

Classification des symptômes causés par le watermelon mosaic virus (WMV) sur melon par Deep Learning: conception d'un modèle léger et déploiement mobile

Naïma Dambrine¹
Matthieu Deloget²
Jacques Lagnel³
Lucie Tamisier³

CORRESPONDANCE

naima.dambrine@inrae.fr; lucie.tamisier@inrae.fr

RÉSUMÉ

Ce projet explore l'application du Deep Learning à la classification automatique des symptômes causés par le watermelon mosaic virus (WMV) sur melon, une culture fortement affectée par ce virus. Un modèle léger de classification a été conçu et entraîné à partir d'images de feuilles, puis converti au format TFLite pour un déploiement embarqué sur smartphone. Une attention particulière a été portée à la préparation des données, à la réduction du bruit visuel et à la compatibilité mobile. L'application mobile obtenue permet une prédiction rapide et localisée des stades de la maladie, ouvrant la voie à une aide au diagnostic en conditions expérimentales.

MOTS-CLÉS

Apprentissage Profond; classification d'images; YOLOv8-cls; réseaux de neurones convolutifs; melon; watermelon mosaic virus (WMV); détection de maladies des plantes; phénotypage; application mobile; modèles légers

1. INRAE, EPIdémiologie des maladies Animales et zoototiques, 69280, Marcy L'Etoile, France

2. Avignon Université, Master Ingénierie M1 Intelligence Artificielle, 84000, Avignon, France

3. INRAE, Génétique et Amélioration des Fruits et Légumes, F-84143, Montfavet, France

Classification of symptoms caused by watermelon mosaic virus (WMV) on melons using Deep Learning: design of a lightweight model and mobile deployment

Naïma Dambrine¹
Matthieu Deloget²
Jacques Lagnel³
Lucie Tamisier³

CORRESPONDENCE

naima.dambrine@inrae.fr; Lucie.Tamisier@inrae.fr

ABSTRACT

This project explores the application of Deep Learning to the automatic classification of symptoms caused by the watermelon mosaic virus (WMV) on melon, a crop heavily affected by this virus. A lightweight classification model was designed and trained on leaf images, then converted to TFLite format for on-board deployment on smartphones. Particular attention was paid to data preparation, visual noise reduction and mobile compatibility. The resulting mobile application enables rapid, localised prediction of disease stages, paving the way for diagnostic support in experimental conditions.

KEYWORDS

Deep Learning; image classification; YOLOv8-cls; convolutional neural networks; melon; watermelon mosaic virus (WMV); plant disease detection; phenotyping; mobile application; lightweight mode

1. INRAE, EPIdémiologie des maladies Animales et zoototiques, 69280, Marcy L'Etoile, France

2. Avignon Université, Master Ingénierie M1 Intelligence Artificielle, 84000, Avignon, France

3. INRAE, Génétique et Amélioration des Fruits et Légumes, F-84143 Montfavet, France

Introduction

Les virus des plantes représentent une menace majeure en agriculture, compromettant à la fois les rendements et la qualité des récoltes. Le watermelon mosaic virus (WMV; genre *Potyvirus*, famille *Potyviridae*) est un des principaux virus infectant les cucurbitacées comme le melon, la pastèque, la courgette ou le concombre. Présent dans le monde entier, le WMV est particulièrement répandu dans les zones tempérées et méditerranéennes (Desbiez, 2020).

En l'absence de méthodes curatives, l'utilisation de variétés de plantes génétiquement résistantes constitue l'un des principaux moyens de lutte contre les virus (Kang *et al.*, 2005). Le développement de ces variétés nécessite de phénotyper plusieurs centaines de génotypes de plante afin d'évaluer leur niveau de résistance à l'agent pathogène, puis d'exploiter ces données à l'aide de modèles génétiques permettant d'identifier les gènes de résistance impliqués (Korte & Farlow, 2013).

Chez le pathosystème melon-WMV, une méthode de phénotypage couramment employée consiste à noter la sévérité des symptômes foliaires selon une échelle de notation standardisée (Díaz-Pendón *et al.*, 2005 ; Gilbert *et al.*, 1994 ; López-Martín *et al.*, 2024 ; Palomares-Rius *et al.*, 2011). Les symptômes peuvent aller de faibles mosaïques à des déformations foliaires sévères selon les souches virales et les génotypes de plante (Lecoq & Desbiez, 2012). Ce phénotypage visuel reste cependant subjectif et dépend fortement de l'expertise de l'observateur, ce qui peut introduire des variations entre évaluateurs. De plus, ce travail de phénotypage demeure une étape longue et fastidieuse dans les programmes de recherche et de sélection variétale. Le développement d'outils permettant de standardiser et d'automatiser la reconnaissance et la quantification des symptômes ouvre donc des perspectives majeures pour l'amélioration génétique des cucurbitacées et la gestion durable des maladies virales.

Dans ce contexte, nous avons cherché à développer un outil d'analyse automatisée des symptômes du WMV, en nous appuyant sur des approches de vision par ordinateur et d'intelligence artificielle (Ferentinos, 2018 ; Mohanty *et al.*, 2016). L'objectif est de concevoir un modèle léger, embarquable sur smartphone, capable de classifier l'intensité des symptômes à partir de photos de feuilles asymptomatiques ou infectées, dans le cadre du phénotypage de masse.

L'essor de l'intelligence artificielle (IA) et des méthodes d'analyse d'image offre de nouvelles perspectives pour automatiser et standardiser ces observations. L'analyse par Deep Learning, en particu-

lier, permet de classifier automatiquement les symptômes à partir de simples photographies de feuilles (Barbedo, 2019).

Une étude exploratoire a été menée pour évaluer cette approche dans le contexte du WMV, en s'appuyant sur la constitution d'un jeu d'images annotées et l'expérimentation de plusieurs architectures de classification, jusqu'au déploiement du modèle retenu dans une application mobile dédiée (Mrisho *et al.*, 2020). Nous en présentons ici les principales étapes : préparation et traitement du jeu de données, entraînement et évaluation du modèle retenu, puis intégration dans un outil léger, simple d'usage et adapté aux besoins du phénotypage en conditions expérimentales.

Préparation et traitement du jeu de données

Constitution du jeu de données

Le jeu de données a été obtenu en semant 130 génotypes de melon (*Cucumis melo*) représentatifs de la diversité mondiale de l'espèce et présentant différents niveaux de résistance au WMV, avec 6 plantes par génotype. Une semaine après semis, les plantes ont été inoculées mécaniquement avec la souche LL1A. Un mois après inoculation, une feuille par plante a été prélevée pour réaliser la prise de photo. L'expérience a été effectuée en conditions contrôlées en chambre de culture (photopériode : 16h lumière/8h obscurité, températures : 24 °C jour/18 °C nuit).

Les photographies ont été réalisées dans des conditions maîtrisées (fond noir, distance fixe grâce à un trépied) afin d'assurer l'homogénéité des images et de limiter les biais d'analyse. Les images ont ensuite été classées manuellement selon une échelle en quatre stades, du stade 0 : feuille asymptomatique au stade 3 : forte déformation foliaire (Figure 1).

En vue de leur utilisation dans un modèle d'apprentissage automatique, les images ont ensuite fait l'objet d'un prétraitement structuré comprenant les étapes suivantes : un nettoyage des images (recadrage, redimensionnement, suppression des doublons ou images floues), une gestion des déséquilibres entre classes (par échantillonnage et équilibrage), ainsi qu'une augmentation des données via des transformations simples (rotations, symétries, variations de luminosité).

Les images brutes, d'une résolution initiale de 4496×3 000 pixels, ont fait l'objet d'un prétraitement systématique (Figure 2) visant à améliorer la qualité du signal et à normaliser les entrées du modèle.



Figure 1. Exemples d'images de feuilles de melon représentatives de l'intensité des symptômes causés par le WMV pour chacun des 4 stades (0 à 3). *Prétraitement des images*



Figure 2. Prétraitement : détection et sélection de la feuille dominante, fond uniforme, redimensionnement pour l'entraînement

Une procédure automatique a permis de détecter la feuille dominante en exploitant son contraste avec le fond noir et la proportion de pixels verts. Le contour présentant le meilleur ratio de pixels verts dans l'espace HSV était sélectionné, puis utilisé pour extraire la feuille sous forme de masque binaire.

Le fond était ensuite remplacé par une couleur uniforme (gris moyen), et l'image recadrée autour de la feuille. Chaque image était finalement redimensionnée à 640×640 pixels sans distorsion, via une mise à l'échelle conservant le rapport hauteur/largeur, suivie d'un padding homogène sur fond gris.

Équilibrage du jeu de données

Le jeu de données ainsi obtenu présente un déséquilibre notable entre classes, en particulier pour le stade 3 (formes sévères). Pour anticiper les biais d'apprentissage, les données ont été réparties de manière stratifiée en deux ensembles : 80 % pour l'entraînement et 20 % pour la validation, tout en conservant une distribution représentative des stades dans les ensembles originaux (Figure 3).

Augmentation de données ciblée

Afin d'atténuer le déséquilibre observé, une stratégie d'augmentation de données ciblée a été mise en œuvre pour la classe 3.

À chaque image ont été appliquées des transformations contrôlées : retournement horizontal, zoom, translation, rotation légère, et flou modéré.

Cette démarche a permis d'augmenter la diversité intra-classe sans introduire d'artéfacts visuels.

Les matrices de confusion ont été établies à partir des prédictions du modèle de classification YOLOv8m-cls (Ultralytics) sur l'ensemble de validation : chaque image est affectée à la classe la plus probable (argmax), puis la matrice est normalisée par classe vraie afin d'exprimer des proportions. Compte tenu du faible effectif de validation pour la classe 3 ($n = 14$), nous rapportons en complément une analyse sur une validation enrichie (169 images originales + 28 variantes transformées), interprétée comme un test de sensibilité à des transformations légères. Les matrices de confusion correspondantes sont présentées en Figure 4. Les résultats suggèrent une amélioration de la détection du stade 3, au prix d'une baisse du score top-1 global (0.905 sans augmentation vs 0.878 avec augmentation).

Entrainement et évaluation du modèle

Cette étude vise avant tout à illustrer une démarche de classification supervisée et un pipeline d'entraînement reproductible dans un contexte de données limitées. En l'absence d'un volume suffisant pour constituer un ensemble de test indépendant, les performances sont rapportées sur un ensemble de validation, complétées par une analyse par classe (matrices de confusion) et une évaluation de sensibilité aux transformations lorsque pertinent.

Plusieurs modèles de classification visuelle ont été évalués : YOLOv11n-cls, YOLOv8n-cls, et YOLOv8m-cls. Le modèle YOLOv8m-cls a été retenu pour son bon compromis entre robustesse et rapidité d'inférence.

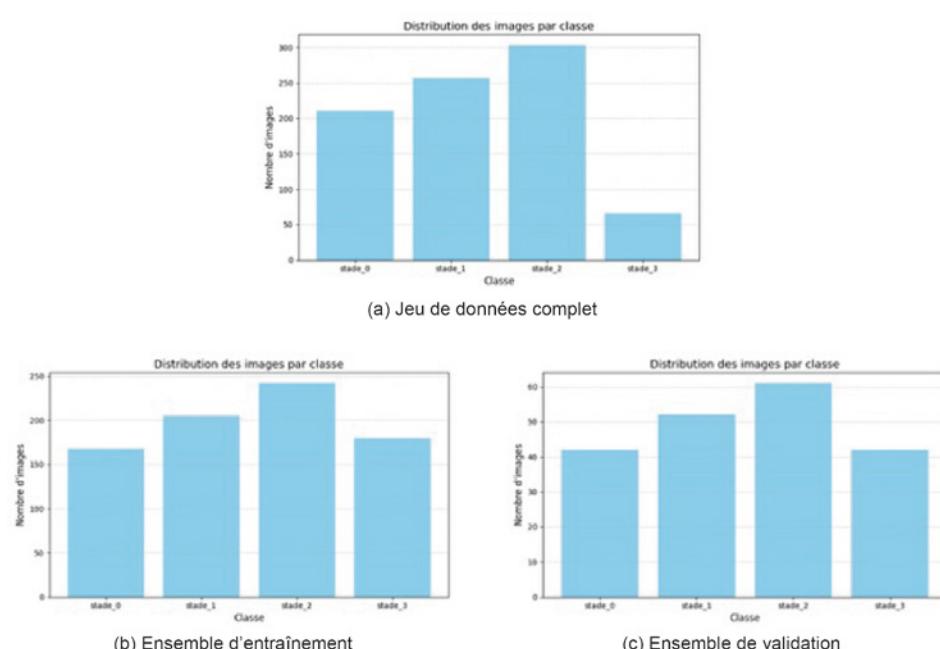


Figure 3. Distribution des classes dans le jeu de données complet, puis dans les ensembles d'entraînement et de validation après répartition stratifiée (80 % / 20 %).

Le déséquilibre du stade 3, motive l'exploration d'une augmentation ciblée.

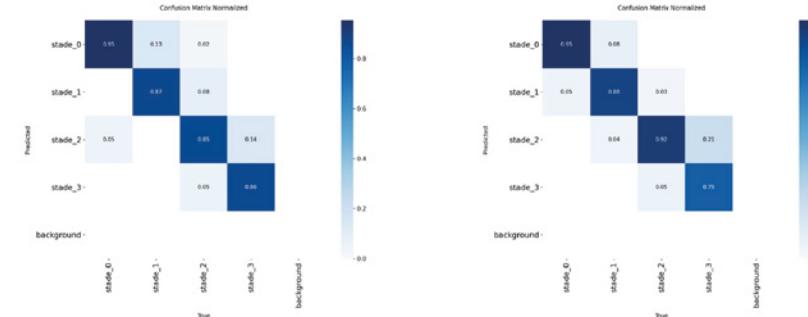


Figure 4. Matrices de confusion normalisées obtenues avec un réseau de classification YOLOv8m-cls (Ultralytics), avant et après augmentation ciblée de la classe 3 appliquée à l'ensemble d'entraînement.

L'évaluation est réalisée sur une validation enrichie utilisée comme test de sensibilité aux transformations.

L'entraînement a été réalisé avec la bibliothèque Ultralytics, en mode classification supervisée, sur des images 640 x 640 pixels avec les paramètres suivants :

- **Nombre d'époques :** 50
- **Batch size :** 16
- **Optimiseur :** Adam avec un taux d'apprentissage initial de 0.001
- **Scheduler du taux d'apprentissage :** automatique (avec *early stopping* si stagnation)
- **Patience :** 10
- **Augmentations :** désactivées initialement, car les transformations par défaut d'Ultralytics introduisaient des artefacts visuels peu cohérents avec les cas réels
- **Classes :** 4 niveaux de gravité (de l'absence de symptômes jusqu'à la forte déformation de la feuille)
- **Répartition :** 80 % entraînement, 20 % validation (aléatoire mais stratifiée)

Une expérimentation ultérieure avec la bibliothèque Albumentations a permis d'identifier des augmentations plus adaptées, notamment pour la classe 3, sous-représentée. En appliquant des transformations réalistes (rotation légère, flou modéré, etc), il a été possible d'augmenter la diversité sans nuire à la lisibilité des symptômes.

Sur la validation originale, le modèle atteint une précision top-1 de 0.905. L'augmentation ciblée de la classe 3 vise à améliorer la reconnaissance des formes sévères et est analysée via les matrices de confusion (Figure 4); sur l'évaluation enrichie, le score top-1 global est de 0.878. Ces tendances devront être confirmées sur un jeu de données de validation plus large.

- Quantification float16 lors de l'export TFLite (*half=True*), réduisant la taille du modèle sans perte notable de précision.
- Chargement avec délégation GPU (*GpuDelegateV2*) assurant une inférence fluide :
 - moins de 700 ms sur smartphone récent (moins d'un an);
 - moins de 2 700 ms sur un appareil ancien (6 à 7 ans).

Développement de l'application mobile

L'application mobile, développée avec Flutter pour une compatibilité Android/iOS, permet à l'utilisateur de prendre une photo ou d'en sélectionner une dans la galerie, d'envoyer l'image au modèle embarqué, puis d'afficher le stade estimé accompagné d'une probabilité (Figure 5).

L'inférence se fait hors-ligne, garantissant un usage autonome sans connexion réseau.

L'interface est minimalistique et accessible à un public non spécialiste. Elle sert également de pré-test avant une phase 2 axée sur la détection en temps réel sur le terrain.

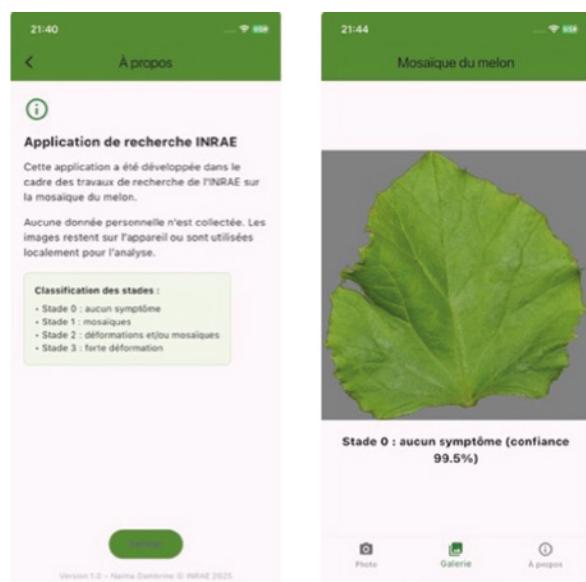


Figure 5. Aperçu de l'interface mobile (iPhone)

Intégration dans une application mobile

Conversion et déploiement mobile

Le modèle final a été converti au format TFLite pour permettre son exécution directe sur smartphone. Deux optimisations complémentaires ont été appliquées :

Résultats

Précision de classification

Le modèle YOLOv8m-cls atteint une précision top-1 de 0.905 sur l'ensemble de validation, avec un bon équilibre entre les différentes classes (Tableau 1).

Tableau 1. Récapitulatif des classes / métriques

Classe	Label	Nombre d'images	Précision top-1 (%)
0	asymptomatique	212	92.0
1	mosaïque faible	259	85.3
2	mosaïque forte	309	84.2
3	forte déformation	71	91.1

L'augmentation ciblée de la classe 3 s'est avérée déterminante pour améliorer la détection des formes sévères, initialement sous-représentées.

La matrice de confusion indique une nette amélioration de la reconnaissance du stade 3, sans dégradation significative des performances globales (Figure 4).

Temps d'inférence

Des tests ont été réalisés sur différents appareils afin d'évaluer le temps d'inférence du modèle embarqué. Les résultats varient selon les performances matérielles, allant de 700 ms sur les smartphones récents à 2 700 ms sur des smartphones plus anciens (> 5 ans) (Tableau 2).

Précisions techniques

- Android : Le délégué GpuDelegateV2 de TensorFlow Lite est disponible et améliore nettement les performances. À activer dès que possible.
- iOS : Le plugin tflite_flutter ne prend pas encore en charge Metal (GPU d'Apple). L'inférence se fait donc sur CPU, mais les appareils récents (comme l'iPhone 16) montrent une excellente réactivité.
- L'application adapte dynamiquement la configuration de l'interpréteur selon la plateforme, (Platform.isAndroid, Platform.isIOS).

Précision et robustesse

Les prédictions faites avec le modèle TFLite float16 sont cohérentes avec celles générées en Python avec le modèle YOLOv8 natif, avant conversion. Cela confirme la fidélité de la conversion et la robustesse du modèle embarqué.

Tableau 2. Comparaison des temps d'inférence par plateforme

Plateforme	Déléguer	Temps moyen (ms)	Commentaire technique
MacBook (YOLO)	CPU/GPU natif	~55	YOLOv8m, inférence directe avec Ultralytics
MacBook (TFLite)	CPU TFLite	~800	Modèle float16, sans délégué GPU
Galaxy A40	CPU (sans délégué)	~9 000	Très lent, traitement purement CPU
Galaxy A40	GPUDelegateV2	~3 600	Temps divisé par 2-3 grâce au GPU Delegate
iPhone XR	CPU uniquement	~1 800	Bonne performance sans Metal
iPhone 16	CPU uniquement	~720	Excellent réactivité malgré l'absence de support Metal

Conclusion

Ce projet confirme la faisabilité d'un diagnostic embarqué des symptômes foliaires induits par le WMV chez le melon basé sur l'intelligence artificielle, via un modèle de classification intégré à une application mobile.

Cll démontre la possibilité d'un cycle complet, de l'annotation à l'inférence sur smartphone, sans infrastructure serveur ni connexion Internet.

Le prototype mis en place est robuste, rapide (inférence < 1 seconde sur iPhone récent), et facilement transposable. Il pourrait être adapté à d'autres maladies du melon ou à d'autres cultures, en ajustant les jeux de données et les classes.

Enfin, ce travail montre qu'un tel système peut être conçu en autonomie par une petite équipe, dans une logique ouverte et réutilisable. Bien qu'à ce stade aucun test collectif structuré n'a encore été réalisé, il constitue un socle utile pour une phase 2, axée sur les tests en conditions réelles et l'amélioration du système pour un usage terrain, dans un esprit de réciprocité et d'application concrète.

Perspectives

Le prototype développé ouvre plusieurs pistes d'évolution.

- Généralisation à d'autres maladies ou espèces : La chaîne de traitement (constitution du jeu de données, entraînement, conversion, intégration mobile) est entièrement réplicable. Un projet similaire pourrait être mené sur d'autres virus du melon, ou élargi à d'autres cultures présentant des symptômes visuels (tomate, vigne, etc).

Dans cette perspective, une phase d'extension du jeu de données, en particulier pour les stades sévères, serait nécessaire pour renforcer la robustesse du modèle. L'objectif serait de disposer de plusieurs centaines, voire milliers d'images par classe, afin d'améliorer la généralisation et de préparer un déploiement en conditions réelles.

- Ajout d'un module de traçabilité :

Un historique local ou exportable (par exemple en CSV) pourrait être ajouté afin de suivre des diagnostics dans le temps, associant à chaque image sa date, sa localisation et le stade détecté.

- Plateforme collaborative :

L'application pourrait s'intégrer à une plateforme web ouverte, dans une logique de science participative, permettant à des collecteurs de transmettre leurs observations et de contribuer à la veille sanitaire.

- Amélioration du traitement IA:

Des pistes plus avancées peuvent être explorées :

- étude du développement de la maladie sur plante entière (et non feuille détachée) comme l'ont déjà entrepris Fuentes *et al.* (2017) pour la tomate ;
- segmentation des zones atteintes (et non une simple classification) ;
- analyse de courtes vidéos pour stabiliser les prédictions ;
- intégration de modèles multitâches (diagnostic, géolocalisation, suivi temporel).

Ces développements pourraient être amorcés dans le cadre d'un stage de Master 2, notamment pour explorer l'extension du diagnostic de la feuille isolée à la plante entière, étape plus complexe nécessitant un jeu de données adapté.

Références

- Barbedo, J. G. A. (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, 180, 96-107. <https://doi.org/10.1016/j.biosystemseng.2019.02.002>
- Desbiez, C. (2020). The never-ending story of cucurbits and viruses. *Acta Horticulturae*, 1294, 173-192. <https://doi.org/10.17660/actahortic.2020.1294.23>
- Díaz-Pendón, J. A., Fernández-Muñoz, R., Gómez-Guillamón, M. L., & Moriones, E. (2005). Inheritance of resistance to Watermelon mosaic virus in *Cucumis melo* that impairs virus accumulation, symptom expression, and aphid transmission. *Phytopathology*, 95(7), 840-846. <https://doi.org/10.1094/phyto-95-0840>
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311-318. <https://doi.org/10.1016/j.compag.2018.01.009>
- Fuentes, A., Yoon, S., Kim, S., & Park, D. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9), 2022. <https://doi.org/10.3390/s17092022>
- Gilbert, R. Z., Kyle, M. M., Munger, H. M., & Gray, S. M. (1994). Inheritance of resistance to Watermelon Mosaic Virus in *Cucumis melo* L. *HortScience*, 29(2), 107-110. <https://doi.org/10.21273/horts.29.2.107>
- Kang, B., Yeam, I., & Jahn, M. M. (2005). Genetics of plant virus resistance. *Annual Review of Phytopathology*, 43(1), 581-621. <https://doi.org/10.1146/annurev.phyto.43.011205.141140>
- Korte, A., & Farlow, A. (2013). The advantages and limitations of trait analysis with GWAS: a review. *Plant Methods*, 9(1), 29. <https://doi.org/10.1186/1746-4811-9-29>
- Lecoq, H., & Desbiez, C. (2012). Viruses of cucurbit crops in the mediterranean region. *Advances in Virus Research*, 84, 67-126. <https://doi.org/10.1016/b978-0-12-394314-9.00003-8>
- López-Martín, M., Montero-Pau, J., Ylla, G., Gómez-Guillamón, M. L., Picó, B., & Pérez-De-Castro, A. (2024). Insights into the early transcriptomic response against watermelon mosaic virus in melon. *BMC Plant Biology*, 24(1), 58. <https://doi.org/10.1186/s12870-024-04745-x>
- Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1419. <https://doi.org/10.3389/fpls.2016.01419>
- Mrisho, L. M., Mbilinyi, N. A., Ndalahwa, M., Ramcharan, A. M., Kehs, A. K., McCloskey, P. C., Murithi, H., Hughes, D. P., & Legg, J. P. (2020). Accuracy of a smartphone-based object detection model, PlantVillage Nuru, in identifying the foliar symptoms of the viral diseases of Cassava-CMD and CBSD. *Frontiers in Plant Science*, 11, 590889. <https://doi.org/10.3389/fpls.2020.590889>
- Palomares-Rius, F. J., Viruel, M. A., Yuste-Lisbona, F. J., López-Sesé, A. I., & Gómez-Guillamón, M. L. (2011). Simple sequence repeat markers linked to QTL for resistance to Watermelon mosaic virus in melon. *Theoretical and Applied Genetics*, 123(7), 1207-1214. <https://doi.org/10.1007/s00122-011-1660-2>



Cet article est publié sous la licence Creative Commons (CC BY-SA). <https://creativecommons.org/licenses/by-sa/4.0/>.

Pour la citation et la reproduction de cet article, mentionner obligatoirement le titre de l'article, le nom de tous les auteurs, la mention de sa publication dans la revue « NOV'AE », la date de sa publication et son URL.