Reconnaissance de plantes
Rendu 2 : Modélisation

DataScientest - Promotion Février 2024

—

Équipe

Felipe Souto

Nicolas Papegaey

Contexte et motivation

La détection des maladies des plantes est un enjeu crucial pour l'agriculture moderne. Les maladies végétales peuvent entraîner des pertes de rendement importantes, menacer la sécurité alimentaire, et engendrer des coûts élevés pour les agriculteurs en termes de traitements et de prévention. Dans le contexte du changement climatique, la propagation de nouvelles maladies ou l'intensification des maladies existantes devient un risque accru.

L'objectif de ce projet est donc de développer un modèle capable de détecter automatiquement les maladies végétales à partir de simples images de feuilles, ce qui peut aider les agriculteurs à prendre des décisions précoces et informées, réduisant ainsi les pertes économiques et améliorant l'efficacité des traitements.

Classification du problème

Type de problème

Ce projet porte sur la détection automatisée de maladies végétales à partir d'images de feuilles. En utilisant des techniques de vision par ordinateur, chaque image est classée dans l'une des 38 catégories, représentant soit des maladies spécifiques (ex. : "Tomato Yellow Leaf Curl Virus"), soit un état sain ("Healthy").

Le problème que nous abordons est un problème de classification multi-classes supervisée, où nous entraînons un modèle capable de prédire une étiquette discrète (la maladie ou l'état sain) à partir des caractéristiques visuelles extraites des images. L'objectif est de permettre une reconnaissance automatique et précise des maladies à partir de simples photos de feuilles de plantes.

Tâche de machine learning

Ce projet correspond à une tâche de reconnaissance d'images appliquée à la détection de maladies végétales. En utilisant des techniques de vision par ordinateur, nous tentons de classifier les images de feuilles dans des catégories correspondant soit à des maladies spécifiques (ex: “Tomato Yellow Leaf Curl Virus”), soit à un statut sain (“Healthy”).

Cette tâche est similaire à d'autres applications de diagnostic automatisé basées sur des images, telles que la détection de pathologies en imagerie médicale ou industrielle, où l'identification précise de motifs visuels est cruciale pour la prise de décision.

Métrique de performance principale

La métrique principale retenue pour ce projet est l'accuracy, qui mesure la proportion d'images correctement classées dans leur catégorie respective. L'accuracy est une métrique adaptée lorsque les classes sont équilibrées, ou que l'objectif est de maximiser le nombre de prédictions correctes globalement.

Toutefois, dans un dataset déséquilibré comme PlantVillage, où certaines classes sont sur-représentées, l'accuracy seule peut donner une vision incomplète des performances du modèle. Elle peut masquer les erreurs sur les classes minoritaires. Pour cette raison, il est crucial de ne pas s'appuyer uniquement sur cette métrique, mais de la compléter avec d'autres mesures plus appropriées aux déséquilibres de classe.

Autres métriques utilisées

En plus de l'accuracy, nous avons également utilisé plusieurs métriques d'évaluation complémentaires pour obtenir une vision plus fine de la performance du modèle, notamment dans le cas des classes sous-représentées :

- Précision (Precision) : La précision mesure la proportion de prédictions correctes parmi celles effectuées pour une classe donnée. Elle est essentielle lorsque les faux positifs (prédictions incorrectes pour une classe donnée) doivent être minimisés.
- Rappel (Recall) : Le rappel mesure la proportion de prédictions correctes par rapport au nombre total de cas réels d'une classe. Elle est particulièrement utile pour évaluer les performances sur les classes minoritaires.
- Score F1 : Le score F1 est la moyenne harmonique entre la précision et le rappel, offrant un bon compromis entre ces deux métriques. Il est utile pour mesurer la performance du modèle sur des classes déséquilibrées.

Ces métriques sont cruciales pour évaluer les performances du modèle sur les classes minoritaires. Par exemple, la précision et le rappel aident à comprendre si le modèle est biaisé en faveur des classes majoritaires, tandis que le score F1 permet un équilibre entre la précision et le rappel, particulièrement pertinent dans les contextes de déséquilibre.

Pour ce faire, nous avons utilisé des outils comme le rapport de classification et la matrice de confusion, qui offrent une vue détaillée de la performance du modèle sur chaque classe.

Contexte du déséquilibre des classes

Le dataset PlantVillage présente un fort déséquilibre entre les différentes classes, certaines maladies étant beaucoup plus représentées que d'autres. Par exemple, des catégories comme "Tomato Yellow Leaf Curl Virus" sont sur-représentées tandis que d'autres comme "Potato Healthy" ont beaucoup moins d'exemples. Ce déséquilibre peut influencer négativement la capacité du modèle à généraliser, et il est donc nécessaire d'en tenir compte dans le choix des algorithmes et dans l'évaluation des résultats. L'accuracy seule, sans les métriques complémentaires, risque de donner une impression faussement positive des performances du modèle, en surestimant sa capacité à bien classer les images.

Choix du modèle et optimisation

Démarche

Dans le cadre du projet de reconnaissance d'images de plantes et de maladies, nous avons exploré plusieurs algorithmes afin d'identifier une solution efficace permettant d'obtenir de bon résultats dans la classification d'un dataset de diverses classes (38).

Nous nous sommes donné comme objectif de mettre en place un modèle qui soit à la fois efficace dans la distinction des différentes classes de plantes, mais également dans la reconnaissance des détails relatifs aux caractéristiques propres de chaque maladie.

Notre première étape a été de travailler sur un réseau de neurones convolutif (CNN) de base créé à partir de la librairie Keras. Ce modèle initial comprenait deux couches de convolution suivies de couches de pooling, permettant de capturer des caractéristiques visuelles essentielles des images. L'objectif de cette première approche était d'établir un point de référence pour la complexité du problème, en fournissant un modèle baseline sur lequel nous pourrions itérer.

La deuxième étape a impliqué des itérations successives sur ce modèle de base pour améliorer l'accuracy, tout en réduisant l'overfitting et en ajustant les hyperparamètres.

La troisième étape nous a amené à explorer la mise en place du transfer-learning en partant de modèles pré-entraînés : **MobileNet**, **VGG**, **ResNet** et **NasNet** dans le but d'améliorer la capacité de notre modèle à généraliser et à reconnaître plus précisément les symptômes distinctifs des maladies des plantes. Cette approche nous a permis d'améliorer considérablement les performances du modèle, tout en réduisant le temps d'entraînement nécessaire grâce à l'utilisation de caractéristiques déjà optimisées pour la reconnaissance d'images complexes.

Cependant les modèles VGG ,ResNet etNasNet bien qu'efficaces dans de nombreux scénarios, n'ont pas montré des résultats compétitifs ici en raison de la complexité de leurs architectures, qui a conduit à un sur-apprentissage rapide. De plus, les ressources limitées sur Google Colab ont prolongé les temps d'entraînement de manière significative, freinant l'expérimentation avec ces architectures.

CNN - Modèle de base

Dans cette première version, nous avons implémenté un réseau de neurones convolutif (CNN) de base avec les spécifications suivantes :

- **Architecture** : Deux couches de convolution (32 et 64 filtres) avec des fonctions d'activation ReLU, suivies de couches de max pooling. La couche dense intermédiaire comprend 256 neurones avant la couche de sortie, qui utilise une activation softmax pour la classification multi-classes.

- **Taille des images** : 224x224 pixels.

- **Hyperparamètres de base** :
  - Taille de batch : 32.
  - Taux d'apprentissage initial : 0.0001.
  - Nombre d'époques : initialement 5, puis augmenté à 20 en itération 2.

- **Optimisation** :
  - Ajout d'un **Early Stopping** (itération 2) pour éviter l'overfitting.
  - Ajout du **ReduceLROnPlateau** et **Dropout** (itération 3) pour stabiliser l'apprentissage et améliorer la régularisation.

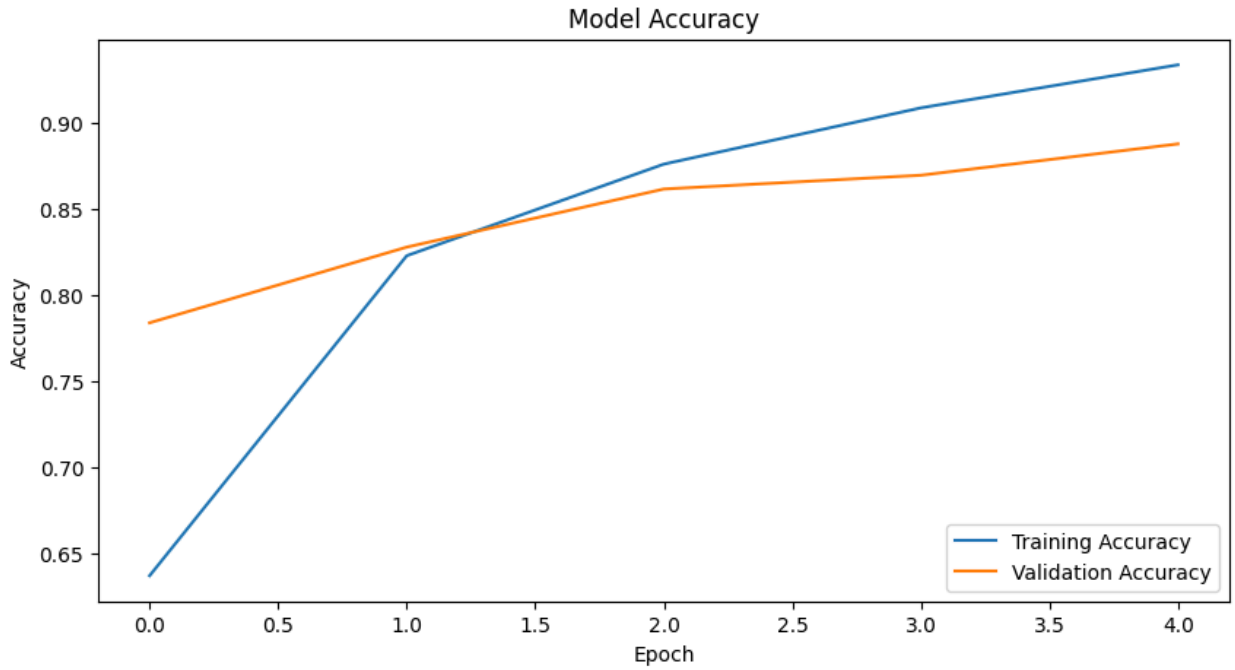
Tableau des amélioration et résultats

Itération	**Améliorations effectuées**	**Accuracy (Entraînement /Validation)**	**Perte (Loss)**	**Macro Average (Accuracy/ Recall)**	**Weighted Average (Accuracy /Recall/F1)**
Itération 1	Modèle de base (5 époques)	93% / 89%	Diminue régulièrement	86% / 84%	89% / 89% / 89%
Itération 2	Ajout de l'**Early Stopping** et augmentation des époques (20 époques)	99% / 90% (stabilisé après 8 époques)	Validation Loss à 0.33, augmente après la 13ème époque	89% / 87%	91% / 91% / 91%

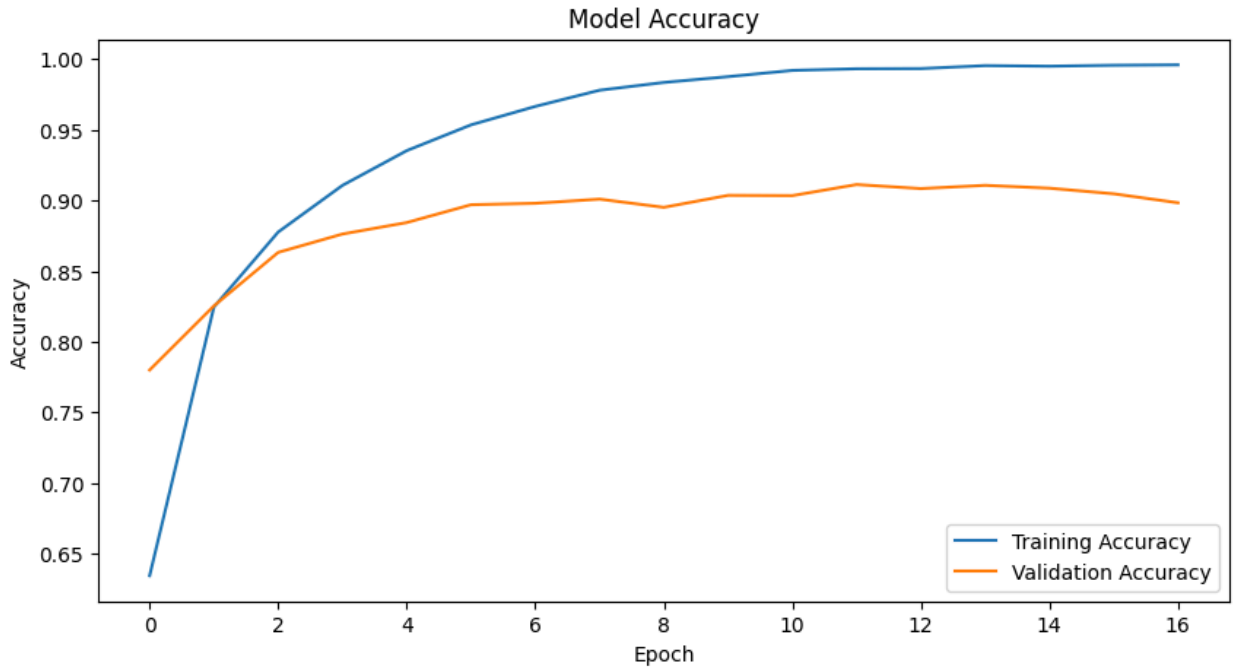
Itération 3	Ajout de **ReduceLROnPlateau** et **Dropout (0.2)**	99% / 91% (stable sur 20 époques)	Validation Loss stabilisée à 0.32	88% / 88%	91% / 91% / 91%

Analyse des résultats

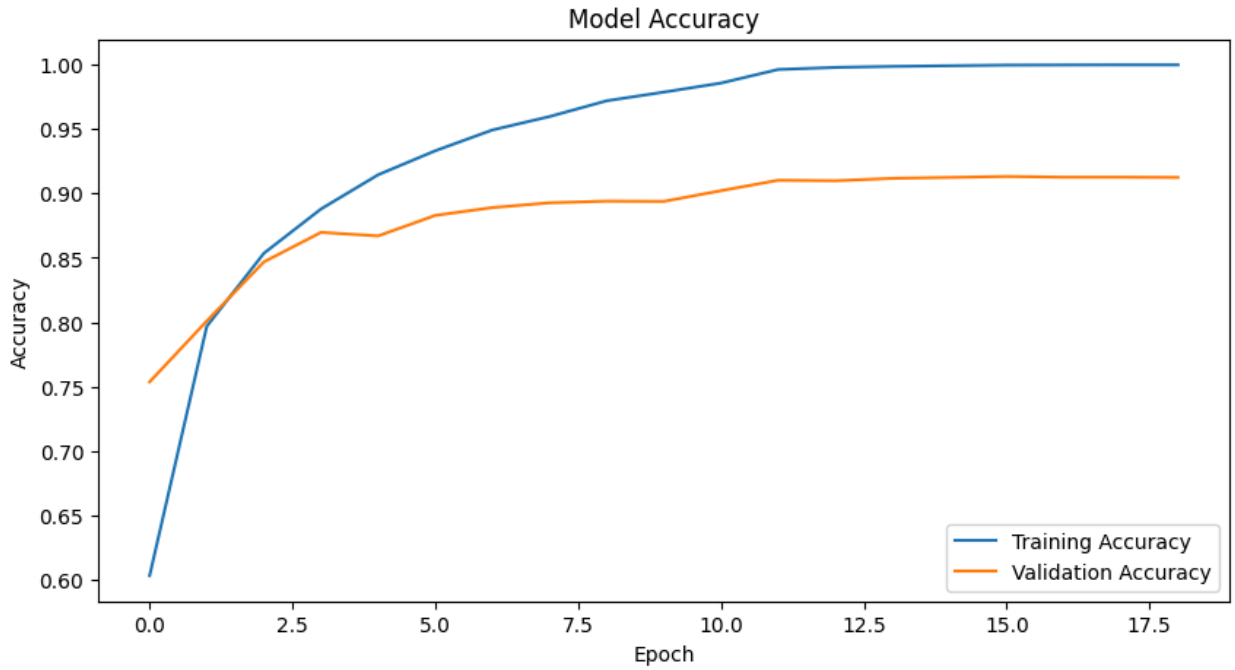
- **Itération 1** : Le modèle de base a montré de bons résultats, avec une accuracy de validation à 89%. Cependant, il y avait encore un certain potentiel d'amélioration, notamment pour les classes moins fréquentes.



- **Itération 2** : L'ajout de l'early stopping et l'augmentation des époques ont permis d'améliorer la performance, bien que des signes d'overfitting apparaissent après la 13ème époque.



- **Itération 3** : L'introduction du ReduceLROnPlateau a permis d'adapter dynamiquement le taux d'apprentissage lorsque les performances du modèle stagnaient, tandis que le Dropout a amélioré la régularisation en empêchant le sur-apprentissage des données d'entraînement. Le modèle final montre une performance solide avec une accuracy de validation stable à 91%, tout en maintenant des scores équilibrés de précision, rappel et F1-score, ce qui est essentiel dans le contexte des classes déséquilibrées.



Références de notebooks

Vous pouvez accéder aux notebooks détaillant les différentes itérations du modèle CNN à partir des liens ci-dessous :

- Notebook de l'Itération 1 : Modèle de base (5 époques)
- Notebook de l'Itération 2 : Ajout de l'Early Stopping et augmentation des époques (20 époques)
- Notebook de l'Itération 3 : Introduction du ReduceLROnPlateau et Dropout

Conclusion

Le modèle CNN a été amélioré à travers plusieurs itérations, en ajoutant des techniques d'optimisation qui ont permis de stabiliser l'apprentissage et de réduire l'overfitting. Le modèle final atteint une performance globale solide avec une accuracy de validation stable à 91% et une bonne gestion du déséquilibre des classes.

CNN - Modèle amélioré

Dans cette seconde version nous avons implémenté un réseau de neurones convolutif (CNN) plus élaboré avec plusieurs couches de convolution, de pooling et de normalisation de lot pour stabiliser l'entraînement.

- **Architecture : Trois couches de convolution** (32, 64, et 128 filtres), suivies de couches de **batch normalization** et de **max pooling**. Ce modèle améliore la stabilité et la capacité de généralisation par rapport au modèle précédent grâce à l'ajout d'une couche de convolution supplémentaire (128 filtres) pour capturer des caractéristiques plus complexes. De plus, la couche de **Global Average Pooling** remplace la traditionnelle couche **Flatten**, réduisant le risque d'overfitting en réduisant la dimensionnalité des données sans perte d'information importante. Une couche dense intermédiaire de 256 neurones est ensuite utilisée avant la couche de sortie, qui applique une **activation softmax** pour la classification multi-classes. Un **dropout de 50%** est ajouté pour renforcer la régularisation et stabiliser davantage l'entraînement.

- **Taille des images** : 224x224 pixels.

- **Hyperparamètres de base** :
  - Taille de batch : 32.
  - Taux d'apprentissage initial : 0.0001.
  - Nombre d'époques : 20

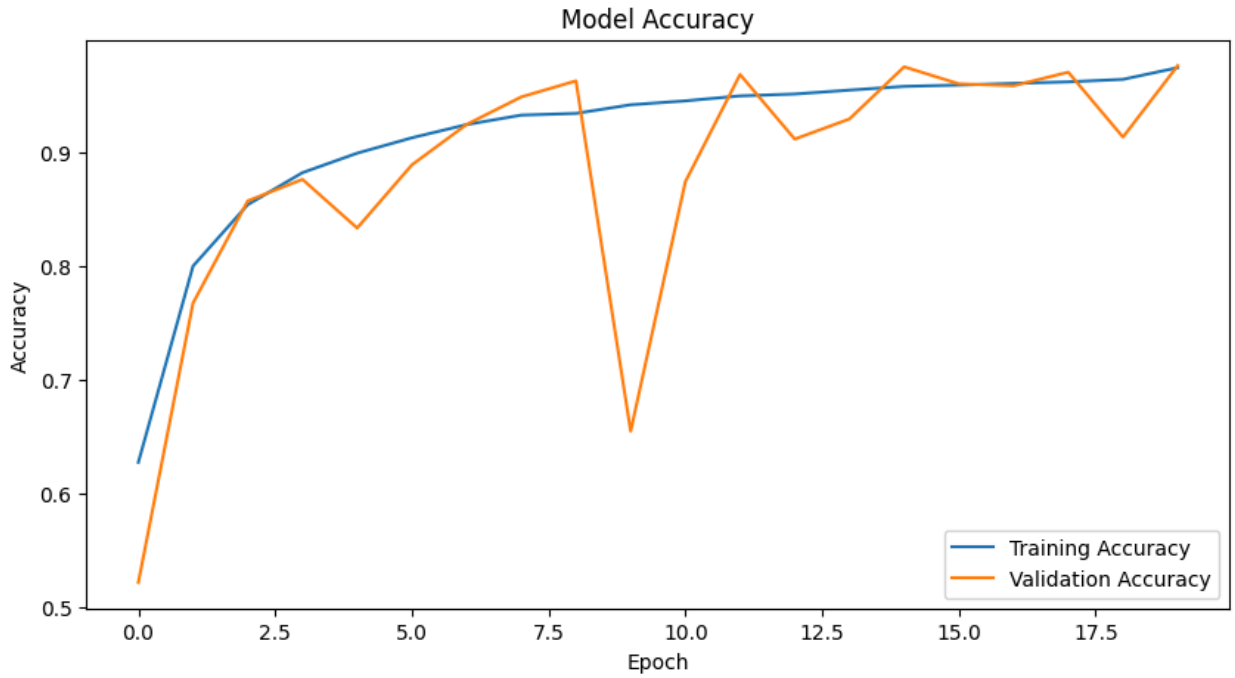
Tableau des amélioration et résultats

Itération	**Améliorations effectuées**	**Accuracy (Entraînement /Validation)**	**Perte (Loss)**	**Macro Average (Accuracy / Recall)**	**Weighted Average (Accuracy/ Recall/F1)**
Itération 1	Modèle amélioré avec Batch Normalization, 3 couches	97.23% / 97.56% Pics d'accuracy observés en	Validation Loss stabilisée à 0.0745	97% / 96%	98% / 97% / 97%

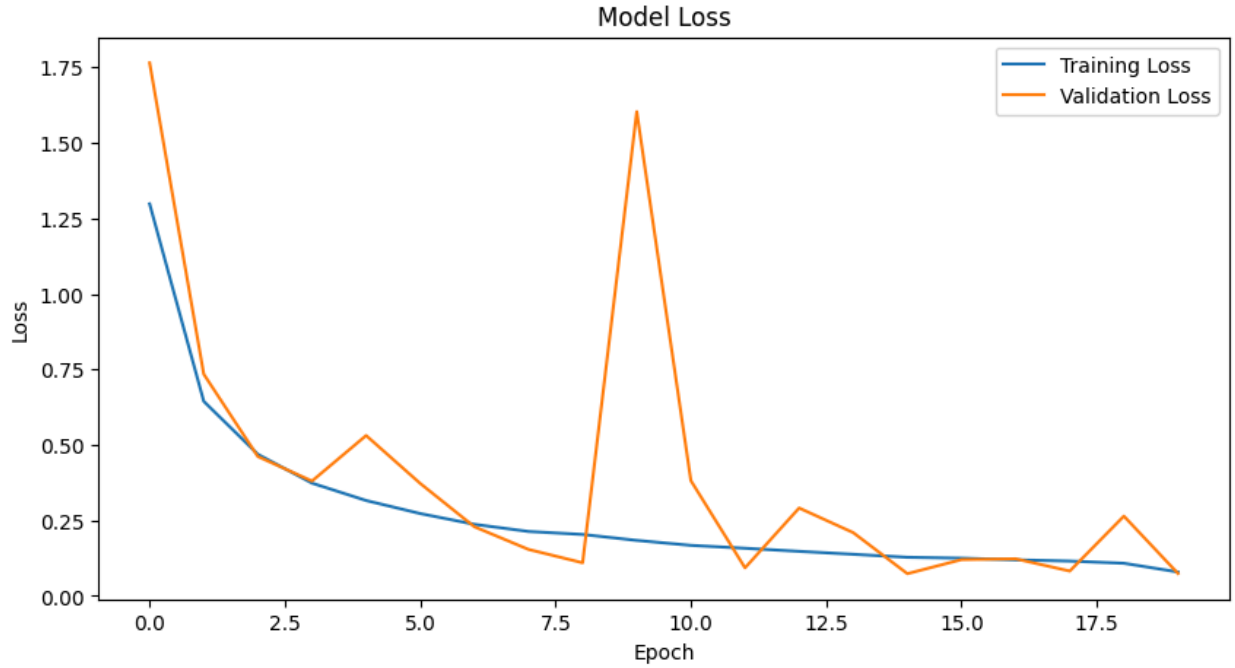
	convolutionnelles, GlobalAveragePooling, et Dropout à 50%	validation probablement dus à l'effet du Dropout.			

Analyse des résultats

Le modèle amélioré a permis d'atteindre une **accuracy d'entraînement de 97.23%** et une **accuracy de validation de 97.56%** après 20 époques.



La **perte de validation** a été réduite et stabilisée à **0.0745**, montrant une bonne généralisation du modèle.



Références de notebooks

Vous pouvez accéder aux notebooks détaillant les différentes itérations du modèle CNN amélioré à partir des liens ci-dessous :

- Notebook de l'Itération 1 : Modèle amélioré avec Batch Normalization, GlobalAveragePooling, et Dropout à 50%

Conclusion

Comparé au modèle de base, qui atteignait une **accuracy de validation de 91%** et une **validation loss stabilisée à 0.32** sur la 3ème itération, ce modèle amélioré montre une performance nettement supérieure. Avec une **accuracy de validation de 97.56%** et une **validation loss de 0.0745**, le nouveau modèle présente une meilleure capacité de généralisation et une plus grande robustesse.

CNN - Modèle MobileNetV1 avec Transfer Learning

Dans cette troisième version, nous avons utilisé une approche de **transfer learning** avec **MobileNetV1**, un modèle pré-entraîné sur ImageNet.

- **Architecture :** MobileNetV1 extrait des caractéristiques complexes grâce à ses couches convolutionnelles optimisées pour les dispositifs à faible puissance. Une couche de **GlobalAveragePooling** réduit la dimensionnalité des caractéristiques extraites. Ensuite, une couche dense de **256 neurones** avec activation **ReLU** traite ces caractéristiques, suivie d'un **dropout** de 20% pour prévenir le sur-apprentissage. Enfin, une couche de sortie **softmax** réalise la classification multi-classes.

- **Taille des images** : 224x224 pixels.

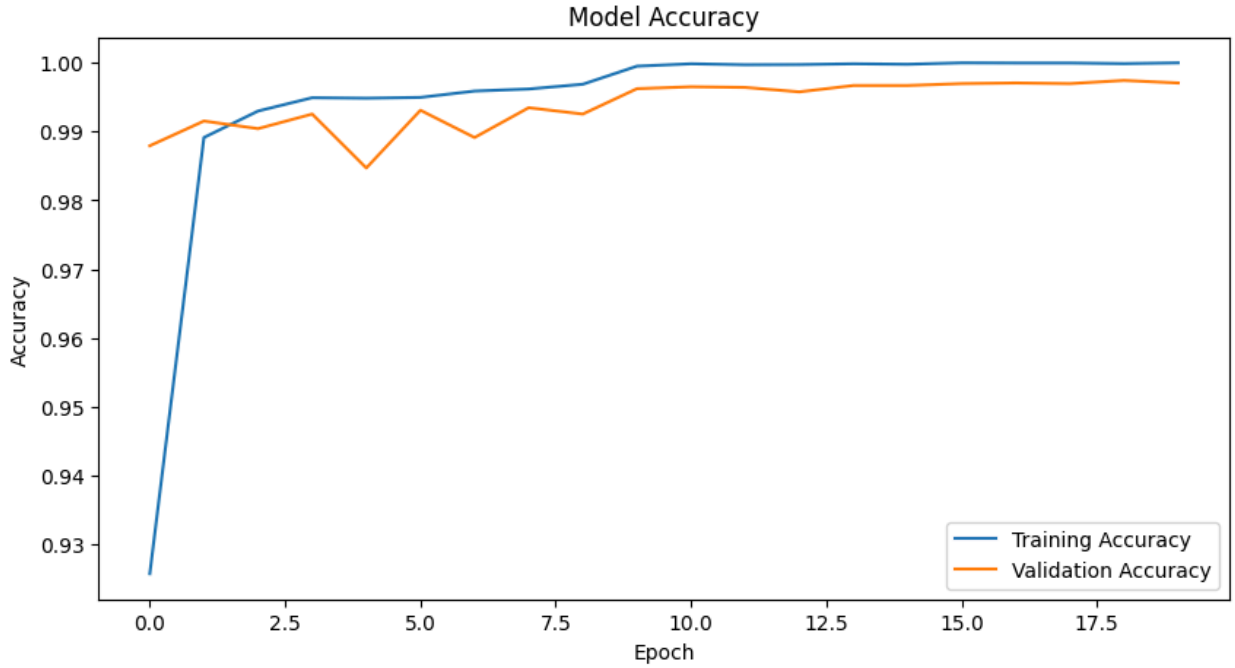
- **Hyperparamètres de base** :
  - Taille de batch : 32.
  - Taux d'apprentissage initial : 0.0001.
  - Nombre d'époques : 20

Tableau des amélioration et résultats

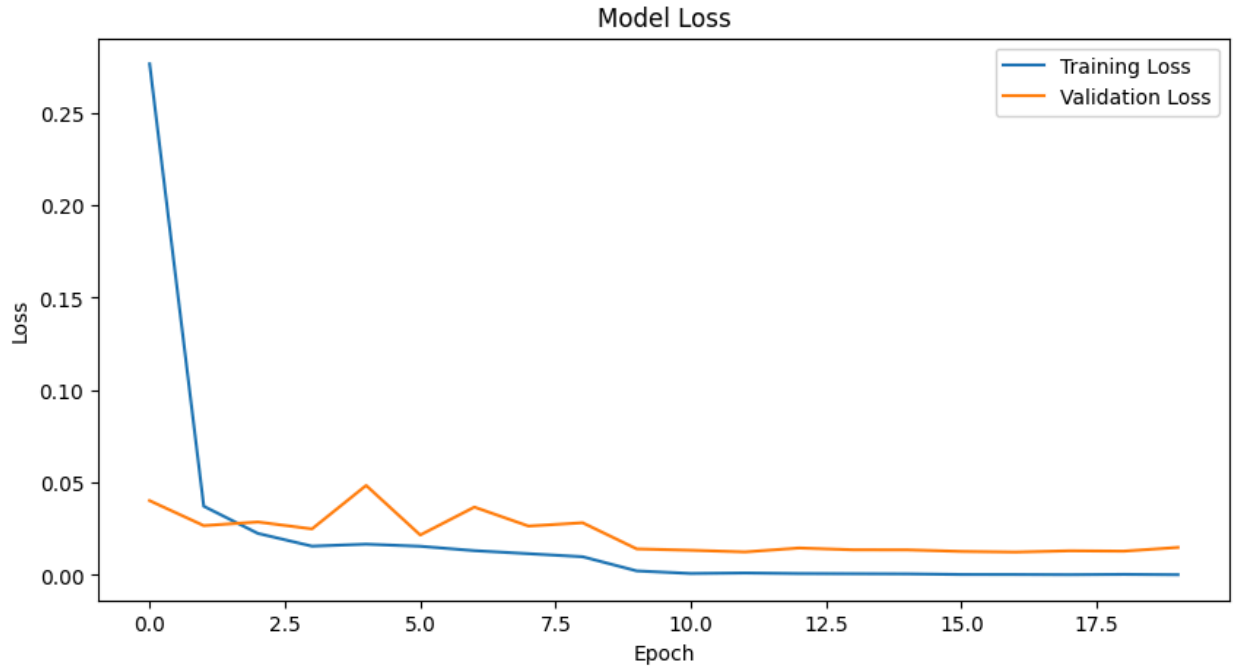
Itération	Améliorations effectuées	Accuracy (Entraînement/ Validation)	Perte (Loss)	Macro Average (Accuracy /Recall)	Weighted Average (Accuracy/ Recall/F1)
Itération 1	Modèle basé sur MobileNet pré-entraîné (transfer learning), avec GlobalAveragePooling et Dropout à 20%	100% / 99.71%	Validation Loss stabilisée à 0.0149	100% / 100%	100% / 100% / 100%

Analyse des résultats

Le modèle MobileNet basé sur le **transfer learning** atteint une **accuracy d'entraînement de 100%** et une **accuracy de validation de 99.71%** après 20 époques.



La **perte de validation** est faible et stabilisée à **0.0149**, ce qui indique une excellente capacité de généralisation sur des données non vues, tout en atteignant des performances presque parfaites sur l'ensemble de validation.



Références de notebooks

Vous pouvez accéder aux notebooks détaillant les différentes itérations du modèle MobileNetV1 avec Transfer Learning à partir des liens ci-dessous :

- Notebook de l'Itération 1 : Modèle basé sur MobileNet pré-entraîné, avec GlobalAveragePooling et Dropout à 20%

Conclusion

Le modèle basé sur **MobileNetV1** avec **transfer learning** a surpassé les 2 modèles précédents. Ce modèle affiche des performances quasiment parfaites avec une **accuracy de validation de 99.71%** et une **validation loss de 0.0149**.

Les principales raisons de cette amélioration sont l'utilisation de **MobileNetV1 pré-entraîné**, qui a permis de transférer des connaissances d'un grand dataset généraliste comme ImageNet, ainsi que l'optimisation fine via le taux d'apprentissage ajusté et le **Dropout**. Cela a permis d'atteindre un équilibre optimal entre la performance et la généralisation.

Autres modèles essayés

Dans le cadre de ce projet, nous avons testé d'autres architectures de Transfer Learning afin de maximiser les performances de notre modèle de reconnaissance d'images. Cependant, ces modèles n'ont pas été approfondis pour diverses raisons.

Le premier modèle de transfer learning a avoir été testé est le **VGG16** (vgg sur 30 époques). La longueur de l'entraînement et les résultats qui restaient en deçà de ceux obtenus par le CNN amélioré, nous ont poussé à explorer d'autres modèles.

Après avoir testé **MobilenetV1,** nous avons ensuite exploré l'architecture **ResNet50** (ResNet50 sur 20 époques) avec laquelle nous avons obtenu des résultats peu performants probablement à cause de la complexité de l'architecture et des ressources disponibles sur Google Colab.

Nous avons testé une version **MobilenetV2** , cependant, malgré les mêmes paramètres de départ que ceux utilisés avec succès pour **MobileNetV1**, nous avons rencontré des difficultés à faire converger le modèle. Étant donné les limites de nos ressources de calcul sur notre plan Google Collab et les excellents résultats obtenus avec **MobileNetV1**, nous avons pris la décision de ne pas approfondir davantage ce modèle.

Optimisation des paramètres

Modification des paramètres de dropout (régularisation)

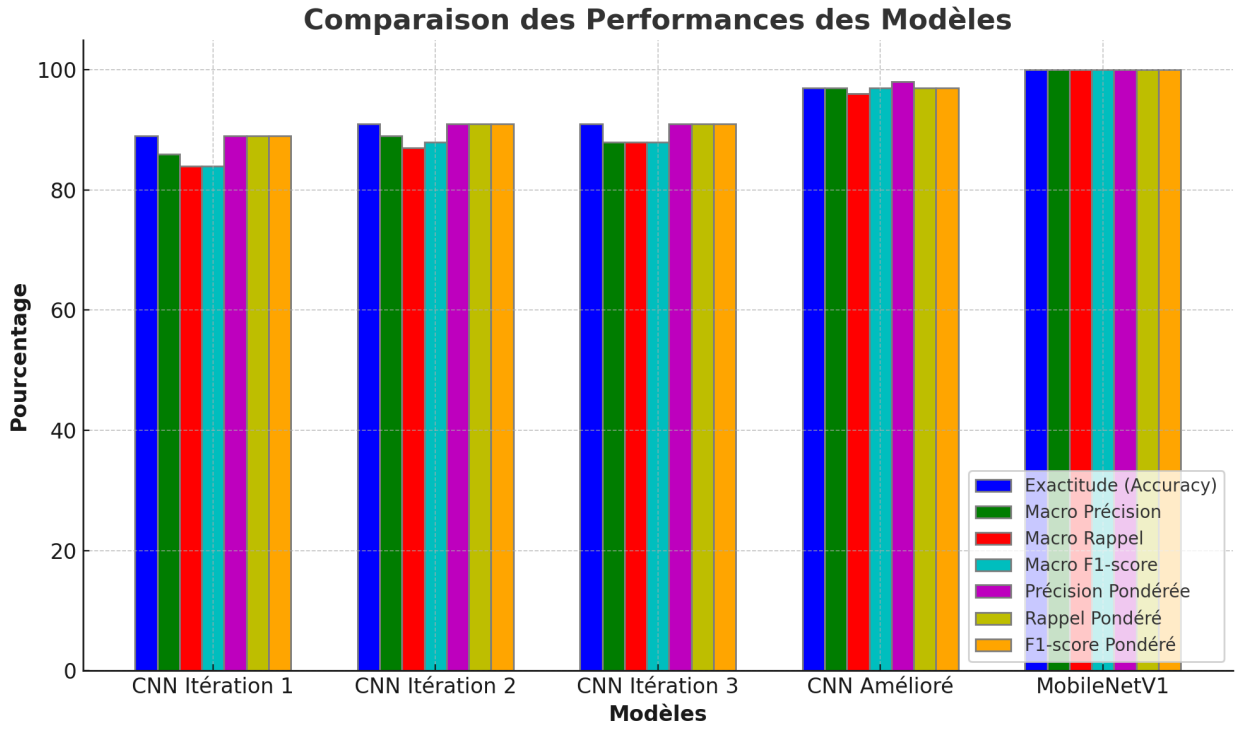
Lors des phases d'entraînement, la valeur de **dropout** a pu être augmentée dans certains cas pour éviter les phases de surapprentissage. La base de départ était de 0.2 et a été augmentée jusqu'à des valeurs entre 0.5 et 0.6.

Modification du taux d'apprentissage de départ

Lors de l'approche de la solution optimale, nous avons pu constater des fluctuations qui empêchaient le modèle de se stabiliser. Dans ces cas, une diminution du taux d'apprentissage initial (par exemple passage de 0.0001 à 0.00005) a permis un apprentissage plus stable.

Graphe récapitulatif

Le graphe suivant présente l'évolution des principales métriques au travers des itérations sur les modèles étudiés en détail.

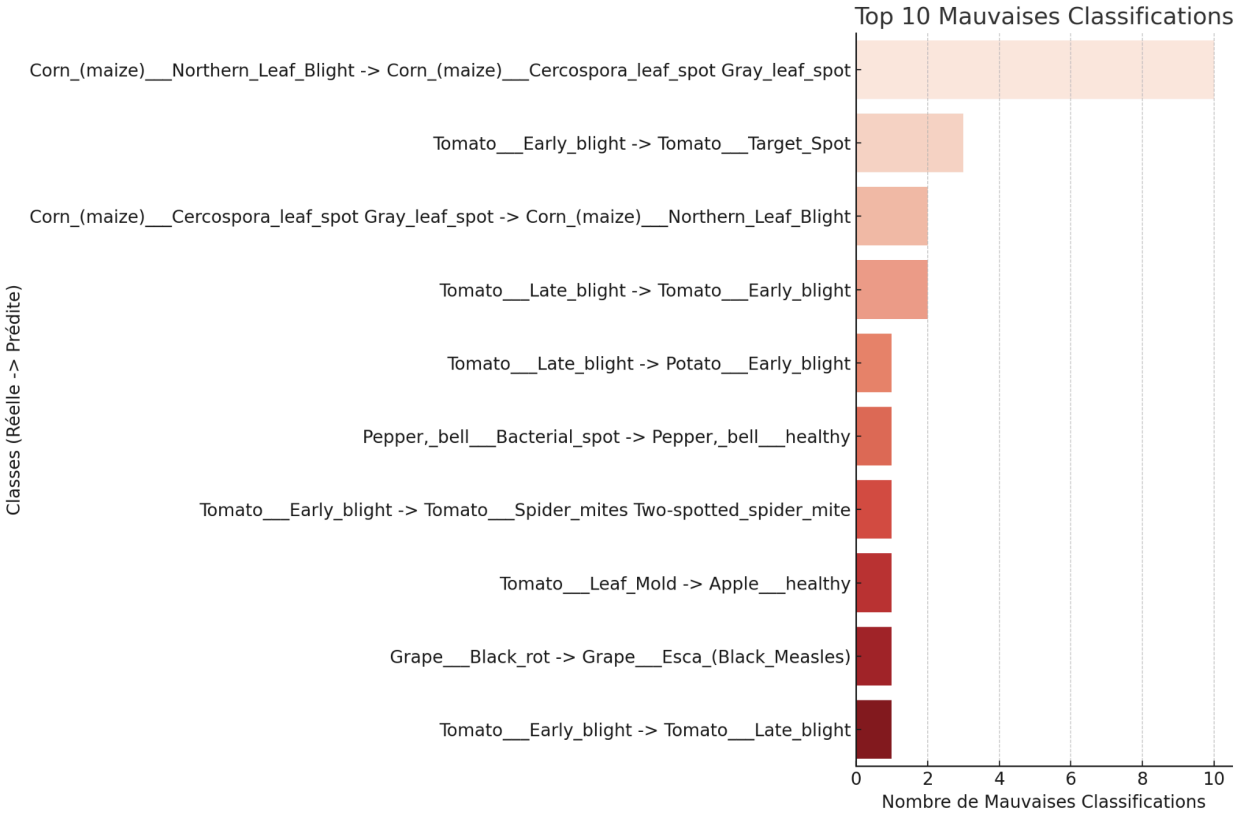


Interprétation des résultats

Analyse des erreurs du modèle

L'outil principal que nous avons utilisé pour analyser les erreurs du modèle a été la matrice de confusion.

Etant donné les très bons résultats de la classification et le nombre de classes (38), pour des soucis de lisibilité nous avons privilégié un focus sur les erreurs les plus fréquentes plutôt que l'affichage complet de la matrice de confusion :



Analyse de la Matrice de Confusion

L'analyse de cette matrice révèle une très faible confusion entre les différentes espèces de plantes. Les erreurs observées sont principalement des cas marginaux. Par exemple, il y a un cas où une image de **Tomate atteinte de Leaf Mold** (moisissure des feuilles) a été classée à tort comme étant un **Pommier sain**. On peut supposer que le modèle s'est

surtout basé sur la forme allongée de la feuille mais à mal appris certains détails différenciateurs comme la courbure de la feuille.

Tomate Leaf Mold classée en Pommier sain

Exemple de pommier sain

Confusions au sein d'une même espèce

Les autres erreurs concernent essentiellement des prédictions erronées de maladies au sein d'une même espèce, en particulier le maïs. Par exemple, la **Northern Leaf Blight** (Brûlure septentrionale) a été confondue 10 fois avec la **Cercospora Leaf Spot** (Tâche foliaire de Cercospora). Ces deux maladies présentent probablement des caractéristiques visuelles similaires, telles que des tâches et des lésions sur les feuilles, ce qui peut expliquer ces confusions.

 Northern Leaf Blight classée en Cercospora Leaf Spot	 Exemple de Cercospora Leaf Spot	 Cercospora Leaf Spot classée en Northern Leaf Blight

Confusions liées aux maladies de la tomate

Les confusions les plus fréquentes concernent les maladies de la tomate, qui représente 33% des images dans le dataset. Certaines maladies de la tomate, comme la **Brûlure précoce** (**Early Blight**) et la **Brûlure tardive** (**Late Blight**), partagent des symptômes visuels très similaires. Ces symptômes ne peuvent parfois être différenciés que par la gravité ou l'étendue des lésions, rendant la tâche difficile pour le modèle de classification.

Il est également possible que certaines erreurs soient dues à une mauvaise classification initiale des images dans le dataset, induisant ainsi le modèle d'apprentissage en erreur.

 Early Blight classée en Late Blight	 Late Blight classée en Early Blight	 Early Blight classée en Target Spot

 Tomate Late Blight classée en Pomme de terre Early Blight	 Exemple de Early Blight	 Exemple de Late Blight

Interprétabilité

Pour mieux comprendre le fonctionnement du modèle de classification, nous avons utilisé deux méthodes d'interprétabilité : **GradCAM** (Gradient-weighted Class Activation Mapping) et **LIME** (Local Interpretable Model-agnostic Explanations).

GradCAM

GradCAM nous a permis de visualiser les régions d'une image influençant le plus les décisions du modèle, révélant ainsi les zones auxquelles le modèle accorde le plus d'importance lors de la classification. Cela nous a aidé à identifier les erreurs de classification liées à un mauvais apprentissage des détails spécifiques à certaines maladies.

LIME

LIME est une méthode agnostique au modèle qui perturbe l'image d'entrée en modifiant différentes zones, et observe comment ces modifications affectent la prédiction. Cela aide à déterminer quelles parties de l'image contribuent le plus à la décision du modèle.

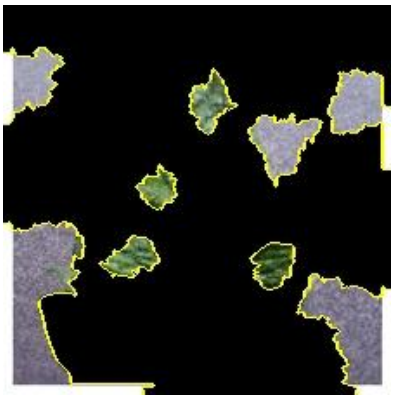
Exemple d'Interprétation pour une Image Mal Classifiée

Nous pouvons supposer que le modèle a bien détecté des taches caractéristiques de la brûlure (taches noires), comme le montre GradCAM. Cependant, il semble avoir mal interprété les zones où l'infection est plus avancée, notamment le trou en haut à gauche et la brûlure en bas à droite, comme confirmé par l'analyse LIME, qui n'a pas suffisamment mis en évidence ces parties critiques.

		
Late Blight classée en Early Blight	Interprétabilité GradCAM	Interprétabilité LIME

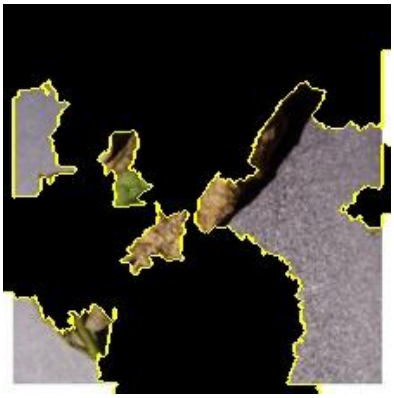
Influence de la Luminosité sur la Classification

Dans certains cas, la luminosité de l'image peut influencer la classification. Par exemple, une image étiquetée comme **saine** présente des taches blanches visibles à l'œil nu, probablement dues à l'éclairage lors de la prise de la photo. L'analyse GradCAM montre que la partie de la feuille contenant ces taches blanches a influencé la classification, menant à l'identification erronée du **virus Spider Mites**.

		
Tomate saine classée avec le virus Spider Mites	Interprétabilité GradCAM	Interprétabilité LIME

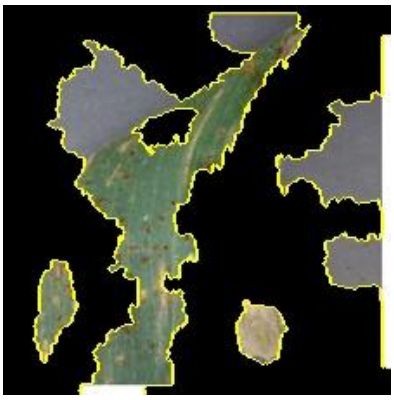
Confusion entre Deux Espèces pour une Maladie Commune

Un autre exemple concerne la **Late Blight**, une maladie commune à la pomme de terre et à la tomate. Bien que le modèle ait correctement identifié la maladie, il a eu du mal à distinguer l'espèce végétale, probablement à cause du stade avancé de l'infection. GradCAM montre que le modèle s'est concentré sur la partie recroquevillée de la feuille, tandis que LIME n'a pas mis en avant la forme de la feuille, qui aurait pu aider à mieux identifier l'espèce.

 Tomate Late Blight classifiée en Pomme de terre Late Blight	 Interprétabilité GradCAM	 Interprétabilité LIME

Confusion entre Deux Maladies sur une Même Feuille

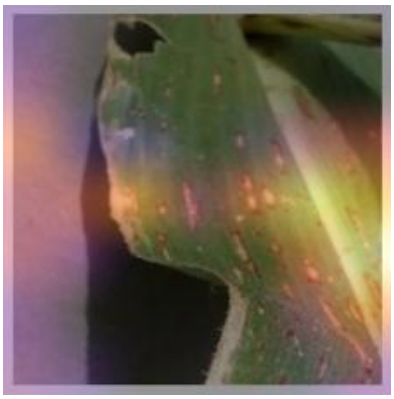
Dans cet exemple, la feuille présente des symptômes de deux maladies : **Northern Leaf Blight** (grande lésion nécrotique) et **Cercospora Leaf Spot** (petites taches dispersées).

 Cercospora Leaf Spot classifiée en Northern Leaf Blight	 Interprétabilité GradCam	 Interprétabilité Lime

GradCAM montre que l'erreur de classification est due au fait que le modèle s'est concentré sur la grande zone nécrotique à droite de la feuille. Cette zone est plus typique de la **Northern Leaf Blight**, alors que la **Cercospora Leaf Spot** est plutôt caractérisée par de petites taches bien définies.

L'analyse LIME révèle également que certaines parties du fond de l'image ont influencé la décision du modèle. Cela suggère que des éléments non pertinents, comme le fond, ont pu détourner l'attention du modèle des symptômes visibles sur la feuille, indiquant un possible manque de robustesse dans la segmentation.

Enfin , dans ce dernier cas, le modèle a probablement utilisé la couleur caractéristique des taĉhes commune des maladies Cercospora et Common Rust.

		
Maïs Cercospora classifié Maïs Common Rust	Interprétabilité GradCAM	Interprétabilité LIME

Améliorations basées sur l'analyse des erreurs

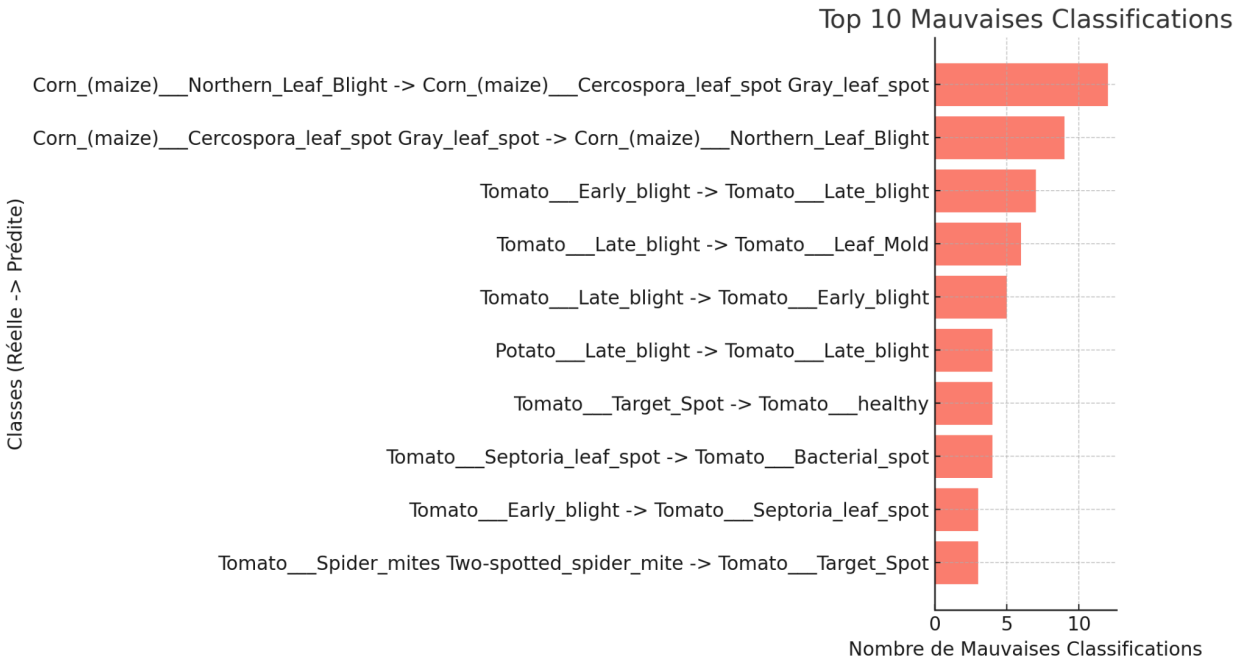
Les erreurs de classification soulignent l'importance de s'assurer que le modèle se concentre sur les caractéristiques clés des maladies et ne soit pas influencé par des éléments non pertinents, comme l'arrière-plan de l'image. Des améliorations dans la segmentation des images ou dans la mise en évidence des symptômes distinctifs pourraient renforcer la robustesse du modèle.

Segmentation des images du dataset initial

Les analyses d'interprétabilité nous ont amené à considérer l'utilisation du pré-traitement des images avec segmentation. En effet, l'interprétabilité LIME a montré sur les images en couleur une prise en compte non négligeable du fond. La segmentation permet d'uniformiser le fond afin que le modèle ne cible que les zones d'intérêt de la plante.

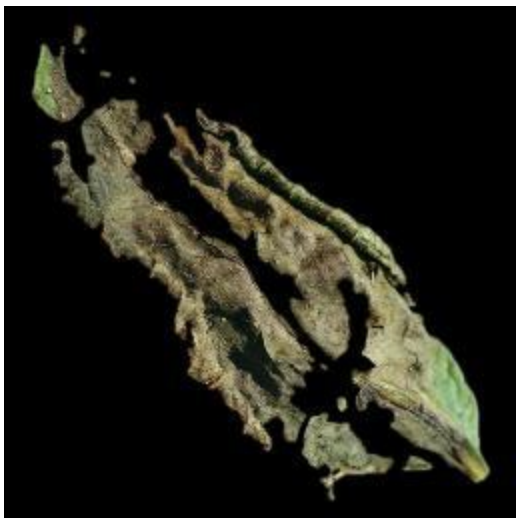
Le dataset contenait déjà les images segmentées. Nous avons utilisé cette partie du dataset pour créer un nouveau modèle.

L'effet de la segmentation semble être mineur dans la plupart des cas, avec quelques erreurs de classification apparaissant dans l'ensemble segmenté, mais absentes dans l'ensemble non segmenté. Cela peut suggérer que la segmentation élimine parfois des caractéristiques contextuelles ou des éléments de l'arrière-plan qui aident à la classification.



Il est à noter qu'une exploration de certaines images qui ont été mal classifiées par le modèle montre que le procédé de segmentation utilisé peut générer une perte d'information.

Le cas suivant (Pomme de terre Late Blight classifiée en Tomate Late Blight) :



On observe que certaines zones d'ombre de la feuille ont été confondues avec l'arrière-plan, ce qui entraîne une segmentation inexacte et déforme la représentation de la forme réelle de la feuille.

Le cas suivant (Raisin Black Rot classifiée en Tomate Early Blight) :

Dans ce cas, la forme complète de la feuille a été perdue, ne laissant que deux petites portions visibles dans l'image segmentée. Cela souligne l'importance de mettre en place un processus de contrôle rigoureux pour évaluer et garantir la qualité de la segmentation des images.

Nous n'avons pas eu accès aux informations détaillées concernant le mode de segmentation utilisé, mais il est évident que ce processus est essentiel pour garantir la qualité des images utilisées dans la classification. **Des erreurs de segmentation peuvent entraîner une perte d'informations précieuses**, ce qui affecte négativement les performances du modèle. Il est donc important de mettre en place un contrôle rigoureux pour vérifier la qualité de la segmentation afin d'assurer des résultats fiables. **Dans notre cas, la segmentation n'a pas apporté d'améliorations notables.**

Remarque : Depuis la soumission du rapport de modélisation, des ajustements méthodologiques ont été apportés pour optimiser l'entraînement du modèle et réduire le surapprentissage identifié dans la version initiale.

Conclusion de la phase de modélisation

Ce projet a permis de développer plusieurs modèles pour détecter les maladies des plantes à partir de photos de feuilles. Le modèle **MobileNetV1**, utilisant le **transfer learning**, a obtenu les meilleurs résultats avec une **précision de 99,71%** sur l'ensemble de validation. Ce modèle a bien géré le **déséquilibre des classes**.

Des outils comme **GradCAM** et **LIME** ont aidé à comprendre les erreurs de classification. Ils ont montré que certaines maladies aux symptômes similaires, comme la **Brûlure précoce** et la **Brûlure tardive** des tomates, sont souvent confondues. Le modèle se concentre parfois sur des zones inutiles, comme le fond ou la luminosité de l'image, ce qui fausse les résultats.

La segmentation d'images n'a pas toujours amélioré la précision. Dans certains cas, elle a même créé des erreurs. Par exemple, certaines feuilles ont été mal découpées, ce qui a perturbé la classification. La segmentation a parfois supprimé des détails importants qui aidaient à distinguer les maladies.

Pour améliorer encore les résultats, il est important de mieux **segmenter les images** et de corriger les **confusions entre maladies** similaires. Il faudrait aussi ajouter plus de données

pour les classes moins fréquentes en utilisant des techniques d'**augmentation de données** et en enrichissant le dataset.

Le modèle est très performant pour reconnaître les espèces de plantes. Cependant, pour des activités de surveillance spécifiques à l'apparition de maladies dans des cultures comme le maïs ou la tomate, bien que ses performances soient prometteuses, certains points de vigilance, tels que les conditions de collecte des nouvelles images ou les attentes vis-à-vis des objectifs de l'application (détection de la présence d'une maladie versus identification précise de celle-ci), doivent être pris en compte avant une mise en œuvre opérationnelle.