UNIVERSITE DE PAU ET DES PAYS DE L'ADOUR

CIRAD-Montpellier

Option : Méthodes Stochastiques et Informatique pour la Décision
Promotion : 2018-2019

Rapport Final

Évaluation des traits architecturaux et de fonctionnement d'arbres fruitiers par imagerie aéroportée

Mr. **A.Rafik**
Promo 2018/2019
Université de Pau et Pays de l'Adour

Encadrants : F.Boudon
B.Pallas
M.Delalande
E.Faure
M.Chaumont

Remerciements

La réalisation de ce stage a été possible grâce à l'aide de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Dans un premier temps, je voudrais remercier mes encadrants : Fréderic Boudon, Benoit Pallas, Magalie Delalande, Emmanuel Faure, Marc chamont et Jean Luc Regnard pour l'aide qu'ils m'ont apporté et le suivi de mon travail.

Je remercie également le GIS Fruit pour l'organisation et le financement de mon stage.

Je remercie ma femme pour son soutien autant moral que financier et pour son sacrifice ainsi que mes deux parents et mes frères pour leurs conseils.

Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de ma démarche.

Table des matières

Introduction Générale

Dans le contexte climatique actuel, les épisodes de sécheresse apparaissent de plus en plus fréquentes notamment dans les zones de culture méditerranéenne. D'ici à 2050, il faudra nourrir près de 9 milliards personnes. La production agricole devra donc doubler, et puiser encore plus dans les réserves en eau. Face à de telles contraintes, les chercheurs en agronomie ont d'immenses défit à relever pour faire que demain nous ayons une agriculture durable en préservant au mieux les ressources. Dans l'objectif d'accompagner les changements agricoles ([1]) dus aux changements climatiques, les chercheurs ont commencé à étudier leurs impacts sur les arbres fruitiers.

De nombreuses études ont été faites pour éclaircir et analyser le fonctionnement [1] et le développement architectural [2] des arbres en réponse à l'environnement ([2], [3]) afin de sélectionner des variétés d'arbres fruitiers plus tolérantes, c-à-d capables de maintenir leurs performances de production dans des conditions de culture sub-optimales.

Pour atteindre ces objectifs de sélection, il est nécessaire de développer des méthodes de phénotypage [3] à haut débit pour pouvoir évaluer la diversité variétale d'un grand nombre d'arbres en analysant notamment l'efficience de l'utilisation de la ressource hydrique. Une partie des méthodes de phénotypage à haut débit [4] est basée sur les techniques d'analyse d'images aériennes.

Dans le cadre de notre étude, nous travaillons sur des données d'imagerie multi-spectrales et thermiques aéroportées par drone [4]. La démocratisation de la télédétection ainsi que l'accessibilité de la technologie des drones (embarquant des capteurs multi-spectraux, thermiques...) et les tarifs de plus en plus abordables, ont permis l'émergence de l'acquisition d'images à haute résolution et pour différentes longueurs d'ondes.

1. **Fonctionnements des arbres :** Photosynthèse des feuilles, transpiration, conductance stomatique...
2. **Architecture des arbres :** La hauteur des plantes, nombre de ramifications, surface totale du feuillage...
3. **Phénotypage :** Consiste à identifier les caractéristiques physiques et physiologiques d'un nombre fini de génotypes pour certains caractères d'intérêts agricoles, économiques... comme la résistance aux maladies, la résistance à la sècheresse...**Le phénotypage** est une activité consommatrice en temps, main d'oeuvre et moyens de mesure.
4. **Le phénotypage à haut débit :** est l'étude de différents génotypes en réponse à différents traitements environnementaux. **Le phénotypage à haut débit** fait appel à des outils numériques, souvent à une chaîne de mesures automatisées permettant la caractérisation d'un grand nombre de génotypes pour de nombreux trait d'intérêt en un minimum du temps en préservant les arbres. L'imagerie aéroportée par drone avec des caméras RVB(Rouge, Vert, Bleu), multi-spectrale et thermique est de plus en plus utilisée à des fin de phénotypage au champ et de caractérisation de grandes collections de diversités. Source : Christophe Salon, `https://www.vitagora.com/blog/2019/agroecologie-phenotypage-hautdebit`.

L'acquisition de données dans plusieurs bandes du spectre solaire visible (VIS), proche infrarouge (NIR), l'infrarouge thermique (TIR), permet l' étude précise et simultanée du fonctionnement et de la structure d'arbres. Les bandes VIS, NIR permettent le calcul des indices de végétation de fonctionnements (NDVI [5], GNDVI [6], MCARI2 [7], ...), de stress (PRI) et de la température du couvert végétal (Température de surface : Ts) par la bande spectrale TIR. Il a été montré, qu'il est possible d'utiliser l'imagerie multi-spectrale et thermique pour le calcul de plusieurs indicateurs de végétation qui sont en corrélation avec la structure de la canopée et l'état du fonctionnement des arbres à l'échelle du champ (Berni (2009), [5]). D'autre part, Virlet (2014) [6] a pu montrer que l'emport de la caméra thermique sur le drone permet le calcul d'un indice du stress qui peut être considéré comme un proxy du taux de transpiration relatif au feuillage. Delalande et al (2018)[7] ont évalué les caractéristiques phénotypiques du pommier liées au stress hydrique à partir de différentes bandes spectrales de l'imagerie aéroportée par drone sur une collection de diversité variétale du pommier en 2017.

Pour obtenir des indicateurs précis de l'état de la végétation, il faut d'abord identifier les arbres et leurs feuillage. A l'heure actuelle, cette étape est difficile et longue à réaliser et necessite des interventions manuelles (création d'exemples de zone de végétation dans l'image). Pour ce faire, il existe des outils qui semblent intéressants à adopter pour améliorer la rapidité et la fiabilité d'identification des feuillages : les méthodes de **deep learning**.

Dans notre travail, nous nous focalisons sur les méthodes d'**apprentissage profond** de classe d'apprentissage automatique [8]supervisé [9].

Les méthodes de **deep learning** sont puissantes et efficaces pour ce genre de tâche dans l'imagerie. De nombreux chercheurs se sont intéressés à ce type de méthodes pour leurs performances et adaptabilité. Elles sont basées sur les données et ne nécessitent aucune hypothèse préalable sur les variables à étudier, ce qui n'est pas le cas dans les méthodes d'analyse statistique et mathématiques classiques. Elles sont beaucoup utilisées dans différents domaines : traitements et analyse de l'image(pour les tâches de classification d'image, détection d'objets dans une image)... [8].

5. **NDVI :** Normalized Difference Vegetation Index.
6. **GNDVI :** Green Normalized Difference Vegetation Index.
7. **MCARI2 :** Modified Chlorophyll Absorption in Reflectance Index
8. **Apprentissage automatique :** est une approche statistique, constituée de deux étapes fondamentales, la première de l'entrainement, qui consiste à la construction, en faisant de l'estimation du modèle pour la réalisation d'une tâche pratique et la secode étape de production et de prédiction. Source : `https://fr.wikipedia.org/wiki/Apprentissage_automatique`
9. **Apprentissage supervisé :** consiste à apprendre un modèle de prédiction à partir des données annotées.

Ce stage s'inscrit en préambule d'un projet ambitieux d'"Évaluation des traits architecturaux et de fonctionnement des arbres fruitiers (pommiers) par imagerie aéroportée et données LiDAR".

Ce projet est conçu pour l'évaluation de l'apport des méthodes de deep learning pour les tâches :

— Segmentation et identification des organes des arbres, particulièrement des pixels des feuilles des arbres.
— Alignement de l'imagerie aéroportée et données LiDAR
— Projection de l'information issue de l'imagerie aéroportée sur la structure 3D issue du LiDAR.

Dans le cadre de mon stage, nous avons réalisé la "segmentation et l'identification des feuillages des arbres" dans les images multi-spectrales et thermiques. Dans un premier temps, nous nous sommes posé la question de recueillir les valeurs spectrales du feuillage des pommiers du reste de la parcelle. Cela nécessite au préalable de réaliser une segmentation. Pour cela, nous avons choisi la méthode **U-NET** afin de séparer des pixels du feuillage des arbres de l'arrière plan. **U-NET** est une méthode connue pour sa précision de prédiction à l'échelle du pixel et sa rapidité en temps de calcul, grâce à sa structure "Décodeur-Encodeur". Elle peut être appliquée à un petit échantillons d'images.

Cette méthode **U-NET** a généralement été appliquée à des images RVB (Rouge, Vert, Bleu) ayant des valeurs entières positives en 0 et 255, donc de tailles et de statistiques différentes de celles des images de notre jeu de données. Pour pouvoir appliquer cette méthode, nous l'avons adaptée à notre jeu de données par des paramétrisations spécifiques au cœur de son architecture. Comme nous travaillons sur des images multi-spectrales et thermiques à 7 longueurs d'ondes avec des valeurs réelles plus petites et de différentes grandeurs (réflectance et température), pour adapter ces valeurs des images en entrée, il nous a fallu normaliser les données. De plus, nous voulons garder des tailles d'images en entrée pour les masques en sortie, afin de pouvoir faire l'extraction des valeurs des pixels des feuilles. Pour cela, nous avons adapté les paramètres du modèles (voir, chapitre U-NET). Une fois la segmentation réalisée, nous proposons d'utiliser la méthode **WaterShed** pour identifier chaque arbre individuel. Finalement, a partir des pixels de chaque arbres, des indices de végétation sont dérivés et nous donne potentiellement des informations sur l'état végétatif de chaque arbre. Cette approche et ces résultats sont présentés dans le chapitre 4.

Une contrainte du premier modèle que nous avons mis en place est qu'elle nécessite les coordonnées GPS de chaque arbre pour les identifier grâce au **WaterShed**. Pour simplifier cette contrainte, nous avons dans une deuxième partie de mon stage étudié la possibilité d'utiliser une méthode de segmentation par instance (voir, Figure 1). Pour la réalisation des cette tâche, nous avons choisi la méthode **Mask R-CNN** [44]. Nous présentons nos premiers tests avec cette méthode dans le chapitre 5.

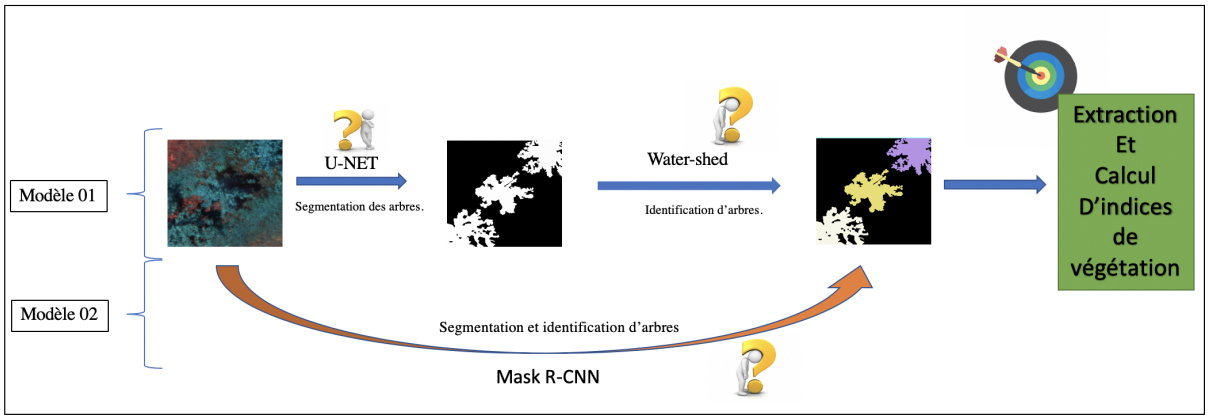


FIGURE 1 – Plan et objectif du stage.

Chapitre 1

Présentation des organismes d'accueil

Introduction :

Dans ce chapitre, je vais présenter les instituts dans lesquels j'ai été accueilli, les équipes avec lesquelles j'ai travaillé. Durant mon stage, j'ai travaillé avec les équipes **AFEF**et **M2P2** de l'UMR AGAP spécialisé sur la modélisation et le phénotypage des pommiers et avec l'équipe **ICAR** du **LIRMM** sur les méthodes de deep learning.

Le stage a été réalisé sous la tutelle de B.Pallas (Chargé de recherche, INRA) spécialisé dans la modélisation des plantes, M.Delalande(Ingénieur de recherche, INRA) spécialisée dans le phénotypage aéroporté, F.Boudon (Chercheur PhD, CIRAD) spécialisé dans modélisation, traitement et analyse de données de phénotypage de plantes de l'**UMR AGAP**, ainsi qu' E.Faure (Chercheur du CNRS, LIRM) et M.Chaumont (Enseignant chercheur, Université de Nîmes) du LIRMM spécialisés dans le traitement et analyse d'image et le deep learning.

1.1 Les laboratoires et les équipes

1.1.1 L'UMR AGAP

L'**U**nité **M**ixte de **R**echerche **A**mélioration **G**énétique et **A**daptation des **P**lantes [1] a été créée en Janvier 2011 dans l'objectif de produire, de diffuser des connaissances dans le domaine de l'agronomie et de répondre aux enjeux sociétaux de l'agriculture. Elle est placée sous triple tutelle : **CIRAD**, **INRA** et **Montpellier SupAgro**. **L'UMR AGAP** est une grande unité mixte de Recherche constituée de 14 équipes cordonnées autour de trois thématiques de base : "Diversités & Génomes : structure, domestication, milieux, sociétés", "Développement et adaptation des plantes et des peuplements", "Approches intégrations pour l'innovation variétale". Elle travaille sur différentes cultures tropicales et méditerranéennes (riz, blé, pommier, vigne, olivier, etc.).

L'équipe AFEF

L'équipe de recherche **A**rchitecture et **F**onctionnement des **E**spèces **F**ruitières fait partie de l'unité de recherche **UMR AGAP**. **AFEF** mène des recherches dans trois disciplines, "l'éco-physiologie", "la génétique" et "la modélisation" sur des espèces fruitières pérennes (pommiers, oliviers...). Elle a pour objectif de comprendre et de modéliser le développement architectural de l'arbre sur des bases écophysiologiques et de déterminisme génétique des caractères d'intérêt.

1. **UMR AGAP** :`https://umr-agap.cirad.fr/l-unite`, cite web de l'unité mixte de recherche AGAP

L'équipe M2P2

L'équipe de recherche **M**odèles et **M**éthodes pour le **P**hénotypage des **P**lantes est une équipe de **UMR AGAP** spécialisée dans la modélisation et le développement des méthodes et modèles d'analyse de données de phénotypage des plantes sur des bases de la biologie, l'agronomie, l'informatique et mathématiques appliquées. L'équipe **M2P2** s'oriente vers le développement des méthodes et algorithmes de phénotypage des plantes avec l'objectif d'automatiser le phénotypage à haut débit et remplacer le phénotypage manuel classique.

1.1.2 Le LIRMM

Le **L**aboratoire d'**I**nformatique, de **R**obotique et de **M**icroélectronique de **M**ontpellier [2] est une grande unité mixte de recherches, il a été créé en 1992 sous tutelle du Centre National de la Recherche Scientifique (CNRS), et de l'Université de Montpellier. Il est situé sur le Campus Saint-Priest de l'Université de Montpellier. Le LIRMM regroupe de nombreuses équipes menant les recherches dans trois thématiques principales : l'Informatique, la Robotique et la Microélectronique. Il a pour objectif de développer les connaissances dans le domaine de l'informatique, de former des chercheurs et de produire des objets matériels et logiciels informatiques, etc.

L'équipe ICAR

L'équipe **I**mage & **I**nteraction [3] du département informatique et du département rebotique du **LIRMM**. Elle mène les recherches sur trois thématiques de base : l'analyse et le traitement, le codage et la protection de données, la modélisation et la visualisation de données. L'équipe ICAR a pour objectif d'améliorer et de développer l'implication de la recherche associant l'interaction et le traitement des données visuelles.

1.2 Les instituts

1.2.1 L'INRA

l'**Institut National de la Recherche Agronomique (INRA** [4] **)** est le premier organisme public national français de recherches scientifiques et agronomiques. L'INRA a été crée en 1946, juste après la deuxième guerre mondiale dans le but de répondre aux besoins alimentaires en France et pour différentes missions : Il produit, il transmet et diffuse des nouvelles connaissances dans le domaine de l'agronomie, favorise les échanges entre scientifiques et accompagne le développement agricole.

1.2.2 Le CIRAD

Le **Centre de Coopération Internationale en Recherche Agronomique pour le Développement(CIRAD** [5] est un organisme public de recherche français. Il a été créé en 1984 pour la recherche agronomique dans les régions tropicales et méditerranéennes. Il est placé sous la tutelle du Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation et du Ministère de l'Europe et des Affaires Étrangères.

Il a pour missions de produire et de transmettre les nouvelles connaissances en agronomie. Il a pour but d'accompagner le développement agricole et de contribuer aux débats sur les enjeux majeurs internationaux, en coopération dans un cadre international pour le développement ; il développe ses recherches en fonction de différents enjeux comme la lutte contre la pauvreté, la sécurité alimentaire des populations, etc.

2. **LIRMM** : `http://www.lirmm.fr`, cite web de LIRMM.
3. **ICAR** : `https://www.lirmm.fr/recherche/equipes/icar`
4. INRA :`http://institut.inra.fr`), cite web de l'INRA.
5. CIRAD : `https://www.cirad.fr`, cite web du CIRAD.

1.2.3 Le CNRS

Le Centre National de la Recherche Scientifique [6] est un organisme public français à caractère scientifique et technologique. Il a été créé en octobre 1939, placé sous tutelle du ministère de l'Enseignement Supérieur, de la Recherche et de l'Innovation. Il a pour mission : d'évaluer l'avancement des recherches, développer et diffuser les informations et connaissances scientifiques, coordonner et subventionner la recherche, créer et gérer les unités de recherche, etc.

6. **CNRS** : `http://www.cnrs.fr/fr/le-cnrs`

Chapitre 2

Les données expérimentales

Intrduction :

2.1 Présentation de données expérimentales

2.1.1 Matériel végétal

Notre étude a été effectuée sur une collection [1] [13] de 241 génotypes [2] de pommiers plantés sur 1.2 ha (le champ encadré en rouge dans la figure 2.1). La parcelle est située sur l'unité expérimentale Inra DiaScope [3] à Mauguio (43 ° 36N, 03 ° 58E), 8 km à l'Est de Montpellier, France.

FIGURE 2.1 – Image de parcelle de la core-collection de Mauguio.
Source : Google earth.

1. **Une core collection :** la parcelle de 1000 pommiers, est une collection de base de 241 génotypes de pommiers, qui ont été multipliés et développés à l'INRA D'ANGERS [9] avant la plantation à mauguio.
2. **Un génotype :** un ensemble d'informations génétiques héréditaires d'un individu. Par extension, le génotype désigne aussi la plante porteuse de ces informations génétiques. Source : Wikipedia
3. **L'Unité Expérimentale Inra DiaScope :** est une unité expérimentale et d'étude sur l'adaptation, et diversité des plantes dans différents environnements **INRA**. Source : `https://www6.montpellier.inra.fr/diascope`.

Les arbres ont été plantés en 2014 dans un dispositif expérimental composé de 10 rangées de 200 m de long, chaque rang étant composé de 100 pommiers, à des distances de 5m (entre rangs)× 2m (sur le rang). Les arbres sont irrigués à l'aide de micro-asperseurs situés entre les arbres . Pour pouvoir étudier l'architecture naturelle des différents variétés, les arbres ne sont pas taillés, mais seulement palissés sur des fils de fer horizontaux tendus sur des piquets. Chaque variété est représentée par 4 arbres : 2 arbres sont irrigués à l'optimum et 2 arbres subissent un stress hydrique estival en juin-juillet.

La figure 2.2, montre la répartition des plantations au niveau de la parcelle :

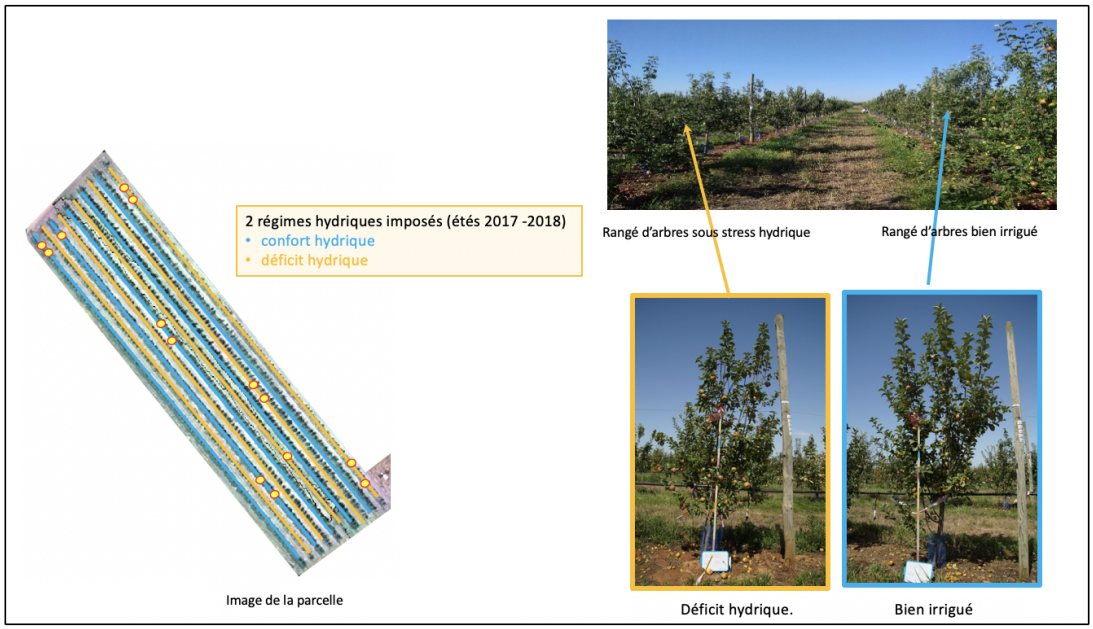


FIGURE 2.2 – Implantation des arbres dans la parcelle.

2.1.2 Acquisition de données par drone

Dans le cadre de cette étude nous nous focaliserons sur les données acquises les 7 et 12 juillet 2017 par un drone Mikrokopter® hexa-rotor à une altitude de ≈ 25 m et par un temps clair et ensoleillé sans nuages avec un vent inférieur à $4ms^{-1}$. Ce drone [9] embarque une caméra AirPhen v3 [4] multi-spectrale avec 6 capteurs : un capteur de réflectance dans le bleu (450nm), 2 capteurs dans le vert (530 et 570nm) et un capteur dans le Proche Infra-Rouge (850nm). La caméra FLIRTau2.7, capte les valeurs dans la gamme LWIR (7,5 - 13m) avec une résolution de 640 x 512 pixels.

Le drone dispose d'un nano ordinateur lié à une carte GPS, le GPS tague chaque image avec de nombreuses méta-données, dont sa position GPS, son altitude et son attitude (c-à-d la position en 3 D dans laquelle il se trouve au moment de la prise de vue). Ces images ont été prises en vue zénithale.

Sur la parcelle sont disposées des cibles, géo-référencées précisement grâce à l'utilisation d'un GPS centimétrique. Lors de la réalisation de l'ortho-mosaique (avec Agisoft photoscan), ces points GPS, les Ground Control Points (GCP) permettent une ortho-rectification et un géo-référencement précis de la dalle, ce qui permettra de superposer les ortho-images aux positions GPS de chaque arbre, relevées également au GPS centimétrique, afin d'identifier chaque variété lors des traitements ultérieurs.

La bande infrarouge thermique permet de calculer la température au niveau du feuillage. Elle est obtenue par une régression linéaire [5] entre les valeurs des pixels des cibles thermiques et la température réelle mesurée sur ces cibles avec un radio thermomètre IR120 (Campbell) calibré.

4. AirPhen v3 : `www.hiphen-plant.com`

5. **Régression linéaire [9] :** est une relation entre la valeur moyenne des pixels des cibles chaude, froide, humide, sèche et les températures IR 120.

2.1.3 Pré-traitement des images et construction des ortho-mosaïques multi-spectrales et thermiques

Avant mon arrivée, toutes les données ont été pré-traitées selon le même protocole. Pour chaque vol, on acquiert $\simeq$ 800-1000 images (35 GO, 1 minute de vol $\simeq$ 1 Go de données). Dans un premier temps, on fait une sélection visuelle en supprimant les images de décollage et atterrissage, et les images floues (très peu nombreuses). À la fin, une fois les images transformées, on les aligne avec Agisoft : corrections des erreurs de la lentille, alignement, construction du nuage de point dense 3D, contruction du DEM -Digital Elevation Model- sur le nuage de points, puis création de l'ortho-mosaique sur le DEM et on réalise une ortho-mosaique.

La surface de la parcelle étant importante (1,20 ha) il faut réaliser 3 vols de 10 à 12 minutes (durée de la batterie du drone) pour en assurer une couverture intégrale de bonne qualité. Avec les images de chaque vol, on réalise une ortho-mosaique (voir, Figure 12), qu'on va nommer des dalles (les dalles Ouest, Centre et Est).

2.1.4 Annotation des données

L'annotation des données est une étape délicate et importante pour notre étude. Ces annotations expérimentales ont été effectuées manuellement sur différentes parties de la dalle, puis extraite en **subset** c-à-d par zone de dimension réduite avec le logiciel ERDAS©IMAGINE. Elles consistent à la construction des polygones des pixels couvrant l'ensemble des feuilles des arbres, sur lesquels chaque pixel a été attribué manuellement à une des classes (classe "Feuilles" ou classe "Autre"). Ces annotations constituent la "vérité terrain". L'affichage de la mosaïque a été faite en trois canaux permettant une meilleure visualisation. Certaines longueurs d'onde comme le 730 et le 850nm permettent une bonne distinction entre l'arbre et l'arrière plan (le sol enherbé...). Les étapes suivies pour l'annotation et la construction sont décrite dans la figure ci-dessous (voir, Figure 2.3). Au total, nous avons annoté 45 subsets de tailles variant entre 250×250 et 600×560 pixels.

Durant l'annotation des données, nous avons fait face à plusieurs difficultés : les arbres sont en port libre, c'est à dire qu'ils ne sont pas taillés, ce qui engendre des volumes et des formes des couverts des arbres non homogènes. Le sol enherbé rend la tâche difficile, il est difficile visuellement de différencier le vert des feuilles des arbres du mélange végétal au sol. Enfin, les arbres sont jeunes, ils ont un couvert discontinu, on voit le sol entre les branches de certains arbres. Cette tâche de construction de données a été très longue en temps et coûteuse en mémoire.

2.1.5 Extraction de données

L'imagerie multi-spectrale et thermique permet plusieurs bandes de réflectance et de température, que nous allons nommer par la suite : B1 de reflectance du bleu (450 nm), B2, B3 de reflectance du vert (530 et 570nm) respectivement, B4 de reflectance du rouge (675nm), B5 de reflectances de red edge (730nm) et B6 de NIR (850nm) et B7 de la camera thermique FLIR® TAU2.7 pour la température IR, LWIR (7,5-13 m).

L'extraction des subsets a généré des fichiers composés de 9 colonnes (voir le tableaux Excel, Figure 2.3) représentant les colonnes décrites dans la section 2.1.2 plus la colonne B8 [6] et B9 est la colonne de la classe "feuilles" ou "autres". Au total, nous avons annoté 45 subsets, situés sur différents partie dans les dalles.

La figure 2.3, montre le processus d'annotation, d'extraction et construction de données :

6. **B8** : est la colonne des identifiants des arbres. Cette identification a été faite de façon manuelle, par superposition des cercles de pixels sur les centres de gravité des arbre. Les pixels des cercles sont labellisés par les numéros des arbres au niveau parcelle.

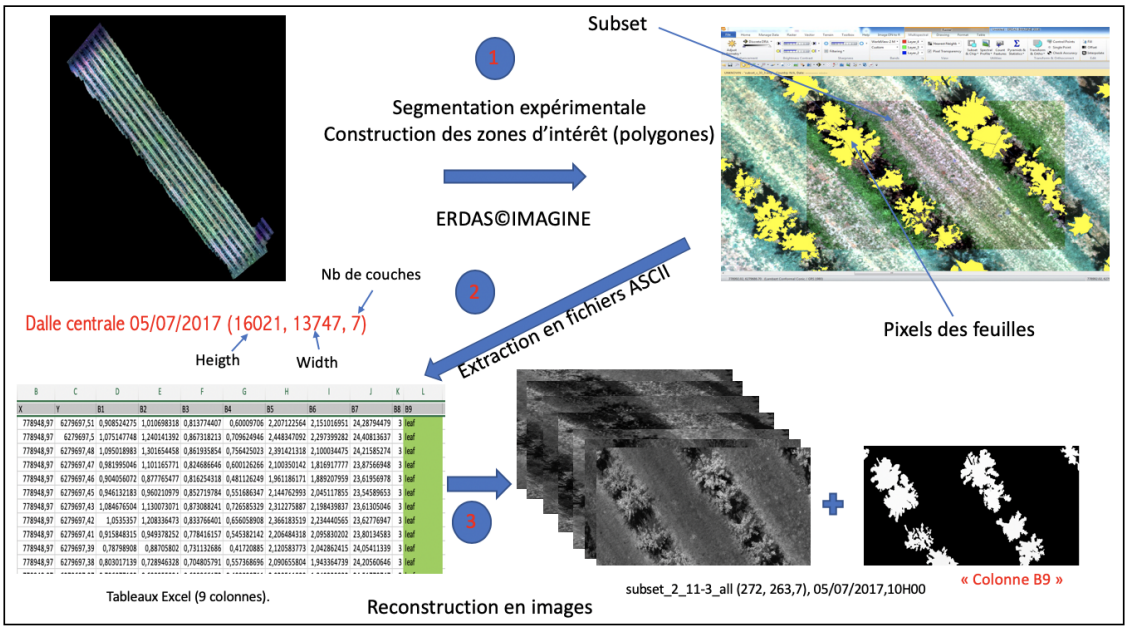


FIGURE 2.3 – Schéma d'annotation et de construction de données.

2.1.6 Calcul des Indices de Végétation et de la Température de surface du feuillage

L'extraction des valeurs des pixels permet d'obtenir des informations sur le comportement et le fonctionnement des arbres en calculant les indices de végétation. Ces indices sont calculés à partir de valeurs des pixels individuels. Pour chaque pixels feuilles nous calculons : les indices de végétation (2.1) et la température de surface, ainsi que l'écart entre la température de surface du feuillage et la température de l'air, puis nous calculons la moyenne, écart-type, minimum, maximum par arbre. Le tableau ci-dessous montre les bandes utilisées pour le calcul de quelques indices de végétation qui nous intéressent :

TABLE 2.1 – Tableau des indices de végétation :

Indices	Équation du calcul	Intérêts
NDVI	$(R850 - R675)/(R850 + R675)$	indice des couverts végétaux
GNDVI	$(R850 - R570)/(R850 + R570)$	la concentration en chlorophylle...
MCARI2	$\frac{1.5[2.5(R850-R675)-1.3(R850-R570)]}{\sqrt{(2R850+1)-(6R850-5R675)-0.5}}$	chlorophylle..
PRI	$(R570 - R530)/(R570 + R530)$	efficience photo-synthétique, chlorophylle...

2.2 Analyse descriptive de données

Pour étudier la dispersion des valeurs des variables de réflectaces et de température, nous avons effectué une analyse descriptive, une analyse des corrélations et une **Analyse des Composantes Principales** de manière séparée entre les deux dates d'acquisition sur les différents subsets, et au sein des subsets.

2. **Chlorophylle** : indice de masse verte et d'efficience d'interception du rayonnement solaire.
3. **NDVI** : Normalized Vegetation Index, développé par Rouse al, 1974.
4. **GNDVI** : Green Normalized Vegetation INdex, développé par Gitelson al, 1996.
5. **MCARI2** : Modified Chlorophyll Absorption Ratio Index 2, développé par Haboudane al, 2004.
6. **PRI** : Photochemical Réflectance Index, développé par Gamon al, 1992.

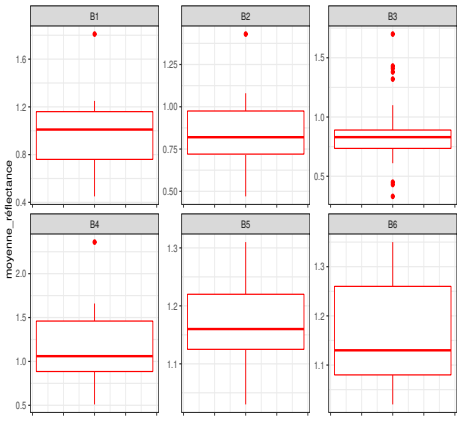
1. **Note :** Ces analyses statistiques descriptives, d'**ACP** et visualisations graphiques ont été effectuées sous le logiciel r-studio, avec les packages "factoextra" et "factominer".

Le tableau ci-dessous a pour but de décrire la variabilité intra et inter-subset. Nous avons calculé les moyennes totales des moyennes et écart-types, ainsi que les écart-types des moyennes et écart-types totaux des moyennes et écart-types des réflectences et températures de l'ensemble des subsets des deux dates d'acquisition.

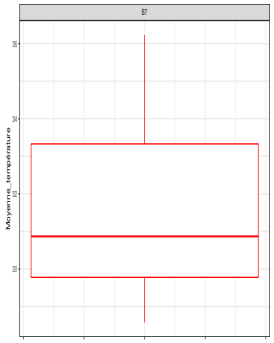
TABLE 2.2 – Tableau descriptif des variables (B$i, i = 1..7$) des réflectances et des températures des subsets sur l'ensemble des subsets des deux dates :

Subsets	Date et heure d'acquisition	Moyenne, Ecart type, Coefficient de variation relatif						
		B1	B2	B3	B4	B5	B6	B7
Moyenne totale	12/07/2017 à 10h00	6.2,2.8,0.4	1.5, 0.6, 0.3	2.8,0.9,0.3	1.6,0.8,0.4	1.7,0.5,0.3	1.2,0.4,0.3	39.3,6.7,0.1
Ecart type total	12/07/2017 à 10h00	0.69, 0.19	0.22, 0.05	0.22 , 0.04	0.24, 0.08	0.04, 0.05	0.04, 0.03	1.68, 0.86
Moyenne totale	05/07/2017 à 10h00	0.9,0.5,0.5	0.8,0.3,0.4	0.8,0.3,0.3	1.1,0.7,0.6	1.1,0.3,0.3	1.1, 0.3, 0.3	31.6, 6.9, 0.2
Ecart type total	05/07/2017 à 10h00	0.30, 0.16	0.19, 0.06	0.32, 0.11	0.39, 0.22	0.07, 0.06	0.10, 0.07	2.36, 1.08

Par la suite, nous avons construit des boites à moustaches pour les moyennes des variables reflectances (B$i, i = 1..6$) et températures B7 des subsets de deux dates.

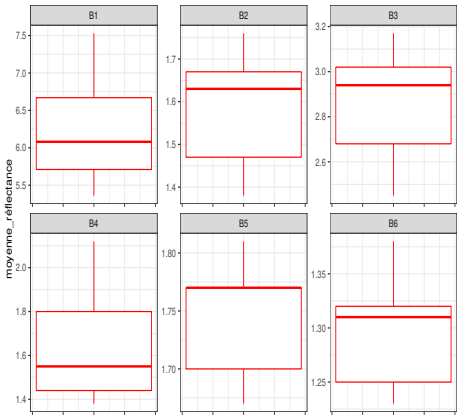


Valeurs moyennes des valeurs des réflectances(nm) (a)

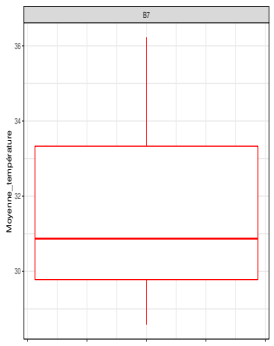


Valeurs moyennes des températures(Degré Celsius) (b)

FIGURE 2.4 – Boites à moustaches des moyennes des variables (B$i, i = 1..7$) de réflectance et température des subsets du 05/07/2017.



Valeurs moyennes des valeurs de réflectances(nm) (a)



Valeurs moyennes des températures(Degré Celsius) (b)

FIGURE 2.5 – Boites à moustaches des moyennes des(B$i, i = 1..7$) de réflectance et température (B$i, i = 1..7$) des subsets du 12/07/2017.

1. **Note :** le trait en rouge au milieu correspond la médiane, et la boite au deuxieme et troisieme quartiles. les points rouges l'extérieur de la boite sont les moyennes des de variables des subsets.

Le tableau et les boites à moustache ci-dessus sont pour la comparaison inter et intra-subsets de la variation des variables (B$i, i = 1..7$) de réflectance et de température entre les deux dates d'acquisition :

Comparaison inter-subset :

— Nous remarquons que les moyennes et écart-types des valeurs de réflectance et de température des subsets acquis la date du 12/07/2017 sont plus élevées que celles du 05/07/2017. Nous remarquons ainsi que les moyennes des écart-types des variables réflectances et température des subsets de la date du 05/07/2017 sont plus élevés que ceux de 12/07/2017, donc une forte variabilité des moyennes pour les subsets du 05/07/2017. Dans les boites à moustaches ci-dessus, nous observons que les moyennes des variables réflectance et de température sont asymétrique alors ces valeurs moyennes des subsets ne suivent pas une loi normale. Cela peut être expliqué par le fait que, au 12/07/2017 on a appliqué un stress sur une partie des arbres qu'on a sélectionné au hazard. On a donc de la variabilité due aux nombreuses variétés, plus de la variabilité due à la réaction au stress hydrique. De plus les conditions météo étaient très différentes : 22.4° et 74% HR (Humidité relative de l'air) le 05/07/2017, alors que le 12/07/2017 était une journée très chaude avec un air très sec (31.6° et 39% HR), ce qui a pu contribuer à une dispersion des valeurs (comportements plus extrêmes dans ces conditions météo).

— Pour comparer les dispersions des moyennes des reflectances et des températures entre les subsets de deux dates, nous avons calculé les coefficients de variation [7] relatif (voir, Tableau 2.2) des variables de réflectance et de température. Ces coefficients varient entre 0.3 et 0.5 pratiquement pour toutes les variables de réflectance et de températures des deux dates. Ils varient de façon similaire pour tous les subsets de différentes dates.

Comparaison intra-subset :

— Nous remarquons qu'il existe des valeurs moyennes de réflectance extrêmes (exemple des moyennes de variables B1, B2, B3, B4 de réflectances de la date du 05/07/2017), cela peut être expliqué par le fait que le couvert végétal est mal réparti conséquence de l'irrigation contrastée de la parcelle. Ce qui explique ainsi, la variabilité élevée des moyennes de température des subsets, les régions arrosées normalement ont plus de couvert végétal, elles sont plus humides et herbées donc elles sont moins chaudes, contrairement aux zones sous stress hydrique (où la température du sol nu parfois atteint les 55° Celsius).

— Nous remarquons une variation concomitante entre les valeurs moyennes de (B1, B2, B3) et de (B5 avec B6). Les subsets de la date 05/07/2017, les moyennes obtenues des valeurs de reflectance des subsets varient de la même façon d'un subset à un autre (voir la section, Tableau des corrélation 2.2.1).

Les graphiques des corrélations 2.7 montrent que les valeurs des variables (B$i, i = 1..6$) ne sont pas symétrique donc ne se répartissent pas selon une loi normale. Nous remarquons aussi, une forte corrélation de 90% à 94% entre les variables (B$i, i = 1..4$) des bandes du visible surtout pour les subsets du 12/07/2017 et moins pour le 05/07/2017, cela peut être expliqué par la température élevée le jour du 12/07/2017.

Pour les températures, nous avons observé que les valeurs de pixels des feuilles se répartissent de façon plus homogène pour les deux dates, contrairement à celles des pixels correspondant à la classe "autres", cela peut être du à la répartition du couvert végétal au niveau du sol, et à l'existence d'autres éléments au niveau de la parcelle (tuyaux, poteaux, sol, enherbement).

7. **Coefficient de variation** : une mesure relative de la dispersion des données autour de la moyenne, il est calculé par le ration de l'écart type sur la moyenne.

FIGURE 2.6 – Tableau de variation des variables (B$i, i = 1..7$) de réflecrance et de température des subsets de la date du 05/07/2017.

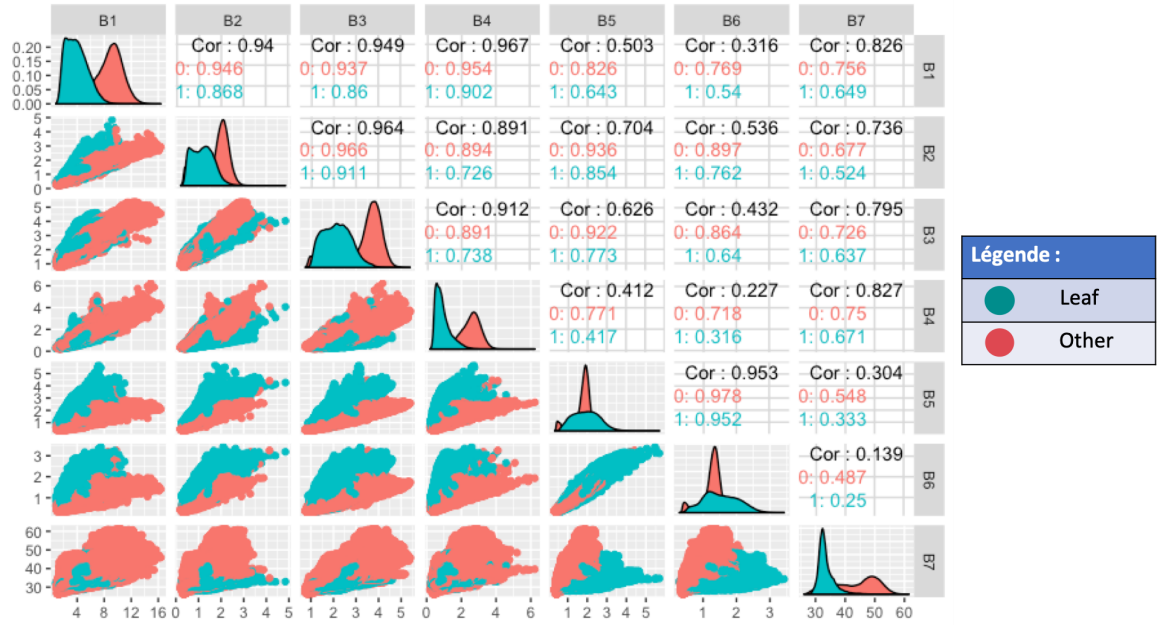
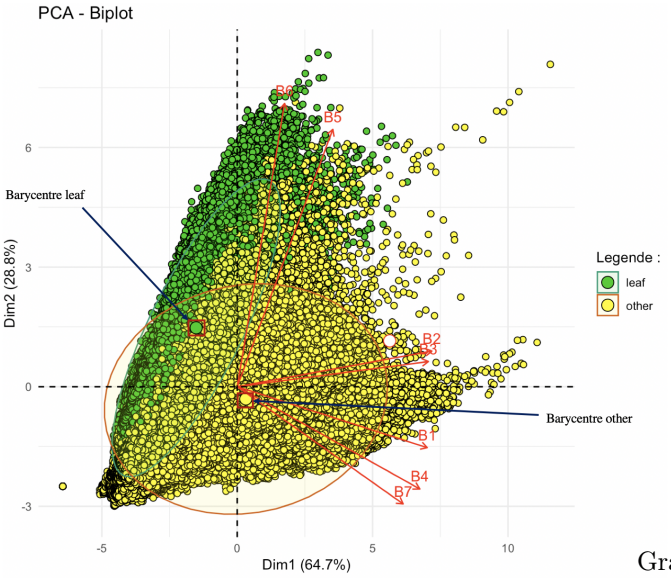


FIGURE 2.7 – Tableau de variation des variables (B$i, i = 1..7$)de réflecrance et de température des subsets de la date du 12/07/2017.

2.2.1 Analyse des Composantes Principales

À la suite de ces analyses descriptives et de corrélations, nous avons effectué une **Analyse en Composantes Principales** sur une l'ensemble des subsets, de deux dates d'acquisition séparément, afin mesurer les contributions des variables B$i, i = 1..7$ à notre jeu de données et les classer par la suite selon la qualité contributions. Le tableau ci-dessus montre les résultats obtenus de l'**ACP** :



Graphe des individus ACP (a)

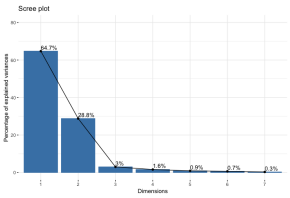


FIGURE 2.8 – Graphe des valeurs propres.

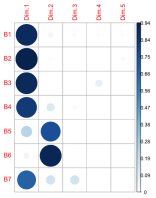
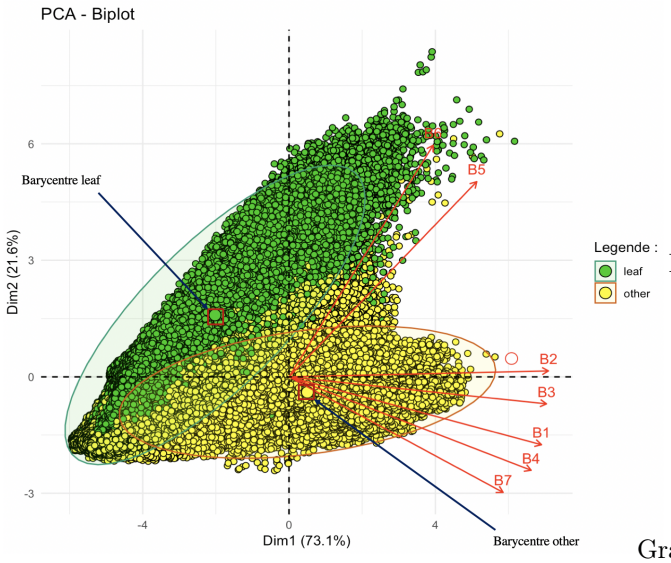


FIGURE 2.9 – Graphe des contributions.
Graphes des valeurs propres et contributions (b)

FIGURE 2.10 – Résultas de ACP sur les valeurs de réflectance et de température des subsets du 05/07/2017.



Graphe des individus (a)

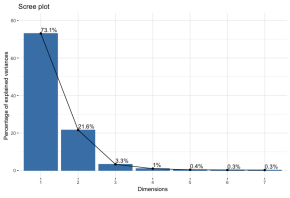


FIGURE 2.11 – Graphe des valeurs propres.

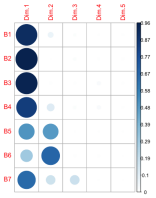


FIGURE 2.12 – Graphe des contributions.
Graphes des valeurs propres et contributions (b)

FIGURE 2.13 – Résultats de ACP sur les valeurs de réflectance et de température des subsets du 12/07/2017.

Ces **ACP**s ont été effectuées sur l'ensemble des variables ($Bi, i = 1..7$) de réflectance et de température. Les individus sont les pixels, sur les subsets de la première date du 05/07/2017 nous avons 1043877 et 740900 pixels de la date du 12/07/2017.

Pour la date du 05/07/2017, Il apparait que 94.7% de variance expliquée par les deux premières dimensions, et une qualité des contributions importantes de toutes les variables de deux classes "feuilles" et "autres" sur ces deux premières dimensions. Les variables B1, B2, B3, B4, B7 sont corrélées positivement, elles contribuent plus à la première dimension. cependant, les variables B6 et B5 contribuent plus à la deuxième dimension.

Pour la date du 12/07/2017, la variance expliquée par les deux premières dimensions est de 95.7%. Nous remarquons des contributions importantes de toutes les variables et de deux classes "feuilles" et "autres" sur les deux premières dimensions. Les variables B1, B2, B3, B4, B7 sont corrélées, elles contribuent le plus à la première dimension, ainsi que B5 et B6 sont fortement corrélées et elles contribuent à la deuxième dimension.

Conclusion :

Dans ce chapitre, nous avons présenté les données multi-spectrales et thermiques, le mode d'acquisition, le pré-traitement de données et pour conclure, nous avons effectué des analyses descriptives sur les données annotées.

Les analyses descriptives, de variances, des corrélations et d'ACP sur les variables des réflectance et de température nous révèlent que les corrélations en groupes des variables du spectre solaire visible (VIS), proche infrarouge (NIR), l'infrarouge thermique (TIR). Ces variables ont des dispersions homogènes, elles gardent des structures de corrélations assez semblables pour les deux dates.

Les représentations des individus (pixels des subsets des deux dates) dans l'ACP , montrent qu'il est possible de discriminer ce qui est "sol" et "arbre" de manière efficace par une approche multivariée. Les autres résultats de l'ACP montrent que toutes les bandes contribuent de la même manière, mais il est plus intéressant d'utiliser toutes les bandes pour les méthode du deep learning.

Chapitre 3

Les réseaux de neurones, Deep learning

Introduction :

Dans le cadre de notre étude, nous nous sommes intéressés aux méthodes de deep learning (Apprentissage profond). Ces méthodes sont une des principales branches du Machine learning [1](Apprentissage automatique) dans le domaine de l'intelligence artificielle. Celles-ci se sont montrées très utiles notamment pour le traitement des gros volumes de données (Big data). Ces méthodes permettent de réaliser des tâche de classification et par extension permettent de résoudre des problèmes, tels que l a segmentation, la détection et localisation d'objets dans une image.

Dans ce chapitre, nous allons donner un bref historique sur le deep learning, puis quelques applications et contributions de ces méthodes dans le phénotypage et l'agriculture. À la fin, nous allons présenter les réseaux de neurones et le type de réseaux qu'on va utiliser dans les prochains chapitres.

Histoire du deep learning :

Le deep learning est un ensemble de méthodes d'apprentissage automatique. Certaines méthodes de deep learning ont montré des résultats très impressionnants. En 2017, l'algorithme de deep learning Alpha Go construit par google a battu le champion du monde dans le jeu de GO chinois. Ces méthodes sont basées sur des structures de données particulières appelées "Réseaux de Neurones Profonds (DNN)". Ils existent différents type de Réseaux de Neurones Profonds : les réseaux de neurones à convolution (CNN), les réseaux de neurones récurrents (RNN) [2], etc.

Dans notre étude, nous nous focalisons sur les réseaux de neurones à convolution pour les tâches de traitements d'image. Ces réseaux de neurones ont été inventés dans les années 80, puis délaissées car ils demandent du matériel puissant et une quantité de données importante pour les paramétriser.

En 1999, LeCun [30]proposent un algorithme de rétro-propagation du gradient plus performant qui permet une paramétrisation plus simples de ces réseaux. Cela permet aux méthodes de deep learning de connaitre une seconde vie. Cette réapparition en force s'impose notamment dans le domaine du traitement et de l'analyse d'images. En 2012 lors d'un challenge **ILSVRC** [3], les méthodes de **deep learning** se sont montrées les plus efficaces et les plus fiables pour le traitement d'images. Différentes architectures de deep learning sont apparues comme des leaders dans ces domaines : (**AlexNet(2012)**, **VGG-16(2013)**, **GoogLeNet (2014)** et **ResNet(2016)**)...

1. **Apprentissage automatique :** est une approche statistique constituée de deux étapes fondamentales, la première de l'entrainement pour la construction de modèle et la seconde étape consiste à prédiction. Source : `https://fr.wikipedia.org/wiki/Apprentissage_automatique`
2. **Les réseaux de neurones récurrents** : ces réseaux s'appliquent sur des données séquentielles (temporelle). Ils sont beaucoup utilisés dans le domaine de la reconnaissance de l'écriture manuscrite ainsi que la reconnaissance vocale par des moteurs de recherche connus tel que : GOOGLE.
3. **ILSVRC** (IMAGENET Scale Visual Recognition Challenge) : est une compétition internationale annuelle, organisée par plusieurs chercheurs pour la comparaison et l'évolution des algorithmes de détection et de classification des objets dans une image. Url : `http://image-net.org/challenges/LSVRC/`.

Elles ont apporté des contributions largement reconnues pour les tâches de classification, de détection et de segmentation d'objet dans une image [17] [18][20]. Elles ont ainsi permis des progressions dans plusieurs domaines scientifiques (médecine[32], agriculture, etc.).

Une bonne partie des méthodes de deep learning s'appuient sur des méthodes d'apprentissage automatique supervisé [4]. Ces méthodes fonctionnent en deux étapes : une première étape de conception du modèle ou a lieu une phase d'apprentissage sur les données et une seconde étape de prédiction.

Quelques applications du deep learning en agronomie et phénotypage :

Les méthodes de deep learning ont beaucoup d'interet agronomie. F. Coppens et al. 2017 [10] ont montré que de telles approches permettent d'éviter différentes difficultés (Pas d'hypothèses préalables, etc.). On trouve également un certains nombres d'applications du deep learning pour le phénotypages [23] dans la littérature. Singh et al. 2018 [11] ont utilisé ces approches pour le phénotypage des plantes sous stress à haut débit en montrant que différentes tâches peuvent être réalisés par deep learning : identification et classification des cultures, la quantification des récoltes ou des fruits, et prédiction des récoltes. S. Ghosal et al. 2017 [24] ont étudié la précision et la robustesse des méthodes d'imagerie par deep learning pour l'identification et la classification des récoltes dans le milieu végétal. Z. Khan et al. 2018 [25] ont également utilisé des méthodes d'imagerie par deep learning pour estimer des indices de végétation pour le phénotypage du blé. Pour la détection et le comptage d'objets dans un milieu végétal (détection des fruits dans un arbre), A. Dore et al. 2018 [28] ont montré la répétabilité et la précision que ces méthodes peuvent apporter pour le phénotypage. Dans le cadre de la conception d'une plateforme de phénotypage appelée "Deep Plant Phenomics" et basée sur les méthodes de deep learning [26], J. Ubbens al. 2018[27] proposent une méthode de deep learning pour le comptage des feuilles de rosettes d'Arabidopsis Thaliana .

3.1 Les réseaux de neurones

Le réseau de neurones artificiels [14] sont des structures composées d'un ensemble de neurones artificiels relié entre eux. Un type de réseau particulier est particulièrement utilisé, le perceptron multicouche ou les neurones artificiel sont organisés en plusieurs couches. Chacune de ces couches est composé d'un nombre variable de neurones artificiels. Les neurones d'une couches sont connectés partiellement ou totalement aux neurones des couches précédentes. Pour bien comprendre les réseaux de neurones, nous allons définir ce qu'est un neurone artificiel.

3.1.1 Neurone artificiel et le neurone biologique

Un neurone artificiel peut être défini comme une fonction mathématique qui imite le comportement de neurone biologique. Un neurone biologique est une cellule constituée de noyau, des synapses, des dendrites et axones(voir, Figure 3.1). Elle fonctionne comme suit : elle reçoit des signaux électriques à partir des cellules auxquelles elle est connectée, et en fonction de ces signaux elle peut renvoyer un signal par ses terminaisons et ses axones. Le neurone artificiel fonctionne de la même façon, il reçoit des signaux d'entrée comme une combinaison de ces entrées $X_i, i = 1 : n$ et des poids qui le définissent $W_{ij}, i = 1 : n, j = 1 : m$. Une somme pondérée des entrées et des poids peut être réalisée $X_i W_{ij}$ et est appelée fonction d'agrégation. En fonction de ce signal agrégé, le neurone renvoie le produit de cette somme par une fonction de transfert appelée **" fonction d'activation "** [14]. Les neurones se distinguent par leurs fonction d'activation [15]. Les fonctions d'activations les plus connues sont : **fonction linéaire, seuil :** $g(x) = 1_{[0,+\infty[}(x)$**, sigmoïde :** $g(x) = \frac{1}{(1+\exp(x))}$ et de **correction Relu :** $g(x) = max(0, x)$, etc. Notons que la sortie σ est souvent calculée par : $\sigma =$ 1 si $f(\sum_{i=1}^{n}(X_i W_{ij})) > seuil$, 0 sinon .

4. **Apprentissage supervisé :** consiste à apprendre un modèle de prédiction à partir des données annotées. Source : `https://www.supinfo.com/articles/single/6041-machine-learning-introduction`

La figure ci-dessous illustre le schéma du fonctionnement d'un neurone artificiel :

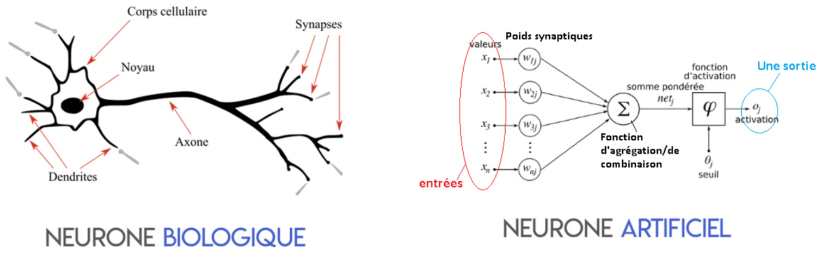


FIGURE 3.1 – Schéma du fonctionnement de neurone artificiel et biologique.
Source : `https://deeplylearning.fr/cours-theoriques-deep-learning/fonctionnement-du-neurone-artificiel/`.

3.1.2 Fonctionnement de l'apprentissage profond

La conception des modèles du deep learning s'effectue généralement en trois phases : la phase d'apprentissage, la phase du test et de validation et la phase de prédiction. Pour réaliser ces trois phases, nous divisons le jeu de données en trois parties : données d'apprentissage ($\simeq$ 80% de jeu de données), données du test et données de validation (20% du jeu de données) puis on généralise la prédiction sur d'autre donnée de même type que celui des vérité terrain.

Phase d'apprentissage

Cette phase consiste en la calibration des paramètres du réseau. Durant cette étape d'apprentissage, on fournit au réseau de neurones les données d'entrée puis en comparant les sorties avec les vérités-terrain, on réajuste les paramètres du réseaux (poids, etc.) pour s'approcher au mieux des valeurs des vérités terrain. Il existe plusieurs méthode pour la correction et l'ajustement des poids des réseaux. La méthode la plus utilisée est la **Rétro-propagation du gradient** [29], elle est basée sur le calcul des gradient pour la minimisation des erreurs des poids des réseaux. Cette méthode fonctionne dans deux sens : "propagation (propagation en anglais)" et "retro-propagation (Back-propagation en anglais)".

Dans l'étape de propagation, le réseaux prend les données d'entrée, puis il réajuste les poids des neurones de chaque couche de réseaux jusqu'à l'obtention d'une sortie. Dans la deuxième étape de de retro-propagation (Back-propagation), le réseau cette fois-ci prend les résultats obtenus dans l'étape précédente et il les compare avec les données de vérité de terrain puis il modifie les poids de la dernière couche de réseaux qui précède pour que l'erreur soit minimisée. Puis le réseau propage en arrière ces erreur sur les couches précédentes jusqu'à la mise à jour de tous les poids du réseau. Ces étape se répètent jusqu'à l'obtention d'une erreur globale qui soit acceptable par l'utilisateur.

Phase du test et de validation

Pour évaluer les performances du modèle entrainé, on génère des prédictions sur des données du test et de validation (non connues par le modèle). Les prédictions faites sur les données d'apprentissage sont souvent bonnes. Pour valider le modèle, il faut donc le tester sur d'autre données. Notons qu'il existe deux phénomènes qui peuvent se produire lors de l'entrainement : le "sur-apprentissage (Overfitting)" et "sous-apprentissage (Underfitting)" du modèle. On dit qu'un modèle "sur-apprends" si il apprend bien sur les données d'apprentissage mais il ne peut pas être généralisé sur les données test et de validation, c-à-d il produit de mauvaise prédictions sur des données non vues par le modèle. Un modèle est dit "sous-apprends" si il apprend mal sur les données d'apprentissage, ainsi qu'il génère des prédictions non constantes sur les données de test et validation (voir, figure 3.3).

1. Information : pour plus d'information sur l'apprentissage et test des modèles du deep learning (Machine learning en général), voici un lien : `https://docs.microsoft.com/fr-fr/azure/machine-learning/studio/algorithm-parameters-optimize`.

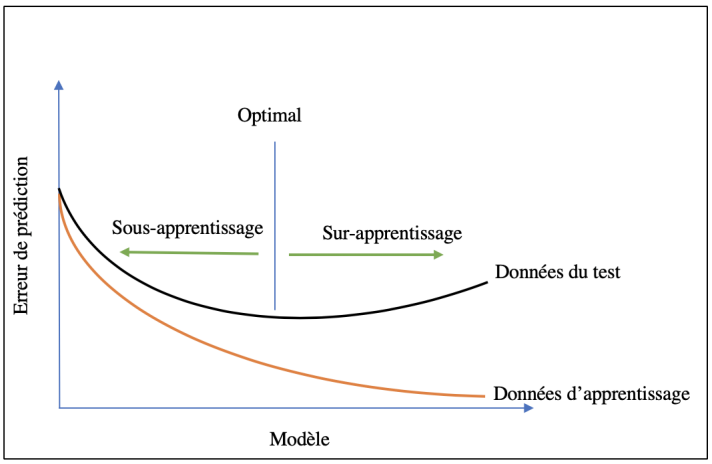


FIGURE 3.2 – Courbe des erreurs d'apprentissage et test.

Les phases d'apprentissage et test sont souvent appliquées de façon simultanée sur les données d'apprentissage et du test et pour un nombre d'itérations définies (appelé epoch) que nous allons appeler "nombre d'époques". Pour faciliter et accélérer l'apprentissage du modèle, on fixe généralement un paramètre de réseau appelé "Bach-size", il est la quantité moyenne de données lue aléatoirement par le modèle lors de l'apprentissage par époque. Ce paramètre est dépendant des capacités de mémoire de la machine utilisée.

En résumé, les poids du réseau de neurones sont réajustés à chaque époque en minimisant l'erreur commise par chaque neurone. Ces erreurs sont estimées en fonction des comparaisons des sorties du modèles à des données de vérité de terrain.

Pour chaque époque d'apprentissage, on définit une erreur globale appelée "valeur de perte", le terme le plus utilisée est "Loss en anglais". Cette valeur correspond à la valeur à optimiser (minimiser) pour l'apprentissage. Elle peut par exemple être calculée comme la somme des carrés de différences entre les valeurs de vérité de terrain et les valeurs prédites sur le nombre de données. Donc, pour chaque époque on obtient deux valeur Loss : une des données d'apprentissage et une deuxième des données du test.

Avec les sortie du modèle durant l'apprentissage, on obtient ainsi une autre valeur appelée "précision (accuracy en anglais)". Cette valeurs est calculée comme suit : lors de l'apprentissage, les valeurs prédites de sorties seront par la suite discrétisées et classées par un seuil de discrimination. Ensuite, les valeurs discrète obtenue seront de nouveau comparées aux valeurs de vérité de terrain. Elles sont calculées comme le ratio du nombre de données correctement prédites sur le total de données. Notons que, durant chaque époque d'apprentissage on obtient deux valeurs d'accuracy : une des données d'apprentissage et une deuxième des données du test.

Ces valeurs de Loss et d'accuracy pour les données d'apprentissage et test sont d'importance. Elles permettent de suivre l'évolution d'apprentissage et de performance des modèle lors de l'entrainement.

2. Information : Il existe d'autres valeurs de précision de modèle de deep learning comme : F-score, Precision, Reccal, etc. Vous pouvez trouvez dans le lien suivant des explication sur le calcul des valeurs de précision. Url :`https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234`

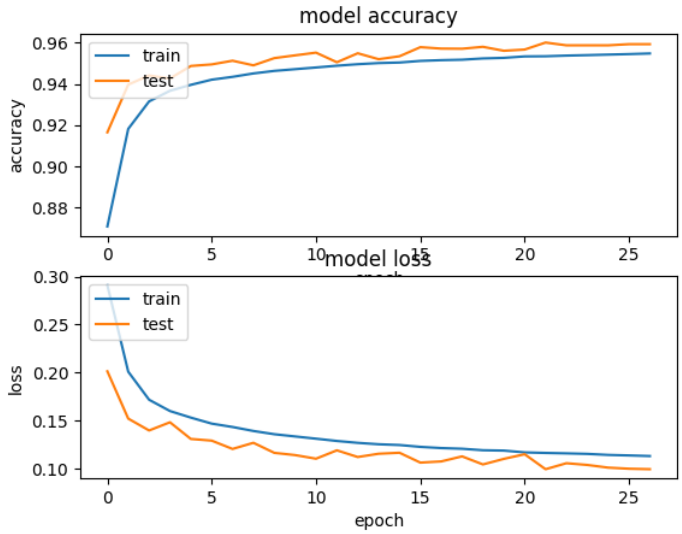


FIGURE 3.3 – Courbe des valeurs loss et accuracy.

Phase de prédiction

Cette étape consiste en la génération de prédictions par des modèles entrainés jugés pertinents dans les étapes d'apprentissage et test.

3.2 Réseau de neurones à convolution

Les réseaux de neurones à convolution [5] (sous forme abrégée CNN ou ConvNet) est l'approche la plus utilisée par les méthodes de deep learning pour le traitement d'images. Ces réseaux se sont révélées intéressants pour différentes tâches dans le domaine d'imagerie telles que la classification, la détection, et la localisation d'un objet dans une image,etc... [17]. Les CNNs sont apparus dans les années 1980 après l'invention des modèles de "Neocognitron" proposés par Fukushima [22]. Un CNN [6] est généralement constitué de quatre types de couches [21], une couche de convolution, une couche pooling, une couche Relu, une couche complètement connectée.

3.2.1 Couche de convolution

C'est la couche essentielle dans les réseaux de neurones à convolution. Elle est utilisée pour l'extraction des caractéristiques d'une image d'entrée. Elle construit des filtres qui sont appliqué sur l'image d'entrée, et extrait en sortie ces informations (les traits, luminosité, les intensités des couleurs, les contours, etc...) sous forme de cartes appelées "cartes de caractéristiques" (feature map).

5. **Convolution :** est une opération appliquée au cœur de la couche de convolution, son origine revient au "produit de convolution" des fonctions mathématiques. La convolution est le fait d'appliquer les filtres sur l'image d'entrée pour l'extraction des caractéristiques.

6. Note : ces information sont trouvées dans les sites web suivants : 1- Deep Learning, le réseau à convolution, Florent SIMON , 05/10/2018,`http://penseeartificielle.fr/focus-reseau-neurones-convolutifs/#1_Lrsquooperation_de_convolution`. 2- What is max pooling in convolutional neural networks ? Jay Ricco, Fourth Year, Undergraduate BS Computer Science, `Whatismaxpoolinginconvolutionalneuralnetworks?JayRicco,FourthYear,UndergraduateBSComputerScience`. 3-How do Convolutional Neural Networks work ? End-to-End Machine Learning, Brandon Rohrer,`https://brohrer.github.io/how_convolutional_neural_networks_work.html`.

Filtre

Le filtre est une matrice de valeurs souvent de petites dimensions, ses valeurs sont générées et améliorées lors de l'apprentissage du modèle. Le filtre prend le parcours montré dans la figure ci-dessus :

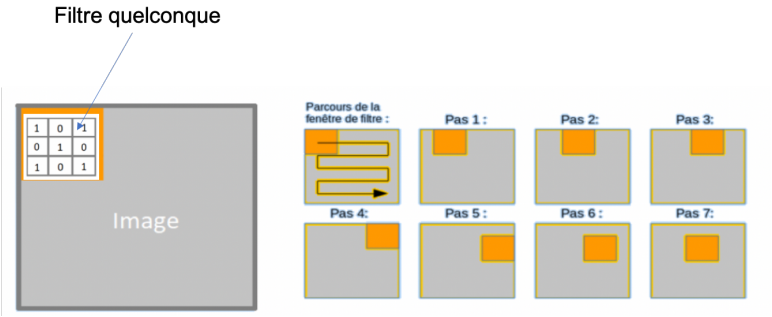


FIGURE 3.4 – Parcours du filtre sur l'image.
Source : `https://www.supinfo.com/articles/single/8037-deep-learning-reseau-convolution`.

Le calcul de la convolution se fait comme illustré dans la figure ci-dessus : Certain paramètres de

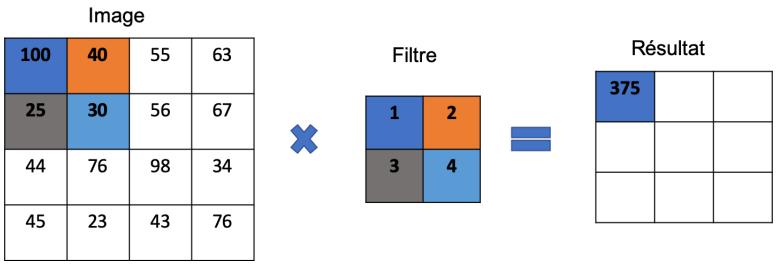


FIGURE 3.5 – Exemple du calcul de filtre.

On fait le produit cartésien entre les éléments des deux matrices pixels par pixels, pour le calcul de notre exemple, nous avons : $100\times1 + 40\times2 + 25\times3 + 30\times4 = 375$.

filtres sont fixés. les paramètres essentiels sont : Le nombre de filtres [7], la taille du filtre [8], le pas de déplacement [9], et le contour [10]. Ces paramètres dépendent directement de la taille de l'image. Exemple (Voir, Figure 3.5) : le nombre de filtre = 1, kernel = 3×3, stride = 1, padding = 0.

Une opération inverse existe : la convolution transposée. Cette convolution prend une carte de caractéristiques en entrée et, en utilisant le même noyau de convolution, régénère une information détaillé. Les mêmes paramètres que la convolution classique sont utilisés.

3.2.2 Couche pooling

La couche pooling est cruciale dans les réseaux de neurones à convolution. Elle reçoit des cartes des caractéristiques (feature map) et les réduit en cartes de plus petites dimensions en conservant des informations simplifiée. Cela permet de réduire le temps de calcul et d'éviter tout sur-apprentissage [11] du modèle. En effet, la réduction de tailles de cartes de caractéristiques permet ainsi une réduction en nombre de paramètres du modèles et conditionne mieux l'apprentissage.

7. **Nombre de filtres** :le nombre de filtre est déterminé fonction des images traitées, de leurs aspects, de leurs dimension. il est fixé par l'utilisateur, puis le réseaux de neurones ne retient que les filtre qui détecte plus d'informations.
8. **Taille du filtre** : est la dimension longueur×largeur du filtre, elle souvent appelée **kernel**(noyau en français).
9. **Le pas de déplacement** : est les nombres de pixel que le filtre parcours entre deux convolution, il est appelé ainsi **stride**.
10. **Contour** : pour avoir des cartes de caractéristique de même dimension que l'image d'entrée, on rajoute des zero autour, ce paramètre est pour fixer le nombre de ligne de zéros qu'on rajoute. Il est appelé aussi **zero_padding**
11. **Sur-apprentissage :** est le fait que le modèle apprend bien sur les données d'apprentissage, et qu'il se généralise mal sur les données de validation.

Il existe différentes types de couche de pooling :

— **Max-pooling** : garde la valeur maximale de la fenêtre de convolution (voir, Figure 3.6). C'est l'opération généralement utilisée dans les réseaux de neurones à convolution.

— **Sum-pooling** : prend la moyenne des valeurs de la fenêtre de convolution.

— **Stochastique-pooling** : sélectionne de façon stochastique une des valeurs de la fenêtre de convolution.

La figure suivante illustre le calcul de la couche Max-pooling :

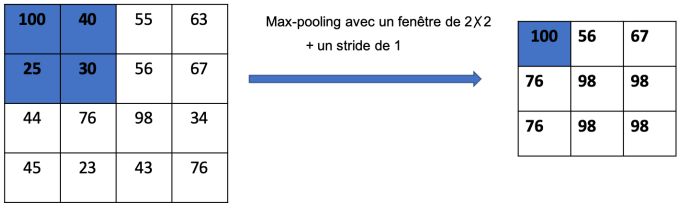


FIGURE 3.6 – Exemple du calcul de max-pooling.

Ce calcul s'effectue de manière similaire à celui d'une convolution. L'image d'entrée est parcourus avec une fenêtre et le maximum des pixels de la fenetre est calculé à chaque étape. Les paramètres les plus importants pour la couche de max-pooling sont : kernel (taille de la fenêtre), stride (pas de deplacement). Notons que la couche de pooling produit le même nombre de cartes de caractéristiques qu'elle prend en entrée.

3.2.3 Couche Relu (Rectified Linear Unit)

La couche de correction linéaire **Relu** est une fonction d'activation qui retourne zéro pour toute valeur négative en entrée ou la valeur de l'entrée si elle est positive. La figure ci-dessus montre un exemple de calcul de la couche Relu :

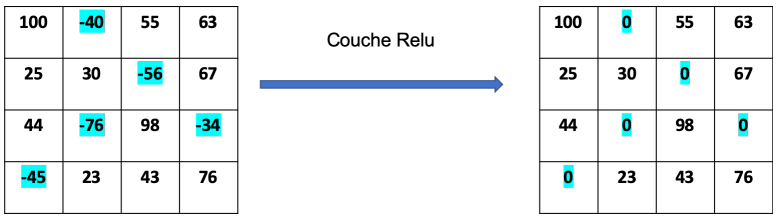


FIGURE 3.7 – Exemple du calcul de la couche Relu.

Il existe ainsi différent type de fonction d'activation . On citera notamment la fonction **Elu** [31]. Introduit par D. A. Clevert et al. 2017 [31] elle utilise une fonction exponentielle non-linéaire. Ce type de transformations permet d'étendre l'intervalle des valeurs de données (avec des valeurs négatives) en cassant la linéarité qu'il peut y avoir après les convolutions.

Ces transformations sont calculées comme suit :

$$f(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha(\exp(x) - 1) & \text{sinon.} \end{cases}$$

Les valeurs x sont les valeurs de données d'entrée et α est une constante. Pour la couche **Elu** la constante α=1.

3.2.4 Couche complètement connectée

Ces couches sont généralement positionnées comme dernière couche des réseaux de neurones à convolution. Elles prennent en entrée un vecteurs de cartes de caractéristiques, et en fonction de ces cartes elle classifie les valeurs des pixels.

Conclusion :

Dans ce chapitre, nous avons donné quelques notions sur les réseaux de neurones, les réseaux de neurones à convolution, de l'historique et les domaines d'application ainsi des exemples d'application dans le domaine de l'agronomie et le phénotypage. Puis nous avons présenté leur structure et fonctionnement de chaque partie de la structure. À la fin, nous avons montré le processus de conception et d'utilisation des modèle de deep learning.

Chapitre 4

Segmentation d'images multi-spectrales et thermiques aéroportées par la méthode U-NET pour le calcul d'indices de végétation de pommiers

Introduction :

L'objectif du travail présentées dans ce manuscrit consiste en l'analyse d'images multi-spectrales et thermiques d'une parcelle de pommiers acquises par vol de drône pour l'estimation d'indices de végétation pour chaque arbre. Un point critique d'une telle analyse est la segmentation et la localisation des feuillages de chaque arbre de la parcelle.

Afin de segmenter les pixels du feuillage des arbres, nous avons choisi d'adapter la méthode de deep learning **U-NET** [32]. Une première partie de ce chapitre sera consacré à la définition et la présentation de cette méthode, puis son adaptation et application sur nos données. En résultat, nous présentons les segmentations obtenues par **U-NET**, puis une comparaison avec la méthode d'apprentissage automatique classique **SVM** [38] [39]. Ensuite une méthode basée sur la méthode du WaterShed et sur des relevés GSP des centres des arbres permettant d'identifier les pixels de chaque arbre est présentée. Finalement, nous présentons le processus d'extraction des valeurs des feuilles et du calcul d'indices de végétation (voir, Section 2.1.6) des arbres de la parcelle que nous avons mise en place a partir des segmentations d'arbres individuel.

Un objectif important de ce chapitre est d'évaluer l'apport de la méthode **U-NET** pour la segmentation d'images muti-spectrales et thermiques d'un milieu végétal. Le but dans ce cas est alors de séparer les pixels du feuillages des arbres de ceux du reste de la parcelle, c-à-d de décomposer les images multi-spectrales et thermiques en deux classes d'objets : classe **"feuilles"** correspondant aux feuillages des arbres (la canopée) et la classe **"autres"** qui correspond au reste (sols et à l'herbe...) de la parcelle comme le montre la figure ci-dessous :

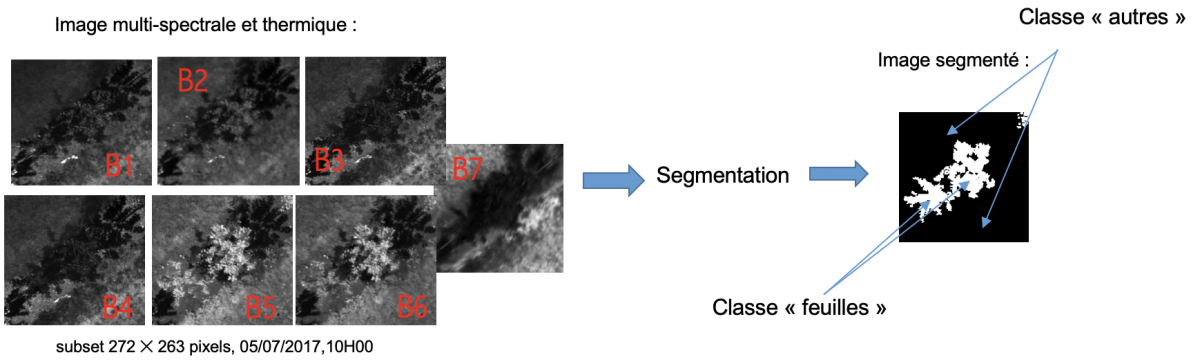


FIGURE 4.1 – Exemple de segmentation d'un subset de la parcelle.

La segmentation des images multi-spectrales et thermiques n'est pas une tâche facile. Le choix de la méthode **U-NET** a été fait par rapport à ces avantages (voir, Section 4.1) et sa capacité d'application sur la quantité de données annotées dont on dispose.

4.1 La Méthode U-NET

La méthode **U-NET** a été une véritable avancée dans le domaine de deep learning pour la segmentation d'image. Elle a été introduite originellement pour segmenter des images biomédicales par **Olaf Ronneberger** et al. 2015 [32]. Elle est connue pour être rapide en temps de calcul et précise en prédiction. La méthode **U-NET** a une structure particulière de type "Encodeur-Décodeur". La structure du réseau sous-jacent est formée d'une succession de blocs de couches de type convolution, pooling et Relu. Les couches de chaque bloc sont complètement connectées entre elles.

La méthode **U-NET** peut être appliquée sur un échantillon de données (images) relativement petit contrairement à d'autres méthodes de deep learning. C'est intéressant dans notre cas, car nous n'avons pu annoter que 45 subsets. La méthode **U-NET** est rapide en temps de calcul notamment sur des machines équipées d'un GPU [1]. **U-NET** a une bonne précision de la segmentation, cela est du à sa structure "Encodeur-Décodeur" que nous allons présenter dans la section 4.1.1 ci-dessous et à son utilisation des réseaux de neurones complètement connectés.

1. **GPU :** est un micro-processeur graphique(Graphics Processing Unit en anglais) récent. Il a été récemment inventé pour le traitement graphique de données. Le GPU a largement contribué au développement et à la réapparition des méthodes de deep learning par ses capacités de calculs parallèles souvent importantes.

4.1.1 Structure de la méthode

la méthode **U-NET** [2] est constituée de 3 étapes (voir, Schéma 4.2) :

— Étape 01 : Elle est appelée "Down-convolution" (Convolution). Cette étape est constituée de 4 blocs de couches dont chacun est composé de :
  — 2 × Couche de convolution 3 × 3, avec une fonction de correction « Relu ».
  — Couche Max_pooling de 2 × 2.

  Le nombre de filtres de convolution commence à 64, puis il double à chaque couche de convolution jusqu'à 512 filtres. Il dépend de la taille de l'image d'entrée et du nombre de couches de convolution. Cette étape sert à l'extraction des cartes des caractéristiques. Chaque bloc prend l'image d'entrée et lui applique deux couches de convolution pour extraire des cartes de caractéristiques. Ensuite, il applique une couche "Relu" pour rectifier et éliminer les valeurs négatives et les remplacer par des zéros dans les cartes des caractéristiques. À la fin, il utilise une couche Max-pooling pour ne conserver que les caractéristiques les plus importantes en réduisant les tailles des ces cartes de caractéristiques.

— Étape 2 : Elle est appelée "bottleneck". Elle est composée d'un seul bloc de 3 couches de convolution 3 × 3 et 3 couches de correction "Relu" :

  — Ce bloc de couches permet le lien entre la première partie de "Down-Convolution" et la dernière de "Up-convolution". il constitue une couche de convolution 3 × 3 et une couche de correction "Relu" ainsi que, une couche de convolution inverse pour agrandir les tailles des cartes pour la prochaines étape.

— Étape 03 : Elle est appelée Up-Convolution (Convolution transposée). Cette étape est composée de 4 blocs dont chaque bloc est composé de :

  — Couche de convolution transposée 3 × 3.
  — 2 × Couche de concaténation avec les cartes de caractéristiques obtenues dans l'étape de "Down-convolution" de même niveau.
  — Couche de convolution 3 × 3, avec une fonction de correction « Relu ».

  Cette étape consiste en la construction de l'image segmentée (masque). Chaque bloc de couches reçoit des cartes de caractéristiques et agrandit ces carte en tailles par une convolution transposée. Ensuite une concaténation des cartes de caractéristiques est faite avec les cartes obtenues dans l'étape de "Down-convolution" de même niveau. À la fin de chaque bloc, deux convolutions sont appliqués sur ces cartes pour l'extraction de nouvelles cartes de caractéristiques. Ce processus se répète un certain nombre de fois en fonction de la taille des images d'entrée et de sortie souhaitée.

2. **Note :** Ces notations sont citées dans l'article [32] de **U-NET**, ainsi dans le tutoriel du site web :`http://www.deeplearning.net/tutorial/unet.html`. Où les auteurs montrent l'architecture et les avantages de la méthode, ainsi que l'importance de l'utilisation des réseaux complètement connectés pour la segmentation d'images.

La figure suivante montre visuellement la structure de la méthode **U-NET** :

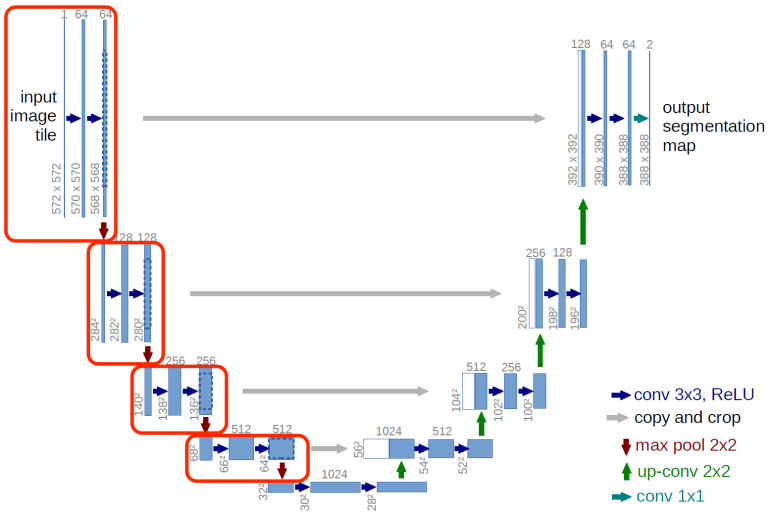


FIGURE 4.2 – Structure du modèle U-NET

Nous remarquons dans la figure ci-dessus, que la taille des images en sorties(388×388 pixels) est différente de celles en entrées (572×572 pixels). Pour conserver les dimensions des images en entrée du modèle **U-NET**, il est nécessaire d'étudier et de fixer les hyper-paramètres les plus pertinents pour les couches de convolution, les couches de convolution transposée et max-pooling (voir, Sections 3.2.1 3.2.2) vus dans le chapitre de deep learning.

4.2 Entrainement de U-NET

L'entrainement du modèle **U-NET** sur nos données a été effectué en quatre étapes :

1. Étape d'étiquetage et d'annotation de données décrite dans la section 2.1.4.

2. Étape de séparation de données : au total nous avons annoté 45 subsets de dimension variable de (260×280 à 580×600 pixels). Afin de pouvoir appliquer **U-NET**, nous avons découpé ces subsets d'apprentissage et de test en images selon deux tailles de 96×96 ou 256×256 pixels (voir, Section 4.2).

3. Étape d'apprentissage (voir, 3.1.2) : Pour tout entrainement de **U-NET**, nous avons pris 80% de données pour l'apprentissage ($\approx$ 38 subsets, voir, Tableau 4),. Le choix des paramètres du modèle **U-NET** a été fait par rapport à la quantité des données d'apprentissage et de test et par rapport à la tailles des images (voir, Section 4.2.3).

4. Étape du test : cette étape consiste à tester et évaluer les modèles entraînés obtenus. Les performances des modèles ont été évaluées par les courbes d'accuracy 3.1.2 résultantes d'apprentissage du modèle (voir, 4.5) ainsi que la mesure de prédiction (voir, Section **??**).

Découpage des subsets en images

Pour l'entrainement et le test de **U-NET**, nous avons découpé de façon identique les subsets multi-spectrales et thermiques annotés (Voir, 2.1.4) en images de tailles de 96×96 pixels (resp. 256×256 pixels). Afin d'augmenter la quantité de données, nous avons réalisé des découpes des subsets. Sur chaque subset, on fait parcourir une fenêtre glissante de 96×96 pixels (resp. 256×256 pixels) couvrant la totalité du subset dans deux sens : en partant de haut à gauche jusqu'en bas à droite et en partant du bas a droite jusqu'en haut à gauche (voir, Figure 4.4)). Ces découpes sont effectuées de la même manière sur tous les subsets. Cette fenêtre se déplace par un pas défini (entre 20 et 45 pixels). Chaque déplacement de la

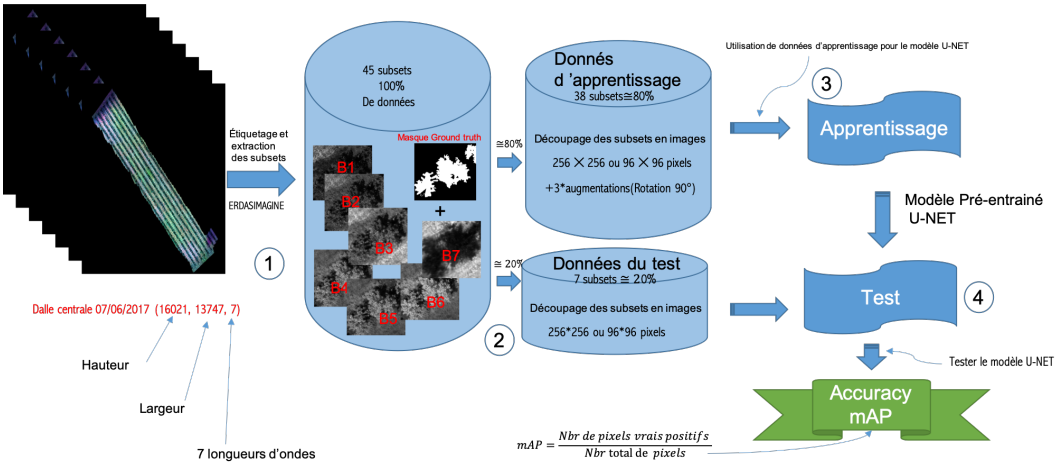


FIGURE 4.3 – Schéma d'application de la méthode U-NET

fenêtre permet une extraction d'une image multi-spectrale et thermique et un masque correspondant de 96×96 pixels (resp. 256×256 pixels).

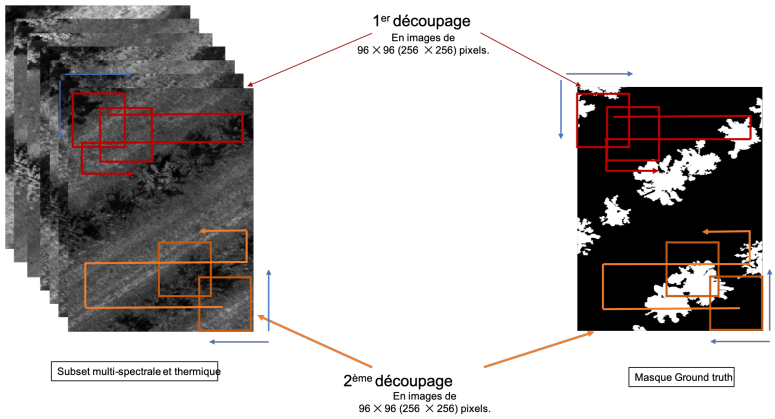


FIGURE 4.4 – Schéma de découpage parallèle des subsets annotés en images et masques de $96 \times 96 (256 \times 256)$ pixels.

Notons que ces découpages en 96×96 pixels (respectivement, 256×256 pixels) nous ont permis de multiplier la quantité de jeu de donneés de 45 subsets annotés en 3624 images(respectivement, 374 images) d'apprentissage et test (voir, Tableau 4).

Augmentation des données d'apprentissage par transformation

Dans le cas des méthodes de deep learning basées sur les réseaux de neurones à convolution, il est toujours intéressant d'augmenter la quantité de données d'apprentissage pour améliorer les performances et précisions du modèle. Dans notre étude, l'augmentation de données a été réalisée par 3 rotations de 90° sur les images et masques d'apprentissage. Cela permet aussi d'éviter tout sur-apprentissage (voir, 3.1.2) du modèle et le rend ainsi insensible à l'orientation (et potentiellement à l'échelle) de la prise de vue des images en entrée.

4.2.1 Mesures d'apprentissage et de prédiction

Nous avons mesuré les performances des modèles entrainés par deux indicateurs : les valeurs d'accuracy trouvées à chaque époque de l'entrainement sur les données d'apprentissage et test (voir, Figure 3.3). Une fois que le modèle est évalué par ces courbes, nous faisons des prédictions sur les données du test, puis sur une partie de la dalle indépendante des données d'apprentissage et test pour valider ces modèles, par le calcul de valeur de précision de prédiction (voir, Figure 4.2.1).

Précision d'apprentissage

Pour évaluer l'apprentissage du modèle, nous avons regardé la tendance et la variation des courbes d'accuracy d'apprentissage et du test. À chaque époque [3] (voir, Section 3.1.2) de l'apprentissage, nous obtenions deux valeurs de précision (accuracy en anglais) : une des données d'apprentissage et une autre des données test. Ces valeurs sont obtenues par le calcul des moyennes de précisions des toutes les images d'apprentissage ou test utilisée par chaque époque.

Dans notre étude la précision ou l'accuracy du modèle est calculée pour chaque image. Elle est obtenue de la façon suivante :

$$Accuracy = \frac{\text{Le nombre de pixels bien prédits}}{\text{Le nombre total des pixels}} \tag{4.1}$$

Pendant l'entrainement, ces valeurs de précision sont calculées sur toutes les images chargées durant chaque époque. Donc pour chaque époque, on obtient la précision moyenne suivante :

Soit $Accuracy_i$ la précision (accuracy) de l'image i, nous avons donc, la précision moyenne des images d'apprentissage ou test utilisées par époque égale à :

$$Accuracy = \frac{\sum_{i=0}^{n} Accuracy_i}{n} \text{ où n : le nombre d'images utilisées par époque.} \tag{4.2}$$

Durant l'entrainement du modèle **U-NET**, nous avons obtenu deux graphiques : d'accuracy et de loss (Voir, Section 3.1.2) des données d'apprentissage et test (voir, Figure 4.5). Ces deux métriques permettent de mesurer les performances du modèle ajusté par période lors de l'entrainement. Elles varient généralement de manière opposée, quand l'accuracy croit la valeur loss diminue. Dans nos simulations, nous n'avons pris en compte que les valeurs d'accuracy.

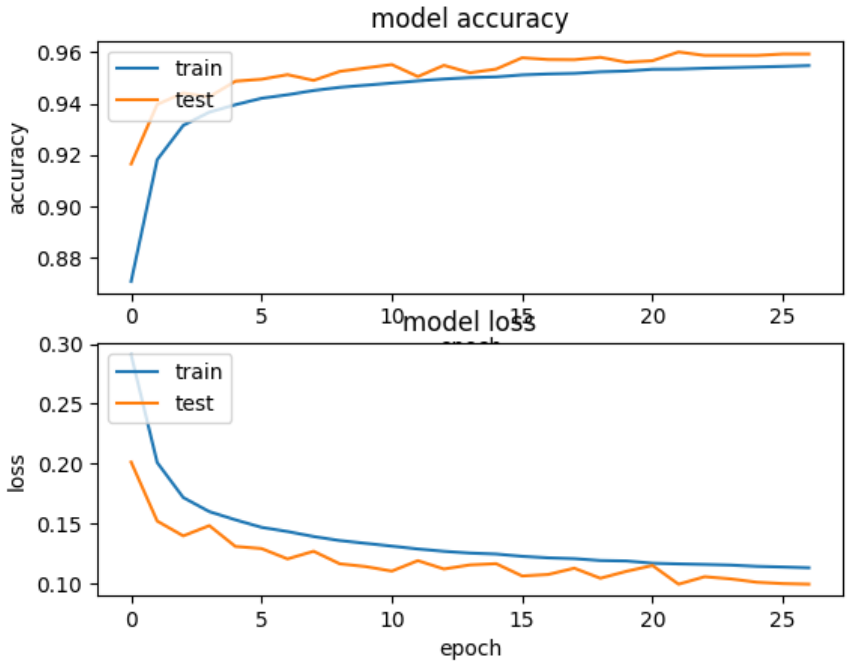


FIGURE 4.5 – Graphe d'Accuracy et de Loss de U-NET.

3. **Époque :** est la période durant laquelle les poids du modèle ont été ajustés pour correspondre aux images en entrée les masques de sortie. Source : `http://fann.sourceforge.net/fann_fr.pdf`.

Précision de prédiction

Pour évaluer la précision et la qualité de la prédiction du modèle entrainé **U-NET**, nous avons calculé la précision des prédiction qui est une estimation empirique des comparaisons entre les valeurs des masques prédits par le modèle entrainé et les masques de vérité terrain, elle est obtenue de la façon suivante :

Étant donné qu'il y a 2 classes de pixels dans notre images ("feuillages" versus "autres"), les valeurs des pixels des images prédites par le modèle lorsqu'ils sont comparées aux vérités terrain peuvent être répartis en 4 classes :

1. Classe des Vrais Positif **VP** : désigne le nombre des pixels des feuilles bien prédits.
2. Classe des Vrais Négatif **VN** : désigne le nombre des pixels de la classe "autres" bien prédits.
3. Classe des Faux Positif **FP** : est le nombre des pixels des feuilles mal prédits.
4. Classe des Faux Négatif **FN** :désigne le nombre des pixels de la classe "autres" mal prédits.

Sur chaque image, nous avons la précision égale au nombre des pixels bien prédits des deux classes "feuilles" et "autres" divisé par le nombre total des pixels, elle est donc :

$$Prcision = \frac{VN + VP}{VN + FN + FP + VP} = \frac{VN + VP}{\text{Le nombre total des pixels}} \tag{4.3}$$

Et la moyenne des précisions de toutes les images du test qu'on va nommer mP (mean Precision) est égale alors :

Étant donné $i \in \{1..n\}$ avec i représentant la ième image et n le nombre d'image utilisé par le modèle par époque.

$$mP = \frac{\sum_{i=0}^{n} Prcision_i}{n} \tag{4.4}$$

Il est important de montrer la différence entre les deux mesures de prédiction d'Accuracy et de mP introduites ci-dessous. Alors que la valeur d'accuracy mesure les performances du modèle durant l'apprentissage sur des images de données d'apprentissage et test de façon simultanée et sur un nombre (Batch-size) petit d'images(voir, Section 3.1.2), la précision mP est généralisée sur tous les subsets du test. Elle est donc plus informative. Elle est utilisée pour la validation des modèles sur d'autres parties de la dalle.

4.2.2 Méthode de prédiction

Il est toujours intéressant de trouver un moyen pour améliorer les prédictions faites par les modèles entrainés de **U-NET**. Alors, pour générer de bonnes prédictions sur des parties de la dalle, nous avons suivi les étapes ci-dessous :

1. Phase de prédiction : elle est effectuée comme suit :

   a- Découpage en image de 96×96 (respectivement, 256×256) : il n'est pas possible de faire de la prédiction sur des subsets de tailles différentes de 96×96 (respectivement, 256×256). Pour cela, nous avons effectué plusieurs découpes. Ces découpes ont été effectuées par des fenêtres de 96×96 (respectivement, 256×256) parcourant le subset par un pas de déplacement donné démarrant d'un point initial donné.

b- Prédiction : elle s'agit de l'étape des prédiction sur image découpées du subset de 96×96 (respectivement, 256×256) par le modèle entrainé de **U-NET**.

c- Construction de masque : La prédiction des subsets a été construite à partir de ces images prédites, en assignant chaque image prédite à la même position dans le masque prédit que l'image de départ dans le subset.

Cette phase est répétée 10 fois. À chaque itération, on change la position initial de la fenêtre glissante. À la fin, on obtient une matrice 3D de 10 prédictions superposée l'une sur l'autre.

2. Phase de sélection des pixels : cette étape consiste à la sélection des pixels les mieux prédit sur les 10 prédictions, c-à-d que nous assignons chaque pixel à la classe "feuilles" ou "autres" en regardant sa probabilité de prédiction pour chacune des classes sur les 10 prédictions.

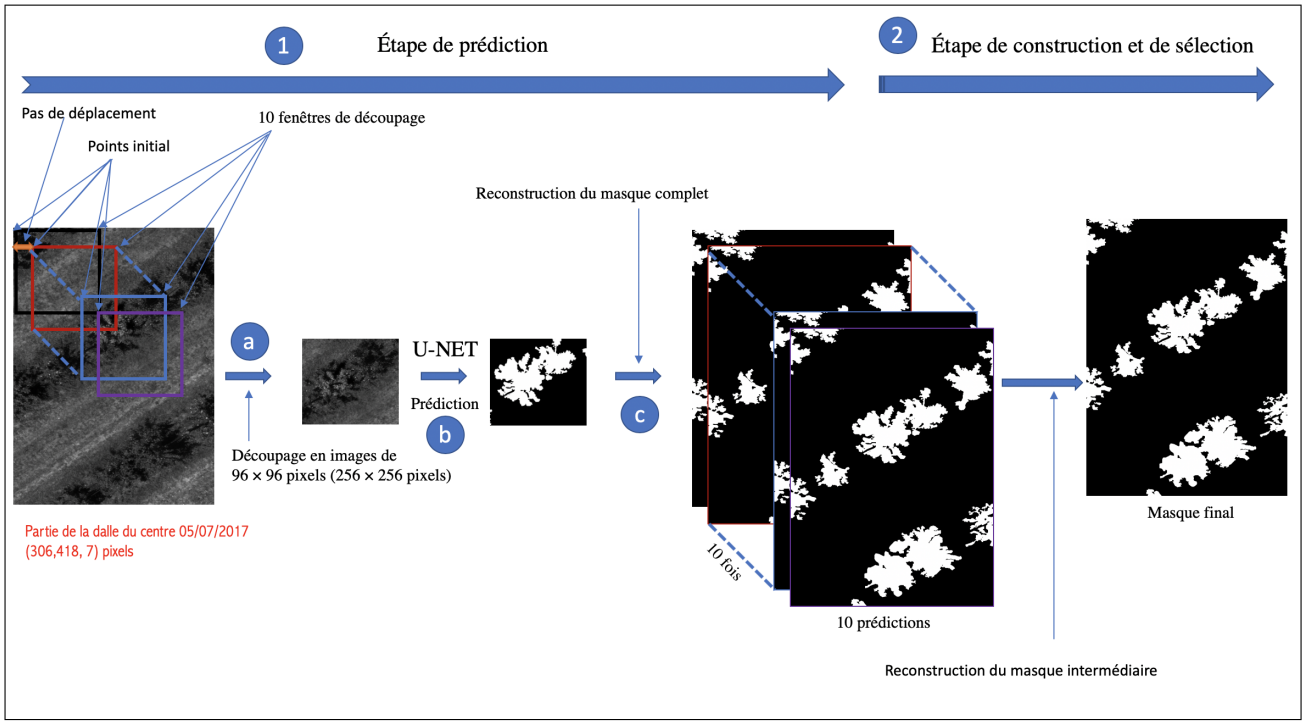


FIGURE 4.6 – Schéma de la méthode de prédiction des subsets(dalles).

4.2.3 Implémentation de la méthode U-NET

Les manipulations de construction des subsets, de découpages des subsets annotés en images et masques, d'augmentation de données et la méthode de prédiction ont été effectuées sous la librairie de programmation python3.6. Pour implémenter le modèle de **U-NET**, nous avons utilisé le programme écrit Sukriti Paul 2018 sur GitHub [4]. Ce modèle est inspiré du modèle d'**Olaf Ronneberger et.al** [32]. Les librairies utilisées sont Tensorflow [5] [35], Keras [6] >= 1 [36].

4. Modèle **U-NET** : Sukriti Paul a publié le 7 juin 2018, un modèle plus simple de **U-NET** écrit sous python. Ce modèle a été conçu pour les image RGB, puis nous l'avons adapté à notre jeu de donnée. Sukriti Paul, Learn How to Train U-Net On Your Dataset. Source : `https://github.com/zhixuhao`.
5. **Tensorflow :** An Open-Source Machine Learning Framework for Everyone. Url : `https://www.tensorflow.org/`.
6. **Keras :** The Python Deep Learning Library. Url : `https://keras.io/`.

Le modèle utilisé de **U-NET** comporte 4 blocs de "Down-convolution" , un bloc de "bottelneck" et 4 blocs de "Up-convolution". Chaque bloc de "Down-convolution' est constitué : d'une couche de convolution avec 16 filtres de taille de 3×3 avec une correction Elu (voir, Section 3.7), une couche Dropout (d'une probabilité de 0.1) [7][33][34]. Puis une autres couche de convolution avec un filtre 3×3 avec 16 filtres de taille de 3×3 avec une correction Elu et une couche de max-pooling de 2×2.

Ce processus se répète 4 fois en doublant le nombre de filtre de convolution jusqu'à 128 filtres. Ensuite, une couche de convolution de 256 filtres de taille de 3×3 avec une correction Elu, un couche Dropout(0.2) et une autre couche de convolution de 256 filtres de taille de 3×3 dans le bloc de "bottelneck". Dans la dernière partie de "Up-convolution", nous avons ainsi 4 blocs : d'une couche de convolution transposée de 128 filtres de 2×2 avec une correction Elu, une couche de concaténation et 2 autres couches de convolution de 128 filtres de taille de 3×3 avec une régularisation Dropout. Ce processus se répète 4 fois, en divisant le nombre de filre par 2 à chaque itération. À la sortie du réseau, nous avons une couche de convolution d'un seul filtre de 1×1 avec une fonction d'activation sigmoïde.

Pour pouvoir l'appliquer, nous l'avons adapté à notre jeu de données. Nous avons en entrée, des images multi-spectrales et thermiques de plusieurs longueurs d'ondes et de taille de 96×96 pixels (respectivement, 256×256 pixels) et en sortie des masque binaire à une seule longueur d'onde de même tailles que les images d'entrée. Notons que pour les deux tailles d'images de 96×96 et 256×256 pixels, nous n'avons utilisé le même modèle. Pour tous les entrainements de **U-NET**, nous avons initialisé les réseaux à une pondération aléatoire. Nous avons choisis un batch$-$size [8]$=5$ (respectivement, batch$-$size$=2$) pour les images de 96×96 (respectivement, 256×256). Le nombre d'époques a été fixé à 50 pour tous les apprentissages.

4.2.4 Méthode SVM de discrimination

Afin d'évaluer l'apport de **U-NET** pour la segmentation sur nos données, nous nous sommes intéressés à la comparaison avec une méthode classique de classe **SVM** [9] de classification uni- et multivariée. Le terme **SVM** signifie "Séparateur à Vastes Marge". C'est un ensemble de méthodes d'apprentissage automatique supervisé pour les tâches de "régression" et de classification".

Ce genre des méthodes s'est montré puissante et efficace pour réaliser les tâches de segmentation et de la détection d'objet dans une image RVB [38][39], elle sont beaucoup utilisées pour les tâches de classification multivariée [37]. Le principe des ces méthodes est de construire un hyperplan séparateur qui sépare efficacement deux ensembles d'individus (pixels) donnés. Une fois paramétré, on peut l'utiliser pour faire des prédictions sur d'autres données de même type.

Ces méthodes fonctionnent de la manière suivante :

1. Dans un premier temps, elles cherchent tous les hyperplans des ensembles. Ces hyperplan sont appelés supports frontaliers des ensembles.

2. Puis, dans un second temps, elles choisissent l'hyperplan de marges optimal, c-à-d l'hyperplan séparant les ensembles à une marge maximale(voir, Figure 4.7).

La méthode **SVM** a été choisie suite aux résultats obtenus dans l'analyse descriptive de données. En effet, les résultats vus dans l'ACP (voir, Section 2.2.1), dans la présentation des pixels (voir, Figure 2.13) des deux classes : "feuilles" et "autres", nous avons remarqué qu'il est possible d'utiliser une approche multivariée pour identifier la nature de ces pixel, c'est donc pour cela que nous avons utilisé une méthode de classe SVM pour la segmentation, puis pour la comparaison avec **U-NET** afin d'évaluer son apport pour la tâche de segmentation sur notre jeu de données par rapport aux méthodes d'apprentissage classiques.

7. **Dropout** : cette manipulation permet la suppression au hazard des neurones du réseau, cela permet de rendre le réseau plus consistant et facilite sa convergence

8. **Batch$-$size** : le nombre d'images lues par périodes lors de l'apprentissage.

9. **Méthodes SVM :** les méthodes SVM ont été apparues les années 1990. Leur apparition était la conséquence des développements théoriques présentés par Vladimir Vapnik. Source : Wikipédia.

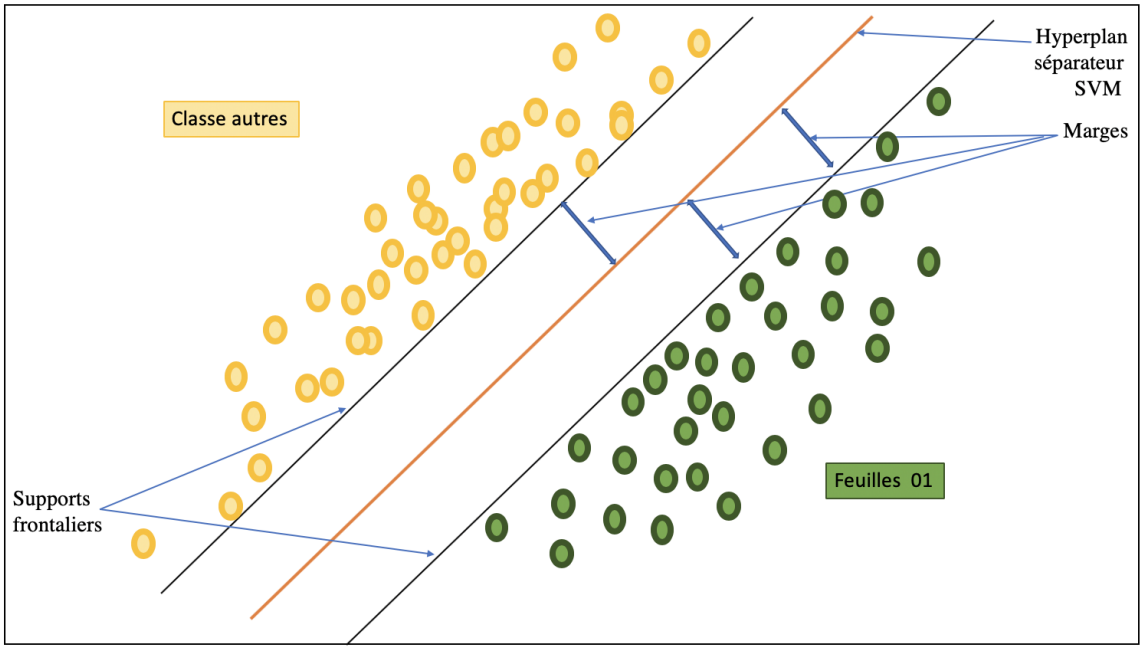


FIGURE 4.7 – Séparateurs SVM.

Implémentation de la méthode SVM

L'implémentation de la méthode SVM a été effectuée sous python en utiisant les librairies scikit-images et scikit-learn [40]. Nous avons utilisé le programme écrit par B. Mathieu [10] pour la localisation d'objets dans une images RVB.

Dans cet algorithme,la fonction SLIC (Simple Linear Iterative Clustering) est utilisée. Elle sert à découper une image en super-pixels [11] pour réduire la taille des données, de l'échelle de pixel à l'échelle de super-pixel, en gardant le maximum d'information sur l'image.. Elle fonctionne comme suit : Dans un premier temps elle regroupe les pixels les plus proches (similaires en couleurs, similaires en textures, etc.) en super-pixels rectangulaires et réguliers. Les super-pixels sont décrits par leur couleur moyenne et la localisation de son barycentre. Elle ré-attribue les pixels en terme de couleur et de localisation au super-pixels jusqu'à qu'ils soient stables [12].

Cette méthode a été appliquée sur des images RGB (comme cité au dessous), donc à trois longueurs d'ondes et de valeurs entières de 0 à 255. Pour pouvoir l'appliquer sur les données multispectrales et thermiques, nous l'avons adapté à notre jeu de données en changeant certains paramètres du modèle, comme le nombre de super-pixels.

Pour entrainer le modèle SVM, nous avons fournis 38($\simeq$ 80% de données) subsets normalisés comme entrées et leurs masques comme sorties. Pour l'apprentissage de **SVM**, nous avons appliqué la méthode SLIC de façon parallèle sur tous les subsets et leurs masques. Donc, nous avons obtenu deux sortes de matrices : une matrice que nous allons nommer **X** des barycentres des super-pixels (des vecteurs multi-spectrale et thermique) et un vecteur **Y** des classes "feuilles" et "autres" correspondantes aux super-pixels. Au final, le modèle **SVM** a été entrainé sur la matrice **X** des barycentres comme entrée et le vecteur **Y** comme sortie. la prédiction a été effectuée sur 7($\simeq$ 20% de données) subsets normalisés.

10. **Programme SVM :** Récupéré du tutoriel intitulé : Localisation d'un objet par classification de super-pixels avec les méthode de machine learning (SVM)— publié 31/10/2017, makina-corpus, Toulouse, France. Source : `https://makina-corpus.com/@@search?Creator=bmt`.
11. **Les super-pixels :** sont de des régions des pixels voisins homogènes (de même intensité).
12. Source : `https://makina-corpus.com/@@search?Creator=bmt`.

4.3 Prétraitement et normalisation de données d'apprentissage et test

La préparation de données est une étape importante pour les méthodes de deep learning. Dans l'analyse descriptive des subsets (voir, Section 2.2), nous avons remarqué les variations inter-date, inter-parcelle et inter-subset, ceux-là peuvent créer des incohérences et de aberrances au niveau de données. De plus, les images multi-sepectrales et thermiques sont de longueurs d'ondes de différentes métriques (réflectance et thermique) et ces longueurs varient dans des intervalles différents. Pour éviter toutes anomalies et mettre ces variables à la même échelle de variation, nous avons fais des transformations appelées "**Normalisation Min-Max (Min-Max scaling)**". Ces transformations permettent de réduire les intervalles de toutes les variations en intervalle $[0, 1]$. Elles ont été effectuées de trois façons différentes, comme nous allons le montrer ci-dessous :

4.3.1 Normalisation par image par canal

La **Normalisation par image par canal** a été effectuée indépendamment sur chaque image d'apprentissage et test. Avant la construction des tableaux d'apprentissage et test du modèle **U-NET**, nous avons fait des normalisations des valeurs de chaque image de la façon suivante :

Soient $V_{i,j,k}$ la j-ème valeur du canal i de l'image k et $V^*(k)_{i,j}$ les valeurs normalisées. Pour toute image $k \in K$. Nous avons pour chaque canal i :

$$V_{i,j}^{*(k)} = \frac{V_{i,j}^{(k)} - \underset{j \in J}{min}(V_{i,j}^{(k)})}{\underset{j \in J}{max}(V_{i,j}^{(k)}) - \underset{j \in J}{min}(V_{i,j}^{(k)})} \tag{4.5}$$

Avec : $\underset{j \in J}{min}(V_{i,j}^{(k)})$: la valeur minimale du canal $j \in J$ et $\underset{j \in J}{max}(V_{i,j}^{(k)})$: la valeur maximale du canal $j \in J$ de l'image $k \in K$.

4.3.2 Normalisation par image

La **Normalisation par image** a été effectuée indépendamment sur chaque image d'apprentissage et du test du modèle **U-NET**. Pour chaque image, nous avons réalisé la transformation suivante :

Soient $V_{i,j,k}$ la j-ème valeur du canal i de l'image k et $V^*(k)_{i,j}$ les valeurs normalisées. Nous avons pour toute image $k \in K$:

$$V_{i,j}^{*(k)} = \frac{V_{i,j}^{(k)} - \underset{k \in K}{min}(V_{i,j}^{(k)})}{\underset{k \in k}{max}(V_{i,j}^{(k)}) - \underset{k \in K}{min}(V_{i,j}^{(k)})} \tag{4.6}$$

Avec : $\underset{k \in K}{min}(V_{i,j}^{(k)})$: la valeur minimale de l'image $k \in K$ et $\underset{k \in K}{max}(V_{i,j}^{(k)})$: la valeur maximale de l'image $k \in K$.

4.3.3 Normalisation par canal

La **Normalisation par canal** a été effectuée sur l'ensemble des images d'apprentissage et test du modèle **U-NET**. Pour chaque canal $j \in J$, nous avons effectué la transformation suivante :

Soient $V_{i,j,k}$ la j-ème valeur du canal i de l'image k et $V^*(k)_{i,j}$ les valeurs normalisées. Pour toutes les image $k \in K$, nous avons pour chaque canal $j \in J$:

$$V_{i,j}^{*(k)} = \frac{V_{i,j}^{(k)} - \underset{j \in J}{min}(V_{i,j}^{(k)})}{\underset{j \in J}{max}(V_{i,j}^{(k)}) - \underset{j \in J}{min}(V_{i,j}^{(k)})} \tag{4.7}$$

Avec : $\underset{j \in J}{min}(V_{i,j}^{(k)})$: la valeur minimale du canal $j \in J$ de toutes les images $k \in K$ et $\underset{j \in J}{max}(V_{i,j}^{(k)})$ la valeur maximale du canal $j \in J$ de toutes les images $i \in I$.

4.3.4 Comparaison des méthodes de normalisation

Pour pouvoir décrire ces trois méthodes de normalisation, nous avons réalisé une comparaison visuelle entre les dispersions des valeurs des variables de réflectances et températures $B_i, i : 1..7$ après chaque normalisation.

Les graphiques ci-dessous montrent les distributions des variables de réflectances et de température d'un subset de la dalle centre 05/07/2017 avant et après chaque normalisation du tableau d'apprentissage du **U-NET**. Nous remarquons que les dispersions des variables des réflectances et températures $B_i, i : 1..7$ obtenues après la normalisation faite par image et par canal varient de façon assez similaire aux subset bruts : elles sont notamment distribuées sur le long de l'intervalle $[0, 1]$, contrairement aux résultats des deux autres méthodes de normalisation.

Les dispersions obtenues des variables $B_i, i : 1..7$ du subset après la normalisation par canal ne sont pas similaire à celles des variables des subsets bruts c-à-d que les subsets d'apprentissage soient hétérogènes. Cela est la conséquence de la mauvaise répartition du couvert végétal et les horaires différents d'acquisition de 10h00 et de 12H00 (voir, Section 2.2).

Les graphiques obtenus de distributions des variables du subset après la normalisation par image montrent que les valeurs des variables de réflectance varient dans des petits intervalles et proche de zéro contrairement à la variable de température. Cela est la conséquence de la différence de grandeurs (unités) et d'échelles entre les variables de réflectance et la variable des températures. En résumé, ces résultats ci-dessous montrent de fortes variabilités inter et intra-image (voir, Section 2.2). La méthode qui semble la plus intéressante est celle par image et par canal.

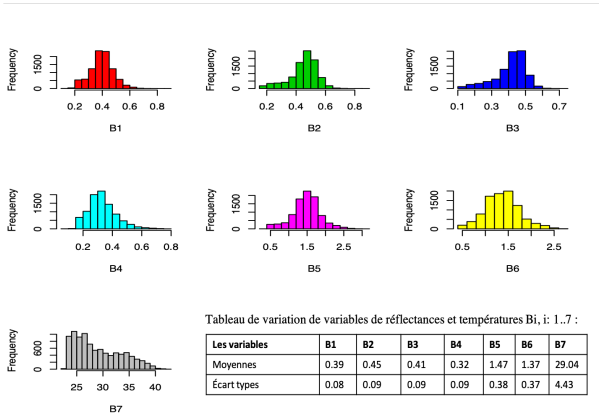


Tableau de variation de variables de réflectances et températures Bi, i: 1..7 :

Les variables	B1	B2	B3	B4	B5	B6	B7
Moyennes	0.39	0.45	0.41	0.32	1.47	1.37	29.04
Écart types	0.08	0.09	0.09	0.09	0.38	0.37	4.43

Sans normalisation (a)

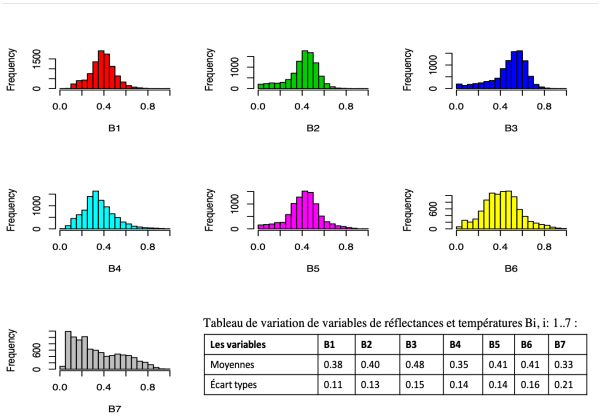


Tableau de variation de variables de réflectances et températures Bi, i: 1..7 :

Les variables	B1	B2	B3	B4	B5	B6	B7
Moyennes	0.38	0.40	0.48	0.35	0.41	0.41	0.33
Écart types	0.11	0.13	0.15	0.14	0.14	0.16	0.21

Normalisation par image par canal (b)

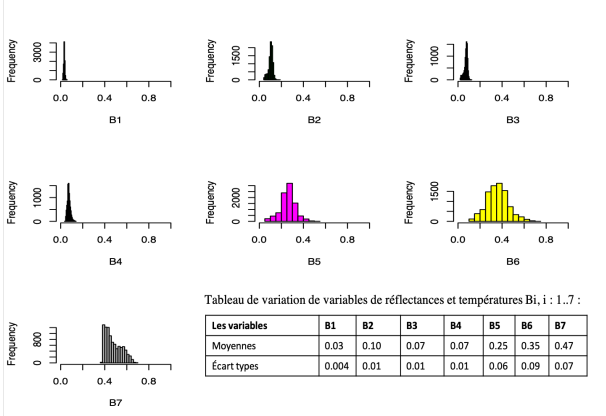


Tableau de variation de variables de réflectances et températures Bi, i : 1..7 :

Les variables	B1	B2	B3	B4	B5	B6	B7
Moyennes	0.03	0.10	0.07	0.07	0.25	0.35	0.47
Écart types	0.004	0.01	0.01	0.01	0.06	0.09	0.07

Normalisation par canal (c)

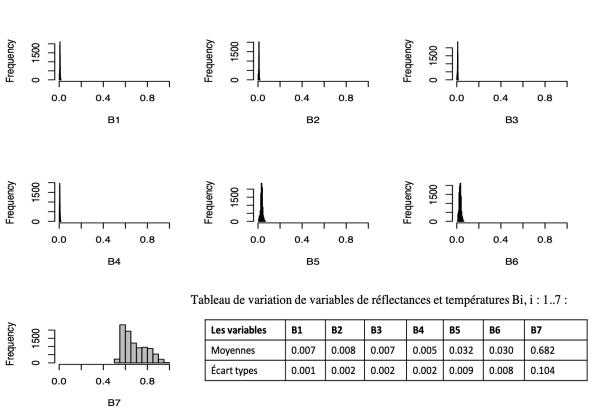


Tableau de variation de variables de réflectances et températures Bi, i : 1..7 :

Les variables	B1	B2	B3	B4	B5	B6	B7
Moyennes	0.007	0.008	0.007	0.005	0.032	0.030	0.682
Écart types	0.001	0.002	0.002	0.002	0.009	0.008	0.104

Normalisation par image (d)

FIGURE 4.8 – Graphes de dispersion des variables de réflectance et de température $B_i, i : 1..7$ d'une image d'apprentissage.

4.4 Impact des paramètres d'apprentissage sur la qualité des prédictions

Les performances des modèles entrainés de **U-NET** ont été évaluées avec différents paramètres d'apprentissage 4(Méthode de normalisation, Taille des images, Seuil de discrimination) par les trois méthodes de normalisation citées précédemment (voir, Section 4.4.1).

4.4.1 Méthodes de normalisation

Pour la sélection de la méthode de normalisation (voir, Section) donnant de meilleurs résultats avec **U-NET**, nous avons effectué des comparaisons entre des modèles entrainés sur des mêmes données d'apprentissage et test mais normalisées en utilisant les différentes méthodes présentées ci-dessus. Ces comparaisons ont été faites avec des modèles entrainés sur des mêmes données d'apprentissage et test de 96×96 pixels(voir, Tableau 4). Nous avons fixé un seuil de discrimination de 0.5 et un batch-size = 5.

Le graphique obtenu ci-dessous (voir, Figure 4.10) montre que la méthode qui donne de meilleurs résultats est celle de normalisation par image et par canal. Cette méthode de normalisation semble meilleure car elle rend mieux compte du niveau de variabilité des différentes longueurs d'onde dans les images multi-sperctrales et thermiques.

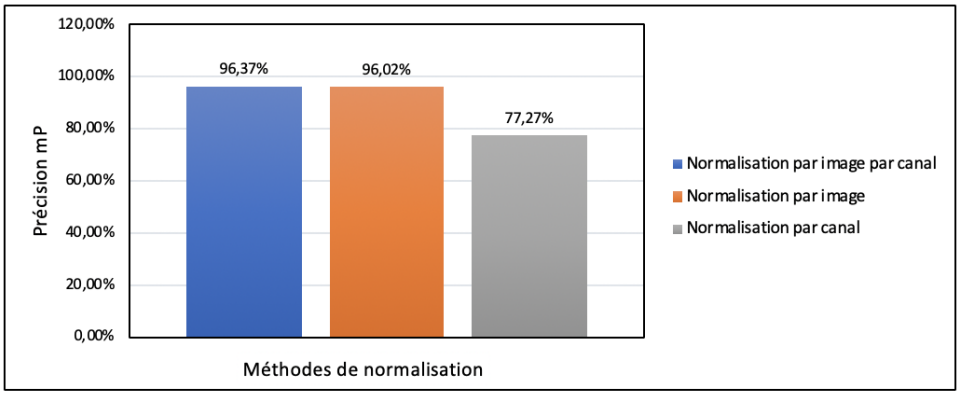


FIGURE 4.9 – Graphe de comparaison entres les méthodes de normalisation de données.

4.4.2 Dates d'acquisition

Pour évaluer la généralisation et la répétabilité de la méthode **U-NET** sur notre jeu de données, nous avons réalisé des comparaisons entre des modèles entrainés et testés sur des données à différentes périodes(voir, Tableau 4). Nous avons effectué deux apprentissages, le premier a été fait sur des données du 05/07/2017 et le test a été réalisé sur des données du 12/07/2017. Le second apprentissage a été réalisé sur les données obtenues du 12/07/2017 que nous avons testé par la suite sur des données acquises le 05/07/2017.

Ces entrainements ont été effectués sur des images de 96×96 pixels avec le même modèle de mêmes paramètres. Nous avons pris un $batch - size = 5$, un nombre d'époque $epoch = 50$ et un seuil de discrimination de 0.5 .

Les résultats obtenus ci-dessous, montrent qu'un modèle entrainé sur des données d'une date peut être utilisé pour générer de bonnes prédictions sur des données acquises à une autre période.

Nous remarquons que le modèle entrainé sur les données du 12/07/2017 génère de meilleurs prédictions que le modèle entrainé sur les données du 05/07/2017. Cela peut être expliqué par le fait de l'heure de l'acquisition, nous avons déjà vu dans la section 2.2 d'analyse de données, que les valeurs des feuilles des arbres et du reste de la parcelle sont mieux séparées pour les données acquises le 12/07/2017. Cela peut être du à la variation de données.

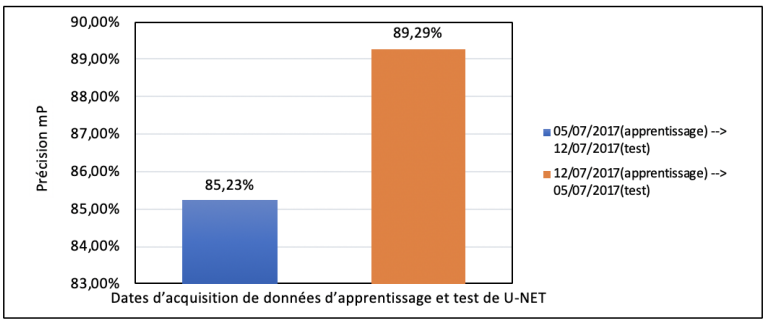


FIGURE 4.10 – Graphe de comparaison des résultats de **U-NET** entre dates d'acquisition de données.

4.4.3 Tailles des images

Pour pouvoir évaluer les performances du modèle **U-NET** par rapport à la taille des images d'apprentissage et du test, nous avons appris le même modèle **U-NET** (de mêmes paramètres et seuil) sur 100 images de 96×96 pixels (respectivement, 256×256 pixels) et le test a été fait sur 10 images de 96×96 pixels (respectivement, 256×256 pixels) de 45 subsets de deux dates d'acquisition. La normalisation des image a été faite par image par canal. Nous avons pris un nombre d'époques à 50, un seuil de discrimination de 0.5 et un $batch - size = 5$.

Les résultats obtenus par rapport à la taille des images (figure ci-dessous), montrent que les valeurs de précision mP sont plus élevées pour les images de 96×96 pixels que pour celles de 256×256 pixels, cela témoigne du fait que le modèle **U-NET** est plus performant quand il apprend sur des images de plus petites tailles. Pour un même nombre de données en entrées, un modèle 256×256 pixels a plus de paramètres à ajuster et il est donc de moins bonne qualité. À noter cependant que des réseaux trop petits même avec un nombre de données suffisants peuvent être limités dans leurs capacités d'apprentissage et donc de prédiction. Dans notre cas, une taille d'image de 96×96 pixels semble donner un bon compromis entre capacité d'apprentissage et capacité de prédiction.

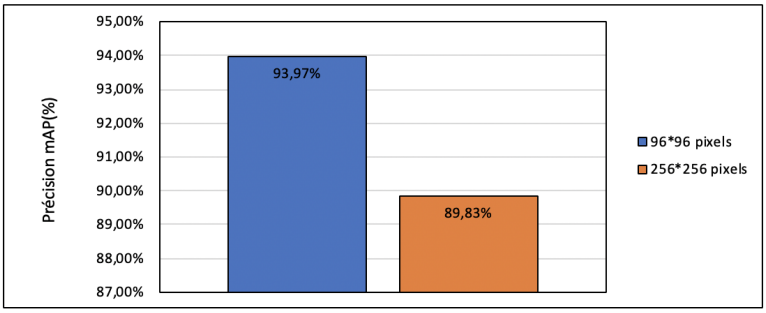


FIGURE 4.11 – Graphe de comparaison des résultats de **U-NET** selon les tailles des images.

4.4.4 Seuil de discrimination

Le seuil de discrimination est un paramètre important pour la prédiction car il définit le seuil pour l'appartenance d'un pixels à une classe. Dans notre étude, nous avons deux classes des pixels prédits : "feuilles" et "autres". Un pixel prédit est dit dans la classe "feuilles" si sa valeur prédite est au-dessus du seuil. Il est intéressant de comprendre le comportement et les précisions du modèle entrainé en fonction du seuil de discrimination. Cela va permettre par la suite de mieux classer les pixels dans les classes "feuilles et "autres".

Pour cela nous avons fais apprendre un modèle sur des images de 96×96 pixels normalisées par image et par canal. Cet apprentissage a été faits sur 37 subsets et le test sur 8 subsets restants des deux dates d'acquisition. Nous avons fixé un $bach - size = 5$ et un nombre d'époque à $epoch = 50$.

Dans la figure 4.12 ci-dessous, nous remarquons que les valeurs de précision mP obtenues sur les données test varient de façon constante par rapport aux valeurs du seuil. Cela veut dire que le modèle arrive à différencier les pixels des feuilles des arbres des pixels de la végétation basse et du sol pratiquement sur tout l'intervalle sauf aux extrémités proches de 0 ou 1. Pour ces subsets du test, nous avons trouvé le seuil optimal donnant la précision maximale (mP$\simeq$ 96%) égal à 0.8. La sensibilité du modèle a ce seuil semble faible.

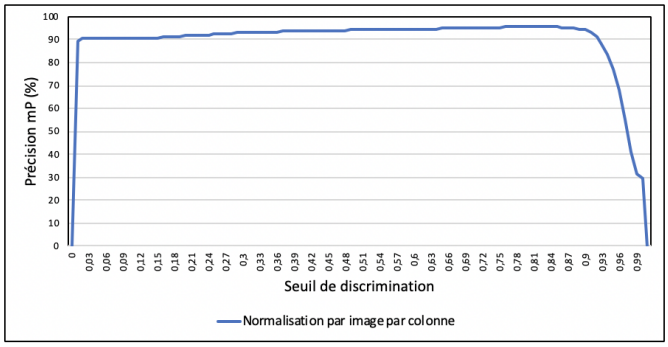


FIGURE 4.12 – Graphe d'évaluation de la précision (mP) du modèle **U-NET** par rapport au seuillage.

4.5 Comparaison U-NET/SVM

Pour évaluer l'apport de la méthode **U-NET**, nous avons fait une comparaison de celle-ci avec la méthode d'apprentissage automatique **SVM**. Afin de pouvoir réaliser cette comparaison, nous avons fait de l'apprentissage sur les mêmes subsets et masques des deux dates d'acquisition du 5 et 12/07/2017 et nous avons utilisé la même méthode de normalisation "par image et par canal" (voir, Section 4.3.3). La figure ci-dessous présente les moyennes mP des prédictions sur les mêmes subsets du test, elles sont obtenues à partir des différents modèles entrainés sur différents paramètres :

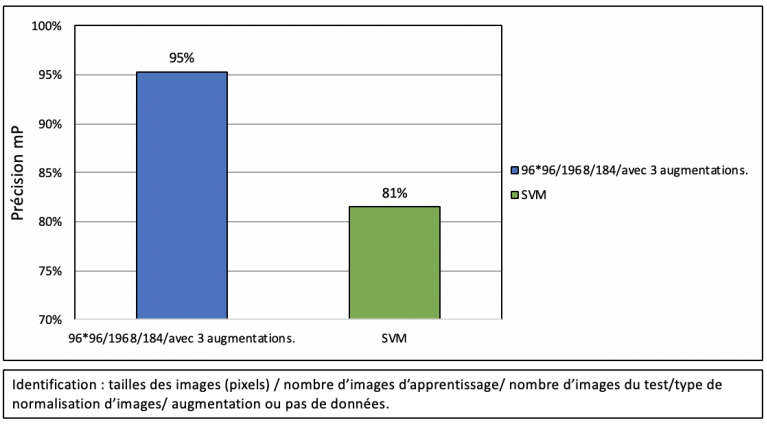


FIGURE 4.13 – Graphe de comparaison U-NET/SVM.

Les résultats figurant ci-dessus montrent montrent que la méthode **U-NET** est plus précise que la méthode SVM ; sa précision mP sur les données du test atteint $\simeq$ 96%.

Comparaison visuelle des prédictions U-NET/SVM d'un subset du test

Pour mieux comparer ces deux méthodes, nous avons fait deux prédictions sur le même subset de données. Ces prédictions ont été réalisées avec les modèles entrainés cités dans la section précédente 4.5 pour les deux méthodes et sur les mêmes données d'apprentissage. Pour les prédictions, nous avons fixé le seuil de discrimination à 0.5 pour les deux méthodes.

Les visualisation suivantes montrent que la méthode **U-NET** est plus précise que la méthode **SVM**, la méthode **U-NET** distingue mieux les pixels des feuilles des arbres du reste de la parcelle contrairement à la méthode **SVM**. Cela peut être expliqué par le fait que la méthode **SVM** est uniquement basée sur les intensités des canaux des pixels qui peuvent etre proches entre les pixels de végétation basse et des pixels des feuilles des arbres. La méthode **U-NET** a aussi la possibilité de prendre en compte des motifs d'organisation des pixels que ne considèrent pas **SVM**.

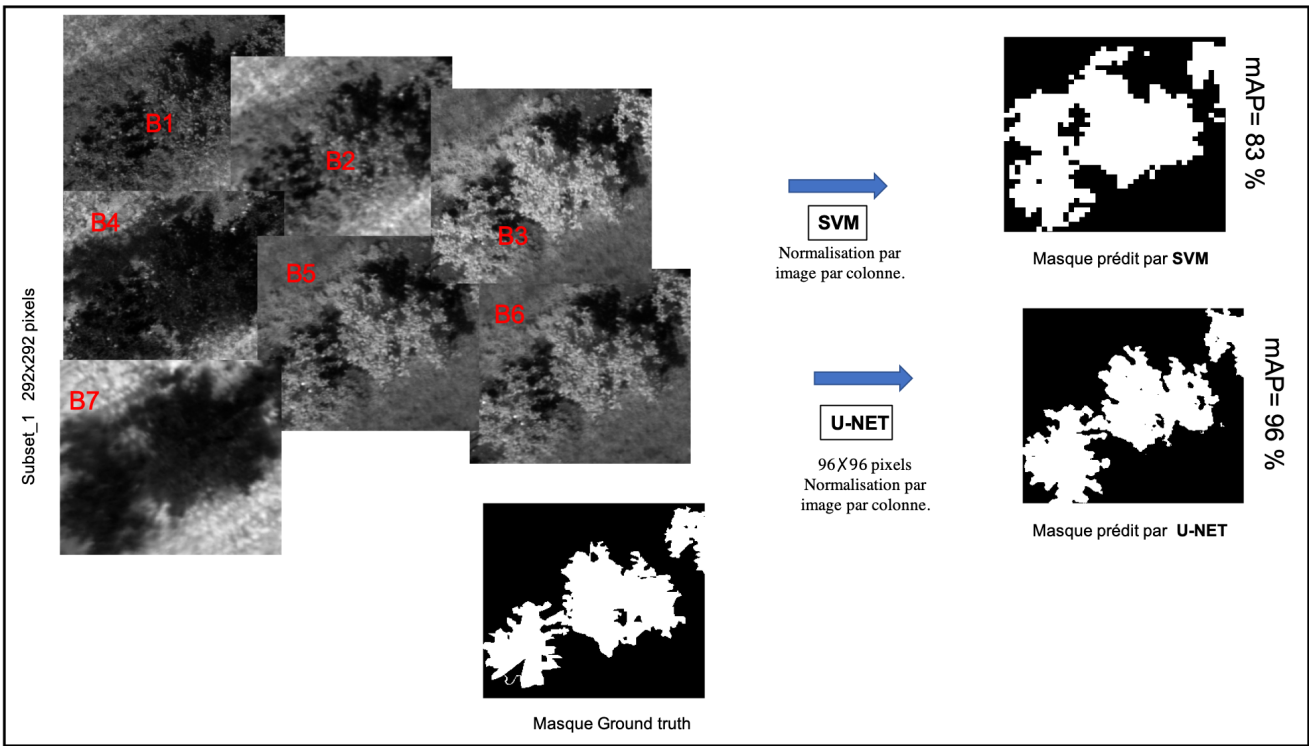


FIGURE 4.14 – Comparaison visuelle U-NET/SVM sur un subset de données test.

4.5.1 Comparaison U-NET/ SVM sur une partie de la dalle

Pour confirmer la précision et l'efficacité de la méthode **U-NET** par rapport à la méthode **SVM**, nous avons fait des prédictions sur un subset de plus grandes dimensions. Ce subset est indépendant des données d'apprentissage et de test. Pour les prédictions, nous avons utilisé les modèles entrainés sur les mêmes données d'apprentissage cités précédemment (voir, Section 4.5). Pour les prédiction, nous avons utilisé un seuil de 0.5.

Les résultats obtenus dans la figure ci-dessous montrent que la méthode **U-NET** est plus précise, elle produit de bonne prédiction que la méthode **SVM**, elle sépare et discrétise mieux les pixels des feuilles des pixels de végétation et du sol.

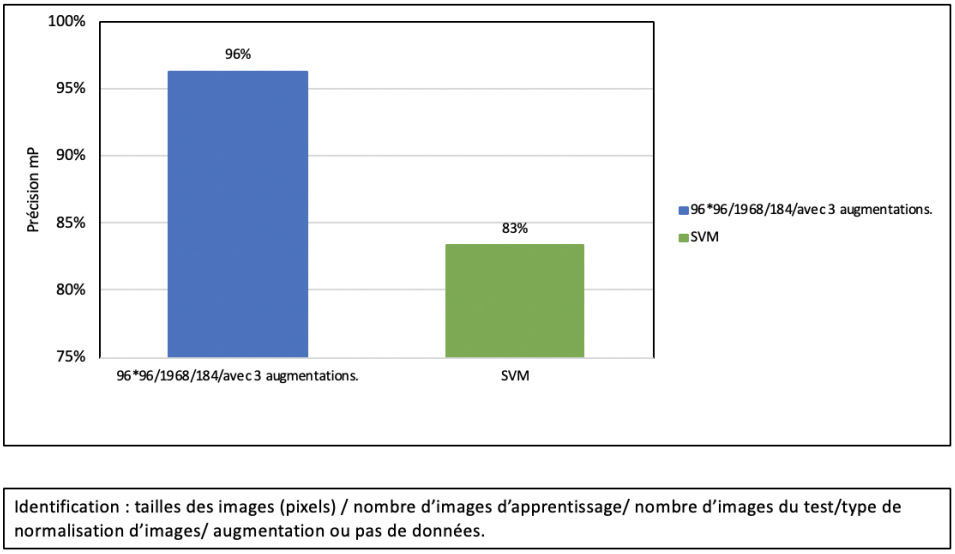


FIGURE 4.15 – Graphique en barres de comparaison des prédiction **U-NET**/**SVM** sur le subset 33_5.

Le masque prédit obtenu par le modèle entrainé de **U-NET** est plus précis, nous voyons clairement les différents arbres du subsets. Contrairement à la prédiction obtenue par le modèle de **SVM**, où son modèle entrainé ne parvient pas à séparer les feuillages de la végétation à proximité des arbres.

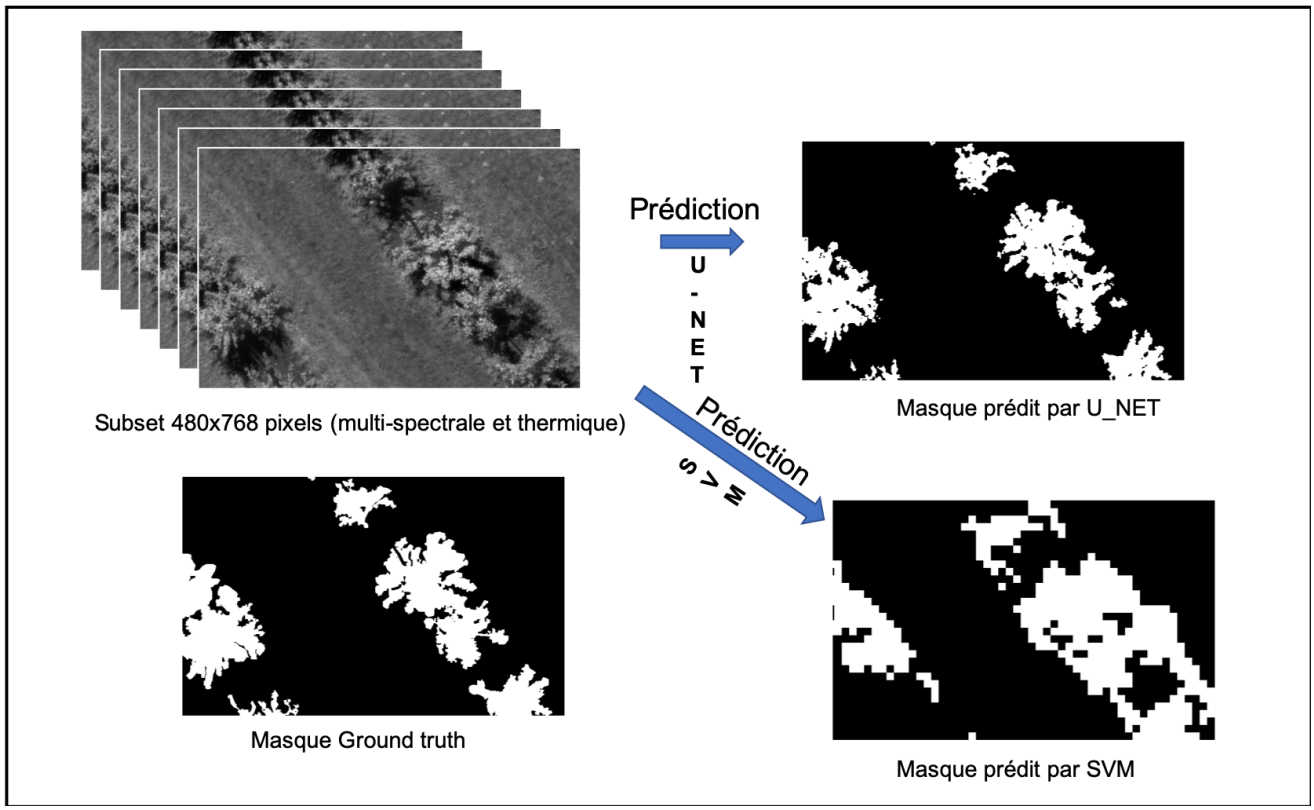


FIGURE 4.16 – Comparaison visuelle.

4.6 Prédiction d'une partie de la dalle

Pour l'utilisation du modèle entraîné de **U-NET** sur notre jeu de données, nous avons fait une prédiction sur une partie de la parcelle. Nous avons pris le meilleur modèle entraîné sur 37 subsets des deux dates d'acquisition. Nous avons fixé un $batch - size = 5$ et un nombre d'époque à $epoch = 50$. Pour la prédiction nous avons utilisé un seuil de discrimination égal à 0.8 . Les résultats montrent une bonne identification des pixels de feuillage sur toute la parcelle.

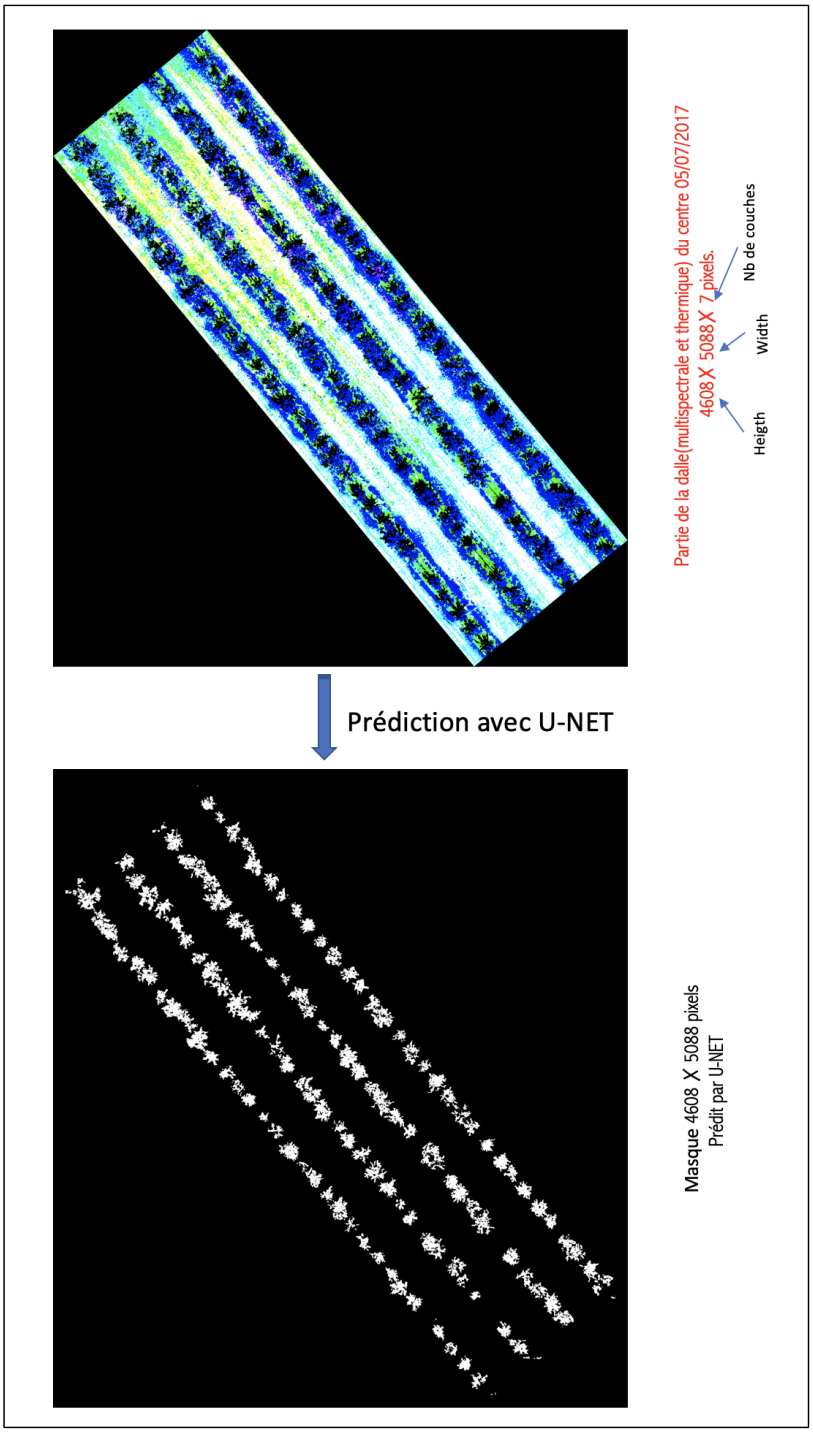


FIGURE 4.17 – Prédiction d'une partie de la dalle par **U-NET**.

4.7 Extraction des valeurs des feuilles et calcul d'indices de végétation

Le but final de ce travail a été de pouvoir réaliser des extractions des valeurs multi-spectrales et thermiques du feuillage de différents arbres pour déterminer des indices de végétation à l'échelle de l'arbre. Pour cela, nous avons mis en place le pipeline suivant (voir, Figure 4.18) :

1. Étape de prédiction : consiste à identifier les feuillages des arbres de la parcelle.
2. Étape de séparation et d'identification des arbres : nous avons utilisé la méthode **WATER-SHED** (voir, Section 4.7.1) pour séparer les arbres dans le masque prédit. Pour cela, nous avons pris les centres des arbres (identifiés par des coordonnées GPS) comme des graines et puis nous avons assigné à chaque pixel des feuilles l'identifiant de l'arbre qui le contient.
3. Étape du calcul des indices de végétation : Les indices de végétation des arbre sont calculés à partir des valeurs des pixels de feuilles de chaque arbre. Nous avons calculé pour chaque pixel de feuilles les valeurs d'indices de végétation (NDVI, GNDVI, PIR, TS-TA) et nous en avons calculé la moyenne pour chaque arbre. (voir, Tableau 4.1).

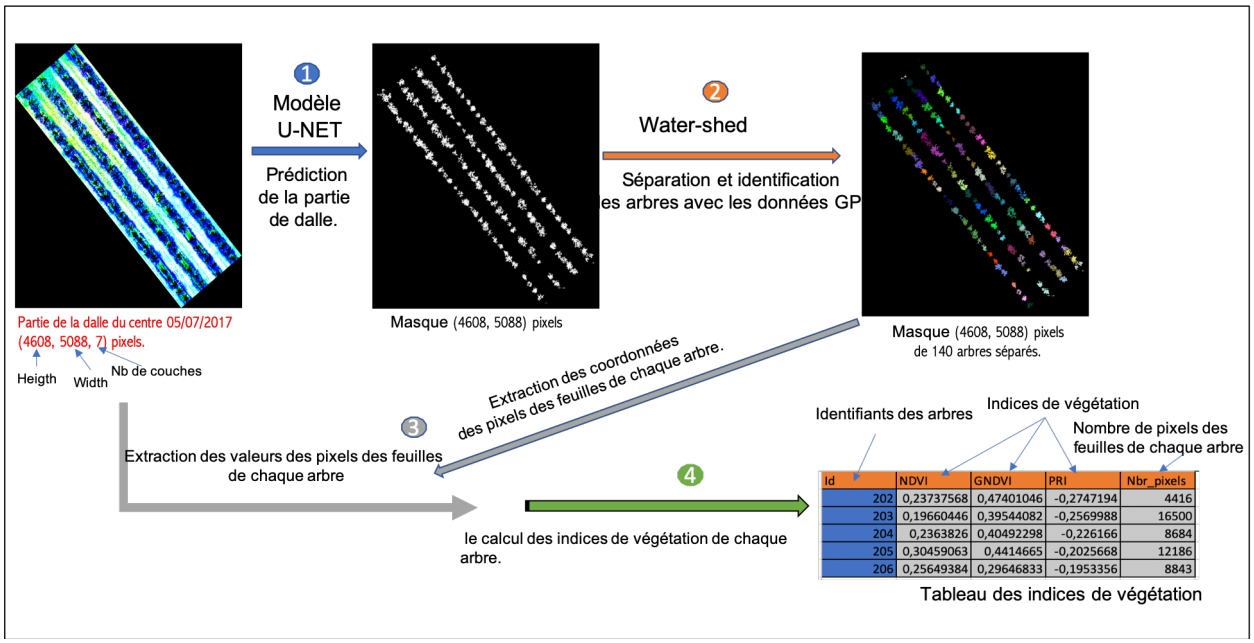


FIGURE 4.18 – Schéma d'extraction des valeurs des feuilles et du calcul des indices de végétation de chaque arbre.

4.7.1 Séparation des feuillages des arbres par la méthode Water-shed

Pour la séparation et l'identification des feuillages des des arbres de la parcelle, nous avons utilisé l'algorithme Water-shed. En effet, la méthode water-shed [41] est un ensemble de transformations mathématiques classiques conçues pour la tâche segmentation par instances de l'image. Elle fonctionne en deux étapes principales : dans un premier temps, elle considère l'image comme un relief topographique avec la hauteur de chaque pixel définit en fonction de son intensité. En second temps, elle calcule les frontières des régions (appelées "région-bassin") par des estimations en fonctions mathématiques [41]. Ces frontières sont appelées "lignes de partage des eaux". L'identification des régions de l'image peut être par une classification hiérarchique ou par indication de source de ces régions appelées graines.

Implémentation de la méthode Water-shed

L'implémentation de cette méthode a été faite sous python 3.6. Nous avons utilisé le package "scikit-image" pour séparer les arbres dans les masques prédits en prenant les centres des arbres identifiés par des coordonnées **GPS** comme graines.

Les difficultés de la paramétrisation de la méthode Water-shed résident dans la définition des positions d'arbres au niveau de la parcelle. Si un subset contient une partie de l'arbre sans l'information de positionnement GPS, nous n'aurons pas sa graine, alors le water-shed attribuera cette partie aux autres arbres situés aux alentours. Les cas difficile sont lorsqie des arbres sont "collés" et n'ont donc pas de frontière claire.

Pour évaluer cette méthode, nous avons pris 10 subsets annotés manuellement (utilisés pour **U-NET**) acquis le 05/07/2017. La méthode de water-shed a été appliquée sur les masques binaires. Pour l'identification des arbres, nous avons enregistré une précision de 100%, et pour la classification de pixels, nous avons trouvé une précision de 98%.

4.7.2 Évaluation des calculs d'indices de végétation obtenus par U-NET

Pour évaluer la qualité d'extraction des valeurs des feuilles et de calcul d'indices de végétation par la méthode **U-NET**, nous avons comparé avec les résultats obtenus à partir des segmentations manuelles (vérités terrain), les résultats déjà obtenus avec le logiciel ERDASIMAGINE et les résultats obtenus avec la méthode **U-NET** (voir, Tableau 13) sur les mêmes arbres de la parcelle.

Pour ce faire, nous avons fait des prédictions avec un modèle entrainé de **U-NET** sur 7 subsets du test de la dalle centrale et ouest acquis le 05/07/2017. Nous avons identifié et séparé les arbres de chaque subset (au total, nous avons identifié 20 arbres), puis nous avons extrait les valeurs de réflectances et températures des feuilles des arbres. Ensuite, nous avons calculé pour chaque pixel feuille les indices : NDVI, GNDVI, PIR, TS-TA, puis nous calculons la moyenne de ces indices pour chaque arbre.

Nous avons remarqué dans le tableau des résultats ci-dessous 4.1 de fortes corrélations entre les valeurs d'indices de **U-NET** et les indices de vérité terrain. Cela montre que les estimations d'indices de végétation obtenues par **U-NET** semble aussi précis que les vérité terrain. Par contre, des mauvaises corrélations sont obtenus par la méthode défini via le logiciel ERDAS. Cela peut s'expliquer par une sélection des pixels representifs de chaque arbre très différent dans cette méthode.

TABLE 4.1 – Tableau d'indices de végétation des valeurs extraites par U-NET.

		Extraction par ERDAS				Extraction par U-NET			
		Ts -Ta	NDVI	PRI	GNDVI	Ts - Ta	NDVI	PRI	GNDVI
Les valeurs de vérité terrain	Ts -Ta	-0.03				0.98			
	NDVI		0.78				0.88		
	PRI			-0.27				0.99	
	GNDVI				0.15				0.99

Conclusion :

Dans ce chapitre, nous avons pu adapter et appliquer la méthode **U-NET** pour notre jeu de données. Puis nous avons comparé par rapport à la méthode classique **SVM** pour la segmentation sur notre jeu de données. Nous avons ensuite introduit l'utilisation de la méthode **Water-shed** pour la séparation des feuillages des arbres, puis une méthode d'extraction des valeurs des feuilles de chaque arbre. À la fin, nous avons évalué la qualité et la précision des calculs d'indices de végétation trouvés par l'utilisation de **U-NET**. Les résultats semblent de bonnes qualités. Il faudrait maintenant tester cette approche à d'autres dates de manière plus ample pour voir si une telle approche automatise de manière efficace et précise la détermination d'indice de végétation à partir d'images multispectrales et thermiques acquises par drone.

Chapitre 5

Segmentation et séparation des feuillages d'arbres par le **MASK R-CNN**

Introduction :

L'objectif de notre étude est de segmenter et détecter les différents pommiers de la parcelle sur les images multi-spectrales et thermiques (voir, Section 2.1.1). Un premier pipeline basé sur les méthodes **U-NET** et **Water-shed** a été introduit dans le chapitre précédent. Il nécessite cependant les coordonnées GPS de chaque arbre pour pouvoir les identifier efficacement. Récemment des méthodes de deep learning ont été introduites pour segmenter et identifier sans a priori les différents instances d'une même classe d'objet dans une image. Dans ce chapitre, nous décrivons nos premières tentatives pour appliquer cette méthode, appelée **mask R-CNN** , sur notre jeu de données. Nous décrivons d'abord la méthode ainsi que les processus de construction des données d'apprentissage et test pour celle-ci. Ensuite, nous présentons les premiers résultats de prédiction obtenus sur nos données. Pour illustrer cela, la figure ci-dessous 5.1 montre un exemple de segmentation par instance faite manuellement sur une des partie de la parcelle.

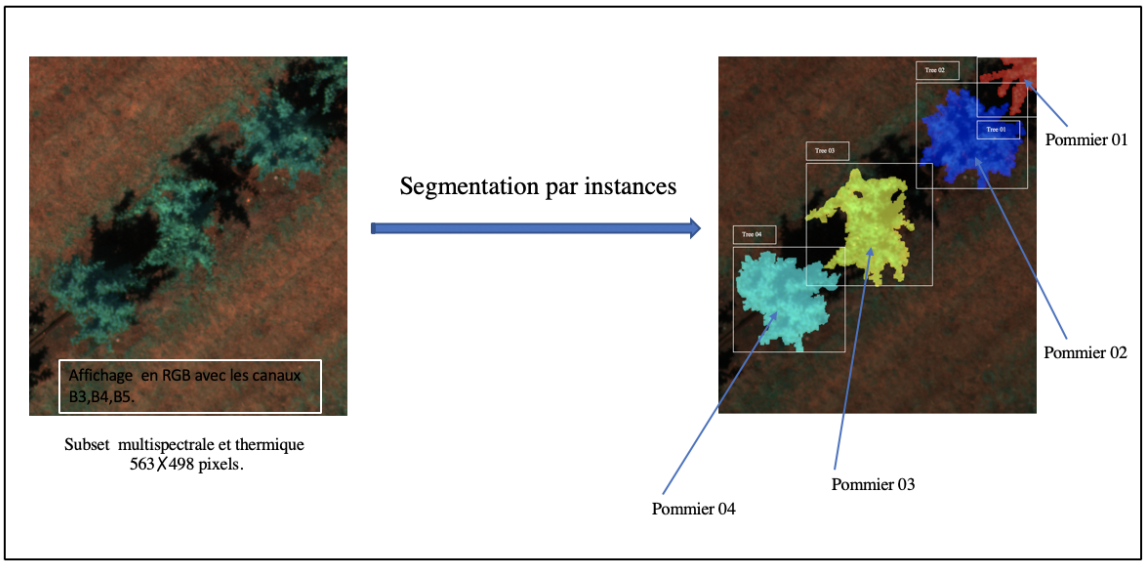


FIGURE 5.1 – Exemple de segmentation par instances d'une partie de la parcelle.

5.1 Présentation de la méthode

La méthode **Mask R-CNN** a été inventée par Kaiming He et al. 2017 [44]. Cette méthode permet de résoudre le problème en vision par ordinateur qui consiste à identifier et segmenter des instances d'un objet dans une image. Cette méthode fonctionne en plusieurs étapes principales : dans un premier temps, elle identifie les instances d'objets de différentes classes dans l'image, puis pour chaque instance, elle identifie les pixels lui correspondant par une approche de segmentation.

Au final, la méthode **Mask R-CNN** classifie chaque pixel de l'image à une instance ce qui permet de faciliter la localisation, l'identification et la séparation de différents objets dans une image.

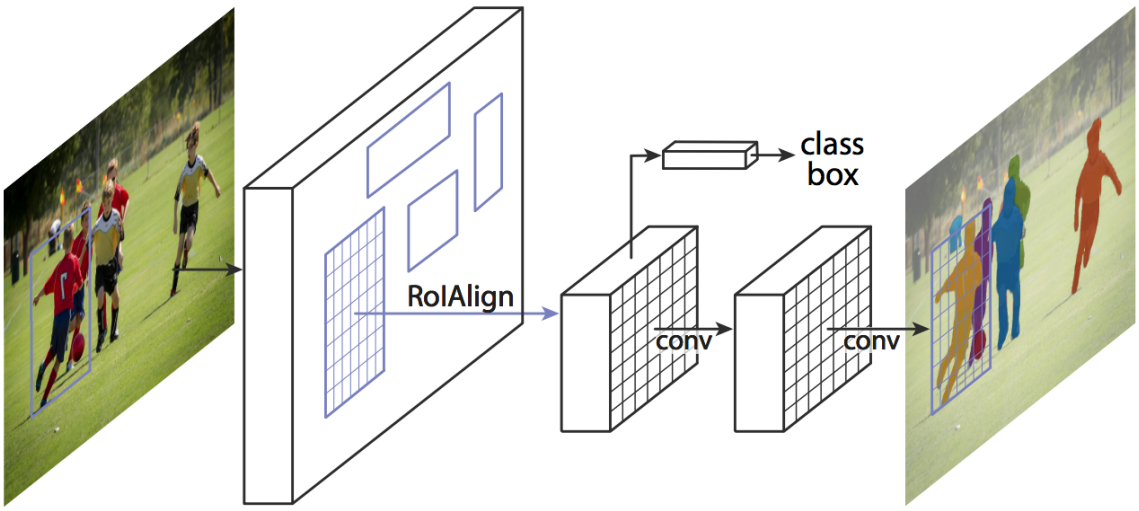


FIGURE 5.2 – Architecture Mask R-CNN.
Source : Kaiming He. `https://arxiv.org/abs/1703.06870`

5.1.1 Historique et quelques applications du Mask R-CNN

La méthode **Mask R-CNN** a été le résultat d'une suite d'amélioration de différents algorithmes. En 2014, R. Girshick [42] a présenté une architecture appelée **R-CNN** (Region Convolution Neural Network) pour la détection et la localisation d'objets dans une image. Cette structure permet de prendre une image brute comme entrée puis elle fait une extraction de $\simeq$ 2000 régions d'intérêt en utilisant la méthode de **recherche sélective des régions** inventée par Uijlings et al.2013 [43]. La méthode de la **recherche sélective des régions** est basée sur les échelles, la variation et graduation des couleurs pour la détection et la propositions de régions d'intérêt d'une images.

Pour chaque région extraite de l'image, la méthode **R-CNN** applique : un CNN pour l'extraction des caractéristiques, un classifieur SVM pour l'identification de l'objet et un modèle de régression linéaire pour la construction des contours des régions. Ces différentes transformations au niveau de chaque région de l'image multiplient le temps de calcul de l'apprentissage et de prédiction.

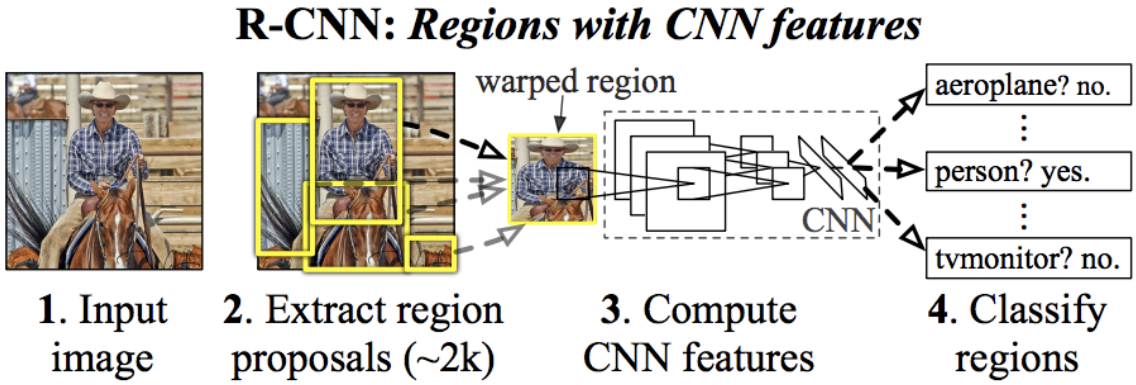


FIGURE 5.3 – Architecture du R-CNN.
Source : R.Girshick.2015 [42], Source : `https://arxiv.org/abs/1703.06870`

En 2015, R. Girshick [44] a construit un nouveau modèle **Fast R-CNN**, en améliorant la structure précédente de **R-CNN**. La méthode du **Fast R-CNN** n'utilise qu'un seul CNN pour toutes les régions d'intérêts afin de réduire le temps de calcul. Avec le **Fast R-CNN**, il faut en moyenne deux secondes pour détecter des objets dans une image de 1024×1024 pixels (avec une machine équipée d'un GPU), ce qui reste lent, surtout quand il s'agit de traiter de grande quantité d'images. Le temps de prédiction est une vraie contrainte de celle-ci, car elle utilise la méthode de **recherche sélective des régions** qui est un processus très long. De plus, elle applique un classifieur SVM et un régression pour chaque région d'intérêt. Ce qui rend la méthode lente en temps d'apprentissage et de prédiction.

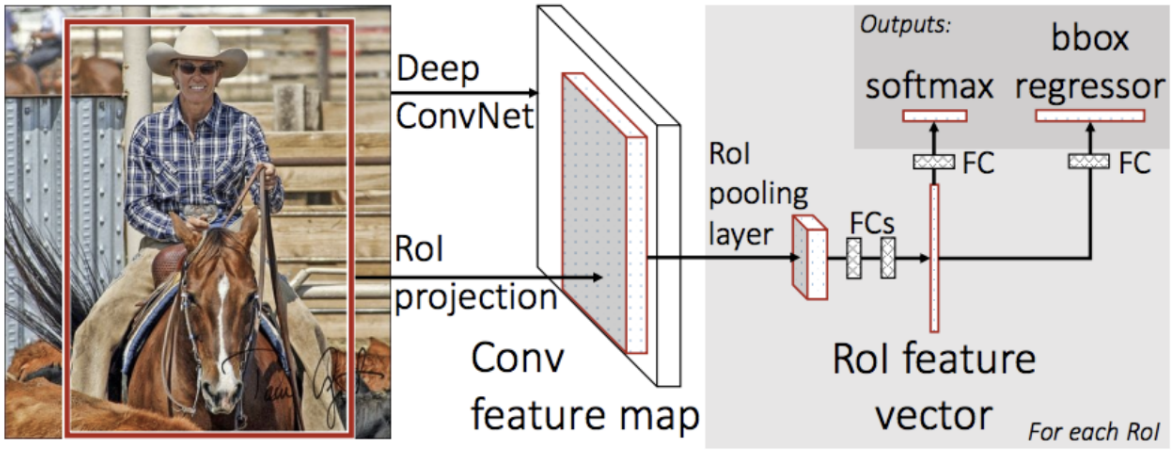


FIGURE 5.4 – Architecture du Fast R-CNN.
Source : R.Girshick.2016 [42], Source : `https://arxiv.org/pdf/1504.08083.pdf`.

Dans la même année, R.Girshick [45] a proposé une nouvelle architecture améliorée du **Fast R-CNN** pour la détection et localisation d'objets dans une image, elle est appelée **Faster R-CNN**. Dans la méthode **Faster R-CNN**, les auteurs introduisent l'utilisation d'un réseau spécifique pour la détection de région appelé **RPN** (région proposal network). Grâce à cela, les temps de calculs sont grandement améliorés (0.2 seconde pour traiter/apprendre sur une image).

La méthode **Faster R-CNN** fonctionne de la façon suivante : d'abord, elle applique un réseau de neurones à convolution sur l'image d'entrée pour extraire des cartes de caractéristiques, puis elle réduit le nombre de régions d'intérêt en appliquant un réseau de proposition de régions **RPN** [1] (Region Proposal Network). Ensuite, elle utilise une couche **ROI pooling layerROI pooling layer** : c'est une couche des neurones, elle sert à réajuster les tailles des images d'entrée par l'utilisation des fenêtres glissantes comme vu dans la couche max-pooling 3.6. pour mettre les régions d'intérêt à la même dimension. Ces régions seront transformées par des couches complètement connectées pour les remettre en vecteur d'une seule dimension, ils sont appelés **ROI feature vector**. Ces vecteurs **ROI feature vector** seront transmis par la suite pour la prédiction des classes des régions en appliquant une couche **Softmax**, ainsi que pour la construction des boites englobantes des régions en utilisant d'autres couches complètement connectées.

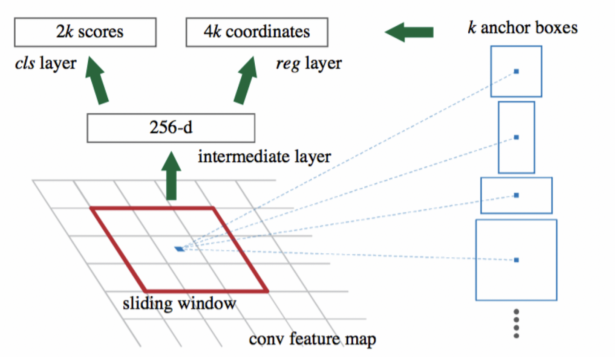


FIGURE 5.5 – Architecture du RPN.

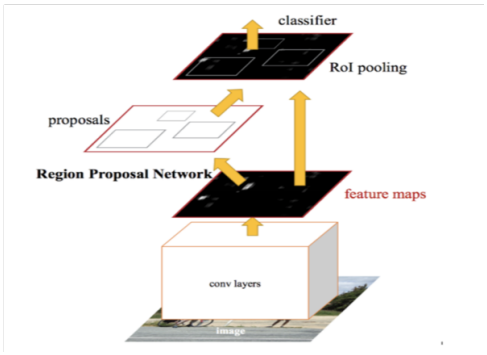


FIGURE 5.6 – Architecture du Faster R-CNN.

Source : R.Girshick.2015 [45], Source : `https://arxiv.org/pdf/1506.01497.pdf`.

En 2017, Kaiming He et al [46] ont proposé une extension de **Faster R-CNN** appelée **Mask R-CNN**. Celle-ci permet la détection et la localisation d'objets dans une image ainsi que la construction du masque de chaque région d'intérêt.

La méthode du **Mask R-CNN** fonctionne comme suit : elle applique un CNN pour l'extraction de cartes de caractéristiques, puis elle sélectionne les régions d'intérêt les plus pertinentes par un réseau de proposition de régions **RPN**. Ensuite, elle met toutes les régions sélectionnées en mêmes dimensions par une couche de neurones appelée **ROI Align**. À la fin, ces régions seront fournies à une branche des couches complètement connectées pour la classification et la construction de boite englobante pour chaque région et à une autre branche de réseau de neurones pour la construction des masques.

La méthode **Mask R-CNN** a été largement utilisée dans de nombreux domaines de vision par ordinateur tels que : la médecine [47] pour l'analyse des troubles du rythme circadien. Ces analyses ont été faites par le rapport du nombre de trouble atteintes la patient, par le nombre de personne détectées par le **Mask R-CNN** visitant le patient par jour. le milieu marin [48] pour la détection et la quantification du nombre de navires côtiers.

Cette méthode a été largement utilisée dans le domaine de l'agriculture pour le comptage et la détection des fruits (pommes) dans un verger par S.Bargoti et al.2017 [49]. G.Bernotas et al.2018 [50] ont montré l'importance et la précision que peut apporter la méthode **Mask R-CNN** pour la segmentation des feuilles ainsi que pour le phénotypage des plantes.

1. **RPN** : est un type des réseau de neurones artificiels, il utilise des fenêtres glissantes de différentes tailles pour une analyse fine et en profondeur de cette carte de caractéristiques afin de générer un nombre réduit et précis de régions d'intérêt.

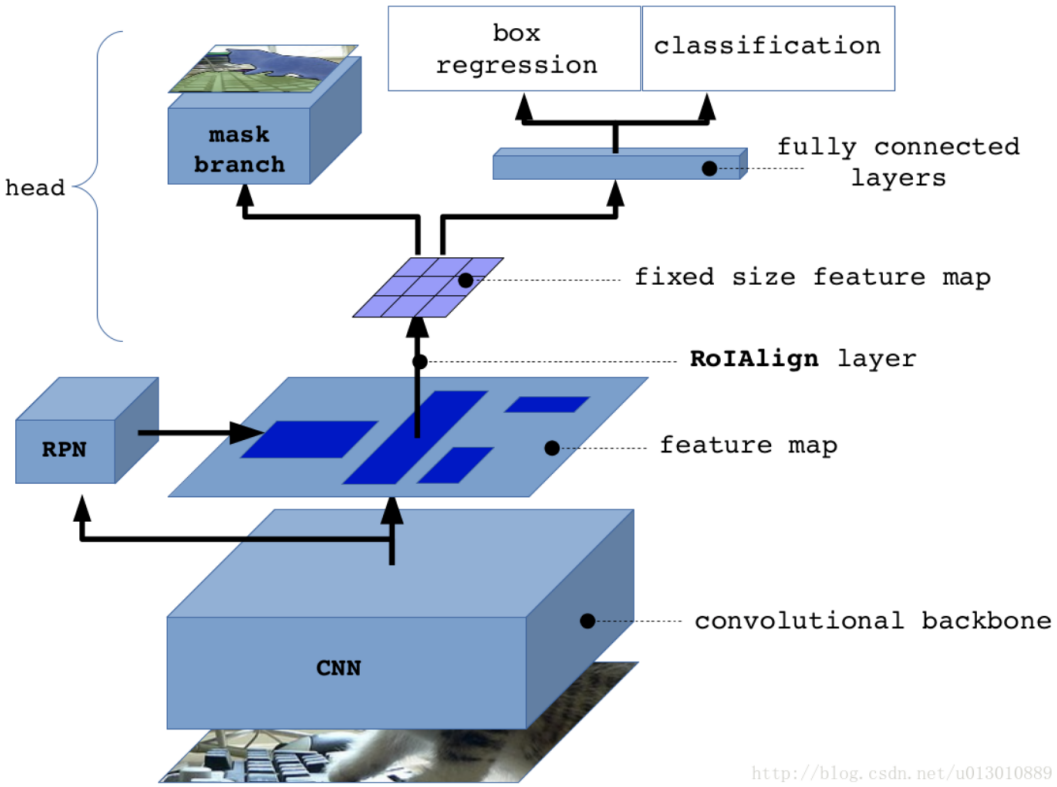


FIGURE 5.7 – Architecture du Mask R-CNN.
Source : Kaiming He et al.2017 [46], Source : `https://arxiv.org/abs/1703.06870`.

5.2 Préparation de données du Mask R-CNN

Dans cette partie, nous allons présenter les différentes manipulation effectuées pour la construction de données d'apprentissage et du test de la méthode **Mask R-CNN**.

5.2.1 Construction des masques de différents arbres

La construction des données d'apprentissage et test du **Mask R-CNN** s'effectue en 4 étapes :

1. Étiquetage et annotation de données multi-spectrales et thermique : nous avons pris les 45 subsets annotés utilisés dans le chapitre précédent pour l'apprentissage et test de la méthode **U-NET**.

2. Séparation des masques des feuillages des arbres : elle a été appliquée sur les masques binaire. Pour ce faire, nous avons utilisé la méthode des composantes connexes (connected components) pour distinguer les arbres connexes, puis la méthode de Water-shed (voir, 4.7.1) pour séparer les arbres dans chaque subset.

3. Identification et séparation en masque d'arbres : Pour chaque subsets contenant plusieurs arbres, un masque par arbre est généré.

4. Découpage des subsets et masques en images de 256×256 pixels : Durant cette étape, nous faisons un découpage des subsets multi-spectrales et thermiques en images de 256×256.

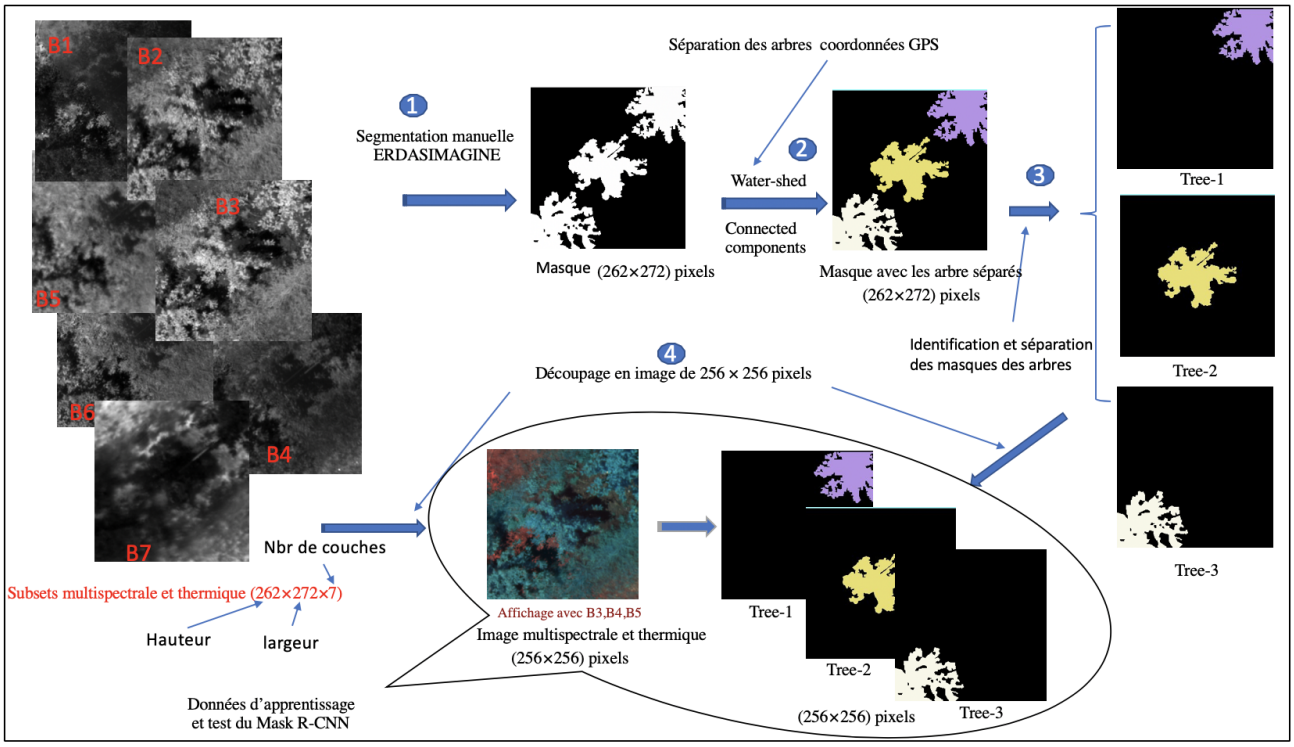


FIGURE 5.8 – Schéma de construction de données d'apprentissage et test du Mask R-CNN.

5.2.2 Normalisation et augmentation des données d'apprentissage

Pour l'apprentissage et test du **Mask R-CNN**, nous avons normalisé les images multi-spectrales et thermiques normalisées. Cette normalisation a été faite par image et par canal (voir, Section 4.3.3).

Pour multiplier la quantité de données d'apprentissage, nous avons utilisé la librairie Python Imgaug. En tout, nous avons généré 7 transformations : Par symétrie verticale, par symétrie horizontale, par trois rotations de 90° , 180° et 270° et par une multiplication des valeurs des images de 80% à 150%.

Comme pour la méthode **U-NET** dans le chapitre 4 de segmentation 4.1, nous avons utilisé sous python les librairies scikit-images et scikit-learn [40] pour les manipulation : construction des masques, découpage des subsets en images de 256 × 256 pixels, l'augmentation et normalisation de données.

5.3 Implémentation du MASK R-CNN

Pour l'implémentation du **Mask R-CNN**, nous avons utilisé la version de **Matterport** dans implémenté par Waleed Abdulla. Elle a été publiée en 2017 dans Github (Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow, url : `https://github.com/matterport/Mask_RCNN`).

Pour pouvoir appliquer la méthode sur notre jeu de données, nous avons changé la configuration des tailles des images d'entrée dans le backbone [2] (voir, Figure 5.10) du **Mask R-CNN**, car nous travaillons sur des images avec 7 canaux et cette méthode a été conçue pour des images RVB avec 3 canaux.

Les configurations effectuées sur le modèles sont :

— Backbone = ResNet101.

— MAX_GT_INSTANCES = 10 (le nombre maximum d'instance par image).

— DETECTION_MAX_INSTANCES = 10 (le nombre maximum d'instance à détecter par image).

— Epoch =200 (Nombre d'époque d'apprentissage du modèle).

— IMAGES_PER_GPU = 2 (Batchsize =2) (le nombre moyen d'images lues à la fois par étape d'apprentissage).

— Learning_rate =0.001 (Il est le paramètre définissant a quelle vitesse les poids des réseaux du **Mask R-CNN** seront modifiés).

— DETECTION_MIN_CONFIDENCE = 0.7 (le seuil de détection), etc.

Les ResNet101 (Residual Network) sont des réseaux de neurones artificiels basés sur les connexions indirectes entre couches. Kemaing He et al.2015 [51] ont montré l'intérêt et la facilité qu'apporte ce type de réseaux pour l'apprentissage profond. Ceux-là permettent de créer des raccourcis entre couches et d'éviter les couches ayant de difficultés à converger, ainsi que de réduire le risque de tomber dans le sur-apprentissage.

5.3.1 Apprentissage et test MASK R-CNN

L'apprentissage et test du modèle du **Mask R-CNN** ont été effectués sur des images multi-spectrales et thermiques de 256×256 pixels (voir, Section 5.2.1).

Nous avons pris 160 images($\approx$ 80% de données) pour l'apprentissage et 16 images pour le test du modèle. Nous avons fixé le nombre d'époques à 50.

Résultats d'apprentissage du MASK R-CNN

Pour évaluer l'apprentissage de la méthode **Mask R-CNN**, nous avons construit trois graphiques des valeurs loss du modèle : modèle (loss), masque (mrcnn-mask-loss) et de boites englobantes (val-mrcnn-bbox-loss). Ces valeurs sont obtenues pour chaque partie du réseaux(Voir, Section 5.10) par période de façons indépendante.

2. **Backbone** : il est le réseau de neurones artificiels utilisé pour l'extraction des cartes de caractéristiques des images d'entrée.

Dans les graphiques obtenus, nous avons remarqué que les valeurs de loss du modèle décroissent, puis elles convergent vers une valeur $\simeq 1.2$. Nous avons constaté aussi que les valeurs de loss du réseau de masques et de boites englobantes décroissent, mais elles ne convergent pas. Nous avons remarqué une divergence entre les résultats produits par modèle de chaque période, c-à-d un modèle entrainé sur un échantillon d'images d'apprentissage d'une période produit un résultat complètement différent sur un autre échantillon de données, donc le modèle subissait certainement un sur-apprentissage.

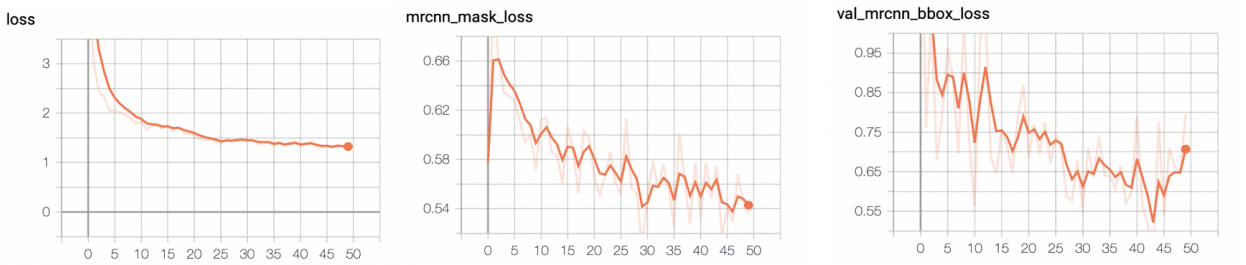


FIGURE 5.9 – Graphe de valeur loss d'apprentissage du Mask R-CNN.

5.3.2 Prédiction du MASK R-CNN

Nous avons effectué des prédictions sur 3 images multi-spectrales et thermiques de 256×256 pixels. Les résultats obtenus montrent que le modèle entrainé détecte bien les arbres, il les classifie et il leurs construit des boites englobantes avec des scores (des probabilités) assez élevés, proche de $\simeq 1$. Les masques prédits ne couvrent pas la totalité des pixels des feuilles des arbres, ils ne couvrent que l'intérieur des arbres. Cela peut être du à la quantité de données d'apprentissage.

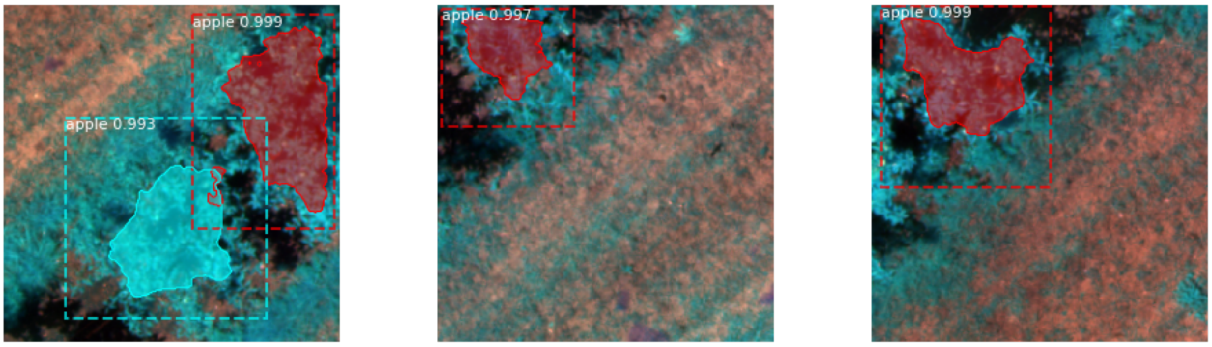


FIGURE 5.10 – Prédiction par Mask R-CNN.

Conclusion et perspective :

Dans les premiers résultats obtenus du **Mask R-CNN**, nous avons remarqué que la méthode ne converge pas. Le modèle ne semble pas fonctionner pour la segmentation. C'est peut être du à un manque de données. Pour augmenter la précision et améliorer les performances de modèles du **Mask R-CNN**, il serait nécessaire de rajouter plus de données d'apprentissage. Pour cela, il est possible d'utiliser la méthode **U-NET** (vue dans le cinquième chapitre4) pour la construction de données d'apprentissage.

Conclusion et perspectives

Les méthodes de traitement d'image a base de deep learning se sont montrées pertinentes pour la segmentation de nos images agronomiques. Dans une première partie de cette étude, nous avons utilisé la méthode **U-NET** pour la segmentation du feuillage d'une parcelle de pommiers et la méthode **WaterShed** pour l'identification des pommiers individuels a partir d'images multi-spectrales et thermiques et de coordonnées GPS. Des indices de végétations ont pu être dérivés de ces segmentation. La méthode **U-NET** s'est montrée très précise : 96% sur nos données. Cette approche permet l'automatisation du traitement d'un grand nombre de données (parcelle de 1000 arbres) par une approche fiable et reproductible et de réduire ainsi grandement le temps de travail manuel nécessaire à la caractérisation du matériel végétal.

Dans une deuxième partie de mon stage, nous avons commencé à travailler à la généralisation de cette approche, notamment pour des parcelles ou nous n'aurions pas accès aux coordonnées GPS de chaque arbre. Pour cela, nous avons étudié la méthode **Mask R-CNN** qui permet en plus de la segmentation du feuillage, l'identification des arbres de la parcelle. Malheureusement, nos premiers tests ont donné des résultats encourageants mais peu précis.

Cette étude a ouvert plusieurs perspectives et pistes de recherches intéressantes :

Bien qu'il existe une forte variabilité d'architecture de pommier dans le verger, nous avons testé notre algorithme sur un nombre limité de condition de végétation. Il serait intéressant de tester nos méthodes dans d'autres conditions de verger (par exemple : verger plus dense en terme de densité de plantation), dans d'autres date d'acquisition (vol de drone à différentes altitudes, avec d'autres résolutions, à d'autres stades de développement de l'arbre) ou pour d'autres variables (identification des fruits qui créera d'autres problèmes sans doute liée par exemple aux fortes variations de couleurs des fruits sur les parcelles). Éventuellement, on peut envisager tester aussi sur des données provenant de vergers avec d'autres espèces d'arbres fruitiers.

Meme si les résultats avec notre méthode basée sur **UNET** sont de bonnes qualité, différentes pistes existent pour l'améliorer et ou rendre plus robuste cette approche. En particulier, les données ont été fait par un petit nombre d'experts (2) avec des outils limités. Il serait intéressant de consolider ces données en faisant des validations croisées avec un panel d'experts i.e. la segmentation réalisée par un expert est reprise, corrigée et validée par d'autres experts. Idéalement une plus grande base d'exemples d'apprentissage serait pertient.

Notre approche est basée sur l'idée d'utiliser des données riches i.e. multispectrales et thermiques.

Pourtant il n'est pas clair ce qu'apporte ces données par rapport aux prises de données plus classique (RVB). Il serait intéressant de tester notre approche en considérant un nombre limité de canaux, parmi les 7 acquis, et tester la dégradation des résultats qui en résulte. Comme le montre les analyses des données préliminaires présentées dans ce manuscrit, certains canaux varient de manière très similaire et peuvent être partiellement redondants pour la segmentation. Déterminer un nombre minimal pertinent de canaux pour la segmentation permettrait d'optimiser la méthode et d'accelerer les temps de calcul.

Du a des contraintes techniques des réseaux et des machines actuelles, notre étude s'est concentrée sur des images de taille 96x96 et 256x256. Une première sensibilité de notre méthode basée sur **UNET** a pu être montrée. Pourtant d'autres tailles d'images sont imaginables et seraient intéressantes a tester. En particulier, il faudrait identifier le lien entre la taille de l'image et la capacité à détecter les objets considérés (feuillage, arbres).

Pour la méthode **Mask R-CNN**, différentes améliorations sont imaginables. La méthode est un assemblage complexe de plusieurs réseaux. Certains ne semblent pas complétement adaptés à nos données et peuvent être optimisés. En particulier, le test des boites englobantes peut être contraint en fonction des tailles des objets que l'on recherche. Le réseau utilisé pour réaliser la segmentation ne semble pas bien fonctionner. L'intégration d'un sous réseau de type **UNET** permettrait d'améliorer les résultats.

Comme perspective finale, dans le projet, des scans Lidar des arbres de la parcelle ont été réalisés en plus des images multi-spectrales et thermiques. Aligner et coupler les scans Lidar et l'imageries multi-spectrale et thermique permettrait de mieux spatialiser en 3D les indices de végétations et ainsi mieux étudier leur variabilité dans l'arbre.

En conclusion personnelle, ce stage m'a permis de découvrir un nouveau domaine, l'agriculture, que j'ai énormément apprécié. Il m'a également permis de mettre en application et de me familiariser avec certaine notions du deep learning. D'un point de vue technique, ce stage m'a permis également de consolider mes connaissances en programmation informatique.

Table des figures

Liste des tableaux

Bibliographie

[1] J-F. Soussana, coord. *S'adapter au changement climatique, Agriculture, écosystèmes et territoires.* Édition Quæ, RD10, 78026 Versailles Cedex.

[2] J. L. Monteith, *Climate and the efficiency of crop production in Britain*, Philos. Trans. R. Soc. Lond. B 281, 277–294 (1977). Url : https://doi.org/10.1098/rstb.1977.0140,(1977).

[3] T.B. Johnson, *Correlations of stomatal conductance with hydraulic, chemical and environmental variables in five urban tree species*, Tennessee Agricultural Experiment Station, University of Tennessee, Knoxville, TN 37901-1071, USA,Scientia Horticulturae , Url : http://www.esalq.usp.br/lepse/imgs/conteudo_thumb/,13 December 2000.

[4] N. Vigneau et al. *Imagerie aérienne par drone : Exploitation de données pour l'agriculture de précision*, Colloque scientifique francophone : Drones et moyens légers aéroportés d'observation, Montpellier, France, Url : https://hal.archives-ouvertes.fr/hal-01336243, 2014.

[5] J.A.J.Berni et al. *Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle.* IEEE Trans. Geosci. Remote Sens. 47, 722-738, Url : http://dx.doi.org/10.1109/TGRS.2008.2010457, 2009.

[6] N.Virlet et al. *Stress indicators based on airborne thermal imagery for field phenotyping a heterogeneous tree population for response to water constraints.* J. Exp. Bot. 65, 5429-5442. Url :http://dx.doi.org/10.1093/jxb/eru309, 2014.

[7] M. Delalande et al (submitted). *Do multispectral and thermal IR high-resolution UAS-borne imagery help in phenotyping the tree response to water stress at field ? Case studies in apple diversity population and varietal assays*, Mechanization, Precision Horticulture, and Robotics, 2nd Int. Congress ISHS, Istanbul (Turquie), 2018.

[8] Y.LeCun et al. *Deep Learning*, Nature 521, 436–444. Url : https://pdfs.semanticscholar.org/425f/c05d848f0318289f496f7b5e8cad26b123f6.pdf, 28 May 2015.

[9] A. Coupel-Ledru et al. *Multi-scale high-throughput phenotyping of architectural and functional traits in field/orchard conditions : application to the genotypic variability in an apple tree core-collection under contrasted watering regimes*, Horticulture Research. Url : https://doi.org/10.1038/s41438-019-0137-3, 23 January 2019.

[10] F.Coppens et al. *Unlocking the potential of plant phenotyping data through integration and data-driven approaches*,Current Opinion in Systems Biology 2017,58–63 . Url : http://dx.doi.org/10.1016/j.coisb.2017.07.002, July 2017.

[11] A.K.Singh et al. *Deep Learning for Plant Stress Phenotyping : Trends and Future Perspectives*, Trends in Plant Science, Vol. 23, No. 10, 883-898 . Url : https://doi.org/10.1016/j.tplants.2018.07.004, October 2018.

[12] A. Gitelson et al. *Use of a green channel in remote sensing of global vegetation from EOS- MODIS. Remote Sensing of Environment*, 58(3), 289-298. Url : https://doi.org/10.1016/S0034-4257(96)00072-7, 1996.

[13] L.Lassois et al. *Genetic diversity, population structure, parentage analysis, and construction of core collections in the french apple germplasm based on SSR Markers*, Plant Mol. Biol. Report. 34, 827–844. Url : https://orbi.uliege.be/bitstream/2268/191546/, 2016.

[14] C.Touzet. *LES RÉSEAUX DE NEURONES ARTIFICIELS, INTRODUCTION AU CONNEXIONNISME : COURS, EXERCICES ET TRAVAUX PRATIQUES.* EC2, Collection de l'EERIE, N. Giambiasi. hal-01338010, 1992.

[15] P.Borne, M.Nenrejeb, J.Haggère. *LES RESEAUX DE NEURONES, Présentation et application.* Edition TECHNIP, Paris. ISBN : 978-2-7108-0896-1, ISSN : 1152-0647. URL :www.books.google.fr, 2007.

[16] O.Ezratty. *Les usages de l'intelligence artificielle.* Edition 2018, page : 5-521, Url : http://creativecommons.org/licenses/by-nc-nd/2.0/fr/, Novembre 2018.

[17] A. Garcia et al. *A survey on deep learning techniques for image and video semantic segmentation*, 3D Perception Lab, University of Alicante, Spain, Applied Soft Computing 70 (2018) 41–65. Url : www.elsevier.com/locate/asoc, May 2018.

[18] P. Benedetti et al (submitted). *M3Fusion : A Deep Learning Architecture for Multi-Scale/Modal/Temporal satellite data fusion*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Volume : 11, pages : 4939-4949, Dec 2018.

[19] S.Indolia et al. *Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach*, ScienceDirect, Procedia Computer Science 132 (2018), 679–688. Url : https://reader.elsevier.com/reader/sd/pii/, 2018.

[20] A.R.Pathak et al. *Application of Deep Learning for Object Detection*, ScienceDirect, Procedia Computer Science 132 (2018), 1706–1717. Url : https://creativecommons.org/licenses/by-nc-nd/3.0/, 2018.

[21] S.Indolia et al. *Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach*, rocedia Computer Science 132, 679–688. Url : https://creativecommons.org/licenses/by-nc-nd/3.0/, 2018.

[22] K.Fukushima et al. *Neocognitron : A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*, NHK BroadcastingScienceResearchLaboratories,Kinuta, Setagaya,Tokyo,Japan, Biol. Cybernetics36, 193–202. Url : https://www.rctn.org/bruno/public/papers/Fukushima1980.pdf, 1980.

[23] M.Pound et al.*Deep machine learning provides state-of-the-art performance in image-based plant phenotyping*, School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK, GigaScience, 6 1–10. Url : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5632296/, August 2017.

[24] S.Ghosal et al. *An explainable deep machine vision framework for plant stress phenotyping*, Department of Energy, Plant Research Laboratory, Michigan State University, East Lansing, MI, PNAS vol. 115 18 4613–4618 . Url : www.pnas.org/cgi/doi/10.1073/pnas.1716999115, March 2018.

[25] Z.Khan et al. *Estimation of vegetation indices for high-throughput phenotyping of wheat using aerial imaging*, Phenomics and Bioinformatics Research Center, University of South Australia, Plant Methods volume 14, Article number : 20 (2018), ISSN : 1746-4811 . Url : https://doi.org/10.1186/s13007-018-0287-6, March 2018.

[26] J.Ubbens et al. *Deep Plant Phenomics : A Deep Learning Platform for Complex Plant Phenotyping Tasks*, University of Saskatchewan, 105 Administration Place, Saskatoon S7N 5C5, Canada, Frontiers in Plant Science volume 8, Article number : 1190 (2017), ISSN : 1746-4811 . Url : https://doi.org/10.3389/fpls.2017.01190, July 2017.

[27] J.Ubbens et al. *The use of plant models in deep learning : an application to leaf counting in rosette plants*, University of Saskatchewan, 105 Administration Place, Saskatoon S7N 5C5, Canada, Plant Methods volume 14, Article number : 6 (2018), ISSN : 1746-4811 . Url : https://doi.org/10.1186/s13007-018-0273-z, January 2018.

[28] A.Dore et al. *Détection d'objets en milieu naturel : application à l'arboriculture*, 16 ième Jourées Francophones des Jeunes Chercheurs en Vision par Ordinateur (ORASIS 2017), GREYC, Jun 2017, Colleville-sur-Mer, France. 8p. hal-01866622. Url : https://hal.archives-ouvertes.fr/hal-01866622, Submitted on Sep 2018.

[29] Y.LECUN et al. *Handwritten digit recognition with a back-propagation network.* In : Advances in neural information processing systems, p. 396-404. Url : http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf, 1990.

[30] Y.LECUN et al. recognition with gradient-based learning. In Shape, contour and grouping in computer vision (pp. 319-345). Springer, Berlin, Heidelberg. Url : https://link.springer.com/chapter/10.1007/3-540-46805-6_19, (1999).

[31] D.-A. Clevert et al. *Fast and accurate deep network learning by exponential linear units (elus).*Institute of Bioinformatics Johannes Kepler University, Linz, Austria. Url : https://arxiv.org/pdf/1511.07289.pdf, 2015.

[32] O. Ronneberger et al. *U-Net : Convolutional Networks for Biomedical Image Segmentation*, Computer Science Department and BIOSS Centre for Biological Signalling Studies,University of Freiburg, Germany. Url : arXiv:1505.04597v1[cs.CV], May 2015.

[33] Y.Gal et al.*Dropout as a bayesian approximation : Representing model uncertainty in deep learning. In international conference on machine learning*, University of Cambridge, pp. 1050-1059. Url : http://proceedings.mlr.press/v48/gal16.pdf, June 2016.

[34] N. Srivastava et al. *Dropout : A simple way to prevent neural networks from overfitting*, The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958. Url : http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf, 2014.

[35] M.Abadi et al. *Tensorflow : A system for large-scale machine learning.* In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (pp. 265-283). Url :https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi, 2016.

[36] A.Gulli et al. *Deep Learning with Keras.* Packt Publishing Ltd. Url : https://books.google.fr/books?hl=fr&lr=&id=20EwDwAAQBAJ&oi=fnd&pg=PP1&dq=keras&ots=lHeCco7PS7&sig=E06iT0dfASl4_XS8o2mCD0sBOds&redir_esc=y#v=onepage&q=keras&f=false , 2017.

[37] A.Cornuéjols . *Une nouvelle méthode d'apprentissage : Les SVM. Séparateurs à vaste marge.* Bulletin de l'AFIA, 51, 14-23. Url :http://www2.agroparistech.fr/ufr-info/membres/cornuejols/Papers/PUBLIES/SVM-synthese.pdf, 2002.

[38] G.Lebrun et al. *Sélection de modèles pour la classification supervisée avec des SVM (Séparateurs à Vaste Marge). Application en traitement et analyse d'images*, Traitement des images [eess.IV]. Université de Caen Basse-Normandie. Url : https://tel.archives-ouvertes.fr/tel-01282893, Submitted on 7 Mar 2016.

[39] T.Chi Zhang et al. *SVM Methods in Image Segmentation*, COLLA 2016 : The Sixth International Conference on Advanced Collaborative Networks, Systems and ApplicationsISBN : 978-1-61208-517-3 62-65. Url : https://www.thinkmind.org/download.php?articleid=colla_2016_4_10_50022, 2016.

[40] F.Pedregosa et al. *Scikit-learn : Machine learning in Python. Journal of machine learning research*, vol. 12, no Oct. Url :http://www.jmlr.org/papers/v12/pedregosa11a. 12(Oct) 2011, 2825-2830.

[41] S.Beucher. *The Morphological Approach to Segmentation : The Watershed Transformation*, Mathematical Morphology in Image Processing, Editors : Dougherty, E.R., pp.433–481. Url : https://www.researchgate.net/publication/230837870_The_Morphological_Approach_to_Segmentation_The_Watershed_Transformation, January 1993.

[42] R.Girshick. *Rich feature hierarchies for accurate object detection and semantic segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Url :doi:10.1109/CVPR.2014.81, 2014.

[43] J. R. R.Uijlings et al. *Selective Search for Object Recognition.* International Journal of Computer Vision, 104(2), 154–171. doi :https://link.springer.com/article/10.1007/s11263-013-0620-5, 2013.

[44] R.Girschick. *Fast R-CNN.* Proceedings of the IEEE International Conference on Computer Vision : 1440–1448. Url : arXiv:1504.0808, 2015.

[45] R.Girshick. *Faster R-CNN.* The IEEE International Conference on Computer Vision (ICCV), pp. 1440-1448. Url : http://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html, 2015.

[46] Kaiming He et al. *Mask R-CNN.*The IEEE International Conference on Computer Vision (ICCV), pp.2961-2969 . Url : http://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html, 2017.

[47] K.R.Malhotra et al. *Autonomous Detection of Disruptions in the Intensive Care Unit Using Deep Mask R-CNN.* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 1863-1865. Url : http://openaccess.thecvf.com/content_cvpr_2018_workshops/w36/html/Malhotra_Autonomous_Detection_of_CVPR_2018_paper.html, 2018.

[48] S.Nie, Zhiguo et al. *Inshore Ship Detection Based on Mask R-CNN*, Published in IGARSS - IEEE International Geoscience and Remote Sensing Symposium. DOI : 10.1109/IGARSS.2018.8519123, 2018.

[49] S.Bargoti et al. *Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards*. Special Issue : Agricultural Robotics, volume34, issue6 , pp. 1039-1060. Url :https://doi.org/10.1002/rob.21699 ,septembre 2017.

[50] G.Bernotas et al. *Photometric stereo data for leaf segmentation and plant phenotyping : a collation study" Machine vision and applications*. Url :https://www.plant-phenotyping.org/lw_resource/datapool/systemfiles/elements/files/e8a1faf0-949b-11e8-8a88-dead53a91d31/current/document/0018.pdf ,2018.

[51] Kaiming He et al. *Deep Residual Learning for Image Recognition*. Computer Vision and Pattern Recognition. Url : https://arxiv.org/abs/1512.03385, Dec 2015.

Annexes

5.4 Tableau des utilisations des mesures de réflectances pour indices de végétation :

Bandes spectrales	Valeur centrale (nm)	Utilisation
Bleu (Blue)	450	Glaucescence
Vert (PRI1)	530	MCARI, MCARI2, PRI,...
Vert (PRI2)	570	GNDVI, PRI,...
Rouge (RED)	675	MCARI, NDVI,...
Red-edge (RE)	730	MCARI,...
Proche infra-rouge (NIR)	850	chlorophylle...
Infra-Rouge Thermique (TIR)	850	(Ts-Ta), CWSI, WDI du stresse hydrique

TABLE 5.1 – Tableau des utilisations des mesures de réflectances pour indices de végétation.

5.5 Tableau descriptif des valeurs de réflectance et température de plusieurs subset :

Subsets	Date et heure d'acquisition	Moyenne, Ecart type						
		B1	B2	B3	B4	B5	B6	B7
subset 1	12/07/2017 à 12h00	5.85, 2.87	1.63, 0.47	2.74, 1.00	1.49, 0.79	1.74, 0.60	1.25, 0.43	38.76, 6.38
subset 2	12/07/2017 à 12h00	5.71, 2.60	1.42, 0.59	2.66, 0.90	1.44, 0.71	1.70, 0.58	1.25, 0.43	38.42, 5.88
subset 3	12/07/2017 à 12h00	5.36, 2.84	1.51, 0.68	2.76, 1.03	1.40, 0.82	1.78, 0.64	1.33, 0.45	37.18, 5.65
subset 4	12/07/2017 à 12h00	6.67, 2.80	1.64, 0.61	2.97, 0.94	1.80, 0.82	1.67, 0.51	1.23, 0.37	39.71, 6.59
subset 5	12/07/2017 à 12h00	6.62, 2.94	1.72, 0.63	3.11, 0.99	1.80, 0.85	1.77, 0.53	1.30, 0.38	40.48, 7.44
subset 6	12/07/2017 à 12h00	5.85, 2.87	1.47, 0.63	2.47, 1.00	1.49, 0.79	1.74, 0.60	1.25, 0.43	38.76, 6.38
subset 7	12/07/2017 à 12h00	5.71, 2.60	1.42, 0.59	2.66, 0.90	1.44, 0.71	1.70, 0.58	1.25, 0.43	38.42, 5.88
subset 8	12/07/2017 à 12h00	5.51, 2.42	1.40, 0.54	2.68, 0.89	1.38, 0.68	1.77, 0.56	1.37, 0.45	39.57, 7.22
subset 9	12/07/2017 à 12h00	6.08, 2.50	1.53, 0.55	2.94, 0.92	1.55, 0.69	1.81, 0.48	1.38, 0.38	41.56, 7.72
subset 10	12/07/2017 à 12h00	5.36, 2.84	1.51, 0.68	2.76, 1.03	1.40, 0.82	1.78, 0.64	1.33, 0.45	37.18, 5.65
subset 10	12/07/2017 à 12h00	6.32, 2.92	1.67, 0.64	3.02, 0.96	1.69, 0.85	1.80, 0.55	1.32, 0.40	39.57, 6.91
subset 12	12/07/2017 à 12h00	6.67, 2.80	1.64, 0.61	2.97, 0.94	1.80, 0.82	1.67, 0.51	1.23, 0.37	39.71, 6.59
subset 13	12/07/2017 à 12h00	6.72, 2.94	1.72, 0.63	3.11, 0.99	1.81, 0.85	1.77, 0.53	1.30, 0.38	40.48, 7.44
subset 14	12/07/2017 à 12h00	5.53, 2.64	1.38, 0.59	2.45, 0.90	1.50, 0.93	1.68, 0.62	1.31, 0.47	35.64, 5.31
subset 15	12/07/2017 à 12h00	7.10, 3.17	1.65, 0.62	2.99, 0.99	1.95, 0.98	1.77, 0.52	1.32, 0.39	40.99, 8.11
subset 16	12/07/2017 à 12h00	7.33, 3.00	1.68, 0.58	3.06, 0.93	2.08, 0.95	1.77, 0.49	1.32, 0.37	41.06, 7.67
subset 16	12/07/2017 à 12h00	7.53, 2.99	1.76, 0.61	3.17, 0.97	2.12, 0.91	1.77, 0.49	1.31, 0.36	41.88, 7.60
subset 18	05/07/2017 à 10h00	0.64, 0.39	0.55, 0.23	0.45, 0.16	0.53, 0.36	1.31, 0.47	1.26, 0.49	29.77, 4.93
subset 19	05/07/2017 à 10h00	0.96, 0.59	0.72, 0.36	0.61, 0.26	0.92, 0.63	1.13, 0.47	1.26, 0.46	31.38, 6.76
subset 20	05/07/2017 à 10h00	0.94, 0.60	0.72, 0.38	0.63, 0.27	0.88, 0.62	1.13, 0.50	1.15, 0.48	30.35, 6.30
subset 21	05/07/2017 à 10h00	1.06, 0.71	0.80, 0.41	0.73, 0.31	0.97, 0.71	1.16, 0.45	1.23, 0.47	30.15, 5.67
subset 22	05/07/2017 à 10h00	0.99, 0.54	0.82, 0.36	0.81, 0.28	0.89, 0.58	1.21, 0.44	1.34, 0.48	29.68, 5.06
subset 23	05/07/2017 à 10h00	1.03, 0.56	0.81, 0.37	0.82, 0.28	0.92, 0.60	1.23, 0.43	1.35, 0.48	29.69, 4.93
subset 23	05/07/2017 à 10h00	1.14, 0.60	0.85, 0.35	0.85, 0.29	1.01, 0.66	1.23, 0.41	1.34, 0.46	31.07, 6.02
subset 25	05/07/2017 à 10h00	0.99, 0.51	0.72, 0.33	0.79, 0.31	0.80, 0.53	1.12, 0.43	1.26, 0.45	30.59, 6.56
subset 26	05/07/2017 à 10h00	1.03, 0.47	0.78, 0.29	0.83, 0.28	0.84, 0.51	1.30, 0.43	1.33, 0.44	30.87, 6.56
subset 27	05/07/2017 à 10h00	0.89, 0.48	0.70, 0.30	0.75, 0.30	0.75, 0.52	1.24, 0.45	1.27, 0.47	29.40, 5.89
subset 28	05/07/2017 à 10h00	0.54, 0.25	0.59, 0.21	0.33, 0.11	0.51, 0.25	1.03, 0.38	1.03, 0.39	30.12, 5.51
subset 29	05/07/2017 à 10h00	0.85, 0.38	0.80, 0.30	0.43, 0.15	0.88, 0.52	1.05, 0.38	1.09, 0.39	28.58, 6.61
subset 30	05/07/2017 à 10h00	1.18, 0.73	1.08, 0.50	0.78, 0.35	1.43, 1.13	1.14, 0.40	1.11 , 0.38	31.26, 7.66
subset 31	05/07/2017 à 10h00	1.81, 0.72	1.43, 0.37	1.10, 0.27	2.36, 1.15	1.15, 0.21	1.08, 0.23	30.62, 7.81
subset 32	05/07/2017 à 10h00	0.59, 0.28	0.85, 0.32	1.41, 0.50	1.29, 0.77	1.29, 0.38	1.13, 0.37	34.06, 7.28
subset 33	05/07/2017 à 10h00	0.45, 0.27	0.71, 0.32	1.32, 0.52	0.99, 0.71	1.12, 0.40	1.16, 0.37	29.79, 7.62
subset 34	05/07/2017 à 10h00	0.47, 0.22	0.47, 0.26	1.43, 0.49	1.06, 0.68	1.17, 0.39	1.13, 0.38	29.64, 7.22
subset 35	05/07/2017 à 10h00	0.67, 0.31	0.87, 0.32	1.70, 0.60	1.66, 0.90	1.30, 0.30	1.02, 0.27	36.23, 8.55
subset 36	05/07/2017 à 10h00	0.50, 0.24	0.73, 0.27	1.38, 0.49	1.12, 0.72	1.16, 0.35	1.08, 0.35	31.40, 8.30
subset 37	05/07/2017 à 10h00	1.01, 0.63	0.87, 0.40	0.74, 0.33	1.14, 0.83	1.08, 0.39	1.10, 0.39	32.19, 8.01
subset 38	05/07/2017 à 10h00	1.10, 0.63	0.96, 0.40	0.84, 0.33	1.23, 0.85	1.16, 0.39	1.14, 0.39	32.60, 7.04
subset 39	05/07/2017 à 10h00	1.25, 0.62	0.99, 0.38	0.91, 0.34	1.52, 0.91	1.09, 0.32	1.06, 0.31	36.15, 8.41
subset 40	05/07/2017 à 10h00	1.21, 0.61	1.02, 0.37	0.87, 0.31	1.52, 0.90	1.20, 0.31	1.07, 0.30	34.96, 7.31
subset 41	05/07/2017 à 10h00	1.22, 0.63	0.99, 0.37	0.87, 0.32	1.51, 0.91	1.16, 0.33	1.08, 0.32	35.31, 8.06
subset 42	05/07/2017 à 10h00	1.19, 0.61	1.00, 0.37	0.86, 0.31	1.49, 0.89	1.20, 0.32	1.07, 0.31	35.14, 7.63
subset 43	05/07/2017 à 10h00	1.24, 0.63	0.99, 0.37	0.87, 0.32	1.52, 0.93	1.17, 0.33	1.07, 0.32	35.34, 8.01
subset 44	05/07/2017 à 10h00	1.25, 0.61	1.01, 0.38	0.92, 0.34	1.53, 0.90	1.09, 0.33	1.07, 0.30	35.23, 7.88
subset 45	05/07/2017 à 10h00	1.02, 0.68	0.95, 0.52	0.73, 0.37	1.22, 1.04	1.04, 0.43	1.09, 0.43	29.55, 6.71
Total	Date et heure d'acquisition	Moyenne, Ecart type, Coefficient de variabilité						
Moyenne	12/07/2017 à 12h00	6.2,2.8,0.4	1.5, 0.6, 0.3	2.8,0.9,0.3	1.6,0.8,0.4	1.7,0.5,0.3	1.2,0.4,0.3	39.3,6.7,0.1
Ecart type	12/07/2017 à 12h00	0.69, 0.19	0.22, 0.05	0.22 , 0.04	0.24, 0.08	0.04, 0.05	0.04, 0.03	1.68, 0.86
Moyenne	05/07/2017 à 10h00	0.9,0.5,0.5	0.8,0.3,0.4	0.8,0.3,0.3	1.1,0.7,0.6	1.1,0.3,0.3	1.1, 0.3, 0.3	31.6, 6.9, 0.2
Ecart type	05/07/2017 à 10h00	0.30, 0.16	0.19, 0.06	0.32, 0.11	0.39, 0.22	0.07, 0.06	0.10, 0.07	2.36, 1.08

TABLE 5.2 – Tableau descriptif des variables (B$i, i = 1..7$) des subsets.

5.6 Schéma des pré-traitements, extraction et analyse des images

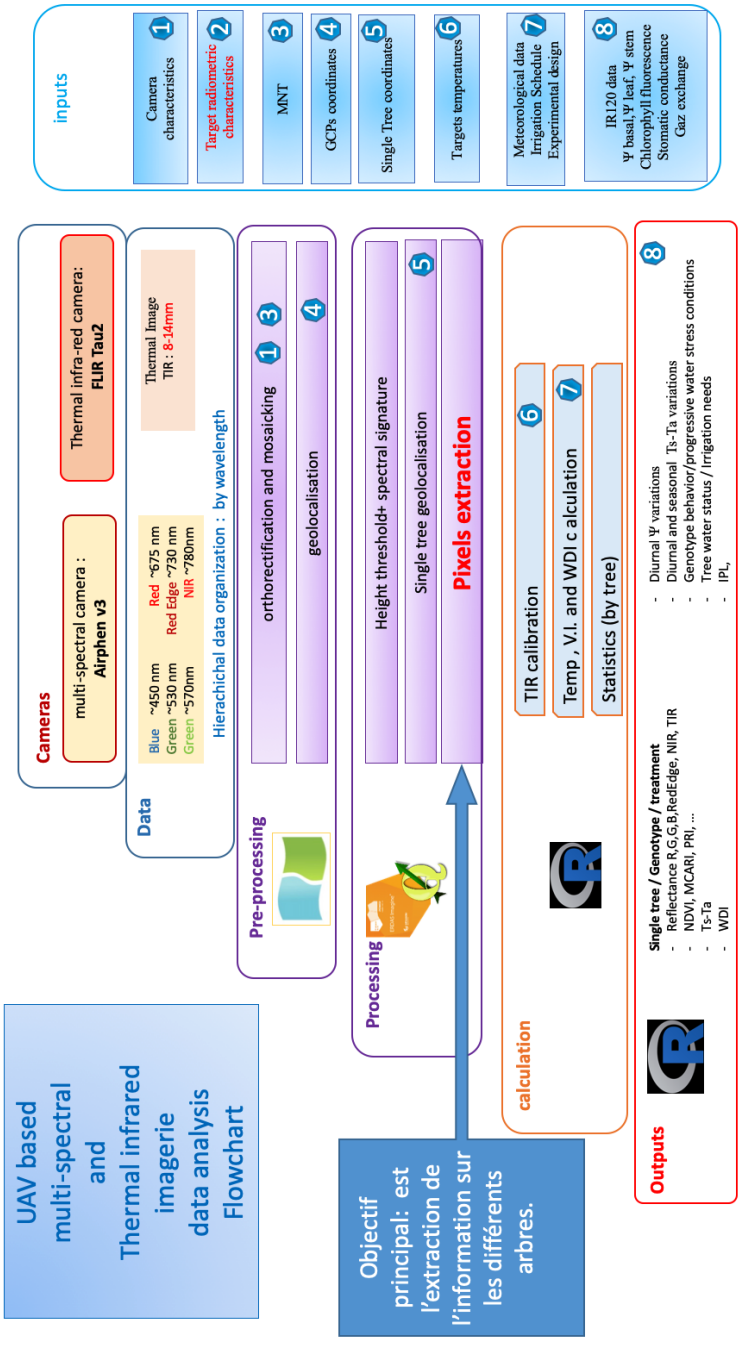


FIGURE 5.11 – Schéma des pre-traitements, extraction et analyse des images.
Source : M. Delalande...

0.7 Image ortho-mosaïque de la parcelle

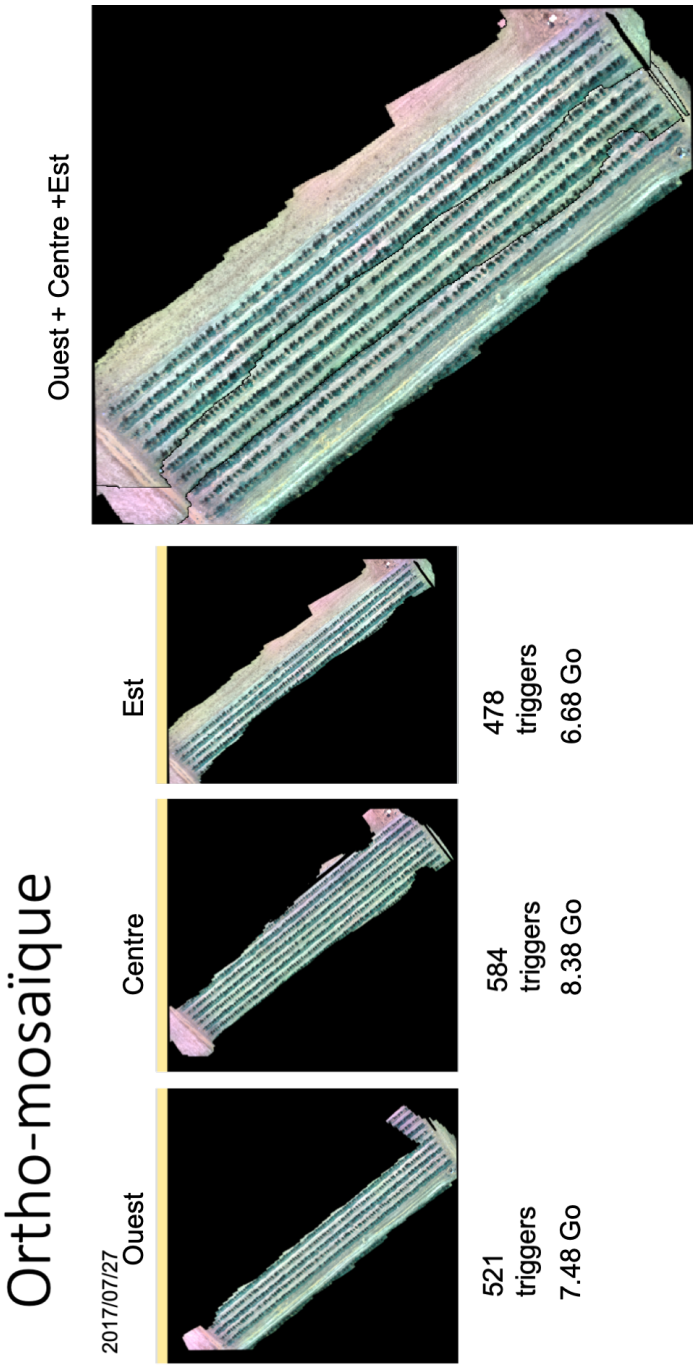


FIGURE 12 – Otho-mosaiques (Est, Centre, Ouest) de la parcelle.
Source : M. Delalande...

1. **Triggers** : un ensemble d'images prises en simultané, dans différentes longueurs d'onde. Ici un trigger est composé de 7 images : 1 par longueur d'onde.

0.8 Tableau de comparaison U-NET/SVM

Nom des subset	Précision U-NET (mP)				Précision SVM (%)
	96 ×96/1968/184 /3 augmentations. (%)	96 ×96/1968/184/ sans augmentation. (%)	256 ×256/135/7/ 3 augmentations. (%)	256 ×256/135/7/ sans augmentation. (%)	
Subset 1	96.3	95.2	90.6	90.9	83.3
Subset 2	97.7	97.1	96.3	91.2	85.9
Subset 3	91.1	87.3	79.4	79.1	81.1
Subset 4	96.1	96.5	95.5	83.82	77.5
Subset 5	92.2	89.8	92.7	89.3	82.9
Subset 6	96.5	95.1	94.9	93.6	93.9
Subset 7	96.2	95.4	95.9	94.2	65.6
Totaux	95.2	93.8	92.2	88.9	81.4

TABLE 3 – Tableau de comparaison U-NET/SVM.

0.9 Tableau des simulations de U-NET

N	Taille de données d'apprentissage	Tailles données de tes	Nbr d'augmentation	Date Acquisition apprentissage	Date acquisition test	Type de normalisation	Nbr canaux	Canaux	Batch -size	Seuil	Accuracy	Mp image de test	Mp Subset 33-5	Temps d'apprentissage
1	45/38/7/3148/476	96*96	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image by column	7	tous	5	0.5	Train :95% Test : 96%	96.37%	91.40%	4855.93 s
2	45/38/7/3148/476	96*96	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By column	7	tous	5	0.5	Train :94% Test : 90%	77.26%	52.51%	4476.66 s
3	45/38/7/3148/476	96*96	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image	7	tous	5	0.5	Train :96% Test : 95%	96.02%	80.01%	4452.75 s
4	45/38/7/3148/476	96*96	0	05/07/2017 12/07/2017	05/07/2017 12/07/2017	Byimage by column	7	tous	5	0.5	Train :97.5% Test : 96.5%	95.19%%	90.97%%	3301.50 s
5	45/38/7/3148/476	96*96	0	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By column	7	tous	5	0.5	Train :91.5% Test : 89%	77.10%	52.10%	3115.56 s
6	45/38/7/3148/476	256*256	0	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image	7	tous	5	0.5	Train :97.5% Test : 96%	96.02%	91.40%	3255.86 s
7	45/38/7/3148/476	256*256	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	Byimage by column	7	tous	5	0.5	Train :98% Test : 97%	96.76%	91.47%	300,86 s
8	45/38/7/3148/476	256*256	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By column	7	tous	5	0.5	Train :91% Test : 80%	60.30%	80.61%	151.30 s
9	45/38/7/3148/476	256*256	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image	7	tous	5	0.5	Train :97% Test : 96%	96.80%	80.66%	257.24 s
10	45/38/7/3148/476	256*256	0	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image by column	7	tous	5	0.5	Train :95% Test : 94%	97.69%	94.90%	142.61 s
11	45/38/7/3148/476	256*256	0	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By column	7	tous	5	0.5	Train :91% Test : 84%	89.37	72.54	74.82 s
12	45/38/7/151 /36	256*256	0	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By column	7	Tous	2	0.5	Train :91% Test : 84%	89.37%	72.54%	74.82 s
13	45/38/7/151 /36	256*256	0	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image	7	Tous	2	0.5	Train :97% Test : 96.8%	96.31%	80.74%	175.62 s
14	21/17/4/382/92	96*96	3	05/07/2017	05/07/2017	Byimage by column	7	Tous	1	0.5	Train :95.5% Test : 95%	96.29%	89.96%	791.87 s
15	21/17/4/382/92	96*96	3	05/07/2017	05/07/2017	By column	7	Tous	1	0.5	Train :91.5% Test : 84%	62.86%	80.59%	285.55 s
16	21/17/4/382/92	96*96	3	05/07/2017	05/07/2017	By image	7	Tous	1	0.5	Train :95% Test : 95%	95.19%	80.63%	860.93 s
17	21/17/4/382/92	96*96	0	05/07/2017	05/07/2017	By image by column	7	Tous	1	0.5	Train 94% Test : 93%	91.73%	88.76%	225.15 s
18	21/17/4/382/92	96*96	0	05/07/2017	05/07/2017	By column	7	Tous	1	0.5	Train :92% Test : 87%	87.00%	79.25	377.93 s
19	21/17/4/382/92	96*96	0	05/07/2017	05/07/2017	By image	7	Tous	1	0.5	Train 94% Test : 93%	94.01%	80.62%	548.03 s
20	45/34/11/2764/1720	96*96	0	05/07/2017	12/07/2017	By image by column	7	Tous	5	0.5	Train : 96% Test : 89%	93.03%	89.57%	463.664 s
21	45/34/11/2764/1720	96*96	0	05/07/2017	12/07/2017	By column	7	Tous	5	0.2	Train 95% Test : 87.5%	89.20%	85.07%	466.96 s
22	45/34/11/2764/1720	96*96	0	05/07/2017	12/07/2017	By image	7	Tous	5	0.2	Train 96% Test : 87.5%	90.00%	88.76%	331.46 s
23	45/34/11/2764/1720	96*96	3	05/07/2017	12/07/2017	Byimage by column	7	Tous	5	0.1	Train : 95% Test : 87%	82.45%	91.91%	588.35 s
24	45/34/11/2764/1720	96*96	3	05/07/2017	12/07/2017	By column	7	Tous	5	0.1	Train : 94% Test : 84%	80.00%	76.00%	565.21 s
25	45/34/11/2764/1720	96*96	3	05/07/2017	12/07/2017	By image	7	Tous	5	0.1	Train : 95% Test : 85%	85.83%	82.00%	487.34 s
26	45/34/11/2764/1720	256*256	3	05/07/2017	12/07/2017	By image by column	7	Tous	1	0.1	Train : 96% Test : 90%	92.36%	92.14%	76.06 s
27	45/34/11/2764/1720	256*256	3	05/07/2017	12/07/2017	By column	7	Tous	1	0.1	Train : 94% Test : 85%	87.60%	76.13%	87.17 s
28	45/34/11/2764/1720	256*256	3	05/07/2017	12/07/2017	By image	7	Tous	1	0.01	Train : 95% Test : 87%	84.45 %	91.57%	66.18 s
29	45/34/11/2764/1720	256*256	0	05/07/2017	12/07/2017	By image by column	7	Tous	1	0.5	Train : 91% Test : 88%	91.62%	88.93%	86.31 s
30	45/34/11/2764/1720	256*256	0	05/07/2017	12/07/2017	By column	7	Tous	1	0.5	Train : 90% Test : 85%	87.60%	87.63%	79.28 s
31	45/11/34/1720/2764	96*96	3	12/07/2017	05/07/2017	By image by column	7	Tous	5	0.5	Train : 95% Test : 92%	89.29%	87.00%	445.34 s
32	45/11/34/1720/2764	96*96	3	12/07/2017	05/07/2017	By column	7	Tous	5	0.5	Train : 95% Test : 82%	85.22%	76.46%	422.24 s
33	45/11/34/1720/2764	96*96	3	12/07/2017	05/07/2017	By image	7	Tous	5	0.5	Train : 95% Test : 90%	90.93%	82.20%	390.10 s
34	45/11/34/42/84	256*256	3	12/07/2017	05/07/2017	By image by column	7	Tous	2	0.5	Train : 95% Test : 93%	92.26%	87.74%	77.02 s
35	45/11/34/42/84	256*256	3	12/07/2017	05/07/2017	By column	7	Tous	2	0.5	Train : 95% Test : 82%	80.49%	80.49%	80.00 s
36	45/11/34/42/84	256*256	3	12/07/2017	05/07/2017	By image	7	Tous	2	0.5	Train : 93% Test : 86%	85.46%	83.13%	76.01 s
37	45/34/11/2764/1720	256*256	0	05/07/2017	12/07/2017	By image	7	Tous	1	0.1	Train : 91% Test : 84%	87.60%	80.61%	75.30 s
38	45/38/7/2722/852	96*96	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image by column	5	1,2,3,4,5	2	0.5	Train : 96% Test : 95%	95.28%	80.22%	4536.45 s
39	45/38/7/2722/852	96*96	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By column	5	1,2,3,4,5	2	0.5	Train : 95% Test : 90%	91.28%	78.30%	4550.37 s
40	45/38/7/2722/852	96*96	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image	5	1,2,3,4,5	2	0.2	Train : 96% Test : 96%	96.55%	75.61%	4563.01 s
41	45/38/7/125/62	256*256	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image by column	5	1,2,3,4,5	1	0.5	Train : 96% Test : 95.5%	95.83%	80.58%	497,45 s
42	45/38/7/125/62	256*256	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By column	5	1,2,3,4,5	1	0.5	Train : 95% Test : 94%	95.09%	67.27%	559 ,80 s
43	45/38/7/125/62	256*256	3	05/07/2017 12/07/2017	05/07/2017 12/07/2017	By image	5	1,2,3,4,5	1	0.5	Train : 95,5% Test : 94%	94.53%	74.66%	516,67 s

TABLE 4 – Tableau des simulations de **U-NET**.

0.10 Tableau des indices de végétation

Paramètres et identifiants d'arbres			Valeurs des feuilles extraites avec ERDAS					Valeurs extraites avec U-NET et Water-shed				Valeurs extraites avec la segmentation manuelle			
Id	clone	Ta	dalle	mean_Ts	mean_NDVI	mean_PRI	mean_GNDVI	Nbr_pixels_leafs	mean_Ts	NDVI	GNDVI	PIR	Nbr_pixels_leafs	mean_Ts	NDVI	GNDVI	PIR	Nbr_pixels_leafs
199	X6468	22,41	C	35,49	0,49	-0,21	0,09	8060	25,40	0,60	0,68	-0,27	9476	25,32	0,62	0,69	-0,27	9190
200	X6471	22,41	C	33,81	0,32	-0,17	0,01	10755	24,97	0,45	0,59	-0,30	6928	23,64	0,42	0,62	-0,30	4100
201	X6518	22,41	C	34,03	0,31	-0,18	-0,05	13017	24,16	0,40	0,56	-0,29	10062	24,23	0,44	0,57	-0,29	10882
202	X6905	22,41	C	34,38	0,12	-0,17	-0,14	11818	24,00	0,21	0,50	-0,29	1954	23,23	0,25	0,53	-0,29	4966
403	X8934	22,41	C	32,17	0,22	-0,38	-0,22	11757	25,18	0,30	0,39	-0,17	9536	25,18	0,30	0,39	-0,17	9536
404	X8937	22,41	C	33,16	0,18	-0,38	-0,24	12341	25,83	0,28	0,38	-0,15	5203	25,83	0,28	0,38	-0,15	5203
405	X8939	22,41	C	34,05	0,24	-0,36	-0,20	11760	26,00	0,20	0,33	-0,13	5817	25,75	0,28	0,39	-0,14	4866
406	X8972	22,41	C	33,74	0,36	-0,34	-0,16	12799	25,55	0,31	0,36	-0,11	26850	25,21	0,38	0,40	-0,12	21211
407	X9080	22,41	C	32,16	0,33	-0,35	-0,17	12891	24,36	0,30	0,38	-0,10	1724	24,44	0,38	0,44	-0,12	1433
459	X9148	22,41	C	32,35	0,28	-0,31	-0,17	10873	32,77	0,35	-0,13	0,29	2224	33,31	0,28	-0,18	0,30	8134
463	X8380	22,41	C	31,85	0,32	-0,33	-0,12	9859	31,72	0,23	-0,23	0,37	410	31,43	0,22	-0,23	0,37	630
708	X0700	22,41	E	24,63	0,28	-0,28	0,05	4111	26,37	0,26	0,36	-0,08	404	24,82	0,40	0,40	-0,11	356
709	X9153	22,41	E	23,97	0,35	-0,27	0,10	4568	25,84	0,39	0,41	-0,09	3757	25,93	0,41	0,42	-0,10	3650
710	X4915	22,41	E	22,01	0,50	-0,27	0,16	14139	25,05	0,49	0,45	-0,09	13305	25,40	0,49	0,44	-0,09	15835
711	X8933	22,41	E	22,70	0,40	-0,27	0,13	10516	24,50	0,43	0,44	-0,09	6893	24,54	0,42	0,43	-0,09	7314
810	X4681	22,41	E	24,82	0,33	-0,24	0,13	9183	27,36	0,29	0,11	0,25	11296	27,16	0,33	0,14	0,24	9798
811	X0036	22,41	E	24,93	0,37	-0,25	0,12	10768	28,10	0,33	0,10	0,25	17242	27,71	0,40	0,15	0,24	12953

FIGURE 13 – Tableau d'indices de végétation des arbres.

1. **info** : Ce tableau est résultat des images multi-spectrales et thermiques du 05/07/2017. Ces calculs sont effectués avec trois segmentations différentes : en premières partie, nous avons segmentées ces images de façons manuellement, puis en deuxième partie, avec le logiciel ERDASIMAGINE. En troisième partie, nous avons segmenté les mêmes images avec la méthodes **U-NET** puis nous avons identifié les arbres avec la méthode WATER-SHED. Les indices de végétation ont été calculés à partir des extractions des valeurs des feuilles de chaque arbre de la même façon.