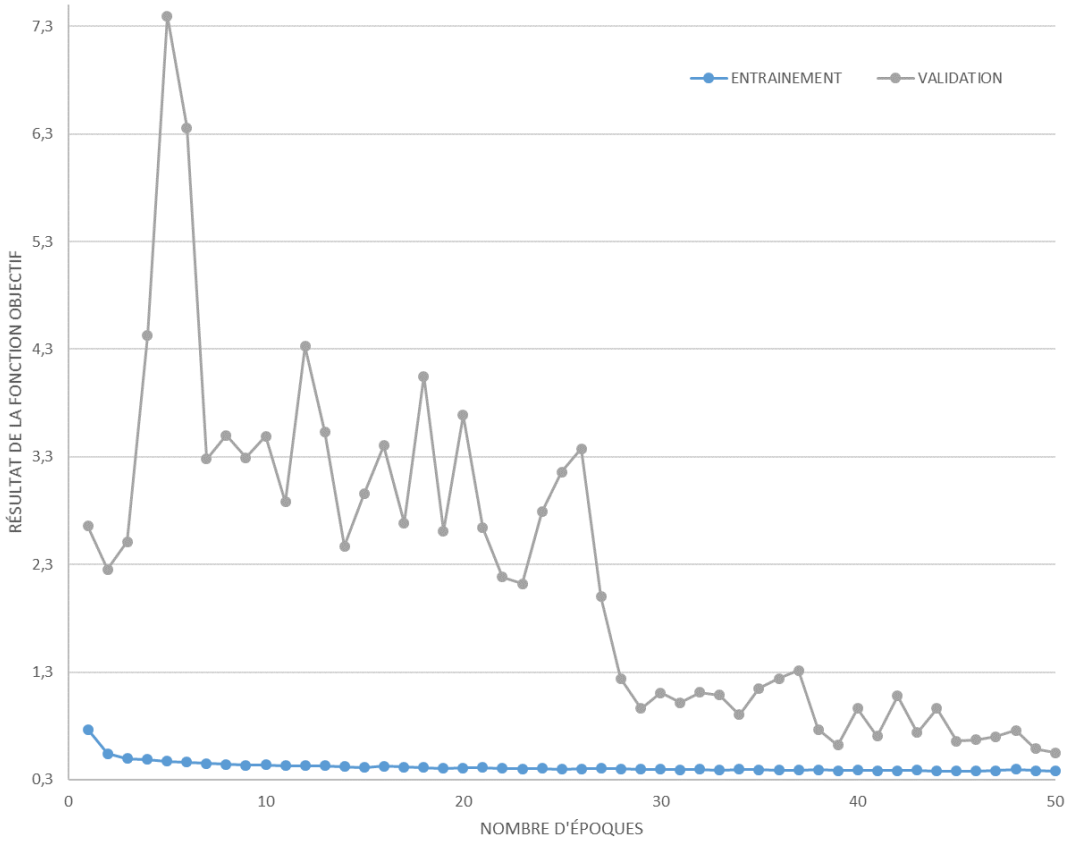

Figure 5.5 Résultat de l'entraînement et de la validation sur le JD virtuel seulement

de bleuets dans les images de synthèses et que nous cherchons principalement à confirmer que nous avons un RN capable de faire de la segmentation dans nos images, cette étape ne sera pas validée contre des données réelles mais contre un échantillon d'images synthétiques.

Notre RN a été en mesure d'apprendre sur le JD et de produire des prédictions d'une qualité suffisante comme on peut le constater à la figure 5.5. Il est possible qu'un entraînement plus long aurait donné de meilleurs résultats, mais considérant que cet apprentissage va servir de base à un autre JD et considérant le risque

de surapprentissage, le résultat de la dernière époque sera retenu pour démarrer l'entraînement sur les images réelles dans l'approche combinée.

PERFORMANCE SUITE À L'ENTRAÎNEMENT UTILISANT DE PETITS JEUX DE DONNÉES RÉELLES

Afin de valider si l'apprentissage sur les images de synthèse apporte une différence significative sur les résultats, nous réalisons un entraînement directement sur les images réelles sans passer par les images de synthèse. Pour justifier la création et l'utilisation des images de synthèse, il est attendu que la performance avec l'usage du petit JD d'images réelles donne un résultat inférieur à l'approche combinée.

PERFORMANCE COMBINÉE DES DEUX APPROCHES

Finalement, nous utilisons l'approche qui a eu du succès durant les tests, à savoir d'entraîner d'abord le RN sur les images de synthèse, puis de procéder à un nombre restreint d'époques sur les données réelles afin d'éviter le surapprentissage.

On peut constater deux différences avec l'approche combinée. Premièrement, le résultat de classification est meilleur. La table 5.4 présente les 3 meilleurs résultats de validation obtenus dans les deux scénarios. La différence est significative et le RN entraîné uniquement sur le petit JD d'images réelles ne semble pas s'améliorer en augmentant le nombre d'époque, suggérant l'atteinte de la limite de cette forme d'entraînement. Le RN ayant passé par l'entraînement sur les images virtuelles a donc été en mesure de développer des mécanismes de détection utiles et transférables aux images réelles.

Deuxièmement, il a été plus rapide d'arriver à ce résultat à partir du RN pré-entraîné sur les données virtuelles qu'à partir de MobileNet. L'entraînement combiné a mené à une perte (loss) inférieure à 2 après environ 25 époques tandis qu'il

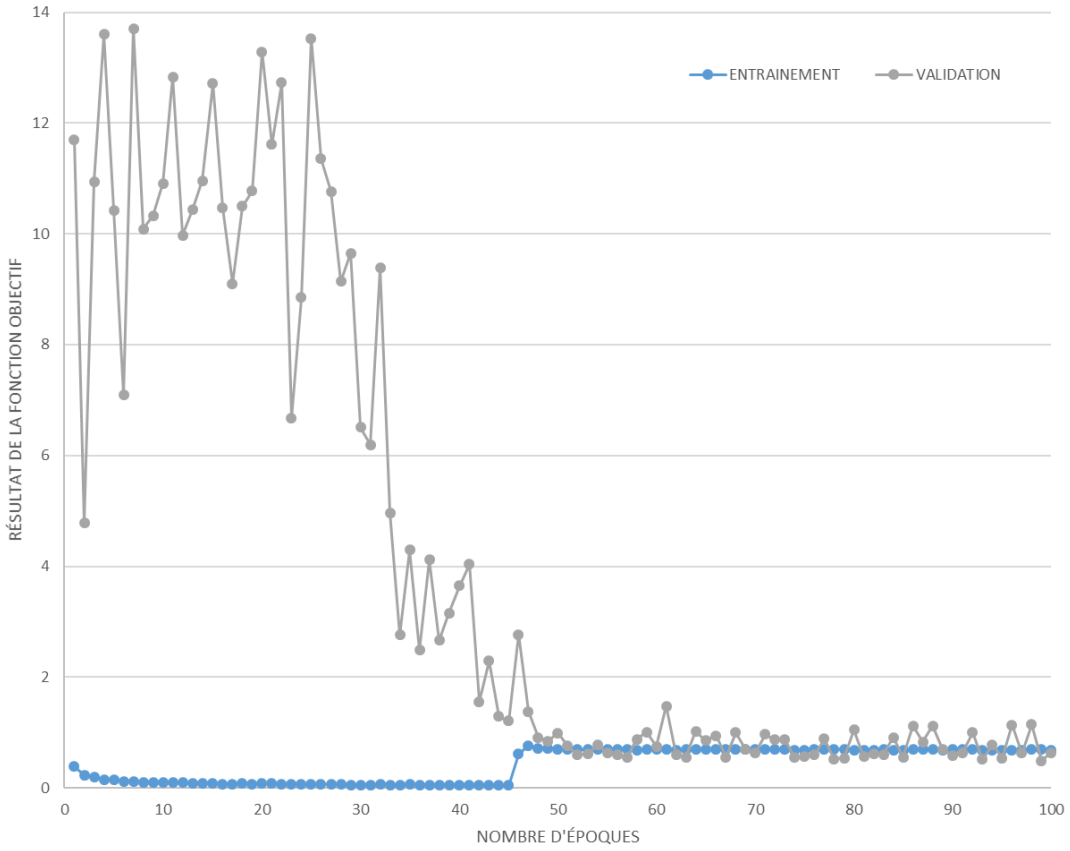

Figure 5.6 Résultat de l'entraînement et de la validation sur le JD réel seulement

JD	3 meilleurs résultats			Moyenne
Données réelles seulement	0.489	0.517	0.521	0.509
Données virtuelles et réelles	0.261	0.298	0.304	0.288

Tableau 5.4 Comparatif des résultats avec et sans l'entraînement sur les données virtuelles (taux de perte, loss)

en a fallu environ 45 pour obtenir la même performance sur les images réelles uniquement. Bien que cela n'a ultimement pas d'impact sur la performance du RN, cette amélioration de la vitesse d'entraînement permet d'affirmer que l'entraînement virtuel a été bénéfique pour l'utilisation sur les images réelles. Dans notre cas la différence sur le nombre d'époques n'avait comme impact que quelques heures, mais dans des situations plus complexe les durées d'entraînement sont beaucoup plus longues et une telle accélération aurait des impacts significatifs.

RÉSULTATS EN TERME DE SUCCÈS DE CLASSIFICATION

L'utilisation du "loss" plutôt que d'une valeur binaire pour l'entraînement et l'évaluation des résultats a des avantages au niveau de l'efficacité de l'entraînement. Le "loss" tient compte du degré de réussite ou d'échec de prédiction. Par exemple, une prédiction de 0.1 alors que la vraie valeur était 1 sera pénalisée plus fortement qu'une prédiction de 0.4 dans le même cas. Si on avait utilisé une réponse binaire avec un seuil, les 2 cas auraient subi le même traitement alors que l'un s'était davantage trompé que l'autre et nécessite une variation plus grande des valeurs du RN.

Cependant, ces valeurs ne représentent pas bien pour l'imaginaire humain si une prédiction est valide ou non car ultimement, nous cherchons à obtenir une réponse binaire. Pour cette raison, nous allons discuter les exemples de résultats en termes

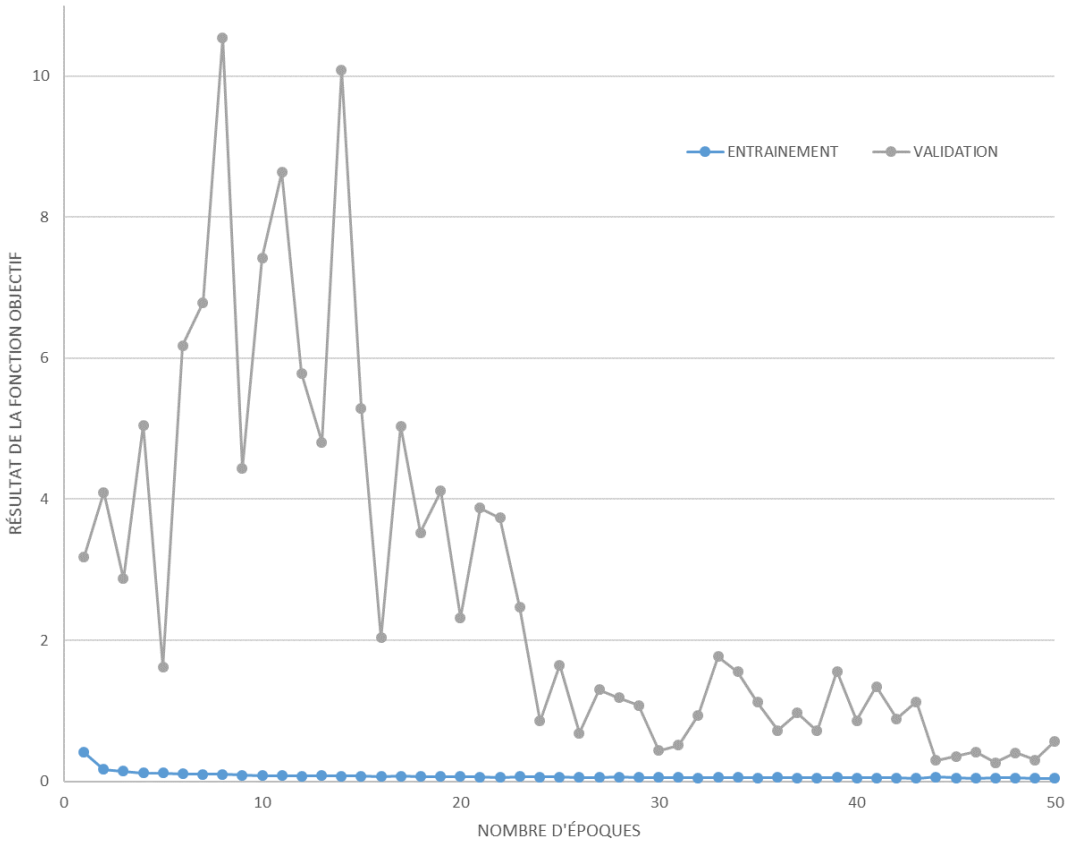

Figure 5.7 Résultat de l'entraînement et de la validation sur le JD réel suite à une série d'entraînements sur les données virtuelles

de succès/échec de prédiction. Ces résultats sont basés sur 100 prédictions aléatoires utilisant des données réelles. Les 3 meilleurs scores de chaque entraînement sont représentés de la même façon que dans la table précédente.

JD	3 meilleurs résultats			Moyenne	Taux d'erreur
Données réelles seulement	68.7%	65.6%	67.6%	67.3%	32.7%
Données virtuelles et réelles	86.8%	87.4%	89.0%	87.7%	12.3%

Tableau 5.5 Comparatif des résultats avec et sans l'entraînement sur les données virtuelles (taux de succès de prédiction par pixel)

On peut constater une différence significative dans la performance de classification des deux approches. Le résultat des données réelles seulement est près de la fraction des données représentant la classe majoritaire, laissant supposer que le RN ne fait que prédire la classe majoritaire (ce qui est effectivement le cas et sera analysé plus bas). Pour ce qui est du RN entraîné sur les deux types de données, son score laisse plutôt prévoir qu'une réelle capacité de segmentation a été atteinte. Nous ne sommes pas pour autant au stade auquel un grand JD réel serait inutile. En effet, il est courant de voir des performances de segmentation de 95%+ dans la recherche lorsque suffisamment de données d'entraînement sont disponibles.

EXEMPLES DE RÉSULTATS DE CLASSIFICATION

Dans cette section, nous analyserons des cas spécifiques de classification afin de mieux comprendre pourquoi nous avons obtenu ces résultats.

EXEMPLES DE RÉSULTATS DE CLASSIFICATION SUR LE PETIT JEU DE DONNÉES D'IMAGES RÉELLES

Lorsqu'on regarde les prédictions de cet entraînement à la figure 5.8, on constate que le RN a rapidement basculé vers une stratégie de prédiction de la classe prédominante pour toutes les images, ce qui est la meilleure stratégie si un RN est incapable de développer une approche plus discriminante permettant une réelle segmentation. Du moment où cette stratégie a été adopté par le RN, elle est demeurée. À la lumière de cette analyse on peut conclure que l'entraînement utilisant seulement un petit JD d'images réelles est un échec, la stratégie n'ayant aucune utilité et n'ayant pas besoin d'un RN neuronal pour parvenir au même résultat. Pour cette raison nous n'avons pas inclus la matrice de confusion de ces résultats : l'entièreté des prédiction concernant la même classe, il n'y a pas d'intérêt à analyser plus en détail.

Il demeure possible qu'en variant le RN et les paramètres d'entraînement une autre stratégie de segmentation aurait pu émerger. Il est cependant peu probable qu'elle ait été performante si on avait voulu la comparer à d'autres méthodes de segmentation d'images (par exemple, utiliser la couleur du pixel pour prédire sa classe). La complexité des RN suppose qu'on doit au minimum égaler la performance des méthodes classiques pour justifier leur utilisation.

EXEMPLES DE RÉSULTATS DE CLASSIFICATION SUR L'USAGE COMBINÉ RÉEL ET SYNTHÉTIQUE

On constate en analysant les exemples de la figure 5.9 que l'utilisation combinée des deux types de données a produit un RN capable de faire une segmentation réelle des images et de produire un résultat pouvant être utile.

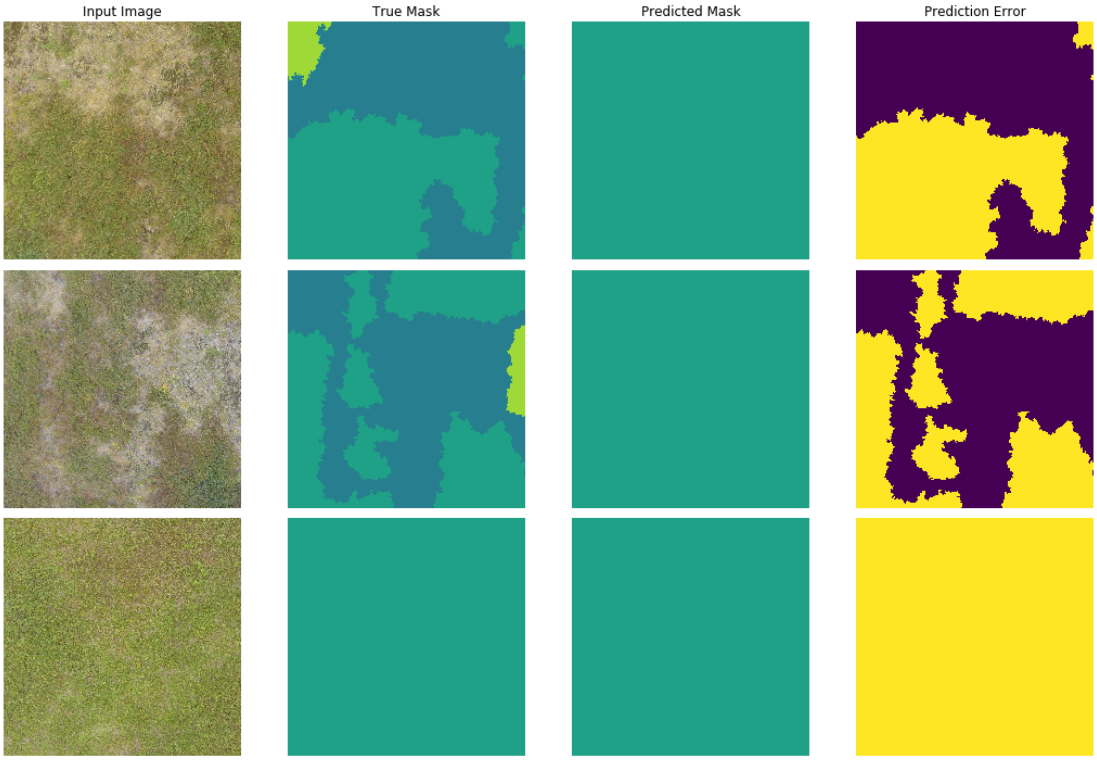

Figure 5.8 Exemples des prédictions utilisant un petit JD réel

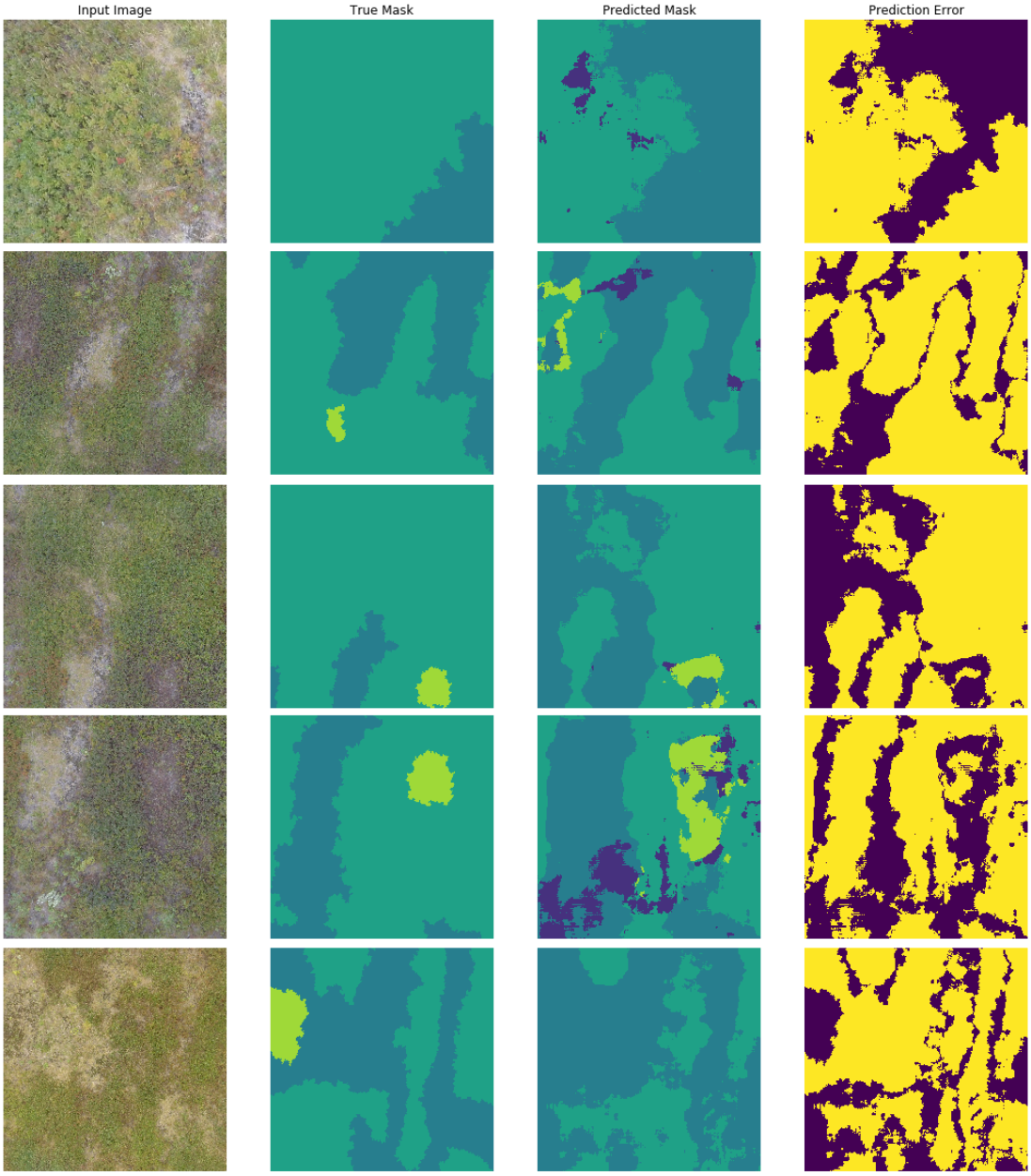

Figure 5.9 Exemples des prédictions utilisant les données combinées

	Prédiction					
	Arbre	**Débris**	**Autre plante**	**Bleuet**	**Eau**	**Sol**
Arbre	**89%**	0%	3%	2%	0%	6%
Débris	68%	**0%**	11%	11%	0%	11%
Autre Plante	1%	0%	**88%**	10%	0%	0%
Bleuet	2%	0%	9%	**89%**	0%	1%
Eau	8%	0%	55%	11%	**0%**	26%
Sol	4%	0%	50%	4%	0%	**42%**

Tableau 5.6 Matrice de confusion des résultats obtenus à l'usage des deux JD d'entraînement (pourcentages arrondis). Les classes absentes des données de test et jamais prédites (roche et ciel) ont été retirées du tableau.

Même avec ce RN on peut constater un écart entre ce qui a été marqué à la main et la prédiction du RN. Il est intéressant de constater qu'il y a beaucoup de différences entre les deux lors d'un changement de région, et qu'il n'est pas forcément vrai que ce qui a été marqué à la main dans ces images est plus véridiques que la prédiction du RN lorsqu'il s'agit de déterminer la classe d'un pixel dans ces zones transitoires. Une amélioration intéressante à apporter dans de futures recherches serait un meilleur support de ces zones afin de voir comment on pourrait supporter une notion de transition plutôt qu'une coupure directe d'une classe à l'autre. Dans plusieurs cas le RN tire la ligne à des endroits qui seraient acceptables dans un contexte opérationnel même si le RN diverge de la division manuelle sur les images d'entraînements.

La matrice de confusion des résultats obtenus (table 5.6) permet d'avoir un aperçu global des performances. À première vue on constate ce qui semble être une bonne performance sur la plupart des classes majoritaires. Cependant, une telle classification des résultats est problématique lors de l'utilisation de jeux de données qui

Résultats IoU (Multiclasse)	
Arbre	0,08
Débris	0,00
Autre plante	0,68
Bleuet	0,86
Eau	0,00
Sol	0,32
Moyenne	**0,32**

Tableau 5.7 Résultat IoU de l'approche multiclasse des résultats de segmentation. Les classes absentes des données de test et jamais prédites (roche et ciel) ont été retirées du tableau.

présentent un déséquilibre des classes comme le nôtre. Il est facile de prédire principalement les classes majoritaires et d'obtenir un excellent score dans une telle matrice sans avoir une réelle capacité de segmentation des classes. L'approche IoU (Intersection Over Union) est méthode de représentation des résultats qui tient compte aussi des erreurs de prédiction et donc cerne la capacité réelle du réseau à fournir un résultat utile. On constate des résultats (table 5.7) bien différents de l'approche par matrice de confusion lorsqu'on utilise cette approche.

À la lumière de ce résultat, il semble que notre classificateur n'est pas performant d'un point de vue multiclasses. Nos données d'entraînement ont un fort déséquilibre de classes et ont mené à un réseau ayant principalement appris à identifier la classe principale. Si on le considère comme un classificateur binaire en groupant toute les classes différentes de bleuets 5.8), on obtient une performance nettement supérieure à l'approche multiclasse. Il semble donc que nous avons réussi à entraîner un bon classificateur binaire sur notre classe principale, mais que nous ne

Résultats IoU (Binaire)	
Non-Bleuet	0,62
Bleuet	0,86
Moyenne	**0,74**

Tableau 5.8 Résultat IoU de l'approche binaire des résultats de segmentation.

sommes pas parvenus à mettre en place une méthode d'entraînement capable de segmenter les différents éléments pouvant se trouver dans nos images.

Devrait-on se préoccuper de ces classes minoritaires ? La réponse dépend de l'utilisation que l'on voudrait faire du réseau. Par exemple dans un cas d'application d'herbicide, il pourrait être suffisant d'aller vers un classificateur binaire capable de séparer les bleuets du reste du terrain et de laisser tomber ces classes minoritaires. En contrepartie, il y aurait le coût économique et environnement d'appliquer de l'herbicide sur autre chose que les espèces indésirables (par exemple, le sol nu). Mais en autant que l'on ne touche pas aux plantes de culture, ce compromis pourrait être acceptable.

Dans le cas où l'on voudrait faire de la cartographie du territoire ou appliquer un herbicide de façon restreinte, il est important de garder la capacité à discerner ces classes minoritaires. Pour de telles utilisations, il faut donc une stratégie de réduction de l'effet d'imbalance du JD. Les JD réels sont, comme la réalité, rarement balancés et il faut souvent gérer ces impacts. Il s'agit d'un problème affectant l'apprentissage machine en général et non quelque chose de spécifique aux réseaux neuronaux. Deux groupes de solutions existent à cette fin.

Le premier est d'ajuster les poids de la rétroaction en fonction de la classe, et donc de pénaliser plus fortement le réseau lorsqu'il commet une erreur sur une classe mineure. Par exemple, dans un JD fictif de deux classes, la majeure représentant

90% des données et l'autre le 10% restant pour un ratio 9 pour 1, les poids d'entraînement seraient de 1 pour 9 (l'inverse du ratio des classes). Les 2 classes auraient ainsi un impact théorique équivalent.

Un approche plus raffinées de l'ajustement des poids est possible en se basant sur les erreurs de prédictions plutôt que sur les classes. Le « Focal Loss » (Lin *et al.*, 2020) pénalise plus fortement le réseau lorsqu'il se trompe sur les données qu'il arrive difficilement à classer. En évitant que les données qu'il est déjà capable de classer facilement aient beaucoup d'impact, les classes sous-représentées auront une priorité automatique. Cette priorité sera également mieux distribuée, car un exemple difficile à classer de la classe majoritaire (ou simple à classer de la classe minoritaire) aura un effet plus cohérent sur l'entraînement que l'approche élémentaire d'y aller strictement selon la classe.

Le deuxième est la sur/sous-représentation de classe dans la préparation des données d'entraînement. Dans le cas de notre JD fictif réparti 9 pour 1 entre deux classes, le générateur de données d'entraînement pourrait piger 9 fois plus souvent dans les données de la classe mineure. Réduire la quantité de données des classes majoritaires pour revenir à un ratio 1 pour 1 est également possible comme stratégie.

Ces méthodes ne sont cependant pas parfaites. Elles vont forcer le réseau à considérer fortement une petite quantité de données pour son entraînement, possiblement au détriment de sa précision globale. Le degré de débalancement, la complexité de l'information, la taille totale du JD et le classificateur lui-même sont tous des facteurs ayant un impact sur les résultats (Japkowicz et Stephen, 2002). Cette optimisation est un exercice de compromis et selon le but recherché, elle peut prendre beaucoup de temps. Pour cette raison, il n'a pas été considéré d'utiliser

ces méthodes pour nos travaux mais il s'agit d'une excellente piste pour de futures améliorations.

5.6 APPRENTISSAGE

L'utilisation de données simulées en complément aux données réelles a permis d'entraîner un RN capable de segmentation, ce qui ne s'est pas produit lorsque nous avons utilisé uniquement les images réelles. Il est probable qu'un entraînement sur un grand JD d'images réelles donne une performance supérieure à ce que nous avons obtenu ici.

Dans des scénarios d'usage comme l'identification de mauvaises herbes dans une culture particulière, ces grands JD sont pratiquement inexistants et le resteront, considérant l'intérêt minime pour une organisation de mettre en place ces JD et de les rendre publics. Les données virtuelles et le transferts de RN entraînés sur des données génériques vers les cas spécifiques demeurera probablement la voie pour plusieurs années. Il faudrait développer des méthodes d'apprentissage machines capables d'extrapoler elles-mêmes le général à partir du spécifique, ce qui n'est pas encore à notre portée.

Notre RN a développé une expertise spécifique au petit JD dans notre cas d'expérimentation, et cette performance n'est probablement pas généralisable. Il serait intéressant de confirmer les résultats avec plusieurs petits JD d'images réelles afin de voir comment il serait possible d'arriver à un RN généraliste sans pour autant devoir créer manuellement des JD massifs.

CONCLUSION

L'objectif de nos travaux était de construire et d'entraîner un classificateur multi-classe de segmentation d'image utilisable pour la culture du bleuet sauvage. Pour atteindre cet objectif nous avons :

- utilisé des technologies du jeu vidéo et créé un environnement virtuel,
- complété ces images de synthèse avec des images réelles acquises par drone,
- utilisé un réseau inspiré de U-Net comme classificateur,
- entraîné ce réseau à l'aide de notre banque d'image et
- effectué des tests de performance des résultats obtenus.

Il fut relativement simple de créer des environnements de simulation. L'industrie du jeu vidéo a pavé la voie autant au niveau des technologies que des banques d'éléments visuels nécessaires à leur mise en place. La course au réalisme est un moteur de l'évolution technologique du jeu vidéo et peut s'utiliser à d'autres fins. Les données réelles et l'étiquetage manuel sont une alternative coûteuse à cette capacité de synthèse. Environ une semaine d'effort a permis de générer plusieurs dizaines de milliers d'images virtuelles. L'acquisition et la classification d'un JD réel d'une taille similaire aurait requis un volume de travail se comptant plutôt en mois.

La photographie par drone fut un élément crucial de nos travaux. Les alternatives aux drones (par exemple, les hélicoptères) sont trop dispendieuses et auraient rendu impossible notre approche. Leur usage comporte toutefois des difficultés,

et des quelques sorties qui ont été effectuée, une seule a engendré des images utilisables.

L'usage de code existant pour le réseau a grandement simplifié nos travaux en évitant de mettre les efforts sur l'écriture de code et de la concentrer plutôt sur la génération des données et l'optimisation de l'entraînement. Considérant la simplicité du réseau utilisé, l'entraînement fut possible dans un court laps de temps sur de l'équipement facilement disponible. Bien que nous ayons sûrement sacrifié en qualité des résultats en cherchant un entraînement rapide, ce compromis a permis une expérimentation plus facile dans le contexte de ces travaux.

Nos résultats analysés selon l'approche IoU n'ont pas été concluants d'un point de vue multiclasse avec un score IoU de seulement **0,32**. Nos données d'entraînement n'étaient pas de la même qualité pour toutes les classes et n'ont pas permis un entraînement de qualité pour chacune d'entre elles. Cependant, en se limitant à une approche binaire (bleuet ou autre), nous avons obtenu des résultats concluants avec un score IoU de **0,74**. Un score supérieur à 0,5 est habituellement considéré comme étant une bonne prédiction.

Il s'avère difficile d'entraîner un classificateur multiclasse, chaque classe augmentant rapidement le besoin en données. Nos images d'entraînement manquaient en quantité et en qualité pour plusieurs classes et notre réseau n'a pas toujours obtenu des résultats utilisables. C'est lorsqu'on le considère comme un classificateur binaire qu'on obtient un résultat utilisable. Nous avons donc seulement atteint partiellement notre objectif. Plusieurs pistes sont possibles pour améliorer la qualité de ces résultats.

LIMITES DE NOS TRAVAUX ET AMÉLIORATIONS POSSIBLES

Les méthodes utilisées dans ces travaux pourraient être grandement améliorées. Notre approche se voulait simple pour le contexte (2017) et depuis de nombreuses améliorations ont été développées. Des techniques comme les réseaux génératifs antagonistes (GAN) et le zero-shot learning auraient de grandes chances de donner de meilleurs résultats que ceux que nous avons obtenus. Ce travail demeure un bref aperçu de toutes les possibilités qui existent pour attaquer le problème de la segmentation d'images.

Nous avions un seul environnement virtuel et n'avons pas fait appel aux services d'un designer 3D. Une simulation plus précise des réalités agricoles serait un meilleur terrain d'entraînement que les éléments de forêt que nous avions. Créer des plants de bleuets sauvages en simulation aurait sûrement amélioré la performance de l'entraînement. Nos données manquaient également de variété au niveau des différentes classes d'éléments. Un classificateur de terrain agricole doit être en mesure de distinguer les roches, l'eau, le ciel, les arbres, et bien d'autres. Il est possible de trouver de tels éléments sur le marché, simplifiant leur intégration. Les difficultés de notre classificateur à distinguer les classes entre elles vient probablement du manque de qualité des données d'entraînement.

Suite à quelques essais infructueux d'acquisition de données, une sortie en drone a produit des données utilisables pour notre projet. Bien que l'on cherche à minimiser le temps en acquisition et classification d'images réelle, plusieurs acquisitions d'images dans des lieux différents auraient donné un réseau plus robuste aux différences.

Nos essais non concluants avec l'adaptation de domaine ne signifie pas pour autant que la méthode est inutile, celle-ci ayant fait ses preuves dans d'autres circons-

tances. Il est possible qu'en utilisant différents paramètres ou en la greffant à un autre réseau, nous aurions de meilleurs résultats et un meilleur transfert des apprentissages en environnements virtuels vers le réel.

Nous avons utilisé des architectures réseaux plutôt élémentaires. Bien qu'il ne soit pas certain qu'un réseau plus complexe ait une meilleure performance dans nos cas d'utilisations, plusieurs de ces architectures sont disponibles publiquement et pourraient être testées.

Le matériel utilisé pour les entraînements (Nvidia GTX 1070) est maintenant plusieurs générations derrière le marché. Les mêmes entraînements pourraient se dérouler plus rapidement, ou être plus complets dans la même période de temps. L'offre infonuagique en apprentissage profond continue de s'améliorer, rendant accessible une capacité de calcul ponctuelle qui ne serait pas justifiable d'acheter pour un usage personnel.

AUTOMATISATION ET CULTURE DU BLEUET SAUVAGE

Le bleuet sauvage est une activité exclusive à quelques territoires dans le nord de la côte Atlantique et se fait selon des méthodes différentes du bleuet cultivé. Bien qu'il s'agit d'un secteur économique important pour ces régions, la taille et l'unicité de cette culture fait en sorte que les grandes entreprises d'équipements agricoles ne la considèrent pas comme une priorité. À ce jour, les équipements utilisés sont soit des équipement à usage général, soit des équipements produits par des entreprises locales à petite échelle. Aucun grand groupe industriel ne produit d'équipement pour cette culture.

En permettant un accès à l'automatisation à l'industrie du bleuet sauvage, on aide à la garder vivante à long terme. Les dernières années ont été difficiles pour les producteurs du secteur, des chutes de prix ayant créé de lourde pertes financières.

Le coût de production élevés du bleuet sauvage par rapport au bleuet cultivé est un grand responsable de ces difficultés. Dans les dernières années, le prix payé pour la livre de bleuet sauvage (0.20$ CAN) est passé sous le seuil de rentabilité de bien des producteurs (0.50$ CAN) [3]. Si cette industrie n'arrive pas à diminuer ses coûts de production et que les prix demeurent aussi bas, il est probable que l'industrie se contracte. La moitié des producteurs sont à risque de faillite dans un tel contexte économique [4]. La baisse des coûts de production, des besoins de main-d'oeuvre, de la quantité de pesticides et de l'empreinte écologique passe en grande partie par l'automatisation des processus.

Nos travaux démontrent qu'il est possible, avec peu de moyens, d'appliquer l'intelligence artificielle à cette industrie. Ces technologies ne sont pas réservées aux grands industriels et de petits fournisseurs pourraient mettre en marché des solutions novatrices propre à ce domaine.

---

3. www.cbc.ca/news/canada/nova-scotia/wild-blueberry-prices-reach-all-time-low-says-association-1.4269412

4. www.acadienouvelle.com/actualites/2017/08/20/bleuets-sauvages-producteurs-n-b-ne-a-labri-de-faillite/

RÉFÉRENCES

Arellano, C. et Tapia Farias, J. (2019). Deepblueberry : Quantification of blueberries in the wild using instance segmentation. *IEEE Access*, *PP*, 1–1. http://dx.doi.org/10.1109/ACCESS.2019.2933062

Badrinarayanan, V., Kendall, A. et Cipolla, R. (2017). SegNet : A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495. http://dx.doi.org/10.1109/TPAMI.2016.2644615

Breheret, A. (2017). Pixel Annotation Tool. Récupéré le 2020-01-25 de https://github.com/abreheret/PixelAnnotationTool

Chen, B.-C. et Kae, A. (2019). Toward realistic image compositing with adversarial learning. Dans *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8407–8416. http://dx.doi.org/10.1109/CVPR.2019.00861

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K. et Yuille, A. L. (2018). DeepLab : Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848. http://dx.doi.org/10.1109/TPAMI.2017.2699184

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. et Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding.

Di Cicco, M., Potena, C., Grisetti, G. et Pretto, A. (2017). Automatic model based dataset generation for fast and accurate crop and weeds detection. *IEEE International Conference on Intelligent Robots and Systems*, *2017-September*, 5188–5195. http://dx.doi.org/10.1109/IROS.2017.8206408

Fortuner, B., Viana, M., murphy66 et Kowshik, B. (2022). Loss Functions - Cross-Entropy. Récupéré le 2022-02-12 de https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.

html#:~:text=Cross%2Dentropy%20loss%2C%20or%20log,diverges%20from%20the%20actual%20label.

Ganin, Y. et Lempitsky, V. (2015). Unsupervised Domain Adaptation by Backpropagation. *Proceedings of the 32nd International Conference on Machine Learning*, *37*(Proceedings of Machine Learning Research), 1180—-1189. http://dx.doi.org/10.1109/CVPR.2012.6247911

Garford Farm Machinery Ltd (2021). Weed-it quadro. Récupéré le 2021-02-06 de https://garford.com/fr/bineuse-robocrop-inrow/

Geiger, A., Lenz, P., Stiller, C. et Urtasun, R. (2013). Vision meets robotics : The kitti dataset. *International Journal of Robotics Research (IJRR)*.

Girshick, R., Donahue, J., Darrell, T. et Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Dans *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 580–587. http://dx.doi.org/10.1109/CVPR.2014.81

Giselsson, T. M., Jørgensen, R. N., Jensen, P. K., Dyrmann, M. et Midtiby, H. S. (2017). A Public Image Database for Benchmark of Plant Seedling Classification Algorithms. *arXiv preprint*.

He, K., Gkioxari, G., Dollar, P. et Girshick, R. (2017). Mask R-CNN. Dans *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-October, 2980–2988. http://dx.doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S. et Sun, J. (2016). Deep residual learning for image recognition. Dans *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, 770–778. http://dx.doi.org/10.1109/CVPR.2016.90

Hennessy, P., Esau, T., Zaman, Q., Corscadden, K., Schumann, A. et Farooque, A. (2020). Viability of using convolutional neural networks for real-time fescue and sheep sorrel detection in wild blueberry fields. http://dx.doi.org/10.32393/csme.2020.1184

Huang, G., Liu, Z., Van Der Maaten, L. et Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*, 2261–2269. http://dx.doi.org/10.1109/CVPR.2017.243

Japkowicz, N. et Stephen, S. (2002). The class imbalance problem : A

systematic study. *Intell. Data Anal.*, *6*, 429–449. http://dx.doi.org/10.3233/IDA-2002-6504

Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S. N., Rosaen, K. et Vasudevan, R. (2017). Driving in the matrix : Can virtual worlds replace human-generated annotations for real world tasks ?

Kamikawa, Y. (2017). SegNet including indices pooling for Semantic Segmentation with tensorflow and keras. Récupéré le 2019-03-03 de https://github.com/ykamikawa/SegNet

Kraemer, F., Schaefer, A., Eitel, A., Vertens, J. et Burgard, W. (2017). From Plants to Landmarks : Time-invariant Plant Localization that uses Deep Pose Regression in Agricultural Fields. Récupéré le 2020-02-01 de https://arxiv.org/abs/1709.04751

Krizhevsky, A., Sutskever, I. et Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 1–9. http://dx.doi.org/http://dx.doi.org/10.1016/j.protcy.2014.09.007

LeCun, Y., Bottou, L., Bengio, Y. et Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2323. http://dx.doi.org/10.1109/5.726791

Leguilly, Y. (2019). Semantic Segmentation with tf.data in TensorFlow 2.0 and ADE20K dataset. Récupéré le 2020-02-12 de https://yann-leguilly.gitlab.io/post/2019-12-14-tensorflow-tfdata-segmentation/

Lin, T., Goyal, P., Girshick, R., He, K. et Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 318–327. http://dx.doi.org/10.1109/TPAMI.2018.2858826

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y. et Berg, A. C. (2016). SSD : Single shot multibox detector. Dans *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9905 LNCS, 21–37. http://dx.doi.org/10.1007/978-3-319-46448-0_2

Milioto, A., Lottes, P. et Stachniss, C. (2018). Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs. Dans *Proceedings - IEEE International Conference on Robotics and Automation*, 2229–2235. http://dx.doi.org/10.1109/ICRA.2018.8460962

Neuhold, G., Ollmann, T., Bulo, S. R. et Kontschieder, P. (2017). The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. Dans *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-October, 5000–5009. http://dx.doi.org/10.1109/ICCV.2017.534

Olsen, A., Konovalov, D. A., Philippa, B., Ridd, P., Wood, J. C., Johns, J., Banks, W., Girgenti, B., Kenny, O., Whinney, J., Calvert, B., Azghadi, M. R. et White, R. D. (2019). DeepWeeds : A Multiclass Weed Species Image Dataset for Deep Learning. *Scientific Reports*, *9*(1). http://dx.doi.org/10.1038/s41598-018-38343-3

Pérez, A. J., López, F., Benlloch, J. V. et Christensen, S. (2000). Colour and shape analysis techniques for weed detection in cereal fields. *Computers and Electronics in Agriculture*, *25*(3), 197–212. http://dx.doi.org/10.1016/S0168-1699(99)00068-X

Redmon, J., Divvala, S., Girshick, R. et Farhadi, A. (2016). You only look once : Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 779–788. http://dx.doi.org/10.1109/CVPR.2016.91

Richter, S. R., Vineet, V., Roth, S. et Koltun, V. (2016). Playing for data : Ground truth from computer games. Dans *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9906 LNCS, 102–118. http://dx.doi.org/10.1007/978-3-319-46475-6_7

Robbins, H. et Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, *22*(3), 400–407. http://dx.doi.org/10.1214/aoms/1177729586. Récupéré le 2020-02-01 de http://projecteuclid.org/euclid.aoms/1177729586{\%}5Cnpapers2://publication/uuid/0E522956-F1DE-4A56-885A-E780F05A8297

Ronneberger, O., Fischer, P. et Brox, T. (2015). U-net : Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9351*, 234–241. http://dx.doi.org/10.1007/978-3-319-24574-4_28

Ros, G., Sellart, L., Materzynska, J., Vazquez, D. et Lopez, A. M. (2016). The SYNTHIA Dataset : A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. Dans *Proceedings of the IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, volume 2016-December, 3234–3243. http://dx.doi.org/10.1109/CVPR.2016.352

Sa, I., Chen, Z., Popovic, M., Khanna, R., Liebisch, F., Nieto, J. et Siegwart, R. (2018). http://dx.doi.org/10.1109/LRA.2017.2774979

Shropshire, G. J. (1989). Weed detection in row crops using the red near -infrared reflectance ratio and frequency transforms of video images. *University of Nebraska - Lincoln*. Récupéré le 2020-02-01 de https://digitalcommons.unl.edu/dissertations/AAI8925260/

Simonyan, K. et Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICRL)*, 1–14. http://dx.doi.org/10.1016/j.infsof.2008.09.005

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. et Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 2818–2826. http://dx.doi.org/10.1109/CVPR.2016.308

Thambawita, V., Salehi, P., Sheshkal, S. A., Hicks, S. A., Hammer, H. L., Parasa, S., de Lange, T., Halvorsen, P. et Riegler, M. A. (2021). Singan-seg : Synthetic training data generation for medical image segmentation.

Tik Chiu, Mang ; Xu, Xingqian ; Wei, Yunchao ; Huang, Zilong ; Schwing, Alexander G. ; Brunner, Robert ; Khachatrian, Hrant ; Karapetyan, Hovnatan ; Dozier, Ivan ; Rose, Greg ; Wilson, David ; Tudor, Adrian P. ; Hovakimyan, N. et S., T. (2020). Agriculture-Vision : A Large Aerial Image Database for Agricultural Pattern Analysis. *ArXiv*, *abs/2001.0*. Récupéré le 2020-02-01 de https://arxiv.org/abs/2001.01306

Tiu, E. (2022). Metrics to Evaluate your Semantic Segmentation Model. Récupéré le 2022-02-12 de https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

Viola, P. et Jones, M. J. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, *57*(2), 137–154. http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb

WEED-IT Precision Spraying (2021). Weed-it quadro. Récupéré le 2021-02-06 de https://www.weed-it.com/weedit-quadro

White, B. W. et Rosenblatt, F. (1963). Principles of Neurodynamics :

Perceptrons and the Theory of Brain Mechanisms. *The American Journal of Psychology*, *76*(4), 705. http://dx.doi.org/10.2307/1419730

Wikipédia (2020). Ligne de partage des eaux (segmentation) — Wikipédia, l'encyclopédie libre. Récupéré le 2020-02-12 de https://fr.wikipedia.org/wiki/Ligne_de_partage_des_eaux_(segmentation)

Wikipédia (2022). Précision et rappel — Wikipédia, l'encyclopédie libre. Récupéré le 2022-02-12 de https://fr.wikipedia.org/wiki/Pr%C3%A9cision_et_rappel

Yang, X. et Sun, M. (2019). A survey on deep learning in crop planting. *IOP Conference Series : Materials Science and Engineering*, *490*, 062053. http://dx.doi.org/10.1088/1757-899x/490/6/062053. Récupéré le 2021-02-20 de https://doi.org/10.1088/1757-899x/490/6/062053

Zarco-Tejada, P. J., Camino, C., Beck, P. S., Calderon, R., Hornero, A., Hernández-Clemente, R., Kattenborn, T., Montes-Borrego, M., Susca, L., Morelli, M., Gonzalez-Dugo, V., North, P. R., Landa, B. B., Boscia, D., Saponari, M. et Navas-Cortes, J. A. (2018). Previsual symptoms of Xylella fastidiosa infection revealed in spectral plant-trait alterations. *Nature Plants*, *4*(7), 432–439. http://dx.doi.org/10.1038/s41477-018-0189-7

Zeiler, M. D. et Fergus, R. (2014). Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8689 LNCS*(PART 1), 818–833. http://dx.doi.org/10.1007/978-3-319-10590-1_53

Zellinger, W., Moser, B. A., Grubinger, T., Lughofer, E., Natschläger, T. et Saminger-Platz, S. (2019). Robust unsupervised domain adaptation for neural networks via moment alignment. *Information Sciences*, *483*(Cmd), 174–191. http://dx.doi.org/10.1016/j.ins.2019.01.025

Zoph, B., Vasudevan, V., Shlens, J. et Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 8697–8710. http://dx.doi.org/10.1109/CVPR.2018.00907