

# VOLTRON

## **Section IA**

---

*Explication du CNN - Détection des maladies*

Version	Date	Auteur	Statut
1.0	23/07/2023	PIGNON Nathan	Finalisation rédaction
1.0	23/07/2023	CLAVIER Eliott	Relecture et validation

<b>Préambule</b>	<b>4</b>
Contexte du projet	4
Jeu de données	4
Modèle de base	5
<b>Expérimentations</b>	<b>7</b>
Affinage du modèle	7
Dataset	8
<b>Modèle final</b>	<b>10</b>
<b>Articles utiles</b>	<b>10</b>

# Préambule

## Contexte du projet

Le projet VOLTRON vise à améliorer la gestion des vignobles grâce à l'application de technologies innovantes, mêlant plusieurs spécialités telles que l'IOT, la Big Data, le Cloud, la sécurité ou encore l'Intelligence Artificielle (IA) qui est la partie que nous allons aborder.

Notre équipe de 4 experts IA est composée de:

- Eliott CLAVIER
- Paul RIPAULT
- Valentin BOUCHET
- Nathan PIGNON

L'un des objectifs principaux de cette équipe est de créer une intelligence artificielle permettant de détecter de manière prématurée les maladies de la vigne. Pour cela, des images des vignes vont être fournies en temps réel à l'intelligence artificielle qui aura comme objectif de détecter les maladies et d'en informer les vignerons.

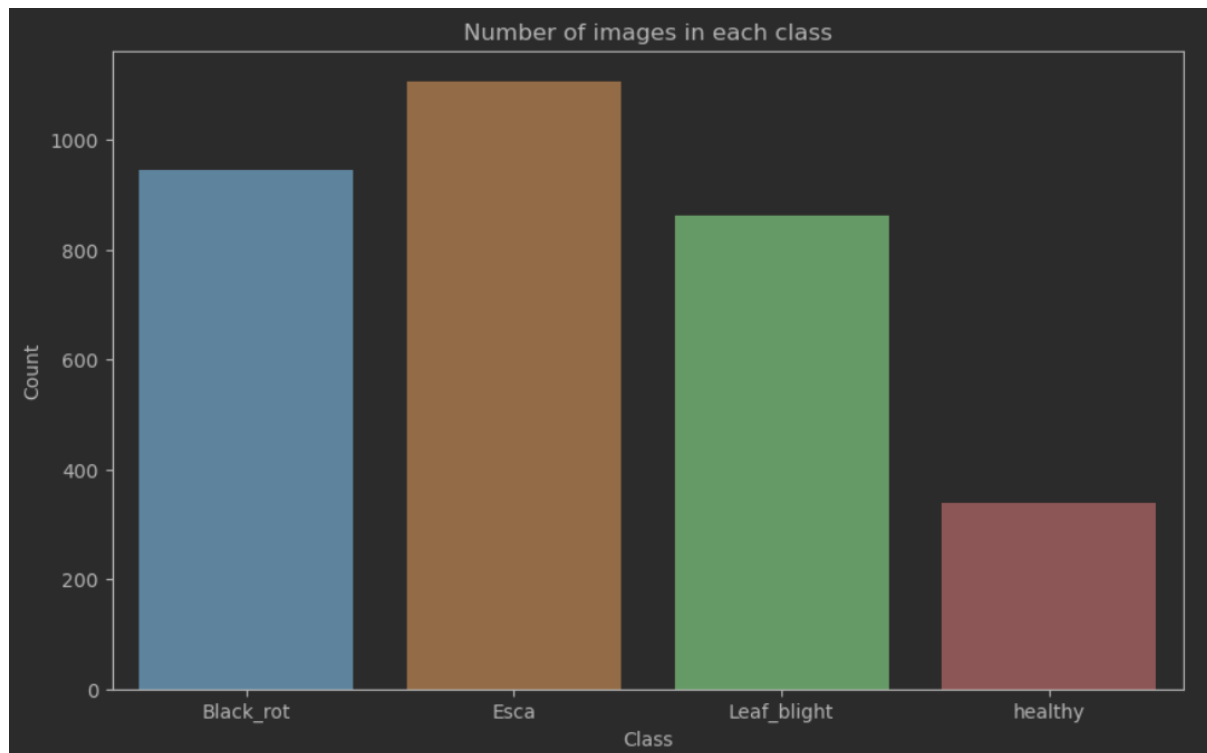
Un apprentissage continu sera implémenté grâce à une interface de feedback, afin que le modèle soit constamment entraîné et apprenne de ses erreurs grâce aux nouvelles données. C'est un élément essentiel de notre projet car il garantit une progression de notre modèle et une précision de plus en plus fiable au cours des années.

## Jeu de données

L'ensemble de données utilisé pour ce projet provient de la plateforme Kaggle, une ressource en ligne largement reconnue pour sa richesse en jeux de données divers et variés. Le lien spécifique pour accéder à l'ensemble de données [est le suivant](#).

Cet ensemble de données est particulièrement adapté à notre projet car il comprend quatre classes distinctes, chacune représentant une maladie différente affectant les vignes. Ces maladies sont les suivantes :

- "Black Rot"
- "Esca"
- "Leaf Blight"
- et le statut "Sain"



Chaque classe comprend un ensemble d'images de vignes affectées par la maladie correspondante, ou dans le cas de la classe "Healthy", des images de vignes saines. Ces images serviront de base pour l'entraînement de notre modèle d'intelligence artificielle, lui permettant d'apprendre à identifier et à distinguer ces maladies.

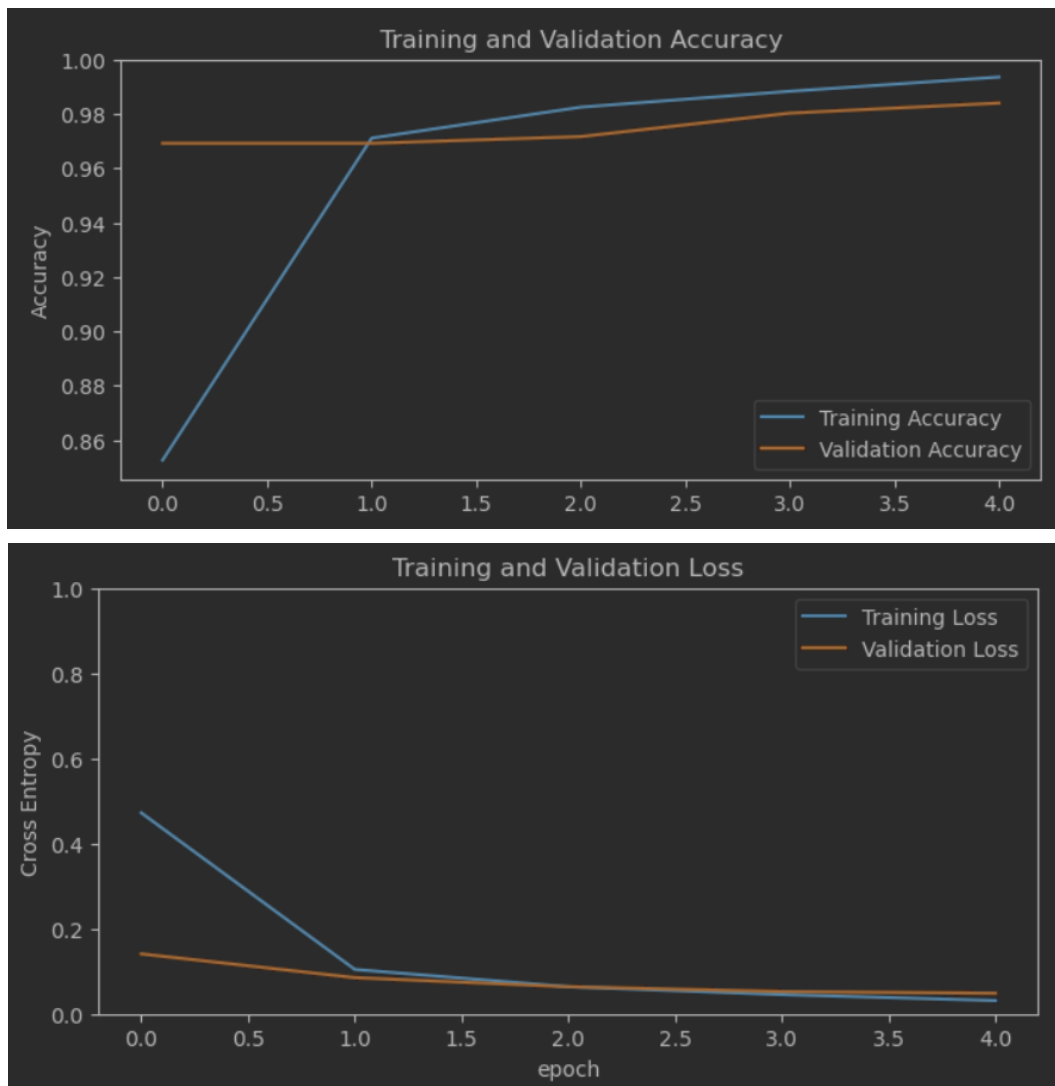
## Modèle de base

Le point de départ de notre travail sur la détection des maladies des vignes est un modèle de réseau de neurones convolutionnels (CNN) que nous avons trouvé sur Kaggle. Ce modèle, disponible [à l'adresse suivante](#), a démontré une précision de 99% dans la classification des maladies des vignes.

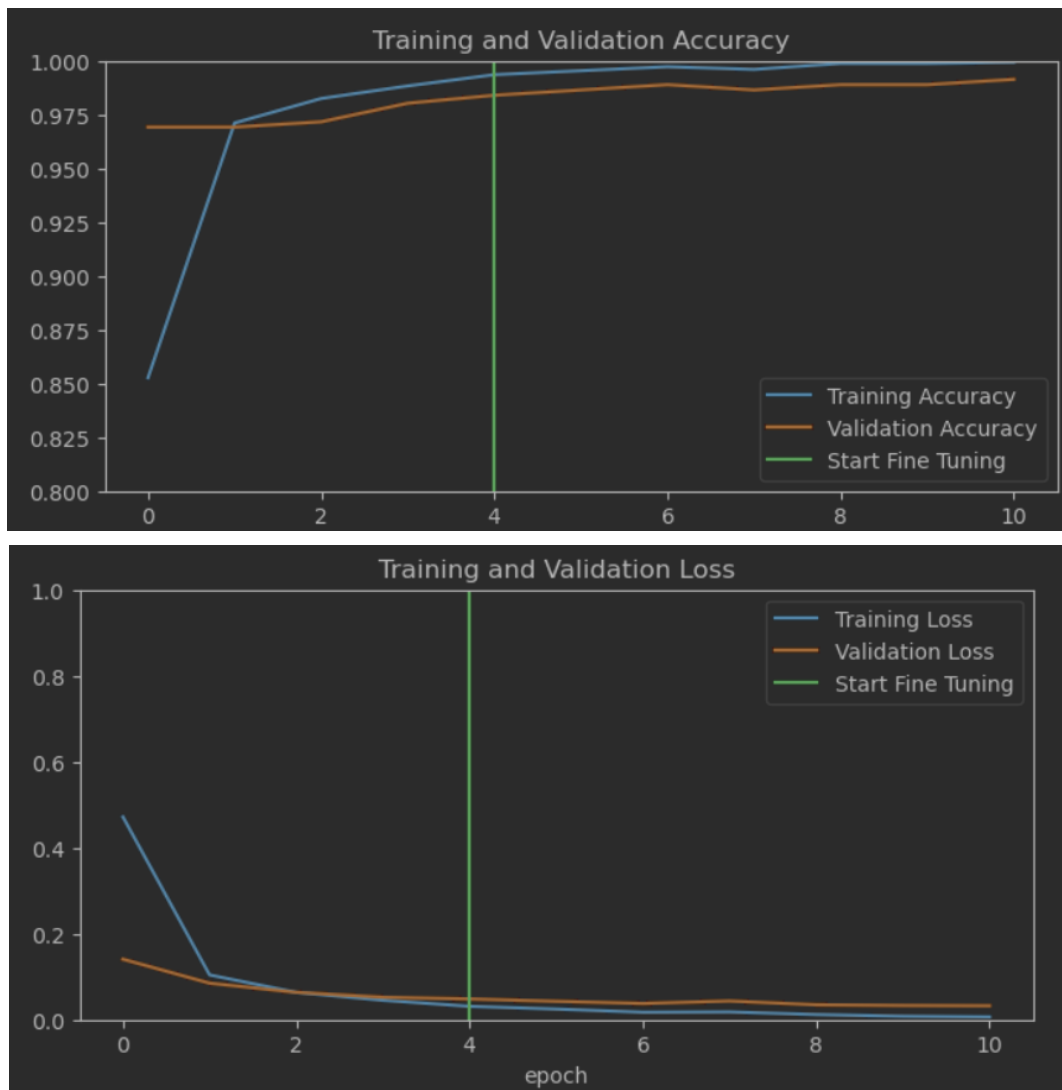
Notre modèle de base s'appuie sur un modèle pré-entraîné appelé MobileNetV2. MobileNetV2 est une architecture de réseau de neurones convolutionnels largement utilisée pour la vision par ordinateur, connue pour sa capacité à fournir une haute précision tout en étant relativement légère en termes de ressources de calcul.

Pour adapter le modèle MobileNetV2 à nos besoins spécifiques, plusieurs couches ont été ajoutées à l'architecture de base. Ces ajouts comprennent une couche de pooling global et trois couches denses. Ces couches supplémentaires permettent au modèle d'apprendre des caractéristiques plus complexes et spécifiques à notre ensemble de données.

Une fois que notre architecture de modèle a été définie, nous avons procédé à un premier entraînement du modèle. Cela a permis au modèle d'apprendre à identifier les différentes maladies des vignes à partir de notre ensemble de données.



Après l'entraînement initial, le modèle va passer par une phase d'affinage. L'affinage, ou fine-tuning, est une technique qui consiste à ré-entraîner une partie des couches d'un modèle pré-entraîné. Dans notre cas, les 100 premières couches du modèle MobileNetV2 vont être gelées et toutes les autres couches seront ré-entraînées. Cette approche a été choisie après avoir testé plusieurs combinaisons, et elle a permis d'améliorer les performances de notre modèle.



## Expérimentations

### Affinage du modèle

L'affinage est une technique couramment utilisée en apprentissage profond qui consiste à **geler une partie des couches d'un modèle pré-entraîné et à ré-entraîner les couches restantes**. Cette technique permet de conserver les caractéristiques apprises par le modèle pré-entraîné tout en permettant au modèle d'apprendre des caractéristiques plus spécifiques à notre ensemble de données.

L'objectif de cette expérimentation était de **déterminer le nombre optimal de couches à geler** pour **maximiser les performances** de notre modèle, sachant que notre modèle de base a les 100 premières couches gelées. En gelant différentes quantités de couches, nous avons cherché à **trouver un équilibre** entre la conservation des caractéristiques apprises par le modèle pré-entraîné et la capacité du modèle à apprendre de nouvelles caractéristiques à partir de notre ensemble de données.

Nous avons mené cette expérimentation en essayant de **ne geler aucune couche dans un premier temps**, puis **en gelant successivement les 50 et 150 premières couches** du modèle MobileNetV2 et en

ré-entraînant les couches restantes. Pour chaque configuration, nous avons évalué les performances du modèle sur notre ensemble de données.

**Modèle de base (100 premières couches) :**

- Loss : 0.0260
- Accuracy : 0.9902

**0 couches :**

- Loss : 0.0328
- Accuracy : 0.9877

**50 premières couches :**

- Loss : 0.0326
- Accuracy : 0.9889

**150 premières couches :**

- Loss : 0.0341
- Accuracy : 0.9877

Après avoir testé ces différentes configurations, nous avons conclu que **le gel des 100 premières couches offrait légèrement de meilleures performances**. Cette configuration permettait de conserver une grande partie des caractéristiques apprises par le modèle pré-entraîné tout en laissant suffisamment de couches pour que le modèle puisse apprendre des caractéristiques spécifiques à notre ensemble de données.

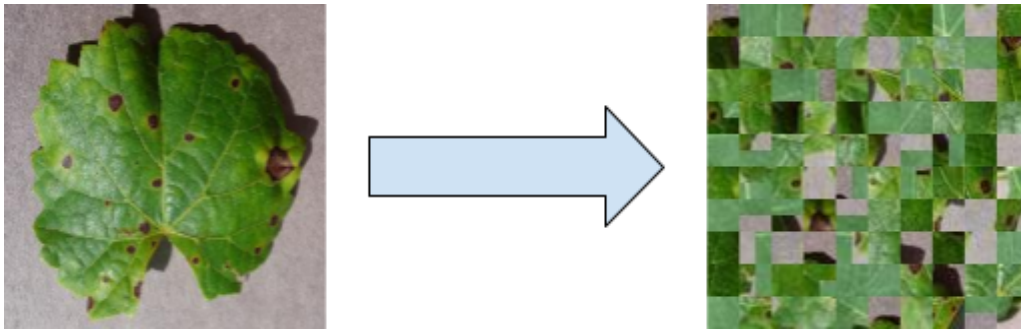
## Dataset

Le problème de notre modèle de base est qu'il est entraîné sur des feuilles seules, posées sur une table, ce qui n'est pas vraiment représentatif de la réalité. Nous avons exploré différentes techniques pour augmenter la diversité de notre ensemble de données afin de pallier ce problème. Nous avons donc revu nos priorités et avons décidé d'accepter une baisse de précision si les heatmaps correspondaient plus à ce que nous recherchons, c'est-à-dire fixer les zones qui différencient les maladies sur une feuille.

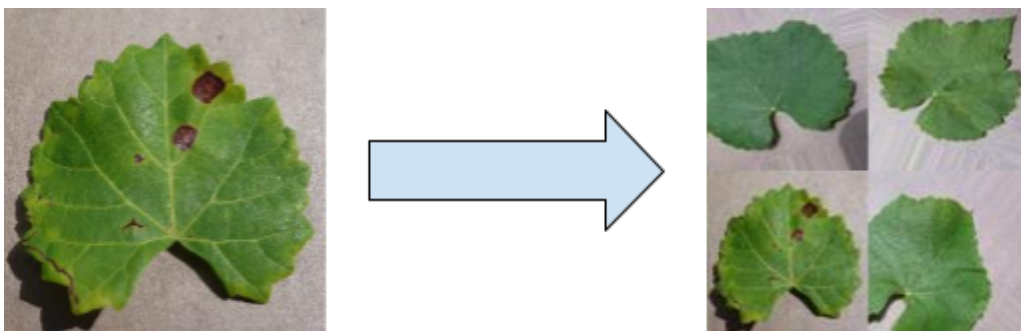
La première technique que nous avons essayée est le "shuffle". Cette technique consiste à découper une image en plusieurs morceaux (dans notre cas, 100 morceaux), à mélanger ces morceaux, puis à reformer l'image. Cette méthode permet d'introduire de la variabilité dans notre ensemble de données sans avoir besoin de collecter de nouvelles images.

Dans le but d'avoir des heatmaps qui fixent les zones contaminées par la maladie que nous essayons de détecter, nous avons essayé de remplacer quelques parties de la table en fond par des zones d'une feuille de la classe "Healthy".





La deuxième technique que nous avons explorée est la méthode "grid". Cette technique consiste à prendre une image de base et trois images de la classe "Healthy", puis à combiner ces quatre images en une seule. Cette méthode permet d'avoir plusieurs feuilles sur une seule image contenant des feuilles saines et une feuille malade (sauf lors du parcours des images de la classe "Healthy" qui contiendra alors 4 images saines). Notre modèle est donc entraîné à reconnaître une feuille malade parmi des feuilles saines.

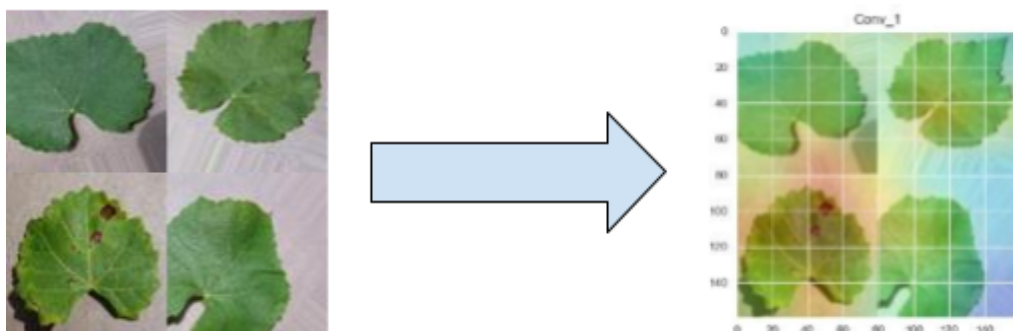


Nous avons également tenté de combiner les ensembles de données obtenus par les méthodes "shuffle" et "grid", ainsi que les images normales. Cette combinaison avait pour but de créer un ensemble de données plus diversifié et plus représentatif des différentes conditions que notre modèle pourrait rencontrer en production.

Parmis ces expérimentations, la technique du grid était la plus encourageante, malgré le fait que sa précision ait considérablement baissé (80% de précision) les heatmaps étaient plus précises que notre modèle de base, et surtout les données étaient plus représentatives de la réalité. Nous avons donc décidé de pousser un peu plus cette partie en essayant d'augmenter le nombre de données.

Pour cela, nous avons eu recours à une méthode de "flip" sur les images obtenues par la méthode "grid". Cette technique consiste à effectuer un flip horizontal et vertical de l'image, ce qui permet de multiplier par trois la taille de notre ensemble de données. De plus, pour une même image, cette méthode permet d'avoir la feuille malade à trois endroits différents, ce qui peut aider notre modèle à mieux apprendre à détecter les maladies indépendamment de leur position sur l'image.

Les résultats sont très satisfaisants car nous sommes passé à une précision de 87% et les heatmaps se concentrent désormais essentiellement sur notre feuille malade :



Malgré le fait que la précision ait baissé, nous avons décidé d'opter pour ce dataset car il est plus représentatif des futures données et est beaucoup plus fiable au niveau de la détection d'une feuille malade comme le montre les heatmaps.

## Modèle final

Après une série d'expérimentations et d'optimisations, nous avons abouti à notre modèle final. Ce modèle est le fruit de notre travail sur l'affinage du modèle, la manipulation de l'ensemble de données, et l'application de diverses techniques pour améliorer la précision et la généralisation du modèle.

Notre modèle final utilise la méthode "grid" pour la préparation des données, qui consiste à combiner une image de base et trois images de la classe "Healthy" en une seule image. Cette méthode a permis d'augmenter la diversité de notre ensemble de données et d'améliorer la capacité de notre modèle à reconnaître une feuille malade parmi des feuilles saines.

De plus, nous avons appliqué une méthode de flip sur les images obtenues par la méthode grid. Cette technique a permis de multiplier par trois la taille de notre ensemble de données et d'augmenter la variabilité des positions de la feuille malade sur l'image.

Malgré une légère baisse de la précision par rapport à notre modèle de base (de 99% à 87%), nous avons décidé d'adopter ce modèle final car il est plus représentatif des conditions réelles que notre modèle pourrait rencontrer en production. De plus, les heatmaps générées par notre modèle final se concentrent désormais essentiellement sur la feuille malade, ce qui indique une amélioration significative de la capacité de notre modèle à localiser les maladies.

## Articles utiles

### Explication MobileNetV2 :

[https://wandb.ai/wandb\\_fc/french/reports/L-volution-des-Architectures-Mobiles-CNN--Vmlldzo0NzM5NTc#:~:text=La%20V2%20des%20s%C3%A9ries%20MobileNet,am%C3%A9liorer%20les%20performances%20de%20MobileNets](https://wandb.ai/wandb_fc/french/reports/L-volution-des-Architectures-Mobiles-CNN--Vmlldzo0NzM5NTc#:~:text=La%20V2%20des%20s%C3%A9ries%20MobileNet,am%C3%A9liorer%20les%20performances%20de%20MobileNets)