UCLouvain

AGRO LOUVAIN

Faculté des bioingénieurs

Segmentation sémantique faiblement supervisée de canopées multi-espèces à l'aide d'apprentissage actif et de réseaux de neurones entièrement convolutifs

Auteur : Charles Rongione

Promoteurs : Prof. Xavier Draye
Prof. Guillaume Lobet

Lecteurs : Prof. Christophe De Vleeschouwer
M. Victor Joos de ter Beerst

Année académique 2021-2022

Mémoire de fin d'études présenté en vue de l'obtention du diplôme de Bioingénieur : Sciences agronomiques

Remerciements

Mes premiers remerciements vont à mes promoteurs. De façon générale pour avoir pris le temps de m'encadrer de façon irréprochable, et plus particulièrement,

Xavier Draye, pour sa créativité, son bon sens, sa confiance et la source intarissable de bonnes idées qu'il a été tout au long de ce travail.

Guillaume Lobet, pour sa disponibilité, sa bienveillance ainsi que ses nombreux conseils et relectures tout au long de l'écriture de ce travail.

Ensuite je tenais à remercier mes lecteurs. Pour le temps qu'ils passeront à lire mon travail

Christophe De Vleeschouwer pour son implication, son expertise et sa disponibilité.

Victor Joos de ter Beerst pour son expertise et son aide concernant l'utilisation des ressources informatiques de l'Uclouvain.

Enfin, je souhaiterais remercier ma famille.

Mes parents pour leur soutien indéfectible tout au long de mon parcours sinueux.

Mon oncle Marc pour ses conseils concernant la partie mathématique de ce travail.

Ma mamie pour son amour inconditionnel, tout au long de ma vie.

Table des matières

Remerciements *2*

Table des Figures *4*

Table des Tableaux *5*

Liste des acronymes *5*

1. Introduction *6*

1.1. État de l'art du phénotypage de plantes par segmentation d'images **8**

1.1.1. Méthodes Non Supervisées 8

1.1.2. Méthodes Supervisées 10

1.1.3. Conclusion de l'état de l'art 14

1.2. Objectifs **15**

2. Matériel et Méthodes *16*

2.1. Jeu de données **18**

2.1.1. Étape 0 : collecte d'images de cultures pures dont l'espèce est indiquée 19

2.2. Segmentation Binaire et Apprentissage actif **20**

2.2.1. Les arbres de décision Boostés par gradient : explication de l'algorithme 20

2.2.2. Les arbres de décision Boostés par gradient : Implémentation 23

2.2.3. Les arbres de décision Boostés par gradient : Optimisation des hyperparamètres 24

2.2.4. Étape 1 : La segmentation binaire de ces images grâce à des arbres de décision boostés sous forme d'apprentissage actif 25

2.3. Segmentation Multi-Classes **26**

2.3.1. Les Réseaux de neurones complètement convolutifs (FCNN) : explications 26

2.3.2. Description de l'architecture du réseau de neurones complètement convolutif utilisée . 32

2.3.3. Étape 3 : L'entraînement d'un réseau de neurones convolutifs sur les résultats de LeafPainter 36

2.3.1. Étape 4 : Inférence sur des images de cultures associées 41

2.4. Hardware et Software **41**

2.4.1. Software 41

2.4.2. Hardware 42

3. Resultats *42*

3.1. Segmentation Binaire LeafPainter **42**

3.2. Segmentation Multi-Classes **44**

3.2.1. Cultures pures 44

3.2.2. Cultures associées 46

4. Discussion *53*

4.1. Segmentation Binaire LeafPainter **53**

4.2. Segmentation Multi-Classes **54**

4.2.1. Cultures pures 54

4.2.2. Cultures Associées 55

5. Conclusion *57*

Bibliographie *58*

6. ANNEXES *66*

6.1. Indices de couleurs principaux **66**

Table des Figures

Figure 1 : Segmentation d'une image de fenouil.....**10**

Figure 2 : Schémas récapitulatif de la méthode globale.....**19**

Figure 3 : Localisation des quatorze parcelles où des photos ont été prises.....**21**

Figure 4 : Exemple illustratif de conversion des zones annotées en table de données structurées.....**25**

Figure 5 : Représentation schématique d'un réseau de neurones.....**29**

Figure 6 : Illustration d'une succession de deux couches de convolution.....**31**

Figure 7 : Illustration de l'architecture de FCNN utilisée lors de ce travail.....**35**

Figure 8 : Exemple d'intrusion de haricot sur une image de culture pure de choux cabus....**39**

Figure 9 : Types d'augmentation de données utilisés.....**43**

Figure 10 : Exemple de 2 images particulièrement difficiles du test de validation et leur segmentation par inférence de LeafPainter.....**45**

Figure 11 : Image du set de validation d'une culture pure de choux cabus.....**47**

Figure 12 : Culture pure de fenouil avec intrusion de chou cabu.....**47**

Figure 13 : Matrice de confusion du set de validation du FCNN.....**48**

Figure 14 : Exemple de différentes combinaisons d'associations de cultures.....**49**

Figure 15 : Comparaison d'erreur du FCNN entre cultures pures et associées pour le chou cabu.....**50**

Figure 16 : Comparaison de l'erreur du FCNN associée aux haricots ombragés, en culture pure et associée.....**51**

Figure 17 : Image d'une culture associée de haricot et de fenouil.....**52**

Figure 18 : Comparaison de 2 images de culture associées avec intrusion de haricots.....**53**

Figure 19 : Illustration de la corrélation entre erreur et incertitude.....**54**

Table des Tableaux

Tableau 1 : Hyperparamètres des GBDT les plus importants, leurs valeurs après optimisation et leur nature.....**25**

Tableau 2 : Description des caractéristiques et hyperparamètres du FCNN utilisé lors de ce travail.....**34**

Tableau 3 : Répartition des classes du jeu de données du CNN.....**37**

Tableau 4 : Score obtenus sur les différentes étapes.....**42**

Liste des acronymes

- **ExGR** : Indice d'excédent vert
- **NDI** : Indice de différence normalisée
- **NN** : Réseau de neurones entièrement connectés
- **CNN** : Réseau de neurones convolutifs
- **Resnet** : Réseau de neurones résiduels
- **FCNN** : Réseau de neurones complètement convolutifs
- **GBDT** : Arbres de décisions amplifiés par gradient
- **RGB** : Rouge, Vert, Bleu
- **HSV** : Teinte, Saturation, Valeur
- **CMYK** : Cyan, Magenta, Jaune, Noir
- **NIR** : Proche infra-rouge
- **RF** : Foret aléatoire
- **SVM** : Machine à vecteur de support
- **BR** : Brocolis
- **FE** : Fenouil
- **HN** : Haricot Nain
- **HR** : Haricot à Rame
- **CC** : Chou Cabus
- **CF** : Chou Frisé
- **OB** : Oignon Blanc
- **L** : Laitue

1. Introduction

La nécessité d'une intensification durable des systèmes agroalimentaires devient inévitable pour continuer à nourrir une population croissante, tout en minimisant les impacts environnementaux engendrés. Parmi les options à l'essai, l'association de cultures peut être utilisée pour augmenter les rendements de ces cultures grâce à une meilleure répartition des ressources ainsi qu'à l'amélioration de leur accessibilité (Guvenc & Yildirim, 2008). En plus d'atteindre une plus grande productivité (Gaudio, et al., 2019) , la diversification des systèmes de culture grâce à des associations stratégiques de cultures fournissent des avantages intéressants comparativement aux cultures pures à de nombreux égards. D'abord en termes économiques, en améliorant la stabilité des rendements (Iqbal, et al., 2019) , ensuite, en termes de services agro-systémiques, en réduisant le risque associé aux ravageurs et pathogènes (Boudreau, 2013). Leur intérêt se manifeste également en améliorant qualité des sols (Tang, et al., 2020) ou encore en permettant le développement et le maintien d'une biodiversité plus importante (Sayed Roholla & Hamdollah, 2011).

Malgré leur potentiel évident pour l'intensification durable, les paramètres écophysiologiques, tels que la complémentarité, la compétitivité ou encore les dynamiques spatiales et temporelles du développement architectural des différentes espèces associées sont nombreux. Bien qu'étant déterminants dans le succès ou non d'une association, ils ne sont que peu abordés dans la littérature scientifique. De plus, les données disponibles les concernant sont rares et parfois même controversées en ce qui concerne les cultures en maraîchage (Chevalier et al., 2021). Il est donc clair que la détermination de paramètres fiables conditionnant le succès d'une association serait une avancée déterminante pour l'établissement de systèmes de cultures en association et ainsi ouvrir une nouvelle voie vers une agriculture plus durable.

Pour ce faire, il est important de tester empiriquement certaines associations qui fonctionnent, c'est à dire qui démontrent une supériorité en termes de services agro-systémiques et/ou économiques comparativement à leurs cultures non-associées, et d'autres qui ne fonctionnent pas. En étudiant et en comparant de telles associations, il devrait être possible d'identifier et de quantifier les paramètres les plus importants, conditionnant le succès ou l'échec d'une association. Il est donc essentiel de pouvoir comparer efficacement de nombreuses répétitions pour plusieurs associations. Pour rendre cela possible, il est nécessaire de pouvoir compter sur des méthodes de phénotypage simples et rapides, comme celles basées sur l'acquisition et l'analyse d'images. Les méthodes de phénotypage par images offrent de nombreux avantages. Par exemple, elles sont non-destructives, ce qui signifie que les données phénotypiques peuvent être collectées sur les mêmes individus, tout au long de l'expérience.

L'analyse par imagerie numérique est également abordable et peut être adaptée à un large éventail de systèmes expérimentaux (Lobet et al., 2011). L'une des causes de cette flexibilité est le découplage temporel que l'analyse d'image permet entre l'expérience et la collecte des données (Lobet et al., 2013) puisque les images sont virtuellement éternelles et ne se dégradent pas avec le temps. Elle se prête également à l'automatisation, ce qui permet d'étudier des échantillons de grande taille pour une puissance statistique accrue (Mutka & Bart, 2015). Un cas particulier de l'analyse

d’images est la segmentation sémantique. Cette méthode consiste à attribuer une classe à chacun des pixels de l’image, et est le sujet principal de ce travail.

Lorsque les photos sont prises en champs, elles peuvent être très hétérogènes, même si elles capturent le même objet. Il peut y avoir des différences de luminosité, de contraste, il peut y avoir des ombres. Le contexte peut aussi être différent avec la présence d’autres objets (comme des outils, des rochers, etc.) ou encore des sols de différentes textures. Tous ces éléments font qu’il est extrêmement difficile de réaliser du phénotypage automatisé sur base d’images en extérieure. Ce problème est exacerbé dans le cas d’associations de cultures puisque ces différences peuvent se matérialiser différemment pour chacune des espèces ou variétés, ainsi qu’à toutes leurs combinaisons possibles. La solution fonctionnant théoriquement le mieux actuellement est l’apprentissage supervisé (Pound, et al., 2017). Théoriquement parce que, en pratique, les méthodes issues de ce paradigme requièrent aujourd’hui l’annotation manuelle d’images (voir Figure 1). Dans le cas particulier de la segmentation sémantique, ce travail d’annotation devient très laborieux puisqu’il doit se faire sur chacun des pixels de l’image et peut prendre jusqu’à une dizaine d’heures par image (Fuentes-Pacheco, et al., 2019).

1.1. État de l'art du phénotypage de plantes par segmentation d'images

Il existe 2 grandes catégories de méthodes pour segmenter des images. Les méthodes supervisées et les méthodes non-supervisées.

Avant de discuter ces paradigmes en détails, il convient d'introduire le concept de segmentation d'images. Segmenter une image signifie attribuer à chacun de ses pixels une classe, en fonction du type d'objet auquel elle appartient. On peut réaliser une segmentation binaire, c'est-à-dire classifier selon 2 classes chaque pixel, lorsque l'on cherche à identifier un seul type d'objet. On attribuera alors la classe 1 à cet objet et la classe 0 à tout le reste. Cet exemple est illustré sur la Figure 1.

Ce type d'analyse peut permettre de rendre cet objet quantifiable. Par exemple, dans le cas de plantes, cela peut permettre de quantifier sa surface foliaire en additionnant tous les pixels qui se voient attribuer la classe 1. On peut procéder de façon similaire lorsque l'on segmente des canopées multi espèces.

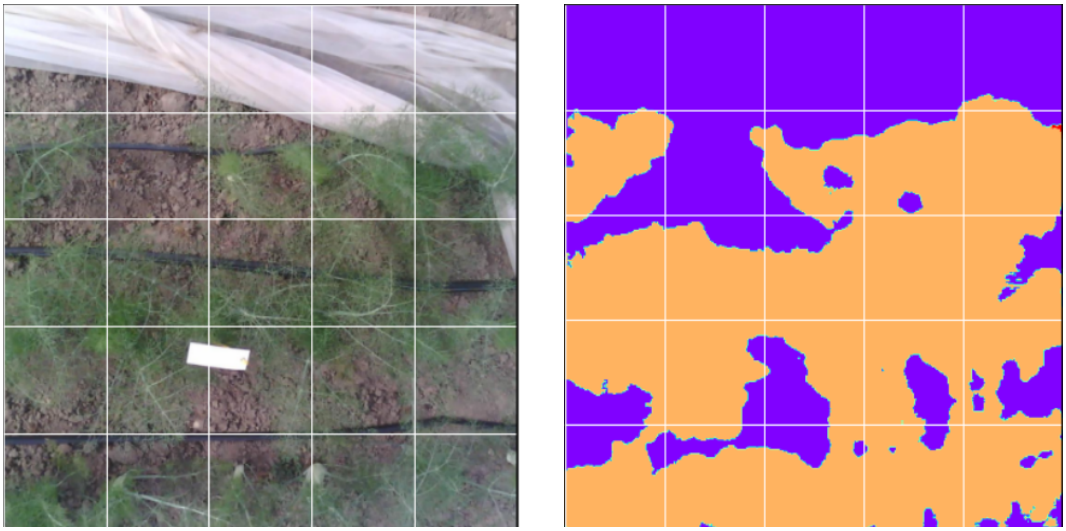


Figure 1 : Segmentation d'une image de fenouil. On voit que chaque pixel de l'image de gauche se voit attribué une classe sur le résultat de droite. Cette classe est représentée par la couleur : jaune pour le fenouil et mauve pour le reste. L'image de droite est communément appelée « masque de segmentation ».

1.1.1. Méthodes Non Supervisées

Dans la recherche en agronomie, les méthodes de phénotypage par analyse d'images sont utilisées couramment depuis longtemps. En effet, l'introduction de l'automatisation et de l'imagerie numérique à large échelle par Leister et al., (1999) a permis la collecte rapide d'images de plantes de manière non destructive. Toutefois, l'analyse requérait encore le passage d'un expert afin d'extraire l'information utile des images collectées. Par la suite, les analyses de phénotypes par analyse d'images se sont multipliées : Arvidsson et al (2011) ont créé une plateforme de segmentation d'*Arabidopsis thaliana* afin de pouvoir comparer les dynamiques de croissances de

différents génotypes. Granier et al. (2006) ont créé une plateforme de phénotypage pour *Arabidopsis thaliana* par prise d'images afin d'étudier sa réponse à différentes amplitudes de stress hydrique. Walter et al. (2007) ont, quant à eux, pu établir une relation entre régime d'intensité lumineuse et vitesse de croissance chez *Nicotiana tabacum*, grâce à des séries temporelles d'images.
Ces exemples ont pour point commun de requérir des conditions strictes, telles qu'une grande homogénéité de contexte des images ou une seule et même espèce de plante. C'est une conséquence directe du fait qu'elles fonctionnent principalement par indice de couleur ou par seuillage.

1.1.1.1. Indices de couleur

Le spectre de couleur est l'une des caractéristiques les plus couramment utilisées pour distinguer les plantes de leur environnement (Hamuda & Glavin, 2016). Plusieurs travaux de recherche ont utilisé la couleur pour séparer la plante du sol. Par exemple, dans les travaux suivants : (Rasmussen *et al.*, 2007), (Meyer & Neto, 2008) et (Kirk et al., 2009), les couleurs d'une image ont été utilisées pour distinguer les plantes du sol et ainsi estimer leur surface foliaire. Le principe de cette méthode est de convertir les images ayant trois canaux de couleurs, Rouge, Bleu, Vert (RGB), en image n'ayant plus qu'un canal, le plus discriminant possible vis-à-vis de l'objet que l'on souhaite isoler de son environnement. Pour ce faire, de nombreux indices de couleurs (combinaisons linéaires, non linéaires ou rapports de valeurs dans différents canaux) ont été développés. Une description des principaux indices utilisés dans la littérature est disponible à l'annexe 6.1.

Utilisation des indices de couleur

Les indices de couleur ne sont pas une méthode de segmentation d'image *per se*. Ils sont plutôt apparentables à des méthodes de réduction de la dimensionnalité des images. Une fois l'image réduite à un seul canal, il faut encore pouvoir classifier chaque pixel selon qu'il fait partie de la plante d'intérêt ou non. Afin d'effectivement segmenter l'image, il faut appliquer un seuil de décision. Les pixels présentant une valeur sous ce seuil vont se voir attribuer une classe et ceux sous ce seuil vont se voir en attribuer une autre. Par exemple, Bhanu & Jones (1993), ont comparé différents indices: des images prises en serres ont été analysés en utilisant l'excèdent vert (ExGR) avec un seuil fixe de zéro, l'indice de différence normalisée (NDI) avec un seuil de zéro. Ce seuil peut être, comme ici, arbitraire et fixe ou être calculé selon la méthode d'Otsu (Otsu, 1979), par exemple. Bhanu & Jones (1993) ont également comparé les indices appliqués à des images prises en champ : ExGR avec un seuil de zéro, NDI avec un seuil Otsu et ExG avec un seuil Otsu.

Bhanu & Jones (1993) ont obtenu les meilleurs résultats sur leurs images prises en extérieur, par ExGR avec un seuil de zéro et ExG avec un seuil déterminé par la méthode de (Otsu, 1979).

Inconvénient majeur des indices de couleurs

Les méthodes ayant recours aux indices de couleurs et au seuillage peuvent être performantes lorsque l'arrière-plan est uniforme. Dans ce cas, l'utilisation d'une seule métrique pour séparer les pixels végétaux et non végétaux peut être simple. Cependant, même lorsque les plantes sont imagées dans des conditions de serre, une complexité de fond supplémentaire est souvent présente. Dans certains contextes, les surfaces réfléchissantes peuvent imiter la plante et rendre les seuils choisis incorrectes (Adams et al., 2020).

Toutes ces difficultés font qu'à moins d'avoir des conditions parfaitement contrôlées et des contextes très homogènes, la performance de ces méthodes est inadéquate pour pouvoir réaliser des analyses précises (Lee et al., 2018).

1.1.1.2. Différenciation de calque

Une autre méthode populaire de segmentation d'image est la différenciation de frame introduite par Das Choudhury et al. (2018) : une image contenant l'arrière-plan est utilisée comme référence. Ensuite, la différenciation de calque identifie tous les pixels d'une image donnée qui sont différents des pixels de l'image d'arrière-plan. Une contrainte majeure est qu'il faut trouver une image de référence utilisable, représentative de l'arrière-plan de l'ensemble de la collection d'images. La différenciation de calque n'est donc possible que dans le cas d'expériences où l'arrière-plan ne change jamais. Si les images sont prises avec des arrière-plans différents, par exemple dans un champ, comme dans le cas de ce travail, la différenciation des images présente un intérêt limité.

1.1.1.3. Classification *K-Means*

Une autre approche souvent utilisée est le clustering *K-means* (MacQueen, 1967). Chaque pixel est ici considéré comme une observation, et ses intensités spectrales (e.g. RGB) sont les caractéristiques utilisées pour regrouper les pixels similaires. Il est également possible de moduler l'information des images en considérant des espaces de couleur alternatifs tels que Teinte, Saturation, Valeur (HSV) ou encore Cyan, Magenta, Jaune, Noir (CMYK), pour espérer améliorer la performance du clustering, comme par exemple dans (Klukas, Chen, & Pape, 2014). Lorsque les plantes sont jeunes, cet algorithme fonctionne particulièrement bien (Adams et al., 2020). Cependant, au fur et à mesure que les plantes vieillissent, la complexité des images croît et le nombre de clusters nécessaires à déterminer pour avoir une segmentation correcte augmente exponentiellement (Adams et al., 2020).

1.1.2. Méthodes Supervisées

Une façon de s'affranchir du manque de robustesse des méthodes non-supervisées aux contextes hétérogènes des images prises en extérieur est l'utilisation de méthodes supervisées. Le terme supervisé signifie que ces méthodes requièrent des données d'entraînement sous le format suivant :

une partie nommée mesure (ou variable dépendante) qui constitue l'information visant à être prédite, et une partie nommée « caractéristiques » (ou variables indépendantes) qui contient l'information que l'opérateur voudrait que son algorithme soit capable d'utiliser pour prédire la variable dépendante.

Dans le cas de la segmentation d'image, la variable indépendante est la classe mesurée d'un pixel et les variables indépendantes sont son spectre de couleur, par exemple RGB.

1.1.2.1. Apprentissage Automatique « Traditionnel »

L'apprentissage automatique est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune (Apprentissage automatique, 2022). Le mot « traditionnel » est couramment employé dans la littérature pour en exclure l'apprentissage profond, abordé plus loin.

De nombreuses méthodes d'apprentissage automatique ont été testées dans le domaine de la segmentation d'images de plantes : les forêts aléatoire (RF) (Breiman, 2001), les machines à vecteur de support (SVM) (Hearst et al., 1998) et les réseaux de neurones complètement connectés (FCNN). Ces derniers font partie du domaine de l'apprentissage profond et seront largement discutés dans ce travail.

Adams et al. (2020) ont comparé ces trois algorithmes avec différentes méthodes de seuillage et ont conclu que ces dernières performaient le moins bien, suivi par les SVM, puis les RF et enfin les FCNN qui ont montré la meilleur performance pour une tâche une segmentation binaire plante/arrière plant. Leurs plantes étaient vues de profil, sur fond blanc uniforme.

Lee et al. (2018) ont réalisé une analyse similaire sur des images zénithales sur fond de terre. Ils sont arrivés à des conclusions similaires avec toutefois de meilleures capacités de généralisation pour les RF.

Toutefois, une conclusion commune de ces 2 analyses est la difficulté à fournir des données d'entraînement à ces algorithmes. Lee et al. (2018) ont même créé leur propre software d'annotation à cette fin car ils ont considéré que les outils disponibles sont inadaptés à la segmentation de plantes.

1.1.2.2. Apprentissage profond

L'apprentissage profond comprend un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires (Apprentissage profond, 2022). Les concepts de non-linéarité et d'architecture sont discutés et expliqués sections 2.3.1.1 et 2.3.1.2.

L’apprentissage profond a été largement étudié pour l’analyse d’images, y compris dans l’agriculture (Kamilaris & Prenafeta-Boldú, 2018). Comparativement aux méthodes plus conventionnelles d’apprentissage automatique, l’apprentissage profond peut apprendre automatiquement les caractéristiques architecturales des plantes (telles que la forme des feuilles, de la tige, leurs couleurs), ce qui évite de devoir manuellement identifier les plus discriminantes ou d’en créer manuellement (Tang, et al., 2017). De plus, quand elles sont encodées manuellement, ces caractéristiques ne sont pas assez robustes aux contextes hétérogènes des images prises en champs. Cela mène à une faible robustesse et à une faible capacité de généralisation des modèles plus traditionnels et rend difficile l’application pratique de ces modèles, par exemple en agriculture de précision (Lottes et al., 2018 ; McCarthy et al., 2010).

Les applications de l’apprentissage profond en matière d’analyse d’images en agriculture sont nombreuses :

Dyrmann, et al. (2017) ont entraîné un réseau de neurones convolutifs (CNN) basé sur l'architecture du GoogLeNet (Szegedy, et al., 2015) pour détecter les plantes adventices dans les cultures céréalières.

Lottes et al. (2018) ont utilisé un FCNN composé d’un encodeur et d’un décodeur. Ils se sont servis de séquences d’images, ce qui a permis d’exploiter l’information contenue dans l’arrangement spatial des cultures, et d’ainsi améliorer la robustesse de leur modèle dans la segmentation sémantique d’images de cultures contenant des plantes adventices.

Milioto et al. (2018) ont utilisé un FCNN pour la segmentation sémantique d’images de cultures contenant des plantes adventices, et y ont ajouté 14 indices de végétations. Ceci leur a permis d’améliorer la segmentation des plantes avec un grand nombre de chevauchement, ainsi que la robustesse à une grande variété de stades de croissance.

Wang et al. (2020) ont montré que seul l’ajout d’un canal proche infra-rouge (NIR) augmentait significativement la performance de leur FCNN, toujours dans la tâche de segmentation sémantique d’images de cultures contenant des plantes adventices. Cet ajout est particulièrement intéressant quand les conditions d’illumination sont fort changeantes, avec la présence d’ombres, par exemple.

Ces 5 exemples montrent que la reconnaissance d’adventices sous toutes ses formes est un sujet très étudié par la littérature. Bien que ce problème puisse également être considéré comme de la segmentation de canopées multi-espèce, il est très différent de celui d’identifier chacune des espèces d’une canopée. En effet, dans la très grande majorité des cas, les adventices restent à un stade où elles sont de petite taille comparativement aux cultures d’intérêt. La taille devient donc une caractéristique très discriminante dans la plupart des cas. Tandis que dans les canopées multi-espèces de cultures maraîchères, les différentes plantes sont très souvent intriquées les unes dans les autres et la taille des individus n’est plus discernable de la masse. Ceci sera illustré dans la section Résultats.

1.1.2.3. Le problème de l'obtention de données

Comme vu plus haut, plus l'hétérogénéité des images est grande, plus les méthodes non-supervisées perdent leur intérêt. Pour un problème de segmentation multi-espèce dont les données présentent une grande hétérogénéité par nature, il paraît logique de s'orienter vers des méthode supervisées. Parmi ces méthodes, l'apprentissage profond est à même de pouvoir segmenter des images avec plusieurs espèces différentes de façon plus robuste que l'apprentissage automatique traditionnel (Lottes et al., 2018 ; McCarthy et al., 2010 ; Picon Ruiz et al. 2020).

La contrainte majeure de la segmentation sémantique par apprentissage profond est qu'elle nécessite un très grand travail d'annotation manuelle pour les images en extérieur. Ce travail vient du fait que, comme mentionné précédemment, l'apprentissage supervisé requiert de fournir aux algorithmes des exemples de la tâche que l'on souhaite les voir accomplir. Fournir ces exemples implique, dans le cas de la segmentation de canopées multi-espèces, d'annoter des images de cultures en entourant des zones contenant des plantes et d'attribuer à ces zones des classes, correspondant à leurs espèces. Cela peut s'avérer compliqué car dans le cas où les espèces sont très intriquées, les zones à entourer peuvent devenir complexes.

Ces annotations (ou labels) peuvent s'avérer non seulement pénible, mais aussi coûteuses à produire (Picon et al., 2022 ; Rawat et al., 2022 ; Zhou et al., 2017 ; Sayez & De Vleeschouwer, 2022 ; Ratner, Varma, & Hancock, 2019). Une durée de 10 heures par image a même été rapportée par Fuentes-Pacheco et al. (2019).

De plus en plus d'auteurs tentent de s'affranchir de cette contrainte. On voit, par exemple, l'émergence de l'utilisation de jeux de données synthétiques (Picon et al., 2022 ; Barth et al., 2019 ; Barth et al. 2018).

On voit aussi la recrudescence de l'apprentissage actif, un sous-domaine de l'apprentissage automatique, qui a pour objectif de réduire le travail d'annotation en sélectionnant et en annotant uniquement les échantillons les plus informatifs. L'hypothèse clé de l'apprentissage actif est que si l'algorithme choisit les données à partir desquelles il apprend, il sera plus performant avec moins de données, par rapport à un apprentissage basé sur des données choisies aléatoirement (Settles, 2019). Une des façons les plus communes de sélectionner les images les plus importantes à annoter est d'utiliser l'incertitude des modèles afin d'identifier les images les plus rentables à annoter (Zhou et al., 2017 ; Guyon et al., 2011 ; Wang et al., 2016). Un autre indicateur couramment utilisé est la diversité de prédiction, par exemple, entre différents morceaux d'une même image (Zhou, et al., 2017).

L'autre grand paradigme utilisé pour réduire le labeur nécessaire à l'annotation est la supervision faible. C'est une approche de l'apprentissage automatique qui consiste à utiliser des données moins exhaustives et souvent plus bruitées pour créer des jeux de données d'entraînement plus volumineux et plus rapidement que ce qui pourrait être produit par une supervision complète (Ratner et al., 2019). Il a également été suggéré que le fait d'avoir un jeu de données moins précis mais plus diversifié et plus volumineux qu'un jeu de donnée parfait mais moins diversifié et moins volumineux pouvait amener à de meilleur résultats (Sayez & De Vleeschouwer, 2022). Ce paradigme a déjà été utilisé en l'agriculture par Wang et al. (2022) qui ont entraîné un modèle sur

des champs en France et l'ont utilisé pour délimiter des champs en Inde avec succès. Kim et al. (2021) ont d'abord entraîné un classificateur de champs récolté/non-récolté, en ont extrait les cartes d'activation de classes (CAM). Celles-ci sont grossièrement des cartes des zones discriminantes d'une image, les zones qu'un réseau de neurones classificateurs a utilisées pour prendre sa décision sur la classe à prédire (Zhou et al., 2016). Elles permettent donc d'utiliser l'avantage des classificateurs d'être plus simples à entraîner, par la plus faible complexité de leurs données d'entraînement, pour en extraire une information sur la localisation des objets. Cette information étant une zone de l'image mettant en évidence les pixels discriminants, elle est parfois suffisante pour sélectionner automatiquement les pixels qui serviront donnée d'entraînement pour une tâche de segmentation.

Plusieurs voies peuvent être suivies pour réduire le travail d'annotation dans la segmentation de plantes. (Sheikh, et al., 2020) ont, dans le but de segmenter des betteraves et des mauvaises herbes, utilisé des pseudo-labels. Ceux-ci sont basés sur une supervision faible à base de clustering K-means. Ensuite, ils ont calculé les normes des gradients de rétropropagation, une mesure qui indique quels échantillons affecteront le plus les paramètres, dans le cas de modèles entraînés par descente de gradient. Les échantillons ainsi détectés feront alors l'objet d'une supervision plus forte. Les auteurs montrent qu'une telle stratégie fonctionne mieux qu'une sélection aléatoire des échantillons à annoter de façon précise. Picon et al. (2022) et Güldenring et al. (2021) ont synthétisé un jeu de données pour entraîner un modèle puis l'utiliser pour segmenter de vraies images. Les premiers ont ainsi traité un cas de segmentation de mauvaises herbes dans un fonds de maïs tandis que les seconds ont généré quelques exemples de feuilles de mauvaises herbes qu'ils ont ensuite « collé » sur des images de différents fonds pour pouvoir segmenter de vraies mauvaises herbes de façon robuste au type de champ. Riehle et al. (2020) ont utilisé des indices de couleurs ainsi qu'une méthode de seuil automatique pour obtenir des masques de segmentation approximatifs, utilisables par d'autres algorithmes pour segmenter des images de maïs et de betteraves. Petti & Li (2022) travaillent sur la segmentation de fleurs de coton et utilisent un modèle entraîné sur une petite partie de leur jeu de données annotée manuellement, puis l'utilisent pour inférer sur une autre petite partie de leur jeu de données et corrigent les erreurs. Ils montrent qu'il peut être plus rapide de corriger des erreurs d'annotations que de les construire à partir de rien.

1.1.3. Conclusion de l'état de l'art

Plusieurs éléments ressortent de cette revue de la littérature :

1. Les méthodes non supervisées ne fonctionnent que pour des cas bien particuliers d'images de parcelles mono-espèces, très homogènes ou en conditions contrôlées.
2. Lorsque la segmentation à réaliser est binaire (i.e. Plante/Fond) les méthodes dites traditionnelles d'apprentissage automatique peuvent suffire et ont l'avantage de requérir un travail d'annotation plus faible que l'apprentissage profond.
3. Quand la segmentation est non-binaire (multi-label), l'apprentissage profond apporte des résultats plus robustes, par ses facultés à apprendre automatiquement les caractéristiques

architecturales des plantes. Cependant, dans sa forme brute, il requiert un travail d'annotation souvent trop coûteux pour être raisonnable.

4. L'apprentissage actif et la supervision faible ont montré des résultats très prometteurs pour utiliser l'apprentissage profond dans la segmentation d'images de cultures en permettant de réduire le coût d'annotation des images
5. A notre connaissance, il n'y a à ce jour aucune étude évaluant les possibilités d'utilisation de l'apprentissage actif et/ou de la supervision faible dans la segmentation de canopées multi-espèces, hormis dans le cas particulier des adventices.

Au vu de la complexité des images de canopées multi-espèces et, par conséquent du coût de leur annotation manuelle, il paraît nécessaire d'explorer les possibilités qu'offrent la supervision faible et l'apprentissage actif afin de faciliter la segmentation sémantique de ce type d'image.

1.2. Objectifs

L'objectif de ce travail est d'aborder un des problèmes de la segmentation supervisée : l'obtention de données d'entraînement. Au travers des notions d'apprentissage actif et de supervision faible, ce travail présente une méthode combinant arbres de décision et réseaux de neurones convolutifs ayant pour but de réduire le temps et le travail nécessaire pour entraîner pour fournir à un modèle des masques de qualité suffisante pour pouvoir segmenter efficacement des images de canopées multi-espèces.

C'est, à notre connaissance, la première fois que des arbres de décision boostés par gradient sont utilisés pour générer des pseudo-labels pour une tâche de segmentation. C'est également la première fois qu'une tâche de segmentation de cultures associées utilise des données d'entraînement basées uniquement sur des cultures pures.

2. Matériel et Méthodes

Cette section vise tout d'abord à exposer le type d'images utilisées dans ce travail, avant de présenter la méthode proposée.

Ce travail s'est déroulé en 5 étapes :

0. la collecte d'images de cultures pures dont l'espèce est indiquée (réalisé dans le cadre de Chevalier (2022))
1. la segmentation binaire de ces images grâce à des arbres de décision boostés sous forme d'apprentissage actif
2. la conversion de ces masques binaires en masques multi-classe grâce aux labels associés aux parcelles
3. l'entraînement d'un réseau de neurones complètement convolutifs (FCNN) sur ces images et leurs masques de segmentation obtenus à l'étape précédente
4. l'utilisation du FCNN pour prédire les masques de segmentation sur des images des canopées de cultures associées.

Chacune des étapes 1 à 4 sera vue en détails. Un schéma illustratif est disponible Figure 2.

Enfin, le lecteur pourra trouver une description des logiciels utilisés, ainsi que le matériel informatique.

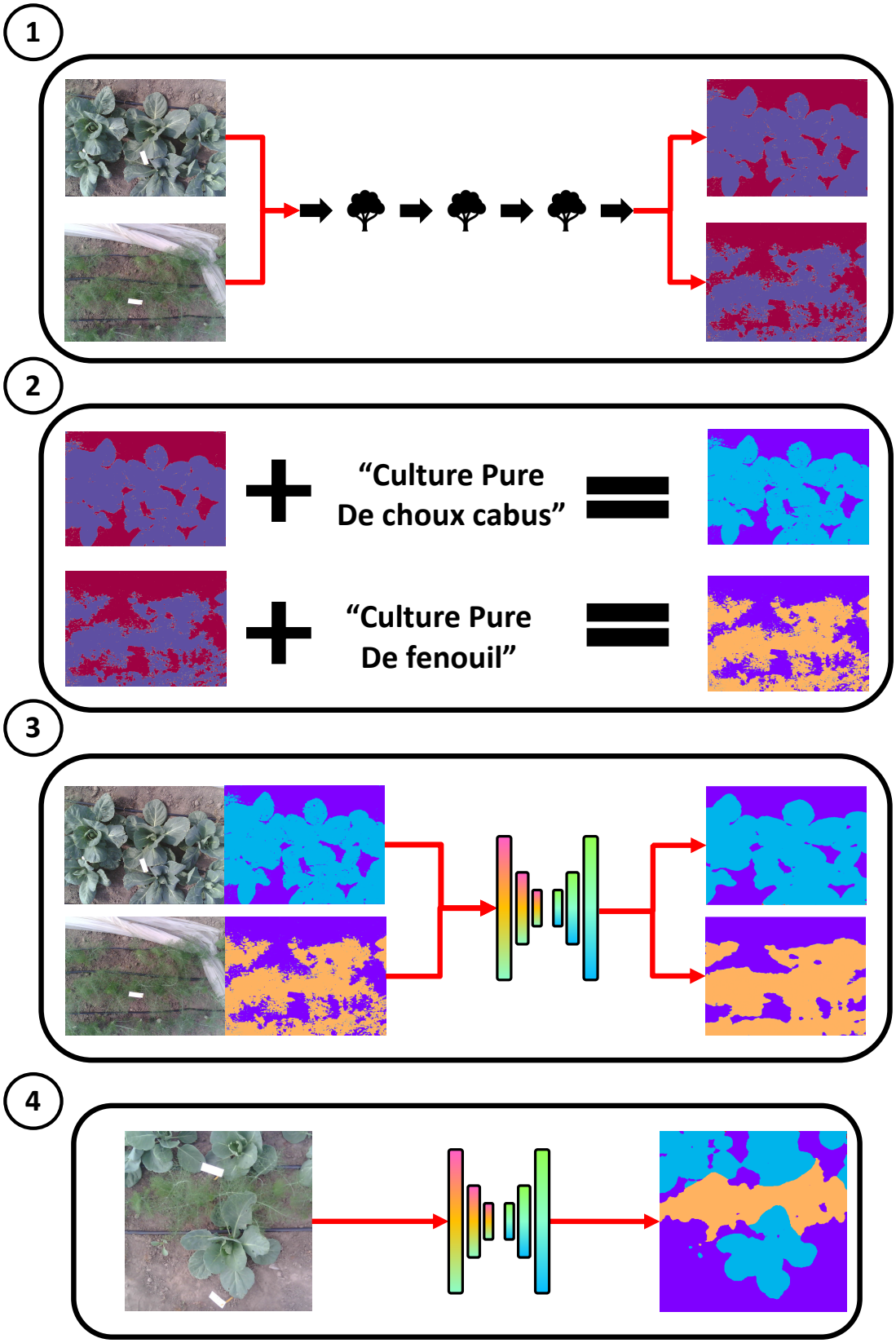


Figure 2 : Schéma récapitulatif de la méthode globale. 1) segmentation binaire des images de cultures pures grave à LeafPainter. 2) Conversion des masques binaires en masques multi classes grâce à la connaissance de la culture pure concernée. 3) Entrainement d'un FCNN à segmenter les images de cultures pures en fonction de l'espèce des plantes. 4) Utilisation du FCNN entraîné pour inférer sur les images de cultures associées et obtenir leurs segmentations.

2.1. Jeu de données

Le jeu de données contient 2386 images RGB zénithales de parcelles de cultures, prises à approximativement 2 mètres du sol. Les parcelles peuvent contenir soit une culture pure de légumes, soit une culture associée de 2 légumes. Il y a au total 8 espèces de légumes : Brocolis (BR), Fenouil (FE), Haricot Nain (HN), Haricot à Rame (HR), Chou Cabus (CC), Chou Frisé (CF), Oignon Blanc (OB) et Laitue (L). Le jeu de données comporte des images de chaque espèce en culture pure et de chaque combinaison possible de 2 légumes, excepté pour les laitues qui sont soit en culture pure, soit en culture associée avec des Haricots Nains. Plusieurs plantes de la même espèce apparaissent sur chaque photo. A chacune de ces photos est associé le nom de l'espèce ou des espèces y figurant. C'est une information importante pour la suite car cela signifie que pour les images en cultures pures, on peut poser l'hypothèse que l'espèce de plante de chacun des pixels de la classe « plante » est connue. Cela n'est pas exact car il peut y avoir des mauvaises herbes ou des plantes d'une parcelle voisine qui débordent sur celle prise en photo, mais pour la grande majorité des pixels, cette hypothèse est vérifiée.

Chaque parcelle a été prise en photo 4 fois, au cours des 4 mois de l'expérience : Juillet, Aout, Septembre, Octobre. Cela implique que les plantes ont été prises en photo à 4 stades de croissance différents et selon des ensoleillements différents.

Ces photos ont été prises soit à la ferme de Lauzelle, soit chez un des 20 maraîchers volontaires. Ces images présentent donc des contextes variables par leurs localisations, lesquelles amène différents types de sols, agencements des cultures, outils présents sur les parcelles, etc… Un autre type de diversité est amenée par le fait que des maraîchers différents peuvent avoir des façons différentes de cultiver, ce qui peut se refléter sur les plantes elles-mêmes. On peut par exemple avoir des plantes ayant des contenus en azote différents, ce qui va directement se répercuter sur leur couleur.

Nous avons donc une très grande hétérogénéité, due à 5 facteurs : les combinaisons d'espèces, les stades de croissance, les conditions lumineuses, le contexte géographique et les techniques culturales des agriculteurs.

L'acquisition des images a été réalisé dans le cadre des travaux de Chevalier (2022). l'appareil photo utilisé était constitué de 6 modules OEM de caméra OV2640 (Omnivision), RGB, 2Mpixels, embarqué sur un module ESP32-Cam (https://docs.ai-thinker.com/en/esp32-cam), installés parallèlement l'un à l'autre, à 3 cm l'un de l'autre sur une grille de 3 colonnes et 2 lignes et équipés d'un filtre passe-bande à 450x10, 568x10, 676x10, 700x25, 730x10, 850x25 nm de la marque Edmund Optics.

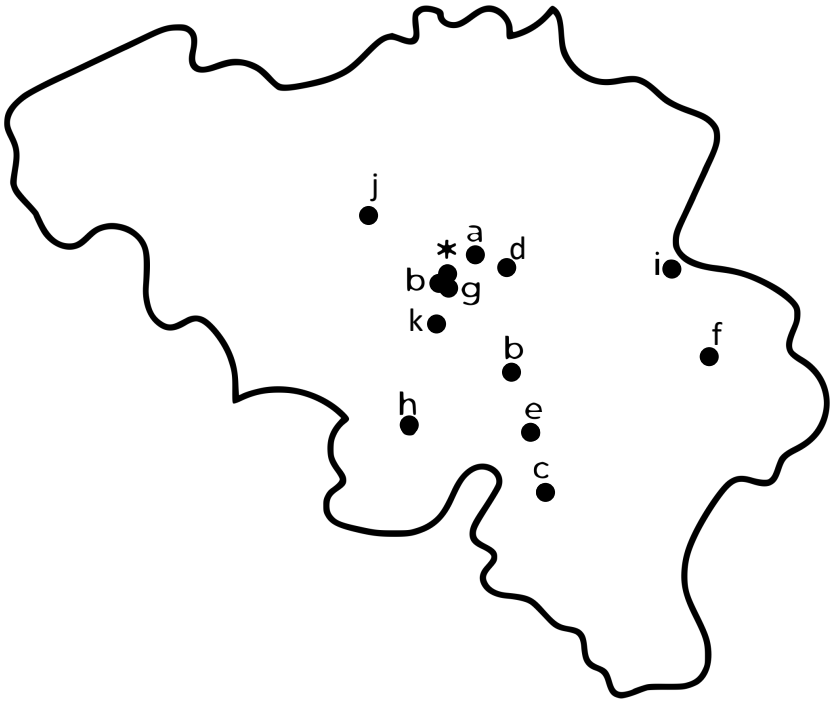


Figure 3 - Localisation des quatorze parcelles où des photos ont été prises. Les lettres représentent les onze maraîchers participants. Le symbole de l'astérisque représente la parcelle de la ferme de Lauzelle. Seuls treize points sont représentés sur la carte, car deux des quatorze parcelles étaient situées au même endroit (c). Ceci est la figure 33 de (Chevalier, 2022)

2.1.1. Étape 0 : collecte d'images de cultures pures dont l'espèce est indiquée

Ces images ont été collectées en même temps que les images d'associations par Chevalier (2022). Puisqu'il n'y a en théorie qu'une seule espèce sur la photo, tous les pixels segmentés comme « plante » sont de cette espèce. Cela signifie que nous n'avons besoin de segmenter ces images que de façon binaire (plante : fond) pour constituer un jeu de données bi-classes. Comme vu précédemment, la segmentation binaire peut être réalisée par des techniques d'apprentissage automatique traditionnelles bien moins exigeantes en données, puisqu'elles peuvent fonctionner uniquement sur base de la couleur des pixels, contrairement aux FCNN. Cela implique cependant qu'il faut entraîner le modèle en ne lui fournissant que des exemples de pixels isolés. Pour ce faire, nous avons développé une petite application appelée « LeafPainter » dans laquelle l'utilisateur peut charger une image, entourer manuellement une zone à l'aide de la souris et lui attribuer un label (plante ou fond). L'application va extraire chaque pixel de la zone puis stocker ses valeurs RGB ainsi que sa classe dans une table de données, comme illustré sur la Figure 4.

2.2. Segmentation Binaire et Apprentissage actif

2.2.1. Les arbres de décision Boostés par gradient : explication de l'algorithme

Cette étape a pour but de segmenter les images de façon binaire, c'est-à-dire de n'avoir que 2 classes à prédire : Fond ou Plante.

L'algorithme choisi est un ensemble d'arbres de décision boostés par gradient (GBDT). Cet algorithme est une technique d'apprentissage automatique utilisée dans les tâches de régression et de classification, entre autres. Le principe repose sur un modèle de prédiction qui est un ensemble d'arbres de décision faibles. Ces arbres sont créés de manière séquentielle, chaque arbre ayant grossièrement pour tâche de corriger les erreurs de l'arbre qui le précède dans la séquence (Breinman, 1997).

Explication intuitive de l'algorithme

Le principe d'un ensemble d'arbres de décision boosté par gradient pour un problème de classification est le suivant.

1. Prédire pour chaque échantillon (c'est-à-dire un la classe d'un pixel et son spectre de couleur RGB) le logarithme de la cote associée aux valeurs observées.
   Par exemple, si on a 4 échantillons appartenant à la classe 1 et 2 échantillons appartenant à la classe 0, la prédiction sera de 0.69
2. Convertir cette prédiction en probabilité d'appartenir à la classe 1.
   Par exemple, dans ce cas la probabilité d'appartenir à la classe 1 sera de 0.66
3. Calculer l'erreur de prédiction (ou pseudo-résidu) de chaque échantillon par rapport à la probabilité : (réalité – prédiction)
   Par exemple, dans le cas d'une prédiction de 0.66, les échantillons appartenant à la classe 1 auront un pseudo-résidu de 0.44 et ceux appartenant à la classe 0 auront un pseudo-résidu de -0.66.
4. Construire un arbre de décision ayant pour but de prédire les pseudo-résidus pour chaque échantillon.
5. Convertir les pseudo-résidus prédis à chaque feuille (c'est-à-dire les régions terminales des arbres) en logarithmes de cotes.
6. Pour chaque échantillon, ajouter à la prédiction de l'arbre précédent (ou au logarithme de la cote prédite à l'étape 1 si on est dans la première itération de l'algorithme) l'output de l'étape 5 associé à cet échantillon, multiplié par une constante appelée taux d'apprentissage
7. Revenir à l'étape 2

Explication formelle de l'algorithme

Soient

- y_i : la classe mesurée de l' échantillon *i*
- $\widehat{y_i}$: la classe prédite de l'échantillon *i*
- n : le nombre total d'échantillons
- M : le nombre total d'arbres de décisions du modèle
- f_0 : le tout premier sous-modèle, qui initialise la prédiction pour chaque échantillon
- f_l : arbre de décision avec $l \geq 1$)
- F_j : le modèle au stade *j,* c'est l'ensemble des sous-modèles f_0 à f_j : $F_j = \{f_0, \dots, f_j\}$
- F : $\{f_0, f_j, \dots, f_M\}$, le modèle au stade final
- p_{ij} : la probabilité prédite par F_j que l'échantillon i appartienne à la classe 1
- $cote_{ij}$: la cote de l'échantillon *i* d'appartenir à la classe 1 au stade *j*. $cote_{ij} = \frac{p_{ij}}{1-p_{ij}}$
- x_i : les variables indépendantes de l'échantillon *i*
- **X** : l'ensemble des variables indépendantes de tous les échantillons $\mathbf{X} = \{x_0, \dots, x_n\}$
- r_{ij} : le pseudo-résidu de prédiction de l'échantillon *i* par F*j*
- R_{kj} : la $k^{ème}$ feuille de l'arbre *j*
- lr : la constante d'apprentissage
- L : la fonction de coût à minimiser : le log(vraisemblance) :

$$\mathbf{L(y_i, p_{ij}) = -(y_i log(p_{ij}) + (1 - y_i) log(1 - p_{ij}))} \qquad \textbf{Eq. 10}$$

où log est le logarithme népérien.
Puisque le modèle ne prédit pas des probabilités mais des logarithmes de cotes, et au vu de l'identité

$$\mathbf{p_{ij} = \frac{exp\left(log(cote_{ij})\right)}{1 + exp\left(log(cote_{ij})\right)}} \qquad \textbf{Eq. 11}$$

On peut montrer que l'équation 10 peut être réécrite sous la forme :

$$\mathbf{L\left(y_i, log(cote_{ij})\right) = -y_i log(cote_{ij}) + log\left(1 + exp(log(cote_{ij}))\right)} \qquad \textbf{Eq. 12}$$

et son gradient :

$$\frac{d\,L\left(y_i, \log(cote_{ij})\right)}{d\,\log(cote_{ij})} = -y_i + \frac{\exp(\log(cote_i))}{1 + \exp(\log(cote_i))} \qquad \textbf{Eq. 13}$$

Grace à l'identité de l'équation 11, l'équation 13 peut être exprimée en fonction de p_{ij} :

$$\frac{d\,L\left(y_i, \log(cote_{ij})\right)}{d\,\log(cote_{ij})} = p_{ij} - y_i \qquad \textbf{Eq. 14}$$

L'étape **1** devient donc: initialiser le modèle avec une valeur constante telle que $f_0(\mathbf{X}) = \underset{\log(cote)}{argmin} \sum_{\{i=1\}}^{n} L(y_i, \log(cote))$. Cette valeur correspond en réalité au logarithme de la cote associée aux valeurs observées.

Les étapes **2 et 3** deviennent : calculer les pseudo résidus : $r_{ij} = \frac{dL\left(y_i\, F_j(x_i)\right)}{dF_j(x_i)}$

L'étape **4** devient : créer un arbre de décision $f_{j+1}(x_i) = r_{ij}$ pour chaque *(i,j)*.

Comme on limite le nombre maximal de feuilles de chaque arbre, il peut y avoir plus de pseudo-résidus (r) que de feuilles (R). Une même feuille R_{kj} peut donc contenir plusieurs pseudo-résidus r.

L'étape suivante consiste à déterminer la prédiction de chaque $R_{k,j+1}$. Au stade actuel, elles ne contiennent que des pseudo-résidus et le modèle doit prédire des log(cote). Cette valeur de log(cote) prédite à partir des résidus de chaque feuille $R_{k,j+1}$ est calculée à partir de l'équation suivante :

$$\log\left(cote_{k,j+1}\right) = \underset{\log(cote)}{argmin} \sum_{x_i \in R_{k,j+1}} L\left(y_i, \mathrm{F_j}(x_i) + \log(cote)\right) \qquad \textbf{Eq. 15}$$

On peut montrer que la solution de l'équation 15 est :

$$\log\left(cote_{k,j+1}\right) = \frac{\sum_{r_{ij} \in R_{k,j+1}} r_{ij}}{\sum_{i=1}^{n} p_{ij}\left(1 - p_{ij}\right)} \qquad \textbf{Eq. 16}$$

L'étape **5** devient : conversion des $r_{ij} = f_{j+1}(x_i)$ en $\log(cote_{k,j+1})$âce à l'équation 16.

L'étape **6** devient : mise à jour du modèle : $F_{j+1}(x_i) = F_j(x_i) + lr \log(cote_{k,j+1})$'étape **7** devient : retour à l'étape **2** jusqu'à convergence ou arrêt prématuré de la part de l'utilisateur.

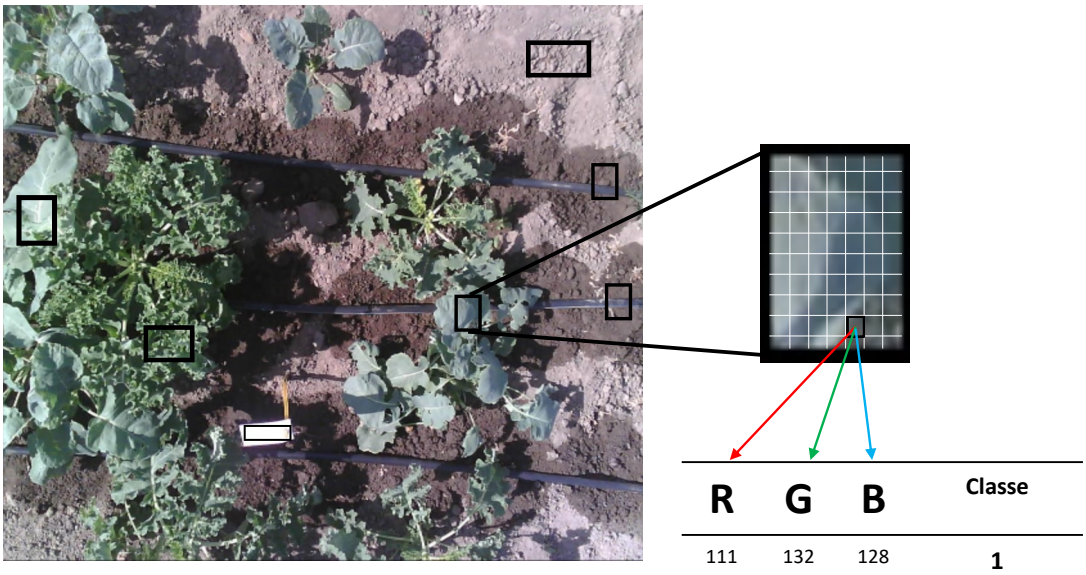


Figure 4: Exemple illustratif de conversion des zones annotées en table de données structurées. Des zones de l'image sont annotées, les pixels en sont extraits et leurs valeurs Rouge (R) Verte (V) et Bleue (B) sont notées dans une table. L'utilisateur spécifie ensuite la classe de la zone qu'il a annotée (Plante = 1, Fond = 0) et elle est ajoutée à toutes les lignes correspondant aux pixels de la zone annotée.

2.2.2. Les arbres de décision Boostés par gradient : Implémentation

Le choix de l'implémentation s'est porté sur la librairie LightGBM version 3.2.0 de Microsoft (Ke, et al., 2017). Cette librairie, écrite en Python, offre une implémentation robuste, testée et validée par des milliers d'utilisateurs dans le monde. C'est également l'implémentation qui offre la meilleure performance en termes de vitesse, de mémoire utilisée et de précision (Ke, et al., 2017), bien que dernier point soit contesté (Hancock & Khoshgoftaar, 2020). Même si ces trois points ont une grande importance, c'est surtout celui de la vitesse qui a été déterminant dans ce choix de l'implémentation (voir section Apprentissage Actif).

2.2.3. Les arbres de décision Boostés par gradient : Optimisation des hyperparamètres

Afin de tirer le meilleur parti des arbres de décision boostés par gradient, il convient d'optimiser leurs hyperparamètres (i.e. les paramètres non apprenables par le model, que l'utilisateur doit fournir lui-même). Les hyperparamètres communément acceptés comme étant les plus importants (Boehmke & Greenwell, 2017), (Scikit-Learn, 2021) sont listés dans le Tableau 1.

Afin d'optimiser ces hyperparamètres, un jeu de données d'approximativement 10 millions de pixels a été constitué en annotant des zones d'images choisies arbitrairement, suivant le protocole illustré sur la Figure 4. Ces pixels proviennent de 68 images. Ces images ont été sélectionnées de sorte à ce que le jeu de données contiennent des images de chaque espèce de plante, vue chaque mois, à chaque stade de croissance. Cette diversité permet de s'assurer que ce jeu de données restreint est bien représentatif de la distribution du jeu de données complet. Le jeu de données a également été créé de sorte à avoir approximativement autant de pixels de plantes que de fonds.

Ensuite, les paramètres ont été optimisés par validation croisée :

1. Un ensemble de valeur des hyperparamètres est choisi.
2. 70% des pixels du jeu de données restreint, choisis aléatoirement, sont utilisés comme données d'entraînement, tandis que les 30% restant sont utilisés pour mesurer la performance du modèle, à la fin de l'entraînement. Ces 2 sous jeux de données se nomment respectivement le set d'entraînement et le set de validation.
3. Le modèle s'entraîne sur le set d'entraînement et mesure sa performance sur le set de validation.
4. L'étape 3 est répétée 20 fois
5. Les 20 performances mesurées sur le set de validation sont moyennées et deviennent le score de l'ensemble de valeurs d'hyper paramètres déterminés à l'étape 1
6. On retourne à l'étape 1 ou on arrête de la validation croisée si l'on estime avoir testé suffisamment de combinaisons d'hyperparamètres.
7. L'ensemble d'hyperparamètres ayant obtenu le meilleur score devient l'ensemble choisi pour le modèle final.

Cette validation a été réalisée avec la fonction « *gridsearch* » de la plateforme *Weights & Biases* (Biewald, 2020). Cette fonction prend en entrée une liste de valeurs pour un ensemble d'hyperparamètres et réalise la validation croisée sur toutes les combinaisons possibles de ces valeurs d'hyperparamètres. Les intérêts de cette plateforme sont surtout son ergonomie, son interface utilisateur permettant de suivre l'avancement de la validation croisée à distance et les possibilités qu'elle offre de pouvoir sauvegarder ses résultats et les consulter à postériori de manière très simple.

Afin d'évaluer la performance d'un modèle, il convient tout d'abord de déterminer une métrique. Dans le cas d'une classification, la précision peut ne pas suffire. En effet, il est important de considérer les taux de faux positifs et de faux négatifs afin de pouvoir caractériser la nature des erreurs commises. Dans notre cas, grâce en partie à l'attention portée à l'équilibrage des données, ces taux étaient similaires et la précision seule s'est avérée suffisante pour déterminer le choix des valeurs des hyper paramètres. Ces valeurs sont indiquées dans le Tableau 1.

La précision obtenue est de 84.8% sur le set de validation.

Tableau 1 : ***Hyperparamètres des GBDT les plus importants, leurs valeurs après optimisation et leur nature.***

Paramètre	*Valeur optimisée*	*Nature du Paramètre*
Nombre d'arbres	100	Le nombre d'arbres que contiendra le modèle à la fin de l'apprentissage
Constante d'apprentissage	0.05	La constante par laquelle sera multiplié la sortie du dernier arbre créé avant d'être ajoutée au modèle temporaire. Autrement dit, la contribution de chaque arbre à la sortie finale.
Profondeur maximale d'un arbre	8	Le nombre maximal d'étages que peut avoir un arbre avant une feuille.
Nombre minimal d'échantillons par feuille	2	Le nombre minimal d'échantillons pouvant être regroupés dans une seule feuille.

2.2.4. Étape 1 : La segmentation binaire de ces images grâce à des arbres de décision boostés sous forme d'apprentissage actif

L'annotation de zones aléatoires d'images ne parait pas très efficiente, ni termes de temps, ni en termes de données. L'idéal serait de pouvoir connaitre pour chaque image les régions causant de la confusion au modèle afin de n'annoter que celles-là. Comme expliqué dans la revue de la littérature, l'apprentissage actif a précisément cette vocation.

Nous proposons dans ce travail une méthode itérative simple ayant pour but de n'annoter que les zones des images pour lesquelles les prédictions sont fausses. La méthode consiste en 3 étapes (Une démonstration visuelle est disponible à l'adresse https://www.youtube.com/watch?v=lMpzxDt40Lc):

1. Annoter une zone de sol et de plante et faire une première prédiction sur le reste de l'image.
2. Annoter quelques zones où le modèle se trompe.
3. Refaire une prédiction sur le reste de l'image.
4. Revenir à l'étape 2 si l'utilisateur n'est pas satisfait de la qualité du masque généré.

Pour implémenter cette méthode, nous avons créé une application appelée LeafPainter. Celle-ci permet de se connecter à une base de données SQLite contenant la collection d'images, de rédiger des requêtes pour charger certaines images dans l'application, de réaliser la segmentation binaire décrite par les étapes ci-dessus et de télécharger ensuite le masque binaire généré. Il est également possible d'y charger n'importe quelle image directement, sans recourir à une base de données.

Elle a été réalisée avec le langage de programmation Python (Van Rossum & Drake Jr, 1994), grâce à la librairie Flask (Grinberg, 2018) qui permet de créer des applications en Python. L'interface a été réalisée en HTML5 (Hickson & Hyatt, 2011), CSS (Lie, 1994) et Javascript (Eich, 2005).

2.3. Segmentation Multi-Classes

Une fois les masques binaires sur images de monocultures générés, nous pouvons entraîner un FCNN à segmenter ces images, de façon multi-classes, cette fois. Cela signifie que le FCNN devra non seulement reconnaitre les pixels de plantes mais également l'espèce à laquelle ils appartiennent. Pour rappel, comme expliqué section 2.1, cela n'est possible que parce que les masques que nous avons générés avec LeafPainter concernent des images de monocultures pour lesquelles les espèces sont connues, et donc les espèces de chaque pixel de plantes, sont connues.

2.3.1. Les Réseaux de neurones complètement convolutifs (FCNN) : explications

2.3.1.1. Les réseaux de neurones complètement connectés

Avant d'aborder le sujet des FCNN, il convient d'expliquer ce que sont les réseaux de neurones (NN).

Un NN est un ensemble d'unités connectées appelées neurones. Chaque connexion, peut transmettre un signal entre les neurones qu'elle connecte. Quand un neurone reçoit un signal, il lui applique une transformation, puis le transmet à d'autres neurones. Un neurone peut recevoir des signaux de plusieurs neurones, auquel cas ils seront sommés puis cette somme subira une transformation. Cette

transformation est une fonction qui porte le nom de fonction d'activation. La sortie de cette fonction d'activation est l'activation du neurone. Les connexions entre neurones ont un poids qui s'ajuste au fur et à mesure de l'apprentissage. Ce poids est un facteur multiplicatif appliqué au signal de sortie d'un neurone. Il peut donc l'amplifier ou le réduire. En règle générale, les neurones sont rassemblés en couches. Ces différentes couches peuvent effectuer différentes transformations sur leurs entrées. Les signaux voyagent de la première couche (la couche d'entrée) à la dernière couche (la couche de sortie) en passant par les couches intermédiaires (les couches cachées). Cette étape s'appelle « la passe avant ». La fonction d'activation est dans la grande majorité des cas une transformation non-linéaire. Les plus connues sont les fonction sigmoïde et ReLU :

$$\boldsymbol{Sigmoide(x) = \frac{1}{1 + exp(-x)}} \qquad \textbf{Eq. 17}$$

$$\boldsymbol{ReLU(x) = \max(0, x)} \qquad \textbf{Eq. 18}$$

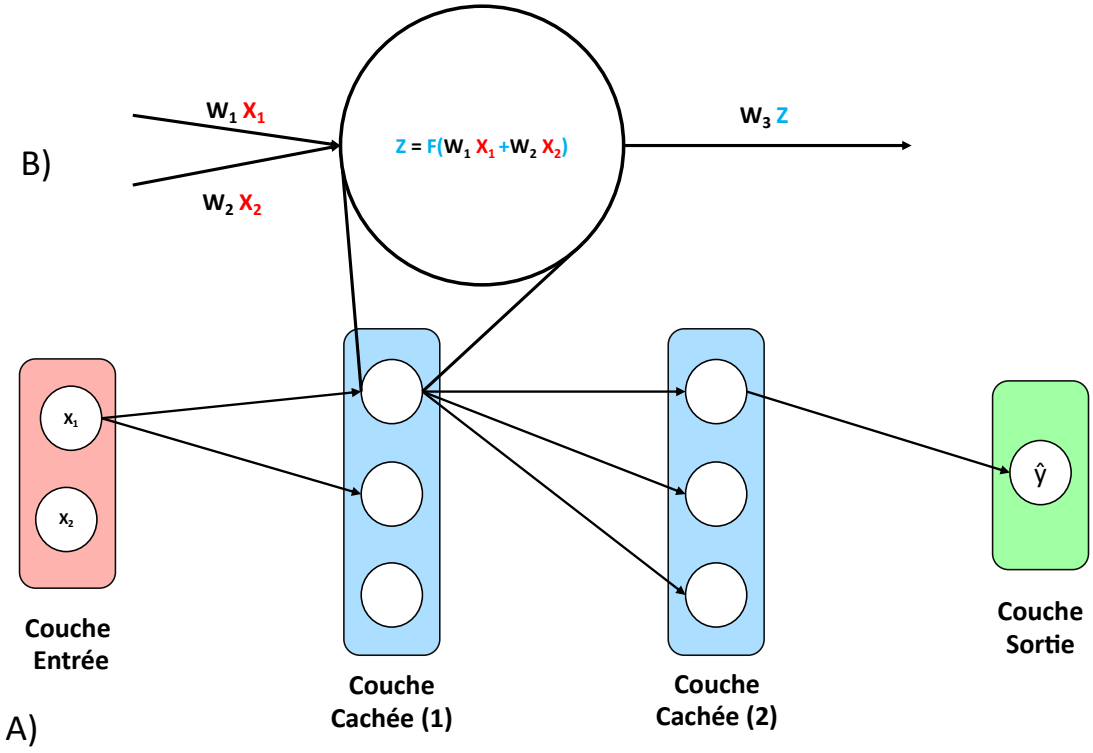


Figure 5 : Représentation schématique d'un réseau de neurones. A) Réseau entier. On peut y voir 3 types couches : la couche d'entrée, en rouge, contenant les variables indépendantes d'un échantillon, 2 couches dites cachées en bleu et la couche de sortie contenant la prédiction du réseau. Les flèches symbolisent les connexions entre neurones ainsi que leurs poids. Par souci de clarté, seules les connexions d'un neurone par couche ont été représentées. B) Vue plus détaillée d'un neurone. Les W sont les poids. Les X sont les variables indépendantes. F est la fonction d'activation et Z est la valeur d'activation du neurone. Le biais est volontairement omis par souci de cohésion.

Le but de la phase d'apprentissage d'un réseau de neurones va être d'optimiser les valeurs de ses poids afin de minimiser une fonction de coût. Par exemple, dans le cas d'une régression, cela peut être l'erreur carrée moyenne. Pour ce faire, les réseaux de neurones mettent itérativement à jour les différents poids et biais en fonction de l'impact qu'ils ont eu sur l'erreur de prédiction, cet impact est calculé lors d'une étape appelée « passe arrière ». Cette mise à jour itérative des paramètres s'appelle la descente de gradient :

$$w := w - lr\,\nabla Q(w) \qquad \text{Eq. 19}$$

Où w est l'ensemble des paramètres du réseau, lr est la constante d'apprentissage et Q est la fonction de coût à optimiser. Le symbole Nabla signifie « gradient ». Par « gradient », on entend la dérivée partielle de l'erreur de prédiction par rapport à un paramètre particulier.

Il existe de nombreuses méthodes pour utiliser le gradient pour optimiser les paramètres d'un NN. Ces méthodes sont couramment nommées des optimiseurs. Les 2 les plus connus sont :

- La descente de gradient stochastique (SGD) qui est quasiment identique à l'équation 19, sauf qu'elle s'applique successivement à des sous-échantillons aléatoires du jeu de données (appelés des « mini-lots » de données), plutôt qu'à l'ensemble du jeu de données.
- L'estimation adaptative du moment (Adam) (Kingma & Ba, 2014): qui est un des optimiseurs les plus utilisés dans la littérature (He et al., 2021) :

$$w := w - lr \frac{m_{\nabla Q(w)}}{\sqrt{v_{\nabla Q(w)}} + e} \qquad \textbf{Eq. 20}$$

où $m_{\nabla Q(w)}$ est le moment d'ordre 1 du gradient de la fonction de coût en fonction des paramètres, calculé en réalisant une moyenne mobile exponentiellement pondérée de ce gradient. Grâce à ce terme, la mise à jour des paramètres est réalisée, non pas uniquement sur le gradient en un point précis de la fonction de coût, mais en tenant compte de l'historique de ce gradient. Intuitivement, ce terme peut être comparé à l'inertie du gradient, où à une balle qui roule sur une pente. Son mouvement ne dépend pas uniquement du point où elle se trouve, mais également de sa vitesse accumulée sur son chemin. $v_{\nabla Q(w)}$ est le moment d'ordre 2 de ce même gradient, calculé en réalisant une moyenne mobile exponentiellement pondérée du carré de ce gradient. Son intérêt est d'empêcher les mises à jour trop brusques des paramètres puisqu'il pénalise celle-ci en fonction du carré du gradient. e a été présenté par (Kingma & Ba, 2014) comme une constante de stabilité numérique, pour éviter la division par 0.

Pour plus d'information sur les différents optimiseurs existants, une revue récente a été réalisée par (Haji & Abdulazeez, 2021).

2.3.1.2. Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN) sont un type particulier de réseaux de neurones particulièrement adaptés aux images. Ils sont très similaires aux réseaux neuronaux complètement connectés abordés plus tôt : ils sont constitués de neurones dont les poids et les biais peuvent être appris par descente de gradient, multiplient un signal d'entrée et y appliquent ou non une non-linéarité. Leur principe est de pouvoir transformer des pixels bruts d'une image en entrée en des scores de classes en sortie (lorsqu'il s'agit d'une tâche de classification).

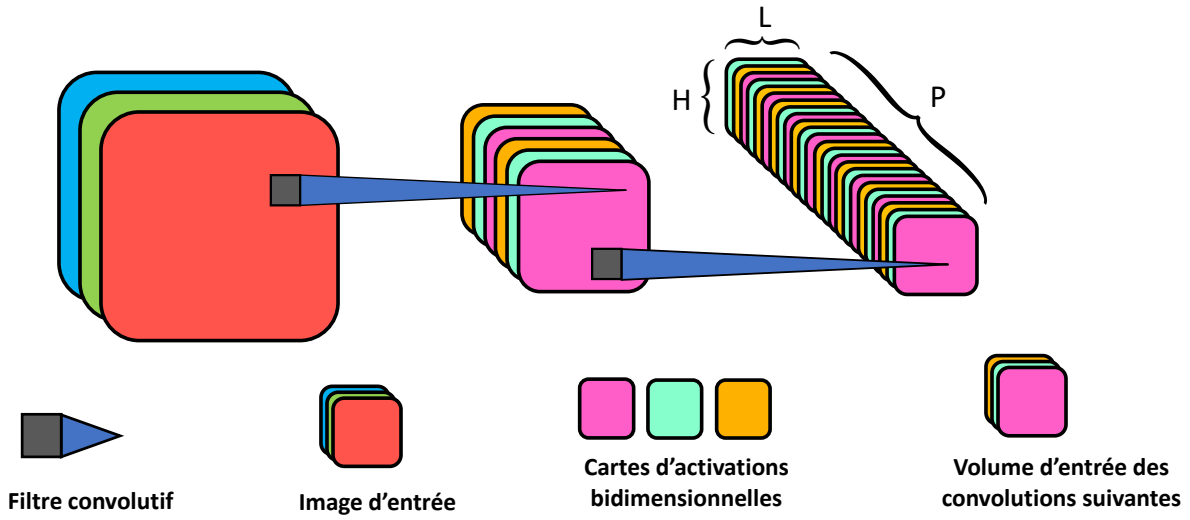


Figure 6 : Illustration d'une succession de deux couches de convolution (CONV). Les filtres de la première couche convolent sur l'ensemble de l'image d'entrée. Ils ont donc ici une profondeur P de 3, chacun. Seul le premier filtre de chaque CONV est illustré. Comme le volume d'entrée de la 2ème couche a une P de 6, cela signifie que 6 filtres au total ont convolé sur l'image d'entrée. En effet, chaque filtre ne produit qu'une seule carte d'activation bidimensionnelle de largeur L et de hauteur qui s'empilent sur une dimension de profondeur P. Cet empilement constitue le volume d'entrée de la CONV suivante. Comme le premier volume d'entrée illustré ici a une P de 6, tous les filtres de la deuxième CONV auront également une profondeur de 6. Encore une fois, leur nombre sera le même que la P du deuxième empilement de cartes d'activation bidimensionnelles, c'est-à-dire 24.

La différence principale avec les réseaux neuronaux complètement connectés est que les CNN possèdent des couches particulières, appelées couches de convolution (CONV). Les paramètres des CONV consistent en un ensemble de filtres apprenables. Chaque filtre est de faible taille en largeur et en hauteur, mais s'étend sur toute la profondeur du volume d'entrée. Par exemple, un filtre typique de la première couche d'un CNN serait de dimension 5x5x3 (c'est-à-dire 5 pixels en largeur et en hauteur, et 3 pour les canaux RGB) et contiendrait donc 75 paramètres. Pendant la passe avant, chaque filtre convolve sur la largeur et la hauteur du volume d'entrée et calcule les produits scalaires entre les paramètres du filtre et les valeurs des pixels correspondant à la position du filtre avant de leur appliquer une fonction non-linéaire telle que celles des équations 17 et 18, par exemple. Lorsque ce filtre convolve sur la largeur et la hauteur du volume d'entrée, il produit une carte d'activation bidimensionnelle contenant les sorties de ce filtre à chaque position spatiale. Intuitivement, le réseau apprend des filtres qui s'activent lorsqu'ils voient un certain type de caractéristique visuelle, comme un bord d'une certaine orientation ou une tâche d'une certaine couleur. Chaque couche CONV contient la plupart du temps plusieurs filtres et chacun d'entre eux produira une carte d'activation bidimensionnelle distincte. Ces cartes sont empilées sur une dimension de profondeur. La Figure 6 illustre les notions de filtres, de volumes d'entrée ainsi que de cartes d'activation bidimensionnelles.

Il existe de multiples architectures de CNN qui diffèrent les unes des autres par leurs caractéristiques, comme le nombre de CONV qu'ils contiennent (i.e. leurs profondeurs), les types de fonctions d'activations qu'ils utilisent, la taille et le nombre de leurs filtres,… Une parmi les plus

célèbres et les plus utilisées s'appelle « Réseaux de neurones résiduels » (Resnet). Elle fut présentée par He et al. (2016) pour répondre à une problématique de l'époque qui était que les CNN perdaient en précision lorsque le nombre de CONV augmentait. Leur travaux ont ainsi permis de pouvoir créer des architectures bien plus profondes et performantes. Il existe plusieurs variétés de Resnet, tels que les Resnet18, Resnet50, Restnet152. Les nombres en suffixe correspondent au nombre de CONV que l'architecture possède. Ce travail sera basé autour d'un Restnet50 (voir section 2.3.2).

2.3.1.3. Les réseaux de neurones complètement convolutifs

Les réseaux de neurones complètement convolutifs (FCNN), pour la première fois présentés par Long et al. (2015), sont un sous ensemble des CNN ne possédant que des CONV. En effet, certains CNN sont rattachés à un NN pour les tâches de classifications, par exemple. Dans cet exemple de classification, les CONV servent d'extracteurs de caractéristiques qui sont ensuite utilisées en entrée du NN pour la tâche finale de classification. Les FCNN sont particulièrement adaptés aux tâches de segmentation d'images (Hesamian et al., 2019 ; Liu et al., 2019 ; Liu et al., 2021). Comme pour les CNN, il en existe plusieurs architectures. Une des plus connues et des plus utilisées est le U-Net, développé par Ronneberger et al. (2015) et qui sera central dans ce travail.

Le U-net se compose d'une partie contractive (ou encodeur) et d'une partie expansive (ou décodeur), ce qui lui donne une architecture en forme de U. Dans sa forme originelle, l'encodeur est un réseau convolutif typique qui consiste en l'application de CONV suivies d'une activation de type ReLU (voir équation 18). Pendant la contraction, l'information spatiale est réduite tandis que l'information sur les caractéristiques est augmentée, via une profondeur de plus en plus importante et par un nombre croissant de cartes d'activation bi-dimensionnelles. Le décodeur combine l'information spatiale et l'information sur les caractéristiques par une séquence de convolutions ascendantes. Cette combinaison implique des opérations de concaténation entre les même étages de l'encodeur et du décodeur (Voir Figure 7).

La littérature montre que n'importe quel CNN peut faire office d'encodeur (Chaurasia & Culurciello, 2017). L'évolution des encodeurs par rapport à la première version de U-net a permis d'utiliser des CNN plus performants. L'utilisation de Resnets comme encodeur de U-net a été réalisée pour la première fois par Zhang et al. (2018). Ils rapportent avoir eu une meilleure performance qu'avec un U-net traditionnel. Ils expliquent cela par le fait que les connexions résiduelles amélioreraient la propagation de l'information. Un autre intérêt majeur d'utiliser une architecture populaire de CNN comme encodeur est qu'il est souvent aisé de retrouver des paramètres pré-entraînés.

Le pré-entraînement d'un CNN est à la base d'un paradigme appelé transfert d'apprentissage (Weiss et al., 2016). Celui-ci consiste en entraîner un CNN sur une tâche étalon, puis à utiliser ces paramètres appris pour raccourcir, voir même améliorer l'apprentissage d'une tâche similaire (Weiss et al., 2016). La tâche étalon la plus célèbre pour les CNN est la classification d'image appliquée à un jeu de données appelé Imagenet (Deng, et al., 2009). Il s'agit d'une base de données de plus de 10 millions d'images annotées, réparties en 1000 classes. Il est depuis lors devenu une des références principales du transfert d'apprentissage. Il est possible de trouver en ligne des

paramètres appris sur celui-ci pour pratiquement toutes les architectures de CNN, et de les télécharger. En effet, vu la taille de ce jeu de données, il est très difficile de réaliser le pré-entraînement soi-même. Il existe néanmoins de nombreuses autres manières de pré-entraîner un CNN comme l'utilisation d'autres jeux de données de référence (Kamilaris & Prenafeta-Boldú, 2018), le clustering d'images similaires (George et al., 2018) et bien d'autres. Zhuang et al. (2020) ont réalisé une revue complète des différentes possibilités. Dans le cadre de ce travail, seul le pré-entraînement sur Imagenet a été étudié.

2.3.2. Description de l'architecture du réseau de neurones complètement convolutif utilisée

Comme mentionné précédemment, l'architecture utilisée est constituée d'un Resnet50 comme encodeur, ainsi que d'un décodeur contenant une couche de suréchantillonnage pour chaque couche de l'encodeur où les dimensions de longueur et de largeur du volume sont divisées par 2 (voir Figure 7). Pour complètement saisir la nature de l'architecture, il convient d'introduire encore les concepts suivants :

- ***Maxpooling*** - Il est courant d'insérer une couche (dite de « *maxpooling* ») dans une architecture CNN. Sa fonction est de réduire progressivement la taille spatiale de la représentation afin de réduire la quantité de paramètres et de calculs dans le réseau, et donc de contrôler également *l'overfitting*. Cette couche de *maxpooling* consiste en un filtre qui va glisser le long des cartes d'activation de façon similaire aux filtres de convolution et appliquer l'opération max() sur les éléments se trouvant à la même position que le filtre (Li, 2022).

- **Normalisation de lot** - Il a été montré que CNN fonctionnent mieux (vitesse, stabilité, robustesse) lorsque les données qu'ils reçoivent sont normalisées (Ioffe & Szegedy, 2015). Puisque chaque CONV d'un CNN peut être considérée comme un mini CNN, Ioffe & Szegedy (2015) ont eu l'idée de renormaliser les entrées des CONV successives, à intervalles réguliers.

- **Suréchantillonnage par mélange de pixel** - Dans l'implémentation de base du U-net, Ronneberger et al., (2015) ont utilisé des convolutions ascendante pour le décodeur. En 2016, Shi et al. (2016) ont mis au point une méthode de super-résolution (c'est-à-dire augmenter artificiellement la résolution d'une image) basée sur le suréchantillonnage par mélange de pixel. La méthode est basée sur une série de convolutions pour laquelle chaque filtre aura pour tâche d'apprendre une direction particulière d'extrapolation. Shi et al. (2016) montrent que leur méthode peut apprendre des transformations plus complexes et plus précises que de simples convolutions ascendantes. En effet, le domaine de la super-résolution est très lié à la partie expansive du U-net puisque celle-ci augmente progressivement la résolution spatiale du signal transmis par la partie contractive. Il n'est donc pas surprenant que le suréchantillonnage par mélange de pixel apporte de bons résultats.

- **Initialisation des paramètres** - Avant l'entrainement, les paramètres des réseaux de neurones se voient attribués une valeur. Cette attribution de valeur s'appelle l'initialisation. Une initialisation non optimale peut compliquer la tâche d'apprentissage (He, Zhang, Ren, & Sun, 2015). He et al. (2015) montrent que leur initialisation améliore les propriétés de convergence des réseaux de neurones utilisant l'activation ReLU. Chaque paramètre sera tiré d'une distribution normale $\sim \mathrm{N}\left(0, \frac{2}{n_w}\right)$ où n_w est le nombre de paramètres dans un filtre. Seul le décodeur est concerné puisque l'encodeur utilise des paramètres pré-entrainés.

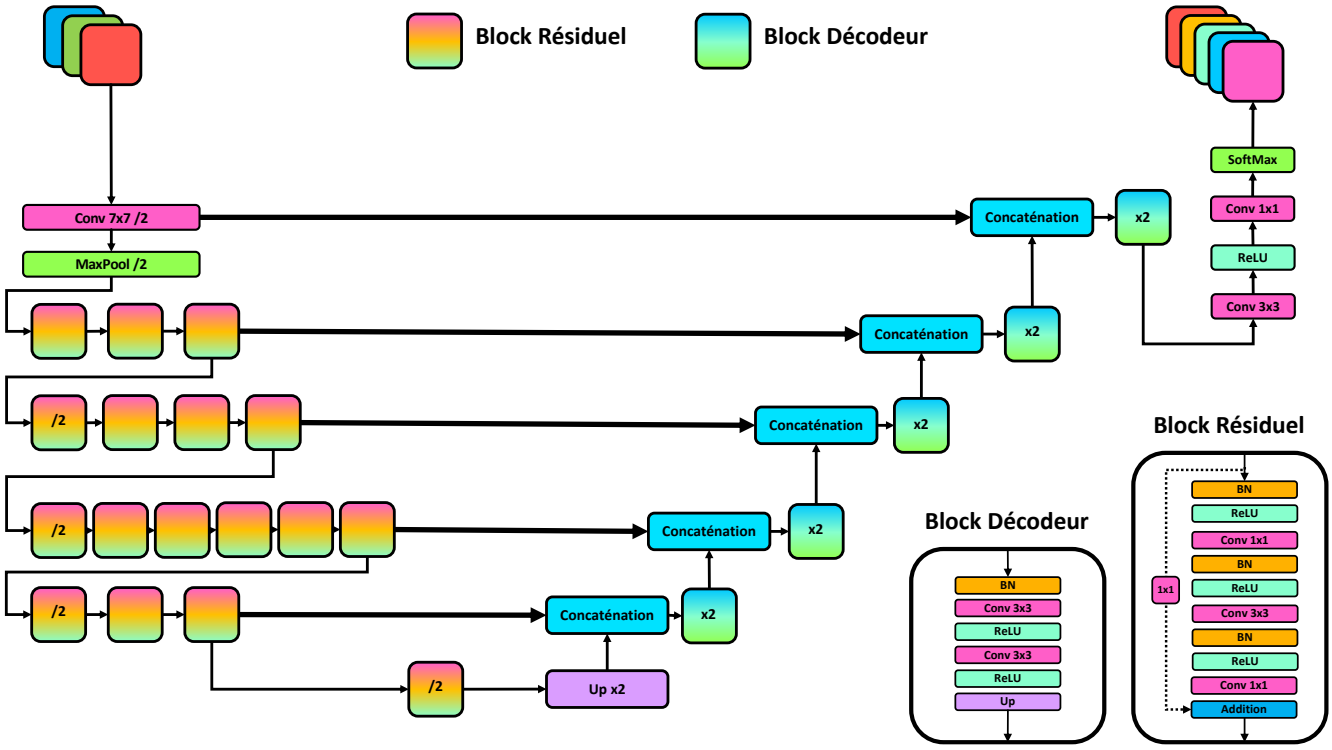


Figure 7 : Illustration de l'architecture de FCNN utilisée lors de ce travail. Les deux bloques de couches principaux utlisés sont résumés sous forme de « block décodeur » et de « block résiduel ». Leurs contenus sont illustrés dans le coin inférieur droit de la figure. Les élément utilisés sont : la Normalisation de lot (BN), des convolutions (Conv), dont les dimensions de hauteur et de largeur sont indiquées respectivement sous la forme (HxW), l'activation ReLU, du *maxpooling* (MaxPool) et le surréchantillonage par mélange de pixel (Up). L'opération de concaténation signifie la superposition des volumes d'entrée sur la dimention de profondeur. L'opération d'addition signifie l'addition élément par élément des volumes d'entrée, sur la dimension de profondeur. Les symboles « /2 » et « x2 » signifie qu'après passage dans cet élément, le volume d'entrée verra ses dimensions de hauteur et de largeur respectivement divisées ou multipliées par 2 (arrondi au nombre entier supérieur).

Tableau 2 : ***Description des caractéristiques et hyperparamètres du FCNN utilisé lors de ce travail.***

Paramètre	***Valeur utilisée***	***Nature du Paramètre***
*Encodeur***	Resnet50	Le type d'encodeur utilisé pour le U-net. Voir sections 2.3.1.2 et 2.3.1.3
*Pré-entraînement de l'encodeur***	Oui, Imagenet	Si l'encodeur a été pré-entraîné et de quelle façon. Voir section 2.3.1.3, dernier paragraphe.
*Initialisation du décodeur**	He	Valeurs des paramètres avant entrainement. Voir section 3.3.2
*Optimiseur***	Adam	Fonction d'optimisation utilisée pour optimiser les paramètres du modèle. Voir section 2.3.1.1, dernier paragraphe.
β_1*	[0.85, 0.95]	Poids pondérant la moyenne mouvante exponentiellement pondérée d'ordre 1 utilisé par l'optimiseur Adam. Valeurs minimales et maximales que prendra le paramètre au cours du planning. Voir sections 2.3.1 et 2.3.3.3
β_2*	[0.85, 0.95]	Poids pondérant la moyenne mouvante exponentiellement pondérée d'ordre 2 utilisé par l'optimiseur Adam. Valeurs minimales et maximales que prendra le paramètre au cours du planning. Voir sections 2.3.1 et 2.3.3.3
ϵ_1*	1e-5	Constante de stabilité numérique utilisée par

		l'optimiseur Adam. Voir section 2.3.1
*Fonction de coût***	Combinaison entre FL et LCDL	Fonction que l'optimiseur cherchera à minimiser par descente de gradient. Voir section 38,
γ	2	Constante d'importance de l'incertitude de la FL. Augmente la pénalité attribuée aux pixels pour lequel le FCNN est incertain. Voir section 38
α*	1	Facteur de pondération de déséquilibre des classes de la FL
*Constante d'apprentissage***	[1.2e-6, 3e-5]	Facteur multiplicatif du gradient conditionnant l'amplitude de la mise à jour des paramètres lors de la descente de gradient. Valeurs minimales et maximales que prendra la constante d'apprentissage au cours du planning. Voir section 2.3.3.3
Taille des mini-lots	2	Le nombre d'échantillons utilisés par itération de descente de gradient. Voit section 2.3.3.1
Taille ajustée des images	500x500	Les dimensions de hauteur et de largeur auxquelles les images ont été redimensionnées. Voir section 2.3.3.1

*Les paramètres suivis de ** ont été optimisés de la même façon que ceux des GBDT (voir section 2.2.3) Les paramètres suivis d'un * sont les valeurs par défaut de la librairie Fastai (voir section 2.4.1).*

2.3.3. Étape 3 : L'entraînement d'un réseau de neurones convolutifs sur les résultats de LeafPainter

Pour entraîner l'architecture présentée dans les sections 2.3.1.3 et 2.3.2 nous avons généré, grâce à la méthode présentée section 2.2.4, 55 masques dont la répartition par classe est visible dans le Tableau 3. On peut y voir que les classes sont réparties de façon homogène, hormis la classe Fond (ou fond) qui est dominante, ce qui est attendu dans un problème de segmentation.

Le point important de cette méthode est que nous utilisons uniquement des photos de cultures pures pour lesquelles l'espèce cultivée est connue. Cette connaissance de l'espèce permet de convertir les masques binaires en masques multi-classes ne contenant qu'une espèce, et donc d'entraîner le modèle à une tâche de segmentation multi-classes.

2.3.3.1. Redimensionnement et mini-lots

Un des facteurs importants à prendre en compte lors de modélisation par réseaux de neurones est leur consommation de mémoire vive. Il faut tenir compte de cette limitation lors de la conception du mode d'entraînement. Pour cela on peut jouer sur 2 paramètres :

- **Les dimensions des images**, où le nombre de pixels contenu dans une image. On l'exprime sous la forme d'un produit entre le nombre de lignes et de colonnes (hauteur x largeur). Évidemment, une image plus petite occupe moins d'espace mémoire. Nous avons donc choisi la taille maximale, telle que l'information contenue soit toujours suffisante pour être facilement analysée par un œil humain : 500x500, sachant que la taille originelle des images était de 1600x1200.

- **La taille des mini-lots**, c'est-à-dire le nombre d'échantillons (d'images dans notre cas) utilisés par itération de descente de gradient. Cela va jouer sur la vitesse du processus d'entraînement puisque les librairies d'apprentissage profond permettent d'effectuer toutes les opérations de façon parallèle sur un processeur graphique (voir section 2.4.2). Cela va aussi jouer sur la vitesse d'apprentissage en diminuant la variance des gradients de la fonction de coût quand on augmente la taille des mini-lots (Qian & Klabjan, 2020). Nous avons donc choisi la taille maximale des mini-lots qui pouvait être traitée par le processeur graphique pour cette taille d'image : 2.

La réduction de taille des images présente un autre intérêt. Puisque les essais ont été réalisés sur plusieurs espèces en même temps, il est fréquent de voir des espèces d'une parcelle déborder sur une autre (voir Figure 8), ce qui fait que des espèces différentes peuvent parfois être présentes sur la même image de culture pure. Ces intrusions créent un problème qui est que, pour de la conversion de masques binaires en masques multi-classes, nous posons l'hypothèse que tous les pixels labélisés comme plante par LeafPainter appartiennent à la même espèce. Ensuite, nous convertissons ces

masques binaires en masques multi-classes grâce au label de la culture pure, pour ensuite entrainer entraîner le FCNN à prédire cette espèce. Les pixels appartenant à une intrusion seront dont mal labélisés. Réduire ce phénomène est donc primordial pour ne pas introduire d'erreur dans son apprentissage.

Le fait de réduire les dimensions des images en rognant les bords permet de fortement limiter ce phénomène.

Figure 8 : Exemple d'intrusion de haricot (entouré en rouge) sur une image de culture pure de choux cabus. Garder cette image telle quelle pour l'entraînement du FCNN aurait eu pour conséquence que le modèle apprenne à identifier le haricot comme choux cabus.

Tableau 3 : ***Répartition des classes du jeu de données du CNN***

Espèce	*Nombre d'images*	*Pourcentage de pixels total*
Fond	55	45
Choux Cabu	18	13

Choux frisé	11	13
Fenouil	10	14
Haricot Nain	12	15

2.3.3.2. La fonction de coût

L'apprentissage des paramètres optimaux par un FCNN implique la minimisation d'une fonction de coût. Pour les tâches de classification, y compris la segmentation sémantique, la fonction de coût la plus populaire a pendant longtemps été l'entropie croisée (CE):

$$CE(y_n, p_n) = -\sum_{c=1}^{C} log(p_{n,c})\, y_{n,c} \qquad \textbf{Eq. 22}$$

où y_n est le vecteur de classe du pixel n d'un échantillon. Il contient $\boldsymbol{C}$ éléments. $\boldsymbol{C}$ est le nombre total de classes et $\boldsymbol{c}$ une classe particulière. Par exemple, si un pixel appartient à la classe 3 et qu'il y a 4 classes au total, la valeur de y_n sera de [0,0,1,0]. p_n est la prédiction du FCNN, de même dimension. Chaque élément de p_n est la probabilité prédite que ce pixel soit de chaque classe. $y_{n,c}$ et $p_{n,c}$ sont respectivement les probabilités observées et prédites que le pixel appartienne à la classe $\boldsymbol{c}$.
La CE a le désavantage qu'elle favorise l'apprentissage des classes les plus représentées puisqu'il ne s'agit que d'une somme sur les différentes classes prédites, sans pondération (Jadon, 2020).

Une alternative tenant compte de ce problème est la fonction de coût de Dice (DL), proposée pour l'apprentissage profond par Sudre et al. (2017). Ils montrent qu'en cas de déséquilibre de classes, la DL est au moins équivalente en termes de performance à la CE et supérieure quand le déséquilibre est grand, même lorsque cette dernière est pondérée pour tenir compte de ce déséquilibre.

$$DL(y_n, p_n) = 1 - 2\,\frac{\sum_{c=1}^{C} p_{n,c} y_{n,c}}{\sum_{c=1}^{C} p_{n,c} + y_{n,c}} \qquad \textbf{Eq. 23}$$

Le problème de la DL est qu'elle a tendance à être numériquement instable (Jadon, 2020). Ceci s'est également manifesté dans ce travail. Pour résoudre ce problème, Jadon (2020) a proposé d'utiliser le Logarithme du cosinus hyperbolique de la DL (LCDL) comme nouvelle fonction de coût :

$$LCDL(y_n, p_n) = \log\left(\cosh\left(DL(y_n, p_n)\right)\right) \qquad \text{Eq. 24}$$

Les résultats de ce travail sont cohérents avec Jadon (2020) puisque l'utilisation de la LCDL a bien résolu les problèmes de stabilité rencontré lors de l'utilisation de la DL.

Une autre fonction de coût célèbre pour la segmentation d'image est la fonction de coût focale (FL) (Lin et al., 2017). La FL peut être considérée comme une variation de la CE. Elle diminue la contribution des exemples que le FCNN classifie bien avec une certitude élevée et permet au modèle de se concentrer davantage sur l'apprentissage des exemples difficiles. Elle fonctionne bien pour les scénarios de classes fortement déséquilibrées.

$$FL(y_n, p_n) = \alpha(1 - p_{*n})^{\gamma} CE(y_n, p_n) \qquad \text{Eq. 25}$$

Où α et γ sont des hyperparamètres. Plus γ est grand, plus le FCNN attribuera d'importance aux pixels pour lesquels il est incertain. Lin et al. (2017) rapportent qu'une valeur de 2 mène à de meilleurs résultats. p_{*n} est la probabilité associée à la classe prédite pour la classe réelle. α est un facteur de pondération de déséquilibre des classes. Il sera laissé à 1 dans ce travail.

Enfin, Wang et al. (2020) ont montré que la FL est particulièrement performante lorsque le jeu de données contient beaucoup de petits objets. La DL, elle, est particulièrement efficace quand le fond prend beaucoup de place dans le jeu de données puisqu'elle se concentre sur la région de classe positive (les pixels Plantes dans notre cas). La FL et la DL ont donc des avantages différents. Ces auteurs montrent que combiner la DL et la FL augmente la performance sur leur jeu de données de façon considérable.

Cette combinaison de fonction de coût est également utilisée dans ce travail et est dénommée fonction de coût combinée (CL). La seule différence par rapport à (Wang et al., 2020) est que, dans ce cas, c'est la LCDL qui est utilisée au lieu de la DL pour sa stabilité supérieure, comme évoqué plut tôt.

$$CL(y_n, p_n) = FL(y_n, p_n) + LCDL(y_n, p_n) \qquad \text{Eq. 26}$$

2.3.3.3. Les planificateurs de paramètres

Smith & Topin (2019) ont développé l'idée que les paramètres initiaux ne sont généralement pas adaptés à la tâche que l'on essaye de résoudre. Par conséquent, il est dangereux de commencer l'entraînement avec une constante d'apprentissage élevée. Ils ont conçu un planificateur de constante d'apprentissage séparé en 2 phases linéaires : une où la constante d'apprentissage croît de la valeur minimale à la valeur maximale (échauffement), et une où elle décroît pour revenir à la valeur minimale. Smith & Topin (2019) ont appelé cette combinaison d'approches "apprentissage en un cycle".

Ils montrent que ce planificateur permet d'utiliser une constante d'apprentissage maximale beaucoup plus élevé que les autres types de formation, ce qui présente 2 avantages :

- Entraîner un modèle avec des constantes d'apprentissage plus élevées lui permet d'apprendre plus rapidement. Smith & Topin (2019) nomment ce phénomène la « super-convergence ».
- Entraîner un modèle avec des constantes d'apprentissage plus élevées lui permet de moins sur-ajuster (*overfitting*). En effet, cela permet à la descente de gradient d'éviter plus de minima locaux.

L'explication intuitive donnée par Smith & Topin (2019) est qu'un modèle qui généralise bien est un modèle dont le gradient de la fonction de coût est faible. Si un modèle apprend à un taux d'apprentissage élevé pendant un certain temps avec de bons résultats, c'est qu'il se trouve dans une zone de la fonction de coût de topographie homogène et plate (donc de faible gradient).

Au cours de l'apprentissage en un 1 cycle, la constante d'apprentissage va donc progressivement croître vers sa valeur maximale dans le but de trouver cette région de faible gradient de coût. Puis, la constante va diminuer progressivement afin de permettre à l'optimiseur de trouver, au sein de cette zone de faible gradient, le point minimum.

Enfin, Smith (2018) a montré que l'apprentissage en un cycle doit s'accompagner d'une dynamique inverse pour les poids pondérant la moyenne mouvante exponentiellement pondérée des optimiseurs utilisant l'inertie du gradient. Plus simplement, quand la constante d'apprentissage croit, ces poids décroissent et inversement, toujours de façon linéaire.

2.3.3.4. Augmentation des données

Il est courant, dans l'entrainement de réseaux de neurones convolutif d'utiliser un concept appelé « l'augmentation de données ». Ce concept consiste en l'application de transformation aux images afin d'augmenter artificiellement la richesse et la diversité du jeu de données à moindre cout. Les augmentations utilisée dans ce travail sont illustrées Figure 9. L'augmentation de données doit être utilisée de façon réfléchie. Elle ne doit pas, par exemple, mener à des formes ne représentant plus les feuilles de façon correctes. Par exemple, changer leurs formes ou leur couleurs de façon non réaliste peut dégrader la performance du modèle sur des images non augmentées.

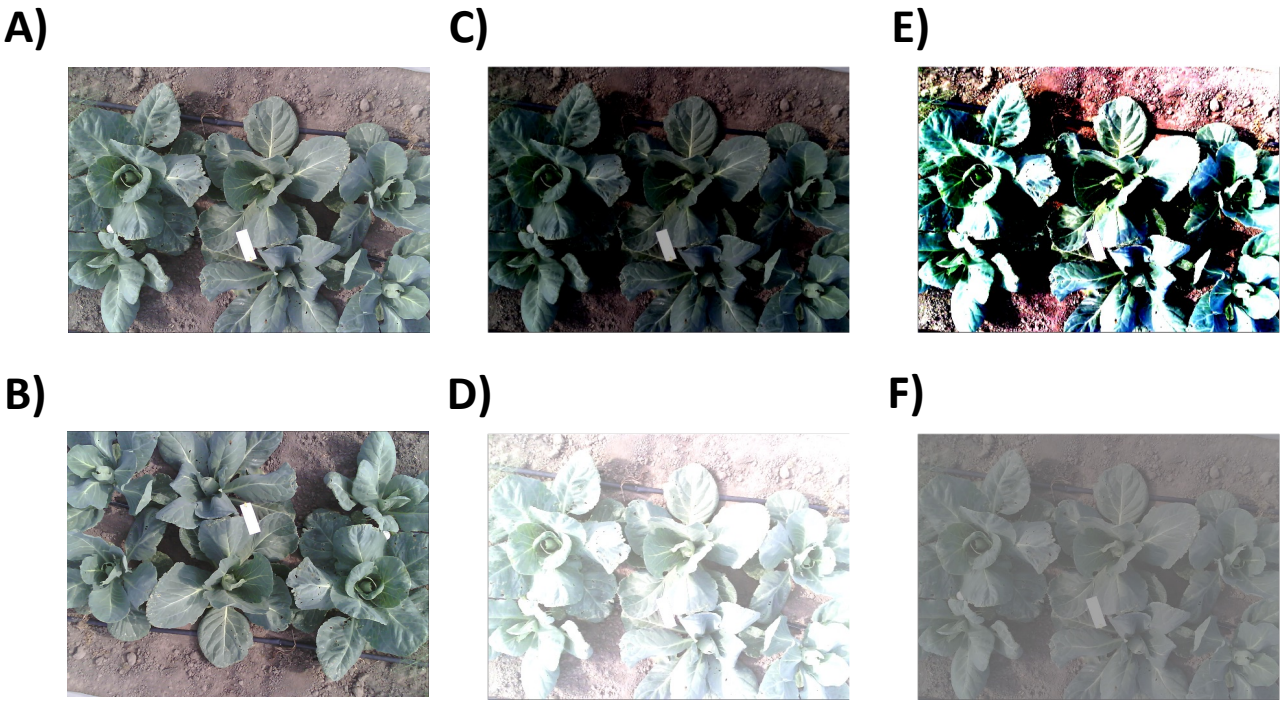


Figure 9 : Types d'augmentation de données utilisés. A) Image non transformée B) Rotation selon un angle aléatoire (ici 180°). C) et D) Respectivement baisse et augmentation de la luminosité. E) et F) Respectivement augmentation et baisse du contraste.

2.3.1. Étape 4 : Inférence sur des images de cultures associées

Cette dernière étape consiste simplement à utiliser le FCNN entraîné à l'étape 3 pour inférer sur des images de cultures associées.

2.4. Hardware et Software

Ce travail a nécessité un recours important à des ressources soft- et hardware qui sont décrites ci-dessous.

2.4.1. Software

Tous les algorithmes ont été implémentés dans le langage de programmation Python v3.8 (Van Rossum & Drake Jr, 1994). Les modèles d'apprentissage profond ont été implémentés en pyTorch v1.8.1 (Paszke, et al., 2019). C'est une des librairies en Python les plus célèbres et les plus recommandées pour implémenter ce type d'algorithme. Enfin, la librairie Fastai v2.5.3 (Howard & Gugger, 2020) a été utilisée pour faciliter l'implémentation de l'architecture. C'est une collection d'abstractions de haut niveau basées sur la librairie pyTorch.

2.4.2. Hardware

Pour entraîner l'architecture, nous avons utilisé le centre de Calcul Intensif et Stockage de Masse (CISM) de l'Uclouvain. Le CISM fournit l'accès à plusieurs noyaux d'unités de calculs. Parmi ces noyaux, nous avons utilisé le noyau Manneback, sur recommandation d'expert. Le processeur graphique (GPU) utilisé est un A100-PCIE 40 Gigabytes de mémoire, de la marque Nvidia. Plus d'information sont disponibles sur la fiche technique : https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/A100-PCIE-Prduct-Brief.pdf

3. Resultats

Ce chapitre aborde de façon pragmatique les différents des différentes étapes du travail : la segmentation binaire par apprentissage actif par LeafPainter, la segmentation multi-classe du FCNN entrainé sur les masques de cultures pures fournis par LeafPainter et, finalement l'inférence du FCNN entrainé sur des images d'associations de cultures.

Tableau 4 : ***Score obtenus pour les différentes étapes***

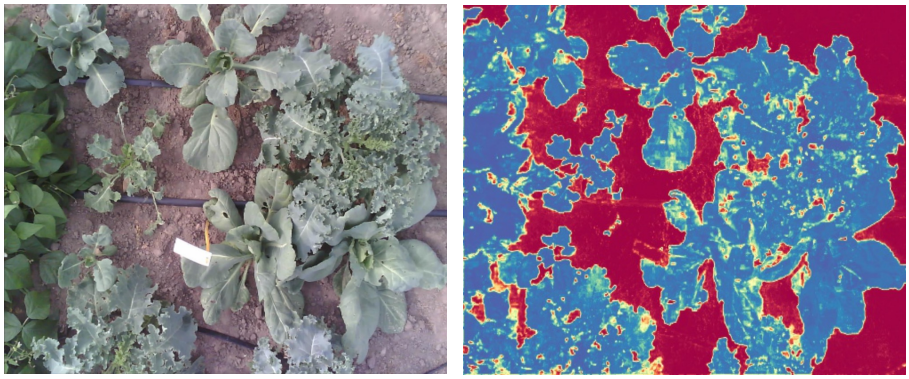
Étape	*Précision*
Segmentation Binaire	84.8 %
Segmentation Multiclasses sur cultures pures FCNN	92.16 %

3.1. Segmentation Binaire LeafPainter

Pour rappel, la deuxième étape de la méthode (section 2.2.4), consiste à utiliser LeafPainter pour générer des masques de segmentation binaire par apprentissage actif. La précision obtenue après application du protocole (section 2.2.3) est de 84.8 %. Il est utile de préciser que le terme « précision » utilisé ici signifie que nous avons fait le rapport entre les pixels bien classifiés et le nombre total de pixels. Cette métrique peut s'avérer biaisée en cas de déséquilibre de classes mais,

comme décrit section 2.2.3, ce jeu de donnée est équilibré et la précision est donc une métrique bien représentative et facile interpréter.

A)



B)

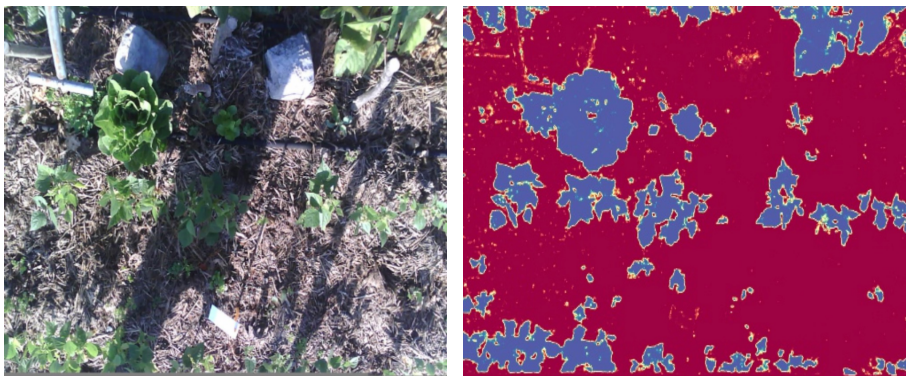


Figure 10 : Exemple de 2 images particulièrement difficiles du test de validation et leur segmentation par inférence de LeafPainter. La couleur symbolise la probabilité d'appartenir à la classe « Plante ». Bleu foncé = 1, Jaune clair = 0.5 et Rouge foncé = 0. A) La principale difficulté de cette image est qu'elle contient 3 espèces de plantes différentes. La segmentation est assez bonne, malgré quelques zones d'incertitude. B) Image difficile par la présence d'ombres, d'outils et de rochers. Ces derniers amènent un peu d'incertitude mais le résultat est bon, malgré tout.

L'intérêt principal de LeafPainter est la réduction du travail nécessaire à l'obtention de masques. Pour quantifier le temps de ce travail, nous allons parler du temps de latence, c'est-à-dire le temps de réentrainement additionné du temps d'inférence du modèle, et du temps d'annotation qui est la somme des temps de sélection des zones de l'image et de latence. Le temps d'annotation dépend de deux facteurs :

- La complexité de l'image, qui est d'autant plus importante que l'image contiendra des zone sombres, éléments étrangers (outils, etc.), ou encore des espèces différentes. Plus celle-ci est grande, plus une image prend d'itérations à annoter, pour un même stade d'entrainement des GBDT.
- Le stade d'entrainement du GBDT. Plus les GBDT auront été entrainés sur un grand nombre de pixels, moins il y aura de corrections à faire, mais plus le temps de latence sera grand.

Le score de 84,8% de précision a été obtenu en annotant 10 millions de pixels. Le temps de latence à ce niveau était d'environ 2 minutes. Lorsqu'on débute une nouvelle phase, avec un jeu de données vierge, le temps de latence est d'environ 3 secondes. Ceci fait donc un temps moyen de latence d'environ une minute, par nouvelle correction, pour la tâche complète. Ceci dépendant évidemment du matériel informatique de l'utilisateur.

En ajoutant à cela le temps moyen de sélection des zones, de l'ordre de 20 secondes, en fonction de la complexité des images, nous arrivons à un temps d'annotation moyen de 2 minutes et 30 secondes par image, en sachant que quand 10 millions de pixels ont été annotés, l'inférence a pu être réalisée sur tout le reste des images, sans travail supplémentaire.

3.2. Segmentation Multi-Classes

3.2.1. Cultures pures

L'étape 3, décrite section 2.3.3 consiste à utiliser des masques générés par LeafPainter pour entrainer un FCNN à reconnaitre l'espèce présente sur les images de cultures pures. Comme métrique d'évaluation, pour le choix du modèle, nous avons utilisé la « précision d'avant-plan », c'est-à-dire la précision calculée uniquement sur les pixels appartenant à une des classes de plante et pas sur les pixels de fond. Cela permet d'utiliser une métrique facile à interpréter, sans le biais induit par le fait que la majorité des pixels soient du fond.

La précision d'avant plan ainsi obtenue sur le set de validation est de 92.16 %.

Il faut toutefois se méfier de ce nombre qui mesure à quel point les prédictions du modèle sont similaires aux masques fournis par LeafPainter. Ces derniers n'étant pas parfaits, la meilleure précision possible n'est pas de 100% puisque cela reviendrait à reproduire les erreurs de LeafPainter. De plus, nous avons observé que le modèle est capable de fournir des prédictions de qualité supérieure aux masques sur lesquels il a été entraîné. La Figure 11 illustre bien ce phénomène : le FCNN est parvenu à corriger le masque d'entraînement. Bien que la prédiction du FCNN soit différente du masque, et tire donc la précision vers le bas, cette prédiction est pratiquement parfaite.

Un autre exemple d'évaluation quantitative difficile se présente lorsqu'il reste des inclusions de parcelles voisines au sein d'une image de culture pure. En effet, même si le FCNN identifie correctement l'espèce intrue, le masque fourni reste un masque de culture pure et va considérer l'intru comme faisant partie de l'espèce pure en question. Donc si le FCNN identifie correctement l'intru, la métrique d'évaluation va être sous-évaluée, tandis que s'il l'identifie comme faisant partie de la même espèce que la culture pure, la métrique va être surévaluée. Cet exemple est illustré sur la Figure 12

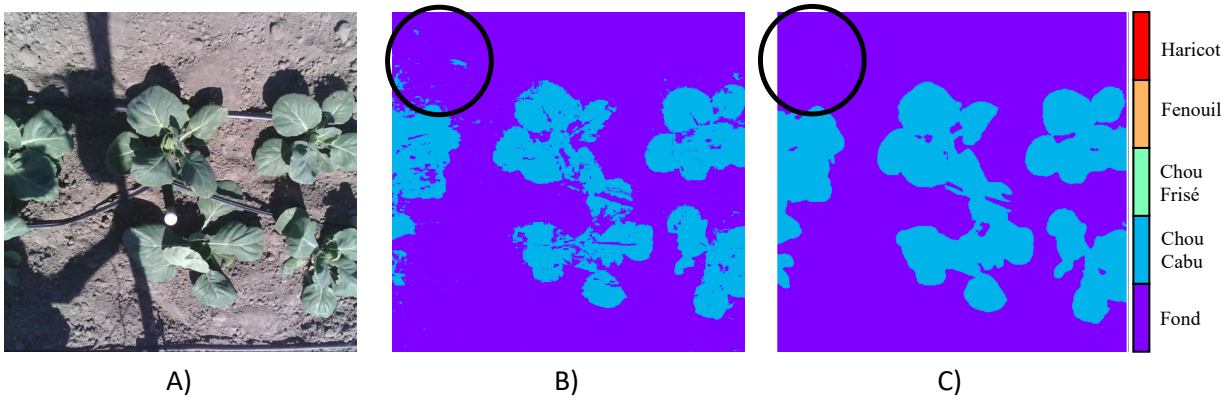


Figure 11 : Image du set de validation d'une culture pure de choux cabus. A) Photo de la parcelle. B) Segmentation binaire réalisée par LeafPainter. On voit que certaines zones sont segmentées de manière éparse. Particulièrement dans les zones d'ombres, comme dans le cercle noir. C) Prédiction par le FCNN. On voit que les zones segmentées sont des zones compacte, ce qui permet de s'affranchir des zones de segmentations éparses. La prédiction est ici visuellement de meilleure qualité que le masque, ce qui implique que l'interprétation de la métrique d'évaluation sera biaisée.

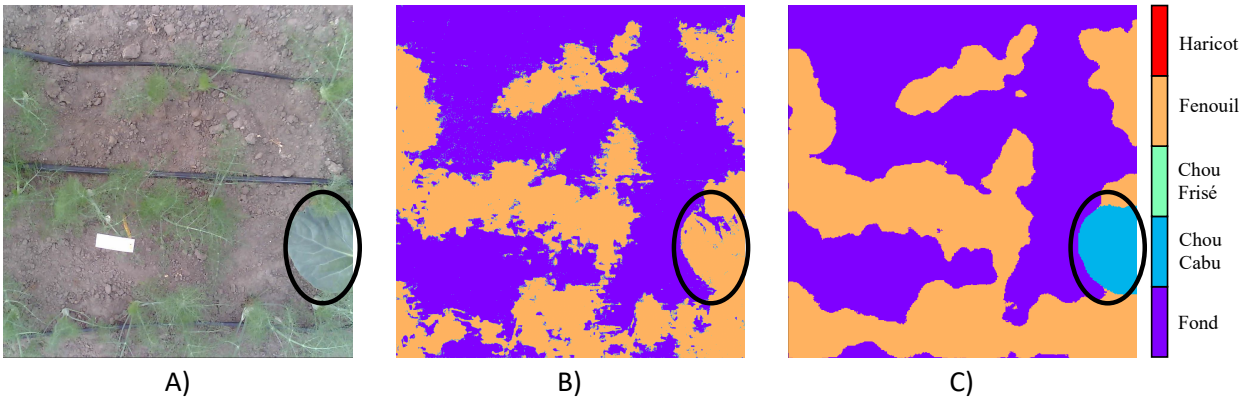


Figure 12 : Culture pure de fenouil avec intrusion de chou cabu. A) Image catégorisée comme culture pure de Fenouil avec une intrusion de chou cabu dans le cercle noir. B) Labélisation par LeafPainter. Comme attendu, l'intrusion de chou cabu est labélisée comme fenouil. C) L'intrusion est correctement reconnue comme chou cabu par le FCNN. Cela a pour conséquence de baisser la métrique d'évaluation, à tort.

Il est également intéressant d'analyser les types d'erreurs ayant été commises par le FCNN. Pour cela nous avons réalisé une matrice de confusion (voir Figure 13). Celle-ci nous indique que la plupart des erreurs proviennent de la classe du fond. Comme illustré sur la Figure 11, ceci peut s'expliquer par le fait que LeafPainter fournit des résultats contenant une segmentation éparse avec beaucoup de pixels de sols isolés labelisés comme plante. Comme mentionné plus tôt, la bonne identification de ces pixels comme pixels de fond par le FCNN a pour conséquence de faire baisser la précision pour cette classe de façon erronée. On constate aussi sur la Figure 13 qu'il ne semble

pas y avoir de confusion particulière entre espèce. La confusion la plus forte est entre les haricots nains et les choux cabus, de l'ordre de 2%.

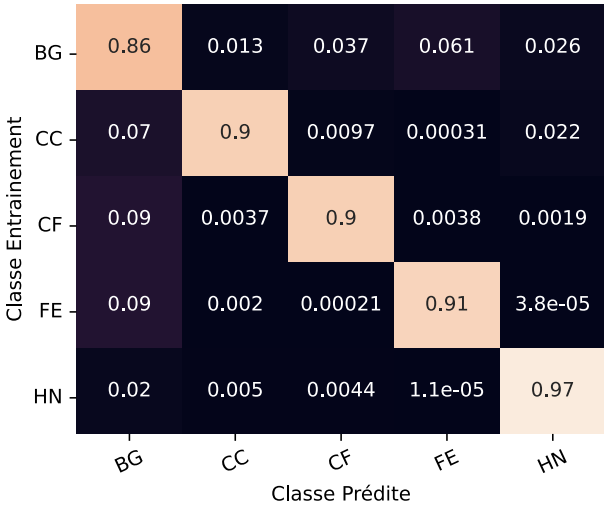


Figure 13 : Matrice de confusion du set de validation du FCNN. Les valeurs sont le pourcentage de pixels totaux du jeu de données de validation qui ont été prédits à une intersection donnée.

3.2.2. Cultures associées

L'étape 4 consiste à finalement utiliser le FCNN entraîné pour segmenter des images de cultures associées. Elle constitue l'étape final de ce travail.

Comme montré sur la Figure 14, la segmentation est bonne dans les différentes combinaisons d'associations, même lorsqu'on observe un grand enchevêtrement des plantes. Ceci est un résultat très positif car on aurait pu craindre, vu l'entraînement en culture pure, que le modèle ne soit capable de reconnaitre une espèce que lorsqu'elle est majoritaire sur la photo.

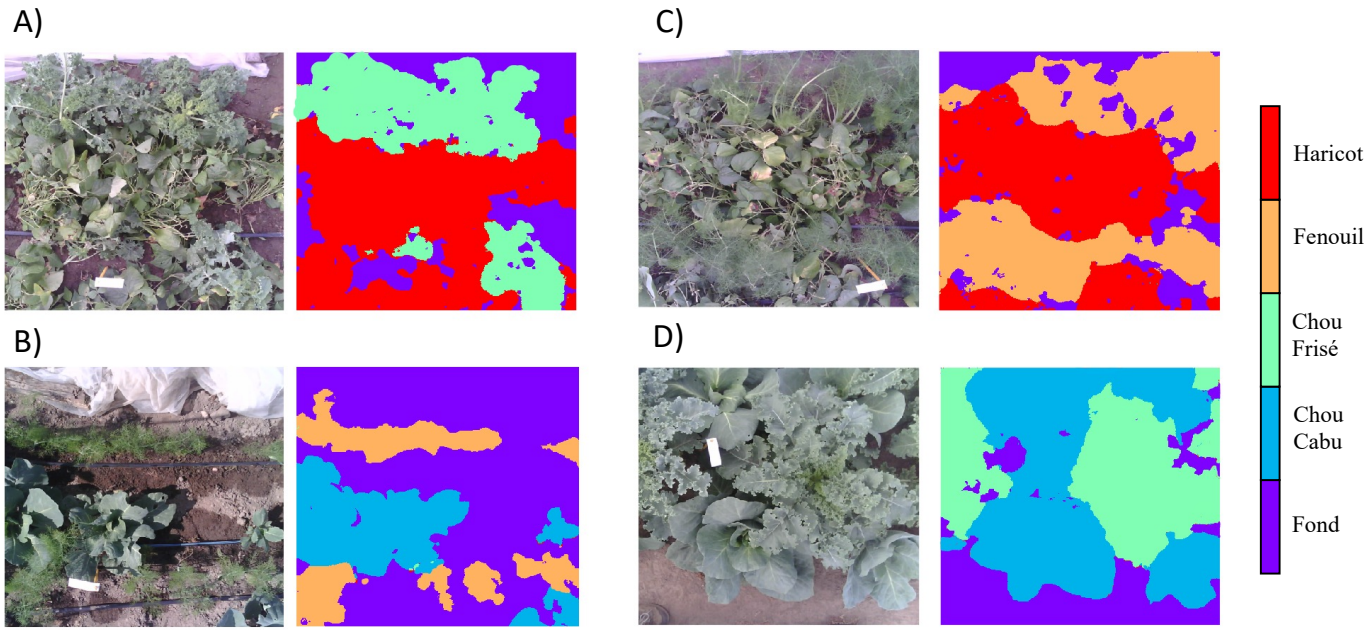


Figure 14 : Exemple de différentes combinaisons d'associations de cultures. A) Association de fenouil et de haricot. La segmentation est bien réalisée malgré le fait que les plantes soient très enchevêtrées. B) Association de fenouil et de chou cabus. Hormis quelques erreurs mineures, la segmentation est bien réalisée. C) Association de haricot et de fenouil. La segmentation est bonne malgré l'enchevêtrement des plantes. D) Association de chou cabu et de chou frisé. Encore une fois la segmentation est bonne malgré un grand enchevêtrement entre les plantes.

3.2.2.1. Analyse de l'erreur

Encore une fois, le point faible de la méthode pour cette étape est la difficulté d'en avoir une évaluation quantitative précise. En effet, l'intérêt est d'utiliser des données plus faciles à annoter pour inférer sur des données pour lesquelles cette annotation serait beaucoup trop laborieuse. La conséquence est que nous n'avons donc pas de point de comparaison parfait. Toutefois il est possible d'analyser qualitativement les erreurs faites et d'en tirer des généralités.

Tout d'abord, nous pensons que les erreurs sur les images de canopées associées apparaissent dans les mêmes circonstances que celles qui arrivent en cultures pures. Par exemple, la Figure 15 illustre bien le fait que lorsque les feuilles de choux cabus sont déchirées ou trouées, le modèle à tendance à confondre leurs extrémités par du haricot, que ce soit en culture pure, ou en association. Un autre exemple, illustré Figure 16, est la difficulté particulière qu'a le modèle à reconnaitre les haricots lorsqu'ils sont ombragés. Encore une fois cette difficulté est présente tant en associations de cultures qu'en culture pure, sans différence entre les deux. Il est également utile de remarquer que dans les deux exemples cités la confusion survient entre choux cabus et haricot, ce qui est cohérent avec l'analyse de l'erreur faite par la matrice de confusion à la Figure 13. Celle-ci indiquait bien que la majorité de la confusion entre espèces survenait entre choux cabus et haricots.

Le point principal de ces exemples et du paragraphe précédent est que même s'il est très difficile d'évaluer quantitativement la performance de la segmentation en culture associée, celle-ci semble visuellement très corrélée à la performance de la segmentation en culture pure. Le modèle commet

des erreurs similaires sur les deux jeux de données et performe bien dans les cas similaires. Cela permet de poser l'hypothèse que la performance de la segmentation en culture pure, qui elle est quantifiable, est un bon proxy pour quantifier la performance de la segmentation en culture associée.

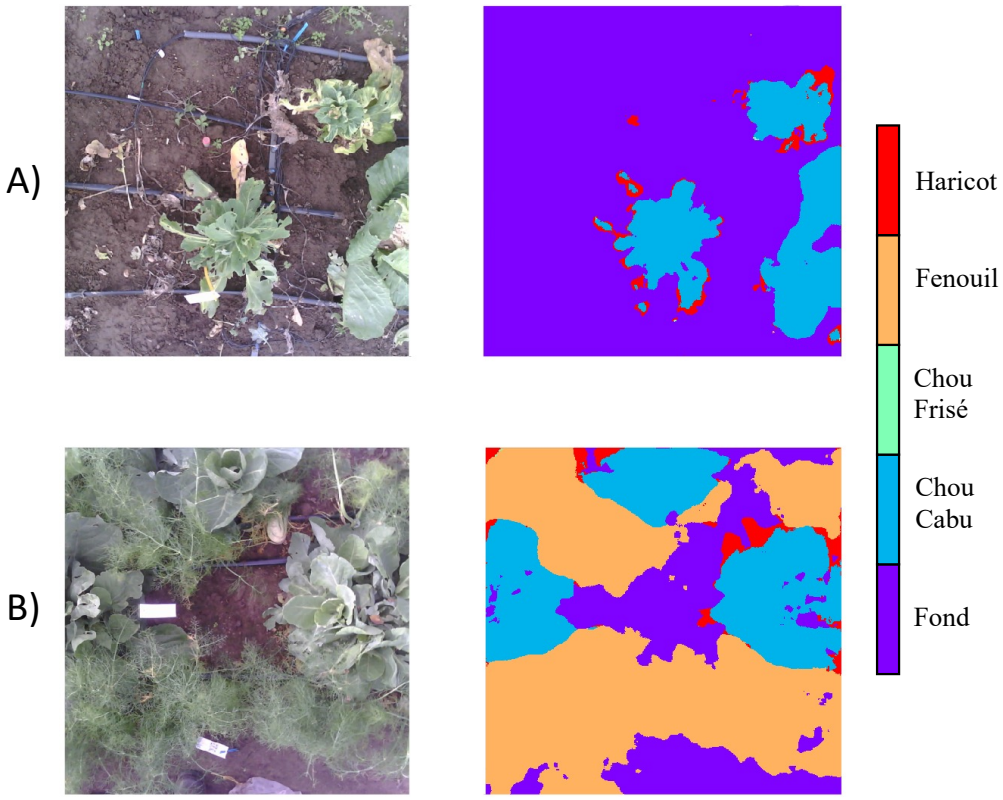


Figure 15 : Comparaison d'erreur du FCNN entre cultures pures et associées pour le chou cabu. A) Culture pure de choux cabu, on remarque que quand les feuilles sont déchirées/froissées, le FCNN a tendance à confondre leurs extrémités avec du haricot. B) Association de cultures de choux cabu / fenouil. On remarque la même chose : les feuilles abimées sont confondues avec du haricot.

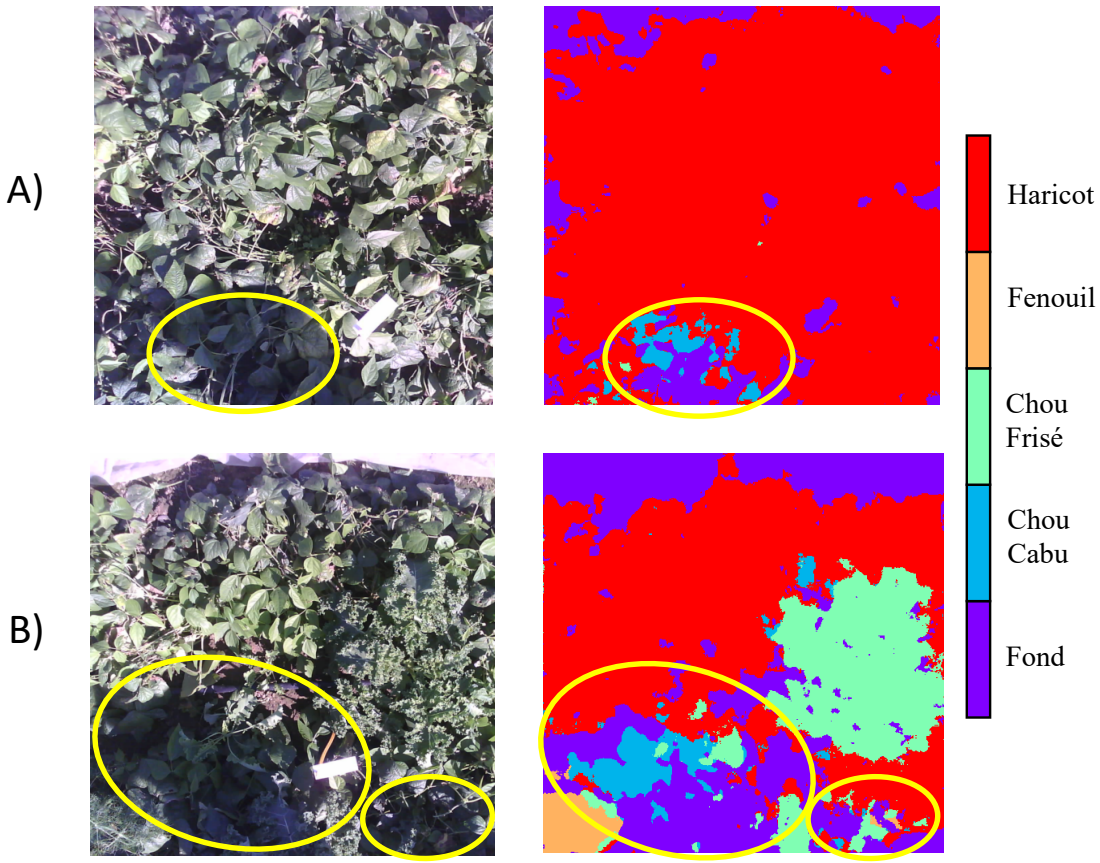


Figure 16 : Comparaison de l'erreur du FCNN associée aux haricots ombragés, en culture pure et associée. A) Culture pure : l'ombre crée une confusion entre haricots, choux cabus et fond. B) Culture associée de haricots et de choux frisés : l'ombre crée une confusion et le haricot n'est plus bien identifié non plus.

Il existe cependant un type d'erreur inhérent à la culture associée. Le fenouil a pour caractéristique particulière d'avoir des feuilles extrêmement fines. Ceci lui donne une apparence légèrement transparente. Cela implique que lorsqu'il est en culture associée, il arrive fréquemment qu'il soit au-dessus d'une autre espèce et qu'on puisse voir une autre espèce au travers. Cela fait émerger de nouveaux types de textures, par exemple, comme sur la Figure 17, où voir une feuille de haricot au travers d'un fenouil lui donne un aspect plus granuleux, ce que le modèle peut confondre avec du chou frisé.

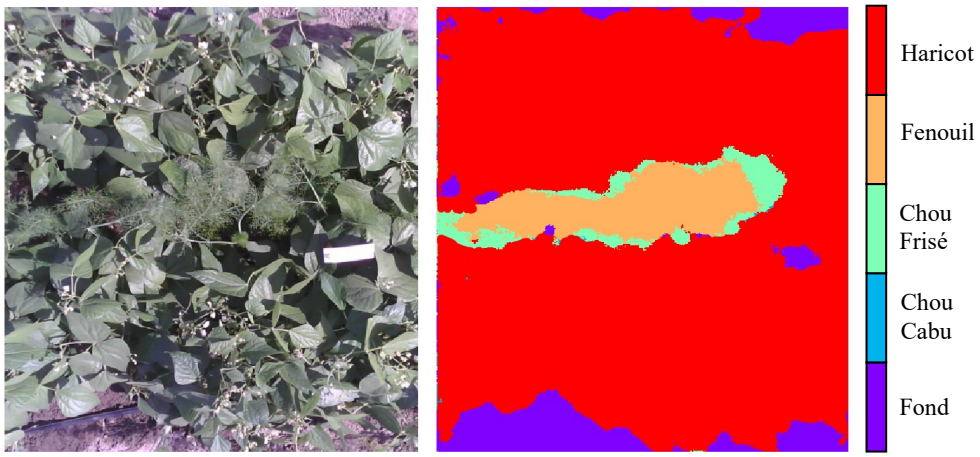


Figure 17 : Image d'une culture associée de haricot et de fenouil. On y voit, sur le contour de la zone de fenouil, là où celui-ci est le plus fin, une confusion claire entre le haricot vu au travers du fenouil et le chou frisé. Ceci est probablement dû à la texture granuleuse que le fenouil donne aux feuilles de haricot vues à son travers.

Enfin, l'autre grand type d'erreur inhérent à l'association de culture est un petit peu plus subtil. Pour la comprendre il faut se souvenir que le FCNN utilisé pour l'inférence a été entrainé sur des images ne contenant qu'une espèce à la fois. Or, une des caractéristiques dont se servent les FCNN pour classifier les pixels est leur contexte. Il est donc possible et attendu que le FCNN aie appris à prédire de façon différente la classe d'un pixel en fonction des classes des autres pixels déjà présents. Comme il a été habitué à ne voir qu'une classe par image, il est possible que le FCNN aie construit une trop grande corrélation spatiale amenant parfois trop de prédictions d'une même classe. Par exemple si le FCNN voit du haricot dont il est certain de la classe, il est probable que les prédictions pour les pixels incertains soient tirées vers la classe haricot. Ce cas est illustré sur la Figure 18 où une grosse confusion est observée entre haricots et choux cabus, sur une image où il y a dominance de choux cabus. En prenant une image similaire où il y a cette fois dominance de choux frisé, ce type de confusion disparait. On l'a déjà vu précédemment, la confusion entre haricots et choux cabus est la plus grande. Il ne serait donc pas étonnant qu'elle soit exacerbée par ce phénomène.

En effet, Les images de la Figure 18 sont très similaires entre elles, la différence entre-elles étant la présence de choux cabus. Ayant connaissance des possibles difficulté que peut éprouver le FCNN à différencier les haricots des choux cabus, on peut raisonnablement croire que la prédiction de l'espèce d'un pixel peut dépendre des autres espèces en présence. Ceci peut être un indicateur d'un biais créé par le fait que le réseau n'ait été entrainé que sur des images de cultures pures.

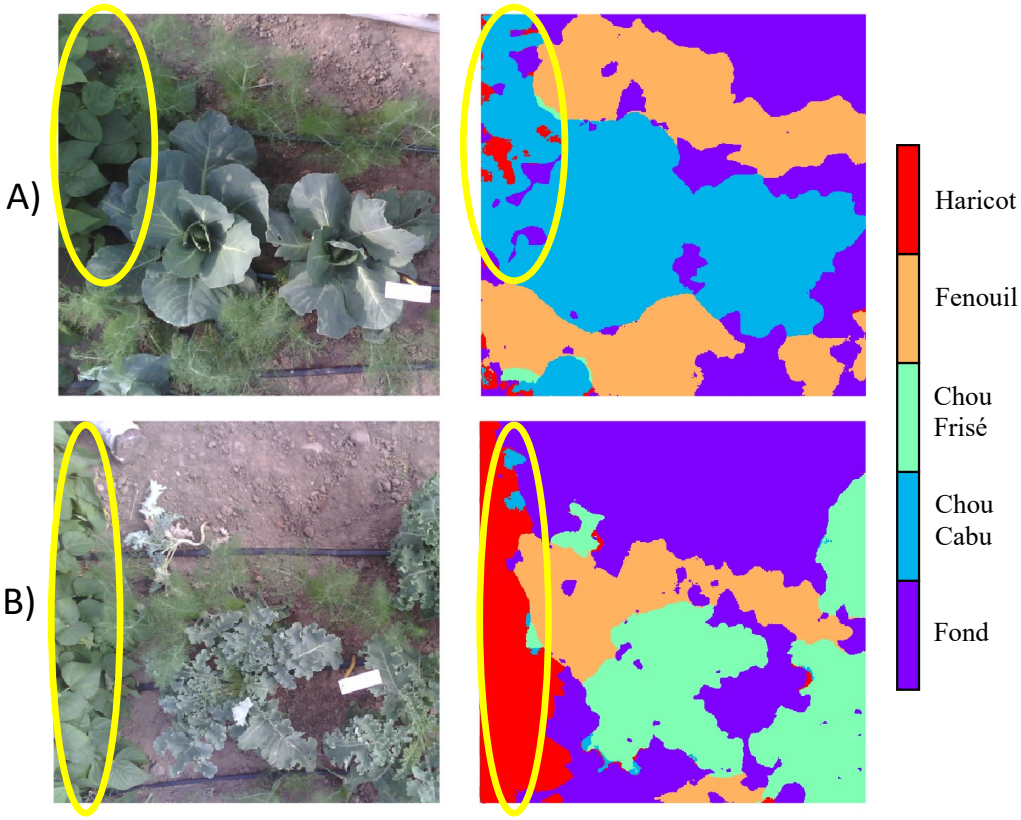


Figure 18 : Comparaison de 2 images de culture associées avec intrusion de haricots dans le cercle jaune. A) Culture associée de choux cabus et de fenouil. On remarque la grande confusion qu'il y a entre l'intrusion de haricots et le chou cabu. B) Culture associée de choux frisés et de fenouil. Cette fois-ci les haricots sont correctement identifiés. Les images étant très similaires, la différence entre-elles étant la présence de choux cabus et ayant connaissance des possibles difficulté que peut éprouver le FCNN à différencier ces 2 espèces, on peut raisonnablement croire que la prédiction de l'espèce d'un pixel peut dépendre des autres espèces en présence.

3.2.2.2. Utilisation de la certitude

Bien qu'il soit impossible de quantifier la qualité des prédictions du modèle sur les cultures associées, il est possible de leur attribuer un score de confiance. En effet, nous avons constaté qu'il existe une corrélation entre l'incertitude du modèle et la justesse de la prédiction. Par incertitude, on entend la probabilité associée à la classe prédite. Plus celle-ci est proche de 1, plus on peut considérer que le modèle est « sûr de lui ». Voir Figure 19.

Puisqu'il existe un lien entre cette certitude et la justesse, il est possible d'identifier les images « probablement mal segmentées » en triant les résultats par ordre de certitude croissant. Cette certitude peut obtenue en sommant les probabilités associées aux prédictions de l'ensemble des pixels que l'on divisera ensuite par le nombre total de pixels de l'image. On obtiendra ainsi un score de certitude entre 0.2 et 1. En effet, la probabilité minimale d'une prédiction est strictement supérieure à 0.2 car il y a 5 classes possibles.

L'utilisateur peut donc décider de regarder les images ayant les plus faibles scores de certitude pour identifier les types d'erreurs que le modèle commet et agir en conséquence. Il pourrait également décider de ne garder que les images qui ont une certitude au-delà d'un certain seuil pour ses analyses.

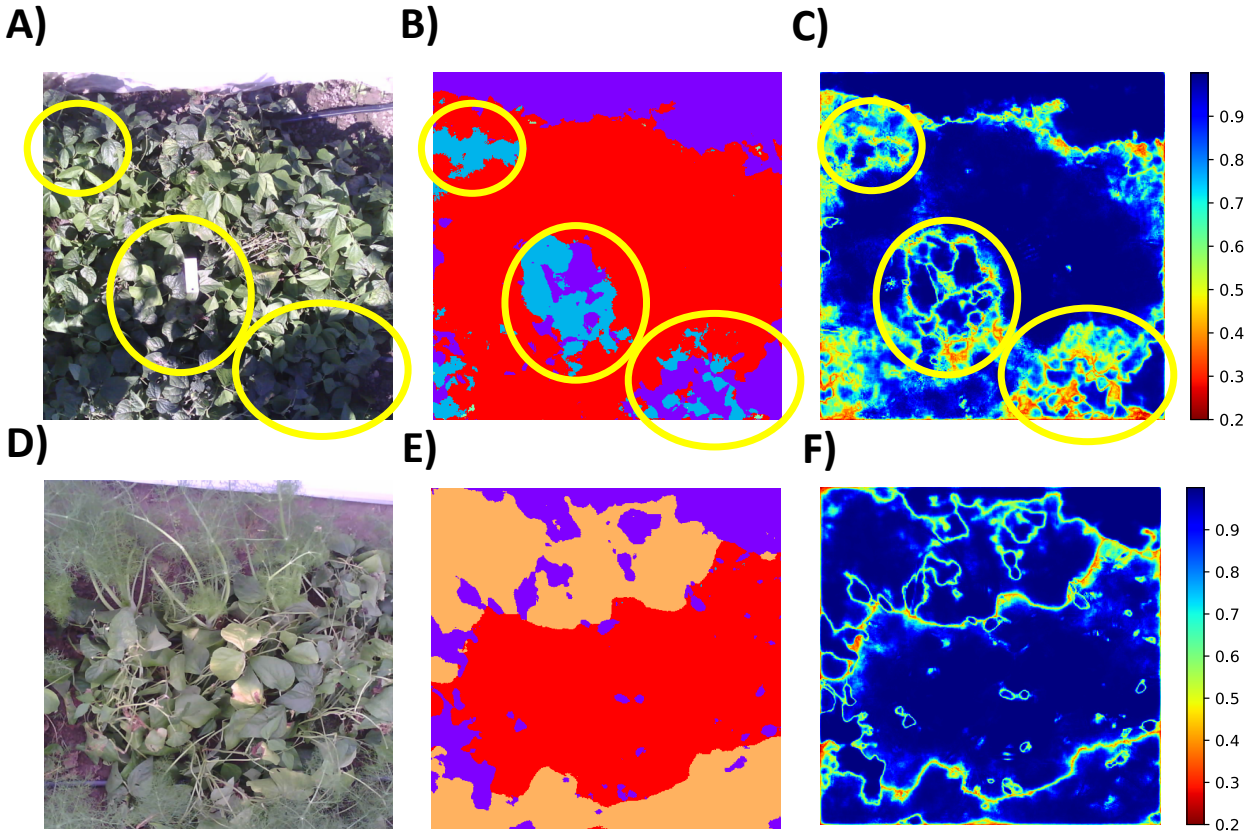


Figure 19 : Illustration de la corrélation entre erreur et incertitude. A) Culture pure de haricot. Les zones entourées en jaune sont des exemples de zones où le modèle s'est trompé. B) Prédictions du modèle. C) Probabilité que le modèle associe à la prédiction. On constate clairement que les zone où le modèle se trompe sont associée à une faible probabilité. Autrement dit, à une grande incertitude. D) Culture associée de haricot et de fenouil. E) la prédiction, pratiquement parfaite. F) Probabilité de prédiction, on voit que le modèle associe partout une haute probabilité à ses prédictions, hormis aux zones frontières entre 2 classes. Ceci est attendu car les frontières sont naturellement plus difficiles à identifier précisément.

4. Discussion

Nous avons montré dans ce travail que la segmentation d'images d'associations de cultures était possible sans avoir la contrainte du travail d'annotation très complexe et très couteux qu'il aurait impliqué. L'alternative proposée consiste à annoter des images de cultures pures, pour apprendre à un FCNN à les segmenter, puis d'utiliser ce FCNN entrainé pour inférer sur des images de cultures associées. Il en résulte un gain de temps important car les images de cultures pures sont plus simples et moins couteuses à annoter.

Au travers de ces résultats, la méthode proposée semble fonctionner : Il est possible segmenter des images de cultures associées avec un FCNN sans avoir à lui fournir des masques d'associations de cultures segmentées, des masques de cultures pures suffisent. De plus, il n'est pas nécessaire d'annoter de façon exhaustive manuellement.

4.1. Segmentation Binaire LeafPainter

L'apprentissage actif a montré son potentiel de réduction du temps d'annotation, puisque le résultat d'approximativement une minute et 30 secondes par image (voir section 3.1) est une grande avancée par rapport au temps d'annotation de 10h par image rapporté par Fuentes-Pacheco et al. (2019). Le temps d'annotation est un paramètre que peu d'auteurs rapportent alors qu'il est crucial, puisqu'il est l'intérêt principal des méthodes d'active learning. Par exemple, Sheikh et al. (2020) comparent différentes méthodes d'apprentissage actif mais ne les évaluent que sur base du nombre d'échantillon nécessaires d'entrainement pour arriver à une même performance, sans parler du temps nécessaire pour traiter ces échantillons. Même si la quantité de données nécessaires pour un entrainement est également un paramètre intéressant à minimiser, il ne faut pas perdre de vue que l'objectif essentiel est de minimiser le temps de travail. Il sera certainement toujours plus intéressant d'avoir, pour une même performance, un jeu de données volumineux mais rapide à créer qu'un jeu de données léger mais couteux en termes de temps pour le générer, comme c'est le cas pour les jeux de données synthétiques. Sayez & De Vleeschouwer (2022) arrivent également à cette conclusion, en montrant qu'une annotation grossière peut requérir jusqu'à 10 fois moins de temps qu'une annotation de grande qualité pour parvenir à une même performance.
Il parait donc, encore une fois, crucial que les travaux qui étudient ce problème rapportent également l'efficacité temporelle de leur méthode. L'utilisation d'un jeu de données commun, d'images de cultures, dédié à l'évaluation de l'efficacité temporelle de méthodes d'annotation serait idéal afin d'avoir des comparaisons justes.

En ce qui concerne LeafPainter, de futurs travaux pourraient évaluer sa performance sur d'autres jeux de données qui ont pour objet la segmentation de plantes. Il est également important de préciser que LeafPainter n'est, par essence, pas lié à la segmentation de plantes et il serait aisé de l'adapter à d'autres domaines, ce que de futurs travaux pourraient également étudier.

4.2. Segmentation Multi-Classes

4.2.1. Cultures pures

La stratégie consistant à utiliser des masques fournis par LeafPainter pour entrainer un FCNN à segmenter des images de cultures pures en fonction de leurs espèces s'est avérée un bon choix. C'est un résultat attendu puisque l'apprentissage de la tache correspond aux données fournies. Ce qui est plus intéressant c'est que, comme expliqué section 3.2.1 et illustré Figure 11, le FCNN parvient à améliorer les masques de LeafPainter. Ceci a plusieurs implications.

D'abord, le fait que les masques ne soient pas parfaits complexifie l'analyse quantitative de la performance. En effet, comme mentionné section 3.2.1, une précision de 100% indiquerait une reproduction des erreurs de LeafPainter. Cela rend la méthode difficilement évaluable et comparable, du point de vue de la performance. D'autant plus qu'historiquement, les progrès algorithmiques en apprentissage profond sont souvent très légers en termes, par exemple de précision, souvent de l'ordre quelques pourcents (Liu et al., 2022 ; Yu et al., 2018). Cet ordre de grandeur n'est probablement pas significatif lorsque les données d'entrainement sont imparfaites et il est donc difficile d'identifier les paramètres amenant des petits gains de performance.
Le même souci est observé pour la procédure d'entrainement. Il est commun, en apprentissage profond, d'arrêter l'entrainement de son modèle manuellement lorsqu'on observe que la fonction de cout appliquée aux données de validation atteint son minimum. Cela permet d'éviter le sur-apprentissage (Ying, 2019). Ceci n'est pas forcément pertinent dans notre cas puisque ce minimum pourrait représenter la meilleure copie des erreurs de LeafPainter. Il est donc important d'avoir un aperçu qualitatif des prédictions sur les jeux de données d'entrainement et de validation en les visualisant tout au long de la phase d'apprentissage. Cela rend cette méthode moins automatique et rend la phase d'entrainement non triviale, par rapport à un entrainement sur un jeu de données parfait.

Ensuite, le fait que le FCNN soit capable d'améliorer les masques est cohérent avec Zhang et al. (2018), qui ont observé la faculté des réseaux de neurones à régulariser des annotations imparfaites et avec Sayez & De Vleeschouwer (2022), puisque cette amélioration montre que les annotations n'ont pas besoin d'être parfaites pour avoir d'excellents résultats. Ce type d'analyse est plutôt rare dans la littérature et de futurs travaux pourraient en étudier plusieurs aspects :

- réaliser une comparaison entre les masques imparfaits et masques parfaits pour mesurer à quel point le FCNN peut améliorer la qualité des masques ;
- étudier le type éventuel d'erreur qui subsiste pour permettre de vérifier si les masques peuvent être imparfaits de n'importe quelle façon ou si l'erreur résiduelle pourrait être corrélée au type d'imperfection des masques ;
- mesurer à quel point les masques peuvent être imparfaits pour arriver à une même performance. Cela permettrait d'avoir une idée du niveau de qualité requis pour une performance donnée et donc de pouvoir prendre une décision informée quant au temps investi par image en fonction du résultat souhaité.

4.2.2. Cultures Associées

Enfin, l'utilisation du FCNN entrainé sur images de cultures pures pour inférer sur des images de cultures associées permet d'obtenir une segmentation multi-espèce de qualité. Le FCNN arrive à généraliser son apprentissage à reconnaitre des espèces uniques par image vers des images plus complexes où les plantes peuvent être enchevêtrées. Cela montre que la méthode proposée répond bien à la problématique de départ et qu'il est possible de s'affranchir de la plupart du travail d'annotation, inhérent à l'apprentissage supervisé pour segmenter de façon sémantique des images d'associations de cultures. De plus, cette tâche ne nécessite que du travail sur des images de cultures pures, bien plus faciles à obtenir. En effet, si la tâche avait dû être réalisée de façon classique, c'est-à-dire en labelisant des images d'associations de culture, elle aurait été non seulement plus longue par la difficulté d'annotation de ces images, mais il aurait également été bien plus couteux d'obtenir suffisamment d'images puisque le nombre de combinaisons d'associations possible croit de manière factorielle avec le nombre d'espèces.
Il faut toutefois admettre que cette méthode apporte quelques difficultés à prendre en compte.

Tout d'abord, la non quantifiabilité de la performance sur les associations de culture est problématique puisqu'il n'est pas possible de connaitre de façon triviale la performance du modèle sur les cultures associées. Seule une évaluation qualitative est possible et prend donc un peu plus de temps, puisqu'il faudra visualiser les résultats. De plus, identifier des récurrences dans les erreurs, et ainsi pouvoir les corriger, va requérir connaissance expertise et exploration du jeu de donnée afin d'identifier de potentielles récurrences dans l'erreur pour pouvoir la corriger.

Il est toutefois, comme montré section 3.2.2.1, possible de connaitre à priori le type d'erreur qui sera commise en regardant les erreurs sur les images de cultures pures qui ont servi à l'entrainement et à la validation du FCNN. Cela permet de fournir au FCNN des images de cultures pures présentant les caractéristiques ayant causé l'erreur et d'ainsi être relativement confiant quant à la robustesse du FCNN à ce type d'erreur. Par exemple pour l'erreur illustrée Figure 15, il devrait suffire d'ajouter au set d'entrainement des images de choux cabus présentant ces caractéristiques. De la même façon, pour l'image Figure 16, en ajoutant des images ombragées de haricot au set d'entrainement, l'erreur devrait être résolue pour les images d'associations de cultures également.

Il reste cependant un type d'erreur plus difficiles à prévoir, celles inhérentes à la complexité émergente amenée par l'association de culture. Les cas de ce genre rencontrés dans ce travail sont illustrés Figure 17 et Figure 18. Parmi ces cas, il y a les prévisibles et les plus difficile à prévoir. Par exemple, l'erreur causée par la transparence du fenouil est une erreur qui ne se reproduira que dans le cas de présence d'une autre espèce ayant un aspect transparent, est donc assez difficile à prévoir, puisqu'elle est inhérente aux données et pas à la méthode. En revanche, l'erreur illustrée à la Figure 18 causée par une probable corrélation spatiale trop grande induite par l'entrainement en culture pure uniquement est prévisible, puisque directement lié à la méthode et cohérente avec la matrice de confusion Figure 13. Ce type d'erreur est donc prévisible et potentiellement évitable. Une solution pourrait être d'entrainer le modèle, non pas sur des images entières mais sur des mosaïques

d’images de cultures pures afin de, déjà au cours de cette étape, forcer le modèle à reconnaitre plusieurs espèces par images.

Nous avons également montré qu’il était possible d’identifier les images pour lesquelles le risque d’erreur était grand en utilisant la certitude du modèle. Cela peut permettre d’avoir une évaluation probabilistique la performance qui permet d’attribuer un risque d’erreur aux analyses réalisées en aval de la segmentation ou encore de pouvoir identifier plus facilement les images présentant une erreur et de commencer l’analyse de l’erreur, comme expliqué plus tôt. Une fois des récurrences ou des patterns identifiés, l’utilisateur peut vérifier si ces patterns se répètent dans les images présentant une certitude élevée.

On voit donc que le prix à payer pour la réduction du travail manuel d’annotation se traduit par deux nouvelles contraintes :

- Une évaluation plus compliquée de la performance. On peut toutefois se servir de la performance sur les cultures pures comme proxy raisonnable, comme discuté section 3.2.2.1.
- Une nouvelle source d’erreur, due à l’association.

Enfin, pouvoir obtenir ce type de masques à cout plus faible devrait permettre d’étudier plus facilement et plus rapidement les dynamiques spatio-temporelles d’évolution de la superficie des canopées, comme cela a été réalisé par Chevalier (2022), en partie grâce à cette méthode. Cela pourrait également aider de futurs travaux à accroitre la vitesse de production de leurs données et ainsi permettre à toute la communauté scientifique de disposer de plus d’images segmentées d’associations de culture et d’accélérer l’acquisition de nouvelles connaissances sur les mécanismes complexes conditionnant le succès ou non d’une association.

5. Conclusion

Nous avons montré dans ce travail que la segmentation d'images d'associations de cultures était possible sans avoir la contrainte du travail d'annotation très complexe et très coûteux qu'il aurait impliqué. L'alternative proposée consiste à annoter des images de cultures pures, pour apprendre à un FCNN à les segmenter, puis d'utiliser ce FCNN entrainé pour inférer sur des images de cultures associées. Il en résulte un gain de temps important car les images de cultures pures sont plus simples et moins couteuses à annoter.

Nous avons également présenté une méthode d'annotation pour images en culture pure à base de GBDT, nommée LeafPainter, ne nécessitant en moyenne qu'un travail de deux minutes et 30 secondes par image, ce qui est un gain de temps significatif par rapport au méthodes existantes. Nous avons également montré que, même si les annotations renvoyées par LeafPainter n'étaient pas parfaites, le FCNN est quand même capable d'apprendre la segmentation et même d'améliorer les masques sur lesquels il s'en entrainé.

Enfin, nous avons identifié les limitations et inconvénients de cette méthode, à savoir la difficulté d'évaluation quantitative de sa performance, par l'absence de masques de référence. Toutefois, cet inconvénient peut être limité puisque nous avons vu qu'une partie de l'erreur est identique à celle du jeu de données d'entrainement contenant les images de cultures pures et l'utilisation de l'erreur de validation en culture pure comme approximation de l'erreur de prédiction sur les images de cultures associées a du sens. Il existe toutefois un type d'erreur inhérent à l'association requérant expertise et réflexion pour être corrigée, telle que la corrélation spatiale induite par l'entrainement en culture pure.

Bibliographie

Adams, J., Qiu, Y., Xu, Y., & Schnable, J. C. (2020). Plant segmentation by supervised machine learning methods. *The Plant Phenome Journal,, 3*(1), e20001.

Apprentissage automatique. (2022, Janvier 27). Consulté le Avril 2022, sur Wikipédia, l'encyclopédie libre: https://fr.wikipedia.org/w/index.php?title=Sp%C3%A9cial:Citer&page=Apprentissage_automatique&id=190282589&wpFormIdentifier=titleform

Apprentissage profond. (2022, Février 3). Consulté le Avril 2022, sur Wikipédia, l'encyclopédie libre: https://fr.wikipedia.org/w/index.php?title=Sp%C3%A9cial:Citer&page=Apprentissage_profond&id=190494975&wpFormIdentifier=titleform

Arvidsson, S., Pérez-Rodriguez, P., & Bernd, M.-R. (2011). A growth phenotyping pipeline for Arabidopsis thaliana integrating image analysis and rosette area modeling for robust quantification of genotype effects. *New Phytologist, 191*(3), 895-907.

Barth, R., IJsselmuiden, J., Hemming, J., & Van Henten, E. J. (2018). Data synthesis methods for semantic segmentation in agriculture. *Computers and electronics in agriculture, 144*, 284-296.

Barth, R., IJsselmuiden, J., Hemming, J., & Van Henten, E. J. (2019). Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Computers and Electronics in Agriculture, 161*, 291-304.

Bhanu, B., & Jones, T. L. (1993). Image understanding research for automatic target recognition. *EEE Aerospace and Electronic Systems Magazine, 8*(10), 15-23.

Biewald, L. (2020). *Experiment Tracking with Weights and Biases*. Récupéré sur Weights & Biases: wandb.com

Boehmke, B., & Greenwell, B. (2017). *Hands-on machine learning with R.* Chapman and Hall/CRC.

Boudreau, M. A. (2013). Diseases in Intercropping Systems. *Annual Review of Phytopathology, 51*, 499-519.

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

Breinman, L. (1997). Arcing The Edge. *Statistics Department, University of California, Berkeley*, Technical Report 486.

Chaurasia, A., & Culurciello, E. (2017). Linknet: Exploiting encoder representations for efficient semantic segmentation. *2017 IEEE Visual Communications and Image Processing*, 1-4.

Chevalier. (2022). Exploring Intercropping in Market Gardening A story of combinations. *Thèse de doctorat, Université Catholique de Louvain*, 454.

Chevalier, C., Draye, X., & Baret, P. (2021). Reliability of growth cycle and architectural characteristics as criteria for intercropping design in market gardening: Strategies to cluster vegetable crops according to their architecture. *Aspects of Applied Biology, 146*, 7.

Das Choudhury, S., Bashyam, S., Qiu, Y., A., S., & Awada, T. (2018). Holistic and component plant phenotyping using temporal image sequence. *Plant Methods, 14*(1), 1-21.

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

Dyrmann, M., Jørgensen, R. N., & Midtiby, H. S. (2017). RoboWeedSupport-Detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. *Advances in Animal Biosciences, 8*(2), 842-847.

Eich, B. (2005). JavaScript at ten years. *ICFP, 5*, 129.

Fuentes-Pacheco, J., Torres-Olivares, J., Roman-Rangel, E., Cervantes, S., Juarez-Lopez, P., Hermosillo-Valadez, J., & Rendón-Mancha, J. M. (2019). Fig plant segmentation from aerial images using a deep convolutional encoder-decoder network. *Remote Sensing, 11*(10), 1157.

Güldenring, R., Boukas, E., Ravn, O., & Nalpantidis, L. (2021). Few-leaf learning: Weed segmentation in grasslands. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3248-3254.

Gaudio, N., Escobar-Gutiérrez, A. J., Casadebaig, P., Evers, J. B., Gérard, F., Louarn, G., & Justes, E. (2019). Current knowledge and future research opportunities for modeling annual crop mixtures. A review. *Agronomy for Sustainable Development, 39*(2), 1-20.

George, D., Shen, H., & Huerta, E. A. (2018). Classification and unsupervised clustering of LIGO data with Deep Transfer Learning. *Physical Review D, 97*(10), 101501.

Granier, C., Aguirrezabal, L., Chenu, K., Cookson, S. J., Dauzat, M., Hamard, P., Lebaudy, A. (2006). PHENOPSIS, an automated platform for reproducible phenotyping of plant responses to soil water deficit in Arabidopsis thaliana permitted the identification of an accession with low sensitivity to soil water deficit. *New Phytologist, 169*(3), 623-635.

Grinberg, M. (2018). *Flask web development: developing web applications with python.* Reilly Media, Inc.

Guerrero, J. M., Pajares, G., Montalvo, M., Romeo, J., & Guijarro, M. (2012). Support vector machines for crop/weeds identification in maize fields. *Expert Systems with Applications, 39*(12), 11149-11155.

Guerrero, J. M., Pajares, G., Montalvo, M., Romeo, J., & Guijarro, M. (2012). Support vector machines for crop/weeds identification in maize fields. *Expert Systems with Applications, 39*(12), 11149-11155.

Guijarro, M., Pajares, G., Riomoros, I., Herrera, P. J., Burgos-Artizzu, X. P., & Ribeiro, A. (2011). Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture, 75*(1), 75-83.

Guijarro, M., Pajares, G., Riomoros, I., Herrera, P. J., Burgos-Artizzu, X. P., & Ribeiro, A. (2011). Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture, 75*(1), 75-83.

Guvenc, I., & Yildirim, E. (2008). Increasing Productivity with Intercropping Systems in Cabbage Production. *Journal of Sustainable Agriculture, 28*(4), 29-44.

Guyon, I., Cawley, G. C., Dror, G., & Lemaire, V. (2011). Results of the active learning challenge. *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010* , 19-45.

Hague, T., Tillett, N. D., & Wheeler, H. (2006). Automated crop and weed monitoring in widely spaced cereals. *Precision Agriculture, 7*(1), 21-34.

Haji, S. H., & Abdulazeez, A. M. (2021). Comparison of optimization techniques based on gradient descent algorithm: A review. *PalArch's Journal of Archaeology of Egypt/Egyptology, 18*(4), 2715-2743.

Hamuda, E., & Glavin, M. J. (2016). A survey of image processing techniques for plant extraction and segmentation in the field Author links open overlay panel. *Computers and Electronics in Agriculture, 125*, 184-199.

Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review. *Journal of big data, 7*(1), 1-45.

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

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

He, X., Xue, F., Ren, X., & You, Y. (2021). Large-Scale Deep Learning Optimizations: A Comprehensive Survey. *arXiv preprint*, 2111.00856.

Hearst, M., Dumais, S., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications, 13*(4), 18-28.

Hesamian, M. H., Jia, W., He, X., & Kennedy, P. (2019). Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging, 4*(32), 582-596.

Hickson, I., & Hyatt, D. (2011). Html5. *W3C Working Draft WD-html5*, 53.

Howard, J., & Gugger, S. (2020). Fastai: a layered API for deep learning. *Information, 11*(2), 108.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448-456.

Iqbal, N., Hussain, S., Ahmed, Z., Yang, F., Wang, X., Liu, W., Liu, J. (2019). Comparative analysis of maize–soybean strip intercropping systems: a review. *Plant Production Science, 22*(2), 131-142.

Jadon, S. (2020). A survey of loss functions for semantic segmentation. *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 1-7.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science, 156*(3), 312-322.

Kataoka, T., Kaneko, T., Okamoto, H., & Hata, S. (2003). Crop growth estimation system using machine vision. *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2*, b1079-b1083.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems, 30*.

Kim, W. S., Lee, D. H., Kim, H., Sim, T., & Kim, Y. J. (2021). Weakly supervised crop area segmentation for an autonomous combine harvester. *Sensors, 21*(14), 4801.

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

Kirk, K., Andersen, H. J., Thomsen, A. G., Jørgensen, J. R., & Jørgensen, R. N. (2009). Estimation of leaf area index in cereal crops using red–green images. *Biosystems Engineering, 104*(3), 308-317.

Klukas, C., Chen, D., & Pape, J. M. (2014). Integrated analysis platform: an open-source information system for high-throughput plant phenotyping. *Plant physiology, 165*(2), 506-518.

Knipling, E. B. (1970). Knipling, E. B. (1970). Physical and physiological basis for the reflectance of visible and near-infrared radiation from vegetation. *Remote sensing of environment, 1*(3), 155-159.

Kumar, S., Luo, W., Kantor, G., & Sycara, K. (2019). Active Learning with Gaussian Processes for High Throughput Phenotyping. *arXiv preprint* , arXiv:1901.06803.

Lee, U., Chang, S., Putra, G., Kim, H., & Kim, D. H. (2018). An automated, high-throughput plant phenotyping system using machine learning-based plant segmentation and image analysis. *PloS one, 13*(4), e0196615.

Leister, D., Varotto, C., Pesaresi, P., Niwergall, A., & Salamini, F. (1999). Large-scale evaluation of plant growth in Arabidopsis thaliana by non-invasive image analysis. *Plant Physiology and Biochemistry, 37*(9), 671-678.

Li, F.-F. (2022). *Convolutional Neural Networks (CNNs / ConvNets)*. Consulté le Mars 2022, sur CS231n: Deep Learning for Computer Vision: https://cs231n.github.io/

Lie, H. W. (1994). Cascading HTML style sheets – a proposal.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2980-2988.

Liu, X., Deng, Z., & Yang, Y. (2019). Recent progress in semantic image segmentation. *Artificial Intelligence Review, 52*(2), 1089-1106.

Liu, X., Song, L., Liu, S., & Zhang, Y. (2021). A review of deep-learning-based medical image segmentation methods. *Sustainability, 13*(3), 1324.

Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. *arXiv preprint*, arXiv:2201.03545.

Lobet, G., Draye, X., & Périlleux, C. (2013). An online database for plant image analysis software tools. *Plant methods, 9*(1), 1-8.

Lobet, G., Pagès, L., & Draye, X. (2011). A Novel Image-Analysis Toolbox Enabling Quantitative Analysis of Root System Architecture. *Plant physiology, 157*(1), 29-39.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431-3440.

Lottes, P., Behley, J., Milioto, A., & Stachniss, C. (2018). Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters, 3*(4), 2870-2877.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1*(14), 281-297.

McCarthy, C. L., Hancock, N. H., & Raine, S. R. (2010). Applied machine vision of plants: a review with implications for field deployment in automated farming operations. *Intelligent Service Robotics, 3*(4), 209-217.

Meyer, G. E., & Neto, J. C. (2008). Verification of color vegetation indices for automated crop imaging applications. *Computers and Electronics in Agriculture, 63*(2), 282-293.

Meyer, G. E., Hindman, T. W., & Laksmi, K. (1999). Machine vision detection parameters for plant species identification. *Precision agriculture and biological quality, 3543*, 327-335.

Meyer, G. E., Neto, J. C., Jones, D. D., & Hindman, T. W. (2004). Intensified fuzzy clusters for classifying plant, soil, and residue regions of interest from color images. *Computers and electronics in agriculture, 42*(3), 161-180.

Milioto, A., Lottes, P., & Stachniss, C. (2018). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. *2018 IEEE international conference on robotics and automation (ICRA)*, 2229-2235.

Mutka, A., & Bart, R. (2015). Image-based phenotyping of plant disease symptoms. *Frontiers in Plant Science, 5*, 734.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics, 9*(1), 62-66.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems 32*, 8024--8035.

Petti, D., & Li, C. (2022). Weakly-supervised learning to automatically count cotton flowers from aerial imagery. *Computers and Electronics in Agriculture, 194*, 106734.

PICON RUIZ, A. R., ALVAREZ GILA, A. I., Irusta, U., & ECHAZARRA HUGUET, J. O. (2020). Why deep learning performs better than classical machine learning? *Dyna Ingenieria E Industria*.

Picon, A., San-Emeterio, M. G., Bereciartua-Perez, A., Klukas, C., Eggers, T., & Navarra-Mestre, R. (2022). Deep learning-based segmentation of multiple species of weeds and corn crop using synthetic and real image datasets. *Computers and Electronics in Agriculture, 194*, 106719.

Pound, M. P., Atkinson, J. A., Townsend, A. J., Wilson, M. H., Griffiths, M., Jackson, A. S., & French, A. P. (2017). Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *Gigascience, 6*(10), gix083.

Qian, X., & Klabjan, D. (2020). The impact of the mini-batch size on the variance of gradients in stochastic gradient descent. *arXiv preprint*, arXiv:2004.13146.

Rasmussen, J., Nørremark, M., & Bibby, B. M. (2007). Assessment of leaf cover and crop soil cover in weed harrowing research using digital images. *Weed Research, 47*(4), 299-310.

Ratner, A., Varma, P., & Hancock, B. (2019). *Weak Supervision: A New Programming Paradigm for Machine Learning*. Consulté le 2022, sur The Stanford AI Lab Blog About: https://ai.stanford.edu/blog/weak-supervision/

Rawat, S., Chandra, A. L., Desai, S. V., Balasubramanian, V. N., Ninomiya, S., & Guo, W. (2022). How Useful Is Image-Based Active Learning for Plant Organ Segmentation? *Plant Phenomics*.

Riehle, D., Reiser, D., & Griepentrog, H. W. (2020). Robust index-based semantic plant/background segmentation for RGB-images. *Computers and Electronics in Agriculture, 169*, 105201.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, 234-241.

Sayed Roholla, M., & Hamdollah, E. (2011). A General Overview on Intercropping and Its Advantages in Sustainable Agriculture. *Journal of Applied Environmental and Biological Sciences, 1*(11), 482-486.

Sayez, N., & De Vleeschouwer, C. (2022). Accelerating the creation of instance segmentation training sets through bounding box annotation.

Scikit-Learn. (2021, Octobre). *Gradient-boosting decision trees Hyperparameter tuning*. Récupéré sur Scikit-Learn course: https://inria.github.io/scikit-learn-mooc/python_scripts/ensemble_hyperparameters.html#:~:text=Gradient%2Dboosting%20decision%20trees,-For%20gradient%2Dboosting&text=The%20important%20parameters%20are%20n_estimators,as%20previously%20discussed%20random%20

Settles, B. (2019). Active learning literature survey.

Sheikh, R., Milioto, A., Lottes, P., Stachniss, C., Bennewitz, M., & Schultz, T. (2020). Gradient and log-based active learning for semantic segmentation of crop and weed for agricultural robots. *EEE International Conference on Robotics and Automation (ICRA)*, 1350-1356.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1874-1883.

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. *arXiv preprint*, arXiv:1803.09820.

Smith, L. N., & Topin, N. (2019). Super-convergence: Very fast training of neural networks using large learning rates. *Artificial intelligence and machine learning for multi-domain operations applications, 11006*, 1100612.

Sudre, C. H., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Deep learning in medical image analysis and multimodal learning for clinical decision support*, 240-248.

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

Tang, J., Wang, D., Zhang, Z., He, L., Xin, J., & Xu, Y. (2017). Weed identification based on K-means feature learning combined with convolutional neural network. *Computers and electronics in agriculture, 135*, 63-70.

Tang, X., Zhong, R., Jiang, J., He, L., Huang, Z., Shi, G., . . . He, L. (2020). Cassava/peanut intercropping improves soil quality via rhizospheric microbes increased available nitrogen contents. *BMC Biotechnol, 20*, 13.

Van Rossum, G., & Drake Jr, F. L. (1994). Python reference manual.

Walter, A., Scharr, H., Gilmer, F., Zierer, R., Nagel, K. A., Ernst, M., Uhlig, B. (2007). Dynamics of seedling growth acclimation towards altered light conditions can be quantified via GROWSCREEN: a setup and procedure designed for rapid optical phenotyping of different plant species. *New Phytologist, 174*(2), 447-455.

Wang, A., Xu, Y., Wei, X., & Cui, B. (2020). Semantic segmentation of crop and weed using an encoder-decoder network and image enhancement method under uncontrolled outdoor illumination. *IEEE Access, 8*, 81724-81734.

Wang, K., Zhang, D., Li, Y., Zhang, R., & Lin, L. (2016). Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology, 27*(12), 2591-2600.

Wang, L., Wang, C., Sun, Z., & Chen, S. (2020). An improved dice loss for pneumothorax segmentation by mining the information of negative areas. *IEEE Access, 8*, 167939-167949.

Wang, S., Waldner, F., & Lobell, D. B. (2022). Unlocking large-scale crop field delineation in smallholder farming systems with transfer learning and weak supervision. *arXiv preprint*, arXiv:2201.04771.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 1-40.

Woebbecke, D. M., Meyer, G. E., Von Bargen, K., & Mortensen, D. A. (1993). Plant species identification, size, and enumeration using machine vision techniques on near-binary images. *Optics in Agriculture and Forestry, 1836*, 218-219.

Woebbecke, D. M., Meyer, G. E., Von Bargen, K., & Mortensen, D. A. (1995). Color indices for weed identification under various soil, residue, and lighting conditions. *Transactions of the ASAE, 38*(1), 259-269.

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series, 1168*(2), 022022.

Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., & Tang, Y. (2018). Methods and datasets on semantic segmentation: A review. *Neurocomputing, 304*, 82-103.

Zhang, J., Zhang, T., Dai, Y., Harandi, M., & Hartley, R. (2018). Deep unsupervised saliency detection: A multiple noisy labeling perspective. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9029-9038.

Zhang, Z., Liu, Q., & Wang, Y. (2018). Road extraction by deep residual u-net. *EEE Geoscience and Remote Sensing Letters, 15*(5), 749-753.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. *IEEE conference on computer vision and pattern recognition*, 2921-2929.

Zhou, Z., Shin, J., Zhang, L., Gurudu, S., Gotway, M., & Liang, J. (2017). Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7340-7351.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE, 109*(1), 43-76.

6. ANNEXES

6.1. Indices de couleurs principaux

L'indice de différence normalisée (NDI)

Développé par (Woebbecke, Meyer, Von Bargen, & Mortensen, Plant species identification, size, and enumeration using machine vision techniques on near-binary images, 1993) cet indice transforme les valeurs des pixels en valeurs entre −1 et +1. Puis, pour avoir des valeurs entre 0 et 255, l'indice est traité en ajoutant 1 puis est multiplié par 128. Sa formule est la suivante :

$$NDI = 128 * \left(\frac{G - R}{G + R} + 1\right) \qquad \textbf{Eq. 1}$$

Son intérêt est de profiter des propriétés de réflectance des plantes qui, en règle générale, absorbent beaucoup dans le rouge et réfléchissent beaucoup dans le vert (Knipling, 1970).

Indice d'excédent vert (ExG)

Cet indice a été développé par (Woebbecke, Meyer, Von Bargen, & Mortensen, 1995). en utilisant des coordonnées chromatiques et une teinte (HUE ???) modifiée pour distinguer la plante du sol. L'indice final a pour formule :

$$ExG = 2g - r - b \qquad \textbf{Eq. 2}$$

Où r, g et b sont les coordonnées chromatiques, calculée comme suit :

$$r = \frac{R^*}{(R^* + G^* + B^*)} \quad , \quad g = \frac{G^*}{(R^* + G^* + B^*)} \quad , \quad b = \frac{B^*}{(R^* + G^* + B^*)} \qquad \textbf{Eq. 3}$$

où R^*, G^* et B^* sont les valeurs RGB normalisées comprises entre 0 et 1, et sont calculés comme suit :

$$R^* = \frac{R}{R_{max}} \quad , \quad G^* = \frac{G}{G_{max}} \quad , \quad B^* = \frac{B}{B_{max}} \qquad \textbf{Eq. 4}$$

L'indice ExG est l'un des indice les plus utilisé et a montré de très bon résultats pour distinguer les plantes de leur environnement (Guerrero, Pajares, Montalvo, Romeo, & Guijarro, 2012).

Indice d'excédent rouge (ExR)

(Meyer, Hindman, & Laksmi, 1999) Ont créé cet indice en s'inspirant de la distribution des cônes dans la rétine humaine (4% de bleu, 32% de vert et 64% de rouge) pour développer cet indice. Il est défini comme suit :

$$ExR = 1.3R - G \qquad \textbf{Eq. 5}$$

Malgré des performances satisfaisantes, (Meyer, Hindman, & Laksmi, 1999) ont montré qu'il n'était pas aussi performant que le ExG.

Indice de couleur de l'extraction de la végétation (CIVE)

Proposé par (Kataoka, Kaneko, Okamoto, & Hata, 2003), Cette méthode a été élaborée dans le but de séparer les plantes vertes du sol afin de pouvoir analyser la croissance de cultures de betteraves et de haricot. La formule du CIVE est la suivante :

$$CIVE = 0.441R - 0.811G + 0.385B + 18.78745 \qquad \textbf{Eq. 6}$$

Kataoka et al. ont constaté que le CIVE segmente les plantes de façon plus précise que la méthode proche infrarouge (NIR) car il met, d'après eux, davantage l'accent sur les corps verts.

Indice d'excédent verts moins l'indice d'excédent rouge (ExGR)

Cette méthode a été introduite par (Meyer G. E., Neto, Jones, & Hindman, 2004) et combine 2 indices de couleur, (ExG) et (ExR), abrodés précédement. Ces 2 indices ont appliquées simultanément pour séparer les plantes de l'environnement : ExG pour extraire la plante et ExR pour éliminer le bruit de fond (sol et résidus) où des matériaux similaires aux plantes d'intérêt (tiges, branches ou pétioles) peuvent exister. La formule du ExGR est définie comme suit :

$$ExGR = ExG - ExR \qquad \textbf{Eq. 7}$$

Indice végétatif (VEG)

Cet indice, proposé par (Hague, Tillett, & Wheeler, 2006) pour séparer les pixels de plantes (céréales et mauvaises herbes) des pixels de sol. L'étude a été menée en conditions extérieures. Pour réaliser la segmentation, une image RGB a été convertie, comme pour les indices précédents, en niveaux de gris à l'aide de la formule suivante :

$$VEG = \frac{G}{R^a B^{1-a}} \qquad \textbf{Eq. 8}$$

Où a est une constante de valeur 0.667. Les auteurs ont constaté que cette transformation permet d'avoir un bon contraste entre la plante et le sol. Un avantage de celle-ci est qu'elle est robuste au changement d'éclairage, caractéristique indispensable pour être utilisable en condition extérieures.

Indices Combinés 1 (COM1)

(Guijarro, et al., Automatic segmentation of relevant textures in agricultural images, 2011) ont sélectionné 4 indices précédemment abordés, ExG, CIVE, ExGR et VEG. Il les ont ensuite appliqués simultanément plutôt qu'individuellement pour tenter d'améliorer la qualité de la segmentation :

$$COM1 = ExGR + ExG + CIVE + VEG \qquad \textbf{Eq. 9}$$

Les auteurs ont montré que l'indice combiné donnait de meilleurs résultats que les indices utilisés individuellement. La méthode a été testée dans des champs d'orge et de maïs et a démontré être stable même avec des conditions d'éclairage variable, dans des environnements extérieurs.

Indice d'excédent vert modifié (MExG)

L'indice d'excédent vert modifié a été développé par (Guijarro, et al., 2011) et est défini comme suit :

$$CIVE = 1.262G - 0.884R + 0.311B \qquad \textbf{Eq. 10}$$

La discrimination entre la plante et son environnement par indice MExG a présenté une grande robustesse aux conditions d'éclairage changeantes, encore meilleure qu'avec le ExG dont il est inspiré.

Indices Combinés 2 (COM2)

Cet indice, introduit par (Guerrero, Pajares, Montalvo, Romeo, & Guijarro, 2012), et est similaire au COM1 car il combine également plusieurs indices de couleur : ExG, CIVE et VEG. Il a été construit en par moyenne pondérée de ces trois indices, la somme des poids valant 1. Sa formule est :

$$COM2 = 0.36\,ExG + 0.47\,CIVE + 0.17VEG \qquad \textbf{Eq. 11}$$

Segmentation sémantique faiblement supervisée de canopées multi-espèces à l'aide d'apprentissage actif et de réseaux de neurones entièrement convolutifs

Présenté par Charles Rongione

Le suivi des caractéristiques écophysiologiques des canopées multi-espèces est essentiel pour comprendre comment diverses espèces végétales poussent et interagissent ensemble et pour développer des pratiques agro-écologiques. Le suivi basé sur l'image est devenu de plus en plus populaire en raison d'un exellent rapport information/coût. La segmentation sémantique a montré un grand potentiel pour l'analyse des images de canopées et a déjà été appliquée dans un large éventail de tâches, telles que le phénotypage, la différenciation des mauvaises herbes et des cultures ou le comptage des feuilles. Cependant, elle souffre toujours d'un important goulot d'étranglement : l'acquisition de masques de segmentation. L'obtention de masques adéquats est une tâche fastidieuse qui rend souvent la segmentation d'images impraticable. La majorité des travaux réalisés pour réduire l'inefficacité du processus d'annotation ont été effectués sur des images de monoculture provenant de champs homogènes. Bien qu'elles soient efficaces pour de telles images, la plupart des méthodes proposées perdent une grande partie de leurs performances lorsqu'elles sont appliquées à des couverts hétérogènes. Par conséquent, ces méthodes ne sont pas transférables à des parcelles de culture multi-espèces.

Dans ce travail, nous présentons une méthode d'apprentissage actif basée sur des arbres de décision boostés par gradient qui permet d'obtenir des masques de segmentation binaire plante/fond de haute qualité en quelques minutes en annotant seulement de petites régions des images, même sur des images de cultures mixtes très diverses. Nous montrons ensuite qu'un réseau de neurones entièrement convolutif entraîné avec les masques générés par cette méthode sur des images de monoculture peut être généralisé aux images de cultures mixtes et classer de manière robuste chaque pixel de plante selon son espèce. Cette approche réduit considérablement la quantité de travail humain nécessaire pour segmenter sémantiquement les images de cultures mixtes, passant de plusieurs heures à quelques minutes par image, et facilite donc grandement l'étude des caractéristiques éco-physiologiques des couverts végétaux multi-espèces.

Nous réalisons également une analyse rigoureuse des limitations associées à cette méthodes et proposons des pistes d'amélioration de la méthode, ainsi que des directions vers lesquelles la future recherche en facilitation de la segmentation sémantique appliquée aux cultures pourrait s'orienter.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des bioingénieurs Croix du Sud, 2 bte L7.05.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/agro