Introduction

l'agriculture intelligente ou l'agriculture 4.0 est l'application des technologies de l'information et des données pour optimiser des systèmes agricoles complexes. Cela implique des machines individuelles et toutes les opérations agricoles.

Agriculture
Importance économique
Besoins en alimentation

Informatique
Intelligence Artifice
Deep Learning

Agriculture renforce par meilleur technologies

Problématique

- Les maladies foliaires constituent une menace majeure pour la productivité globale et la qualité des vergers de pommiers.
- La méthode actuelle repose sur un dépistage manuel coûteux et chronophage. coûteux par des humains.
- La détection et le diagnostic précoces de la maladie réduisent le taux de maladie.

Objectifs du projet

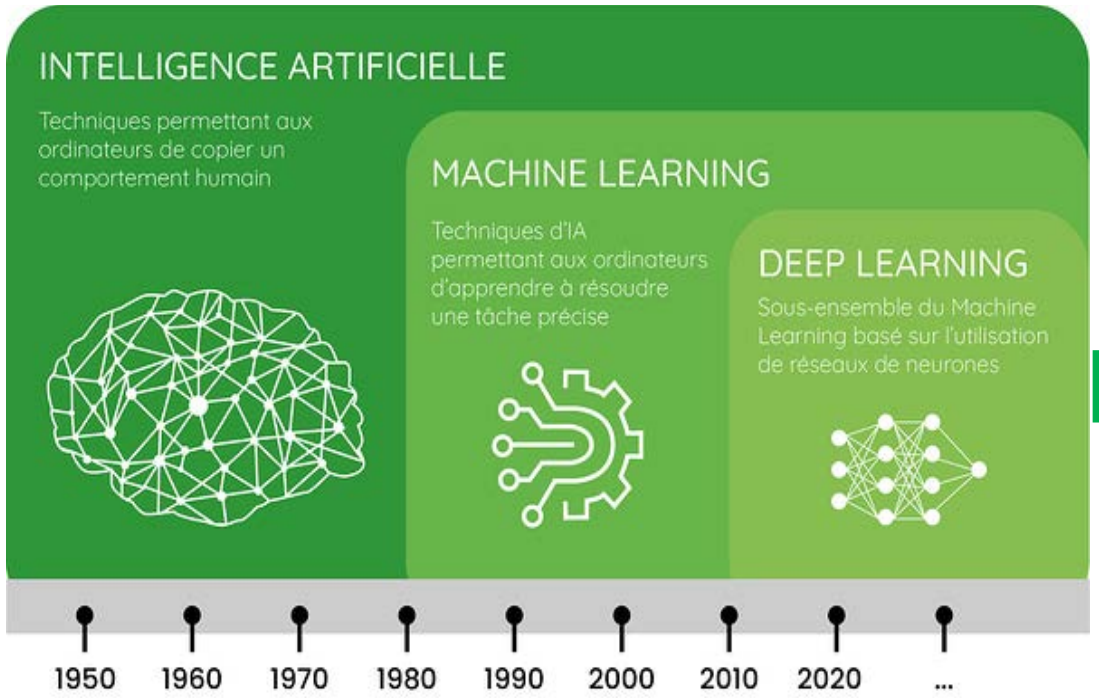
- Etablir une solution collaborative entre le traitement d'image et la phytopathologie permettra de réduire le temps du travail humain nécessaire par l'utilisation des algorithmes afin de faciliter l'identification des maladies des plantes.

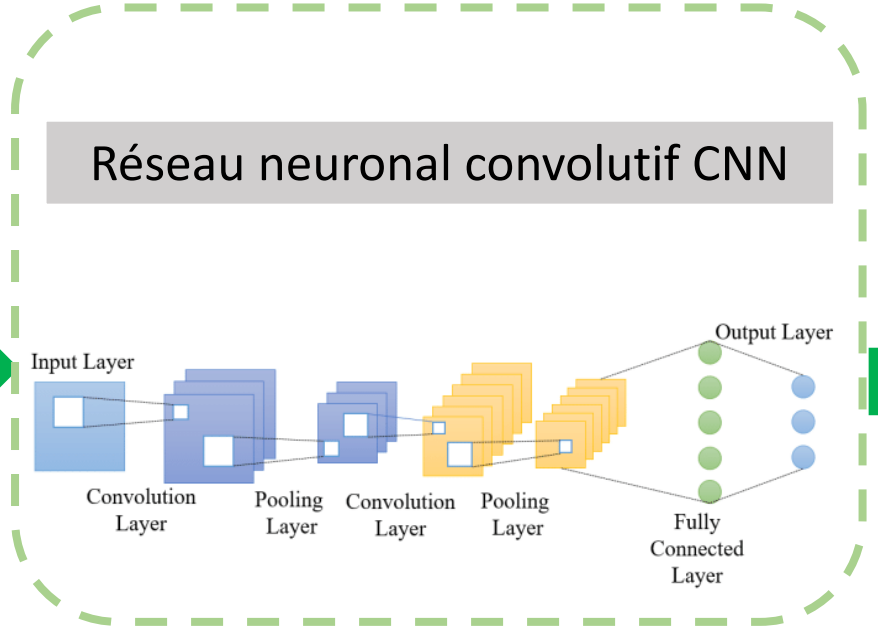
- développer un modèle basé sur l'apprentissage en profondeur pour identifier les maladies sur des images de feuilles de pommier.

- Mettre à la disposition des agriculteurs un système dédié à la prédiction des maladies des plantes, de telle façon que les gens pourront prendre une photo de leur plante et le transmettre vers le cloud afin d'obtenir un diagnostic en quelques secondes.

Techniques de l'intelligence artificielle

L'intelligence artificielle est un vaste domaine scientifique, apparue dès les années 50, peut-être définie comme l' ensemble de techniques permettant à des machines d'accomplir des tâches et de résoudre des problèmes normalement réservés aux humains et à certains animaux

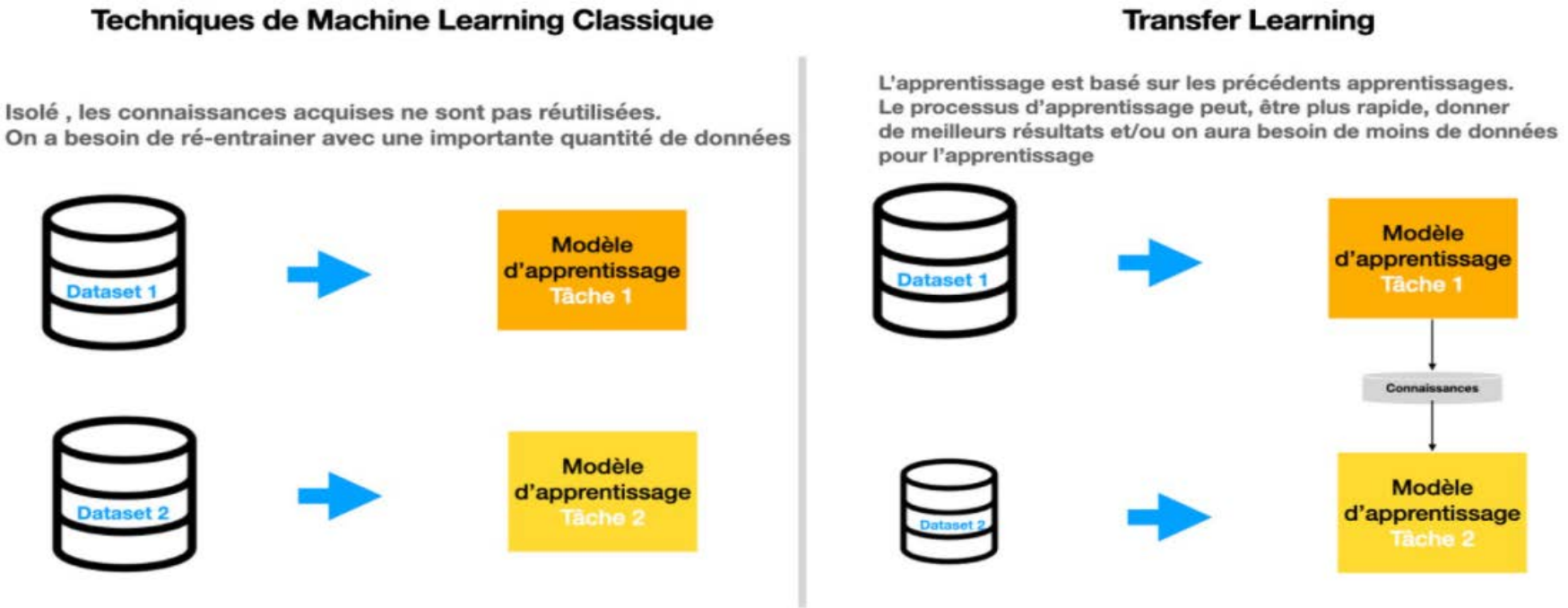




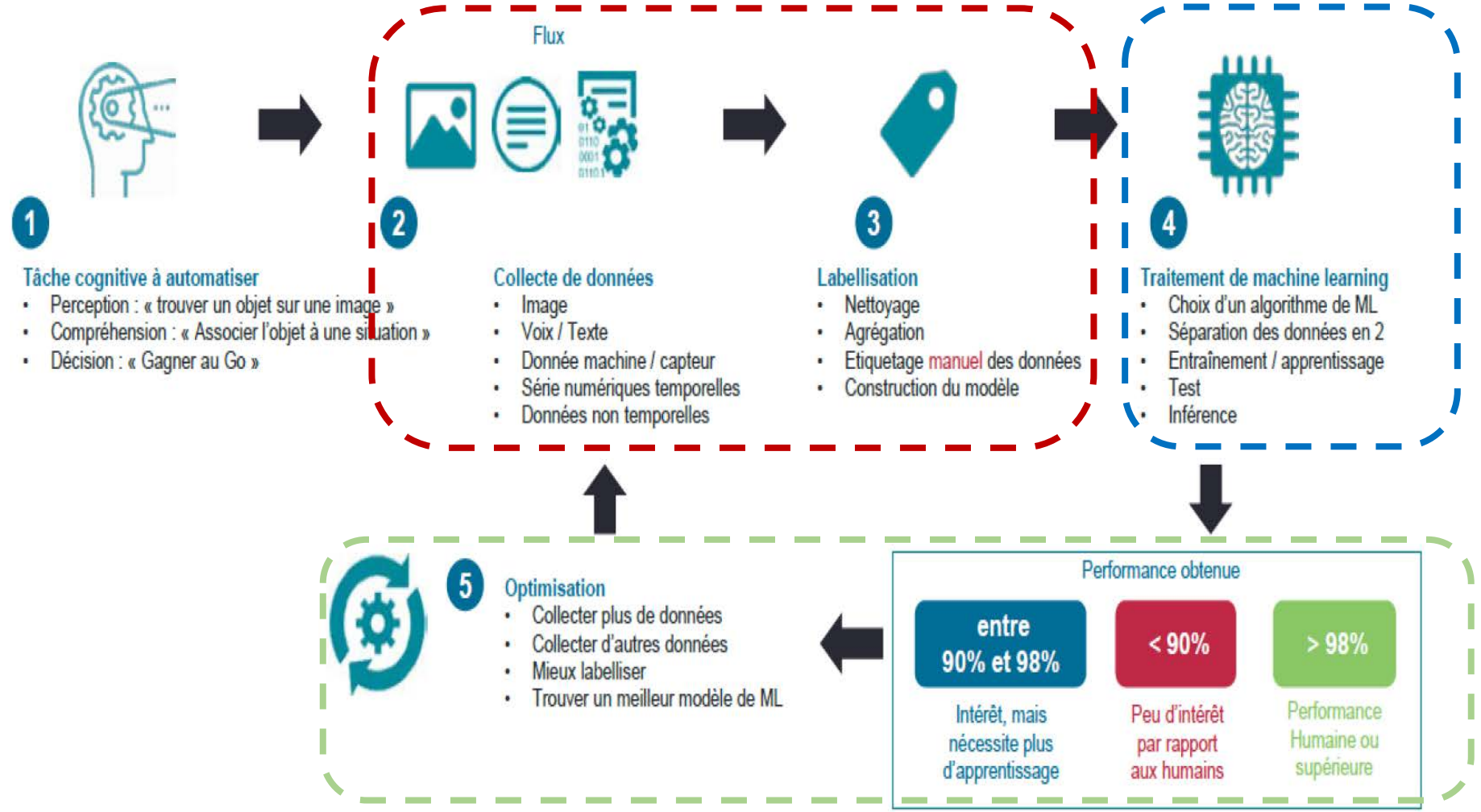
Modèle base sur CNN
Xception
VGG16
VGG19
ResNet50
AlexNet
ResNet152V2
InceptionV3
InceptionResNetV2
MobileNet
DenseNet121
EfficientNetB1

Transfer Learning

Le Transfer Learning permet de faire du Deep Learning sans avoir besoin d'y passer un mois de calculs. Le principe est d'utiliser les connaissances acquises par un réseau de neurones lors de la résolution d'un problème afin d'en résoudre un autre plus ou moins similaire.



Les étapes de développement du modèle



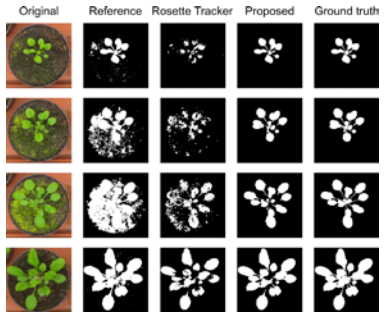
Préparation des données

Construction du modèle

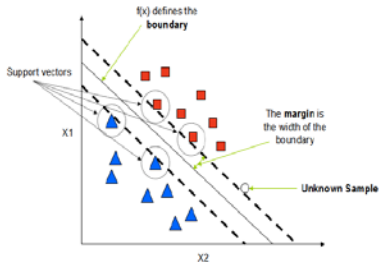
Évaluation et validation

Réseaux de classification

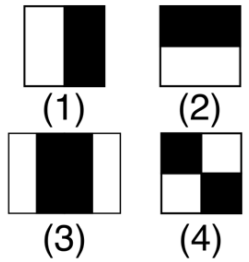
✓ Classification traditionnels



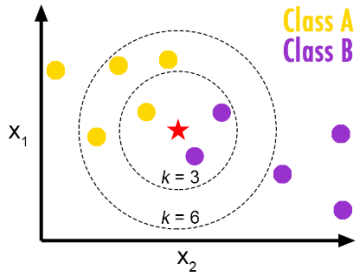
segmentation d'images



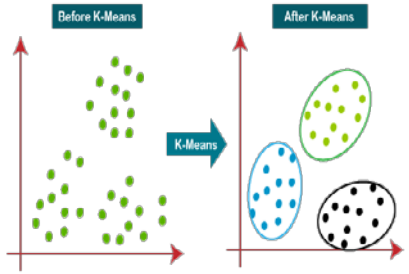
Support Vector Machines



pseudo-Haar

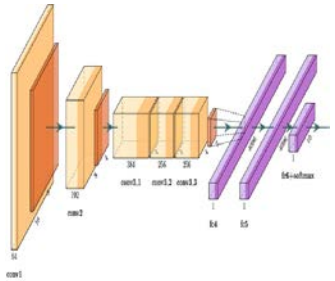


K-Nearest Neighbors

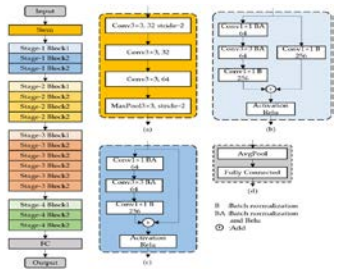


K-means

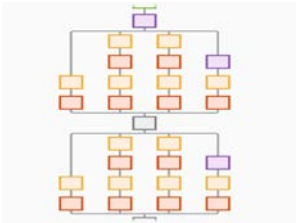
✓ réseaux convolutifs célèbres basés sur le CNN



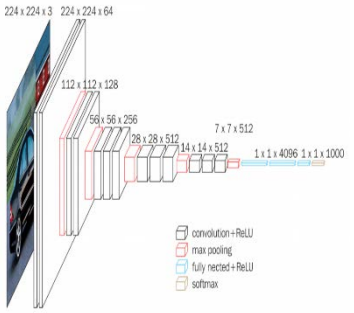
AlexNet



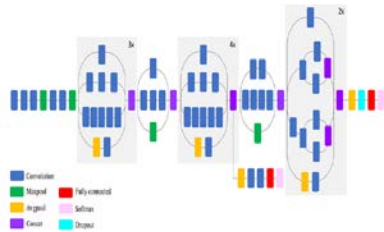
Resnet50



GoogleNet



VGGNet



Inception

Revue du littérateur

Travail	Pays	Plant	Data	Méthode de classification	Nbre maladies	Précision
Qiufeng Wu, 2018	Chine	Tomate	5550 (feuille de tomate)	CNN (SGD , ADM)	8	95%
José G. 2019	Brazil	Café	1747 (feuille de café)	CNN (t-SNE)	4	94 ,05 %
Jie Hang 2019	Chine	Blé Cerise Pomme	6108 (Feuilles de : Blé, Cerise et pomme).	CNN (SE)	7	91.70 %

Définition d'ensemble de données

Fine-Grained Visual Categorization

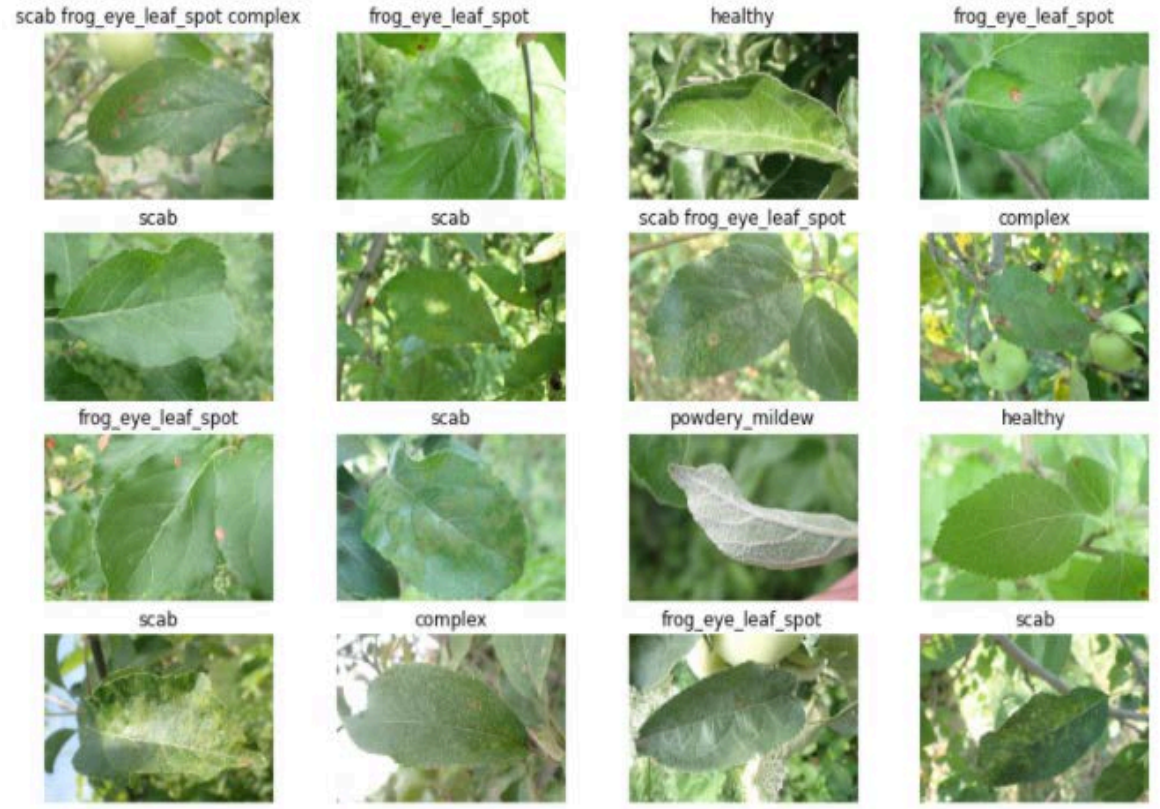
Data Explorer

16.1 GB

- test_images
- train_images
- sample_submission.csv
- train.csv

Summary

- 18.6k files
  - .jpg
  - .csv
- 4 columns



Echantillon de feuilles avec différentes maladies

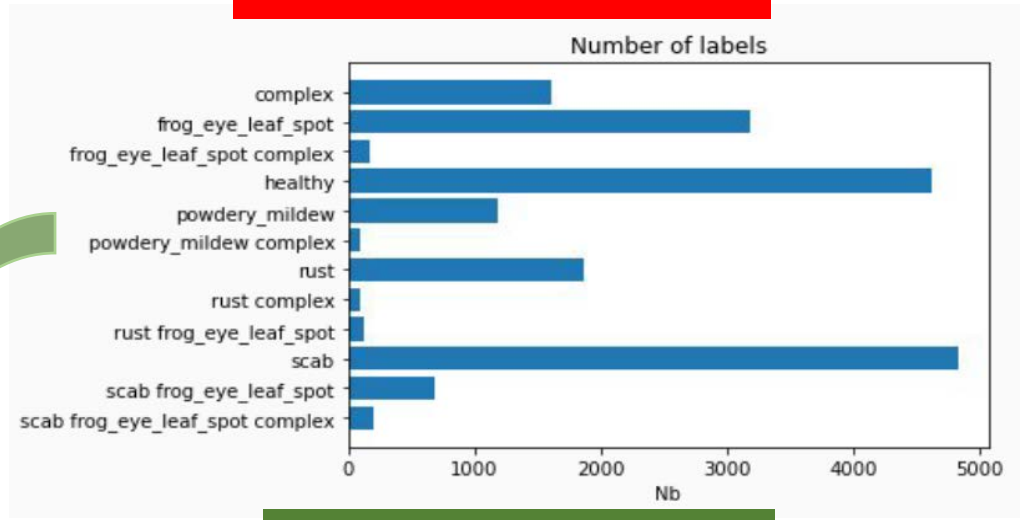
Prétraitement des données

✓ Analyse exploratoire

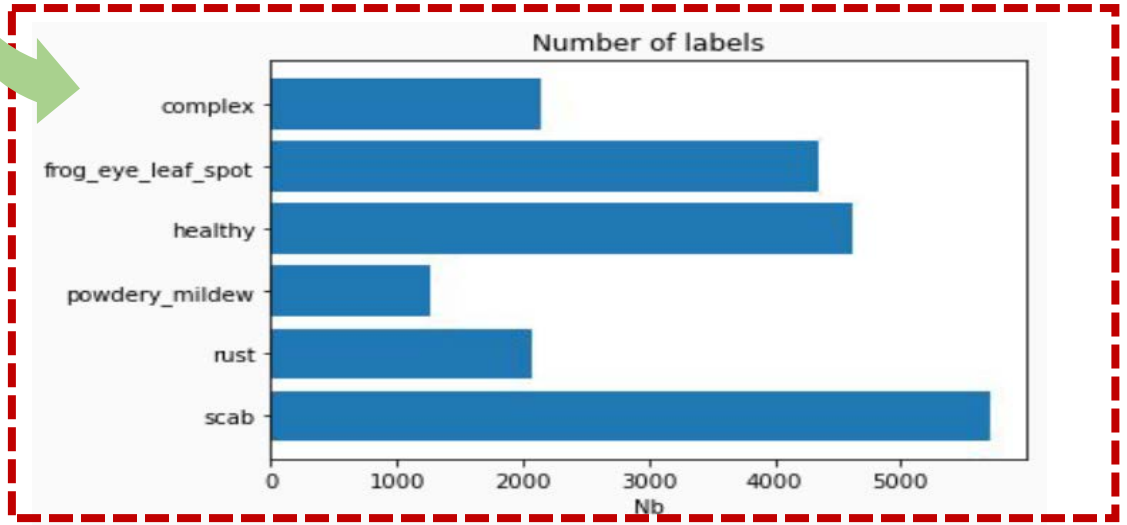
Étiquette	Nombre d'images	Pourcentage
scab	4826	25,9%
healthy	4624	24,8%
Forg_eye_leaf_spot	3181	17,1%
rust	1860	10,0%
complex	1602	8,6%
powdery_mildew	1184	6,4%
Scab Forg_eye_leaf_spot	686	3,7%
Scab Forg_eye_leaf_spot complex	200	1,1%
Forg_eye_leaf_spot complex	165	0,9%
Rust Forg_eye_leaf_spot	120	0,6%
Rust complex	97	0,5%
powdery_mildew complex	87	0,5%
Totale	**18632**	**100%**

déséquilibré énorme déséquilibre dans l'ensemble de données

Classification multi-class

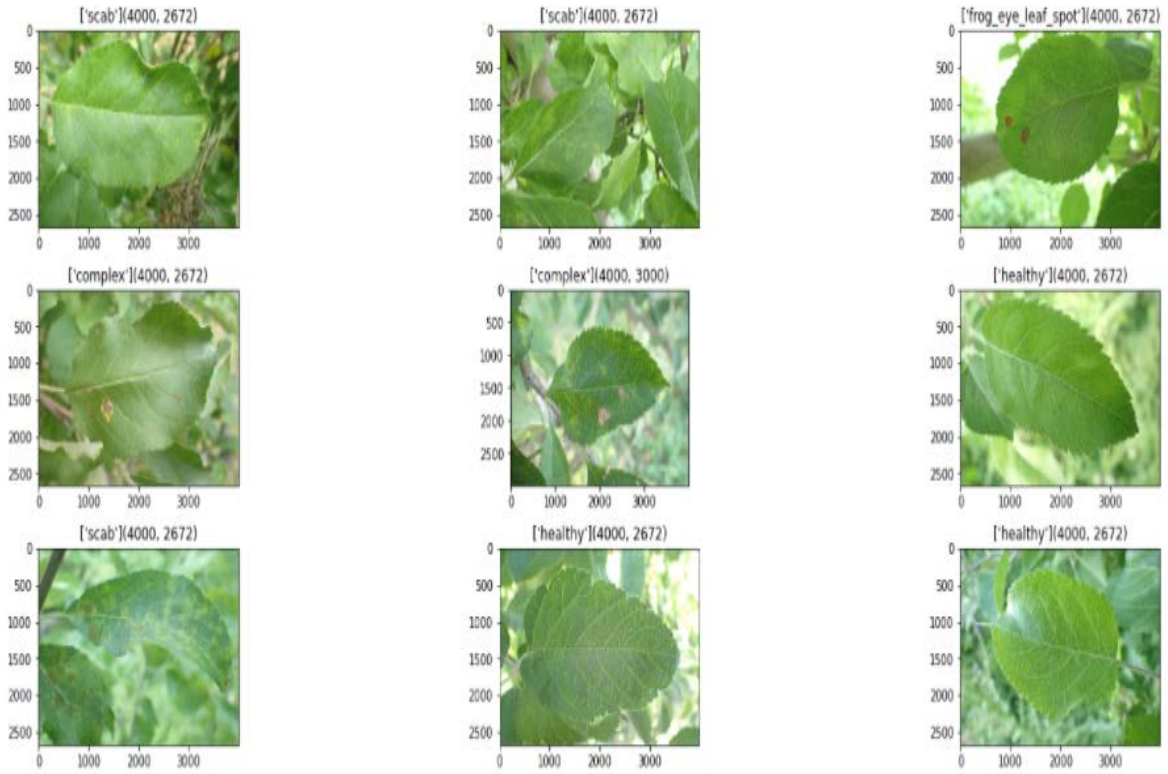


Classification multi-label

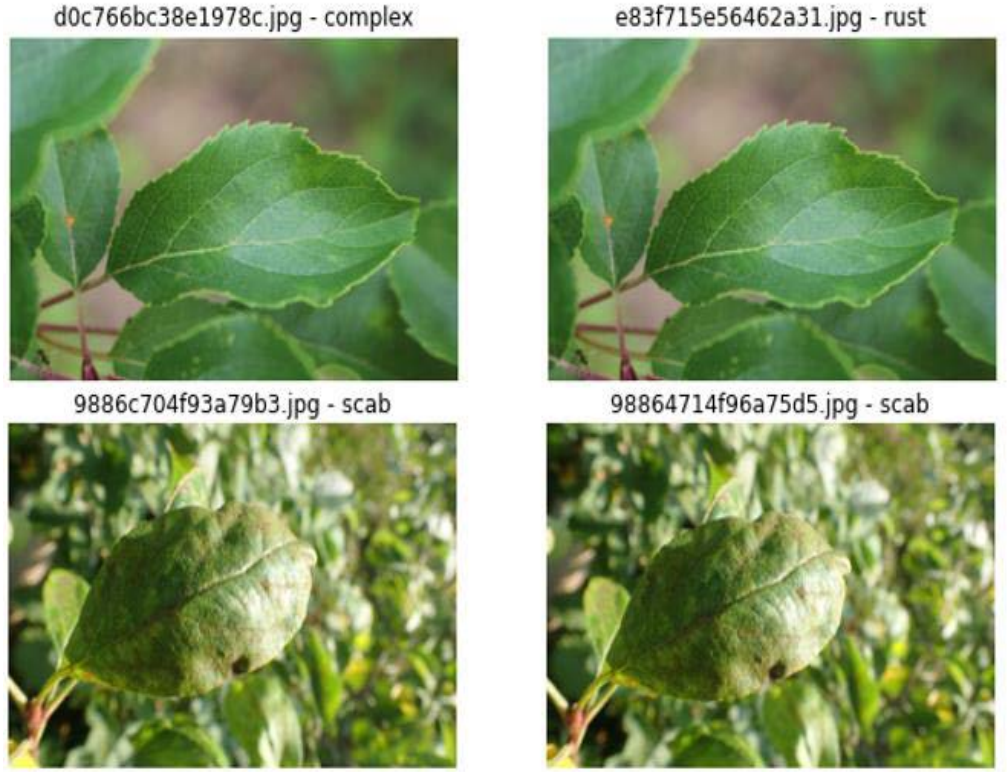


Prétraitement des données

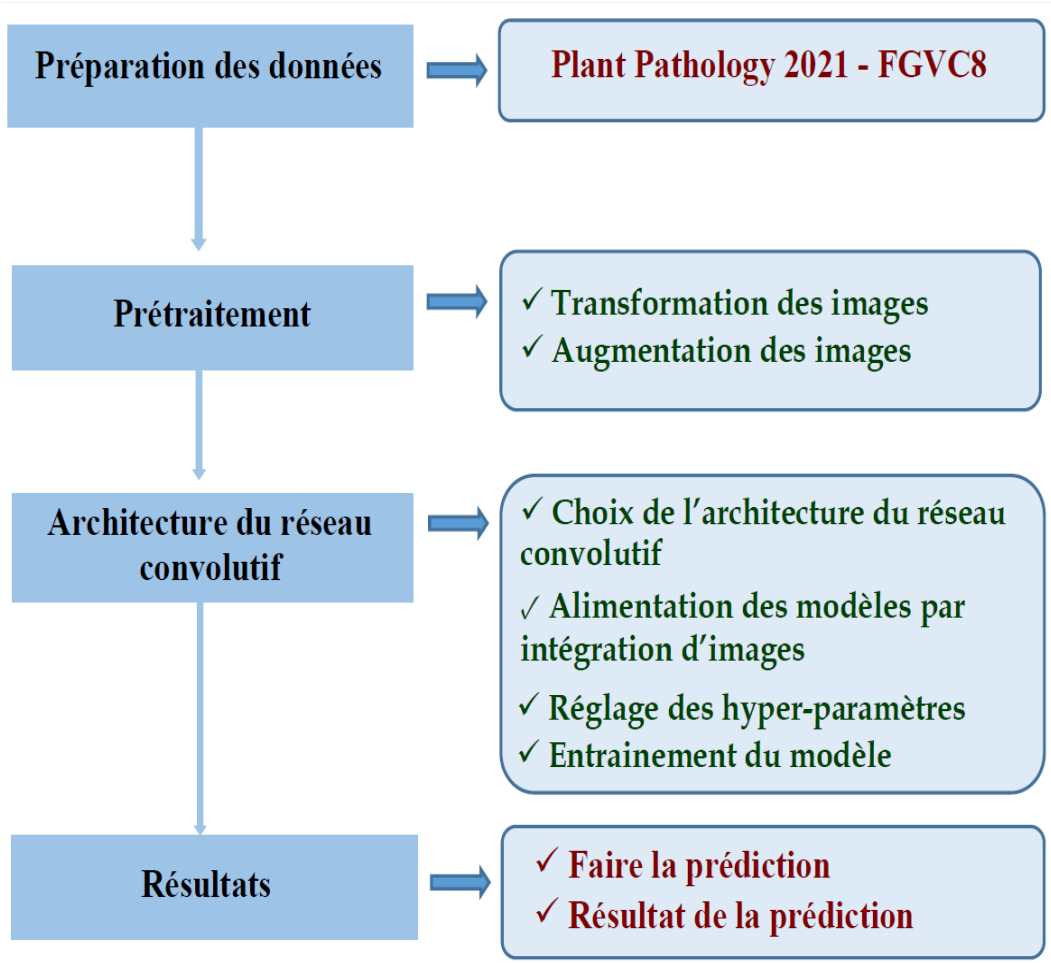
Exploration d'images: Redimensionner et traitement des images



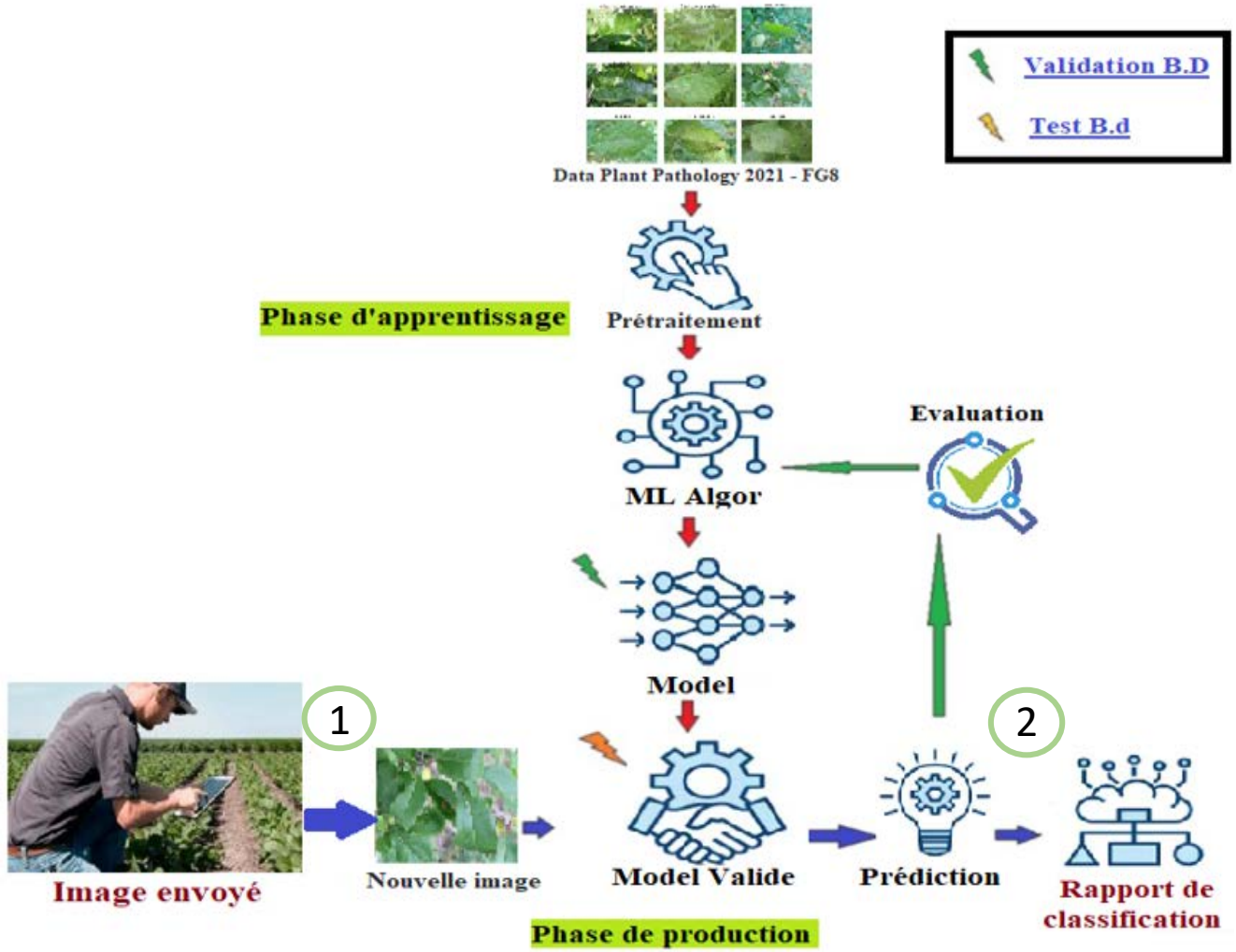
Suppression des doublons de l'ensemble de données



Architecture de système

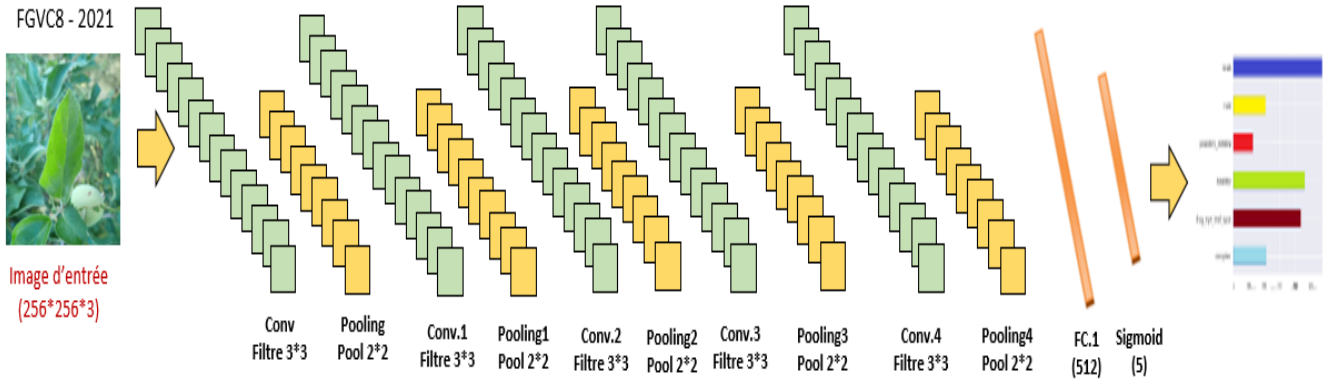


Organigramme d'architecture de système



Architecture des réseaux proposés

Modèle Proposé N° 01 :



Input	Layer Type	Output Size	kernel size	Activation
1	Conv	254x254x32	3x3	Relu
2	Pooling/max	127x127x32	3x3	Relu
3	Conv1	125 × 125× 64	3x3	Relu
4	Pooling1/max	62x62x64	2x2	Relu
5	Conv2	60x60x128	3x3	Relu
6	Pooling2/max	30x30x128	2x2	Relu
7	Conv3	28x28X128	3x3	Relu
8	Pooling3/max	14x14x128	2x2	Relu
9	Conv4	12x12x128	3x3	Relu
10	Pooling4/max	6x6x128	2x2	Relu
	Flatten	4608	-	Relu
11	Dense	512	-	Relu
12	Dense	5	-	Sigmoid

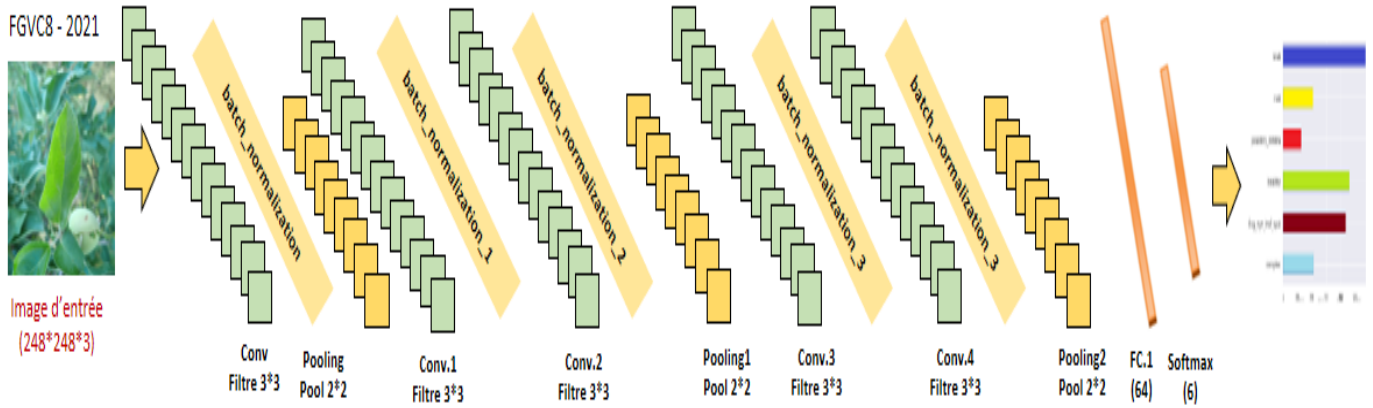
Couches du Modèle N°01

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 254, 254, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 127, 127, 32)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 125, 125, 64)      18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 62, 62, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 60, 60, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 30, 30, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 128)       147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 14, 14, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 12, 12, 128)       147584
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 6, 6, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0
_________________________________________________________________
dense (Dense)                (None, 512)               2359808
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 2565
=================================================================
Total params: 2,750,789
Trainable params: 2,750,789
Non-trainable params: 0
```

Architecture des couches du Modèle N° 01

Architecture des réseaux proposés

Modèle Proposé N° 02 (**dropout, Bach normalisation)**



Input	Layer Type	Output Size	kernel size	Activation
1	Conv	248x248x32	3x3	Relu
2	Pooling/max	82x82x32	3x3	Relu
3	Conv1	82x82x64	3x3	Relu
4	Conv2	82x82x64	3x3	Relu
5	Pooling1/max	41x41x64	3x3	Relu
6	Conv 3	41x41x128	3x3	Relu
7	Conv 4	41x41x128	3x3	Relu
8	Pooling3/max	20x20x128	2x2	Relu
	Flatten	51200	-	Relu
9	Dense	64	-	Relu
10	Dense	6	-	softmax

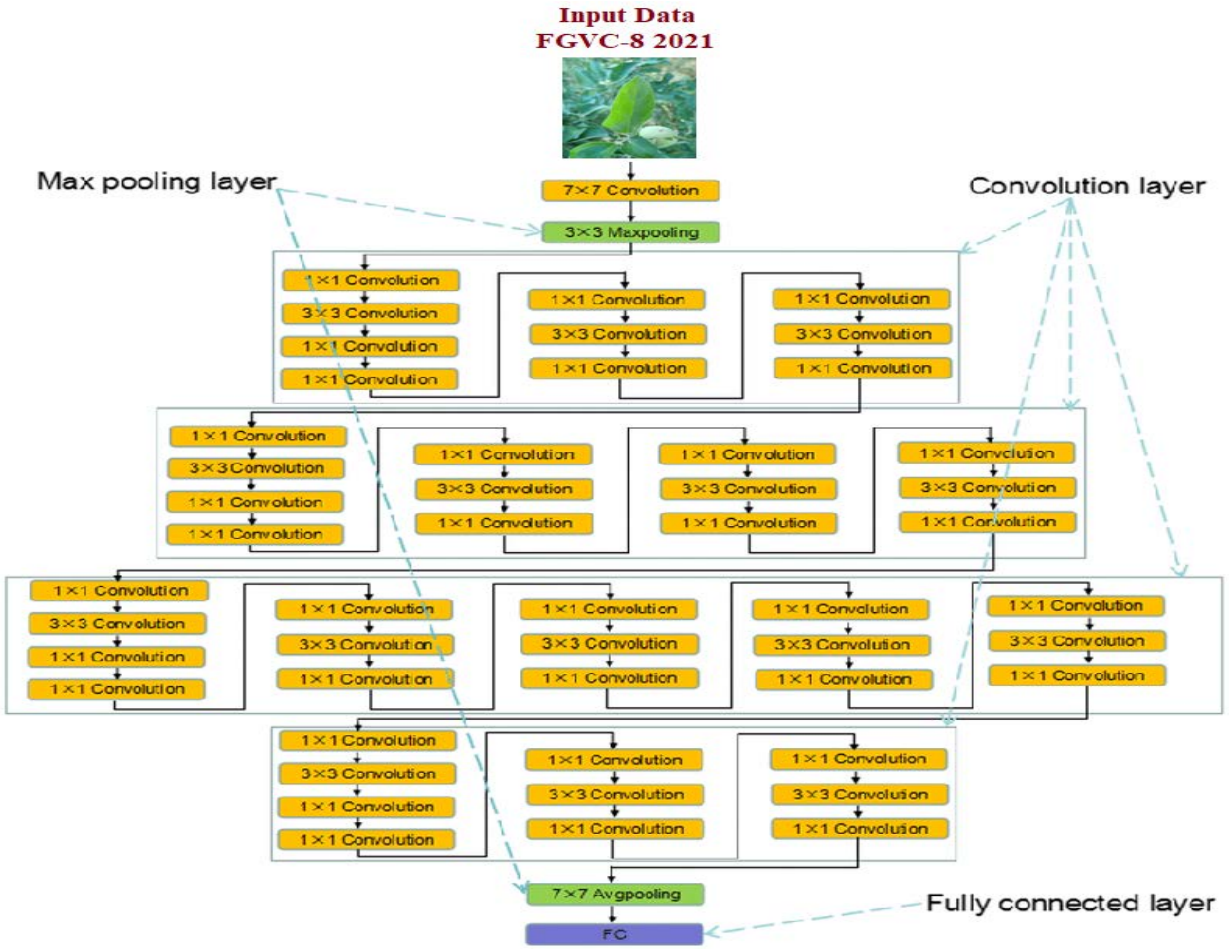
Couches du Modèle N°02

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 248, 248, 32)      896
_________________________________________________________________
batch_normalization (BatchNo (None, 248, 248, 32)      128
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 82, 82, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 82, 82, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 82, 82, 64)        18496
_________________________________________________________________
batch_normalization_1 (Batch (None, 82, 82, 64)        256
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 82, 82, 64)        36928
_________________________________________________________________
batch_normalization_2 (Batch (None, 82, 82, 64)        328
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 41, 41, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 41, 41, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 41, 41, 128)       73856
_________________________________________________________________
batch_normalization_3 (Batch (None, 41, 41, 128)       512
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 41, 41, 128)       147584
_________________________________________________________________
batch_normalization_4 (Batch (None, 41, 41, 128)       512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 20, 20, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 20, 20, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 51200)             0
_________________________________________________________________
dense (Dense)                (None, 64)                3276864
_________________________________________________________________
activation (Activation)      (None, 64)                0
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
batch_normalization_5 (Batch (None, 64)                256
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 390
_________________________________________________________________
activation_1 (Activation)    (None, 6)                 0
=================================================================
Total params: 3,557,006
Trainable params: 3,556,010
Non-trainable params: 996
```

Architecture des couches du Modèle N° 01

Architecture des réseaux proposés

Modèle Proposé N° 03 :



Architecture de réseau ResNet50

Model: "Resnet50"

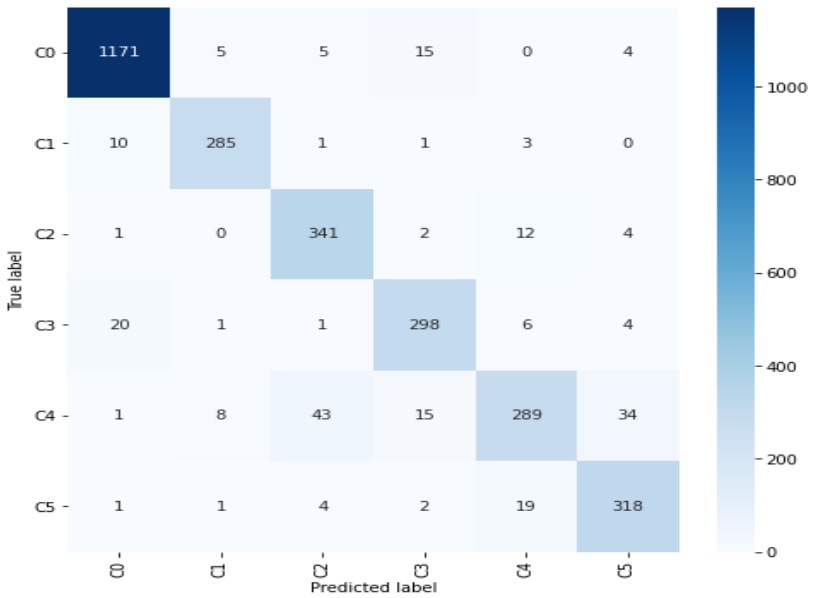
Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)	0	
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	input_2[0][0]
conv1_conv (Conv2D)	(None, 112, 112, 64)	9472	conv1_pad[0][0]
conv1_bn (BatchNormalization)	(None, 112, 112, 64)	256	conv1_conv[0][0]
conv1_relu (Activation)	(None, 112, 112, 64)	0	conv1_bn[0][0]
pool1_pad (ZeroPadding2D)	(None, 114, 114, 64)	0	conv1_relu[0][0]
pool1_pool (MaxPooling2D)	(None, 56, 56, 64)	0	pool1_pad[0][0]
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 64)	4160	pool1_pool[0][0]
conv2_block1_1_bn (BatchNormali	(None, 56, 56, 64)	256	conv2_block1_1_conv[0][0]
conv2_block1_1_relu (Activation	(None, 56, 56, 64)	0	conv2_block1_1_bn[0][0]
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 64)	36928	conv2_block1_1_relu[0][0]
conv2_block1_2_bn (BatchNormali	(None, 56, 56, 64)	256	conv2_block1_2_conv[0][0]
conv2_block1_2_relu (Activation	(None, 56, 56, 64)	0	conv2_block1_2_bn[0][0]
conv2_block1_0_conv (Conv2D)	(None, 56, 56, 256)	16640	pool1_pool[0][0]
.			
.			
.			
conv5_block2_out (Activation)	(None, 7, 7, 2048)	0	conv5_block2_add[0][0]
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 512)	1049088	conv5_block2_out[0][0]
conv5_block3_1_bn (BatchNormali	(None, 7, 7, 512)	2048	conv5_block3_1_conv[0][0]
conv5_block3_1_relu (Activation	(None, 7, 7, 512)	0	conv5_block3_1_bn[0][0]
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 512)	2359808	conv5_block3_1_relu[0][0]
conv5_block3_2_bn (BatchNormali	(None, 7, 7, 512)	2048	conv5_block3_2_conv[0][0]
conv5_block3_2_relu (Activation	(None, 7, 7, 512)	0	conv5_block3_2_bn[0][0]
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	conv5_block3_2_relu[0][0]
conv5_block3_3_bn (BatchNormali	(None, 7, 7, 2048)	8192	conv5_block3_3_conv[0][0]
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_add[0][0]
global_average_pooling2d (Globa	(None, 2048)	0	conv5_block3_out[0][0]
dense (Dense)	(None, 64)	131136	global_average_pooling2d[0][0]
dense_1 (Dense)	(None, 16)	1040	dense[0][0]
dense_2 (Dense)	(None, 6)	102	dense_1[0][0]

Total params: 23,719,990
Trainable params: 23,666,870
Non-trainable params: 53,120

couches du Modèle N° 03

Comparaison des résultats obtenus

Modèle	Description d'Architecture utilisée	Nombre de paramètres	Paramètres traitable	Paramètres non traitable	Batch Size	Epoque	Précision sur la base d'apprentissage	Précision sur la base de validation	Erreur
Modèle N°01	5 couches de convolution	2, 750,789	2, 750,789	---	128	30	78,95%	83,09%	15,09%
	5 couches de pooling					40	78,01%	82,85%	18,65%
	2 couches de fully-connected.								
Modèle N°02	5 couches de convolution	3, 557,006	3, 556,010	996	32	30	77,44%	78.37%	19,30%
	5 couches de pooling								
	2 couches de fully-connected.					40	79,51%	73,99%	22,03%
	5 batch normalization et dropout								
Modèle N°03	ResNet50 (50 couche convolution)	23, 719,990	23, 666,870	53,12	32	30	90,68%	85,95%	46,60%
						40	92,46%	91,83%	28,34%



Matrice de Confusion pour le Modèle 03 (40 époque)

le modèle 3 pré-entraîne base sur ResNet50 présente les meilleurs résultats trouvés (91.83%)

Pour le modèle 1 on remarque que pour 30 époques donne de meilleurs résultats (une différence de 4.72%), et quand le nombre d'époques passe à 40, on remarque que les deux modèles diminués.


Keras
TensorFlow™
Flask
kaggle
python™
ANACONDA.

Démonstration

Conclusion et perspectives