



PLANTSPY

Your botanist on hand

*Projet réalisé par les “**PlantSpy Girls**” :*

- Marie-Laure Fichoux
- Ilham EL Bouloumi

Data Scientist Bootcamp - Avril 2022

PlantSpy

Détecter le type de plante et sa maladie grâce au Deep Learning

I- Contexte:	3
II- Exploration des données :	4
III- Premiers modèles Deep Learning : classification de l'espèce d'une plante	11
1- Baseline : Modèle CNN classique	11
Le premier modèle testé en Baseline, est un simple modèle CNN classique avec l'architecture suivante:	11
3- Modèle CNN combiné au générateur d'images	12
4- Modèle LeNet combiné au générateur d'images	14
5- Modèle LeNet avec Rééquilibrage des données et générateur d'images	17
5- Transfer Learning	19
5- Conclusions- Synthèse tirées des modèles précédents	21
IV- Détection de la maladie et classification du type de maladie	22
1- Détection de la maladie	22
2- Classification du type de maladie de la plante	23
2.1- L'espèce "Tomato" comme exemple	23
2.2- Résultats sur les autres espèces	27
V- Approche : un modèle pour triple classification	28
VI- Combinaison de modèles	31
1- Détermination de l'espèce et détection de la maladie	32
2- Détermination de l'espèce et détection du type de maladie	34
3- Classification finale avec agrégation des modèles:	35
VI- Points d'amélioration:	36
VII- Annexes	38
Annexe 1: Classification type de Maladie	38
A- Espèce "Grape"	38
B- Espèce "Corn"	38
C- Espèce "Potato"	38
C- Espèce "Apple"	38
Annexe 2 : Modèle combine final- Performance de la classification Espèces+Type de Maladie	39
VIII- Programmes/Code	42

I- Contexte:

D'après Wikipedia, "il existerait (en 2015) plus de 400 000 espèces décrites, dont la grande majorité sont des plantes à fleurs (369 000 espèces répertoriées), sachant que près de 2000 nouvelles espèces sont découvertes chaque année."

Devant ce nombre évolutif d'espèces, l'intelligence artificielle pourrait apporter son aide afin de faciliter la reconnaissance et la classification des différentes plantes, la finalité étant d'assurer une meilleure compréhension de la biodiversité de l'écosystème et de son évolution.

Dans ce contexte, le projet "PlantSpy" apporte une pierre à l'édifice en jouant le rôle du botaniste. En effet, l'objectif du projet est de localiser et classer l'espèce d'une plante dans une image. Une fois la classification est faite, l'application informe l'utilisateur si la feuille de la plante est saine ou malade. Et si maladie il y a, "PlantSpy" devrait détecter le type de maladie.

Dans les pages qui suivent, nous vous présenterons la démarche effectuée afin de trouver le modèle Deep Learning qui permet d'apporter les meilleures réponses possibles aux questions suivantes :

1. Quelle est la catégorie de la plante?
2. Est-ce que la feuille est malade ou saine?
3. Quel est le type de maladie détecté ?

A partir des dataset fournis par la plateforme Kaggle, nous avons développé une première version de "PlantSpy" permettant de classer 14 catégories de plantes différentes.

Dans le rapport suivant, nous vous présenterons :

- Le dataset choisi pour le projet et ses spécificités
- Les différentes approches de Deep Learning testées et leurs performances respectives
- Le choix du meilleur modèle possible et ses performances

Enfin, nous finirons avec des propositions d'améliorations à apporter au projet.

II- Exploration des données :

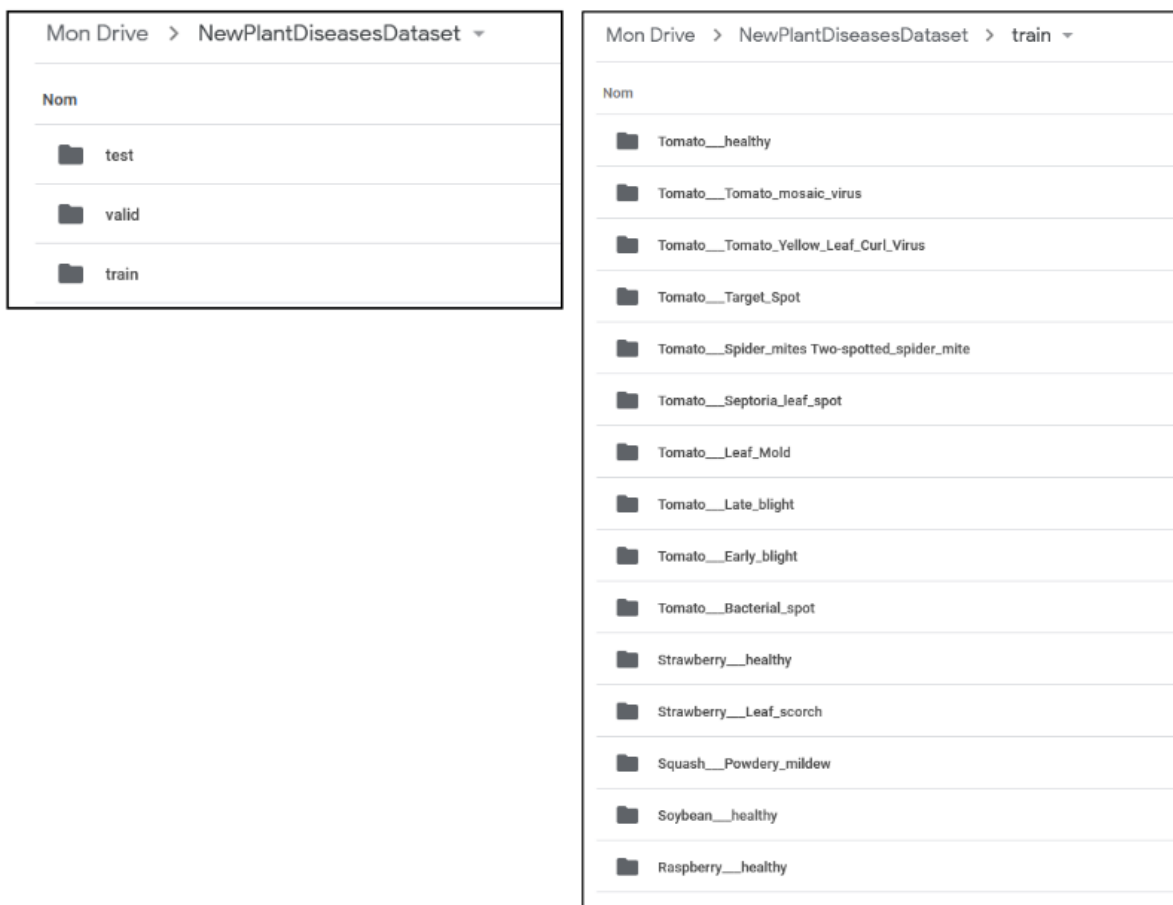
- Description du dataset New Plant Disease

Nous avons essentiellement travaillé sur une base de données Kaggle (new-plant-disease-dataset) contenant près de 90K images de plantes labellisées en fonction de l'espèce de la plante et du type maladie.

Le dataset est constitué 3 répertoires :

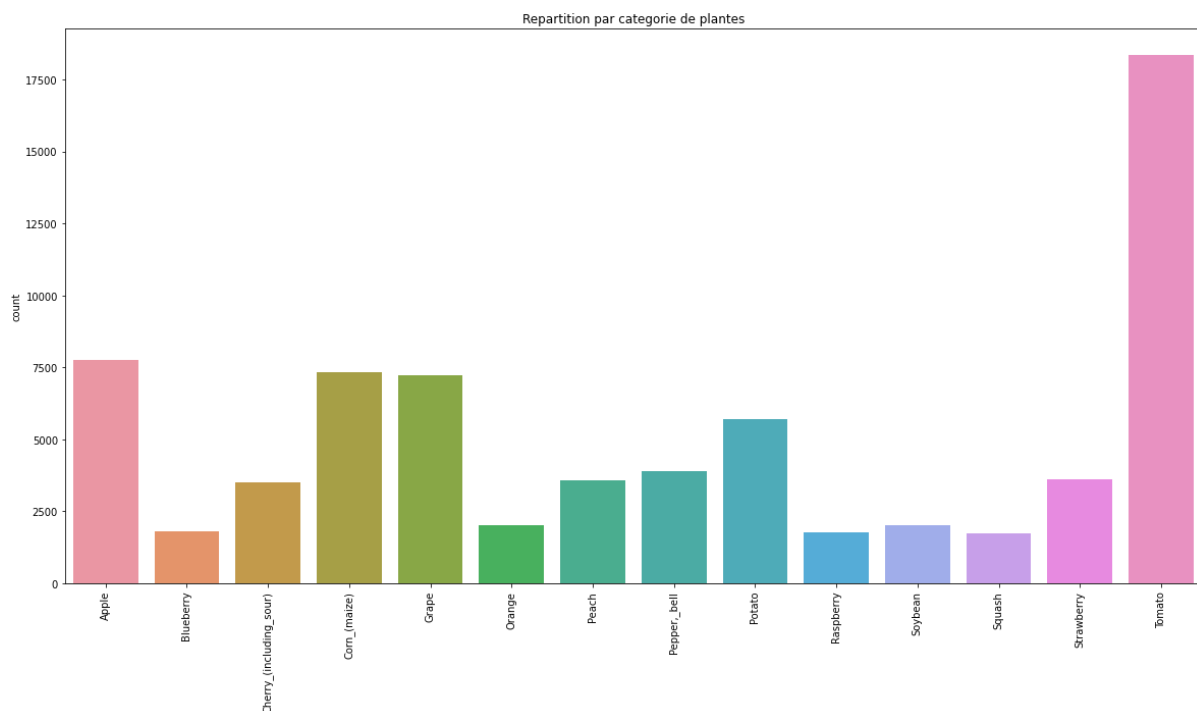
- train : 38 sous-Dir, 70 295 images
- valid : 38 sous-Dir, 17 552 images
- test: 33 images

Le nom de chaque sous-répertoire définit l'espèce de la plante et sa maladie:



La base de données contient au total **14 espèces de plantes (listée en anglais)** :

1- Apple	6- Orange	11- Soybean
2- Blueberry	7- Peach	12- Squash
3- Cherry	8- Pepper	13- Strawberry
4- Corn	9- Potato	14- Tomato
5- Grape	10- Raspberry	

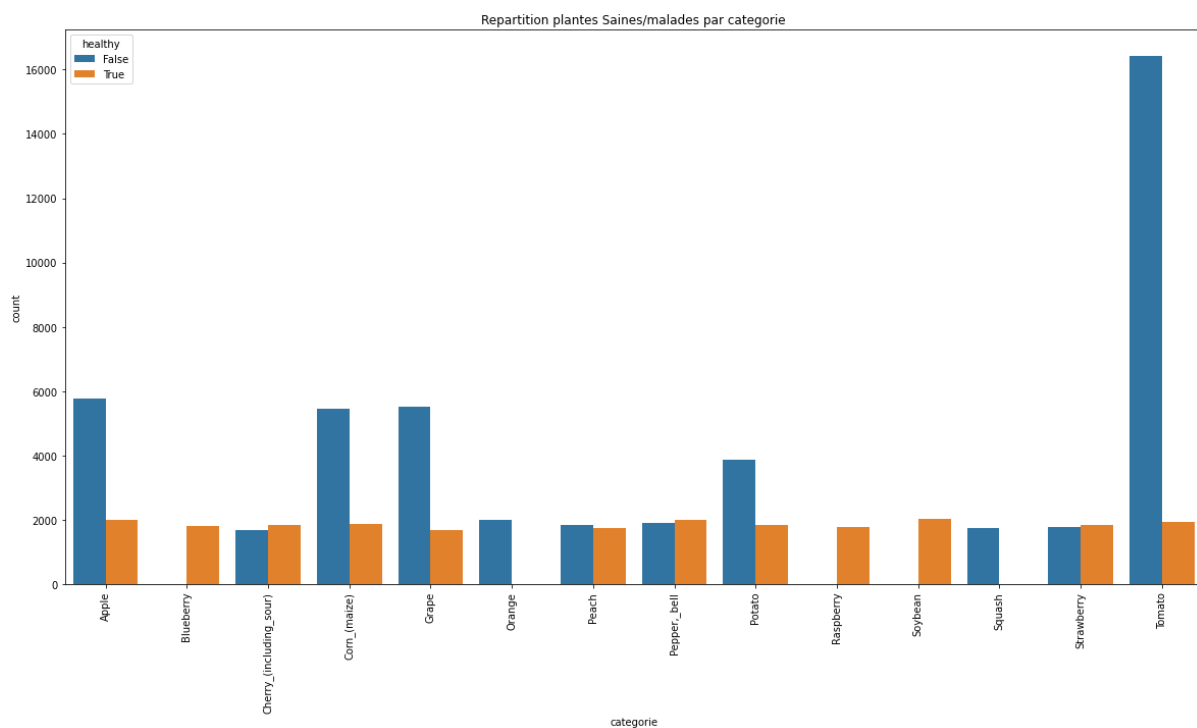
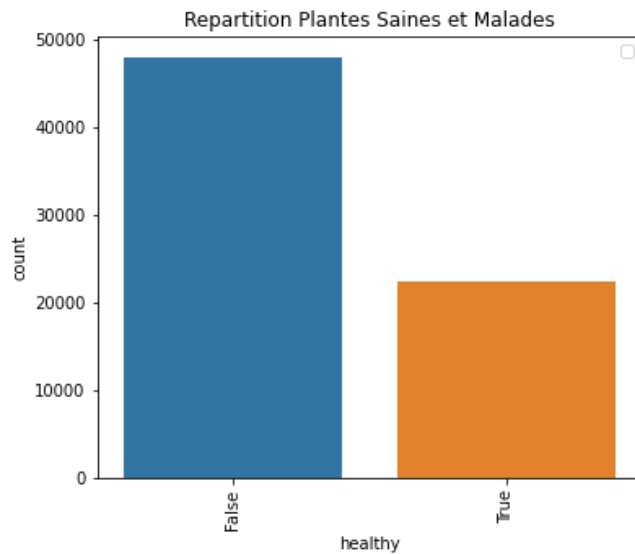


La répartition entre les différentes espèces de plantes n'est pas uniforme. La catégorie "Tomato" contient plus de 3 fois plus d'images que les autres espèces.

Dans l'échantillon d'apprentissage, nous avons plus de 18000 images "Tomato" contre 7771 images "Apple" et seulement 1736 "Squash". Il y a donc un fort déséquilibre dans notre base de données qu'il conviendra de corriger par des techniques de ré-échantillonnage (Over ou underSampling) ou de pondération.

Plantes saines et malades:

Notre dataset contient des photos de ces 14 espèces de plantes saines et malades. Près de 68% des images correspondent à des photos de plantes malades.

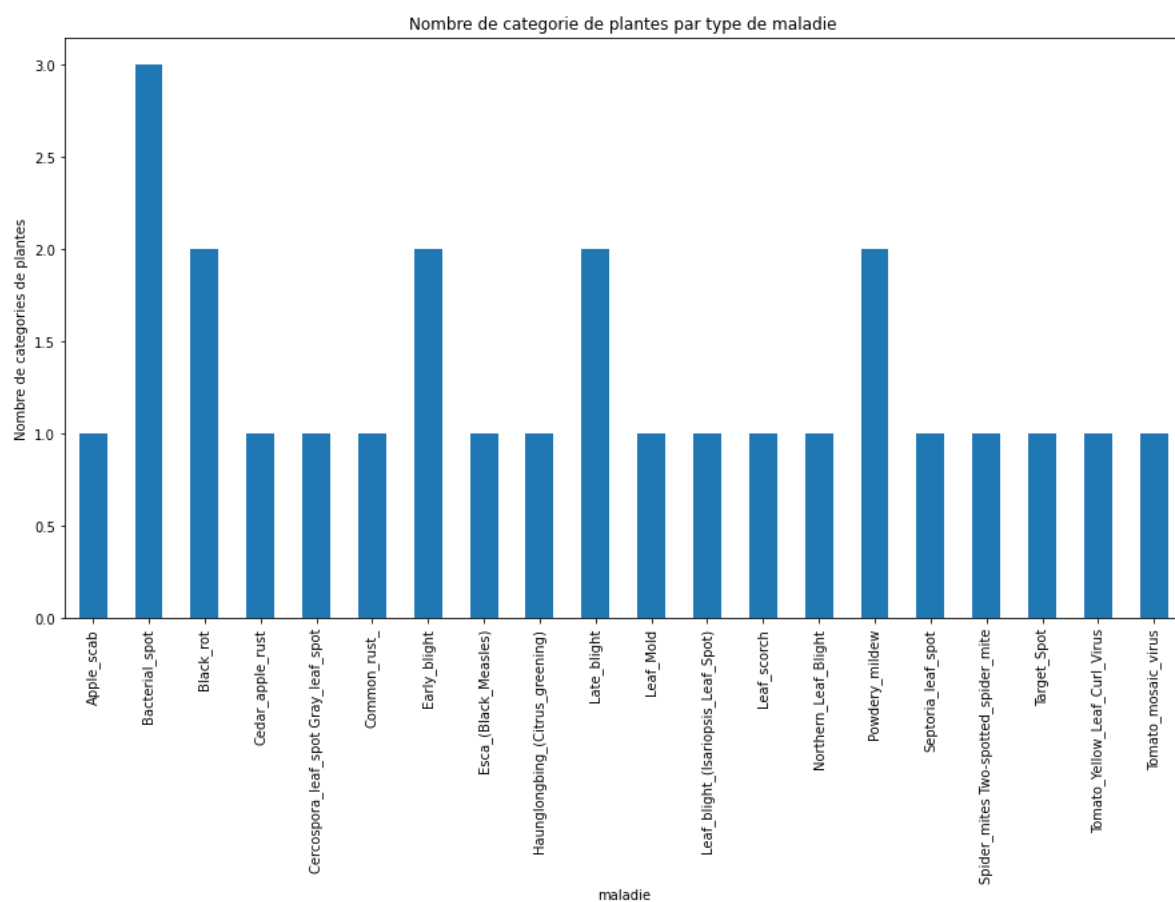
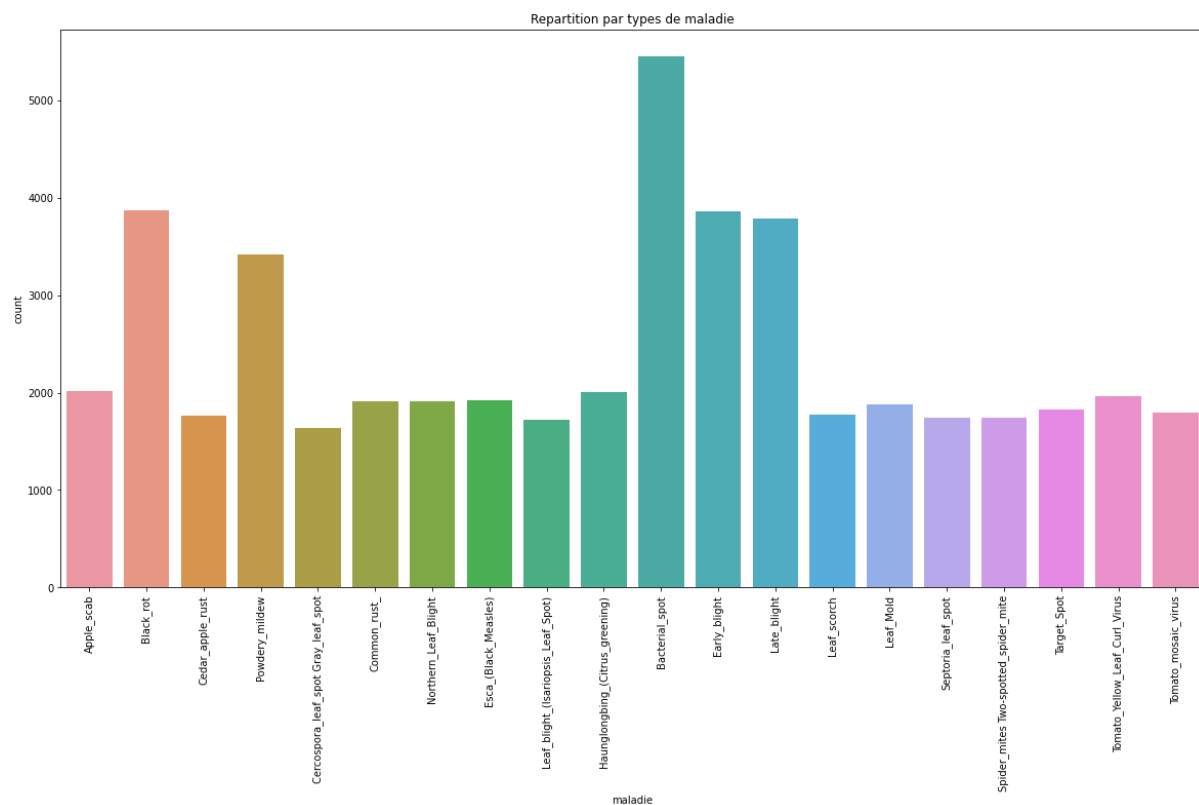


Encore une fois, nous observons un déséquilibre dans la répartition entre les espèces saines et malades:

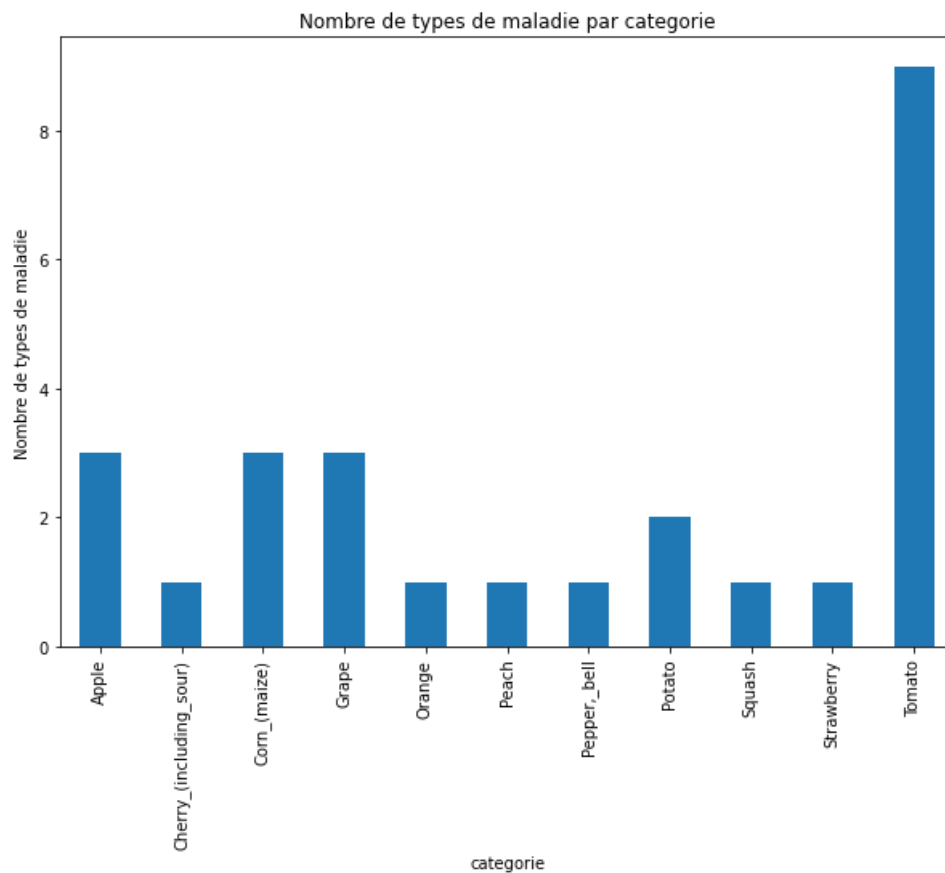
- L'espèce "Tomato" contient 10% d'image de plantes saines,
- Les espèces "Blueberry", "Raspberry" et "squash" ne contiennent aucune photo de plante malade
- A l'inverse, les espèces "Squash" et "Orange" ne présentent que des plantes malades.

Types de maladie:

Notre dataset contient des photos de 20 types de maladies différentes observées sur les 14 espèces de plantes. Seulement 5 types de maladie peuvent être communs à plusieurs espèces de plantes ("Bacterial Spot", "black_rot", "powdery_mildew" et "early" and "late blight")



La répartition du nombre de type de maladie par espèces:



La répartition du type de maladie par espèce n'est pas uniforme. L'espèce "Tomato" contient 9 types de maladies différentes, quant aux autres espèces, elles contiennent entre 1 et 3 types de maladies.

Tableau de synthèse:

Categorie de plante	Echantillon d'apprentissage	Echantillon de validation	% plantes saines	# types de Maladie
Apple	7771	1943	26%	3
Blueberry	1816	454	100%	0
Cherry	3509	877	52%	1
Corn	7316	1829	25%	3
Grape	7222	1805	23%	3
Orange	2010	503	0%	1
Peach	3566	891	48%	1
Pepper	3901	975	51%	1
Potato	5702	1426	32%	2
Raspberry	1781	445	100%	0
Soybean	2022	505	100%	0
Squash	1736	434	0%	1
Strawberry	3598	900	51%	1
Tomato	18345	4585	10%	9
Total	70295	17572	32%	26

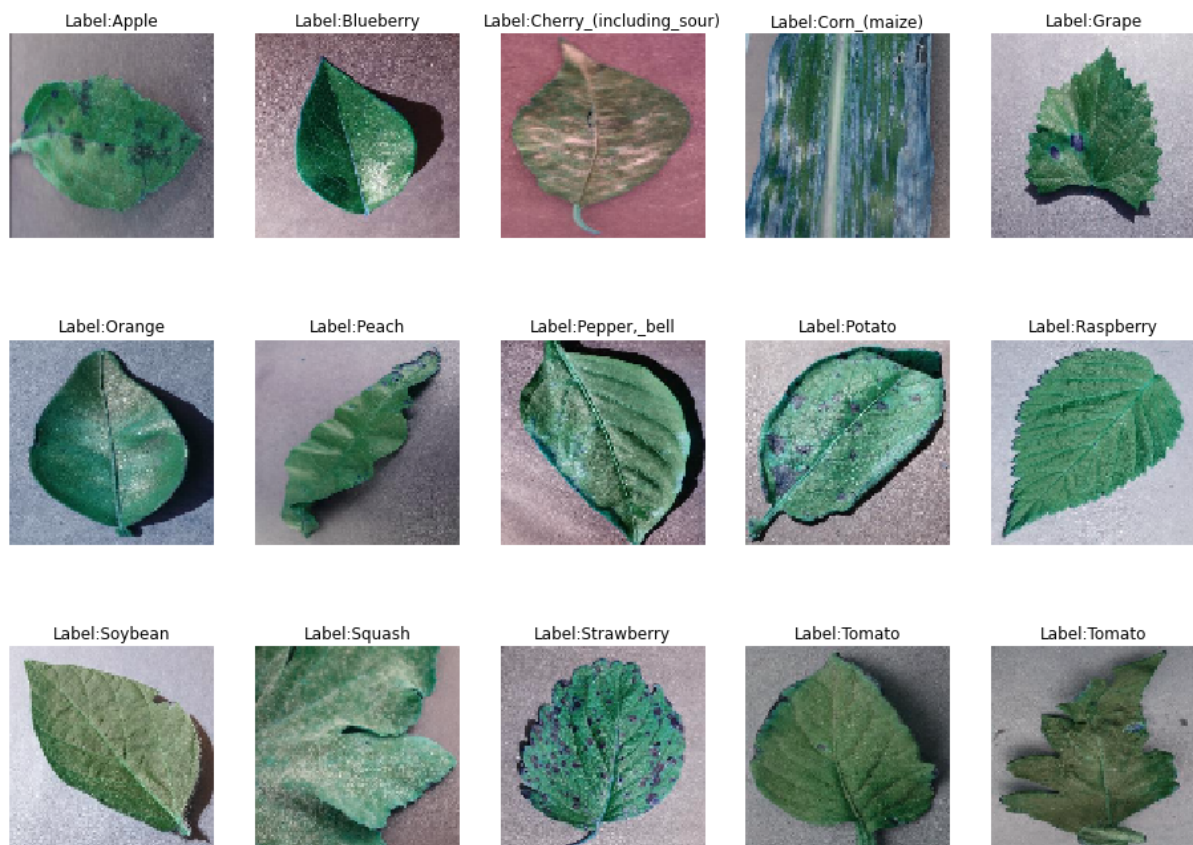
Analyse des images:

Les images de la base de données ont une haute définition et ont toutes le même format de (256x256x3). Pour réduire la mémoire (RAM) nécessaire et le temps de traitement sans pour autant perdre d'information, nous avons choisi de réduire la taille des images à (100,100,3).

Certaines espèces et surtout les types de maladies ont de grandes similarités, il nous a donc paru important de garder tout de même une bonne définition et la couleur pour mieux les distinguer.

Pour un format, (255x255x3) la modélisation porterait sur 195.075 params alors que pour des images de format (100,100,3) ce nombre passe à 30.000, ce qui paraît bien plus raisonnable sans pour autant perdre trop en résolution par rapport au format d'origine.

Nous présentons ci-dessous une sélection aléatoire d'images correspondant à chacune des espèces référencées dans notre dataset:



Nous pouvons constater que les photos de notre dataset présentent une structure assez similaire et standard: Il s'agit de la photo de la feuille d'une plante posée sur un support de couleur homogène (gris, rose, beige, noir). Il n'y a donc dans nos données aucune information ou biais liés à l'environnement/ background. Il n'aura donc aucun impact sur la modélisation.

Après une première observation de ces photos nous pouvons déjà repérer quelques espèces qui seront plus facilement identifiables et celles qui présentent de probables erreurs de prédictions. Par exemple:

- Les espèces "Corn" et "squash" devraient être facilement identifiables car la forme de leur feuillage est assez différente des autres espèces.
- En revanche sur cet exemple, nous voyons qu'il peut être délicat de distinguer "Blueberry" et "Orange".

Bien que les photos d'origine présentent le même format d'origine/résolution, nous observons de grande disparités de qualité entre les photos (voir échantillon ci-dessous)

- Photos surexposées ou sous-exposées
- photo avec ombre de la feuille
- Photos floues, etc.

Ces caractéristiques rendent la tâche de classification plus difficile mais elles sont essentielles dans l'apprentissage pour une meilleure performance du modèle.

Un échantillon de photos :



III- Premiers modèles Deep Learning : classification de l'espèce d'une plante

Comme nous avons pu le voir dans la section précédente, les images de notre modèle se définissent par plusieurs niveaux de classification/labels:

- l'espèce de plantes (tomate, pomme, etc.)
- la dimension Malade ou Saine
- le type de maladie qui sera différent en fonction du type de plante.

Dans ce contexte, nous avons décidé de procéder en plusieurs étapes pour la modélisation. Dans cette section, nous nous intéresserons uniquement à la classification de l'espèce

1- Baseline : Modèle CNN classique

Nous nous sommes intéressés aux Réseaux de neurones convolutifs car ce type de modèle est particulièrement adapté au traitement des images du fait de son format matriciel.

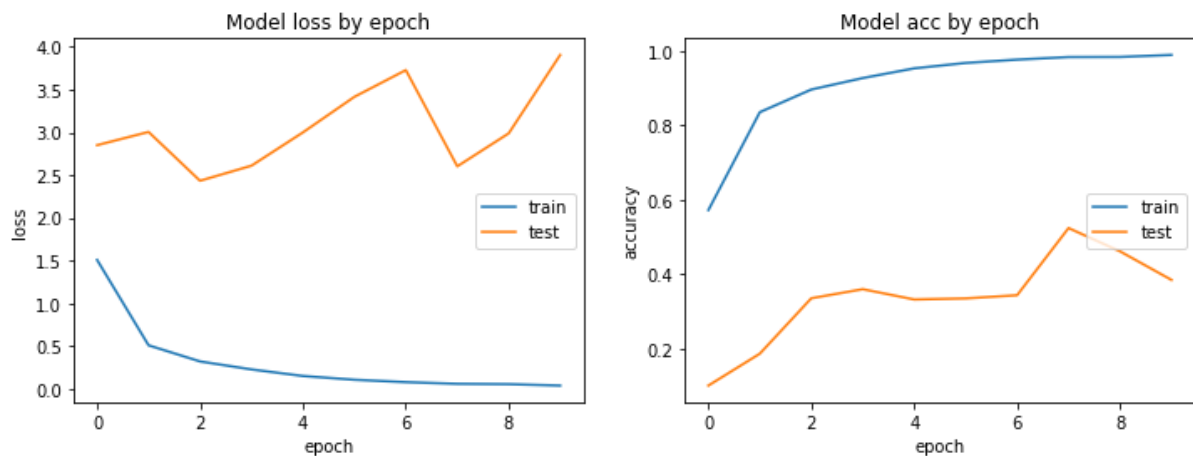
Le premier modèle testé en Baseline, est un simple modèle CNN classique avec l'architecture suivante:

- une première couche de convolution composée de 32 matrices de convolution
- une couche de MaxPooling
- une couche de Dropout avec un taux de 20%
- une couche d'aplatissement en vecteur
- une couche dense à 128 neurones
- une dernière couche dense "output" à 14 neurones, correspondant aux nombres de catégorie d'espèce de notre classification avec une fonction d'activation Softmax

Layer (type)	Output Shape	Param #
Input (InputLayer)	[(None, 100, 100, 3)]	0
conv2d_4 (Conv2D)	(None, 96, 96, 32)	2432
max_pooling2d_4 (MaxPooling 2D)	(None, 48, 48, 32)	0
dropout_3 (Dropout)	(None, 48, 48, 32)	0
flatten_2 (Flatten)	(None, 73728)	0
dense_4 (Dense)	(None, 128)	9437312
dense_5 (Dense)	(None, 14)	1806
=====		
Total params: 9,441,550		
Trainable params: 9,441,550		
Non-trainable params: 0		

Nous avons utilisé la fonction de perte "Sparse-sparse categorical crossentropy" adaptée au problème de classification à plusieurs classes et nous utilisons la métrique "Accuracy" pour évaluer les performances.

L'entraînement du modèle sur 10 Epochs et des batch de taille 100 donne les résultats suivants:



Nous voyons bien que l'apprentissage se stabilise assez rapidement (présence de plateau après 2 Epochs). En revanche, les performances sur l'échantillon de validation sont médiocres et instables. Ces résultats révèlent un sur apprentissage du modèle.

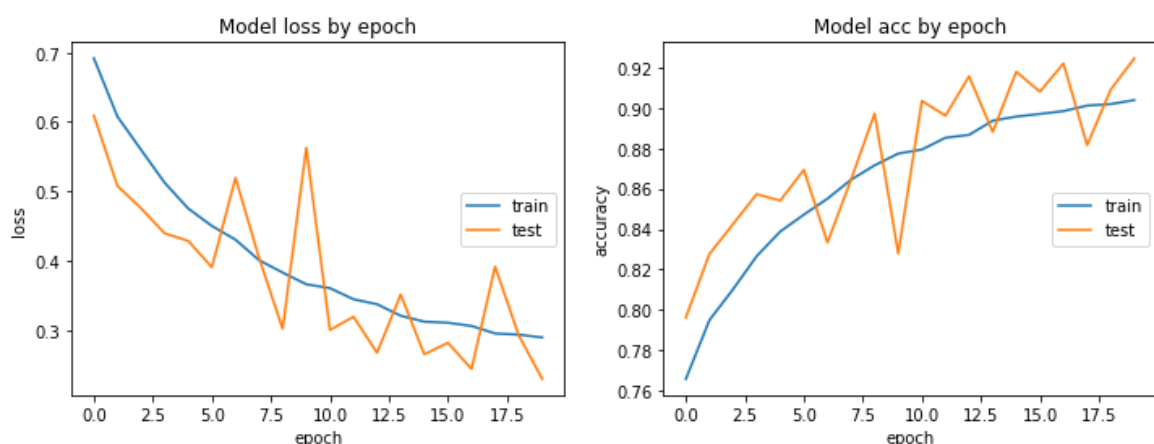
3- Modèle CNN combiné au générateur d'images

Afin de pallier au problème de surapprentissage, nous avons utilisé le générateur d'images. Les générateurs d'images permettent de générer des lots d'images transformées (rotation, zoom, décalage, ...) pour entraîner notre modèle. Cette approche permet d'augmenter notre jeu de données et d'éviter au maximum le sur-apprentissage.

Nous avons donc instancié un générateur d'images avec les paramètres suivants:

```
train_data_generator = ImageDataGenerator(
    rotation_range=50,
    width_shift_range=1,
    height_shift_range=1,
    zoom_range=[0.7, 1.2],
    horizontal_flip=True)
```

Nous entraînons ensuite le même modèle que précédemment mais cette fois-ci en y intégrant le générateur d'images et nous obtenons les résultats suivants:



Cette fois-ci les performances convergent entre l'échantillon d'entraînement et de test. Nous évitons le surapprentissage et les performances (niveau d'accuracy) atteignent des niveaux très satisfaisants dès 10 Epochs. Notons toutefois que les performances sur l'échantillon de validation présentent une certaine instabilité.

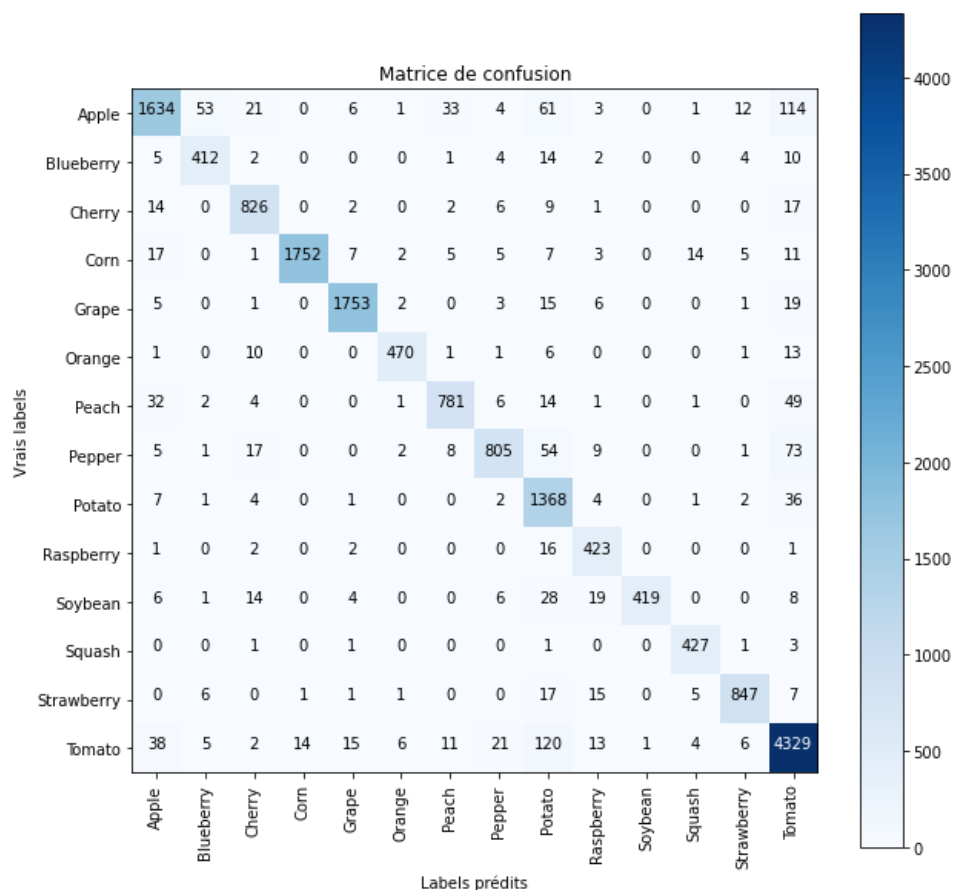
Classification Report:

	precision	recall	f1-score	support
Apple	0.93	0.84	0.88	1943
Blueberry	0.86	0.91	0.88	454
Cherry	0.91	0.94	0.93	877
Corn	0.99	0.96	0.97	1829
Grape	0.98	0.97	0.97	1805
Orange	0.97	0.93	0.95	503
Peach	0.93	0.88	0.90	891
Pepper	0.93	0.83	0.88	975
Potato	0.79	0.96	0.87	1426
Raspberry	0.85	0.95	0.90	445
Soybean	1.00	0.83	0.91	505
Squash	0.94	0.98	0.96	434
Strawberry	0.96	0.94	0.95	900
Tomato	0.92	0.94	0.93	4585
accuracy			0.92	17572
macro avg	0.93	0.92	0.92	17572
weighted avg	0.93	0.92	0.92	17572

L'accuracy globale du modèle est très bonne (0.92 sur l'échantillon de validation). Les niveaux de f1-score par espèces de plantes varient mais restent correctes. La catégorie "Apple" présente la performance la plus faible avec une accuracy de 0.88 et "Corn" la performance la plus élevée avec 0.97.

La matrice de confusion:

une analyse de la matrice de confusion permet de mieux comprendre les erreurs de prédictions:



Nous voyons que les catégories qui présentent le plus d'erreurs de prédiction sont "Tomato" et "Apple".

La catégorie "Tomato" étant sur-représentée dans notre dataset il est assez logique d'observer que beaucoup d'images ont été classées par erreur dans cette catégorie. En effet, à chaque fois qu'une photo sera difficile à classer, le modèle aura tendance à lui affecter par défaut la catégorie "Tomato".

4- Modèle LeNet combiné au générateur d'images

Nous avons vu que le générateur d'images améliore considérablement notre modèle en évitant le surapprentissage. Dans cette partie, nous gardons le générateur mais nous considérons cette fois-ci une d'architecture de réseaux de neurones convolutifs plus complexe: l'architecture LeNet.

L'architecture du LeNet est constituée des couches suivantes:

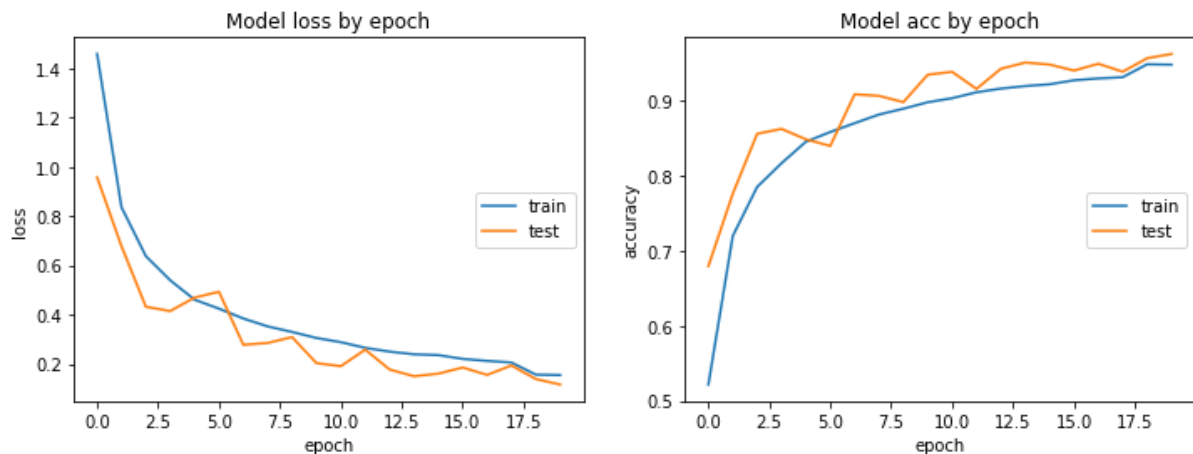
- Une première couche de **Convolution**
- Une couche de **Max-Pooling 1**
- Une nouvelle couche de **Convolution**
- Une couche de **Max-Pooling 2**
- Une couche **Dropout** (Connexions coupées: 20%)
- Une couche d'**Aplatissement**
- **Une couche Dense 1** a 128 neurones
- Une dernière couche Dense Output a 14 neurones correspondant au nombre de catégorie de plantes - fonction d'activation **Softmax**

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 96, 96, 30)	2280
max_pooling2d_3 (MaxPooling 2D)	(None, 32, 32, 30)	0
dropout_2 (Dropout)	(None, 32, 32, 30)	0
conv2d_4 (Conv2D)	(None, 30, 30, 64)	17344
max_pooling2d_4 (MaxPooling 2D)	(None, 15, 15, 64)	0
conv2d_5 (Conv2D)	(None, 13, 13, 32)	18464
max_pooling2d_5 (MaxPooling 2D)	(None, 6, 6, 32)	0
dropout_3 (Dropout)	(None, 6, 6, 32)	0
flatten_1 (Flatten)	(None, 1152)	0
dense_2 (Dense)	(None, 64)	73792
dense_3 (Dense)	(None, 14)	910

=====
Total params: 112,790
Trainable params: 112,790
Non-trainable params: 0
=====

L'entraînement du modèle LeNet avec générateur d'images donne les résultats suivants:



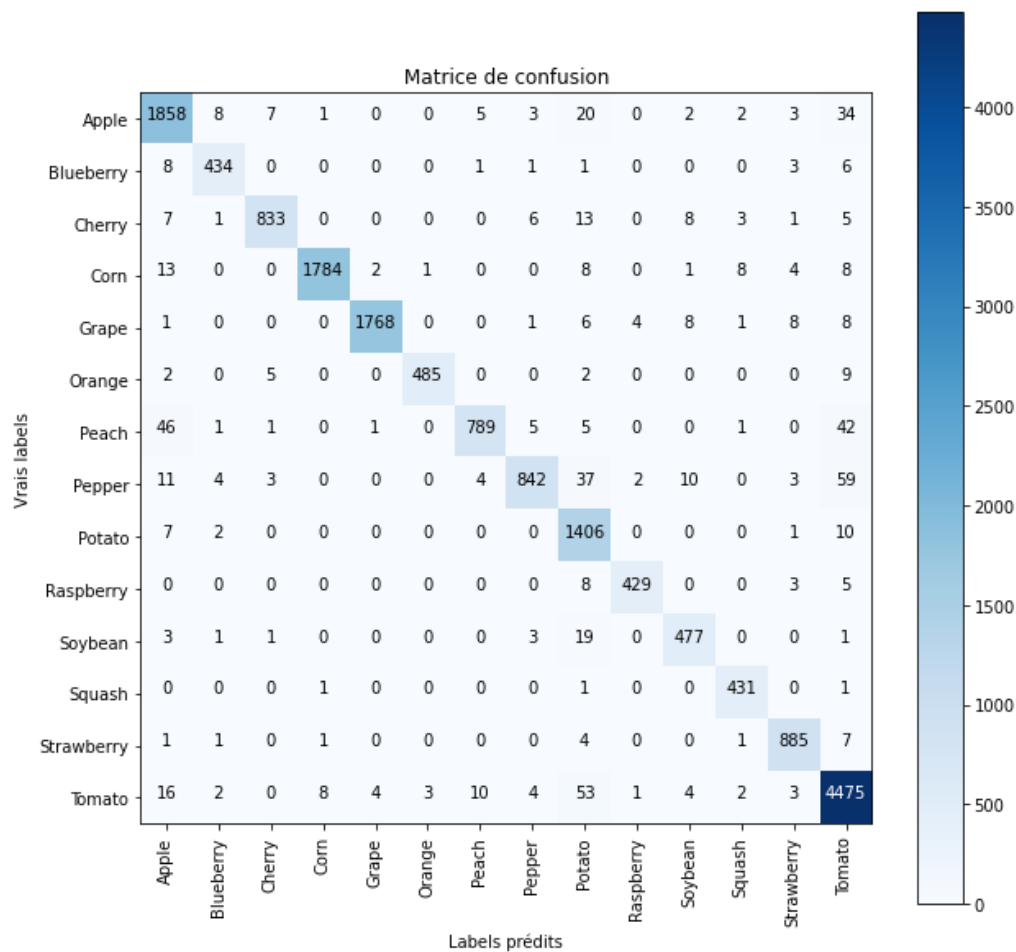
Encore une fois les performances convergent entre l'échantillon d'entraînement et de test. Nous évitons le surapprentissage et les performances atteintes (niveaux d'accuracy) sont excellentes et beaucoup plus stables.

Classification Report:

	precision	recall	f1-score	support
Apple	0.94	0.96	0.95	1943
Blueberry	0.96	0.96	0.96	454
Cherry	0.98	0.95	0.96	877
Corn	0.99	0.98	0.98	1829
Grape	1.00	0.98	0.99	1805
Orange	0.99	0.96	0.98	503
Peach	0.98	0.89	0.93	891
Pepper	0.97	0.86	0.92	975
Potato	0.89	0.99	0.93	1426
Raspberry	0.98	0.96	0.97	445
Soybean	0.94	0.94	0.94	505
Squash	0.96	0.99	0.98	434
Strawberry	0.97	0.98	0.98	900
Tomato	0.96	0.98	0.97	4585
accuracy			0.96	17572
macro avg	0.96	0.96	0.96	17572
weighted avg	0.96	0.96	0.96	17572

Les performances avec l'architecture LeNet sont excellentes avec une accuracy globale de 0.96. Les niveaux de f1-score par espèces sont également très bons variant entre 0.92 et 0.98.

La matrice de confusion:



La matrice de confusion confirme les bons résultats du modèle LeNet par rapport au modèle CNN en révélant moins d'erreurs de prédiction même sur les catégories "Tomates" et "Apple"

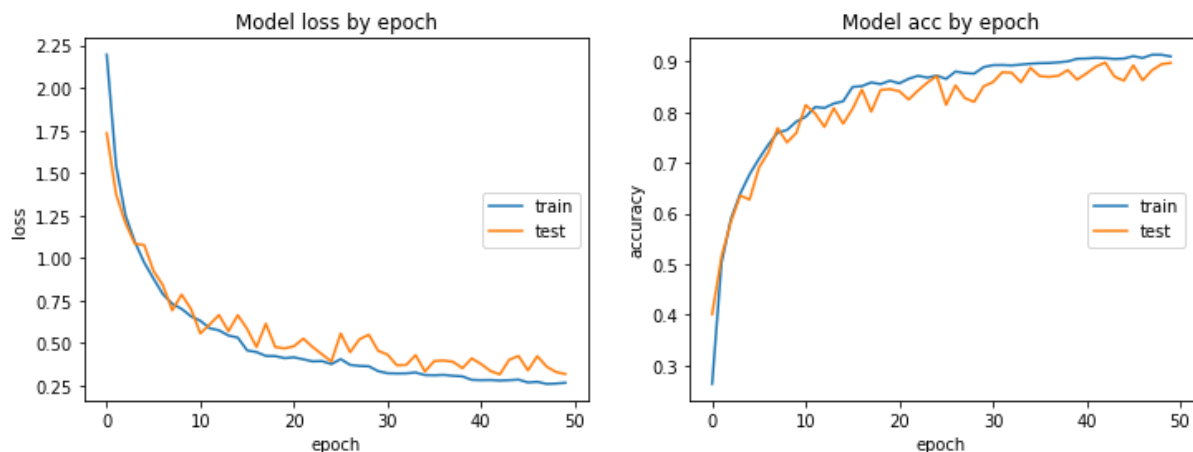
5- Modèle LeNet avec Rééquilibrage des données et générateur d'images

Comme évoqué dans la partie exploration des données, notre dataset présente un fort déséquilibre entre les différentes espèces. En effet, la catégorie "Tomates" à elle seule contient plus de 23.000 images différentes, 4 fois plus que la moyenne des 13 autres catégories. Il nous a paru pertinent de tester l'impact d'un rééquilibrage de nos données sur l'échantillon d'entraînement sur la performance du modèle.

Au-delà du rééquilibrage de nos données, il nous a également semblé important de réduire la taille de notre échantillon d'apprentissage. En effet, cela engagera moins de ressources de calculs : RAM, disque dur. Durant les tests des premiers modèles, nous avons été mis en épreuve, à plusieurs reprises, par des crash lors de l'exécution du programme faute de disponibilité de mémoire RAM, ressource vitale pour le Deep Learning en particulier.

Afin de réduire la taille du dataset tout en le rééquilibrant, nous avons appliqué le modèle "**RandomUnderSampler**". Cela nous a permis d'obtenir un nouveau dataset de **18K** images réparties en **1286** images par catégories.

Les résultats du Modèle LeNet combiné au générateur de d'images sur ce nouveau dataset:

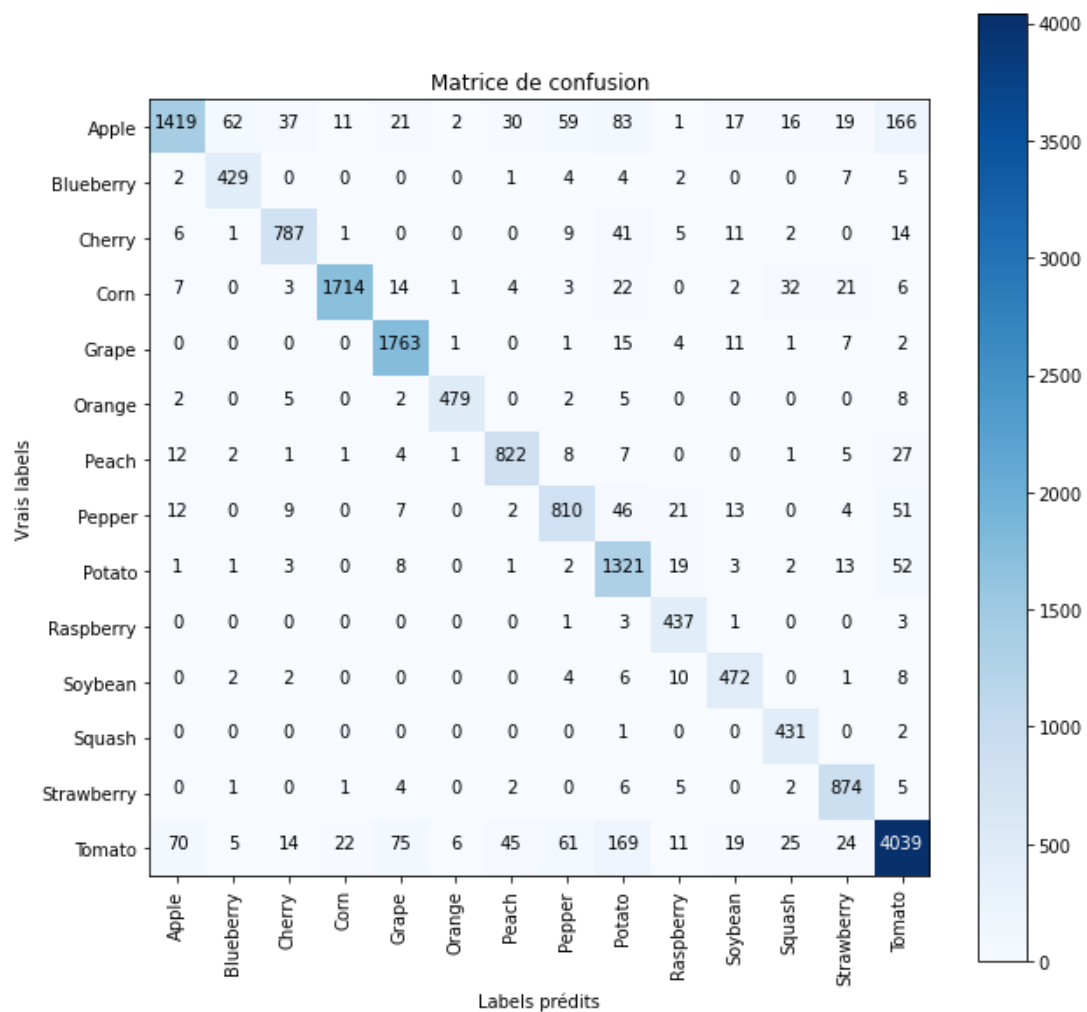


L'apprentissage a partir de ce sous-échantillon d'entraînement est très satisfaisant. Les performances convergent et atteignent de très bon niveaux. Il nous a fallu en revanche un peu plus d'Epoch pour atteindre le plateau d'apprentissage.

Classification Report:

	precision	recall	f1-score	support
Apple	0.93	0.73	0.82	1943
Blueberry	0.85	0.94	0.90	454
Cherry	0.91	0.90	0.91	877
Corn	0.98	0.94	0.96	1829
Grape	0.93	0.98	0.95	1805
Orange	0.98	0.95	0.96	503
Peach	0.91	0.92	0.91	891
Pepper	0.84	0.83	0.84	975
Potato	0.76	0.93	0.84	1426
Raspberry	0.85	0.98	0.91	445
Soybean	0.86	0.93	0.90	505
Squash	0.84	0.99	0.91	434
Strawberry	0.90	0.97	0.93	900
Tomato	0.92	0.88	0.90	4585
accuracy			0.90	17572
macro avg	0.89	0.92	0.90	17572
weighted avg	0.90	0.90	0.90	17572

Matrice de confusion:



Les performances du modèle restent excellentes bien que légèrement moins bonnes que le modèle entraîné sur les tout le dataset d'entraînement (70K images) avec déséquilibre. Toutefois, il est important de noter que nous avons maintenu la performance tout en réduisant notre dataset de 75%.

5- Transfer Learning

Nous allons dans cette section appliquer la technique de transfert de connaissance à partir d'une modèle de classification d'image existant VGG16 (d'imageNet). En effet, dans la littérature scientifique sur la classification de plante, l'utilisation de modèles existants (VGG, ResNet, etc) est très souvent utilisée.

Le principe est le suivant: ces modèles existants sont composés de **deux grandes parties**. La première est un ensemble de convolutions qui va permettre l'extraction des features. La seconde est une succession de dense layers qui a pour but de classifier le modèle de la voiture. Dans ce contexte, pour construire notre modèle, nous allons **initialiser les poids de la partie d'extraction de features par les poids du modèle pré-entraîné**. Les couches de la partie de classification seront remplacées et initialisées de manière aléatoire.

Model: "sequential"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, None, None, 512)	14714688
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 1024)	525312
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 14)	7182

```

=====
Total params: 15,771,982
Trainable params: 8,136,718
Non-trainable params: 7,635,264

```

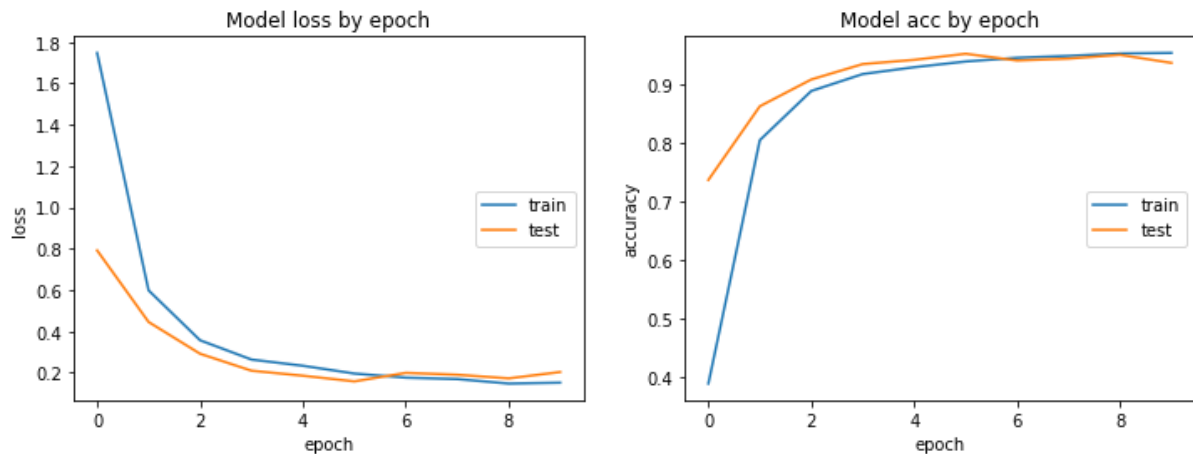
Au cours de l'apprentissage, nous avons testé deux approches:

- Approche 1: "freezer" entièrement les poids de la partie pré-entraînée
- Approche 2: entraîner 4 couches et "freezer" le reste

Pour la suite, nous ne présentons que les résultats de la deuxième approche qui nous a donné de meilleurs résultats.

Compte tenu du temps de traitement d'un tel modèle, nous avons réalisé l'apprentissage sur l'échantillon "Undersampling"

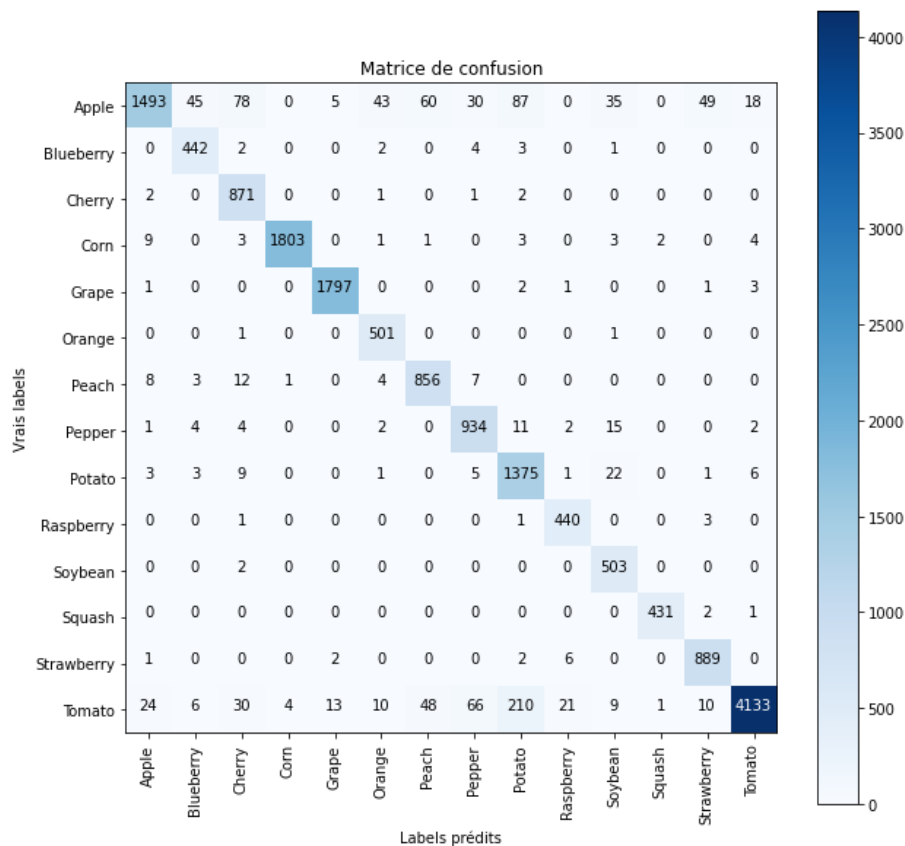
L'entraînement du modèle sur l'échantillon ré-échantillonne (Undersampling) donne les résultats suivants:



Avec le transfert, l'apprentissage atteint plus vite un plateau (des 4 epochs vs 50 epochs) en revanche le temps d'exécution est extrêmement long (presque 5H) même en ayant "unfreeze" que 4 couches du modèle VGG16.

	precision	recall	f1-score	support
Apple	0.97	0.77	0.86	1943
Blueberry	0.88	0.97	0.92	454
Cherry	0.86	0.99	0.92	877
Corn	1.00	0.99	0.99	1829
Grape	0.99	1.00	0.99	1805
Orange	0.89	1.00	0.94	503
Peach	0.89	0.96	0.92	891
Pepper	0.89	0.96	0.92	975
Potato	0.81	0.96	0.88	1426
Raspberry	0.93	0.99	0.96	445
Soybean	0.85	1.00	0.92	505
Squash	0.99	0.99	0.99	434
Strawberry	0.93	0.99	0.96	900
Tomato	0.99	0.90	0.94	4585
accuracy			0.94	17572
macro avg	0.92	0.96	0.94	17572
weighted avg	0.94	0.94	0.94	17572

Les performances sont très bonnes avec une accuracy globale de 0.94 et des f1 score par espèces entre 0.86 et 0.99.



5- Conclusions- Synthèse tirées des modèles précédents

		Baseline: CNN Classique	CNN + Générateurs de données	LeNet + Générateur de Données	LeNet + Générateur de Données + Rééquilibrage des données (UnderSampling)-50 Epochs	Transfer Learning VGG6+ UnderSampling (18K) - 10Epochs
F1 Score	Apple	0.76	0.88	0.95	0.82	0.86
	Blueberry	0.81	0.88	0.96	0.90	0.92
	Cherry	0.78	0.93	0.96	0.91	0.92
	Corn	0.96	0.97	0.98	0.96	0.99
	Grape	0.95	0.97	0.99	0.95	0.99
	Orange	0.92	0.95	0.98	0.96	0.94
	Peach	0.79	0.90	0.93	0.91	0.92
	Pepper	0.80	0.88	0.92	0.84	0.92
	Potato	0.78	0.87	0.93	0.84	0.88
	Raspberry	0.84	0.90	0.97	0.91	0.96
	Soybean	0.89	0.91	0.94	0.90	0.92
	Squash	0.90	0.96	0.98	0.91	0.99
	Strawberry	0.89	0.95	0.98	0.93	0.96
	Tomato	0.77	0.93	0.97	0.90	0.94
Accuracy		0.83	0.92	0.96	0.90	0.94

Nous avons entraîné au total 5 modèles pour la classification d'espèces, avec des performances très satisfaisantes (aussi bien en accuracy global que les F1 Score par espèces)

- le générateur d'images s'est avéré particulièrement efficace pour éviter le sur-apprentissage
- L'architecture LeNet semble plus adapté à notre problématique apportant la meilleure performance

- Undersampling n'améliore pas les performances du modèle mais permet de réduire notre dataset d'apprentissage de 75% tout en conservant le même niveau d'accuracy.
- l'approche par transfert learning paraît prometteuse en performance mais très gourmande en temps d'exécution.

IV- Détection de la maladie et classification du type de maladie

Lors de l'exploration des données, nous avons montré que l'information disponibles sur les maladie des plantes n'est pas homogène:

- Les types de maladies ne sont pas les mêmes d'une espèce à l'autre.
- Certaines espèces ne présentent aucune photo malade dans le dataset et à l'inverse certaines espèces n'ont que des images avec maladie.
- Certaines espèces ne présentent qu'un seul type de maladie et enfin d'autres espèces comme "Tomato" présentent différents types de maladies.

Compte tenu de ses disparités, nous avons décidé de travailler en deux temps:

- Tout d'abord, nous avons entraîné un modèle générique permettant de détecter la présence de maladie indépendamment de l'espèce de la plante,
- et dans un deuxième, nous avons travaillé sur des modèles dédiés par catégorie de plantes pour classifier le type de maladie.

1- Détection de la maladie

Dans cette partie, on entraîne un modèle de Deep Learning à détecter si la plante est malade ou pas, toutes catégories confondues. Comme pour la classification par catégorie, on a utilisé le modèle LeNet combiné du générateur de Data.

Après entraînement du modèle, on obtient une "accuracy" de **98%**. Le modèle arrive facilement à détecter les plantes malades des non malades car la maladie de la plante se présente sous forme d'une non-uniformité de la surface de la plante et un changement de la couleur verte de la plante : présence de taches noires, taches marrons ou encore jaunes, couleur verte dégradées, ... etc. Ce qui facilite l'apprentissage binaire du modèle.

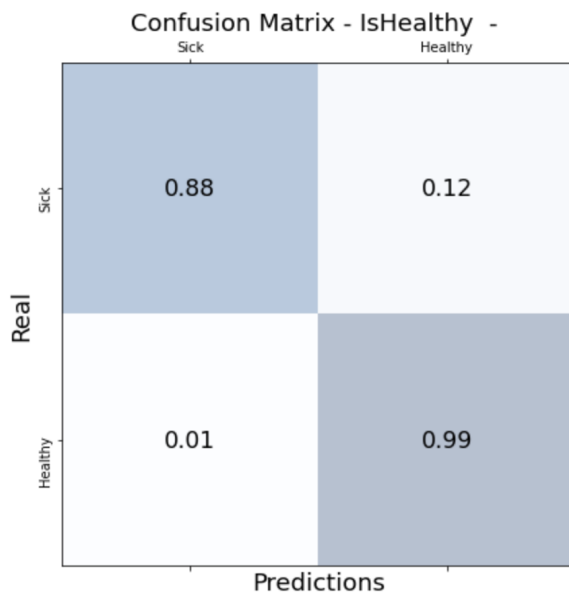
- Dataset d'entraînement : un répertoire réduit de 18K images
- Dataset de validation : le repo "Valid" contenant 17572 images

Ci-dessous les résultats du modèle appliqué au dataset de validation. D'après le rapport de classification, on obtient une "accuracy" de **92%** :

```
print(metrics.classification_report(y_valid, isHealthy_valid_pred))
```

	precision	recall	f1-score	support
0	1.00	0.88	0.94	12000
1	0.80	0.99	0.88	5572
accuracy			0.92	17572
macro avg	0.90	0.94	0.91	17572
weighted avg	0.93	0.92	0.92	17572

Les mêmes valeurs sont représentées sous forme de matrice de confusion comme suit :



Ce modèle de détection de présence de maladie a été aussi sauvegardé pour servir pour les prochaines classifications et combinaisons avec d'autres modèles Deep Learning.

2- Classification du type de maladie de la plante

Seules les espèces de plantes présentant au moins 2 types de maladie en plus du statut "healthy" ont été prises en comptes dans cette section:

- 1) Apple
- 2) Corn
- 3) Grape
- 4) Potato
- 5) Tomato

Nous allons détailler le modèle Tomato puis présenter la synthèse des résultats des 5 modèles sur les 5 catégories de plantes.

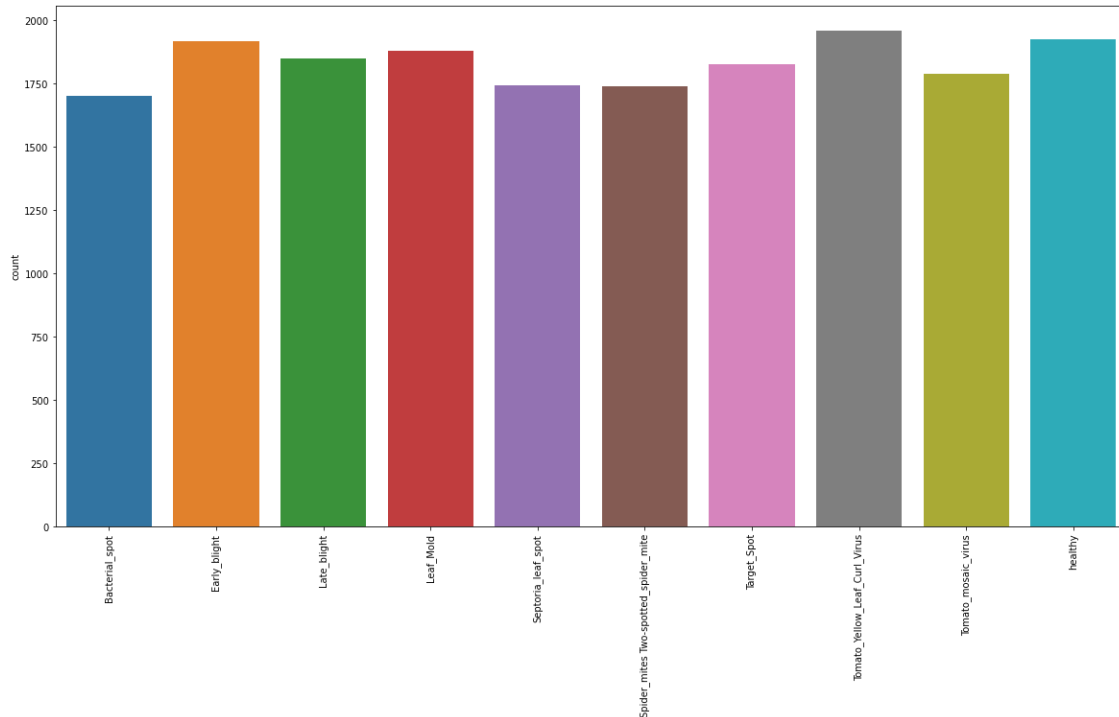
2.1- L'espèce "Tomato" comme exemple

Il s'agit dans cette partie d'évaluer la possibilité de classification par type de maladie. Nous présentons ici les résultats obtenus pour l'espèce "Tomato". Nous avons choisi la tomate qui présente non seulement un nombre important d'images mais également une grande variété de type de maladie dans la base de données.

On part de l'hypothèse que l'on connaît la catégorie de la plante. On va chercher alors à classer de type de maladie de la plante.

Dataset:

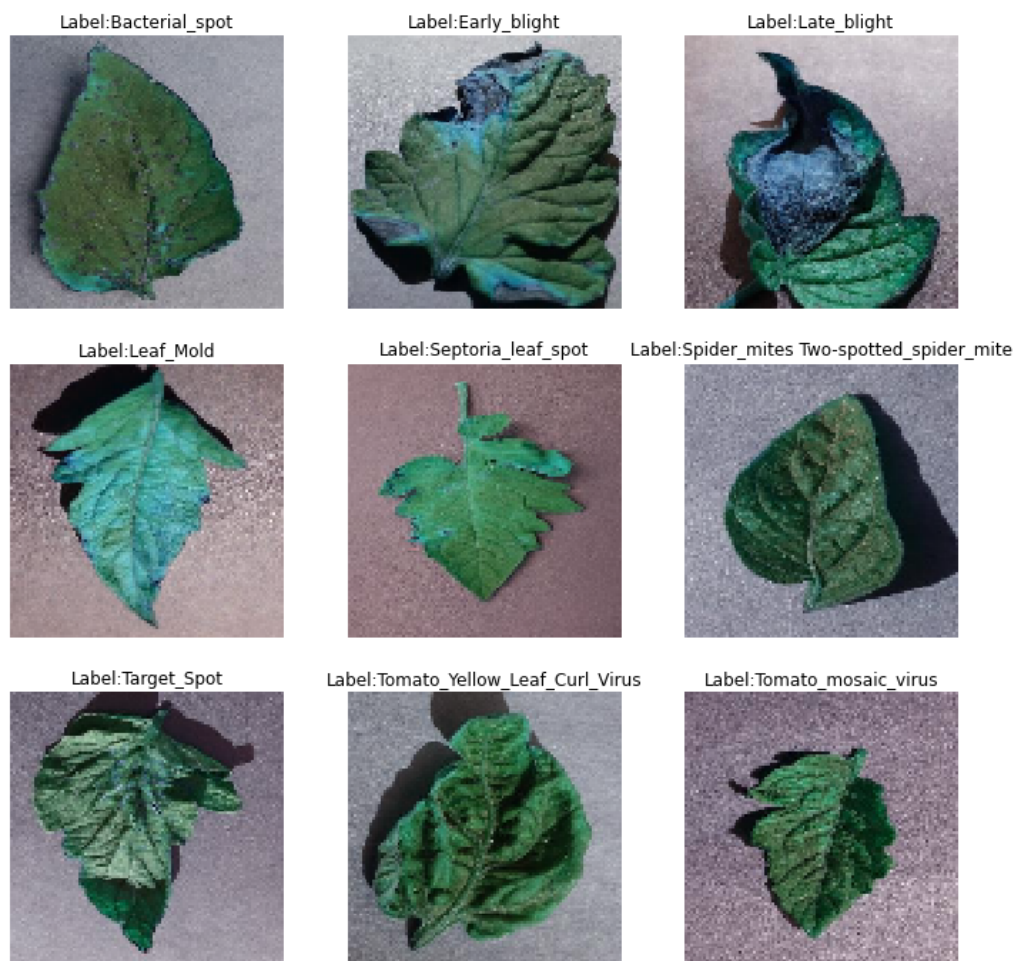
La base de données d'apprentissage contient 18 845 photos de tomates réparties en 10 catégories de maladie (dont la catégorie "Healthy" plus 9 maladies différentes):



La répartition entre les différentes classes est bien équilibrée.

L'échantillon de validation contient 4585 photos de tomates également uniformément réparties entre les différentes catégories de maladies.

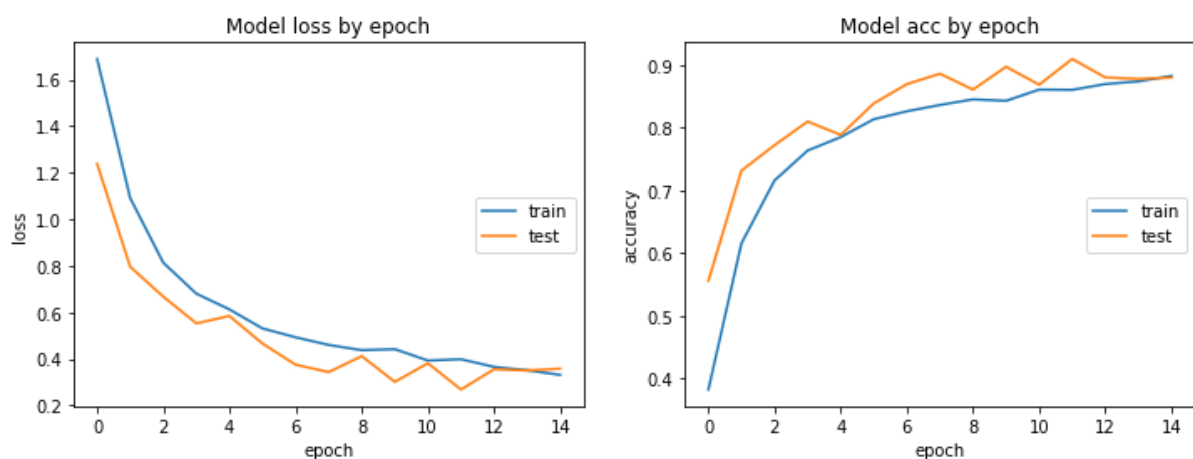
Data visualisation:



Ces images montrent à quel point la classification n'est pas une tâche aisée. La distinction entre les différentes maladies semble particulièrement difficile à première vue!

Deep Learning: Classification des maladies de "Tomates"

Suite aux conclusions précédentes pour la classification d'espèce, nous avons choisi d'appliquer un modèle de **Deep Learning** de classification, utilisant une **architecture LeNet** avec **générateur de données** pour éviter le sur apprentissage.



Les performances du modèles atteignent des niveaux satisfaisants et stables après 15 Epochs et nous évitons bien le surapprentissage avec des performances convergentes entre l'échantillon d'apprentissage et l'échantillon de validation.

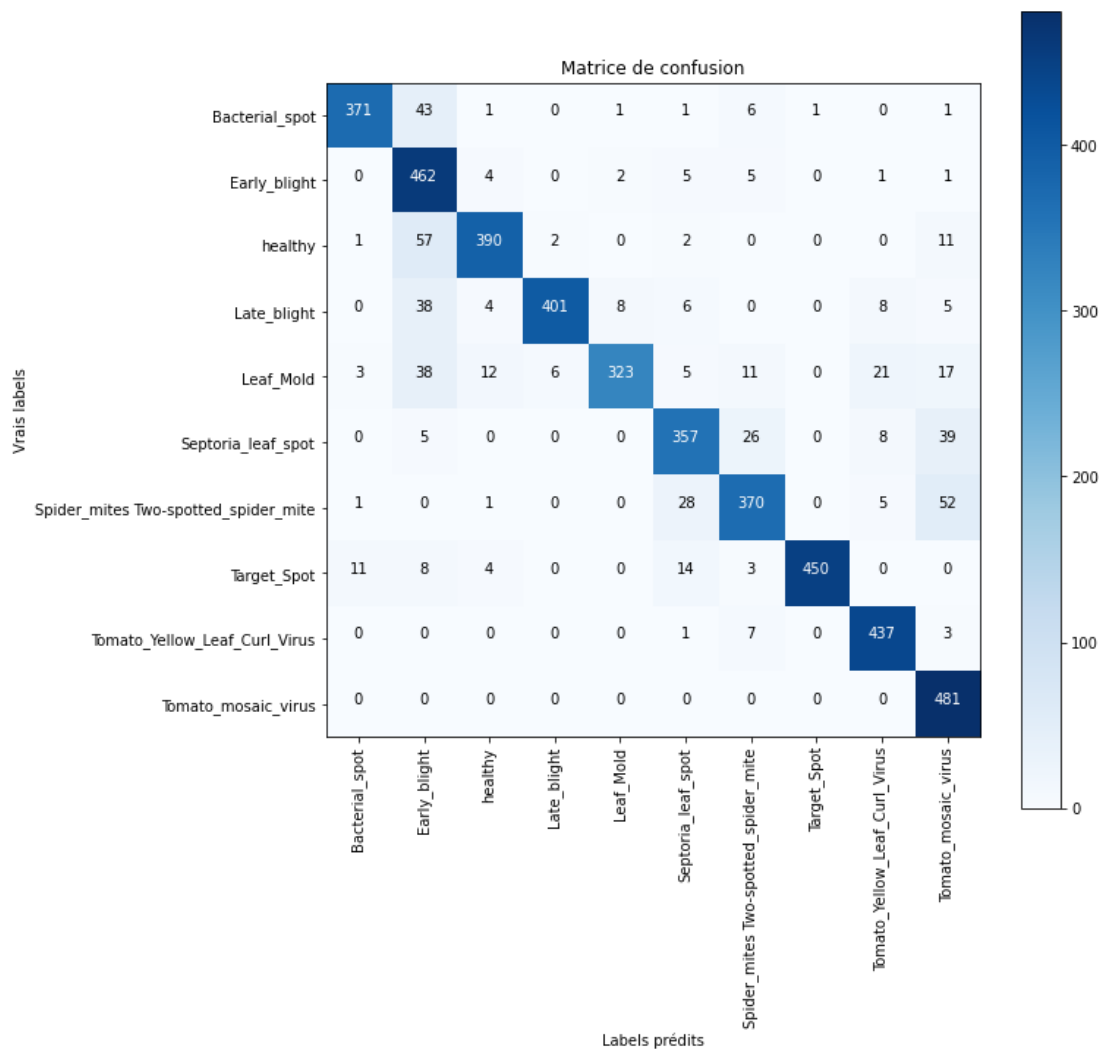
Classification Report

	precision	recall	f1-score	support
Bacterial_spot	0.96	0.92	0.94	425
Early_blight	0.75	0.97	0.85	480
Late_blight	0.95	0.82	0.88	463
Leaf_Mold	0.94	0.95	0.95	470
Septoria_leaf_spot	0.92	0.88	0.90	436
Spider_mites	0.93	0.85	0.89	435
Two-spotted_spider_mite	0.92	0.80	0.86	457
Target_Spot	0.98	0.93	0.95	490
Tomato_Yellow_Leaf_Curl_Virus	0.98	0.98	0.98	448
Tomato_mosaic_virus	0.87	0.99	0.93	481
healthy				
accuracy			0.91	4585
macro avg	0.92	0.91	0.91	4585
weighted avg	0.92	0.91	0.91	4585

Le rapport de classification confirme les performances du modèles avec une “accuracy” globale de 0.91 et avec des valeurs de f1-score correctes pour toutes les classes (Minimum 0.85, maximum: 0,98).

Les types de maladie qui présentent les meilleures performances sont “Tomato_mosaic_virus”, “Tomato_Yellow_leaf_curl_virus” et “Leaf_mold” car elles se démarquent par des caractéristiques très différentes des autres types de maladie (Couleur, forme) et donc plus facilement identifiables.

Matrice de confusion - Prédiction des types de maladie de Tomates:



2.2- Résultats sur les autres espèces

Nous avons procédé de même pour les 4 autres espèces de plantes présentant au moins 2 types de maladies (Apple, Corn, Grape et Potato).

On a gardé aussi la même stratégie de dataset que celle des modèles précédents:

- Dataset d'entraînement : un répertoire réduit de 18K images
- Dataset de validation : le repo "Valid" contenant 17572 images
- Modèle utilisé : architecture LeNet + générateur d'images

Ci-dessous une synthèse des résultats obtenus (plus de détails dans l'Annexe-1):

	# de types de maladie	Accuracy du modèle	F1-Score Min	F1-Score Max
Tomato	9	0.91	0.85	0.98
Apple	3	0.91	0.89	0.95
Corn	3	0.84	0.73	0.98
Grape	3	0.98	0.96	1
Potato	2	0.98	0.97	1

Les modèles de classification obtiennent d'excellentes performances sur toutes ces catégories.

V- Approche : un modèle pour triple classification

Dans cette approche on va explorer la piste d'entraîner un modèle de Deep Learning avec toutes les classes / catégories de plantes, avec leurs maladies associées.

Selon le dataset disponible, cela reviendrait à classifier 38 classes différentes. La sortie du modèle devrait donner une concaténation de deux informations qui peuvent prendre un des deux formats suivants :

- 1- "catégorie"_healthy : si on prédit que la feuille est saine.
- 2- "catégorie"_typeMaladie : si on prédit que la feuille est malade,
où "catégorie" peut prendre une des 14 valeurs disponibles dans le dataset
et "typeMaladie" peut prendre une des 20 valeurs disponibles dans le dataset.

Ce modèle, à première vue, permettrait de répondre rapidement et efficacement à l'objectif du projet : répondre aux 3 questions (catégorie, si maladie, quelle type de maladie). En effet, on obtient la performance suivante :

- **R1 :** Entraînement et validation du modèle sur le dataset réduit de 18K images dont 4501 images utilisées pour le test :
 - accuracy = 91%
 - F1-Score min = 53%
 - F1-Score max = 98%
- **R2 :** Application du modèle obtenu sur la dataset "Valid" contenant 17K images :
 - accuracy = 84%
 - F1-Score min = 55%
 - F1-Score max = 98%

Les rapports de Classification associés aux résultats R1 et R2 :

- Pour R1 :

```
print(metrics.classification_report(y_test_label, test_pred_LeNet1_label))
```

	precision	recall	f1-score	support
Apple_Apple_scab	0.86	0.70	0.78	81
Apple_Black_rot	0.91	0.84	0.87	82
Apple_Cedar_apple_rust	0.97	0.84	0.90	70
Apple_healthy	0.81	0.76	0.78	76
Blueberry_healthy	0.96	0.95	0.95	316
Cherry_Powdery_mildew	0.97	0.94	0.95	172
Cherry_healthy	0.99	0.95	0.97	160
Corn_Cercospora_leaf_spot Gray_leaf_spot	0.89	0.82	0.85	76
Corn_Common_rust	0.95	0.99	0.97	77
Corn_Northern_Leaf_Blight	0.95	0.91	0.93	76
Corn_healthy	0.99	0.93	0.96	74
Grape_Black_rot	0.79	0.75	0.77	84
Grape_Esca_(Black_Measles)	0.83	0.93	0.87	97
Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	0.94	0.92	0.93	98
Grape_healthy	0.98	0.79	0.87	70
Orange_Huanglongbing_(Citrus_greening)	1.00	0.97	0.99	327
Peach_Bacterial_spot	0.92	0.92	0.92	167
Peach_healthy	0.98	0.99	0.99	143
Pepper_Bacterial_spot	0.92	0.87	0.89	152
Pepper_healthy	0.93	0.86	0.90	150
Potato_Early_blight	0.87	0.98	0.92	99
Potato_Late_blight	0.77	0.89	0.83	116
Potato_healthy	0.85	0.91	0.88	103
Raspberry_healthy	0.91	1.00	0.95	321
Soybean_healthy	0.93	0.97	0.95	342
Squash_Powdery_mildew	0.97	0.99	0.98	316
Strawberry_Leaf_scorch	0.94	0.97	0.95	150
Strawberry_healthy	0.96	0.96	0.96	167
Tomato_Bacterial_spot	0.88	0.76	0.82	38
Tomato_Early_blight	0.61	0.74	0.67	23
Tomato_Late_blight	0.71	0.48	0.58	31
Tomato_Leaf_Mold	0.65	0.68	0.67	41
Tomato_Septoria_leaf_spot	0.67	0.47	0.55	38
Tomato_Spider_mites Two-spotted_spider_mite	0.77	0.60	0.68	45
Tomato_Target_Spot	0.47	0.62	0.53	32
Tomato_Tomato_Yellow_Leaf_Curl_Virus	0.85	0.82	0.84	28
Tomato_Tomato_mosaic_virus	0.78	0.92	0.85	39
Tomato_healthy	0.73	0.92	0.81	24
accuracy			0.91	4501
macro avg	0.86	0.85	0.85	4501
weighted avg	0.91	0.91	0.91	4501

- Pour R2 :

```
print(metrics.classification_report(y_valid_label, test_pred_LeNet1_validlabel))
```

	precision	recall	f1-score	support
Apple_Apple_scab	0.90	0.66	0.76	504
Apple_Black_rot	0.91	0.82	0.86	497
Apple_Cedar_apple_rust	0.92	0.83	0.87	440
Apple_healthy	0.87	0.76	0.81	502
Blueberry_healthy	0.84	0.95	0.89	454
Cherry_(including_sour)_Powdery_mildew	0.90	0.91	0.91	421
Cherry_(including_sour)_healthy	0.95	0.92	0.93	456
Corn_(maize)_Cercospora_leaf_spot Gray_leaf_spot	0.78	0.78	0.78	410
Corn_(maize)_Common_rust_	0.97	0.99	0.98	477
Corn_(maize)_Northern_Leaf_Blight	0.89	0.84	0.87	477
Corn_(maize)_healthy	0.99	0.91	0.95	465
Grape_Black_rot	0.82	0.81	0.81	472
Grape_Esca_(Black_Measles)	0.87	0.95	0.91	480
Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	0.89	0.95	0.92	430
Grape_healthy	0.99	0.80	0.88	423
Orange_Huanglongbing_(Citrus_greening)	0.94	0.97	0.95	503
Peach_Bacterial_spot	0.81	0.85	0.83	459
Peach_healthy	0.94	0.97	0.96	432
Pepper,_bell_Bacterial_spot	0.76	0.88	0.81	478
Pepper,_bell_healthy	0.85	0.85	0.85	497
Potato_Early_blight	0.84	0.96	0.90	485
Potato_Late_blight	0.63	0.89	0.74	485
Potato_healthy	0.78	0.91	0.84	456
Raspberry_healthy	0.79	0.99	0.88	445
Soybean_healthy	0.78	0.95	0.86	505
Squash_Powdery_mildew	0.83	0.99	0.90	434
Strawberry_Leaf_scorch	0.90	0.96	0.93	444
Strawberry_healthy	0.96	0.94	0.95	456
Tomato_Bacterial_spot	0.87	0.74	0.80	425
Tomato_Early_blight	0.73	0.62	0.67	480
Tomato_Late_blight	0.78	0.47	0.59	463
Tomato_Leaf_Mold	0.80	0.68	0.74	470
Tomato_Septoria_leaf_spot	0.68	0.47	0.55	436
Tomato_Spider_mites Two-spotted_spider_mite	0.72	0.78	0.75	435
Tomato_Target_Spot	0.74	0.62	0.67	457
Tomato_Tomato_Yellow_Leaf_Curl_Virus	0.95	0.84	0.89	490
Tomato_Tomato_mosaic_virus	0.87	0.91	0.89	448
Tomato_healthy	0.81	0.94	0.87	481
accuracy			0.84	17572
macro avg	0.85	0.84	0.84	17572
weighted avg	0.85	0.84	0.84	17572

Mais en analysant de plus près les résultats, nous trouvons que le modèle est limité à cause du biais du dataset. En effet, il existe des catégories avec des données non équilibrée:

- Les catégories (Cherry, Peach, Pepper, Squash) ont une seule et unique maladie. Donc le modèle ne saurait prédire qu'un seul type de maladie pour ces plantes.
- Pour Orange, même si la détection de catégorie avec ce modèle est bonne, en termes de maladie. Avec ce modèle, cette prédiction sera toujours associée à la maladie "Huanglongbing" même si la plante aurait pu être saine.
- Les catégories (Raspberry, Blueberry, Soybean) sont toutes saines dans le dataset. Donc le modèle associera ces catégories à l'état "health" même si la photo correspondait à une feuille malade.
- Pour la classe "Tomato_Septoria_Leaf_spot", on voit que le F1-score est de juste 55%, ce qui reste relativement faible. On va voir dans la suite avec l'approche combinant 3 modèles que ce F1-score peut être amélioré. En l'occurrence il aura la valeur de 89,29% sur le dataset de validation

Nous nous apercevons vite que pour ces catégories, les prédictions vont être tout le temps associées aux mêmes labels pour le type de maladie.

Afin de pallier ce biais introduit par le déséquilibre du dataset, on a choisi de faire appel aux modèles de prédictions indépendants. Cette approche sera expliquée dans le chapitre suivant.

Ci-dessous le tableau des 38 classes possibles à prédire :

Apple_Apple_scab	Corn_Common_rust	Peach_Bacterial_spot	Soybean_healthy	Tomato_Septoria_leaf_spot
Apple_Black_rot	Corn_Northern_Leaf_Blight	Peach_healthy	Squash_Powdery_mildew	Tomato_Spider_mites
Apple_Cedar_apple_rust	Corn_healthy	Pepper_Bacterial_spot	Strawberry_Leaf_scorch	'Tomato_Target_Spot'
Apple_healthy	Grape_Black_rot	'Pepper_healthy	Strawberry_healthy	Tomato_Yellow_Leaf_Curl_Virus
Blueberry_healthy	Grape_Esca	Potato_Early_blight	Tomato_Bacterial_spot	Tomato_mosaic_virus
Cherry_Powdery_mildew	Grape_Leaf_blight	Potato_Late_blight	Tomato_Early_blight	Tomato_healthy
Cherry_healthy	Grape_healthy	Potato_healthy	Tomato_Late_blight	
Corn_Cercospora_leaf_spot	Orange_Huanglongbing	Raspberry_healthy	'Tomato_Leaf_Mold	

- Les cases vertes correspondent aux plantes ayant un seul type de maladie
- Les cases avec des degrés de bleu correspondent aux plantes ayant juste une seule maladie répertoriée.

VI- Combinaison de modèles

Après avoir entraîné et comparé les performances de plusieurs modèles et compte tenu des biais de la database utilisée pour l'entraînement, il a été décidé de suivre l'approche suivante :

* Etape 1 : Classifier les plantes par catégorie avec le modèle 1 (qui classifie les 14 catégories de fruits/légumes du dataset)

* Etape 2 : En fonction du résultat de la prédiction de l'étape 1, décider quel modèle enchaîner pour la prédiction suivante:

- Si la prédiction est une catégorie qui fait partie de la liste [Apple, Corn, Grape, Potato, Tomato], on fait alors appel au modèle qui prédit directement l'état de santé de la feuille liée à sa catégorie.

Par exemple, si la catégorie prédite est "Tomato", appeler le modèle qui prédit les types de maladies de "Tomato"

En sortie on aura comme résultat deux types d'infos :

- catégorie + "healthy"
- catégorie + "nom de l'éventuelle maladie"

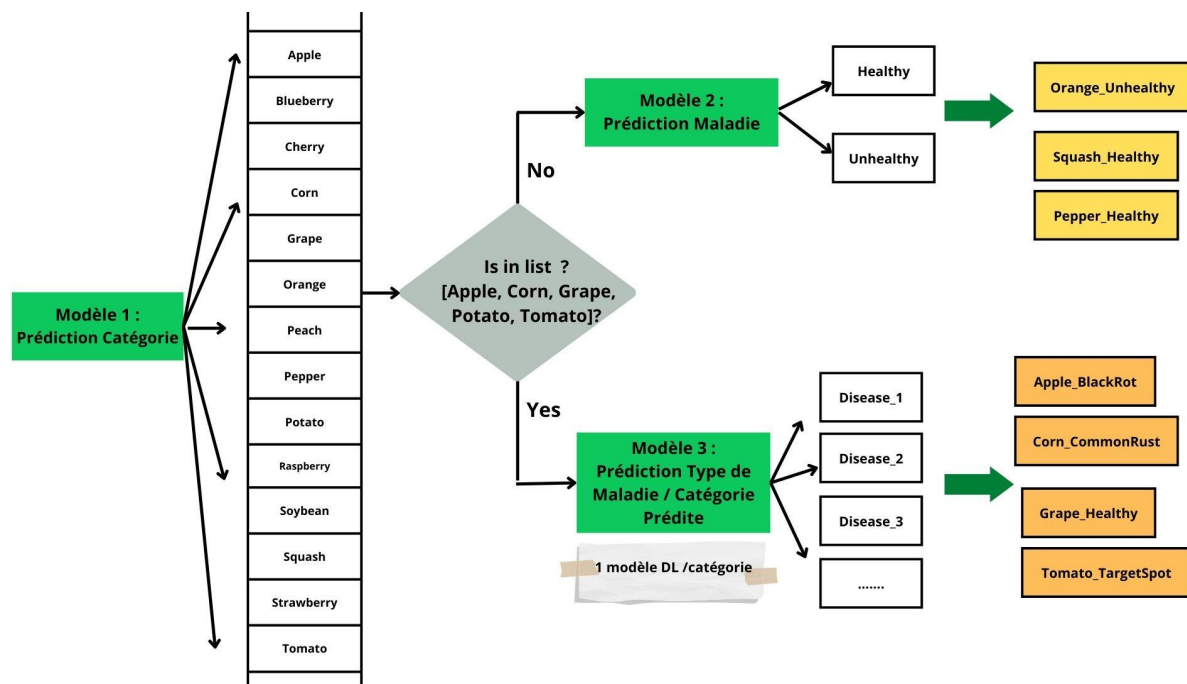
- b) Sinon, appeler le modèle qui détecte seulement la présence ou pas de maladie sur la feuille de la plante.

En sortie on aura comme résultat deux types d'infos :

- catégorie + "healthy"
- catégorie + "unhealthy"

Cette approche permet d'éviter d'associer par défaut un nom de maladie à une plante alors qu'elle aurait pu être "saine" juste parce que le modèle qui prédit le type de maladie n'est entraîné qu'avec des photos de la plante malade.

Le schéma suivant résume la démarche expliquée ci-dessus :

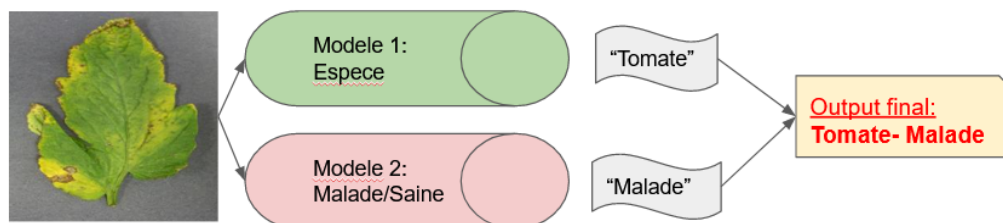


Dans la partie suivante, nous présentons les résultats de cette approche après avoir appliqué les différents modèles sur la dataset de validation (Plus de 17k images)

1- Détermination de l'espèce et détection de la maladie

Nous nous intéressons dans cette partie à la possibilité de classer les photos à la fois selon leur catégorie (Orange, Peach,, etc.) et la présence de maladie (sans spécification du type de maladie).

Etant donné que nous avons déjà entraîné deux modèles dédiés (un modèle pour déterminer la catégorie de plante et un modèle pour déterminer la présence de maladie), nous avons tout simplement combiné les résultats de ses deux modèles:



Rapport de classification:

	precision	recall	f1-score	support
Apple_0	0.94	0.83	0.88	1441
Apple_1	0.65	0.91	0.76	502
Blueberry_0	0.00	0.00	0.00	0
Blueberry_1	0.96	0.95	0.96	454
Cherry_0	0.98	0.74	0.85	421
Cherry_1	0.82	0.95	0.88	456
Corn_0	0.99	0.97	0.98	1364
Corn_1	0.98	0.98	0.98	465
Grape_0	0.99	0.97	0.98	1382
Grape_1	0.93	0.93	0.93	423
Orange_0	1.00	0.89	0.94	503
Orange_1	0.00	0.00	0.00	0
Peach_0	0.96	0.81	0.88	459
Peach_1	0.95	0.93	0.94	432
Pepper_0	0.97	0.82	0.89	478
Pepper_1	0.92	0.86	0.89	497
Potato_0	0.91	0.94	0.93	970
Potato_1	0.78	0.98	0.87	456
Raspberry_0	0.00	0.00	0.00	0
Raspberry_1	0.99	0.96	0.98	445
Soybean_0	0.00	0.00	0.00	0
Soybean_1	0.94	0.94	0.94	505
Squash_0	0.97	0.90	0.93	434
Squash_1	0.00	0.00	0.00	0
Strawberry_0	0.97	0.97	0.97	444
Strawberry_1	0.96	0.99	0.98	456
Tomato_0	0.95	0.76	0.84	4104
Tomato_1	0.33	0.96	0.49	481
accuracy			0.88	17572
macro avg	0.74	0.75	0.74	17572
weighted avg	0.93	0.88	0.89	17572

Nous obtenons une bonne performance de **0.88 d'accuracy**. En quelque sorte la combinaison des deux modèles cumule les erreurs de prédiction des deux modèles.

Pour rappel, l'accuracy du modèle de classification par espèces était de l'ordre de 0.96 et celui de la détermination de présence de maladie de 0.93.

Le modèle qui classe les 38 classes du dataset (chapitre V) obtient une performance globale de **84% sur le dataset de validation**. Donc la combinaison des deux modèles permet ici d'améliorer la performance à 88% tout en ajoutant plus de précision sur l'état de certaines plantes.

Il convient toutefois de remarquer que la performance est à relativiser par espèces:

- Tomato: le modèle présente des difficultés à classer correctement les Tomates Saines. Probablement car ces dernières sont sous-représentées dans notre dataset relativement aux tomates malades. L'autre explication est que certaines maladies de tomate sont difficiles à repérer sur la feuille : le changement de la couleur verte est tellement peu visible que le modèle peut confondre une feuille saine avec une feuille malade.
- Les catégories de plantes qui ne présentent aucune image avec maladie ("Blueberry", "Soybean", "Raspberry") et à l'inverse sans maladie ("Squash", "Orange") dans notre dataset obtiennent des performances biaisées et bonnes par construction
- Pour les autres catégories par contre les performances sont bonnes et les deux modèles se complètent bien.

2- Détermination de l'espèce et détection du type de maladie

Comme expliqué précédemment, cette approche a été limitée aux 5 catégories : Apple, Corn, Grape, Potato et Tomato. Si une prédiction donne comme résultat une catégorie parmi les 5 citées, on applique le modèle de prédiction de type de maladie associée à cette catégorie. Les performances du modèle correspondant à chacune de ces 5 catégories est dans Annex 1.

Cette approche a été testé sur le dataset de validation et on a pu combiner les deux modèles (prédiction de catégorie + prédiction type de maladie). Ci-dessous, on a les résultats pour la catégorie "Apple".

Ci-dessous la matrice de confusion et le rapport de classification. La classe "not_Apple" correspond aux images qui ont été classifiées par erreur comme "Apple" par le modèle 1.

Malgré les erreurs de prédictions de catégorie et les erreurs de prédictions de type de maladie d'Apple, la performance globale reste très satisfaisante. En effet, la valeur min de F1-score est de 84% qui correspond à la détection de la maladie "Black_rot".

Matrice de confusion - Prédiction type de maladie d'"Apple" :

maladie_pred	Apple_scab	Black_rot	Cedar_apple_rust	healthy	not_Apple
Maladie_reelle					
Apple_scab	410	45	5	18	26
Black_rot	27	464	0	0	6
Cedar_apple_rust	1	7	399	4	29
healthy	10	28	0	440	24
not_Apple	16	58	10	31	15514

Rapport de classification - Prédiction type de maladie d'Apple :

	precision	recall	f1-score	support
Apple_scab	0.88	0.81	0.85	504
Black_rot	0.77	0.93	0.84	497
Cedar_apple_rust	0.96	0.91	0.93	440
healthy	0.89	0.88	0.88	502
not_Apple	0.99	0.99	0.99	15629
accuracy			0.98	17572
macro avg	0.90	0.90	0.90	17572
weighted avg	0.98	0.98	0.98	17572

NB : Pour les 4 autres catégories, les résultats des modèles sont disponibles dans l'annexe-2

3- Classification finale avec agrégation des modèles:

Ce tableau résume les performances obtenues à la sortie de la combinaison des 3 modèles (cf schéma page 32)

	Accuracy (% Bonnes predictions)
Classification Globale Especies + Maladie	89,24%
Classification Espece de la Plante	96,15%
Plantes Groupe I - Detection de la maladie	
Total	96,00%
Peach	97,76%
Cherry	88,83%
Pepper	96,72%
Strawberry	99,78%
Squash	91,01%
Orange	91,05%
Blueberry	99,34%
Raspberry	100,00%
Soybean	100,00%
Plantes Groupe II - Healthy ou Type de Maladie	
Total	89,80%
Tomato	87,35%
Apple	89,60%
Corn	82,89%
Grape	97,45%
Potato	97,12%

- La performance globale est très bonne avec une accuracy de 89,24%. Il s'agit de la bonne prédiction de la catégorie et l'état de santé de la plante, tout en associant le type de maladie aux 5 catégories choisies précédemment.
- En comparaison avec le modèle classifiant 38 classe qui avait 84% d'accuracy, il est clair que notre combinaison de modèles permet d'augmenter l'accuracy de 5,25%
- L'accuracy pour prédire la bonne catégorie est de 96,15%
- L'accuracy pour prédire la catégorie et l'état de santé de 9 catégories de plantes est en moyenne de 96%.
Pour les catégories comme "Soybean", l'accuracy était de 100% car le modèle est entraîné avec un seul état de santé, en l'occurrence toutes les photos de "Soybean" sont en bonne santé.
- L'accuracy pour prédire la catégorie et le type de maladie de 5 catégories est de 89,80%

VI- Points d'amélioration:

Après avoir présenté notre démarche de recherche du meilleur modèle de prédiction possible pour répondre aux objectifs de "PlantSpy", nous présentons ici quelques pistes d'améliorations :

1- La majeure contrainte de ce projet était le déséquilibre de database, notamment le manque de données pour certaines plantes. Afin de pouvoir générer un modèle généralisable, l'idéal serait d'inclure plus de photos avec plus de types de maladie pour avoir une meilleure pertinence de détection des types de maladie.

2- Aussi les photos de l'entraînement étaient prises sur un background uniforme. Donc on peut se poser la question sur la performance du modèle si le background change de couleur/forme ou si la plante n'est plus centrée sur la photo.

3- Dans une perspective plus long terme, on pourrait étendre le scope du projet aux 20 fruits et 20 légumes les plus cultivés en France. Cette approche pourrait donner naissance à un projet "PlantSpy" commercialisable et d'une utilité pour les agriculteurs ou les ingénieurs agronomes.

VII- Annexes

Annexe 1: Classification type de Maladie

A- Espèce “Grape”

	precision	recall	f1-score	support
Black_rot	0.99	0.93	0.96	472
Esca_(Black_Measles)	0.96	0.99	0.97	480
Leaf_blight_(Isariopsis_Leaf_Spot)	0.98	1.00	0.99	430
healthy	1.00	1.00	1.00	423
accuracy			0.98	1805
macro avg	0.98	0.98	0.98	1805
weighted avg	0.98	0.98	0.98	1805

B- Espèce “Corn”

	precision	recall	f1-score	support
Cercospora_leaf_spot Gray_leaf_spot	0.60	0.95	0.73	410
Common_rust_	0.98	0.61	0.75	477
Northern_Leaf_Blight	0.95	0.87	0.90	477
healthy	1.00	0.95	0.98	465
accuracy			0.84	1829
macro avg	0.88	0.85	0.84	1829
weighted avg	0.89	0.84	0.85	1829

C- Espèce “Potato”

	precision	recall	f1-score	support
Early_blight	0.95	1.00	0.97	485
Late_blight	1.00	0.94	0.97	485
healthy	0.99	1.00	1.00	456
accuracy			0.98	1426
macro avg	0.98	0.98	0.98	1426
weighted avg	0.98	0.98	0.98	1426

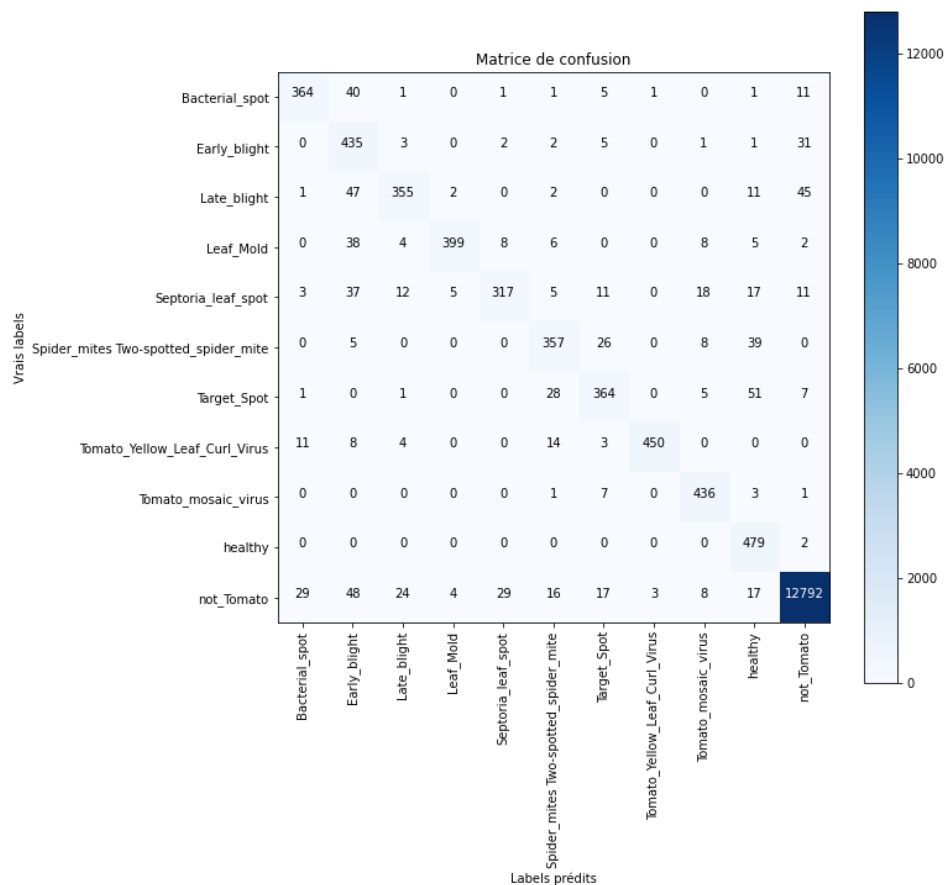
C- Espèce “Apple”

	precision	recall	f1-score	support
0	0.89	0.89	0.89	756
1	0.89	0.93	0.91	750
2	0.98	0.93	0.95	639
3	0.91	0.90	0.91	767
accuracy			0.91	2912
macro avg	0.92	0.91	0.91	2912
weighted avg	0.91	0.91	0.91	2912

Annexe 2 : Modèle combine final- Performance de la classification Espèces+Type de Maladie

Categorie "Tomato":

	precision	recall	f1-score	support
Bacterial_spot	0.89	0.86	0.87	425
Early_blight	0.66	0.91	0.76	480
Late_blight	0.88	0.77	0.82	463
Leaf_Mold	0.97	0.85	0.91	470
Septoria_leaf_spot	0.89	0.73	0.80	436
Spider_mites Two-spotted_spider_mite	0.83	0.82	0.82	435
Target_Spot	0.83	0.80	0.81	457
Tomato_Yellow_Leaf_Curl_Virus	0.99	0.92	0.95	490
Tomato_mosaic_virus	0.90	0.97	0.94	448
healthy	0.77	1.00	0.87	481
not_Tomato	0.99	0.98	0.99	12987
accuracy			0.95	17572
macro avg	0.87	0.87	0.87	17572
weighted avg	0.96	0.95	0.95	17572



Categorie “Potato”

maladie_pred	Early_blight	Late_blight	healthy	not_potato
Maladie_reelle				
Early_blight	480	0	0	5
Late_blight	27	447	3	8
healthy	0	0	449	7
not_potato	28	65	84	15969

	precision	recall	f1-score	support
Early_blight	0.90	0.99	0.94	485
Late_blight	0.87	0.92	0.90	485
healthy	0.84	0.98	0.91	456
not_potato	1.00	0.99	0.99	16146
accuracy			0.99	17572
macro avg	0.90	0.97	0.93	17572
weighted avg	0.99	0.99	0.99	17572

Categorie “Grape”

maladie_pred	Black_rot	Esca_(Black_Measles)	Leaf_blight_(Isariopsis_Leaf_Spot)	healthy	not_grape
Maladie_reelle					
Black_rot	437	22	7	1	5
Esca_(Black_Measles)	3	475	1	0	1
Leaf_blight_(Isariopsis_Leaf_Spot)	0	0	423	1	6
healthy	0	0	0	398	25
not_grape	2	3	2	0	15760

	precision	recall	f1-score	support
Black_rot	0.99	0.93	0.96	472
Esca_(Black_Measles)	0.95	0.99	0.97	480
Leaf_blight_(Isariopsis_Leaf_Spot)	0.98	0.98	0.98	430
healthy	0.99	0.94	0.97	423
not_grape	1.00	1.00	1.00	15767
accuracy			1.00	17572
macro avg	0.98	0.97	0.97	17572
weighted avg	1.00	1.00	1.00	17572

Catégorie “Apple”

maladie_pred	Apple_scab	Black_rot	Cedar_apple_rust	healthy	not_Apple
Maladie_reelle					
Apple_scab	410	45	5	18	26
Black_rot	27	464	0	0	6
Cedar_apple_rust	1	7	399	4	29
healthy	10	28	0	440	24
not_Apple	16	58	10	31	15514

	precision	recall	f1-score	support
Apple_scab	0.88	0.81	0.85	504
Black_rot	0.77	0.93	0.84	497
Cedar_apple_rust	0.96	0.91	0.93	440
healthy	0.89	0.88	0.88	502
not_Apple	0.99	0.99	0.99	15629
accuracy			0.98	17572
macro avg	0.90	0.90	0.90	17572
weighted avg	0.98	0.98	0.98	17572

Catégorie “Corn”

maladie_pred	Cercospora_leaf_spot	Gray_leaf_spot	Common_rust_	Northern_Leaf_Blight	healthy	not_Corn
Maladie_reelle						
Cercospora_leaf_spot	Gray_leaf_spot	366	0	20	0	24
Common_rust_		185	291	0	0	1
Northern_Leaf_Blight		55	1	408	1	12
healthy		14	4	2	437	8
not_Corn		9	1	1	0	15732

	precision	recall	f1-score	support
Cercospora_leaf_spot	0.58	0.89	0.70	410
Gray_leaf_spot	0.98	0.61	0.75	477
Common_rust_	0.95	0.86	0.90	477
Northern_Leaf_Blight	1.00	0.94	0.97	465
healthy	1.00	1.00	1.00	15743
not_Corn				
accuracy			0.98	17572
macro avg	0.90	0.86	0.86	17572
weighted avg	0.99	0.98	0.98	17572

VIII- Programmes/Code

Nom de Fichier	Description du contenu
Model3_Categorie_TypeMaladie.ipynb	<p>Modele Combine - Classification Espece + type de maladie</p> <ul style="list-style-type: none"> - Especies: Tomato, Potato, Corn, Apple, Grape - Pas de modele detection de maladie
Model3_Categorie_DetectionMaladie_TypeMaladie_VersionFinale.ipynb	<p>Modele Combine final</p> <ul style="list-style-type: none"> A- Espece plantes B- DetectionMaladie (toutes plante sauf Tomato, Apple, Corn, Grapes, Potato) C- definir le type de maladie/Healthy (seulement pour Tomato, Apple, Corn, Grape, Potato)
Model2_Tomato_Diseases_CNN.ipynb	Classification du Type de Maladie "Tomato"
Model2_Potato_Diseases_CNN.ipynb	Classification du Type de Maladie "Potato"
Model2_Grape_Diseases_CNN.ipynb	Classification du Type de Maladie "Grape"
Model2_Corn_Diseases_CNN.ipynb	Classification du Type de Maladie "Corn"
Model2_Apple_Diseases_CNN.ipynb	Classification du Type de Maladie "Apple"
Model1_Classification_EspeciesPlantes.ipynb	Contient tous les modeles de la partie classification Espece
DoubleClassification_Categorie_DetectionMaladie.ipynb	Modèle combine: Espèces et Detection Maladie (2 modeles/ 2 outputs)
Model_Detection_Maladie_Classif_Binaire.ipynb	Modèle permettant de prédire si la feuille de plante contient une maladie ou pas.
Model_Classification_38_Classes_Categorie_TypeMaladie.ipynb	Modèle permettant de classier les 38 classes du dataset
Download_Dataset_from_Kaggle.ipynb	Code permettant de télécharger les dataset depuis Kaggle

