Proximal sensing and neural network processes to assist in diagnosis of multi-symptom grapevine diseases

Malo Tardif

▶ **To cite this version:**

Malo Tardif. Proximal sensing and neural network processes to assist in diagnosis of multi-symptom grapevine diseases. Automatic Control Engineering. Université de Bordeaux, 2023. English. NNT : 2023BORD0369 . tel-04438748

HAL Id: tel-04438748
https://theses.hal.science/tel-04438748

Submitted on 21 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR

DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE SCIENCES POUR L'INGÉNIEUR

AUTOMATIQUE, PRODUCTIQUE, SIGNAL ET IMAGE, INGÉNIERIE COGNITIQUE

Par **Malo TARDIF**

Proximal sensing and neural network processes to assist in diagnosis of multi-symptom grapevine diseases

Proxidétection et approches neuronales pour l'aide au diagnostic de maladies multi-symptômes de la vigne

Sous la direction de: **Jean-Pierre DA COSTA**

Soutenue le 7 décembre 2023

Membres du jury :

M. Jean-Pierre DA COSTA	Professeur	Bordeaux Sciences Agro	Directeur
M. Adel HAFIANE	Maître de conférences	INSA Centre Val de Loire	Rapporteur
M. Luiz Angelo STEFFENEL	Professeur	Université de Reims	Rapporteur
Mme Véronique BELLON-MAUREL	IPEF	INRAE Montpellier	Examinatrice
M. François DELMOTTE	Directeur de recherche	INRAE Bordeaux-Aquitaine	Président du jury

Membres invités :

M. Marc GREVEN	Maître de conférences	Bordeaux Sciences Agro	Co-encadrant
M. Damian MARTIN	Docteur	Plant and Food Research	Invité

Acknowledgments

I would like to express my sincere gratitude to the reviewers for accepting to assess and evaluate my thesis work, as well as to the members of the jury for their presence during my defense.

A special thanks goes to my thesis supervisor, Jean-Pierre Da Costa, and my co-supervisor, Marc Greven, for giving me the opportunity to contribute to this exciting scientific project and for their guidance throughout these three years of research.

I also want to extend my appreciation to the members of the IMS and PFR laboratories with whom I had the privilege to collaborate during my thesis, including Ahmed, Barna, Aymeric, Paul, Dion, Damian, and many others, for their invaluable assistance and warm welcome.

I wish to express my gratitude to Bordeaux Sciences Agro and The Marlborough Research Centre, without whose support, this thesis would not have been possible. Additionally, I extend my appreciation to the teams of GDON, BNIC, and IFV for their valuable contributions to this research.

I would like to express my sincere gratitude to the owners and managers of the vineyards we visited for their cooperation.

Finally, a big thanks to all those who contributed to improving the quality of my manuscript through their meticulous reviewing efforts, as well as to my family and friends who supported me throughout this journey.

Your support and collaboration have been crucial to the success of this research work. Thanks, from the bottom of my heart.

List of abbreviations frequently used throughout the manuscript

BD: Botryosphaeria Dieback

CNN: Convolutional Neural Network

ED: Eutypa Dieback

FCN: Fully Connected Network

FD: Flavescence Dorée

GNN: Graph Neural Network

GTDs: Grapevine Trunk Diseases

IOU: Intersection Over Union

MLP: MultiLayer Perceptron

MP: Message Passing

p: precision

r: recall

RF: Random Forest

RGB: Red Green Blue

Proximal sensing and neural network processes to assist in diagnosis of multi-symptom grapevine diseases

Abstract:

Grapevine is a plant susceptible to numerous diseases. Some of these diseases can lead to significant yield losses and the death of the infected grapevine. Among these diseases, some present symptoms of different nature on various organs of the same vine. Their diagnosis, typically performed by experts, is even more complex as many confounding factors are present. This research focuses on the development of methodologies for acquiring, annotating, and processing data related to multi-symptom grapevine diseases to study their automated diagnosis. Two groups of diseases are targeted: grapevine yellows such as Flavescence dorée (FD) and grapevine trunk diseases (GTDs) with Eutypa and Botryosphaeria diebacks as specific diseases.

RGB image acquisitions were conducted directly in grapevine rows to build datasets for each disease type. The dataset for FD covers five different grape varieties and takes into account many diseases that have symptoms similar to FD, referred to as confounding diseases. The GTDs dataset includes images of a single grape variety and no confounding disease. Three methods for the automatic diagnosis of these diseases are proposed, compared, and discussed. The first method, inspired by state-of-the-art techniques, uses a convolutional neural network-based classifier applied to raw images (method A). The results show that this methodology delivers good results on datasets containing very few confounding diseases. Precision (p) and recall (r) of (p=0.94, r=0.92) are achieved for classifying images of grapevines affected by GTDs, while they are (p=0.87, r=0.84) for classifying images of vines affected by FD in a dataset containing 16% of confounding disease images.

To improve these results, two methods were developed, both consisting of two steps: (1) individual symptom detection using a detection algorithm composed of neural convolutional layers and a neural segmentation algorithm; (2) diagnosis based on the association of detected symptoms, either using a Random Forest classifier (method B) or a graph neural network (method C). The results of these two methodologies on the dataset containing 16% of confounding disease images for FD are (p=0.86, r=0.90) for method B and (p=0.90, r=0.96) for method C. These results demonstrate the better effectiveness of two-step methodologies in distinguishing confounding diseases from targeted diseases. They also highlight the relevance of embedded RGB imaging combined with artificial intelligence approaches for diagnosing these diseases.

Finally, these three methods are tested on whole-block acquisitions to establish their validity in real-world use cases. The results highlight the advantages of the two-step methodology based on symptom association by graph, the significant contribution of considering the surrounding vines and both sides of the vines during their automated diagnosis, and emphasize the challenges of real-world application of these methodologies.

Keywords: proximal sensing**,** computer vision, deep learning, graph neural networks, image processing, grapevine diseases.

Search unit:

IMS, UMR 5218 CNRS, Univ. Bordeaux – Bordeaux INP, 33405 Talence, France.

Proxidétection et approches neuronales pour l'aide au diagnostic de maladies multi-symptômes de la vigne

Résumé :
La vigne est une plante sujette à de très nombreuses maladies. Certaines de ces maladies peuvent entraîner d'importantes pertes de rendement et la mort du pied de vigne infecté. Parmi ces maladies, certaines présentent des symptômes de nature différente, sur plusieurs organes de la même vigne. Leur diagnostic, réalisé en pratique par des experts, est d'autant plus complexe qu'elles présentent de nombreux facteurs confondants. Cette recherche se concentre sur le développement de méthodologies pour l'acquisition, l'annotation et le traitement des données liées aux maladies de la vigne multi-symptômes afin d'étudier leur diagnostic automatique. Deux groupes de maladies sont ciblés : les jaunisses de la vigne telles que la Flavescence dorée (FD) ou le bois noir, et les maladies du bois de la vigne (GTDs) avec l'Eutypiose et le Botryosphaeria comme maladies ciblées.

Des acquisitions d'images RGB, directement dans les rangs de vigne, ont été réalisées pour construire des ensembles de données pour chaque type de maladie. Le jeu de données pour la FD couvre cinq cépages différents et tient compte de nombreuses maladies présentant des symptômes similaires à ceux de la FD, appelées maladies confondantes. L'ensemble de données pour les GTDs comprend des images d'une seule variété de vigne et aucune maladie confondante. Trois méthodes de diagnostic automatique de ces maladies sont proposées, comparées et discutées. La première méthode, inspirée de l'état de l'art, utilise un classifieur basé sur un réseau neuronal convolutif appliqué aux images brutes (méthode A). Les résultats montrent que cette méthodologie délivre de bons résultats sur les ensembles de données contenant très peu de maladies confondantes. Une précision (p) et un rappel (r) de (p=0,94 ; r=0,92) sont obtenus pour la classification des images de vigne affectées par les GTDs, tandis qu'ils sont de (p=0,87 ; r=0,84) pour la classification des images de vignes affectées par la FD sur un ensemble de données contenant 16% d'images de maladies confondantes.

Dans le but d'améliorer ces résultats, deux méthodes ont été développées, toutes deux comportant deux étapes : (1) la détection individuelle des symptômes à l'aide d'un algorithme de détection constitué de couches de neurones convolutives et d'un algorithme neuronal de segmentation ; (2) le diagnostic basé sur l'association des symptômes détectés, soit à l'aide d'un classificateur de type forêt d'arbres décisionnels ou Random Forest (méthode B), soit à l'aide d'un réseau neuronal sur graphe (méthode C). Les résultats de ces deux méthodologies sur l'ensemble de données contenant 16% d'images de maladies confondantes à la FD sont de (p=0,86 ; r=0,90) pour la méthode B et de (p=0,90 ; r=0,96) pour la méthode C. Ces résultats démontrent une meilleure efficacité des méthodologies en deux étapes pour distinguer les maladies confondantes des maladies ciblées. Ils démontrent également la pertinence de l'imagerie RGB embarquée associée aux approches neuronales d'intelligence artificielle pour le diagnostic de ces maladies.

Enfin, ces trois méthodes sont testées sur des acquisitions à l'échelle de parcelles entières afin d'établir leur validité dans des cas d'utilisation concret. Les résultats mettent en évidence les avantages de la méthodologie en deux étapes basée sur l'association des symptômes par graphe, la contribution significative des vignes environnantes et des deux faces des vignes lors de leur diagnostic automatique et soulignent les défis de l'application réelle de ces méthodologies.

Mots-clés : proxidétection, vision par ordinateur, apprentissage profond, réseaux de neurones sur graphes, traitement d'image, maladies de la vigne.

Unité de recherche :

IMS, UMR 5218 CNRS, Univ. Bordeaux – Bordeaux INP, 33405 Talence, France.

Table of content

Introduction générale

Les vignes sont sensibles à divers ravageurs et maladies pouvant avoir un impact significatif sur leur croissance, leur productivité et leur santé globale. Ces maladies peuvent provoquer une diminution de la qualité et du rendement, et potentiellement conduire à la mort de la vigne atteinte. La sensibilité des vignes à ces ravageurs et maladies dépend de divers facteurs, tels que la variété de vigne, les conditions environnementales, les pratiques culturales et la présence de ravageurs ou de maladies spécifiques dans une région donnée. Les maladies de la vigne peuvent être transmises par différents vecteurs ou agents causaux, notamment des champignons, des bactéries et des insectes.

En raison de cette pression liée aux maladies et aux ravageurs, la viticulture fait l'objet de nombreuses critiques, en particulier en ce qui concerne son impact sur l'environnement. En 2003, les vignobles représentaient 3 % des terres cultivées en Europe, mais étaient responsables de l'application de 13 % (en masse) de tous les pesticides synthétiques en Europe (Muthmann & Nadin, 2007). Bien que la viticulture ne représente que 3 % des terres agricoles en France, le secteur utilise 20 % des fongicides du pays (Robert, 2019).

La viticulture est confrontée à des défis majeurs. Cependant, grâce à l'adoption de pratiques durables et à la recherche continue, l'industrie vinicole s'efforce de s'adapter à ces défis afin de garantir sa durabilité. Face à l'obligation de réduire l'utilisation de pesticides nocifs pour l'environnement et notre santé, il existe un réel besoin d'adapter les mesures actuelles de protection de la vigne. La détection efficace et ciblée des problèmes phytosanitaires est cruciale pour le contrôle efficace des ravageurs et des maladies dans les vignobles. L'inspection des vignobles contribue à une viticulture durable en minimisant la nécessité de pulvérisations préventives systématiques aux profits d'interventions ciblées. La réalisation d'inspections fréquentes dans tous les vignobles tout au long de l'année représente un défi important pour les experts en maladies de la vigne. Pour résoudre ce problème, le développement de dispositifs de détection numérique, combinés à des outils de prise de décision, pourrait s'avérer d'une grande aide pour organiser les surveillances. De tels outils faciliteraient la priorisation des parcelles nécessitant une attention particulière, permettant ainsi aux experts de se concentrer sur ces zones et d'inspecter uniquement une partie des parcelles où les vignes sont potentiellement affectées par les maladies. En rationalisant le processus d'inspection, ces outils permettraient d'économiser du temps aux experts de manière considérable et leur permettraient de réaliser plusieurs surveillances consécutives dans un plus grand nombre de vignobles.

Les avancées technologiques récentes, notamment l'amélioration des dispositifs d'acquisition données, la disponibilité d'ordinateurs embarqués toujours plus puissants, les progrès sans précédent de l'intelligence artificielle et l'utilisation généralisée de drones, ont considérablement impulsé la recherche scientifique dans ce domaine. Notamment, plusieurs études ont démontré l'efficacité du processus de détection automatique des maladies de la vigne présentant des symptômes distinctifs (Bendel *et al.*, 2020 ; Boulent *et al.*, 2019 ; Kerkech *et al.*, 2018 ; Nguyen *et al.*, 2021). Cependant, des vides critiques existent en ce qui concerne la détection automatique des maladies manifestant plusieurs types de symptômes de manière simultanée et la prise en compte des facteurs confondants.

L'association des symptômes est souvent essentielle pour établir un diagnostic précis des maladies multi-symptômes de la vigne car de nombreuses maladies peuvent présenter des symptômes similaires. Par exemple, le diagnostic de la Flavescence dorée (FD) repose sur l'observation de la présence simultanée de symptômes sur feuilles, rameaux et grappes de la même vigne. Le diagnostic de l'Eutypiose (ED) et du dépérissement à Botryosphaeria (BD) est basé sur l'examen du tronc et du feuillage de la vigne. Actuellement, il y a peu de considération et de compréhension de ces cas complexes. Améliorer le diagnostic des maladies multi-symptômes de la vigne, en particulier en tenant compte des défis posés par les changements de variétés de vigne et la présence de maladies aux symptômes similaires, est une préoccupation cruciale.

L'objectif général de cette étude est d'identifier, de développer et d'évaluer des méthodologies automatiques pour le diagnostic des maladies multi-symptômes de la vigne. Deux types de maladies multi-symptômes distinctes sont étudiés : les jaunisses de la vigne, en mettant l'accent sur la

Flavescence dorée et le Bois noir (Tessitori *et al.*, 2018) comme cas d'études, et les maladies du bois de la vigne, avec le dépérissement à Botryosphaeria et l'Eutypiose comme maladies sélectionnées. Cette étude comporte quatre objectifs spécifiques :

1. Proposer un protocole complet intégrant l'imagerie RGB (Rouge-Vert-Bleu), la collecte de données de référence sur le terrain et la réalisation de divers types d'annotations à l'échelle de la plante et à l'échelle des symptômes.
2. Développer des chaînes innovantes d'algorithmes d'intelligence artificielle pour l'analyse automatisée des données liées aux maladies multi-symptômes.
3. Effectuer une évaluation de ces méthodologies en utilisant des données reflétant fidèlement les conditions des vignobles, garantissant ainsi leur applicabilité dans des scénarios d'application réels. Ces évaluations comprendront des tests sur diverses variétés de vignes, prendront en compte les symptômes confondants et évalueront la viabilité des méthodologies développées pour le diagnostic de parcelles entières.
4. Fournir une solution pour identifier les zones à haut risque de présence de ces maladies afin d'aider les efforts de prospection.

Le chapitre I présentera une introduction étendue du sujet de recherche. Le chapitre II fournira une revue des méthodologies existantes pour le diagnostic automatique des maladies de la vigne trouvées dans la littérature. L'identification des méthodologies prometteuses et la mise en lumière de lacunes dans les précédentes études permettront pour la sélection des approches et des bases de données proposées dans les chapitres suivants. Dans le chapitre III, les maladies étudiées ainsi que les données utilisées pour le développement, les tests et la comparaison des différentes méthodologies pour le diagnostic automatique de ces maladies seront présentées. Une explication du système et du protocole d'acquisition et d'annotation des données y sera présentée. Les trois chapitres suivants se concentreront sur le développement, l'évaluation et la comparaison des algorithmes de diagnostic automatique des maladies de la vigne. Tout d'abord, dans le chapitre IV, l'évaluation de la méthodologie standard en une étape basée sur les réseaux de neurones convolutifs (CNNs) sera effectuée sur un ensemble de données complet, mettant en évidence ses limitations et la nécessité de détecter précisément les différents symptômes. Dans le chapitre V, des algorithmes de détection de symptômes seront ainsi proposés et évalués. Les résultats mettront en évidence les défis liés à la tâche et la nécessité de les associer, comme le font les experts en maladies de la vigne sur le terrain, afin de délivrer le bon diagnostic. Deux méthodologies pour l'association des détections de symptômes seront présentées, évaluées et comparées dans le chapitre VI. La première méthodologie repose sur une approche d'apprentissage automatique plutôt classique, tandis que la seconde est une méthode d'apprentissage profond utilisant des données représentées sous forme de graphes, largement utilisée dans de nombreux domaines mais jamais adaptée à la détection de maladies sur cultures. Dans le chapitre VII, une évaluation des méthodes développées sera effectuée à l'échelle de sites entiers, simulant une mise en pratique de ces méthodes dans des cas d'utilisation réels pour le diagnostic automatique des maladies multi-symptômes de la vigne.

Résumé étendu

Le travail de recherche présenté dans ce mémoire propose de répondre à la problématique suivante : comment diagnostiquer efficacement les maladies de la vigne multi-symptômes par imagerie numériques en tenant compte des conditions réelles rencontrées au sein des vignobles ?

Quatre hypothèses guidant cette recherche
Une revue de littérature des précédentes études sur ce sujet a permis de formuler quatre hypothèses qui ont guidé ce travail.

1. La proxidétection semble être la méthode d'acquisition de données optimale pour identifier les différents symptômes exprimés par les vignes malades. De plus, il semble que l'imagerie RVB soit appropriée pour visualiser correctement les symptômes, pouvant être distingués par leurs couleurs et leurs formes.
2. Les techniques d'apprentissage profond semblent adaptées et prometteuses pour le traitement automatique des images et le diagnostic à l'échelle de la vigne.
3. L'approche traditionnelle utilisée dans les recherches précédentes sur ce sujet, fournissant un diagnostic en une seule passe sur les images, ne semble pas en mesure d'appréhender toute la complexité des maladies multi-symptômes, notamment en présence de maladies confondantes. Les approches en deux étapes, consistant en une première étape dédiée à la détection et à la distinction des symptômes unitaires, suivie d'une deuxième étape dédiée à la combinaison de ces détections, devraient permettre un diagnostic plus fiable des maladies multi-symptômes à l'échelle du pied de vigne.
4. Lors du diagnostic de l'état phytosanitaire d'une vigne par imagerie embarquée, tirer parti de la redondance d'information entre positions voisines (sur le rang) ou opposées (de part et d'autre du rang) permettrait d'améliorer le diagnostic à l'échelle de la vigne ciblée.

Maladies ciblées
Pour traiter ce sujet de recherche et évaluer ces quatre hypothèses, deux types de maladies multi-symptômes de la vigne ont été ciblés, au cœur de deux projets distincts auxquels cette recherche a contribué. Premièrement, la Flavescence dorée (FD), appartenant à la catégorie des maladies de jaunisse de la vigne, est au cœur du projet français ProspectFD qui vise à proposer des outils d'aide à la décision pour la prospection de la FD. Cette maladie a un impact significatif en Europe et présente des symptômes à la fois sur les feuilles, rameaux et grappes. La présence simultanée de ces symptômes est nécessaire pour différencier la FD de ses maladies confondantes sur le terrain. Cette maladie se caractérise par une décoloration des feuilles (vers le rouge pour les variétés de vigne rouge et vers le jaune pour les variétés de vigne blanche), la non-lignification des rameaux et le dessèchement des grappes. Un deuxième type de maladie multi-symptômes a été étudié : les maladies du bois de la vigne (GTDs), en relation directe avec le programme Vineyard Ecosystems se déroulant en Nouvelle-Zélande, au laboratoire Plant and Food Research du centre de recherche de Marlborough. Les dépérissements par Botryosphaeria et Eutypiose ont été les maladies ciblées. Ces deux maladies présentent trois symptômes différents, cependant, contrairement à la FD, leur présence simultanée n'est pas essentielle pour le diagnostic. La croissance entravée des rameaux (nommée « short shoot »), la croissance de la végétation d'un seul côté du tronc (nommée « half head ») et les chancres sont les symptômes de ces maladies.

Acquisition des données
L'établissement de bases de données spécifiques à ces deux types de maladies a été la première étape de ces travaux. En ce qui concerne l'étude de la FD, 1483 images ont été acquises à l'échelle de la vigne, englobant 5 variétés de vigne différentes (Cabernet sauvignon, Cabernet franc et Merlot pour les variétés rouges, Ugni blanc et Sauvignon blanc pour les variétés blanches). Parmi ces images, 754 présentaient des vignes atteintes par la FD, tandis que les autres présentaient des vignes avec des

symptômes pouvant être confondus avec ceux de la FD, en particulier des symptômes foliaires. Ces derniers ont été classés en trois catégories différentes : 'Esca' pour les vignes affectées par l'Esca, 'CONF' pour les vignes présentant des feuilles visuellement différentes des feuilles saines, et 'CONF+' pour les vignes présentant des symptômes foliaires presque identiques à ceux de la FD.

En ce qui concerne l'étude des GTDs, un total de 10 305 images ont été acquises. Parmi celles-ci, 2 830 images présentent des vignes affectées par les GTDs, et 7 475 présentent des vignes saines. Malheureusement, les parcelles étudiées lors de la collecte de données ne présentaient pas de maladies confondantes avec les GTDs. De plus, les vignes photographiées sont uniquement de la variété Sauvignon Blanc, la variété prédominante dans la région d'acquisition. Cette seconde base de données ne permet donc pas l'étude de l'impact des maladies confondantes, ni celle de la variation de l'expression des symptômes en fonction de la variété de vigne, sur le diagnostic automatique. Cependant, elle permet de comparer les résultats entre une base de données contenant des maladies confondantes et plusieurs cépages (celle liée à la FD) et une base de données sans maladies confondantes et ne contenant qu'un seul cépage (celle liée aux GTDs).

Diagnostic en 1 étape utilisant les réseaux de neurones convolutifs

Les résultats initiaux du diagnostic à l'échelle de la vigne ont été obtenus grâce à des réseaux de neurones convolutifs (CNNs). Cette méthode en une étape, classifiant directement l'image brute, est l'approche la plus couramment utilisée dans les recherches sur le diagnostic automatique des maladies de la vigne (Boulent *et al.*, 2020 ; Ji *et al.*, 2020 ; B. Liu *et al.*, 2020). Le test de cette méthodologie sur les ensembles de données d'images des deux types de maladies avait deux objectifs principaux : i) obtenir de premiers résultats de diagnostic automatique des maladies ciblée et ii) confirmer ou réfuter l'hypothèse selon laquelle cette méthodologie ne permet pas une distinction efficace de maladies présentant des symptômes similaires. Trois architectures de CNN différentes, parmi les plus rependues, ont été testées (MobileNetV3-large, ResNet50 et EfficientNetB5), ainsi que différents pré-traitements des images d'entrée pour ces réseaux. Au cours de l'étude pour le diagnostic automatique de la FD, des tests ont été effectués en regroupant les différents cépages rouges d'un côté et les cépages blancs de l'autre. Le premier ensemble contenait des images de Cabernet Sauvignon acquises en 2020 et 2021, de Cabernet franc et de Merlot acquises en 2021, avec 16 % des images de la classe 'CONF+'. En comparaison, l'ensemble de données de cépages blancs ne contenait que 3 % des images de la classe 'CONF+' et comprenait des images de cépages Ugni Blanc acquises en 2020 et 2021, et de Sauvignon Blanc acquises en 2021. Les meilleurs résultats pour la classification des images de la classe 'FD' pour les cépages rouges sont, en précision (p) et rappel (r), de (p=0,87 ; r=0,84), tandis que ceux pour les variétés de vigne blanche sont de (p=0,97 ; r=0,96). Ces résultats semblent démontrer les moins bonnes performances des CNNs dans la distinction entre la FD et ses maladies très confondantes. Ils semblent également confirmer leurs efficacités affirmées dans la littérature pour le diagnostic de la FD en l'absence de maladies confondantes ainsi que leur capacité à considérer efficacement les symptômes de la FD sur des vignes de différents cépages.

L'étude des CNNs pour le diagnostic automatique des GTDs a également été réalisée. L'ensemble de données testé comprenait 4 128 images, dont 2 212 montraient des vignes affectées par GTDs. Le CNN ResNet50 obtient les meilleurs résultats, à savoir (p=0,94 ; r=0,92) pour la classification des images de vignes affectées par les GTDs. Ce résultat confirme avec un autre type de maladie que les CNNs fournissent d'excellents résultats pour le diagnostic automatique des maladies de la vigne lorsque des maladies confondantes ne sont pas prises en compte.

Afin d'obtenir un meilleur diagnostic automatique des maladies et d'améliorer leur différenciation par rapport aux maladies confondantes, l'hypothèse émise était qu'une approche en deux étapes, avec une première étape dédiée uniquement à la détection des symptômes et une deuxième étape dédiée à leur association, serait bénéfique. L'étude menée après celle sur les CNNs s'est alors concentrée sur la détection automatique des symptômes des maladies ciblées.

Détections des symptômes unitaires

Afin de détecter automatiquement les symptômes, certaines des images de FD et de GTDs ont été annotées à l'échelle des symptômes. Deux types d'annotations, correspondant au mieux aux propriétés des symptômes étudiés, ont été réalisés. Concernant la FD, les feuilles symptomatiques de FD (classe 'feuille FD'), d'Esca (classe 'feuille Esca'), celles visuellement différentes des feuilles saines et similaires à celles de la FD (classe 'feuille CONF') ont été annotées avec des boîtes englobantes sur 744 images. Ces annotations ont donné lieu à 11 279 feuilles de la classe 'feuille FD', 2 467 de la classe 'feuille Esca' et 22 421 de la classe 'feuille CONF'. Les rameaux et les grappes symptomatiques de FD ont été annotées à l'aide de masques de segmentation. 128 masques de segmentation ont été créés, englobant 833 rameaux symptomatiques et 468 grappes symptomatiques. De plus, 660 grappes saines ont également été annotées sur ces masques de segmentation, leur présence en grand nombre sur une image pouvant être une preuve de l'absence de la maladie. Pour les GTDs, 505 images ont été annotées par boîtes englobantes. Parmi celles-ci, 268 symptômes de type 'Half-head', 1 480 symptômes de type 'Short shoot' et 139 symptômes de type 'Chancre' ont été annotés de cette manière. Des masques de segmentation ont également été créés pour les GTDs, mettant en évidence la distinction entre le tronc de la vigne étudiée (classe 'Tronc'), son feuillage (classe 'Feuillage') et le reste (classe 'Arrière-plan'). L'objectif était d'extraire des informations potentiellement caractéristiques de cette segmentation, telles que la différence de taille et de feuillage de certaines pousses (caractéristique du symptôme « short shoot ») ou de feuillage en contact avec un seul côté du tronc (caractéristique du symptôme « half head »). À cet égard, 113 images de GTDs ont été annotées avec des masques de segmentation comprenant ces trois classes.

La détection des feuilles symptomatiques de FD a obtenu ses meilleurs résultats avec l'algorithme de détection YOLOv8m, avec (p=0,78 ; r=0,61) pour les cépages rouges et (p=0,70 ; r=0,48) pour les cépages blancs. Les détections des rameaux et des grappes symptomatiques de FD ont obtenu de meilleurs résultats avec l'algorithme ResUNet, à savoir (p=0,81 ; r=0,67) pour les rameaux et (p=0,84 ; r=0,69) pour les grappes. En ce qui concerne la détection des symptômes des GTDs, le modèle YOLOv8m a obtenu les meilleurs résultats, avec (p=0,58 ; r=0,58) pour la classe 'Short shoot', (p=0,85 ; r=0,85) pour la classe 'Half head' et aucune détection pour la classe 'Chancre'. La segmentation entre le tronc, le feuillage et le reste de d'image a obtenu comme meilleurs résultats (p=0.86 ; r=0.95) pour la classe 'Tronc', (p=0.93 ; r=0.92) pour la classe 'Feuillage' et (p=0.97 ; r=0.97) pour la classe 'Arrière-plan'.

Ces résultats, bien que pouvant être espérés meilleurs, doivent être considérés à la lumière de la complexité de la tâche. Cependant, ces résultats ont pu servir de base pour le développement de la deuxième étape de la méthodologie, qui a concentré une grande partie des efforts de cette recherche, à savoir l'association de ces détections afin de délivrer le diagnostic final à l'échelle de la vigne. Ces méthodologies d'association de détections ont été exclusivement développées et testées pour la maladie FD, pour laquelle le nombre de symptômes détectés par image et le nombre d'images de la classe 'CONF+' étaient substantiels. Cet ensemble de données s'est révélé essentiel pour déterminer si les méthodologies en deux étapes permettaient d'obtenir de meilleurs résultats dans la distinction entre les vignes affectées par la maladie ciblée de celles affectées par des maladies présentant des symptômes très confondants.

Méthodologies d'association des symptômes détectés

Deux méthodologies différentes d'association de symptômes ont été testées : la première utilise un vecteur d'informations de taille fixe pour chaque image, basé sur les symptômes détectés, et le classe avec un classifieur de type Random Forest (RF). La deuxième approche représente les détections de symptômes de chaque image sous la forme d'un graphe, classé par la suite par un réseau neuronal sur graphe (GNN). Pour l'ensemble de données de cépages rouges, contenant 16 % d'images de la classe 'CONF+' et sur lequel les CNNs ont montré des performances perfectibles (p=0,87 ; r=0,84), les méthodologies RF et GNN ont permis d'améliorer de façon significative les résultats de classification avec respectivement (p=0,86 ; r=0,90) et (p=0,9 ; r=0,96). Pour l'ensemble de données des cépages blancs, contenant 3 % d'images de la classe 'CONF+' et sur lequel les CNNs ont obtenu leurs meilleures

performances avec (p=0,97 ; r=0,96), les méthodologie RF et GNN ont obtenus des résultats équivalents voire en léger retrait avec respectivement (p=1 ; r=0,94) et (p=0,97 ; r=0,90). Par conséquent, les méthodes d'association de symptômes ont donné de meilleurs résultats sur l'ensemble de données lorsque la proportion d'images de la classe 'CONF+' était substantielle (ensemble de données des cépages rouges). Lorsque cette proportion était très faible (cépages blancs), les résultats n'étaient pas meilleurs que ceux obtenus par les CNNs. Ces résultats de classification d'images, à la vue des résultats de détection de symptômes sur lesquels reposent les méthodes de classification, sont très prometteurs. Ils semblent indiquer que le choix d'acquérir des données par proxidétéction en utilisant l'imagerie RGB, détecter puis associer les symptômes à l'aide d'algorithmes d'apprentissage profond, sont des choix pertinents pour le diagnostic automatique des maladies multi-symptômes de la vigne.

Diagnostic à l'échelle de sites entiers

Enfin, l'efficacité des CNNs et des méthodologies en deux étapes a été évaluée lors du diagnostic à l'échelle d'un site entier (i.e. acquisition continue des deux faces de chaque rang de chaque parcelle), simulant un cas d'utilisation concret. Des acquisitions ont été menées sur trois sites distincts, chaque site ayant des caractéristiques particulières. Le site 'Neuffons' présente des vignes de cépage Merlot, et la plupart des vignes affectées par la FD présentent très peu de symptômes. Le site 'Cognac' présente des vignes de cépage Ugni Blanc et un grand nombre de vignes atteintes des jaunisses de la vigne sans être atteintes par la FD. Le site 'Couture', composé de vignes de cépage Cabernet Sauvignon, contient de nombreuses vignes atteintes par la FD exprimant de manière visible les symptômes de la maladie. Les méthodes CNN, RF et GNN ayant obtenu les meilleurs résultats mentionnés ci-dessus ont été testés pour classifier chaque image acquise sur ces sites. La méthodologie GNN a obtenu les meilleurs résultats sur les trois sites, bien que ces résultats ne soient pas toujours satisfaisants. Ils étaient de (p=0,25 ; r=0,26) pour le site 'Neuffons', (p=0 ; r=1) pour le site 'Cognac', et (p=0,96 ; r=0,22) pour le site 'Couture'.

Dans le but d'obtenir de meilleurs résultats, des méthodes RF et GNN utilisant les détections de symptômes effectuées sur les images à gauche et à droite de l'image à classifier (méthode des 3 voisins) ainsi que sur les trois images acquises sur l'autre face du rang (méthode des 6 voisins) ont été développées. Ces méthodologies avaient pour objectif de tirer au mieux parti des acquisitions denses réalisées à l'échelle de sites entiers. Ces méthodologies procèdent soit en créant un vecteur d'information à partir de toutes ces détections (méthodes RF 3 et 6 voisins), soit en créant un graphe composé de ces mêmes détections (méthodes GNN 3 et 6 voisins). La prise en compte des symptômes détectés sur l'autre face du rang (méthode des 6 voisins) permet : i) de contourner le cas pour lequel les symptômes d'une vigne ne seraient visibles que sur une seule face de celle-ci, et ii) de profiter de la redondance des détections dans le cas où les symptômes seraient visibles sur les 2 faces.

Des validations croisées intra-site ont été effectuées pour les CNNs et les méthodes RF et GNN. Ces tests visaient à déterminer la méthode la plus adaptée pour le diagnostic à l'échelle d'un site entier. La méthodologie GNN 6 voisins a obtenu les meilleurs résultats dans le diagnostic automatique de la FD pour chacun des trois sites. Ces résultats pour la classe 'FD' sont de (p=0,76 ; r=0,71) pour le site 'Neuffons', (p=0,36 ; r=0,93) pour le site 'Cognac', et (p=0,94 ; r=0,98) pour le site 'Couture'. Ces résultats mettent en évidence que la prise en compte des vignes environnantes ainsi que des deux faces d'une même vigne conduit à un meilleur diagnostic. Enfin, la méthodologie GNN 6 voisins a été testée sur chaque site en étant entrainée sur les images de l'un ou des deux autres sites, afin d'établir son efficacité dans un scénario réaliste où le modèle n'aurait jamais rencontré les images du site ciblé. Lors du test sur le site 'Neuffons', dont les vignes atteintes de FD présentent très peu de symptômes, le modèle, entrainé sur des vignes présentant de nombreux symptômes, n'a détecté aucun cas de FD. Le test sur le site 'Cognac' a donné comme meilleur résultats (p=0,08 ; r=0,18). La méthodologie n'arrive pas à différencier les pieds atteints de FD de ceux atteints de maladies confondantes en étant entrainée sur des sites contenant peu de maladies confondantes. Enfin, le test sur le site 'Couture',

dont les très nombreux pieds atteints de FD expriment bien les symptômes, a conduit à de très bons résultats (p=0,88 ; r=0,93).

Malgré la construction d'une base de données pluri-annuelles et multi-cépages, comportant de nombreuses expressions de symptômes différentes et la présence de symptômes confondants, les modèles entrainés sur cette base peinent à obtenir des résultats convaincants lors de leurs applications sur de nouveaux sites. Ces difficultés soulignent combien la constitution d'une base de données d'apprentissage consistante, embrassant une multiplicité de situations et de conditions d'acquisition, reste primordiale dans la perspective d'un passage à l'échelle. Les résultats obtenus dans des situations mieux maîtrisées car déjà rencontrés lors de la mise au point des modèles restent cependant prometteurs sur la capacité des approches à être appliquées en situation professionnelle.

Réponse à la problématique

Cette recherche s'est proposée de répondre à la question : "Comment diagnostiquer efficacement les maladies multi-symptômes de la vigne par des moyens numériques en tenant compte des conditions réelles rencontrées au sein des vignobles ?". La réponse, selon cette recherche, réside dans la création de la base de données la plus complète possible en termes d'expression des symptômes des maladies étudiées et de leurs maladies confondantes, la détection précise des symptômes et l'association intelligente de ces détections de symptômes. Tenir compte des symptômes détectés des deux côtés de la même vigne ainsi que sur les vignes voisines semble apporter un avantage significatif. Cependant, en plongeant dans la réalité complexe des vignobles, il devient évident que ce problème est très compliqué à résoudre. Créer une base de données et obtenir de bons résultats sur celle-ci est une chose, mais appliquer le modèle développé à un site jamais étudié auparavant en est une autre. Chaque vignoble semble avoir sa propre spécificité, sa propre expression des symptômes des maladies ciblées et des maladies confondantes, rendant le diagnostic sur des sites autres que ceux sur lesquels les algorithmes ont été entrainés compliqué. Malgré les efforts déployés pour construire une base de données solide et très diversifiée, avoir développé des méthodologies innovantes obtenant de très bons résultats sur cette base de données, les résultats obtenus lors de la mise en pratique de ces méthodologies sont décevants. Cela semble indiquer qu'un manque persiste dans la variété d'expression de symptômes de la maladie ciblée et de ses maladies confondantes considéré lors de l'entrainement des méthodes développées.

Position par rapport aux recherches antérieures

Cette recherche confirme les conclusions d'études antérieures sur ce sujet, à savoir que l'utilisation de CNNs permet d'obtenir de très bons résultats dans le diagnostic automatique des maladies de la vigne lorsqu'on ne considère que les vignes affectées par la maladie ciblée et les vignes saines (Boulent *et al.*, 2020 ; Ji *et al.*, 2020 ; B. Liu *et al.*, 2020). Cependant, elle se distingue de ces études précédentes en tenant également compte des maladies aux symptômes confondants lors du développement de modèles de diagnostic automatique. Lors de la considération de ces maladies confondantes, les CNNs ne sont plus les meilleurs modèles pour le diagnostic automatique, et les méthodologies innovante en deux étapes proposées dans cette étude délivrent de meilleurs résultats. Enfin, même en tenant compte de ces maladies très confondantes lors de l'entrainement des modèles, cette étude confirme la conclusion de Al Saddik (2019) et Albetis *et al*. (2018) selon laquelle, lors de l'application des modèles dans des cas d'utilisation réels, il est très difficile de distinguer les maladies multi-symptômes de la vigne de leurs maladies confondantes.

Limites de la recherche

En ce qui concerne les limites de cette recherche, davantage d'acquisitions à l'échelle de site entier auraient pu améliorer les modèles par graphe combinant des images multiples, qui ont montré les meilleurs résultats pour le diagnostic automatique de la FD. Bien que différentes architectures de CNN aient été testées comme méthodologie en une étape, les vision transformers (S. Khan *et al*., 2022), offrant une approche nouvelle et prometteuse pour le traitement des données visuelles différente des CNN traditionnels, auraient pu être testés. Une plus grande diversité d'images avec des symptômes

annotés et une étude plus approfondie des algorithmes de détection des symptômes auraient pu être menées pour obtenir de meilleurs résultats lors de leur association. En ce qui concerne les GTDs, des méthodes d'association des détections de symptômes auraient pu être développées pour comparer les résultats à ceux obtenus lors de l'étude de la FD. Une étude plus approfondie sur les maladies pouvant être confondues avec les GTDs, la prise en compte de plusieurs variétés de vigne et des acquisitions à l'échelle de sites entiers auraient pu être menées pour fournir une étude plus complète du diagnostic automatique des GTDs. Une étude plus approfondie des paramètres choisis concernant les méthodes RF (paramètres des vecteurs) et GNN (type de graphe, paramètres des nœuds) aurait pu être réalisée.

Conclusion générale

Le but principal de cette recherche était le développement et la comparaison de méthodes pour le diagnostic automatique des maladies multi-symptômes de la vigne. Afin de diagnostiquer au mieux ces maladies, cette recherche préconise tout d'abord une approche novatrice concernant les protocoles d'acquisitions de données et de la vérité terrain. Cette approche comprend l'intégration des maladies confondantes à la maladie ciblée dans les jeux de données, ainsi que des annotations à l'échelle des images et des symptômes des maladies ciblées et confondantes. Les données ont été collectées par imagerie RGB, directement dans les rangs de vignes, afin d'avoir la vue la plus précise possible des différents symptômes. La méthodologie de diagnostic automatique très largement répandue dans la littérature a été testée et a donné des résultats très satisfaisants dans la plupart des cas. Cependant, lorsque les jeux de données contenaient une grande proportion de vignes présentant des symptômes confondants à ceux de la maladie ciblée, les résultats de cette méthode devenaient moins bons. Dans le but d'améliorer les performances de différenciation entre maladie ciblée et maladies aux symptômes confondants, des méthodologies en 2 étapes ont été développées. La première étape se focalisant sur la détection des symptômes unitaires, réalisée par deux algorithmes neuronaux, un de segmentation et un de détection. La deuxième étape est dédiée à l'association de ces détections de symptômes. Leurs représentations sous la forme de graphes et leur traitement par graph neural network ont été, entre autres, proposés. Cette méthodologie innovante a permis d'améliorer la distinction entre maladie ciblée et maladies aux symptômes confondants. Les méthodologies développées ont par la suite été testée sur trois acquisitions réalisées à l'échelle de vignobles entiers. Cette recherche a pu démontrer l'efficacité supérieure d'approche prenant en compte les détections de symptômes obtenues sur un ensemble de vignes environnantes et leurs représentations par graphe pour le diagnostic de maladies à l'échelle de l'ensemble d'un vignoble. Ces innovations méthodologiques semblent avoir un potentiel très prometteur par leur capacité à tirer parti de la redondance de l'information dans les parcelles. Cependant, si pour un vignoble, les résultats sont très satisfaisants, ils sont décevants pour les 2 autres. Ces résultats soulignent toute la difficulté de la mise en pratique généralisée des méthodes de diagnostic automatique de maladies de la vigne. Chaque vignoble semble présenter des caractéristiques qui lui sont propre. Afin de couvrir toutes les spécificités pouvant être rencontrées, un changement d'échelle concernant le nombre d'acquisition et d'annotation des données d'entrainements des modèles semble nécessaire. Les méthodologies innovantes développées dans cette recherche, délivrant un diagnostic vigne par vigne uniquement à partir d'images, semblent constituer un premier pas dans l'amélioration de la détection des maladies dans les vignobles. Cependant, pour une meilleure identification des zones à haut risque de présence de maladies au sein des vignobles, il semble essentiel de développer un outil complet d'aide à la décision, prenant en compte des caractéristiques telles que l'historique des symptômes et l'environnement du vignoble, et adaptant des critères de diagnostic en conséquence.

I. Introduction

Table of content

A. Viticulture: a dynamic but threatened sector

a. Viticulture: a look ahead

Today, grapevine cultivation and winemaking are practiced in numerous countries across the globe. Famous wine-producing regions include France, Italy, Spain, the United States, Argentina, Australia, and South Africa, among many others. Each region has its own unique climate, soil, grape varieties, and winemaking traditions, resulting in a diverse range of wines (Charters, 2006; Leeuwen & Darriet, 2016).

In 2022, the global vineyard area was estimated to be around 7.3 million hectares (International Organisation of Grapevine and Wine, 2023), indicating the widespread cultivation of grapes for wine production and table grapes. Notably, Spain, France, China and Italy emerged as the countries with the most substantial vineyard surface areas, reinforcing their prominence in the wine industry.

The wine trade stands as a significant global enterprise, with exports amounting to several billion dollars annually. The worldwide wine consumption for 2022 reached an impressive 232 million hectolitres, highlighting the widespread appreciation of wine as a popular beverage choice.

The wine industry encompasses a range of activities, including grape cultivation, winemaking, marketing, distribution, and wine tourism, which collectively create employment opportunities worldwide. Wine tourism, in particular, has gained significant popularity, attracting travellers to explore iconic wine regions like Napa Valley in California, Tuscany in Italy, and Bordeaux in France

However, viticulture faces challenging times due to the impact of climate change. The undeniable reality of global warming, largely driven by human activities and greenhouse gas emissions, presents one of the most significant environmental hurdles humanity must address (IPCC, 2022). Rising temperatures have profound implications for grapevine life cycles, affecting key phenological stages such as bud burst, flowering, veraison, and maturity (Parker *et al.*, 2013). Consequently, the quality and reputation of renowned wine-growing regions may be influenced by these temperature changes (Gutiérrez-Gamboa *et al.*, 2021).

Moreover, the anticipated increase in extreme weather events, including heatwaves, frost events, storms, and hailstorms, can cause potential reductions in yield and degradation of wine quality, adding to the industry's challenges (Droulia & Charalampopoulos, 2021). As a result, viticulturists must adapt to shifting environmental conditions and reconsider their approaches to cultivating and preserving grapevines (Droulia & Charalampopoulos, 2022).

One notable consequence of rising temperatures is the potential emergence of new diseases and pests that affect grapevines. Known diseases may also undergo shifts in distribution due to changing climate conditions, affecting regions previously considered too cold or too warm for certain pathogens. Fungal pathogens like powdery mildew (Gadoury *et al.*, 2012) and downy mildew (Caffi *et al.*, 2013), which thrive in warm and humid environments, may become more prevalent as temperatures increase, further complicating grapevine management.

High temperatures and prolonged heat stress can weaken the defence mechanisms of grapevines, making them more vulnerable to disease. Heat stress can compromise the plant's ability to produce defence compounds and activate defence responses (Berry & Bjorkman, 1980), leaving it more susceptible to infections.

Additionally, insects that act as disease vectors (Reineke & Thiéry, 2016) can be affected by climate change, impacting disease dynamics and possibly leading to the emergence of new grapevine pests and diseases (Bocca *et al.*, 2020; Deutsch *et al.*, 2018). Changes in the phenology of grapevine disease vectors over the past three decades have already been observed, further highlighting the importance of understanding and managing these risks.

In conclusion, while viticulture plays a crucial role in the economy and culture of many countries worldwide, it faces significant challenges. The changing temperatures and their subsequent effects on grapevines as well as the sensitivity of grapevine to diseases and pests necessitate innovative and adaptive approaches to sustainably manage vineyards and preserve the wine industry for future generations.

b. Grapevine sensitivity to pests and diseases

Grapevines are sensitive to a variety of pests and diseases, which can significantly impact their growth, productivity, and overall health. These diseases cause a decrease in quality and yield, and can eventually lead to the death of the grapevine. The sensitivity of grapevines to these pests and diseases depends on various factors, including grapevine cultivar, environmental conditions and cultural practices. Grapevine diseases can be caused by fungi, viruses or bacteria that can be transmitted through different vectors like insects, birds, machinery and rain splatter.

Fungal diseases such as powdery mildew and downy mildew are common grapevine diseases. Powdery mildew is spread through airborne spores, while downy mildew is primarily transmitted by spores carried by water droplets from splashing water. Prevalence for both diseases is strongly favoured by high humidity. Downy Mildew is causing damage all over the world (Fontaine *et al.*, 2021).

Trunk diseases are a growing concern for grape growers worldwide, as, contrarily to mildews, the pathogens are difficult to control by means of pesticides (wounds can still be protected with them to stop infection). Many wine production regions around the world reported trunk diseases as major issues (Bois *et al.*, 2017). There are four major grapevine trunk diseases caused by different fungi. These diseases are called Esca (Larignon & Dubos, 1997) Eutypa dieback (ED) (Carter, 1991), Botryosphaeria dieback (BD) (Slippers & Wingfield, 2007) and Phomopsis dieback (Úrbez-Torres *et al.*, 2013). According to Fontaine *et al.* (2016), Spain had an estimated vineyard trunk diseases incidence of 10.5% in 2007. In France, about 13% of vineyards were unproductive due to trunk diseases, resulting in an estimated loss of €1 billion in 2014. Trunk diseases have caused in California losses of at least US $260 million annually. Esca infections represent a loss of about 2.000-3.000$ per hectare per year in California. A national survey of symptomatic material from 43 vineyards showed that 88% had some degree of infection by Botryosphaeriaceous species in New Zealand.

Apart from fungi, grapevines worldwide are also impacted by bacterial diseases, three of which are significant: bacterial blight (*Xylophilus ampelinus*) (Prunier *et al.*, 1970), Pierce's disease (*Xylella fastidiosa*) (Davis *et al.*, 1978), and crown gall (*Agrobacterium vitis*) (Burr & Otten, 1999). These bacteria thrive within the vascular system of the grapevine, allowing them to invade and inhabit the entire plant. The presence of latent infections in propagating material is thought to play a significant

role in the spread of these diseases (Szegedi & Civerolo, 2011). In Europe, Flavescence dorée (FD) (Lefol *et al.*, 1993) was classified as a quarantine disease in 1993 (European Directive 2000/29/EC) and is subject to mandatory reporting. It is caused by a phytoplasma, a type of bacteria-like organism, and is transmitted by leafhoppers *Scaphoideus titanus* Ball (Cicadellidae). The disease causes a decline in grape quality and yield. FD is a significant concern for grape growers and winemakers, as it can lead to severe economic losses and the destruction of vineyards if not properly managed through strict control measures and quarantine protocols. In 2018 in France, 75% of vineyards (i.e. 568,507 ha) is within the compulsory control perimeters defined in the prefectural decrees (Barthellet *et al.*, 2018). Oliveira *et al.* (2020) showed that infected plants showed reduction in the yield between 51% and 92%.

In addition to fungi and bacteria, viruses also pose a threat to grapevines. Nearly 70 different viruses have been identified in grapevines. They are characterized by a wide array of symptoms, e.g. malformations of leaves and twigs, foliar discolourations (reddening, yellowing, chlorotic or bright yellow mottling, ringspots, and line patterns), grooving and/or pitting of the woody cylinder, delayed bud break, stunting and dieback (Martelli, 2017). Grapevine leafroll-associated virus 3 (GLRaV3) (Naidu *et al.*, 2014) is primarily spread through propagation material and certain insects like mealy bugs (Pseudococcidae spp). Atallah *et al.* (2011) estimated that the economic impact of grapevine leafroll disease ranges from $25,000 to $40,000 per hectare over the 25-year lifespan of the studied vineyards in the USA.

It is important for grape growers to monitor their vineyards regularly, employ integrated pest management strategies, and take appropriate preventive measures to minimize the damage caused by pests and diseases. Local agricultural extension services or viticulture experts can provide region-specific information and recommendations for managing pest and disease issues in grapevines.

c. Environmental impact

Because of this disease pressure, viticulture is subject to a lot of criticisms, particularly with regard to its impact on the environment due to the high quantities of phytosanitary products being used.
Since 1850, pest and diseases and have spread like never before. The globalization of world trade, the expansion of the international market of seeds and crops and the increase of monoculture being the main causes. Grapevines are a very sensitive to pests and diseases, and the introduction of new pesticides during the second half of the 20th century have enabled pest and disease control and a significant increase in yield. The use of these chemical agents became a necessity for winegrower in order to meet yield and quality targets. In 2003, vineyards covered 3% of the European cropland, but were involved in the application of 13% (in mass) of all pesticides in Europe (Muthmann & Nadin, 2007). Although viticulture represents 3% of France's agricultural land, the sector uses 20% of the country's fungicides (Robert, 2019).

The advantages for the winegrowers are obvious. However, there is now overwhelming evidence that their intensive use has resulted in serious health implications to humans and their environment (Van der Werf, 1996). For example, copper-based fungicides, widely used against downy mildew, (even in organic vineyards) are responsible for the reduction of soil fertility and reduce the soils ability to filter contaminants (Keesstra *et al.*, 2012). As copper is highly persistent in soils (Babcsányi *et al.*, 2016; Fernández-Calviño *et al.*, 2009), it can seep into groundwater and affect the water quality and with it, aquatic organisms (Fernández-Calviño *et al.*, 2010; Mackie *et al.*, 2012). Secondly, the impact of any pesticides on human health is real. Vineyard workers are obviously the first in danger while spraying pesticides. Studies have demonstrated the effect on mental health (Fuhrimann *et al.*, 2022; N. Khan *et al.*, 2019) and an elevated sensitivity to illness and certain forms of cancer (Curl *et al.*, 2020) for people directly exposed to pesticides. However, these people are not the only one exposed to the pesticides. All wine consumers are exposed too. A study of 40 bottles coming from different countries in the world demonstrated that all the bottles coming from conventional blocks contained pesticides residues (Pesticide Action Network, 2008).

d. Adapting grapevine protection for a sustainable future

Viticulture is facing major challenges. However, through the adoption of sustainable practices and ongoing research, the wine industry is striving to adapt to these challenges and ensure its sustainability. Faced with this vision of the future of viticulture and the obligation to reduce the use of pesticides harmful to the environment and our health, there is a real need to adapt the current protection measures of the grapevine.

Many vineyards are adopting integrated pest management (IPM) strategies (Isaacs *et al.*, 2012; Perria *et al.*, 2022; Wilson & Daane, 2017). IPM involves using a combination of techniques such as biological control, cultural practices, and targeted pesticide application to minimize reliance on pesticides and reduce their negative impacts on the environment and human health. Sustainable viticulture practices aim to strike a balance between effective pest management and minimizing the use of pesticides, promoting the long-term health of both the vineyard and the surrounding ecosystem. In IPM, monitoring of the vineyard and correct pest identification assist in deciding whether control of the pest or disease is needed. Correctly identifying the pest and diseases is key to knowing whether a pest is likely to become a problem and determining the best management strategy. However, despite these measures, residues of pesticides in grapes that are equal or higher than the maximum residues levels, still can be found (Baša Česnik *et al.*, 2008). Breeding can also be part of a solution for the control of the current and future pests and diseases. Combining grapevine resistant to diseases with grapevines with great grape quality is an option (Bavaresco, 2019). Researches to identify resistant genotype to major diseases, as for example powdery and downy mildew (Buonassisi *et al.*, 2017; Merdinoglu *et al.*, 2018; Nardi *et al.*, 2019), are in progress. Organic farming may be a way to reduce the use of phytosanitary products. However, here again, it implies a very efficient monitoring of the blocks in order to prevent diseases from spreading (Merot *et al.*, 2020), which may turn catastrophic.

B. Effective vineyard inspections for sustainable viticulture

Efficient and targeted detection of phytosanitary problems is crucial for effective pest and disease control in vineyards. To achieve this, vineyard inspections play a vital role in promoting sustainable viticulture by minimizing the need for systematic preventive spraying and instead enabling precise interventions. While winegrowers themselves can contribute to inspections, certain diseases require inspection by experts. Grapevines are vulnerable simultaneously to various biotic and abiotic stresses, exhibiting diverse symptoms that can be very challenging to detect or differentiate. Furthermore, disease symptoms can vary based on the grapevine variety, phenological stage and even exhibit a random pattern of disappearance and reappearance from one year to another. Accurate diagnosis often necessitates meticulous inspection, association of symptoms across grapevine organs and DNA analysis (Hren *et al.*, 2007; Úrbez-Torres *et al.*, 2006). Early-stage disease symptoms, such as subtle discolouration (Christen *et al.*, 2007) and small leaf spots (Gessler *et al.*, 2011), are particularly elusive, underscoring the importance of precise and consistent inspections by experts to contain their spread promptly.

Conducting frequent inspections across all vineyards throughout the year poses a significant challenge for grapevine disease experts. To address this issue, the development of digital detection devices combined with decision support tools could prove invaluable in organizing surveys. Such tools would facilitate the prioritization of blocks that require attention, allowing experts to focus on those areas and inspect only a portion of the blocks where grapevines have been found to likely be affected by diseases. By streamlining the inspection process, these tools would save experts considerable time and enable them to conduct multiple consecutive surveys across more vineyards.

Recent technological advancements, including improved acquisition devices, powerful embedded computers, artificial intelligence, and the widespread use of drones or unmanned aerial

vehicles (UAVs), have significantly propelled scientific research in this field. Notably, several studies have demonstrated the effectiveness of automated detection process for grapevine diseases that exhibit distinctive symptoms (Bendel *et al.*, 2020; Boulent *et al.*, 2019; Kerkech *et al.*, 2018; Nguyen *et al.*, 2021). However, a critical gap exists when it comes to diseases that manifest multiple symptoms. The association of these symptoms is essential to establish an accurate diagnosis, as numerous diseases can display similar symptoms. For example, the diagnosis of FD relies on the observation of the simultaneous presence of symptoms on leaves, shoots and bunches on the same grapevine. The diagnosis of ED and BD is based on the examination of the grapevine's trunk and foliage. Currently, there is limited consideration and understanding of such complex cases. Improving the diagnosis of multi-symptom diseases in grapevines, especially considering the challenges posed by changes in grape varieties and the presence of similar diseases, is a crucial concern.

How to effectively diagnose grapevine multi-symptom diseases by digital means considering real vineyard conditions?

C. Objectives of the study

The general objective of this study is to identify, develop and evaluate automated methodologies for multi-symptoms grapevine diseases diagnosis. Two types of distinct multi-symptomatic diseases will be investigated: grapevine yellows, focusing on Flavesence dorée and Bois noir (Tessitori *et al.*, 2018) as case studies, and grapevine trunk diseases, with Botryosphaeria and Eutypa diebacks as the selected diseases studies. There are three specific objectives to this study:

- Propose a comprehensive protocol that integrates RGB imaging (Red-Green-Blue imaging. In this technique, each pixel of an image is represented by a combination of intensity values for these three colours), ground truth dataset collections and various types of annotations at both the plant and symptom scales.
- Develop innovative chains of artificial intelligence algorithms for the automated analysis of data related to multi-symptom diseases.
- Conduct an evaluation of these methodologies using data that accurately reflect the vineyard conditions, thus ensuring their applicability in real-world application scenarios. This evaluation will encompass testing on various grapevine varieties, include the consideration of confounding symptoms and assess their usefulness for the diagnosis of whole blocks.

D. Flow of the manuscript

Chapter II will provide a comprehensive review of existing methodologies for automated diagnosis of grapevine diseases found in the literature. The identification of promising methodologies and knowledge gaps in considering certain aspects will be used for the selection of approaches and databases proposed and developed in subsequent chapters. In Chapter III, the studied diseases as well as the data used for the development, testing, and comparison of the different methodologies for the automated diagnosis of the studied diseases will be presented. This will include an explanation of the data acquisition system and protocol as well as the annotated data (and annotation protocols). The subsequent three chapters will focus on the development, evaluation and comparison of the automated grapevine diseases diagnosis algorithms. Firstly, in Chapter IV, the evaluation of the standard CNN-based 1-step mainstream methodology will be conducted on a complete dataset, highlighting its limitations and the need of accurately detecting the symptoms. Subsequently, in Chapter V, symptom detection algorithms will be proposed and evaluated. Results will emphasize the challenges involved in the task and the need for the symptom detections association, as grapevine diseases expert are doing in the field to deliver the right diagnosis. Two methodologies for the association of symptom detections will be presented, evaluated and compared in Chapter VI. The first

methodology is based on a rather classical machine learning approach, whereas the second one is a deep learning method using data represented as graphs, widely used in many domains but never adapted to crop diseases detection. In Chapter VII, a comparison of the developed methods will be carried out on vineyard scale, simulating a practical implementation of these methods in real-use cases for the automated diagnosis of multi-symptom grapevine diseases.

II. Building up a strategy to address multi-symptom disease diagnosis

Table of content

A. Literature review on grapevine diseases automated detection

The in-situ characterization of crops using digital tools has resulted in an extensive body of literature that is challenging, and perhaps not entirely necessary, to synthesize within this thesis. This literature covers a wide array of crops (such as cereals (Pujari *et al.*, 2015), open-field/greenhouse vegetables (Cubero *et al.*, 2011), fruit trees, and horticulture (Usha & Singh, 2013)), various topics of interest (including phenology (Yalcin, 2017), health status (Fuentes *et al.*, 2018; Johannes *et al.*, 2017), and yield (Van Klompenburg *et al.*, 2020)), and a diverse range of technologies (involving the selection of carriers, sensors, and processing methods). While it is valuable to explore all available technological options, the focus was chosen to narrow to the specific issue of disease detection in perennial crops, with a particular emphasis on viticulture. This field poses unique questions and challenges, making it a compelling area for investigation. In fact, detecting grapevine diseases in field conditions, rather than controlled environments, presents significant challenges. In the field, grapevines are arranged in rows and can grow to significant heights, which poses limitations on data acquisition and data capture. Capturing high-quality data becomes more challenging due to the overlapping and intertwined nature of grapevine foliage. Additionally, variations in lighting, background clutter, and environmental conditions further complicate the accurate identification and differentiation of disease symptoms on grapevines. These factors emphasize the need for robust and adaptable techniques specifically designed to handle the unique challenges of grapevine disease detection in real-world field settings. Therefore, grapevine disease detection methods necessitate the development of grapevine-specific approaches to overcome these inherent complexities and ensure reliable and precise disease identification.

Capturing close-up images of leaves, either in the field or in a controlled laboratory, using standard or multispectral cameras provides a promising avenue for obtaining detailed information about grapevine diseases. By examining leaves at such a close range, it becomes possible to detect and analyze disease leaf symptoms with greater precision and accuracy. This approach has been widely adopted in numerous studies to not only detect diseases but also differentiate between them.

In an insightful study conducted by Al Saddik (2019), a combination of spectral and textural analyses was employed to differentiate between healthy and diseased leaves. The results were correct, with an accuracy exceeding 0.85 for distinguishing between healthy and diseased leaves. Moreover, the degree of infection and the distinction between specific diseases such as FD, Bois noir, and Esca could be determined with accuracies surpassing 0.74. This demonstrates the potential of utilizing spectral and textural features to precisely characterize different diseases in grapevine leaves. Another effective approach, as demonstrated by Pantazi *et al.* (2016), involves analysing proximal sensing images by leveraging techniques such as colour space transformations, texture operator applications, and parameter extractions. A classifier achieved an impressive accuracy of over 0.93 in classifying leaves exhibiting symptoms of three specific diseases: powdery mildew, downy mildew, and black rot. This highlights the efficacy of advanced image analysis and machine learning methods in accurately identifying and classifying diseases affecting grapevine leaves. However, the most notable advancements in classifying symptomatic leaves captured in close-up have been driven by the

application of deep learning techniques, particularly Convolutional Neural Networks (CNNs). Ji *et al.* (2020) achieved remarkable results, surpassing 0.99 accuracy, when classifying leaves into four distinct classes: healthy leaves, black rot, Esca, and Phomopsis leaf spots. The use of CNNs also demonstrated an outstanding accuracy of 0.97 when classifying grapevine leaves into six classes: anthracnose, brown spot, moth damage, black rot, downy mildew, and leaf blight (Liu *et al.*, 2020). These impressive outcomes underscore the power of deep learning and CNNs in accurately differentiating between a wide range of disease types affecting grapevine leaves. A comparative study conducted by Rançon (2019) evaluated two approaches, namely Scale-Invariant Feature Transform (SIFT) and transfer learning, for the classification of Esca symptomatic leaves. Both methods exhibited excellent performance, with approximately 0.99 accuracy in predicting advanced-stage Esca leaves using either approach. Pérez-Roncal *et al.* (2022) aimed to evaluate the effectiveness of near-infrared hyperspectral imaging for detecting Esca disease in grapevine leaves before visible symptoms appear. The researchers collected 72 leaves from 6 different grapevines classified in 3 classes: 'Healthy', 'Asymptomatic' (from grapevine showing symptoms of Esca) and 'Symptomatic'. Researchers employed advanced image processing techniques and multivariate statistical analysis. Partial least squares discriminant analysis, along with different pre-processing techniques, were used for classification. The classification rates ranged from 83% to 97% in validation datasets for both three-class (healthy, asymptomatic, and symptomatic) and two-class (healthy versus asymptomatic) classifiers. In the study performed by Morellos *et al.* (2022), an RGB camera sensor was used to capture images from vineyard. The objective was to develop disease identification classifiers using images of healthy leaves and leaves infected with a fungal disease. Several CNNs were trained on the public plantvillage dataset, which consisted of close-up leaf images on a uniform background. The classes included 'healthy', 'Black rot', 'Leaf blight spot', and 'Esca'. Subsequently, fine-tuning and testing were performed on their dataset, which comprised groups of leaves in close-up under field conditions with the classes 'Esca', 'Powdery mildew', and 'healthy'. The results obtained were highly promising. However, it is important to note that their test set was limited, consisting of only 10 'healthy', 11 'Powdery Mildew' and 10 'Esca' samples, although these diseases present really different symptoms. It calls for further validation and testing on larger and more diverse datasets. Nonetheless, the use of deep learning and transfer learning in this study shows great potential for accurately identifying and classifying vineyard diseases based on leaf RGB images.

The findings from these studies show great promise for diagnosing grapevine diseases. However, the methodology relies on capturing close-up images of leaves, which presents certain limitations. Automating the process of capturing and focusing sensors on symptomatic grapevine leaves is challenging and may not be compatible with full automation. Additionally, this approach may not be suitable for diagnosing grapevine diseases that exhibit multiple symptoms across different organs, and not only the leaves. Consequently, this data acquisition method does not align with the objectives of this research. However, the algorithms employed for accurate leaf classification have the potential to inform the selection of future methods.

The widespread adoption of drones or UAVs (Unmanned Aerial Vehicles) has revolutionized the acquisition of data in various fields, including viticulture (Sassu *et al.*, 2021; Singh *et al.*, 2022). Drones offer a practical and efficient solution for capturing images of vineyard blocks due to their speed and manoeuvrability. Multiple studies have highlighted the benefits of using drones for image acquisition, enabling accurate estimation of crucial crop parameters such as biomass, canopy temperatures and size (Holman *et al.*, 2016; Ludovisi *et al.*, 2017; Madec *et al.*, 2017). Building upon these successes, researchers have explored the potential of using drones for diagnosing grapevine diseases. For instance, in certain studies (Kerkech *et al.*, 2020b, 2020a), a segmentation approach was employed to classify each pixel into one of four categories: soil pixel, shadow pixel, healthy grapevine or diseased grapevine. The results of these studies have been promising, with the first study achieving an accuracy rate of 0.92 for identifying diseased grapevines. Additionally, vegetation indices, such as Excess Green (ExG) and Green-Red Vegetation Index (GRVI), can be calculated from UAV images to detect areas of diseased vineyards (Kerkech *et al.*, 2018). The patch classification, involving the

grouping of pixels into 16*16, 32*32 or 64*64 pixel clusters, even reached an impressive accuracy of 0.95 by the conclusion of the study.

The studies mentioned above have shown highly encouraging results in the context of diagnosing grapevine diseases using drone imaging. However, these positive outcomes are limited to the discrimination between healthy and diseased grapevines, rather than the identification of specific diseases or the differentiation between diseases. This may contribute to the favourable results observed in those studies.

Bendel *et al.* (2020) introduced ground-based hyperspectral and airborne multispectral approaches for detecting foliar Esca symptoms. The study compared models developed using field data and manually annotated data (symptomatic leaves annotated on images). A segmentation model utilizing SWIR (Short Wave InfraRed) and VNIR (Visible and Near-InfraRed) spectral ranges was employed. Hyperspectral disease detection models have been developed using either original field data or annotated data. Both sensor systems demonstrated suitability for in-field disease detection. These models were subsequently applied at the plant scale. Initial findings indicate the potential for pre-symptomatic detection of external symptoms, but further evaluations are required. MacDonald *et al.* (2016) employed a spectral unmixing approach with a spatial resolution of approximately 0.25 to 0.50 meters per pixel to detect grapevine leafroll-associated virus 3 (GLRaV-3) in Cabernet sauvignon vineyards. Reflectance signatures of diseased and healthy grapevines were compared to measurements acquired by a hyperspectral camera from the air, specifically targeting GLRaV-3 reflectance spectra. The resulting continuous image represented the relative probability of a pixel containing the GLRaV-3 signature. A customized Geographic Information System (GIS) methodology was developed to compare visual symptoms to hyperspectral imaging results. On average, the detection sensitivity was 94.1%, varying between 88% and almost 100% per vineyard. The study demonstrates the usefulness and cost-effectiveness of remote hyperspectral imaging for mapping GLRaV-3 infected Cabernet Sauvignon vineyards. However, only five blocks were studied and further research is needed to explore its application in detecting GLRaV-3 on other grape varieties and identifying other grapevine pathogens. In work conducted by Albetis *et al.* (2017), the employment of both univariate and multivariate classification approaches, to categorize grapevines as either affected by FD or healthy, is studied. To accomplish this, they utilized 20 variables derived from UAV images, including spectral bands, vegetation indices, and biophysical parameters. The researchers found that the classification results were promising for red varieties, severely infected by FD. The characteristics captured by the UAV images provided useful information for distinguishing between healthy and FD-affected grapevines. However, when it came to white varieties, the results were not as convincing. This discrepancy could potentially be attributed to variations in the expression of the disease in different grape varieties, highlighting the complexity of disease detection using drone imaging. In another study by Albetis *et al.* (2018), the researchers aimed to differentiate between FD and trunk diseases, specifically black dead arm and Esca. They expanded their analysis to include 24 variables comprising spectral bands, vegetation indices, and biophysical parameters. The study involved photographing seven vineyards, encompassing five different red grape varieties, using drone imagery. The obtained results were promising for distinguishing between grapevines affected by FD and healthy grapevines. However, the discrimination between FD and trunk diseases did not yield satisfactory outcomes. This indicates that the visual cues captured by the drone imagery might not be sufficiently distinct to identify and differentiate between these specific diseases accurately.

A novel automated method using 3D point cloud data has been developed, enabling the detection of individual grapevine trunks, posts, and missing plants (Jurado *et al.*, 2020). Extensive testing in different vineyards demonstrated its high performance, particularly when applied to 3D point clouds during phases with less dense foliage (because there is less overlap between the grapevine rows in the images). This methodology could enable the detection of certain grapevine trunk symptoms in future developments.

Despite the potential advantages offered by drone imaging for data collection in the vineyard, several limitations hinder the accurate diagnosis of grapevine diseases. One prominent limitation is the inability of drone imagery to capture the lower parts of the grapevines, including the bunches and

shoots. Consequently, the analysis is restricted to the upper leaves, which may not provide a comprehensive representation of the overall health and condition of the grapevines. Additionally, the resolution of drone images, with each pixel equivalent to several millimetres, poses challenges in detecting subtle symptoms, such as small spots on leaves. These limitations of the drone imaging acquisition method may explain the difficulties encountered in achieving satisfactory results for disease discrimination. Particularly, the challenge lies in achieving good results for both red and white grape varieties while considering a wide range of data (various varieties and years) and confounding diseases. These findings underscore the need for more detailed information and a higher level of precision to effectively differentiate between various diseases affecting vineyards.

One effective method to capture detailed information of entire grapevines is by directly collecting data within the rows. This is called Proximal sensing (Oerke *et al.*, 2014). Only a limited number of studies have utilized grapevine images captured from distances ranging from 50 to 200 cm for the purpose of diagnosing grapevine diseases. One such study conducted by Bourgeon (2015) assessed the efficacy of a vehicle-mounted device in characterizing grapevine foliage. However, this study only focused on calculating vegetation indices and did not explore further disease diagnosis. In another study by Abdelghafour *et al.* (2019), a computer vision approach was employed to analyse images captured in the field. This approach utilized joint colour and texture analysis with extended structure tensors to differentiate various grapevine organs. Building upon this, a subsequent evaluation was conducted by Abdelghafour *et al.* (2020) to assess the potential of high-resolution embedded imagery for epidemiological monitoring, using Downy Mildew as a case study. The results obtained in this second step were promising, demonstrating the feasibility of estimating the overall health status of a block of grapevines without requiring a decision for each individual grapevine. This approach offers the advantage of providing block-level information for disease monitoring. In the detection of Esca symptomatic leaves, Rancon (2019) explored the use of a detection network called RetinaNet (Lin *et al*., 2018). Additionally, a comparison was made between RGB and hyperspectral images to identify early stages of Esca. However, the results indicated that hyperspectral imaging did not yield superior outcomes compared to RGB imaging for the detection of Esca at its early stages. Proximal hyperspectral sensing was utilized in (Wang *et al.*, 2023) to detect virus infection on two grapevine varieties, Pinot Noir and Chardonnay. Leaf spectral reflectance data were collected using a portable hand-held spectroradiometer. The measurement distance was approximately 50cm from the canopy, which represented a 20 cm diameter circle on the canopy. Spectral data were collected at six time points during the grape growing season for each cultivar. The presence or absence of Grapevine Leafroll Disease was predicted using Partial Least Squares-Discriminant Analysis. The temporal changes in canopy spectral reflectance revealed that the harvest time point provided the most accurate predictions. Pinot Noir achieved a prediction accuracy of 96%, while Chardonnay achieved 76%. Nguyen *et al.* (2021) used hyperspectral imagery at the plant scale to detect and classify grapevines infected with grapevine vein-clearing virus during the early asymptomatic stages. Images were captured using a hyperspectral sensor, and statistical analysis and vegetation indices were employed for classification. The study identified discriminative wavelength regions and important indices for accurate classification. A comparison of machine learning techniques, including support vector machine (SVM) and random forest (RF), and 2D and 3D CNNs was performed for classification. The results showed promising performance of the automated 3D convolutional neural network (3D-CNN) in learning features from limited sample hyperspectral data cubes. The results were promising but the small size of the set of images used in the study (40 images, only of Chardonnay grapes) does not allow to draw wide-ranging conclusions from this study. Daglio *et al.* (2022) conducted experiments to detect FD and Esca diseases using data captured directly in the field. The detection system involved a quad-bike, a mobile PC, GIS software, and RTK-GNSS. Two sensors were mounted on a structure in front of the quad-bike to gather canopy data. The sensors recorded reflectance at specific wavelengths, which were then used to calculate various vegetation indices such as NDVI, NDRE, and VIS/NIR. Initially, a statistical analysis was performed on these indices using a one-way ANOVA followed by a post-hoc test. The goal was to determine threshold values that could effectively distinguish between healthy and diseased grapevines. However, it proved challenging to establish an absolute threshold for the

vegetation indices that could reliably differentiate between grapevines affected by FD and Esca, and those that were healthy. As an alternative approach, a second methodology was tested, which involved not only considering the indices of the target grapevine individually but also taking into account the indices of nearby grapevines. The first step was the calculation of the differences between the target grapevine's indices and the means of the vegetation indices for the surrounding grapevines. The second step consisted of computing ratios from these differences and means. This method successfully identified thresholds that effectively discriminated between diseased and healthy grapevines. However, it was still not possible to differentiate between FD and Esca using this methodology. In a study conducted by Boulent *et al.* (2020), deep learning methods, specifically CNNs and FCNs, were employed to classify images of grapevines affected by FD. These images were captured by a camera positioned at a distance of approximately 100 cm. The study achieved a notable true positive rate of 0.98, indicating the efficacy of neural networks in detecting grapevine diseases without solely relying on close-up images of leaves. However, it is crucial to consider specific nuances associated with these findings. While the overall true positive rate of 0.98 was achieved, it is important to recognize that the results varied depending on the grapevine variety. Specifically, the true positive rate for the Chardonnay grape variety reached 0.98, while it significantly dropped to 0.08 for the Ugni blanc grape variety. This disparity suggests that the pronounced variations in symptom expression between different grape varieties play a vital role in the accurate detection of grapevine diseases. Furthermore, it is worth noting that this study solely utilized symptoms present on the leaves to provide the diagnosis, without considering other parts of the grapevine. The findings from Boulent's study underscore the potential of deep learning techniques in detecting FD beyond the scope of close-up leaf images. However, the variation in detection performance across different grape varieties highlights the importance of accounting for varietal-specific symptom expression patterns when developing disease detection models. Additionally, it is essential to expand the analysis beyond leaves and consider the possibility of incorporating symptoms from other parts of the grapevine to enhance diagnostic accuracy.

Overall, the utilization of grapevine images captured from distances of 50 to 200 cm for disease diagnosis has been limited. It may be due to the challenges associated with navigating through the blocks. Capturing images from moving vehicles presents multiple difficulties (no industrial solution available, equipment robustness and control over image capture). The recent development of dedicated onboard imaging solutions, coupled with the emergence of robotic means for blocks navigation, now allows for considering alternative methods of vineyard observation in close proximity. While some studies have focused on characterizing grapevine foliage and differentiating grapevine organs, few have delved into the specific task of disease diagnosis. As technology continues to evolve, there is immense potential for further advancements in proximal sensing and image analysis techniques. The integration of more sophisticated sensors and the exploration of novel algorithm architectures hold great promise for even higher levels of accuracy and specificity in the diagnosis of grapevine diseases. These advancements will undoubtedly contribute to the sustainability and productivity of vineyards, ensuring the production of high-quality grapes and wines.

B. Proximal sensing, visible imagery and IA: the most promising technologies for disease symptom detection?

Recently developed vectors and sensors enabling massive data acquisition, coupled with advanced image analysis and deep learning methodologies, have made significant strides in the field of grapevine disease diagnosis. These approaches offer a non-destructive and efficient means of assessing the health of grapevine leaves and may facilitate timely disease management strategies.

There are three questions to address: the choice of the acquisition tool (machine/robot vs. drone), the selection of the imaging sensor (visible/multispectral/hyperspectral), and the algorithms to be used. The aim is to find a combination of these different options that will allow the diagnosis of

diseases in the field (for real-world applications). These diseases can manifest themselves through various symptoms, affect different organs, and vary significantly depending on the context and grape variety. Moreover, the symptoms can be mistaken for the effects of various other biotic or abiotic factors.

Let's take the example of FD. The visibility of all grapevine organs is required as prospectors rely on a combination of symptoms to accurately diagnose this multi-symptom disease and differentiate it from confounding diseases. Discussions with groups of experts in FD diagnosis, such as GDON[1] des Bordeaux and BNIC[2], have revealed that during their surveys, the initial indicator leading them to identify diseased grapevines is the discolouration of the leaves. This symptom is the most prominent and can be observed from a distance. Their subsequent step involves approaching the grapevines closely to examine other symptoms present on various organs. If these symptoms are observed and align with the characteristic patterns associated with the disease (for example, symptomatic leaves connected to symptomatic shoots), the diagnosis can be made with a high level of confidence. When it comes to grapevine trunk diseases like Eutypa or Botryosphaeria diebacks, symptoms on the grapevine trunk or shoots may not occur simultaneously if a grapevine is affected. However, when symptoms manifest on both organs, it ensures a definitive diagnosis of the disease. For these diseases, drones could potentially aid in detection, particularly during early grapevine growth stages, although challenges remain. FD expresses symptoms on leaves, shoots and bunches, predominantly close to harvest, with significant foliage obscuring the view of bunches and shoots when using drones. The use of this acquisition tool seems therefore not suitable for a unified diagnostic methodology of multi-symptom grapevine diseases.

In addition, if automated detection is desired, solely acquire and diagnose each symptom in close-up (which yields the best results according to the literature review) seems not suitable to a real-world application.

The necessity for visibility of different organs, located at various heights and depths within the canopy, suggests that capturing images facing the trellising plane, is necessary. The size of the symptoms and the existence of potentially confounding factors demand the use of a high enough resolution that allows characterizing the geometry and texture of the symptoms. Lastly, the multiplicity and co-occurrence of symptoms call for algorithmic approaches with a certain versatility/plasticity and a good capacity for generalization.

The literature review thus confirms the choice made at the IMS laboratory (Abdelghafour *et al.*, 2020; Rançon *et al.*, 2023) to prioritize the use of visible-light cameras, embedded in the grapevine rows. Multispectral or hyperspectral imaging is often employed to calculate indices or identify characteristic wavelengths associated with specific diseases. Some studies have shown very promising results for the automated diagnosis of diseased leaves. However, in the case of multi-symptom diseases, where symptoms are diverse and dependent on grape varieties, and various diseases can be very confounding, multispectral imaging does not seem to provide more information than RGB imaging. The acquisition setup will be presented in Chapter III Section B.

The main focus of this thesis is the use of algorithmic approaches for the determination of disease presence. The assumption is that AI/Deep learning approaches, as evidenced by the community's enthusiasm, have the potential to enable symptom detection (Garcia-Garcia *et al.*, 2017; Kaur & Singh, 2023). However, the need to identify the right methods, optimize their use, and, most importantly, propose innovative solutions to achieve complex diagnoses in the presence of multiple confounding factors are part of this research.

[1] Groupement de Défense contre les Organismes Nuisibles (Pest Control Group)

[2] Bureau National Interprofessionnel du Cognac (National Interprofessional Bureau of Cognac)

C. Four hypotheses guiding this research

Based on a comprehensive literature review, the selected data acquisition approach, and with the aim to emulate the diagnostic methods employed by field experts, several intermediate hypotheses have been formulated. These hypotheses will serve as foundations for further investigation and exploration in the subsequent stages of this study.

- **Hypothesis 1 - Opting for RGB image acquisition through proximal sensing is a suitable approach for the detection and diagnosis of multi-symptom grapevine diseases.**
  The use of RGB imaging in proximal sensing allows for a comprehensive view of the entire grapevine, providing a detailed assessment of the symptoms present. The RGB images effectively capture the visible spectrum of light, enabling the detection and discrimination of various grapevine diseases. By analysing the colour and texture patterns exhibited by the grapevines and their organs, such as leaves, shoots and bunches, significant insights can be gained regarding the health status of the vineyard.
- **Hypothesis 2 - Deep learning-based approaches are highly suitable for addressing the specific challenges of this research.**
  It seems imperative to thoroughly test and explore deep learning approaches in this research. The field of deep learning has gained significant momentum in recent years and offers immense potential in solving complex problems across diverse domains (Kamilaris & Prenafeta-Boldú, 2018; LeCun *et al.*, 2015; Shinde & Shah, 2018). Its ability to automatically learn and extract meaningful patterns from large datasets makes it particularly well-suited for tackling the challenges associated with grapevine disease detection. Furthermore, the flexibility of deep learning algorithms allows them to adapt well to different grapevine disease scenarios. By gathering more data and refining the models, the deep learning approaches can evolve and improve, continuously enhancing the accuracy and robustness of the disease detection system.
- **Hypothesis 3 - Detecting and associating all the symptoms of individual vines, enables a more reliable diagnosis for multi-symptom diseases.**
  The believe is that the conventional approach found in the literature (Boulent *et al.*, 2020; Ji *et al.*, 2020; B. Liu *et al.*, 2020), being the widespread use of CNNs, is not well-suited for diagnosing multi-symptom diseases. A single-step approach cannot fully capture the complexity of the task, which involves identifying symptoms of various forms on different organs, varying according to grape variety, and understanding the significance of their concurrent presence in the diagnostic process. To address this, a two-step approach appears to be more appropriate: a first step focused solely on detecting individual symptoms, allowing the algorithms to grasp the complexity of the symptoms. This first step is followed by a second step of associating these detections to distinguish diseases of interest from confounding diseases.
- **Hypothesis 4 - Considering not only an image taken on an individual grapevine but also images of neighbouring grapevines enhances the reliability of the diagnosis.**
  Analysing images on an individual basis can potentially result in incomplete information and limited insights. The presence of neighbouring grapevines can play a crucial role in refining the diagnostic process. Erroneous detections on one grapevine may be rectified by accurate detections on adjacent grapevines. Uncertain predictions can be further validated by the presence of other nearby detections. This may lead to a more reliable overall diagnosis. Furthermore, taking into account both sides of a grapevine can provide valuable insights. It is not uncommon for symptoms to manifest predominantly on one side, making it critical to capture images from multiple angles to ensure comprehensive symptom detection.

D. Methodological options

By combining RGB imaging and proximal sensing, the aim is to leverage the strengths of these techniques to enhance the performance of deep learning algorithms in accurately identifying and classifying different grapevine disease symptoms as well as produce an automated decision at the grapevine scale. The intention is to comprehensively outline the methodologies available in order to effectively accomplish the objectives of this research. Carefully examining and documenting the range of approaches available ensure a well-informed decision-making process and a strategic path towards achieving the objectives of this research.

However, computer vision approaches, including deep learning methods, constitute a wide range of approaches offering numerous options for image processing. Concerning the recognition of diseases in images, three main categories of approaches are of significant interest: classification networks to classify an entire image, detection networks to find occurrences of one or multiple sought-after classes within an image, and segmentation networks to delineate groups of pixels of the same class within an image.

It should be noted that while neural approaches are now mainstream, other more traditional methods could be considered. These approaches generally operate in two steps: feature extraction, carefully designed following expert analysis of the relevant images, and a decision-making step based on a machine learning algorithm. However, the generalization ability of neural approaches, linked to their "learnability" property, has led the computer vision community to widely adopt these techniques.

In more detail, the 3 categories of neural approaches that appear to be of particular interest regarding disease recognition in images are:

- ***Classification*** - When it comes to automatically classifying images without pre-processing, deep learning algorithms, particularly CNNs, are the most commonly used approaches (Albawi *et al.*, 2017). CNNs have the ability to learn and extract relevant features directly from raw image data, eliminating the need for extensive pre-processing. By leveraging the hierarchical architecture of CNNs, these algorithms can automatically learn and identify discriminative patterns and features from the images. The basic building blocks of a CNN are convolutional layers, pooling layers, and fully connected layers. The input to a CNN is an image represented as a grid of pixels. The convolutional layers consist of filters (also called kernels) that convolve over the input image. Each filter extracts features by performing element-wise multiplications and summations at each location of the image. This process generates a feature map, which represents the presence of certain visual patterns or features in the input image. Pooling layers downsample the feature maps, reducing the spatial dimensions and the number of parameters in the network. Common pooling operations include “max pooling”, which selects the maximum value within a pooling window, or “average pooling”, which takes the average value. The final feature maps are flattened and passed through fully connected layers. These layers resemble a traditional artificial neural network, where each neuron is connected to every neuron in the previous layer. The fully connected layers learn to classify the image based on the extracted features. Some popular CNN architectures for image classification include AlexNet (Krizhevsky *et al.*, 2012), VGGNet (Simonyan & Zisserman, 2015), GoogLeNet (Szegedy *et al.*, 2015), ResNet (He *et al.*, 2015), EfficientNet (Tan & Le, 2019).

  Transformers are a type of deep learning architecture that has recently revolutionized the field of artificial intelligence, particularly in natural language processing (NLP) tasks. They were introduced by Vaswani *et al.* (2017). They have been adapted and applied to image processing tasks with great success. The application of transformers in image processing is known as Vision Transformers or ViTs (S. Khan *et al.*, 2022). In CNNs, information is processed locally by looking at small receptive fields within the input image and gradually build up hierarchical representations of the image. On the other hand, transformers excel at capturing global dependencies and relationships between image patches. This mechanism is known as "self-attention" or "scaled dot-product attention". ViTs have demonstrated good performance

in image classification tasks, sometimes surpassing traditional CNN-based approaches when trained on large-scale datasets.

- ***Detection*** - If the aim is to surpass mere image classification and delve into individual symptom detection, object detection techniques, such as template matching (Duda & Hart, 1974), Haar classifiers (Viola & Jones, 2001), Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005), and advanced deep learning-based algorithms like Fast R-CNN (Girshick, 2015), YOLO (Redmon *et al.*, 2016), and SSD (W. Liu *et al.*, 2016), allow for the identification and localization of objects within images. These methods have revolutionized fields like autonomous driving, surveillance systems, and object recognition in real-time scenarios.
- ***Segmentation*** - Object segmentation approaches take the detection a step further by providing precise boundaries for objects within an image. Semantic segmentation assigns class labels to every pixel in an image, allowing for detailed understanding of the scene (Garcia-Garcia *et al.*, 2017). Instance segmentation goes even further by differentiating individual instances of objects, enabling accurate object counting, tracking, and analysis. The advent of advanced architectures like Mask R-CNN (He *et al.*, 2017) has propelled instance segmentation to new heights, enabling pixel-level object masks and precise delineation of objects even in complex scenes.

Choosing either a naïve or an "expert-like" algorithmic strategy

Regarding to the subject of this research, which is the automated identification of diseases in images, different strategies could be adopted.

The first approach would involve submitting an entire image to an algorithm that would determine the presence or absence of the disease (without necessarily localizing the symptoms). In this case, the use of a classification neural network would be a viable option. This is, in fact, the approach chosen by Boulent *et al.* (2020). This methodology will be explored in Chapter IV.

On the other hand, the second option would be to identify the symptoms first and then associate and diagnose the disease, as implied by hypothesis 3 (see Chapter II Section C). The identification of individual symptoms can be done either through detection algorithms or segmentation algorithms: these approaches will be explored in Chapter V.

It is worth noting that once the symptoms are detected, it is necessary to consider how to associate them in order to reach a diagnosis at the grapevine scale. When it comes to providing classification based on feature extraction, a diverse range of techniques is commonly employed. Support Vector Machines (SVM) (Cortes & Vapnik, 1995) and k-Nearest Neighbours (k-NN) (Cover & Hart, 1967) are popular machine learning algorithms that leverage extracted features to assign class labels. SVMs find an optimal hyperplane to separate classes, while k-NN assigns labels based on the nearest neighbours in the feature space. Decision trees and ensemble methods like Random Forests (Ho, 1995) and Gradient Boosting (Friedman, 2002) also utilize extracted features for classification, enabling accurate and interpretable results. Additionally, Naive Bayes classifiers (Rish, 2001) leverage probabilistic assumptions for efficient classification. An emerging technique in this field is the Graph Neural Network (GNN), which excels in classification tasks involving graph-structured data (Wu et al., 2021; Xia et al., 2021). GNNs capture relationships and dependencies between data points modelled as graphs, making them ideal for scenarios like social network analysis, recommendation systems, or molecule classification. GNNs allow for the incorporation of node features and graph structure to learn meaningful representations and make accurate predictions. The use of GNNs has shown promising results in various domains, where the data can be naturally represented as a graph (Zhou *et al.*, 2019). These approaches of symptom combination will be explored in Chapter VI.

Building a comprehensive database for training and validation purposes

In order to validate and assess the effectiveness of some of these methods, it becomes essential to build a substantial and diverse image database. This database will serve as the foundation for training and evaluating the developed models. Regardless of the approach considered, a crucial and essential first step is the creation of a qualified database: from image acquisition and ground truth data to the

annotation of symptoms in the images. The literature review demonstrates how critical this step is, as it determines both the successful learning of decision-making algorithms and their robustness to the variability in the acquisition process and plant material.

To initiate the research and provide a point of comparison for more innovative methodologies, the initial phase of the research was focused on the exploration of the straightforward approach offered by CNNs, enabling direct classification of the images. To accomplish this, there was a necessity to establish a robust and reliable image acquisition protocol that ensures accurate ground truthing for each photographed grapevine. This involves carefully documenting the characteristics and conditions of each grapevine, including disease presence or absence, as well as the specific symptoms observed. To achieve accurate symptom detection, it is crucial to ensure precise labelling of the symptoms in the images. This may require the involvement of experts or skilled annotators who can carefully identify and mark the specific symptoms present on the grapevines. The accuracy and consistency of symptom labelling are crucial to train the deep learning models effectively and enable them to distinguish between different diseases and healthy vines accurately.

Furthermore, for comprehensive diagnosis at both the individual grapevine and plot scale, it is necessary to capture automated and geolocated acquisitions. This enables the collection of data from multiple vines in close proximity and obtain a holistic view of the vineyard. Automated acquisitions provide efficiency and scalability, allowing to cover larger areas and monitor multiple vines simultaneously. Geolocation information enhanced the contextual understanding of the vineyard, facilitating spatial analysis and correlation of disease patterns with environmental factors.

To ensure the robustness and generalizability of the developed methodologies in real-world applications, it is imperative to acquire images representing various grapevine cultivars and encompassing diseases that may exhibit similar symptoms. This diversity in the dataset will enable the developed models to learn and differentiate between different diseases accurately, even when faced with challenging scenarios and potential confusion between similar symptoms.

In the upcoming chapter, the protocol for acquiring and annotating the dataset will be presented. The systematic approach employed to capture high-quality images, establish ground truth labels, and ensure comprehensive coverage of grapevine varieties and diseases will be outlined.

III. Building up a consistent database of grapevine disease symptoms

Table of content

This study revolves around two distinct projects. The first one is the French project ProspectFD, which aims to create a comprehensive decision support tool for Flavescence dorée prospecting. This decision support system takes into account the plot's historical data, its environment, and the results of disease detection algorithms in order to provide relevant choices of areas for inspection to prospectors. A prospector, in this context, refers to someone who explores or scouts the vineyard to identify potential diseased grapevines. The various stakeholders involved in this project, funded by the French Research Agency (ANR) [grant ANR-19-ECOM-0004 Prospect FD], include the IFV[3], INRAE[4], GDON des Bordeaux, BNIC, Bordeaux Sciences Agro, and the IMS laboratory.

The second project takes place in New Zealand at the Plant and Food Research (PFR) laboratory of the Marlborough Research Centre. PFR has been actively involved in a range of research

[3] Institut Français de la Vigne et du vin (French Institute of Vine and Wine)

[4] Institut national de recherche pour l'agriculture, l'alimentation et l'environnement (National Institute for Agricultural, Food, and Environmental Research)

programmes investigating grapevine trunk diseases in the past. Some of these investigations have been government and/or industry funded, and other works were funded by Plant and Food Research itself. Reviewing GTDs research conducted over the last two decades had determined that a gap existed for the automated detection of GTDs for both research and application uses. This project was designed to investigate possible tools to fill the pipeline for subsequent projects and was carried out at the PFR laboratory of the Marlborough Research Centre.

In the following, the targeted diseases, the image acquisition system used for both projects and the collected and annotated data will be presented.

A. Targeted diseases

a. Flavescence dorée

i. Historic, impacts and control methods of the disease

Flavescence dorée is a relatively new grapevine disease. This devastating grapevine disease was first identified in the early 1950s in the French Côte d'Or region. The name "Flavescence dorée", which translates to "golden yellowing", vividly describes the yellow discolouration exhibited by infected vines of white grape varieties. Initially, FD was regarded as a localized problem confined to specific areas of France. However, over time, it has spread relentlessly, impacting other prominent wine-producing regions across Europe, and affecting both red (red leaf symptoms) and white grape varieties (yellow leaf symptoms).

The leafhopper *Scaphoideus titanus* Ball has emerged as the primary vector responsible for the transmission of Flavescence dorée (Lefol *et al.*, 1993). This small insect, measuring about 3-4 mm in length and belonging to the Cicadellidae family, originates from North America. Probably accidentally introduced to Europe in the early 20th century (Papura *et al.*, 2009), *Scaphoideus titanus* has established itself in various wine-growing regions. The leafhopper becomes infected when it feeds on grapevine phloem sap from an infected plant. During this feeding process, the phytoplasma responsible for FD, known as "Candidatus Phytoplasma vitis," is ingested by the insect, subsequently the spread of FD can occur through various mechanisms. Infected leafhoppers can move from one grapevine to another, transferring the phytoplasma and spreading the disease within a vineyard. Additionally, the insects can be carried by wind currents, facilitating their dispersal over longer distances. Human activities, such as the movement of plant material or agricultural machinery, can also contribute to the spread of infected leafhoppers and the transmission of FD to new areas. In addition, if scions or rootstocks are taken from an infected plant without appropriate desinfection, the resulting plants are highly likely to carry the phytoplasma (S. Robert *et al.*, 2007).

FD poses a significant challenge in vineyards, and current practices revolve around meticulous inspection of each grapevine to detect disease symptoms. Directly targeting the phytoplasma causing FD is not feasible; therefore, control measures focus on indirect approaches. The European Union classified FD as a quarantine organism in 1993. France and Italy have implemented mandatory declarations and control measures through ministerial decrees. Prefectural orders are also issued annually in FD-contaminated winegrowing regions, outlining legal obligations such as reporting disease cases, treatment requirements in specific areas, insecticide-based vector control with treatment documentation, uprooting of affected vines or plots, and the elimination of abandoned vineyards.

Population dynamics of *S. titanus* are influenced by temperature, humidity, vineyard management practices, and host plant availability. Effective monitoring and control of the vector are vital in preventing FD spread (Chuche & Thiéry, 2014). Integrated pest management strategies, including targeted insecticide applications and physical barriers, are used to manage *S. titanus* populations and reduce disease transmission risks.

FD imposes significant consequences on vineyards and the wine industry, leading to substantial economic losses attributed to diminished grape yields and quality (Morone *et al* 2007). In addition to the direct effects on grapevine health and grape production, managing FD requires costly practices such as establishing control perimeters and administering insecticide treatments, thus increasing production costs for growers. In France in 2018, the mandatory control perimeters encompassed 75% of the vineyard area. 61% of the vineyards were classified as contaminated by FD, with surveillance efforts covering 19% of the vineyard. Presently, various countries in the Mediterranean region are affected by this disease (EFSA Panel on Plant Health (PLH), 2016). Furthermore, FD is subject to strict regulation in many other nations, necessitating quarantine measures and mandatory reporting. The spread of FD can have devastating consequences for entire vineyards, impacting both current and future harvests. Therefore, it is crucial to implement effective prevention, early detection, and appropriate management strategies to minimize the effects of FD on vineyards.

ii. Symptom description

FD manifests itself through various symptoms in affected grapevines. These symptoms typically become visible during the summer, approximately one year after the plant becomes contaminated. The signs of FD are concurrently evident on three primary organs of the grapevine: the leaves, shoots, and bunches. Leaf symptoms include a red discolouration in red grape varieties and a yellow discolouration in white grape varieties. The leaves may also exhibit rolling or curling. Shoots affected by FD stay green in the absence of browning, meaning they do not undergo the normal lignification that provides resistance to frost in colder seasons. At the bunch level, the berries are wilting and the inflorescence are drying out (Figure 1).

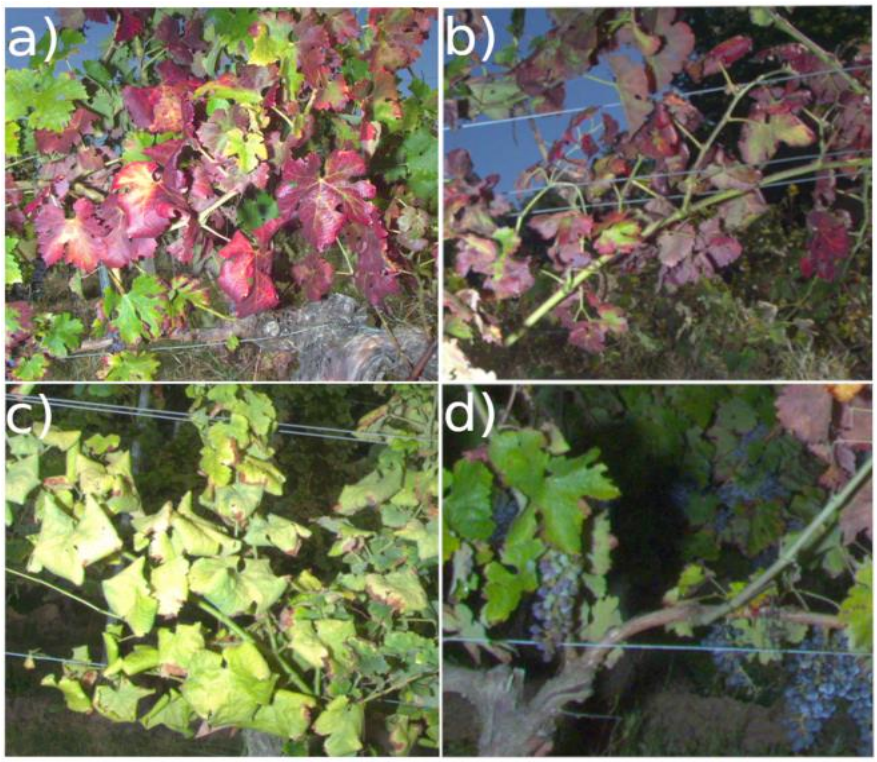

Figure 1. Images of Flavescence dorée symptoms : a) Symptomatic leaves on Cabernet sauvignon ; b) Symptomatic leaves and shoots on Cabernet sauvignon ; c) Symptomatic leaves and shoots on Sauvignon blanc ; d) Symptomatic bunches and shoot on Cabernet franc.

The symptoms of FD can vary significantly depending on the grape variety affected, especially those on leaves, the most obvious symptom of FD. Some grape varieties may display more pronounced and noticeable symptoms of FD. The leaf discolouration in these varieties may be more vivid, with intense red or yellow hues that are easily distinguishable. In contrast, other grape varieties may exhibit milder symptoms of FD. The leaf discolouration may be less intense or restricted to certain areas of the plant, making it less conspicuous. The rolling or curling of leaves may also be less pronounced, resulting in a more subtle visual effect (Figure 2). Understanding these variations is crucial for accurate diagnosis and the development of effective management strategies tailored to specific grape varieties.

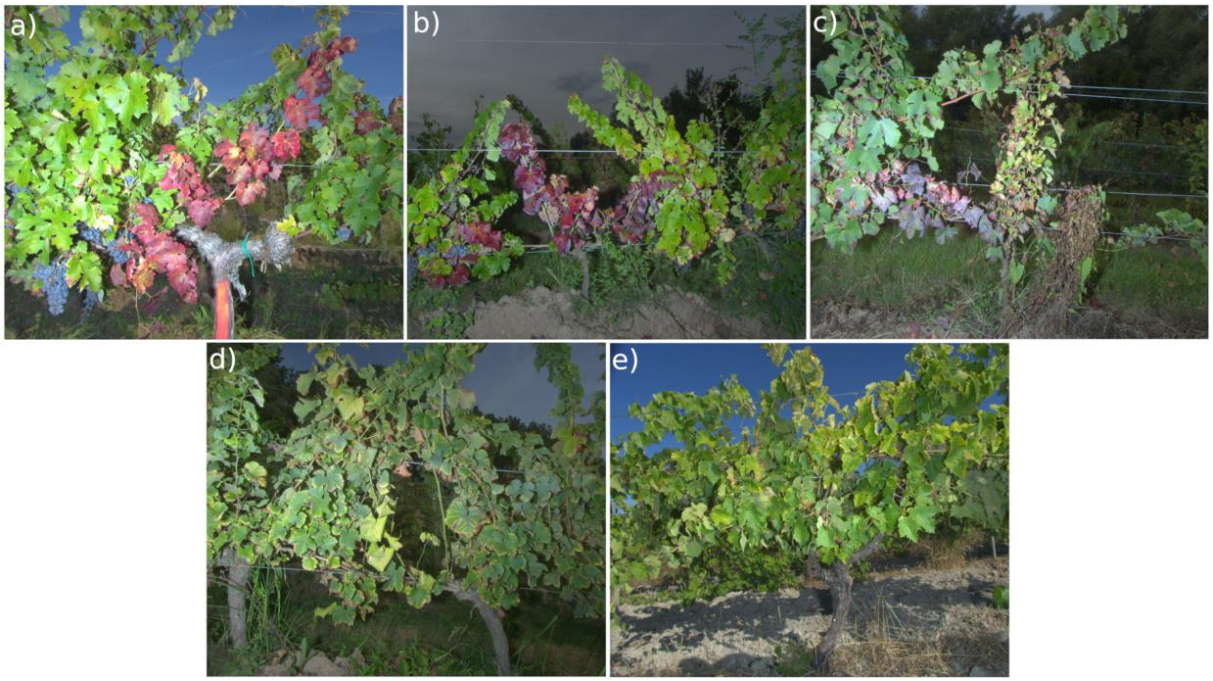

Figure 2. Images illustrating the differences in symptom expression among varieties. The symptomatic leaves of FD are bright red for: a) Cabernet sauvignon variety; dark red for :b) Cabernet franc variety; pale purple for: c) Merlot variety; yellowing and curling for: d) Sauvignon blanc variety; slightly yellowing for: d) Ugni blanc variety.

It is important to note that while these colour changes in the leaves are common indicators of FD, they are not exclusive to this disease. Other phytosanitary conditions or environmental factors can also cause similar leaf discolouration. When diagnosing FD, it is essential to consider other confounding biotic or abiotic stresses that may exhibit similar symptoms. These include conditions such as leaf roll (Naidu et al., 2014), where the symptoms visually resemble FD with faded veins in red grape varieties and yellowing in white varieties, but the lignification of shoots remains normal. Magnesium deficiencies or Esca (Scalabrelli, 2014) can also lead to similar symptoms, but in these cases, the leaf veins remain unchanged, and lignification is unaffected. The presence of the green leafhopper or Buffalo treehopper can cause symptoms similar to FD, with red (or yellow on white varieties) discolouration primarily occurring at the leaf margins, forming mosaic-like patterns delimited by small veins. Grapevine yellows can exhibit symptoms that may be confused with FD. This group of diseases includes diseases caused by phytoplasmas or viruses that can lead to yellowing and other symptoms similar to FD. Laboratory analysis or molecular testing is often necessary to distinguish them from FD. The tips of the shoots can also be non-lignified even though the grapevine is healthy. Lignification is a process that starts from the base of the shoot and extends to its tip, it is then possible to find only partially lignified shoots while the disease is not present. This can be due simply to a lack of vine vigour. Additionally, other factors such as broken shoots are leading to a visually green shoot and discoloured leaves (Figure 3). Finally, the most confounding disease of FD is Bois noir, caused by the phytoplasma "*Candidatus* Phytoplasma solani" (Quaglino *et al.*, 2013) and also transmitted by a leafhopper

(*Hyalesthes obsoletus*). Bois noir can cause very similar symptoms, including leaf discolouration and wilting of berries (Tessitori *et al.*, 2018). Distinguishing between FD and Bois noir disease is visually impossible, even for experts. Laboratory analysis of grapevine samples becomes necessary to obtain an accurate diagnosis and differentiate between the two diseases. While the symptoms may resemble each other, a proper diagnosis ensures appropriate disease management strategies can be implemented to protect the vineyard. The laboratory investigations were not included in the scope of this study. Hence, for the remaining sections of the manuscript, the term "FD" will be used to encompass both FD and Bois noir, acknowledging the potential presence of either condition.

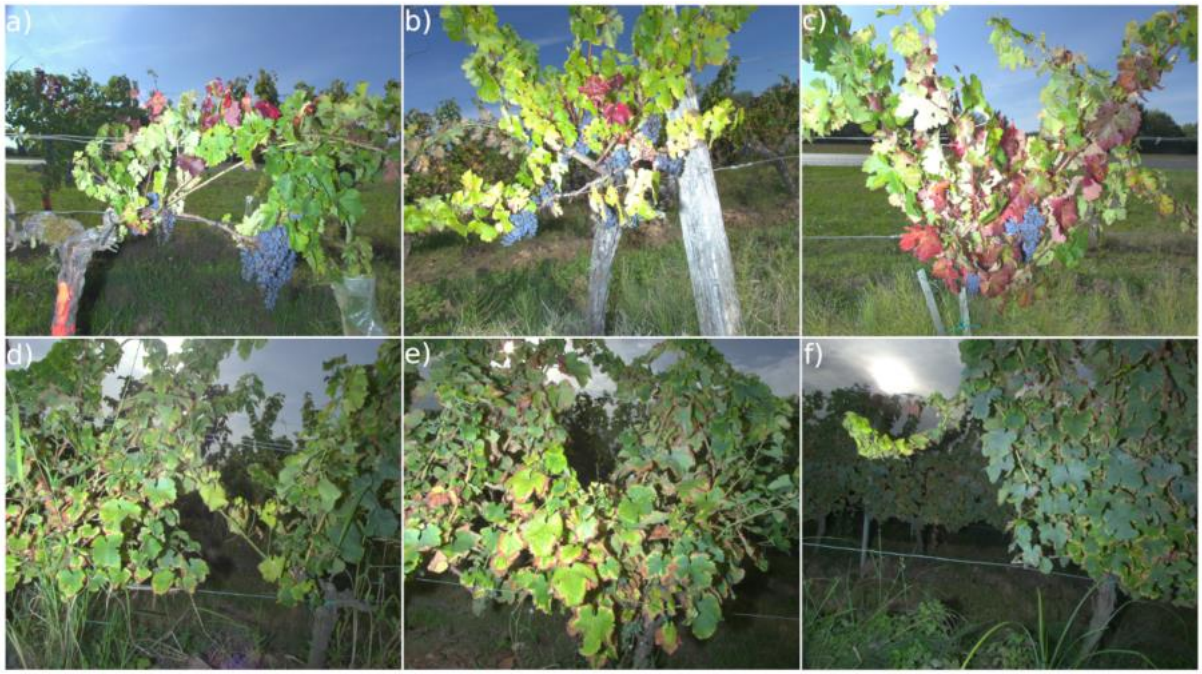

Figure 3. Example of confounding symptoms of FD. First row: a) FD; b) Buffalo Treehopper; c) Leafroll virus on Cabernet sauvignon. Second row: d) FD; e) Other yellowing disease; f) Broken shoot on Sauvignon blanc.

b. Botryosphaeria and Eutypa diebacks

i. Historic, impacts and control methods of the diseases

Botryosphaeria dieback (BD), a fungal disease, has had a notable impact on grapevines for some decades. This plant pathogen, belonging to the Botryosphaeriaceae family, affects grapevine species worldwide. Historically, BD has posed persistent challenges in vineyards, with outbreaks occurring in various wine-producing regions. Its prevalence can be attributed to factors such as climatic conditions, susceptible grapevine cultivars, and inadequate disease management practices. Early reports on BD, also known as black dead arm, mentioned the presence of both *Phomopsis viticola* and *Sphaeropsis malorum* (synonymous with *Botryosphaeria stevensii*). Initially, *P. viticola* was considered the primary pathogen responsible for the symptoms (Chamberlain *et al.*, 1964). However, research since the early 2000s has increasingly shown the significant role played by species within the Botryosphaeriaceae family in grapevine decline (Larignon *et al.*, 2015).

Eutypa dieback (ED) is the result of an infection caused by the fungus *Eutypa lata*, alternatively recognized during its asexual phase as *Libertella blepharis*. It is prevalent in grape-growing regions worldwide where annual rainfall exceeds 600 mm (Pearson & Goheen, 1988). The primary means of spreading Eutypa lata is through spores, which are discharged from infected wood when it rains. These spores have the potential to be disseminated through either rain splash or wind to nearby pruning wounds, thereby posing a risk of infecting otherwise healthy vines (Mundy & Manning, 2010).

Controlling these two diseases requires an integrated approach that combines cultural, chemical, and biological control methods. Cultural practices include proper pruning techniques, maintaining adequate plant spacing for air circulation, and removing and destroying infected plant material to minimize the disease's spread. Chemical control measures can be employed using fungicides at or following pruning to protect the wounds from infections. Biological control methods involve using beneficial microorganisms or natural antagonists that can suppress the growth and activity of the pathogen. In the absence of effective chemical, biological and cultural control, removal and destruction of diseased grapevine parts is recommended (Pearson & Goheen, 1988).

The impact of these two diseases on grapevine cultivation is substantial, with infected plants experiencing reduced vigour, stunted growth, and diminished fruit quality and yield. Dieback of shoots can result in fewer and smaller grape bunches or smaller-sized grapes. These consequences can lead to significant economic losses for grape growers by potentially impacting the fruit availability and the quality of the wine produced. Furthermore, BD and ED are chronic and can persist in a vineyard for many years. The fungus can survive in infected wood or grapevine debris, leading to ongoing infections and recurring outbreaks. This long-term impact necessitates appropriate management strategies to control the disease effectively.

ii. Symptom description

Similarly to FD and Bois noir, Botryosphaeria and Eutypa diebacks both exhibit the same symptoms and a laboratory analysis is required to differentiate between them. In the rest of the document, the acronym ‘GTDs’ (for Grapevine Trunk Diseases) will refer to both of these diseases. Note that the symptoms described in this section are specific to the GTDs observed in the Marlborough region of New Zealand.

In the early stages of infection mild leaf chlorosis is one such sign, characterized by slight yellowing or discolouration of the leaves. Leaf wilt, where the leaves droop and lose their turgidity, is another indication.

Vascular occlusions form within the xylem, blocking the flow of water and nutrients. This process is known as vascular occlusion formation. Visually, this leads to stunted branches with sparse foliage, referred to as "short shoots" (Figure 4). Moreover, the cambium layer, responsible for the growth of new vascular tissue, is also damaged This results in the death of the cambium layer itself.

As a consequence, a canker, a localized dead area, becomes visible on the outer surface of the grapevine's trunk or arm.

The impact on grapevine health becomes more apparent as the symptoms progress. Affected grapevines may experience sudden wilting during the growing season, losing their vigor and vitality. Additionally, some infected grapevines may fail to emerge from their dormant state, leading to delayed or stunted growth. A highly distinctive symptom of the disease is shoot development occurring exclusively on one side of the trunk. This characteristic is commonly known as "half head"

Figure 4. The 3 visual symptoms associate with GTDs. First row: a) Short shoots; b) Canker; c) Half head on Sauvignon blanc. Second row: zoom on the symptoms.

B. Acquisition device

The data acquisition system used to collect the data was developed by the IMS laboratory, used at the core of several projects, and presented by Rançon *et al.* (2023).

The core of the data acquisition device is the Basler Ace (acA2440-20gc GigE) industrial RGB camera, boasting a 5-megapixel resolution and a global shutter mechanism (Figure 5). Complemented by a 6mm lens, the camera offers a wide 70-degree horizontal field of view, capturing detailed imagery with clarity.

To address the variability in luminance levels arising from diverse weather conditions, the sensor system incorporates a powerful lighting solution: the high-power Phoxene Sx-3 xenon flash. In addition, the system leverages a unique "day for night" effect, carefully designed to selectively illuminate the foreground while maintaining a dark background. This technique not only adds visual appeal to the images but also enhances analysis accuracy by creating a distinct contrast between the subject of interest and the surroundings. Moreover, to combat motion blur resulting from vehicle vibrations, a combination of techniques was employed: firstly, the implementation of a short exposure time allowed for sharp image capture even in the presence of movement and secondly, the use of a small aperture, for an optimal depth of field.

The sensor system integrates a Global Navigation Satellite System (GNSS) module, enabling precise positioning and geolocalization of captured images. For optimal precision, two options are proposed: a high-precision ublox Zed-F9P unit and a standard precision generic USB GNSS unit based on the ublox Neo-6 chip. Since real-time georeferencing was not necessary for the study, postprocessed kinematics (PPK) was performed to achieve subdecimetric precision, allowing accurate geolocalization of images within vineyards.

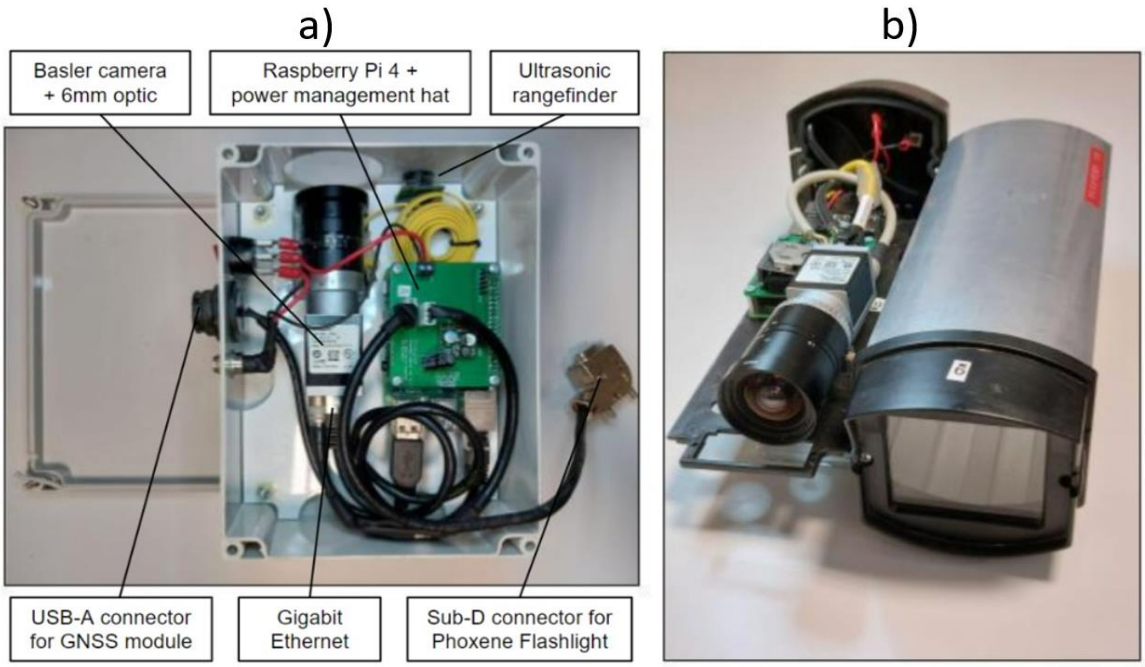

Figure 5. Presentation of the acquisition device used for the acquisition of FD and GTDs images in a) a large case; b) a narrow case. Illustration from (Rançon et al., 2023).

The need for versatility and ease of integration is ensured by a sensor system designed to be mounted on various vehicles, as presented in Figure 6. The central unit of the system is the Raspberry Pi 4+, providing a standardized platform for integrating different devices, including the camera, flash unit, GNSS system, and rangefinder.

Figure 6. Image acquisition device for the acquisition of FD images attached to a wheelbarrow for targeted acquisitions, a tractor and a quad during automated image acquisitions.

The sensor system supports two modes of operation: autonomous and manual control. In the autonomous mode, image acquisition is triggered and halted based on precise positioning data. Active plots are defined as polygons using geographical information systems (GIS), such as Google Earth, and saved as ".kml" files. This approach enables seamless image capture within designated areas, streamlining data acquisition workflows.

For manual control, an Android smartphone application that establishes a remote Wi-Fi connection with the sensor system was developed. The interface provides comprehensive status updates for each component, real-time preview of captured images, storage space monitoring, and more. Users can manually initiate and terminate acquisitions, adjust camera parameters (exposure time, frequency) and manage data operations.

C. Acquisition at the grapevine scale

a. Flavescence dorée

In collaboration with experts in FD diagnosis, a meticulous image acquisition protocol was developed to ensure accurate labelling of the acquired images.

Prior to collecting images, experts identified specific blocks known to have a high prevalence of FD cases. Upon reaching these locations, experts pointed out the target grapevines for photography. These included grapevines affected by FD, as well as other diseases such as Esca, mildew, deficiencies, and phytosanitary issues that could exhibit symptoms resembling FD. For added convenience, the acquisition device was mounted on a wheelbarrow, enabling swift movement through the grapevine rows while maintaining stability during image capture. During the image acquisition process, an annotation file was completed, specifying the identified disease at the grapevine level and any additional symptoms observed, such as non-lignified shoots, desiccated bunches, burnt leaves, and nutrient deficiencies.

Over a period of two years, images were acquired accompanied by scouting experts to 14 vineyard blocks, planted with five different grape varieties, all identified as having a significant presence of FD cases. Images were acquired of the most prevalent grape varieties in the Nouvelle-Aquitaine region in France: Ugni blanc for white wine and Cabernet sauvignon for red wine. Additionally, to account for variations in symptom expression across different grape varieties, images of two other red grape varieties, Cabernet franc and Merlot, as well as one other white grape variety, Sauvignon blanc were acquired. The image acquisition took place in September and October 2020 and 2021, just prior to the harvest in France, as this is when the symptoms are most prominently expressed. The images were captured from a distance of 100 to 200 centimetres, depending on the size of the rows, in order to encompass the entire grapevine within the frame. The primary focus during the acquisitions was on FD and its confounding diseases. To facilitate data analysis, the images were classified into four distinct classes based on the disease symptoms present in the images: 'FD', 'ESCA', 'CONF', and 'CONF+'. The ‘Esca’ class was created due to the substantial number of images exhibiting this particular disease. Images displaying visual symptoms different from those of FD but potentially confusing for algorithms (e.g. sunburned leaves or symptomatic leaves of deficiencies) were classified into the 'CONF' class. Images exhibiting visual symptoms, particularly on the leaves, closely resembling those of FD, were categorized under the 'CONF+' class. This classification aimed to assess the algorithm's efficacy in handling the most challenging and complex cases. A summary of the acquisition for the diagnosis of FD is presented in Table 1. The precise locations of each parcel in which acquisitions were made are available in Appendix 1.

Table 1. Summary of the number of acquired images and associated ground truth at the grapevine *scale for FD.*

Grape variety, acquisition year	Cabernet sauvignon, 2020	Ugni blanc, 2020	Cabernet sauvignon, 2021	Merlot, 2021	Cabernet franc, 2021	Ugni blanc, 2021	Sauvignon blanc, 2021	Total
Dataset name	**CS20**	**UB20**	**CS21**	**M21**	**CF21**	**UB21**	**SB21**	
Total number of	**405**	**463**	**116**	**98**	**86**	**161**	**154**	**1483**
of which FD	159	211	107	53	56	112	56	754
of which Esca	97	49	3	5	8	28	59	249
of which CONF	90	153	6	23	22	17	35	346
of which CONF+	59	10	0	17	0	4	4	94

b. Grapevine trunk diseases

The GTDs acquisitions took place in the region of Marlborough, New Zealand. Spanning a period from November 20 to December 22, 2022, the data collection efforts were focused on understanding the presence and manifestation of GTDs.

To ensure a comprehensive study, strategically acquisitions were conducted at different phenological stages of the grapevines. This approach allowed us to investigate the ease of detection in early stages (flowering and fruit set) when foliage density is lower, enabling clearer observation of characteristic symptoms such as stunted and deformed shoots. By capturing images at these crucial stages, the aim was to enhance the understanding of the disease's progression and develop more effective diagnostic methods.

To manoeuvre through the vast vineyards and take full advantage of the ample row width, the acquisition setup was fixed to the rear of a compact utility van (Figure 7). Accompanying the team was an expert in grapevine trunk diseases, whose expertise ensured accurate image-based diagnostics.
During the expeditions, vineyard blocks identified in advance were visited, known to harbour a significant prevalence of GTDs. To introduce an additional layer of variability, the selected plots encompassed a mix of conventional and organic cultivation methods, allowing the exploration of potential management influences on disease expression. The acquired images were captured during the early stages of development, precluding the identification of confounding diseases that may emerge at later phenological stages. Nevertheless, this enabling the creation of a robust dataset, predominantly focused on the primary target of GTDs, laying the foundation for accurate disease analysis.

Figure 7. Presentation of the acquisition device attached to the back of a van, remotely controlled via WiFi.

In this process, the acquisition device was set in "manual" mode, allowing an expert to identify and designate the diseased grapevines that needed to be photographed.

Subsequently, to ensure a well-rounded study, additional acquisitions were conducted in "automatic" mode to capture images of healthy grapevines. This involved systematically photographing all the grapevines in one or more rows. These images, captured through the automated process, provided a broader scope for analysis and comparison.

Once the data collection phase was complete, each image was scrutinized to determine whether it belonged to the 'GTDS' class, indicating the grapevine was affected by the disease, or to the 'Healthy' class, indicating the grapevine was disease-free. This classification process ensured the dataset was accurately labelled and served as a foundation for subsequent analysis and research.

Table 2 provides a comprehensive summary of the specific vineyard blocks, corresponding acquisition dates, cultivation methods employed, and the number of images acquired for both healthy grapevines and those exhibiting symptoms of GTDs. Figure 8 provides an overview of an image of each date of acquisition.

Table 2. Summary of the number of acquired images and associated ground truth at the grapevine *scale for GTDs.*

Acquisition date (2022)	**20/11**	**22/11**	**25/11**	**28/11**	**01/12**	**08/12**	**20/12**	**22/12**	**Total**
Management	Organic	Organic	Conventional	Conventional	Conventional	Conventional	Organic	No herbicide	
Total number of images	624	895	145	307	95	1898	2986	3355	10305
of which healthy	268	617	0	243	5	1052	2343	2947	7475
of which GTDs	356	278	145	64	90	846	643	408	2830

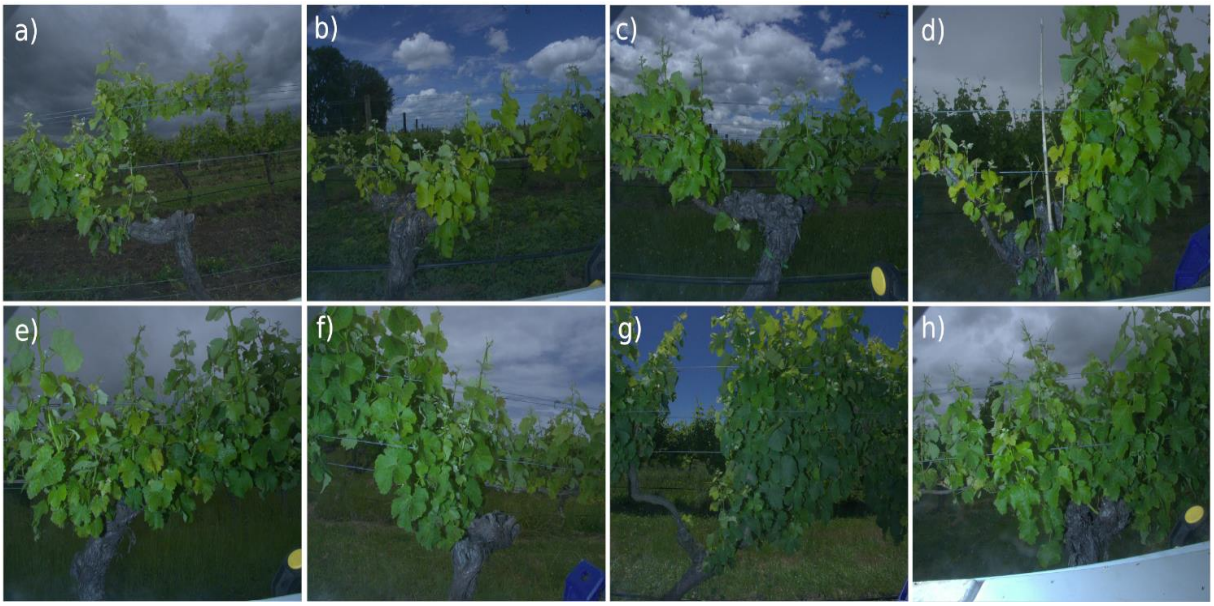

Figure 8. Images of GTDs from each acquisition dates: a) 20/11/2022; b) 22/11/2022; c) 25/11/2022; d) 28/11/2022; e) 01/12/2022; f) 08/12/2022; g) 20/12/2022; h) 22/12/2022.

D. Symptom annotations

There are several types of annotations used to detect objects in images. The type of annotation used depends on the type of object one wishes to detect, the complexity of the object, and the specific requirements of the data type. The most common object annotation methodologies are:

- Bounding boxes: This method involves surrounding the object with a rectangle (or a parallelepiped for 3D objects). The coordinates of the rectangle define the position and size of the object in the image.
- Semantic Segmentation: Instead of defining a bounding box, this method involves labelling each pixel of the image with the class of the object it belongs to. This allows for a more precise detection of the object's contours.
- Instance Masks: Similar to semantic segmentation, but instead of grouping all pixels of the same class, this method assigns a unique mask to each individual instance of the class.
- Keypoints: This method involves annotating specific points on the object of interest, such as corners, endpoints, or centres. These points can be used to reconstruct the shape and orientation of the object.
- Polygons: For objects with complex shapes, such as animals or characters, polygons are used to define the exact shape of the object using a series of interconnected points.

In order to best detect the symptoms of the two diseases (of different size and shape), several types of annotation have been tested. In this chapter, a comprehensive description of these annotation types is provided.

a. Annotation of FD symptoms

i. Bounding boxes for leaves

After acquiring the images, individual symptoms on leaves and bunches were annotated directly on the computer screen using bounding boxes. This method of annotation, employed by FD specialists, offered the advantage of being swift and straightforward. During the annotation process, the leaves were categorized into three classes: 'FD symptomatic leaf,' 'Esca leaf,' and 'Confounding leaf.' The 'Confounding leaf' class encompassed all leaves visually distinct from a healthy leaf. Annotate them in a separate class may allow the algorithms to better discern between different leaf types. To facilitate this annotation approach, the Labelme software (Wada, 2021) was used, which supports bounding box annotations (Figure 9).

The meticulous process of annotating each leaf individually proved time-consuming, often requiring up to 20 minutes for an image laden with symptoms. Nevertheless, the necessity of such detailed annotation was confirmed through rigorous testing, as omitting the 'Confounding leaf' class significantly compromised the accuracy of prediction results. The annotated dataset comprises images along with their corresponding symptom annotations. These annotations are saved in the '.json' format, providing comprehensive descriptions of each annotated symptom, including its label (symptom class), the coordinates of its bounding box or line strip ([[xmin, ymin], [xmax, ymax]]), and its shape type (bounding box or line strip). These annotations serve as resources for detection algorithms (presented in Chapter V).

Among the various symptoms of FD, leaf symptoms present the greatest challenge in differentiating them from confounding symptoms. To maximize the number of annotated leaves while considering time constraints, the decision was made to focus expert's annotation efforts primarily on leaves and not bunches.

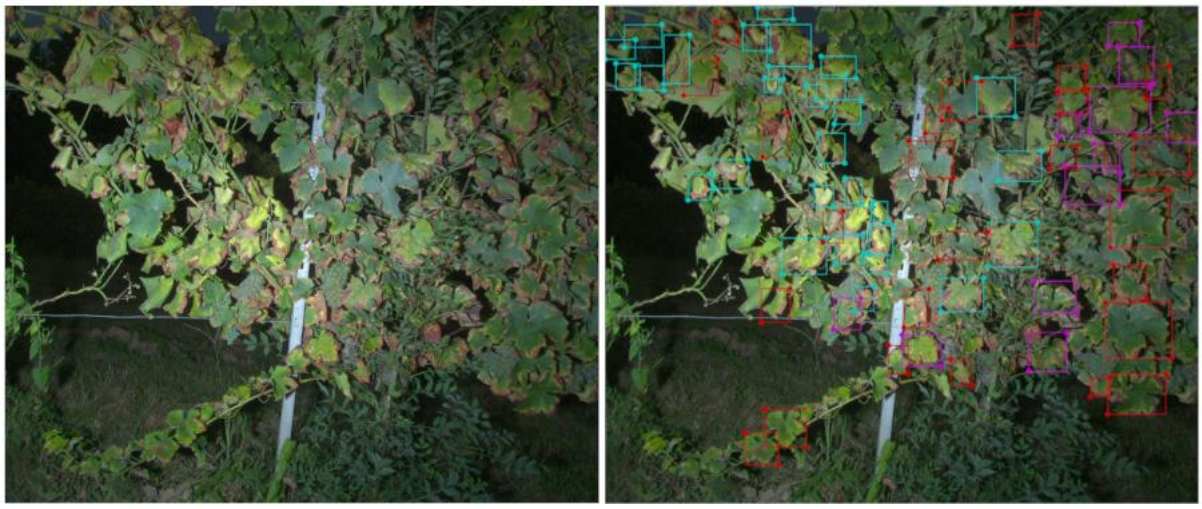

Figure 9. Example of annotation with bounding boxes of foliar symptoms. Presentation of a raw image (at the left) and its annotation with bounding boxes (at the right). In red: Symptomatic leaf of Esca; in blue: Symptomatic leaf of FD; in purple: Confounding leaf.

A summary of the number of annotated images and symptoms per grape variety and acquisition year is available in Table 3.

Table 3. Summary of images annotated with bounding boxes for the detection of FD symptoms, detailing each object class.

Grape variety, acquisition year	**Cabernet sauvignon, 2020**	**Ugni blanc, 2020**	**Cabernet sauvignon, 2021**	**Merlot, 2021**	**Cabernet franc, 2021**	**Ugni blanc, 2021**	**Sauvignon blanc, 2021**	**Total**
# of annotated images with	**67**	**221**	**116**	**43**	**86**	**161**	**50**	**744**
# of bounding boxes per class								
FD leaves	1086	2962	1101	475	1871	3155	629	11279
Esca leaves	299	425	357	0	274	758	354	2467
Conf. leaves	2956	3317	1473	2026	2930	7790	1929	22421

ii. Segmentation masks for shoots and bunches

Segmentation masks were manually created to further refine the annotations. This annotation methodology was chosen because it offers the highest level of precision in capturing the intricate shapes and contours exhibited by the shoots. Annotating them by bounding boxes was not possible, as shoots can take curved forms and the area covered by the shoots within the bounding box is relatively small. Consequently, there would have been a considerable amount of additional information present within the bounding box beyond just the shoot information.

A segmentation mask, also known as a pixel-level mask or annotation mask, is a visual representation that assigns a specific label or category to each pixel in an image. It is used to precisely

outline and identify different objects or regions of interest within an image. Each pixel in the segmentation mask corresponds to a pixel in the original image, and the assigned label or colour indicates the category or class to which that pixel belongs. The segmentation masks were created with the GIMP software (The GIMP Development Team, 2019).

Moreover, the choice to include symptomatic bunches in these masks was done, considering their limited occurrence in the expert-annotated images. These additional annotations of the 2 other symptoms of FD were performed on a separate dataset. It was carried out with the intention of achieving a comprehensive coverage of as many possible occurrences of these symptoms as possible. When dealing with machine learning tasks, it is crucial to have a well-balanced and representative dataset to train the model effectively. If there are only a few instances of a specific class or symptom in the dataset, the model may struggle to learn and generalize patterns accurately.

Given that symptomatic bunches, typically dried, often occupy only a few pixels in the images, challenges in accurate detection by algorithms were anticipated. To mitigate this issue, the healthy bunches were also annotated on the segmentation masks. Detecting healthy bunches tends to be comparatively easier, and their presence or absence may provide valuable insights into the presence of the disease. The symptomatic shoots maintain a consistent visual appearance across the images, irrespective of the grape variety, be it red or white. Annotating shoots from different grape varieties does not require attention due to the uniformity of this symptom among varieties. However, grape bunches, both symptomatic and healthy, exhibit colour variations based on whether they belong to red or white grape varieties. Therefore, the decision was made to focus annotation efforts on the red and white grape varieties with the largest available image datasets, namely Cabernet sauvignon and Ugni blanc. These annotations serve as resources for segmentation algorithms (presented in Chapter V). An example of a segmentation mask is presented in Figure 10, and the summary of the number of annotated images and symptoms per grape variety is provided in Table 4.

Figure 10. Example of annotation by segmentation mask of symptomatic shoots and bunches of FD and healthy bunches for a grapevine of Ugni blanc variety. Presentation of a raw image (at the left) and its associated annotation mask (at the right). Pixels in red: Symptomatic shoot; in green: symptomatic bunch; in blue: healthy bunch; in black: all the rest.

Table 4. Summary of the number of images annotated by segmentation mask for the symptom segmentation of FD.

Grape variety, acquisition year	**Cabernet sauvignon, 2020**	**Ugni blanc, 2020**	**Total**
# of annotated images with segmentation masks	**78**	**50**	**128**
# of annotated object per class			
Symptomatic shoot	575	258	833
Symptomatic bunch	348	120	468
Healthy bunch	441	219	660

b. Annotation of grapevine trunk diseases symptoms

i. Segmentation masks for foreground, trunk and foliage identification

Segmentation masks were created to facilitate the detection of disease symptoms. These masks were designed with three distinct classes: 'Foliage,' 'Trunk,' and 'Background'. An example of a segmentation mask with these three classes is provided in Figure 11. The purpose of this segmentation is to enhance the detection of the 'Half head' symptom by accurately delineating the grapevine trunk from the surrounding elements in the image. This separation is particularly crucial because the presence of background grapevines or shoots from neighbouring grapevines can create an optical illusion, making it appear as if the foliage is growing on both sides of the trunk. By employing a dedicated algorithm specifically designed for this task, the accuracy and efficacy of detecting this particular symptom can be improved.

Furthermore, the segmentation of foliage can also contribute to better detection of symptoms on the shoots, specifically the short shoots. This segmentation enables a clearer differentiation between the shoots and the background, resulting in improved identification and classification of symptoms affecting the shoots. By isolating the shoots from the background, the algorithm can focus on analysing the specific characteristics and abnormalities present in the short shoots, leading to enhanced detection and analysis of these symptomatic structures. Segmentation masks were created for images of the different dates of acquisition. The summary of the number of images annotated with segmentation mask per acquisition date is available in Table 5.

Figure 11. Example of a raw image (at the left) of a grapevine suffering from GTDs and its associated segmentation mask (at the right). Pixels of trunk have been set in white, foliage in red and all the rest in black.

Table 5. Summary of the number of images annotated by segmentation masks for the symptom segmentation of GTDs per date of acquisition.

Acquisition date (2022)	**20/11**	**22/11**	**25/11**	**28/11**	**01/12**	**08/12**	**20/12**	**22/12**	**Total**
# of annotated images with segmentation masks	29	9	5	5	34	11	11	9	113

ii. Bounding boxes for symptomatic organs

An annotation process using bounding boxes to identify and label the symptoms was conducted. Using the same software as the one used for the annotation of FD leaves, three classes of bounding boxes for GTDs symptoms were created: 'Half Head,' 'Short shoot,' and 'Canker' (Figure 12). This approach was suitable for annotating symptomatic shoots because their characteristic symptom is stunted growth, resulting in small, thin shoots without curved shapes. The annotations for GTDs symptoms underwent rigorous expert evaluation to ensure accuracy. Due to the minimal presence of confounding factors in this particular disease, it was not necessary to create a separate class for 'confounding symptoms' in the dataset. Instead, the focus remained on accurately annotating the specific symptoms associated with GTDs. To obtain a well-rounded annotated dataset that encompasses the diverse range of symptom expressions, annotations on a subset of images from each plot were performed. This approach aimed to capture the various manifestations of symptoms and provide comprehensive coverage across the dataset. Table 6 provides an overview of the number of annotated images and symptoms by bounding boxes.

Figure 12. Example of a row image (at the left) of a grapevine suffering from GTDs and its symptom annotation by bounding boxes. In red, symptoms of the class 'Short shoot', in yellow, symptom of the class 'Half head'.

Table 6. Summary of the annotation of GTDs symptoms by bounding boxes per image acquisition date.

Acquisition date (2022)	**20/11**	**22/11**	**25/11**	**28/11**	**01/12**	**08/12**	**20/12**	**22/12**	**Total**
# of annotated images with bounding boxes	172	141	27	21	12	0	69	63	505
# of annotated Half head	89	78	10	14	7	0	54	16	268
# of annotated Short shoot	695	451	62	13	20	0	12	227	1480
# of annotated Canker	73	34	11	12	2	0	6	1	139

E. Acquisitions at the vineyard block scale

a. Description of the acquisitions

As part of the automated detection of FD, acquisitions were conducted at the block level across three diverse vineyards in France, encompassing different regions and grape varieties. A description of these 3 vineyards is provided in Table 7. A comprehensive approach was taken, involving the photography and geolocation of all the grapevines within these designated parcels. Notably, both sides of each grapevine were captured in the images to ensure a thorough examination of potential disease symptoms. Symptoms such as leaf or shoot abnormalities may only be visible on one side of the grapevine due to the foliage arrangement during the specific time of observation. To establish reliable ground truth data, expert scouts were deployed to prospect the grape vineyards. They reported the geolocations of the affected grapevines, providing valuable reference points for subsequent analysis (as illustrated in Figure 13). This pre-established ground truth data served as a benchmark for the automated detection algorithms, facilitating the identification of diseased grapevines based on acquired imagery. To carry out the acquisitions, two acquisition devices were mounted on a quad

vehicle, and the acquisitions were efficiently conducted, capturing the necessary data while ensuring minimal disruption to the vineyard operations. This acquisition strategy aimed to gather comprehensive data from different regions and grape varieties, enabling analysis and the development of automated diagnosis algorithms for FD at the block scale.

Table 7. Summary of the acquisitions at the block scale for the diagnosis of FD.

Acquisition date	**20/09/22**	**23/09/22**	**09/09/22**
localisation	922.8, 5566875.2	44.651314, 0.024368	45.739181, -0.228742
Grape variety	Merlot	Cabernet sauvignon	Ugni blanc
Dataset name	Neuffons	Couture	Cognac
# of acquired images	8243	18 076	17 221
# of grapevines suffering from FD	61	1764	16

These three vineyards each had their own characteristics and distinguish themselves from one another. The 'Neuffons' vineyard contained two separate blocks, with Merlot grapevines and 61 cases of FD, where symptoms were sometimes very mild. The 'Cognac' vineyard featured Ugni blanc grapevines and only 16 grapevines affected by FD, the lowest number among the three vineyards. Additionally, numerous vines exhibited yellowing symptoms (leaves turning yellow without FD infection) whereas symptoms expressed by FD affected grapevine were also very mild. Lastly, the 'Couture' vineyard boasted a staggering 1764 vines afflicted with FD. Symptoms were very pronounced but here again, confounding diseases of FD were present. Grapevines were of the Cabernet sauvignon variety.

Although these 3 vineyards represented a limited number of study vineyards to evaluate the methodologies described earlier at the block scale, the diversity in grape varieties, number of vines affected by FD, and the presence or absence of confounding diseases provided an initial representative foundation of the diversities encountered in a real-world use case.

Figure 13. Aerial view of the three study vineyards displaying the positions of each acquired image (red dots) as well as the vines affected by FD (green triangles). Vineyard a) 'Neuffons' comprised two separate blocks but was treated as a single vineyard, vineyards b) and c) correspond to the vineyards named 'Cognac' and 'Couture'.

For each acquired image, a file was automatically filled in with the time the image was taken ('timestemp'), the latitude ('lat') and longitude ('lon') coordinates. A PPK correction was then applied to these coordinates. In addition to the PPK correction, the image coordinates were repositioned on

the grapevine and not at the camera location (Figure 14). Folders for saving images and files describing the properties of each image were created separately for each camera. A procedure was also integrated to prevent the camera from taking an image if there was no grapevine present (for example, in cases of already uprooted plants or changes in rows).

Figure 14. Illustration of the PPK correction and the repositioning of the coordinates at the location of the grapevine photographed. The yellow and pink dots are the uncorrected coordinates of images of the same grapevine row acquired on either side of the row. The orange and red points correspond respectively to the corrected coordinates of the yellow and pink points.

b. Creating a database of image triplets

When capturing images of a grapevine automatically at the block scale, images may not be centred on the targeted grapevine, or may not contain the entirety of that grapevine. For example, a long shoot carrying symptomatic leaves may not be fully captured in one image, but the missing part of this shoot can be found in the image acquired just before or just after. It is in this sense that the '3-neighbouring grapevines' dataset was developed for each vineyard.

The procedure to find the images to the left and right of each image was as follows:

1. For each targeted image, the camera that captured it was retrieved, and the file describing the image was opened. The ‘timestamp’ of the targeted image is then obtained, along with the image whose timestamp was just before it, and the one immediately after.
2. A timestamp threshold was applied to manage cases where an image was not captured right next or before to the one being studied (possible missing grapevine or the target image is the first or last in the row). If the timestamp difference is greater or less than 1 second (an image was acquired every 0.5 second), the acquired image was too far from the one being studied, and the image was not considered in the creation the image triplet.
3. It was necessary to determine the orientations in which the images were taken in order to properly align the images with each other. To achieve this, the timestamp, original position,

and corrected position of each image were required. There were four different configurations, as illustrated in Figure 15. The timestamp first indicated the direction of camera movement. To determine which side of the grapevine the camera was on, the uncorrected coordinates of the target image (Figure 14) indicated the camera's position relative to the grapevine.

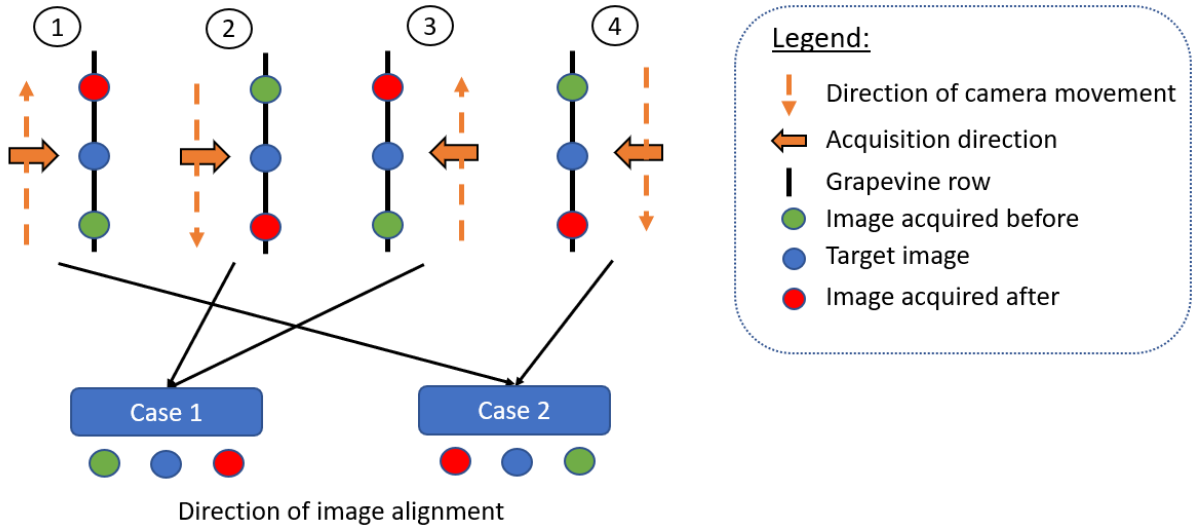

Figure 15. The 4 scenarios of image capture and their impacts on the reconstruction orientation.

c. Creating a database of image sextuplets

This database allowed for the full potential of dense acquisitions made at the parcel scale to be realized by considering, as in the dataset of images triplets, the images of the preceding and following grapevines of the targeted one, but also the images acquired on the opposite side of these 3 grapevines.

The procedure to create the '6-neighbouring grapevines' database was as follows:

1. For the image of the studied grapevine, the three steps of the procedure to create the image triplets database were applied to locate the images of the vines to the left and right if they existed.
2. The image of the opposite side of the studied grapevine was found by searching for the image with the closest corrected latitude and longitude coordinates (Figure 14). For this opposite face, the images of the vines to the left and right were located in the same way as in step 1).
3. To ensure that the symptoms on each side were aligned as closely as possible, considering that the images may not have been taken from the exact same positions on each side of the grapevine, this offset was taken into account. By examining a sample of images (each plot was treated independently), the average overlap in terms of pixel width between two consecutive images was calculated. Similarly, the average difference in latitude and longitude between two consecutive images was also calculated. Consequently, an estimation of how a difference in corrected latitude and longitude (as they are placed on the same line) impact the difference in terms of pixels. This was calculated as follows: if the average width of overlap was X pixels, the distance in pixels between two consecutive images is 2448 (the width in pixels of the images) minus 2*X. The positions of the images on the opposite face were therefore updated to account for the offset in corrected latitude and longitude between those of the target image and its opposite face.

F. Summary of the datasets

A concise overview of the acquisitions made and the diverse datasets created for the purpose of symptom detection and diagnosis of the two diseases is presented in this section. Specifically, Table 8 provides a comprehensive summary of these acquisitions and annotations related to the diagnosis of FD, while Table 9 focuses on the different datasets related to the diagnosis of GTDs.

Table 8. Summary of all acquisition and split of the data among tasks for FD automated diagnosis.

Purpose	Name of the dataset	Number of images	Details
Acquisition at the grapevine scale		1483	
Segmentation mask annotations	SegFD	128	
Bounding box annotations	DetFD	568	
Grapevine scale annotation	ClaFD	787	
Block scale annotation	BloFD	43540	

dataset	FD shoot	FD bunch	Healthy bunch
CS20	575	348	441
UB20	258	120	219

dataset	FD leaf	Esca leaf	CONF leaf
CS20	1086	299	2956
UB20	2962	425	3317
CS21	225	104	412
M21	475	0	2026
CF21	1376	236	2231
UB21	2479	534	4890
SB21	629	354	1929

dataset	FD	Esca	CONF	CONF+
CS20	72	45	87	56
UB20	83	13	88	8
CS21	86	2	6	0
M21	32	5	7	11
CF21	23	3	2	5
UB21	33	12	4	0
SB21	41	44	17	2

dataset	Variety	FD vines
Neuffons	Merlot	61
Couture	Cabernet sauvignon	1764
Cognac	Ugni blanc	16

Image-level acquisitions and symptom annotations by bounding boxes and segmentation masks for FD diagnosis have been published (Tardif *et al.*, 2023). The datasets 'SegFD,' 'DetFD,' and 'ClaFD' are publicly available, as indicated in the publication.

As detailed in Tables 8, separated class were created for Esca for both image and symptom scales diagnosis. Consequently, it became possible, alongside the study of FD automated diagnosis, to obtain results for the automated diagnosis of Esca. The results presented in the upcoming chapters

will be exclusively for FD, the targeted multi-symptom disease. However, all results related to Esca are available, detailed and discussed in Appendix 2.

Table 9. Summary of all acquisition and split of the data among tasks for GTDs automated diagnosis

Purpose	Name of the dataset	Number of images	Details
Acquisition at the grapevine scale		10305	
Segmentation mask annotations	SegGTD	113	**dataset** / **# of annotated images**: **20_11** 29; **22_11** 9; **25_11** 5; **28_11** 5; **01_12** 34; **08_12** 11; **20_12** 11; **22_12** 9
Bounding box annotations	DetGTD	505	**dataset** / **Half head** / **Short shoot** / **Canker**: **20_11** 89 / 695 / 73; **22_11** 78 / 451 / 34; **25_11** 10 / 62 / 11; **28_11** 14 / 13 / 12; **01_12** 7 / 20 / 2; **08_12** 0 / 0 / 0; **20_12** 54 / 12 / 6; **22_12** 16 / 227 / 1
Grapevine scale annotation	ClaGTD	9687	**dataset** / **GTDs** / **Healthy**: **20_11** 155 / 268; **22_11** 128 / 617; **25_11** 113 / 0; **28_11** 38 / 243; **01_12** 44 / 5; **08_12** 835 / 1052; **20_12** 563 / 2343; **22_12** 336 / 2947

IV. One-step diagnosis using Convolutional Neural Networks

Table of content

In the initial phase of the research, the objective was to thoroughly examine the widely adopted methodology found in the existing literature for diagnosing grapevine diseases (Boulent *et al.*, 2020; Ji *et al.*, 2020; B. Liu *et al.*, 2020). This entailed leveraging the power of CNNs in image processing and assess their capability to detect mutli-symptom grapevine diseases by directly providing them with the entire image. By utilizing CNNs, the aim was to extract meaningful insights from grapevine disease images and potentially establish a robust diagnostic framework.

The results obtained through this initial phase have served as a baseline for subsequent investigations. Building upon this foundation, innovative two-step approaches that go beyond this conventional method have been explored.

A. Selection of baseline algorithms for 1-step classification of grapevine images

a. About Convolutional Neural Networks (CNNs)

A CNN is a type of deep learning model specifically designed for processing and analysing visual data such as images (LeCun *et al.*, 2015). The key building blocks of a CNN are convolutional layers, pooling layers, and fully connected layers. In the convolutional layers, filters (also called kernels) are applied to the input image to extract relevant features that indicate for instance the presence of edges, corners or textures. The pooling layers downsample the spatial dimensions, reducing the computational complexity, allowing feature extractions at different scales and providing some degree of translational invariance. Finally, the fully connected layers process the extracted features to make predictions, such as classifying an image into different categories. Figure 16 represents these different blocks (typically, multiple convolutional/pooling blocks are used in succession).

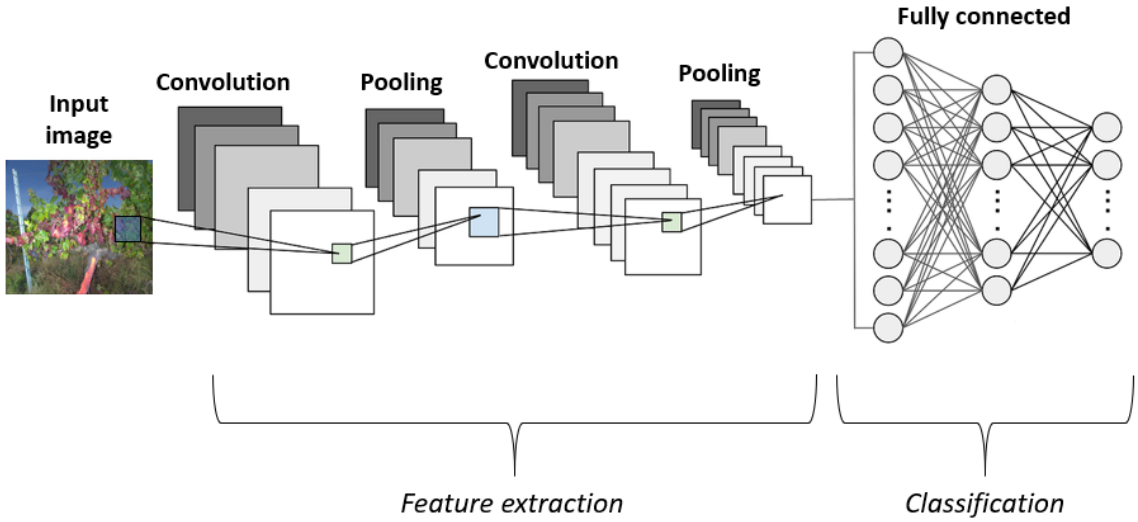

Figure 16. Basic workflow of a CNN for image classification. It consists of key layers, including convolutional and pooling layers, as well as a fully-connected layer responsible for delivering the final predicted class.

CNNs are trained through a process called backpropagation, where the network adjusts the weights of the filters and connections to minimize the difference between predicted outputs and ground truth labels. This difference is calculated by a loss function. The choice of the loss function depends on the specific task being performed and the goal is to minimize the loss value, as it indicates how well the model is performing on the training data. This training is typically performed using large labelled datasets.

One of the key advantages of CNNs is their ability to automatically learn hierarchical representations of data. The early layers capture low-level features, while deeper layers learn high-level and more abstract features. This hierarchical feature extraction allows CNNs to excel at tasks like image classification (Jmour *et al.*, 2018), object detection (Girshick, 2015; W. Liu *et al.*, 2016; Redmon *et al.*, 2016), image segmentation (Ajmal *et al.*, 2018; Geng *et al.*, 2017), and even more complex tasks like visual question answering (Anderson *et al.*, 2018; Chen *et al.*, 2016) and image generation (S.-Y. Wang *et al.*, 2020). In the following, the abbreviation CNN will be used to refer to CNNs used for image classification.

CNNs are powerful deep learning models tailored for visual data analysis. They leverage convolutional layers to extract features, pooling layers to downsample the data, and fully connected layers for final prediction. CNNs have had a significant impact on computer vision tasks and continue to drive advancements in the field. They emerge as a pertinent choice for the automated detection of grapevine diseases in image analysis.

b. Choice of three architectures

CNNs can have different architectures, each with its own characteristics and design choices. Each architecture has its own strengths and trade-offs, depending on the specific task and computational requirements. Researchers continuously explore new architectures and variations to improve the performance and efficiency of CNNs for various computer vision tasks (Bhatt *et al.*, 2021).

Three state-of-the-art CNN architectures have been selected for the study, each with different combinations of parameters and depths. This deliberate choice enabled to perform a comprehensive

comparative analysis of their respective outcomes. This enabled to optimize the selection in terms of network and architecture and potentially enhance computation times.

The first one, MobileNetV3-large (Howard *et al.*, 2019), is a CNN architecture specifically designed for efficient inference on mobile and embedded devices with limited computational resources. It is an extension of the MobileNet family of architectures (Howard *et al.*, 2017), aiming to further reduce model size and improve performance. Here are some notable features of MobileNetV3-large:

- Efficient architecture: MobileNetV3-large employs a combination of depthwise separable convolutions and inverted residual blocks. Depthwise separable convolutions split the convolution operation into depthwise and pointwise convolutions, reducing the computational complexity. Inverted residual blocks introduce a linear bottleneck layer followed by a non-linear activation function, enabling efficient feature extraction.
- Neural architecture search (NAS): MobileNetV3-large incorporates Neural Architecture Search techniques to automatically discover optimal network configurations. This search process helps to identify efficient network architectures with improved accuracy.
- Activation functions: MobileNetV3-large introduces novel non-linear activation functions such as h-swish and h-swish with a hard sigmoid, which provide a good balance between non-linearity and computational efficiency.
- Squeeze-and-Excitation (SE) blocks: MobileNetV3-large incorporates SE blocks, which selectively recalibrate feature maps by adaptively scaling channel-wise information. This mechanism enhances the representation power of the network and improves performance.

The overall goal of MobileNetV3-large is to provide lightweight and efficient models that can be deployed on resource-constrained devices without compromising too much on accuracy.

It was chosen as the initial CNN to test, as achieving good results would eliminate the need for CNNs with a higher parameter count. Such models would require more time for inference and pose challenges in terms of integration into acquisition devices. MobileNet have multiple mode variants. The MobileNetV3-large, which focuses on achieving higher accuracy at the expense of a larger model size (only around 5 million parameters), has been chosen for the study. Figure 17 describes the succession of neural layers of the model. The input has a shape of 224*224*3 and the output is a classification among k classes. Between these layers, simple convolutional layers and bottleneck layers coming from MobileNetV2 (Inverted Residual and Linear Bottleneck) (Sandler *et al.*, 2018), either with Squeeze-and-Excitation (SE) blocks or not.

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Figure 17. Specification for MobileNetv3-large (Jain, 2019). SE denotes whether there is a Squeeze-And-Excite in that block. NL denotes the type of nonlinearity used. HS denotes h-switch and RE denotes ReLU. NBN denotes no batch normalization. s denotes stride.

The second CNN studied is called ResNet50 (He *et al.*, 2016). ResNet50 is a popular deep learning model that belongs to a family of models called Residual Neural Networks (ResNets). It was introduced by Microsoft Research in 2015 and has since been widely used for various computer vision tasks.

The main idea behind ResNet50 is to address the problem of vanishing gradients in very deep neural networks. As the network gets deeper, the gradients tend to become extremely small, making it difficult for the network to learn effectively. ResNet50 addresses this problem by introducing a "skip connection" or a "shortcut connection" that allows the gradient to bypass one or more layers and directly propagate from one layer to a later layer. ResNet50 architecture is presented in Figure 18.
Here's how ResNet50 works at a high level:

- Convolutional Layers: The input image passes through a series of convolutional layers, which are responsible for extracting visual features from the image. ResNet50 uses a combination of 1*1 and 3*3 convolutions to capture different levels of spatial information.
- Residual Blocks: The core building block of ResNet50 is the residual block. A residual block consists of two or three convolutional layers with a shortcut connection. Each residual block learns to model the residual mapping instead of directly trying to learn the desired underlying mapping. This allows the network to effectively learn the incremental changes required to improve the performance.

- Stacking Blocks: ResNet50 consists of several stacked residual blocks. The number of blocks can vary depending on the specific ResNet variant (e.g., ResNet18, ResNet34, ResNet50, etc.). ResNet50 has 50 layers in total, hence the name.
- Global Average Pooling: After the residual blocks, the network applies global average pooling, which averages the spatial dimensions of each feature map to obtain a fixed-length feature vector.

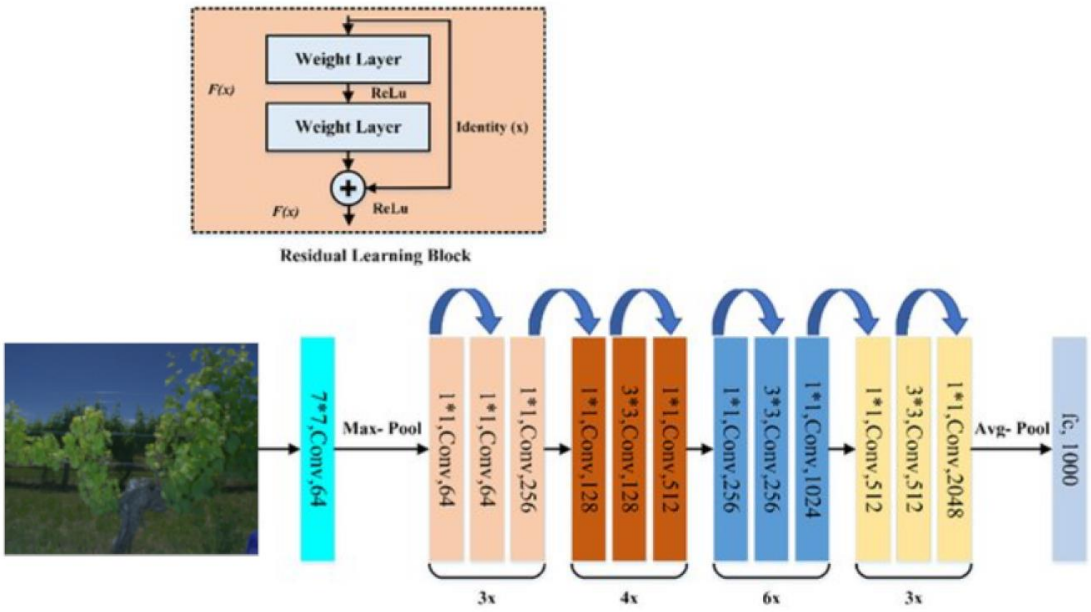

Figure 18. The architecture of ResNet-50 model (Ali et al., 2021).

Overall, ResNet50's depth, residual connections, strong performance, transfer learning capabilities, and versatility make it a good choice for image classification tasks. ResNet50 uses approximately 25 million parameters.

The third CNN studied, EfficientNetB5, is deeper (312 layers) and with more parameters (about 30 million) than the others. EfficientNetB5 is a CNN architecture that is part of the EfficientNet family. It was introduced by Tan & Le (2019) as an extension to the EfficientNet models, which are known for their superior performance in image classification tasks while maintaining efficiency in terms of model size and computational cost. Higher versions exist, but they were not chosen because they were even deeper, had more parameters, and consumed too many resources during training. EfficientNetB5 is designed to handle more complex visual recognition tasks that require a higher level of feature extraction and representation. Here are some key features of EfficientNetB5:

- Depth and Width: EfficientNetB5 has a greater depth and width compared to earlier EfficientNet models. It consists of multiple stacked layers, with increasing depth as the network progresses. The increased width helps capture more fine-grained details in the input images.
- Compound Scaling: Similar to other EfficientNet models, EfficientNetB5 incorporates compound scaling to balance model size and performance. It scales the depth, width, and resolution of the network in a principled manner, optimizing the trade-off between accuracy and computational efficiency. This allows EfficientNetB5 to achieve high accuracy without excessively increasing the model size or computational requirements.
- Convolutional Blocks: EfficientNetB5 utilizes various convolutional blocks, including depthwise separable convolutions, bottleneck structures, and squeeze-and-excitation modules. These blocks enhance the efficiency and effectiveness of feature extraction, enabling the network to capture complex patterns and dependencies in the input data.

The architecture of EfficienNetB5 is presented in Figure 19.

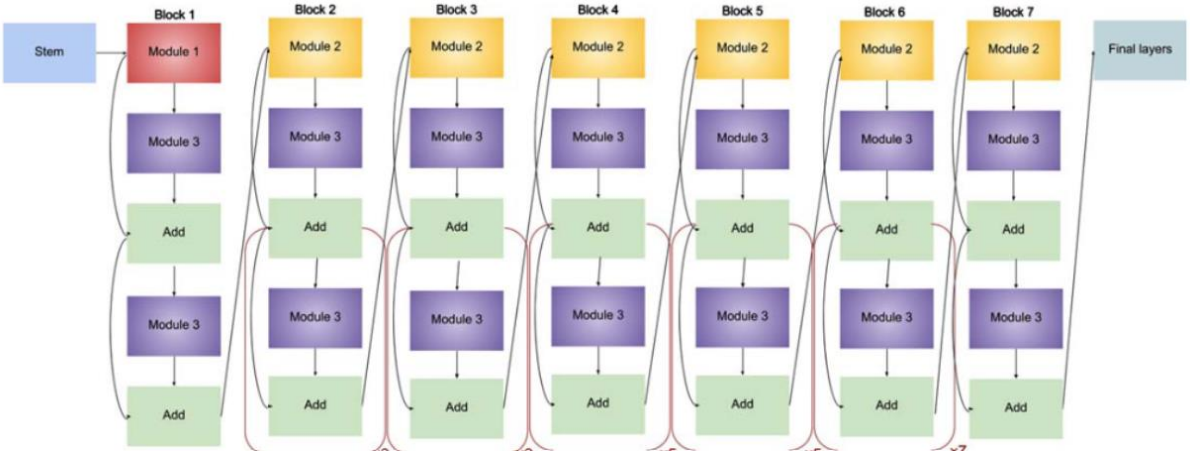

Figure 19. Architecture of EfficientNetB5 (Agarwal, 2020).

EfficientNetB5 has demonstrated impressive performances on various image classification benchmarks, surpassing many state-of-the-art models in terms of accuracy while still maintaining a reasonable model size and computational cost. It is particularly beneficial in scenarios where accuracy is crucial, but computational efficiency is also a consideration.

A summary of the different chosen CNNs is provided in Table 10.

Table 10. Summary of the tested CNNs MobileNetV3-large, ResNet50 and EfficientNetB5 for the automated diagnosis of FD and GTDs.

Architecture	Top 1 accuracy*	# of parameters	Depth**
MobileNetV3-large	75.2%	5.4 M	105
ResNet50	75.8%	25.6 M	107
EfficientNetB5	83.6%	30.5 M	312

**Refers to the model's performance on the ImageNet validation dataset*
***Number of layers with parameters*

c. A common training methodology

In this section, a comprehensive overview of the methodology employed to train the different CNNs is presented. A detailed description of deep learning algorithm training is offered in Appendix 3.

The fine-tuning technique was employed during the training of these algorithms. Fine-tuning played a crucial role in training these algorithms by adapting pre-trained models to the specific task at hand. This technique involves taking a pre-trained CNN model, originally trained on a large dataset, and fine-tuning it on a new, smaller dataset or a different task. By leveraging the pre-trained model's learned representations, fine-tuning allows for improved performance and faster convergence on the new task. Instead of starting from scratch, the fine-tuning process builds upon the existing knowledge encoded within the pre-trained model.

Here is an overview of the fine-tuning process for a CNN:

- Select a pretrained model: Choose a pre-trained CNN model that was trained on a large-scale dataset, such as ImageNet (Deng *et al.*, 2009). The choice of the model depends on the specific requirements of the new task and the availability of pre-trained models.

- Freeze initial layers: Initially, the pre-trained model's layers are frozen, which means their weights and parameters are not updated during training. This freezing is done to preserve the learned representations from the original task and prevent the model from forgetting what it has already learned.
- Modify output layers: Replace or modify the final fully connected layers of the pre-trained model to match the number of classes or the requirements of the new task.
- Training: With the modified output layers in place, the fine-tuning process involves training the model on the new dataset. The new dataset typically has a smaller number of samples compared to the original dataset. During training, the weights of the modified output layers are updated to learn the task-specific features while keeping the initial layers frozen.
- Gradual Unfreezing: After some initial training on the modified output layers, a technique called gradual unfreezing can be applied. Gradual unfreezing involves unfreezing and selectively fine-tuning some of the earlier layers of the pre-trained model, allowing them to adapt to the new task. This process helps the model to generalize better and capture task-specific features.
- Training Parameters: Fine-tuning involves setting the appropriate hyperparameters such as learning rate, batch size, and regularization techniques like dropout or weight decay. These parameters may need to be tuned to achieve optimal performance on the new task.
- Evaluation and Testing: Once training is completed, the fine-tuned model is evaluated on a validation set to measure its performance. Additional adjustments and fine-tuning may be performed based on the evaluation results. Finally, the model can be tested on a separate test set to assess its performance on unseen data.

The added output layers at the top of these algorithms were the same for all three models. They consisted of a GlobalAveragePooling2D layer following the pre-trained model, followed by a dense layer with 64 neurons using the 'ReLU' non-linear activation function. Predictions were then obtained through a final dense layer using the 'sigmoid' activation function for binary classification (GTDs) or 'Softmax' activation function for three-class prediction (FD). The 'Binary_crossentropy' loss function was used to train the models for binary classifications, while 'Sparse_Categorical_crossentropy' was used for three-class predictions.

The hyperparameters are consistent across all three CNNs: 100 epochs (complete pass of the model through the entire training dataset) with a batch size (determines how many data are used to update the model's parameters in one optimization step) of 10. There was no extensively fine-tuning of the hyperparameters. To ensure fair comparison among the CNNs, uniform hyperparameter settings across all models were chosen.

To evaluate the performances of these algorithms, precision and recall metrics are computed for each class. The precision and recall are commonly used measures to evaluate the performance of a classification model. Precision is a measure of the model's ability to provide trustworthy results for a specific class. It is calculated by dividing the number of true positives by the sum of true positives and false positives. In other words, precision measures the proportion of positive results given by the model that are actually correct. Recall, also known as sensitivity or true positive rate, is a measure of the model's ability to correctly identify all positive examples of a given class. It is calculated by dividing the number of true positives by the sum of true positives and false negatives. In other words, recall measures the proportion of actual positive examples that were correctly identified by the model.

These CNNs were developed in Python using the TensorFlow library, designed for the development, training, and testing of artificial intelligence algorithms. The CNNs were pre-trained on ImagetNet. The fine-tuning was performed on an NVIDIA GeForce RTX 3090 graphics card.

B. Datasets

a. Grapevine trunk diseases

To conduct a comprehensive analysis of the influence of training data size and data uniformity (images taken at different growth stages), two distinct datasets for CNN-based image-level classification of GTDs-infected grapevines (Table 11) were created. This division allows the comparison of one dataset with fewer images but less variability in the growth stage and another containing many more images but greater variability in symptom expression:

- The first dataset, named 'ClaGTD_till25', is composed of three subsets ('20_11', '22_11', and '25_11') extracted from the larger 'ClaGTD' dataset. To ensure a balanced representation of classes, only 200 images of healthy vines from the '22_11' subset were selected. Consequently, this dataset comprises 416 images of GTDs-infected grapevines and 468 images of healthy grapevines.
- For the second dataset, the complete 'ClaGTD' dataset was used with the exception of achieving class balance. 200 images of healthy grapevines from the '22_11' subset, along with 200 images from '08_12', 500 images from '20_12', and another 500 images from '22_12' were randomly sampled. As a result, this dataset encompasses 2212 images of DTD-infected grapevines and 1916 images of healthy vines.

Table 11. Summary of the two studied datasets for the automated diagnosis of GTDs by CNNs.

Dataset name	Acquisition period	# of GTDs	# of Healthy	Symptom variability
'ClaGTD_till25'	20/11/22 – 25/11/22	416	468	low
'ClaGTD'	20/11/22 – 22/12/22	2212	1916	high

b. Flavescence dorée

To enhance the automated diagnosis of FD, the study took advantage of a larger variety of images. Multiple datasets were examined to investigate different aspects (described in Table 12), with the most important ones being the impact of confounding factors (‘CONF+’ images) and the grape variety.

- Firstly, the impact of images from the ‘CONF+’ class on algorithm performance was explored for specific grape varieties. This analysis was conducted on two subsets: ‘CS20’ for red grape varieties, which contains a substantial number of ‘CONF+’ images, and ‘UB20’ for white grape varieties, which had a lower quantity of ‘CONF+’ images but provided sufficient training data for the training on one grape variety. The performance of algorithms trained on these subsets was evaluated using cross-validation within the same datasets. Additionally, the trained models were tested on images from other grape varieties, taking into account the colour of the grape variety. This allowed for an assessment of how the grape variety influenced the inference results.
- A series of trainings and tests were carried out subsequently on grape varieties sharing the same colour. The objective was to determine the ability of CNNs to generalize when presented with various expressions of symptoms during the training phase. This was particularly important because different grape varieties may exhibit distinct manifestations of symptoms despite belonging to the same colour category.
- Finally, an experiment was conducted to evaluate the possibility of creating a unified model that encompasses both red and white grape varieties. The datasets containing images of both grape

colours were combined for training. This approach aimed to assess whether a single model can effectively handle the diagnosis of FD across grape varieties of different colours.

By systematically examining these different training and testing scenarios, the study aimed to gain insights into the effects of image diversity, grape variety, and the inclusion of 'CONF+' images on the performance of automated FD diagnosis algorithms.

Table 12. Training and testing scenarios for the automated diagnosis of FD by CNNs

Training set	Testing set	Usage	% of 'CONF+' (training set ; testing sets)
CS20 without 'CONF+'	CS20 without 'CONF+'	Training and testing on the same grape variety without the 'CONF+' images	(0%; 0%)
CS20	CS20	Test the impact of the 'CONF+' images	(22%; 22%)
CS20 without 'CONF+'	(CS21, CF21, M21) without 'CONF+'	Testing grape variety change resilience (red grape varieties)	(0%; 0%, 0%, 0%)
UB20	UB20, UB21, SB21	Testing grape variety change resilience (white grape varieties)	(4%; 4%, 0%, 2%)
CS20 + CS21 + CF21 + M21	CS20 + CS21 + CF21 + M21	Training and testing on the entire set of red grape varieties	(16%; 16%)
UB20 + UB21 + SB21	UB20 + UB21 + SB21	Training and testing on the entire set of white grape varieties	(3%; 3%)
ClaFD	ClaFD	Training and testing with both red and white grape varieties	(10%; 10%)

c. Data pre-processing

Three types of data preprocessing were tested on the images before training the CNNs. Firstly, the images were loaded as they were, with a size of 2048*2448*3 pixels. Processing the images directly at this size would preserve the most complete information as input for the algorithms. However, this would result in longer training and prediction times, and the models might receive more information than necessary to accurately predict the image class. Moreover, the average image resolution on which the algorithms have been pre-trained (ImageNet dataset) is 469*387 pixels. Therefore, resolution degradation techniques were tested. Reductions by a factor of 16, 64, and 256 in the number of pixels were tested by selecting every 4th column and row (8 and 16, respectively). These different degradations allowed for studying the optimal approach and making choices based on computation times. Prior to this selection, a Gaussian filter was applied to the images to ensure that the retained pixels also carried the information of the discarded pixels. The input images for the algorithms were then of dimensions (512*640*3), (256*320*3), and (128*160*3). Padding of the images (adding pixels

with a value of 0) was performed to ensure that the images could be properly processed by the algorithms.

The images were subsequently divided into training, validation, and testing sets using the cross-validation method. Cross-validation is a widely adopted technique in machine learning for assessing model performance and generalization. It allows us to evaluate how well a model will perform on unseen data by splitting the available data into multiple subsets, known as folds. The dataset was divided into K equal-sized folds, where K is typically determined based on the dataset size and computational resources. For each fold, the model was trained on K-1 folds (the training set) and evaluated on the remaining fold (the test set). This process was repeated K times, with each fold serving as the test set once. The final metrics are averaged over the K test folds. Data were split into 5 folds, resulting in 80% of the data allocated to the training set and 20% to the test set. Additionally, 20% of the training data was set aside as the validation data. Validation data refer to a separate dataset, distinct from the training data, used to evaluate the performance of a model during training Consequently, the final distribution across training/validation/testing sets was as follows: 56% in the training set, 24% in the validation set, and 20% in the test set. Furthermore, to maintain proportional representation of each class across the different sets, the technique of stratification was employed. Stratification ensures that the distribution of classes remains the same in the training, validation, and testing sets, which is particularly important for classification tasks.

Following the preprocessing stage, the training images underwent a data augmentation process. Data augmentation is a widely used technique in deep learning to increase the size and diversity of training datasets. Its aim is to improve model performance and generalization by introducing synthetic variations into the training data, simulating real-world scenarios and enabling models to better generalize. Data augmentation techniques can include operations such as rotation, resizing, cropping, horizontal/vertical flipping, zooming, adding noise, adjusting contrast and brightness, etc. These geometric and colour transformations increase the variability of images and expose the model to different perspectives and lighting conditions. The augmented data is used alongside the original data during model training. This exposes the model to a greater variety of situations and variations, promoting better generalization and robustness to real-world data. Data augmentation helps prevent overfitting by providing more diversity in the training data, reducing the model's sensitivity to minor variations in the training examples. It can also address class imbalances by creating synthetic examples of minority classes. The same data augmentation process was applied during all training sessions. It consisted of a combination of operations applied randomly to each image, including vertical flipping, random cropping (0-10% of the image), adding Gaussian noise, setting 1% to 5% of pixels to 0, setting 3% to 10% of pixels to 0 in square patches, adding a value of -10 to 10 to each pixel channel, changing brightness (70-130% of the original value), adjusting contrast (70-130% of the original value), affine transformations including zooms or dezooms (80%-120% of the original value), vertical and horizontal translations (up to 20% of pixels in one direction), and rotation (-45 to 45 degrees).

C. Results

a. Grapevine trunk diseases

In the study of automated diagnosis of GTDs, two different datasets were used. These datasets were subjected to experiments involving three pre-processing techniques and three CNNs. The three pre-processing techniques applied to the datasets involved dividing the number of pixels by 16, 64, and 256, respectively (Figure 20). This division aimed to reduce the dimensionality of the images and potentially enhance the features relevant for GTDs diagnosis as well as reducing the inference time. Each pre-processing technique was applied to both datasets separately.

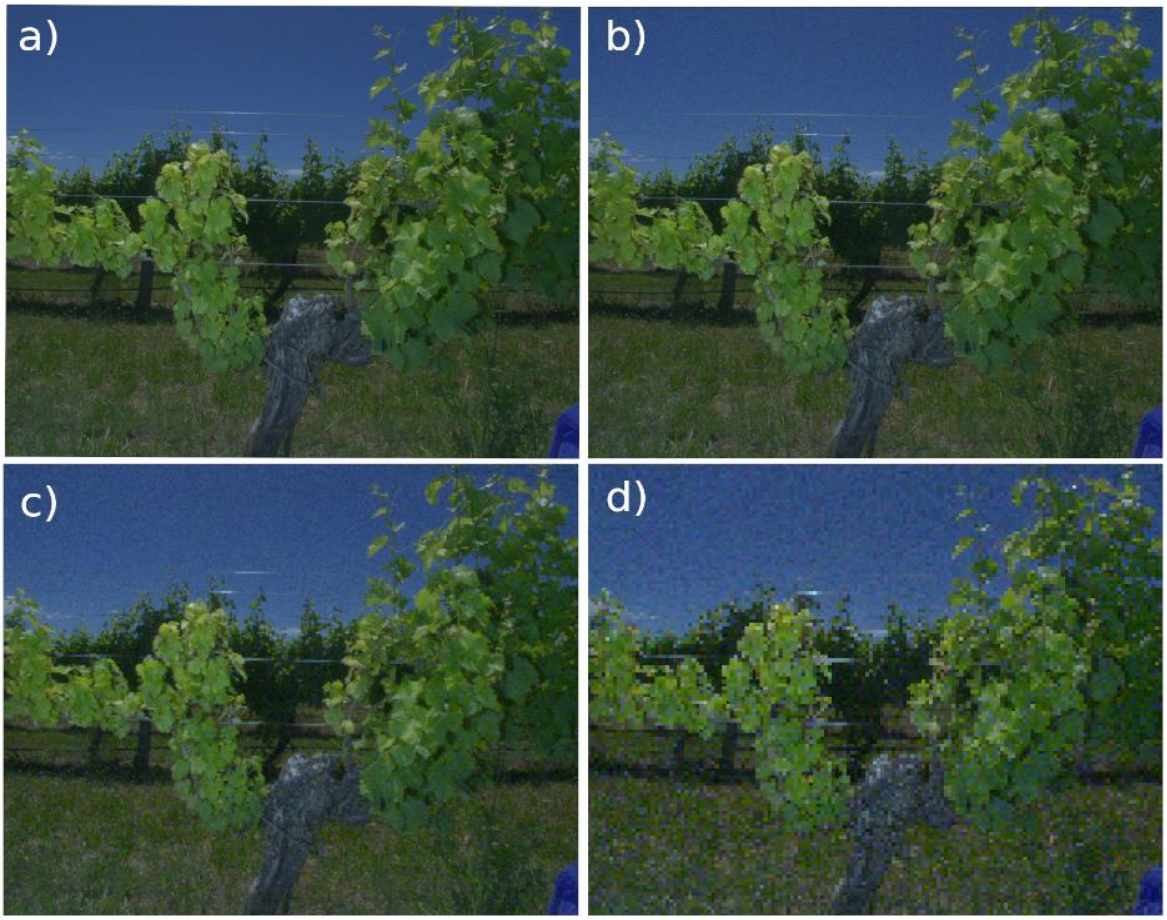

Figure 20. Illustration of the different resolution degradation: a) The original image; b) resolution divided by 8; c) by 64; d) by 256.

The CNNs were trained on the pre-processed datasets, and their performance were evaluated using a cross-validation approach. The results obtained from the experiments are presented in Table 13. It is important to note that the values displayed in the table represent the average performance metrics calculated from the results on the 5 test sets of the 5-fold cross-validation. A plot of the loss and accuracy curves for the methodology reaching the best results is presented in Figure 21.

Table 13. Summary of the results of the 5-fold cross-validation for each combination of dataset, preprocessing and CNN for the GTDs diagnosis.

Dataset	Pre-process	MobileNetV3-large (p, r)* 'GTDs' class	ResNet50 (p, r) 'GTDs' class	EfficientNetB5 (p, r) 'GTDs' class
ClaGTD_till25	/16	(0.82, 0.77)	**(0.88, 0.83)**	(0.87, 0.82)
	/64	(0.86, 0,78)	(0.85, 0.85)	(0.87, 0.83)
	/256	(0.82, 0.83)	(0.85, 0.8)	(0.83, 0.79)
ClaGTD	/16	(0.9, 0.92)	**(0.94, 0.92)**	(0.92, 0.93)
	/64	(0.89, 0.92)	(0.91, 0.93)	(0.9, 0 .89)
	/256	(0.87, 0.89)	(0.86, 0.89)	(0.87, 0.83)

**(precision, recall)*

The results showed that the ResNet50 architecture achieved the best performance on the first dataset when the input image resolution was downscaled by a factor of 16. The precision and recall for the 'GTDs' class reached p=0.88 and r=0.83 respectively. Interestingly, there was no substantial disparity in performance among the three architectures when using different preprocessing methods, such as downsampling the image resolution by factors of 16 or 64. The crucial information for diagnosing this disease in the images, such as the distinct symptoms like 'half head' or 'short shoot', remains clearly visible and easily recognizable by the algorithms even in low resolution images. These important visual cues enable the algorithms to effectively identify and classify the presence of these specific symptoms, aiding in accurate disease diagnosis. However, the achieved results were relatively low compared to those considering the entire dataset.

When considering the entire dataset, which encompassed more images and a greater variability in disease expression, results were better for all the CNNs. The ResNet50 architecture with a downsampling factor of 16 for image resolution yielded the best results. This preprocessing approach consistently outperformed the others across all three architectures. The precision-recall pairs for the 'GTDs' class were notably higher, reaching (p=0.90, r=0.92), (p=0.94, r=0.92), and (p=0.92, r=0.93) for MobileNetV3-large, ResNet50, and EfficientNetB5 architectures, respectively.

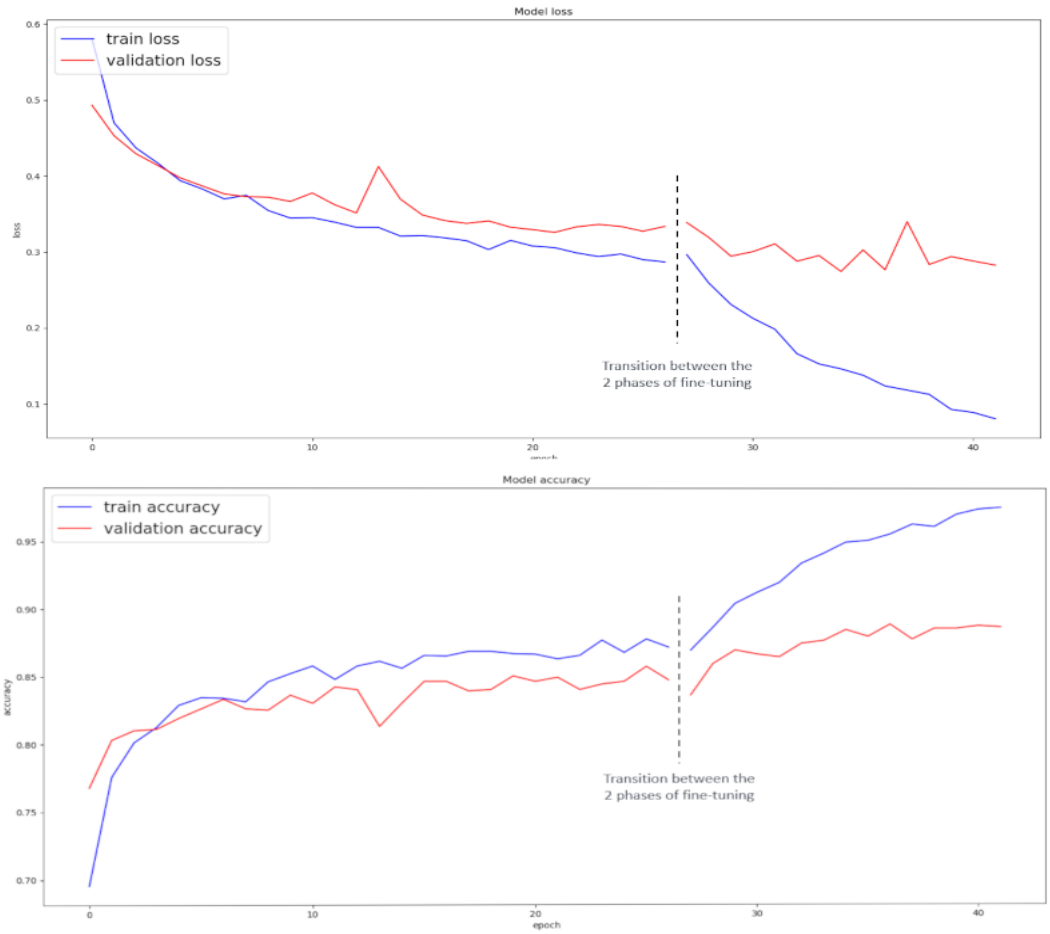

Figure 21. Loss and accuracy curves for the Resnet50 architecture with a 1/16 reduction in input image resolution on the training and validation sets for the GTDs diagnosis.

Figure 21 illustrates the impact of the two phases in the fine-tuning process. Once the training of the final layers started to plateau, early stopping was employed to halt this training. Subsequently, the remaining layers of the network were unfrozen, and a second round of training started, resulting in further enhancements to the results.

Upon closer examination of the results, particularly the mispredictions (Figure 22), it seemed that these very good results could likely have been further improved. Concerning false negatives (images from the ‘GTDs’ class predicted as 'Healthy'), it was observed that images containing the 'Half head' symptom were not correctly predicted. This could have been due to the difficulty in distinguishing between the foreground and background of the image, leading the background to be perceived as an extension of the foreground grapevine. When analysing the false positives (images from the 'Healthy' class predicted as ‘GTDs’), it appeared that the algorithm classified these images as belonging to the ‘GTDs’ class when some shoots were smaller than others, which could occur at the ends of healthy grapevines. The algorithm seemed to have learned to discriminate between healthy and diseased vines based on variations in vegetation within the same grapevine.

*Figure 22. Images of the first test set of the cross-validation for the ClaGTD dataset. First row: Images of the class ‘*GTDs*’ predicted as 'Healthy'. Second row: Images of the class 'Healthy' predicted as ‘*GTDs*’.*

The study revealed that the depth of the architecture and the number of parameters did not directly correlate with improved results. The ResNet50 architecture, which was not the deepest but had a relatively large number of parameters, achieved the best performance. Conversely, the MobileNet architecture, with significantly fewer parameters, demonstrated nearly equivalent results. These findings suggested that, in scenarios where computational time needs to be minimized, selecting the ResNet50 architecture with a downsampling factor of 256 for image resolution or the MobileNet architecture with a downsampling factor of 64 could be beneficial. Although slightly lower precision and recall were observed (0.9, 0.93) and (0.89, 0.92) respectively, these results remained high while greatly reducing inference time, as presented in Table 14 with the comparison of the prediction time of each architecture and image pre-processing.

Table 14. Prediction time (in seconds) for a batch of 30 images, depending on the image pre-processing and the architecture of the CNNs. Average time over 10 repetitions calculated on an NVIDIA GeForce RTX 3090 graphics card.

Pre-process	MobileNetV3-large	ResNet50	EfficientNetB5
/16	0.12	0.18	0.39
/64	0.06	0.08	0.14
/256	0.04	0.05	0.07

b. Flavescence dorée

The first aspect studied in the automated diagnosis of FD was the training and testing on single grape variety. Cabernet sauvignon (CS20) and Ugni blanc (UB20) were used for red and white grape varieties respectively. Furthermore, the training and testing on CS20 were conducted in two different ways: without and with the inclusion of the 'CONF+' images (Table 15). This was done to assess the impact of highly confounding diseases related to FD on the results and to determine if the algorithm was capable of distinguishing them. This particular dataset in Table 15 had the most 'CONF+' images (22%).

Table 15. Results of the cross-validation for CS20 with/without the 'CONF+' class for the FD diagnosis using CNNS.

Dataset	Pre-processing	MobileNetV3-large (p,r) 'FD' class	ResNet50 (p,r) 'FD' class	EfficientNetB5 (p,r) 'FD' class
CS20 without 'CONF+'	/16	(0.91, 0.95)	(0.84, 0.82)	(0.88, 0.95)
	/64	**(0.96, 1)**	**(0.84, 0.93)**	**(0.93, 0.91)**
	/256	(0.96, 0.95)	(0.75, 0.76)	(0.89, 0.91)
CS20	/16	(0.8, 0.88)	(0.76, 0.8)	(0.82, 0.83)
	/64	**(0.84, 0.86)**	(0.76, 0.69)	**(0.83, 0.8)**
	/256	(0.77, 0.6)	(0.74, 0.4)	(0.78, 0.71)

The results presented in Table 15 without 'CONF+' images were excellent. For all three CNNs, the best results were achieved with a downsampling factor of 64 for input image resolution. MobileNetV3-large, ResNet50, and EfficientNetB5 attained precision-recall pairs of (p=0.96, r=1), (p=0.84, r=0.96), and (p=0.93, r=0.91), respectively. Notably, the least deep architecture yielded the highest performance. The results were also excellent for the same architecture with a downsampling factor of 256 for image resolution (p=0.96, r=0.95).

However, when including the 'CONF+' class images, which depict symptoms similar to those of FD, the results declined. The confusion matrix of the first test fold in the training on the 'CS20' dataset is presented in Table 16 and shows that the 'CONF+' images are responsible of this decrease in results.

Table 16. Confusion matrix of the first test fold of the 5-fold cross-validation for the CS20 dataset using MobileNetV3-large with the resolution degradation by 64. It shows the classifier's performance by comparing its predicted results with the actual true values. The numbers represent the count of correctly or incorrectly classified images.

Predicted label / True label	FD	Esca	CONF	CONF+
FD	12	0	1	1
Esca	0	2	7	1
CONF	0	2	12	4
CONF+	3	0	1	7

MobileNetV3-large and EfficientNetB5 architectures achieved the best results of (p=0.84, r=0.86) with a downsampling factor of 64 and (p=0.82, r=0.83) with a downsampling factor of 16,

respectively. By visualizing the results (Figure 23), one can note that the classifier does not rely on the 3 symptoms to make its diagnosis. Indeed, the first and third images classified as 'CONF+' (positive for the disease) by the CNN do not have symptomatic shoots and bunches, only reddening leaves. Additionally, using a downsampling factor of 256 for the MobileNetV3-large architecture led to poor results of (p=0.77, r=0.6). In addition to significantly degrading the CNNs' classification performance, the 'CONF+' images required higher input image resolution to be effectively processed by the CNNs. The architecture that performed the best in the study of GTDs, ResNet50, delivered the lowest results in this case. This demonstrates the crucial importance of selecting an appropriate architecture tailored to the targeted disease and the dataset.

Figure 23. Images of the 'CONF+' class of the first test set of the CS20 dataset. First row: images correctly predicted as 'CONF+' images. Second row: images of the 'CONF+' class predicted as 'FD' class. Results obtained for MobileNetV3-large with the resolution degradation by 64.

In a second step, these trained models were evaluated on images from other datasets to test the robustness of these algorithms when faced with a change in grape variety, where they had never been trained on these specific varieties. Results are presented separately for red and white grape varieties in Table 17 and 18 respectively.

Table 17. Results of the training of the CNNs on CS20 and test on the other red grape varieties for the diagnosis of FD.

Training set	Pre-processing, testing set	MobileNetV3-large	ResNet50	EfficientNetB5
		(p,r) 'FD' class	(p,r) 'FD' class	(p,r) 'FD' class
CS20 without 'CONF+'	/64, CS21 without 'CONF+'	(1, 0.15)	(0.96, 0.21)	(0.96, 0.24)
	/64, CF21 without 'CONF+'	(1, 0.67)	(1, 0.55)	(1, 0.71)
	/64, M21 without 'CONF+'	(0.83, 0.1)	(0.75, 0.16)	(0.96, 0.21)

Table 18. Results of the training of the CNNs on UB20 and test on UB20 and the other white grape varieties for the diagnosis of FD.

Training set	Pre-processing, testing set	MobileNetV3-large	ResNet50	EfficientNetB5
		(p,r) 'FD' class	(p,r) 'FD' class	(p,r) 'FD' class
UB20	/64, UB20	**(0.91, 1)**	(0.84, 0.96)	(0.84, 1)
	/64, UB21	(0.94, 0.79)	(0.78, 0.89)	(0.63, 0.78)
	/64, CS21	(0.94, 0.35)	(0.81, 0.69)	(0.68, 0.85)

The study on the impact of grape variety change revealed that CNNs struggle to generalize their training, and even slight variations in symptom expression between the training and testing datasets significantly affect the quality of the results. When it came to red grape varieties, training on the 'CS20' dataset and testing on images of other red grape varieties highlighted this difficulty. Even without including the 'CONF+' images, the results for all three approaches on the 'CS21' dataset showed high precision (1, 0.96 and 0.96) but very low recall (0.15, 0.21 and 0.24). This means that the algorithms rarely misclassified an image as affected by FD, but they failed to identify a significant number of vines actually affected by FD (resulting in many false negatives). Similar patterns were observed in other datasets. However, the EfficientNetB5 approach appeared to perform slightly better each time, although the results were still not excellent ((p=0.96, r=0.24), (p=1, r=0.71) and (p=0.96, r=0.21) for CS21, CF21 and M21 respectively).

In contrast, the results for white grape varieties were significantly better. Training and testing on the 'UB20' dataset yielded similar good results (p=0.91 and r=1 for MobileNetV3-large) to training and testing on 'CS20' without the 'CONF+' images (since 'UB20' contains only 4% 'CONF+' images compared to 22% in 'CS20'). Once again, MobileNetV3-large achieved the best results. However, compared to red grape varieties, testing this training on other white grape datasets produced relatively better results. However, the recalls (0.79 and 0.35) remained lower compared to the precisions (0.94 and 0.94). This could be attributed to the similarity in symptom expression between white grape varieties such as Ugni blanc and Sauvignon blanc, where leaf yellowing was a common symptom, unlike the distinct red leaf discolouration seen, for example, between Cabernet sauvignon and Merlot. This similarity explained the improved results in white grape datasets.

Trainings and tests were then conducted on all grape varieties of the same colour to test whether CNNs were capable of generalizing when they have multiple different expressions of symptoms during training. The 'CONF+' images of the different grape varieties were also included. The results of these tests are presented in Table 19

Table 19. Results of trainings and tests of CNNs for each grapevine colour for the diagnosis of FD.

Dataset	Pre-processing	MobileNetV3-large	ResNet50	EfficientNetB5
		(p,r) 'FD' class	(p,r) 'FD' class	(p,r) 'FD' class
CS20 + CS21 + CF21 + M21	/16	(0.85, 0.85)	(0.85, 0.84)	(0.84, 0.86)
	/64	(0.8, 85)	(0.84, 0.81)	**(0.87, 0.84)**
	/256	(0.82, 0.81)	(0.86, 0.79)	(0.82, 0.88)
UB20 + UB21 + SB21	/16	(0.92, 0.95)	(0.89, 0.94)	(0.92, 0.93)
	/64	**(0.97, 0.96)**	(0.92, 0.93)	(0.9, 0.93)
	/256	(0.86, 0.87)	(0.9, 0.84)	(0.84, 0.89)

Two separate training processes were carried out: one involving all red grape varieties and the other involving all white grape varieties. The 'CONF+' images were included in the training, although there was a notable difference in the proportion of these images between the two trainings. Specifically, the 'CONF+' class accounted for 16% of the training and testing images for red grape varieties, whereas it comprised only 3% for white grape varieties.

The results clearly reflected this discrepancy, with significantly better outcomes observed for the white grape varieties. In addition to having symptoms that are easier to generalize, the lower presence of 'CONF+' images contributed to excellent results, akin to the performance achieved in trainings and predictions focused on a single grape variety without 'CONF+' images. Once again, the MobileNetV3-large architecture demonstrated the highest performance, achieving a precision-recall pair of (p=0.97, r=0.96). The other two architectures also yielded excellent results. However, for the red grape varieties, the results were slightly worse while still remaining satisfactory for training and predicting across multiple grape varieties. The architectures demonstrated similar performance, with precision-recall pairs of (p=0.85, r=0.85), (p=0.85, r=0.84), and (p=0.87, r=0.84) for MobileNetV3-large, ResNet50, and EfficientNetB5 architectures, respectively. The impact of 'CONF' and 'CONF+' images on these results can be visualized through the confusion matrix presented in Table 20, which refers to the results on the first test fold of the cross-validation of EfficientNetB5 with a resolution degradation on the input images by 64.

Table 20. Confusion matrix of the first test fold of the 5-fold cross-validation for the red grape varieties dataset using EfficientNetB5 with the resolution degradation by 64 for the diagnosis of FD.

Predicted label / True label	FD	Esca	CONF	CONF+
FD	65	1	3	5
Esca	5	10	7	2
CONF	8	5	28	2
CONF+	4	1	6	5

The results from Table 20 indicate that among the 16 images in the 'CONF+' class, 4 of them were classified as 'FD' by the CNN. Furthermore, 5 images from the 'FD' class were classified as belonging to the 'CONF+' class by the model. It is also noteworthy that 8 images from the 'CONF' class were classified as belonging to the 'FD' class by the model. Even more surprising, 5 images from the 'Esca' class were classified as belonging to the 'FD' class. The addition of different symptom expressions depending on the grape varieties and acquisition year has significantly disrupted the model for red grape varieties. This is not the case for white grape varieties, where the 'CONF+' images are in a much smaller proportion and the expression of symptoms is less variable.

Finally, a training was conducted that included all available images for classification (dataset 'ClaFD'). Both red and white grape varieties were mixed, and the 'CONF+' class images were included. The results are presented in Table 21.

Table 21. Results of the training and test of the CNNs on all the images of the 'ClaFD' dataset for the diagnosis of FD.

Dataset	Pre-processing	MobileNetV3-large	ResNet50	EfficientNetB5
		(p,r) 'FD' class	(p,r) 'FD' class	(p,r) 'FD' class
ClaFD	/16	**(0.85, 0.83)**	(0.76, 0.79)	(0.84, 0.77)
	/64	(0.84, 78)	(0.8, 0.82)	(0.8, 0.82)
	/256	(0.79, 0.76)	(0.75, 0.74)	(0.74, 0.71)

The objective was to determine whether CNNs could generalize and comprehend that diverse symptom expressions could lead to the same ultimate diagnosis. While the results presented in Table 21 are reasonably good considering the complexity of the task, they remain insufficient to provide a reliable FD automated diagnosis model. Once again, MobileNetV3-large emerged as the top-performing architecture, achieving a precision-recall pair of (p=0.85, r=0.83) when the input image resolution was downscaled by a factor of 16.

D. Synthesis

In conclusion, the study successfully used CNNs for classifying grapevine images affected by GTDs. Two datasets were utilized, and ResNet50 emerged as the top-performing architecture, achieving the best results when the input image resolution was downscaled by a factor of 16 ((p=0.88, r=0.83) for the 'ClaGTD_till25' dataset and (p=0.94, r=0.92) for the ClaGTD dataset). The best results were obtained for the dataset containing the most images, despite the presence of more variability in the grapevine's development stage and symptoms. These excellent results can be explained by the fact that when a grapevine is affected by GTDs, there is a marked difference between the symptomatic and non-symptomatic parts of the grapevine (either the absence of vegetation on one side of the trunk due to the presence of a 'half head', or a significant difference in the size and foliage of the shoots between the 'short shoot' and non-symptomatic ones).

Regarding the automated diagnosis FD using CNNs, excellent results were obtained when training and testing on the same grape variety without the 'CONF+' images ((p=0.96, r=1) for the MobileNetV3-large-large with a preprocessing of /64 on the 'CS20 without 'CONF+' dataset). However, these results declined when 'CONF+' images were incorporated, indicating the difficulty of distinguishing them from 'FD' images for these algorithms ((p=0.84, r=0.86) for the same methodology on the 'CS20' dataset). The study highlights the importance of including test grape varieties in the training datasets. Finally, the attempt to create a single model encompassing both red and white grape varieties proved difficult. It demonstrated the advantage of separating them into two distinct training processes.

Based on the study of automated diagnoses of GTDs and FD using CNNs, the results with a single grape variety were excellent when none, or very few, confounding diseases were present. This was the case of the GTDs dataset and the 'CS20 without CONF+' dataset. However, for the FD study, as the proportion of grapevine images presenting symptoms similar to FD increased, the results declined. This can be a realistic scenario, as numerous vines can be affected by diseases that resemble FD. On the other hand, the robustness of CNNs has been demonstrated when dealing with changes in grape varieties, as long as those varieties are included in the training dataset. For both diseases, the pre-processing of the images with a division of the resolution by 16 or 64 gave the best results. The division by a factor 256 resulted in the loss of too much information.

In the upcoming chapters, the focus will be on further improving the results, particularly by enhancing the differentiation between the 'FD' and 'CONF+' classes for the FD automated diagnosis as well as a better understanding of the symptoms for the automated diagnosis of GTDs. To accomplish this, the implementation of a dedicated initial phase focused on symptom detection was performed. The hypothesis was that by incorporating this additional step, the accuracy and reliability of the classification process could be enhanced.

V. Unitary symptom detection

Table of content

In the previous chapter, it was observed that directly feeding the image to the model, without additional information about the various symptoms of the disease, yielded very good results on 'simple' datasets containing few diseases with confusing symptoms. However, the results were less favourable as soon as the studied datasets contained more diseases. It was then proposed, in order to achieve a better distinction between the targeted diseases and diseases presenting confusing symptoms, to mimic the method of field prospectors in distinguishing diseases from each other: First, ensure the presence of one or more symptoms of the disease, and then focus on the relationship between these symptoms and their arrangements to provide the most reliable diagnosis possible. In this chapter, the first phase of this diagnostic method will be study, namely the detection of individual symptoms of the targeted diseases.

For the two types of multi-symptom diseases under investigation, two distinct approaches, namely detection and segmentation, were employed to identify the diverse symptoms associated with these conditions. This chapter will exclusively address the symptoms, refraining from drawing

diagnostic conclusions on a broader scale. Within this chapter, a comprehensive breakdown of the various algorithms that were tested, accompanied by their respective outcomes will be provided.

A. Detection networks for bounding box identification of symptoms

a. State-of-the-art detection neural networks

A detection network is a specialized neural network architecture extensively used in computer vision tasks for locating and identifying objects within images. Each detected object in the image comes with a bounding box encompassing the object in question and a prediction of the object's class. These networks can be separated in primary categories: two-stage and single-stage detectors. Two-stage detectors typically involve a proposal generation step followed by object classification and localization, while single-stage detectors directly predict object positions and categories in one pass. In this family of detection networks, some of the most popular ones can be mentioned:

- **Faster R-CNN**: Faster R-CNN (Ren *et al.*, 2015) is a two-stage object detection algorithm that combines region proposal generation and object classification into a single model. It is known for its accuracy but can be relatively slow.
- **YOLO (You Only Look Once)**: YOLO (Redmon *et al.*, 2016) is a single-stage object detection algorithm that can achieve real-time detection on standard hardware. It is known for its speed and is often used in real-time applications like video analysis and robotics.
- **SSD (Single Shot MultiBox Detector)**: SSD (W. Liu *et al.*, 2016) is another single-stage object detection algorithm that aims to balance speed and accuracy. It can detect objects at multiple scales and aspect ratios within a single pass.
- **RetinaNet**: RetinaNet (Lin *et al.*, 2018) is a one-stage detector known for its accuracy and the ability to handle a wide range of object scales. It introduces the focal loss to address class imbalance in object detection.
- **Mask R-CNN**: While primarily designed for instance segmentation (detecting objects and their precise outlines), Mask R-CNN (He *et al.*, 2017) is also capable of object detection. It provides pixel-level object masks in addition to bounding boxes and class labels.

Studies have compared the methodologies and results of these different types of detection algorithms (Kaur & Singh, 2023; Srivastava *et al.*, 2021; Zhao *et al.*, 2019).

Among the various deep detection algorithms, the preference has been for YOLO-style models. It is a fully convolutional network, meaning that it exclusively uses convolutional layers. The selection of YOLO for detection was motivated by several compelling reasons. YOLO offers distinct advantages that make it a suitable choice for the detection task within this study:

- Real-time Performance: YOLO's architecture is designed to process images in a single pass, enabling real-time object detection which is especially beneficial for scenarios where timely detection is crucial.
- Speed and Efficiency: YOLO's unified architecture allows it to simultaneously predict multiple object classes and their bounding boxes. This design contributes to its speed and efficiency, as it reduces redundant computations and streamlines the detection process.
- Multi-Scale Detection: YOLO is capable of detecting objects of varying sizes within the same image. This is particularly valuable when dealing with multi-symptom diseases, where symptoms can manifest itself at different scales.
- Minimal Post-Processing: YOLO's direct bounding box predictions mean that it requires minimal post-processing steps. This simplifies the pipeline and reduces the risk of errors being introduced during post-processing.

Given these reasons, the YOLO architecture aligned well with the objectives of efficiently and accurately detecting the diverse symptoms associated with the multi-symptom diseases studied and is suited for potential future real-time image processing.

YOLO takes a different approach by framing object detection as a regression problem. Here follows a quick explanation of how YOLO is working:

- Grid-based Approach: YOLO divides the input image into a grid of cells. Each cell is responsible for predicting the presence of objects within its boundaries. The size of the grid depends on the chosen architecture and configuration.
- Bounding Box Prediction: Within each cell, YOLO predicts bounding boxes that define the location of potential objects.
- Objectness Score: For each bounding box, YOLO calculates an "objectness" score that estimates the probability of an object's presence within the box. This score helps filter out false positives.
- Class Prediction: In addition to the bounding box and objectness score, YOLO predicts the probability distribution over predefined classes for each bounding box. YOLO can detect and classify multiple object classes within a single cell.
- Non-maximum Suppression: After predictions are made, a post-processing step called non-maximum suppression is applied. This step eliminates redundant bounding box detections and retains only the most confident and accurate ones.

In order to determine the accuracy and reliability of the detection process during the learning and validation steps, the IOU (Intersection over Union) threshold and confidence threshold play crucial roles. An IOU is a measurement of how much two bounding boxes overlap. It is calculated as follows (equation 1):

$$IOU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \quad (1)$$

The IOU threshold is used to determine whether a predicted bounding box accurately captures an actual object in the image. If the IOU between a predicted bounding box and the ground truth bounding box exceeds the set IOU threshold, the detection is considered accurate. If the IOU falls below the set threshold, the detection is treated as a false positive.

The confidence threshold is associated with the predicted probability of an object's presence within a bounding box. In YOLO, each detected bounding box is associated with a confidence score that represents the model's confidence in its prediction. This confidence score is a value between 0 and 1. The set confidence threshold is used to filter out low-confidence detections. Detections with confidence scores below the threshold are discarded as they are considered less reliable.
Figure 24 represents the basic operation of the algorithm.

During the experiments, the IOU and confidence thresholds were set respectively to 0.5 and 0.25.

During the initial trials (Tardif *et al.*, 2022), the latest released version of YOLO was YOLOv4. YOLOv4 incorporates a more powerful backbone architecture than YOLO, using CSPDarknet53 as the base network. This helps to extract more meaningful and representative features from input images. YOLOv4 incorporates a Feature Pyramid Network (FPN), which enables the model to better detect objects at different scales. This is crucial for detecting objects of varying sizes within an image. It introduces various improvements with respect to previous version in object detection, including better handling of small objects and dense object clusters, leading to increased accuracy in detection results. In the head of the network, the non-maximum suppression (NMS) step improve the selection of the most relevant bounding boxes while eliminating redundant detections. Finally, YOLOv4 incorporates self-attention mechanisms, enabling the model to focus on relevant features and ignore irrelevant

ones. In order to minimize inference times, the most compact architecture, YOLOv4-tiny, was chosen. YOLOv4-tiny is optimized for faster inference while sacrificing some accuracy.

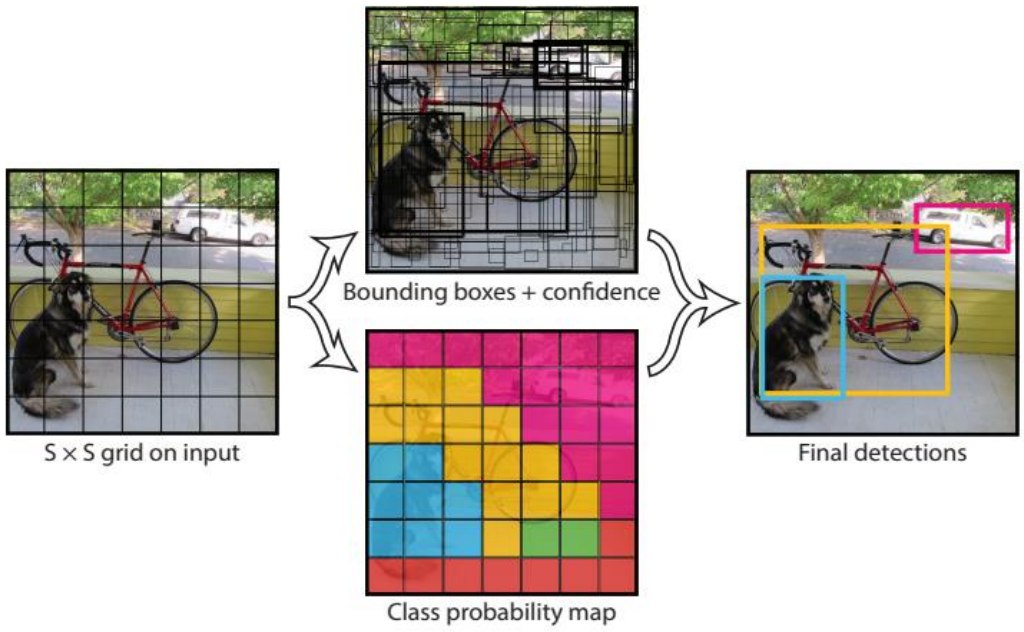

Figure 24. Representation of the YOLO model (Redmon et al., 2016). It divides the image into an S × S grid and for each cell predicts bounding boxes, their confidence score and the class probabilities.

Subsequently, the performance of YOLOv4-tiny, chosen for its inference time, was compared with another one, YOLOv8, less fast but presumed to be more accurate. YOLOv8 is faster (without taking into account the "tiny" versions) and more accurate than the previous versions. It incorporates a new backbone network, a new anchor-free head and a new loss function. The regression branch employs both Distribution Focal Loss (DFL) (Li *et al.*, 2022) and Complete Intersection over Union (CIoU) loss (Zheng *et al.*, 2019).

YOLOv4-tiny and YOLOv8 architectures are presented in Appendix 4.

b. Data pre-processing

Each dataset was partitioned into three sets with consistent proportions: 70% of images for training, 15% for validation, and 15% for testing. Furthermore, the class distribution was maintained identically across these three sets. All subsequent results presented will pertain to outcomes on the test sets within each dataset which remain consistent regardless of the specific training set. Due to the considerable training time required by these algorithms, cross-validation was applied exclusively to the methodology that yielded the most promising results, aiming to validate their performance.

To best match the input resolution required by both algorithms, the images were divided into corresponding-sized patches (416*416*3 for YOLOv4-tiny, 640*640*3 for YOLOv8m). The choice was made not to degrade the image resolution, with the assumption that degrading the resolution would have led to a decrease in detection results. For instance, in the case of YOLOv4-tiny, an image (2048*2448*3) was partitioned into 5*6=30 patches 416*416*3. (by adding lines and columns of zeros to the original image) This approach accomplished two objectives: maintaining the desired input resolution while retaining the full resolution of the images (crucial for accurate symptom detection) and augmenting the training dataset's quantity. This partitioning was conducted after the distribution

into training/validation/test sets, ensuring that patches from the same original image did not end up in different datasets

During the patch partitioning, some bounding boxes might be split into two parts. These bounding boxes were adjusted to ensure completeness across every patch in which they appeared. If the resulting corresponding bounding box measured less than 30 pixels, it was removed, as the enclosed symptom portion became less significant for detection. Additionally, for training and test images, an overlap among the patches was introduced. Sometimes bounding boxes were cut in half during patching. To prevent them from not being fully visible to the algorithm, an extra patch was created between each adjacent patch (as illustrated in Figure 25). Consequently, every bounding box appeared at least once in its entirety during training. For prediction and evaluation of test images, only predictions within the central portion of each patch were retained. For instance, in the case of 640*640*3 patches, predictions made within the 160-pixel border regions— which did not correspond to the true borders of the complete image— are not factored in during the reassembly of the entire images. This approach guaranteed that every symptom was seen in its entirety during its prediction, thereby enhancing diagnostic accuracy.

Figure 25. Thumbnail creation with/without overlapping. The thumbnails bordered in green are those created with overlapping.

The YOLOv8m algorithm was also tested to provide the whole image in order to compare the results to pre-processing results with thumbnails. The images were decimated by a factor of two to avoid memory error and having an input image size that is excessively large compared to the one most suitable for this network (640*640*3). The algorithm automatically resizes the image to match its architecture. The image's largest dimension (in that case, 2448) is reduced to 1216 (must by divisible by 32), degrading the resolution. The second dimension is scaled down while maintaining the original image's proportions (2048 becomes 1088). The advantages were that no symptoms were cut off and the algorithm processed the image in its entirety. The disadvantages were a loss of image resolution and a lower number of images available for training.

For the YOLOv4-tiny, a data-augmentation applied randomly one or more of the following operations on the images in the training set before the training: horizontal flip, image resize, rotation, crop, horizontal and/or vertical translation, luminosity and brightness variation. The YOLOv8m, contains an integrated data-augmentation, which includes a variety of transformations such as random crops, flipping, rotation, distortion, luminosity and brightness variation, mosaic augmentation

technique (combining four images into one, which helped prevent overfitting and augment the training data).

c. Detection datasets and experimental design

i. Flavescence dorée

The detection algorithms were tested to automatically detect symptomatic leaves of FD. Three classes of leaves were annotated in the images: 'FD Leaf', 'Esca Leaf', and 'Confounding Leaves'. Initially, tests were conducted on the ‘CS20’ subset of ‘DetFD’ (described in Chapter III Section F, Table 8) dataset to determine the optimal way to annotate the images. Thus, two annotation methods for leaves were tested: either a bounding box per leaf, attempting to closely fit the leaf (A1), or larger bounding boxes grouping clusters of leaves of the same class together (A2). This latter annotation approach could potentially save a significant amount of annotation time if the results were promising. Secondly, tests were conducted with and without leaves from the 'Confounding Leaves' class (A3). Once again, there would be a substantial time-saving in annotation if not annotating the confounding leaves didn't negatively impact the algorithm's performance. The 3 annotating ways are presented in Figure 26.

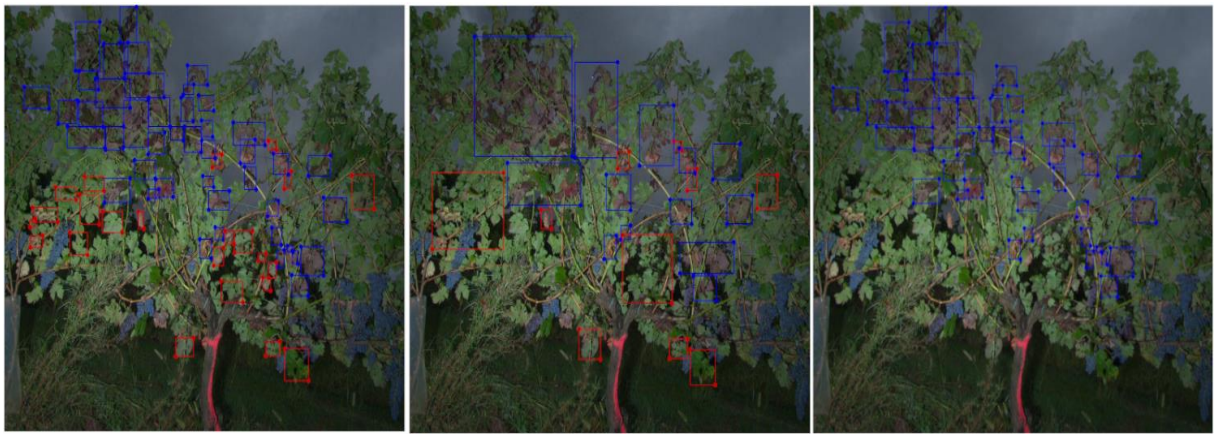

Figure 26. Three ways to annotate the leaves for FD leaf detection. Left image: One box per leaf, confounding leaves annotated (A1). Middle image: Annotation by cluster of leaves, confounding leaves annotated (A2). Right image: One box per leaf, no annotation of confounding leaves (A3).

Tests were subsequently conducted to assess the algorithm's ability to detect FD leaves on grape varieties not included in the training dataset using the annotation method that yielded the best results in the previous step. By separating the images of the white and red grape varieties, the tests began with only one grape variety used for training and testing on the others. Then, the training dataset was gradually supplemented with images of the other grape varieties of the same colour to monitor the progression of the results. The objective was to determine if adding new grape varieties to the training dataset improved the test results on these same grape varieties without decreasing the results for the grape varieties already present in the training set. A summary of the tests conducted for the detection of FD symptomatic leaves is presented in Table 22.

Table 22. Summary of the tests conducted for the algorithms detecting symptomatic FD leaves. The colours of the rows indicate the colour of the studied grape varieties: red for red grape varieties, yellow for white grape varieties. Annotation method A1 refers to leaf-by-leaf annotation, including confounding leaves. Method A2 refers to annotation by leaf clusters, including confounding leaves. Method A3 refers to leaf-by-leaf annotation, excluding confounding leaves.

Training set	Testing set	algorithm	Annotating method	Input Image size	Purpose
CS20	CS20	YOLOv4-tiny	A1	416*416*3	Choice of the annotation method
CS20	CS20	YOLOv4-tiny	A2	416*416*3	
CS20	CS20	YOLOv4-tiny	A3	416*416*3	
CS20	CS20 + CF21 + M21 + CS21	YOLOv4-tiny	A1	416*416*3	Testing on out-of-training grape varieties + tracking results evolution with dataset expansion
CS20 + CF21	CS20 + CF21 + M21 + CS21	YOLOv4-tiny	A1	416*416*3	
CS20 + CF21 + M21 + CS21	CS20 + CF21 + M21 + CS21	YOLOv4-tiny	A1	416*416*3	
UB20	UB20 + UB21 + SB21	YOLOv4-tiny	A1	416*416*3	
UB20 + UB21	UB20 + UB21 + SB21	YOLOv4-tiny	A1	416*416*3	
UB20 + UB21 + SB21	UB20 + UB21 + SB21	YOLOv4-tiny	A1	416*416*3	
CS20 + CF21 + M21 + CS21	CS20 + CF21 + M21 + CS21	YOLOv8m	A1	640*640*3	YOLO versions comparison + image pre-processing selection
CS20 + CF21 + M21 + CS21	CS20 + CF21 + M21 + CS21	YOLOv8m	A1	2048*2448*3	
UB20 + UB21 + SB21	UB20 + UB21 + SB21	YOLOv8m	A1	640*640*3	
UB20 + UB21 + SB21	UB20 + UB21 + SB21	YOLOv8m	A1	2048*2448*3	

ii. Grapevine trunk diseases

Regarding the GTDs, the 3 symptoms ('Half Head', 'Short shoot', and 'Canker') found in Sauvignon blanc grapevines were annotated with bounding boxes and tested for detection. Unlike the images used for FD diagnosis, acquired at the same period of the year (a few days before harvest), the GTDs images were acquired much earlier in the season and over a longer period during grapevine growth. Thus, the symptoms vary in size depending on the acquisition date. To assess the impact of the acquisition date, dataset ‘DetGTD’ was divided into 2 subsets: 'DetGTD_till2 ' consisting of images from ‘DetGTD’ acquired between the 20/11/22 and 25/11/22, and 'DetGTD_since28' consisting of images acquired between 28/11/22 and 22/12/22.

The training on 'DetGTD_till25’ was tested separately on the test sets of ‘DetGTD_till25‘ and ‘DetGTD_since28' (same for the training on ‘DetGTD_since28') to evaluate the algorithm's capabilities

in detecting symptoms on vines at different stages of development. These results were then compared with those from the entire 'DetGTD' dataset. The chosen annotation methodology (A1) and the algorithm (Yolov8m) were the ones that yielded the best results during FD leaf detection. The image size in input was either patches of 640*640*3 or the whole rescaled image in 1088*1216*3. A summary of these experiment is presented in Table 23.

Table 23. Summary of the tests conducted for the algorithms detecting GTDs symptoms. Annotation method A1 refers to leaf-by-leaf annotation, including confounding leaves.

Training set	Testing set	algorithm	Annotating method	Image size	Purpose
DetGTD_till25	DetGTD_till25, DetGTD_since28	YOLOv8m	A1	640*640*3, 1088*1216*3	Identifying the phenological stage(s) that allow for the best detection of GTDs symptoms
DetGTD_since28	DetGTD_till25, DetGTD_since28	YOLOv8m	A1	640*640*3, 1088*1216*3	
DetGTD_till25 + DerGTD_since28	DetGTD_till25, DetGTD_since28	YOLOv8m	A1	640*640*3, 1088*1216*3	

d. Results

i. Flavescence dorée

The first aspect studied was the impact of annotating the symptoms in order to avoid wasting time later by changing and redoing annotations. The three annotation methods were tested on the CS20 dataset, and the results are presented in Table 24.

Table 24. Results of the YOLOv4-tiny of the 3 annotating methods on the CS20 dataset for the detection of symptomatic leaves of FD. Annotation method A1 refers to leaf-by-leaf annotation, including confounding leaves. Method A2 refers to annotation by leaf clusters, including confounding leaves. Method A3 refers to leaf-by-leaf annotation, excluding confounding leaves.

Training set	Testing set	Annotating method	Results of the 'FD leaf' class in (precision, recall)
CS20	CS20	A1	**(0.65, 0.45)**
CS20	CS20	A2	(0.53, 0.43)
CS20	CS20	A3	(0.51, 0.48)

It can be observed from Table 24 that the best results (p=0.65, r=0.45), when combining precision and recall, were achieved with annotation method A1, referring to individual small bounding boxes for each symptom, compared to method A2 referring to the annotation of clusters of leaves (p=0.53, r=0.43). Furthermore, including the annotation of confounding leaves enhanced the results in symptomatic leaf detection (A1 vs. A3 (p=0.51, r=0.48)). This annotation methodology, although the most time-consuming, was subsequently adopted and applied to other datasets for the detection of symptomatic FD leaves.

The results of the 'FD leaf' class detection were not really high, due to the complexity of the task and to the small number of training images.

It was then decided to investigate the algorithm's capability to effectively predict symptomatic leaves of FD on grapevine varieties it hadn't been exposed to previously. The training datasets had been gradually expanded with new grape varieties to monitor the progress of the results. These results

are presented separately for the white grape varieties in Table 25 and for the red grape varieties in Table 26.

Table 25. YOLOv4-tiny's results of the detection of FD symptomatic leaves for white grape varieties.

	Results of the 'FD leaf' class in (precision, recall)		
Training set	**UB20**	**UB21**	**SB21**
UB20	(0.64, 0.40)	(0.61, 0.39)	(0.52, 0.08)
UB20 + UB21	(0.52, 0.53)	**(0.51, 0.74)**	(0.67, 0.42)
UB20 + UB21 + SB21	**(0.57, 0.49)**	(0.54, 0.69)	**(0.66, 0.57)**

As for identifying symptomatic leaves of FD in white grape varieties, initial training using the UB20 images yielded an acceptable precision (p=0.64 and 0.61) for UB20 and UB21 datasets respectively. However, the recall (r=0.40 and 0.39 for UB20 and UB21 respectively) was poor for both datasets. Incorporating the UB21 dataset into the training process significantly enhances the recall of 'FD leaf' in the UB20 (r=0.53) and UB21 (r=0.74) sets, leading to a more comprehensive recovery of FD-symptomatic leaves. Moreover, both precision (p=0.67) and recall (r=0.42) exhibit an improvement in SB21 images. This enhancement can be attributed to the greater diversity of symptom expressions and image acquisition conditions offered by the UB21 images, even though the symptoms on the leaves in SB21 images differed. Subsequently, the introduction of SB21 images into the training dataset results in improved performance on SB21 images (p=0.66, r=0.57), without adversely affecting the results on UB20 (p=0.57, r=0.49) and UB21 (p=0.54, r=0.69) sets.

Table 26. YOLOv4-tiny's results of the detection of FD symptomatic leaves for red grape varieties.

	Results of the 'FD leaf' class in (precision, recall)			
Training set	**CS20**	**CF21**	**M21**	**CS21**
CS20	(0.65, 0.45)	(0.70,0.51)	(0.24, 0.13)	(0.54, 0.31)
CS20 + CF21	(0.67, 0.50)	(0.77, 0.63)	(0.54, 0.29)	(0.77, 0.32)
CS20 + CF21 + M21 + CS21	**(0.63, 0.63)**	**(0.71, 0.79)**	**(0.61, 0.58)**	**(0.73, 0.68)**

When addressing red grape varieties, the initial training using the CS20 dataset, which comprises only 51 images, demonstrates good precision results (p=0.65 for CS20, p=0.70 for CF21). Notably, the results on CS21 images (p=0.54, r=0.31) were interesting as they were captured from a greater distance compared to the images in CS20. This highlighted the pivotal role of data augmentation, as the algorithm's ability to train on leaf sizes akin to those in CS21 is made possible through randomized scaling operations. This proved beneficial despite the absence of such examples in the original training dataset. Nonetheless, the results on M21 images were not so good, (p=0.24, r=0.13), likely due to the distinctiveness of symptoms in these images. Incorporating CF21 images into the training dataset enhanced the identification of FD symptomatic leaves across CF21 (p=0.67, r=0.50), M21 (p=0.54, r=0.29), and CS21 (p=0.77, r=0.32) images. This augmentation introduces a wider spectrum of symptoms, leading to a significant improvement in FD detection. The conclusive training phase involving red grape varieties yields highly satisfactory results. The inclusion of M21 and CS21 sets improved the recall results on CS20 (p=0.63, r=0.63) and CF21 (p=0.71, r=0.79) sets, while

substantially enhancing the prediction accuracy for symptomatic FD leaves in M21 (p=0.61, r=0.58) and CS21 (p=0.73, r=0.68) images.

In summary, the investigation into detecting symptomatic leaves of FD across different grapevine varieties underscored the importance of dataset diversity and the crucial role of data augmentation. The results were not excellent, due to the numerous confounding leaves of those of FD symptomatic leaves and the rather limited quantity of annotated data. However, the results demonstrated the algorithm's adaptability, by sometimes even achieving good performances on varieties not present in the training sets. This study highlights the evolving nature of object detection training and the crucial role of holistic dataset enrichment in enhancing detection accuracy across varied conditions and grapevine varieties.

Finally, in order to ascertain whether opting for YOLOv4-tiny resulted in a significant performance loss, the utilization of YOLOv8m was tested on the training sets that yielded the best results with YOLOv4-tiny. This was done with a training on all red grape varieties (as presented in Table 26) and all white grape varieties (as presented in Table 25). To validate the image patch cropping methodology, complete images were fed into the detection network as inputs. Results of these tests are presented in Table 27 for the red grape varieties and in Table 28 for white grape varieties.

Table 27. Results of the YOLOv8m and comparison with the best results of the YOLOv4-tiny for the detection of the FD symptomatic leaves on red grape varieties. Best results for each set are put in bold.

		Results of the 'FD leaf' class in (precision, recall)				
Training set	**Algorithm, pre-processing**	**CS20**	**CF21**	**M21**	**CS21**	**Global**
CS20+CF21+ M21+CS21	YOLOv4-tiny, patches	(0.63, 0.63)	**(0.71, 0.79)**	(0.61, 0.58)	(0.73, 0.68)	(0.66, 0.67)
	YOLOv8m, patches	**(0.70, 0.70)**	(0.76, 0.70)	(0.51, 0.6)	**(0.79, 0.72)**	(0.71, 0.68)
	YOLOv8m, whole images	(0.73, 0.57)	(0.81, 0.65)	**(0.79, 0.48)**	(0.88, 0.59)	(0.78, 0.61)

The first thing that can be noted is the superior overall performance of the YOLOv8m algorithm (global precision and recall) of (p=0.71, r=0.68) and (p=0.78, r=0.61) compared to that of YOLOv4-tiny of (p=0.66, r=0.67). Three out of four grape varieties achieve better results with the former. Only the CF21 set obtained better results with YOLOv4-tiny, but the results of YOLOv8m for this variety are not far behind. For the other three grape varieties, the results are indeed better, though not significantly superior to YOLOv8m.

Regarding the pre-processing of images using patches or full images, once again, for three out of four grape varieties, the patch-based pre-processing yields better results. Sometimes, there is a significant difference (for the CS20 dataset, patch pre-processing achieves (p=0.70, r=0.70), compared to full image pre-processing (p=0.73, r=0.53)). At other times, the difference is similar (for the CF21 dataset, patches yield (p=0.76, r=0.7), while full images yield (p=0.81, r=0.63). Only the M21 dataset obtains improved results with the algorithm trained on full images. The YOLOv8m algorithm with patch-based pre-processing appears to stand out, although it doesn't achieve superior results across all test sets.

Figure 27 demonstrates the significance of overlapping predictions during image cropping and prediction by patch. In the highlighted area, it can be observed that an inaccurate prediction of 'FD

leaf' is avoided at the centre of the white circle, and the confounding leaves in the top left and the FD leaves in the bottom right of the white circle are better delineated.

Raw image | Predictions without overlapping | Predictions with overlapping

Figure 27. Comparison between the predictions by patches with and without overlapping of patches on a red grape variety. Boxes of the class 'FD leaf' in red and of the class 'Confounding leaf' in orange. Inaccurate prediction of 'FD leaf' is avoided at the centre of the white circle, and the confounding leaves in the top left and the FD leaves in the bottom right of the white circle are better delineated.

Table 28. Results of the YOLOv8m and comparison with the best results of the YOLOv4-tiny for the detection of the FD symptomatic leaves on white varieties. Best results for each set are put in bold.

		Results of the 'FD leaf' class in (precision, recall)			
Training set	**Algorithm, pre-processing**	**UB20**	**UB21**	**SB21**	**Global**
UB20+UB21+ SB21	YOLOv4-tiny, patches	(0.57, 0.49)	(0.54, 0.69)	(0.66,0.57)	(0.57,0.57)
	YOLOv8m, patches	(0.42, 0.56)	(0.40, 0.72)	(0.54, 0.78)	(0.42, 0.64)
	YOLOv8m, whole image	**(0.44, 0.64)**	**(0.52, 0.76)**	**(0.62, 0.75)**	**(0.48, 0.69)**

Regarding the results for white grape varieties in Table 28, the difference between the results of YOLOv4-tiny (same as Table 25) and YOLOv8m is even smaller. Precision is consistently better for YOLOv4-tiny (global precision of 0.57), but recall is further improved for YOLOv8m (recall of 0.64 for the patches and 0.69 for the whole images), making it the best algorithm when both indices are considered ((p=0.57, r=0.57) for Yolov4-tiny with patches, (p=0.42, r=0.64) for YOLOv8m with patches, (p=0.48, r=0.69) for YOLOv8m with whole images). However, unlike on the red grape varieties, the precision and recall results are better across all four datasets for the processing of the entire image. This can be explained by the fact that, unlike red grape varieties, the symptoms on white grape varieties are much less pronounced (slight yellowing of the leaves), making them harder to differentiate from confusing symptoms, even in full resolution with patch-based slicing. On the other hand, providing the entire image can help the algorithm calculate the distribution of symptomatic leaves on the grapevine and achieve better results.

Figure 28 illustrates a comparison of predictions from the three modalities (choice of algorithm and image pre-processing). For the first example (first row, symptomatic leaves of FD of M21 set, the 3 modalities gave good results, except a false prediction of 'Confounding leaf' for the YOLOv4-tiny. The second example (second row) displays confounding leaves of an image of CF21. The 3 modalities wrongly predicted some leaves of the 'FD leaf' class. This was a very hard image to predict as the leaves

appeared to be exactly the same in shape and colour as leaves symptomatic of FD in this grape variety. However, the YOLOv8m algorithm on large images still managed to predict 2 of these leaves as belonging to the 'confounding leaf' class. It is the only algorithm that achieved this. Finally, the last example (last row of Figure 28) presents leaves that are very similar to those symptomatic of FD on Sauvignon Blanc grape variety. Once again, these leaves are very challenging to distinguish from FD leaves, sharing the same colour and shape. The YOLOv4-tiny algorithm predicted them as belonging to the 'FD leaf' class; however, both YOLOv8m algorithms predicted them correctly, either as belonging to the 'Confounding leaf' class or by not making any predictions on them (indicating that, for the algorithm, these leaves do not belong to any of the classes 'FD leaf,' 'Esca leaf,' or 'Confounding leaf').

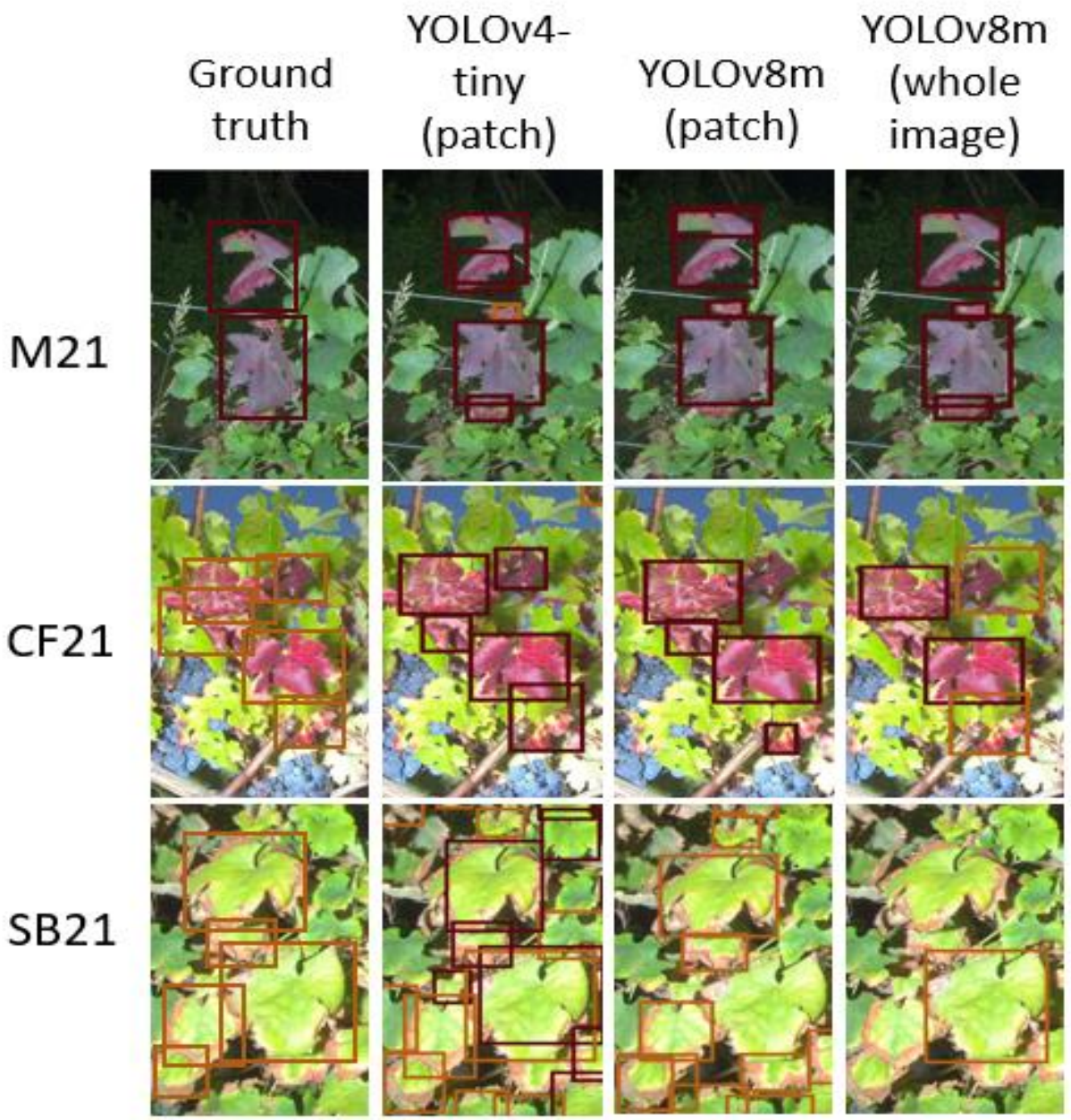

Figure 28. Comparison of the leaf predictions. First column: Ground truth images for Merlot (M21), Cabernet franc (CF21) and Sauvignon blanc (SB21). Second column: Predictions by the YOLOv4-tiny. Third column: predictions by the YOLOv8m on patches. Fourth column: predictions by the YOLOv8m on whole images. Box colours are dark red for the 'FD leaf' class and brown for the 'Confounding leaf' class. First row: FD symptomatic leaves of M21 set. Second row: Confounding leaves of CF21 set. Third row: Confounding leaves of SB21 set.

ii. Grapevine trunk diseases

For the detection of GTDs symptoms, the methodology that achieved the best results in FD leaf detection was adopted. This includes annotating images using the A1 method, employing the YOLOv8m architecture, and cutting images into patches (640*640*3) or whole image (1088*1216*3) as network inputs. Three training sets were compared to analyse the impact of grapevine growth stages on symptom detection. Table 29 compares these approaches for input images cut into patches and Table 30 for entire input images.

Table 29. Comparison of the results of the YOLOv8m on patches from images acquired before and after the 25/11/22 for the GTDs symptoms detection.

	DetGTD_till25			**DetGTD_since28**		
Training set	**'Half head'**	**'Short shoot'**	**'Canker'**	**'Half head'**	**'Short shoot'**	**'Canker'**
DetGTD_till25	(0.36, 0.65)	(0.24, 0.56)	(-, 0)	(0.64, 0.65)	(0.43, 0.34)	(-, 0)
DetGTD_since28	(0.32, 0.45)	(0.23, 0.45)	(-, 0)	(0.56, 0.43)	**(0.47, 0.53)**	(-, 0)
DetGTD_till25 + DerGTD_since28	(0.56, 0.61)	(0.26, 0.43)	(-, 0)	**(0.63, 0.67)**	(0.42, 0.53)	**(-, 0)**

Regarding the results of Table 29, the first thing to note is the best performance on the DetGTD_since28 test set, even when trained on the DetGTD_till25 images (p=0.64, r=0.65) for the 'Half head' class, (p=0.43, r=0.34) for the 'Short shoot' class). This can be explained by the fact that during the analysis of a grapevine in an early stage, symptoms like short shoots can be easily mistaken for healthy shoots that have just emerged or are not yet densely covered in leaves. Regarding the 'half head', once again, the more substantial the vegetation, the more pronounced the absence of growth on one side becomes. As for the cankers, their very low count (139 in total across all annotations) prevents their accurate detection. The score (p=-, r=0) indicates that no canker was detected. The best results are achieved when training on the entire dataset, which includes a wider variety of symptom expressions, and testing on the DetGTD_since28 dataset ((p=0.63, r=0.67) and (p=0.42, r=0.53) for the classes 'Half head' and 'Short shoot' respectively).

Table 30. Comparison of the results of the YOLOv8m on entire images acquired before and after the 25/11/22 for the GTDs symptoms detection.

	DetGTD_till25			**DetGTD_since28**		
Training set	**'Half head'**	**'Short shoot'**	**'Canker'**	**'Half head'**	**'Short shoot'**	**'Canker'**
DetGTD_till25	(0.43, 0.71)	(0.35, 0.65)	(0.12, 0.09)	(0.62, 0.71)	(0.38, 0.24)	(-, 0)
DetGTD_since28	(0.45, 0.58)	(0.1, 0.32)	(-, 0)	(1, 0.25)	(0.45, 0.62)	(-, 0)
DetGTD_till25 + DerGTD_since28	(0.74, 0.59)	(0.47, 0.35)	(-, 0)	**(0.85, 0.81)**	**(0.58, 0.58)**	**(-, 0)**

The training and testing results on the full images followed the trend of patch-based processing, with the best outcomes observed for each training scenario when tested on the 'DetGTD_since28' dataset. The best results are also achieved when training on all the available data and evaluating on the 'DetGTD_since28' test set, with (p=0.85, r=0.81) and (p=0.58, r=0.58) for the classes ‘Half head’ and ‘Short shoot’ respectively. These results are notably superior to those obtained with patch-based processing, except for the cankers which still remain undetected. Supplying the entire image enhances the detection of 'Half head' symptoms, as it allows for a clearer recognition that no growth occurs on one side of the grapevine. Additionally, for 'short shoots', comparing their sizes and shapes to the healthy ones on the same grapevine is easier with the full image, than with segmented patches.

Reducing the resolution should not result in significant loss of performance since it is primarily the shape of the symptoms and their surroundings that are being analysed (such as the lack of foliage on shoots or the absence of vegetation on one side of the trunk). Figure 29 displays results of both successful and unsuccessful symptom detection on images from the ' DetGTD_till25 + DetGTD_since28' test dataset, using the training that yields the best outcomes for this specific dataset (training on whole images from the entire dataset).

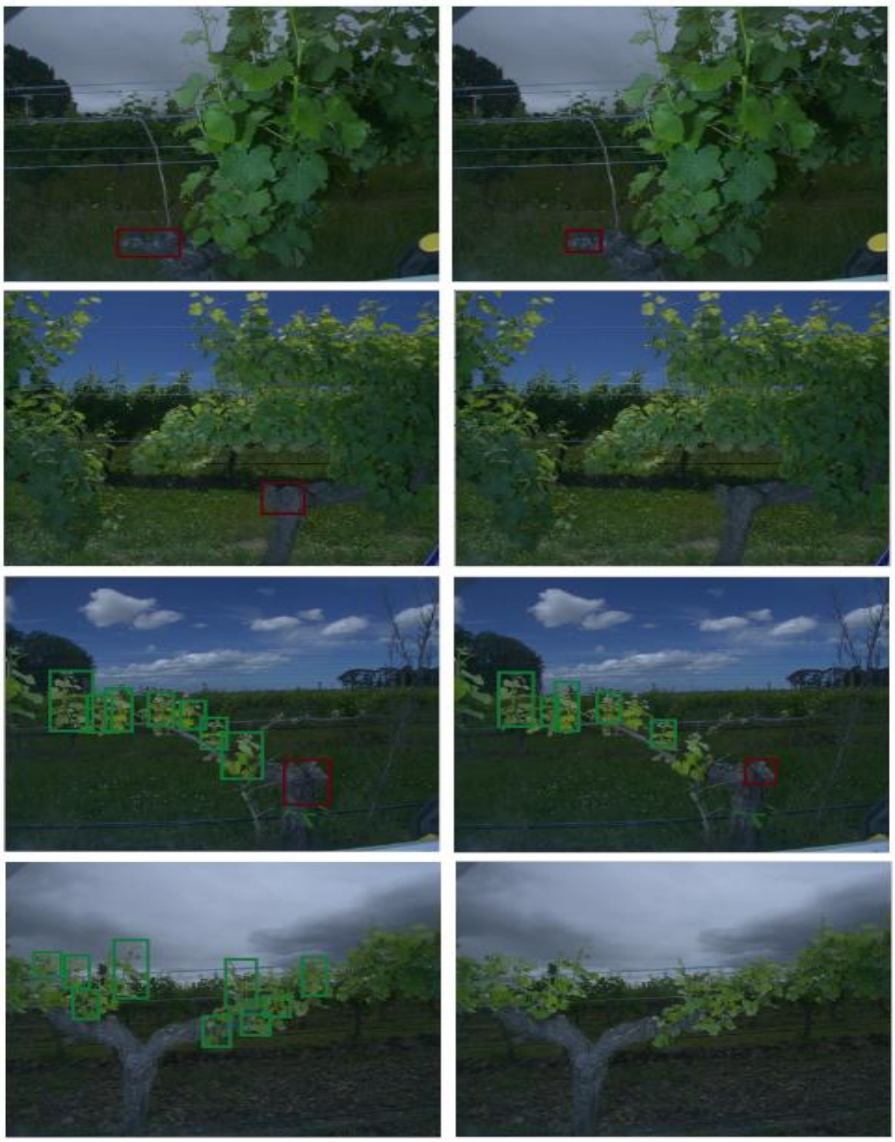

Figure 29. Examples of symptom predictions on images of grapevine affected by GTDs. First column: Ground truth. Second columns: predictions by the YOLOV8m. Boxes in red: 'Half head', in green: 'Short shoot'.

In Figure 29, the first example (first row) depicts a correctly detected 'half head' by the YOLOV8m algorithm. In the second example (second row), a 'half head' is not detected by the algorithm. The hypothesis was that, unlike the first example, the grass touching the 'half head' in the image is almost the same colour as the foliage, and therefore the algorithm doesn't recognize it as a 'half head'. The third example (third row) show cases the accurate detection of a 'half head' (again with grass of a different colour than the foliage), along with correct identifications of 'short shoots,' although a few are missing. Finally, the last example (last row) shows 'short shoots' that are not detected by the algorithm. The photographed grapevine is in an early stage, and as seen before, the algorithm struggles to achieve good results on such images, even though the symptoms seem clear to the human eye. While some 'short shoots' were not detected in the previous example, in this case, none of the 'short shoots' were detected.

B. Segmentation of symptomatic grapevine organs

Another category of algorithms, tailored for predicting symptoms in images, has also been explored. These are segmentation algorithms, potentially better suited for identifying specific symptoms.

a. Segmentation neural networks and related methodology

i. Choice of a relevant architecture

Segmentation algorithms belong to a category of computer vision methods employed to divide an image into various segments or regions, where each segment corresponds to a unique object, area, or characteristic present in the image. The main objective of image segmentation is to break down the complexity of an image into meaningful and controllable components, simplifying the task of computer analysis and comprehension of the visual content. Semantic segmentation involves assigning a label (like 'Trunk' or 'Foliage') to individual pixels in an image, essentially classifying each pixel into a particular object category. This process offers a broad overview of the image's content.

Several methods are available for achieving this task. One common approach is thresholding, where a specific intensity threshold is used to separate regions in an image based on pixel intensity values. Alternatively, clustering can be employed, grouping similar pixels into clusters by considering attributes like colour, intensity, or texture. Region Growing (Shih & Cheng, 2005) is another technique that begins with a seed pixel and progressively expands a region by incorporating adjacent pixels that satisfy particular similarity criteria. In recent years, the domain of image segmentation has been revolutionized by deep learning models, particularly CNNs. Architectures like U-Net (Ronneberger *et al.*, 2015), FCN (Long *et al.*, 2015), and Mask R-CNN (He *et al.*, 2017) have introduced a paradigm shift in image segmentation by harnessing the power of deep learning to understand intricate features and relationships within images, resulting in significantly enhanced accuracy and performance of the pixels classification.

ResUNet (Diakogiannis *et al.*, 2020), a CNN architecture combining the advantages of two other popular architectures, ResNet and U-Net, has been chosen to be tested for symptom segmentation. Here's a breakdown of these two components:

- U-Net: The U-Net architecture is widely used in image segmentation tasks. It consists of a contracting path (encoder) to capture contextual information and a symmetric expanding path (decoder) to generate a precise segmentation mask. The U-Net's architecture enables it to localize and segment objects within images effectively.
- ResNet: Residual Networks, or ResNets, are a type of deep convolutional neural network architecture known for their skip connections, which help mitigate the vanishing gradient problem and enable the training of very deep networks. As already explained, ResNet

introduced the concept of residual blocks, where the network learns the residual (difference) between the input and the desired output, making it easier to optimize deep networks.

The ResUNet architecture combines the U-Net's ability to perform semantic segmentation with the ResNet's skip connections for efficient training of deep networks. By integrating skip connections, the ResUNet architecture allows the network to capture both fine-grained and contextual features, making it well-suited for tasks like image segmentation where accurate boundary localization is crucial. The architecture essentially takes advantage of the U-Net's architecture for segmentation tasks and leverages the ResNet's residual connections to facilitate the training of deeper networks, leading to improved accuracy and convergence during training. Figure 30 illustrates the workflow of both U-Net and ResNet architecture.

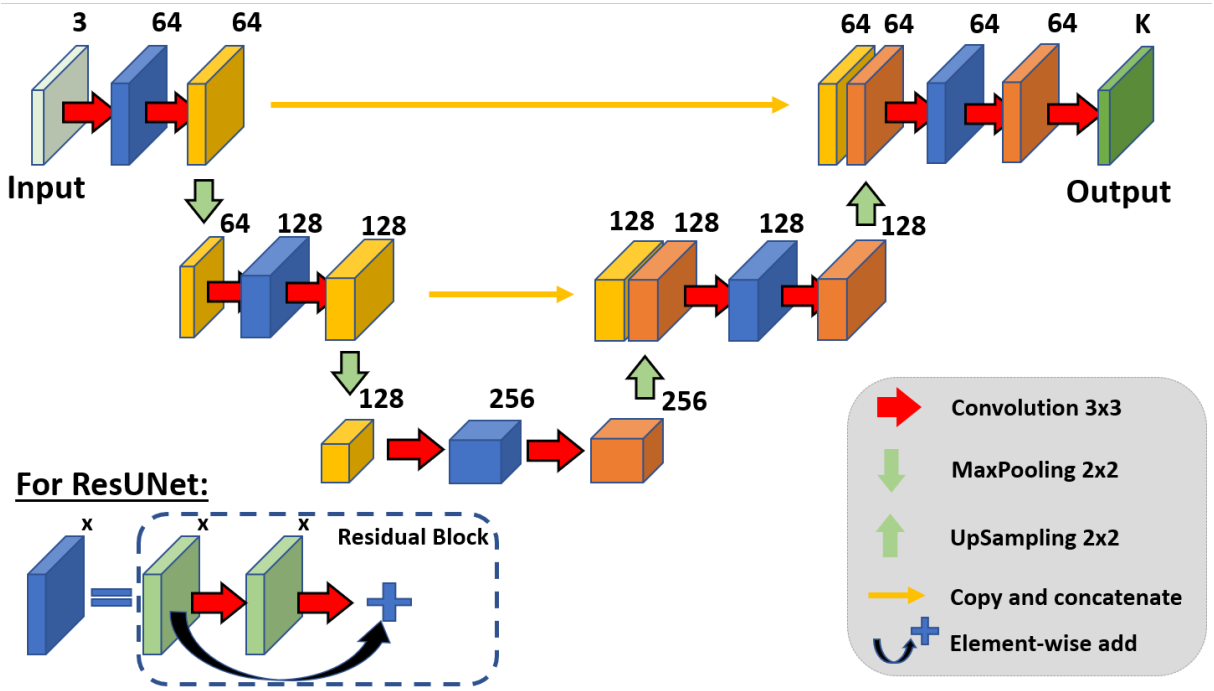

Figure 30. U-Net and ResUNet architecture. The input image has a depth of 3 (3 values per pixel), and the output is transformed into K dimensions (for pixel classification into K classes).

Each training example is associated with a true class label. The true class label for each pixel is represented as a vector, where the element i, corresponding to the true class i is 1, and the rest are 0. The final convolution of the ResUNet produces raw scores or logits for each class. These logits are then transformed into probabilities using the Softmax function. The Softmax function ensures that the predicted probabilities sum up to 1.

A more classical image processing algorithm was also tested to segment a certain type of symptom. This is the structure tensor (Budde & Frank, 2012). The results, although interesting, were outperformed by the neural network approach. For the sake of coherence and space saving, this approach has been removed from the main body of the manuscript. The structure tensor approach is detailed in Appendix 5, and its results are compared to those obtained by ResUnet for the segmentation of the target symptom.

ii. Metrics

In order to study and compare the results of these segmentation algorithms, two metrics were used. Firstly, since the classification is conducted at the pixel level, the 'pixel' metric was employed. It reflects

the classification outcomes at the pixel scale. However, it might not be very informative as numerous classification errors can occur, especially at the edges of objects, even when these are relatively well detected. For small symptoms or elongated symptoms, these edge errors can constitute a significant percentage. Additionally, precisely detecting every pixel of a symptom is not necessarily essential; what matters is accurate localization and classification of the symptom. Hence, a second metric was examined, termed the 'Object' metric. In this metric, if at least 50% of an object's pixels were correctly classified, the object is deemed correctly predicted. This metric enables a comparison of the number of symptoms correctly predicted or not.

iii. Loss functions

A study was also conducted regarding the loss function. Indeed, for the segmentation of certain symptoms, the number of pixels belonging to these classes can be considerably smaller compared to the number of pixels in other classes. This disparity in proportions can pose challenges during model training. In fact, deep learning models, like neural networks, learn by adjusting their parameters based on the examples in the training data. When one class is heavily outnumbered, the model might prioritize learning the majority class and perform poorly on the minority class. This results in an unbalanced model that cannot make accurate predictions for the minority class. They tend to overfit to the majority class, leading to poor performance on the minority class. These models use a loss function to quantify the difference between predicted and actual outcomes. In imbalanced datasets, it is important to choose a loss-function that takes into account these class imbalances. To study the impact of the choice of loss function, four loss functions have been tested:

1. **Categorical cross-entropy**. The categorical cross-entropy loss calculates the dissimilarity between the predicted probabilities and the true class labels. Its formula is given in equation 2:

$$CE = -\sum_{i=1}^{N} y_i \,.\, log(\hat{y}_i) \tag{2}$$

   where N is the number of classes, y_i is the i-th element of the true class label vector and $\hat{y}_i$ is the i-th element of the predicted probability vector.

   The closer the probability of belonging to the correct class is to 1, the closer the logarithm of that probability will be to 0. As a result, the value of the loss function will be close to 0. This is a basic loss function that doesn't take into account class imbalance issues.

2. **Weighted categorical cross-entropy**. The weighted categorical cross-entropy is the same as the previous one except that class-specific weights are designed to balance the contribution of each class to the overall loss. Typically, the weight of a class is inversely proportional to its frequency in the training dataset. That is, classes with fewer samples get higher weights, and classes with more samples get lower weights. The formula is given in equation 3:

$$WCE = -\sum_{i=1}^{N} w_i . y_i \,.\, log(\hat{y}_i) \tag{3}$$

   with w_i the weight associated with the class i.

3. **Dice loss**. The Dice loss measures the similarity between the predicted segmentation mask and the ground truth mask by comparing the overlapping regions of these masks. It is used to

evaluate how well the predicted segmentation matches the true segmentation. The formula (equation 4) for Dice Loss is:

$$\boldsymbol{D}L = 1 - \frac{2\,.\,\sum(Prediction \cap GroundTruth)}{\sum Prediction\ +\ \sum Ground\ Truth} \tag{4}$$

where Prediction represents the predicted segmentation mask, Ground Truth represents the true segmentation mask, ∩ denotes the intersection between two sets and ∑ represents the number of pixels in a set.

By minimizing the Dice Loss during training, the model learns to generate segmentation masks that align closely with the ground truth masks. It focuses more on the overlapping regions and is less affected by the background regions that are usually larger in size.

4. **Twersky loss**. Twersky Loss is a loss function introduced as an alternative to traditional loss functions, like the Dice Loss, for addressing the class imbalance issue in image segmentation tasks. This formula (equation 5) is as follow:

$$\boldsymbol{TL} = \frac{\boldsymbol{TP}}{\boldsymbol{TP}\ +\ \boldsymbol{\alpha}.\boldsymbol{FN} + \boldsymbol{\beta}.\boldsymbol{FP}} \tag{5}$$

where *TP* represents true positives (correctly classified pixels), *FN* represents false negatives (missed pixels), *FP* represents false positives (misclassified pixels), *α* is a parameter that controls the emphasis on false positives and *β* is a parameter that controls the emphasis on false negatives.

The Twersky Loss allows adjusting parameters *α* and *β* to customize the loss function's behaviour. By setting different values for these parameters, the loss function can be tailored to prioritize either precision or recall. For example, if *α* is set to a larger value, the loss function will be more sensitive to false positives, which might be desirable in scenarios where minimizing false positives is crucial. Conversely, if *β* is set to a larger value, the loss function will focus more on reducing false negatives, which could be important in tasks where avoiding missed detections is critical as the detection of symptoms.

b. Data pre-processing

Three different image sizes were tested as input for the ResUNet: the whole image, thumbnails from the original image of size 512*512*3, following the methodology presented for the detection algorithms, with overlapping, and the same thumbnails shape (512*512*3) but also with a reduction of 4 times the number of pixels (removing every other row and column, resulting in 256*256*3 thumbnails). This reduction aimed to improve computation times and potentially enhance algorithm results (for instance, in the case of symptomatic shoots of FD segmentation, this resolution reduction might prevent the detection of very thin green shoot tips that are not indicative of FD). During the prediction of test images using this last methodology, the resolution of the original image is reconstructed using bilinear interpolation. For the structure tensor, the whole images are used.

The images were separated in the following way: 75 % in training set, 10 % in validation and 15 % in test, while making sure that the proportions of pixels of each class were similar in each set. A stratified 5-fold cross-validation was performed on the methodology yielded to the best results (70% in training, 15% in validation, 15% in test). For the images in training, a data augmentation randomly applies one or more of the following operations: image rotation, pixel dropout, contrast adjustment, and brightness change.

c. Datasets

i. Flavescence dorée

After having studied the main markers for FD, the coloured leaves, by using deep detection algorithm, the first symptom studied by segmentation method was the unlignified shoots. The ResUNet and structure tensor algorithms were initially tested for the segmentation of this symptom alone, using images from the dataset 'SegFD' where only the shoots were annotated. Subsequently, healthy bunches and symptomatic bunches of FD were annotated on the same images, and the ResUNet algorithm was retrained to segment these three classes on the same dataset. The description of these symptoms and the creation of segmentation masks were detailed in Chapter III Section D.

ii. Grapevine trunk diseases

Segmentation for the GTDs images was explored to delineate the grapevine trunk, its foliage, and the background, with the aim of potentially enhancing disease diagnosis. For this disease, only the ResUNet model was investigated using the 'SegGTD' dataset.

d. Results

i. Flavesence dorée

The comparison of results between the Structure Tensor algorithm and the ResUNet algorithm for segmenting symptomatic shoots of FD is available and discussed in Appendix 5. This section will only address the results obtained by ResUNet, which are superior for the segmentation of the symptomatic shoots of FD, in addition to the fact that ResUNet is capable of segmenting multiple types of symptoms. The results of the symptomatic shoots of FD segmentation by the ResUNEt are available in Table 31.

Table 31. Symptomatic shoots of FD segmentation results at pixel and object scales using ResUNet.

Algorithm	Input size	Pixel precision	Pixel recall	Object precision	Object recall
ResUNEt	2048*2448*3 (whole image)	0.53	0.59	0.74	0.52
ResUNEt	512*512*3 (patches)	0.64	0.53	0.76	0.58
ResUNEt	256*256*3 (1:4 patches)	**0.69**	**0.58**	**0.82**	**0.59**

The ResUnet algorithm reached the best results for the Tversky loss function (those presented in this table), five levels of depth, $\alpha = \beta = 0.8$ for batches of 30 images during 300 epochs.

According to the results in Table 31, optimal performance was attained for the ResUNet by employing the 256*256*3 patches (512*512*3 image patches with 1:4 downsizing) with (p=0.82, r=0.59) for the object metric. This technique significantly bolsters precision during result comparisons (for the object metric, p=0.74 for the whole images as input and p=0.76 for the 512*512*3 patches). This phenomenon stems from the fact that petioles (leaf stems), resembling shoots in their elongated shape and green colour, are frequently predicted by the algorithm as they naturally appear. Nevertheless, the resolution reduction diminishes their distinctiveness in the images, given their slender nature, resulting in fewer predictions by the algorithm of them being branches. For comparison, none of the pixels of symptomatic shoots were predicted by the classic 'Categorical cross-entropy' loss function.

Subsequently, annotations for healthy and symptomatic bunches were added to those of shoots on the segmentation masks. The algorithm was then retrained to detect both symptomatic

shoots and bunches, as well as healthy bunches. The results of the segmentation of these 3 classes are presented in Table 32 and examples of prediction of these 3 classes are presented in Figure 31.

*Table 32. Results of the ResUNet for a 5-fold cross-validation with image patch of size 256*256*3.*

Class	Pixel precision	Pixel recall	Object precision	Object recall
Symptomatic shoot	0.72	0.58	0.83	0.57
Symptomatic bunch	0.85	0.47	0.78	0.40
Healthy bunch	0.89	0.8	0.85	0.68

For the segmentation of the symptomatic bunches, the precision was high, but the recall not so good ((p=0.78, r=0.40) for the object metric). This can be explained by the fact that i) the symptomatic bunches that had lost all their berries were very complicated to predict because of their very small size (a few pixels of thickness), and ii) the division of the resolution by 2 of the original images (to reduce the prediction time) reduced the size of the symptomatic bunches even further. The segmentation of the healthy bunches showed pretty good results ((p=0.85, r=0.68) for the object metric). Recall was also lower than precision. Once again, smaller bunches and bunches overlaid by leaves were less accurately detected.

Figure 31. Examples of predictions of the ResUNet for the 3 classes. First column: raw image of a white grape variety and two of red grape variety. Second column: Prediction of the ResUNet overlaid with the image. In green: Symptomatic shoots, in yellow: symptomatic bunches, in blue: healthy bunches.

It can be seen in Figure 31 that the symptomatic shoots and bunches and the healthy bunches were rather well detected in these three examples.

ii. Grapevine trunk diseases

The results of the segmentation into 3 classes ('background', 'trunk', and 'foliage') in GTDs were studied using the ResUNet algorithm. The input consisted of either full images, image patches of size 512*512*3, or 256*256*3 (with the resolution divided by 4 as in the previous section). Only the cross-entropy and weighted cross-entropy (with weights corresponding to the inverse of the proportion of pixels for each class.) loss functions have been tested here, as the classes are not significantly imbalanced. While the 'trunk' class is a minority, it is in proportions such that the network cannot disregard this class to achieve good results. The results are presented in Table 33. The object-level metric is not used here since the objects of interest (trunk and foliage) are contiguous (there was only one object of these class in each image).

Table 33. Results of the ResUNet for the segmentation of the GTDs images with the weighted cross-entropy loss function.

Input shape	Trunk (p, r)	Foliage (p, r)	Background (p, r)
2048*2448*3	**(0.86, 0.95)**	**(0.93, 0.92)**	**(0.97, 0.97)**
512*512*3	(0.82, 0.89)	(0.88, 0.87)	(0.95, 0.94)
256*256*3	(0.76, 0.54)	(0.82, 0.84)	(0.94, 0.93)

Segmentation on large images performs better than using patch-based processing. This seems logical because providing the entire image to the network helps it understand the location of the trunk and foliage, which are almost always in the same positions. An overall view seems more appropriate for this segmentation. On a small patch, it can be challenging for the network to distinguish between the foliage of the studied grapevine and the foliage of the grapevine in the row behind for instance. The weighted cross-entropy performed better for each of the input shape. Best results for the cross-entropy were for the input shape of 2048*2448*3 and were of (p=0.83, r=0.85) for the 'Trunk', (p=0.81, r=0.84) for the 'Foliage', and (p=0.95, r=94) for the 'Background' class.

Figure 32 presents three examples of segmentation using the algorithm that achieves the best performance. In the first image, an example is shown where everything is working well: the grapevine is correctly segmented from the background, and the trunk is clearly distinguished from the rest. In the second image, two problems are apparent: grass is predicted as part of the foliage, and a small portion of the foliage is missing. However, it's noteworthy that the grass predicted as foliage is very close to the trunk, which also exhibits a 'Half head' symptom on that side. It's possible that the detection network might not have identified this 'Half head' due to the grass resembling foliage. Yet, through segmentation, even though some pixels of grass are wrongly detected as pixels of the 'Foliage' class, a clear distinction between them and the trunk is visible, likely enabling the accurate identification of the symptom later on. Regarding the third image, a portion of the vehicle is predicted as either trunk or foliage. The training dataset lacks sufficient images of this type (where the vehicle is visible) to accurately differentiate it from the rest. However, this should not be a problem moving forward, as long as it doesn't distort symptoms.

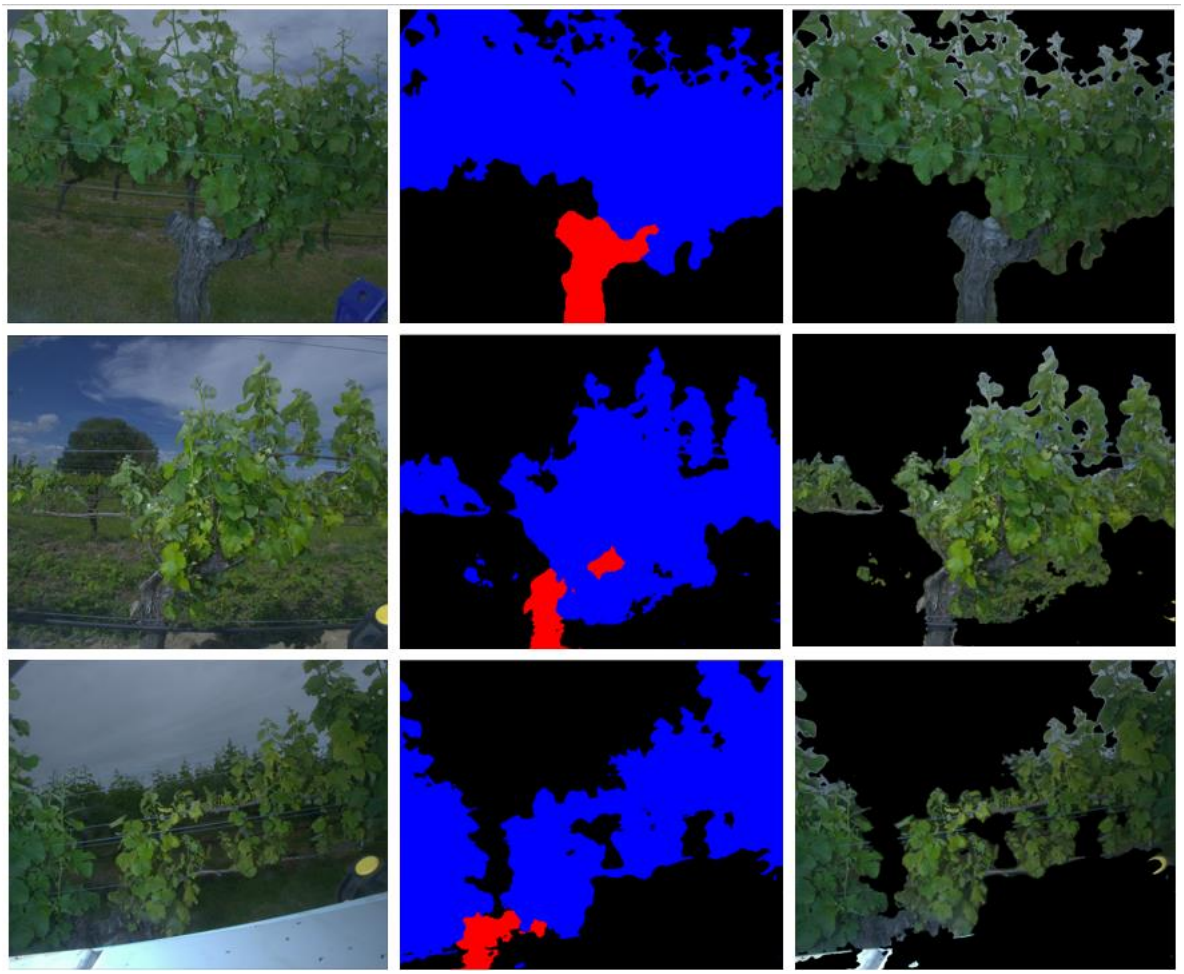

Figure 32. Example of prediction with ResUNet for the segmentation of GTDs images. First column: raw images of white grape variety. Second column: prediction mask by the algorithm (in red: 'Trunk', in blue: 'Foliage', in black: all the rest). Third column: Overlay of the raw image and the predictions to segment the studied grapevine to the background.

C. Synthesis

Whether it was for the detection or segmentation of FD symptoms, the results in precision and recall of the analyses were modest. In the case of detecting symptomatic FD leaves, the best results were achieved for red grape varieties because white grape varieties displayed less prominent symptomatic FD leaves. There was no methodology for red grape varieties that consistently outperforms the other analysis methods studied in this study. However, the YOLOv8m algorithm seems to perform better on average across these red varieties. For white grape varieties, one algorithm stands out: YOLOv8m using whole image inputs. Results for this methodology are, for the red and white variety, reaching the bests results with (p=0.81, r=0.63) for the CF21 test set and (p=0.62, r=0.75) for the SB21 test set. As for the segmentation of symptomatic shoots, symptomatic bunches, and healthy bunches, the best results are obtained with the ResUNet model by dividing images into 512*512*3 patches and reducing the resolution of each patch by a factor of four. The object metrics results are (p=0.83, r=0.57), (p=0.78, r=0.40) and (p=0.85, r=0.68) for respectively the 'Symptomatic shoot', 'Symptomatic bunch' and 'Healthy bunch' classes.

Regarding the detection of GTDs symptoms, the best precision and recall results are achieved with YOLOv8m using whole image inputs. Additionally, there is a noticeable difference in results between images acquired before and after 25/11/2022 (acquisition date appearing to mark a change

in vegetation density). The best results for more advanced grapevine development stages are (p=0.85, r=0.81) for 'Half head' symptoms and (p=0.58, r=0.58) for 'Short shoot' symptoms. Unfortunately, the 'Canker' symptoms are never detected due to their low frequency in the training dataset. Finally, the segmentation of GTDs images into three classes achieves its best results by using the entire images as input for the ResUNet algorithm. The results (using the pixel metric) are (p=0.86, r=0.95) for the 'Trunk' class, (p=0.93, r=0.92) for the 'Foliage' class, and (p=0.97, r=0.97) for the 'Background' class.

The modest results of the symptom detection were somewhat expected, considering the complexity of the task (high variability in symptom expression, occlusion, presence of confounding symptoms). However, the redundancy and structure of the detected symptoms on the grapevine could potentially overcome some of their inaccuracies, in order to achieve a precise diagnosis at the vine scale. In the next chapter, the way to combine these detections will be explored in order to obtain the best possible plant-scale diagnostic outcomes.

VI. Symptom detections association

Table of content

In this chapter, the methodologies developed to address the challenge of disease diagnosis based on prior symptom detections are explored. The previous chapter demonstrated that it was feasible to associate a rich collection of indications with each image, encompassing both the evident disease symptoms and the confounding symptoms. These indications, originating from various sources, serve as a resource for potential disease detection. In this regard, two approaches have been explored to achieve the most accurate diagnosis possible. The first approach relied on feature extraction from the detected symptoms, followed by the application of a machine learning method, specifically Random Forest (RF) (Ho, 1995). The second innovative approach, hinges on representing the intricate network of detected symptoms in images as a graph. To efficiently exploit this graph structure, a Graph Neural Network or GNN (Scarselli *et al.*, 2009) model has been applied. This approach has the potential to capture relationships between symptoms and minimize information loss during the association phase of detections. Throughout this chapter, both approaches will be examined in detail, highlighting the steps of their implementation, challenges encountered, and the results obtained. The workflows of these two 2-step methodologies are presented in Figure 33.

Only the diagnosis of FD will be studied in this chapter. The numerous symptoms of the disease and confounding symptoms present in the images are well suited to the 2-step methodologies. However, when it comes to GTDs, an image of a grapevine affected by this disease contains on average only 2 symptoms. It thus seems inappropriate to apply such methods to associate detected symptoms.

The symptom association methodologies (using either random forests or graphs) have partly been presented in previous articles (Tardif *et al.*, 2022; Tardif *et al.*, 2023a; Tardif *et al.*, 2023b). This chapter brings an additional dimension by a deeper explanation of the algorithm's principles and a more in-depth experimental evaluation of these methodologies. Regarding experimental design, variations are introduced in algorithm hyperparameters and new leaf detections obtained with the YOLOv8m algorithm were used as priors (while only the YOLOv4-tiny algorithm was used in the previous articles).

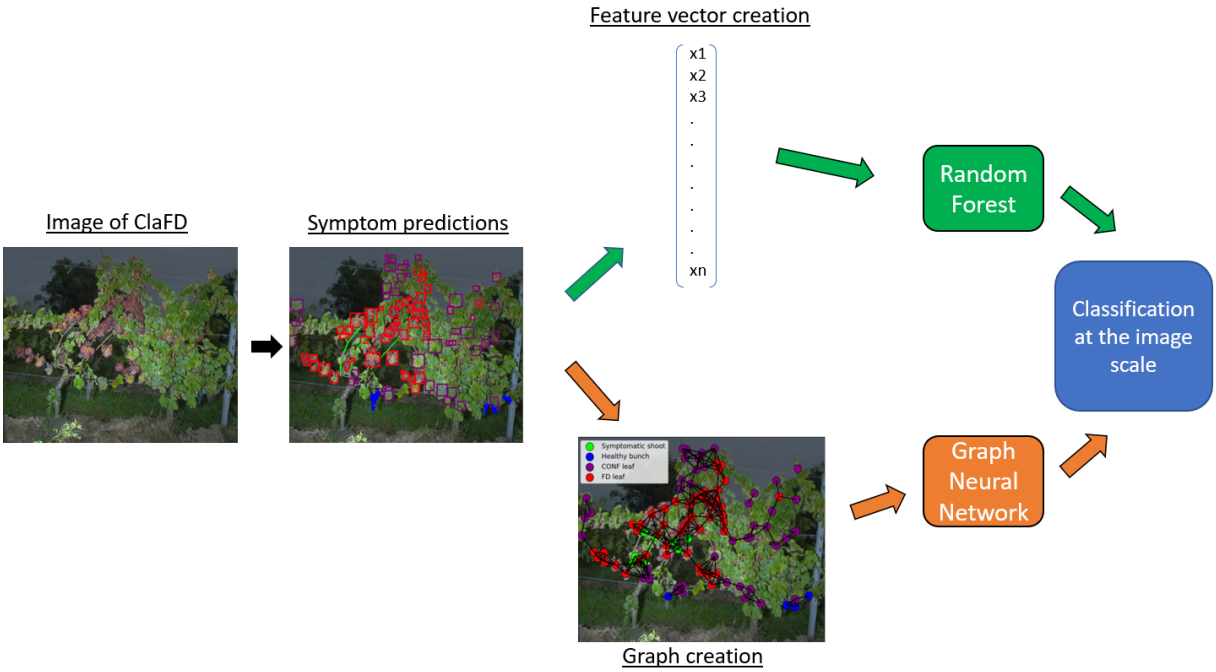

Figure 33. Representation of the two symptom association methodologies. Symptoms are detected in the same manner for both methods. In the RF methodology, a feature vector is calculated from these detections and is used as input by an RF classifier to provide a decision at the image scale. In the graph-based methodology, a graph is created from the detected symptoms and uses as input of a Graph Neural Network to provide a decision at the image scale.

A. Symptoms association using a Random Forest classifier

a. Feature vectors creation

The RF methodology involves the creation of an input information vector. A specific information vector was created for each image. The procedure for creating the information vector was as follows: For each image, symptoms were predicted using symptom prediction algorithms trained on an independent dataset, as discussed in Chapter V. The 'SegFD' dataset was used for the segmentation of symptomatic shoots, symptomatic and healthy bunches. The 'DetFD' dataset was used for the detection of symptomatic leaves. The chosen segmentation algorithm that achieved the best results in Chapter V, ResUNet (Diakogiannis *et al.*, 2020), took 256*256*3 pixel patches as input. Symptomatic leaves from FD were predicted in two ways: either by the YOLOv4-tiny algorithm (Bochkovskiy *et al.*, 2020), taking 416*416*3 pixel patches as input, or by the YOLOv8m (Jocher *et al.*, 2023) algorithm, taking entire images of 1088*1216*3 pixels as input. This allowed the study of the impact of slightly improved leaf prediction on the final classification, and answered the question if even better results could be expected by improving the average leaf detection outcomes. The trainings used to predict the symptomatic leaves were those whose results were presented in Chapter V. The symptomatic leaves of white grape varieties were predicted using networks trained on all white grape varieties (UB20 + UB21 + SB21), and those of red grape varieties were predicted using networks trained on all red grape varieties (CS20 + CF21 + M21 + CS21). A feature vector was computed from the detections made on the image. This vector was put at the input of the RF classifier which delivered a classification at the image scale in output (Figure 33). For the creation of the feature vector, 16 features were studied, as described below.

From the detection of individual leaves, 8 parameters were calculated. The number of elements in each class, as well as the average confidence score of each class, were computed. A high

number of elements in either the ‘FD leaf’ or ‘Esca leaf’ classes, along with a high average confidence score, could indicate the presence of one of these two diseases. For the 'FD leaf' and 'Esca leaf' classes, the number of 'neighbouring' symptomatic leaves was calculated. This measurement was intended to help deciding whether symptomatic leaves were located on the same shoot, as was the case for a grapevine affected by FD or Esca. This measurement was calculated as follows: The number of leaves of the same class that were pairwise within a distance of less than 200 pixels (with the centre of bounding boxes as the reference point) was counted. This yielded the first 8 parameters of the feature vector:

- x1: The total number of leaves in the ‘FD leaf’ class.
- x2: The average of the ‘FD leaf’ class confidence scores.
- x3: The total number of leaves in the ‘Esca leaf’ class.
- x4: The average of the ‘Esca leaf’ class confidence scores.
- x5: The total number of leaves in the ‘CONF leaf’ class.
- x6: The average of the ‘Conf leaf’ class confidence scores
- x7: The number of spatially close leaves of the ‘FD leaf’ class.
- x8: The number of spatially close leaves of the ‘Esca leaf’ class.

Then, based solely on the results of the segmentation algorithm, the following parameters were computed: the numbers of elements (a connected component issued by the segmentation algorithm) in the classes 'Symptomatic shoot', 'Symptomatic bunch', and 'Healthy bunch'. A large number of symptomatic shoots or bunches could indicate the presence of the disease, whereas a large number of healthy bunches could indicate the absence of the disease.

In an attempt to differentiate between petioles or shoot tips that could be segmented as symptomatic shoots, the maximum thickness among segmented symptomatic shoots was calculated and included as a parameter in the feature vector. The minimum distance between a symptomatic shoot detection and a symptomatic bunch was indicated in the information vector. If the FD symptoms were localized in a specific part of the grapevine, as could be the case, this parameter could help confirm true predictions or, conversely, distinguish false prediction if the distance was large. Similarly, the minimum distance between a healthy bunch and a symptomatic branch was calculated, as the healthy bunch, if present, should ideally be far from the affected side of the grapevine. The significance of these last two parameters (distances between a symptomatic shoot and either a symptomatic bunch or a healthy bunch) was not certain, and the importance of the parameters in the final algorithm decision was produced to understand their real impacts on disease diagnosis. This introduced six new parameters to the feature vector:

- x9: The total number of shoots in the ‘Symptomatic shoot’ class.
- x10: The total number of shoots in the ‘Symptomatic bunch’ class.
- x11: The total number of shoots in the ‘Healthy bunch’ class.
- x12: The maximum thickness of the shoots in the ‘Symptomatic shoot’ class
- x13: The minimum distance between instances of the ‘Symptomatic shoot’ and ‘Symptomatic bunch’ classes.
- x14: The minimum distance between instances of the ‘Symptomatic shoot’ and ‘Healthy bunch’ classes.

Finally, by combining the detections from the two algorithms, 2 new parameters were calculated. The minimum distances between an element of the 'symptomatic shoot' class and the 'FD leaf' class, as well as the 'Esca leaf' class, were computed. Again, a small distance could indicate the presence of FD. Regarding Esca, which doesn't produce symptoms on shoots or bunches, this parameter shouldn't typically hold significant importance in the final classification. This has been confirmed or refuted in the subsequent stages of the study.

- x15: The minimum distance between instances of the ‘Symptomatic shoot’ and ‘FD leaf’ classes.

- x16: The minimum distance between instances of the ‘Symptomatic shoot’ and ‘Esca leaf’ classes.

Many other studies (Albetis *et al.*, 2017; Boulent *et al.*, 2020; Musci *et al.*, 2020) consider leaf information alone, sufficient to accurately diagnose FD and differentiate it from its confounding diseases. To determine if the addition of symptomatic information from shoots and bunches would bring a real improvement, two different information vectors were investigated. The first vector consisted only of features calculated from leaf detection (from x1 to x8), while the second contained all the parameters presented above (from x1 to x16). A total of 4 feature vectors were examined and summarized in Table 34.

Table 34. Description of the studied feature vectors used for the diagnosis of FD with the RF methodology.

Feature vector name	Detection algorithm	Features
VectYOLOv4leaves	YOLOv4-tiny	Only leaf features
VectYOLOv4all	YOLOv4-tiny	All features
VectYOLOv8leaves	YOLOv8m	Only leaf features
VectYOLOv8all	YOLOv8m	All features

The Random Forest classifier was chosen to process these feature vectors. The following section provides a detailed explanation of its functioning and the reasons behind its selection among other possible approaches.

b. Random Forest classifier

Random Forest is a powerful machine learning algorithm that falls under the category of ensemble learning methods. It is widely used for both classification and regression tasks. The algorithm is known for its effectiveness, robustness, and ability to handle complex datasets.
Here's how the RF algorithm works:

- Ensemble Learning Concept: RF is an ensemble of decision trees. Ensemble learning involves combining multiple models (in this case, decision trees) to improve the overall performance and generalization of the model. RF creates a "forest" and then combines their predictions to make more accurate and robust predictions.
- Decision Trees: Decision trees are a type of supervised machine learning model that makes decisions based on a series of if-else conditions. Each node in the tree represents a feature, and the tree is split into branches based on the feature values. At the leaf nodes, the model provides the final prediction.
- Voting or Averaging: During the prediction phase, each tree in the RF makes its own prediction based on the input features. For classification tasks, the class that receives the most votes from the individual trees becomes the final prediction. For regression tasks, the predictions from all trees are averaged to obtain the final prediction.
- Bootstrapping: RF employs a technique called "bootstrapping" to create different subsets of the original dataset. Bootstrapping involves randomly sampling the data with replacement, which means some samples may appear multiple times in a subset while others may not appear at all. This process creates diversity among the subsets.
- Random Feature Selection: For each tree in the RF, only a subset of features is considered at each split. This introduces further randomness and diversity among the trees. The number of features to consider at each split is typically smaller than the total number of features. This process prevents the model from becoming too biased towards dominant features.

- Building Trees: Each decision tree in the RF is built using the bootstrapped dataset and the randomly selected subset of features. The trees are grown deep but not pruned, which can lead to overfitting for individual trees. However, since the RF combines predictions from multiple trees, the overfitting tendencies of individual trees are mitigated.

Three hyperparameters are important during the training phase of a Random Forest classifier: i) The Number of Decision Trees (NDT), ii) the Maximum number of Samples (MS) that were used to train each individual decision tree in the ensemble. This parameter controls the amount of data that is randomly sampled for each tree's training process. iii) the Maximum number of Features (MF) that were considered when making a split at each node of an individual decision tree in the ensemble. Each node uses only one feature at a time to make its splitting decision, but this feature is chosen from a limited subset of features based on the value of MF. The workflow of a RF classifier is illustrated in Figure 34.

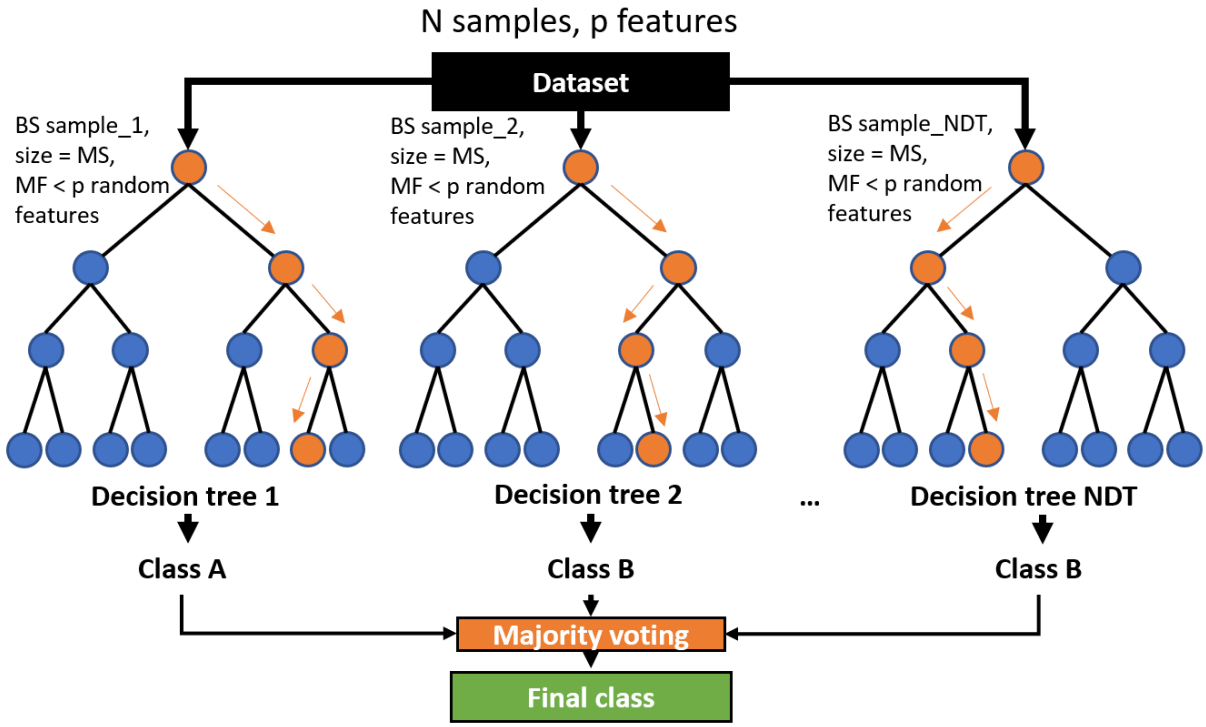

Figure 34. Workflow of a Random Forest (RF) classifier. NDT stands for the Number of Decision Trees, MS for the Maximum number of Samples and MF for the maximum number of Features.

The RF classifier has been chosen as the first methodology to associate the symptom detections and deliver a diagnosis at the image scale because of its advantages:

- Feature Importance: RF provides a measure of feature importance. It calculates the contribution of each feature in the model's predictions. This information is valuable for understanding which features are most relevant to the task and can aid in feature selection or feature engineering. This renders RF a suitable methodology to initially test.
- Easily and quickly implementable: RF allows a practical association of symptom methodology in order to obtain preliminary results and test different types of associations.

Other approaches could have also been tested instead of RF for classifying the feature vectors. Among them, some popular can be mentioned:

- Support Vector Machines (SVM) (Cortes & Vapnik, 1995): SVMs seek to find an optimal hyperplane that separates classes in a feature space.

- K-Nearest Neighbours (KNN) (Cunningham & Delany, 2007): KNN is a proximity-based classification method. It assigns a class to a vector based on the classes of the k nearest neighbours.
- Artificial Neural Networks (ANN) (McCulloch & Pitts, 1943): Neural networks are deep learning models that can be used for classification tasks.

B. Symptom association using a graph-based methodology

To better preserve information about individual symptom detections (without reducing them to a feature vector) during the image-scale diagnosis, an attempt was made to represent these detections in the form of a "graph". This graph was then processed by a GNN to deliver a classification at the graph scale, i.e. at the image scale.

a. Understanding graphs: types and characteristics

A graph is a mathematical representation of structured data where entities, termed 'nodes' are interconnected via relationships known as "edges" or "links". Nodes symbolize individual entities or elements present in the dataset. Each node is associated with a feature vector that encodes pertinent details about that entity. For example, in text data, this might include word embeddings, while in social networks, it could contain user attributes. Edges denote the connections or links between nodes. For instance, in a social network, edges could represent friendships. Not all nodes are required to be linked, and some nodes may have no connection to any other nodes.

Graphs can be undirected or directed. In an undirected graph, edges have no direction. If there's an edge between nodes A and B, it implies that there's a relationship between A and B in both directions. Undirected graphs are used to represent relationships without a clear "source" and "destination". However, in a directed graph, edges have a direction. An edge from node A to node B signifies a relationship that goes from A to B only. Directed graphs are used to model asymmetric relationships or flows of information.

These connections can be bi-directional or unidirectional, and they might carry additional attributes or weights that signify relationship strength or type.

Graphs can also be either heterogeneous (Wang *et al.*, 2023) or homogeneous (Bo, 2023). A homogeneous graph is a type of graph where all nodes and edges belong to the same type or category. In other words, all nodes in a homogeneous graph share the same set of attributes and have the same type of relationships with other nodes. A heterogeneous graph, on the other hand, contains multiple types of nodes and edges, each associated with distinct attributes and relationship types. Heterogeneous graphs are particularly useful for representing complex systems where entities have various characteristics and interact in diverse ways. Figure 35 illustrates four different graph architectures.

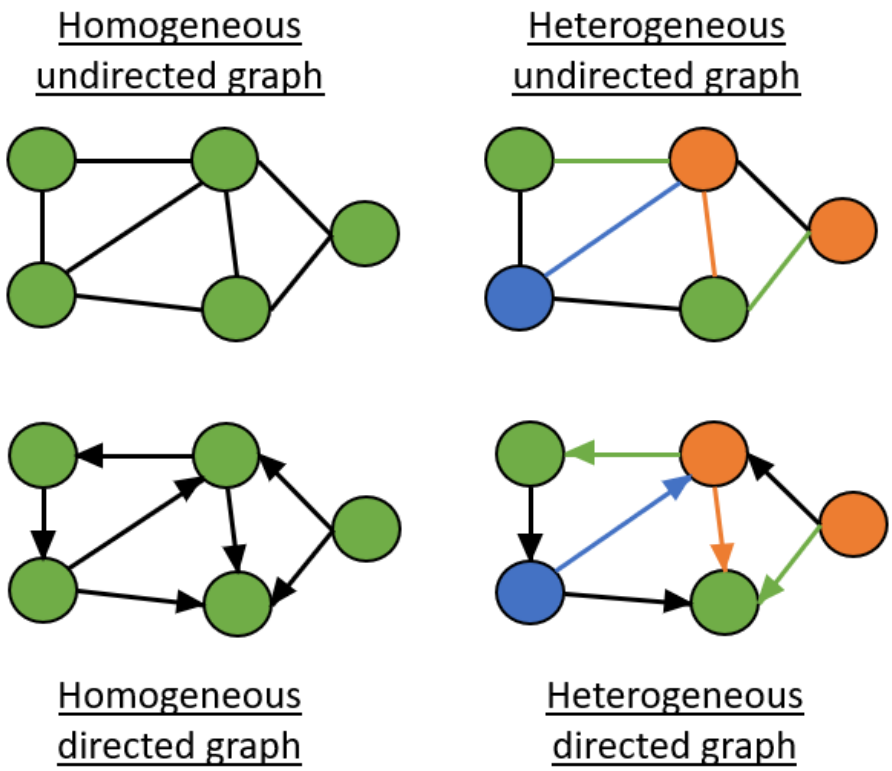

Figure 35. Four possibilities when creating a graph (not exhaustive). The homogeneous graph has all its nodes in the same class (symbolized here by nodes of the same colour) and is undirected if the edges do not contain information about the target and source nodes. The heterogeneous graph has nodes of different types (nodes of different colours) and can also have edges of different types (edges of different colours). Heterogeneous graphs can be either directed or undirected.

b. Graph creation from symptom detections

The initial part of the graph-based methodology is the same as that of the RF (see Figure 33). Symptoms were predicted using the two symptom detection algorithms. However, instead of creating an information vector from the detections, each detection is represented as a node in the graph. This way, each node can be provided with a distinct information vector, minimizing the loss of information. The hypothesis made here is that the proximity between symptoms is a relevant indicator for disease diagnosis, and therefore, it is necessary to connect them during the construction of the graph. To account for these distances in the graphs, nodes were connected if they were within a certain threshold distance (calculated in pixels using the centres of bounding boxes or predicted segmented objects). In this case, the created graphs were undirected, as there was neither a starting symptom nor and a target symptom. The decision was made to create homogeneous graphs because they are easier to manipulate. In the case of the final classification results had not been satisfactory, the creation of heterogeneous graphs would have been tested. All the detected symptoms have been placed in the same class and therefore share a feature vector of the same type. However, the classes of these symptoms have been encoded in the feature vector as dummy variables (1 if the node belongs to the class, 0 otherwise). Figure 36 illustrates the process of creating the graphs. The image has been overlaid by the graph for better understanding. However, the nodes did not have any specific spatial arrangement when graphs were created. The only known information was which nodes were connected to each other (which means: which symptoms were close).

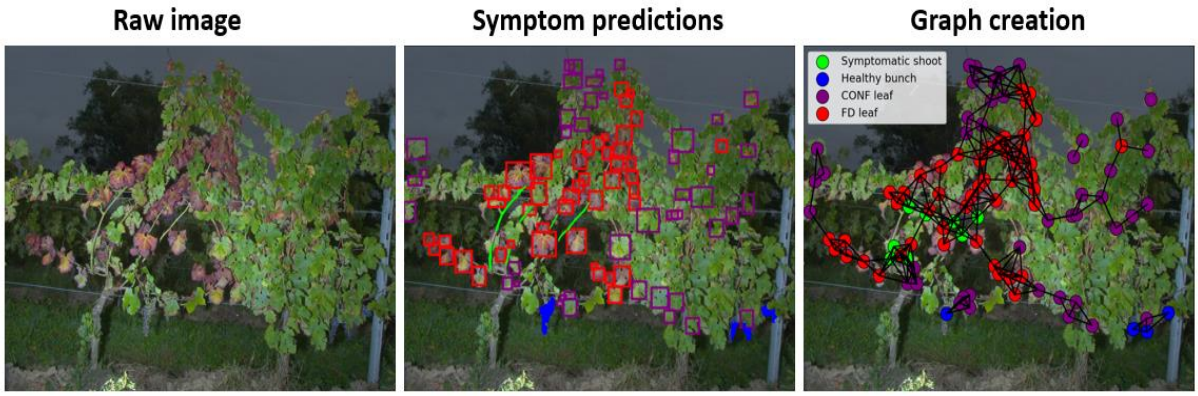

Figure 36. Process of creating a graph. Example for an image showing a grapevine of red grape variety. Symptoms were first detected by the symptom predictions algorithms. For the creation of the graph, each predicted symptom represented a node (the circles) and has been connected to another one if their distance was less than 200 pixels.

As for mathematical representation and computer implementation, a graph G is divided into two matrices: the feature matrix X and the adjacency matrix A:

- The feature matrix X provides the initial node attributes. Each row corresponds to a node, and the columns represent the various attributes or features assigned to the nodes. The features could be binary, categorical, or continuous values that describe the properties of the nodes. In the case of a categorical feature, each modality (category) can be encoded by a dummy variable, i.e. a binary variable that indicates the belonging to the modality.
- The adjacency matrix A is a square matrix of binary values that represents the relationships between nodes in a graph. In an undirected graph, the entry at position *(i,j)* indicates whether there is an edge between nodes *i* and *j* (*i* and *j* refer to the row number of the nodes in the feature matrix) with a 1 if they are linked or 0 otherwise.

Figure 37 illustrates this decomposition into feature and adjacency matrices.

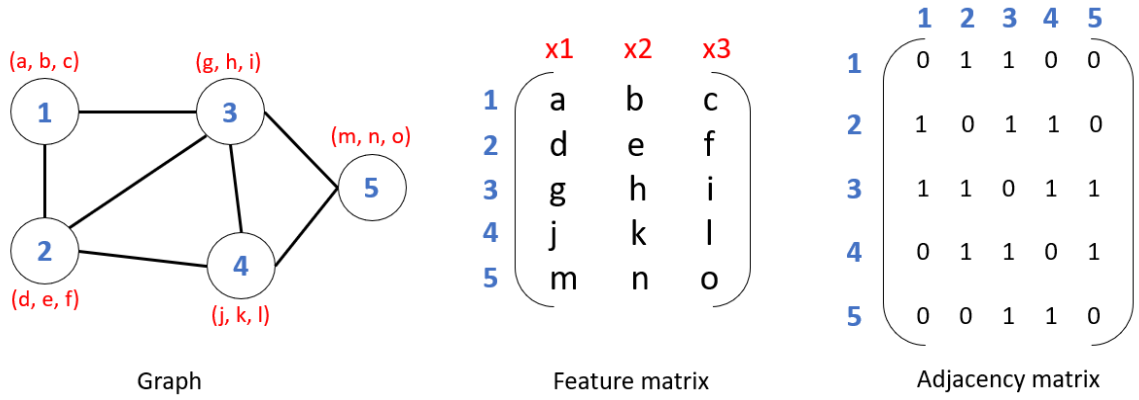

Figure 37. Example of decomposing a 5-nodes graph into feature and adjacency matrices. The nodes are numbered according to their order of appearance in the feature matrix. Each node contains a 3-features vector.

To study the impact of node choice in graphs as well as of node linkage distance, various distance thresholds (100, 200, 400, 800 pixels) were evaluated (illustrated in Figure 38), and three distinct node sets (Figure 39) were compared:

- The first node set, termed 'leafgraph,' was constructed exclusively from symptomatic leaf detections. Each node corresponding to a detected 'FD symptomatic leaf,' an 'Esca leaf,' or a 'confounding leaf' was represented by an 8-feature vector:

$$\text{leafgraph}: [1_{FD}, 1_{ESCA}, 1_{CONF}, Score_{FD}, Score_{ESCA}, Score_{CONF}, x, y],$$

where 1_Z equals 1 if the detected leaf symptom is of class Z, and 0 otherwise. $Score_Z$ reflected the confidence score from the leaf detection algorithm for a leaf of class Z, remaining 0 otherwise. The coordinates x and y in pixels of each symptoms were also added in order to potentially assist the algorithm in recognizing spatial arrangements of symptoms within the image, in addition to the linked nodes.

- The second node set encompassed all detected symptoms ('allgraph'). In addition to the first graph's attributes, the feature vector encoded the symptomatic shoots' length and width and the areas of both symptomatic and healthy bunches. Each node was thus described by a 15-parameter vector:

$$\text{allgraph}: [1_{FD}, 1_{Esca}, 1_{Conf}, 1_{Shoot}, 1_{Sympt.bunch}, Score_{FD}, Score_{Esca}, Score_{Conf}, Length_{Shoot}, \text{Width}_{Shoot}, \text{Area}_{Sympt.bunch}, \text{Area}_{Health.bunch}, x, y]$$

- The third node set ('allgraph_withoutconf') mirrored the second, but without the information on the 'Confounding leaf' to investigate the impact of this symptom on the prediction:

$$\text{allgraph\_withoutconf}: [1_{FD}, 1_{Esca}, 1_{Shoot}, 1_{Sympt.bunch}, \text{Score}_{FD}, \text{Score}_{Esca}, \text{Length}_{Shoot}, \text{Width}_{Shoot}, \text{Area}_{Sympt.bunch}, \text{Area}_{Health.bunch}, x, y]$$

These symptoms, often highly prominent in the images, could either obscure useful information and lead to poorer results or enhance the ability to effectively distinguish FD from its confounding diseases.

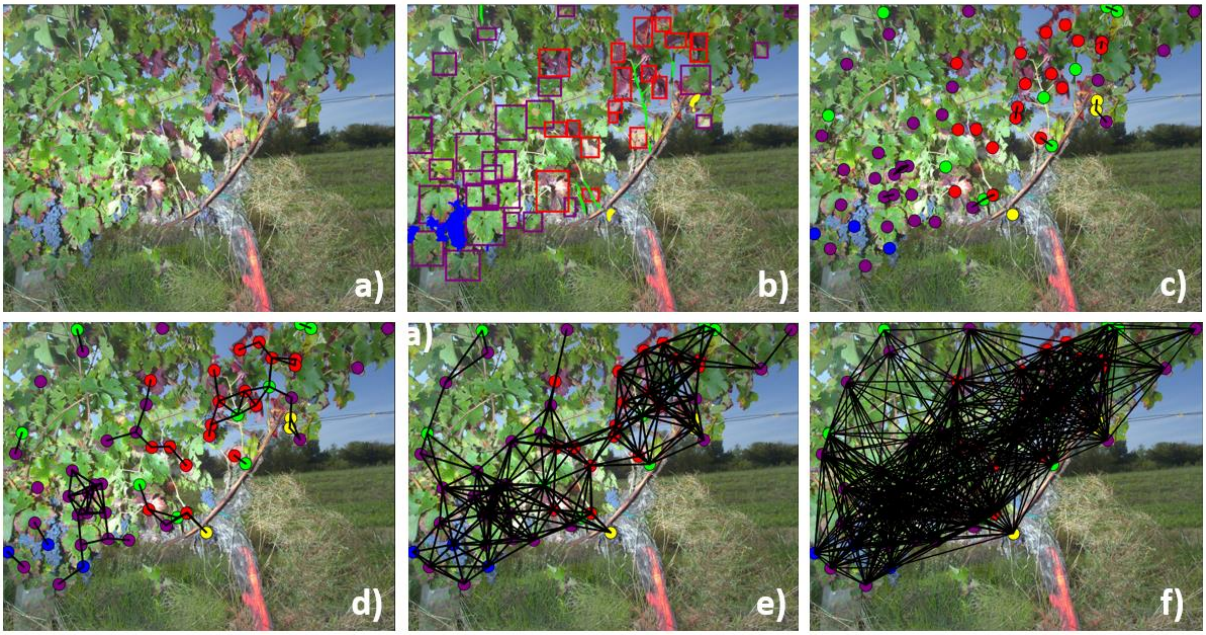

Figure 38. Illustration of the graph creation with increasing link distance thresholds. a) Raw image, b) symptom detections. Graph with a distance threshold of c) 100, d) 200, e) 400 and f) 800 pixels. The detected symptoms and other occurrences are represented by different colours: in purple 'Confounding leaf', in red 'FD leaf', in yellow 'Symptomatic bunch', in green 'Symptomatic shoot', in blue 'Healthy bunch'.

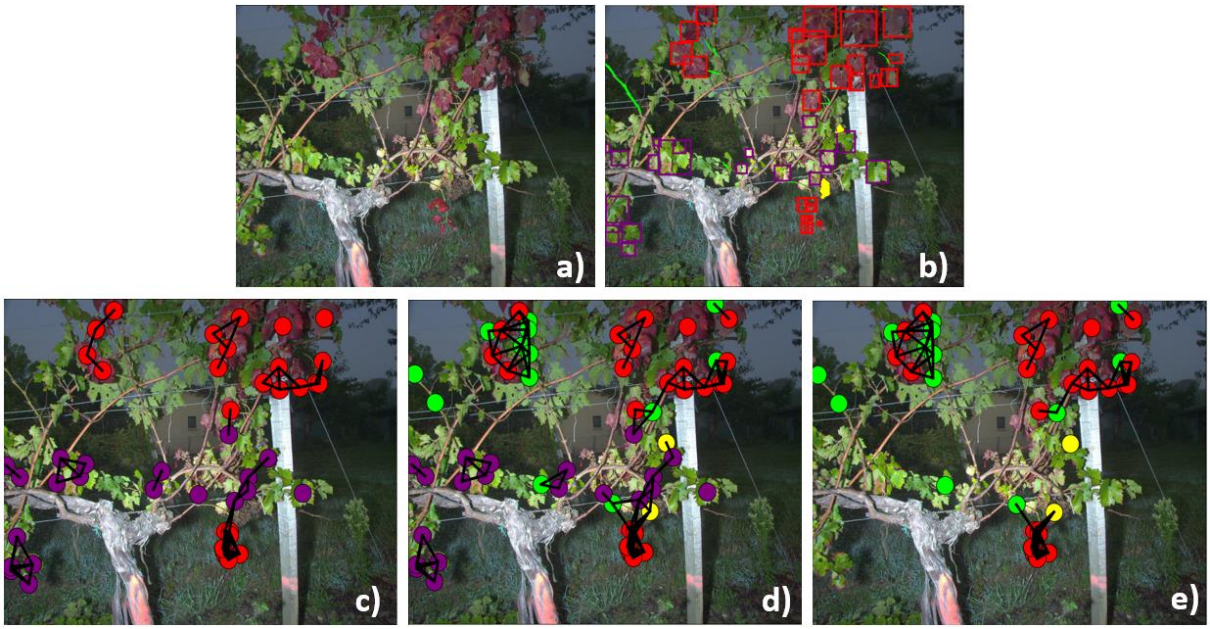

Figure 39. Comparison of the graphs issued from 3 node sets. a) Raw image, b) symptom predictions. Graphs created from the 3 node sets: c) 'leafgraph' only, d) 'allgraph', e) 'allgraph_withoutconf'. Distance threshold was 100 pixels for the 3 configurations.

c. Graph classification algorithm

Algorithms adapted to graph data can be applied to various tasks, which often fall into three main categories: node-level tasks, link-level tasks and graph-level tasks. Node-level tasks focus on individual nodes, including classification (Bhagat *et al.*, 2011; Xiao *et al.*, 2021), embedding learning (Garcia Duran & Niepert, 2017; Pan *et al.*, 2020) and anomaly detection (Akoglu *et al.*, 2015; Ma *et al.*, 2021). These tasks offer insights into the characteristics and behaviours of individual nodes. Link-level tasks revolve around analysing and predicting the connections within a graph. These tasks include link prediction and classification (M. Wang *et al.*, 2021; M. Zhang & Chen, 2018). Graph-level tasks are centred around understanding and analysing entire graphs as holistic entities. These tasks involve the classification (Morris, 2022), regression (Saigo *et al.*, 2009), or generation (Yang *et al.*, 2018; Zhu *et al.*, 2022) of entire graphs based on their structural, topological, or attribute properties. Graph classification entails assigning labels or categories to entire graphs. Graph regression focuses on predicting continuous graph-level properties. Graph generation aims to create new graphs that adhere to certain predefined patterns or distributions.

In this chapter, the focus will be on the graph-level task, specifically graph classification. Several algorithms and approaches are suitable for graph classification. For instance, graph kernels (Kriege *et al.*, 2020; Vishwanathan *et al.*, 2010) assess graph similarities by considering subgraph patterns and structural features. Inspired by the success of transformers in natural language processing, graph transformers (Hu *et al.*, 2020; Yun *et al.*, 2019) leverage self-attention mechanisms for effective graph classification. Graph autoencoders (Pan *et al.*, 2018) enable unsupervised representation learning for graphs, and the acquired embeddings can be applied to downstream classification tasks. GraphSAGE (Hamilton *et al.*, 2017) emerges as a noteworthy graph-based deep learning algorithm designed to address various graph-related tasks, with a particular emphasis on efficiency for handling large graphs. Graph Convolutional Networks (GCNs) (S. Zhang *et al.*, 2019) are also a prominent choice for graph classification, employing convolutional operations to learn node representations that can be employed for graph-level classification. GCNs are part of a broader class of GNNs (Wu et al., 2021). GNNs have been studied for graph classification in this chapter.

iii. Graph Neural Networks

A GNN is a type of neural network model specifically designed to work with data structured as graphs. The main idea behind GNNs is to extend traditional neural network architectures to accommodate graph-structured data. Indeed, there is a lot of similarity between CNNs and GNNs. Both architectures are based on a stacked layer structure. Each layer performs specific operations to extract information at different scales or levels of abstraction. Both GNNs and CNNs use local operations to process data. CNNs apply local convolutions to image regions, while GNNs aggregate information from local neighbours in a graph. Both methods can use pooling techniques to reduce the size of either the image or the graph, in order to extract information at different scales and to obtain a condensed and representative form for the classification step. This is often performed in the same way for both methods: through an MLP. These common steps are illustrated and compared in Figure 40.

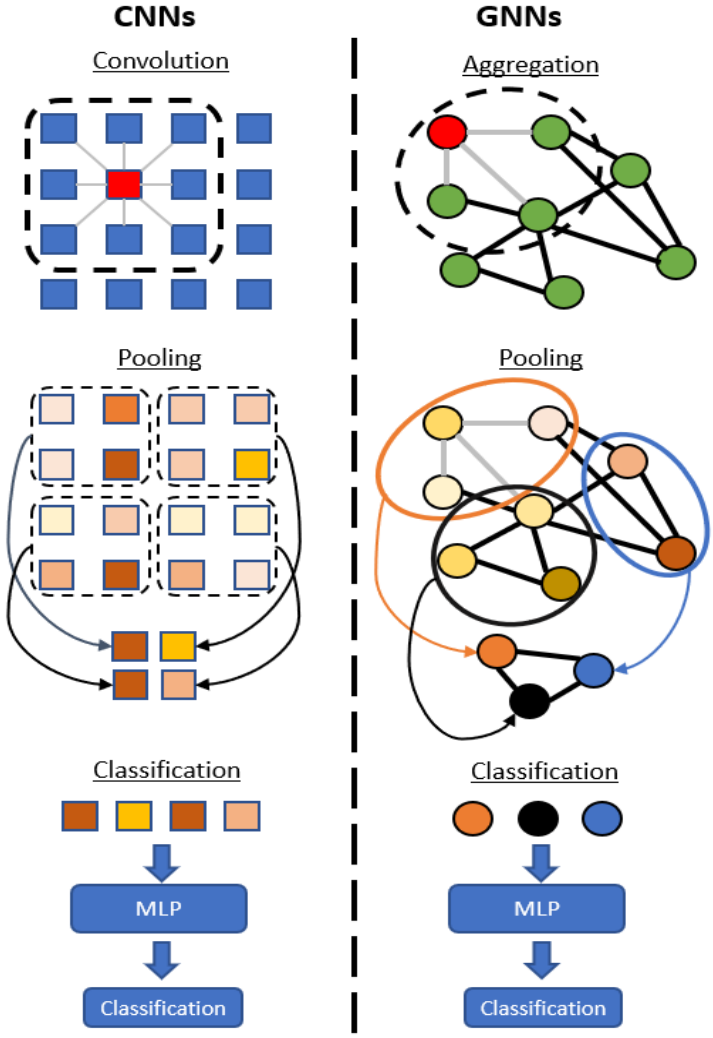

Figure 40. Comparison between the key steps of CNNs and GNNs. For CNNs, image pixels are represented by squares. Local information is processed through convolution operations. A 3x3 convolution is shown here. Information from neighbouring pixels is associated with information on each processed pixel (in red). The same principles are applied for GNNs, except that only information from nodes connected to the target node (also shown in red) is associated with the node under study. For the pooling step, the one depicted for CNNs is the widely used MaxPooling2D 2x2 pixels, where a 2x2 sliding window extracts the pixel with the highest score (darker colours indicating higher scores). For GNNs, this operation is performed by forming node clusters. These two steps (convolution/aggregation and pooling) can be applied multiple times to extract features at different levels. Finally, the mainly

used classification step is the same for both architectures: the final representation of the image or graph is flattened into a 1D vector and fed into an MLP that outputs the class prediction.

GNNs can operate directly on graph data by considering both the nodes and their neighbouring nodes' information. Here's a basic overview of how GNNs work:

- *Graph Representations*: Provide to the network a pair of feature and adjacency matrix for each graph.
- *Neighbourhood Aggregation*: GNNs update a node's representation by aggregating information from its neighbouring nodes. This can be done through a *message-passing* function (Equation 6). The idea is that the nodes are updated based on their features and the features of their neighbours, capturing both local and global structure of the graph. That is why the choice of nodes linked together is important. This aggregation is usually performed by computing a weighted sum or a combination of the neighbouring nodes' features.
- *Learnable Parameters*: GNNs include learnable parameters, just like traditional neural networks. These parameters are used to combine and transform node features during the aggregation process.
- *Layer Stacking*: Like other neural networks, GNNs can have multiple layers stacked on top of each other. In each layer, node representations are updated based on their aggregated neighbourhood information. This allows the model to capture information from nodes that are further away in the graph.
- *Activation Function*: After aggregation, an activation function (usually ReLU or similar) is applied to the aggregated features, introducing non-linearity to the model.
- *Output and Prediction*: The final node representations can be used for various downstream tasks, such as node classification, link prediction, or graph classification. In this study case, the final node representation is flattened and an MLP with a final Softmax operation is used for graph classification.

Message passing: given a graph G (A, X) with *n* nodes and *c* feature channels, $A \in M_n(n)$ representing its adjacency matrix, $X \in \mathrm{R}^{n.c}$ representing its feature matrix and c' the number of output feature channels, the message passing function used throughout the experiences of this chapter can be written as in equation 6:

$$Z = f\left(\widehat{D}^{-1}.\hat{A}.X.W\right) \tag{6}$$

with $\hat{A} = A + I$ the adjacency matrix of the graph with added self-loops, $\widehat{D}$ is the diagonal degree matrix of $\hat{A}$ with $\widehat{D}_{ii} = \sum_j \hat{A}_{ij}$, $W \in \mathrm{R}^{c.c'}$ the matrix of trainable parameters, f the non-linear activation function and $Z \in \mathrm{R}^{n.c'}$ the output of the message passing function.

This message passing function can be separated into 4 steps:

1. A linear transformation of the node matrix is executed using $Y = X.W$. This process maps the original c feature channels to c' channels.
2. The operation $\hat{A}.Y$ disseminates node-related information to both neighbouring nodes (nodes linked to the studied node) and the node itself. That is why self-loops were added to A, to take into consideration the feature vector of the studied node when aggregating the information of the linked nodes).
3. Normalization of each new feature vector by multiplying with $\widehat{D}^{-1}$, in order to keep a fixed feature scale after feature vector aggregation. In fact, $\widehat{D}_{ii}$ contain the number of linked nodes to the *i*-th node +1 with the addition of the self-loops.
4. Application of the non-linear function *f*.

iv. Pooling functions

Similar to image processing, pooling layers are crucial for enhancing performance in graph classification. They help reduce the dimensionality of graph representations, capture important features, and improve efficiency in downstream tasks like classification. The challenge was how to reduce the number of nodes in the graphs without losing useful information for classification. These pooling functions can occur between each aggregation phase, gradually reducing the graph's size, or at the end of the sequence of message passing functions to leverage a comprehensive understanding of the graph, enabling a more effective selection of important nodes. These pooling functions can be trainable or non-trainable and may have hyperparameters. Three pooling functions have been tested:

- **SortPooling** (M. Zhang *et al.*, 2018). It is used at the end of the several consecutive message passing functions. The idea behind the SortPooling is to sort nodes based on a certain criterion, select the top nodes, and then aggregate their features to create a representative graph-level feature vector. It is then fed into an MLP for the final graph classification. The hyperparameter *K* in this pooling function allow the selection of the *K*-top nodes. Following the procedure described in the paper by Zhang *et al.* (2018), the several outputs of the message passing functions are concatenated together, allowing the extraction of multi-scale substructure features. In the concatenated output, each row can be regarded as the "feature descriptor" (of size *m*) of a node, encoding its multi-scale local substructure information. The last message passing function maps the feature channels into 1 single feature channel, and this channel is used to perform the sorting.

  The nodes in the concatenated output are subsequently ranked based on the scores obtained from this final message passing layer. Only the nodes achieving the top *K* scores are retained. In doing so, a consistent ordering is enforced for the graph vertices, enabling the training of conventional neural networks on the sorted graph representations. When using this pooling layer, 1-D convolutions (sliding a filter over a one-dimensional sequence) are used before the MLP. The first 1-D convolution layer has a filter size and step of size *m*, in order to sequentially apply filters on node's feature descriptors. Subsequent to this, a sequence of MaxPooling layers and 1-D convolutional layers are added in order to learn local patterns on the node sequence. The workflow of the GNN with the SortPooling layer is presented in Figure 41.

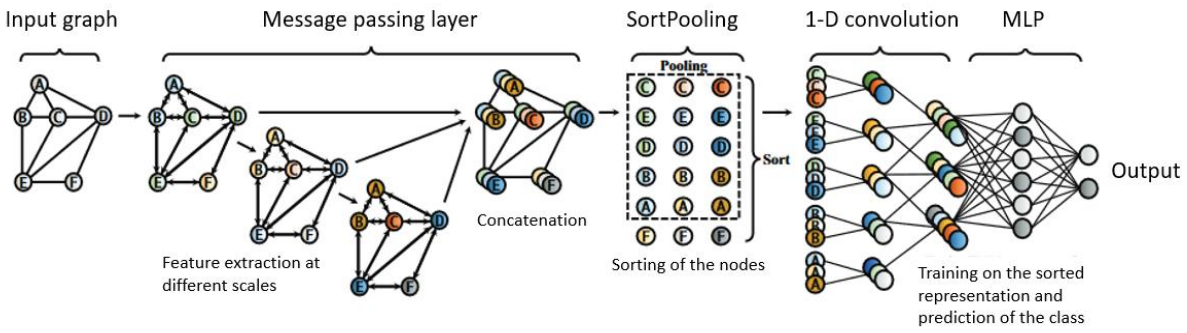

Figure 41. Workflow of the GNN with the SortPooling layer (Zhang et al., 2018).

- **DiffPool** (Ying *et al.*, 2018). It aims to learn hierarchical representations of graphs by iteratively coarsening the graph structure and features to reduce the computational complexity of GNNs while preserving important information. Unlike SortPooling, this pooling function is applied after each message passing function or stack of message passing functions in order to gradually

reduce the size of the graph. DiffPool performs graph pooling by selecting a subset of nodes to form a coarser representation of the graph. DiffPool employs a learnable mechanism to assign pooling scores to nodes. To achieve this, two GNN layers are defined with distinct roles (each composed of a stack of message passing functions and with l the layer number):

- $GNN_{l,embed}$ which, as defined above, is responsible for learning a new representation Z of each node through linear transformation, aggregation, and non-linear transformation. It can be written as in equation 7 (with $A^{(l-1)}$ and $X^{(l-1)}$ respectively the adjacency matrix and the feature matrix at layer $l-1$:

$$Z^{(l)} = GNN_{l,embed}\left(A^{(l-1)}, X^{(l-1)}\right) \tag{7}$$

- $GNN_{l,pool}$ which is responsible for calculating the pooling scores s for each node. These scores are represented in the matrix S, where at row *i* and column *j* is indicated the probability for node *i* to belong to cluster *j*. Its calculation at layer *l* is as written in equation 8:

$$S^{(l)} = softmax\left(GNN_{l,pool}\left(A^{(l-1)}, X^{(l-1)}\right)\right) \tag{8}$$

After calculating the new representation of each node $Z^{(l)}$ and the probabilities for each node to belong to each future cluster j contained in $S^{(l)}$, the new feature matrix $X^{(l)}$ (equation 9) and adjacency matrix $A^{(l)}$ (equation 10) accounting for the clustering are calculated in the following way:

$$X^{(l)} = S^{(l)T} Z^{(l)} \tag{9}$$
$$A^{(l)} = S^{(l)T} A^{(l-1)} S^{(l)} \tag{10}$$

The graph representations at different stages are not stacked, and after a certain number of iterations of message passing functions and poolings, the final graph representation at layer l, $X^{(l)}$, is passed to the layers responsible for the classification. $X^{(l)}$ is flattened and the MLP classifies it. The entire DiffPool process is differentiable, allowing it to be trained in an end-to-end manner with backpropagation. The pooling scores, coarsened graphs, and GNN parameters are all learned simultaneously through gradient-based optimization. The workflow of the GNN with DiffPool layer is presented in Figure 42.

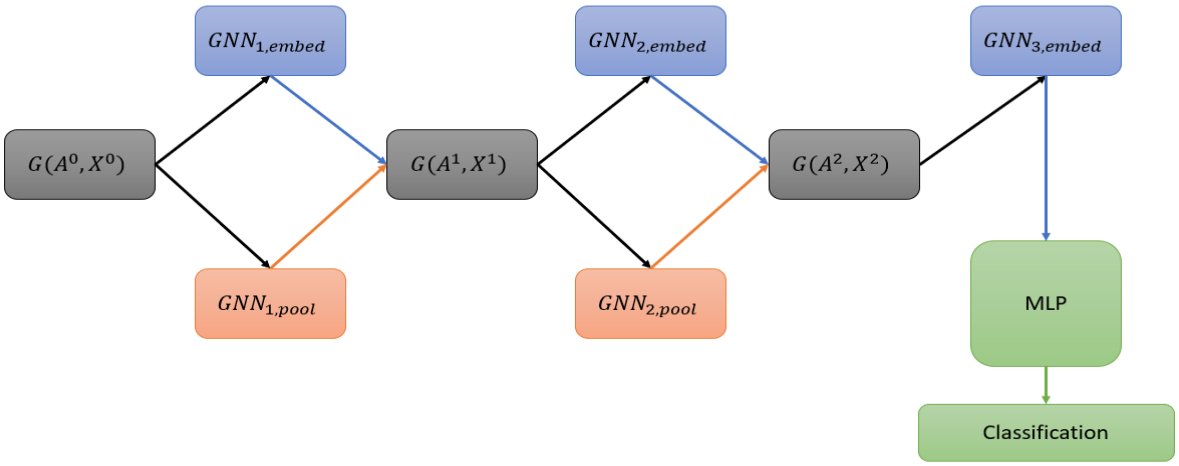

Figure 42. Workflow of the graph classification with 2 DiffPool layers. The input graph is $G(A^0, X^0)$ and the output is the classification of the graph.

- **MinCUT Pooling** (Bianchi *et al.*, 2020)**.** The principle of MinCUT Pooling is the same as for DiffPool, which is to form node clusters and aggregate nodes belonging to the same clusters in order to create a new graph representation. However, unlike DiffPool which employs a separate stack of MP function dedicated to determine pooling score ($GNN_{l,pool}$), MinCUT Pooling calculates these scores directly on the new node features (output of the $GNN_{l,embed}$). The goal of the partitioning is to create clusters of nodes that are densely connected within the clusters but have sparse connections between clusters. The aim is to exert greater control over node grouping by clustering together nodes that are interconnected while establishing clusters as distinct as possible. This is achieved by finding a cut in the graph that minimizes the number of edges between clusters while maximizing the number of intra-cluster edges.
  The learning of a new representation $Z^{(l)}$of the nodes is the same as presented in equation 7. However, the clustering is not performed through a second GNN, but by directly applying to Z an MLP with a Softmax on the output layer, as presenting in equation 11:

$$S^{(l)} = MLP\left(Z^{(l)}\right) \tag{11}$$

  Each row i of $S^{(l)}$ corresponds to the soft cluster assignment of the feature vector of node i contained in $Z^{(l)}$. The cluster assignment is computed from the node feature. However, due to the MLP function, clusters are likely to contain nodes that are both strongly connected and with similar features. Then, an auxiliary loss L_u is computed at each new cluster assignment to optimize this clustering (equation 12):

$$L_u = L_c + L_o$$

$$with\ L_c = -\frac{Tr\left(S^{(l)T} A^{(l-1)} S^{(l)}\right)}{Tr\left(S^{(l)T} D^{(l-1)} S^{(l)}\right)} \tag{12}$$

$$and\ L_o = \left\| \frac{S^{(l)T} S^{(l)}}{\left\| S^{(l)T} S^{(l)} \right\|_F} - \frac{I_k}{\sqrt{k}} \right\|_F$$

  where Tr is the trace of the matrix (the sum of its diagonal coefficients), $D^{(l-1)}$ the degree matrix of $A^{(l-1)}$, $\| \|_F$ the Frobenius norm (the square root of the sum of the squared values of all the elements in a matrix), *k* the number of clusters required at layer l and I_k the identy matrix of order k.
  The term L_c can be rewritten simply as in equation 13:

$$-\frac{number\ of\ links\ within\ clusters}{total\ number\ of\ links}, \text{ or}$$

$$-\frac{(total\ number\ of\ links) - (number\ of\ links\ among\ clusters)}{total\ number\ of\ links} \tag{13}$$

  Hence, the closer this term is to -1 (the minimum), the more the total number of links within the clusters equals the total number of links in the graph, indicating very few links between nodes of different clusters. Minimizing this term indeed aims to create the most distinct clusters possible. However, this minimum can be achieved for clusters that are not desirable. For example, one could consider a graph with 10 000 nodes all interconnected in a certain way, and 4 other nodes not connected to any others. Five node clusters are desired to be created. Minimizing the term Lc would lead to the 10 000 interconnected nodes being grouped in the same cluster, and the 4 isolated nodes each placed in one of the other 4 clusters. This does not

seem like a relevant way to cluster the graph. Another undesirable outcome of minimization would be to classify all the nodes of a graph into the same cluster. L_c is bounded by: $-1 \leq L_c \leq 0$.

To avoid this, the term L_o is added to encourage node clusters to be of similar size. $S^{(l)T}S^{(l)}$ is a diagonal matrix containing at row and column i the number of nodes grouped in cluster i. The matrix $\frac{S^{(l)T}S^{(l)}}{\|S^{(l)T}S^{(l)}\|_F}$ can be seen as the diagonal matrix of the number of nodes in each cluster relatively to the total number of nodes. The trace could have been used instead of the Frobenius norm, but this norm, squaring each element individually, takes better into account the individual variations of each element, and thus the differences in size between the clusters. Finally, I_k can be seen as the rescaled $S^{(l)T}S^{(l)}$ where the number of nodes is perfectly partitioned among the k clusters. In addition, if the number of nodes is perfectly partitioned among the k clusters, $\|S^{(l)T}S^{(l)}\|_F$ can be written as $\sqrt{k}.n$ with *n* the number of nodes in each cluster. The two matrices of L_o are then element-wise equal and the L_o is minimised. $0 < L_o \leq 2$ as the two matrices of L_o have unitary norm.

L_o is commensurable to L_c and the two terms can be safely summed without rescaling them. $L_u = L_c + L_o$ is bounded by $-1 < L_u \leq 2$ and minimizing it simultaneously aims to evenly distribute the nodes across each cluster while also striving to create groups of nodes well-distinct from each other. The rest of the algorithm follows the same steps as explained for DiffPool. The main advantage of MinCut Pooling is that it can lead to more controlled and interpretable pooling results.

C. Experiments

Similar to the investigation of CNNs for image-scale diagnosis (Chapter IV), several different experiments were examined. In fact, the same datasets as those used for CNNs study were used to allow for a comparison of the performance of these three methodologies. The same data splitting was also applied (5-fold stratified cross-validation with the same images in each fold as in Chapter IV Section B). The only difference was that a validation set didn't play a role in training the RF. As a result, the data that constituted the validation sets during the training of the CNNs or GNNs have been placed into the training sets of the RF. The testing sets remained the same for the 3 methodologies.

Initially, the 'CS20' dataset (Cabernet sauvignon, red variety) from 'ClaFD' was studied with the RF methodology because it contained the highest proportion of 'CONF+' images (22 %, as a percentage of the total number of images). The RF was trained and tested on this dataset with/without the 'CONF+' images, using the 4 feature vectors, in order to analyse performances and parameter importances in each scenario.

The study of the RF was also conducted on the 'UB20' dataset from 'ClaFD' to compare the results in parameter importances with those obtained on the 'CS20' dataset with images from a white grape variety (Ugni blanc, white variety). This dataset contains very few 'CONF+' images (4%).

Training and testing were performed separately on all red grape varieties (16% of 'CONF+ images) and all white grape varieties (3% of 'CONF+ images) within 'ClaFD' for both RF and GNN methodologies. The purpose of these tests was to compare the results of the two methodologies and to assess whether or not increasing the training dataset and the diversity of symptom expression improved performance or not for both methodologies.

Lastly, training including all images (both red and white grape varieties combined, 10% of 'CONF+' images) was tested to investigate whether any particularities in detections, distributions, or spatial arrangements of red and white grape varieties, if present, enhanced the RF and GNN methodologies performance when present in the same training database.

The different datasets studied, along with their total number of images, the number of images in the 'FD' class, and the number of images in the 'CONF+' class, are indicated in Table 35.

Table 35. Summary of the various datasets on which the RF methodology was trained and tested, along with a reminder of their respective total image counts, the number of images grapevines affected by FD, and the number of images displaying highly confounding symptoms.

Training and testing set	Total number of images	Number of 'FD' images	Number of 'CONF+' images
CS20 without 'CONF+'	204	72	0
CS20 with 'CONF+'	260	72	56
UB20	192	83	8
CS20 + CS21 + CF21 + M21	442	214	72
UB20 + UB21 + SB21	413	157	10
ClaFD	787	370	82

The same data augmentation was also performed for all experiments. It involved augmenting the raw images and then performing symptom detections on these new images. As a result, the symptom detections were different, and new vectors and graphs were created. This data augmentation was the same as for the CNNs methodology (Chapter IV Section B).

During each training of the RF, three of its hyperparameters were studied: i) The Number of Decision Trees (NDT, either 50, 100 or 500), ii) the Maximum number of Samples (MS, either 50%, 60% or 70% of the training set), iii) the Maximum number of Features (MF, either 1, 2 or 3).

For each of the algorithms, a range of MP functions between 0 and 4 were tested for the classification of the graphs, with 0 indicating that the graph is directly flattened and fed into the MLP. For each MP function, the c feature channels of each node feature were mapped into 16 feature channels by the linear transformation of the node matrix.

The SortPooling layer was integrated after the consecutive MP functions. The primary concept of the SortPooling involved selecting the top *k* nodes based on the score obtained after applying an additional MP function, which maps each node features into a single feature channel. These selected top *k* nodes were subsequently employed for the graph classification process. Tests were conducted using different values of *k*, specifically 10, 20, 50, and 100.

The DiffPool layer was also applied after each MP function. The value of *k* assigned to the final DiffPool layer (which represents the number of node clusters created by the pooling layer) was set to either 10 or 20. For each consecutive MP function and DiffPool layer leading up to this final DiffPool layer, the value of *k* was doubled. For instance, in the case of 3 stacks of MP function and DiffPool layer, *k* took respectively the values {40,20,10} or {80, 40, 20}. The same approach was adopted during the testing of the MinCut pooling layer.

D. Results

The initial results only concern those obtained through the RF methodology as it allowed visualizing the importance of each parameter in the information vector. Therefore, it allowed confirming or disproving the significance of considering symptoms other than foliar symptoms for the automated classification of FD. The results presented for the RF methodology are the best obtained for each feature vector. Therefore, the parameters of the Random Forest indicated in the tables are the ones that led to the best results.

The initial results presented in Table 36 are from the cross-validation conducted on the 'CS20' dataset without its 'CONF+' images.

Table 36. RF results for the 5-fold cross-validation on CS2O without 'CONF+' images for the FD study.

Feature vector	RF parameters (NDT, MF, MS)	FD precision	FD recall
VectYOLOv4leaves	(500, 1, 0.5)	**1**	**1**
VectYOLOv4all	(100, 3, 0.6)	0.93	1
VectYOLOv8leaves	All except (50, 1, 0.5) (100, 2, 0.6)	1	0.93
VectYOLOv8all	(100 & 500, 2 & 3, 0.5) (50 & 100 & 500, 3, 0.7)	1	0.93

NDT, MF and MS stand for the Number of Decision trees, the Maximum number of Features and the Maximum number of Samples respectively.

It can be observed that without the 'CONF+' images, the vector constructed solely with leaf information achieved the best results for leaves predicted with YOLOv4-tiny (outperforming YOLOv8m on the grape variety Cabernet sauvignon). Moreover, the results were perfect, with both precision and recall of 1. Furthermore, it can be observed that adding information about other symptoms (on shoots and bunches) besides the leaves decreased precision (p=0.93), leading to false positives in predicting images of vines affected by FD.

The analysis of parameter importance scores during one of the cross-validation training sessions using the 'VectYOLOv4all' vector was performed and is presented in Figure 43. This score indicates how much each feature contributes to the predictive capacity of the model and to the reduction of error. The higher the importance score, the greater the impact the feature has on the model's predictions. The calculation is based on how individual trees within the RF utilize each feature to conduct splits. Specifically, when trees make decisions at each node by dividing the data, the features that lead to the most informative and distinct splits contribute more to the overall prediction.

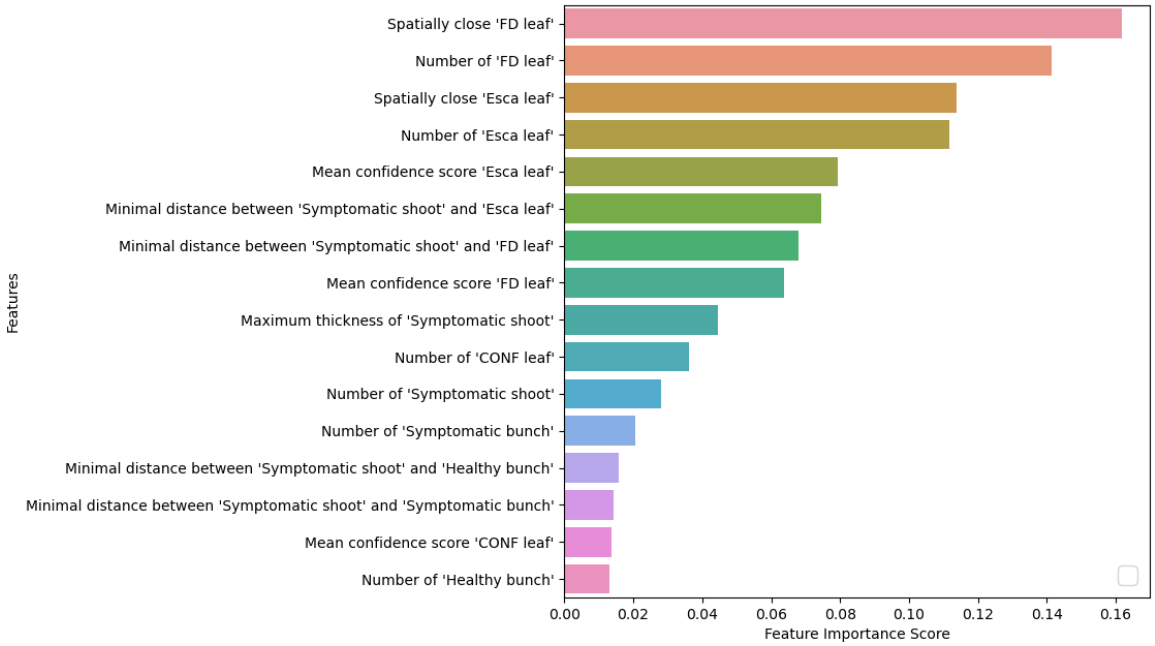

Figure 43. Feature importance scores of the training on CS20 without 'CONF+' with VectYOLOv4all for the FD study.

It can be observed in Figure 43 that even when training the algorithm with other symptoms apart from the leaves, the 5 most important criteria are solely related to the leaves. Specifically, the top 4 criteria that stand out from the rest are the number of elements and spatially close elements of the 'FD leaf' and 'Esca leaf' classes. When the dataset doesn't include 'CONF+' images, adding parameters related to other symptoms only confuses the algorithm, as demonstrated by the results presented in Table 37. These results were compared to those from the same dataset, CS20, but this time including the images from the 'CONF+' class. Results are presented in Table 37.

Table 37. RF results for the cross-validation on CS2O images, including CONF+ images, for the FD study.

Feature vector	RF parameters (NDT, MF, MS)	FD precision	FD recall
VectYOLOv4leaves	All	0.85	0.79
VectYOLOv4all	(500, 1 & 2 & 3, 0.5) (500, 3, 0.6 & 0.7) (100, 1, 0.6 & 0.7)	**0.87**	**0.93**
VectYOLOv8leaves	(50, 1 & 3, 0.5) (500, 3, 0.6) (500, 2, 0.7)	0.8	0.86
VectYOLOv8all	(50, 3, 0.7) (100, 3, 0.5) (500, 3, 0.6)	0.86	0.88

NDT, MF and MS stand for the Number of Decision trees, the Maximum number of Features and the Maximum number of Samples respectively.

Firstly, it can be observed in Table 37 that the addition of 'CONF+' images led to worse results for all the studied feature vectors. Moreover, this time, the vectors that took into account all the symptoms achieved better results (p=0.87, r=0.93) for the 'VectYOLOv4all'. The vectors created with the YOLOv4-tiny detection algorithm once again yielded superior results. By analysing the importance scores of features for the feature vector VectYOLOv4all, as proposed in Figure 44, the change in their scores became evident.

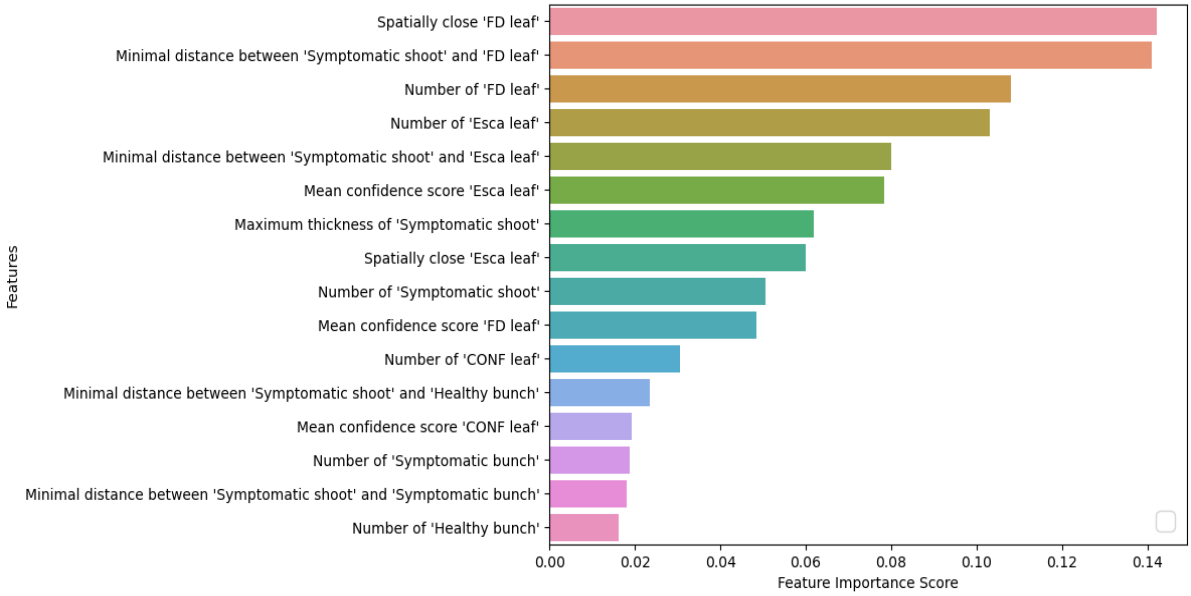

Figure 44. Feature importance scores of the training on CS20 with 'CONF+' with VectYOLOv4all

When looking at Figure 44, the most important criterion remained the number of 'FD leaf' spatially close. However, the second most important criterion became the minimum distance between a 'Symptomatic shoot' and an 'FD leaf,' with almost the same importance score. This criterion seemed crucial in distinguishing vines affected by FD from confounding diseases. Following this were the number of 'FD leaf' and 'Esca leaf,' which remained important criteria. In the fifth position, came the minimum distance between a 'Symptomatic shoot' and an 'Esca leaf.' Criteria related to grape bunches are scarcely considered by the model and can even lower the model's precision. Therefore, an attempt was made to remove these 4 parameters related to bunches from the VectYOLOv4all vectors, resulting in a precision and recall of (0.93, 0.93) for the 'FD' class.

The same tests were conducted on the 'UB20' dataset, which consisted of white grape varieties and contained very few 'CONF+' images. The results of these tests are presented in Table 38.

Table 38. RF results for the cross-validation on UB2O images for the FD study.

Feature vector	RF parameters (NDT, MF, MS)	FD precision	FD recall
VectYOLOv4leaves	All	**0.94**	**0.94**
VectYOLOv4all	All	0.89	0.94
VectYOLOv8leaves	(50 & 100, 1 & 2, 0.5) (50 & 100 & 500, 1, 0.7)	**0.94**	**0.94**
VectYOLOv8all	All except (50 & 100 & 500, 2, 0.5) (50 & 100 & 500, 3, 0.6)	**0.94**	**0.94**

NDT, MF and MS stand for the Number of Decision trees, the Maximum number of Features and the Maximum number of Samples respectively.

As shown in Table 38, the feature vectors considering only leaf parameters achieved the best results (p=0.94, r=0.94). Training with the 'VectYOLOv8all' vector also yielded these same excellent results (p=0.94, r=0.94). This further supported the notion that when studying a dataset with very few confounding diseases to FD, leaf information alone was sufficient. Adding parameters related to other symptoms contributed no additional value. It was also tested to remove the parameters related to bunches from the VectYOLOv8all vectors, but this did not result in an improvement of the results.

Table 39 shows the results of the RF on the entire set of red grape varieties and Table 40 those on the entire set of white grape varieties.

Table 39. RF results for the cross-validation on red grape datasets CS20 + CF21 + M21 + CS21 for the FD study.

Feature vector	RF parameters (NDT, MF, MS)	FD precision	FD recall
VectYOLOv4leaves	(50, 1, 0.6)	0.89	0.76
VectYOLOv4all	(500, 2, 0.6) (500, 3, 0.6 & 0.7)	0.89	0.76
VectYOLOv8leaves	(50, 3, 0.5)	0.86	0.88
VectYOLOv8all	(50, 2, 0.5)	**0.86**	**0.9**

NDT, MF and MS stand for the Number of Decision trees, the Maximum number of Features and the Maximum number of Samples respectively.

As shown in Table 39, the two vectors created with YOLOv8m produced better results (p=0.86, r=0.88) and (p=0.86, r=0.9), especially in terms of recall. Furthermore, adding information other than leaves led to a slight improvement in recall compared to the vector created solely with leaf information (r=0.88 vs r=0.9). This was different from the results obtained on CS20, where YOLOv4-tiny yielded the best results (p=0.87, r=0.93). This is explained by the fact that on the entire set of red grape varieties, YOLOv8m achieved better results for leaf detection. Both types of vectors created from YOLOv4-tiny achieved the same results (p=0.89, r=0.76). Once again, the parameters related to bunches ended up in the last positions during the study of their importance. However, their removal did not lead to better results this time.

Table 40. RF results for the cross-validation on white grape datasets UB21 + UB21 + SB21 for the FD study.

Feature vector	RF parameters (NDT, MF, MS)	FD precision	FD recall
VectYOLOv4leaves	(50 & 100 & 500, 1, 0.5) (50 & 100 & 500, 1, 0.7)	1	0.87
VectYOLOv4all	All	1	0.91
VectYOLOv8leaves	(50 &100 & 500, 2 & 3, 0.5) (50 & 500, 2 & 3, 0.7)	0.97	0.91
VectYOLOv8all	(100, 1, 0.6)	**1**	**0.94**

NDT, MF and MS stand for the Number of Decision trees, the Maximum number of Features and the Maximum number of Samples respectively.

The results shown in Table 40 for training and testing on all combined white grape varieties were very good ((p=1, r=0.91), (p=0.97, r=0.91), (p=1, r=0.94)). They were superior to those obtained for all red grape varieties ((r=0.86, p=0.9)), despite the fact that leaves were detected less accurately by the YOLOs. These better results for white varieties are explained by the fact that only 3% of the images in the dataset of combined white grape varieties are 'CONF+' images, compared to 16% for the combined red grape varieties dataset. However, it can be noted that considering the other symptoms further enhanced the results, unlike the results from the 'UB20' dataset alone.

Finally, RF and GNN were evaluated on all the images from the 'ClaFD' dataset, white and red grape varieties combined. The results are presented in Table 41 and 42 for the RF methodology.

Table 41. RF results for the cross-validation on ClaFD images for the FD study

Feature vector	RF parameters (NDT, MF, MS)	FD precision	FD recall
VectYOLOv4leaves	(100 & 500, 3, 0.5) (100 & 500, 1, 0.6) (50 & 100, 2, 0.6)	0.85	0.96
VectYOLOv4all	(100, 1, 0.5)	0.85	0.96
VectYOLOv8leaves	(100, 3, 0.7) (500, 2, 0.6)	**0.82**	**1**
VectYOLOv8all	(50 & 100, 2, 0.5)	**0.84**	**1**

NDT, MF and MS stand for the Number of Decision trees, the Maximum number of Features and the Maximum number of Samples respectively.

Results in Table 41 shown that once again, the 'VectYOLOv8all' vector yielded the best results, with a precision of 0.84, but most importantly, a recall of 1 (all images showing a grapevine affected

by FD were correctly classified as such by the algorithm). Mixing all the grape varieties had the effect of predicting all cases of FD, at the expense of precision (more false positives). To determine whether it was preferable to choose this model over those that separate red and white grape varieties, the results were tested on separate test sets for red and white grape varieties, and presented in Table 42.

Table 42. VectYOLOv8all best results for red and white separate in test for the FD study.

Test set	Feature vector, RF parameters (NDT, MF, MS)	FD precision	FD recall
ClaFD	VectYOLOv8all, (50, 2, 0.5)	0.84	1
CS20+CF21+M21+CS21	VectYOLOv8all, (50, 2, 0.5)	0.79	1
UB20+UB21+SB21	VectYOLOv8all, (50, 2, 0.5)	0.88	1

NDT, MF and MS stand for the Number of Decision trees, the Maximum number of Features and the Maximum number of Samples respectively.

When looking at the results shown in Table 42, for the white grape varieties, the recall is slightly better (0.94 when training exclusively with white grape varieties, Table 40), but the precision is lower by a larger proportion (previously 1). For white grape varieties, it was preferable to choose training exclusively with white grape varieties.

For red grape varieties, the recall was also significantly better (increased from 0.9 to 1, Table 41), and the precision was worse (decreased from 0.86 to 0.79). Both models were comparable, and one can choose the model based on whether the goal is to correctly detect all cases of FD at the expense of more false positives (training on combined white and red) or to have fewer false positives but potentially miss more cases of FD (training exclusively on red grape varieties).

Regarding the parameters of the Random Forest, their choices did not lead to a significant difference in the results. There was no specific triplet of parameters that stood out from the others; the choice of parameters did not seem to be critical. For example, for the cross-validation performed on the 'ClaFD' dataset using the YOLOv8all vector, the results for the 'FD' class are between 0.78 and 0.84 for average precision and between 0.96 and 1 for average recall. The results of all the hyperparameter combinations tested for this set and this type of feature vector are available in Appendix 6.

The importance scores of the parameters of the VectYOLOv8all vector for the training and testing that achieved the best results on the 'UB20', 'CS20 + CF21 + M21 + CS21', 'UB20 + UB21 + SB21' and 'ClaFD' sets are available in Appendix 7.

Table 43 presents the best results achieved using the GNN methodology for each tested pooling method and graph type for the 5-fold cross-validation on the ‘ClaFD’ data.

Table 43. Best results for each evaluated pooling method and graph type in 5-fold cross-validation on the 'ClaFD' dataset. Distance thresholds tested were either 100, 200, 400 or 800 pixels. 0 to 4 consecutive MP layer were tested. Tests were conducted using different values of k, specifically 10, 20, 50, and 100 for the SortPooling and 10 or 20 for the DiffPooling and MinCut pooling with 1 MP layer, {20, 10} or {40, 20} with 2 MP layers, {40, 20, 10} or {80, 40, 20} with 3 MP layers, {80, 40, 20, 10} or {160, 80, 40, 20} for 4 MP functions.

Pooling	Graph	Link distance (pixel)	Number of MP layer	K	FD precision	FD recall
Without	leafgraph	200	2		0.76	0.86
	allgraph_withoutconf	100	2		0.64	0.87
	allgraph	200	2		0.66	0.92
SortPooling	leafgraph	400	1	20	0.85	0.81
	allgraph_withoutconf	200	1	20	0.77	0.91
	allgraph	200	2	20	0.81	0.83
DiffPooling	leafgraph	200	1	10	0.83	0.81
	allgraph_withoutconf	100	2	[20, 10]	0.82	0.86
	allgraph	200	2	[20, 10]	0.84	0.87
MinCut Pooling	leafgraph	200	1	10	0.91	0.92
	allgraph_withoutconf	200	1	10	0.90	0.93
	allgraph	200	1	10	**0.93**	**0.93**

The first thing that can be observed when looking at results in Table 43 is that the best outcomes were consistently achieved for almost the same link distances and number of consecutive MP layer, regardless of the pooling method and graph type used. Smaller link distances (100 or 200) yield superior results for the detection of FD with a precision between 0.64 and 0.93 and a recall between 0.81 and 0.93. Only the SortPooling applied to the graph consisting solely of leaf detections delivered the best results for FD classification on 'ClaFD' with a link distance of 400 (p=0.85, r=0.81). When visually inspecting these distances (shown in Figure 38), it seemed that small distances were more conducive to clustering by clusters compared to larger distances where most symptoms, even when distant, were connected. As a consequence, during the passage through the MP layer, each node carried information about distant nodes and was less conducive to effective clustering. With smaller distances, clusters were more easily identifiable and interpretable. Supporting this notion, a low number (1 or 2) of MP layer iterations yielded the best results. This also reinforced the idea that node information did not need to be aggregated with information from distant nodes to accurately identify nodes to group together (for each MP layer, information was aggregated from nodes that already had aggregated information of their neighbours). This seems logical, as the created graphs were rather small in size: The graph with the largest number of nodes (the 'allgraph' type nodes) contained only 202 nodes, while the average number of nodes in graphs of the 'allgraph' type was only 76 nodes. For each node, the information did not need to be fetched very far from its neighbours to attribute it with an attribute vector characterizing its neighbourhood correctly, enabling it to be accurately classified within one of the node clusters. Finally, the parameter k, representing the number of nodes or groups

of nodes selected for the graph classification phase, yielded the best results for smaller values: either 20 for the pooling method selecting the top-k nodes (SortPooling), or 10 for the methods creating node clusters (DiffPooling, MinCut Pooling). As depicted in Figure 39, for each graph type, nodes can be effectively grouped into a small number of relevant clusters for classification purposes. This observation was consistent with the fact that, once again, the generated graphs had a relatively low number of nodes. However, when comparing the results obtained by each pooling method to the models without pooling method, it became evident that all results with pooling surpassed those attained without pooling. Hence, even though the graphs were relatively simple, these methods substantially contributed to enhanced graph classification outcomes.

Notably, SortPooling achieved its optimal results on the 'allgraph_withoutconf' type of graphs, yielding (0.77, 0.91) in terms of precision and recall for the FD class. The addition of confounding symptoms appeared to hinder the results of this pooling method, as fewer instances of actual FD in the vines were correctly identified by the model (recall drops from 0.91 to 0.83). It is also noteworthy that the model performed similarly when considering only leaf information ('leafgraph'). The inclusion of other symptoms did not significantly improve the results.

The pooling method (DiffPool) achieved similar results, but this time the discrepancy in results between graphs containing only leaves and those with other symptoms was more pronounced. Furthermore, the inclusion of confusing leaves slightly enhanced the results this time. One can suppose that the clustering methods better took into account the other symptoms, whereas for SortPooling, the 20 nodes selected for classification of the graphs likely contained leaf information.

Lastly, the finest results for all three graph types were achieved with the MinCut Pooling layer. The incorporation of auxiliary losses during clustering significantly enhanced classification results. The most impressive results were obtained for graphs containing all symptoms (p=0.93, r=0.93) in terms of precision and recall. The results for the other two graph types were quite comparable, suggesting that the information from symptom types other than leaves may be somewhat drowned out by the significance of leaf-related information. A test was conducted by removing the x and y coordinates from the information vector of each node for this combination of pooling function, graph, and hyperparameter, resulting in the best performance. The same results were obtained (p=0.93, r=0.93), suggesting that the model does not take into account these channel features during the node clustering process.

Results on the ‘allgraph’ were further categorized into results for the red grape test set and the white grape test set and presented in Table 44. Additionally, two separate training and testing instances were conducted with red and white grapes separated to assess whether it was more advantageous to separate the grape types or mix them using this symptom detection association method. The results are presented in Table 45

Table 44. Mincut Pooling results with 'allgraph' trained on ‘ClaFD’ and tested separately on red and white grape variety.

Test set	Pooling	Graph	Distance	MP layer	K	FD precision	FD recall
CS20+CS21+M21+CF21	MinCut	allgraph	200	1	10	0.9	0.96
UB20+UB21+SB21	MinCut	allgraph	200	1	10	0.97	0.89

The separated results for red and white grapes presented in Table 44 when training on ‘ClaFD’ were intriguing, featuring remarkably high recall for red grapes (0.96) coupled with a decrease in precision (from 0.93 to 0.9), while the opposite held true for white grapes (precision from 0.93 to 0.97, recall from 0.93 to 0.89). These differences may be due to variations in leaf detection for white and red grape varieties as discussed in Chapter V.

Table 45. MincutPooling with 'allgraph' trained separately on red and white grape variety

Train and test set	Pooling	Graph	Distance	MP layer	K	FD precision	FD recall
CS20+CS21+M21+CF21	MinCut	allgraph	200	1	10	0.85	0.89
UB20+UB21+SB21	MinCut	allgraph	200	1	10	0.87	0.84

The results of separated training between red and white grapes presented in Table 45 yielded inferior results compared to when they were mixed: (p=0.9, r=0.96) and (p=0.97, 0.89) compared to (p=0.85, 0.89) and (p=0.87, r=0.84) for separated training on red and white grape variety respectively. This contrasts with the results obtained thus far for other methodologies, where better results were found when separating red and white grape varieties. However, the hypothesis was that in this case, the clustering of nodes (symptom clusters) compensated for the distribution and detection differences among grape varieties. Moreover, the amalgamation of the two colours significantly augmented the volume of training data for the model, a crucial parameter to consider in deep learning algorithms.

E. Synthesis

A summary of the best results on the red and white grape variety obtained by the CNN methodologies (Chapter IV Section C) and symptom detection and segmentation association using the RF approach and the GNN approach is provided in Table 46. These results are those of the 5-fold cross-validation with the same images in each folder for the 3 methodologies. Standard deviations (std) have been added to the results of precision and recall averages obtained in cross-validation. It indicates how much the results obtained for each test fold are dispersed around the mean value shown in the Table 46.

Results in Table 46 are showing that concerning the red grape varieties, best results are achieved for the GNN methodology with (p=0.90, r=0.96). The RF method yielded slightly inferior results (p=0.86, r=0.90). The less favourable results are delivered by the CNN methodology with (p=0.87, r=0.84). These results contrast with those obtained for white grape varieties, for which the best results are achieved with the CNN methodology (p=0.97, r=0.96) and with the RF methodology (p=1, r=0.94). This reinforces the hypothesis that for datasets without or with very few confounding diseases, as is the case for white grapes (3% of images in the 'CONF+' class), CNNs yield excellent results. However, as soon as the number of 'CONF+' images is higher, as in the case of the red grape dataset (16%), CNN results were less favourable. The two-step methodologies, with a first step dedicated to symptom prediction and a second step for associating these detections, appeared to better differentiate FD from its confounding diseases. The standard deviations range from 0 to 0.05 without any particular method obtaining significantly more variable results than others during the different tests of the 5-fold cross-validation. The results of the 3 methodologies remained stable from one fold to another during their 5-fold cross-validation, and the differences in results between the methodologies, with regards to the standard deviations, can be considered significant.

The hypothesis was that GNN had the potential to outperform the RF method, given its capacity to accommodate extensive symptom detection information without input size limitations. However, it was found that GNN results were not outperforming the RF results. Several explanations are suggested:

- The chosen parameters for constructing the RF input vector aptly encapsulated the grapevine's physiological state, rendering additional information redundant for diagnostics.
- The average symptom detection results capped the potential image-scale classification results for both methods.
- The created graphs might lack a sufficient number of nodes to fully leverage the power of the graph methodology and the cluster creation of pooling methods.

Table 46. Comparison of the most relevant results of a 5-fold cross-validation obtained using the CNN methodology, the RF methodology and the GNN methodology, on red and white varieties. VectYOLOv8all stands for the vector with 16 parameters including features about all the symtpoms and leaves detected with YOLOv8m. 'allgraph' stands for the graph created with all the symptoms (including the confounding leaves) with node feature vectors of 15 parameters. MP stands for Message Passing function. K stands for the number of clusters created by the pooling function. Std stands for the standard deviation.

Training and Testing set	% of 'CONF+' images	Method	Parameters	FD precision (std)	FD recall (std)
Red grape varieties	16%	CNN	EfficientNetB5, /64 of the resolution	0.87 (0.01)	0.84 (0.03)
		RF	VectYOLOv8all, NDT=50, MF=2, MS= 0.5	0.86 (0.02)	0.90 (0.04)
		GNN	'allgraph', link distance = 200, 1 MP layer, MinCut pooling with k=10	**0.9 (0.03)**	**0.96 (0.03)**
White grape varieties	3%	CNN	MobileNetV3-large, /64 of the resolution	**0.97 (0.04)**	**0.96 (0.02)**
		RF	VectYOLOv8all, NDT=100, MF=1, MS= 0.6	**1 (0)**	**0.94 (0.05)**
		GNN	'allgraph', link distance = 200, 1 MP layer, MinCut pooling with k=10	0.97 (0.04)	0.90 (0.02)

NDT, MF and MS stand for the Number of Decision trees, the Maximum number of Features and the Maximum number of Samples respectively.

To explore this last possibility, a comparison of these methods at the scale of several neighbouring grapevines has been performed, allowing for the creation of more complex graphs taking advantage of the redundancy of information on neighbouring grapevines. These methodologies were developed and tested thanks to the acquisitions made at the whole-vineyard scale. This study is discussed in Chapter VII.

VII. Diagnosis methodologies applied at the block scale

Table of content

The objective was to evaluate and compare the CNN methodology detailed in Chapter IV and the symptom association methodologies detailed in Chapter VI, not on images acquired sporadically from blocks, but on images depicting the entirety of a blocks to concretely assess these methodologies in real world use scenarios.

A. Algorithms

The CNN, RF and GNN models trained and obtaining the best results on the 'ClaFD' set (see Table 46) have been tested on the images of these 3 blocks. For each image acquired on these blocks, the unitary symptoms were predicted by the symptom detection algorithms (see Chapter V) and these detections were associated, either via the vector 'VectYOLOv8all' for the RF methodology, or by creating 'allgraph' for the GNN methodology.

In addition, new ways of assembling the detections have been tested in order to make the most of these continuous acquisitions at block scale.

a. Associating the symptoms detected on 3 neighbouring grapevines

The image triplets from the '3-neighbouring grapevines' dataset (Chapter III Section E) were used. During the classification of each image, it took into account the symptoms detected in the images acquired just before/after the targeted image.

i. Random Forest based methodology

The principle of the RF considering 3 neighbouring grapevines was the same as for the RF with a single image, except that the symptoms considered for vector creation were those detected in the 3 images. The same features as those in the 'VectYOLOv8all' vector were then calculated in the same manner as described in Chapter VI Section A, but this time by considering the predictions from these 3 images. These new information vectors were subsequently fed into the RF for the classification step.

ii. GNN-based methodology

Taking into consideration the symptoms detected in 3 neighbouring images allowed the creation of a large graph that aggregated all these detections. Once the alignment direction was determined, the coordinates of the nodes in each graph were updated to match the coordinates relative to the 3 images placed side by side. This allowed connecting the nodes of the 3 graphs below the defined distance threshold, illustrated in Figure 45.

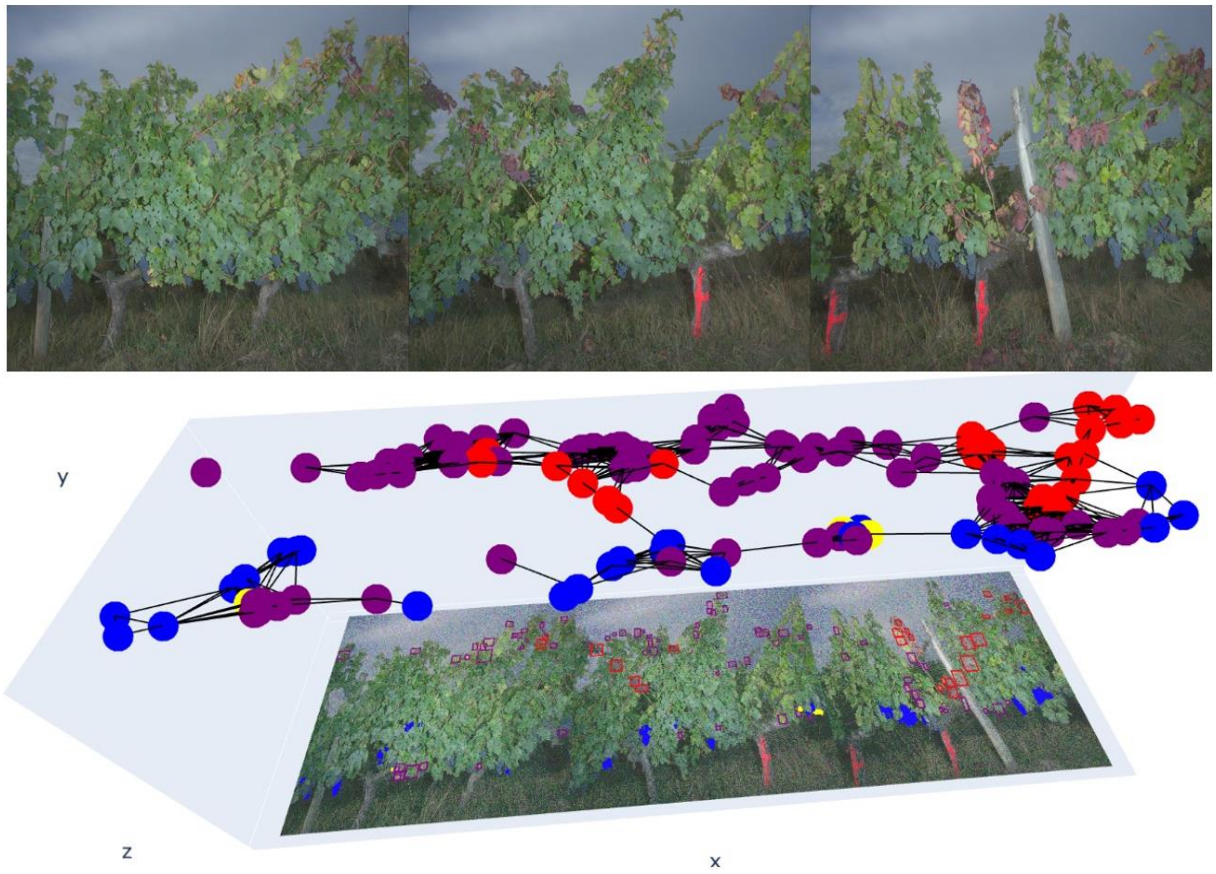

Figure 45. Example of the reconstruction of 3 consecutive images and visualization of the detected symptoms, along with the graph that summarizes the detected symptoms on the 3 images.

On Figure 45, it can be observed that there was some overlap between the images. However, this overlap was not consistent. The images captured at the block scale were taken with cameras fixed on a quad, making it challenging to maintain a constant speed. Due to this variability, it was decided not to remove this overlap between the images when creating the graphs with 3 neighbours. Furthermore, this overlap could be beneficial. If the same symptom was present in two consecutive images and was predicted to belong to the same class on both images, the confidence in this detection could be higher. Conversely, if it was classified into two different classes, this could help identify potential false positives or false negatives more effectively.

b. Associating the symptoms detected on 6 neighbouring grapevines

The image sextuplets from the '6-neighbouring grapevines' dataset (Chapter III Section E) were used. The symptoms detected in the images of the preceding and following grapevines of the targeted one, but also the symptoms detected on the opposite side of these 3 grapevines were considered for the

diagnosis. This methodology enabled the correlation of symptoms detected on each side and, therefore, held the promise of a more precise disease diagnosis.

i. Random Forest based methodology

The same features as those in the 'VectYOLOv8all' vector were then calculated in the same manner as described in Chapter VI, Section A, but this time by considering the symptom predictions from these 6 images.

ii. GNN-based methodology

For the graph-based methodology, two graphs were created from the symptom detections of the three images of each side. Then, the nodes of the two graphs were linked: a third dimension, which would be that of depth, is not taken into account, and the two graphs are superimposed as if they were derived from the same side of the vine. Their nodes were linked to each other if their distances were below the set threshold (Figure 46).

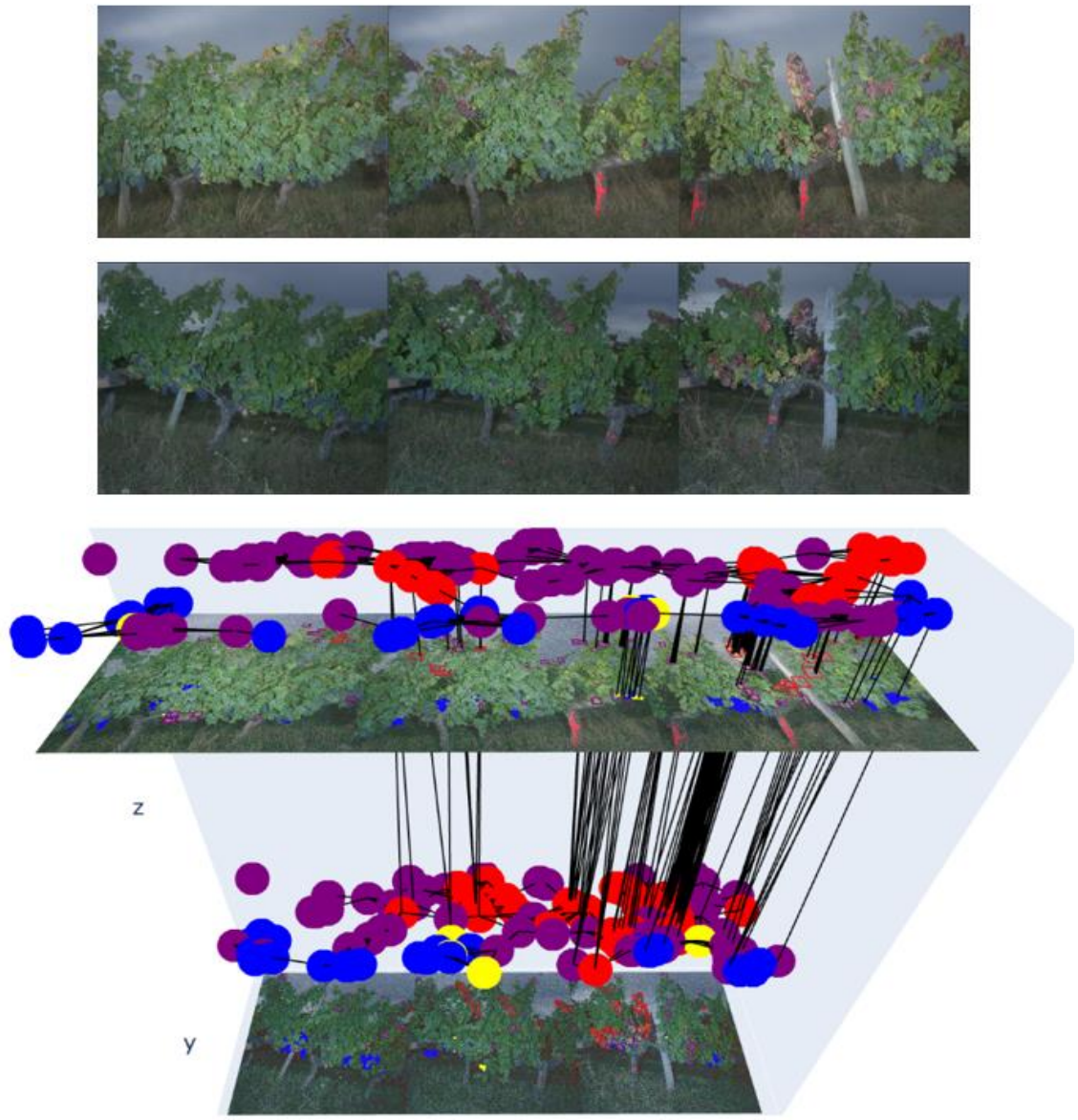

Figure 46. Representation of a 6-neighbours graph. The images in the first row represent the reconstruction into a single large image of one side. The second row represents the images acquired

from the other side, arranged in the correct order and flipped so that the detected symptoms on each side correspond. The graph created by combining these 6 images is depicted below. This illustration was created to provide a better understanding of what a graph representing the detections of the 6 neighbouring images looked like. The visible difference in link size is solely due to a scaling difference between the axes to enhance the figure's representation. The z axis is only present for a better understanding, the 2 graphs are positioned on the same plane when combining them.

B. Algorithm hyperparameters

The data vectors generated for the RF 3 and 6 neighbours methodologies are of the same size and share the same parameters as those used for the RF methodology. It was observed that hyperparameters such as the Number of Decision Trees (NDT), Maximum Number of Features (MF), and Maximum Number of Samples (MS) did not have a significant influence on the final results of the study (Chapter VI Section D). Following this observation, the decision was made to set these specific hyperparameters (NDT=50, MF=2, and MS=0.5) for the RF 3 and 6 neighbours methodologies. These values demonstrated optimal performance during the analysis of red grape varieties (as indicated in Chapter VI Section E, Table 46). It is important to note that two out of the three study vineyards are planted with red grape varieties, further justifying the choice of these hyperparameters for these specific methodologies.

Regarding the GNN methods, the decision was made to test only the method using MinCut pooling, as this method achieved by far the best results when classifying the graphs in the 'ClaFD' dataset (Chapter VI Section D, Table 43). The graphs created by the GNN 3 and 6 neighbours methods, considering the detections on 3 or 6 images, have more nodes and edges than those created from the images in the 'ClaFD' dataset, considering detections on a single image. Consequently, the number of Message Passing functions (MP functions) and successive MinCut pooling function were re-evaluated to find the optimal hyperparameters. The parameter K_l, representing the number of clusters of nodes at pooling step l, was tested with values in {10, 20} for $l = 1$, {20, 40} for $l = 2$, {40, 80} for $l = 3$, and {80, 160} for $l = 4$. Since the graphs are more complex, it was also tested to apply multiple consecutive MP layers before applying the MinCut pooling layer. Up to 4 consecutive MP functions were tested between each MinCut pooling layer.

C. Datasets and experiments

Experiment 1

The CNN, RF and GNN methodologies at the scale of a single grapevine, developed in Chapter IV and Chapter VI Section A and B respectively, have been tested on the three study plots. Thus, the training sets that achieved the best results for red grape varieties and white grape varieties of 'ClaFD' in the CNN methodology were used to predict all the images acquired in these three vineyards. The training sets that achieved the best results for red and white grape varieties of 'ClaFD' in the RF methodology were used to predict all the images acquired in these three vineyards. Similarly, for the GNN methodology, the training sets that achieved the best results for red and white grape varieties was tested on the images of these plots.

Experiment 2

The RF and GNN methodologies at the scale of 3 and 6 neighbouring grapevines were studied. It was not possible to directly test these methodologies on these plots, as was the case for the methodologies at the scale of a single grapevine, because these three plots constituted the only image dataset

containing images of neighbouring vines. Images other than these were not available to train the models beforehand. In order to validate the accuracy of these methodologies, they were tested through cross-validation on each of the three vineyards. The vineyards were divided into three distinct zones, as illustrated in Figure 47. For each of the three vineyards, the RF methodologies with 3 neighbours and 6 neighbours were trained on two parts of these plots and tested on the remaining part. For the GNN methodologies with 3 and 6 neighbouring grapevines, which require a validation set, the training was conducted on one part, validated on another, and tested on the remaining part. The validation set contained a significant amount of data because it was important to have a validation set that included a sufficient number of cases of vines affected by FD. As illustrated in Figure 47, some plots contain very few such cases.

For the cross-validation of the 'Cognac' and 'Neuffons' vineyards, the number of vines affected by FD for training was very low (16 for 'Cognac' and 61 for 'Neuffons' vineyards). The images showing these vines placed in the training set underwent three independent data augmentations (the same as presented Chapter IV Section B for the CNN study), resulting in three different images, vectors and graphs for the same image. Additionally, the number of healthy grapevine images was randomly reduced to achieve a training population consisting of 20% of vectors or graphs from FD-affected vines and 80% from healthy vines. This proportion was chosen to ensure that there were enough elements for the models to train properly, avoiding overfitting. Loss functions to deal with this class unbalance were not tested because the models did not show a tendency to classify all elements as healthy during training.

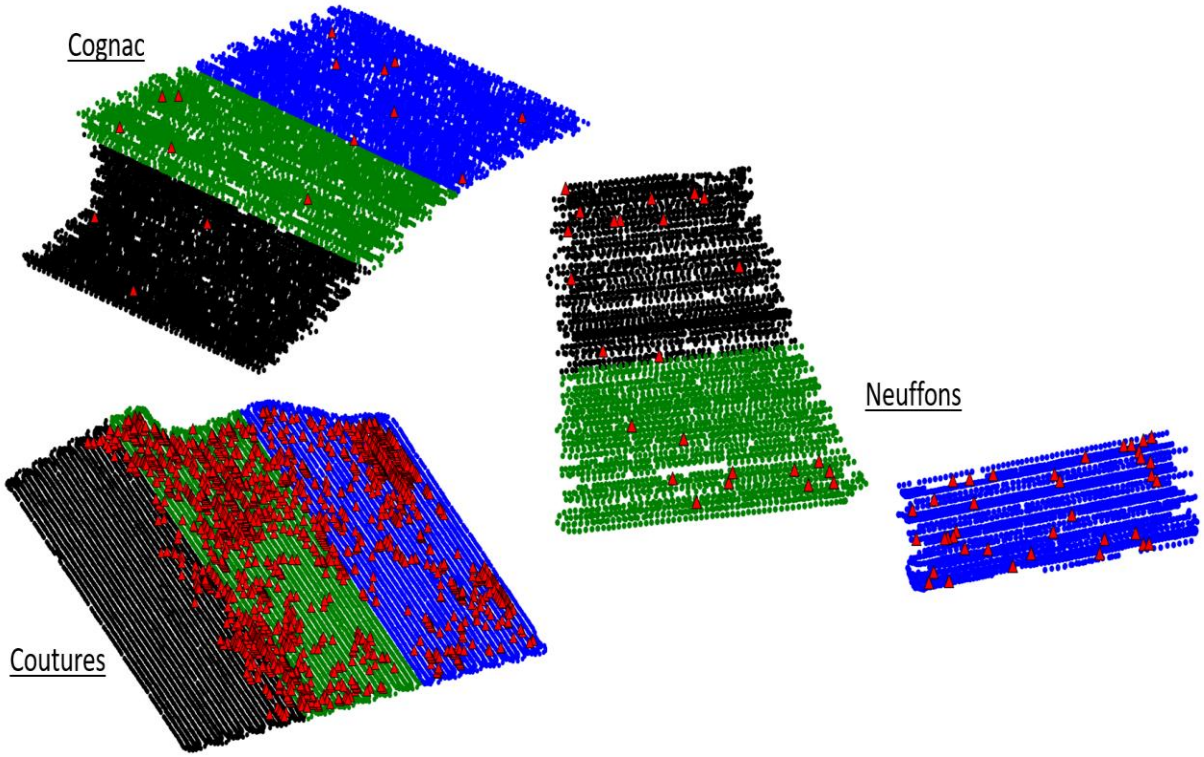

Figure 47. The study vineyards were divided into 3 parts for the study of the RF and GNN methodologies: one grapevine, 3 neighbouring vines, and 6 neighbouring vines. Points of the same colour represent the coordinates of images that were placed in the same folder during cross-validation. The red triangles correspond to the coordinates of the vines affected by FD as recorded by the experts.

Furthermore, the CNN, RF and GNN methodologies at the scale of a single grapevine were also trained and tested in the same manner. The results obtained allowed for a comparison of these seven methodologies among themselves, using the same training and testing dataset, and identified the most promising one for vineyard-scale diagnostic.

Experiment 3

The methodology that achieved the best results in intra-vineyard cross-validation was trained on one or two vineyards and tested on the other(s) in order to validate these methodologies in a real-world scenario where the algorithm had never been exposed to the images of the studied vineyard.

It is important to note that for the training and testing of all these methodologies, the symptoms were predicted in the same way: using the two algorithms YOLOv8-m and ResUNet presented in Chapter V and trained on the DetFD and SegFD datasets, respectively.

A summary of the tests conducted on these three vineyards is available in Table 47.

Table 47. Summary of the tests conducted at the vineyard level for the CNN, RF, RF 3-neighbours, RF-6 neighbours and GNN, GNN 3-neighbours, GNN 6-neighbours methodologies.

Experiment	Methodology	Train	Test	Purpose
1	CNN	CS20+CF21+M21+CS21, UB20+UB21+SB21	BloFD	Evaluate the training conducted on 'ClaFD' across the three vineyards
	RF	CS20+CF21+M21+CS21, UB20+UB21+SB21	BloFD	
	GNN	ClaFD	BloFD	
2	CNN	BloFD	BloFD	Cross-validation on each vineyard to determine the most suitable method for the diagnosis of FD at the vineyard scale.
	RF, RF 3-neighbours, RF-6 neighbours.	BloFD	BloFD	
	GNN, GNN 3-neighbours, GNN 6-neighbours	BloFD	BloFD	
3	GNN 6-neighbours	BloFD	BloFD	Training on two vineyards and testing on the remaining vineyard to evaluate these methodologies in a real-world application scenario.

Examples illustrating the specificities of each vineyard are provided in Figure 48. Grapevines affected by FD at the 'Neuffons' vineyard show very few symptoms. Many grapevines at the 'Cognac' vineyard exhibit symptoms very similar to those of FD without being affected by it. The 'Couture' vineyard contains a large number of grapevines affected by FD and a few confusing diseases.

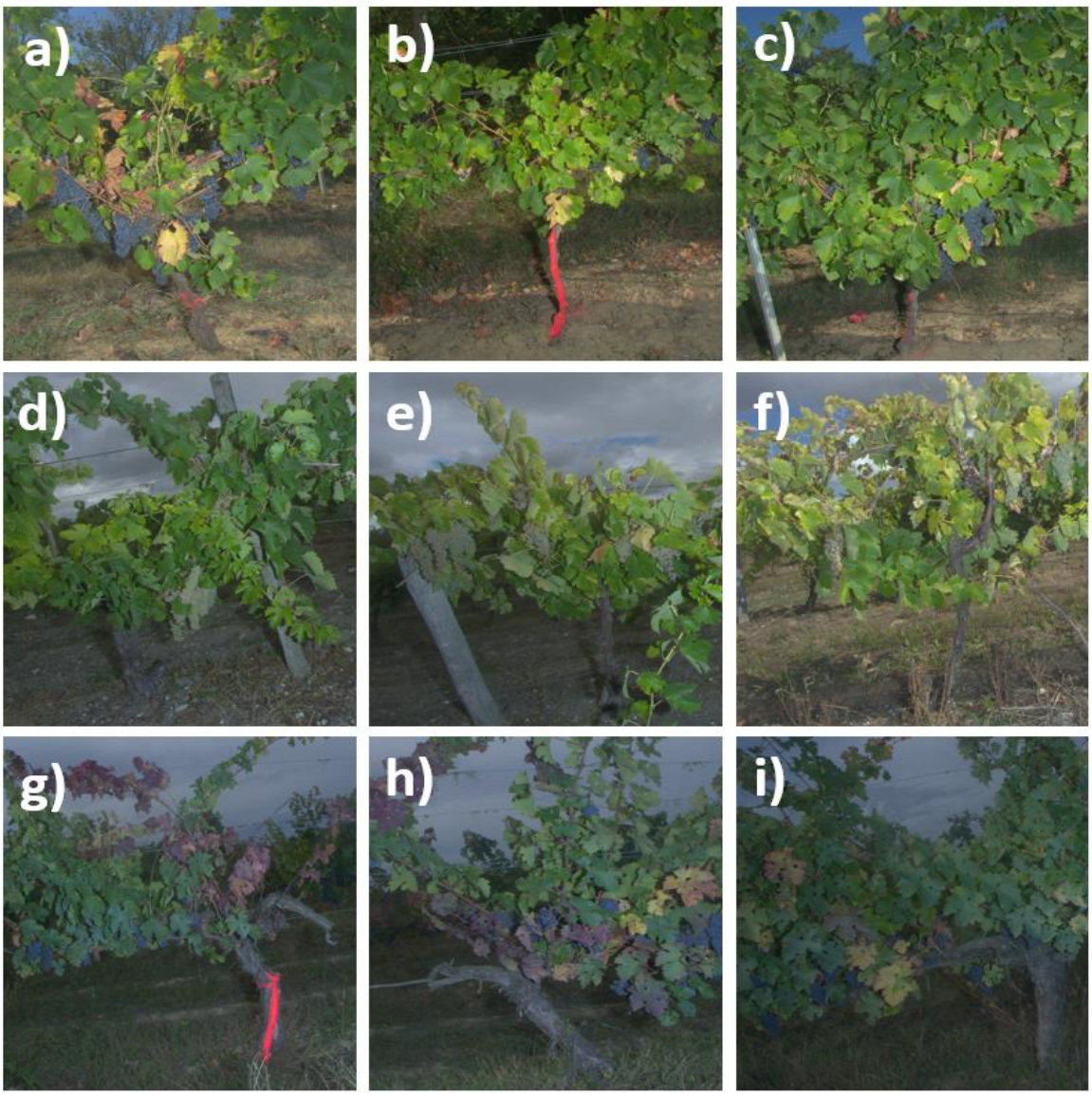

Figure 48. Grapevines affected by FD and confounding diseases in the 3 study blocks. First row: 3 grapevines affected by FD of the 'Neuffons' vineyard. The first one (a)) present few symptoms, while the other two (b) and c)) did not appear to present any symptoms of FD. Second row: Grapevines of the 'Cognac' vineyard. d) Grapevine affected by FD; e) and f) Grapevine non-affected by FD but displaying symptoms similar to those of FD. Third row: grapevine of the 'Couture' vineyard. Grapevine affected by FD (g)) and 2 grapevines showing symptoms similar but without being affected by FD ((h) and i)).

D. Results

Experiment 1

The results of tests on the 3 vineyards (Neuffons, Cognac, Couture) of the methodologies RF and GNN trained on the 'ClaFD' dataset images are presented in Table 48.

Table 48. Experiment 1: results for the class 'FD' obtained with the CNN, RF and GNN methodologies trained on the 'ClaFD' dataset images and tested on the vineyards of the 'BloFD' dataset.

Vineyard	Grapevine affected by FD	Methodology	Grapevine correctly predicted as infected by FD	FD precision	FD recall
Neuffons	61	CNN	51	0.04	0.84
		RF	6	0.3	0.1
		GNN	19	**0.25**	**0.26**
Cognac	16	CNN	15	0.00	0.94
		RF	14	0.00	0.88
		GNN	16	**0.00**	**1**
Couture	1764	CNN	1692	0.13	0.99
		RF	258	0.89	0.15
		GNN	384	**0.96**	**0.22**

The results presented in Table 48 are rather disappointing. For all three vineyards, the GNN method yields the best results, but these results are quite poor. For the 'Neuffons' vineyard, the GNN method achieved (p=0.25, r=0.26) for the 'FD' class, compared to (p=0.3, r=0.1) for the RF method and (p=0.04, r=0.84) for the CNN method. However, the three methods failed to correctly diagnose this vineyard, where the symptoms of FD are very subtle (as illustrated in Figure 49 where the symptom predictions are displayed for two grapevines of the 'Neuffons' vineyard affected by FD). The first reason could be that the symptom detection algorithm is missing a lot of symptomatic leaves of FD. This could be also be because the 'ClaFD' database on which the methodologies were trained predominantly contained vines with highly pronounced symptoms of FD. As a result, the association methods struggle to accurately diagnose vines affected by FD with less visible symptoms. The CNN method correctly predicted most of the grapevine affected by FD (r=0.84) but made a significant number of false predictions of grapevine affected by FD (p=0.04).

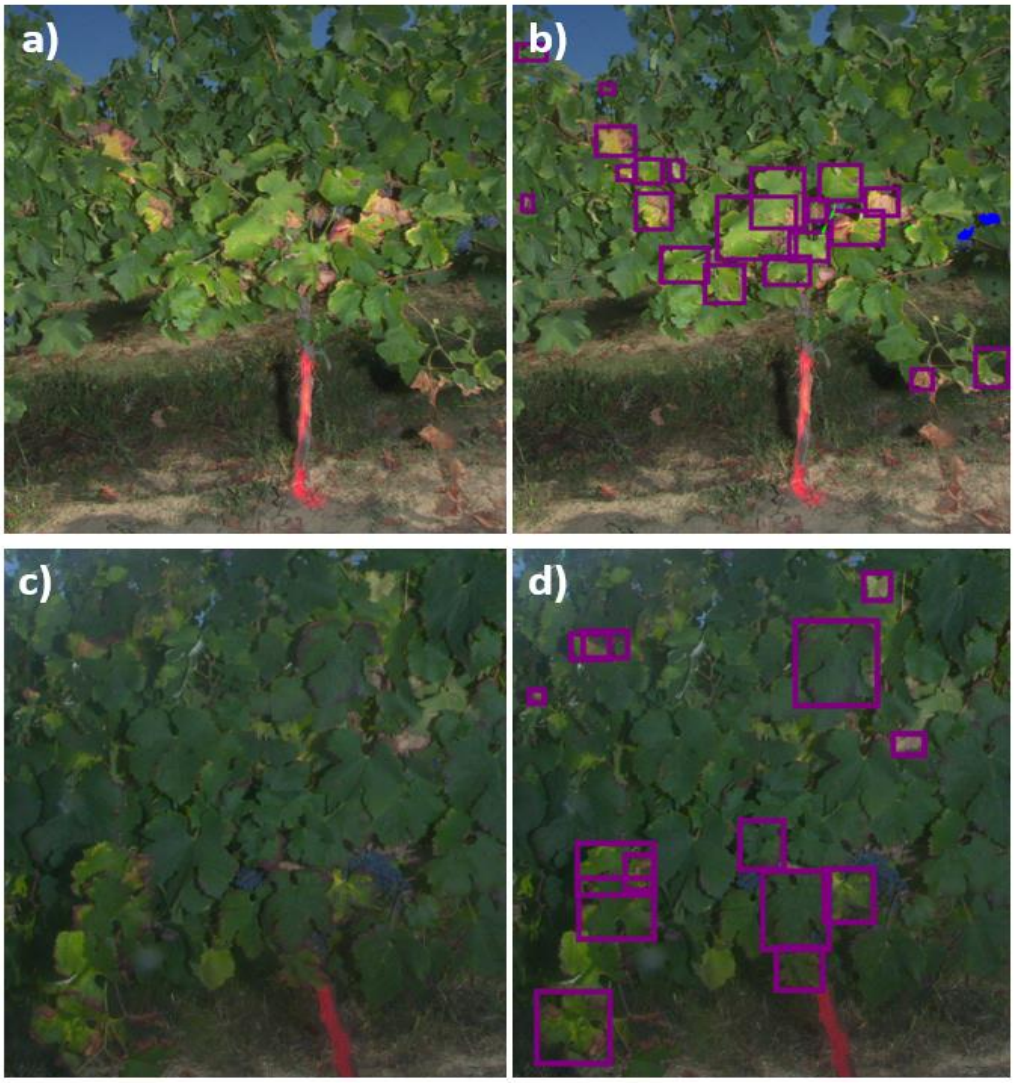

Figure 49. Image and symptom detections on two images acquired on the 'Neuffons' vineyard. Images a) and c) show grapevines affected by FD. The symptoms are very faintly expressed in these two vines. On images b) and c), where symptom predictions have been overlaid onto the images, it can be observed that all symptomatic FD leaves have been predicted as confounding leaves by the YOLOv8m algorithm (purple bounding boxes).

Regarding the 'Cognac' vineyard, the RF method correctly predicted 14 out of 16 vines affected by FD, while the CNN method correctly predicted 15 and the GNN method correctly predicted all 16 affected vines. Unfortunately, the three methods also predicted a significant number of non-FD vines as FD-affected. For results, the three methods reached a precision close to 0. This is due to many vines at the 'Cognac' vineyard suffering from grapevine yellowing. The leaves of these vines had turned yellow and were often misidentified as 'FD leaves' by the leaf detection algorithm, as it is illustrated in Figure 50. Their prevalence in large numbers caused both methods to struggle, even when combined with other symptoms (symptomatic shoots and bunches, confounding leaves), to correctly predict these images of vines affected by yellowing as non-FD. The 'ClaFD' dataset on which both methods were trained may not contain enough images of this type (false detections of FD leaves in large numbers when the grapevine is not affected by FD) to accurately diagnose FD on this vineyard.

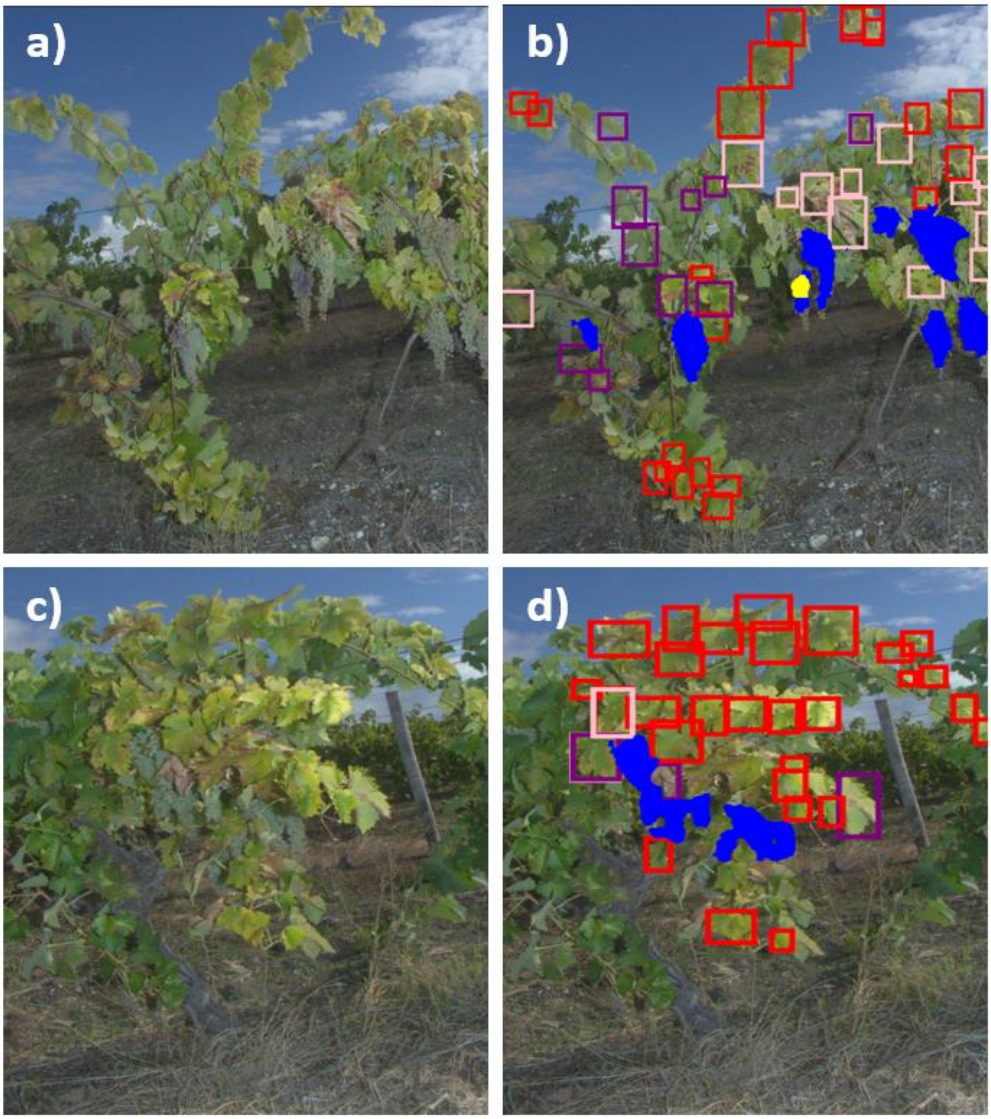

Figure 50. Image and symptom detections on two images acquired on the 'Cognac' vineyard. Images a) and c) are not affected by FD and show yellowing leaves very similar to those symptomatic of FD. The symptoms detected on these images (b) and d)) show that most of these yellowing leaves are predicted as FD leaves by the YOLOv8m algorithm (red bounding boxes). The purple bounding boxes correspond to leaves of the 'Confounding leaf' class, and the pink ones belong to the 'Esca leaf' class. The blue pixels correspond to healthy bunches pixels, and the yellow ones represent symptomatic bunches.

Finally, for the 'Couture' vineyard, both symptom associating approaches achieved high precision in predicting vines affected by FD (0.89 for the RF approach, 0.96 for the GNN approach) but had relatively low recall (0.15 and 0.22 for the RF and GNN approaches, respectively). This vineyard contains a very high number of vines affected by FD (1764), and both approaches struggle to identify most of them. This is likely because, once again, the detected symptoms and their spatial arrangement are not present in the 'ClaFD' dataset. However, on this vineyard, vines displaying symptoms very similar to those of FD are fairly accurately predicted (very few false positives for both methods) in contrast to the results obtained on the 'Cognac' vineyard. The hypothesis is that foliar symptoms are overall better predicted for red grape varieties (‘Couture’ is planted with the Cabernet sauvignon red variety) than for white grape varieties (‘Cognac is planted with the Ugni blanc white variety). Indeed, as presented in Chapter V, Section A, Tables 27 and 28, leaf symptom detection results in the best outcomes (p=0.78, r=0.61) for red grape varieties compared to (p=0.48, r=0.69) for white grape varieties. The CNN approach, on the other hand, allowed for the detection of almost all the vines

affected by FD (1692). However, as with the other two vineyards, this method delivered a significant number of false predictions (p=0.13).

Experiment 2

The CNN, RF and GNN methodologies have been trained and tested independently in cross-validation on each vineyard to determine if learning and testing on the same vineyard resulted in good FD prediction results. These three methodologies were compared to the RF 3 and 6 neighbours and GNN 3 and 6 neighbours methodologies. It should be noted that the symptoms are always predicted using algorithms trained on the 'SegFD' and 'DetFD' datasets. The results of this study are presented in Table 49.

Table 49. Experiment 2: cross-validation results for the class 'FD' obtained using the CNN, RF, RF 3-neighbours, RF 6-neighbours, GNN, GNN 3-neighbours, and GNN 6-neighbours methodologies on the 3 vineyards of the 'BloFD' dataset. The method and results in bold are those that achieved the best results when combining precision and recall. The results for the GNN 3-neighbours were obtained using 2 consecutive layers of MP functions followed by a single layer of MinCut pooling with k=10. For the GNN 6-neighbours, the results were obtained using 4 consecutive layers of MP functions followed by a single layer of MinCut pooling with k=10.

Vineyard	Grapevine affected by FD	Methodology	Grapevine correctly predicted as infected by FD	FD precision	FD recall
Neuffons	61	CNN	52	0.17	0.86
		RF	13	0.87	0.21
		RF 3-neighbours	30	0.16	0.56
		RF 6-neighbours	24	0.37	0.52
		GNN	32	0.43	0.65
		GNN 3-neighbours	21	0.84	0.26
		GNN 6-neighbours	**42**	**0.76**	**0.71**
Cognac	16	CNN	9	0.01	0.63
		RF	9	0.19	0.49
		RF 3-neighbours	7	0.07	0.46
		RF 6-neighbours	8	0.16	0.45
		GNN	10	0.27	0.63
		GNN 3-neighbours	12	0.25	0.71
		GNN 6-neighbours	**15**	**0.36**	**0.93**
Couture	1764	CNN	1694	0.36	0.99
		RF	1663	0.67	0.96
		RF 3-neighbours	1634	0.78	0.96
		RF 6-neighbours	1630	0.82	0.96
		GNN	1572	0.74	0.90
		GNN 3-neighbours	1645	0.69	0.97
		GNN 6-neighbours	**1651**	**0.94**	**0.98**

The results presented in Table 49 first show a significant improvement in the performance of the CNN, RF and GNN methods when trained on the same vineyard (on different subplot). This may seem logical, but it indicates that the 'ClaFD' dataset on which the methods were previously trained (experiment 1) lacked representative images of grapevines affected by FD and confounding diseases in these plots. For example, for the GNN method, the results improved from (p=0.25, r=0.26), (p=0, r=1), and (p=0.96, r=0.22) to (p=0.43, r=0.65), (p=0.27, r=0.63), and (p=0.74, r=0.9) respectively for the 'Neuffons,' 'Cognac,' and 'Couture' vineyards. A comparison of the results for one of the cross-validations of the GNN, GNN 3-neighbours, and GNN 6-neighbours methods on the 'Couture' vineyard is presented in Figure 51, illustrating the improvement in results when considering neighbouring grapevines for diagnosis.

Regarding the comparison between the GNN, GNN 3-neighbours, and GNN 6-neighbours methods, the best results were obtained with the GNN 6-neighbours method. The results for the GNN 6-neighbours methods were (p=0.76, r=0.71), (p=0.36, r=0.93) and (p=0.94, r=0.98) for the ‘Neuffons’, ‘Couture’ and ‘Cognac’ vineyards, respectively.

As for the RF methodologies, the results varied from vineyard to vineyard. For the 'Neuffons' and 'Cognac' vineyards, the best results were achieved by the RF method, with (p=0.87, r=0.21) and (p=0.19, r=0.49), respectively. For the 'Couture' vineyard, the RF 6-neighbours method obtained the best results: (p=0.82, r=0.96).

Regarding the results obtained by CNNs, it can be observed that even in cross-validation on a single vineyard, the results exhibited the same trend: high recall but low precision ((p= 0.17, r=0.86), (p=0.01, r=0.63), and (p=0.36, r=0.99) for the ‘Neuffons’, ‘Cognac’, and ‘Couture’ vineyards, respectively). The method generates a large number of false FD predictions and does not appear to be suitable for diagnosing at the scale of an entire vineyard.

When comparing the best results vineyard by vineyard, considering all methodologies, the GNN 6-neighbours methodology consistently outperforms the others.

The 'Neuffons' vineyard achieved the best results with (p=0.76, r=0.71), which may be seen as somewhat disappointing for cross-validation. However, given the very low expression of FD symptoms in the vines at this vineyard, these results are good. Combining the detections from six images likely provides enough information to accurately classify most images.

The 'Cognac' vineyard obtained the lowest cross-validation results, with the GNN 6-neighbours method achieving (p=0.36, r=0.93). Even when trained on a portion of this plot, the model struggled to deal with the significant number of vines not affected by FD but with symptom predictions indicating FD. This results in a high number of false positives, although it has been reduced compared to the GNN method (p=0.27). The low number of FD cases on the ‘Neuffons’ and ‘Cognac’ vineyards also makes it challenging for the methods to correctly detect affected vines.

For the 'Couture' vineyard, which had the highest number of FD cases and the most pronounced symptom expression but also contained grapevines displaying confounding symptoms, the results of the GNN 6-neighbours method were excellent: (p=0.94, r=0.98). Here, the algorithm successfully distinguishes grapevines affected by FD from healthy vines or those showing confounding symptoms.

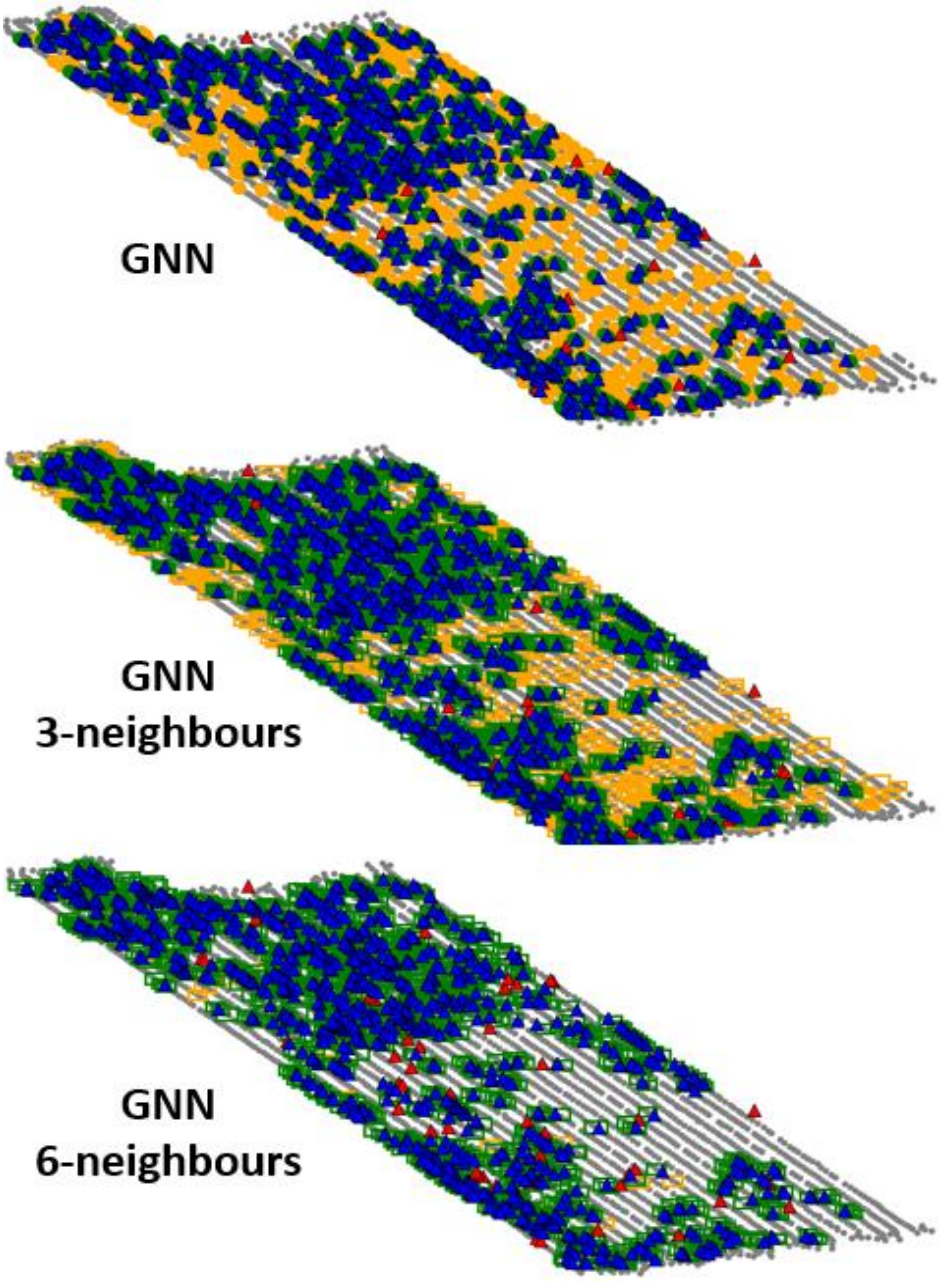

Figure 51. Comparison of the results obtained during the cross-validation of the GNN, GNN 3-neighbours, and GNN 6-neighbours approaches on the 'Couture' vineyard. Training and validation were conducted on two out of the three parts of the vineyard, and the results presented here are the predictions of the models on the third part of the vineyard. The yellow points or areas represent the grapevine (for the GNN method) or groups of grapevines (for the GNN 3-neighbours and GNN 6-neighbours methods) incorrectly predicted as showing grapevines affected by FD. Conversely, the green points or areas indicate the grapevine or groups of grapevines correctly predicted as displaying grapevines affected by FD. Blue triangles indicate the FD-affected grapevines predicted by the models, while red triangles represent the FD-affected grapevines that were not predicted. Grey points represent the grapevines for which the model did not detect FD.

Regarding the hyperparameters of the GNN 6-neighbours methodology, it's interesting to note that using the same number of MinCut pooling functions (1) and clusters of nodes formed (10) as the GNN method on 'ClaFD' yielded the best results. However, in this case, there are 4 consecutive MP functions preceding the MinCut pooling function. Applying multiple MinCut pooling functions seems

to degrade the model's performance. One possible explanation is that each time clusters of nodes are formed, the new node feature for that cluster is the average of the features of the nodes in that cluster. This process degrades the information, and the application of multiple MinCut pooling functions further degrades it. The successive pooling layers are supposed to extract information at different scales and provide a condensed form of the graph that is relevant for classification. However, even though the size of the graphs has been increased, they are still relatively small. The maximum number of nodes is 659 and average number of nodes is 113 on the 3 vineyards for the 6-neighbours graphs. A single pooling layer provides a condensed form of the graph with minimal information loss, relevant for its classification. On the other hand, for the 6-neighbours graphs, using 4 consecutive MP functions before the pooling function yielded the best results. For each node, information needs to be combined with information from distant nodes to obtain a feature vector that best describes it for classification within the clusters.

Experiment 3

The GNN 6-neighbours methodology was subsequently trained on one or two of these vineyards and tested on the remaining vineyard(s). The results of these tests are presented in Table 50.

Table 50. Experiment 3: results for the 'FD' class, when trained on one or two complete vineyards and tested on the remaining vineyard(s) using the GNN 6-neighbours methodology.

Test vineyard	Train vineyard	precision FD	Recall FD
Neuffons	Cognac	0	0
	Couture	0	0
	Cognac + Couture	0	0
Cognac	Neuffons	0.01	0.63
	Couture	0.1	0.13
	Neuffons + Couture	0.08	0.18
Couture	Neuffons	0.88	0.92
	Cognac	0.65	0.88
	Neuffons + Cognac	0.88	0.93

The results presented in Table 50 show significant disparities. First, concerning the 'Neuffons' vineyard, regardless of the training vineyard(s), no graph was classified, correctly or incorrectly, as representing a set of symptoms associated with a grapevine affected by FD. The very low expression of symptoms in the affected vines on this vineyard contrasts with the presence of very pronounced symptoms on the other two vineyards. As a result, the model learned that for a graph to be classified as FD, it must contain a lot of FD symptoms. However, this is not the case for the 6-neighbours graphs created from the detections on the 'Neuffons' vineyard.

Regarding the 'Cognac' vineyard, the precision was very low regardless of the training vineyard(s) (p=0.01, 0.1, and 0.08). Here, the presence of numerous vines displaying confounding symptoms is the cause. The precision was lowest (p=0.01) when training was performed on the

'Neuffons' vineyard. Since the model learned from vines affected by FD with very few symptoms, all vines showing symptoms are classified as FD on the 'Cognac' vineyard. Poor symptom detection on the confounding diseases of the 'Cognac' vineyard resulted in classifying all these vines as FD. However, this led to the highest recall (r=0.63). Training on a vineyard with FD-affected vines displaying more symptoms ('Couture') improved precision (p=0.1) but still led to too many false positives. In return, recall droeds (r=0.13), and fewer FD-affected vines were detected by the model. Finally, combining the graphs from the 'Neuffons' and 'Couture' vineyards for training yielded results that were somewhat of an 'average' of training on these separate vineyards (p=0.08, r=0.18). Combining the two vineyards in training did not help the model better detect FD on the 'Cognac' vineyard.

Finally, the results were the best on the 'Couture' vineyard. Training the model on the 'Neuffons' vineyard achieved a precision of 0.88 and a recall of 0.92. The model learned on the 'Neuffons' vineyard to detect FD even with few symptoms, and even though the vines on the 'Couture' vineyard had many symptoms, most of the grapevine affected by FD were still correctly detected. This contrasts with training on the 'Couture' vineyard and testing on the 'Neuffons' vineyard, where no vines were detected as FD. It can be supposed that if the model is trained on vines with few symptoms, it will be able to predict vines with many symptoms. However, if the model is trained on vines with many symptoms, vines with few symptoms will not be correctly detected. Training on the 'Cognac' vineyard yielded poorer results on the ‘Couture’ vineyard, with a precision of 0.65 and a recall of 0.88. Nevertheless, this is still a decent result considering the symptom detections of the 'Cognac' vineyard and demonstrates a certain adaptability of the method, which manages the numerous false detections on the 'Cognac' vineyard when trained on it. Finally, training on both the 'Neuffons' and 'Cognac' vineyards yielded the best results, with a precision of 0.88 and a recall of 0.93. The model did not get lost when training on these two completely distinct vineyards in terms of symptom expression. This result suggest that the model is capable of achieving good results when trained on very different graphs.

E. Synthesis

The acquisitions made at the scale of three vineyards allowed for the initial investigation of automated FD diagnosis at the whole-vineyard scale. Although this database is limited, it highlights methodologies with greater potential as well as certain challenges when implementing the developed methodologies in a real-world use case.

The 'Neuffons' vineyard was planted with the red Merlot variety, and most of the vines affected by FD displayed very few symptoms. The 'Cognac' vineyard was planted with the white Ugni Blanc variety and had a high number of vines affected by grapevine yellows without being infected by FD. The 'Couture' vineyard was planted with the red Cabernet sauvignon variety and contained numerous vines affected by FD that expressed the disease's symptoms prominently.

To evaluate the effectiveness of the CNN, RF and GNN methods in classifying images from these vineyards, the models that achieved the best results on the 'ClaFD' dataset were initially tested. The GNN methodology outperformed RF and CNN on all three vineyards, although the results were not entirely satisfactory. Specifically, the precision and recall scores for the 'Neuffons' vineyard were (p=0.25, r=0.26), for the 'Cognac' vineyard were (p=0, r=1) and for the 'Couture' vineyard were (p=0.96, r=0.22).

In an effort to improve the results, RF and GNN methods that incorporated symptom detections from neighbouring images were developed. Experiments with a '3 neighbours’ method, considering the images on the left and right of the target image, as well as a '6 neighbours’ method, incorporating detections from opposite sides of the vines in these three images, were conducted. Intra-vineyard cross-validation tests for the CNN, the standard RF and GNN methods as well as the 3 and 6 neighbours methods were conducted to identify the most suitable approach for vineyard-scale automated diagnosis of FD.

The GNN 6 neighbours methodology emerged as the most effective for all three vineyards, with results reaching (p=0.76, r=0.71) for the 'Neuffons' vineyard, (p=0.36, r=0.93) for the 'Cognac' vineyard, and (p=0.94, r=0.98) for the 'Couture' vineyard. These findings underscored the significance of considering the surrounding vines and both sides of the same vine for improved diagnostic accuracy. Finally, the GNN 6 neighbours method was tested across the entire extent of each vineyard, training it on images from one or both of the other vineyards. This allowed the test of its performance in a real-world scenario where the images of the target vineyard have never been encountered before. The results varied: during the 'Neuffons' vineyard test, the model did not detect any FD cases, while the 'Cognac' vineyard test yielded poor results (p=0.08, r=0.18). Ultimately, the 'Couture' vineyard test produced the most promising outcomes (p=0.88, r=0.93).

These latest results emphasize the fact that, in a decision-making process, learning set must absolutely contain similar cases to those encountered during the production deployment of the same process. This is particularly pronounced in the case of FD detection, where it is known that symptomatic expression varies greatly depending on the grape variety or vintage. Even if diagnostic algorithms have shown excellent results in a specific context, their performance can be much reduced in a different context, with cases that have never or rarely been observed before.

Conclusion

Review of the objectives and hypotheses

This research aimed to study the automated diagnosis of multi-symptom grapevine diseases using digital technologies. The objectives were to: i) propose a comprehensive protocol that integrates RGB imaging, ground truth dataset collections and various types of annotations at both the plant and symptom scales; ii) develop innovative chains of artificial intelligence algorithms for the automated diagnostic of multi-symptom grapevine diseases; iii) conduct an evaluation of these methodologies using data that accurately reflect the vineyard conditions and iv) provide a solution to identify high-risk areas of the presence of these diseases in order to assist prospecting efforts.

Before beginning the research, the literature review of previous studies on this subject allowed for the formulation of four hypotheses which have guided this research.

Firstly, proximal sensing appeared to be the most optimal data acquisition method for distinguishing the various symptoms expressed by diseased grapevines. Furthermore, it seemed that RGB imagery was a suitable way for correctly visualizing the investigated disease symptoms, which could be distinguished by their colours and shapes. It was considered that multi or hyperspectral imaging would not provide significant additional information for their visualization.

Secondly, deep learning approaches seemed highly suitable and promising for the automated processing of images and diagnosis at the grapevine scale. It appeared that there was a lack of consideration in the literature regarding the inclusion of confounding diseases in the learning base of the algorithms studied (Al Saddik, 2019; Albetis *et al*., 2018; Boulent *et al*., 2020; Daglio *et al*., 2022; Musci *et al*., 2020). However, grapevines are susceptible to many biotic or abiotic stresses that can lead to highly varied visual symptoms, which can be easily confused with those of the diseases under study. The diagnosis of certain multi-symptom diseases such as Flavescence Dorée (FD) is performed in the field by examining the presence and combination of symptoms on three different organs simultaneously: the leaves, shoots and bunches. This allow the distinction of FD from its confounding diseases, which may exhibit very similar foliar symptoms but not on other organs.

The third hypothesis was that the traditional approach in previous research on this subject, which processes an image in a single pass to provide a grapevine-scale diagnosis, would not be able to deal with the complexity of this diagnostic process. Approaches in two steps, consisting of a first step dedicated solely to the detection and distinction of symptoms of the targeted disease and its confounding diseases, followed by a second step dedicated to combining these detections to provide the vine-scale diagnosis, seemed to hold more potential in the automated diagnosis of these multi-symptom diseases.

Finally, a last hypothesis was that when diagnosing a grapevine disease, taking into account the data acquired from neighbouring vines as well as from both sides of these vines would improve the diagnosis at the grapevine scale.

Research synthesis

In order to study the research topic and affirm or refute the four hypotheses, two distinct types of multi-symptom grapevine diseases were targeted, being at the centre of two distinct projects in which this research participated.

Firstly, FD, belonging to the category of diseases known as grapevine yellows, was at the core of the French project ProspectFD, which aims to create a comprehensive decision support tool for FD prospecting. This disease, which has a significant impact in Europe, presents three distinct symptoms on leaves, shoots, and bunches, and their simultaneous presence is necessary for an optimal in-field diagnosis. In more detail, this disease is characterized by a discolouration of the leaves (turning red for red grape varieties and yellow for white grape varieties), non-lignification of the shoots and drying of the bunches.

A second type of multi-symptom disease was studied: Grapevine Trunk Diseases (GTDs) studied in direct connection with the historical and current projects taking place in New Zealand, at

the Plant and Food Research (PFR) laboratory of the Marlborough Research Centre. Here, Botryosphaeria and Eutypa diebacks were the targeted diseases. These two diseases both present three different symptoms, however, contrary to FD, their simultaneous presence is not essential for diagnosis. Stunted growth of the shoots (named ‘Short shoot’), one-sided vegetative growth on the trunk (named ‘Half head’) and cankers are the symptoms of these diseases.

The establishment of a database specific to these two types of diseases was the first step in this research. Regarding FD, 1483 images were acquired at the vineyard scale, encompassing 5 different grape varieties (Cabernet sauvignon, Cabernet franc and Merlot for red varieties, Ugni blanc and Sauvignon blanc for white varieties). Out of these 1483 images, 754 featured grapevines affected by FD, while the others displayed grapevines with foliar symptoms that could be confused with those of FD. The latter were categorized into three different classes: 'Esca' for grapevines affected by Esca, 'CONF' for grapevines with leaves visually different from healthy leaves, and 'CONF+' for grapevines exhibiting foliar symptoms almost identical to those of FD.

As for GTDs, a total of 10,305 images were acquired to build the dataset for studying their automated diagnosis. Among these, 2830 images displayed grapevines affected by GTDs, and 7,475 showed healthy vines. It must be noted that the surveyed vineyards during the database collection for the GTD study did not have confounding diseases. Also, the photographed vines were of the predominant variety in the acquisition region, namely Sauvignon Blanc.

The initial grapevine-scale diagnostic results were obtained through Convolutional Neural Networks (CNNs). This one-step method, which takes the image as input and directly classifies it, is the most commonly used approach in research on the automated diagnosis of grapevine disease (Boulent *et al.*, 2020; Ji *et al.*, 2020; B. Liu *et al.*, 2020).Testing this methodology on the image datasets of the two types of diseases had two main purposes: i) to obtain initial disease diagnostic results for comparison with subsequent results and ii) to confirm or refute the hypothesis that this methodology did not effectively distinguish between vines affected by a disease and vines affected by a disease with confounding symptoms. Three different CNN architectures, among the most prevalent ones, were tested (MobileNetV3-large, ResNet50, and EfficientNetB5). During this study for the automated diagnosis of FD, tests were conducted by grouping different red grape varieties on one side and different white grape varieties on the other. The dataset of red grape varieties contained images of Cabernet Sauvignon acquired in 2020 and 2021, Cabernet Franc, and Merlot acquired in 2021, with 16% of images from the 'CONF+' class. In comparison, the dataset covering white grape varieties contained only 3% of images from the 'CONF+' class and included images of Ugni Blanc acquired in 2020 and 2021, as well as Sauvignon Blanc acquired in 2021. The best results for classifying 'FD' class images for red grape varieties were (p=0.87, r=0.84), while those for white grape varieties were (p=0.97, r=0.96). These results appeared to demonstrate the improvable performance of CNNs in distinguishing between FD and diseases with very similar symptoms. They also seemed to confirm their effectiveness found in the literature in diagnosing FD in the absence of confounding diseases and their ability to effectively consider FD symptoms from vines of different grape varieties.

The study of CNNs for the automated diagnosis of GTDs was performed too. The dataset tested included 4,128 images, with 2,212 of them featuring vines affected by GTDs. The CNN ResNet50 with input image resolution division achieved the best results, namely (p=0.94, r=0.92) for classifying images of vines affected by GTDs. This result reaffirmed with another type of disease that CNNs deliver excellent results for the automated diagnosis of vine diseases when highly confusable diseases are not taken into account.

In order to achieve a better diagnosis of diseases and to improve their differentiation from confounding biotic and abiotic factors, the hypothesis was that a two-step approach, with a first step solely dedicated to symptom detection and a second step dedicated to their association, would be beneficial. The study conducted after the one on CNNs focused on the automated detection of symptoms of the targeted diseases.

In order to detect the symptoms, some of the FD and GTDs images were annotated at the symptom scale. Two types of annotations, which best corresponded to the properties of the studied symptoms, were carried out. For FD, symptomatic leaves of FD ('FD leaf' class), Esca ('Esca leaf' class),

and visually different leaves from healthy leaves as well as leaves similar to those of FD ('CONF leaf' class) were annotated with bounding boxes on 744 images. These annotations resulted in 11,279 leaves in the 'FD leaf' class, 2,467 in the 'Esca leaf' class, and 22,421 in the 'CONF leaf' class, all annotated with bounding boxes. Symptomatic shoots and bunches of FD were annotated using segmentation masks. There were 128 segmentation masks, encompassing 833 symptomatic shoots and 468 symptomatic bunches. Additionally, 660 healthy bunches were also annotated on these segmentation masks, as their presence in large numbers in an image could be evidence of the absence of the disease. For GTDs, 505 images were annotated with bounding boxes. Of these, 268 'Half head' type symptoms, 1,480 'Short shoot' type symptoms, and 139 'Canker' type symptoms were annotated in this manner. Segmentation masks were also created for GTDs, highlighting the distinction between the vine trunk being studied ('Trunk' class), its foliage ('Foliage' class), and the rest ('Background' class). The goal was to extract characteristic information from this segmentation, such as the difference in size and foliage of certain shoots (characteristic of the presence of ‘Short shoot’) or foliage in contact with only one side of the trunk (characteristic of ‘Half head’). In this regard, 113 GTDs images were annotated with segmentation masks comprising these three classes.

The detection of symptomatic FD leaves achieved the best results with the YOLOv8m detection algorithm, with (p=0.78, r=0.61) for red grape varieties and (p=0.48, r=0.69) for white grape varieties. The segmentation of symptomatic shoots, symptomatic bunches and healthy clusters obtained the best results with the ResUNet segmentation algorithm, with (p=0.83, r=0.57), (p=0.78, r=0.40), and (p=0.85, r=0.68), respectively. Regarding GTDs, the best symptom detection results were also achieved with the YOLOv8 algorithm, with (p=0.85, r=0.85) for the 'Half head' class and (p=0.58, r=0.58) for the 'Short shoot' class. No Canker symptoms were correctly detected. The segmentation between the trunk, foliage, and the rest achieved the best results with the ResUNet algorithm with (p=0.86, r=0.95) for the 'Trunk' class, (p=0.93, r=0.92) for the 'Foliage' class, and (p=0.97, r=0.97) for the 'Background' class.

These results, while not exceptional, should be considered in light of the complexity of the task: presence of confusing symptoms, symptoms masked by the foliage and symptoms with different appearances depending on the grape variety, vine phenological stage and disease stage. A less fast algorithm than YOLOv8m might have potentially yielded slightly better detection results. However, these results served as an initial foundation on which to build the second step of the methodology, on which a significant portion of the efforts of this research was concentrated, namely the association of these detections to deliver the final diagnosis. Specifically, these detection association methodologies were exclusively developed and tested for the FD disease type, where the number of symptoms detected per image and the number of 'CONF+' images were substantial. This dataset was instrumental in determining whether the two-step methodologies would yield better results in distinguishing between grapevines affected by the targeted disease and its highly confusing symptoms of other vine health issues.

Two different symptom association methodologies were tested: The first used a fixed-size information vector for each image based on the detected symptoms and classified it with a Random Forest (RF) classifier. The second represented the detections as graphs for each image, which were then classified by a Graph Neural Network (GNN). For the dataset of red varieties, containing 16% of images from the 'CONF+' class and on which CNNs achieved the best results with (p=0.87, r=0.84) for classifying images of vines affected by FD, the RF methodology achieved the best results with (p=0.86, r=0.9), and the GNN methodology with (p=0.9, r=0.96). For the dataset of white varieties, containing 3% of images from the 'CONF+' class and on which CNNs achieved the best results with (p=0.97, r=0.96) for classifying images of vines affected by FD, the RF methodology achieved the best results with (p=1, r=0.94), and the GNN methodology with (p=0.97, r=0.90). Therefore, the symptom association methods yielded better results on the dataset when the proportion of 'CONF+' class images was substantial (16% for the red grape varieties dataset). When this proportion was very low (3% for the white grape varieties dataset), the results were not better than those obtained by CNNs. These image classification results, in conjunction with the symptom detections results upon which the classification methods rely, are very good. This seems to indicate that the choice to acquire data through proximal

sensing using RGB imagery, to detect and then associate symptoms using deep learning algorithms, were relevant choices for the automated diagnosis of multi-symptoms grapevine diseases.

Finally, the effectiveness of the CNN and the two-step methodologies was tested to deliver a diagnosis at the scale of an entire vineyard, simulating a real-world use case. Acquisitions were made on both sides of each vine from three distinct vineyards, each with unique characteristics. The 'Neuffons' vineyard was planted with the red Merlot variety, and most of the vines affected by FD displayed very few symptoms. The 'Cognac' vineyard was planted with the white Ugni blanc variety and had a high number of vines affected by grapevine yellows without being infected by FD. The 'Couture' vineyard was planted with the red Cabernet sauvignon variety and contained numerous vines affected by FD that expressed the disease's symptoms prominently. The trainings of the CNN, RF and GNN methods that achieved the best results mentioned above were tested to classify each image acquired in these vineyards. The GNN methodology obtained the best results on all three vineyards, although they were not satisfactory. They were (p=0.25, r=0.26) for the 'Neuffons' vineyard, (p=0, r=1) for the 'Cognac' vineyard, and (p=0.96, r=0.22) for the 'Couture' vineyard. In an attempt to achieve better results, RF and GNN methods using symptom detections from the images on the left and right of the image to be classified (3 neighbours method) as well as from the opposite sides of the vines in these three images (6 neighbours method) were developed. Intra-vineyard cross-validation were tested for the CNN, the standard RF and GNN methods considering 1 image, as well as the 3 and 6 neighbours methods. These tests aimed to determine the most suitable method for vineyard-scale diagnosis. The GNN 6 neighbours methodology obtained the best results for each of the three vineyards, namely (p=0.76, r=0.71) for the 'Neuffons' vineyard, (p=0.36, r=0.93) for the 'Cognac' vineyard, and (p=0.94, r=0.98) for the 'Couture' vineyard. These results highlighted the fact that considering the surrounding vines as well as the two sides of the same vine led to better diagnostic results. Finally, the GNN 6 neighbours method was tested on the entirety of each vineyard, being trained on the images from one or both of the other vineyards, to establish its effectiveness in a real-world scenario where the model would have never encountered the images of the target vineyard before. During the test on the 'Neuffons' vineyard, the model did not make any detections of FD. The test on the 'Cognac' vineyard yielded poor results (p=0.08, r=0.18). Finally, the test on the 'Couture' vineyard obtained the best results (0.88, 0.93).

Response to the thesis statement

In response to the question, "How to effectively diagnose grapevine multi-symptom diseases by digital means considering real vineyard conditions?" The answer, according to this research, lies in the creation of the most complete database as possible in term of symptoms expression of the studied diseases and its confounding diseases, precise symptom detection and subsequent intelligent association of the symptoms of these diseases. Taking into account the symptoms detected on both sides of the same vine as well as on neighbouring vines appears to be a significant advantage. However, when delving into the complex reality of vineyards, it becomes apparent how challenging addressing this issue is. Each vineyard seems to have its own specificity, its own expression of symptoms for the targeted diseases, its own confounding diseases, all of which makes diagnosing on vineyards other than those where the algorithms were trained extremely complicated. Creating a database and obtaining good results with it is one thing, applying the developed model to a vineyard that has never been studied is another. Efforts have been made to build a solid database containing multiple grape varieties, expressions of the targeted disease and its confounding health problems very diverse. Methodology that yields good results and innovative methodologies for even better results on this database have been developed. However, there always seem to be expressions of symptoms for the targeted disease or its confounding diseases that have not been encountered before when these methods are put into practice.

Position in relation to other research

This research has confirmed the findings of previous studies on this topic, namely that the use of CNNs allows for very good results in automated diagnosis of grapevine diseases when considering only vines

affected by the targeted disease and healthy vines (Boulent *et al.*, 2020; Ji *et al.*, 2020; B. Liu *et al.*, 2020). However, it distinguishes itself from these previous studies by also taking into account diseases with confounding symptoms during the development of automated diagnosis models. When considering these confounding diseases, CNNs are no longer the best models for automated diagnosis, and the proposed innovative two-step methodology deliver better results. Finally, even when considering these highly confounding diseases during model training, this study aligns with the conclusion of Al Saddik (2019) and Albetis *et al*. (2018) that, during the application of models in real-use cases, distinguishing between grapevines multi-symptom diseases and their very similar symptoms of other health problems is very challenging to achieve.

Limitations of the study

This research has several limitations. Only 3 acquisitions were performed at the scale of an entire vineyards. More acquisition at the vineyard scale could have improved the model that group 6 neighbouring images, which is the method yielding the best results for automated FD diagnosis. While different CNN architectures have been tested as the one-step methodology, vision transformers (S. Khan *et al*., 2022), which offer a novel and promising approach to process visual data that differs from traditional CNN, could have been tested. A greater diversity of images with annotated symptoms and a more in-depth study of symptom detection algorithms could have been conducted to achieve better results when associating them. Regarding GTDs, methods for associating symptom detections could have been developed to compare the results with those obtained during the FD study. A more thorough investigation of the diseases that could be confused with GTDs, a consideration of multiple grape varieties, and vineyard-scale acquisitions could have been conducted to provide a more comprehensive study of GTDs automated diagnosis. A more in-depth study of the chosen parameters of the RF (feature vectors) and GNN methods (node features and type of graph) could have been conducted.

Perspectives for improving the graph-based methodology

The emerging direction from this research for grapevine-scale diagnosis appears to revolve around refining methods for symptom detections association, with a particular focus on graph-based approaches. According to the literature review conducted on the topic of automated disease diagnosis in crops, this method has never been applied before. This approach yielded the best results during the study of automated FD diagnosis. It could be further developed by consolidating the learning databases of symptom detection and association algorithms. The detection and association algorithms for symptoms could also undergo a more in-depth study. For graph-based association, other approaches could have been considered, such as using different sets of nodes, assigning weights to graph links, creating heterogeneous graphs rather than homogeneous ones and employing alternative graph classification algorithms apart from GNNs. In the 3 and 6 neighbours approaches, the image registration process remained rudimentary. The use of registration and 3D reconstruction approaches, potentially relying on different sensors, as well as the use of different and less constrained robotic acquisition vectors, would allow a better leveraging of the redundancy between images and establish better connections between the various instances of symptoms on the same vine.

General conclusion and perspectives

The main goal of this research was the development and comparison of methods for the automated diagnosis of multi-symptom grapevine diseases. Data was collected through RGB imaging directly in the vine rows to obtain the most precise view of the different symptoms. The widely used methodology for automated grapevine diseases diagnosis in the literature was tested and yielded very satisfactory results in most cases. However, when the datasets contained a large proportion of vines with symptoms that were similar to those of the targeted disease, the results of this method became less accurate. In order to improve the differentiation performance between the targeted disease and diseases with similar symptoms, two-step methodologies were developed.

The first step involved the detection of individual symptoms, carried out by two neural algorithms, a segmentation one and a detection one. The second step involved the association of these symptom detections to deliver a diagnosis at the vine scale. The representation of the symptom detections as graphs and their processing by graph neural networks were proposed. This innovative methodology improved the distinction between the targeted disease and diseases with similar symptoms. The developed methodologies were subsequently tested on three acquisitions covering entire vineyards. While the results were very satisfactory for one vineyard, they were disappointing for the other two. These results highlight the challenges of the widespread practical implementation of automated grapevine diseases diagnosis methods. There are numerous variability factors affecting the expression of symptoms which, from one vineyard to another, from one vintage to another, from one grape variety to another, can take different forms. This research seems to confirm, despite the very promising results shown by the developed methodologies, the urgent need to conduct model training on consolidated databases, encompassing all or at least a very large portion of these variability factors.

To address all the specificities that may be encountered when applying the models in real-world use cases, a scaling up of the number of data acquisitions and annotations for model training seems necessary. The innovative methodologies developed in this research, which provide vine-by-vine diagnosis solely based on images, appear to be a first step in improving disease detection in vineyards. However, for a better identification of high-risk areas of disease presence within vineyards, it seems essential to develop a comprehensive decision support tool that takes into account factors such as the history of symptoms and the vineyard environment, and adapts diagnostic criteria accordingly.

Appendix 1. Precise locations of each block in which acquisitions of grapevine affected by FD images were made

Table 51. Summary of acquired images and associated ground truth at the grapevine *scale for FD. The department, city and localisation in (latitude, longitude) are available.*

Grape variety, acquisition year	Cabernet sauvignon, 2020	Ugni blanc, 2020	Cabernet sauvignon, 2021	Merlot, 2021	Cabernet franc, 2021	Ugni blanc, 2021	Sauvignon blanc, 2021	**Total**
City (department number)	Saint-Sève, Louviac, Semens (33)	Reparsac (16)	Saint-Martin (33)	Rions Saint-Martin (33)	Rions (33)	Reparsac Saint-Laurent (16)	Langoiran (33)	
Localisation (latitude, longitude)	44.6126886, -0.0379963 44.6102027, -0.0389426 44.6255832, -0.3070293 44.6071638, -0.2460330 44.6058326, -0.2424245 44.5750228, -0.2468006	45.7406102, -0.2293657 45.7414515, -0.2271748 45.7409606, -0.2255990 45.7405824, -0.2293786	44.5712274, -0.1697558	44.6704526, -0.3561660 44.5712274, -0.1697558	44.6726088, -0.3610193	45.7391416, -0.2286879 45.3702535, 0.0334376	44.6992974, -0.3924154	
Dataset name	**CS20**	**UB20**	**CS21**	**M21**	**CF21**	**UB21**	**SB21**	
Total number of images	**405**	**463**	**116**	**98**	**86**	**161**	**154**	**1483**
of which FD	159	211	107	53	56	112	56	754
of which Esca	97	49	3	5	8	28	59	249
of which CONF	90	153	6	23	22	17	35	346
of which CONF+	59	10	0	17	0	4	4	94

Appendix 2. Automated diagnostic of Esca

During the field visits to acquire images of grapevines affected by FD, many grapevines were also affected by Esca disease. This disease only exhibits symptoms on leaves and did not fall within the scope of this research problem. However, since the foliar symptoms of Esca can be confused with those of FD, images of Esca-affected grapevines were acquired, and symptomatic leaves of Esca were annotated to improve the detection of symptomatic FD leaves. Thus, through these annotations at the image and symptom scale, it was possible, alongside the study of FD automated diagnosis, to obtain results of the automated diagnosis of Esca. These results are presented in this appendix.

A. Description of the disease

Esca, classified as a Grapevine Trunk Disease (GTDs), stems from a diverse array of fungal pathogens with no taxonomic relation (Larignon & Dubos, 1997). These fungi target the wood tissues of grapevines, causing necrosis even within the vital vascular tissues. This process ultimately disrupts the vine's physiology and has the potential to result in the death of the plant.

Regarding Esca, the viticulture industry has been deeply concerned due to the lack of effective control measures. In the past, sodium arsenite was the sole registered pesticide in Europe for Esca control. However, it was prohibited in the early 2000s due to its toxicity, which posed risks to both winegrowers and the environment (Mondello *et al*., 2018).

When it comes to visible symptoms, there are two classic forms of Esca based on leaf manifestations. The first is a slow or chronic form characterized by leaf blade discolouration, with yellow digitations appearing in white grape varieties and dark-red in red grape varieties. These discoloured areas are bordered by yellow regions in red grape varieties, giving the leaves a tiger-stripe-like appearance. The second foliar form, known as apoplectic or apoplexy, is marked by rapid and extensive drying of the entire vine, ultimately leading to the death of the plant. Figure 52 illustrates the different form of Esca. Small, dark-brown to purple spots can also develop on the berries (Essakhi *et al*., 2008).

Figure 52. Example of visual symptoms of Esca. Image 1: Esca symptomatic leaves on white grape variety, slow form of the disease. Image 2: Esca symptomatic leaves on red grape variety, slow form of the disease. Image 3: apoplectic form of Esca. Images from (Rancon, 2019).

B. Database acquisition and annotation

Table 52 summarizes the number of images of grapevines affected by Esca (named 'Esca' class) acquired during the various acquisition campaigns conducted in 2020 and 2021. Only images from grapevines suffering from the slow form of Esca were acquired.

Table 52. Summary of the number of 'Esca' images according to the grape variety and acquisition year.

Number of 'Esca' images	Red varieties		White varieties		Total
	Cabernet sauvignon	Merlot	Cabernet franc	Ugni blanc	Sauvignon blanc	
2020	97	0	0	49	0	146
2021	3	5	8	28	59	103
Total	113			136		249

It can be observed that Esca was more prevalent in certain plots, and the number of images in the 'Esca' class in the dataset is quite varied. However, the number of images of red grapevine varieties exhibiting Esca symptoms is then 113, while for white grape varieties, it is 136. This separation of images in the 'Esca' class between red and white grape varieties is more logical since the symptoms differ between the two grape colours, as illustrated in Figure 53.

Figure 53. Zoom on symptomatic Esca leaves for red and white grape varieties on the acquired images. On the first row, 4 symptomatic leaves of Esca on red grape variety (Cabernet sauvignon) are displayed, from the earliest stage to the most advanced. The second row presents 4 symptomatic leaves of Esca on white grape variety (Ugni blanc), also from the earliest stage to the most advanced.

In Figure 53, it can be observed that during image acquisition, symptomatic leaves of Esca encountered on red grapevines exhibit a pronounced reddening, whereas on white grapevines, symptomatic Esca leaves show diffuse yellowing.

The number of acquired images of grapevine suffering from Esca presented in Table 52 provided an initial foundation for studying its automated diagnosis, either through CNNs (Convolutional Neural Networks) or two-stage methodologies. Furthermore, two-stage methodologies require an initial step of symptom detection. Since the visual symptoms of Esca are confounding from those of FD expressed on the leaves, the symptom annotations for Esca consisted of bounding boxes around the symptomatic Esca leaves, in the same way as symptomatic leaves for FD were annotated (Chapter III Section D). Examples of annotations by bounding boxes of symptomatic Esca leaves are presented in Figure 54.

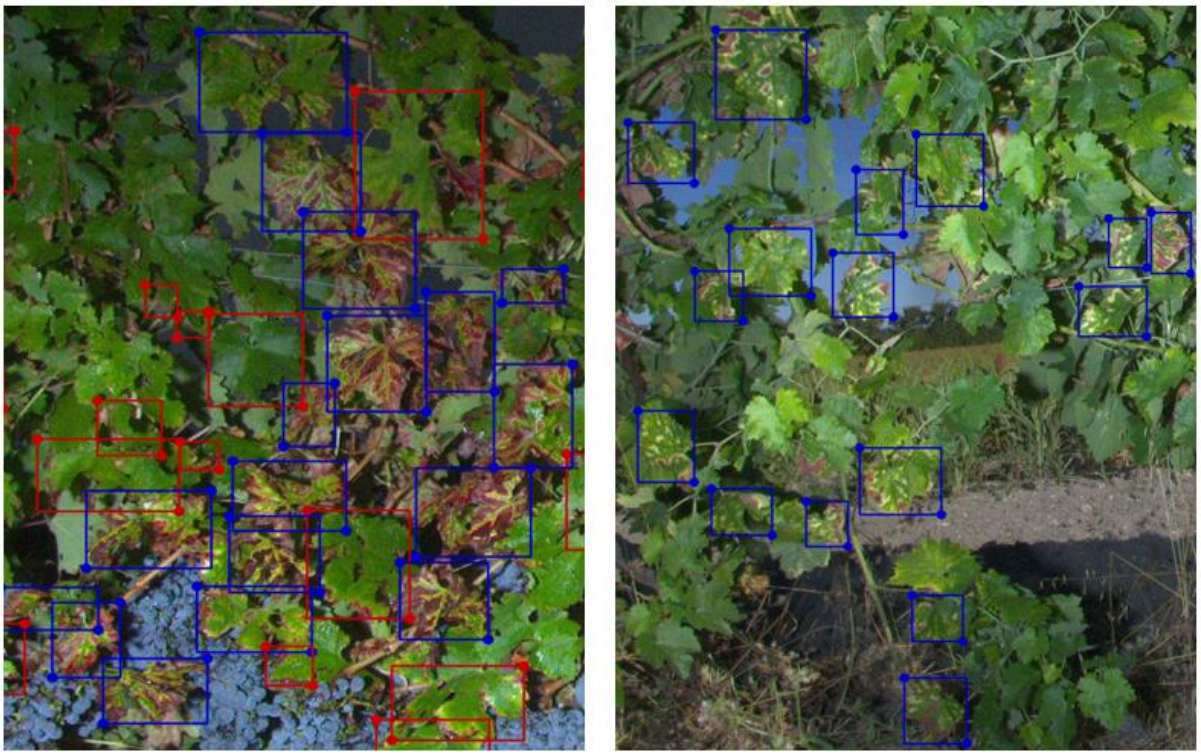

Figure 54. Examples of annotation using bounding boxes for symptomatic Esca leaves. The image on the left shows annotations of Esca symptomatic leaves on Cabernet sauvignon grape variety, while the one on the right is for Ugni blanc grape variety. Leaves belonging to the 'Esca leaf' class are enclosed in blue bounding boxes, while those of the 'Confounding leaf' class are enclosed in red.

The number of symptomatic leaves of Esca annotated by bounding boxes is provided in Table 53.

Table 53. Summary of the number of annotated Esca symptomatic leaves by bounding boxes per dataset.

Number of 'Esca' bounding boxes annotations	**'Det_all_red'**			**'Det_all_white'**		
	Cabernet sauvignon	**Merlot**	**Cabernet franc**	**Ugni blanc**	**Sauvignon blanc**	**Total**
2020	299	0	0	425	O	724
2021	357	0	274	758	354	1743
Total		930		1537		2467

Some of the figures in Table 53 may appear high given the number of images of grapevines affected by Esca in certain datasets. This is because, for these datasets, grapevines suffering from FD also exhibited symptoms of Esca and those were consequently annotated in these images as well. However, in the classification dataset, there are no grapevine exhibiting both FD and Esca symptoms to avoid any confusion. The images annotated with bounding boxes from the Cabernet Sauvignon 2020, Cabernet Sauvignon 2021, and Cabernet Franc datasets were combined during the training of the algorithms into the 'Det_all_red' dataset. Similarly, the images from the Ugni Blanc 2020, Ugni Blanc 2021, and Sauvignon Blanc 2021 datasets were grouped into the 'Det_all_white' dataset. 'Det_all_red' and 'Det_all_white' including leaves annotated as symptomatic of FD and leaves annotated as confounding with FD in the images of their respective datasets datasets).

The images where these symptoms have been annotated were not present in the dataset for study of the classification at the image scale. The dataset used for studying the automated classification of Esca was those indicated in Table 54. This dataset encompassed images of the class 'FD', 'CONF' and 'CONF+' as detailed in Chapter III. The study of automated Esca diagnosis was conducted by grouping red grape varieties on one side and white grape varieties on the other. The images from the datasets for the classification of Cabernet Sauvignon 2020, Cabernet Sauvignon 2021, Merlot 2021, and Cabernet Franc 2021 were all combined into the dataset 'Cla_all_red'. Similarly, the other three datasets were grouped together to form the 'Cla_all_white' dataset. It can be noted that the number of images in the 'Esca' class is balanced between the two datasets.

Table 54. Description of the number of images of the class 'FD', 'Esca', 'CONF' and 'CONF+' for the classification at the image scale.

Dataset	Grape variety, acquisition date	FD	Esca	CONF	CONF+
Cla_all_red	Cabernet sauvignon, 2020	72	45	87	56
	Cabernet sauvignon, 2021	86	2	6	0
	Merlot, 2021	32	5	7	11
	Cabernet franc, 2021	23	3	2	5
Cla_all_white	Ugni blanc, 2020	83	13	88	8
	Ugni blanc, 2021	33	12	4	0
	Sauvignon blanc, 2021	41	44	17	2

All the results presented subsequently for these datasets will be derived from the same 5-fold cross-validation. Each training image also underwent the same data augmentation, as described in Chapter IV, Section B.

The initial method for automated diagnosis on the 'Cla_all_red' and 'Cla_all_white' datasets was carried out using the one-step methodology, which involves the use of Convolutional Neural Networks (CNNs).

C. 1-step diagnosis - CNNs

As presented in Chapter IV, Section A, three CNNs were tested for the automated classification of images. The images also underwent resolution degradation by factors of 16, 64, and 256. Table 55 presents the results obtained during this study for images in the 'Esca' class.

The best results were obtained for red grape varieties with the CNN ResNet50 and image resolution divided by a factor of 16, resulting in p=0.49 and r=0.50. The results for white grape varieties were much better, reaching p=0.81 and r=0.79 as the best results with the CNN MobileNetV3-large and image resolution divided by a factor of 64.

Firstly, this difference in results may seem strange, but it actually makes sense when examining the images in these two datasets. As illustrated in Figure 55, some Esca symptomatic leaves can be confused with symptomatic FD or its confounding diseases for the red grape varieties. In contrast, symptomatic Esca leaves can be distinguished from symptomatic FD leaves and its confounding leaves in the dataset of white grape varieties images.

Table 55. Results of the classification by CNNs for the images of the 'Esca' class.

Dataset	Pre-processing	MobileNetV3-large (p,r) 'Esca' class	ResNet50 (p,r) 'Esca' class	EfficientNetB5 (p,r) 'Esca' class
Cla_all_red	/16	(0.50, 0.37)	**(0.49, 0.50)**	(0.44, 0.35)
	/64	(0.43, 0.19)	(0.54, 0.43)	(0.51, 0.36)
	/256	(0.29, 0.19)	(0.28, 0.16)	(0.44, 0.21)
Cla_all_white	/16	(0.78, 0.76)	(0.84, 0.74)	(0.67, 0.71)
	/64	**(0.81, 0.79)**	(0.81, 0.81)	(0.65, 0.59)
	/256	(0.75, 0.81)	(0.70, 0.71)	(0.66, 0.59)

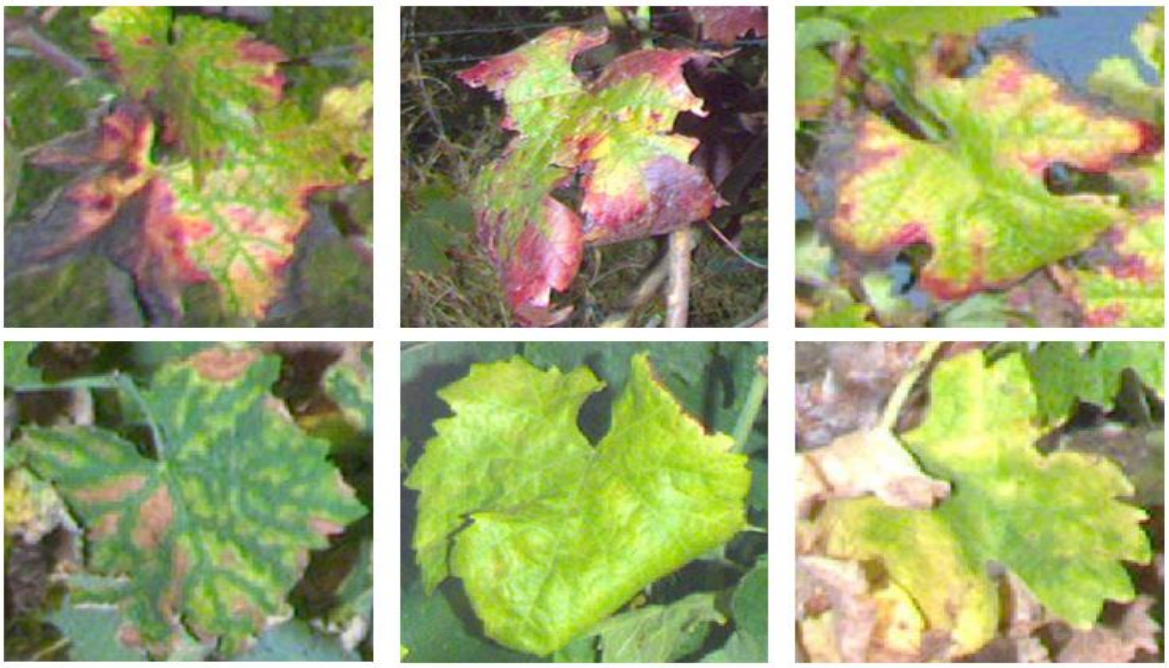

Figure 55. Comparison between symptomatic Esca leaves, symptomatic FD leaves and confounding leaves to those of FD for red and white grape varieties. First row: Leaves present in the images from the dataset 'Cla_all_red'. Second row: Leaves present in the images from the dataset 'Cla_all_white'. First column: Symptomatic Esca leaf. Second column: Symptomatic FD leaf. Third column: Confounding leaf to the symptomatic FD leaf.

Secondly, it is apparent that having the highest resolution is not necessary to achieve the best results in leaf detection, and a resolution division by a factor of 64 yields among the best results. However, resolution division by a factor of 256 yields the worst results for each method and dataset. The two-step methodologies were then developed and tested. The first stage involves the detection of individual disease symptoms, as presented in the next section.

D. Symptom detection

During the detection of symptomatic FD leaves, a separate class was dedicated to Esca symptomatic leaves because they could be confused with FD leaves. Due to the large numbers of Esca symptomatic leaves, a separate class was created for them. As a result, during the training of the symptom detection models presented in Chapter V Section A, results for the detection of Esca leaves were also available. The models tested, as detailed in Chapter V Section A, included two YOLO-type models (Redmon *et al*., 2016) (YOLOv4-tiny (Bochkovskiy *et al*., 2020) and YOLOv8m (Jocher *et al*., 2023)). The images were either divided into patches (see Chapter V Section A) for both models or provided in their entirety for the YOLOv8m model (after a resolution division by 4). Training and results for these models were done separately for red grape varieties on one hand and white grape varieties on the other. The results for the detection of symptomatic Esca leaves for red grape varieties are presented in Table 56, and those for the detection of symptomatic Esca leaves for white grape varieties are in Table 57.

Table 56. Precision and recall results for the detection of Esca symptomatic leaves in red grape varieties, using two different models and comparing the use of patches and whole images.

Training set	Algorithm, pre-processing	'Esca leaf' precision	'Esca leaf' recall
Det_all_red	YOLOv4-tiny, patches	0.73	0.57
	YOLOv8m, patches	0.67	0.66
	YOLOv8m, whole image	0.66	0.73

Table 57. Precision and recall results for the detection of Esca symptomatic leaves in white grape varieties, using two different models and comparing the use of patches and whole images

Training set	Algorithm, pre-processing	'Esca leaf' precision	'Esca leaf' recall
Det_all_white	YOLOv4-tiny, patches	0.49	0.45
	YOLOv8m, patches	0.54	0.52
	YOLOv8m, whole image	0.57	0.54

When comparing the overall performance of the models for red (Table 56) and white (Table 57) grape varieties, it becomes apparent that it contradicts the results obtained by CNNs. In this test the leaves are better detected in red grape varieties (with precision ranging from 0.66 to 0.73 and recall from 0.57 to 0.73) than in white grape varieties (precision ranging from 0.49 to 0.57, and recall from 0.45 to 0.54). However, CNNs achieved better image-level classification results for white grape varieties. This can be explained by the fact that even though symptomatic Esca leaves on white grape varieties look visually different from FD or confounding leaves of FD (as illustrated in Figure 55), their symptoms are less pronounced than those on red grape varieties. Consequently, an algorithm dedicated to their detection has more difficulties to accurately identifying them. On the contrary, even though some Esca leaves visually resemble FD or confounding leaves of FD for red grape varieties, the fact that their symptoms are more pronounced allows a dedicated algorithm for their identification to achieve better results.

Combining the results of precision and recall, the best performance in detecting symptomatic Esca leaves is achieved by the YOLOv8m model when using entire images for both red (p=0.66, r=0.73) and white (p=0.57, r=0.54) grape varieties. This aligns with the results for the detection of symptomatic FD leaves presented in Chapter V, Section A.

E. Symptom detections association

The detection of symptomatic Esca leaves has allowed for their inclusion in the process of associating symptom detections, which is the second step of the two-step methodology for diagnosing diseases at the image level.

a. Random Forest based methodology

Regarding the association methodology using the Random Forest classifier (RF) (Ho, 1995), depicted in Chapter VI Section A, the number of detected Esca leaves, their average confidence scores and the number of spatially close Esca leaves were included in the input information vector provided to the RF. Four different input vectors, as depicted in Chapter VI Section A, have been tested: an input vector taking into account only leaves detection predicted by Yolov4-tiny (VectYOLOv4leaves) or YOLOV8m (VectYOLOv8leaves) or the one taking into account all the symptoms detected for the FD diagnosis (VectYOLOv4all and VectYOLOv8all).

The results of the image classification for the 'Esca' class in the 'Cla_all_red' and 'Cla_all_white' datasets using the RF methodology are presented in Table 58.

Table 58. Results in precision and recall of the RF methodology for 4 different input vectors for the classification of 'Esca' image class for both red and white grape varieties sets.

	Cla_all_red		Cla_all_white	
Feature vector	**Esca precision**	**Esca recall**	**Esca precision**	**Esca recall**
VectYOLOv4leaves	1	0.64	**0.82**	**1**
VectYOLOv4all	1	0.64	0.78	1
VectYOLOv8leaves	**0.86**	**0.88**	**0.82**	**1**
VectYOLOv8all	1	0.64	**0.82**	**1**

The precision and recall results presented in Table 58 show minimal variation based on the different input vectors for the RF. Concerning red grape varieties, the best results are achieved using the vector containing only parameters derived from the detection of leaves by the YOLOv8m algorithm (p=0.86, r=0.88). All other vectors yield exactly the same results (p=1, r=0.64). For white grape varieties, the vector containing predictions for all symptoms, including leaves detected by the YOLOv4-tiny algorithm, obtains the lowest results (p=0.78, r=1), while the other three vectors yield identical results (p=0.82, r=1).

It is concluded that the best input vector for the automated diagnosis of red and white grapevine infections with Esca appears to be the one that takes into account information solely from leaf detections by the YOLOv8m algorithm. These results make sense as YOLOv8m achieves better results for symptomatic Esca leaves detection. It also aligns with the notion that leaf information alone provides better results, as other parameters are related to symptoms specific to FD and potentially confuse the algorithm more than assisting it.

b. Graph neural network based methodology

Regarding the association methodology using the Graph Neural Network (GNN) (Scarselli *et al.*, 2009), depicted in Chapter VI Section B, the detected Esca symptomatic leaves were included in the different graphs at the input of the tested GNNs. Three different input graphs, as depicted in Chapter VI Section B, have been tested: graphs taking into account 1) only leaves detection predicted by Yolov8m

('leafgraph'), 2) all the symptoms (on leaves, shoots and bunches) but without the detection of the confounding leaves of FD ('allgraph_withoutconf') and 3) with the detection of the confounding leaves of FD ('allgraph').

The results of image classification by GNN for the 'Esca' class for red and white grape varieties based on the three types of graphs are presented in Table 59. Only the best results, which are obtained using an MP function followed by MinCut pooling creating 10 clusters of nodes, are shown in this table. These results align with the best results achieved for image classification of the 'FD' class, as presented in Chapter VI, Section D.

Table 59. Results in precision and recall of the GNN methodology using the MinCut pooling function for the classification of 'Esca' image class for both red and white grape varieties sets.

			Cla_all_red		Cla_all_white	
Pooling	**Graph**	**Link distance (pixel)**	**ESCA precision**	**ESCA recall**	**ESCA precision**	**ESCA recall**
MinCut Pooling	leafgraph	100	**0.92**	**0.92**	**1**	**0.82**
	allgraph_withoutconf	100	0.90	0.81	0.80	0.88
	allgraph	100	0.89	0.80	0.81	0.93

The results presented in Table 59 demonstrate that the best image classification results for 'Esca' are achieved with a node-link distance of 100 pixels, compared to 200 pixels for 'FD' (see Chapter VI section D). This indicates that the 3 different symptoms of 'FD' needed to be more interconnected, whereas for 'Esca', with a single symptom on leaves, the proximity of symptomatic leaves was more indicative of the disease for the model.

In contrast to the results obtained for 'FD', the graph containing only leaf symptoms obtains the best results for the classification of 'Esca' images, reaching (p=0.92 and r=0.92) for red grape varieties, and (p=1 and r=0.82) for white grape varieties. The inclusion of symptoms on shoots and bunches for 'FD' did not improve the image classification results for 'Esca' and, in fact, to a decrease in performance.

F. Conclusion

The presence of numerous grapevines affected by Esca in the acquisition plots for grapevines affected by FD, along with foliar symptoms potentially confusing with those of FD, led to the creation of the specific 'Esca' class for the study of automated FD diagnosis at the image level. Additionally, the 'Esca leaf' class was created during the study of symptomatic leaf detection for FD. These specific classes allowed, during studies related to FD automated diagnosis, to obtain results on automated Esca diagnosis. No study was specifically conducted to optimize the results of automated Esca diagnosis, and the number of images of grapevines affected by Esca does not align with a comprehensive study on the subject. However, the results from different diagnostic methodologies highlight methodologies that seem more suitable for its diagnosis than others.

Firstly, the automated diagnosis methodology by CNN demonstrated its limitations, achieving satisfactory results on white grape varieties (p=0.81, r=0.79) but disappointing results on red grape varieties (p=0.49, r=0.50). The results of symptomatic Esca leaf detection contrasted with these findings, achieving (p=0.66, r=0.73) for red grape varieties and (p=0.57, r=0.54) for white grape varieties. These results highlighted that in the dataset of this study, even though symptomatic Esca leaves on white grape varieties look visually different from FD or confounding leaves of FD, their symptoms are less pronounced than those on red grape varieties. Consequently, an algorithm

dedicated to their detection has more difficulty accurately identifying them. On the contrary, even though some Esca leaves visually resemble FD or confounding leaves of FD for red grape varieties, the fact that their symptoms are more pronounced allows a dedicated algorithm for their identification to achieve better results.

The methods of associating Esca leaf detections to deliver a diagnosis at the image level successively outperformed the results obtained by CNNs. The RF methodology achieved image classification results for the 'Esca' class (p=0.86, r=0.88) and (p= 0.82, r=1) for red and white grape varieties, respectively. These results were obtained with the information vector containing only the information from leaf detections of the 'FD leaf,' 'Esca leaf,' and 'Confounding leaf' classes. The automated classification of grapevines affected by Esca by GNN achieved its best results (p=0.92, r= 0.92) and (p=1, r=0.82) for red and white grape varieties, respectively. As with the RF method, the graph containing only the information from leaf detections allowed for the best results. Furthermore, the node-link distance of 100 pixels in the graph, which yielded the best results, suggests that information about Esca leaves in very close spatial proximity is an important criterion for the automated diagnosis of the disease.

Appendix 3. Workflow of a supervised deep learning model

Training a deep learning model involves the process of iteratively updating the model's parameters (weights and biases) based on the input data and corresponding target outputs. The goal is to minimize the difference between the predicted outputs of the model and the ground truth labels, which is typically measured using a loss function. The training process can be divided into several key steps, each of which plays a crucial role in helping the model learn from the data and improve its performance.
Here follows a detailed explanation of the training process for a deep learning model:

1. Data Preparation:
    - The first step is to gather and preprocess the training data. This involves collecting a labelled dataset where the inputs and corresponding outputs (or labels) are known. The data is usually divided into two parts: the training set and the validation set.
    - Preprocessing includes tasks like normalization (scaling the data to a specific range), data augmentation (creating variations of the data to increase diversity), and splitting the data into batches for efficient computation (the model processes a certain number of samples at once).
2. Model Architecture:
    - Choosing an appropriate deep learning model architecture is crucial for the success of the training process. The architecture defines the layout and connectivity of layers in the model, including the number of layers, the types of layers (e.g., convolutional, recurrent, fully connected), and the number of neurons or units in each layer.
    - The choice of architecture depends on the nature of the problem (e.g. image classification, language translation, etc.) and the available resources (computing power and memory).
3. Forward Pass:
    - During training, input data (features) are fed into the model, and the model performs a forward pass. In the forward pass, the input data propagates through the network layer by layer.
    - At each layer, transformations (linear transformations, activation functions) are applied to the input data, and intermediate outputs (activations) are computed and passed to the next layer.
4. Loss Calculation:
    - The model's predictions are compared to the true target outputs using a loss function (also called a cost function or objective function). The loss function quantifies how far off the predictions are from the actual labels.
    - Common loss functions include mean squared error (MSE) for regression problems and categorical cross-entropy for classification problems.
5. Backward Pass (Backpropagation):
    - The backward pass is the core of the training process in deep learning. It involves calculating gradients of the loss function with respect to the model's parameters (weights and biases) using the chain rule of calculus.
    - The gradients indicate the direction and magnitude of the changes needed in each parameter to reduce the loss. These gradients are computed layer by layer, starting from the output layer and propagating backward to the input layer.
6. Parameter Update (Optimization):
    - After computing the gradients, an optimization algorithm is used to update the model's parameters to minimize the loss function. The optimization algorithm adjusts the model's parameters by subtracting a fraction of the gradients from the current values, effectively moving the model in the direction of lower loss.

7. Training Loop:
    - The training process proceeds through multiple iterations called epochs. In each epoch, the entire training dataset is passed through the model, and the gradients and parameter updates are computed and applied.
    - The model is trained for a fixed number of epochs or until a stopping criterion is met (e.g. convergence of the loss function or a predefined maximum number of epochs).
8. Validation:
    - Throughout the training process, the model's performance is evaluated on a separate validation dataset, which the model has not seen during training. This allows monitoring of the model's generalization to unseen data and helps in preventing overfitting (where the model performs well on the training data but poorly on new data).
9. Hyperparameter Tuning:
    - Deep learning models have hyperparameters that are not learned during training but need to be set before training. These include learning rate (controls how quickly or slowly the model adapts and converges to a local minimum), batch size, number of layers, number of neurons in each layer, etc.
    - Hyperparameter tuning involves finding the best combination of hyperparameter values that lead to optimal model performance. This is often done using techniques like grid search, random search, or Bayesian optimization.
10. Test:
    - Once the model is trained and validated, it is tested on a separate test dataset that has not been used during training or validation. The test dataset provides a final evaluation of the model's performance and gives an estimate of how well the model will perform in the real world.

The training process is an iterative and resource-intensive task. It requires a significant amount of computational power and is often performed on GPUs or specialized hardware to speed up the training process. Additionally, the success of deep learning training also depends on the availability of a large and diverse dataset, as well as careful model architecture design and hyperparameter tuning.

Appendix 4. YOLOv4-tiny and YOLOv8 architectures

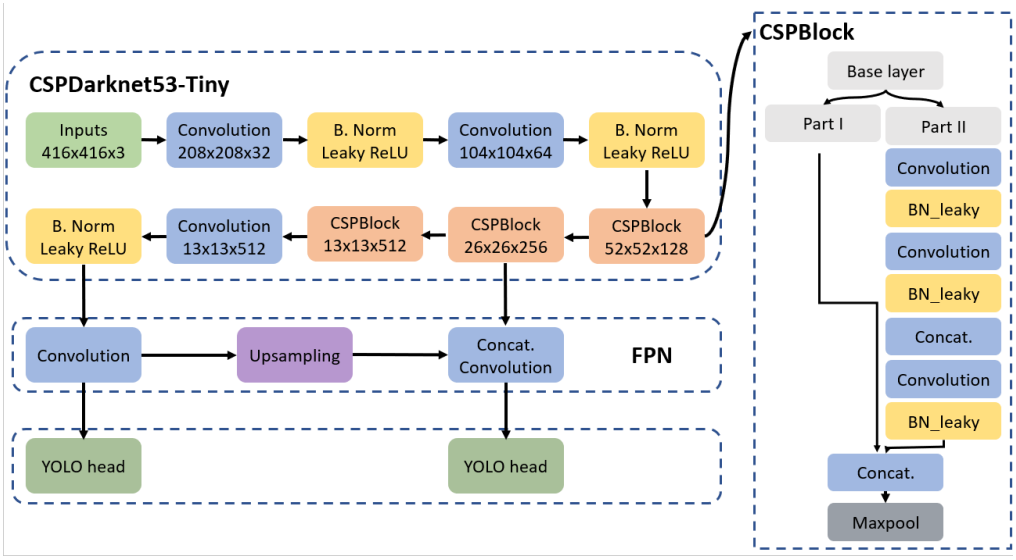

Figure 56. YOLOv4-tiny architecture using CSPDarknet53 as the base network (Saponara et al., 2022). CSP stands for "Cross-Stage Partial Networks", a convolutional neural network architecture used to enhance object detection performance in the YOLOv4 model. CSP aim is to divide the network into two branches so that information flow through these branches in a parallel manner, allowing for feature extraction at different spatial scales. The "head" is responsible for generating the final predictions of the model.

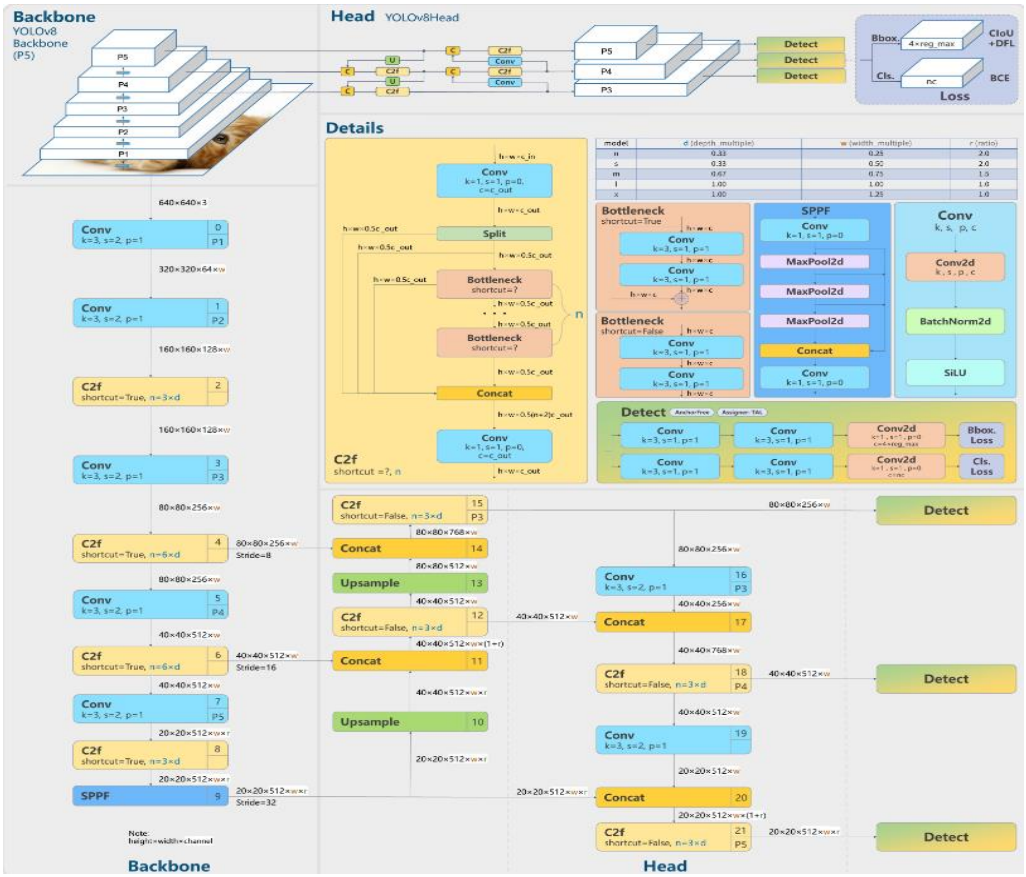

Figure 57. Model structure of YOLOv8 detection models (P5) - YOLOv8n (nano)/s (small)/m (medium)/l (large)/x (extra-large). (Range, 2023)

Appendix 5. Segmentation of the symptomatic shoots of FD by structure tensor

This aims to propose a segmentation method to isolate symptomatic shoots or pieces of symptomatic shoots of FD. A descriptor is proposed to distinguish the pixels of symptomatic shoots (of green colour) from other pixels of similar colour, such as those of petioles, leaves, and potentially present grass. This technique was successfully used by Abdelghafour *et al.* (2019) for grapevine image segmentation. It is known as the structure tensor.

The structure tensor is a mathematical representation extensively used in image processing and computer vision to dissect the local characteristics and orientations of features within an image region. It serves as a fundamental tool for tasks like edge and corner detection, as well as texture analysis. Its main advantage is its rapid calculation speed compared to deep segmentation algorithms. First, the gradient of the image is calculated. The gradient provides information about the direction and magnitude of the image intensity changes. For each pixel in the image, a small local window (also called a neighbourhood) is defined around it. The size of this window determines the scale at which the structure tensor is computed. The gradient values within this window are used to calculate the structure tensor. The structure tensor is a 2*2 matrix that is computed using the gradient values within the local window. The components of the structure tensor are calculated by taking outer products of the gradient vectors. The structure tensor matrix can be represented as in equation 14:

$$S = \begin{bmatrix} I_{x^2} & I_x I_y \\ I_x I_y & I_{y^2} \end{bmatrix} \quad (14)$$

where $\mathrm{I_x}$ is the gradient in the x-direction, $\mathrm{I_y}$ is the gradient in the y-direction.

Smoothing can be applied here to mitigate potential noise stemming from the discretized pixel effect. The eigenvalues and eigenvectors of the averaged structure tensor are computed. These eigenvalues represent the magnitude of the local intensity variations along the dominant and perpendicular directions within the local windows. The eigenvectors provide the corresponding directions of these variations. The orientation of the image features can be estimated using the eigenvector associated with the larger eigenvalue. The anisotropy (how elongated or stretched the features are) can be derived from the ratio of the eigenvalues as shown in equation 15:

$$Anisotropy = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \quad (15)$$

with λ_1 and λ_2 the eigenvalues of S. This value, between 0 and 1, is very close to 1 when the gradient has a preferred direction.

In the present work, attempts have been done to use the anisotropy of the tensor to detect and distinguish certain symptoms with specific shapes and textures. A technique called hysteresis thresholding (Pridmore, 2001) was subsequently used to improve the segmentation results. Hysteresis thresholding, is a technique used in image processing to improve the accuracy of edge detection, particularly in the presence of noise or variations in intensity. The hysteresis thresholding method involves using two threshold values: a high threshold (T_high) and a low threshold (T_low). The main idea behind hysteresis thresholding is to establish a threshold range for potential edge pixels, allowing for more flexibility in distinguishing true edges from noise. The gradient magnitude values are compared with the high threshold (T_high). Pixels with gradient magnitudes above T_high are considered strong edge candidates, while those below T_high are either discarded or marked as

potential weak edges. Pixels with gradient magnitudes between T_low and T_high are considered weak edge candidates. These pixels are not immediately classified as edges but are kept for further evaluation. The connected regions of weak edge candidates are tracked to determine if they are part of a continuous edge structure. If a connected region of weak edges is found to be connected to a strong edge (above T_high), it is promoted to a strong edge. If a weak edge region is not connected to any strong edge, it is discarded as noise.

By applying hysteresis thresholding on the anisotropy of the tensor, the edge detection process becomes more adaptable to variations in edge strength and noisy conditions. The high threshold helps in identifying strong edges, and the low threshold prevents the elimination of weaker edges that are still part of meaningful structures.

Finally, connected components detection was carried out to isolate objects (groups of pixels) that may constitute symptomatic shoots. Morphological operations were tested to compare the detections with the expected shape of the desired symptom (length-to-width ratio) and retain only the corresponding detections.

A comparison was conducted between the structure tensor and ResUNet algorithms for the segmentation of symptomatic shoots of FD. The results are provided in Table 60.

Table 60. Segmentation results obtained for symptomatic shoots at pixel and object scales, using both the structure tensor and the ResUNet methodologies (see Chapter V).

Algorithm	Input size	Pixel precision	Pixel recall	Object precision	Object recall
Structure tensor	2048*2448*3	0.19	0.24	0.12	0.15
Structure tensor + hysteresis threshold	2048*2448*3	0.17	0.44	0.08	0.59
Structure tensor + hysteresis threshold + morphological operation	2048*2448*3	0.7	0.28	0.66	0.42
ResUNEt	2048*2448*3	0.53	0.59	0.74	0.52
ResUNEt	512*512*3	0.64	0.53	0.76	0.58
ResUNEt	256*256*3	**0.69**	**0.58**	**0.82**	**0.59**

The parameters of the structure tensor were the following: derivative filter: sigma = 2. Gaussian smoothing: sigma = 8. Threshold by hysteresis: low threshold = 0.5, high threshold= 0.9. Morphological operations: only the detections verifying: length of the major axis > 90 pixels, 6 pixels < length of the minor axis < 17 pixels were retained. The ResUnet algorithm reached the best results for the Tversky loss function (those presented in this table), five levels of depth, $\alpha = \beta = 0.8$ for batches of 30 images during 300 epochs.

Concerning the structure tensor results, it is evident that the initial outcomes are quite poor, both in relation to the pixel metric (p=0.19, r=0.24) and the object metric (p=0.12, r=0.15). The hysteresis threshold facilitates the retrieval of more pixels from symptomatic shoots at the expense of precision (p=0.08 and r=0.59 for the object metric). Morphological operations enhance the results by greatly improving precision, albeit at the cost of reduced recall (p=0.66 and r=0.42 for the object metric). Figure 58 displays an example of the results of each of these steps.

The ResUNet yielded superior results. Optimal performance was attained by employing the 256*256*3 patches (512*512*3 image patches with downsizing the resolution of each patch by a factor of 4) with p=0.82 and r=0.59 for the object metric. This technique significantly bolsters precision during result comparisons: for the object metric, p=0.74 for the whole images as input and p=0.76 for the patches 512*512*3 as input. This phenomenon stems from the fact that petioles (leaf stems), resembling shoots in their elongated and green form, are frequently predicted as syptomatic shoots by the algorithm as they naturally appear. Nevertheless, the resolution reduction diminishes their

distinctiveness in the images, given their slender nature, resulting in fewer wrong predictions by the algorithm. A comparison of the predictions of the structure tensor and the ResUNet is presented in Figure 59.

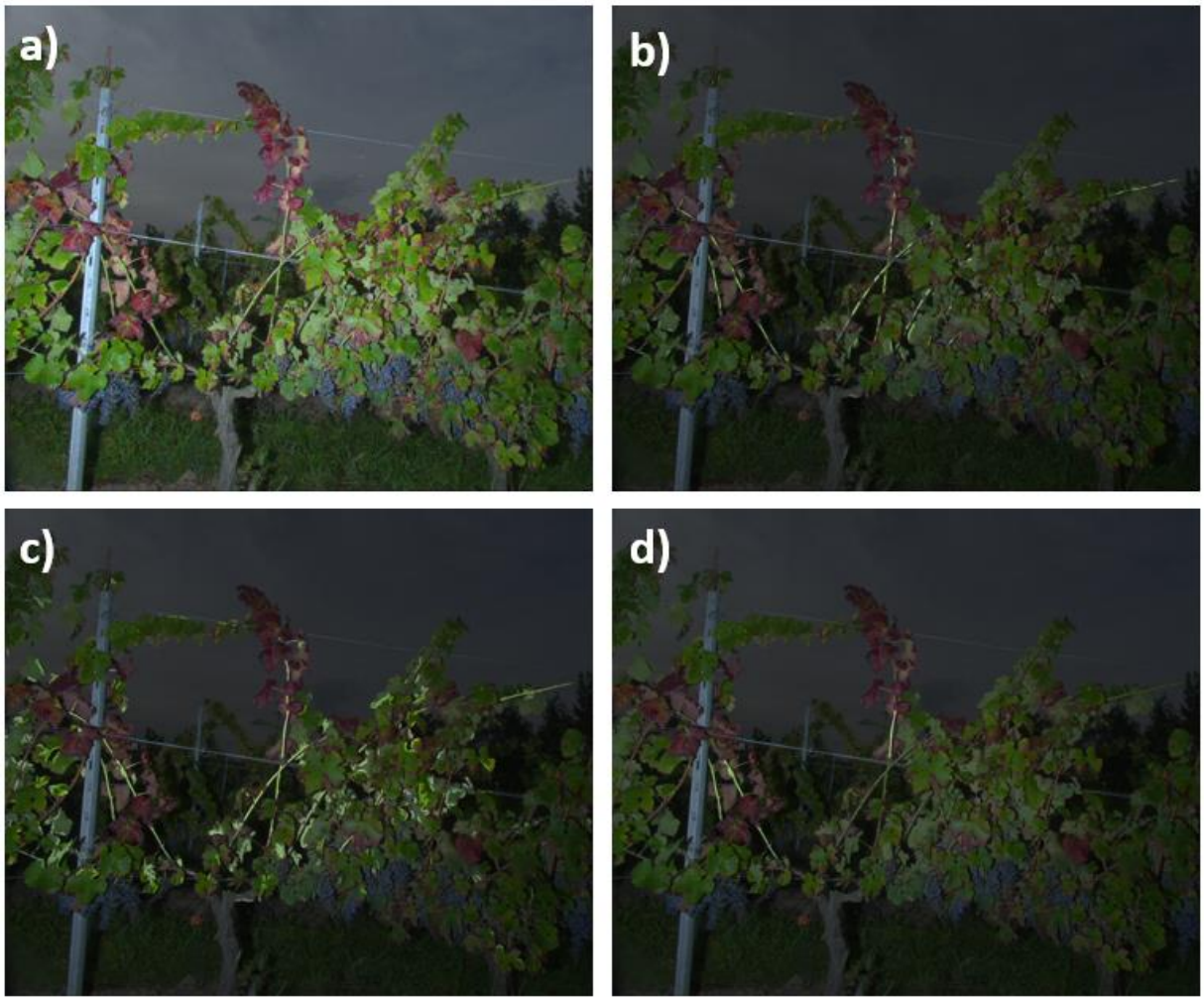

Figure 58. Images of the different steps of the structure tensor algorithm. a) Original image. Original image overlayed with: b) threshold on the anisotropy, c) threshold on the anisotropy with hysteresis treshold, d) threshold on the anisotropy with hysteresis threshold and morphological operations.

It can be observed in Figure 58 that simple thresholding on anisotropy produced false positives almost everywhere, and the detected shoots were not complete. The false positives were exacerbated by hysteresis thresholding, but more shoot pixels were recovered through this operation. Finally, thanks to morphological operations, many false detections were removed at the cost of removing many good detections as well.

Figure 59. Images comparing predictions for symptomatic FD shoots between the structure tensor and ResUNet. First column: Raw image. Second column: Predictions by the structure tensor (with hysteresis thresholding and morphological parameters). Third column: predictions by the ResUnet.

In the first two images of Figure 59 (the first two rows), it can be observed that the shoots predicted by the ResUNet algorithm are generally more complete than those predicted by the structure tensor. Additionally, the last example (bottom row) highlights the better precision of the ResUNet algorithm, with significantly fewer false positives.

Appendix 6. Results of the RF methodology for the 'FD' class depending on the hyperparameter combinations

Table 61. Comparison of the results in precision and recall of the RF for the 'FD' class on the 'ClaFD' dataset using the YOLOv8all vector for all the hyperparameter MS, MF and NDT combinations.

MS (%)	MF	NDT	FD precision	FD recall
0.5	1	50	0,82	0,96
0.5	1	100	0,8	0,97
0.5	1	500	0,8	1
0.5	2	50	0,84	1
0.5	2	100	0,84	1
0.5	2	500	0,82	0,99
0.5	3	50	0,8	0,99
0.5	3	100	0,83	0,99
0.5	3	500	0,83	0,99
0.6	1	50	0,78	0,99
0.6	1	100	0,81	0,99
0.6	1	500	0,8	1
0.6	2	50	0,81	0,99
0.6	2	100	0,81	0,99
0.6	2	500	0,82	0,99
0.6	3	50	0,81	0,99
0.6	3	100	0,82	1
0.6	3	500	0,81	0,99
0.7	1	50	0,79	0,99
0.7	1	100	0,81	1
0.7	1	500	0,81	1
0.7	2	50	0,8	0,97
0.7	2	100	0,8	0,99
0.7	2	500	0,82	0,99
0.7	3	50	0,82	0,99
0.7	3	100	0,82	0,96
0.7	3	500	0,84	0,97

MS, MF and NDT stand for the Maximum number of Samples, the Maximum number of Features and the Number of Decision trees respectively.

Appendix 7: Random Forest's importance scores of the parameters of the VectYOLOv8all vector for the FD study

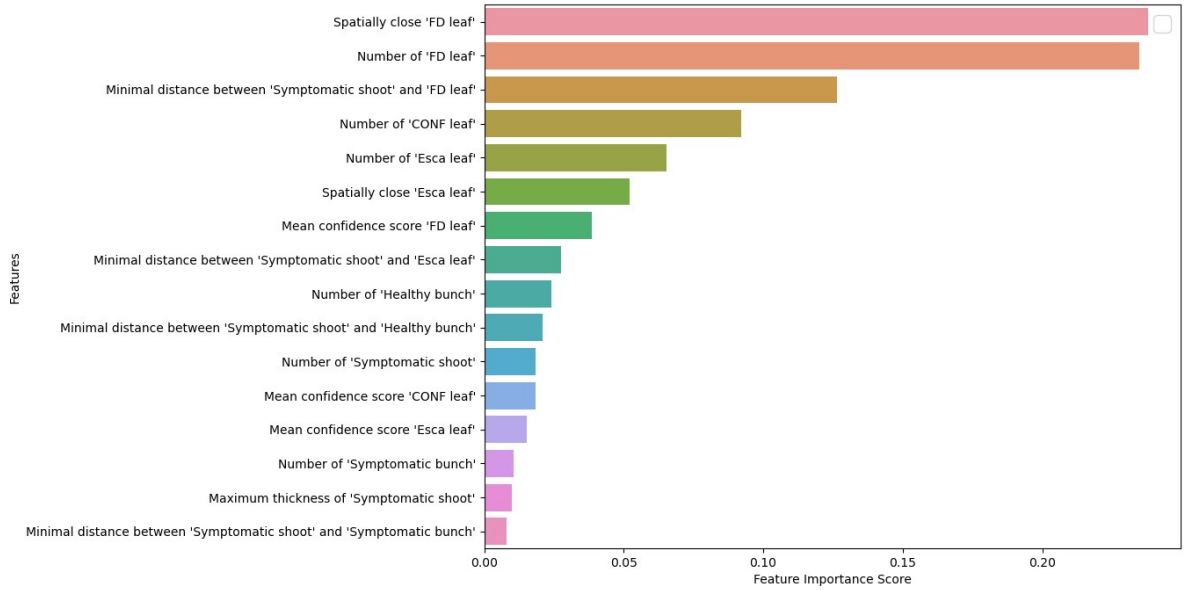

Figure 60. Importance scores of the parameters of the VectYOLOv8all vector for the training and testing of RF that achieved the best results on the 'UB20' set.

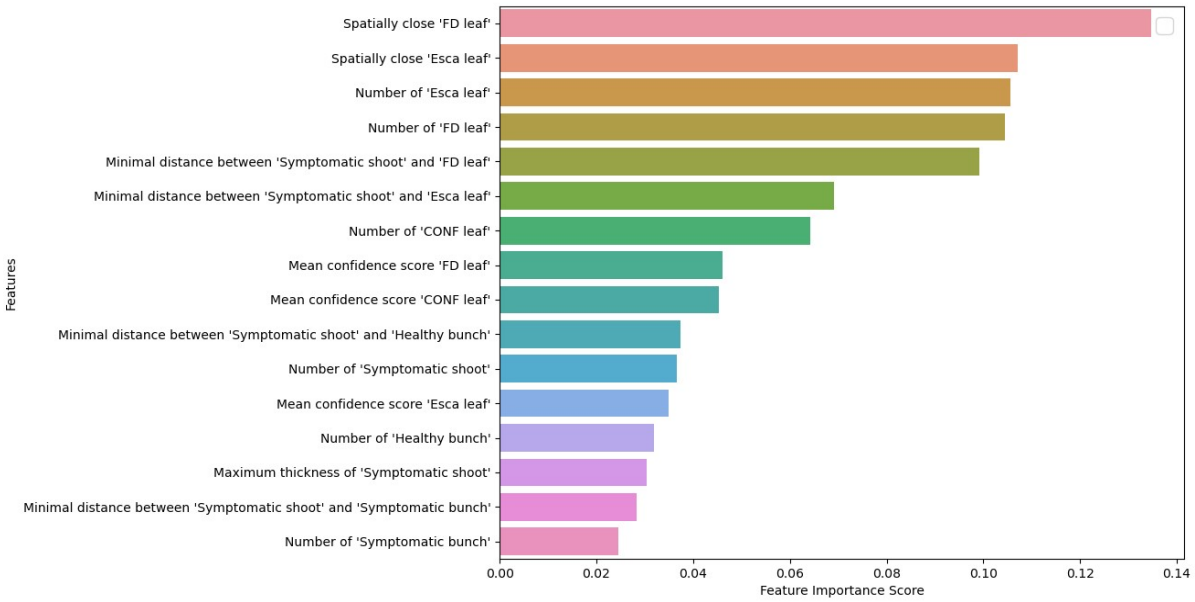

Figure 61. Importance scores of the parameters of the VectYOLOv8all vector for the training and testing of RF that achieved the best results on the 'CS20 + CF21 + M21 + CS21' set.

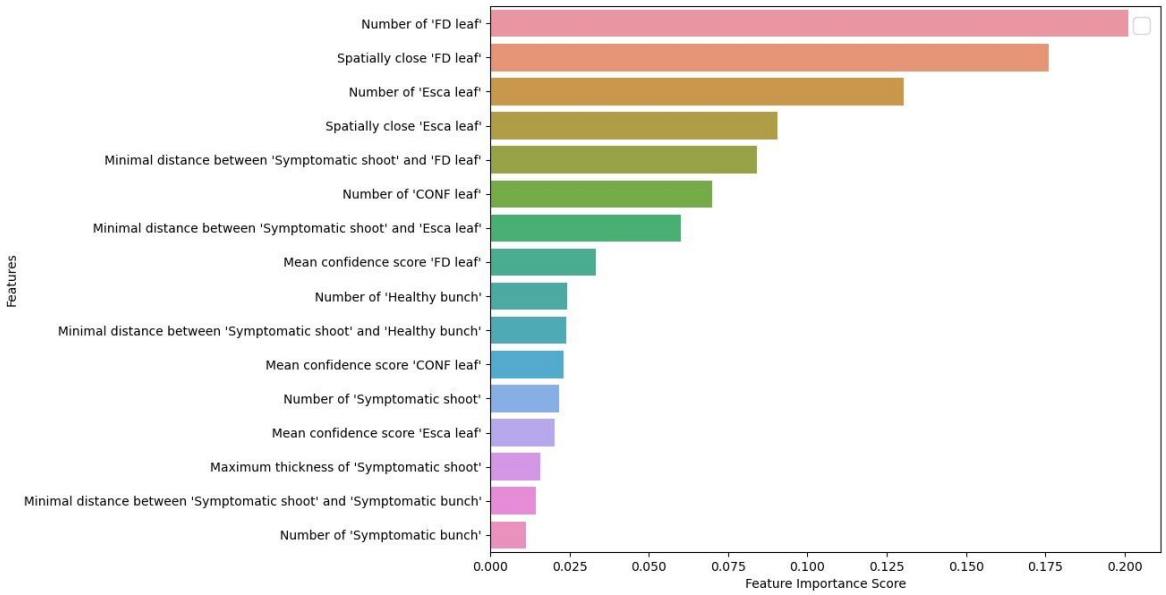

Figure 62. Importance scores of the parameters of the VectYOLOv8all vector for the training and testing of RF that achieved the best results on the 'UB20 + UB21 + SB21' set.

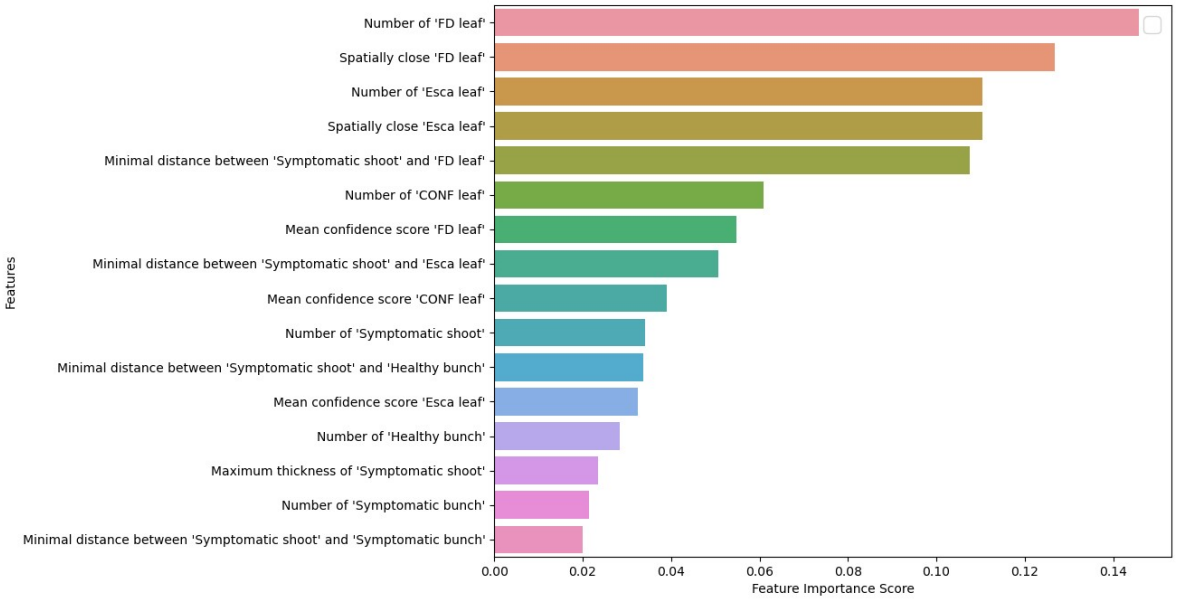

Figure 63. Importance scores of the parameters of the VectYOLOv8all vector for the training and testing of RF that achieved the best results on the 'ClaFD' set.

Author publications and communications.

Peer-reviewed international journals

Tardif, M., Amri, A., Keresztes, B., Deshayes, A., Martin, D., Greven, M., & Da Costa, J.-P. (2022). Two-stage automatic diagnosis of Flavescence Dorée based on proximal imaging and artificial intelligence: A multi-year and multi-variety experimental study. *OENO One*, *56*(3). https://doi.org/10.20870/oeno-one.2022.56.3.5460

Tardif, M., Amri, A., Deshayes, A., Greven, M., Keresztes, B., Fontaine, G., Sicaud, L., Paulhac, L., Bentejac, S., & Da Costa, J.-P. (2023). An expertized grapevine disease image database including five grape varieties focused on Flavescence dorée and its confounding diseases, biotic and abiotic stresses. *Data in Brief*, *48*. https://doi.org/10.1016/j.dib.2023.109230

Rançon, F., Keresztes, B., Deshayes, A., Tardif, M., Abdelghafour, F., Fontaine, G., Da Costa, J.-P., & Germain, C. (2023). Designing a Proximal Sensing Camera Acquisition System for Vineyard Applications: Results and Feedback on 8 Years of Experiments. *Sensors*, *23*(2). https://doi.org/10.3390/s23020847

Conferences with published communication proceedings

Tardif, M., Amri, A., Keresztes, B., Deshayes, A., Martin, D., Greven, M., & Da Costa, J.-P. (2023). Automatic diagnosis of a multi-symptom grape vine disease using computer vision. *Acta Horticulturae*, *1360*, 53-60. https://doi.org/10.17660/ActaHortic.2023.1360.7 - Article and oral presentation.

Tardif, M., Keresztes, B., Deshayes, A., Martin, D., Greven, M., & Da Costa, J.-P. (2023). 127. Automatic diagnosis of a multi-symptom grapevine disease by decision trees and Graph Neural Networks. In *Precision agriculture.* 1011-1017. Wageningen Academic Publishers. https://doi.org/10.3920/978-90-8686-947-3_127 - Article and oral presentation.

Communications without proceedings

Keresztes, B., Amri, A., Tardif, M., Deshayes, A., Da Costa, J.P., Greven, M., Martin, D. (2023). In-field detection of a multi-symptom grapevine disease by proximal sensing and artificial intelligence. KEPAF 2023, Gyula, Hungary.

Tardif, M. (2023). Automatic diagnosis of a multi-symptom grapevine disease by decision trees and Graph Neural Networks. Machine Learning Seminar, Waikato university, Hamilton, New Zealand. Invited speaker.

Tardif, M. (2023). Détection automatique d'une maladie multi-symptômes de la vigne par Graph Neural Network. Journée thématique GDR ISIS Signal/Image & Intelligence artificielle pour l'agriculture numérique. CNRS Ile-de-France Villejuif, Villejuif, France

References

Abdelghafour, F., Keresztes, B., Germain, C., & Da Costa, J.-P. (2020). In Field Detection of Downy Mildew Symptoms with Proximal Colour Imaging. *Sensors*, *20*(16). https://doi.org/10.3390/s20164380

Abdelghafour, F., Rosu, R., Keresztes, B., Germain, C., & Da Costa, J.-P. (2019). A Bayesian framework for joint structure and colour based pixel-wise classification of grapevine proximal images. *Computers and Electronics in Agriculture*, *158*, 345-357. https://doi.org/10.1016/j.compag.2019.02.017

Agarwal, V. (2020). Complete Architectural Details of all EfficientNet Models. *Medium*. https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142

Ajmal, H., Rehman, S., Farooq, U., Ain, Q. U., Riaz, F., & Hassan, A. (2018). Convolutional neural network based image segmentation : A review. *Pattern Recognition and Tracking XXIX*, *10649*, 191-203. https://doi.org/10.1117/12.2304711

Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description : A survey. *Data Mining and Knowledge Discovery*, *29*(3), 626-688. https://doi.org/10.1007/s10618-014-0365-y

Al Saddik, H. (2019). Spectral and textural analysis of high resolution data for the automatic detection of grape vine diseases. https://tel.archives-ouvertes.fr/tel-02408995

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1-6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Albetis, J., Duthoit, S., Guttler, F., Jacquin, A., Goulard, M., Poilvé, H., Féret, J.-B., & Dedieu, G. (2017). Detection of Flavescence dorée Grapevine Disease Using Unmanned Aerial Vehicle (UAV) Multispectral Imagery. *Remote Sensing*, *9*(4). https://doi.org/10.3390/rs9040308

Albetis, J., Jacquin, A., Goulard, M., Poilvé, H., Rousseau, J., Clenet, H., Dedieu, G., & Duthoit, S. (2018). On the Potentiality of UAV Multispectral Imagery to Detect Flavescence dorée and Grapevine Trunk Diseases. *Remote Sensing*, *11*(1). https://doi.org/10.3390/rs11010023

Ali, L., Alnajjar, F., Jassmi, H. A., Gocho, M., Khan, W., & Serhani, M. A. (2021). Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. *Sensors*, *21*(5). https://doi.org/10.3390/s21051688

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. 6077-6086. https://openaccess.thecvf.com/content_cvpr_2018/html/Anderson_Bottom-Up_and_Top-Down_CVPR_2018_paper.html

Atallah, S. S., Gómez, M. I., Fuchs, M. F., & Martinson, T. E. (2011). Economic Impact of Grapevine Leafroll Disease on Vitis vinifera cv. Cabernet franc in Finger Lakes Vineyards of New York. *American Journal of Enology and Viticulture*. https://doi.org/10.5344/ajev.2011.11055

Babcsányi, I., Chabaux, F., Granet, M., Meite, F., Payraudeau, S., Duplay, J., & Imfeld, G. (2016). Copper in soil fractions and runoff in a vineyard catchment : Insights from copper stable isotopes. *The Science of the Total Environment*, 154-162. https://doi.org/10.1016/j.scitotenv.2016.03.037

Barthellet, B., Goglia, R., & Groman, J. (2018). Flavescence dorée, bilan de la surveillance en 2018. Ministère de l'agriculture et de l'alimentation. http://draaf.auvergne-rhone-

alpes.agriculture.gouv.fr/IMG/pdf/1-Bilan_2018_Flavescence_doree-V-def-pub_2_cle0f2948.pdf

Baša Česnik, H., Gregorčič, A., & čuš, F. (2008). Pesticide residues in grapes from vineyards included in integrated pest management in Slovenia. *Food Additives & Contaminants: Part A*, *25*(4), 438-443. https://doi.org/10.1080/02652030701558490

Bavaresco, L. (2019). Impact of grapevine breeding for disease resistance on the global wine industry. *Acta Horticulturae*, *1248*, 7-14. https://doi.org/10.17660/ActaHortic.2019.1248.2

Bendel, N., Kicherer, A., Backhaus, A., Klück, H.-C., Seiffert, U., Fischer, M., Voegele, R. T., & Töpfer, R. (2020). Evaluating the suitability of hyper- and multispectral imaging to detect foliar symptoms of the grapevine trunk disease Esca in vineyards. *Plant Methods*, *16*(1), 142. https://doi.org/10.1186/s13007-020-00685-3

Berry, J., & Bjorkman, O. (1980). Photosynthetic Response and Adaptation to Temperature in Higher Plants. *Annual Review of Plant Physiology*, *31*(1), 491-543. https://doi.org/10.1146/annurev.pp.31.060180.002423

Bhagat, S., Cormode, G., & Muthukrishnan, S. (2011). Node Classification in Social Networks. *Social Network Data Analytics*, 115-148. Springer US. https://doi.org/10.1007/978-1-4419-8462-3_5

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). CNN Variants for Computer Vision : History, Architecture, Application, Challenges and Future Scope. *Electronics*. https://doi.org/10.3390/electronics10202470

Bianchi, F. M., Grattarola, D., & Alippi, C. (2020). Spectral Clustering with Graph Neural Networks for Graph Pooling. *arXiv:1907.00481*. https://doi.org/10.48550/arXiv.1907.00481

Bo, D. (2023). Homogeneous Graph Neural Networks. In *Advances in Graph Neural Networks*, 27-59. Springer International Publishing. https://doi.org/10.1007/978-3-031-16174-2_3

Bocca, F., Picciau, L., & Alma, A. (2020). New insights on Scaphoideus titanus biology and their implications for integrated pest management. *Entomologia Generalis*, *40*. https://doi.org/10.1127/entomologia/2020/0977

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4 : Optimal Speed and Accuracy of Object Detection. *arXiv:2004.10934*. http://arxiv.org/abs/2004.10934

Bois, B., Zito, S., & Calonnec, A. (2017). Climate vs grapevine pests and diseases worldwide : The first results of a global survey. *OENO One*, *51*(2). https://doi.org/10.20870/oeno-one.2017.51.2.1780

Boulent, J., Beaulieu, M., St-Charles, P.-L., Théau, J., & Foucher, S. (2019). Deep learning for in-field image-based grapevine downy mildew identification. 141-148. https://doi.org/10.3920/978-90-8686-888-9_16

Boulent, J., St-Charles, P.-L., Foucher, S., & Théau, J. (2020). Automatic Detection of Flavescence Dorée Symptoms Across White Grapevine Varieties Using Deep Learning. *Frontiers in Artificial Intelligence*, *3*. https://www.frontiersin.org/articles/10.3389/frai.2020.564878

Bourgeon, M.-A. (2015). Conception et évaluation d'un dispositif d'imagerie multispectrale de proxidétection embarqué pour caractériser le feuillage de la vigne. https://tel.archives-ouvertes.fr/tel-01291695

Budde, M. D., & Frank, J. A. (2012). Examining brain microstructure using structure tensor analysis of histological sections. *NeuroImage*, *63*(1). https://doi.org/10.1016/j.neuroimage.2012.06.042

Buonassisi, D., Colombo, M., Migliaro, D., Dolzani, C., Peressotti, E., Mizzotti, C., Velasco, R., Masiero, S., Perazzolli, M., & Vezzulli, S. (2017). Breeding for grapevine downy mildew resistance :

A review of “omics” approaches. *Euphytica*, *213*(5). https://doi.org/10.1007/s10681-017-1882-8

Burr, T. J., & Otten, L. (1999). Crown gall of grape : Biology and Disease Management. *Annual Review of Phytopathology*, *37*(1), 53-80. https://doi.org/10.1146/annurev.phyto.37.1.53

Caffi, T., Gilardi, G., Monchiero, M., & Rossi, V. (2013). Production and Release of Asexual Sporangia in Plasmopara viticola. *Phytopathology*, *103*(1), 64-73. https://doi.org/10.1094/PHYTO-04-12-0082-R

Carter, M. V. (1991). The status of Eutypa lata as a pathogen. The Status of Eutypa Lata as a Pathogen. https://www.cabdirect.org/cabdirect/abstract/19912304635

Chamberlain, G. C., Willison, R. S., Townshend, J. L., & Ronde, J. H. de. (1964). Two fungi associated with the dead-arm disease of grapes. *Canadian Journal of Botany*, *42*(4), 351-355. https://doi.org/10.1139/b64-034

Charters, S. (2006). *Wine and Society*. Routledge.

Chen, K., Wang, J., Chen, L.-C., Gao, H., Xu, W., & Nevatia, R. (2016). ABC-CNN : An Attention Based Convolutional Neural Network for Visual Question Answering. *arXiv:1511.05960*. https://doi.org/10.48550/arXiv.1511.05960

Christen, D., Schönmann, S., Jermini, M., Strasser, R. J., & Défago, G. (2007). Characterization and early detection of grapevine (Vitis vinifera) stress responses to esca disease by in situ chlorophyll fluorescence and comparison with drought stress. *Environmental and Experimental Botany*, *60*(3), 504-514. https://doi.org/10.1016/j.envexpbot.2007.02.003

Chuche, J., & Thiéry, D. (2014). Biology and ecology of the Flavescence dorée vector Scaphoideus titanus : A review. *Agronomy for Sustainable Development*, *34*(2), 381-403. https://doi.org/10.1007/s13593-014-0208-7

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273-297. https://doi.org/10.1007/BF00994018

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964

Cubero, S., Aleixos, N., Moltó, E., Gómez-Sanchis, J., & Blasco, J. (2011). Advances in Machine Vision Applications for Automatic Inspection and Quality Evaluation of Fruits and Vegetables. *Food and Bioprocess Technology*, *4*(4), 487-504. https://doi.org/10.1007/s11947-010-0411-8

Cunningham, P., & Delany, S. (2007). K-Nearest neighbour classifiers. *Mult Classif Syst*, *54*. https://doi.org/10.1145/3459665

Curl, C. L., Spivak, M., Phinney, R., & Montrose, L. (2020). Synthetic Pesticides and Health in Vulnerable Populations : Agricultural Workers. *Current Environmental Health Reports*, *7*(1), 13-29. https://doi.org/10.1007/s40572-020-00266-5

Daglio, G., Cesaro, P., Todeschini, V., Lingua, G., Lazzari, M., Berta, G., & Massa, N. (2022). Potential field detection of Flavescence dorée and Esca diseases using a ground sensing optical system. *Biosystems Engineering*, *215*, 203-214. https://doi.org/10.1016/j.biosystemseng.2022.01.009

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 886-893. https://doi.org/10.1109/CVPR.2005.177

Davis, M. J., Purcell, A. H., & Thomson, S. V. (1978). Pierce's Disease of Grapevines : Isolation of the Causal Bacterium. *Science*, *199*(4324), 75-77. https://doi.org/10.1126/science.199.4324.75

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

Deutsch, C. A., Tewksbury, J. J., Tigchelaar, M., Battisti, D. S., Merrill, S. C., Huey, R. B., & Naylor, R. L. (2018). Increase in crop losses to insect pests in a warming climate. *Science (New York, N.Y.)*, *361*(6405), 916-919. https://doi.org/10.1126/science.aat3466

Diakogiannis, F. I., Waldner, F., Caccetta, P., & Wu, C. (2020). ResUNet-a : A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *162*, 94-114. https://doi.org/10.1016/j.isprsjprs.2020.01.013

Droulia, F., & Charalampopoulos, I. (2021). Future Climate Change Impacts on European Viticulture : A Review on Recent Scientific Advances. *Atmosphere*, *12*(4). https://doi.org/10.3390/atmos12040495

Droulia, F., & Charalampopoulos, I. (2022). A Review on the Observed Climate Change in Europe and Its Impacts on Viticulture. *Atmosphere*, *13*(5). https://doi.org/10.3390/atmos13050837

Duda, R. O., & Hart, P. E. (1974). Pattern classification and scene analysis. *A Wiley-Interscience publication*.

EFSA Panel on Plant Health (PLH), Jeger, M., Bragard, C., Caffier, D., Candresse, T., Chatzivassiliou, E., Dehnen-Schmutz, K., Gilioli, G., Jaques Miret, J. A., MacLeod, A., Navajas Navarro, M., Niere, B., Parnell, S., Potting, R., Rafoss, T., Rossi, V., Urek, G., Van Bruggen, A., Van Der Werf, W., Grégoire, J.-C. (2016). Risk to plant health of Flavescence dorée for the EU territory. *EFSA Journal*, *14*(12). https://doi.org/10.2903/j.efsa.2016.4603

Essakhi, S., Mugnai, L., Crous, P., Groenewald, J. Z., & Surico, G. (2008). Molecular and phenotypic characterization of novel Phaeoacremonium species associated with Petri disease and esca of grapevine. *Persoonia*, *21*, 119-134. https://doi.org/10.3767/003158508X374385

Fernández-Calviño, D., Nóvoa-Muñoz, J. C., Díaz-Raviña, M., & Arias-Estévez, M. (2009). Copper accumulation and fractionation in vineyard soils from temperate humid zone (NW Iberian Peninsula). *Geoderma*, *153*(1), 119-129. https://doi.org/10.1016/j.geoderma.2009.07.024

Fernández-Calviño, D., Soler-Rovira, P., Polo, A., Díaz-Raviña, M., Arias-Estévez, M., & Plaza, C. (2010). Enzyme activities in vineyard soils long-term treated with copper-based fungicides. *Soil Biology and Biochemistry*, *42*(12), 2119-2127. https://doi.org/10.1016/j.soilbio.2010.08.007

Fontaine, F., Gramaje, D., Armengol, J., Smart, R., Nagy, Z. A., Borgo, M., Rego, C., & Corio-Costet, M.-F. (2016). *Grapevine trunk diseases. A review*. OIV Publications. https://hal.science/hal-01604038

Fontaine, M. C., Labbé, F., Dussert, Y., Delière, L., Richart-Cervera, S., Giraud, T., & Delmotte, F. (2021). Europe as a bridgehead in the worldwide invasion history of grapevine downy mildew, Plasmopara viticola. *Current Biology*, *31*(10), 2155-2166. https://doi.org/10.1016/j.cub.2021.03.009

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367-378. https://doi.org/10.1016/S0167-9473(01)00065-2

Fuentes, A. F., Yoon, S., Lee, J., & Park, D. S. (2018). High-Performance Deep Neural Network-Based Tomato Plant Diseases and Pests Diagnosis System With Refinement Filter Bank. *Frontiers in Plant Science*, *9*. https://doi.org/10.3389/fpls.2018.01162

Fuhrimann, S., van den Brenk, I., Atuhaire, A., Mubeezi, R., Staudacher, P., Huss, A., & Kromhout, H. (2022). Recent pesticide exposure affects sleep : A cross-sectional study among smallholder farmers in Uganda. *Environment International*, *158*. https://doi.org/10.1016/j.envint.2021.106878

Gadoury, D. M., Cadle-Davidson, L., Wilcox, W. F., Dry, I. B., Seem, R. C., & Milgroom, M. G. (2012). Grapevine powdery mildew (Erysiphe necator) : A fascinating system for the study of the biology, ecology and epidemiology of an obligate biotroph. *Molecular Plant Pathology*, *13*(1), 1-16. https://doi.org/10.1111/j.1364-3703.2011.00728.x

Garcia Duran, A., & Niepert, M. (2017). Learning Graph Representations with Embedding Propagation. *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/e0688d13958a19e087e123148555e4b4-Paper.pdf

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A Review on Deep Learning Techniques Applied to Semantic Segmentation.

Geng, Q., Zhou, Z., & Cao, X. (2017). Survey of recent progress in semantic image segmentation with CNNs. *Science China Information Sciences*, *61*(5). https://doi.org/10.1007/s11432-017-9189-6

Gessler, C., Pertot, I., & Perazzolli, M. (2011). Plasmopara viticola : A review of knowledge on downy mildew of grapevine and effective disease management. *Phytopathologia Mediterranea*, *50*(1), 3-44.

Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Gutiérrez-Gamboa, G., Zheng, W., & Martínez de Toda, F. (2021). Current viticultural techniques to mitigate the effects of global warming on grape and wine quality : A comprehensive review. *Food Research International*, *139*. https://doi.org/10.1016/j.foodres.2020.109946

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1025-1035.

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv:1512.03385*. https://doi.org/10.48550/arXiv.1512.03385

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. https://doi.org/10.1109/CVPR.2016.90

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd international conference on document analysis and recognition*, *1*, 278-282.

Holman, F. H., Riche, A. B., Michalski, A., Castle, M., Wooster, M. J., & Hawkesford, M. J. (2016). High Throughput Field Phenotyping of Wheat Plant Height and Growth Rate in Field Plot Trials Using UAV Based Remote Sensing. *Remote Sensing*, *8*(12). https://doi.org/10.3390/rs8121031

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets : Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861.* http://arxiv.org/abs/1704.04861

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. *arXiv:1905.02244*. http://arxiv.org/abs/1905.02244

Hren, M., Boben, J., Rotter, A., Kralj, P., Gruden, K., & Ravnikar, M. (2007). Real-time PCR detection systems for Flavescence dorée and Bois noir phytoplasmas in grapevine : Comparison with conventional PCR detection and application in diagnostics. *Plant Pathology*, *56*(5), 785-796. https://doi.org/10.1111/j.1365-3059.2007.01688.x

Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020). Heterogeneous Graph Transformer. *Proceedings of The Web Conference 2020*, 2704-2710. https://doi.org/10.1145/3366423.3380027

International Organisation of Vine and Wine (2023). State of the world vine and wine sector in 2022. https://www.oiv.int/sites/default/files/documents/OIV_State_of_the_world_Vine_and_Wine_sector_in_2022_2.pdf

IPCC (2022). Climate Change 2022 : Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press. https://doi.org/10.1017/9781009157926

Isaacs, R., Vincent, C., & Bostanian, N. J. (2012). Vineyard IPM in a Changing World : Adapting to New Pests, Tactics, and Challenges. *Arthropod Management in Vineyards : Pests, Approaches, and Future Directions*, 475-484. https://doi.org/10.1007/978-94-007-4032-7_20

Ji, M., Zhang, L., & Wu, Q. (2020). Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. *Information Processing in Agriculture*, *7*(3), 418-426. https://doi.org/10.1016/j.inpa.2019.10.003

Jmour, N., Zayen, S., & Abdelkrim, A. (2018). Convolutional neural networks for image classification. *2018 International Conference on Advanced Systems and Electric Technologies*, 397-402. https://doi.org/10.1109/ASET.2018.8379889

Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics YOLOv8*. https://github.com/ultralytics/ultralytics

Johannes, A., Picon, A., Alvarez-Gila, A., Echazarra, J., Rodriguez-Vaamonde, S., Navajas, A. D., & Ortiz-Barredo, A. (2017). Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. *Computers and Electronics in Agriculture*, *138*, 200-209. https://doi.org/10.1016/j.compag.2017.04.013

Jurado, J. M., Pádua, L., Feito, F. R., & Sousa, J. J. (2020). Automatic Grapevine Trunk Detection on UAV-Based Point Cloud. *Remote Sensing*, *12*(18). https://doi.org/10.3390/rs12183043

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture : A survey. *Computers and Electronics in Agriculture*, *147*, 70-90. https://doi.org/10.1016/j.compag.2018.02.016

Kaur, R., & Singh, S. (2023). A comprehensive review of object detection with deep learning. *Digital Signal Processing*, *132*. https://doi.org/10.1016/j.dsp.2022.103812

Keesstra, S., Geissen, V., Mosse, K., Piiranen, S., Scudiero, E., Leistra, M., & van Schaik, L. (2012). Soil as a filter for groundwater quality. *Current Opinion in Environmental Sustainability*, *4*(5), 507-516. https://doi.org/10.1016/j.cosust.2012.10.007

Kerkech, M., Hafiane, A., & Canals, R. (2018). Deep learning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Computers and Electronics in Agriculture*, *155*, 237-243. https://doi.org/10.1016/j.compag.2018.10.006

Kerkech, M., Hafiane, A., & Canals, R. (2020a). VddNet : Vine Disease Detection Network Based on Multispectral Images and Depth Map. *Remote Sensing*, *12*(20). https://doi.org/10.3390/rs12203305

Kerkech, M., Hafiane, A., & Canals, R. (2020b). Vine disease detection in UAV multispectral images with deep learning segmentation approach. *Computers and Electronics in Agriculture*, *174*. https://doi.org/10.1016/j.compag.2020.105446

Khan, N., Kennedy, A., Cotton, J., & Brumby, S. (2019). A Pest to Mental Health? Exploring the Link between Exposure to Agrichemicals in Farmers and Mental Health. *International Journal of Environmental Research and Public Health*, *16*(8). https://doi.org/10.3390/ijerph16081327

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in Vision : A Survey. *ACM Computing Surveys*, *54*(10), 201-241. https://doi.org/10.1145/3505244

Kriege, N. M., Johansson, F. D., & Morris, C. (2020). A survey on graph kernels. *Applied Network Science*, *5*(1). https://doi.org/10.1007/s41109-019-0195-3

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, *25*. https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

Larignon, P., & Dubos, B. (1997). Fungi associated with esca disease in grapevine. *European Journal of Plant Pathology*, *103*(2), 147-157. https://doi.org/10.1023/A:1008638409410

Larignon, P., Spagnolo, A., Bertsch, C., & Fontaine, F. (2015). First Report of Young Grapevine Decline Caused by Neofusicoccum parvum in France. *Plant Disease*, *99*(12). https://doi.org/10.1094/PDIS-03-15-0280-PDN

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553). https://doi.org/10.1038/nature14539

Leeuwen, C. van, & Darriet, P. (2016). The Impact of Climate Change on Viticulture and Wine Quality. *Journal of Wine Economics*, *11*(1), 150-167. https://doi.org/10.1017/jwe.2015.21

Lefol, C., Caudwell, A., Lherminier, J., & Larrue, J. (1993). Attachment of the Flavescence doree pathogen (MLO) to leafhopper vectors and other insects. *Annals of Applied Biology*, *123*(3), 611-622. https://doi.org/10.1111/j.1744-7348.1993.tb04931.x

Li, X., Lv, C., Wang, W., Li, G., Yang, L., & Yang, J. (2022). Generalized Focal Loss : Towards Efficient Representation Learning for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1-14. https://doi.org/10.1109/TPAMI.2022.3180392

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal Loss for Dense Object Detection. *arXiv:1708.02002*. https://doi.org/10.48550/arXiv.1708.02002

Liu, B., Ding, Z., Tian, L., He, D., Li, S., & Wang, H. (2020). Grape Leaf Disease Identification Using Improved Deep Convolutional Neural Networks. *Frontiers in Plant Science*, *11*. https://doi.org/10.3389/fpls.2020.01082

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD : Single Shot MultiBox Detector. *Computer Vision – ECCV 2016*, 21-37. Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2

Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully Convolutional Networks for Semantic Segmentation* . *arXiv:1411.4038*. https://doi.org/10.48550/arXiv.1411.4038

Ludovisi, R., Tauro, F., Salvati, R., Khoury, S., Mugnozza Scarascia, G., & Harfouche, A. (2017). UAV-Based Thermal Imaging for High-Throughput Field Phenotyping of Black Poplar Response to Drought. *Frontiers in Plant Science*, *8*, 1681. https://doi.org/10.3389/fpls.2017.01681

Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q. Z., Xiong, H., & Akoglu, L. (2021). A Comprehensive Survey on Graph Anomaly Detection with Deep Learning. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2021.3118815

MacDonald, S. L., Staid, M., Staid, M., & Cooper, M. L. (2016). Remote hyperspectral imaging of grapevine leafroll-associated virus 3 in cabernet sauvignon vineyards. *Computers and Electronics in Agriculture*, *130*, 109-117. https://doi.org/10.1016/j.compag.2016.10.003

Mackie, K. A., Müller, T., & Kandeler, E. (2012). Remediation of copper in vineyards – A mini review. *Environmental Pollution*, *167*, 16-26. https://doi.org/10.1016/j.envpol.2012.03.023

Madec, S., Baret, F., de Solan, B., Thomas, S., Dutartre, D., Jezequel, S., Hemmerlé, M., Colombeau, G., & Comar, A. (2017). High-Throughput Phenotyping of Plant Height : Comparing Unmanned Aerial Vehicles and Ground LiDAR Estimates. *Frontiers in Plant Science*, *8*. https://doi.org/10.3389/fpls.2017.02002

Martelli, G. P. (2017). An Overview on Grapevine Viruses, Viroids, and the Diseases They Cause. *Grapevine Viruses : Molecular Biology, Diagnostics and Management*, 1-46. Springer International Publishing. https://doi.org/10.1007/978-3-319-57706-7_2

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115-133. https://doi.org/10.1007/BF02478259

Merdinoglu, D., Schneider, C., Prado, E., Wiedemann-Merdinoglu, S., & Mestre, P. (2018). Breeding for durable resistance to downy and powdery mildew in grapevine. *OENO One*, *52*(3). https://doi.org/10.20870/oeno-one.2018.52.3.2116

Merot, A., Fermaud, M., Gosme, M., & Smits, N. (2020). Effect of Conversion to Organic Farming on Pest and Disease Control in French Vineyards. *Agronomy*, *10*(7). https://doi.org/10.3390/agronomy10071047

Mondello, V., Larignon, P., Armengol, J., Kortekamp, A., Vaczy, K., Prezman, F., Serrano, E., Rego, C., Mugnai, L., & Fontaine, F. (2018). Management of grapevine trunk diseases. *Phytopathologia Mediterranea*, *57*(3), 369-383.

Morellos, A., Pantazi, X. E., Paraskevas, C., & Moshou, D. (2022). Comparison of Deep Neural Networks in Detecting Field Grapevine Diseases Using Transfer Learning. *Remote Sensing*, *14*(18). https://doi.org/10.3390/rs14184648

Morone, C., Boveri, M., Giosuè, S., Gotta, P., Rossi, V., Scapin, I., & Marzachì, C. (2007). Epidemiology of Flavescence Dorée in Vineyards in Northwestern Italy. *Phytopathology®*, *97*(11), 1422-1427. https://doi.org/10.1094/PHYTO-97-11-1422

Morris, C. (2022). Graph Neural Networks : Graph Classification. *Graph Neural Networks : Foundations, Frontiers, and Applications*, 179-193. Springer Nature Singapore. https://doi.org/10.1007/978-981-16-6054-2_9

Mundy, D. C., & Manning, M. A. (2010). Ecology and management of grapevine trunk diseases in New Zealand : A review. *New Zealand Plant Protection*, *63*, 160-166. https://doi.org/10.30843/nzpp.2010.63.6558

Musci, M. A., Persello, C., & Lingua, A. M. (2020). UAV images and deep-learning algorithms for detecting Flavescence doree disease in grapevine orchards. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1483-1489. https://doi.org/10.5194/isprs-archives-XLIII-B3-2020-1483-2020

Muthmann, R., & Nadin, P. (2007). The use of plant protection products in the European union. https://ec.europa.eu/eurostat/documents/3217494/5611788/KS-76-06-669-EN.PDF.pdf/36c156f1-9fa9-4243-9bd3-f4c7c3c8286a?t=1414769021000

Naidu, R., Rowhani, A., Fuchs, M., Golino, D., & Martelli, G. P. (2014). Grapevine Leafroll : A Complex Viral Disease Affecting a High-Value Fruit Crop. *Plant Disease*, *98*(9), 1172-1185. https://doi.org/10.1094/PDIS-08-13-0880-FE

Nardi, B. de, Possamai, T., & Velasco, R. (2019). Breeding for mildew resistance in grapevine to improve environmental and socio-economic sustainability in hotspot areas of Veneto. *Acta Horticulturae*, *1248*, 313-318.

Nguyen, C., Sagan, V., Maimaitiyiming, M., Maimaitijiang, M., Bhadra, S., & Kwasniewski, M. T. (2021). Early Detection of Plant Viral Disease Using Hyperspectral Imaging and Deep Learning. *Sensors*, *21*(3). https://doi.org/10.3390/s21030742

Oerke, E.-C., Mahlein, A.-K., & Steiner, U. (2014). Proximal Sensing of Plant Diseases. *Detection and Diagnostics of Plant Pathogens*, 55-68. Springer Netherlands. https://doi.org/10.1007/978-94-017-9020-8_4

Oliveira, M. J. R. A., Castro, S., Paltrinieri, S., Bertaccini, A., Sottomayor, M., Santos, C. S., Vasconcelos, M. W., & Carvalho, S. M. P. (2020). "Flavescence dorée" impacts growth, productivity and ultrastructure of Vitis vinifera plants in Portuguese "Vinhos Verdes" region. *Scientia Horticulturae*, *261*. https://doi.org/10.1016/j.scienta.2019.108742

Pan, S., Hu, R., Fung, S.-F., Long, G., Jiang, J., & Zhang, C. (2020). Learning Graph Embedding With Adversarial Training Methods. *IEEE Transactions on Cybernetics*, *50*(6), 2475-2487. https://doi.org/10.1109/TCYB.2019.2932096

Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., & Zhang, C. (2018). Adversarially regularized graph autoencoder for graph embedding. *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2609-2615.

Pantazi, X. E., Moshou, D., Tamouridou, A. A., & Kasderidis, S. (2016). Leaf Disease Recognition in Vine Plants Based on Local Binary Patterns and One Class Support Vector Machines. *12th IFIP International Conference on Artificial Intelligence Applications and Innovations,* 319-327. https://doi.org/10.1007/978-3-319-44944-9_27

Papura, D., Delmotte, F., Giresse, X., Salar, P., Danet, J. L., Van Helden, M., Foissac, X., & Malembic-Maher, S. (2009). Comparing the spatial genetic structures of the Flavescence dorée phytoplasma and its leafhopper vector Scaphoideus titanus. *Infection, Genetics and Evolution*, *9*(5), 867-876. https://doi.org/10.1016/j.meegid.2009.05.009

Parker, A., de Cortázar-Atauri, I. G., Chuine, I., Barbeau, G., Bois, B., Boursiquot, J.-M., Cahurel, J.-Y., Claverie, M., Dufourcq, T., Gény, L., Guimberteau, G., Hofmann, R. W., Jacquet, O., Lacombe, T., Monamy, C., Ojeda, H., Panigai, L., Payan, J.-C., Lovelle, B. R., Van Leeuwen, C. (2013). Classification of varieties for their timing of flowering and veraison using a modelling approach : A case study for the grapevine species Vitis vinifera L. *Agricultural and Forest Meteorology*, *180*, 249-264. https://doi.org/10.1016/j.agrformet.2013.06.005

Pearson, R. C., & Goheen, A. C. (1988). Compendium of grape diseases. APS Press. http://books.google.com/books?id=xgxJAQAAIAAJ

Pérez-Roncal, C., Arazuri, S., Lopez-Molina, C., Jarén, C., Santesteban, L. G., & López-Maestresalas, A. (2022). Exploring the potential of hyperspectral imaging to detect Esca disease complex in asymptomatic grapevine leaves. *Computers and Electronics in Agriculture*, *196*. https://doi.org/10.1016/j.compag.2022.106863

Perria, R., Ciofini, A., Petrucci, W. A., D'Arcangelo, M. E. M., Valentini, P., Storchi, P., Carella, G., Pacetti, A., & Mugnai, L. (2022). A Study on the Efficiency of Sustainable Wine Grape Vineyard Management Strategies. *Agronomy*, *12*(2). https://doi.org/10.3390/agronomy12020392

Pesticide Action Network. (2008). Message in a Bottle. https://www.pan-europe.info/sites/pan-europe.info/files/public/resources/briefings/message-in-a-bottle.pdf

Pridmore, T. P. (2001). Thresholding Images of Line Drawings with Hysteresis. *Selected Papers from the Fourth International Workshop on Graphics Recognition Algorithms and Applications*, 310-319.

Prunier, J. P., Ride, M., Lafon, R., & Bulit, J. (1970). Bacterial blight of Grapevine. *Progres Agricole et Viticole*, *87*(21), 316-322.

Pujari, J. D., Yakkundimath, R., & Byadgi, A. S. (2015). Image Processing Based Detection of Fungal Diseases in Plants. *Proceedings of the International Conference on Information and Communication Technologies, ICICT 2014, 46*, 1802-1808. https://doi.org/10.1016/j.procs.2015.02.137

Quaglino, F., Zhao, Y., Casati, P., Bulgari, D., Bianco, P. A., Wei, W., & Davis, R. E. (2013). « Candidatus Phytoplasma solani », a novel taxon associated with stolbur- and bois noir-related diseases of plants. *International Journal of Systematic and Evolutionary Microbiology*, *63*(8), 2879-2894. https://doi.org/10.1099/ijs.0.044750-0

Rancon, F. (2019). Imagerie couleur et hyperspectrale pour la détection et la caractérisation des maladies du bois de la vigne. https://tel.archives-ouvertes.fr/tel-03140819

Rançon, F., Keresztes, B., Deshayes, A., Tardif, M., Abdelghafour, F., Fontaine, G., Da Costa, J.-P., & Germain, C. (2023). Designing a Proximal Sensing Camera Acquisition System for Vineyard Applications : Results and Feedback on 8 Years of Experiments. *Sensors*, *23*(2). https://doi.org/10.3390/s23020847

Range, K. (2023). RangeKing. *GitHub*. https://github.com/RangeKing

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once : Unified, Real-Time Object Detection. *arXiv:1506.02640*. http://arxiv.org/abs/1506.02640

Reineke, A., & Thiéry, D. (2016). Grapevine insect pests and their natural enemies in the age of global warming. *Journal of Pest Science*, *89*(2), 313-328. https://doi.org/10.1007/s10340-016-0761-8

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, *3*(22), 41-46.

Robert, A. (2019). Pesticide challenge leaving French viticulture with little choice. *Www.Euractiv.Com*. https://www.euractiv.com/section/agriculture-food/news/pesticide-challenge-leaving-french-viticulture-with-little-choice/

Robert, S., Reisenzein, H., & Zeisner, N. (2007). Analysis of the pest risk from Grapevine flavescence dorée phytoplasma to Austrian viticulture. *EPPO Bulletin*, *37*, 191-203. https://doi.org/10.1111/j.1365-2338.2007.01102.x

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net : Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 234-241. Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Saigo, H., Nowozin, S., Kadowaki, T., Kudo, T., & Tsuda, K. (2009). gBoost : A mathematical programming approach to graph classification and regression. *Machine Learning*, *75*(1), 69-89. https://doi.org/10.1007/s10994-008-5089-z

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2 : Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

Saponara, S., Elhanashi, A., & Qinghe, Z. (2022). Developing a real-time social distancing detection system based on YOLOv4-tiny and bird-eye view for COVID-19. *Journal of Real-Time Image Processing*, *19*, 1-13. https://doi.org/10.1007/s11554-022-01203-5

Sassu, A., Gambella, F., Ghiani, L., Mercenaro, L., Caria, M., & Pazzona, A. L. (2021). Advances in Unmanned Aerial System Remote Sensing for Precision Viticulture. *Sensors*, *21*(3). https://doi.org/10.3390/s21030956

Scalabrelli, G. (2014). Esca symptoms appearance in Vitis vinifera L. : Influence of climate, pedo-climatic conditions and rootstock/cultivar combination. *Vitis*, *53*, 33-38.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, *20*(1), 61-80. https://doi.org/10.1109/TNN.2008.2005605

Shih, F. Y., & Cheng, S. (2005). Automatic seeded region growing for color image segmentation. *Image and Vision Computing*, *23*(10), 877-886. https://doi.org/10.1016/j.imavis.2005.05.015

Shinde, P. P., & Shah, S. (2018). A Review of Machine Learning and Deep Learning Applications. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1-6. https://doi.org/10.1109/ICCUBEA.2018.8697857

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556

Singh, A. P., Yerudkar, A., Mariani, V., Iannelli, L., & Glielmo, L. (2022). A Bibliometric Review of the Use of Unmanned Aerial Vehicles in Precision Agriculture and Precision Viticulture for Sensing Applications. *Remote Sensing*, *14*(7). https://doi.org/10.3390/rs14071604

Slippers, B., & Wingfield, M. J. (2007). Botryosphaeriaceae as endophytes and latent pathogens of woody plants : Diversity, ecology and impact. *Fungal Biology Reviews*, *21*(2), 90-106. https://doi.org/10.1016/j.fbr.2007.06.002

Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., & Pattabiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. *Journal of Big Data*, *8*(1), 66. https://doi.org/10.1186/s40537-021-00434-w

Szegedi, E., & Civerolo, E. L. (2011). Bacterial diseases of grapevine. *International Journal of Horticultural Science*, *17*(3). https://doi.org/10.31421/IJHS/17/3/956

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9. https://doi.org/10.1109/CVPR.2015.7298594

Tan, M., & Le, Q. (2019). EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, 97, 6105-6114. https://proceedings.mlr.press/v97/tan19a.html

Tardif, M., Amri, A., Deshayes, A., Greven, M., Keresztes, B., Fontaine, G., Sicaud, L., Paulhac, L., Bentejac, S., & Da Costa, J.-P (2023). An expertized grapevine disease image database including five grape varieties focused on Flavescence dorée and its confounding diseases, biotic and abiotic stresses. *Data in Brief*, *48*. https://doi.org/10.1016/j.dib.2023.109230

Tardif, M., Amri, A., Keresztes, B., Deshayes, A., Martin, D., Greven, M., & Da Costa, J.-P. (2022). Two-stage automatic diagnosis of Flavescence Dorée based on proximal imaging and artificial intelligence : A multi-year and multi-variety experimental study. *OENO One*, *56*(3). https://doi.org/10.20870/oeno-one.2022.56.3.5460

Tardif, M., Amri, A., Keresztes, B., Deshayes, A., Martin, D., Greven, M., & Da Costa, J.-P. (2023). Automatic diagnosis of a multi-symptom grape vine disease using computer vision. *Acta Horticulturae*, *1360*, 53-60. https://doi.org/10.17660/ActaHortic.2023.1360.7

Tardif, M., Keresztes, B., Deshayes, A., Martin, D., Greven, M., & Da Costa, J.-P. (2023). Automatic diagnosis of a multi-symptom grapevine disease by decision trees and Graph Neural Networks.

In *Precision agriculture, 23,* 1011-1017. Wageningen Academic Publishers. https://doi.org/10.3920/978-90-8686-947-3_127

Tessitori, M., La Rosa, R., & Marzachì, C. (2018). Flavescence Dorée and Bois Noir Diseases of Grapevine Are Evolving Pathosystems. *Plant Health Progress*, *19*(2), 136-138. https://doi.org/10.1094/PHP-10-17-0057-MR

The GIMP Development Team. (2019). *GIMP* (2.10.12). https://www.gimp.org

Úrbez-Torres, J. R., Leavitt, G. M., Voegel, T. M., & Gubler, W. D. (2006). Identification and Distribution of Botryosphaeria spp. Associated with Grapevine Cankers in California. *Plant Disease*, *90*(12), 1490-1503. https://doi.org/10.1094/PD-90-1490

Úrbez-Torres, J. R., Peduto, F., Smith, R. J., & Gubler, W. D. (2013). Phomopsis Dieback : A Grapevine Trunk Disease Caused by Phomopsis viticola in California. *Plant Disease*, *97*(12), 1571-1579. https://doi.org/10.1094/PDIS-11-12-1072-RE

Usha, K., & Singh, B. (2013). Potential applications of remote sensing in horticulture—A review. *Scientia Horticulturae*, *153*, 71-83. https://doi.org/10.1016/j.scienta.2013.01.008

Van der Werf, H. M. G. (1996). Assessing the impact of pesticides on the environment. *Agriculture, Ecosystems & Environment*, *60*(2), 81-96. https://doi.org/10.1016/S0167-8809(96)01096-1

Van Klompenburg, T., Kassahun, A., & Catal, C. (2020). Crop yield prediction using machine learning : A systematic literature review. *Computers and Electronics in Agriculture*, *177*. https://doi.org/10.1016/j.compag.2020.105709

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, 30. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Viola, P., & Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE Conf Comput Vis Pattern Recognit*. https://doi.org/10.1109/CVPR.2001.990517

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., & Borgwardt, K. M. (2010). Graph Kernels. *Journal of Machine Learning Research*, *11*(40), 1201-1242.

Wada, K. (2021). Labelme : Image Polygonal Annotation with Python. https://doi.org/10.5281/zenodo.5711226

Wang, M., Qiu, L., & Wang, X. (2021). A Survey on Knowledge Graph Embeddings for Link Prediction. *Symmetry*, *13*(3). https://doi.org/10.3390/sym13030485

Wang, S.-Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (2020). CNN-Generated Images Are Surprisingly Easy to Spot... For Now. 8695-8704. https://openaccess.thecvf.com/content_CVPR_2020/html/Wang_CNN-Generated_Images_Are_Surprisingly_Easy_to_Spot..._for_Now_CVPR_2020_paper.html

Wang, X., Bo, D., Shi, C., Fan, S., Ye, Y., & Yu, P. S. (2023). A Survey on Heterogeneous Graph Embedding : Methods, Techniques, Applications and Sources. *IEEE Transactions on Big Data*, *9*(2), 415-436. https://doi.org/10.1109/TBDATA.2022.3177455

Wang, Y. M., Ostendorf, B., & Pagay, V. (2023). Detecting Grapevine Virus Infections in Red and White Winegrape Canopies Using Proximal Hyperspectral Sensing. *Sensors*, *23*(5). https://doi.org/10.3390/s23052851

Wilson, H., & Daane, K. M. (2017). Review of Ecologically-Based Pest Management in California Vineyards. *Insects*, *8*(4). https://doi.org/10.3390/insects8040108

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4-24. https://doi.org/10.1109/TNNLS.2020.2978386

Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., & Liu, H. (2021). Graph Learning : A Survey. *IEEE Transactions on Artificial Intelligence*, *2*(2), 109-127. https://doi.org/10.1109/TAI.2021.3076021

Xiao, S., Wang, S., Dai, Y., & Guo, W. (2021). Graph neural networks in node classification : Survey and evaluation. *Machine Vision and Applications*, *33*(1). https://doi.org/10.1007/s00138-021-01251-0

Yalcin, H. (2017). Plant phenology recognition using deep learning : Deep-Pheno. *2017 6th International Conference on Agro-Geoinformatics*, 1-5. https://doi.org/10.1109/Agro-Geoinformatics.2017.8046996

Yang, J., Lu, J., Lee, S., Batra, D., & Parikh, D. (2018). Graph R-CNN for Scene Graph Generation. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., & Leskovec, J. (2018). Hierarchical Graph Representation Learning with Differentiable Pooling. *Advances in Neural Information Processing Systems*, *31*. https://papers.nips.cc/paper/2018/hash/e77dbaf6759253c7c6d0efc5690369c7-Abstract.html

Yun, S., Jeong, M., Kim, R., Kang, J., & Kim, H. J. (2019). Graph Transformer Networks. *Advances in Neural Information Processing Systems*, 32. https://proceedings.neurips.cc/paper_files/paper/2019/file/9d63484abb477c97640154d40595a3bb-Paper.pdf

Zhang, M., & Chen, Y. (2018). Link Prediction Based on Graph Neural Networks. *arXiv:1802.09691 [cs, stat]*. http://arxiv.org/abs/1802.09691

Zhang, M., Cui, Z., Neumann, M., & Chen, Y. (2018). An End-to-End Deep Learning Architecture for Graph Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://ojs.aaai.org/index.php/AAAI/article/view/11782

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks : A comprehensive review. *Computational Social Networks*, *6*(1). https://doi.org/10.1186/s40649-019-0069-y

Zhao, Z.-Q., Zheng, P., Xu, S., & Wu, X. (2019). Object Detection with Deep Learning : A Review. *arXiv:1807.05511*. http://arxiv.org/abs/1807.05511

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2019). *Distance-IoU Loss : Faster and Better Learning for Bounding Box Regression*. arXiv:1911.08287. arXiv. https://doi.org/10.48550/arXiv.1911.08287

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2019). Graph Neural Networks : A Review of Methods and Applications. *arXiv:1812.08434*. http://arxiv.org/abs/1812.08434

Zhu, Y., Du, Y., Wang, Y., Xu, Y., Zhang, J., Liu, Q., & Wu, S. (2022). A Survey on Deep Graph Generation : Methods and Applications. *Proceedings of the First Learning on Graphs Conference* 198, 1-21. https://proceedings.mlr.press/v198/zhu22a.html