Projet de Fin d'Études

« Vers une automatisation de l'analyse d'image pour une technique de caractérisation de vulnérabilité à la sécheresse des plantes »

— Rapport —

Étudiants :
Johan David
Léo Souvay
Jon Stark
Théo Videau

Encadrants :
Mme. Aurélie Bugeau
M. Régis Burlett

https://github.com/theovideau/pfe

Mars 2021

Remerciements

Nous tenions tout d'abord à remercier Mme Aurélie BUGEAU de nous avoir encadrés tout au long de ce projet pour son aide et d'avoir répondu à toutes nos questions sur le projet. Merci aussi à M. Régis BURLETT pour ses explications et son travail.

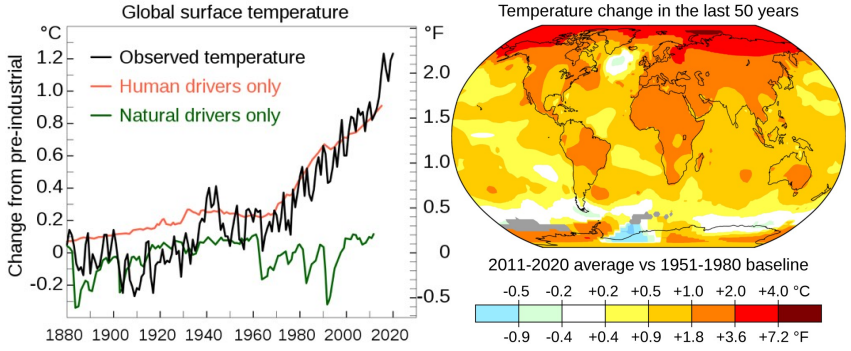
Merci également à notre chargé de TD M. Pascal DESBARATS, avec qui nous avons réalisé le suivi de ce projet tout au long de ce semestre, pour ses précieux conseils.

Table des matières

1 Présentation du projet

Dans le cadre de l'unité d'enseignement *Projet de Fin d'Études (PFE)* au sein de la seconde année de Master Informatique à l'Université de Bordeaux, le sujet que nous avons choisi est le suivant : *Vers une automatisation de l'analyse d'image pour une technique de caractérisation de vulnérabilité à la sécheresse des plantes*. Ce sujet a été proposé par Mme Aurélie BUGEAU chercheuse au LaBRI[1] en collaboration avec M. Régis BURLETT chercheur et membre de l'UMR BIOGECO (Biodiversité, Gènes & Communautés) de l'institut de recherche INRAE[2].

Le climat de notre planète se modifiant, de plus en plus d'événements de sécheresse se produisent dans tous les territoires du globe, dans les forêts, entraînant de nombreuses conséquences souvent irréversibles à de nombreuses espèces de plantes. La mortalité de nombreuses espèces de plantes s'est accélérée ces dernières années et bouleverse ainsi tout l'équilibre des écosystèmes.



(a) Graphique de l'augmentation des températures entre 1880 et 2020. [6]

(b) Moyenne des températures de 2011 à 2020 comparée à la moyenne des températures acquises entre 1951 et 1980. [9]

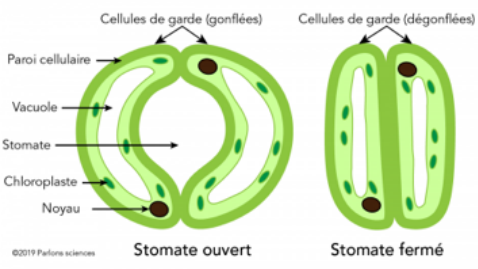
FIGURE 1 – Évolution du climat sur plusieurs années.

Comme nous pouvons le voir sur le graphique de la Figure 1 (a), le climat s'est fortement réchauffé au cours de ces 50 dernières années avec une augmentation de plus de 1.0 °C en moyenne. Depuis 1980, on peut observer

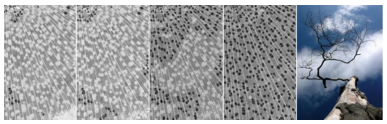
1. Laboratoire Bordelais de Recherche en Informatique.
2. Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement.

une augmentation de +1.0 °C à +2.0 °C en Europe par rapport aux années 1951-1980 et jusqu'à +4.0 °C dans certaines zones du globe Figure 1 (b).

Pour mieux comprendre les mécanismes qui se mettent en place au sein des plantes et étudier les mortalités de nombreuses espèces de plantes, il est important de caractériser la résistance à la sécheresse des végétaux.



(a) Fermeture des stomates ⇒ réversible [1].



(b) Embolie du *xylème* ⇒ irréversible [5].

FIGURE 2 – Mécanismes se produisant au sein de la plante lorsque l'eau vient à manquer.

L'augmentation des températures vue précédemment est corrélée à une augmentation du nombre de sécheresses provoquant une raréfaction de l'eau au sein des sols. Lorsque l'eau vient à manquer dans les feuilles, cela peut provoquer deux mécanismes qui vont avoir des conséquences sur la plante. Tout d'abord, les *stomates* (Figure 2 (a)) vont se fermer à la surface de la feuille pour tenter de préserver un maximum d'eau dans la plante. En effet, ceux-ci sont situés sur la face inférieure des feuilles, et permettent de régler la transpiration de la plante. Cette fermeture est sans risque pour la plante et est réversible. Mais lorsque l'eau vient à manquer pendant de longues périodes, cela peut provoquer un phénomène bien plus grave : l'embolie du *xylème* [3], ce qui n'est pas sans conséquences sur la plante et bien souvent irréversible. Une embolie (Figure 2 (b)) se développe lorsque la circulation est interrompue au sein de la plante et lorsque l'eau n'est plus transportée par le *xylème*. Il s'agit d'un ensemble de vaisseaux au sein des plantes conduisant la sève (eaux et minéraux) des racines jusqu'aux feuilles.

Pour analyser et modéliser les mécanismes se produisant au sein des plantes afin de pouvoir résister au maximum à la sécheresse, le gradient du *potentiel hydrique*, c'est-à-dire la mesure de l'état de tension en eau dans la plante

lorsque le feuillage transpire, va être mesuré pour étudier l'apparition d'embolies dans les feuilles. Cette mesure est réalisée à intervalles de temps réguliers (toutes les 30 minutes) grâce à un *psychromètre*[3] permettant de mesurer le potentiel chimique de l'air au contact de l'eau dans la feuille. Celui-ci correspond à une pression et s'exprime en *MPa (mégapascal)*. Ensuite, les images des feuilles vont être acquises par transmission de la lumière à l'aide d'un scanner ou par caméra et analysées en fonction du temps afin de pouvoir modéliser l'évolution de l'état de la plante pour détecter les embolies. La capture des feuilles par caméra produit des images de moins bonne qualité que par scanner c'est pourquoi ici nous étudierons des images acquises par scanner. Elles sont révélatrices de la réponse des espèces végétales à la sécheresse. Ainsi, l'évolution spatio-temporelle de l'embolie des feuilles va être étudiée. Cette méthode est très efficace quant à l'étude des embolies dues à la sécheresse dans les plantes.

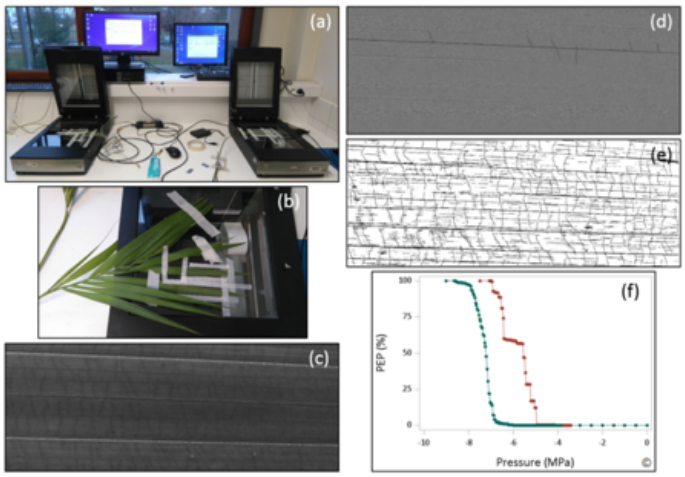


FIGURE 3 – Processus d'acquisition des feuilles [7].

Le dispositif utilise des scanners de bureau dont l'éclairage s'effectue par le haut du *scanner*[4] et l'acquisition se fait par le bas (*cf. Fig. 3(a) 3(b)*), les feuilles sont ensuite scannées à intervalles de temps réguliers tout au long de la déshydratation de la plante (*cf. Fig. 3(c)*). Une fois cette acquisition terminée, l'analyse d'image va permettre de comparer deux images de scans successifs pour déterminer les embolies se produisant dans les vaisseaux du

3. http://www.ictinternational.com/products/psy1/psy1-stem-psychrometer/
4. https://www.epson.fr/products/scanners/consumer-scanners/perfection-v850-pro/

xylème (*cf. Fig. 3(d)*). Le principe de l'analyse est d'évaluer la différence de transmission de lumière dans les vaisseaux de la feuilles lorsque ceux-ci contiennent de l'eau puis de l'air une fois embolisés. À la fin de l'analyse, une cartographie des évènements d'embolie se produisant dans les feuilles est disponible (*cf. Fig. 3(e)*) ainsi que les courbes de vulnérabilité des feuilles à la cavitation, c'est-à-dire la rupture des colonnes d'eau contenue dans le *xylème* (*cf. Fig. 3(f)*).

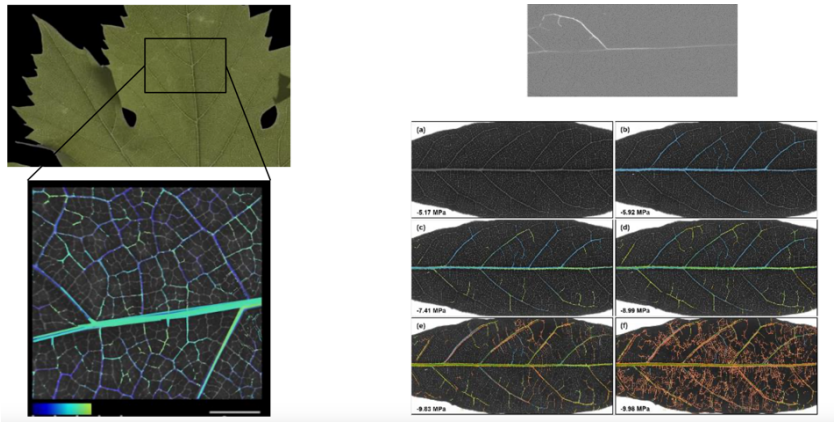


FIGURE 4 – Technique optique de visualisation d'embolies foliaire [4].

Nous pouvons voir sur les images de la Figure 4, la mise en évidence des embolies foliaires dans les feuilles des plantes. En effet, celles-ci sont visibles grâce à un code couleur dépendant du *potentiel hydrique* de la feuille. Le *potentiel hydrique* au sein de la feuille diminue au cours du temps. Et l'embolie est visible sur l'image seuillée en niveaux de gris.

Le principal problème de cette méthode est le temps d'analyse et de traitement des images de feuilles qui est extrêmement long (entre 3 et 6 heures).

L'objectif de ce projet est donc d'accélérer ce processus d'analyse des images en le rendant le plus automatique possible grâce notamment à l'utilisation d'algorithmes de traitement d'images et d'automatisation du processus.

2 Analyse de l'existant

Nous allons évoquer à présent ce qui existe et est à la disposition des chercheurs pour la détection d'embolies au sein d'images de feuilles de plantes.

2.1 Logiciel de traitement d'image `Fiji`

Une méthode manuelle existe actuellement utilisant le logiciel de traitement d'image `Fiji` (version enrichie du logiciel `ImageJ` contenant plusieurs plugins). Un tutoriel de la méthode utilisée est disponible sur `GitHub`[5]. Cette méthode comporte de nombreuses étapes (20 étapes), manuelles, dont il est possible d'automatiser le processus.

2.2 Plugin `OSOV`

Un plugin dédié à l'analyse d'images d'embolies est également disponible. Il s'agit du plugin `OSOV`[6]. *OpenSourceOV* fournit des ressources pour évaluer et analyser l'impact de la sécheresse sur les plantes grâce à l'utilisation de techniques optiques de feuilles. Ce plugin permet ainsi, après importation dans le logiciel `Fiji`, de réaliser des étapes clé pour l'extraction d'informations au sein des images de feuilles pour la détection d'embolies.

2.2.1 Traitement d'images de feuilles

Ce plugin permet par exemple de réaliser une différence pixel à pixel entre 2 images consécutives dans la pile d'images des feuilles, de réaliser un seuillage permettant de mettre en évidence les embolies utiles pour le calcul d'aire (pixel noir = zone d'embolie). Les algorithmes implémentés dans ce plugin seront détaillés dans la partie Choix logiciels.

2.2.2 Analyse des résultats

Dans un second temps, une analyse des résultats est nécessaire après opérations sur les images. Un tutoriel est également disponible sur `GitHub`[7]. Là encore, toutes ces étapes sont réalisées manuellement, nécessitent l'utilisation d'un logiciel tableur (*Microsoft Excel*, *LibreOffice Calc*, ...) et demandent de nombreuses opérations sur les colonnes produites en résultat de l'analyse.

5. https://github.com/OpenSourceOV/image-processing-instructions/blob/master/instructions.md
6. http://www.opensourceov.org/
7. https://github.com/OpenSourceOV/analysis-instructions/blob/master/instructions.md

2.3 Données fournies

Images

Pour procéder à l'analyse d'image d'embolies, trois jeux de données provenant de différentes feuilles d'arbres nous ont été transmis :

- *morus alba* : contenant 1065 images de taille 986×3127.
- *ostrya carpinifolia* : contenant 1709 images de taille 2594×542. (*cf. Figure* 5)
- *quercus cerris* : contenant 892 images de taille 3160×576.

Ces images sont au format de fichier `.tif` et sont de très haute qualité. L'acquisition de ces images de feuilles a été réalisée par scanners et à intervalles de temps réguliers (toutes les 5 minutes). Chaque image a un poids de quelques `Mo`.

FIGURE 5 – Extrait d'une feuille de l'espèce *ostrya carpinifolia*.

Résultats bilans

De plus, des fichiers au format `.csv` contiennent des calculs réalisés au préalable grâce à un autre type d'acquisition et présentent une colonne dont le *potentiel hydrique (WP_MPa)* est mesuré (*cf. Figure* 6). Cette valeur va nous être nécessaire pour calculer le *pourcentage d'embolies* en fonction du *potentiel hydrique*. D'autres informations figurent dans ce fichier comme par exemple l'espèce de plante étudiée, la date et l'heure d'acquisition (au format *Excel*).

Bilan_morus_a... Ouvrir avec Microsoft Excel

esp	ind	date_hour	pep	WP_MPa
morusalba	1	6977.66666666667	0	-1.27
morusalba	1	6977.75	0	-1.255
morusalba	1	6977.83333333333	0	-1.24
morusalba	1	6977.91666666667	0	-1.225
morusalba	1	6978	0	-1.21
morusalba	1	6978.08333333333	0	-1.22
morusalba	1	6978.16666666667	0	-1.23
morusalba	1	6978.25	0	-1.24
morusalba	1	6978.33333333333	0	-1.25
morusalba	1	6978.41666666667	0.0254317265384663	-1.24

FIGURE 6 – Extrait du bilan mesuré pour *morus alba*.

Courbes des résultats

Enfin, les courbes des résultats obtenues manuellement avec cette méthode nous sont fournies (*cf. Figure* 7). Elles correspondent à ce que nous devons obtenir et ce de manière automatique. Chaque point correspond au nombre de pixels embolisés en fonction du potentiel hydrique de la feuille. Chaque point représente la différence entre 2 images consécutives.

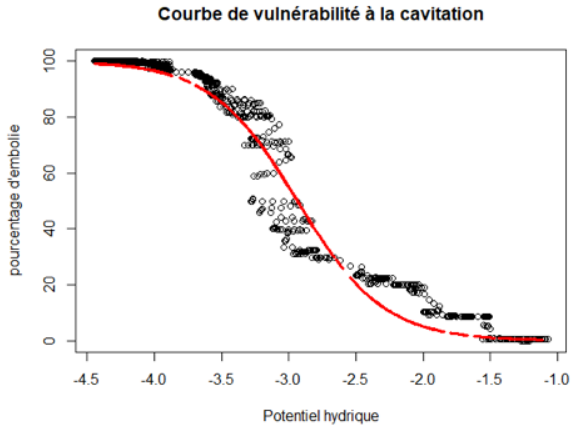


FIGURE 7 – Courbe de vulnérabilité à la cavitation mesurée pour *morus alba*.

2.4 Objectif

L'objectif de ce projet est donc d'automatiser l'ensemble de ces étapes (traitement d'images et analyse des résultats dans le tableur) dans le but d'obtenir une courbe finale correspondant au pourcentage d'embolies en fonction du potentiel hydrique (courbe de vulnérabilité de la plante à la cavitation).

3 Récit utilisateur

Voici les principales étapes du récit utilisateur (*user stories*) exprimées par notre client dans le cadre de ce projet :

Simplicité d'installation : le code que nous devrons fournir doit être facile d'installation au sein du logiciel `Fiji` ou bien facilement compilable et exécutable en dehors de ce logiciel ;

Simplicité d'utilisation : ce que nous devrons développer doit être facile d'utilisation en minimisant le nombre d'étapes nécessaires et le nombre de fenêtres s'affichant à l'écran de l'utilisateur ;

Robuste aux différentes acquisitions : notre programme doit pouvoir s'adapter à tous types d'acquisition ;

Reprendre la main sur l'automatisation : notre programme doit également pouvoir s'interrompre après une étape afin que l'utilisateur puisse réaliser des opérations manuelles supplémentaires si celles réalisées automatiquement ne lui sont pas suffisantes. Cela permet également de visualiser les résultats produits sur la pile d'images une fois les opérations appliquées.

Séparation en 2 parties des étapes : notre programme doit pouvoir s'exécuter en 2 étapes :

- Calcul des zones d'embolies (traitement d'image).
- Calcul de la courbe de vulnérabilité (résultats analytiques).

Cela afin de pouvoir lancer le calcul des résultats sur des piles d'images déjà traitées ou bien si le traitement des images n'a pas été suffisant.

Intégration en un seul programme : actuellement, les traitements des images sont réalisés au sein du logiciel `Fiji` et les calculs sont réalisés à la main dans un logiciel de tableur (*Microsoft Excel, LibreOffice Calc,...*). L'objectif ici serait de tout réunir en un seul programme (par exemple, tout réunir dans le logiciel `Fiji`).

4 Besoins

L'objectif principal de ce projet est donc d'automatiser les étapes réalisées manuellement par les chercheurs après acquisition des différentes feuilles de plantes à intervalles de temps réguliers. Ce projet doit répondre à des besoins fonctionnels et non fonctionnels spécifiques pour son bon déroulement.

4.1 Besoins fonctionnels

Importation des images

L'importation des images doit permettre de travailler sur des images au format `TIFF` et ainsi produire une seule et même pile d'images dans ce même format. Les images doivent conserver les mêmes dimensions au fil des acquisitions et être classées dans l'ordre chronologique des acquisitions.

Temps de traitement

Le temps de traitement doit également constituer un besoin fonctionnel. En effet, il doit permettre un gain important sur le temps de traitement manuel qui est actuellement de plusieurs heures (entre 3h et 6h).

Automatiser les étapes du traitement

Les étapes de traitement d'images pour localiser, mettre en évidence et mesurer la zone d'embolie (aire) sur les feuilles sont réalisées manuellement. Il faut ainsi les automatiser et les appliquer ensuite aux images en entrée. Certains des paramètres clés utilisés seront sélectionnables manuellement afin de conserver la meilleure qualité en fonction des acquisitions.

Suppression des étapes non essentielles

Certaines étapes décrites dans le tutoriel ne sont que visuelles et ainsi non essentielles au traitement d'images. Elles servent uniquement de visualisation à l'utilisateur pour permettre de vérifier ce qu'il produit à chaque étape tout au long du tutoriel.

Conservation d'une qualité de résultat satisfaisante

Un des enjeux de l'automatisation d'un tel processus est de conserver une qualité d'image satisfaisante. En effet, celle-ci doit permettre de réduire le temps de traitement des différentes images mais également conserver et

garantir une qualité de résultat au moins aussi élevée que lors du traitement manuel.

Minimiser le nombre de fenêtres ouvertes dans le logiciel `Fiji`

Afin d'automatiser au maximum le processus de traitement des images, il est important de réduire le nombre de fenêtres ouvertes simultanément lors des différentes étapes dans `Fiji`. L'idée ici est de n'avoir que deux fenêtres ouvertes en même temps : la pile d'images sources et la pile d'images au fur et à mesure du traitement. Enfin, la courbe finale doit s'afficher à la fin du traitement ainsi que les résultats. Nous limitons ainsi les interactions utilisateur qui sont les étapes les plus chronophages.

Sauvegarde et exportation des résultats obtenus

Afin de ne pas avoir à reproduire l'ensemble du processus, le plugin doit permettre à l'utilisateur de sauvegarder ses opérations à chaque étape. Ainsi, il doit pouvoir sauvegarder quand il le souhaite les piles d'images obtenues après traitement mais également sauvegarder les résultats obtenus lors de la phase de calcul des résultats ainsi que les courbes. Afin de conserver le choix des utilisateurs, nous avons mis en place une sauvegarde des résultats au format `.csv`.

4.2 Besoins non fonctionnels

Intégration du programme au logiciel `Fiji`

Pour permettre une simplicité d'utilisation et de mise en place pour l'utilisateur, nous avons choisi de développer ce projet au sein du logiciel `Fiji` et plus particulièrement à la suite du plugin `OSOV` *Image Processing* en intégrant également la partie d'analyse des résultats `OSOV` *Analysis* à la suite de celui-ci.

5 Choix logiciels

Dans cette partie nous allons évoquer les choix logiciels réalisés. Nous avons fait le choix d'automatiser ce processus au sein même du plugin existant. Ce logiciel et ce plugin sont développés en Java et permettent d'être intégrés directement.

Nous nous basons sur les algorithmes de traitement d'images disponibles au sein de `Fiji` comprenant de nombreuses méthodes de seuillages pour mettre en évidence les embolies, de suppression du bruit et de mises à l'échelle.

Nous utilisons le plugin `OSOV Image Processing` ainsi que les méthodes de mesures du plugin `OSOV Analysis` contenant une opération de différence entre deux images consécutives dans la pile et une méthode pour mesurer l'aire de l'embolie.

Nous allons détailler ici le fonctionnement de ces algorithmes.

Importation des images

Tout d'abord, la phase d'importation des images au format `.tif` s'effectue avec la commande *Import > Image Sequence*. Ensuite, il faut veiller à cocher la case *Convert to 8-bit Grayscale* pour garantir la conversion dans le bon format car les traitements se déroulent au format *8-bit* afin de conserver la qualité de l'image tout en réduisant sa taille en mémoire. Il est également possible de choisir le pourcentage d'échelle de l'image, par défaut l'échelle est à 100% pour préserver la qualité de l'image. Ces étapes permettent la préservation de la qualité de l'image, tout en réduisant le nombre de couleur pour réduire l'espace en mémoire utilisé. Réduire la taille de l'image, en utilisant une échelle de 25% ou 50%, peut donner des résultats non valides où il n'est plus possible de visualiser les embolies sur certaines images. Une fois les images importées, celles-ci sont représentées sous la forme d'une seule pile d'images permettant ainsi l'affichage d'une seule fenêtre.

Importation du plugin

Ensuite, l'étape d'importation du plugin `OSOV` au sein du logiciel `Fiji` est nécessaire pour lancer les opérations dédiées à la détection d'embolies sur les différentes piles d'images de feuilles. Ce plugin contiendra le code initial pour les opérations de traitement d'image ainsi que le code que nous avons

développé à la suite de celui-ci pour automatiser le processus du calcul de courbe de vulnérabilité. Nous les avons appelés *Auto Embolism Detection* et *Auto Vulnerability Curve*.

Différence d'images

Nous avons conservé l'implémentation de la méthode *Image Difference V2* dans notre plugin permettant de calculer la différence relative entre deux images consécutives pixel à pixel, sans recadrage et sans égalisation d'histogramme ou de luminance :

$$I_1(x, y) - I_2(x, y) \tag{1}$$

Seuillage

L'étape de seuillage constitue une étape clé du processus d'automatisation. Il s'agit ici des fonctionnalités *Threshold* et *Convert Stack to Binary*. En effet, cette étape permet de seuiller l'ensemble des images de la pile dans le but de mettre en évidence les embolies caractéristiques sur les feuilles des plantes. Les algorithmes de seuillage sont ceux développés dans `Fiji` et leur choix est important, car il faut tenter de minimiser la présence de bruit sur les images, dû à la différence entre deux images consécutives lié aux mouvements de la feuille entre deux acquisitions, tout en conservant au maximum les zones d'embolies et éviter de réduire leur taille à la surface de la plante. L'image ainsi produite est une image seuillée où l'arrière-plan sera blanc et les embolies apparaîtront en noir.

Pour ce faire, nous avons testé les algorithmes de seuillage disponibles et sélectionné quelques algorithmes seulement car certains laissaient trop de bruit sur l'image et il est alors très difficile de le supprimer une fois l'image seuillée sans éliminer la zone d'embolie. Le choix des algorithmes de seuillage va dépendre également de la qualité des acquisitions. Les algorithmes de seuillage utilisés également par les chercheurs que nous avons sélectionnés sont les suivants :

- `IsoData` [2] :
  L'algorithme cherche à séparer les éléments de l'image à son arrière plan. À partir de la plus faible valeur d'un pixel de l'image, G, on détermine L qui est la moyenne de la valeur des pixels inférieurs à G et H qui correspond à la moyenne de la valeur des pixels supérieurs à G. On cherche à obtenir un seuil égal à G tel que G est égal à la moyenne

de L et H. Tant que G est inférieur à cette moyenne, on répète les opérations précédentes en incrémentant G.

Algorithm 1 Isodata

Require: $Hist \leftarrow$ Histogramme de l'image.
  $i = 0$
  while $Hist[i] \neq 0$ **do**
    $i++$
  end while
  $G = i$
  $L = Moyenne(Hist[0 \text{ à } G])$
  $H = Moyenne(Hist[G \text{ à } 255])$
  while $G < (L + H)/2$ **do**
    $G++$
    $L = Moyenne(Hist[i < G])$
    $H = Moyenne(Hist[i > G])$
  end while
  return G

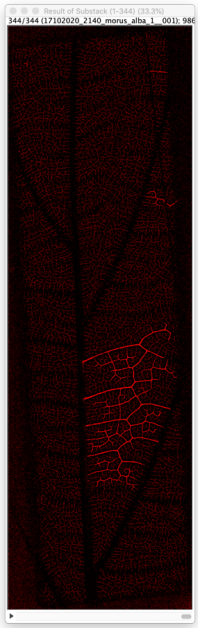


FIGURE 8 – Application de l'algorithme `IsoData` sur *morus alba*.

Cet algorithme produit un résultat assez précis au niveau des embolies mais conserve malgré tout un niveau de bruit assez élevé comme nous pouvons le voir sur la Figure 8.

- `IJ_IsoData` : Cet algorithme de seuillage est l'implémentation originale de l'algorithme précédent `IsoData`. Celui-ci reprend le même principe mais diffère sur le calcul de la moyenne en forçant le typage de certaines variables. Cela entraîne une légère différence sur certaines images.

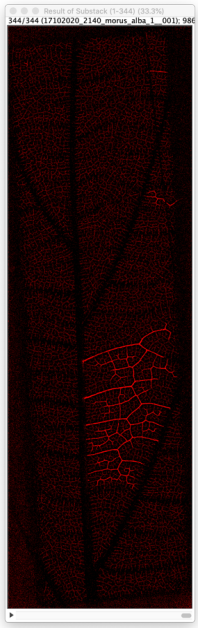


FIGURE 9 – Application de l'algorithme `IJ_IsoData` sur *morus alba*.

Comme nous pouvons le voir sur la Figure 9, le résultat produit est sensiblement identique à celui observé sur la Figure 8.

- `Moments` [10] : Le moment d'une image correspond à la moyenne pondérée des pixels de l'image. L'objectif de cet algorithme de seuillage est de préserver les quatre premiers moments de l'image après seuillage. Le premier moment M_0 est égal à 1 alors que les trois autres moments sont calculés de la manière suivante :

$$M_w = \sum_{i=0}^{255} i^w * Hist, w = 1, 2, 3 \tag{2}$$

avec w les poids attribués aux pixels et $Hist$ l'histogramme normalisé.

À l'aide des quatre moments de l'image, l'algorithme va résoudre un système d'équation afin d'obtenir la variable p_0, correspondant à un pourcentage de pixels permettant de préserver les quatre premiers moments. Enfin, il suffit de trouver l'indice de l'histogramme normalisé cumulé qui permet de se rapprocher le plus possible de p_0, afin d'obtenir notre seuil.

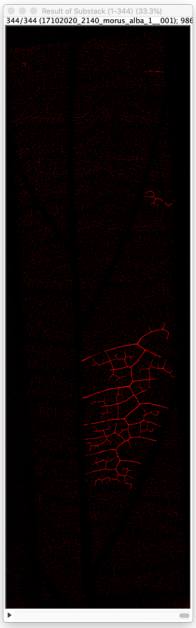


FIGURE 10 – Application de l'algorithme `Moments` sur *morus alba*.

Cet algorithme est assez efficace sur la détection d'embolies comme nous pouvons le voir sur la Figure 10, et contient un faible niveau de bruit.

- `Li` [8] : Cette méthode permet de trouver un seuillage minimisant la *cross entropy* de l'image et de l'image seuillée afin de détacher les éléments de l'image de l'arrière plan.

  Pour cela, on cherche à minimiser la fonction η qui prend en compte la moyenne de l'arrière plan et l'objet de l'image à l'aide de ses *moments*.

La valeur optimale pour le seuillage t est obtenue lorsque la dérivée de η est nulle. On continue de converger vers une solution tant que la différence de deux valeurs de t est supérieure à un pas donné.

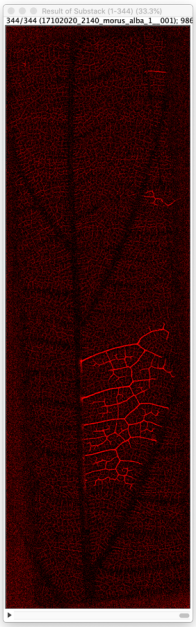


FIGURE 11 – Application de l'algorithme `Li` sur *morus alba*.

Cet algorithme est plus précis que les précédents mais prend en compte forcément plus de bruit. En effet, comme nous pouvons le voir sur la Figure 11, les embolies sont très nettes et précises, mais un fort niveau de bruit est présent dans l'image.

Suppression du bruit

L'étape de suppression du bruit dans les images de feuilles fournies est une étape cruciale dans le bon déroulement de l'automatisation du processus de calcul des zones d'embolies. En effet, après seuillage, une image contient deux couleurs : blanc pour l'arrière-plan et noir modélisant les informations. Si sur les images seuillées du bruit est encore présent, alors celles-ci seront considérées comme zone d'embolies lors du calcul des mesures à l'étape suivante. La présence de bruit dans les images après seuillage va donc fausser les résultats. De plus, celui-ci diffère selon les acquisitions et va donc consti-

tuer un élément clé dans l'automatisation du processus et son bon déroulement. La fonctionnalité que nous avons utilisée pour supprimer le bruit dans l'image est *Remove Outliers*. Cette méthode permet de remplacer un pixel par la médiane des pixels qui l'entourent s'il s'écarte de la médiane de plus d'une certaine valeur (`seuil`). Cette méthode permet de supprimer du bruit dans l'image, mais montre rapidement ces limites. En effet, nous verrons plus en détail ce qui a posé problème dans la partie Tests et les opérations que nous avons testées pour tenter de supprimer un maximum de bruit.

Sélection de l'unité de mesures

L'unité de mesure ici est l'aire. Elle correspond à la surface d'embolie sur chaque image de feuille. Cette étape a été automatisée à la suite de l'étape précédente grâce à la sélection du paramètre `Area` dans le choix de l'unité de mesure des embolies sur les feuilles.

Mise à l'échelle

Les images fournies sont déjà mises à l'échelle. C'est-à-dire que 1 pixel sur l'image correspond à 1 unité. Cette étape est donc automatisée avec les valeurs prédéfinies.

Calculs des mesures

Grâce au plugin `OSOV Analysis`, les mesures sont réalisées automatiquement à partir des zones d'embolies situées sur les feuilles et calculées en fonction de leur aire puis sauvegardées au format `.csv`. Cela est disponible dans le second plugin *Auto Vulnerability Curve*. Une fois ces mesures réalisées, il est alors possible après importation d'un autre fichier `.csv` contenant lui les mesures des *potentiels hydriques (WP_MPa)* réalisées lors d'une précédente acquisition, de tracer la courbe des valeurs du pourcentage d'embolies en fonction du potentiel hydrique. Il s'agit de la courbe finale de vulnérabilité de la plante à la cavitation que les chercheurs vont étudier par la suite. Ces mesures sont calculées de la manière suivante :

- Régression linéaire des valeurs des *potentiels hydriques (WP_MPa)* en fonction du temps
- Calculs des nouveaux *potentiels hydriques* à partir de la régression linéaire et résolution de l'équation :

$$ph[i] = a + bt + ct^2 + dt^3 \tag{3}$$

où $ph[i]$ correspond aux nouveaux *potentiels hydriques* et t le temps.

- Calcul de l'aire cumulative totale :

$$cumulativeArea[i] = sum[i]/sum[nResults - 1] * 100; \qquad (4)$$

où $cumulativeArea[i]$ correspondant au pourcentage total de l'aire d'embolie, $sum[i]$ correspond à la somme de l'aire cumulée et $sum[nResults-1]$ au maximum de l'aire cumulée pour avoir un pourcentage.

- Enfin, nous pouvons tracer les valeurs obtenues aux équations (3) et (4) pour obtenir la courbe finale de vulnérabilité à la cavitation.

6 Architecture

L'architecture de notre logiciel s'organise de la manière suivante :

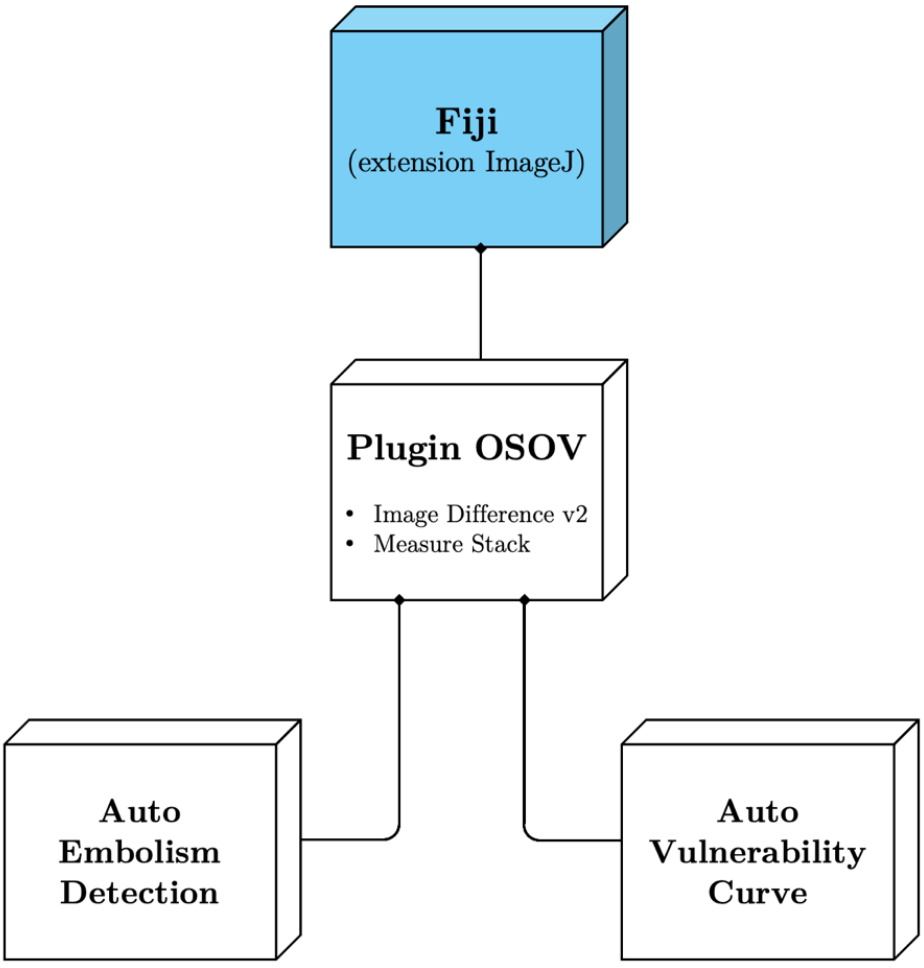


FIGURE 12 – Diagramme de déploiement du logiciel.

7 Diagramme de Gantt prévisionnel

Voici le diagramme de Gantt prévisionnel de notre projet :

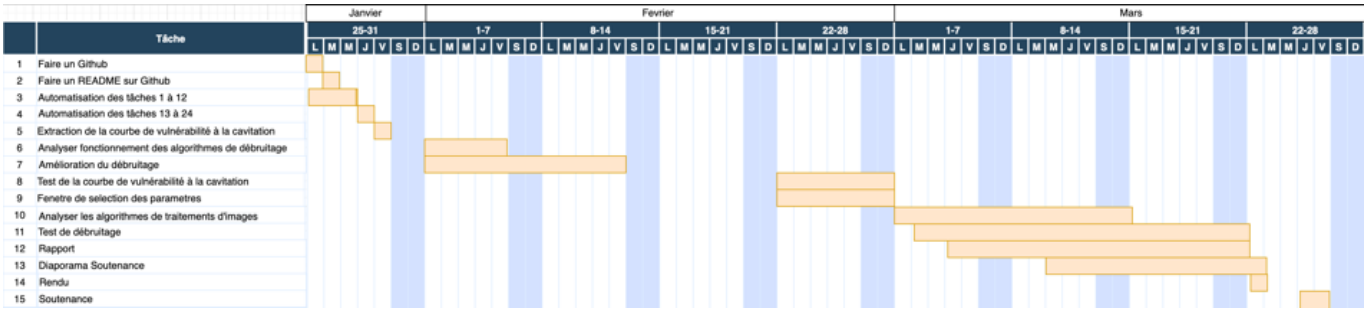


FIGURE 13 – Diagramme de Gantt prévisionnel

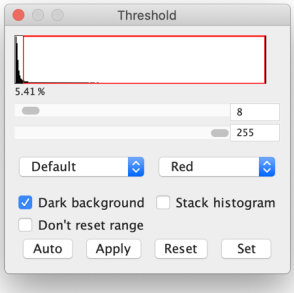
8 Réalisation

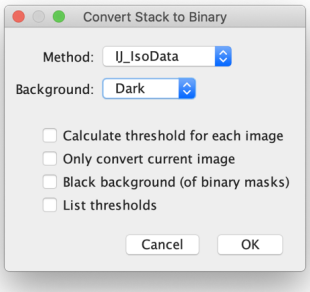
Nous allons voir dans cette partie ce que nous avons produit à partir de l'existant pour automatiser le processus de détection d'embolies et de calcul d'aire sur les différentes images des piles d'images fournies.

8.1 Automatisation de la détection des embolies (`Auto Embolism Detection`)

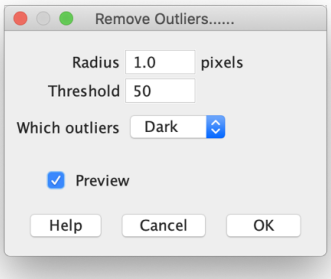
Après chargement du plugin `OSOV` dans le logiciel `Fiji` et importation des images, il est alors possible de lancer le processus d'automatisation de la détection des embolies (`Auto Embolism Detection`). Voici les étapes que l'utilisateur va rencontrer lors du déroulement de l'automatisation :

- `Threshold & Convert Stack to Binary` : Choix de l'algorithme de seuillage.

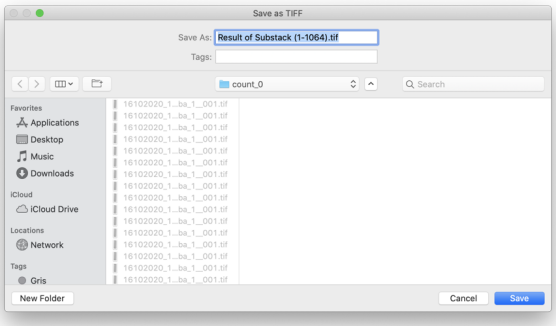




- `Remove Outliers` : Choix des paramètres pour la suppression du bruit.

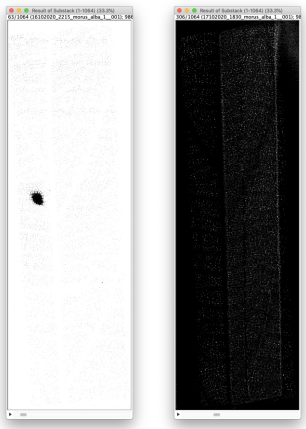


- `Save as TIFF` : Possibilité de sauvegarder la pile d'image après traitement.



En fonction des différentes images, l'utilisateur pourra choisir l'algorithme de seuillage le plus optimal, de même pour la suppression du bruit. Pour ce faire, nous avons permis de passer à une étape suivante en appuyant sur la barre d'espace du clavier de l'utilisateur. Cela permet ainsi à l'utilisateur de revenir sur l'étape qu'il vient de réaliser, vérifier les résultats sur toute la pile d'images, ou bien encore de réaliser des opérations supplémentaires manuellement, si les résultats obtenus ne lui sont pas satisfaisant, entre deux étapes du traitement automatique.

De plus, nous avons procédé à la suppression des images présentant des *artefacts* (problèmes d'acquisition, parasites sur l'image, ...) en cherchant dans l'image des zones de taille 5×5 pixels où ceux-ci sont noirs et représentent ainsi des problèmes d'acquisition.



(a) (b)

FIGURE 14 – Exemples d'images présentant des problèmes d'acquisition.

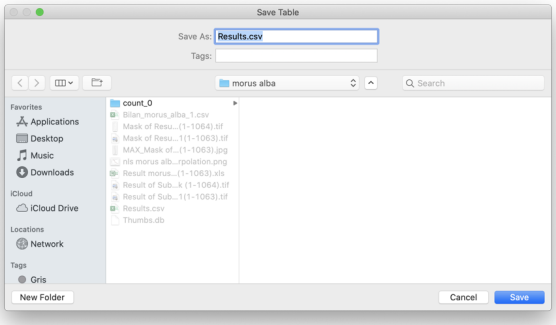
On peut voir sur la Figure 14 que des zones noires seront comptabilisées comme zones d'embolies si elles ne sont pas supprimées et ainsi fausser les résultats.

8.2 Automatisation des calculs de la courbe de vulnérabilité (`Auto Vulnerability Curve`)

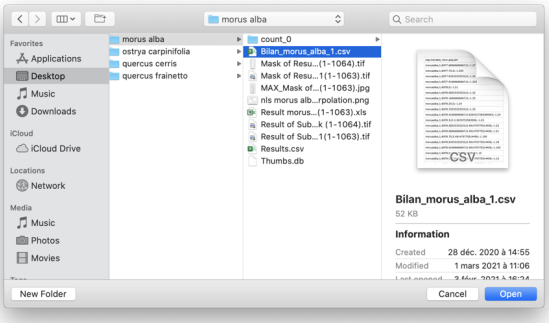
Après traitement sur les différentes images, il est alors possible de lancer le processus d'automatisation des calculs de la courbe de vulnérabilité (`Auto Vulnerability Curve`).

Voici les étapes que l'utilisateur va rencontrer lors du déroulement de l'automatisation :

- `Save Table` : Sauvegarde du fichier résultat contenant les zones d'aires sur chaque image au format `.csv`.



- `Select the csv file with water potential` : Importation du fichier `.csv` contenant les valeurs des *potentiels hydriques* (`WP_MPa`).



Il faut veiller à ce que le nom de la colonne soit `"WP_MPa"` et que les données de chaque ligne du fichier `csv` sont séparées par un point-virgule `";"`.

Voici la courbe de résultat en Figure 15 obtenue après traitement des images et calculs de la courbe de vulnérabilité (`Auto Embolism Detection + Auto Vulnerability Curve`) :

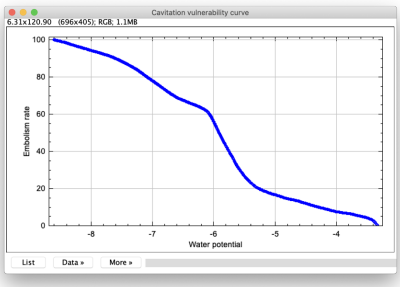


FIGURE 15 – Visualisation des résultats après traitement automatique pour *ostrya carpinifolia*.

Voici la courbe de résultat obtenue Figure 16 uniquement après calculs de la courbe de vulnérabilité à partir des images traitées manuellement (`Auto Vulnerability Curve`) :

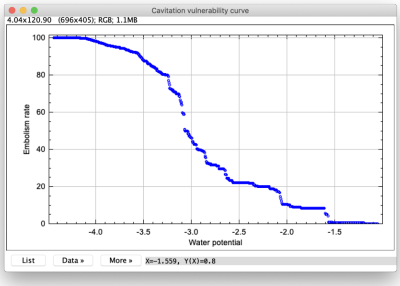


FIGURE 16 – Visualisation des résultats avec traitement manuel pour *morus alba*.

La courbe de vulnérabilité obtenue avec l'ensemble des étapes correspond au résultat final, qui permet de caractériser l'échantillon. Il s'agit de la courbe du pourcentage de pixels appartenant à une embolie (mesuré à partir des images) en fonction du potentiel hydrique (mesuré avec un capteur et qui correspond à la ligne `WP_MPa` dans les fichiers de bilans).

9 Tests

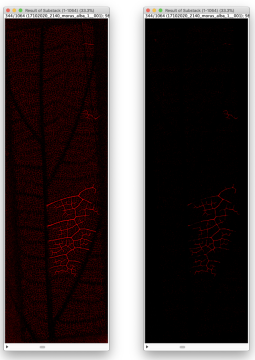
Nous avons consacré une grande partie à la réalisation de tests pour le développement de ce projet. En effet, nous avons testé de nombreux éléments : la méthode utilisée, le traitement et le calcul analytique des résultats, mais également de nombreuses méthodes pour affiner les résultats (suppression du bruit).

9.1 Tests des différents étapes du tutoriel

Tout d'abord, nous avons testé toutes les étapes de la méthode proposée. Nous avons également pu déterminer quelles étapes étaient réellement indispensables au bon déroulement du processus d'automatisation ainsi que celles qui étaient purement visuelles et qui permettaient à l'utilisateur d'avoir un retour des actions qu'il menait. Nous avons rapidement éliminé certaines étapes et testé ensuite les différents paramètres de celles qui restaient.

Nous avons testé :

- Les différents algorithmes de seuillages de la fonction *Threshold* et vu quels algorithmes étaient utilisés par les chercheurs et quels résultats ils produisaient sur les images. Comme décrit précédemment, nous avons sélectionné quelques algorithmes assez efficaces sur les jeux de données fournis (*cf.* `IsoData`, `IJ_IsoData`, `Moments`, `Li`) c'est-à-dire laissant les zones d'embolies assez nettes tout en réduisant au maximum le bruit dans l'image.



(a) (b)

FIGURE 17 – Résultat après seuillage avec `Li` (a) et `IJ_IsoData` (b).

Nous pouvons voir par exemple que certains algorithmes laissent trop de bruit dans l'image avec `Li` (Figure 17(a)) alors que `IJ_IsoData` (Figure 17(b)) semble plus efficace pour localiser les zones d'embolies.

- Les différents paramètres de la fonction *Remove Outliers* pour supprimer le bruit dans l'image

(a) (b)

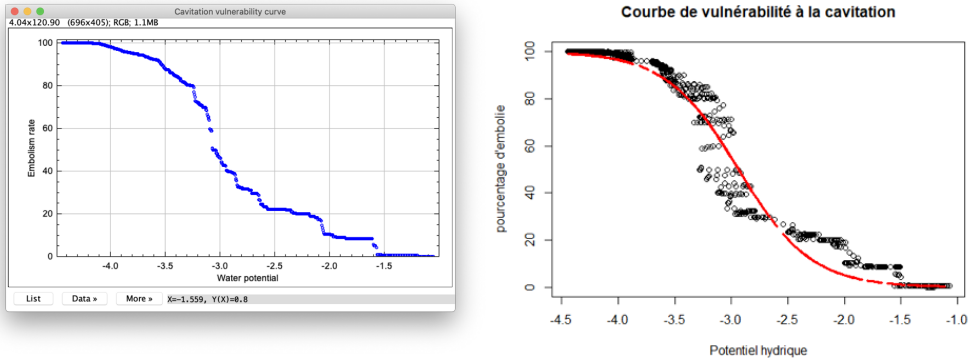
FIGURE 18 – Résultat après suppression du bruit avec un rayon de `1.0` (a) et `2.0` (b).

Nous pouvons voir sur la Figure 18 que plus le rayon est important, moins il y a de bruit, mais plus l'embolie est tronquée.

9.2 Tests des résultats analytiques

Nous avons testé la partie calcul des résultats grâce aux données fournies par les chercheurs et notamment grâce à la pile d'image fournies seuillée et calculée manuellement. Nous avons pu lancer le processus d'automatisation sur celle-ci en ne réalisant aucun traitement supplémentaire et ainsi comparer la courbe finale à celle fournie également.

Voici les résultats que nous avons obtenus :



(a) (b)

FIGURE 19 – Comparaison de la courbe de résultat automatique (a) et réalisée manuellement (b).

Nous pouvons voir sur ces deux courbes (Figure 19(a) et Figure 19(b)) qu'elles suivent la même forme confirmant ainsi que les calculs effectués automatiquement sont corrects.

Nous nous sommes rendus compte également lors de tests que certaines images étaient inversées faussant ainsi la courbe finale des résultats car la zone d'embolie détectée représentait la taille totale de l'image. Cela provoquait une cassure sur la courbe faisant saturer ainsi le pourcentage d'embolie mesuré.

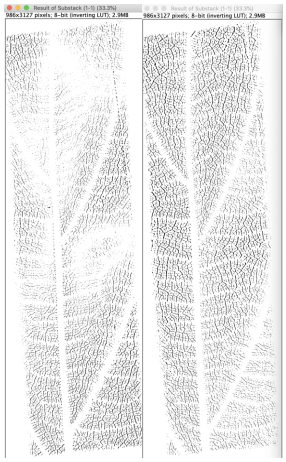
9.3 Tests d'affinage des résultats

Enfin, nous avons testé de nombreuses méthodes pour affiner les résultats en supprimant par exemple davantage de bruit dans l'image. En effet, comme vu précédemment, la méthode *Remove Outliers* fournie de base dans le logiciel `Fiji` n'est pas forcément assez performant pour supprimer le bruit et ce, dans les jeux de données fournies. Ce bruit est provoqué lors de la différence (*Image Difference v2*) car au fil du temps, la feuille se dessèche et ainsi se rétracte au fur et à mesure des acquisitions faussant ainsi la différence pixel à pixel entre deux images d'acquisition. Les images étant acquises toutes les 5 minutes, cela provoque rapidement des disparités lors de la différence.

De plus, le bruit est différent d'une acquisition à une autre. Certaines feuilles vont se dessécher plus rapidement que d'autres, vont se contracter d'une manière différente et le bruit sera ainsi propre à chaque acquisition. Nous n'avons eu que trois jeux de données pour tester les méthodes ce qui n'est pas représentatif de ce que peut être tous types d'acquisitions. Il se peut que sur certaines espèces de feuilles plus épaisses et plus résistantes, le dessèchement s'opère de manière moins rapide et les feuilles se rétracteront moins rapidement. La méthode *Remove Outliers* sera peut-être alors suffisante pour supprimer le bruit présent sur les images.

Recalage

Nous avons d'abord testé le recalage manuel pour vérifier qu'il s'agit bien d'un bruit lié à la différence et à la déformation de la feuille. Nous avons pu constater qu'à l'endroit où nous avions superposé les deux images la différence était nulle et tous les pixels étaient blancs (Figure 20(a) et Figure 20(b)). En revanche la déformation n'étant pas linéaire, sur les extrémités de l'image, le bruit était encore présent.



(a) (b)

FIGURE 20 – Comparaison du seuillage avec (a) et sans recalage manuel (b).

Nous avons aussi essayé d'utiliser le plugin `BUnwarpJ`[8] pour faire du recalage, le temps de traitement est long et le résultat n'est pas concluant car le contraste des images recalées n'est pas la même que celle des images de base, ce qui ne débruite pas les images.
Des opérations utilisant le *flot optique* pourraient permettre de calculer un vecteur représentant le décalage entre deux acquisitions consécutives sur les pixels en mouvement et ainsi permettre ce recalage.

Opérateurs morphologiques

Nous avons également tenté de supprimer le bruit en utilisant des opérateurs morphologiques (ouverture / fermeture / érosion / dilatation) et plus particulièrement une ouverture grâce à la fonctionnalité *Morphology* > *Gray Morphology* dans `Fiji` qui permet de choisir la taille du rayon de détection en pixel. Cette opération est relativement longue sur la totalité de la pile d'image et peu efficace. En effet, de nombreux pixels de bruits étaient encore présents après application de l'ouverture. De plus, cela a également un impact sur la zone d'embolie qui devient ainsi un peu moins visible.

Ajout de flou gaussien

L'ajout de flou gaussien sur les images initiales permet de supprimer le bruit sur l'image finale. Cependant, sur les images fortement bruitées cela ne s'avère pas assez efficace.

Filtre médian

Le filtre médian permet de procéder à une réduction du bruit sur une image tout en gardant les contours de celle-ci. Il permet de remplacer chaque pixel par la valeur médiane des pixels de son voisinage. Ce filtre n'est pas très efficace pour supprimer le bruit tout en conservant les zones d'embolies.

Filtre bilatéral

Nous avons testé d'appliquer un filtre bilatéral sur les images initiales. Il s'agit d'un filtre de lissage préservant les contours et réduisant le bruit des images. Il remplace l'intensité de chaque pixel par une moyenne pondérée des valeurs d'intensité des pixels voisins. Cela ne nous a pas permis d'obtenir une meilleure qualité de résultat surtout au niveau des zones d'embolies. De

8. `https://imagej.net/BUnwarpJ`

plus, l'application de ce filtre sur toutes les images demande un temps de traitement assez élevé.

Deep Learning

Enfin, nous avons tenté une approche avec de l'apprentissage afin de supprimer le bruit. Nous avons d'abord essayé d'utiliser le plugin `DenoiSeg`[9] afin d'effectuer cet apprentissage. Cependant, nous n'avons pas pu avoir de résultats car des erreurs liées à l'exécution nous ont empêché de réaliser un apprentissage et des prédictions.

9. `https://imagej.net/DenoiSeg`

10 Projet

Nous avons globalement respecté le planning initial que nous avions prévu sur le Diagramme de Gantt prévisionnel et l'automatisation de certaines tâches a pu être réalisé facilement, en revanche certaines étapes nous ont demandé beaucoup plus de temps pour affiner les résultats et ainsi supprimer davantage de bruit sur les images. Nous avons testé de nombreuses méthodes comme détaillés dans la section précédente et cela nous a demandé du temps supplémentaire.

Nous avons donc réalisé une version *semi-automatique* de ce projet. C'est-à-dire que nous avons pu automatiser certaines étapes, comme par exemple :

- Le lancement des différentes étapes de traitement d'image.
- Le calcul des résultats à partir des fichiers `csv`.

Mais d'autres sont manuelles :

- Choix de l'algorithme de seuillage.
- Choix des paramètres pour la suppression du bruit.
- Importation des fichiers bilan `csv`.

Nous avons donc une version opérationnelle de ce projet mais reste à affiner les résultats sur les différentes images après seuillage pour que celui-ci soit pleinement efficace et proche des résultats obtenus manuellement par les chercheurs.

11 Conclusion

11.1 Bilan général du projet

Ce projet était très intéressant à réaliser de part son sujet actuel et important pour comprendre ce que le changement climatique peut provoquer comme conséquences sur les différentes espèces de plantes de notre planète. Grâce à cette étude, les chercheurs peuvent ainsi analyser plus rapidement et mieux comprendre les mécanismes qui se mettent en place au sein des feuilles lorsque l'eau vient à manquer entraînant des embolies du *xylème* et des mortalités dans les plantes, comprendre les dégâts que cela peut provoquer sur ces espèces de plantes et ainsi modéliser la réponse des espèces aux changements climatiques.

11.2 Perspectives

L'affinage des résultats pour obtenir des courbes de résultat davantage proche du travail effectué manuellement représente la principale perspective de ce projet. Il faudrait par la suite travailler sur le recalage de l'image car la feuille se dessèche et se déforme au cours du temps provoquant des erreurs lors des calculs des différences sur deux images consécutives. Il faudrait ainsi tenter de recaler les images, effectuer des transformations par rapport à l'image initiale pour pouvoir ensuite effectuer une différence et supprimer le bruit restant après seuillage. Cela représente la principale perspective de ce projet.
Le Deep Learning pourrait être une solution viable afin d'obtenir des bons résultats, dans ce cas-là, il faudra créer un réseau afin de l'entraîner pour ensuite l'utiliser. Cependant, la taille des données fournies étant très élevée, l'apprentissage prend beaucoup de temps, de plus, il faut un ordinateur avec beaucoup de mémoire RAM pour faire tourner le réseau, sachant qu'un ordinateur possédant 16 Go de RAM peut seulement entraîner un réseau avec un seul jeu de données à une échelle de 50%.

12 Annexe – tutoriel d'installation et d'utilisation du plugin

Un fichier `README.md` est disponible sur notre répertoire `GitHub` à l'adresse suivante `https://github.com/theovideau/pfe` détaillant toutes les étapes pour l'installation et l'utilisation du plugin `Auto Embolism Detection` et `Auto Vulnerability Curve`.

Références

[1] https://parlonssciences.ca/sites/default/files/styles/large/public/2019-10/stomate_ouvert_et_ferm%C3%A9.png?itok=DvaiE44J.

[2] Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(8) :630–632, 1978.

[3] Diane Bienaime. *Embolie dans les plantes : dynamique de l'invasion d'air dans des réseaux hydrauliques naturels et artificiels sous pression négative*. Theses, Université Grenoble Alpes, October 2016.

[4] Tim Brodribb, Diane Bienaimé, and Philippe Marmottant. Revealing catastrophic failure of leaf networks under stress. *Proceedings of the National Academy of Sciences*, 113 :201522569, 04 2016.

[5] Brendan Choat, Timothy J. Brodribb, Craig R. Brodersen, Remko A. Duursma, Rosa Ana Lopez Rodriguez, and Belinda Medlyn. Triggers of tree mortality under drought. *Nature*, 558(7711) :531–539, 2018.

[6] Efbrazil. https://upload.wikimedia.org/wikipedia/commons/d/db/Global_Temperature_And_Forces.svg, 2020.

[7] INRAE. PHENOBOIS - Cavitation. https://www6.inrae.fr/phenobois/Expertises-Technologies/Cavitation, 2020.

[8] C. Li and P. Tam. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognit. Lett.*, 19 :771–776, 1998.

[9] NASA's Scientific Visualization Studio, Key and Title by uploader (Eric Fisk). https://commons.wikimedia.org/wiki/File:Change_in_Average_Temperature.svg, 2020.

[10] W. Tsai. Moment-preserving thresholding : a new approach. 1985.