



## Rapport de Stage de M2

Master de Mathématiques appliquées, Statistiques

### Estimation des traits de vigueur par analyse d'images pour le criblage des ressources génétiques des arbres fruitiers



Stage réalisé dans l'entreprise : INRAE

Période : 1<sup>er</sup> Mars au 31 Aout

Tuteur Université : GALERNE Bruno

Tuteurs INRAE : ROTH Morgane  
JAY Sylvain  
WEISS Marie

*Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre de France 2030 portant la référence ANR-16-CONV-0004.*

*This work was supported by the French National Research Agency under France 2030 program, referred as ANR-16-CONV-0004.*

## Remerciements

Je tiens à remercier Marie WEISS, Morgane ROTH et Sylvain JAY qui m'ont proposé ce sujet de stage et plus généralement, je remercie mes trois encadrants pour leur bienveillance et leurs conseils pendant tout ce stage, et pour leur aide à la rédaction du rapport.

Je remercie aussi Sabrina VIRET et Octave CABEL qui m'ont formé et donné de nombreux conseils par rapport à la notation de la floribondité.

Je remercie Hélène BLANCHETEAU qui m'a aidé pour des installations de librairies et d'environnements virtuels python.

Je remercie aussi Cyril LUQUET pour avoir installé un pied à la perche LITERAL. Cela a beaucoup aidé durant les acquisitions qui ont suivi.

Je tiens également à remercier les autres stagiaires et tous les membres du domaine Saint Maurice d'INRAE Avignon présents sur le lieu de mon stage pour leur accueil et leur sympathie.

Je remercie Ayoub NACHITE qui m'a prêté son bureau et qui était toujours disponible pour répondre à mes sollicitations.

Je tiens également à remercier Marie-Pia d'ARGAIGNON pour ses explications de code et pour avoir partagé sa compréhension de mon sujet de stage.

Je remercie Tristan DUBOS et Lucas BERNIGAUD SAMATAN pour leur aide précieuse concernant les codes et la partie technique liés aux réseaux de neurones.

Je remercie Hugo DESRUTINS pour m'avoir aidé lorsque j'en ai eu besoin et pour l'aura positive qu'il a véhiculée au quotidien.

Je tiens également à remercier les autres stagiaires et tous les membres du domaine Saint Paul d'INRAE Avignon présents sur le lieu de mon stage pour leur accueil et leur bienveillance.

Je tiens à remercier Harold DURUFLÉ sans qui je n'aurais jamais eu connaissance de ce stage.

Enfin, je remercie l'équipe administrative et pédagogique du Master Mathématiques Appliquées, Statistiques d'Orléans, l'ensemble des enseignants et intervenants, pour la qualité des cours dispensés et la bonne organisation des formalités administratives.

## Lexique

**ADN** : L'ADN (Acide DésoxyriboNucléique) est une macromolécule biologique présente dans presque toutes les cellules ainsi que chez de nombreux virus. L'ADN contient toute l'information génétique, appelée génome, permettant le développement, le fonctionnement et la reproduction des êtres vivants.

**Agroécosystème** : Écosystèmes cultivés, correspondant généralement à l'unité spatiale qu'est l'exploitation agricole et dont les fonctions écosystémiques sont valorisées par l'Homme sous forme de biens agricoles et de services.

**Agro-Hydrosystème** : Hydrosystèmes sous influence de l'activité agricole.

**Bioagresseur** : Les bioagresseurs, appelés aussi « ennemis des cultures », sont des organismes vivants qui attaquent les plantes cultivées et sont susceptibles de causer des pertes économiques.

**Canopée** : La canopée est la zone d'une forêt qui correspond à la cime des grands arbres.

**Cartographie génétique** : La carte génétique est la représentation graphique de la position des gènes les uns par rapport aux autres sur un génome.

**Écosystème** : En écologie, un écosystème est un ensemble formé par une communauté d'êtres vivants en interaction (biocénose) avec leur environnement (biotope). Les composants de l'écosystème développent un dense réseau de dépendances, d'échanges d'énergie, d'information et de matière permettant le maintien et le développement de la vie.

**Génome** : Le génome est l'ensemble des chromosomes et des gènes (d'une espèce, d'un individu).

**Génotype** : Le génotype est le patrimoine héréditaire (d'un individu) dépendant de l'ensemble des gènes.

**Hydrodynamique** : L'hydrodynamique est l'étude des fluides en mouvement.

**Hydrogéologie** : L'hydrogéologie (de *hydro-*, eau et *géologie*, étude de la terre), également nommée hydrologie souterraine et plus rarement géohydrologie, est la science qui étudie l'eau souterraine. Son domaine d'étude repose essentiellement sur deux branches des sciences de la Terre, la géologie et l'hydrologie, mais aussi sur de nombreuses autres branches comme la géostatistique, la physique, la chimie, la biologie, la géochimie, l'hydrochimie, la géophysique, l'hydrodynamique, l'hydraulique souterraine, l'analyse numérique ainsi que des techniques de modélisation.

**Hydrogéophysique** : L'hydrogéophysique est l'étude des écoulements d'eau souterraine par de méthodes géophysiques.

**Géophysique** : La géophysique est l'étude des propriétés physiques du globe terrestre (mouvements de l'écorce, magnétisme terrestre, électricité terrestre, météorologie).

**Hydrosystème** : Ensemble hydrologique complexe associé à son environnement.

**Inter-nœuds** : Le nœud est le point d'attache d'une feuille ou d'un rameau sur la tige chez les plantes à graines. L'inter-nœud se situe entre deux nœuds successifs.

**Intelligence artificielle** : L'intelligence artificielle est l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine.

**Microsatellite** : Un microsatellite est une portion du génome constituée de petites séquences hautement répétitives de deux à dix nucléotides.

**Nucléotide** : Les nucléotides sont constitués de 3 éléments principaux : un groupe phosphate (groupe composé d'un atome de phosphore lié à quatre atomes d'oxygène [1]), d'un sucre à 5 carbones (pentose), le désoxyribose, et d'une base azotée. Il existe donc 4 types de nucléotides (la cytosine (C), la thymine (T), l'adénine (A) et la Guanine (G)) qui se lient les uns à la suite des autres et forment un polynucléotide. L'ADN est un polynucléotide.

**Segmentation sémantique** : La segmentation sémantique est le processus permettant de classifier chaque pixel d'une image en un label particulier. On ne peut faire de différence entre deux instances d'un même objet. Par exemple, si on a deux voitures sur une image, ce type de segmentation donnera le même label sur l'ensemble des pixels des deux voitures.

**Sélection** : La sélection peut être naturelle ou artificielle. La sélection naturelle provient d'une théorie de Darwin sur l'évolution, selon laquelle l'élimination naturelle des individus les moins aptes dans la « lutte pour la vie » permet à l'espèce de se perfectionner de génération en génération. La sélection artificielle est une forme de sélection adaptative régulée par un opérateur extérieur « conscient ».

**Téledétection** : La téledétection est l'ensemble des sciences et techniques de la détection à distance.

## Table des matières

Remerciements .....	3
Lexique .....	4
1. Introduction.....	7
1.1 Structure d'INRAE .....	7
1.2 La sélection chez le pêcher et l'abricotier .....	9
1.3 Le phénotypage classique et digital.....	10
1.4 État de l'art des méthodes imagerie utilisées pour l'estimation de la vigueur 13	
1.5 Les objectifs de mon stage .....	20
2. Matériel.....	20
2.1 Moyens informatiques .....	20
2.2 Données expérimentales .....	21
3. Méthodes.....	26
3.1 Approche basée sur la simulation.....	26
3.2 Annotation des données RGB .....	30
3.3 Entraînement des réseaux .....	31
4. Résultats et discussion.....	37
4.1 Approche basée sur les simulations .....	37
4.2 L'entraînement des réseaux de neurones .....	39
4.3 Comparaison des estimations de circonférences .....	42
5. Conclusion.....	47
Annexes.....	48
Annexe 1 : Librairies pythons utilisées durant le stage .....	48
Annexe 2 : Tableau récapitulatif des sous-jeux de données d'images utilisés pour l'entraînement des réseaux de neurones.....	50
Annexe 3 : Tableau récapitulatif du jeu de données d'images à l'issue de l'active learning .....	51
Annexe 4 : Évolution de la précision lors du premier entraînement de réseau de neurones.....	52
Annexe 5 : Évolution de la précision lors du 2 <sup>ème</sup> entraînement de réseau de neurones.....	53
Annexe 6 : Évolution de la précision lors du dernier entraînement de réseau de neurones.....	54
Bibliographie.....	55

# 1. Introduction

## 1.1 Structure d'INRAE

INRAE (Institut national de recherche pour l'agriculture, l'alimentation et l'environnement) est créé le 1<sup>er</sup> janvier 2020, issu de la fusion entre l'INRA (Institut National de la Recherche Agronomique) et l'IRSTEA (Institut National de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture).

L'INRA, ancêtre d'INRAE, fut créé en 1946, au lendemain de la Seconde Guerre mondiale pour répondre à une question : « Comment nourrir la France ? » à une époque où le pays souffrait de pénurie alimentaire. La mission de l'INRA était alors d'associer science et technologie pour améliorer les techniques d'agriculture et de l'élevage en France, à une époque où la France était en retard par rapport à d'autres grands pays.

L'IRSTEA, deuxième institut à l'origine de la création d'INRAE, fut créé en 1982 sous le nom de CEMAGREF (Centre national du Machinisme Agricole du Génie Rural, des Eaux et des Forêts), puis la structure a changé de nom en 2012 pour IRSTEA. L'IRSTEA (Institut National de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture) avait pour domaines d'activités : l'environnement, la gestion de l'eau, l'éco-technologie, les risques naturels et les sciences économiques, humaines et sociales.

INRAE a pour missions de « réaliser, organiser et de coordonner, à son initiative ou à la demande de l'Etat, tous les travaux de recherche scientifique et technologique dans les domaines de l'agriculture, de l'alimentation, de la forêt, de l'environnement, de l'eau, de la biodiversité, de la bioéconomie, de l'économie circulaire, de la gestion durable des territoires et des risques dans les champs de compétence précités. En 2020, l'institut comptait un peu plus de 11 000 personnes, avec un peu plus de 200 unités de recherche, une quarantaine d'unités expérimentales implantées dans 18 centres [2]. Et l'entreprise a réalisé un chiffre d'affaire d'environ 31 millions d'euros en 2020.

INRAE est organisé en plusieurs départements de recherche comme Mathématiques et numérique, Biologie et amélioration des Plantes ou Agroécosystèmes [3]. Je réalise mon stage au sein des départements Biologie et amélioration des Plantes et Agroécosystèmes, sur le centre Provence-Alpes-Côte d'Azur (PACA, Figure 1).

Sur ses 18 centres, INRAE est engagé dans 33 sites universitaires

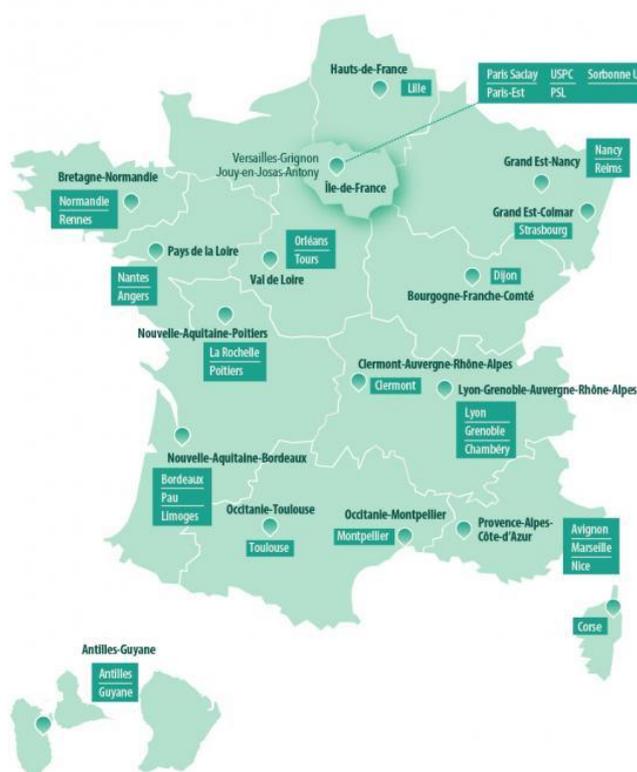


Figure 1. Carte des différentes implantations d'INRAE en France [4]

Les départements et centres sont subdivisés en plusieurs unités. Ces unités regroupent des chercheurs travaillant sur les mêmes thématiques. Mon stage s'est effectué à l'interface entre les unités GAFL qui exprime les besoins (phénotyper des vergers d'arbres fruitiers) et EMMAH qui propose les solutions (notamment grâce à l'utilisation de capteurs et l'analyse d'images). Je suis resté 3 mois dans l'unité GAFL puis j'ai déménagé dans les bureaux de l'unité EMMAH.

### 1.1.1 Structure du laboratoire GAFL

L'Unité de Recherche (UR) GAFL, développe ses recherches selon quatre axes thématiques :

- diversité des espèces et processus de domestication ;
- bases génétiques et moléculaires de la qualité des fruits ;
- caractérisation fonctionnelle des interactions plantes/pathogènes,
- intégration de la résistance aux maladies et de la qualité des fruits dans des innovations variétales.

L'Unité est aussi un centre de ressources génétiques pour les légumes de la famille des solanacées (aubergine, piment, tomate), les cucurbitacées, la laitue, et antenne du CRB-Prunus (Centre de Ressources Biologiques Prunus) d'INRAE [5].

L'UR GAFL est composé de deux équipes :

1. l'équipe DADI (Diversité, Adaptation, Déterminants et Intégration) dans laquelle j'ai effectué la première partie de mon stage. Cette équipe s'intéresse à la tomate et aux principaux fruits à noyau méditerranéens qui sont l'abricot

(*Prunus armeniaca*), la pêche (*Prunus persica*) et leurs porte-greffes. L'équipe développe des études allant de la recherche fondamentale au développement de nouvelles variétés en partenariat avec le secteur privé, pour participer à la conception d'une agriculture durable ;

2. l'équipe ReDD (Résistance aux pathogènes et aux ravageurs, Diversité et Durabilité) qui a pour but de caractériser de nouveaux gènes cibles afin de développer chez les plantes des résistances génétiques aux pathogènes et aux ravageurs, mais aussi de mieux comprendre les interactions entre les plantes et ses bioagresseurs, en vue d'améliorer la durabilité de ces résistances génétiques [6].

### 1.1.2 Structure du laboratoire EMMAH

EMMAH est une Unité Mixte de Recherche (UMR) entre l'Université d'Avignon et des Pays de Vaucluse (UAPV) et INRAE. Elle fait partie du centre de recherche régional PACA d'INRAE. L'UMR regroupe des agents du département Agroécosystèmes d'INRAE, des départements d'Hydrogéologie et de Physique de l'UAPV.

Ses recherches portent d'une part sur l'analyse de l'impact des changements globaux sur la ressource hydrique, la production agricole et leur interaction à l'échelle du territoire (du paysage au bassin de production et de l'aquifère), et d'autre part sur l'identification de voies d'adaptation de l'agriculture à ces changements. Au sein des changements globaux, l'accent est mis sur les changements climatiques et les changements d'usage des terres. Un focus particulier est porté au milieu méditerranéen [7]. L'unité est divisée en 5 équipes :

1. l'équipe Capteurs et Télédétection (CAPTE) dans laquelle j'ai effectué la deuxième moitié de mon stage. Cette équipe est Laboratoire Partenaire Associé (LPA) avec l'entreprise Arvalis (Institut technique agricole français qui a pour objectif d'aider les producteurs agricoles à résoudre les problèmes sociétaux et environnementaux). L'équipe teste, adapte, et développe éventuellement des capteurs fixes pour des mesures continues dans le temps, portatifs, ou bien fixé sur différents vecteurs comme les tracteurs, les drones, les avions, ou encore les satellites. Le LPA développe également les méthodes d'interprétation des données acquises par ces capteurs en intégrant toute la chaîne allant de la mesure à son utilisation dans les différentes applications [8].
2. l'équipe Dynamique des Interaction dans les Sols Contaminants, Organismes Vivants, Eau (DISCOVE).
3. l'équipe Dynamique des territoires, fonctionnement des Espaces Agricoles et Modélisation (DREAM).
4. l'équipe d'Hydrogéologie (HYDRO).
5. l'équipe Soil Water Interactions and transFer Team (SWIFT).

### 1.2 La sélection chez le pêcher et l'abricotier

En botanique, les abricotiers et pêchers sont des arbres du genre *Prunus* qui appartiennent à la famille des Rosacées, dont beaucoup sont cultivées pour leurs

fruits (abricotier, amandier, cerisier, pêcher, prunier) ou pour leur valeur ornementale (cerisier du Japon, cerisier de Virginie, laurier-cerise...) [9].

L'équipe DADI du GAFL contribue aux activités des sélectionneurs, qui développent de nouvelles variétés de pêchers et abricotiers, en mettant au point des outils de diagnostic génétique, qui s'appuient sur des marqueurs génétiques en faisant appel à la cartographie génétique et à des techniques d'analyse spécifiques des caractères quantitatifs (comme le poids ou la longueur de rameaux par exemple) ou qualitatifs [10]. Pour ce faire, l'ADN des individus est extrait à partir d'une simple feuille. Le génotypage consiste à « lire » des milliers de marqueurs génétiques (SNP –Single Nucleotide Polymorphism, ce sont des variations mineures du génome, seul un nucléotide est modifié) répartis sur l'ensemble du génome des individus. Pour établir ces méthodes, il est indispensable de phénotyper et génotyper des populations d'individus.

### 1.3 Le phénotypage classique et digital

Le phénotypage est la détermination de l'ensemble des caractères apparents d'un individu (comme la couleur des fruits ou la date de floraison de l'arbre) [11]. On développe aujourd'hui des méthodes de phénotypage digital. Le phénotypage digital est un domaine pluridisciplinaire défini pour la première fois en 2016 comme la quantification instantanée du phénotype au niveau individuel à l'aide de données provenant d'appareils numériques [12]–[14].

Parmi les avantages du phénotypage digital, il est plus rapide, plus précis et plus facilement répétable. De plus, le phénotypage digital ne confronte pas les mesures au biais du notateur, qui peut exister. Le phénotypage digital permet aussi d'assurer un suivi dans le temps de chaque arbre de façon non-destructrice.

Le phénotypage digital des grandes cultures comme le maïs ou le blé est maintenant utilisé de façon opérationnelle. Cependant, c'est moins le cas pour les arbres fruitiers, car cela demande des dispositifs plus complexes et adaptés. En effet, ce sont des milieux plus hétérogènes autant spatialement que verticalement [15]. Pour phénotyper les arbres fruitiers, on utilise notamment des drones et des acquisitions au sol avec des approches comme le LiDAR (Light Detection and Ranging) [16] ou l'imagerie RGB (Red Green Blue) [17]. On utilise ces méthodes pour observer des traits comme l'architecture de l'arbre [18].

#### 1.3.1 Le phénotypage de la vigueur

La vigueur est un trait phénotypique qui suscite beaucoup d'intérêt chez les arbres fruitiers. La vigueur est notamment un trait recherché chez les porte-greffes (souvent des amandiers) car ce sont eux qui apportent la vigueur aux espèces d'arbres greffées [19]. On cherche parfois à corrélérer la vigueur avec la capacité de résistance aux stress climatiques [20]. Au GAFL, on souhaite établir un lien entre ce trait et la résistance aux maladies : on peut notamment se demander si la vigueur est corrélée ou anti corrélée à la résistance aux maladies [21]. Différents indicateurs permettent de mesurer ce trait comme le nombre d'inter-nœuds, la hauteur de l'arbre, le diamètre de tronc à la base de l'arbre, ou encore le diamètre de la branche au sommet de l'arbre [22]–[25]. Au GAFL, on caractérise la vigueur en mesurant la croissance de la circonférence du tronc à la base de l'arbre [26]. On indique l'endroit

où on mesure à l'aide d'une bombe de peinture pour savoir où mesurer d'une année sur l'autre afin d'avoir un suivi. La circonférence d'un tronc de pêcher ou d'abricotier augmente d'environ 10 cm par an [27].

Des travaux préliminaires (stage de fin d'études de Marie-Pia d'ARGAIGNON) avaient déjà été réalisés au GAFL/EMMAH pour estimer la vigueur des arbres fruitiers. En effet, des images d'arbres avaient été acquises avec une perche LITERAL (voir §2.2.2.a)) mais la précision obtenue était encore insuffisante. Pour estimer la circonférence des troncs, on utilisait : un algorithme de segmentation du bois, des algorithmes de détection du tronc étudié (on détectait le tronc, puis on détectait une marque bleue préalablement réalisée à la bombe de peinture sur l'arbre, puis on extrayait les contours du tronc), un algorithme de stéréovision et un algorithme de calcul de la résolution de l'image au niveau du tronc (Figure 2) :

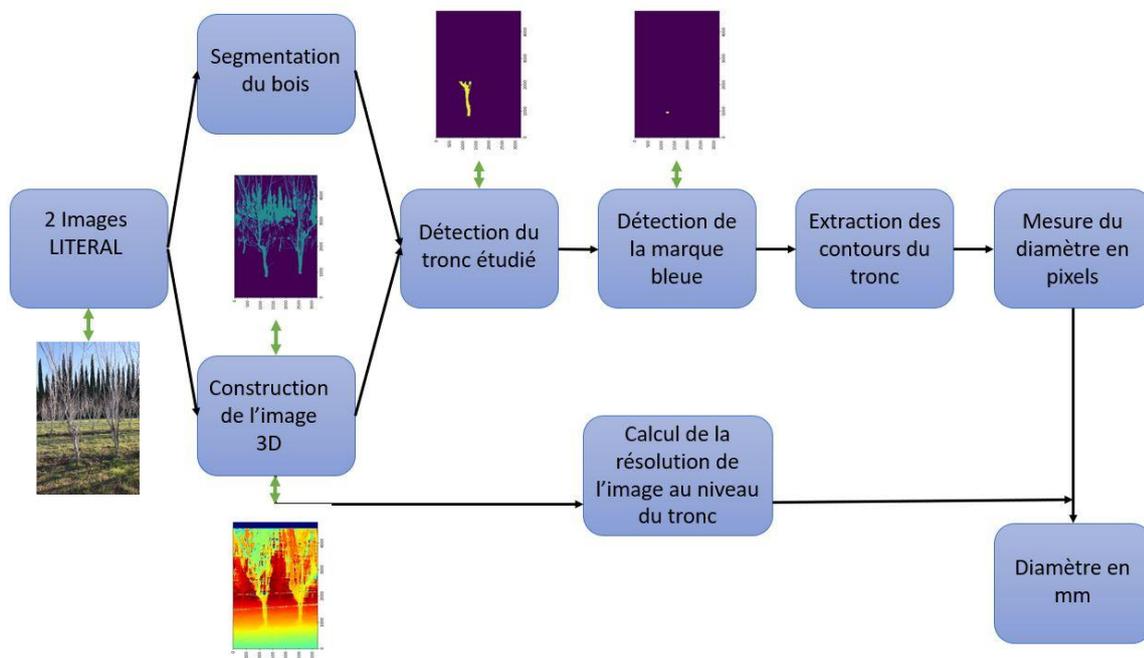


Figure 2. Pipeline pour l'estimation de la circonférence proposée par Sylvain JAY (2022)

Pour estimer la circonférence des troncs, on utilise un algorithme de stéréovision pour pouvoir associer un pixel d'image à une distance entre la caméra et l'objet de l'image grâce aux connaissances de l'ATS (Across Track Sensor Width), de la distance focale, de l'ATNP (Across Track Number of Pixels), de la résolution de la caméra et de l'ATFOV (Across Track Field Of Vue) (cf §3.1). Ensuite, on utilisait un algorithme de segmentation sémantique pour délimiter la partie composée de bois du reste de la sous-image sélectionnée. On couplait ensuite ces 2 images. Et enfin, on utilisait un algorithme d'imagerie pour isoler le tronc de l'arbre. L'estimation des circonférences d'arbre des vergers donne une RMSE (Root Mean Square Error) de 24 mm (Figure 3) ce qui est encore insuffisant puisqu'on souhaite obtenir une précision de l'ordre de 10 mm :

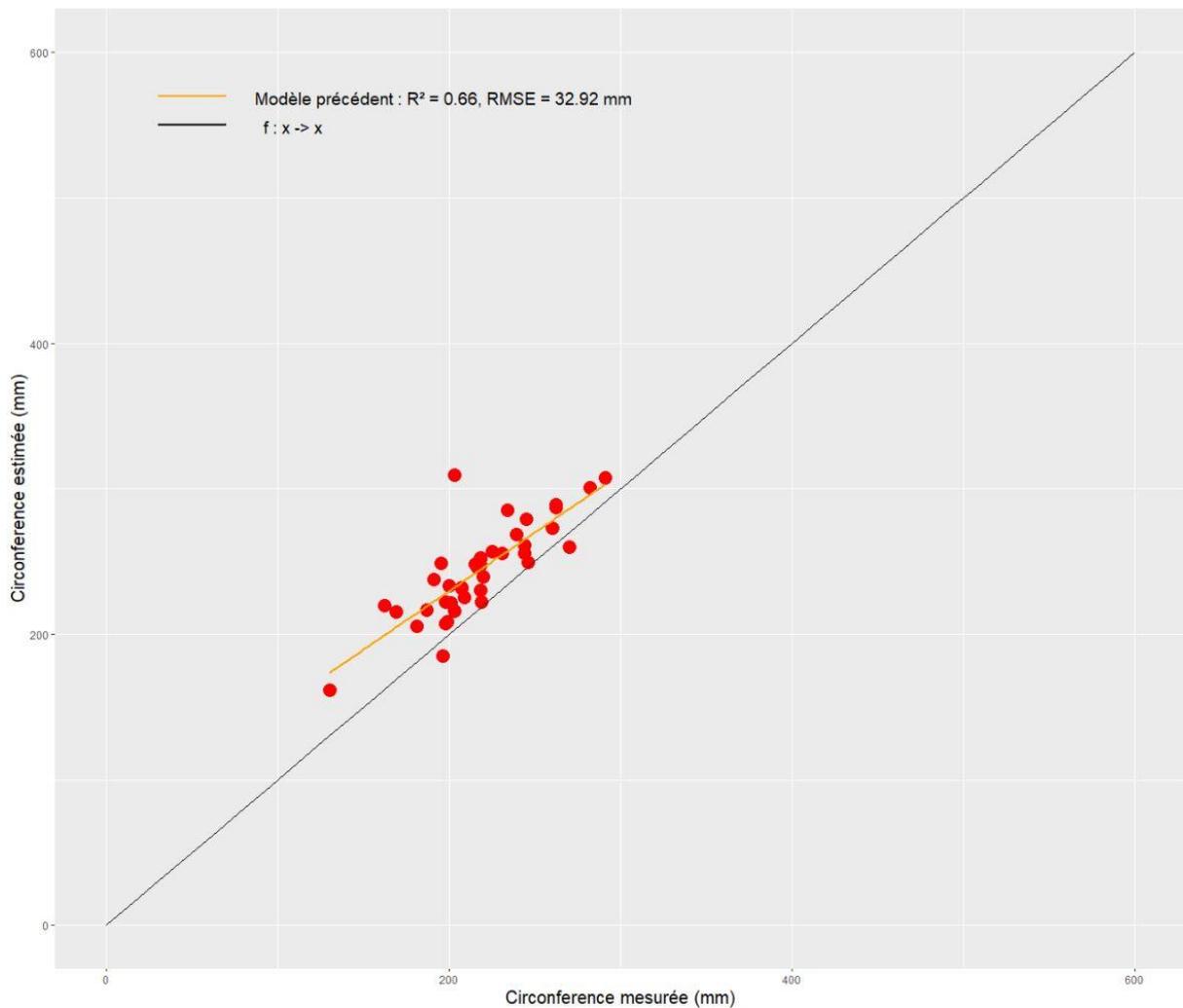


Figure 3. Régression linéaire des circonférences mesurées par le précédent modèle d'estimation de circonférence en fonction des circonférences mesurées en vergers.

Cette erreur est telle que, lorsqu'on étudie la croissance de tronc sur une période donnée de 6 mois (entre janvier et juillet), on n'obtient pas de corrélation entre l'estimation faite par des outils numériques et les mesures manuelles (Figure 4). On peut par ailleurs voir que les croissances estimées sont parfois négatives, ce qui montre que le modèle a besoin d'améliorations :

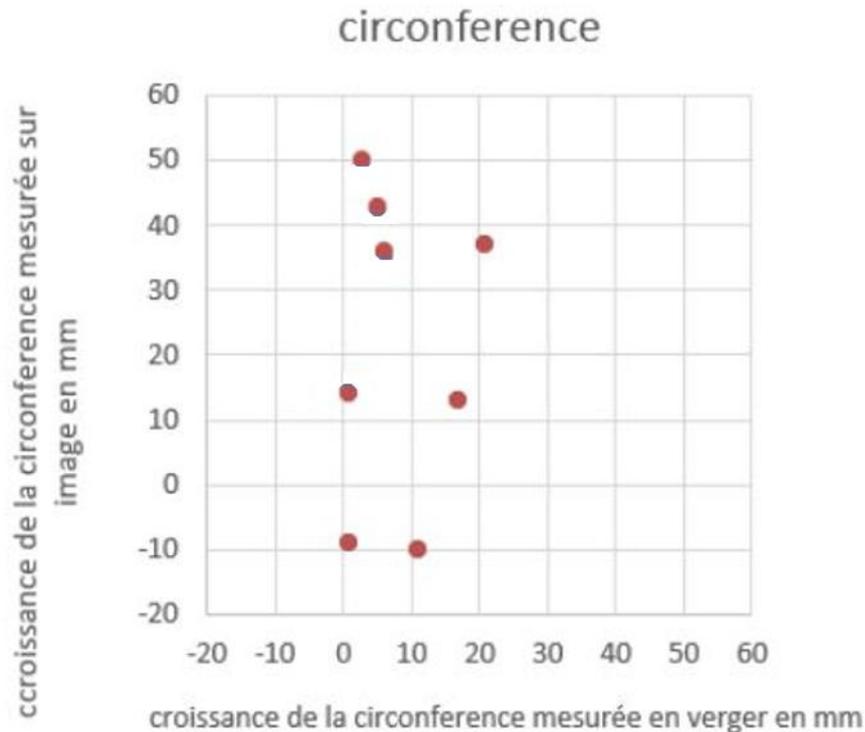


Figure 4. Graphique de croissances estimées par imagerie. Ce graphique représente les croissances de troncs estimées par le modèle en fonction des croissances de troncs mesurées en vergers.

## 1.4 État de l'art des méthodes imagerie utilisées pour l'estimation de la vigueur

### 1.4.1 La stéréovision

La stéréovision est le fait d'utiliser 2 images prises simultanément par système de 2 caméras et de déterminer de ce système, à partir des paramètres des caméras (distances focales, espacement et orientation des caméras), la distance entre les caméras et l'objet observé en chaque pixel. Sur la Figure 5, on voit qu'à partir de l'espacement ( $b$ ) entre les deux caméras ( $O_1$  et  $O_2$ ), la distance focale des caméras ( $f$ ) et la projection du point  $P$  sur les 2 images, on peut obtenir la distance entre l'axe des 2 caméras et le point  $P$  grâce à des relations trigonométriques.

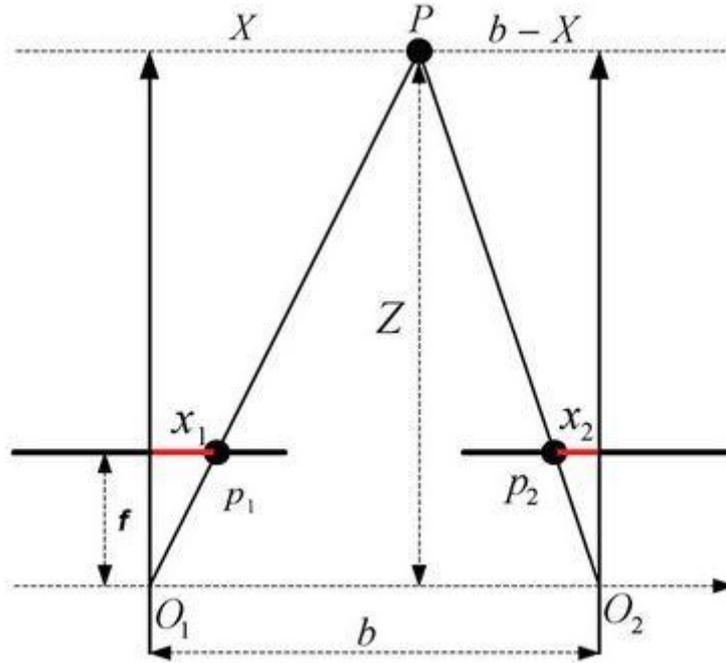


Figure 5. Schéma des variables utilisées dans les calculs de stéréovision [28].  $O_1$  est la 1<sup>ère</sup> caméra,  $O_2$  la 2<sup>ème</sup> caméra,  $f$  la distance focale,  $b$  la distance entre les 2 caméras, et  $p_1$  et  $p_2$  les positions la position de  $P$  (l'objet observé) sur les 2 images.

#### 1.4.2 La segmentation

Si on considère une image  $A$ , la segmentation est la création d'une partition de l'image  $A$  en sous-ensembles  $R_i$ , appelés régions, tels que :

$$\forall i, R_i \neq \emptyset$$

$$\forall i, j : i \neq j, R_i \cap R_j = \emptyset$$

$$A = \cup_i R_i \quad [29]$$

Il existe différentes manières de segmenter les images RGB, on peut notamment les segmenter avec des méthodes comme le Maximum de Vraisemblance [30], les SVM [31] ou encore random forest [32]. Cependant, l'inconvénient de ce type de méthode est qu'elle n'utilise que les valeurs de chaque pixel pour déterminer sa classe là où des algorithmes plus récents prennent aussi en compte les valeurs des pixels adjacents au pixel considéré pour déterminer sa classe. C'est notamment pour cela que le choix fait pour ce stage a été de s'orienter vers la segmentation par Deep Learning qui est omniprésente dans la littérature récente [33]–[35].

#### 1.4.3 Le Machine Learning

Le Machine Learning est un type d'intelligence artificielle qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés. La majeure partie des techniques de Machine Learning suivent la même structure (Figure 6) :

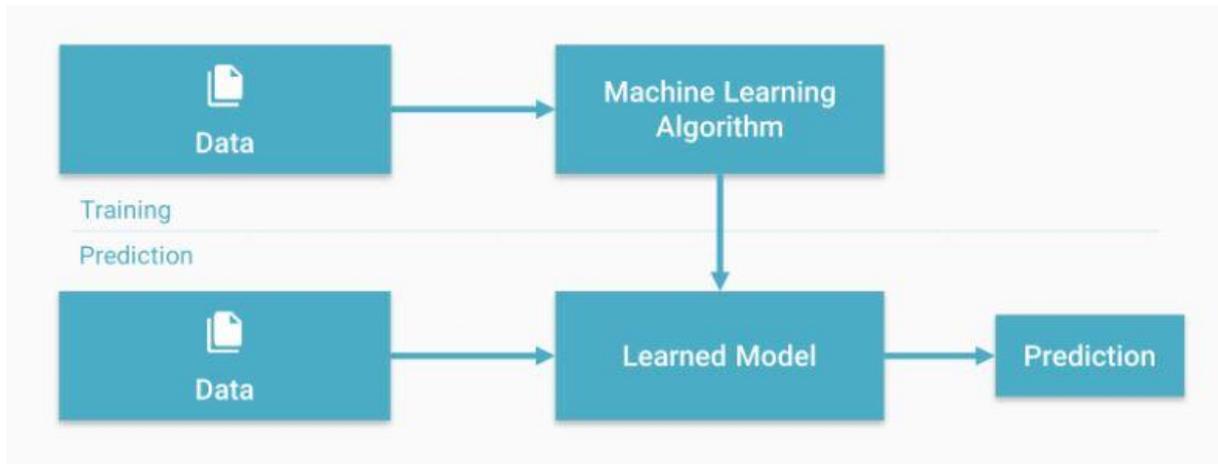


Figure 6. Schéma de fonctionnement des algorithmes de Machine Learning

Le Deep Learning est une sous-partie du Machine Learning qui utilise des algorithmes d'apprentissage dont les architectures sont composées de nombreuses couches qui imitent les réseaux de neurones humains.

Durant ce stage, j'ai utilisé le Deep Learning pour de l'imagerie. J'ai notamment entraîné des réseaux de neurones convolutifs. L'idée était de développer un algorithme capable de prédire le phénotype des différents individus à partir d'images préalablement acquises en vergers.

#### 1.4.4 Les réseaux de neurones convolutifs

Aujourd'hui, lorsqu'on a besoin de réaliser de la segmentation sur des images, on fait souvent appel à des réseaux de neurones convolutifs. Dans les réseaux de neurones utilisés pour le traitement d'images, on retrouve notamment les réseaux de neurones convolutifs comme par exemple U-Net [36], qui est notamment beaucoup utilisé en imagerie biomédicale [37]. Le réseau est composé de couches de convolution 3x3, de couches max pool et de up-convolution 2x2.

Ce réseau est composé de 23 couches convolutives [36] (Figure 7). Le réseau est composé d'une partie contractante (partie gauche de la Figure 7, aussi appelée « encoder ») et d'une partie expansive (partie droite de la Figure 7 aussi appelée « decoder ») ce qui lui confère la forme de « U » :

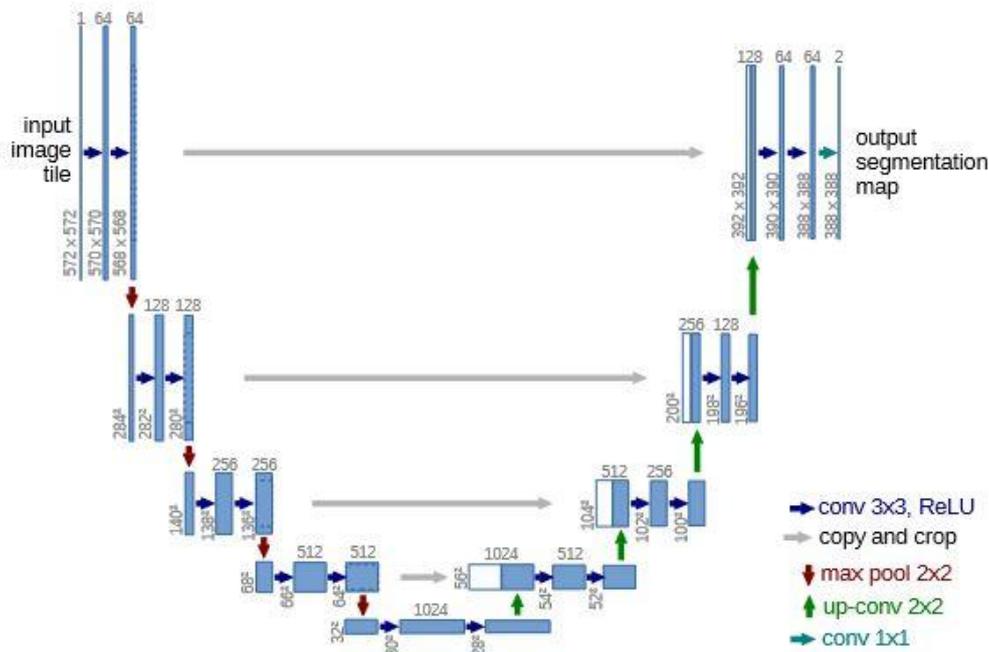


Figure 7. Architecture du réseau U-Net. Chaque rectangle bleu correspond à une matrice à plusieurs canaux. Le nombre de canaux est noté en haut de chaque rectangle. Les nombres de lignes et de colonnes de chaque matrice sont notés en bas à gauche de chaque rectangle [36].

Comme le montre la Figure 7, le réseau est composé de couches de convolution 3x3, de couches max pool et de up-convolution 2x2.

Lors d'une convolution, des matrices de poids appelées features sont superposées au voisinage de chaque pixel. Une matrice différente est superposée au voisinage de chaque pixel. Pour illustrer plus simplement le procédé, par la suite, on considèrera qu'une unique feature de convolution est appliquée à l'ensemble des voisinages de pixels de l'image. À chaque superposition, les poids de la features sont multipliés 2 à 2 avec les valeurs de pixel de l'image (Figure 8). Ensuite, les produits sont sommés, et enfin, cette somme est divisée par le nombre de valeurs de la matrice (qui est aussi le nombre de pixels du voisinage considéré).

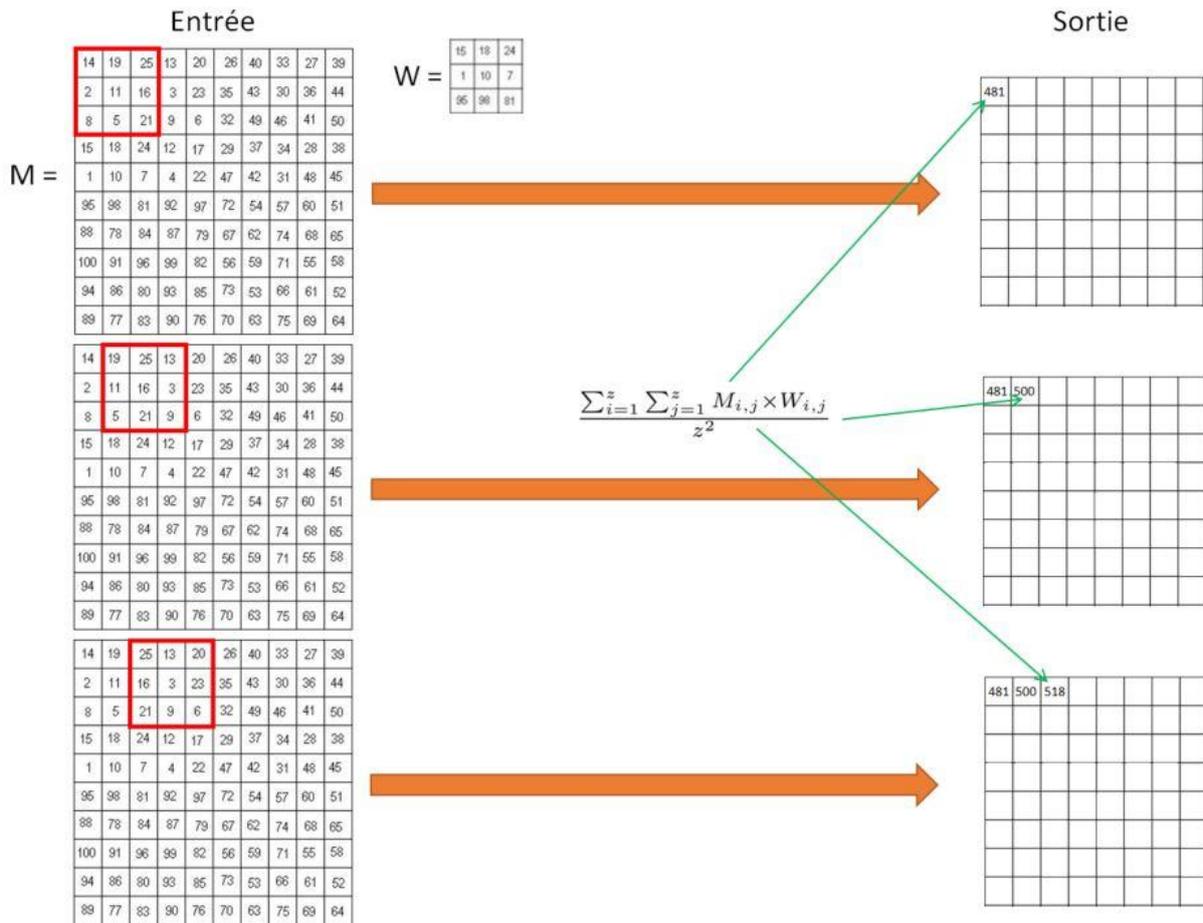


Figure 8. Illustration du procédé de convolution en imagerie. Sur la figure, le noyau est de taille  $z = 3$  [38].

On remarque que l'image en sortie est de dimension  $n-2$  (où  $n$  est la dimension de l'image en entrée). Ce n'est pas toujours le cas, parfois, on rajoute des pixels en bordure d'image avant d'y appliquer la convolution pour que l'image en sortie ait les mêmes dimensions que l'image en entrée.

Le max-pool est une opération de réduction des images. Comme pour la convolution, on considère un « noyau ». Cette fois, à chaque noyau en entrée, on associe un coefficient en sortie qui correspond à la valeur maximale des pixels du noyau en entrée. Cette fois, il n'y a pas de chevauchement (pixels utilisés pour plusieurs noyaux distincts) (Figure 9) :

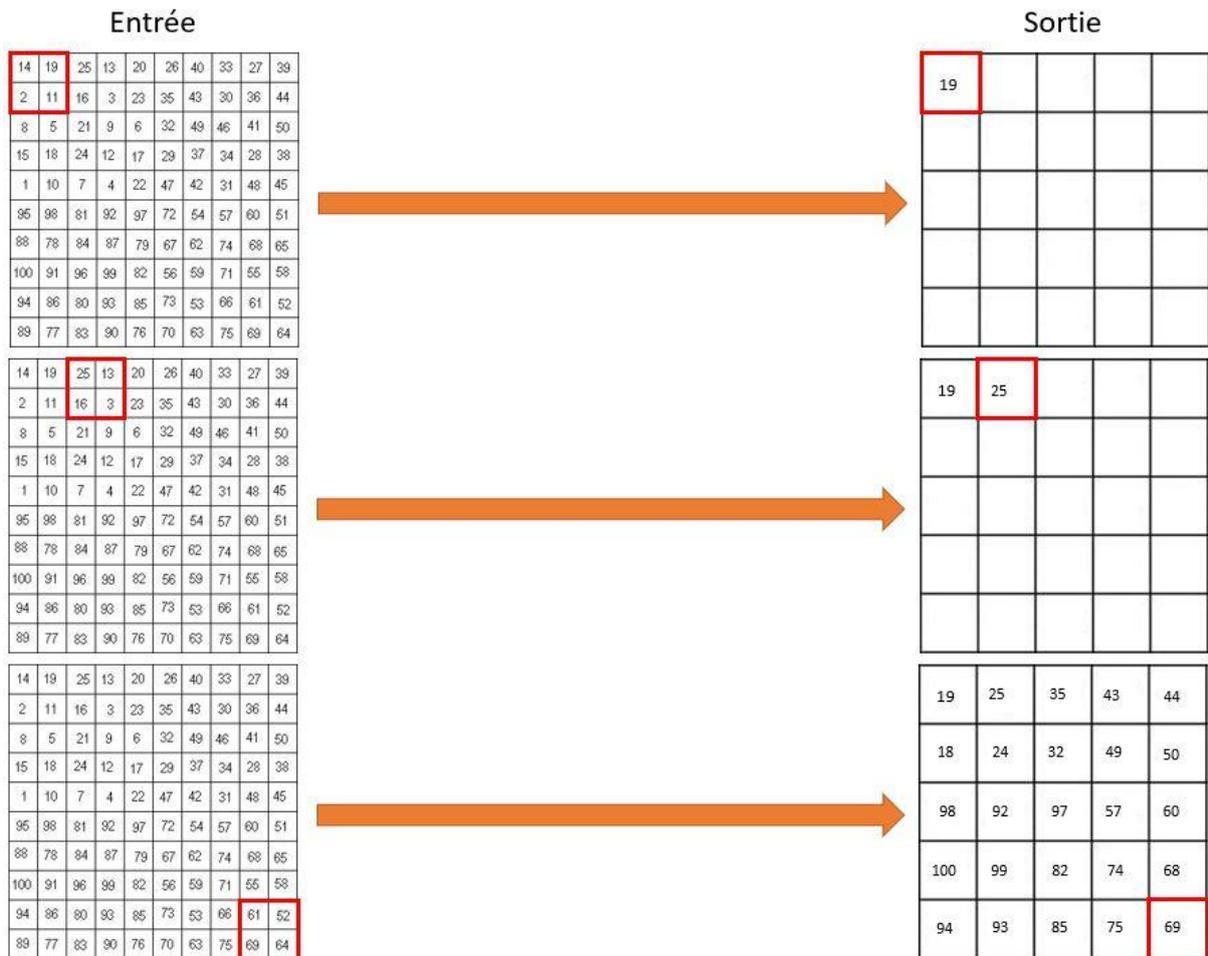


Figure 9. Illustration du procédé de max-pool. Sur la figure, le noyau est de taille  $z = 2$ .

On remarque que pour un max-pool  $2 \times 2$  (taille du noyau), pour une image de taille  $l \times l$  en entrée, on a une image de taille  $\frac{l}{2} \times \frac{l}{2}$ .

Pour finir, la up-convolution est une opération d'augmentation d'images. C'est une sorte d'opération inverse du max-pool : on considère un noyau (le plus souvent de taille  $2 \times 2$ ) et pour chaque pixel en entrée, on obtient un noyau de  $4 (= 2 \times 2)$  fois ce même pixel en sortie (Figure 10) :

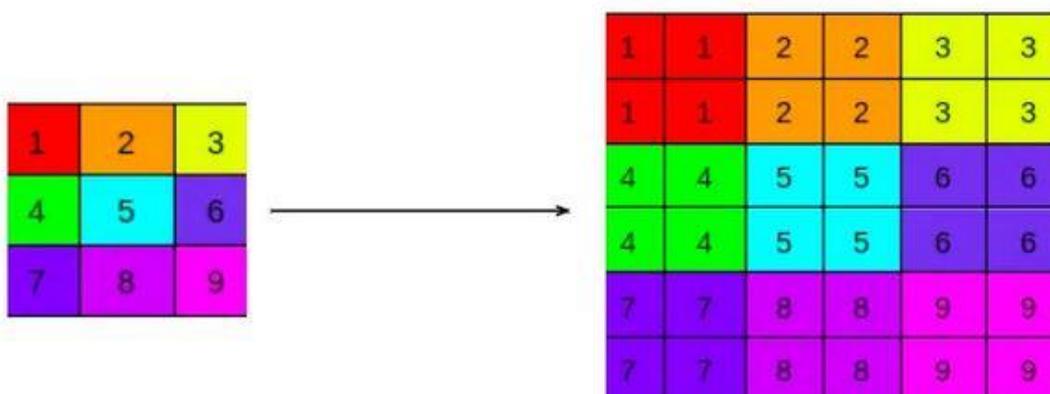


Figure 10. Illustration du procédé de up-convolution. Sur la figure, le noyau est de taille  $z = 2$

Le réseau U-Net n'est pas forcément utilisé sous sa forme originelle, puisqu'il est aussi possible de changer son ossature et d'utiliser l'un des nombreux réseaux de neurones de classification existant à la place l'encoder du réseau. U-Net a été notre choix pour le réseau de neurones car : c'est un réseau simple en terme d'architecture, il ne contient pas de bloc convolutif complexe ; son architecture est relativement simple à comprendre comparée aux autres réseaux de segmentation ; les performances de U-Net sont tout à fait compétitives par rapport aux autres réseaux de neurones [39]. La partie expansive du réseau étant symétrique à la partie contractante, elle devra être adaptée.

En imagerie biomédicale, la partie expansive est souvent remplacée par le réseau resNet50 [40] (Figure 11) :

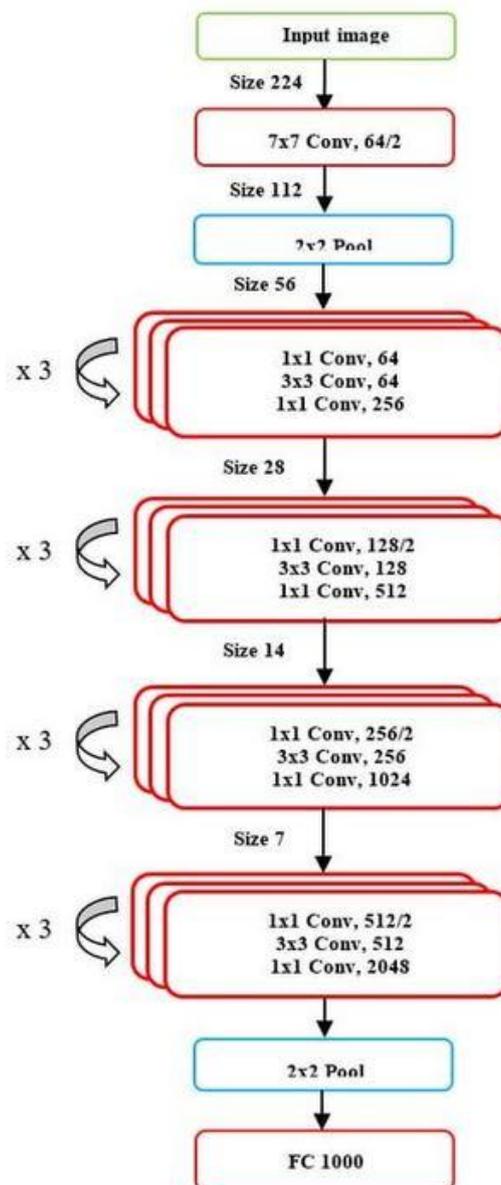


Figure 11. Architecture de réseau resNet50.

Le grand intérêt à reprendre l'architecture d'un réseau de classification pour la partie expansive de U-Net est que cela permet d'importer les paramètres d'entraînement du

réseau de neurones déjà entraîné. On procède généralement par « transfer-learning », cela consiste à utiliser les paramètres d'un réseau de neurones déjà entraîné. En effet, en procédant ainsi, on a besoin d'entraîner le réseau de neurones beaucoup moins longtemps et sur moins de données avant d'avoir des résultats corrects. L'architecture du réseau Resnet50 a été choisie, car c'est un réseau de classification très performant, notamment plus performant que des réseaux comme Alexnet ou GoogleNet [41].

### 1.5 Les objectifs de mon stage

Dans le but de contribuer au phénotypage digital des arbres fruitiers, l'objectif de mon stage consistait à améliorer la méthode de suivi de la circonférence de tronc, permettant le suivi de la vigueur, par stéréovision RGB avec l'instrument LITERAL (voir §2.2.2.a)). Il s'est déroulé en deux grandes étapes :

1. Déterminer quelle étape de l'algorithme décrit dans la Figure 2 a l'impact le plus significatif sur la précision d'estimation de la circonférence de tronc. Les deux étapes cruciales sont l'algorithme de segmentation du bois sur l'image RGB, et l'algorithme permettant de calculer les distances réelles à partir des images par stéréovision. Ce sont par conséquent deux sources d'erreurs non-négligeables. J'ai cherché à quantifier les sources d'imprécision des estimations à l'aide de simulations.
2. Proposer des méthodes d'amélioration pour l'algorithme déterminé précédemment : j'ai développé un nouveau modèle impliquant l'acquisition de nouvelles données après avoir identifié les algorithmes qui posaient problème (cf §3.1). J'ai de plus participé à cette acquisition à l'aide d'une perche LITERAL et d'une caméra RGB (cf §2.2.2.a))
3. Enfin, j'ai comparé les nouvelles estimations aux valeurs de terrain pour juger de la pertinence du phénotypage pour les traits de vigueur et l'amélioration par rapport aux résultats obtenus dans l'étude précédente.

## 2. Matériel

### 2.1 Moyens informatiques

Durant mon stage, j'ai travaillé sur un ordinateur muni d'une carte graphique NVIDIA GeForce RTX 3070 pour les entraînements de réseaux de neurones. Pour ce qui est de la programmation et des calculs, j'ai utilisé le langage Python avec des versions allant de la version 3.6 [42] à la version 3.11 [43] via divers environnements virtuels. Dans ce cadre, j'ai dû me familiariser avec le langage et utiliser différentes bibliothèques (détaillées en Annexe 1 : Bibliothèques pythons utilisées durant le stage).

Pour organiser mes données, j'ai utilisé le logiciel Excel combiné avec R version 4.2.3 (via l'interface RStudio version 2023.03.0 +386). J'ai aussi utilisé les packages suivants :

- dplyr : permet de concaténer des dataframes [44]
- leaflet : sert à créer des cartes interactives sur R [45]
- ggplot2 : permet de tracer des graphiques plus esthétiques [46]

- stringr : permet de manipuler simplement les chaînes de caractères [47]

La gestion des données et des tableaux est aussi possible via python, mais l'environnement R est bien mieux adapté à ces tâches.

## 2.2 Données expérimentales

### 2.2.1 Sites expérimentaux

Des photos étaient nécessaires pour l'entraînement d'un réseau de neurones à la segmentation. Elles ont été prises sur les sites de la SEFRA (Station d'Expérimentation de FRuits d'Auvergne-Rhône-Alpes) (26), à INRAE UERI Gotheron (26), à INRAE GAFL site des Garrigues (84), au domaine de St Paul d'INRAE (84), au CTIFL de Balandran (30) et à INRAE UE A2M site de l'Amarine (30) (Figure 12) :

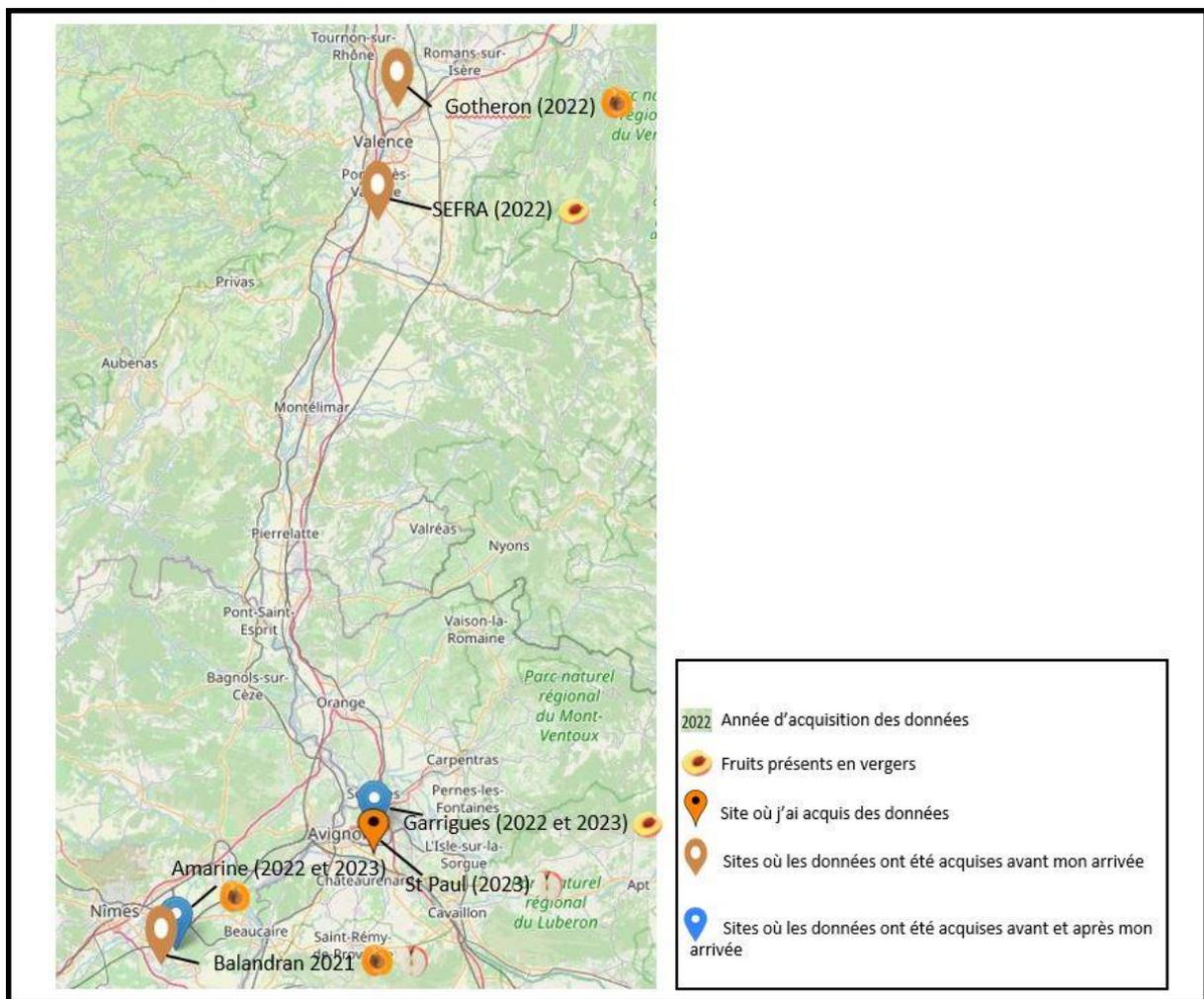


Figure 12. Carte des sites sur lesquels les photos des jeux de données ont été prises [45].

### 2.2.2 Matériel expérimental et protocoles d'acquisition

#### 2.2.2.a) Le matériel d'acquisition

Lors de mes acquisitions en verger, j'ai utilisé une perche LITERAL développée par ARVALIS et INRAE (Figure 13). C'est une perche métallique, au bout de laquelle sont montées deux caméras permettant de faire de la stéréovision,

couplées à une tablette tactile (pour la gestion des prises de vues) et à une batterie.



Figure 13. Illustration du système LITERAL. A) acquisitions de mesures avec la perche LITERAL. B) Valise contenant le système LITERAL. C) boîtier d'acquisition des images en stéréo (cf §1.4)

Chaque acquisition génère deux photos qui permettent d'estimer la distance entre les objets dans l'image et les caméras à partir de l'espacement des deux appareils photo et des paramètres de calibration du système (cf 1.4).

Nous avons également utilisé une caméra Sony RX0 II ainsi qu'une bêche bleue et un trépied (Figure 14) pour les acquisitions dans les vergers du domaine St Paul. En effet, pour faciliter les annotations de bois, les photos ont été prises avec les arbres au premier plan et un fond bleu en arrière-plan (cf §3.2).

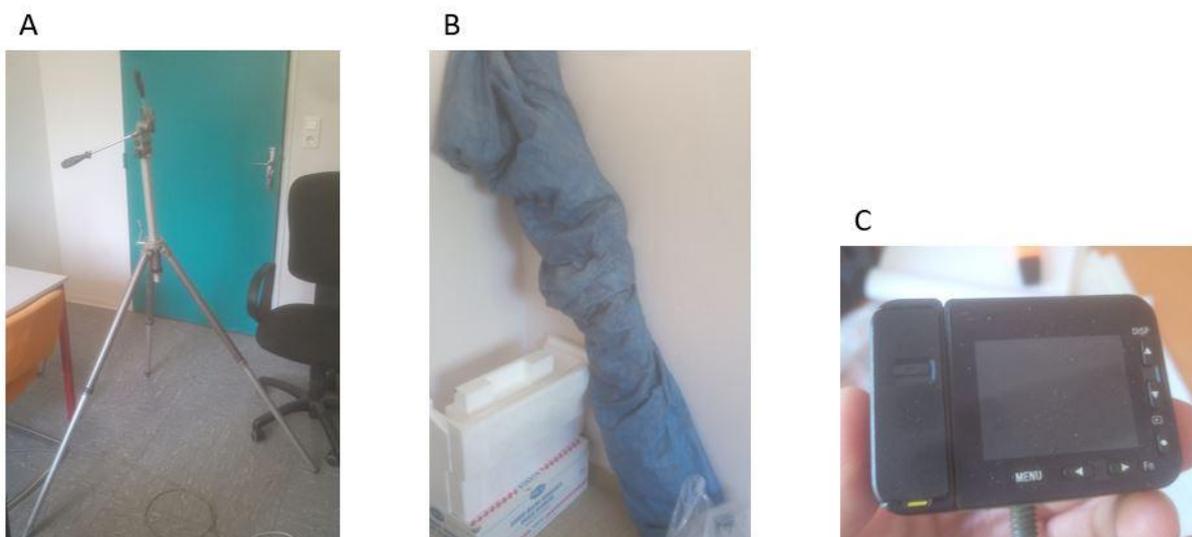


Figure 14. Dispositif d'acquisition utilisé dans le domaine Saint Paul.

#### 2.2.2.b) Le protocole d'acquisition des images d'arbres fruitiers

Les images acquises devaient essentiellement servir à entraîner des réseaux de neurones pour la segmentation du bois. Une partie des images ont été prises en

contre-plongée avec du ciel en fond, cette géométrie d'acquisition permettant de faciliter le traitement des images en éliminant la majorité des arbres situés sur les autres rangs à l'arrière-plan (Figure 15).



Figure 15. Exemple de photo prise en contre-plongée pour limiter le gêne due à l'arrière-plan.

Pour toutes les acquisitions que j'ai faites à l'exception de celles faites au domaine St Paul, le protocole était le suivant :

Au total, 8 acquisitions pour lesquelles il faut :

De chaque côté de la rangée d'arbres :

1. 3 acquisitions « fond ciel en contre-plongée » avec les angles de respectifs de 45, 90 et 135 degrés par rapport à direction du rang, et ce, de chaque côté du rang.

→ Placer l'objectif au minimum à 50 cm de l'arbre (à adapter au volume de l'arbre)

- Incliner la perche à  $45^\circ$  par rapport à l'axe horizontal ; incliner la perche à  $0^\circ$  par rapport à l'axe de la perche ; acquérir l'image.
- 2. 2 acquisitions perche « droite face à l'arbre » donc parallèle au tronc ( $90^\circ$  de tangage et  $0^\circ$  de roulis). Ici, l'objectif est de photographier la majeure partie de l'arbre.
  - Placer l'objectif à environ 2 m de l'arbre et à environ 1,5 m de hauteur (à ajuster en fonction de la taille de l'arbre).

Quant à l'acquisition au domaine de St Paul, elle anticipait l'annotation des images facilitée par le module Woodpicker (cf §3.2). Pour chaque arbre, deux photos étaient prises avec 5 secondes d'intervalle : une photo avec une bâche bleue pour cacher l'arrière-plan et une autre sans bâche bleue. Deux personnes tenaient la bâche bleue et une personne s'occupait de la caméra pour prendre les photos à l'aide d'un trépied et d'un compte à rebours (Figure 14 et Figure 16). Les photos étaient prises à une distance 5 mètres permettant d'imager la totalité de l'arbre sans inclinaison de caméra.

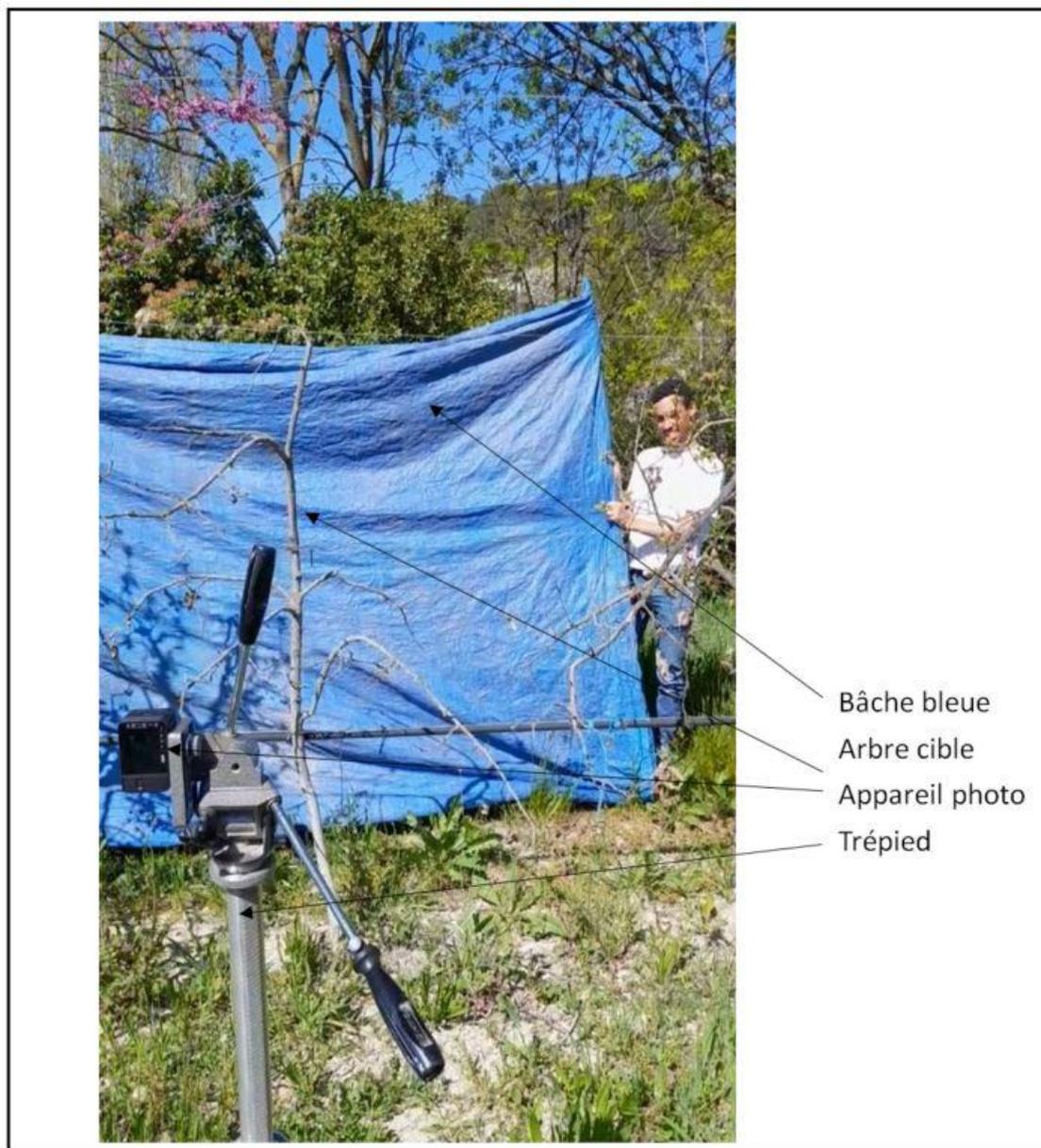


Figure 16. Acquisition d'images dans le domaine de St Paul.

### 2.2.3 Extraction des imagerie RGB utilisées pour l'entraînement du réseau de neurones

Les photos obtenues étaient de dimension 3200 x 4200 pixels. De ces photos, on a extrait des imagerie de taille 512 x 512 pixels, qui ont été utilisées pour entraîner le réseau de neurones.

À l'issue de toutes les extractions d'imagerie du processus d'annotation (cf 3.2), plus de 1300 imagerie étaient disponibles, incluant les données déjà disponibles avant mon stage et celles collectées pendant mon stage.

On pouvait de plus, définir des sous-jeux de données en fonction de la date à laquelle les photos avaient été prises. Pour chaque sous-jeu de données, on

connaissait aussi l'espèce d'arbres présente sur les photos, le type de scène, la couleur du ciel et le type d'éclairage (voir Annexe 2 : Tableau récapitulatif des sous-jeux de données d'images utilisés pour l'entraînement des réseaux de neurones) :

### 3. Méthodes

#### 3.1 Approche basée sur la simulation

Le modèle d'estimation des circonférences est essentiellement basé sur 2 algorithmes qui peuvent chacun être source d'erreur : un algorithme de segmentation et un algorithme de vision stéréo. Dans un premier temps, nous avons donc simulé l'impact de ces deux sources d'erreur à l'aide de la méthodologie décrite sur la Figure 17. Des circonférences (en cm) ont été simulées. Des distances entre caméra et arbre (en m) ont à leur tour été simulées ce qui a permis d'en déduire des diamètres (en pixels). Ensuite, des erreurs de segmentation ont été ajoutées ce qui a permis de simuler des diamètres estimés (en pixels). Enfin, des erreurs de distance entre caméra et arbre ont été ajoutées. Il est ainsi possible d'en déduire le diamètre du tronc et donc sa circonférence ( $circonférence = \pi \times diamètre$ ) en supposant le tronc circulaire.

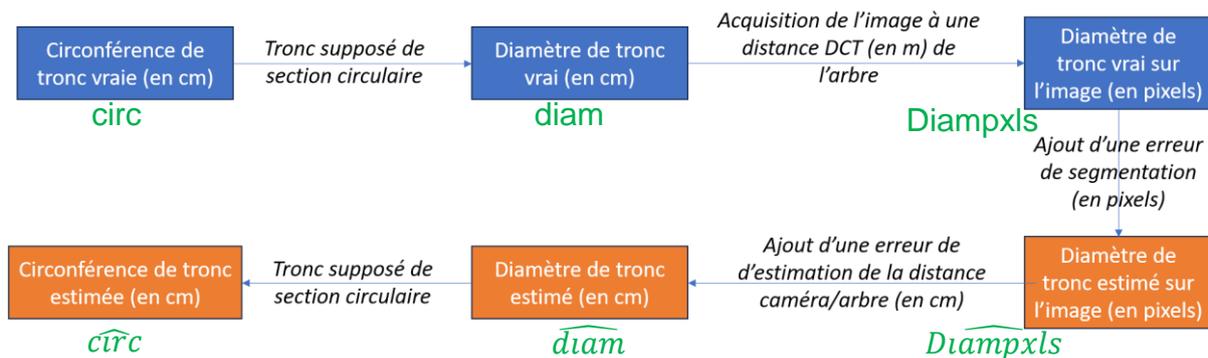


Figure 17. Description du procédé de simulation de données et d'erreurs d'estimation.

Afin de convertir un diamètre de tronc vrai imagé à une distance camera arbre (DCT) de la caméra, la méthode nécessite de calculer la taille d'un pixel (ou la résolution spatiale de l'image) à cette distance DCT en fonction des propriétés de la caméra (Figure 18)

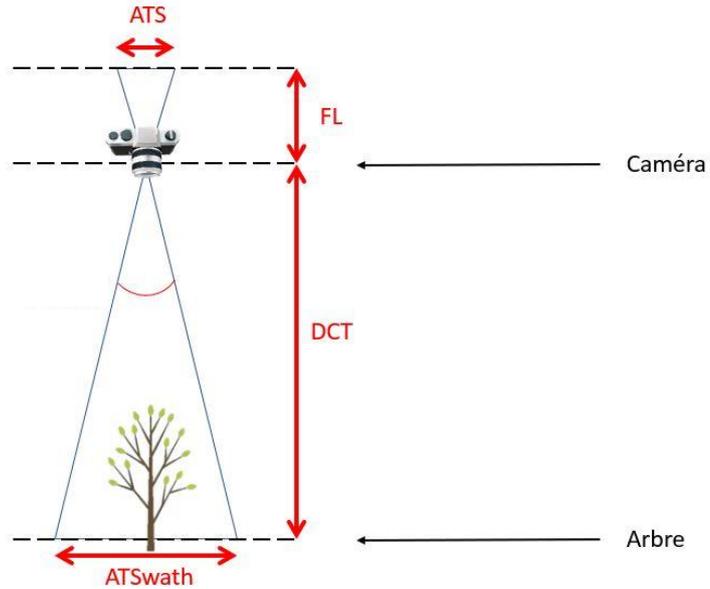


Figure 18. Schéma des différentes variables du calcul de conversion pixels/distance.

Pour cela, l'application du théorème de Thalès à la Figure 18 nous permet de calculer la fauchée selon le grand côté de l'image, à une distance DCT :

$$ATSwath(DCT) = \frac{ATS * DCT}{FL} \quad \text{Équation 1 ;}$$

La résolution spatiale de l'image (Reso) se déduit donc à partir de cette distance en cm et du nombre de pixels qui la représente :

$$Reso(DCT) = \frac{ATSwath(DCT)}{ATNP} \quad \text{Équation 2 ;}$$

Cette résolution spatiale nous permet ainsi de convertir le diamètre du tronc vrai en diamètre de tronc en pixels (Diampxls en pixels) :

$$Diampxls(diam, DCT) = \frac{diam}{Reso(DCT)} \quad \text{Équation 3 ;}$$

	Variable	Signification	Unité	Valeur	Fonction
Capteurs	ATS	Across Track Sensor Width, largeur du capteur CMOS (Complementary Metal-Oxide Semiconductor) de la caméra	cm	1.32	Entrée
	FL	Focal Length, distance focale	cm	1.79	Entrée
	ATNP	Across Track Number of Pixels, nombre de pixels sur le grand côté de l'image	sans dimension	4800	Entrée
	Reso	Resolution, taille d'un pixel dans la réalité pour un pixel situé à une distance donnée	cm		Sortie
Mesure	DCT	Distance Camera-Tree, distance caméra arbre	cm	50-250	Entrée
	ATSwath	Across Track Swath, largeur du champ de vision à une distance donnée	cm		Var. Inter.
Diamètre	diam	Diamètre du tronc	cm	8-40	Entrée
	Diampxls	Diamètre du tronc en pixels sur l'image	pxls		Sortie

Tableau 1. Tableau des variables nécessaires pour calculer les distances expérimentales en cm et en un nombre de pixels sur l'image. Pour la fonction de calcul, on distingue les entrées, les sorties et les variables intermédiaires calculées par la fonction.

Afin d'évaluer l'impact des erreurs de segmentation et d'estimation de la distance caméra/arbre par vision stéréo sur notre méthode d'estimation de la circonférence, nous avons tout d'abord défini un ensemble de couples réalistes circonférence de tronc / distance arbre-tronc ( $circ, DCT$ ) correspondant aux conditions expérimentales (Tableau 2).

	Min	Max	Pas d'éch.	Nb de valeurs
Circonférence de tronc ( $circ$ )	8cm	40cm	0.5cm	65
Distance caméra arbre ( $DCT$ )	50cm	250cm	25cm	9
Erreur sur la circonférence de tronc obtenue après segmentation ( $e_s$ )	-10 pixels	10 pixels	1 pixel	21
Erreur de mesure de la distance Arbre caméra ( $e_d$ )	-20cm	20cm	1cm	41

Tableau 2. Tableau de définition de la stratégie d'échantillonnage des données simulées.

Puis, après avoir converti les circonférences en diamètres, on a calculé la valeur valeur Diampxls pour chaque couple diam / DCT à partir de l' Cette résolution spatiale nous permet ainsi de convertir le diamètre du tronc vrai en diamètre de tronc en pixels (Diampxls en pixels) :

$$Diampxls(diam, DCT) = \frac{diam}{Reso(DCT)} \quad \text{Équation 3 ci-dessus.}$$

	TrueCirconf_(cm)	TrueDiameter_(cm)	TrueDistCam/Tree_(cm)	TrueDiameter_(pxls)
60	38.0	12.095776	50.0	694.957293
61	38.5	12.254931	50.0	704.101468
62	39.0	12.414086	50.0	713.245643
63	39.5	12.573241	50.0	722.389818
64	40.0	12.732395	50.0	731.533993
65	8.0	2.546479	75.0	97.537866
66	8.5	2.705634	75.0	103.633982
67	9.0	2.864789	75.0	109.730099
68	9.5	3.023944	75.0	115.826216
69	10.0	3.183099	75.0	121.922332

Figure 19. Extrait du tableau des valeurs réelles simulées.

On a obtenu un tableau de 585 points (*taille du premier vecteur* × *taille du deuxième vecteur*) correspondant à une condition expérimentale donnée ( $circ_i, DCT_i$ ) avec  $i \in [0; 585]$ . Pour déterminer l'impact des erreurs de mesure sur ces conditions expérimentales, nous avons défini un ensemble de couples réalistes erreur de segmentation ( $e_s$ ) dans l'image / erreur d'évaluation de la distance camera/arbre ( $e_d$ ) (Tableau 2). Ces erreurs ont été appliquées respectivement au vrai diamètre de tronc en pixels,  $Diampxls_i$ , et à la vraie distance,  $DCT$ , lors de la simulation du processus d'estimation (partie inférieure de la Figure 19 ci-dessus) permettant d'obtenir la circonférence estimée,  $circ_i(e_s, e_d)$ . Pour cela, on simule tout d'abord l'erreur de segmentation affectant la valeur estimée du diamètre de tronc en pixels,  $Diampxls_i(e_s)$ , pour le couple  $(circ_i, DCT_i)$  :

$$Diampxls_i(e_s) = Diampxls_i + e_s$$

Où  $Diampxls_i$  est le vrai diamètre de tronc en pixels du  $i^{\text{ème}}$  individu et est donné par l' Cette résolution spatiale nous permet ainsi de convertir le diamètre du tronc vrai en diamètre de tronc en pixels (Diampxls en pixels) :

$Diampxls(diam, DCT) = \frac{diam}{Reso(DCT)}$  Équation 3. Toujours à l'aide de l' Cette résolution spatiale nous permet ainsi de convertir le diamètre du tronc vrai en diamètre de tronc en pixels (Diampxls en pixels) :

$Diampxls(diam, DCT) = \frac{diam}{Reso(DCT)}$  Équation 3, on convertit ensuite  $Diampxls_i(e_s)$  en diamètre de tronc en cm observé à une distance affectée d'une erreur  $e_d$  :

$$diam_i(e_s, e_d) = Reso(DCT + e_d) \times Diampxls_i(e_s)$$

Ce diamètre estimé peut ensuite être transformé en circonférence estimée, en supposant le tronc de section circulaire :

$$circ_i(e_s, e_d) = \pi \times diam_i(e_s, e_d).$$

On a ensuite calculé une matrice de RMSE (Root Mean Square Error) : chaque ligne de la matrice correspond à une erreur de segmentation et chaque colonne correspond à une erreur sur l'estimation de la distance caméra/arbre. Chaque valeur de RMSE de la matrice (de taille 41\*21) est calculée sur les 585 points correspondant à l'ensemble des couples diamètre, distance caméra arbre décrits ci-dessus :

$$RMSE(e_s, e_d) = \sqrt{\frac{1}{n} \sum_{i=1}^n (circ_i - \widehat{circ}_{i, e_s, e_d})^2} \quad \text{Équation 4}$$

Où  $circ_i$  est la circonférence vraie du  $i^{\text{ème}}$  individu,  $\widehat{circ}_{i, e_s, e_d}$  est la circonférence estimée du  $i^{\text{ème}}$  individu pour un couple d'erreurs  $(e_s, e_d)$  et  $n = 585$  est la taille de la population.

$\widehat{diam}_{pxls}$  s'exprime aussi en fonction du diamètre vrai  $diam$ , de la distance caméra/arbre  $DCT$  et des erreurs  $e_s$  et  $e_d$  et de la résolution de l'image :

$$\widehat{diam}_{pxls} = \frac{diam}{Reso(DCT)} + e_s \quad \text{Équation 5}$$

Dans un deuxième temps, on a voulu connaître l'influence de la distance à l'arbre sur la distribution des valeurs de RMSE afin d'effectuer d'éventuelles préconisations sur le protocole de mesure visant à assurer (si possible) la précision d'estimation exigée par le GAFL (précision au à 10 mm près). Pour cela, on a recalculé cette matrice de RMSE en fixant la distance entre la caméra et l'arbre à 50 cm, à 150 cm, puis à 250 cm.

Enfin, on s'est demandé quel était l'impact de la distance entre la caméra et l'arbre sur l'erreur d'estimation de la circonférence des troncs :

$$\text{On pose : } K_1 := \tan\left(\frac{ATFOV \times \pi}{2 \times 180}\right) \times 2 ;$$

On nomme l'erreur de diamètre en pixels  $err\_diam\_pxls$ , l'erreur de diamètre en cm  $err\_diam\_cm$  et l'erreur de circonférence en cm  $err\_circ$  :

$$\begin{aligned} err\_diam\_cm &= \widehat{diam} - diam \\ &= \widehat{diam}_{pxls} \times Reso(DCT + e_d) - Diampxls \times Reso(DCT) \\ &= (Diampxls + e_s) \times Reso(DCT + e_d) - Diampxls \times Reso(DCT) \\ err\_circ &= \pi \times ((Diampxls + e_s) \times Reso(DCT + e_d) - Diampxls \times Reso(DCT)) \\ err\_circ &= a \times e_s + b \end{aligned}$$

$$\text{où } a = \pi \times Reso(DCT + e_d) \text{ et } b = \pi \times Diampxls \times (Reso(DCT + e_d) - Reso(DCT))$$

$$\text{Équation 6}$$

### 3.2 Annotation des données RGB

Après avoir créé le jeu de données d'images prises sur le domaine St Paul, destiné à compléter le jeu de données d'entraînement du modèle de segmentation de bois, il a fallu annoter les images. En effet pour chaque image, il a fallu créer un masque binaire composé uniquement de pixels [0,0,0] (pixels représentant l'arrière-plan) et de pixels [1,1,1] (pixels représentant le bois). Pour annoter les images, deux logiciels ont été utilisés : Woodpicker [48] et SAM (Segment Anything Model) [49].

Le logiciel Woodpicker est basé sur des algorithmes de seuillage. En effet, pour utiliser Woodpicker, il fallait préalablement avoir pris 2 photos par arbre, dont une avec une bâche bleue (Figure 14 et Figure 16). La différence entre les deux images est ensuite multipliée terme à terme avec une indice de végétation. Enfin, l'image avec la bâche bleue est seuillée par rapport à la matrice produit obtenue.

Pour utiliser le logiciel Woodpicker, un curseur devait être déplacé. Des valeurs de curseurs trop faibles avaient pour conséquence des trous dans le coloriage et des valeurs de curseurs trop élevées avaient pour conséquence un coloriage qui débordait.

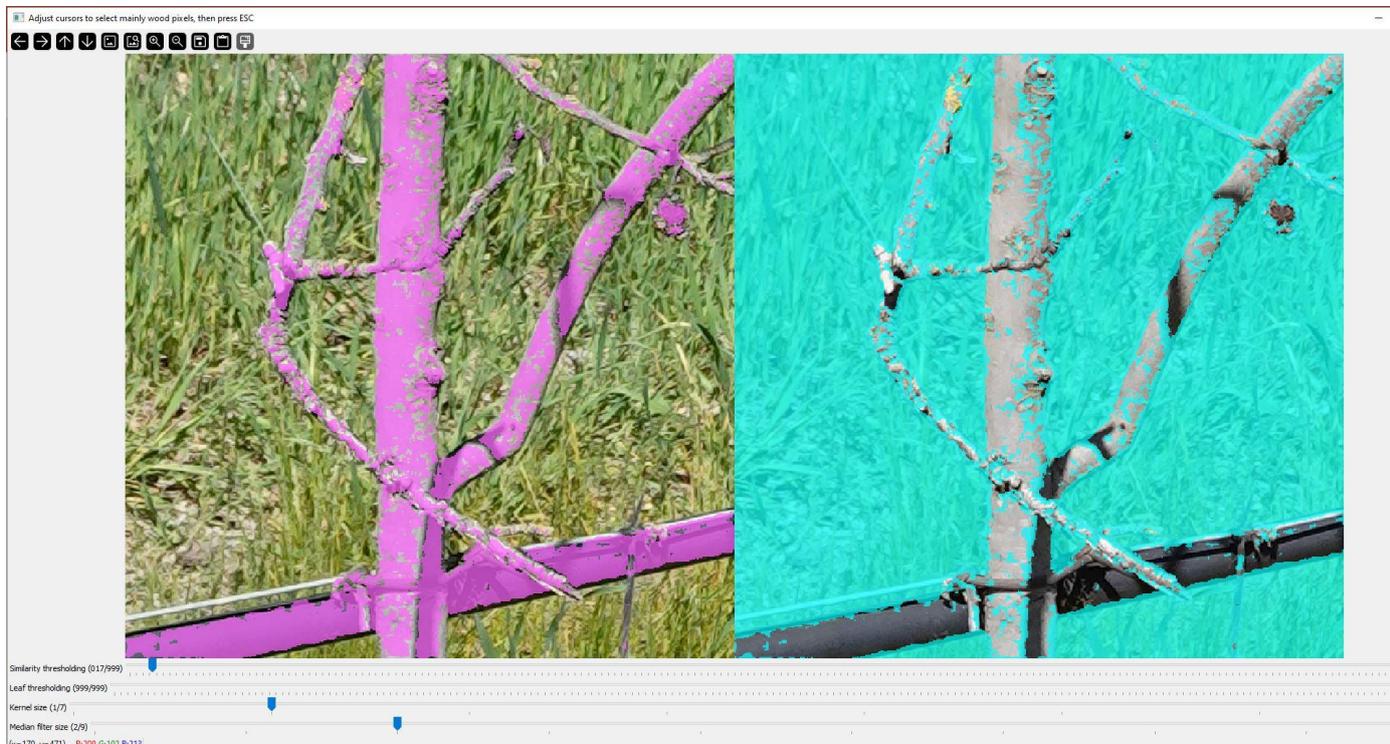


Figure 20. Interface de Woodpicker [48].

Après tous les placements de curseurs, une étape de « correction » permettait de compléter le coloriage à la main.

Le développement de Woodpicker nous a considérablement simplifié la tâche. En effet, sans le modèle, il aurait fallu annoter tous les pixels manuellement tandis qu'avec le modèle, l'essentiel du travail était fait automatiquement. De gros gains de temps ont été faits sur les annotations.

Le logiciel SAM quant à lui nous facilitait de nouveau la tâche. C'est un modèle développé pour réaliser de la segmentation sur tout type d'images. De plus, SAM a été entraîné sur un jeu de données de 11 millions d'images haute définition. Pour utiliser SAM, il suffisait de cliquer sur les zones de bois pour les annoter. Le logiciel était globalement assez performant. Il était toujours possible de rectifier le masquage lorsqu'il n'était pas assez précis. Cependant, on a débuté les annotations avec le logiciel Woodpicker, car on a appris l'existence de SAM qu'assez tard dans le stage. Pour les images qui avaient été acquises les années précédentes, les annotations étaient déjà disponibles avec les images.

### 3.3 Entraînement des réseaux

Pour les entraînements de réseaux de neurones qui vont suivre, on a réalisé de l'augmentation de données sur les sous-jeux de données d'entraînement uniquement, car c'est essentiellement ce jeu de données qui modifie les poids du réseau de neurones. Parmi les différentes augmentations, on pouvait retrouver des

rotations horizontales ou verticale ou encore l'application de filtre gaussien sur certaines images.

### 3.3.1 1<sup>er</sup> entraînement de réseau de neurones

Pour commencer, on a utilisé les images prises sur le domaine de St Paul et celles prises par Marie-Pia d'ARGAIGNON avant mon arrivée. La répartition des données est décrite par un tableau récapitulatif composé de 5 colonnes : le site sur lequel a été prise la photo, l'année durant laquelle l'arbre a été pris en photo, l'espèce prise en photo, le nom du fichier de l'image et si l'image appartient au sous-jeu de données d'entraînement, de validation ou de test (Figure 21) :

A	Site	Année	Espèce	Nom du patch	train/val/test
	Balandran :189	2021:189	Abricotier: 58	Length:635	test : 75
	Gotheron : 18	2022:214	Autre : 69	Class :character	train:497
	Guarrigues :150	2023:232	Pêcher :196	Mode :character	val : 63
	INRAE St Paul:232		Pommier :312		
SEFRA : 46					
B	Site	Année	Espèce	Nom du patch	train/val/test
	Balandran :159	2021:159	Abricotier: 43	Length:497	test : 0
	Gotron : 10	2022:160	Autre : 44	Class :character	train:497
	Guarrigues :117	2023:178	Pêcher :150	Mode :character	val : 0
	INRAE St Paul:178		Pommier :260		
SEFRA : 33					
C	Site	Année	Espèce	Nom du patch	train/val/test
	Balandran :10	2021:10	Abricotier: 4	Length:63	test : 0
	Gotron : 4	2022:28	Autre :25	Class :character	train: 0
	Guarrigues :17	2023:25	Pêcher :24	Mode :character	val :63
	INRAE St Paul:25		Pommier :10		
SEFRA : 7					
D	Site	Année	Espèce	Nom du patch	train/val/test
	Balandran :20	2021:20	Abricotier:11	Length:75	test :75
	Gotron : 4	2022:26	Autre : 0	Class :character	train: 0
	Guarrigues :16	2023:29	Pêcher :22	Mode :character	val : 0
	INRAE St Paul:29		Pommier :42		
SEFRA : 6					

Figure 21. A) Résumé R du tableau récapitulatif du premier jeu de données d'entraînement. B), C) et D) sont respectivement les résumés R des sous-jeux de données d'entraînement, d'évaluation et de test.

La répartition entre jeux d'entraînement, de validation et de test doit se rapprocher respectivement de 80%, 10%, 10% pour ces jeux de données (ce qui correspondrait respectivement à 508/64/63 images). L'objectif de cet entraînement était d'avoir des sous-jeux de données les plus équilibrés possible en termes de répartition des sites et de provenance des images. Le jeu de test est aussi équilibré en termes de répartition des sites et de provenance des images. Par conséquent, le jeu de test

n'est pas indépendant des autres sous-jeux de données. Le réseau a été entraîné pendant 100 époques.

### 3.3.2 2<sup>ème</sup> entraînement de réseau de neurones

Une fois le modèle pour annoter rapidement les images avec un fond de ciel développé, on a pu ajouter toutes les images de bois avec fond ciel, notamment celles acquises sur le Domaine de Saint-Paul en 2023 (Figure 22) :

A	Site	Année	Espèce	Nom du patch	train/val/test
	Balandran :189	2021:189	Abricotier: 58	Length:784	test : 79
	Gotheron : 18	2022:214	Autre :145	Class :character	train:631
	Guarrigues :150	2023:381	Pêcher :196	Mode :character	val : 74
	INRAE St Paul:381		Pommier :385		
SEFRA : 46					
B	Site	Année	Espèce	Nom du patch	train/val/test
	Balandran :179	2021:179	Abricotier: 56	Length:631	test : 0
	Gotheron : 16	2022:180	Autre :115	Class :character	train:631
	Guarrigues :120	2023:272	Pêcher :164	Mode :character	val : 0
	INRAE St Paul:272		Pommier :296		
SEFRA : 44					
C	Site	Année	Espèce	Nom du patch	Train/val/test
	Balandran :10	2021:10	Abricotier: 2	Length:74	test : 0
	Gotheron : 2	2022:34	Autre :30	Class :character	train: 0
	Guarrigues :30	2023:30	Pêcher :32	Mode :character	val :74
	INRAE St Paul:30		Pommier :10		
SEFRA : 2					
D	Site	Année	Espèce	Nom du patch	train/val/test
	Balandran : 0	2021: 0	Abricotier: 0	Length:79	test :79
	Gotheron : 0	2022: 0	Autre : 0	Class :character	train: 0
	Guarrigues : 0	2023:79	Pêcher : 0	Mode :character	val : 0
	INRAE St Paul:79		Pommier :79		
SEFRA : 0					

Figure 22. A) Résumé R du tableau récapitulatif du premier jeu de données d'entraînement. B), C) et D) sont respectivement les résumés R des sous-jeux de données d'entraînement, d'évaluation et de test.

On remarque notamment que les images rajoutées sont toutes d'espèce « Autre » qui sont des espèces diverses et non renseignées dans le jeu de données. On peut aussi voir que le jeu de test est uniquement composé d'images provenant d'INRAE St Paul, il est ainsi uniquement composé d'images avec du ciel en fond. Ceci permet de mettre l'accent sur l'indépendance entre les jeux d'entraînement et d'évaluation et le jeu de test. À nouveau, le réseau a été entraîné pendant 100 époques.

### 3.3.3 Dernier entraînement de réseau de neurones

Ensuite, on a réalisé de l'active learning (apprentissage actif) : on a regardé les IOU et F1-score par site d'acquisition des images pour savoir sur quels sites le modèle avait les meilleurs résultats. IOU signifie Intersection Over Union (intersection sur union). L'IOU est le rapport entre le nombre de pixels de l'intersection de la zone prédite et de la vérité, et le nombre de pixels de l'union de ces deux zones (Figure 23) :

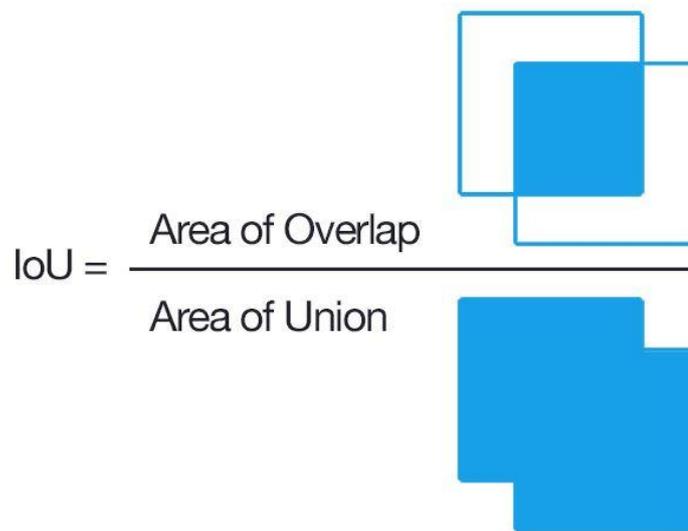


Figure 23. Illustration de l'IOU. Les 2 carrés de contour bleu représentent les zones de prédiction et de vérité. Les parties coloriées en bleu représente les zones considérées [50].

L'IOU est notamment défini par la formule suivante :

$$\text{IOU} = \frac{VP}{VP+FP+FN}, \text{ où VP sont les Vrai Positifs, FP les Faux Positifs et FN les Faux Négatifs}$$

Le F1-score lui est défini par la formule suivante :

$$F_1 = 2 \times \frac{\text{precision} \times \text{rappel}}{\text{precision} + \text{rappel}}, \text{ avec } \text{precision} = \frac{VP}{VP+FP} \text{ et } \text{rappel} = \frac{VP}{VP+FN}$$

Le F1-score quant à lui, est plus difficile à définir qualitativement.

En fonction des sites trouvés, on a essayé d'en déduire le type d'images prises dans ces sites qui posent problèmes. Enfin, une fois qu'on savait quelles images posaient problème, on a rajouté ce type d'images dans le jeu de données. À cette occasion, on a doublé la taille de notre jeu de données en extrayant de nouvelles imageries dans nos images disponibles grâce à SAM [49] (voir Annexe 3 : Tableau récapitulatif du jeu de données d'images à l'issue de l'active learning).

Ensuite, pour essayer de continuer d'améliorer les performances du modèle, on a optimisé la répartition des images dans les jeux d'entraînement, de validation et de test, en fonction de leur proportion de bois (déterminée à partir des annotations). Cela permettait d'assurer des distributions d'images similaires entre les trois jeux de données, et ainsi éviter, par exemple, d'avoir des images de test avec des proportions de bois n'étant pas présentes dans les jeux d'entraînement et de

validation. Pour cela, après avoir utilisé un algorithme d'imagerie assez basique (qui sommait le nombre de pixels de bois et divisait cette somme par le nombre de pixels de l'image), on a obtenu des sous-jeux de données ordonnés par proportion de bois dans l'image. Ensuite, pour la majorité des sous-jeux de données, on a réalisé un algorithme de Kmeans à 8 classes sur les proportions de bois (Figure 24). Pour les sous-jeux de données trop petits ou trop spécifiques, la totalité du sous-jeu de données servait pour le jeu de test.

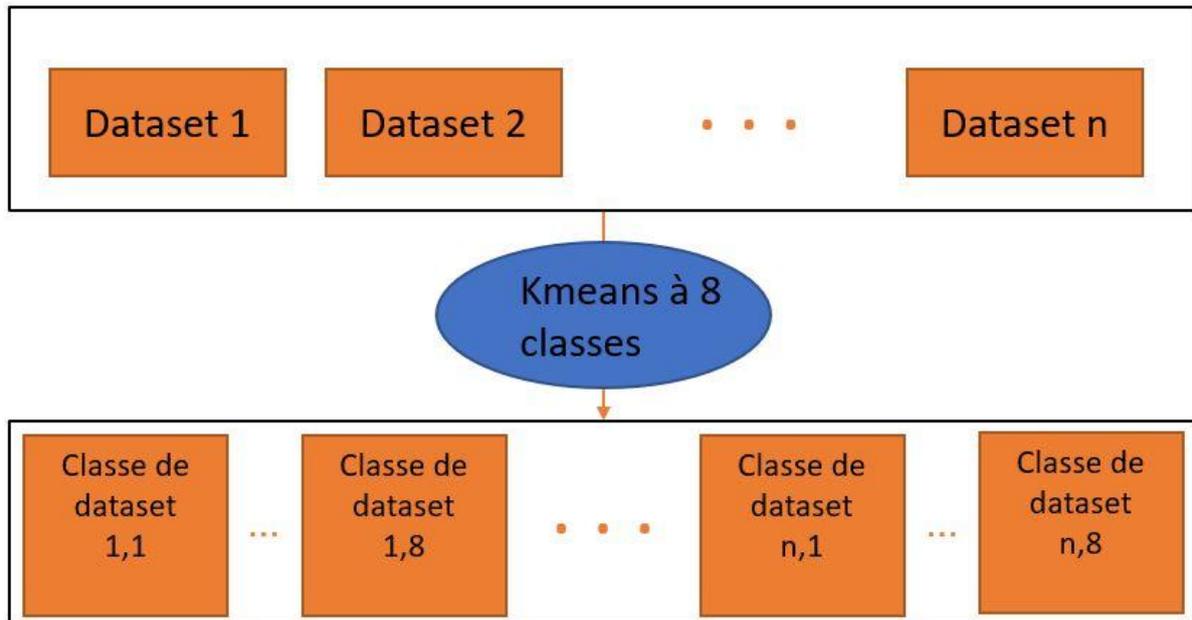


Figure 24. Schéma de la création des classes de jeux de données pour la création du jeu de données d'entraînement du réseau de neurones suivant.

Enfin, pour chaque sous-jeu de données, on a défini un ordre des classes aléatoire (Figure 25). Pour les jeux d'entraînement, d'évaluation et de test, on a tiré sans remise, successivement dans les 8 classes de chaque sous-jeu de données jusqu'à avoir la proportion voulue dans le jeu d'entraînement d'évaluation ou de test (80% du sous-jeu de données dans le jeu d'entraînement 10% du sous-jeu de données pour les jeux d'entraînements d'évaluation et de test) (Figure 26) :

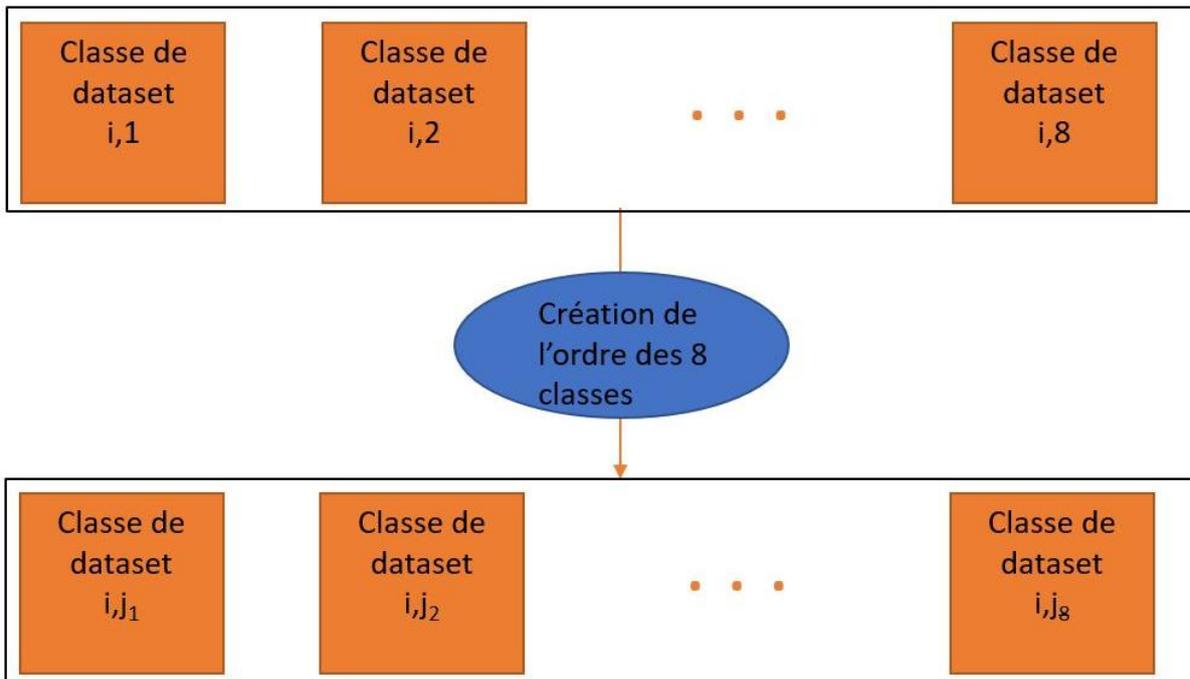


Figure 25. Schéma de la création de l'ordre des classes

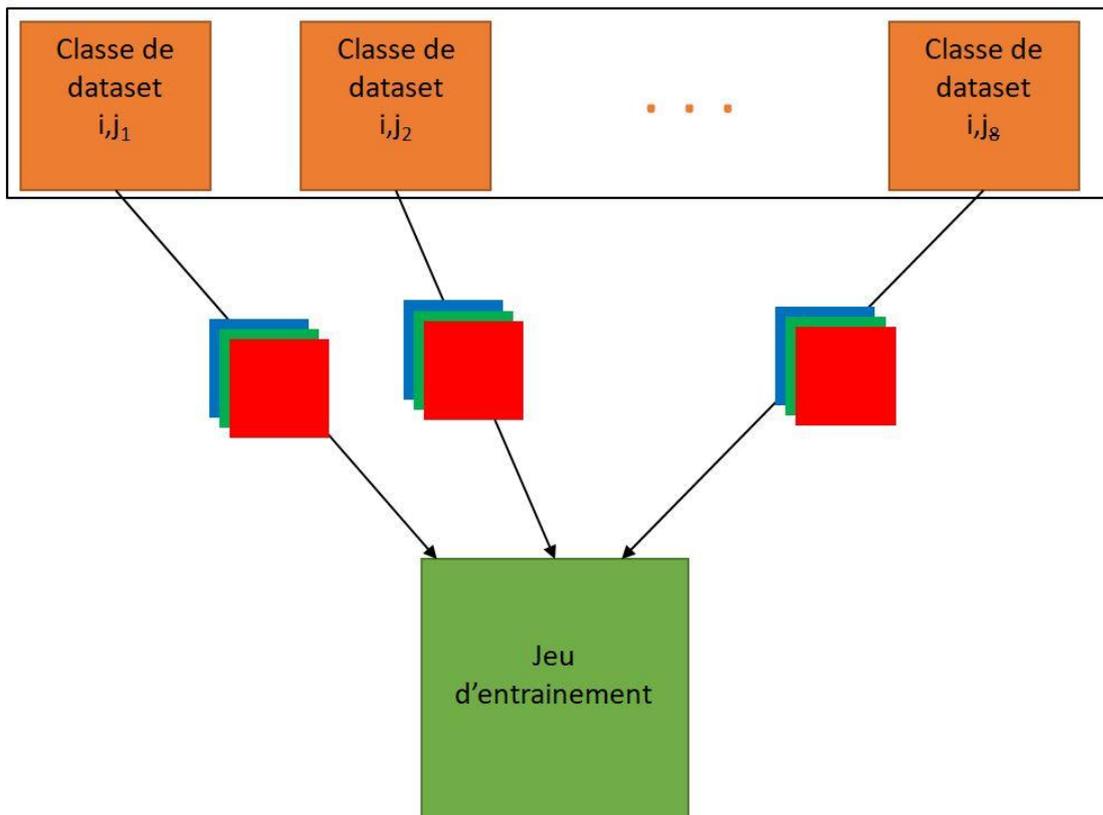


Figure 26. Un tirage avec remise d'une image est effectué à tour de rôle dans chaque classe. On reprend l'ordre des classes défini précédemment.

À l'issue de la création de ce jeu de données, le réseau a été entraîné pendant 250 époques.

## 4. Résultats et discussion

### 4.1 Approche basée sur les simulations

Pour représenter de manière synthétique les valeurs de RMSE pour les couples d'erreurs simulés, une heatmap (carte de chaleur) a été générée avec la matrice de RMSE pour l'ensemble des distances entre la caméra et l'arbre ([50 cm à 250 cm], Figure 27). Puis on a tracé 3 courbes représentant les moyennes des RMSE des circonférences en fonction des erreurs de segmentation, pour les distances entre la caméra et l'arbre de 50 cm, 150 cm et 250 cm et une erreur de distance entre la caméra et l'arbre nulle (Figure 27) :

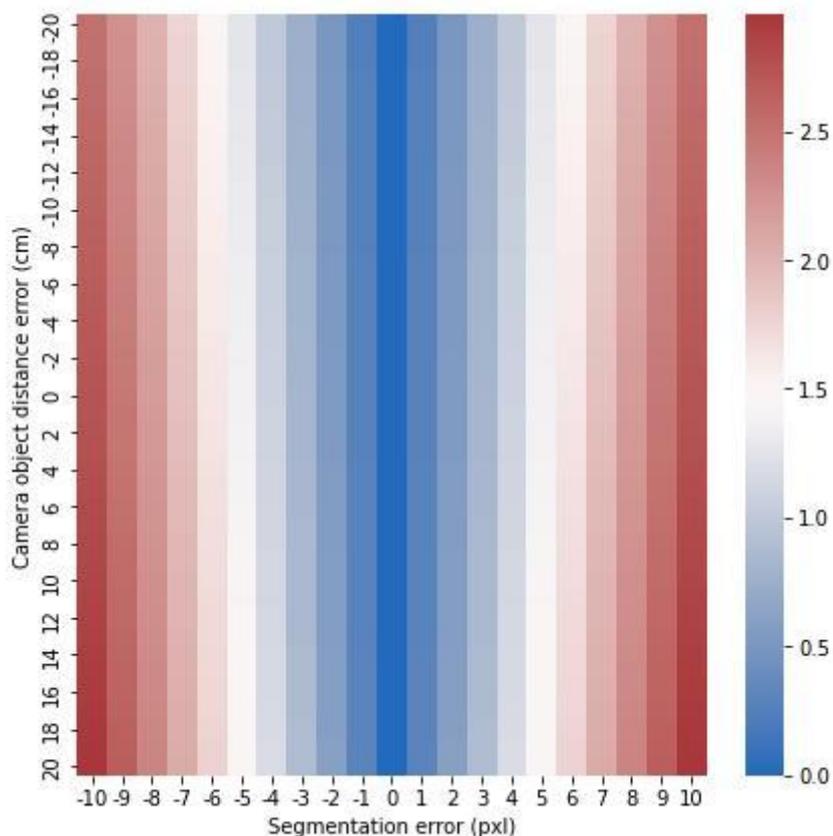


Figure 27. Heatmap de la matrice de RMSE des couples d'erreurs simulés pour les mesures de diamètres des troncs pour toutes les distances entre la caméra et l'arbre. ([50 cm à 250 cm]). L'échelle des RMSE est en cm.

Pour une variation d'erreur de segmentation de -10 pixels à 10 pixels et pour une variation d'erreur d'estimation de la distance entre la caméra et l'arbre qui varie entre -20 cm et 20 cm, l'erreur d'estimation de la circonférence varie entre 0 et 3 cm. On remarque qu'il y a très peu de variabilité lorsqu'on change de ligne, cependant, il y a beaucoup plus de variabilité lorsqu'on change de colonne, ce qui montre l'importance de l'impact de l'erreur de segmentation par rapport à une mauvaise estimation de la distance camera-objet.

Par ailleurs, la Figure 28 confirme ce que l'on a montré précédemment par rapport à la croissance l'erreur d'estimation (cf Équation 6) de circonférence. Cette croissance est affine :

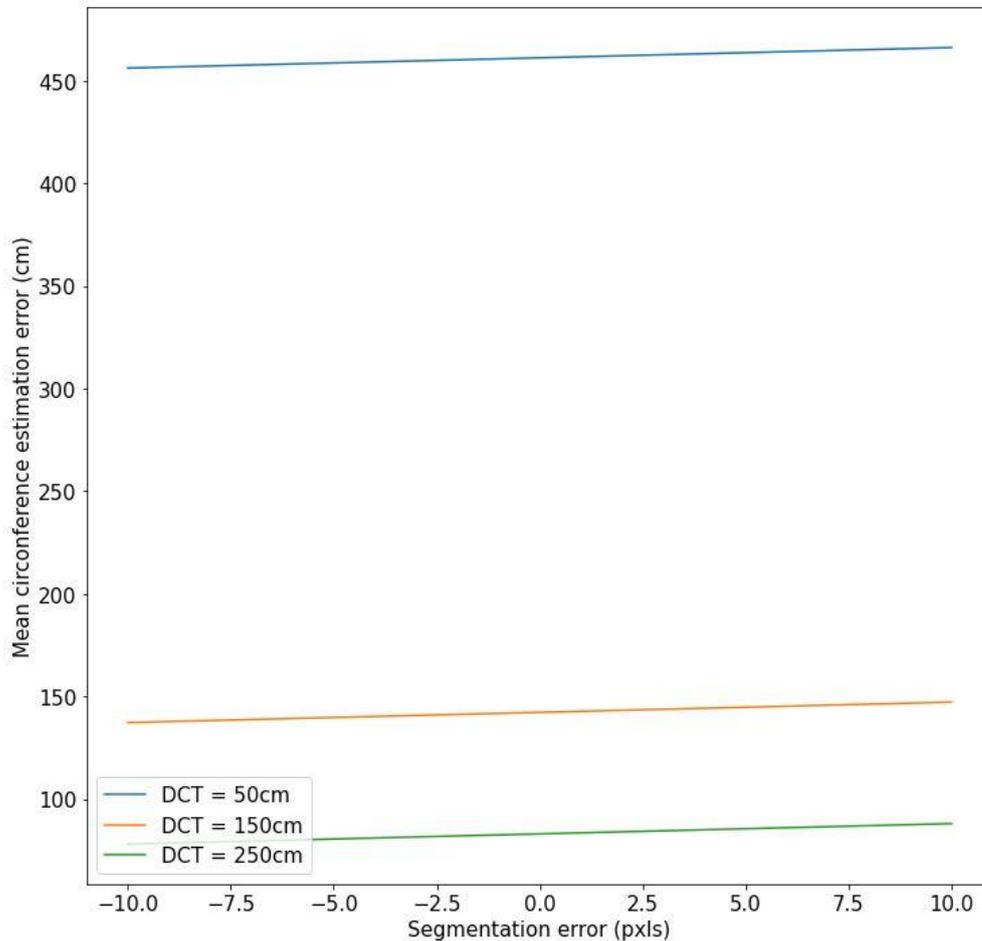


Figure 28. Graphique de comparaison des erreurs en fonction des distances entre la caméra et l'arbre (DCT).

On remarque que l'erreur de segmentation a beaucoup plus d'impact que l'erreur d'estimation de la distance. Par conséquent, dans la suite des travaux, on attachera beaucoup plus d'importance à améliorer les performances du réseau de segmentation plutôt que l'algorithme de vision en stéréo. Pour cela, on pourra rajouter des images pour l'entraînement.

De plus, l'erreur d'estimation de la circonférence croît proportionnellement avec la distance caméra/arbre. On pourrait donc penser obtenir une précision optimale en se rapprochant le plus possible de l'objet d'intérêt. Cependant, le fait de se rapprocher a deux inconvénients majeurs : (i) il faut augmenter le nombre de photos pour pouvoir reconstituer l'arbre dans son ensemble et (ii) on diminue la profondeur de champ, ce qui floute l'arrière-plan. Cela peut avoir une conséquence non-négligeable sur la précision des algorithmes de segmentation (segmentation des rameaux). Par ailleurs, au vu de cette étude, se placer à une distance d'entre un et deux mètres de l'arbre pour prendre les photos semble un bon compromis pour avoir une bonne profondeur de champ tout en gardant une bonne précision.

## 4.2 L'entraînement des réseaux de neurones

Les jeux de données ont été adaptés au fur et à mesure de l'entraînement. Le réseau de neurones a été entraîné 3 fois. Et de ces 3 entraînements, on a extrait les courbes de précision des jeux d'entraînement de validation. De plus, les IOU et F1-score ont été calculés à l'issue de chaque entraînement pour pouvoir évaluer ses performances.

### 4.2.1 1<sup>er</sup> entraînement de réseau de neurones

Avec le jeu de données d'entraînement, le réseau de neurones avait une précision de 88% à la 5<sup>ème</sup> époque. Entre la 5<sup>ème</sup> et la 40<sup>ème</sup>, sa précision a augmenté assez régulièrement jusqu'à 92%, puis cette précision a stagné jusqu'à la 60<sup>ème</sup> époque. Enfin, la précision a à nouveau augmenté plus doucement jusqu'à atteindre 93,5%.

Avec le jeu de données de validation, le réseau était très irrégulier jusqu'à la 6<sup>ème</sup> époque à laquelle il a atteint une précision de 64%. Entre la 6<sup>ème</sup> époque et la 12<sup>ème</sup> époque la précision a fortement augmenté jusqu'à atteindre 84%. Puis cette précision a augmenté plus faiblement jusqu'à atteindre 87% à la 70<sup>ème</sup> époque. Il y a eu un gros creux entre la 70<sup>ème</sup> époque et la 80<sup>ème</sup> époque à laquelle le réseau de neurones a finalement retrouvé sa précision de 87%. Enfin, la précision a continué de croître faiblement jusqu'à 92% (voir Annexe 4 : Évolution de la précision lors du premier entraînement de réseau de neurones).

Si on compare ces deux graphiques, on voit que globalement, les deux courbes ont la même allure, mais les valeurs de la courbe de validation sont plus faibles et plus irrégulières. Cette association de courbes montre un bon ajustement entre les 2 courbes. En effet, le jeu d'entraînement et jeu de validation sont bien équilibrés et reflètent suffisamment bien la variété du jeu de données.

On a ensuite calculé les des IOU et F1-score dans le but d'évaluer le réseau de neurones entraîné. L'IOU global est de 61% tandis que le F1-score global est de 72% (Figure 29).

<b><u>F1-SCORE + IOU PAR SITE :</u></b>		
Site	F1-Score	IOU
Balandran	0,67	0,54
Garrigues	0,72	0,62
Gotheron	0,82	0,72
SEFRA	0,76	0,62
St_Paul	0,73	0,64
Tous sites mélangés	0,72	0,61

Figure 29. Évaluation du premier entraînement par site.

Sur le site de Gotheron, on remarque le IOU et le F1-score sont respectivement de 72% et 82% ce qui est supérieur aux valeurs de tous les sites mélangés de 10%.

Cependant, lorsqu'on regarde de plus près les images de Gotheron, on remarque qu'il n'y a que 3 images provenant de Gotheron et que l'une de ces images ne montre pas de bois ce qui facilite la tâche du réseau de neurones. Par ailleurs, on remarque que le site de la SEFRA montre aussi des valeurs assez correctes surtout en terme de F1-score avec un F1-score de 76%. Le principal point commun des images provenant du site de la SEFRA est que leur arrière-plan est uniquement constitué de ciel (Figure 30 B). Il est fort probable que le contraste créé par le ciel et le bois rende la distinction des classes plus facile au réseau de neurones. En revanche, si on regarde le site de Balandran, les résultats sont assez médiocres avec un F1-score de 0,67 et un IOU de 0,54. En effet, ces images présentent de nombreux poteaux de bois que le réseau de neurones confond avec le bois des arbres (Figure 30 A).

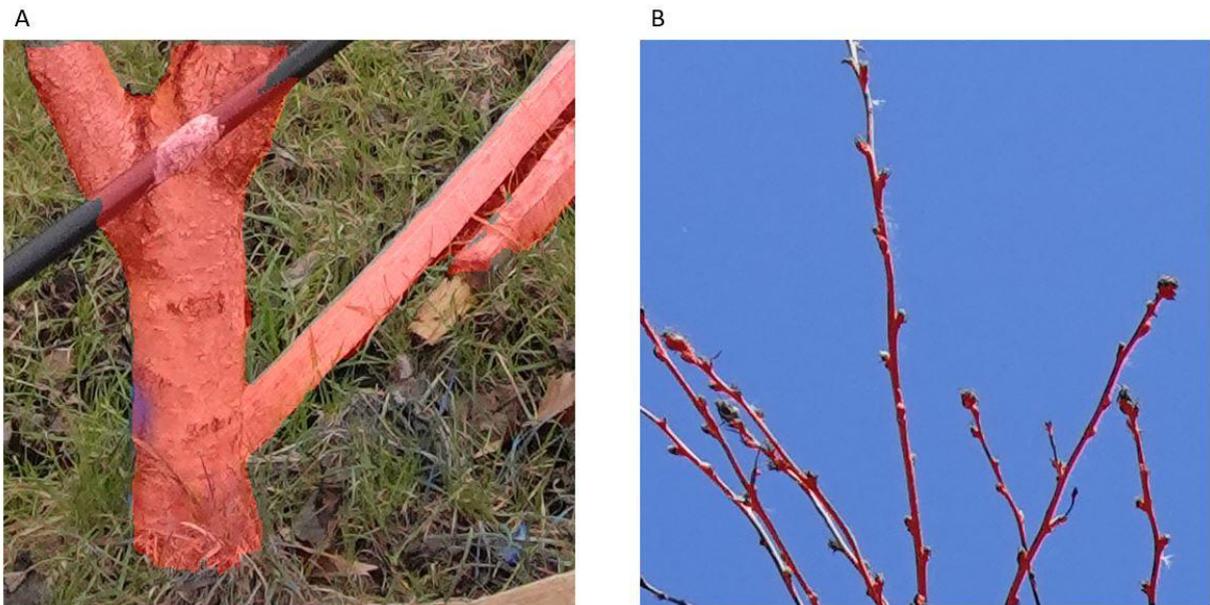


Figure 30. Visualisation de la détection de bois par le réseau de neurones. A) détection lorsqu'il y a du bois autre que l'arbre. B) détection lorsque le ciel est en arrière-plan.

#### 4.2.2 2<sup>ème</sup> entraînement de réseau de neurones

Avec le jeu d'entraînement, à la 5<sup>ème</sup> époque, le réseau de neurones avait une précision de 86%. Cette précision a connu une forte croissance jusqu'à atteindre 89% à la 12<sup>ème</sup> époque. Puis la précision a cru faiblement et constamment jusqu'à atteindre 93,5%.

Avec le jeu de validation, à nouveau, on a eu des irrégularités jusqu'à la 6<sup>ème</sup> époque à laquelle le réseau avait cette fois une précision de 74%. Il y a ensuite eu une forte croissance de la précision jusqu'à 92% un peu avant la 30<sup>ème</sup> époque. Puis la précision a ensuite cru faiblement et constamment jusqu'à atteindre 95% (voir Annexe 5 : Évolution de la précision lors du 2<sup>ème</sup> entraînement de réseau de neurones).

Pour cet entraînement, en comparant les deux graphiques, ces derniers ont à nouveau la même allure globale. Cette fois, le jeu de validation a des valeurs de précision supérieures au jeu d'entraînement. Par conséquent, ce jeu de validation qui est trop peu représentatif et procure trop peu d'exemples. Par ailleurs, le fait que la courbe d'entraînement soit encore en train de croître montre que l'entraînement n'est

pas terminé et que plus d'époques amélioreraient encore un peu les performances du réseau.

Pour cet entraînement, les images présentes dans le jeu de test ne proviennent que d'un site (conséquence de l'indépendance des sous-jeux de données cf §3.3.2). On observe un IOU de 64% et un F1-score de 73%. C'est mieux que lors du précédent entraînement. Cependant, lorsqu'on évalue ce réseau de neurones avec le jeu de test du précédent entraînement (composé en partie d'images qui ont servi à l'entraînement), les résultats sont bien moins bons. En effet, l'IOU redescend à 60% tandis que le F1-score, lui descend à 71%. Même si le réseau de neurones n'a été entraîné avec aucune image de fond ciel, ces images sont bien plus faciles à évaluer pour le réseau de neurones et gonflent ses résultats.

#### 4.2.3 Entraînement avec un jeu de données plus équilibré en répartition des proportions de bois

Avec le jeu d'entraînement, cette fois, on ne voit pas d'irrégularité au début de l'entraînement. La première précision affichée est de 90%. Puis on a une croissance assez régulière de la forme  $x \mapsto e^{-x}$  jusqu'à ce que la précision atteigne 96%.

Avec le jeu de validation, il y a beaucoup d'irrégularités jusqu'à la 15<sup>ème</sup> époque. Puis, la croissance devient globalement de la forme  $x \mapsto 1 - e^{-x}$ , cependant elle est très irrégulière. Les principaux creux dans la précision de l'entraînement sont aux époques 35, 135 et 155. La précision atteint finalement 94,5% (voir Annexe 6 : Évolution de la précision lors du dernier entraînement de réseau de neurones).

L'omniprésence d'irrégularités pour le jeu de validation comparée à la quasi-absence d'irrégularités pour le jeu d'entraînement montre que le jeu de validation est trop petit. En effet, le réseau de neurones n'arrive pas à suffisamment adapter ses paramètres pour ce jeu de données à cause de sa taille.

Pour cet entraînement les IOU et F1-score globaux sont respectivement de 67% et 78 % (Figure 31) :

<b>F1-SCORE + IOU PAR SITE :</b>		
Site	F1-Score	IOU
Amarine	0,86	0,76
Balandran	0,61	0,47
St Paul	0,76	0,65
Garrigues	0,83	0,73
SEFRA	0,78	0,66
Gotheron	0,79	0,72
Tout	0,78	0,67

Figure 31. Évaluation du premier entraînement par site.

On remarque qu'on obtient les meilleures métriques du jeu de données de test avec les images du site de l'Amarine. En effet, elles sont présentent toutes du ciel en fond qui, comme on l'a vu plus tôt, facilite probablement la tâche au réseau de neurones. Cependant, le site de Balandran montre des métriques très faibles. Ces métriques sont même plus faibles que lorsque le premier réseau de neurones avait été évalué sur le site de Balandran alors que ce premier réseau de neurones avait été entraîné pendant 150 époques de moins que le dernier.

L'amélioration des résultats (+6% pour le F1-score et l'IOU) du dernier réseau de neurones par rapport aux précédents n'est pas très surprenante. En effet, on sait que l'efficacité d'un réseau de neurones dépend en majeure partie de la qualité de son jeu de données d'entraînement. Plus particulièrement, on sait que les performances d'un réseau de neurones dépendent de la taille de son jeu de données d'entraînement et de sa diversité. Or, entre le 2<sup>ème</sup> et le 3<sup>ème</sup> entraînement, la taille du réseau de neurones a doublé.

De plus, il semblerait qu'une bonne répartition du jeu de données soit complémentaire à la taille de ce jeu de données pour obtenir un jeu de données efficace. En effet, un jeu de test indépendant, permet d'avoir un test significatif, mais créer un jeu test indépendant sans bien répartir les données du jeu d'entraînement revient à assurer la qualité du test tout en sacrifiant la qualité du réseau de neurones. Évidemment, une bonne répartition des données du jeu d'entraînement couplée à un jeu de test indépendant est l'idéal, mais il faut que le jeu de données le permette.

#### 4.3 Comparaison des estimations de circonférences

L'objectif initial était d'améliorer la qualité du modèle d'estimation de la circonférence des troncs. Une comparaison a donc été faite entre les estimations qui utilisent le précédent réseau de neurones et celles qui utilisent le nouveau (Figure 32) :

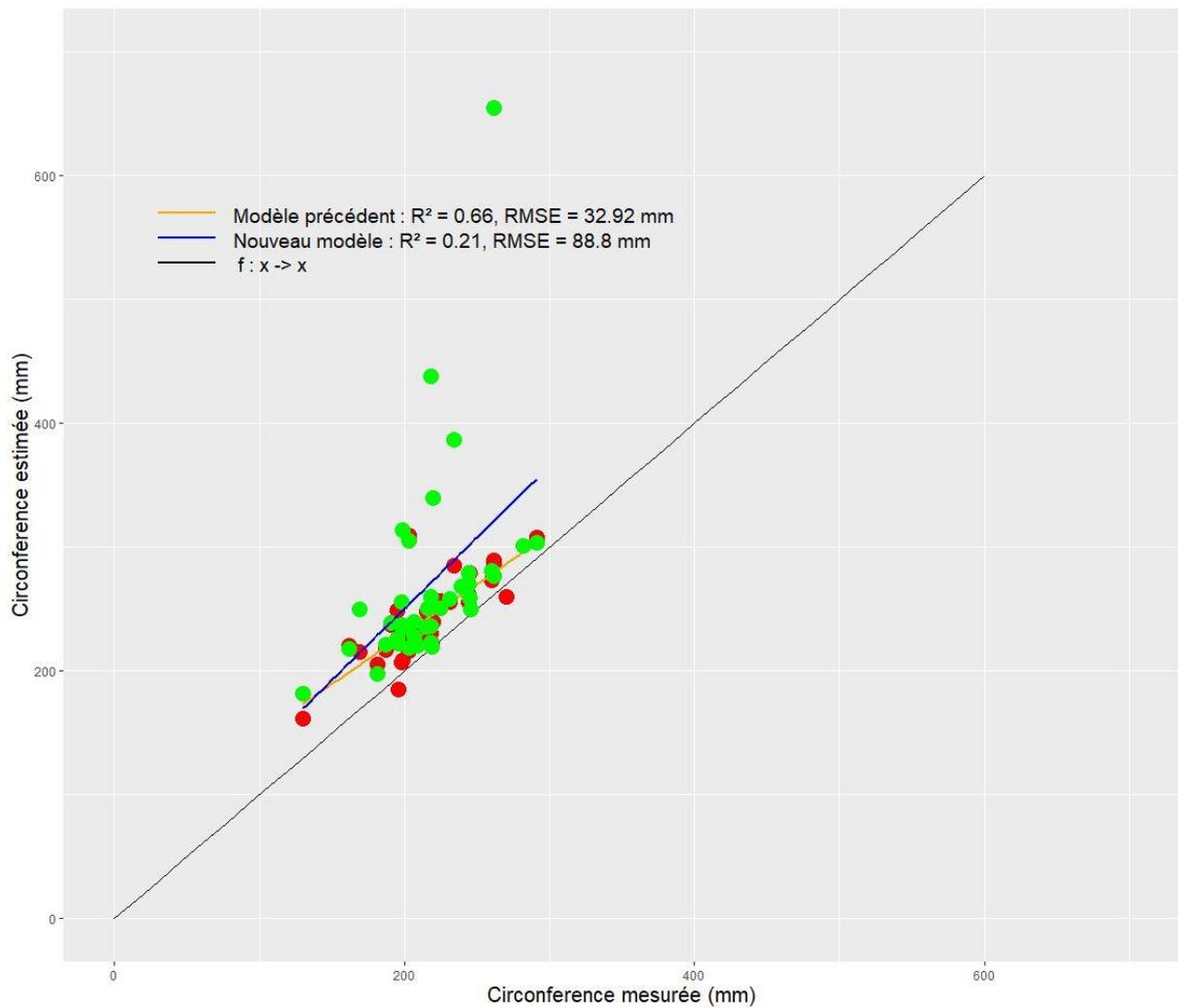


Figure 32. Régression linéaire de la circonférence estimée par le modèle en fonction de la circonférence mesurée en verger.

Les points rouges représentent les estimations du précédent modèle tandis que les points verts les représentent les estimations du nouveau modèle. On remarque que lorsqu'on utilise le nouveau réseau de neurones sur images, on a une précision globale plus faible (RMSE plus élevé). De plus, on constate que les circonférences estimées sont beaucoup moins bien corrélées avec les circonférences mesurées en verger ( $R^2$  beaucoup plus faible). On peut aussi constater que le nouveau modèle est beaucoup plus irrégulier. En effet, on constate la présence de valeurs « aberrantes », anormalement élevées pour ce modèle (Figure 33) :

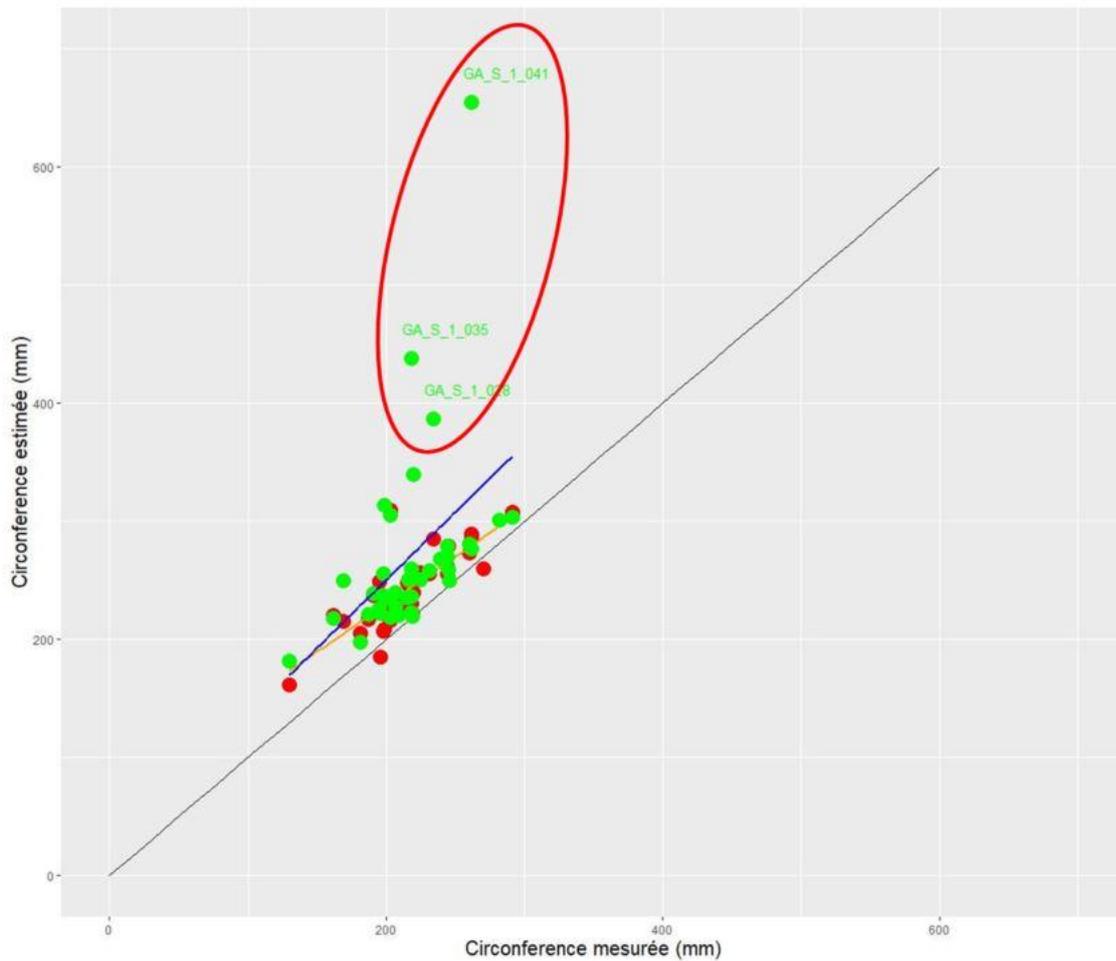


Figure 33. Régression linéaire de la circonférence estimée par le modèle en fonction de la circonférence mesurée en verger avec les valeurs aberrantes mises en évidence.

On peut maintenant se poser la question quant à la cause de ces points aberrants. En effet, le modèle d'estimation des circonférences permet de regarder les différentes étapes qui ont permis d'estimer la circonférence (Figure 34) :

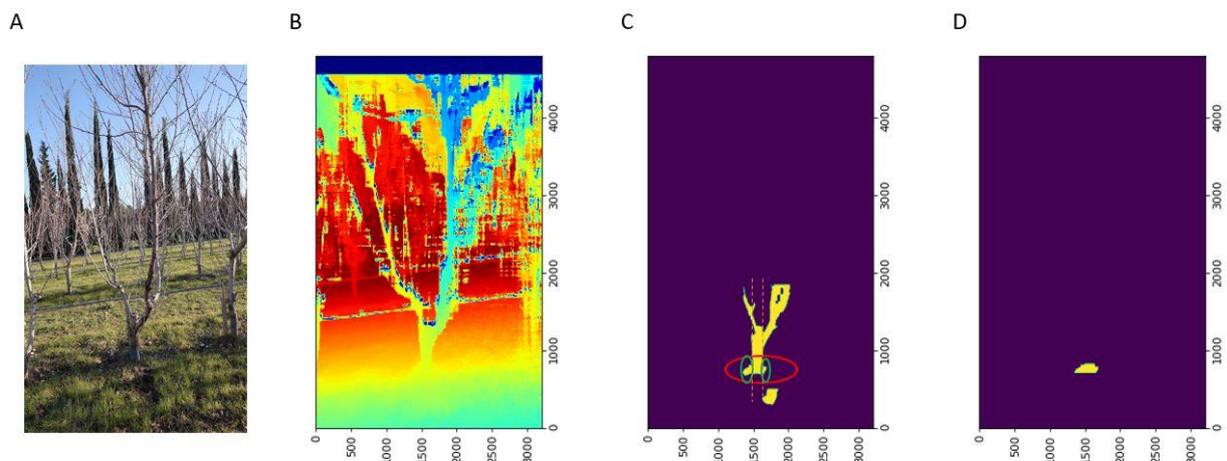


Figure 34. Différentes étapes de l'estimation de circonférence pour l'arbre GA\_041. A) image RGB. B) image de distance au capteur estimée par stéréovision. C) détection du tronc. D) détection de la marque bleue.

Sur la Figure 34C), on comprend que le réseau de neurones a confondu le sol autour du pied du tronc avec le bois du tronc ce qui élargit donc le tronc sur l'image. Enfin, la Figure 34D) nous montre que c'est à cet endroit que la marque bleue a été détectée par le modèle. Par conséquent, le modèle retourne une circonférence anormalement élevée. L'arbre GA\_041 est un exemple, mais c'est ce genre d'erreurs qui est à l'origine des valeurs « aberrantes » du modèle qui sont quasiment absentes avec l'ancien modèle.

Par conséquent, on peut se demander ce qui se passe lorsqu'il n'y a pas d'erreurs importantes de segmentation. Sur la Figure 35, on peut constater que la qualité de la régression linéaire n'a pas beaucoup changé puisque le coefficient de détermination est quasiment le même. Il y a toujours de fortes irrégularités dans la qualité des estimations.

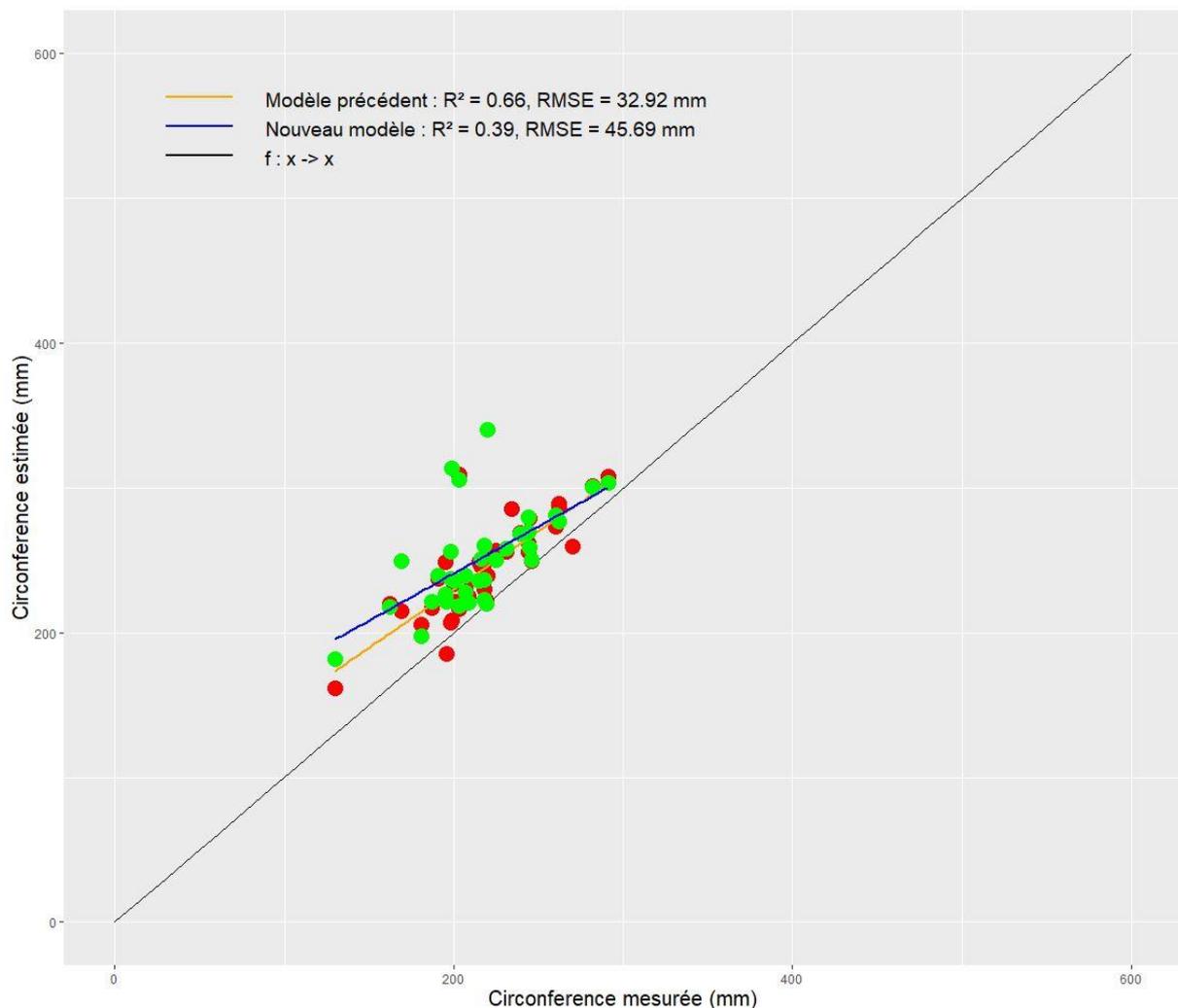


Figure 35. Régression linéaire de la circonférence estimée par le modèle en fonction de la circonférence mesurée en verger après avoir retiré les valeurs aberrantes

Cependant, même, si cela était partiellement attendu, l'erreur globale a fortement diminué puisqu'elle a quasiment été divisée par 2. Cela montre une part importante de la valeur de RMSE anormalement élevée était due aux valeurs aberrantes même si ces dernières étaient peu nombreuses.

Pour encore améliorer la qualité du modèle, on sait que la solution la plus efficace est d'augmenter considérablement la taille du jeu de données. Cependant, l'augmentation de la taille du jeu de données est relativement coûteuse en temps. En effet, il faut récolter beaucoup de données et les annoter. Par ailleurs, la part de la répartition des données dans les sous-jeux de données est aussi très importante et souvent sous-estimée dans les pistes d'améliorations de la qualité de l'entraînement des réseaux de neurones. Il faut évidemment comprendre son jeu de données dans le but de l'équilibrer le plus possible et de mettre le plus de variété possible dans chacun des sous-jeux de données.

## 5. Conclusion

Cette étude visait à améliorer une méthode d'estimation de la circonférence de tronc en vergers à partir d'une paire d'images acquises avec un système de stéréovision. Ce modèle était essentiellement basé sur de la segmentation par réseau de neurones couplée à de la stéréovision. Une matrice de RMSE a été créée dans le but de comparer les impacts des erreurs de segmentation et de stéréovision dans l'estimation de la circonférence des troncs. L'erreur qui a le plus d'impact sur les estimations est l'erreur de segmentation, c'est donc sur la segmentation que le travail d'amélioration a été effectué.

On a eu l'occasion d'observer les différentes méthodes d'amélioration des performances d'un réseau de neurones tout en considérant l'accessibilité de ces méthodes. Ces travaux ont aussi contribué au développement d'un outil de phénotypage digital. Cependant, malgré une légère amélioration du modèle d'estimation de la vigueur, ce dernier manque encore de précision pour pouvoir réaliser des études rigoureuses.

Les axes d'amélioration précédemment évoqués ne sont que des exemples de continuations possibles pour le travail que j'ai commencé au sein d'INRAE. En effet, le sujet est trop complet pour pouvoir être traité en 6 mois. Le travail que j'ai réalisé nous a malgré tout permis de nous rendre compte que la taille du jeu de données est loin d'être le seul levier pour jouer sur la qualité de l'entraînement d'un modèle de deep learning.

Pour encore améliorer ce réseau de neurones, il faut d'une part, encore augmenter la taille du jeu de données d'entraînement. Plus le jeu de données sera important, plus le réseau de neurones sera précis. Les premiers réseaux de neurones de classification étaient entraînés avec des jeux de données de millions d'images [51]. De plus, il faut garder la meilleure répartition possible pour les sous-jeux de données d'entraînement. Le critère de notre étude était la proportion de bois dans les images, mais une recherche de critère plus approfondie peut être nécessaire. On peut notamment utiliser des critères comme le type d'ensoleillement, le type d'arrière-plan (par exemple composé de haies ou de ciel). Enfin, plus le réseau est entraîné longtemps, plus il est précis. Dans notre cas, le réseau a été entraîné sur 250 époques. Il faudrait entraîner le réseau de neurones pendant au moins 10 heures comme cela a déjà été fait lors d'études théoriques [36].

Par ailleurs, ce stage m'a permis de découvrir comment se déroulait une synergie entre plusieurs laboratoires de recherche. J'ai aussi pu apprendre comment créer ou compléter un jeu de données à travers des acquisitions d'images en vergers.

## Annexes

### Annexe 1 : Bibliothèques python utilisées durant le stage

Bibliothèques python usuelles	math : permet d'utiliser les fonctions mathématiques de la bibliothèque C cmath [52]
	matplotlib : ce module permet de réaliser de nombreux types de graphiques [53]
	numpy : aide à réaliser des calculs numériques et à réaliser des tableaux [54]
	random : permet de générer des nombres pseudo-aléatoires [55]
	sys : permet d'interagir avec les variables en ligne de commande [56]
Utilisation du système d'exploitation	argparse : rend l'exécution de codes python en lignes de commande plus ergonomique [57]
	copy : permet la copie d'objets de façon superficielle et récursive [58]
	glob : permet de rechercher tous les chemins (de fichiers) correspondant à un motif particulier [59]
	openpyxl : permet de lire et d'écrire dans des fichiers Excel [60]
	os : permet d'utiliser les fonctionnalités du système d'exploitation à travers python [61]
	pandas : offre des structures ergonomiques et intuitives pour l'analyse de données [62]
	shutil : permet des opérations sur les fichiers et dossiers comme le copié-collé [63]
Imagerie classique	imageio : fournit une interface facile à lire et une large gamme de données d'image, y compris des images animées, volumétriques des données et des formats scientifiques [64]
	opencv-python : permet le traitement d'images en temps réel [65]
	skimage : liste d'algorithmes de traitement d'image [66]

Deep learning	sklearn : boite à outils pour l'analyse de données prédictives [67]
	albumations : permet d'augmenter la taille de jeux de données d'images via des rotations et des changements de couleurs [68]
	datasets : permet la gestion du chargement des jeux de données pour l'utilisation de réseaux de neurones [69]
	evaluate : permet de comparer des modèles et d'obtenir leurs performances plus facilement [70]
	huggingface : permet d'interagir avec les fonctionnalités Huggingface [71]
	keras : Keras est une API (Application Programming Interface) qui permet d'interagir avec les algorithmes de réseaux de neurones, notamment Tensorflow ou Theano [72]
	tensorflow : librairie open source pour le calcul numérique haute performance qui permet l'utilisation de GPU pour les calculs, notamment utilisé pour les algorithmes de réseaux de neurones [73]
	torch : librairie étudiée pour les réseaux de neurones profonds qui permet notamment l'utilisation de GPU pour les calculs [74]
	torchvision : fournit des jeux de données connus et des algorithmes de transformations pour l'imagerie [75]

Autres	collections : permet l'implémentation de types de données provenant de conteneurs [76]
	datetime : permet de manipuler les dates et les heures [77]
	itertools : permet de construire des itérateurs facilement [78]
	json : permet d'encoder et de décoder des fichiers json facilement [79]
	multiprocessing : permet de faire du calcul parallèle [80]
	seaborn : librairie basée sur matplotlib qui permet de visualiser les données statistiques [81]
	time : fournit différentes fonctions liées au temps [82]
	urllib : module de gestion des URL [83]

## Annexe 2 : Tableau récapitulatif des sous-jeux de données d'images utilisés pour l'entraînement des réseaux de neurones

Avant/après	Nom du jeu de données (nom du dossier)	Année(s)	Site(s)	Espèce(s)	Type de scène (par ex, bois+fleurs+fond ciel, tronc+fond herbe, bois+fleurs+fond haie, branches+fond haie...)	Ciel bleu / blanc	Avec / sans ombre	Nombre de patches
Avant	2021_balandran_vergers_CTIFL	2021	Balandran	Abricotiers et Pommiers	Branches + fond herbe	Blanc	Sans	137
Avant	2021_balandran_vergers_CTIFL_fond_ciel	2021	Balandran	Abricotiers et Pommiers	Branches + fond ciel	Blanc	Sans	52
Avant	MPA_2022	2022	SEFRA + Garrigues + Gotheron	Abricotier + Pêcher	Troncs + fond sol	Blanc	Avec	132
Avant	MPA_2022_fond_ciel	2022	SEFRA + Garrigues + Gotheron	Abricotier + Pêcher	Branches + fleurs + fond ciel	Blanc	Avec	82
Avant	JD_2023	2023	St Paul	Diverses	Troncs + fond sol, troncs + fond haie, branches + fond sol, branches + fond haie	Bleu	Avec	232
Après	190122_Garrigues_annoté	2022	Garrigues	Pêcher	Branches + fond ciel	Blanc	Sans	80
Après	190122_verger_ag13_fond_haie_sombre_annoté	2022	Garrigues	Pêcher	Branches + fond haie, piquets/tuyaux d'irrigation avec ou sans tronc	Blanc	Sans	22
Après	02032022_Garrigues_fleurs_annoté	2022	Garrigues	Pêcher	Branches + fleurs + fond ciel	Blanc	Sans	7
Après	09032022_Garrigues_annoté	2022	Garrigues	Pêcher	Branches + fleurs + fond ciel	Blanc	Sans	10
Après	14032022_Gotheron_fleurs_annoté	2022	Gotheron	Abricotier	Branches + fleurs + fond ciel	Blanc	Avec	30
Après	17032022_Sefra_fleurs_annoté	2022	SEFRA	Pêcher	Branches + fleurs + fond ciel	Blanc	Sans	29
Après	24032022_Sefra_fleurs_annoté	2022	SEFRA	Pêcher	Branches + fleurs + fond ciel	Bleu	Avec	29
Après	160222_verger_ag13_set_circonference_troncs_annoté	2022	Garrigues	Pêcher	Troncs + fond herbes	x	Avec	24
Après	160222_verger_ag13_troncs_annoté	2022	Garrigues	Pêcher	Troncs + fond herbes + piquets + tuyaux d'irrigation	x	Avec	94
Après	2022_Dataset_flower_detection_annoté	2022	SEFRA + Garrigues + Gotheron	Pêcher + Abricotier	Branches + fond ciel	Blanc	Sans	79
Après	06042023_StPaul_annoté	2023	St Paul	Diverses	Branches + fond ciel	Bleu	Avec	282
Après	2023_Amarine_fleurs_annoté	2023	Amarine	Abricotier	Branches + fleurs + fond ciel	Bleu	Avec	13
Après	2023_Garrigues_fleurs_annoté	2023	Garrigues	Pêcher	Branches + fleurs + fond haie	Bleu	Avec	2
Après	2023_Garrigues_fleurs_annoté	2023	Garrigues	Pêcher	Branches + fleurs + fond ciel	Bleu	Avec	7
							Total :	1336

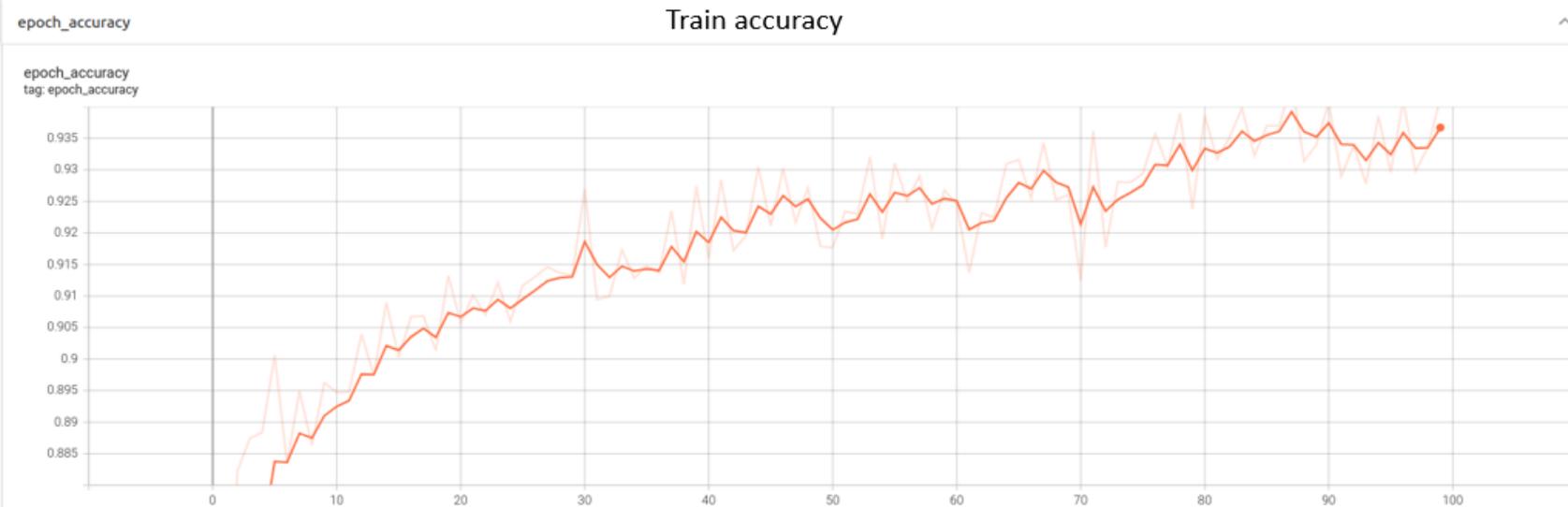
### Annexe 3 : Tableau récapitulatif du jeu de données d'images à l'issue de l'active learning

Avant/après 13/07/23	Nom du jeu de données (nom du dossier)	Année(s)	Site(s)	Espèce(s)	Type de scène (par ex, bois+fleurs+fond ciel, tronc+fond herbe, bois+fleurs+fond haie, branches+fond haie...)	Ciel bleu / blanc	Avec / sans ombre	Nombre de patches	Patches dans le jeu test	
Avant	2021_balandran_vergers_CTIFL	2021	Balandran	Abricotiers et Pommiers	Branches + fond herbe	Blanc	Sans	137	13	
Avant	MPA_2022	2022	SEFRA + Garrigues + Gotheron	Abricotier + Pêcher	Troncs + fond sol, Branches + fond sol	Blanc	Avec	132	13	
Avant	JD_2023	2023	St Paul	Diverses	Troncs + fond sol, troncs + fond haie, branches + fond sol, branches + fond haie	Bleu	Avec	232	23	
Après	190122_verger_ag13_fond_haie_sombre_annoté	2022	Garrigues	Pêcher	Branches + fond haie, piquets/tuyaux d'irrigation avec ou sans tronc	Blanc	Sans	22	2	
Après	160222_verger_ag13_set_circonference_troncs_annoté	2022	Garrigues	Pêcher	Troncs + fond herbes	x	Avec	10	1	
Après	160222_verger_Gotheron_set_circonference_troncs_annoté	2022	Gotheron	Pêcher	Troncs + fond herbes	x	Avec	14	14	
Après	160222_verger_ag13_troncs_annoté	2022	Garrigues	Pêcher	Troncs + fond herbes + piquets + tuyaux d'irrigation	x	Avec	94	9	
Après	2023_Garrigues_fleurs_annoté	2023	Garrigues	Pêcher	Branches + fleurs + fond haie	Bleu	Avec	2	0	
								total :	643	75
Avant	2021_balandran_vergers_CTIFL_fond_ciel	2021	Balandran	Abricotiers et Pommiers	Branches + fond ciel	Blanc	Sans	52	5	
Avant	MPA_2022_fond_ciel	2022	SEFRA + Garrigues + Gotheron	Abricotier + Pêcher	Branches + fleurs + fond ciel	Blanc	Avec	82	8	
Après	190122_Garrigues_annoté	2022	Garrigues	Pêcher	Branches + fond ciel	Blanc	Sans	80	8	
Après	02032022_Garrigues_fleurs_annoté	2022	Garrigues	Pêcher	Branches + fleurs + fond ciel	Blanc	Sans	7	1	
Après	09032022_Garrigues_annoté	2022	Garrigues	Pêcher	Branches + fleurs + fond ciel	Blanc	Sans	10	1	
Après	14032022_Gotheron_fleurs_annoté	2022	Gotheron	Abricotier	Branches + fleurs + fond ciel	Blanc	Avec	30	3	
Après	17032022_Sefra_fleurs_annoté	2022	SEFRA	Pêcher	Branches + fleurs + fond ciel	Blanc	Sans	29	3	
Après	24032022_Sefra_fleurs_annoté	2022	SEFRA	Pêcher	Branches + fleurs + fond ciel	Bleu	Avec	29	3	
Après	2022_Dataset_flower_detection_annoté	2022	SEFRA + Garrigues + Gotheron	Pêcher + Abricotier	Branches + fond ciel	Blanc	Sans	79	8	
Après	06042023_StPaul_annoté	2023	St Paul	Diverses	Branches + fond ciel	Bleu	Avec	282	28	
Après	2023_Amarine_fleurs_annoté	2023	Amarine	Abricotier	Branches + fleurs + fond ciel	Bleu	Avec	13	13	
Après	2023_Garrigues_fleurs_annoté_fond_ciel	2023	Garrigues	Pêcher	Branches + fleurs + fond ciel	Bleu	Avec	7	7	
								total :	700	88

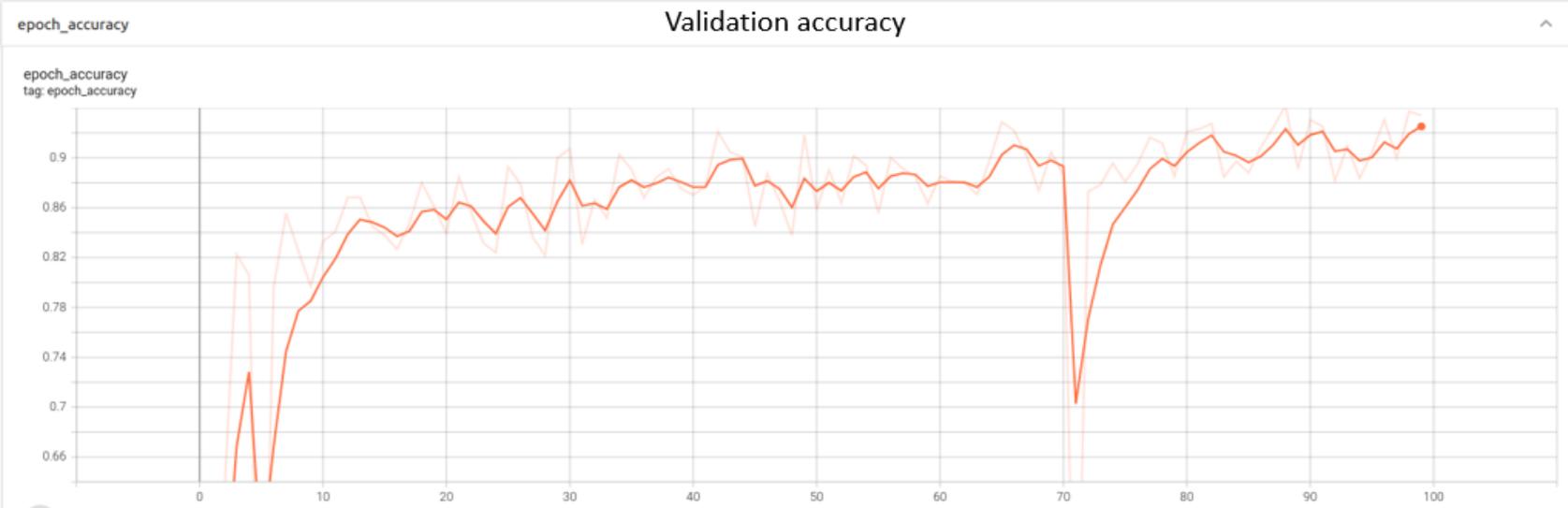
Les lignes en orange correspondent aux images rajoutées après le 13/07 car leur catégorie montrait des IOU et F1-Score moyens plus faibles.

## Annexe 4 : Évolution de la précision lors du premier entraînement de réseau de neurones

A



B



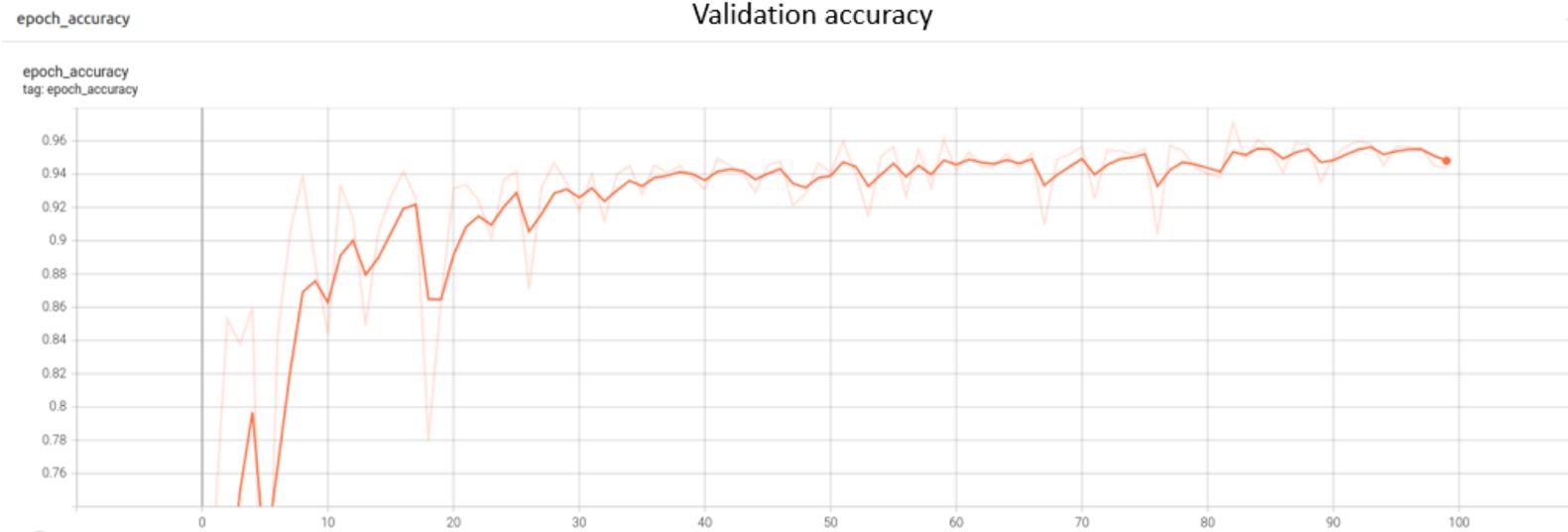
A) précision lors des phases d'entraînement. B) précision lors des phases de validation.

## Annexe 5 : Évolution de la précision lors du 2<sup>ème</sup> entraînement de réseau de neurones

A

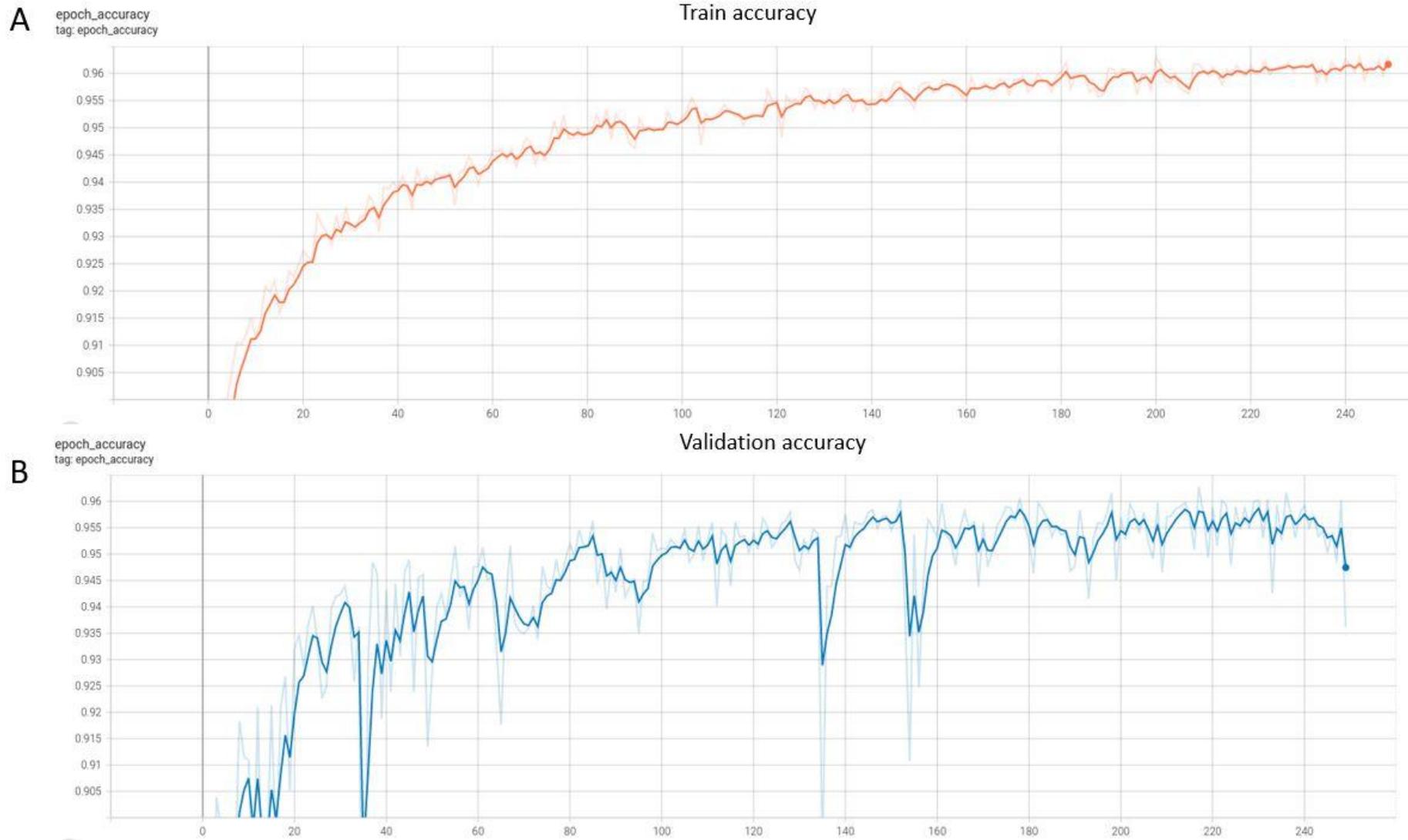


B



A) précision lors des phases d'entraînement. B) précision lors des phases de validation.

## Annexe 6 : Évolution de la précision lors du dernier entraînement de réseau de neurones



A) précision lors des phases d'entraînement. B) précision lors des phases de validation.

## Bibliographie

- [1] « Groupe phosphate : définition et explications ». <https://www.aquaportail.com/definition-13236-groupe-phosphate.html>
- [2] « Rapport d'activité INRAE 2022 ». <https://www.inrae.fr/actualites/rapport-dactivite-inrae-2022>
- [3] « Nous connaître », *INRAE Institutionnel*. <https://www.inrae.fr/nous-connaître>
- [4] « Collaborer avec nous », *INRAE Institutionnel*. <https://www.inrae.fr/collaborer>
- [5] « Site WEB GAFL - Accueil Unité GAFL ». <https://www6.paca.inrae.fr/gafl>
- [6] « Site WEB GAFL - Equipes de recherche ». <https://www6.paca.inrae.fr/gafl/Equipes-de-recherche>
- [7] « EMMAH - Présentation ». <https://www6.paca.inrae.fr/emmah/Qui-sommes-nous/Presentation>
- [8] « EMMAH - Equipe CAPTE ». <https://www6.paca.inrae.fr/emmah/Equipes-de-recherche/Equipe-CAPTE>
- [9] « Prunus L. | Plants of the World Online | Kew Science », *Plants of the World Online*. <http://powo.science.kew.org/taxon/urn:lsid:ipni.org:names:30003057-2>
- [10] P. Lambert, E. Dirlewanger, et F. Laurens, « La sélection assistée par marqueurs (SAM) chez les arbres fruitiers: une approche prometteuse au service de l'innovation variétale », 2009.
- [11] « phénotype - Définitions, synonymes, conjugaison, exemples | Dico en ligne Le Robert ». <https://dictionnaire.lerobert.com/definition/phenotype>
- [12] J.-P. Onnela et S. L. Rauch, « Harnessing Smartphone-Based Digital Phenotyping to Enhance Behavioral and Mental Health », *Neuropsychopharmacology*, juin 2016,
- [13] J. Torous, M. V. Kiang, J. Lorme, et J.-P. Onnela, « New Tools for New Research in Psychiatry: A Scalable and Customizable Platform to Empower Data Driven Smartphone Research », *JMIR Ment Health*, mai 2016,
- [14] 677 Huntington Avenue, Boston, et Ma 02115, « Your phone knows how you feel », *Harvard Public Health Magazine*, 19 juillet 2016. [https://www.hsph.harvard.edu/magazine/magazine\\_article/your-phone-knows-how-you-feel/](https://www.hsph.harvard.edu/magazine/magazine_article/your-phone-knows-how-you-feel/)
- [15] « Plant Phenomics, From Sensors to Knowledge », *Current Biology*, août 2017,
- [16] B. Pallas, S. Martinez, O. Simler, E. Carrié, E. Costes, et F. Boudon, « Assessing T-LiDAR technology for high throughput phenotyping apple tree topological and architectural traits », *Acta Hort.*, juin 2020,
- [17] J. Xue, Y. Fan, B. Su, et S. Fuentes, « Assessment of canopy vigor information from kiwifruit plants based on a digital surface model from unmanned aerial vehicle imagery », *International Journal of Agricultural and Biological Engineering*, Art. n° 1, févr. 2019,
- [18] W. Sangjan et S. Sankaran, « Phenotyping Architecture Traits of Tree Species Using Remote Sensing Techniques », *Transactions of the ASABE*, 2021,
- [19] Berenguer et Egea, « Premiers résultats sur le comportement de l'abricotier "Bulida" sur différents porte-greffe. », *Fruits*, déc. 1977.
- [20] F. Ammad, I. Berkane, et M. Licir, « Etude de comportement de sept (07) variétés du cerisier *Prunus avium* L. Nouvellement introduites dans la région montagneuse de Benchicao (W. Médéa) », 2019.
- [21] K. P. Bleiker, B. S. Lindgren, et L. E. Maclauchlan, « Characteristics of subalpine fir susceptible to attack by western balsam bark beetle (Coleoptera: Scolytidae) », *Canadian Journal of Forest Research*, févr. 2011,

- [22] Á. Montesinos, G. Thorp, J. Grimplet, et M. J. Rubio-Cabetas, « Phenotyping Almond Orchards for Architectural Traits Influenced by Rootstock Choice », *Horticulturae*, Art. n° 7, juill. 2021,
- [23] D. G. McCullough et M. R. Wagner, « Evaluation of four techniques to assess vigor of water-stressed ponderosa pine », *Canadian Journal of Forest Research*, févr. 2011,
- [24] F. Kappel et M. Bouthillier, « Rootstock, severity of dormant pruning, and summer pruning influences on peach tree size, yield, and fruit quality », *Canadian Journal of Plant Science*, mars 2011,
- [25] S. R. MILLER, « SELECTION CRITERIA IN THE SEEDLING STAGE FOR PREDICTING APPLE ROOTSTOCK VIGOR », *Canadian Journal of Plant Science*, mars 2011,
- [26] G. D. Amman, M. D. McGregor, R. F. Schmitz, et R. D. Oakes, « Susceptibility of lodgepole pine to infestation by mountain pine beetles following partial cutting of stands », *Canadian Journal of Forest Research*, févr. 2011,
- [27] W. V. Welker et D. M. Glenn, « Sod Proximity Influences the Growth and Yield of Young Peach Trees », *Journal of the American Society for Horticultural Science*, nov. 1989,
- [28] Y.-C. Du, M. Muslikhin, T.-H. Hsieh, et M.-S. Wang, « Stereo Vision-Based Object Recognition and Manipulation by Regions with Convolutional Neural Network », *Electronics*, Art. n° 2, févr. 2020,
- [29] J. P. Cocquerez, S. Philipp, et R. Zeboudj, « Comparaison de méthodes de segmentation d'images », sept. 1995.
- [30] A. Cornia *et al.*, « Méthode de type Maximum de Vraisemblance pour la détection d'exoplanètes par imagerie différentielle spectrale et angulaire ».
- [31] G. M. Foody et A. Mathur, « Toward intelligent training of supervised image classifications: directing training data acquisition for SVM classification », *Remote Sensing of Environment*, oct. 2004,
- [32] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, et S. Homayouni, « Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review », *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020,
- [33] Z. Liu *et al.*, « Deep learning based brain tumor segmentation: a survey », *Complex Intell. Syst.*, févr. 2023,
- [34] M. Gao, F. Zheng, J. J. Q. Yu, C. Shan, G. Ding, et J. Han, « Deep learning for video object segmentation: a review », *Artif Intell Rev*, janv. 2023,
- [35] « Deep learning-based crack segmentation for civil infrastructure: data types, architectures, and benchmarked performance », *Automation in Construction*, févr. 2023,
- [36] O. Ronneberger, P. Fischer, et T. Brox, « U-Net: Convolutional Networks for Biomedical Image Segmentation », 18 mai 2015.
- [37] X.-Y. Zhou et G.-Z. Yang, « Normalization in Training U-Net for 2-D Biomedical Semantic Segmentation », *IEEE Robotics and Automation Letters*, avr. 2019,
- [38] C. Deledalle, « Chapter IV – Image classification and CNNs ».
- [39] O. Animour et E. Chihani Houssam, « Artificial intelligence paradigms for the automated diagnosis of brain cancer », 2022.
- [40] A. Z. Atiyah et K. H. Ali, « Brain MRI Images Segmentation Based on U-Net Architecture », *IJEEE Journal*, 2021.

- [41] L. Chabbi et A. Chellouche, « Classification d'images de textile par les réseaux de neurones Convolutionnels(CNN). », Thesis, 2022.
- [42] « Overview — Python 3.6.0 documentation ». <https://docs.python.org/release/3.6.0/>
- [43] « 3.11.2 Documentation ». <https://docs.python.org/release/3.11.2/>
- [44] H. Wickham, R. François, L. Henry, K. Müller, et RStudio, « dplyr: A Grammar of Data Manipulation ». 28 avril 2022.
- [45] J. Cheng *et al.*, « leaflet: Create Interactive Web Maps with the JavaScript “Leaflet” Library ». 23 mars 2022.
- [46] H. Wickham *et al.*, « ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics ». 3 mai 2022.
- [47] H. Wickham et RStudio, « stringr: Simple, Consistent Wrappers for Common String Operations ». 10 février 2019.
- [48] S. Jay, « Woodpicker ». 26 mai 2023. [Windows, Linux].
- [49] A. Kirillov *et al.*, « Segment Anything », 5 avril 2023.
- [50] A. Rosebrock, « Intersection over Union (IoU) for object detection », *PyImageSearch*, 7 novembre 2016. <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
- [51] A. Krizhevsky, I. Sutskever, et G. E. Hinton, « ImageNet classification with deep convolutional neural networks », *Commun. ACM*, mai 2017,
- [52] « math — Mathematical functions », *Python documentation*. <https://docs.python.org/3/library/math.html>
- [53] « Matplotlib — Visualization with Python ». <https://matplotlib.org/>
- [54] « NumPy ». <https://numpy.org/>
- [55] « random — Generate pseudo-random numbers », *Python documentation*. <https://docs.python.org/3/library/random.html>
- [56] pythonforge, « Le module sys pour interagir avec l'interpréteur Python », *Pythonforge*, 30 janvier 2021. <https://pythonforge.com/module-sys-python/>
- [57] « argparse — Parser for command-line options, arguments and sub-commands », *Python documentation*. <https://docs.python.org/3/library/argparse.html>
- [58] « copy — Opérations de copie superficielle et récursive », *Python documentation*. <https://docs.python.org/3/library/copy.html>
- [59] « 11.7. glob — Recherche de chemins de style Unix selon certains motifs — Documentation Python 3.6.15 ». <https://docs.python.org/fr/3.6/library/glob.html>
- [60] « openpyxl: A Python library to read/write Excel 2010 xlsx/xlsm files ». [MacOS :: MacOS X, Microsoft :: Windows, POSIX].
- [61] « os — Miscellaneous operating system interfaces », *Python documentation*. <https://docs.python.org/3/library/os.html>
- [62] « Package overview — pandas 2.0.3 documentation ». [https://pandas.pydata.org/pandas-docs/stable/getting\\_started/overview.html](https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html)
- [63] « shutil — High-level file operations », *Python documentation*. <https://docs.python.org/3/library/shutil.html>
- [64] « imageio: Library for reading and writing a wide range of image, video, scientific, and volumetric data formats. » [MacOS :: MacOS X, Microsoft :: Windows, POSIX].
- [65] « opencv-python: Wrapper package for OpenCV python bindings. » [MacOS, Microsoft :: Windows, POSIX, Unix].

- [66] « scikit-image: Image processing in Python ». [MacOS, Microsoft :: Windows, POSIX, Unix].
- [67] « scikit-learn: A set of python modules for machine learning and data mining ». [MacOS, Microsoft :: Windows, POSIX, Unix].
- [68] « albumentations: fast image augmentation library and easy to use wrapper around other libraries ». [OS Independent].
- [69] « datasets: HuggingFace community-driven open-source library of datasets ». [OS Independent].
- [70] « evaluate: HuggingFace community-driven open-source library of evaluation ». [OS Independent].
- [71] «  Hub client library ». [https://huggingface.co/docs/huggingface\\_hub/index](https://huggingface.co/docs/huggingface_hub/index)
- [72] « Keras: Deep Learning for humans ». <https://keras.io/>
- [73] « tensorflow: TensorFlow is an open source machine learning framework for everyone. »
- [74] « torch: Tensors and Dynamic neural networks in Python with strong GPU acceleration ».
- [75] « torchvision: image and video datasets and models for torch deep learning ».
- [76] « collections — Types de données de conteneurs », *Python documentation*. <https://docs.python.org/3/library/collections.html>
- [77] « datetime — Types de base pour la date et l'heure », *Python documentation*. <https://docs.python.org/3/library/datetime.html>
- [78] « itertools — Fonctions creating iterators for efficient looping », *Python documentation*. <https://docs.python.org/3/library/itertools.html>
- [79] « json — Encodage et décodage JSON », *Python documentation*. <https://docs.python.org/3/library/json.html>
- [80] « multiprocessing — Process-based parallelism », *Python documentation*. <https://docs.python.org/3/library/multiprocessing.html>
- [81] M. Waskom, « seaborn: statistical data visualization », *JOSS*, avr. 2021,
- [82] « time — Accès au temps et conversions », *Python documentation*. <https://docs.python.org/3/library/time.html>
- [83] « urllib — Modules de gestion des URLs », *Python documentation*. <https://docs.python.org/3/library/urllib.html>