الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHES CIENTIFIQUE

UNIVERSITEDR.YAHIA FARES DE MEDEA

FACULTE DE TECHNOLOGIE

Département du Génie Électrique



Projet de fin d'études

Présenté pour l'obtention du Diplôme De Master en Électronique

Spécialité : Electronique Des Systèmes Embarqués

THEME

Conception et implémentation d'un système de détection et classification des maladies foliaires

Proposé et dirigé par

Encadreur : Dr M.BOUHEDDA
Co-encadreur : Mr H.BENYEZZA

Préparé et présenté par

BENALLAL Khaoula
BOUATTOU Maroua

Année Universitaire 2020-2021

Dédicace

Je dédie ce modeste travail et ma profonde gratitude

A ma mère et mon père pour l'éducation qu'ils m'ont donnée. Par tous les moyens et au prix de tous les sacrifices qu'ils ont consentis envers moi depuis mon enfance

A ma belle-sœur et mes deux frères.

A toute ma famille.

A tous mes amis et surtout ma moitié, ma meilleur Amis et ma binôme BENALLAL Khaoula d'avoir été dans ma vie tout au long de notre voyage, ainsi qu'à tous les membres de sa famille.

Et aussi j'adresse mes sincères remerciements aux personnes qui nous ont apporté leurs aides et qui ont contribué de proche ou de loin à l'élaboration de ce mémoire.

Bouattou Maroua

Dédicace

Merci Allah (mon dieu) de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire" Ya Kayoum "

A mes chers parents symbole de sacrifice, de tendresse et d'amour,

Sont les moindres sentiments que je puisse vous témoigner. Quoi que je fasse, je ne pourrais jamais vous récompenser pour les grands sacrifices que vous avez faits et continuez de faire pour moi.

Je le dédie particulièrement à ma grand-mère.

Je tiens à faire parvenir mes sincères reconnaissances à ma meilleure amie BOUATTOU Maroua dont, la vigueur et la présence durant tout notre parcours.Et à toute sa famille.

A mes adorables sœurs : Miko, Hayota, Manar, WissO, Sihem khaoula, Chouchou, Sadou et Mimi.

A mes chers frères : Mustapha, Djamel, Nidhal et Moncef.

Toute ma famille en particulier mes tentes hiba, Souad, Hakima et fati .

A mes très chers amis.

A tous mes enseignants et mes amis d'universitéavec lesquels j'ai passé cinq années inoubliables.

A tous ceux qui sont proches de mon cœur et dont je n'ai pas cité le nom.

Khaoula



Remerciement

En préambule à ce mémoire nous remerciant ALLAH qui nous aide et nous donne la patience et le courage durant ces années d'étude.

Nous exprimons toutes nos reconnaissances et gratitudes à l'administration et à l'ensemble du corps enseignant d'Université Yahia Fares pour leurs efforts à nous garantir la continuité et l'aboutissement de cette formation de Master.

Nous tenons à remercier aussi et chaleureusement à ***Dr M.BOUHEDDA*** *et* ***Mr H.BENYEZZA*** *pour l'inspiration, l'aide et le temps qu'ils ont bien voulu nous consacrer.*

Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leurs aides et qui ont contribué de proche ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire et n'ont pas pu être cités ici.

ملخص

يعد تحديد أمراض الأوراق اليوم أمرًا مهمًا للغاية في الزراعة. يتطلب الكشف عن أمراض الأوراق قدرًا هائلاً من العمل والمعرفة في مجال النبات. يهدف هذا المشروع إلى تصميم وتحقيق نظام ذكي يمكن من اكتشاف وتصنيف أمراض النبات باستخدام شبكة عصبية تلافيفية متخصصة. تم تنفيذ هذا النظام المقترح في Raspberry ، وتظهر نتائج الاختبار فعالية ونجاح النظام المقترح.

الكلمات المفتاحية : تحديد أمراض الأوراق، نظام ذكي، شبكة عصبية تلافيفية، Raspberry.

Résumé

L'identification des maladies foliaires est aujourd'hui très importante dans le domaine agricole. La détection des maladies foliaires nécessite d'énormes quantités de travail et de connaissances dans le domaine végétal. Ce projet pour l'objectif de la conception et la réalisation d'un système intelligent qui permit de détecter et classifier les maladies des plantes à l'aide d'un réseau de neurones spécialisé CNN.

Ce système proposé a été implémenté dans un Raspberry, les résultats des tests montrent l'efficacité et la réussite de système proposé.

Mots clés : détection des maladies foliaires, système intelligent, Réseau de neurones convolutifs (CNN), Raspberry.

Abstract

Nowadays the identification of leaf diseases is very important in agriculture. Detecting leaf diseases requires enormous amounts of work and knowledge in the plant field. The aim of this project is the conception and the realization of an intelligent system which made it possible to detect and classify plant diseases using CNN.

This proposed system has been implemented in a Raspberry. The tests results show the effectiveness and the success of the proposed system.

Keywords: leaf disease detection, intelligent system, CNN, Raspberry.

Sommaire

Liste des abréviations

Liste de figures

Liste des tableaux

Introduction générale……..1

Chapitre I Etat de l'art…….……..... 2

I.1 Introduction 2

I.2 Maladies des plantes..... 2

I.2.1 Champignons 3

I.2.2 Oomycètes 4

I.2.3 Phytoplasmes 5

I.2.4 Bactéries 6

I.2.5 Virus 7

I.2.6 Nématodes 8

I.3 Identification des maladies des plantes 9

I.4 Développement de secteur agricole..... 9

I.4.1 Agriculture traditionnelle..... 10

I.4.2 Agriculture de précision 12

I.5 Techniques utilisées pour détecter les maladies des plantes 15

I.5.1 Drones..... 15

I.5.2 Agriculture robotisée 16

I.5.3 Applications pour Smartphone 17

I.5.4 Capteurs connectés 18

I.5.5 Biocapteurs 19

I.5.6 Sentir les plantes malades..... 20

I.5.7 Spectro-imagerie..... 21

I.6 Conclusion……...22

Chapitre II Généralité sur l'AI et CNN 23

II.1 Introduction..... 23

II.2 Intelligence Artificielle (IA) 23

II.2.1 Définition 23

II.2.2. Fonctionnement de l'Intelligence Artificielle 24

II.3 Apprentissage Profond (Deep Learning) 25

II.4 Différentes Type d`Apprentissage 25
II.4.1 Apprentissage Supervisées 25
II.4.2 Apprentissage Non Supervisées 26
II.5 Réseau de Neurones Artificielles (RNA) 26
II.6 Historique des Réseaux de Neurones Artificiels 26
II.7 Fonctionnement des Réseaux de Neurones Artificiels 27
II.8 Avantages et Inconvénients du Réseau de Neurone 28
II.8.1 Avantages 28
II.8.2 Inconvénients 28
II.9 Présentation des Types de Réseaux Neurones 29
II.9.1 Neurone Formel 29
II.9.2 Réseaux de Hopfield 29
II.9.3 ADALINE 30
II.9.4 Réseaux de Kohonen 31
II.9.5 Les Réseaux de RBF (Radial Basis Fonction) 32
II.9.6 Réseaux Neurones Convolutionnels (CNN) 33
II.9.7 Entrainement d'un Nouveau CNN 40
II.9.8 Modèles de CNN 40
II.9.9 Choix des Hyperparamétres 44
II.9.10 Avantages de CNN 45
II.10 Conclusion……..……..45
Chapitre III Présentation hardware et software 46
III.1 Introduction 46
III.2 Partie software 46
III.2.1 Anaconda 46
III.2.2 Spyder 53
III.2.3 TensorFlow 54
III.2.4 Keras 55
III.2.5 Google Colab 55
III.2.6 Caractéristiques 56
III.3 Partie hardware 56
III.3.1 Carte Raspberry Pi 56
III.3.2 Camera OpenMV……...60
III.10 Conclusion……..……..62
Chapitre IV Réalisation et Implémentation 63

IV.1 Introduction……..…….62
IV.2 Description de l'architecture du système proposé 63
IV.3 Conception de système proposé 65
IV.3.1 Equipment utilise 65
IV.3.2 Description de la base de données 65
IV.3.3 Modèles CNN proposés 69
IV.3.4 Architecture du CNN la plus performante 72
IV.4 Implémentation hardware 79
IV.5 Conclusion 83
Conclusion générale……..85

Liste des Abréviations

API: Application Programing Interface.

CNN: Convolutional Neural Network.

CPU: Central Processor Unit.

ELISA: Enzyme Linked Immuno Sorbent Assay.

FC: Fully- Connected.

GPU: Graphic Processor Unit.

IA: Intelligence Artificielle.

MLP: Multi Layer Perception.

MNIST: Modified National Institute of Standards and Technology.

OpenCV: Open Computer Vision.

PCR: Polymerase Chain Reaction.

RBF: Radial Basic Fonction.

RBF: Radial Basis Fonction.

RNA : Réseau de Neurone Artificiel.

RTK: Real Time Kinematic.

TPU: Tensor Processing Unit.

VRT: Variable Rate technology.

Liste des figures

Chapitre Ⅰ :

Figure Ⅰ.1 Exemple d'une maladie de plante. 3
Figure Ⅰ.2 Explique l'un des dommages (oïdium) causés par le champignon. 4
Figure Ⅰ.3 Effet des dommages causés par le mildiou sur les plantes. 5
Figure Ⅰ.4 Phytoplasmes. 6
Figure Ⅰ.5 Bactéries sur une plante. 7
Figure Ⅰ.6 virus chez les plantes. 8
Figure Ⅰ.7 Nématode blanc de la pomme de terre. 9
Figure Ⅰ.8 Aperçu de l'agriculture traditionnelle. 11
Figure Ⅰ.9 Nouvelle révolution agricole. 12
Figure Ⅰ.10 Ombrières permettent de protéger les cultures des aléas climatiques. 14
Figure Ⅰ.11 Principe de fonctionnement d'un autoguidage Rtk. 15
Figure Ⅰ.12 Drone Agricole. 16
Figure Ⅰ.13 La robotique au service de l'agriculture. 17
Figure Ⅰ.14 Une application Smartphone au service du secteur agricole. 17
Figure Ⅰ.15 Capteur pour arbre fruitière. 18
Figure Ⅰ.16 Robot désherbeur agricole 19
Figure Ⅰ.17 Structure et éléments d'un biocapteur. 20
Figure Ⅰ.18 Meilleures techniques chromatographiques. 21

Chapitre II :

Figure II.1 Relation entre l'IA, Machine Learning et Deep Learning. 25
Figure II.2 Architecture d'un réseau de neurones artificiels avec deux entrées, deux couches cachées et une seule sortie. 27
Figure II.3 Un modèle de Neurone Formel. 29
Figure II.4 Un réseau neurone Hopfield Networks. 30
Figure II.5 Structure de réseau ADALINE 31
Figure II.6 Réseau de Kohonen. 32
Figure II.7 Schéma d'un RBF. 32
Figure II.8 Architecture Standard d'un Réseau de Neurone Convolutionnel. 34
Figure II.9 Processus de convolution et de sous-échantillonnage. 35
Figure II.10 Une couche du CNN en 3 dimensions. (Vert : volume d'entrée, bleu : volume du champ récepteur, gris : couche de CNN, cercles : neurones artificiels indépendants. 36
Figure II.11 Différentes couches de CNN. 37
Figure II.12 Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution (bleu) 38
Figure II.13 Déférence entre max pooling et average pooling. 38
Figure II.14 Schéma général de principe de fonctionnement d'un réseau de neurones convolutifs. 40
Figure II.15 Exemple de modèle de CNN. 41
Figure II.16 Affichage d'images avec bruit. 42
Figure II.17 Modèle. 42
Figure II.18 Entrainement du classifier. 43
Figure II.19 Affiche l'image reconstruite (après suppression du bruit)..... 44

Chapitre Ⅲ:

Figure Ⅲ.1 Différentes versions de python. 47

Figure Ⅲ.2 Etape 1..... 48
Figure Ⅲ.4 Etape 2..... 48
Figure Ⅲ.5 Etape 3..... 49
Figure Ⅲ.6 Etape 4..... 49
Figure Ⅲ.7 Etape 5..... 50
Figure Ⅲ.8 Etape 6..... 50
Figure Ⅲ.9 Etape 7..... 51
Figure Ⅲ.10 Etape 8..... 51
Figure Ⅲ.11 Interface d'Anaconda Navigator..... 52
Figure Ⅲ. 12 Présentation de l'interface du logiciel..... 54
Figure Ⅲ.13 Interface de Google Colab. 56
Figure Ⅲ.14 Composants standards d'un Raspberry pi. 58
Figure Ⅲ.15 Raspberry Pi 3..... 60
Figure Ⅲ.16 Camera OpenMV h7. 61

Chapitre Ⅳ:

Figure Ⅳ.1 Stricture générale du système proposé..... 64
Figure Ⅳ.2 Diagramme de détection des maladies..... 64
Figure Ⅳ.3 Classes des différentes images. 67
Figure Ⅳ.4 Echantillons de la base de données (PlantVillage)..... 68
Figure Ⅳ5 Architecture de production..... 68
Figure Ⅳ.6 Présentation du modèle de réseau de neurones convolutifs. 73
Figure Ⅳ.7 Compilation de modèle 3. 74
Figure Ⅳ.8 Précision et Erreur pour le modèle 75
Figure Ⅳ.9 Valeur de précision de modèle 3..... 75
Figure Ⅳ.10 Matrice de confusion..... 76
Figure Ⅳ.11 Exemple sur le premier cas. 77
Figure Ⅳ.12 Exemple photo prise d'internet..... 78
Figure Ⅳ.13 Réalisation hardware..... 79
Figure Ⅳ14 Résultat de test 1. 80
Figure Ⅳ.15 Résultat de test 2. 81
Figure Ⅳ.16 Résultat de test 3. 82

Liste des tableaux

Tableau III.1 Comparaison entre les modèles de RPI.…….... 60
Tableau Ⅳ.1 Caractéristiques de chaque Lap-Top.....66
Tableau Ⅳ.2 Description de la base de données. 67
Tableau Ⅳ.3 Influence du nombre de couche de convolution. 70
Tableau Ⅳ.4 Influence de la taille de filtre. 72
Tableau Ⅳ.5 Influence de nombre d'itérations. 73

Introduction générale

Avec une population mondiale qui devrait atteindre 9 milliards d'ici 2050, en développant des méthodes de détection et d'atténuation les maladies des plantes servent le double objectif d'augmenter la récolte rendement et réduire l'utilisation de pesticides.

La méthode traditionnelle de la détection de la maladie a été d'utiliser un examen manuel par agriculteurs ou experts, ce qui peut prendre beaucoup de temps. Et coûteux, s'avérant infaisable pour des millions de petites et moyennes exploitations à travers le monde.

La détection et la classification de la maladie du plantes par la vision artificielle est un thème de recherche qui a été abordé depuis de nombreuses années et a connu une répandre l'adoption par de nombreuses agences dans le monde pour améliorer leur application, capacités de création.

Pour atteindre cet objectif, nous avons développé une plateforme matérielle composée d'une carte Raspberry Pi, caméra et une plante. Ce mémoire est composé de quatre chapitres:

- **Le 1er chapitre** présente l'état de l'art qui donne une vision générale sur les maladies des plantes ainsi les techniques utilisées pour détecter les maladies des plantes.
- **Le 2ème chapitre** est consacré au deep learning, l'intelligence artificielle et une description plus détaillée sur les réseaux de neurones convolutifs (CNNs) qui est la méthode choisie dans notre projet.
- **Le 3ème chapitre** contient des outils hardware et software Raspberry Pi.3, la caméra et la plante. Ainsi que certaine outils de développement tel que les bibliothèques : OpenCV, Tensorflow et Keras et le langage de programmation Python3.8.
- **Le 4ème chapitre** contient la conception de notre application, ainsi que la méthode d'implémentation de notre travail, ainsi l'ensemble des choix techniques, (langage de programmation Python) utilisés pour la réalisation de cette application. Finalement on va présenter les résultats obtenus avec une discussion sur ces différents résultats.

Cette mémoire est clôturée par la conclusion générale et les perspectives de notre travail.

Chapitre I Etat de l’art

I.1 Introduction

Les maladies des plantes ont toujours été une préoccupation importante en agriculture car elles entraînent une réduction de la qualité des récoltes et par conséquent la production. Les effets des maladies des plantes vont de symptômes mineurs aux dégâts importants de superficies entières de cultures plantées, ce qui entraîne des coûts et des impacts financiers importants fortement sur l'économie agricole, en particulier dans les pays en développement qui dépendent d'une seul culture ou quelques cultures. Afin d'éviter des pertes importantes, diverses méthodes ont été développées pour diagnostiquer la maladie.

Dans ce chapitre, on va présenter tout d'abord les maladies des plantes, ensuite on va décrire le développement de secteur agricole, et les techniques utilisées pour détecter les maladies des plantes.

I.2 Maladies des plantes

Les maladies des plantes sont considérées comme l'un des principaux facteurs influençant la production alimentaire et minimisant les pertes de production, et il est essentiel que les maladies des cultures soient rapidement détectées et reconnues.[1]

Les plantes sont menacées par des maladies causées par des microorganismes : des virus, bactéries, des oomycètes, phytoplasmes, des nématodes et des champignons. Ces maladies causent d'importantes pertes de rendement, sur les cultures vivrières, fruitières, légumières et ornementales, en particulier dans les zones tropicale et tempérée chaude. Ce sont parfois des récoltes, voire des filières tout entières, qui sont anéanties. Les nouvelles maladies émergent régulièrement, à cause de mutations des agents pathogènes, ou encore de leur adaptation à des nouveaux environnements.

Les produits phytosanitaires étant inefficaces contre les bactéries et les virus, il faut trouver d'autres moyens de réduire les risques d'infection, en particulier en évitant d'introduire ces microorganismes. Ces organismes infectent les plantes en détournant les ressources de la plante hôte pour vivre et se reproduire à ses dépens.[2]

Pour que la maladie se manifeste, il doit y avoir quatre facteurs :

- Une plante susceptible d'être hôte.
- Un agent pathogène virulent.
- Des conditions environnement a les favorables à croissance de l'agent pathogène.
- Le temps pour que la maladie se développe.

Figure Ⅰ.1 présente un exemple de maladie des feuilles d'une plante.



Figure Ⅰ.1 Exemple d'une maladie de plante**.**

I.2.1 Champignons

Les champignons sont un groupe d'organismes qui ont certains points communs avec les plantes, mais qui, contrairement à celles-ci, sont dépourvus de chlorophylle, pour pouvoir vivre et se reproduire, ils doivent donc prélever leurs éléments nutritifs à partir d'autres organismes vivants ou morts. Certains champignons forment des filaments délicats (mycélium filamenteux) qui se développent sur ou dans les tissus des plantes infectées, et qui peuvent être invisibles à l'œil nu. La plupart d'entre eux se reproduisent en formant des millions de spores qui sont propagées par les courants atmosphériques, l'eau, la terre et les insectes. Dans des conditions ambiantes propices, et si un hôte approprié se trouve à proximité, les spores germent et

l'infectent. Ils pénètrent par les blessures et les ouvertures naturelles de la plante (stomates, lenticelles) ou en traversant la couche de tissus superficiels externes (épiderme). De nombreux champignons produisent également des spores de réserve qui peuvent résister à des conditions adverses durant de longues périodes.

Les champignons qui infectent le feuillage, les fleurs et les tiges sont généralement propagés par des courants atmosphériques ou par des éclaboussures d'eau. Ceux qui infectent les racines, le collet ou l'appareil vasculaire des plantes (flétrissures) peuvent vivre plusieurs années dans le sol, jusqu'à ce que les conditions redeviennent propices à la germination. Les dégâts qu'ils occasionnent sont la dégradation des tissus (pourriture molle), la dégénérescence cellulaire (nécrose ou taches foliaires) et l'obstruction des vaisseaux conducteurs. Les maladies à champignons sont contrées le plus efficacement en utilisant des variétés résistantes, en améliorant la circulation de l'air sous le couvert végétal de façon à réduire le taux d'humidité, en évitant l'aspersion sur frondaison, en utilisant des techniques d'assainissement, en améliorant le drainage et en appliquant des produits antiparasitaires.[2]

Les champignons parmi les plus communs qui causent des dommages aux cultures spéciales incluent :

- L'oïdium (blanc).
- Les taches foliaires et les brûlures.
- La rouille.
- La moisissure blanche.
- La pourriture du collet et des racines.
- La flétrissure.

Figure Ⅰ.2 Explique l'un des dommages (oïdium) causés par le champignon.



Figure Ⅰ.2 Oïdium de la tomate.

I.2.2 Oomycètes

Les oomycètes sont un groupe d'organismes analogues aux champignons, aussi appelés champignons aquatiques, qui causent nombre de maladies destructrices de plantes. Les symptômes de plusieurs maladies des oomycètes sont similaires à celles causées par les champignons, et ces organismes sont souvent classés avec les champignons pour des raisons de lutte antiparasitaire. Par contre, ces maladies peuvent se répandre très rapidement et sont plus difficiles à contrôler par les seules méthodes de culture. Pour ces raisons, les mesures préventives contre les maladies à oomycètes revêtent une importance considérable.[2]

Les maladies courantes à oomycètes chez les cultures spéciales incluent :

- Le mildiou (Figure Ⅰ.3).
- Brûlure phytophthoréenne.
- La fonte des semis.



Figure Ⅰ.3 Effet des dommages causés par le mildiou sur les plantes.

I.2.3 Phytoplasmes

Les phytoplasmes (Figure Ⅰ.4) sont des organismes spécialisés analogues aux bactéries qui causent plusieurs maladies d'importance économique pour les producteurs de cultures spéciales. La maladie à phytoplasmes la plus courante en Ontario est la jaunisse de l'aster. On retrouve néanmoins plusieurs autres maladies à phytoplasmes chez les cultures spéciales, incluant plusieurs types de balais de sorcières. Les maladies à phytoplasmes se caractérisent généralement par de la chlorose (jaunissement) des feuilles, du rabougrissement, et une prolifération des feuilles ou des branches qui forme un bouquet dense de tissu végétatif. Elles sont aussi la cause d'une production excessive de racines latérales chez les cultures racines telles

que les carottes. Tout comme pour les virus, la transmission de ce parasite nécessite un vecteur, en général des insectes comme les cicadelles. La jaunisse de l'aster est transmise par la cicadelle de l'aster, et le contrôle de cette maladie passe par un contrôle efficace de la cicadelle.[2]



Figure Ⅰ.4 Phytoplasmes.

I.2.4 Bactéries

Les bactéries (Figure Ⅰ.5) sont des micro-organismes unicellulaires qui, dans des conditions favorables, se reproduisent rapidement à l'intérieur de la plante. Elles pénètrent par les blessures, les coupures laissées par la taille ou par les pores des feuilles (stomates). Bien qu'elles puissent également être disséminées par les outils de jardinage, les mains et les vêtements, elles sont le plus souvent propagées par les éclaboussures de la pluie et de l'eau d'arrosage. Une fois à l'intérieur de la plante, elles s'attaquent à la plupart de ses organes.

On associe souvent les bactéries à des flétrissures, à des brûlures ou à des pourritures molles. Les bactéries des flétrissures infectent les racines, et finissent pas boucher les tissus vasculaires, ce qui gêne l'écoulement de l'eau et des éléments nutritifs. Les brûlures consistent en un pourrissement rapide des tissus succulents que sont les pousses, les feuilles et les fleurs. Les pourritures molles se développent sur les tissus charnus des parties blessées de la plante qui restent humides pendant de longues périodes.

Les symptômes de flétrissure se manifestent principalement sur les feuilles : grandes zones de nécrose en forme de V, jaunissement et flétrissement. Des bandes grises ou noires peuvent apparaître sur les feuilles et sur les tiges. Par temps chaud et humide, les plantes s'affaissent très rapidement. Il arrive qu'un exsudat ou que des gouttelettes laiteuses riches de millions de bactéries s'échappent des tiges et des feuilles infectées.

Au début, les brûlures peuvent prendre l'aspect de mouchetures noires sur le feuillage et les tiges, qui s'étendent rapidement. Chez de nombreuses cultures ligneuses, les boutons floraux,

les jeunes feuilles et les pousses peuvent noircir complètement et mourir. L'infection se produit au début du printemps, pendant les périodes d'humidité prolongée.

Les pourritures molles se développent souvent sur les fruits ou les légumes sur lesquels de l'eau stagnante séjourne pendant de longues périodes. Elles apparaissent habituellement là où une blessure existe déjà sur la plante, ou encore aux points de contact des tissus végétales avec le sol. Les pourritures molles bactériennes dégagent généralement une odeur désagréable et ont un aspect humide et visqueux [2].



Figure I.5 Bactéries sur une plante.

I.2.5 Virus

Les virus à base de protéines et d'acides nucléiques ne sont visibles qu'au microscope électronique, et ne peuvent se reproduire que dans les cellules d'un hôte vivant. Les maladies virales les plus communes sont classées en fonction des symptômes qu'elles produisent : mosaïques, chloroses, rabougrissement, marbrures et rayures. Les symptômes des maladies virales (Figure I.6) sont très divers, le plus commun étant un jaunissement inégal de la feuille, qui produit une mosaïque de taches jaunes et vertes. Dans le cas de certains virus systémiques, les parties adjacentes aux nervures virent au jaune. D'autres virus peuvent aussi causer l'apparition de taches annulaires, le rabougrissement, la déformation des feuilles et des fleurs, et la mort prématurée. De nombreux virus vont causer des taches nécrotiques ressemblant aux symptômes d'autres agents pathogènes. De nombreux virus sont propagés des plantes infectées aux plantes saines par certains insectes, tels que les aleurodes et les pucerons. Certains virus sont transmis par le biais des pièces buccales d'insectes suceurs, ne nécessitent pas de période d'incubation dans le corps de l'insecte, et ne sont transmis qu'une seule fois (virus non persistants), d'autres virus requièrent au contraire une période d'incubation chez l'insecte,

accompagnent le vecteur tout au long de sa vie, et peuvent se transmettre un grand nombre de fois (virus persistants). Certains virus peuvent également être transportés mécaniquement, ou par les personnes qui manipulent les plants, les coupent ou les taillent, ou même être transmis par contact des plantes avec un vêtement contaminé. Nombre de virus se conservent d'une saison à l'autre dans les tubercules, les bulbes, les cormes, les herbes et à l'occasion dans les semences.[2]



Figure I.6 virus chez les plantes.

I.2.6 Nématodes

Les nématodes (Figure I.7) sont des vers microscopiques qui s'attaquent aux racines des plantes pour s'en nourrir. Le feuillage peut aussi être touché, bien que rarement. Dans certains cas, les racines sont raccourcies avec des excroissances ou des nœuds enflés. (Par exemple avec le nématode cécidogène, attention de ne pas confondre les nodosités fixatrices d'azote des légumineuses avec les symptômes d'une nématodose). Certains nématodes parasites des plantes causent de petites lésions ressemblant à des égratignures, qui permettent à des champignons pathogènes de pénétrer et de développer des maladies comme la pourriture des racines ou la flétrissure. Souvent, de petites lésions sont également visibles. Au fur et à mesure que la maladie progresse, on assiste généralement à l'apparition du pourridié dû à l'invasion d'autres organismes. La plante dépérit graduellement et montre des symptômes de sécheresse, de stress, de carence nutritionnelle et de retard de croissance. La plupart des nématodes qui infestent les racines des plantes préfèrent les sols sablonneux, mais certains nématodes parasitaires tels que l'anguillule des tiges préfèrent les sols argileux denses. Les nématodes ont une mobilité limitée dans le sol, se déplaçant lentement en nageant dans l'eau interstitielle entre les particules du sol. Ils peuvent couvrir de plus longues distances en cas de déplacements du sol associés à l'érosion,

à la machinerie ou aux bottes. Ils peuvent aussi se propager par les éclaboussures et les mouvements d'eau de surface, ou par la plantation de matières végétales infestées.[2]



Figure I.7 Nématode blanc de la pomme de terre.

I.3 Identification des maladies des plantes

L'identification des maladies des plantes est la clé pour prévenir les pertes de rendement et de quantité du produit agricole, est très importante dans le domaine agricole de nos jours. Les études des maladies des plantes signifient les études des modèles visuellement observables observés sur la plante. Surveillance de la santé et maladie la détection sur les plantes est très critique pour une agriculture durable. Il est très difficile de surveiller manuellement les maladies des plantes. Cela demande une énorme quantité de travail, une expertise, connaissances dans le domaine végétal et nécessitent également un temps de traitement excessif. Par conséquent, l'image le traitement est utilisée pour la détection des maladies des plantes. Il a les caractéristiques suivantes [3]:

- La détection des maladies implique des étapes telles que le prétraitement d'images.
- La segmentation d'images.
- L'extraction de caractéristiques et la classification.

I.4 Développement de secteur agricole

L'être humain pratique l'agriculture depuis des milliers d'années. C'est un des plus vieux métiers au monde. Celle-ci a toutefois fait l'objet d'une modernisation depuis toutes ces années. En effet, avant on utilisait des outils rudimentaires pour cultiver le sol. De ce fait, le travail était difficile et demandait beaucoup de main-d'œuvre. De nos jours, des machines plus sophistiquées permettent de faire le travail plus facilement et plus rapidement tout ayant besoin de oins de main-d'œuvre et ce sont des avantages conséquents. Par conséquent, on peut voir

ces dernières années des machines incroyables qui deviennent de plus en plus indispensables pour faire de bonnes récoltes.[4]

I.4.1 Agriculture traditionnelle

I.4.1.1 Définition

L'agriculture traditionnelle se caractérise par une très faible technicité et une très faible utilisation de la technologie. Cela rend sa production à grande échelle pas trop productive. La production ici est uniquement destinée à la consommation de l'agriculteur et de ceux qui travaillent la terre. Des outils tels que la faucille, la houe ou la pelle sont souvent utilisés. Il est à noter qu'il est encore utilisé par environ 50 % de la population mondiale.

Elle est une activité très rudimentaire et sa production dépend de la plupart des capacités physiques de l'agriculteur et de ses ouvriers. Cela réduit à la fois les performances et l'optimisation des ressources pour de meilleurs produits. A toutes ces caractéristiques, il faut ajouter que la façon dont la terre est travaillée n'est pas celle qu'on fait actuellement. La manière dont la terre est travaillée correspond à des connaissances ou pratiques ancestrales et empiriques qui se communiquent de génération en génération.[5]

I.4.1.2 Caractéristiques de l'agriculture traditionnelle

L'agriculture traditionnelle est caractérisée par [6] :

- Agriculture extensive avec des connaissances et des outils indigènes.
- Outils indigènes comme la hache, la houe et le bâton.
- Le raisin de bétail aide à créer des terres en jachère.
- Absence de responsabilité envers l'environnement.
- Manqué par la production excédentaire.

I.4.1.3 Inconvénients de l'agriculture traditionnelle

Avec le temps, nous nous rendons compte progressivement que ce type d'agriculture finit par présenter plus d'inconvénients que d'avantages[7].

- Mauvaises conditions pour les animaux.
- Déforestation et altération de l'environnement naturel : L'abattage des arbres, les techniques de lutte contre la boue et le brûlage, ainsi que le défrichement des zones forestières pour faire place à l'agriculture ont entraîné une déforestation massive et une érosion des sols. Les habitats naturels, la

diversité biologique et les animaux sauvages ont été fortement touchés. L'agriculture intensive est aussi impliquée dans la production des gaz à effet de serre. Ce sont l'usage d'engrais, les serres chauffées et la mécanisation importante qui en sont à l'origine.

- L'usage intensif de produits agrochimiques : Des pesticides chimiques, des engrais, des herbicides, des insecticides, etc sont à la base de ce type d'agriculture. Les insecticides et les pesticides tuent également les insectes utiles qui contribuent à la perte de biodiversité. L'utilisation d'engrais et d'herbicides chimiques contamine la nappe phréatique, les habitats de la faune et de la flore, ainsi que les masses d'eau comme les estuaires, les rivières, les lacs et même les mers. Les engrais, en particulier, sont la principale cause d'eutrophisation dans la plupart des masses d'eau du monde telles que les mers, les lacs et les rivières.
- La qualité des produits alimentaires : L'agriculture intensive étant principalement axée sur la production de masse de produits alimentaires, les stratégies de production négligent le besoin de produits alimentaires de qualité et nutritifs.

Figure Ⅰ.9 présente les outils de l'agriculture traditionnelle.



Figure Ⅰ.8 Aperçu de l'agriculture traditionnelle.

I.4.2 Agriculture de précision

I.4.2.1 Définition

L'agriculture de précision fait référence à une série de stratégies et d'outils qui permettent d'optimiser et d'augmenter la qualité et la productivité des sols par une série d'interventions ciblées, un résultat qui peut être obtenu grâce à des technologies toujours plus avancées.

Elle est dite (de précision) parce que, grâce aux outils les plus modernes (Figure Ⅰ.10), il est possible d'effectuer la bonne intervention, au bon endroit, au bon moment, en répondant aux besoins spécifiques de chaque culture et de chaque zone de champ, avec un haut niveau de précision.

Les technologies sont utilisées tout d'abord pour collecter des données et des informations qui servent à prendre des décisions sur la façon d'améliorer la production et ensuite pour mettre en œuvre les corrections nécessaires pour atteindre cet objectif.[8]



Figure Ⅰ.9 Nouvelle révolution agricole.

I.4.2.2 Principes de l'agriculture de précision

Le principe de l’agriculture de précision, c’est d’utiliser les nouvelles technologies pour augmenter les rendements d’une parcelle tout en réduisant la consommation d’énergie et d’intrants. L’idée est ainsi de « produire plus avec moins », ou au moins de produire autant, mais de manière plus durable.

L'agriculture de précision utilise ainsi des moyens d'observation (satellites, drones, capteurs connectés, traitement de données en temps réel, intelligence artificielle, applications mobiles, etc.) associés à des outils d'aide à la décision (accessibles sous forme d'applications Web et Mobile) afin de collecter un maximum de données intéressantes. Ces données servent à améliorer le quotidien des agriculteurs. Ils ont également une vocation environnementale forte car ils permettent d'intervenir uniquement lorsque c'est nécessaire. Par exemple, on peut ainsi adapter l'utilisation de produits phytosanitaires en fonction des stricts besoins des cultures, et de traiter les parcelles au mètre carré près.

De cette manière, l'agriculture de précision va se rapprocher de l'agroécologie pratique dont elle est complémentaire et qui devrait permettre d'accélérer la transition de notre modèle agricole conventionnel vers un modèle agricole durable [9].

I.4.2.3 Enjeux de l'agriculture de précision

On dénote trois enjeux majeurs à l'agriculture de précision [9]:

1. Un enjeu agronomique : Cette méthode permet aux exploitants de mieux s'adapter aux contraintes météo et aux types de sols de leurs parcelles. Par exemple Ombrea a développé des outils de régulation intelligents pour protéger les plantes des effets du réchauffement climatique. (Figure I.11)

2. Un enjeu environnemental : cette pratique puisqu'elle vise à diminuer l'empreinte écologique de l'activité agricole. Elle favorise une meilleure gestion de l'eau, des intrants et des machines. Il est ainsi possible de n'arroser par exemple que certaines zones d'une parcelle.

3. Un enjeu économique : qui en découle naturellement. L'agriculture de précision doit ainsi permettre de produire plus avec moins. Cependant, l'utilisation de la technologie pour améliorer le travail d'un agriculteur ne signifie pas que son expérience et son savoir-faire n'ont plus de valeur. Ce sont des outils d'aide à la décision et c'est à chaque exploitant d'en tenir compte pour exercer son métier. Les algorithmes et modèles vont créer des préconisations. Mais ces conseils ne garantissent pas à eux seuls la bonne tenue d'une exploitation.



Figure Ⅰ.10 Ombrières permettent de protéger les cultures des aléas climatiques.

I.4.2.4 Technologies de l’information pour une agriculture plus précise

L’agriculture de précision vise à rendre la pratique de l’agriculture plus précise et contrôlée en matière de culture et d’élevage. Un élément clé de cette approche de gestion agricole optimisée est l’utilisation des technologies de l’information et d’un large éventail d’outils. [10]

- VRT (technologie à taux variable) **:** la VRT fait référence à toute technologie capable de contrôler la quantité d’intrants appliquée à un endroit spécifique.
- Échantillonnage GPS : le test du sol d’un champ révèle les nutriments disponibles, le niveau de pH et une variété d’autres données qui sont importantes pour prendre des décisions éclairées et rentables pour optimiser l’ensemencement et les engrais.
- Technologie de détection **:** outil inestimable pour surveiller et gérer des terres, l’eau et d’autres ressources. Elle peut aider à tout déterminer, des facteurs qui peuvent stresser une culture à un moment précis à l’estimation de la quantité d’humidité dans le sol.
- L’autoguidage par satellite (RTK) **:** permet de corriger le signal envoyé à un tracteur autoguidé ou un robot autonome et d’améliorer sa précision jusqu’au centimètre, afin de limiter les manques et les recouvrements de traitement ou de fertilisation (Figure Ⅰ.12).

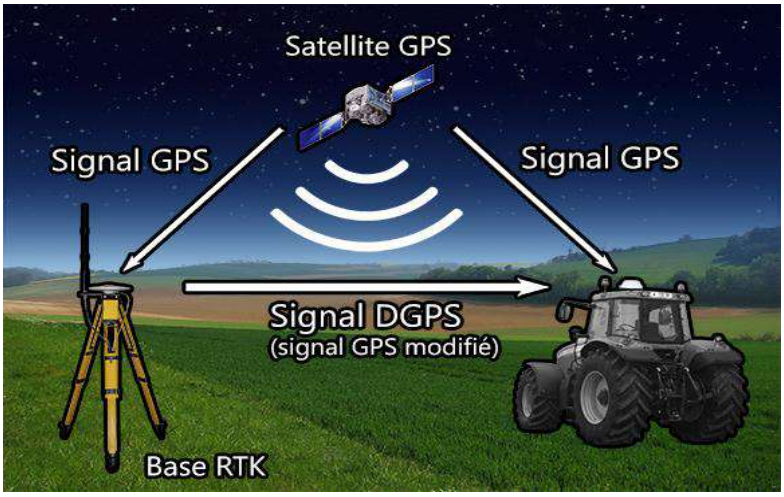


Figure I.11 Principe de fonctionnement d'un autoguidage Rtk.

I.4.2.5 Avantages de l'agriculture de précision

Produire plus avec moins de ressources tout en maintenant des standards de qualité élevés : tel est l'objectif final de l'agriculture de précision. Ce type d'approche présente un double avantage [8]:

- pour les exploitations agricoles, qui peuvent optimiser les efforts et les ressources, réduire la consommation et le gaspillage, et augmenter la productivité des sols. Pour les agriculteurs et les entrepreneurs, le travail devient plus rentable également parce que les coûts horaires sont réduits, grâce à une gestion plus rapide et plus efficace. Enfin, les performances augmentent tout en limitant la fatigue des opérateurs et l'impact
- sur l'environnement, grâce à la réduction des déchets d'engrais et d'herbicides, des émissions et du compactage des sols par une utilisation plus rationnelle des ressources.

I.5 Techniques utilisées pour détecter les maladies des plantes

I.5.1 Drones

Le drone (Figure I.12) est certainement l'une des innovations les plus utiles dans l'agriculture. Il permet aux agriculteurs d'adopter des stratégies de fertilisation précises via un logiciel de planification de vol avec évaluation des conditions des cultures, réduisant ainsi la dépendance aux engrais et augmentant les rendements.

Les drones sont d'excellents outils multifonctionnels favorisant une agriculture de précision. Embarquant plusieurs capteurs selon les mesures à effectuer (estimation de biomasse,

taux de chlorophylle, stress hydrique, imagerie…), les drones permettent de nombreuses applications, comme par exemple [10]:

- L'optimisation de l'apport d'intrants.
- La détection des dégâts de gibier.
- La détection précoce des maladies.
- La détection des adventices.
- L'épandage précis de biocides.



Figure I.12 Drone Agricole.

I.5.2 Agriculture robotisée

Le Français moins de pénibilité, gain de temps, respects des sols et meilleure rentabilité. Ainsi, le robot Oz permet de désherber mécaniquement les cultures maraîchères de moins de 10 hectares, tandis que Dino est un robot enjambeur de désherbage destiné aux plus grandes exploitations. Ted est quant à lui un robot enjambeur spécialement dédié aux cultures viticoles. Tous ntègrent la technologie RTK.
Le projet européen RHEA a mis en place une flotte de tracteurs autonomes et de drones équipés de capteurs pour distinguer les mauvaises herbes des cultures et appliquer des herbicides si nécessaire. Cette combinaison permet d'économiser jusqu'à 75 % de l'herbicide, tout en tuant 90% des mauvaises herbes sur une parcelle.
La startup Bosch Deepfield Robotics a quant à elle conçu des robots qui parcourent les rangées de plantes, les détectent puis envoient les données aux agriculteurs pour les aider à optimiser la sélection des semences. Au besoin, le robot peut enfoncer les mauvaises herbes dans le sol (Figure I.13).[10]



Figure I.13 La robotique au service de l'agriculture.

I.5.3 Applications pour Smartphone

En collaboration avec la MSA, la Chambre d’agriculture du Gard a conçu une application de désherbage localisé en viticulture et arboriculture pour smartphone et tablette. Epand’App est une application d’aide aux réglages des épandeurs de fumier et de tonnes à lisier afin d’appliquer la dose appropriée, lorsque l’agriculteur ne dispose pas de machines d’épandage dotées d’automatismes. Cette application a été développée par les Chambres d’agriculture de Bretagne.[10]

Figure I.14 illustre Une application Smartphone au service du secteur agricole.



Figure I.14 Une application Smartphone au service du secteur agricole.

I.5.4 Capteurs connectés

Divers types de capteurs ont été développés ou sont en cours de développement pour répondre à un grand nombre de problématiques agricoles. [11]

- Capteur pour plantes et arbres fruitiers : Ce capteur s'attache directement à la plante ou au fruit, pour recueillir différents signaux comme son taux de croissance ou son hydratation. Combinées avec d'autres capteurs fournissant les éléments du climat ou l'humidité du sol, ces informations sont analysées et restituées pour fournir un diagnostic sur l'état de la plante et des recommandations (Figure I.15).



Figure I.15 Capteur pour arbre fruitière.

- Capteurs pour serre ou cultures hydroponiques : Ce système développé en open-source comprend un ensemble de capteurs enregistrant de nombreuses données : luminosité, humidité de l'air et du sol, température de l'eau, PH… qui sont traitées sur des serveurs en cloud. Un monitoring à distance et un système d'alerte par SMS permettent de gérer la serre et de décider d'actions correctrices éventuelles.
- Robot désherbeur de vignes : Ce robot spécialisé pour la tonte est alimenté à l'énergie solaire. La définition des zones à traiter se fait via les coordonnées GPS de la parcelle, avec la possibilité de spécifier des endroits à éviter. Ecologique, il permet de réduire l'usage des désherbants et se pilote simplement avec une application par smartphone (Figure I.16).
- Capteur pour drone agricole : Les drones sont logiquement bien adaptés pour les surfaces agricoles, leur utilité dépendant des fonctionnalités des capteurs utilisés.

Proposé par l'un des leaders du drone agricole, Agrosensor est un capteur multi spectral pour la cartographie des cultures. Il peut s'installer sur plusieurs types de drones.

- Système d'irrigation intelligent : Agissant sur des solénoides et valves apposées sur le système d'irrigation, ce dispositif connecté à Internet ajuste les plannings d'irrigation du terrain en fonction des données de prévisions météo locales (pluie, température). Une base de 48000 stations météo est exploitable.



Figure I.16 Robot désherbeur agricole.

I.5.5 Biocapteurs

Parmi les outils déjà utilisés, les biocapteurs sont des technologies permettant de convertir des éléments physiques ou biologiques (anticorps, protéines ou ADN) en signaux mesurables.

Ceux destinés à la détection de protéines (anticorps) ou d'acide nucléique (ADN ou ARN) reposent principalement sur des techniques ELISA et PCR. La technique ELISA (Enzyme Linked Immuno Sorbent Assay) permet de détecter le phytopathogène à partir d'extraits broyés de plante malade en utilisant des anticorps dirigés contre ce pathogène. La technique de « PCR » (pour Polymerase Chain Reaction) consiste à amplifier, pour la quantifier plus facilement, une partie du génome du pathogène, à partir d'un extrait d'ADN du végétal malade.

Mais ces outils de diagnostic nécessitent d'avoir déjà identifié le pathogène et de disposer des anticorps qui permettront de le reconnaître, ou les amorces PCR qui permettent d'amplifier une

partie de sa séquence ADN. De plus, ces techniques sont destructives puisque pour libérer le pathogène ou l'acide nucléique de celui-ci, il faut broyer un organe de la plante. Elles nécessitent également un matériel de laboratoire coûteux et de la main-d'œuvre.

Si ces analyses se font plutôt en laboratoire, il existe aussi des kits de détection commercialisés, utilisant ces marqueurs biologiques pour une utilisation au champ, notamment pour détecter des espèces de Phytophthora, ou mildiou, d'Erwinia amylovora, c'est-à-dire le feu bactérien, qui se révèle par des brûlures du feuillage, et du virus Y de la pomme de terre (Figure Ⅰ.17).[12]

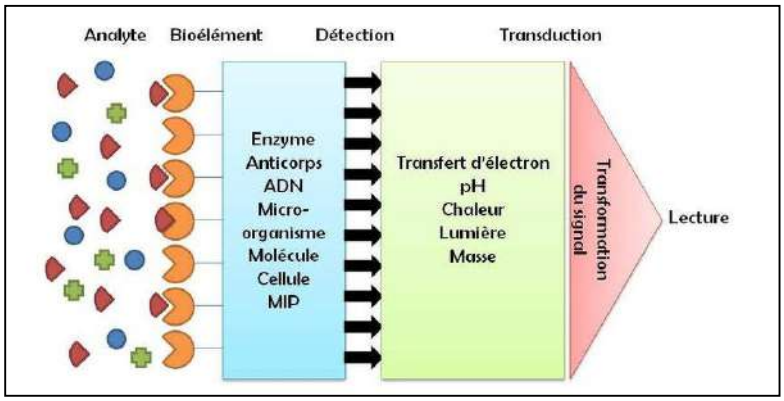


Figure Ⅰ.17 Structure et éléments d'un biocapteur.

I.5.6 Sentir les plantes malades

Le premier exemple de technologie en développement est la chromatographie en phase gazeuse (CPG), qui permet d'analyser des gaz, ou des mélanges de gaz, pour les différencier les uns des autres. Certaines maladies des plantes sécrètent des gaz caractéristiques, appelés composés organiques volatils, détectables par cette technique. (Figure Ⅰ.18)

Par exemple, l'un des principaux problèmes concernant la santé des pommes de terre est le pourrissement. Celui-ci est causé, le plus souvent, par deux agents bactériens : Ralstonia solanacearum et Clavibacter michiganensis subsp sepedonicus, classés dans la liste des pathogènes de quarantaine en Europe. Avant le pourrissement du tubercule de pomme de terre, elle dégage des composés organiques volatils, qui peuvent être captés grâce à des (nez électroniques). Ces techniques permettent de détecter ces pathogènes avant même l'observation des symptômes de maladie.[12]



Figure Ⅰ.18 Meilleures techniques chromatographiques.

I.5.7 Spectro-imagerie

La spectro-imagerie est une autre technologie permettant de détecter une maladie, et parfois même avant que les symptômes ne soient visibles à l’œil nu. Il existe plusieurs types de spectro-imagerie, mais la plus prometteuse, car la plus précise est certainement la technique d’imagerie hyper spectrale. L’idée est de prendre des photos des parties aériennes des plantes dans des bandes spectrales allant du visible au moyen infrarouge (donc certaines que l’on ne voit pas à l’œil nu).

Une plante stressée est différente d’une plante saine, car ses composants biochimiques et structuraux ont changé. Ceux-ci émettent des rayonnements spécifiques, différents de ceux d’une plante saine et qui peuvent être différenciés par l’imagerie hyper spectrale.

En pathologie végétale, on peut alors identifier des maladies de plantes sur la simple base des données spectrales. Une réelle avancée dans le diagnostic. Un système hyper spectral est composé principalement d’une caméra et peut s’installer à bord d’appareils mobiles tels que les drones, ce qui permet d’analyser toutes les plantes d’une parcelle sans les détruire. Un avantage certain pour examiner à la loupe une parcelle agricole par rapport à des prélèvements aléatoires.[12]

I.6 Conclusion

Ce chapitre a été pour nous l'occasion de présenter les déférentes maladies des plantes. De plus, nous avons abordé les développements de secteur agricole (agriculture traditionnel et agriculture moderne). Après on a passé à citer les techniques utilisées pour détecter les maladies des plantes. A l'issue de cette étude, nous avons choisi le Deep Learning comme approche pour détecter les maladies. Dans le chapitre suivant, nous allons présenter les généralités de l'intelligence artificielle, l'apprentissage profond (Deep Learning), les réseaux de neurones et les réseaux de neurones convolutionnels (CNN).

Chapitre II

Généralité sur l'AI et CNN

II.1 Introduction

L'Intelligence Artificielle (IA) comprend des méthodes, des outils et des systèmes pour résoudre des problèmes qui requièrent normalement l'intelligence humaine. Le terme intelligence est toujours défini comme la capacité d'apprendre efficacement, de réagir de façon adaptative, de prendre les bonnes décisions, de communiquer en langage ou en images d'une manière sophistiquée, et de comprendre correctement. L'apprentissage profond (Deep Learning) est une cause de conduire les récentes lancées et de progresser l'Intelligence Artificielle.

Dans ce chapitre, nous allons présenter tout d'abord une généralité sur les réseaux neurones artificiels, ensuite nous allons décrire en détails le réseau CNN.

II.2 Intelligence Artificielle (IA)

II.2.1 Définition

C'est une discipline scientifique relative aux traitements des connaissances et au raisonnement dans le but de permettre à une machine d'exécuter des fonctions normalement associées à l'intelligence humaine. L'intelligence artificielle tente de reproduire les processus cognitifs humains dans le but de réaliser des actions (intelligentes). Le scientifique Marvin Lee Minsky, pionnier de la discipline, défini l'intelligence artificielle comme la construction de programmes informatiques qui s'adonnent à des tâches qui sont pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique.[13]

L'IA fait référence aux systèmes ou aux machines qui imitent l'intelligence humaine pour accomplir des tâches et qui peuvent s'améliorer itérativement en fonction des renseignements qu'ils recueillent. L'IA revêt un certain nombre de formes. Voici quelques exemples :

- Les robots conversationnels utilisent l'IA pour comprendre les problèmes de client plus rapidement et fournir des réponses de façon plus efficace.
- Les assistants intelligents utilisent l'IA pour analyser les informations critiques de grands ensembles de données en forme libre pour améliorer la planification.
- Les moteurs de recommandation peuvent fournir des recommandations automatisées pour des émissions de télévision en fonction des habitudes d'écoute des utilisateurs.

L'IA porte bien plus sur le processus et la capacité de pensée et d'analyse de données super puissantes que sur tout format ou toute fonction particulière. Bien que l'IA évoque des images de robots humanoïdes très fonctionnels prêts à conquérir le monde, elle ne vise pas à remplacer les humains. Elle est conçue pour améliorer considérablement les capacités et les contributions humaines, ce qui en fait un actif opérationnel très précieux.[14]

On peut identifier plusieurs manières d'expliquer l'IA, nous allons en développer trois :

1. L'Intelligence Artificielle est un domaine de l'informatique qui regroupe plusieurs briques technologiques très différentes les unes des autres, dont des algorithmes.
2. L'Intelligence Artificielle est une interface qui s'adapte à l'homme et permet une communication facile et intuitive avec la machine, une technologie qui fait disparaître la technologie.
3. L'Intelligence Artificielle fait partie des sciences cognitives. Ainsi, les offres liées à l'IA de Microsoft et IBM intègrent des (services cognitifs) avec des systèmes apprenants de différents niveaux qui visent à apprendre aux ordinateurs à voir, entendre, lire, mémoriser, structurer, raisonner, décider et agir.

II.2.2. Fonctionnement de l'Intelligence Artificielle

Selon Harry Shum, Président Exécutif de Microsoft, l'IA fonctionne seulement s'il y a présence (d'une vaste quantité de data ; d'une puissance informatique extraordinaire, notamment grâce au cloud, et des algorithmes révolutionnaires, basés sur le deep learning).

L'IA s'applique aujourd'hui dans des domaines variés tels que :

- Les jeux de réflexion.
- La recherche mathématique.
- La finance.
- La médecine.
- Les assistants personnels et la domotique.
- La reconnaissance faciale et la compréhension des langues.
- La robotique.

II.3 Apprentissage Profond (Deep Learning)

L'Apprentissage Profond (Deep Learning) est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années. Dans l'apprentissage automatique, un programme analyse un ensemble de données afin

de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données. L'Apprentissage Profond est basé sur ce qui a été appelé, par analogie, des réseaux de neurones artificiels, composés de milliers d'unités (Les Neurones) qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de neurones servent d'entrée aux calculs d'une deuxième couche et ainsi de suite. Par exemple, pour la reconnaissance visuelle, des premières couches d'unités identifient des lignes, des courbes, des angles, etc. Des couches supérieures identifient des formes, des combinaisons de formes, des objets, des contextes, etc. Les progrès de l'Apprentissage Profond ont été possibles notamment grâce à l'augmentation de la puissance des ordinateurs et au développement de grandes bases de données (Big Data).[15] Figure II.1 illustre la relation entre l'IA, Machine Learning et Deep Learning.

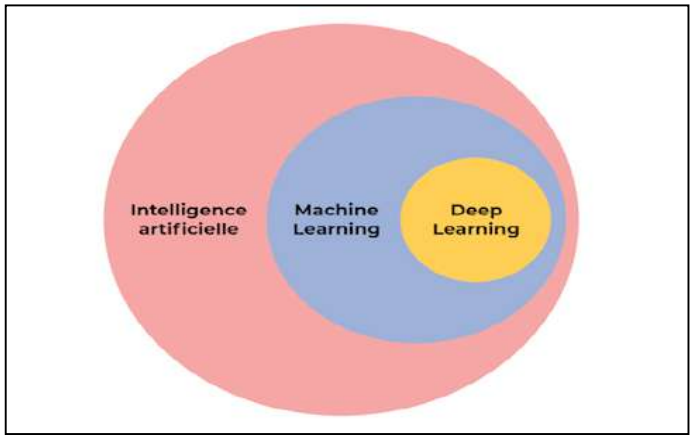


Figure II.1 Relation entre l'IA, Machine Learning et Deep Learning.

II.4 Différentes Type d`Apprentissage

De nombreuses méthodes classiques ont été consacrés, elles peuvent être séparées en deux grandes catégories : les méthodes de classification supervisée et les méthodes de classification non supervisée.

II.4.1 Apprentissage Supervisées

L'objectif de la classification supervisée est principalement de définir des règles permettant de classer des objets dans des classes à partir de variables qualitatives ou quantitatives caractérisant ces objets. On dispose au départ d'un échantillon dit d'apprentissage dont le classement est connu. Cet échantillon est utilisé pour l'apprentissage des règles de classement. Il est nécessaire d'étudier la fiabilité de ces règles pour les comparer et les appliquer, évaluer les cas de sous apprentissage ou de sur apprentissage (complexité du modèle). On utilise souvent un deuxième échantillon indépendant, dit de validation ou de test.[16]

II.4.2 Apprentissage Non Supervisées

Procède de la façon contraire. C'est à dire ne nécessitent aucun apprentissage et aucune tâche préalable d'étiquetage manuel. Elle consiste à représenter un nuage des points d'un espace quelconque en un ensemble de groupes appelé Cluster. Il lié généralement au domaine de l'analyse des données comme l'ACP. Un Cluster est une collection d'objets qui sont similaires entre eux et qui sont dissemblables par rapport aux objets appartenant à d'autres groupes. [16]

II.5 Réseau de Neurones Artificielles (RNA)

Un Réseau de Neurones Artificiels ou Neural Network est un système informatique basé sur le fonctionnement du cerveau humain pour apprendre, dont sont équipés les ordinateurs dotés de fonctions d'Intelligence Artificielle.

Les Réseaux de Neurones Artificiels permettent aux ordinateurs de résoudre des problèmes de façon autonome et renforcent leurs capacités d'une manière générale. Certains nécessitent une supervision initiale, en fonction de la méthode d'Intelligence Artificielle utilisée.

En 2019, l'Intelligence Artificielle est sous le feu des projecteurs. Cette technologie se développe à grande vitesse, et ses cas d'usage se multiplient dans tous les secteurs. Selon les experts, l'IA est amenée à bouleverser tous les aspects de notre société dans les années à venir.

Dans ce contexte, il semble important de comprendre comment fonctionne réellement l'Intelligence Artificielle. C'est pourquoi nous vous proposons aujourd'hui d'en apprendre davantage sur les Réseaux de Neurones Artificiels sur lesquels reposent de nombreuses IA.[17]

II.6 Historique des Réseaux de Neurones Artificiels

Le concept des Réseaux de Neurones Artificiels fut inventé en 1943 par deux chercheurs de l'Université de Chicago : le neuro physicien Warren McCullough, et le mathématicien Walter Pitts. Dans un article publié dans le journal Brain Theory, les deux chercheurs présentent leur théorie selon laquelle l'activation de neurones est l'unité de base de l'activité cérébrale.
En 1957, le Perceptron fut inventé. Il s'agit du plus ancien algorithme de Machine Learning, conçu pour effectuer des tâches de reconnaissance de patterns complexes. C'est cet algorithme qui permettra plus tard aux machines d'apprendre à reconnaître des objets sur des images.

Malheureusement, à l'époque, les Réseaux de Neurones étaient limités par les ressources techniques. Par exemple, les ordinateurs n'étaient pas assez puissants pour traiter les données nécessaires au fonctionnement des Réseaux de Neurones. C'est la raison pour laquelle la

recherche dans le domaine des Neural Networks est restée en sommeil durant de longues années.

Il aura fallu attendre le début des années 2010, avec l'essor du Big Data et du traitement massivement parallèle, pour que les Data Scientists disposent des données et de la puissance de calcul nécessaires pour exécuter des Réseaux de Neurones complexes. En 2012, lors d'une compétition organisée par ImageNet, un Neural Network est parvenu pour la première fois à surpasser un humain dans la reconnaissance d'image.

C'est la raison pour laquell**e** cette technologie est de nouveau au coeur des préoccupations des scientifiques. A présent, les Réseaux de Neurones Artificiels ne cessent de s'améliorer et d'évoluer de jour en jour.[17]

II.7 Fonctionnement des Réseaux de Neurones Artificiels

La conception des Réseaux de Neurones Artificiels s'appuie sur la structure des neurones biologiques du cerveau humain.

Les Réseaux de Neurones Artificiels peuvent être décrits comme des systèmes composés d'au moins deux couches de neurones - une couche d'entrée et une couche de sortie - et comprenant généralement des couches intermédiaires. Plus le problème à résoudre est complexe, plus les Réseaux de Neurones Artificiels ne doit comporter de couches. Chaque couche contient un grand nombre de Neurones Artificiels spécialisés.[18]

Figure II.2 montre l'architecture d'un réseau de neurones artificiels avec deux entrées, deux couches cachées et une seule sortie.

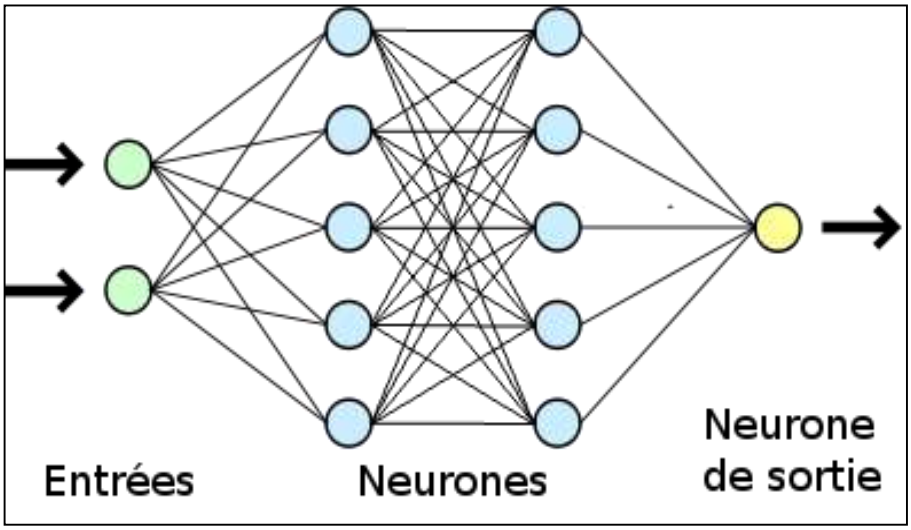


Figure II.2 Architecture d'un réseau de neurones artificiels avec deux entrées, deux couches cachées et une seule sortie.

II.8 Avantages et Inconvénients du Réseau de Neurone

II.8.1 Avantages

- Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe.
- Les Réseaux de Neurones sont capables de tolérer des données bruitées, ainsi que capable de classer les modèles sur lesquels ils ne sont pas été entrainés.
- Ils peuvent être utilisés lorsque nous avons peu d'informations sur la relation entre les attributs et les classes.
- Bien adapté pour les entrées et les sorties d'une valeur continue.
- Le succès sur plusieurs applications du monde réel comme la reconnaissance manuscrite de caractère, de la pathologie et de médecine de laboratoire, et beaucoup plus.
- Comportement moins mauvais en cas de faible quantité de données.
- Pour l'utilisateur novice, l'idée d'apprentissage est plus simple à comprendre que les complexités des statistiques multi variables.[19]

II.8.2 Inconvénients

Parmi ses inconvénients on cite [19] :

- Nécessitent un long temps d'apprentissage, donc ils sont adaptable aux applications dont ceci est faisable.
- Pauvre interprétation de la connaissance représentée sous la forme d'un réseau d'unités connectées par des liens pondérés.
- Exigent un nombre de paramètres qui doivent être déterminées empiriquement, par exemple la topologie du réseau et sa structure, le nombre de couches cachées, le nombre d'unités dans chaque couche cachée et dans la couche de sortie.

II.9 Présentation des Types de Réseaux Neurones

Il y a plusieurs types de Réseaux Neurones, chaque type étant développé pour un objectif particulier :

II.9.1 Neurone Formel

Un Neurone Formel (Figure II.3) est une représentation mathématique et informatique d'un neurone biologique. Le Neurone Formel possède généralement plusieurs entrées et une sortie qui correspondent respectivement aux dendrites et au cône d'émergence du neurone biologique (point de départ de l'axone). Les actions excitatrices et inhibitrices des synapses sont représentées, la plupart du temps, par des coefficients numériques (les poids synaptiques) associés aux entrées. Les valeurs numériques de ces coefficients sont ajustées dans une phase d'apprentissage. Dans sa version la plus simple, un Neurone Formel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation, généralement non linéaire. La valeur finale obtenue est la sortie du neurone.

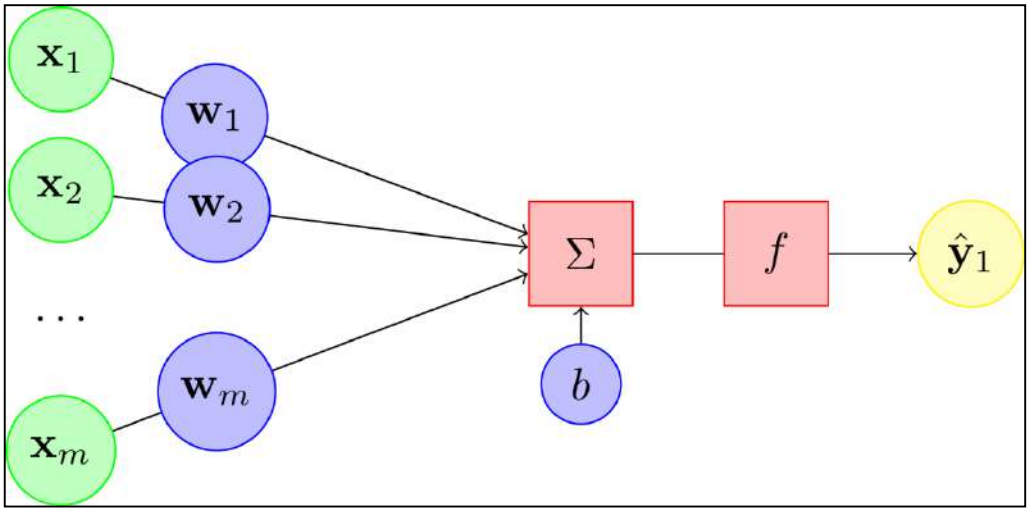


Figure II.3 Un modèle de Neurone Formel.

II.9.2 Réseaux de Hopfield

Le Réseau de Neurones d'Hopfield (Figure II.4) est un modèle de réseau de neurones récurrents à temps discret dont la matrice des connexions est symétrique et nulle sur la diagonale et où la dynamique est asynchrone (un seul neurone est mis à jour à chaque unité de

temps). Il a été découvert par le physicien John Hopfield en 1982. Sa découverte a permis de relancer l'intérêt dans les Réseaux de Neurones qui s'était essoufflé durant les années 1970 à la suite d'un article de Marvin Minsky et Seymour Papert.[15]

Il s'agit d'un réseau constitué de neurones à deux états (-1 et 1, ou 0 et 1), dont la loi d'apprentissage est la règle de Hebb (1949), qui veut qu'une synapse améliore son activité si et seulement si l'activité de ses deux neurones est corrélée (c'est-à-dire que le poids d'une connexion entre deux neurones augmente quand les deux neurones sont activés au même temps).

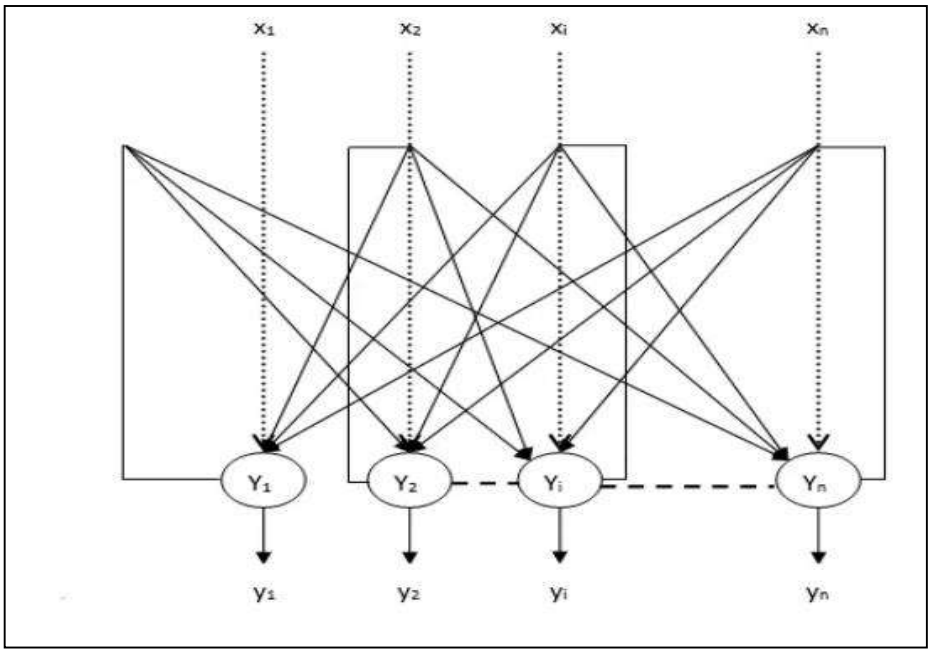


Figure II.4 Un réseau neurone Hopfield Networks.

II.9.3 ADALINE

L'ADAptive LINear Element (Figure II.5) a la même architecture que le perceptron, mais en revanche, il possède des valeurs d'activations continues, une fonction d'activation linéaire est une règle d'apprentissage un peu développée que celle du perceptron, elle s'appelle la règle de WIDROW & HOFF, en effet cette loi minimise l'erreur quadrique, en utilisant la méthode du gradient.

Contrairement à la règle du perceptron cette dernière opère sur les exemples bien classés et mal classés en optimisant toujours les coefficients des hyperplans séparateurs. [20]

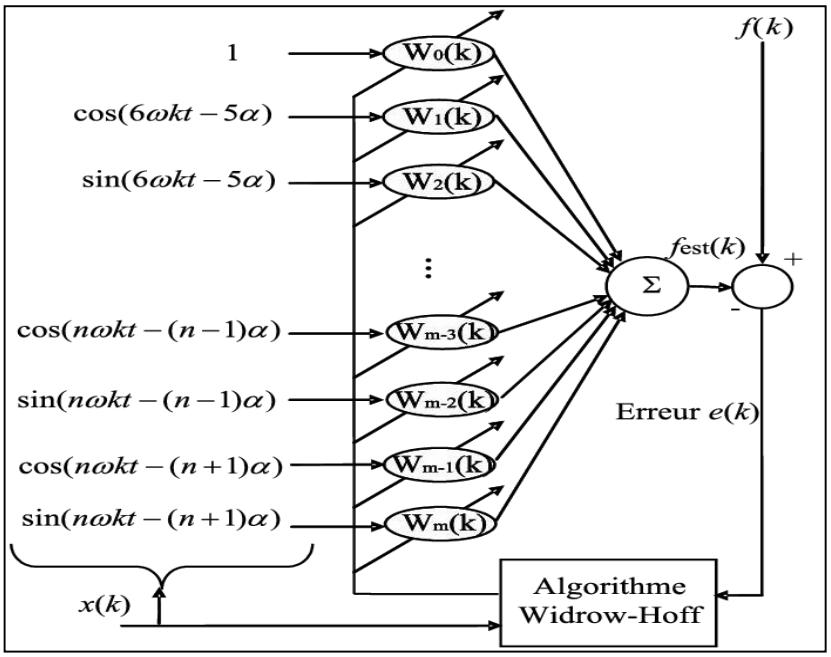


Figure II.5 Structure de réseau ADALINE

II.9.4 Réseaux de Kohonen

Contrairement aux réseaux de Hopfield où les neurones sont modélisés de la façon la plus simple possible, on recherche ici un modèle de neurone plus proche de la réalité. Ces réseaux sont inspirés des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères. Une loi de Hebb modifiée (tenant compte de l'oubli) est utilisée pour l'apprentissage. La connexion est renforcée dans le cas où les neurones reliés ont une activité simultanée et diminuée dans le cas contraire (alors qu'il ne se passait précédemment rien dans ce cas).Tous ces réseaux ont des applications dans la classification, le traitement d'image, l'aide à la décision et l'optimisation. [18]

Un Réseau de Kohonen est constitué d´une couche d´entrée de N neurones connectés aux M neurones d´une couche de sortie elle-même interconnectée. Soit WjT = {wi, j} le vecteur des poids des N connexions reliant la couche d´entrée au neurone j de la couche de sortie. Soit X un vecteur d´entrées de N composantes.

Figure II.6 illustre un réseau de Kohonen.

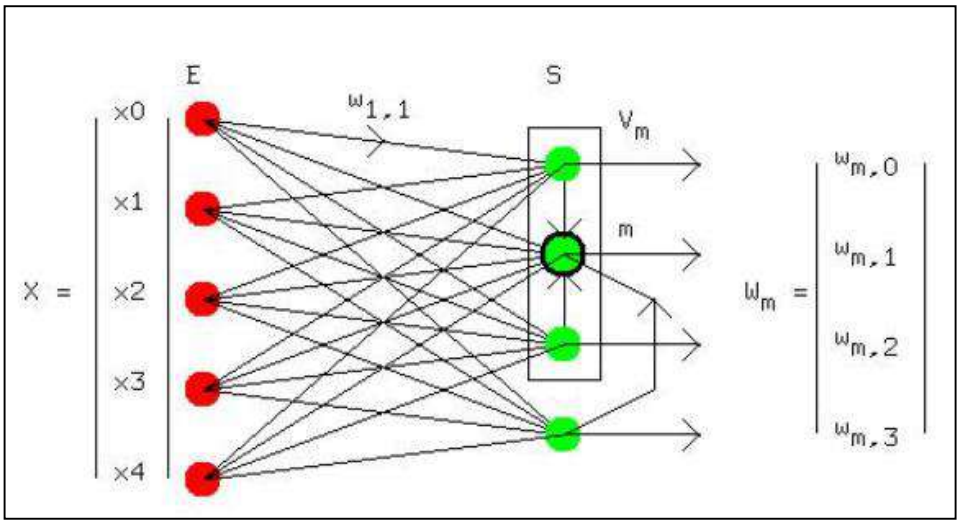


Figure II.6 Réseau de Kohonen.

II.9.5 Les Réseaux de RBF (Radial Basis Fonction)

Le réseau RBF est un réseau de neurones supervisé. Il s'agit d'une spécialisation d'un PMC. Un RBF est constitué uniquement de 3 couches (Figure II.7).

- La couche d'entrée : elle retransmet les entrées sans distorsion.
- La couche RBF : couche cachée qui contient les neurones RBF.
- La couche de sortie : simple couche qui contient une fonction linéaire.

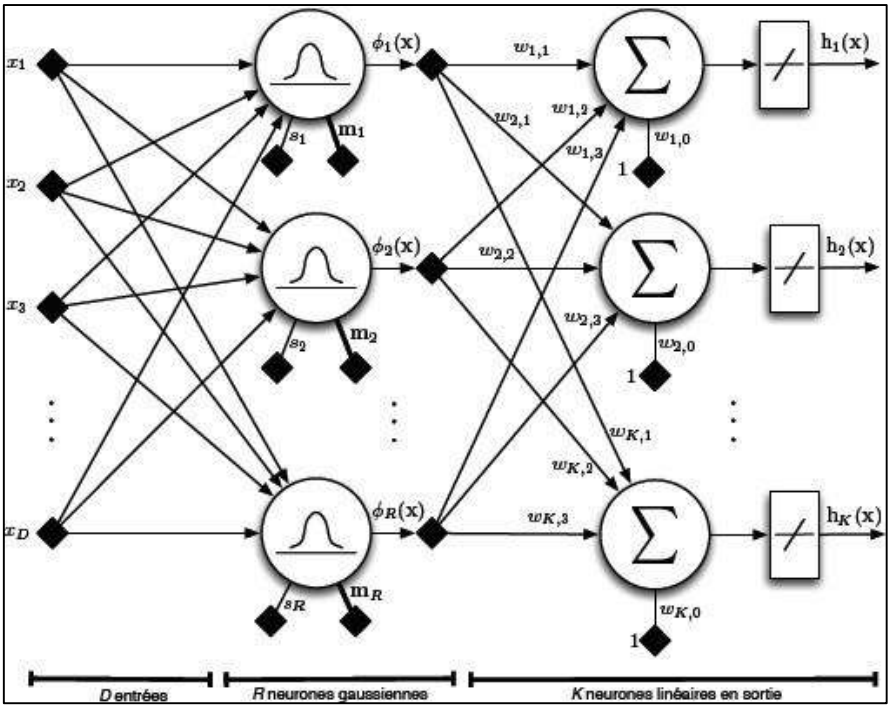


Figure II.7 Schéma d'un RBF.

Chaque neurone RBF contient une gaussienne qui est centrée sur un point de l'espace d'entrée. Pour une entrée donnée, la sortie du neurone RBF est la hauteur de la gaussienne en ce point. La fonction gaussienne permet aux neurones de ne répondre qu'à une petite région de l'espace d'entrée, région sur laquelle la gaussienne est centrée. Donc il y a quatre paramètres principaux à régler dans un réseau RBF.

- Le nombre de neurones RBF (nombre de neurones dans l'unique couche cachée).
- La position des centres des gaussiennes de chacun des neurones.
- La largeur de ces gaussiennes.
- Le poids des connexions entre les neurones RBF et le(s) neurone(s) de sortie.[20]

II.9.6 Réseaux Neurones Convolutionnels (CNN)

Les Réseaux de Neurones Convolutionnels, de l'anglais Convolutional Neural Network (CNN) est un type de réseau neuronal artificiel utilisé dans la reconnaissance et le traitement d'images et spécifiquement conçu pour traiter une énorme quantité des données de pixels, il a été considéré comme l'un des les outils les plus puissants, et est devenu très populaire dans la littérature. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

Un réseau de neurones est un système de matériel et/ou de logiciel conçu d'après le fonctionnement des neurones du cerveau humain. Les réseaux neuronaux traditionnels ne sont pas idéaux pour le traitement des images et doivent être alimentés en images à résolution réduite. Les couches de neurones sont disposées de manière à couvrir l'ensemble du champ visuel, évitant ainsi le problème du traitement d'images par morceaux des réseaux neuronaux traditionnels.

Les réseaux de neurones convolutionnels sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, ils comportent deux parties bien distinctes.

En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentaux [Rouge, Vert, Bleu].
La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une

succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.[20]

L'architecture CNN est formée par un empilement de couches de traitement indépendantes :

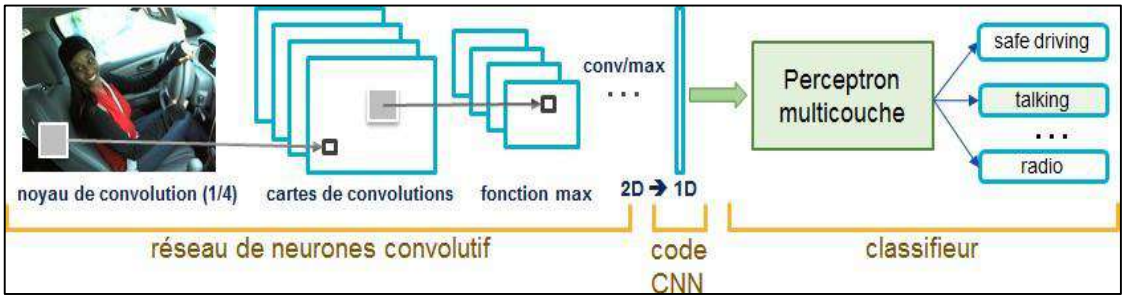


Figure II.8 Architecture Standard d'un Réseau de Neurone Convolutionnel.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

Dans les réseaux CNN, de petites parties de l'image (appelées un champ réceptif local) sont traitées comme des entrées de la couche la plus basse de la structure hiérarchique. L'information se propage généralement à travers les différentes couches du réseau où, à chaque couche, un filtrage numérique est appliqué afin d'obtenir des caractéristiques saillantes des données observées. Le procédé fournit un niveau d'invariance au déplacement, à l'échelle et à la rotation lorsque le champ réceptif local permet aux neurones ou à l'unité de traitement d'accéder à des éléments élémentaires tels que des arêtes ou des coins orientés. Essentiellement, l'image d'entrée est convolutionnée avec un ensemble de N petits filtres dont les coefficients sont soit formés, soit prédéterminés en utilisant certains critères. Ainsi, la première couche (ou la couche la plus basse) du réseau est constituée de «cartes de caractéristiques» qui sont le résultat des processus de convolution, avec un biais additif et éventuellement une compression ou une normalisation des caractéristiques. Cette étape initiale est suivie d'un sous échantillonnage (généralement une opération de moyennage de 2 3 2) qui réduit encore la dimensionnalité et offre une certaine robustesse aux déplacements spatiaux (voir la figure 9). La carte des

caractéristiques sous-échantillonnées reçoit ensuite un biais de pondération et de formation et se propage finalement à travers une fonction d'activation. Certaines variantes existent avec seulement une carte par couche ou des sommations de plusieurs cartes.

Figure II.9 Clarifie le Processus de convolution et de sous-échantillonnage.

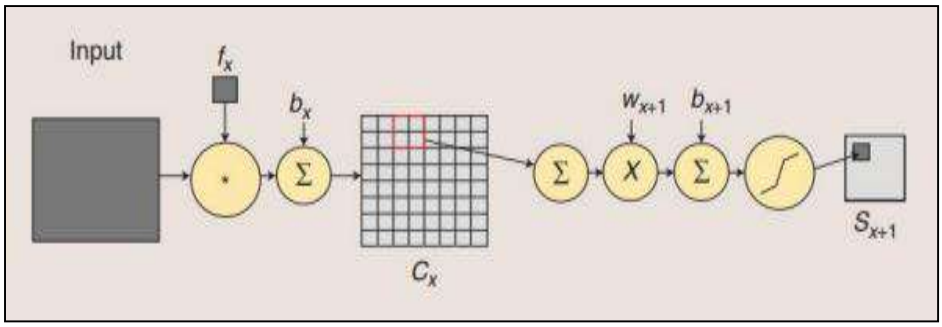


Figure II.9 Processus de convolution et de sous-échantillonnage.

II.9.6.1 Architecture de Réseaux de Neurone Convolutionnel

Les Réseaux de Neurones Convolutionnels (CNN) sont basés sur la perception multicouche(MLP), et inspirés du comportement du cortex visuel des vertébrés. Bien qu'efficaces pour le traitement d'images, les MLP ont beaucoup de mal à gérer des images de grande taille, ce qui est dû à la croissance exponentielle du nombre de connexions avec la taille de l'image.

Par exemple, si on prend une image de taille 32*32*3 (32 de large, 32 de haut, 3 canaux de couleur), un seul neurone entièrement connecté dans la première couche cachée du MLP aurait 3072 entrées (32*32*3). Une image 200x200 conduirait ainsi à traiter 120000 entrées par neurone ce qui, multiplié par le nombre de neurones, devient énorme.

Les CNN visent à limiter le nombre d'entrées tout en conservant la forte corrélation «spatialement locale» des images naturelles. Par opposition aux MLP, les CNN ont les traits distinctifs suivants :

1. Volumes 3D de Neurones : La couche de neurones n'est plus simplement une surface (perceptron), mais devient un volume avec une profondeur. Si on considère un seul champ récepteur du CNN, les n neurones associés (sur la profondeur) forment l'équivalent de la première couche d'un MLP. .
2. Connectivité Locale : Grâce au champ récepteur qui limite le nombre d'entrées du neurone, tout en conservant l'architecture MLP, les CNN assurent ainsi que les «filtres» produisent la réponse la plus forte à un motif d'entrée spatialement localisé, ce qui conduit à une représentation parcimonieuse de l'entrée. Une telle représentation occupe

moins d'espace en mémoire. De plus, le nombre de paramètres à estimer étant réduit, leur estimation (statistique) est plus robuste pour un volume de données fixé (comparé à un MLP).

3. Poids Partagés: Dans les CNNs, les paramètres de filtrage d'un neurone (pour un champ récepteur donné) sont identiques pour tous les autres neurones d'un même noyau (traitant tous les autres champs récepteurs de l'image). Ce paramétrage (vecteur de poids et biais) est défini dans une «carte de fonction». Cela signifie que tous les neurones dans une couche de convolution donnée détectent exactement la même caractéristique. En multipliant les champs récepteurs, il devient possible de détecter des éléments indépendamment de leur position dans le champ visuel, ce qui induit une propriété d'invariance par translation. Ensemble, ces propriétés permettent aux réseaux de neurones convolutionnels d'obtenir une meilleure généralisation (en termes d'apprentissage) sur des problèmes de vision. Le partage de poids permet aussi de réduire considérablement le nombre de paramètres libres à apprendre, et ainsi les besoins en mémoire pour le fonctionnement du réseau. La diminution de l'empreinte mémoire permet l'apprentissage de réseaux plus grands donc souvent plus puissants.[20]

Figure II.10 explique une couche du CNN en 3 dimensions (Vert : volume d'entrée, bleu : volume du champ récepteur, gris : couche de CNN, cercles : neurones artificiels indépendants.

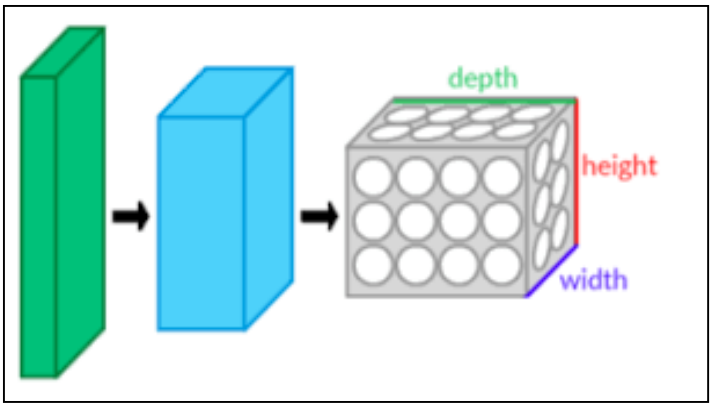


Figure II.10 Une couche du CNN en 3 dimensions. (Vert : volume d'entrée, bleu : volume du champ récepteur, gris : couche de CNN, cercles : neurones artificiels indépendants.

Pour avoir une bonne compréhension de CNN, nous commençons par ses de couches de traitement indépendantes (Figure II.11) :

- Couche d'entrée (Input layer).
- La couche de convolution (CONV) qui traite les données d'un champ récepteur.
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
- La couche de correction (ReLU), souvent appelée par abus 'ReLU' en référence à la fonction d'activation (Unité de rectification linéaire).
- La couche "entièrement connectée" (FC : Fully connected), qui est une couche de type perceptron.
- La couche de perte (LOSS).

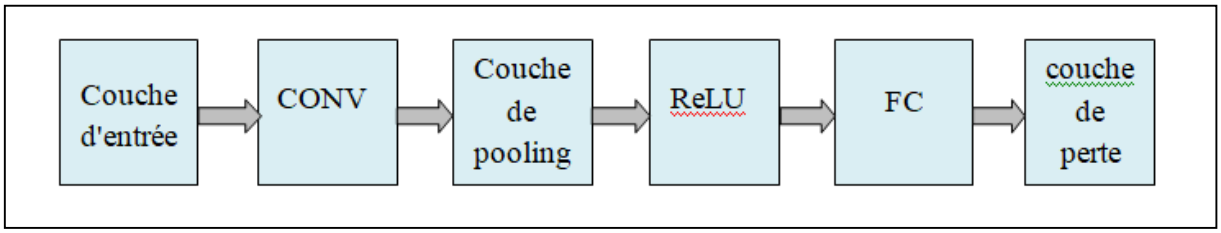


Figure II.11 Différentes couches de CNN.

II.9.6.2 La Couche de Convolution (CONV)

La Couche Convolutionnelle (en anglais convolution layer) (CONV) est le bloc de construction de base d'un CNN. Trois paramètres permettent de dimensionner le volume de la couche de convolution à profondeur, le pas et la marge.

1. Profondeur de la couche **:** nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
2. Le pas **:** contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
3. La Marge(à 0) ou Zero Padding : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce zero-padding est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

Figure II.12 illustre un ensemble de neurones (cercles) créant la profondeur d'une couche de convolution (bleu). Ils sont liés à un même champ récepteur (rouge).

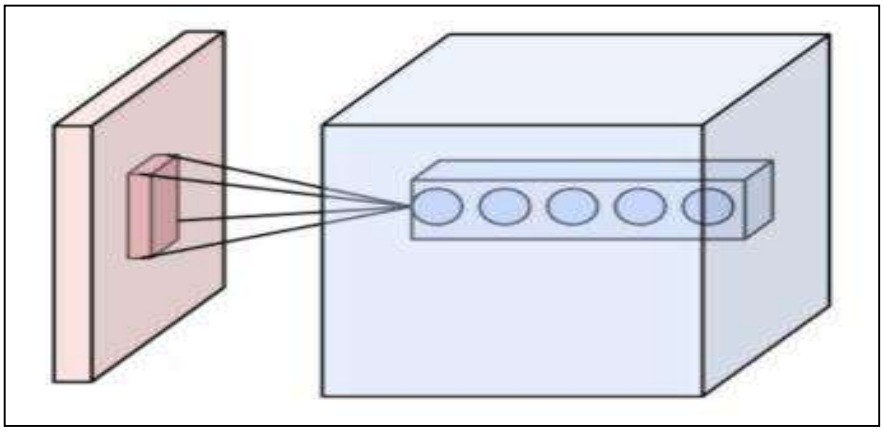


Figure II.12 Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution (bleu). Ils sont liés à un même champ récepteur (rouge).

II.9.6.3 Pooling (POOL)

Après chaque couche convolutive, il peut y avoir une couche de pooling. La couche de pooling (en anglais pooling layer) (POOL) est une opération de sous-échantillonnage typiquement appliquée après une couche convolutionnelle. En particulier, les types de pooling les plus populaires sont le max et l'average pooling, où les valeurs maximales et moyennes sont prises, respectivement.[21]

L'image d'entrée est découpée en une série de rectangles de n pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs prises par les différents pixels de la tuile.

Figure II.13 présente la déférence entre max pooling et average pooling.

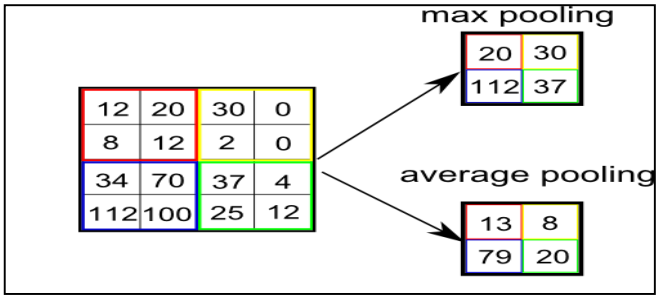


Figure II.13 Déférence entre max pooling et average pooling.

II.9.6.4 Couches de Correction (RELU)

Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie.

La fonction ReLU (abréviation d'Unités Rectifié linéaires):$F(x) = \max(0, x)$, cette fonction force les neurones à retourner des valeurs positives.

II.9.6.5 Couche Entièrement Connectée (FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente. Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation.[20]

II.9.6.6 La Couche de Perte (LOSS)

La couche de perte spécifie comment l'entrainement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La perte « Softmax » est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives. La perte par entropie croisée sigmoïde est utilisée pour prédire K valeurs de probabilité indépendante dans [0,1]. La perte euclidienne est utilisée pour régresser vers des valeurs réelles.

Figure II.14 illustre un schéma général de principe de fonctionnement d'un réseau de neurones convolutifs.

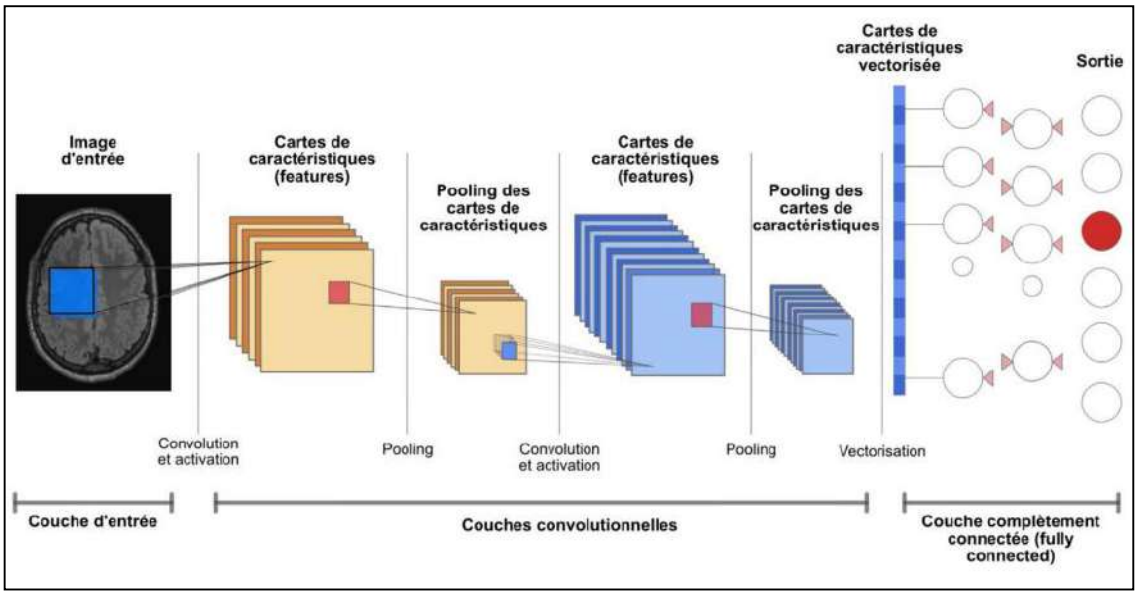


Figure II.14 Schéma général de principe de fonctionnement d'un réseau de neurones convolutifs.

II.9.7 Entrainement d'un Nouveau CNN

Pour créer un nouveau réseau de neurones convolutif est besoin d'une certaine expertise, du matériel informatique et une quantité de données annotées nécessaires.

D'abord il s'agit de fixer l'architecture du réseau ça veut dire leurs tailles, le nombre de couches et les opérations matricielles qui les connectent. L'entraînement consiste à optimiser les coefficients du réseau pour minimiser l'erreur de classification en sortie. Cet entraînement peut prendre plusieurs semaines pour les meilleurs CNN, avec de nombreux GPU travaillant sur des centaines de milliers d'images annotées. Des équipes de recherche se spécialisent dans l'amélioration des CNN. Elles publient leurs innovations techniques, ainsi que le détail des réseaux entraînés sur des bases de données de références.

II.9.8 Modèles de CNN

Cette section décrit quelques exemples de modèles CNN.

II.9.8.1 Exemple 1

La forme la plus commune d'une architecture CNN empile quelques couches Conv-ReLU, les suit avec des couches Pool, et répète ce schéma jusqu'à ce que l'entrée soit réduite dans un espace d'une taille suffisamment petite. À un moment, il est fréquent de placer des couches

entièrement connectées (FC : fully-connected). La dernière couche entièrement connectée est reliée vers la sortie. Voici quelques architectures communes CNN qui suivent ce modèle :

- INPUT -⟩ CONV -> RELU -> FC.
- INPUT -⟩ [CONV -> RELU -> POOL] * 2 -> FC -> RELU -> FC Ici, il y a une couche de CONV unique entre chaque couche POOL.
- INPUT -⟩ [CONV -> RELU -> CONV -> RELU -> POOL] * 3 -> [FC -> RELU] * 2 -> FC Ici, il y a deux couches CONV empilées avant chaque couche POOL.

  L'empilage des couches CONV avec de petits filtres de pooling (plutôt un grand filtre) permet un traitement plus puissant, avec moins de paramètres. Cependant, avec l'inconvénient de demander plus de puissance de calcul (pour contenir tous les résultats intermédiaires de la couche CONV).[22]

Figure II.15 présente un exemple de modèle de CNN.

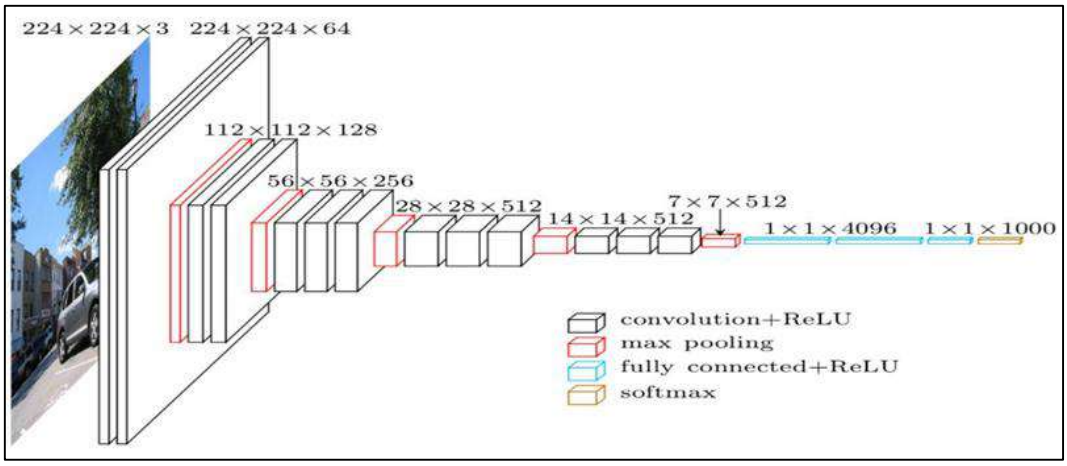


Figure II.15 Exemple de modèle de CNN.

II.9.8.2 Exemple 2

Cet exemple montre la création d'un autoencodeur pour le débruitage d'images utilisant un modèle CNN. Le principe de fonctionnement est simple : il faut trouver une fonction d'encodage puis de décodage permettant de retrouver l'apparence de l'image originale (ou plus généralement de minimiser la fonction de cout entre les données en entrée et la reconstruction en sortie).

- Nous ajoutons un bruit aléatoire sur les images (Figure II.16).

```
[46] plt.figure(figsize=(20, 2))
     for i in range(1,10):
         ax = plt.subplot(1, 10, i)
         plt.imshow(x_test_noisy[i].reshape(28, 28), cmap="binary")
     plt.show()
```

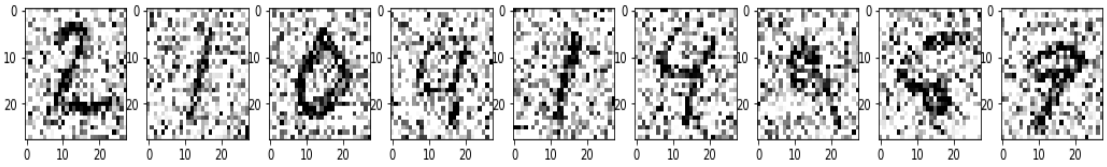


Figure II.16 Affichage d'images avec bruit.

- Construction du modèle : Nous allons maintenant passer à l'implémentation du modèle via les librairies Keras (Figure II.17).

```
[47] model = Sequential()
     model.add(Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(28, 28, 1)))
     model.add(MaxPooling2D((2, 2), padding='same'))
     model.add(Conv2D(8, (3, 3), activation='relu', padding='same'))
     model.add(MaxPooling2D((2, 2), padding='same'))
     model.add(Conv2D(8, (3, 3), activation='relu', padding='same'))

[48] model.add(MaxPooling2D((2, 2), padding='same'))

     model.add(Conv2D(8, (3, 3), activation='relu', padding='same'))
     model.add(UpSampling2D((2, 2)))
     model.add(Conv2D(8, (3, 3), activation='relu', padding='same'))
     model.add(UpSampling2D((2, 2)))
     model.add(Conv2D(32, (3, 3), activation='relu'))
     model.add(UpSampling2D((2, 2)))
     model.add(Conv2D(1, (3, 3), activation='relu', padding='same'))

[50] model.compile(optimizer='adam', loss='mean_squared_error')

[51] model.summary()
```

Figure II.17 Modèle.

- Entrainement de l'autoencodeur : Avant de pouvoir utiliser ce réseau il est nécessaire d'effectuer son entrainement.

- De même qu'un réseau classique il faut alimenter les données en entrée du réseau et la sortie attendue. Il est important de noter que l'entrée sont les images bruitées et les sorties les images non bruitées. Cette combinaison doit permettre à notre modèle d'apprendre une façon de débruiter les images (Figure II.18).

```
Epoch 1/10
200/200 [==============================] - 34s 172ms/step - loss: 0.7534 - acc: 0.7941 -
val_loss: 0.2871 - val_acc: 0.9210
Epoch 2/10
200/200 [==============================] - 34s 170ms/step - loss: 0.2758 - acc: 0.9174 -
val_loss: 0.1973 - val_acc: 0.9431
Epoch 3/10
200/200 [==============================] - 34s 169ms/step - loss: 0.2048 - acc: 0.9395 -
val_loss: 0.1527 - val_acc: 0.9538
Epoch 4/10
200/200 [==============================] - 34s 171ms/step - loss: 0.1486 - acc: 0.9573 -
val_loss: 0.1191 - val_acc: 0.9650
Epoch 5/10
200/200 [==============================] - 34s 171ms/step - loss: 0.1300 - acc: 0.9622 -
val_loss: 0.0969 - val_acc: 0.9705
Epoch 6/10
200/200 [==============================] - 34s 171ms/step - loss: 0.1092 - acc: 0.9684 -
val_loss: 0.0861 - val_acc: 0.9712
Epoch 7/10
200/200 [==============================] - 34s 170ms/step - loss: 0.0901 - acc: 0.9721 -
val_loss: 0.0806 - val_acc: 0.9717
Epoch 8/10
200/200 [==============================] - 34s 170ms/step - loss: 0.0926 - acc: 0.9726 -
val_loss: 0.0735 - val_acc: 0.9755
Epoch 9/10
200/200 [==============================] - 34s 169ms/step - loss: 0.0783 - acc: 0.9767 -
val_loss: 0.0679 - val_acc: 0.9771
Epoch 10/10
200/200 [==============================] - 34s 172ms/step - loss: 0.0701 - acc: 0.9787 -
val_loss: 0.0586 - val_acc: 0.9807
```

Figure II.18 Entrainement du classifier.

- Nous débruitons les images grâce à notre autoencodeur et faisons une prédiction de la classe de l'image. Le résultat ci-dessous (Figure II.19).

```
[60] plt.figure(figsize=(40, 4))
     for i in range(10):
         # display original
         ax = plt.subplot(3, 20, i + 1)
         plt.imshow(x_test_noisy[i].reshape(28, 28), cmap="binary")
         ax = plt.subplot(3, 20, 40 +i+ 1)
         plt.imshow(no_noise_img[i].reshape(28, 28), cmap="binary")

     plt.show()
```

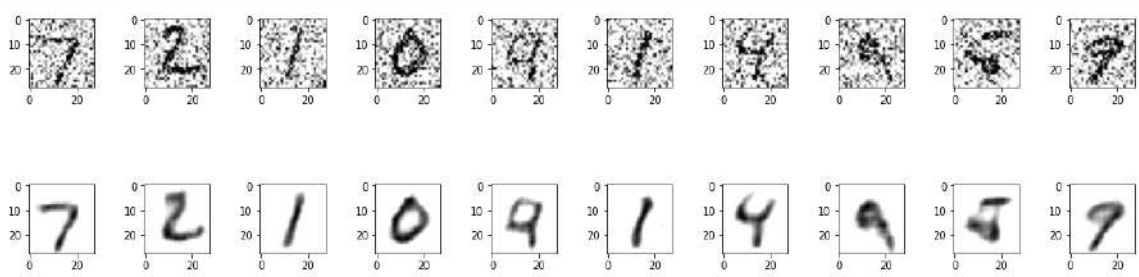


Figure II.19 Affiche l'image reconstruite (après suppression du bruit).

II.9.9 Choix des Hyperparamétres

Les CNNs utilisent plus d'hyper paramètres qu'un MLP standard. Même si les règles habituelles pour les taux d'apprentissage et des constantes de régularisation s'appliquent toujours, il faut prendre en considération les notions de nombre de filtres, leur forme et la forme du max pooling.

II.9.9.1 Nombre de Filtres

Comme la taille des images intermédiaires diminue avec la profondeur du traitement, les couches proches de l'entrée ont tendance à avoir moins de filtres tandis que les couches plus proches de la sortie peuvent en avoir davantage. Pour égaliser le calcul à chaque couche, le produit du nombre de caractéristiques et le nombre de pixels traités est généralement choisi pour être à peu près constant à travers les couches. Pour préserver l'information en entrée, il faudrait maintenir le nombre de sorties intermédiaires (nombre convolutionnels 28 d'images intermédiaire multiplié par le nombre de positions de pixel) pour être croissante (au sens large) d'une couche à l'autre.

le nombre d'images intermédiaires contrôle directement la puissance du système, dépend du nombre d'exemples disponibles et la complexité du traitement.

II.9.9.2 Forme du Filtre

Les formes de filtre varient grandement dans la littérature. Ils sont généralement choisis en fonction de l'ensemble de données. Les meilleurs résultats sur les images de MNIST

(28x28) sont habituellement dans la gamme de 5x5 sur la première couche, tandis que les ensembles de données d'images naturelles (souvent avec des centaines de pixels dans chaque dimension) ont tendance à utiliser de plus grands filtres de première couche de 12x12, voire 15x15.Le défi est donc de trouver le bon niveau de granularité de manière à créer des abstractions à l'échelle appropriée et adaptée à chaque cas.

II.9.9.3 Forme du Max Pooling

Les valeurs typiques sont 2x2. De très grands volumes d'entrée peuvent justifier un pooling 4x4 dans les premières couches. Cependant, le choix de formes plus grandes va considérablement réduire la dimension du signal, et peut entraîner la perte de trop d'information.

II.9.10 Avantages de CNN

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrants dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation. C'est le principal avantage du CNN par rapport au MLP, qui lui considère chaque neurone indépendant et donc affecte un poids différent à chaque signal entrant.

II.10 Conclusion

Dans ce chapitre nous avons décrit les notions importantes qui sont en relation avec l'intelligence artificielle .Aussi qu'une vision générale sur l'apprentissage profond et sa méthode de classification. Ensuit nous avons passé aux réseaux de neurones en citant leur fonctionnement, les différents types de réseaux de neurones, ses avantages et inconvénients. A la fin en détaillant les réseaux de neurones convolutionnels.

Dans le prochain chapitre, nous allons présenter les outils hardware et software utilisés dans notre projet.

Chapitre III

Présentation hardware et software

III.1 Introduction

La réalisation d'un system de détection et classification des maladies foliaire nécessite plusieurs outils software et hardware, en citant le Raspberry Pi, l'environnement de programmation Anaconda utilisant le langage python, ainsi que l'environnement d'exécution Google Colab.

De ce fait, dans ce chapitre nous allons introduire le software utilisé, suivie par une présentation des deux outils hardware Raspberry Pi et le module caméra OpenMV et leurs caractéristiques.

III.2 Partie software

III.2.1 Anaconda

III.2.1.1 Présentation d'Anaconda

L'utilisation d'Anaconda pour installer python pour la data science permet de simplifier un certain nombre d'étapes dans la création de vos projets python pour la data science.

Anaconda regroupe un ensemble d'outils gravitant autour des langages de programmation pour le traitement des données notamment en python et en R, il fournit notamment les deux environnements d'exécution. Cette distribution de Python est orientée (science des données) et (l'apprentissage automatique), elle est certainement la plus populaire.

Anaconda ce sont des solutions pour les entreprises afin de simplifier le passage en production des projets data science. Grâce à ces outils, les data scientistes continuent à utiliser leurs outils de prédilection tout en bénéficiant d'outils de mise en production et de sécurisation professionnels. Elle est à la fois un outil et une entreprise. Celle-ci contribue à une part importante de l'écosystème de la data science en python par son action de soutien à de nombreux projets open source. Elle met à votre disposition une solution vous permettant d'installer tous l'écosystème data avec notamment [23] :

- Une installation de python.
- Les packages de data science tels que numpy, pandas, scikit-learn, etc.
- Des IDE de dernière génération tels que Jupyter (JupyterLab et Jupyter Notebooks) ou Spyder.
- L'outil conda pour gérer les environnements et les répertoires de packages.

III.2.1.2 Téléchargement d'Anaconda pour python

Pour Télécharger Anaconda. Il faut bien :

- Prendre une version pour Python 3 et en 64bits, la dernière version se trouve ici (https://www.anaconda.com/products/individual).

- Choisir un système d'exploitation cible (Windows, Mac, etc).
- Sélectionner la version 3.X (à l'heure de l'écriture de ces lignes, c'est la version 3.6 qui est proposée). Illustre par la figure.
- Privilégier la version 64 bits.



Figure Ⅲ.1 Différentes versions de python.

III.2.1.3 Etapes d'installation

L'installation du logiciel se fait en huit étapes, ordonnées comme suit :

1. Pour entamer l'installation, il faut un double-clic sur le bouton télécharger et le bouton Next par la suite (Figure Ⅲ.2).

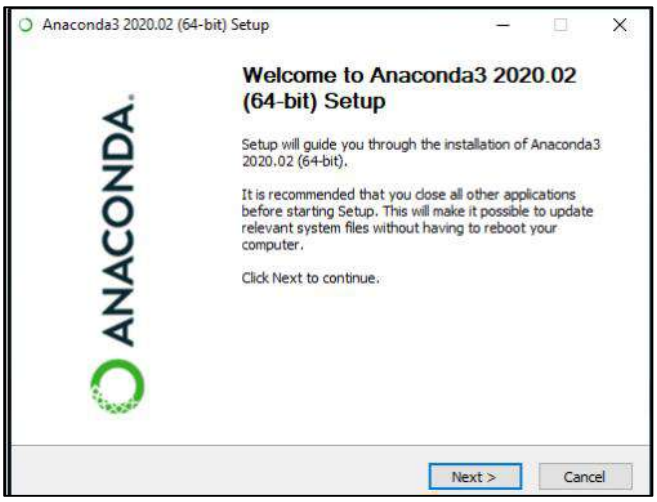


Figure Ⅲ.2 Etape 1.

2. La lecture du contrat de licence en cliquant sur le bouton j'accepte (I Agree). **(**Figure Ⅲ.3)

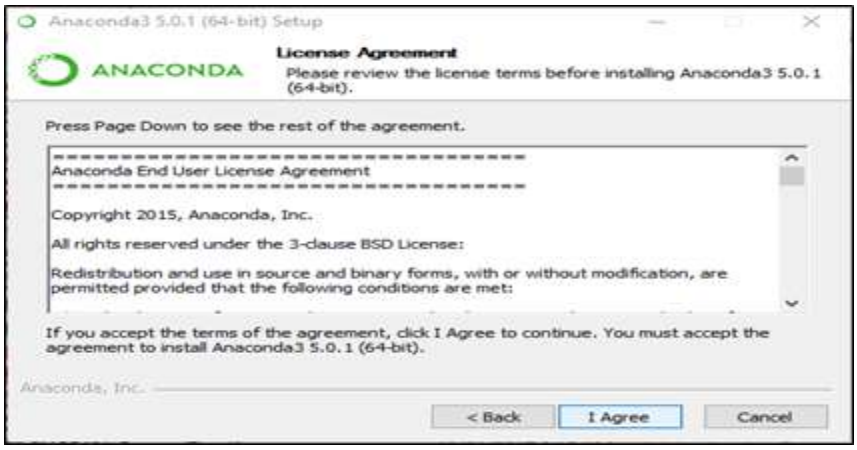


Figure Ⅲ.4 Etape 2.

3. En suite, il faut un clic sur le bouton Next (Figure Ⅲ.4).

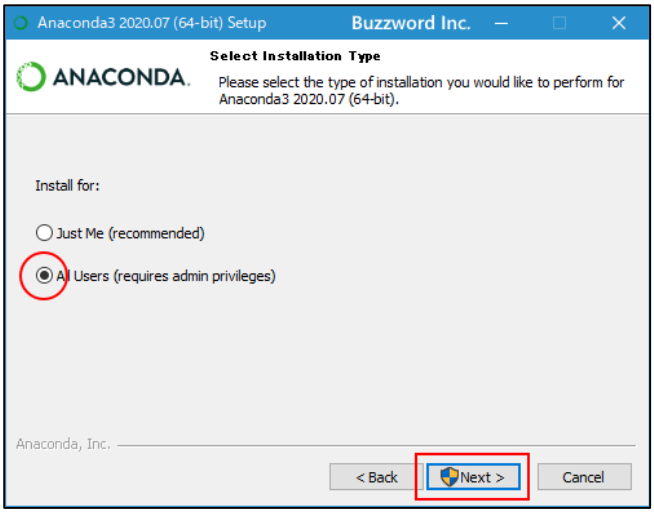


Figure III.5 Etape 3.

4. Dans cet étape, il faut choisir l'emplacement d'installation, en cliquant sur le bouton Next par la suite (Figure III.5).

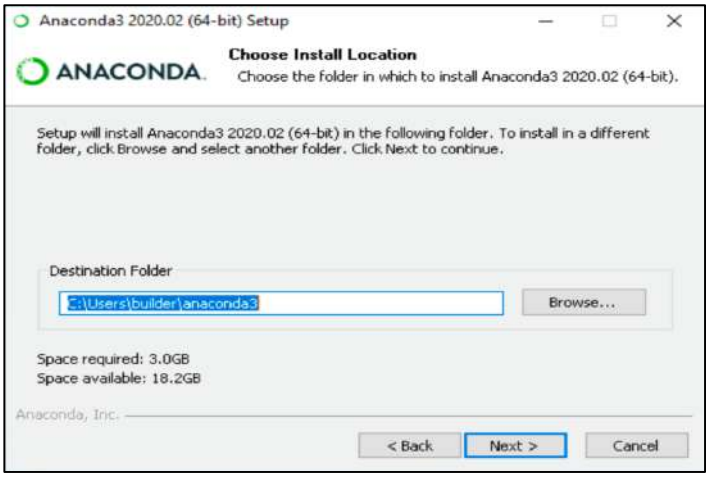


Figure III.6 Etape 4.

5. Cette étape, c'est une partie importante du processus d'installation. L'approche recommandée est de ne pas cocher la case pour ajouter Anaconda à votre chemin. Cela signifie que vous devrez utiliser Anaconda Navigator ou l'invite de commande Anaconda (située dans le menu Démarrer sous "Anaconda"), lorsque vous souhaitez utiliser Anaconda (vous pouvez toujours ajouter Anaconda à votre PATH plus tard si vous ne cochez pas la

case). Si vous souhaitez utiliser Anaconda dans votre invite de commande. Il faut utiliser l'approche alternative et cocher la case (Figure Ⅲ.6).

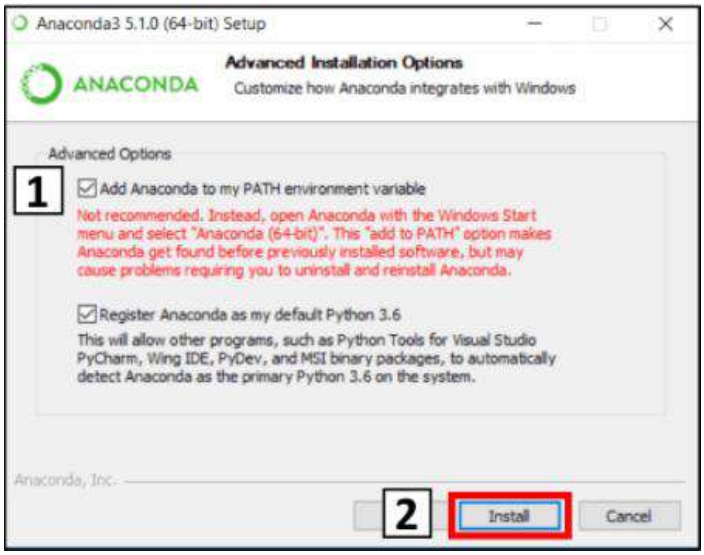


Figure Ⅲ.7 Etape 5.

6. Le clic sur le bouton Next (Figure Ⅲ.7).

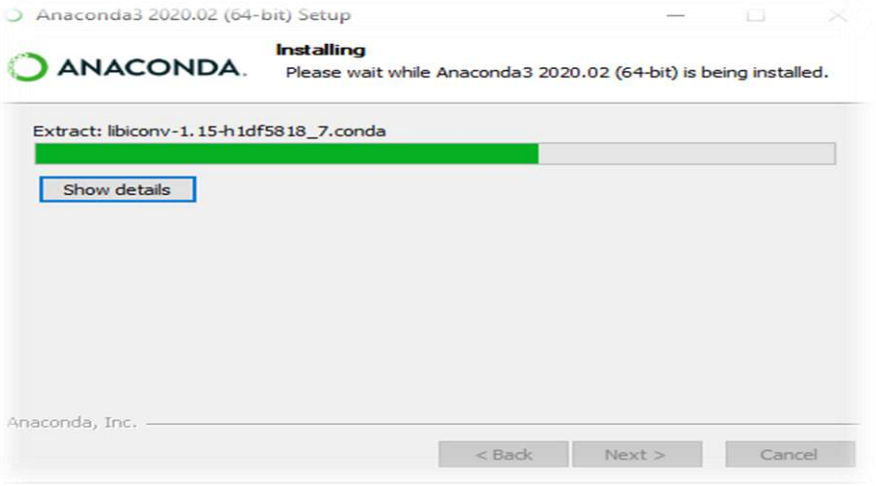


Figure Ⅲ.8 Etape 6.

7. L'installation du Microsoft VSCode si vous le souhaitez, mais c'est facultatif (FigureⅢ.8).

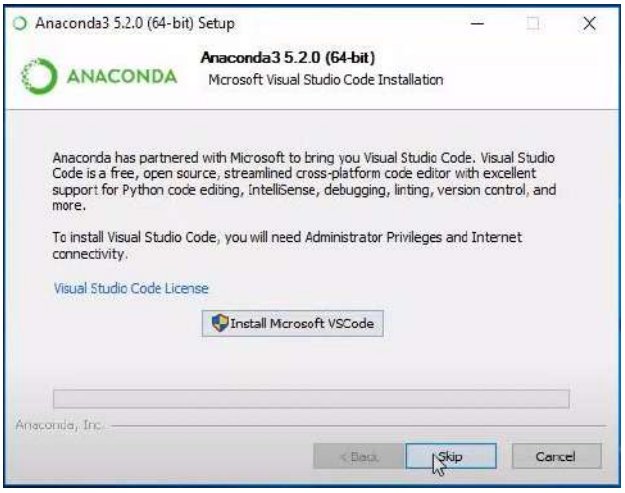


Figure Ⅲ.9 Etape 7.

8. Et pour finir l'installation, il faut juste un clic sur le bouton Finish (Figure Ⅲ.9).



Figure Ⅲ.10 Etape 8.

Une fois le processus terminé, vous aurez dans votre menu Windows accès à différents outils, les deux principaux étant :

- Anaconda Navigator qui est une interface visuelle pour gérer vos environnements.
- Anaconda Prompt qui est un terminal pour gérer vos environnements.

A l'issue de l'installation, vous aurez Anaconda Navigator. Ce dernier est une interface graphique permettant de lancer Jupyter (pour écrire du python au format notebook), ou encore Spyder (IDE pour éditer du code python). Comme montre dans la figure Ⅲ.10)

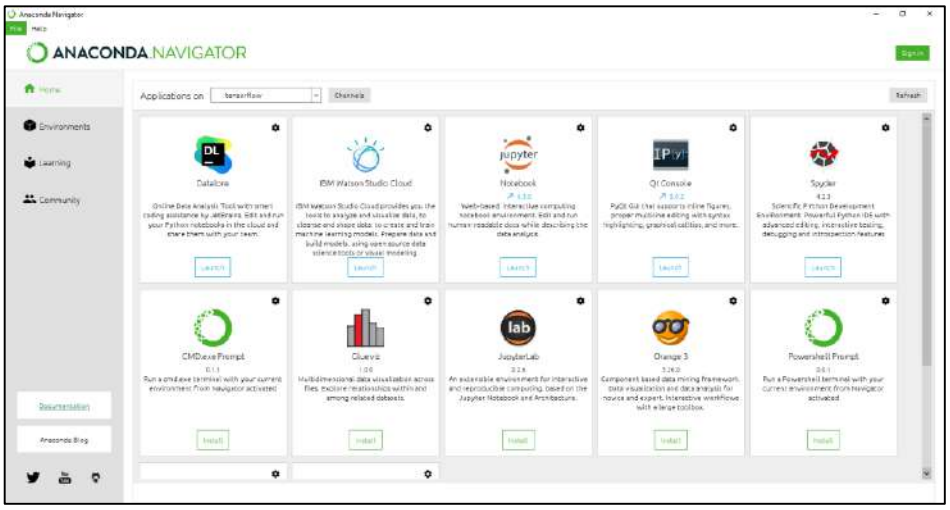


Figure Ⅲ.11 Interface d'Anaconda Navigator.

Après l'installation d'Anaconda, il est toujours utile de s'assurer qu'on a les dernières versions des packages qu'on sera amené à utiliser. Pour les besoins du Machine Learning, on a généralement besoin des librairies suivantes :

- NumPy. : Une bibliothèque qui prend en charge les tableaux multidimensionnels et les matrices.

- SciPy : est un écosystème basé sur Python de logiciels open source pour les mathématiques, les sciences et l'ingénierie.
- Matplotlib : est une bibliothèque complète permettant de créer des visualisations statiques, animées et interactives en Python.
- CV2 : OpenCV représente Computer Vision Open Source. Il contient la bibliothèque de capacités de programmation pour la programmation de l'IA. Un CV ouvert est requis pour les applications de traitement d'images en continu. OpenCV est généralement créé en C, C ++ et son interface de ligne directrice est en langage C ++, mais indépendamment de tout ce qu'il contient, peu importe la façon dont vous le regardez, une interface en langage C large.

- OS : Le module OS est un module fournit par Python dont le but d'interagir avec le système d'exploitation**,** il permet ainsi de gérer l'arborescence des fichiers, de fournir des informations sur le système d'exploitation processus, variables systèmes, ainsi que de nombreuses fonctionnalités des systèmes.
  Le module OS peut être chargé simplement avec la commande : import os

- Pickle : Le module Pickle implémente des protocoles binaires pour sérialiser et désérialiser une structure d'objet Python. Il est le processus par lequel une hiérarchie d'objets Python est convertie en un flux d'octets.
- Pandas : Pandas est un outil d'analyse et de manipulation de données open source. Il est rapide, puissant, flexible et facile à utiliser, construit sur le langage de programmation Python.

- Statsmodels : est une bibliothèque Python spécialement conçue pour les statistiques. Statsmodels est construit sur NumPy, SciPy et matplotlib, mais il contient des fonctions plus avancées pour les tests statistiques et la modélisation que vous ne trouverez pas dans les bibliothèques numériques comme NumPy ou SciPy.
- Scikit-learn : Suppose une connaissance pratique très basique des pratiques d'apprentissage automatique (ajustement de modèle, prédiction, validation croisée, etc.).
  Scikit-learn est une bibliothèque d'apprentissage automatique open source qui prend en charge l'apprentissage supervisé et non supervisé. Il fournit également divers outils pour l'ajustement du modèle, le prétraitement des données, la sélection et l'évaluation du modèle, ainsi que de nombreux autres utilitaires.[24]

Avec Conda, il est possible de télécharger la dernière version d'un package donné à l'aide de la commande suivante : (conda update PACKAGE-NAME), par exemple, pour mettre à jour Sickit-learn, on fera : (conda update scikit-learn).

III.2.2 Spyder

Spyder est un environnement scientifique puissant écrit en Python. Nous recommandons fortement le logiciel gratuit et open source Spyder Integrated Développement Environnement (IDE) pour la programmation scientifique et technique.
Il offre une combinaison unique des fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution

interactive, l'inspection approfondie et les superbes capacités de visualisation d'un progiciel scientifique. Spyder inclus dans Anaconda complet avec l'exploration de données.[25]

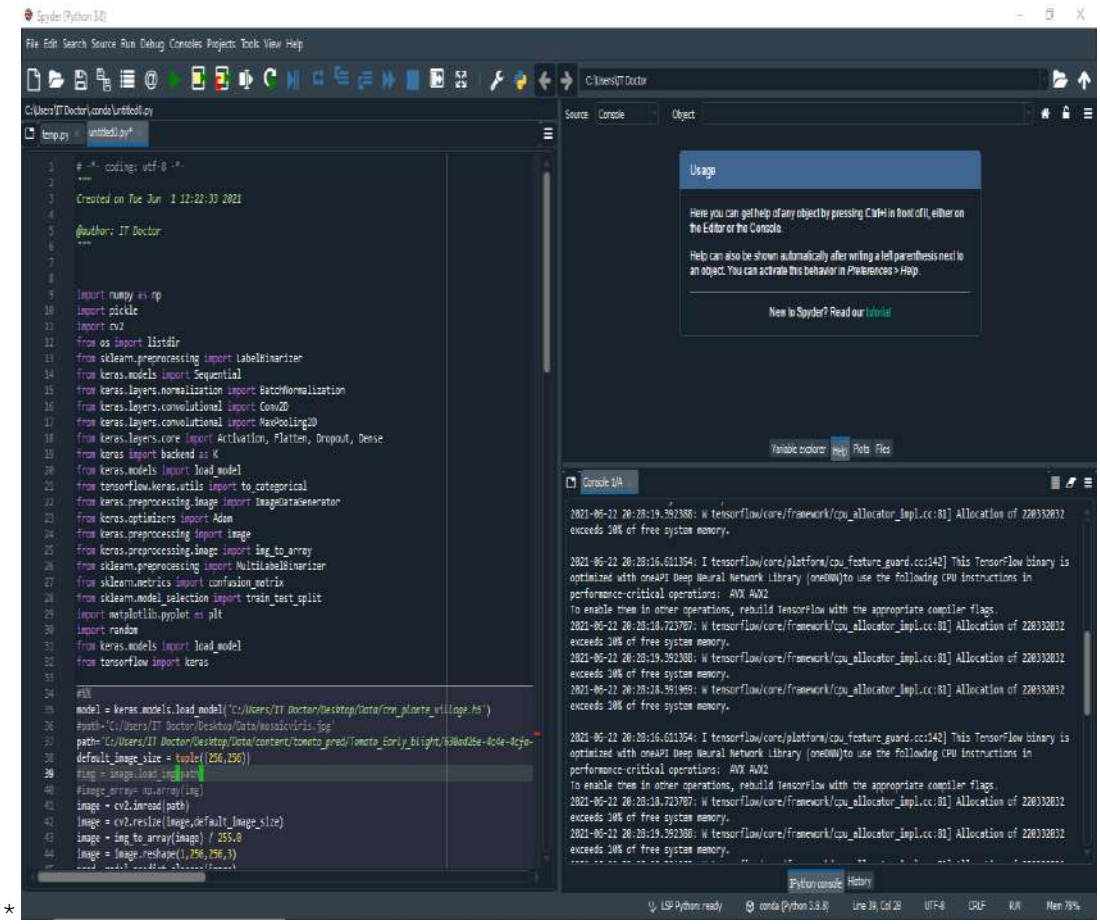


Figure Ⅲ. 12 Présentation de l'interface du logiciel.

III.2.3 TensorFlow

TensorFlow est un environnement (framework) de programmation, il est un cadre d'apprentissage en profondeur et un cadre de calcul général qui permet d'effectuer des opérations mathématiques générales de manière parallèle et distribuée.

Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multidimensionnelles, appelées Tenseurs (Tensor). Un Tenseur à deux dimensions est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow: Gmail, Google Photos, Reconnaissance de voix.

III.2.4 Keras

Keras est une API conçue pour les êtres humains, pas pour les machines. Keras suit les meilleures pratiques pour réduire la charge cognitive, il propose des API cohérentes et simples, il minimise le nombre d'actions utilisateur requises pour les cas d'utilisation courants et il fournit des messages d'erreur clairs et exploitables. Il contient également une documentation complète et des guides de développement.

Keras est l'énervement d'apprentissage en profondeur le plus utilisé parmi les 5 meilleures équipes gagnantes sur Kaggle. Parce que Keras facilite l'exécution de nouvelles expériences, il vous permet d'essayer plus d'idées que vos concurrents, plus rapidement.[26]

III.2.5 Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. Google colab est un outil complet pour entraîner rapidement et tester rapidement des modèles d'apprentissage automatique sans avoir de contrainte matérielle. Sa particularité est que tout le monde peut l'utiliser gratuitement.

Pour Réduire le temps d'apprentissage de vos modèles avec google colab .Il faut entrainer un modèle sur CPU, GPU et TPU afin de mesurer les différentes durées d'apprentissage.

De plus, les documents Colab (Jupyter Notebook) sont enregistrés directement votre compte Google Drive.[27]

Les notebooks Colab vous permettent d'utiliser dans un même document, du code exécutable, du texte enrichi**,** des images, du code HTML, du code LaTeX et bien plus.

Lorsque vous créez des notebooks Colab, ils sont enregistrés dans votre compte Google Drive. Vous pouvez facilement les partager avec vos collaborateurs.

Il permet de :

- D'améliorer vos compétences de codage en langage de programmation Python.
- De développer des applications en Deep Learning en utilisant des bibliothèques Python populaires telles que Keras, TensorFlow, PyTorch et OpenCV.
- D'utiliser un environnement de développement (Jupyter Notebook) qui ne nécessite aucune configuration.
- Colab permet vraiment de démocratiser l'accès à l'Intelligence Artificielle.

III.2.6 Caractéristiques

- Un GPU qui est 60 fois plus rapide qu'un entrainement classique sur un CPU.
- Aucune configuration requise.
- Accès gratuit aux GPU.
- Partage facile.
- Un outil Google simple et gratuit pour vous initier au Deep Learning.
- C'est un outil aussi simple à utiliser que performant.

La Figure Ⅲ.12 représente l'interface de Google Colab.

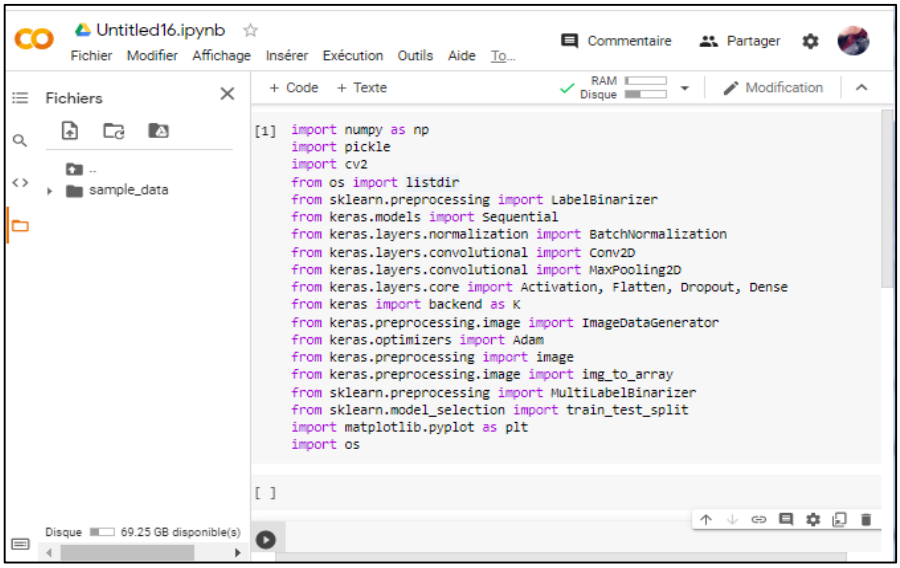


Figure Ⅲ.13 Interface de Google Colab.

III.3 Partie hardware

III.3.1 Carte Raspberry Pi

Comme le Raspberry Pi que l'on a utilisé dans notre projet. Nous le présenterons dans la suite, Cette dernière joue le rôle d'un serveur Web affichant en temps réel plusieurs valeurs stockée dans une base de données donc il est considéré comme le cerveau de notre système.

III.3.1.1 Présentation de Raspberry Pi

Raspberry Pi est un ordinateur réduit à sa plus simple expression, une unique carte à processeur ARM un poil plus grande qu'une carte de crédit. C'est un mini-ordinateur mono-carte à processeur ARM et fait un peu penser à l'Arduino, capable de faire fonctionner des systèmes d'exploitation comme Linux et même une version réduite de Windows 10. Ce circuit imprimé Son intérêt rendre l'informatique abordable et accessible à tous en encourageant

l'apprentissage de la programmation informatique et de ses différents langage (python, scratch, etc.).Ce circuit imprimé dont la plupart des composants sont sous licence libre.
Le Raspberry Pi est livré sans boitier, alimentation et encore moins de clavier, souris ou écran. L'objectif ici est bien entendu de réduire les coûts au maximum et de permettre à son acquéreur de pouvoir recycler du matériel déjà en sa possession.[28]
Il se veut désormais plus puissante, adaptée à plus d'usages, et aussi, plus accessible à tous. Généralement il utilise la bibliothèque Open CV.

III.3.1.2 Les composants standards de Raspberry

Puisqu'il y a plusieurs modèles de Raspberry on peut représenter l'organisation générale et les différents connecteurs disponibles d'un Raspberry standard :

- Le centre de ce circuit est occupé par le processeur.
- L'autre gros circuit intégré visible sur cette face du circuit imprimé n'est autre que le contrôleur réseau, absent bien évidemment sur le modèle A.
- Le double connecteur USB voisin et quant à lui conforme à la norme USB 2.0, Rétrocompatible USB 2.0 comme il se doit.
- Il y a cinq LED d'état du Raspberry.

    - Une LED rouge de mise sous tension.
    - Trois LEDs vertes l'une pour l'activation du processeur, l'autre pour signaler l'activation de réseau. et la dernière pour signaler le mode full duplex de la connexion réseau.
    - Une LED jaune pour indiquer que le réseau fonctionne à 100 Mbits par seconde.

- La carte comporte :
    - Un processeur ARM.
    - Mémoire vive RAM.
    - Une connectique variée :
        - ❖ Port USB.
        - ❖ Emplacement pour carte SD.
        - ❖ Port Ethernet.
        - ❖ Prise audio ou Une prise jack.
        - ❖ Port HDMI (Hight Definition Multimedia Interface).
        - ❖ Port RCA (Radio Corporation of America),

- ❖ Connecteur d'alimentation Micro USB.
- ❖ Ports GPIO (General Purpose Input/Output).

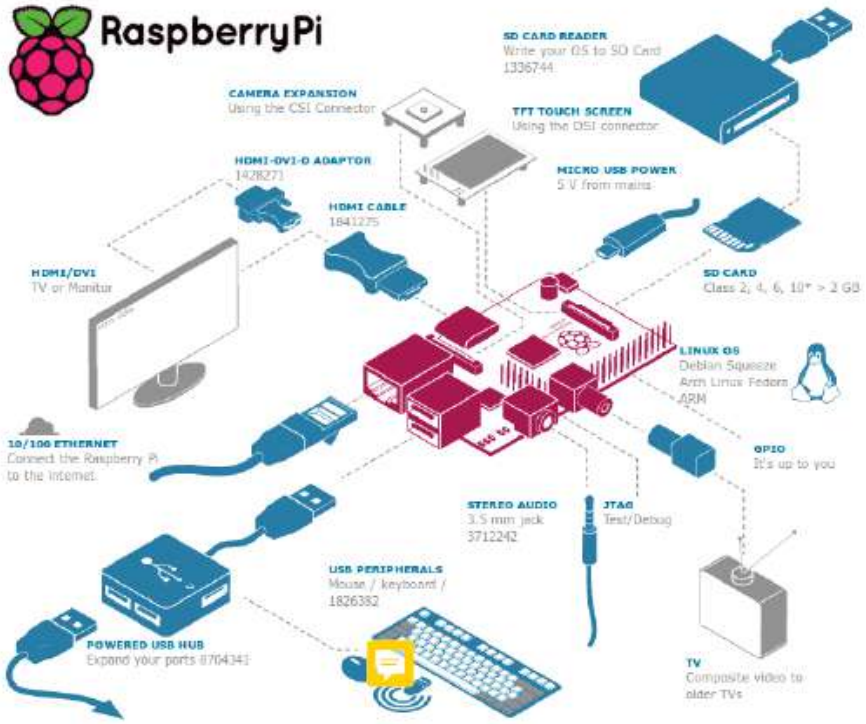


Figure Ⅲ.14 Composants standards d'un Raspberry pi.

III.3.1.3 Différents modèles du Raspberry Pi

Il y a plusieurs modèles du Raspberry Pi depuis sa conception :

- Modèle A : il est utilisé pour un projet à faible coût qui nécessite un ordinateur complet sans capacités de mise en réseau et un support E/S décent.
- Modèle B **:** Ce modèle peut être utilisé pour un projet où le prix n'est pas un problème et où le Pi le plus puissant est nécessaire. Ce modèle contient également des E/S faciles à utiliser, il convient donc aux premiers projets Pi.
- Model Zéro : Ce modèle est idéal pour un projet à très faible coût, dans un espace restreint, qui nécessite un ordinateur en parfait état de fonctionnement et qui bénéficierait d'une connectivité sans fil.

Le tableau III.1 montre les déférences entre certains modèles de Raspberry pi.

Tableau III.1 Comparaison entre les modèles de RPI.

	Génération Zéro	Génération 1		Génération 2	Génération 3	Génération 4
Modèle	Zéro	A	B	2B	3 A+	4 B
RAM	512 Mo	256 Mo	512 Mo	1 Go	512 Mo	1Go / 2Go / 4Go / 8Go
Processor	1GHz ARM11	700 MHz ARM11	700 MHz ARM11	900 quadricoeur ARM Cortex-A7	1,4 quadricoeur ARM Cortex-A53	1,5 quadricoeur ARM Cortex-A72
Réseau	NON	NON	10/100 Mbits/s Ethernet	10/100 Mbits/s Ethernet	Wifi 802.11 ac Bluetooth 4.2	Ethernet Wifi Bluetooth
Sorties Audio	Mini-HDMI	Stéréo Jak 3,5 sortie son 5.1 sur la prise HDMI				
USB	1 Micro-USB 2.0	1 (2.0)	2 (2.0)	4 (2.0)	1 (2.0)	2 (2.0) 2 (3.0)
Sorties Vidéo	Mini HDMI et Composite via Soudures	HDMI et Composite via une Price RCA		HDMI et Composite (via un connecteur Jack		2 Micro HDMI et Composite via un connecteur Jack
Poids	9 g	45G	45G	45G	29G	46G

III.3.1.4 Raspberry Pi 3

Le Raspberry Pi 3 modèle B est le premier modèle de la troisième génération de Raspberry Pi. Il a remplacé le Raspberry Pi 2 Model B, Le Raspberry Pi 3 modèle B a un processeur marque broadcom BCM2837 64 bits quadricœur à 1,2 GHz et 1 Go de RAM, un réseau local sans fil et Bluetooth Low Energy (BLE) tandis que le Raspberry pi 3 modèle B+ a un processeur

Broadcom BCM2837B0 Cortex-A53 (ARMv8) 64-bit à 1.4GHz et 1GB deRAM, un Réseau local sans fil 2,4 GHz et 5 GHz IEEE 802.11.b/g/n/ac et Bluetooth 4.2, ainsi 4 x USB ports et BLE. (La figure Ⅲ.14).



Figure Ⅲ.15 Raspberry Pi 3.

III.3.2 Camera OpenMV

La caméra OpenMV est une petite carte microcontrôleur à faible consommation qui vous permet d'implémenter facilement des applications utilisant la vision industrielle dans le monde réel. Vous programmez l'OpenMV dans des scripts Python de haut niveau (avec l'aimable autorisation du système d'exploitation Micro Python) au lieu de C et C++. Cela facilite la gestion des sorties complexes des algorithmes de vision industrielle et le travail avec des structures de données de haut niveau. Mais, il y a toujours un contrôle total sur OpenMV et ses broches d'E/S en Python. Vous pouvez facilement déclencher la prise de photos et de vidéos sur des événements externes ou exécuter des algorithmes de vision industrielle pour déterminer comment contrôler vos broches d'E/S.[29]

OpenMV H7 est programmable par le biais de scripts Python avec l'OS Micro Python. Il bénéficie d'un environnement de développement, OpenMV IDE, téléchargeable sur Linux ou Windows. Cet IDE permet le téléchargement et l'exécution de vos scripts Python de haut niveau,

ainsi que l'affichage des images visualisées par la caméra et la modification des paramètres de visualisation.

Compatible Arduino et Raspberry Pi, le module caméra OpenMV H7 intègre de nombreuses entrées/sorties lui permettant de piloter lui-même vos applications (ADC/DAC, PWM, port I2C, série et SPI, etc.). (Figure Ⅲ.15.)



Figure Ⅲ.16 Camera OpenMV h7.

III.3.2.1 Spécifications techniques de la caméra OpenMV

- Microcontrôleur ARM Cortex-M7 32 bits @ 400 MHz.
- Largeur de bande: 1 MB (SRAM), 2 MB (Flash).
- Unité à virgule flottante (UVF) double précision.
- Instructions DSP complètes, encodage matériel en JPEG.
- Connexions: 2 x UART, 2 x I2C, 1 x SPI, 1 CAN, 3 TIM/PWM.
- 1 port USB Full Speed pour la programmation.
- 1 LED RGB et 2 LED IR embarquées.
- 1 port pour carte micro SD (jusqu'à 64 Go/s).
- Régulateur de commutation haute performance (sortie 1 A).
- Régulateur de tension LDO à faible bruit pour l'alimentation analogique du capteur.
- Connecteur pour batterie LiPo.
- Consommation: moins de 150 mA.

- Multiples capteurs pris en charge : modules capteurs d'images OV7725 640 x 480 et MT9V034, modules capteurs d'images thermiques FLIR 1, 2 et 3.

III.4 Conclusion

Ce chapitre a été pour nous l'occasion de présenter les outils hardware et software utilisés pour la conception et l'implémentation du système traité.

Dans le chapitre suivant, nous allons expliquer en détails les étapes de développement de notre système de détection et classifications des maladies.

Chapitre IV

Réalisation et Implémentation

IV.1 Introduction

L'identification des maladies des plantes est la clé pour prévenir les pertes de rendement et de quantité du produit agricole. Les études sur les maladies des plantes signifient des études des modèles visuels qui peuvent être observés sur une plante. La surveillance de la santé et la détection des maladies sur les plantes sont très critiques pour l'agriculture. C'est très difficile de surveiller manuellement les maladies des plantes. Cela demande énorme quantité de travail, expertise dans les maladies des plantes, et nécessitent également le temps de traitement excessif. La détection des maladies implique des étapes sont l'acquisition de l'image, le prétraitement, la segmentation, l'extraction et la classification des images.

Dans ce chapitre, nous présentons un aperçu sur l'architecture générale de notre système et aussi l'algorithme CNN utilisé dans notre travail. Finalement, nous allons acheminer notre travail par l'environnement logiciel qui assure la réalisation de notre plateforme électronique et finaliser notre projet par l'obtention de différents résultats de tests.

IV.2 Description de l'architecture du système proposé

L'architecture générale de système proposé est illustrée par la figure Ⅳ.1, elle est divisée en trois parties :

- Première partie (Camera) : permet de capturer l'image de la feuille de la plante à travers d'une caméra et la transférer vers l'unité de traitement.
- Deuxième partie (Rasberry) : au niveau de cette partie, nous avons développé un système intelligent à base de CNN qui permet de faire le traitement et la classification des images capturés.
- Troisième partie (Afficheur) : pour afficher les décisions de block précédent (CNN).

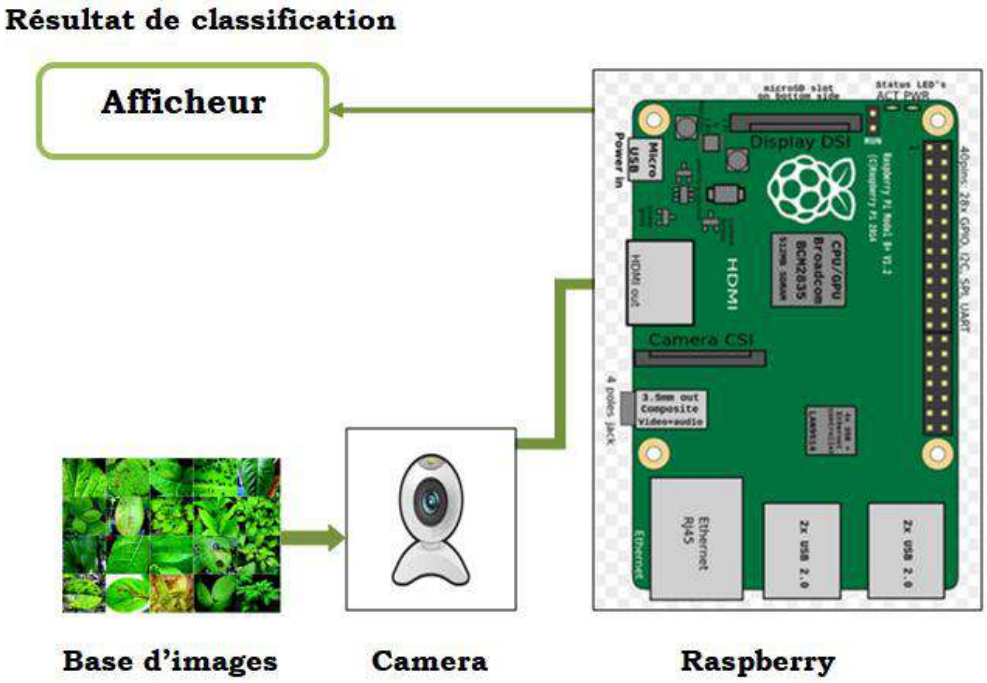


Figure Ⅳ.1 Stricture générale du système proposé.

Figure Ⅳ.2 montre l'organigramme de système proposé.

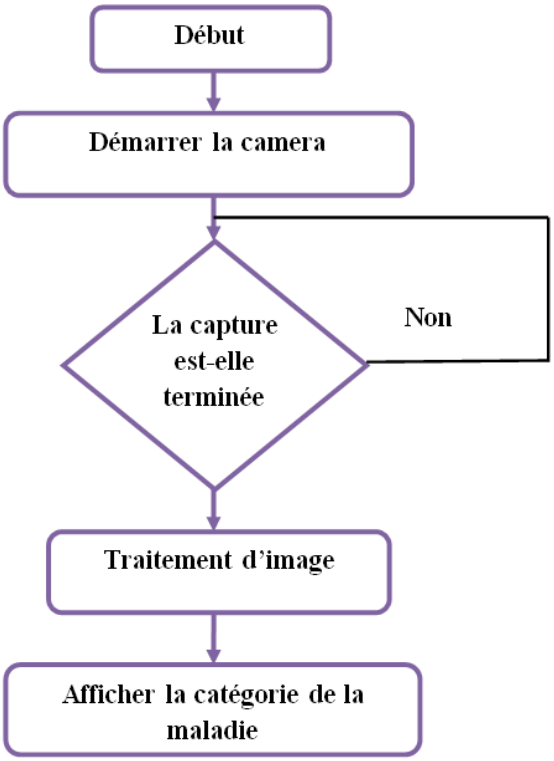


Figure Ⅳ.2 Diagramme de détection des maladies.

IV.3 Conception de système proposé

IV.3.1 Equipment utilise

Deux lap-top ont été utilisés pour l'ensemble du processus de formation et de test du modèle de détection des maladies des plantes décrit dans notre projet. Chaque itération de formation a pris plusieurs minutes sur ces machines spécifiées dont les caractéristiques de base sont présentées dans le tableau (Tableau Ⅳ.1) suivant.

Tableau Ⅳ.1 Caractéristiques de chaque Lap-Top.

	Caractéristiques	
Hardware et software	**Lap-top 1**	**Lap-top 2**
RAM	4 GB	4 GB
Processeur	Intel Core i5-4300U CPU @ 2.50GHz	Intel Core i3-321700U CPU @ 1,8GHz
Système d'exploitation	Windows 10 Professionnel 64 bits	Windows7 Professionnel 64 bits

Pour la partie de programmation, nous avons fait appel à la plate-forme Google Colab. Le choix de cette plateforme est lié au facilité d'utilisation ainsi qu'il est pratique (Par exemple quand nous avons besoin de n'importe quelle bibliothèque, nous l'allons importer. Par contre, chaque librairie doit être installée dans l'anaconda cela Ça perd beaucoup de temps). Il est aussi plus rapide surtout quand nous activons le GPU. Google colab est accordée beaucoup plus avec la connexion.

IV.3.2 Description de la base de données

Nous allons parler sur la base de données utilisée dans la phase de l'apprentissage. C'est une référentielle ouverte qui contient en total quatorze classes (Pepper_bell_Bacterial_spot, Pepper_bell_healthy, Potato_Early_blight,

Potato_healthy, Potato_Late_blight, Tomato_Target_Spot, Tomato_Tomato_mosaic_virus, Tomato_Tomato_YellowLeaf_Curl_Virus, Tomato_Bacterial_spot, Tomato_Early_blight, Tomato_healthy, Tomato_Late_blight, Tomato_Leaf_Mold, Tomato_Septoria_leaf_spot, Tomato_Spider_mites_Two_spotted_spider_mite) de 20639 images. Une fois que cette base de données est téléchargée à l'aide de compte kaggle [https://www.kaggle.com/emmarex/plant-disease-detection-using-keras]. Nous l'organisons et structurons dans un répertoire (nommé PlantVillage). Dans ce dossier PlantVillage, les dix classes du plante de tomate ont été prend en considération, chaque classe a un sous-répertoire qui contient l'ensemble d'images, ils sont nommés de 0 à 9 (Tableau Ⅳ.2). Au total, nous avons utilisé 16012 images de 256×256 pixels en format JPG binaire.

L'application du CNN à associer aux images une étiquette (Labels en anglais). Comme l'intervalle de la probabilité est compris entre [0 1] et que l'intervalle de valeurs des pixels de l'image est compris entre [0 255], nous avons devisé les valeurs de chaque image par 255. Ceci nous permet d'obtenir une image normalisée dont les valeurs sont comprises dans l'intervalle de [0 1].

Enfin, nous précisons les classes de chaque image. Dans notre cas, nous disposons de dix classes. De ce fait, nous créant vecteur de dix éléments. La procédure pour créer ce vecteur binaire est simple. La classe 1 en codée comme suite : on met un (1) à le premier élément et les autres sont mis à zéros (0). Dans le cas de la classe 2, on met le deuxième élément à un (1) et les restes des éléments sont mis à zéros (0) et ainsi de suite.

Figure Ⅳ.3 illustre des exemples des classes des images.

Tableau Ⅳ.2 Description de la base de données.

Classe	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
	Tomato Bacterial Spot	Tomato Early blight	Tomato Late blight	Tomato Leaf Mold	Tomato Septoria leaf spot	Tomato Spider mites_Two Spotted Spider_mite	Tomato Target Spot	Tomato YellowLeaf Curl Virus	Tomato mosaic virus	Tomato healthy

Nombre d'images	2127	1000	1909	952	1771	1676	1404	3209	373	1591
Total	16012									

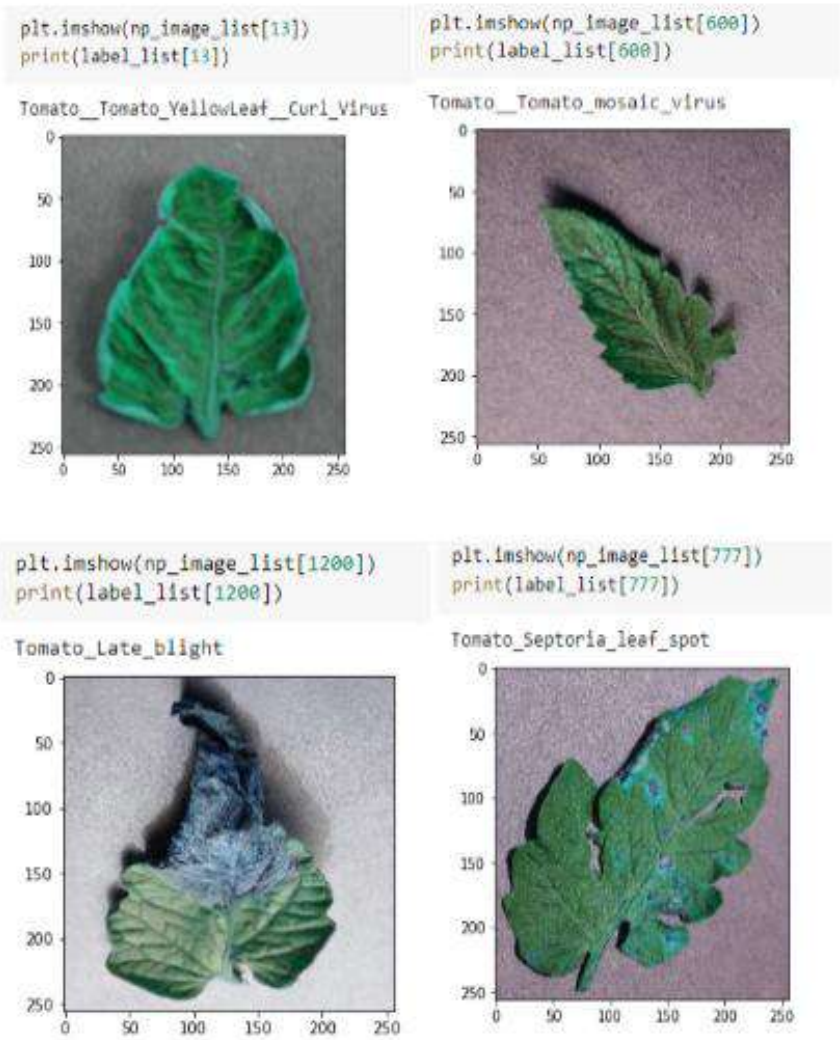


Figure Ⅳ.3 Classes des différentes images.

Notre dataset a été divisée en deux parties :

- **Partie pour l'apprentissage**

Tant que nous avons travaillé sur une seule base de données, nous devons la scinder en deux, 80% pour faire l'entrainement (Train) et 20% pour évaluer ses performances (Test). Dans notre cas, nous avons utilisé 12810 images pour l'apprentissage et 3202 images pour les tests d'évaluations (Voir quelques échantillons de la base dans la figure Ⅳ.4).

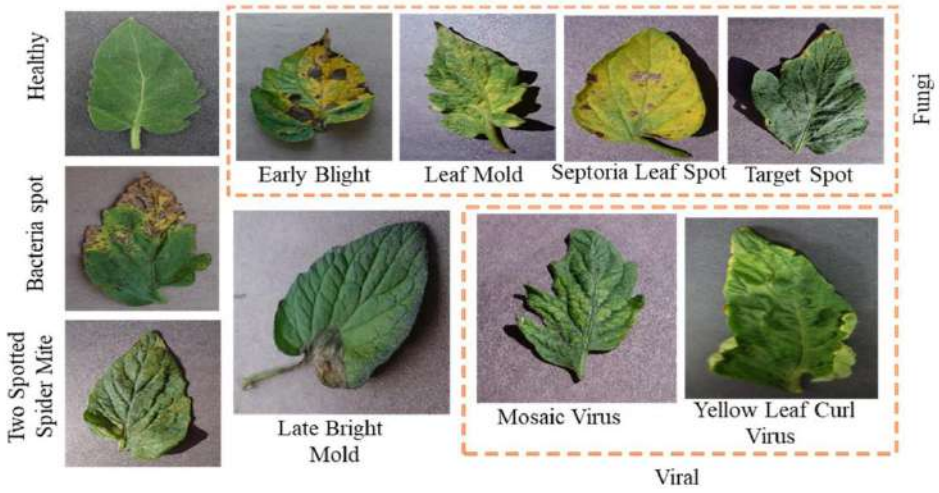


Figure Ⅳ.4 Echantillons de la base de données (PlantVillage).

- **Partie pour la prédiction**

Au début, nous avons créé un répertoire nommée (tomato_pred), qui contient dix images supprimées de chaque catégorie. Les cent images passent par la prédiction (classification des maladies). Le modèle étant déterminé, de nouvelles données peuvent alors être soumises afin d'obtenir le résultat correspondant à la tâche souhaitée. En pratique, l'acquisition d'une photo (d'une feuille de la tomate) par un message indiqué dans le schéma (Figure Ⅳ.5) passé par la transaction numéro (1) afin d'avoir une classification de l'image recevez suivante le modèle généré par la phase d'apprentissage. Suivie par la dernière l'étape de prédiction après le système obtenu un résultat par la voie de transaction numéro (2). La classification consiste à attribuer à l'image entrée une classe d'appartenance parmi les classes déjà vu à la phase d'apprentissage de notre modèle.

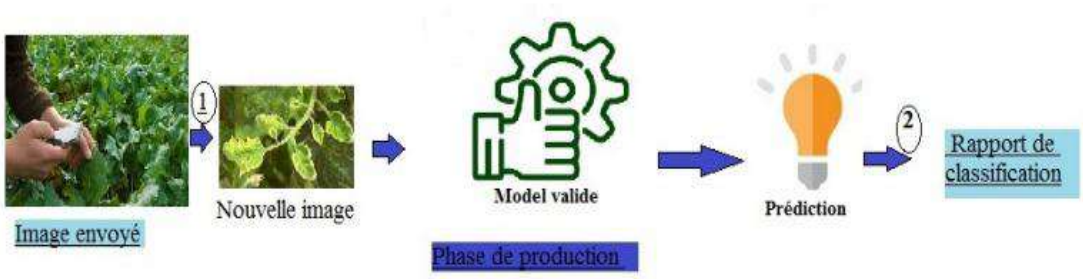


Figure Ⅳ.5 Architecture de production.

IV.3.3 Modèles CNN proposés

Nous dédions cette partie à la description de nos différents tests et expériences. Ainsi que les résultats obtenus en reconnaissant des maladies des plantes. Pour atteindre un taux de précision le plus proche de 100%, nous avons effectué un certain nombre important de tests. Nous avons changé à chaque fois le nombre de couches et la taille des filtres ainsi le nombre d'itérations. Ces expérimentations nous ont permis, d'une part, de mettre à nouveau en évidence les effets de ces paramètres sur l'efficacité du réseau, d'autre part de trouver la configuration optimale à réaliser pour une tâche avec les meilleures performances.

D'abord, nous allons démontrer l'effet du nombre de couches de convolution, puis l'effet de la taille de filtres et enfin l'effet du nombre d'itérations. A noter que toutes les expériences ont été réalisées sur la même base de données avec des images de mêmes tailles 256 * 256 pixels.

IV.3.3.1 Impact du nombre de couche de convolution

Pour démontrer l'influence du nombre de couches de convolution sur la qualité d'apprentissage du réseau CNN, nous avons adopté quatre architectures qui différents par leurs nombre de couches de convolution. Les quatre architectures avec ses résultats obtenus sont regroupées dans le tableau ci-dessous (Tableau Ⅳ.3).

Tableau Ⅳ.3 Influence du nombre de couche de convolution.

Architecture 1	**Architecture 2**	**Architecture 3**	**Architecture 4**
Convolution Filtre 9*9	Convolution Filtre 9*9 Convolution Filtre 9*9	Convolution Filtre 9*9	Convolution Filtre 9*9 Convolution Filtre 9*9
Max pooling			
Convolution Filtre 3*3	Convolution Filtre 3*3	Convolution Filtre 3*3 Convolution Filtre 3*3	Max pooling
Max pooling	Max pooling	Max pooling	Convolution

						Filtre 3*3 Convolution Filtre 3*3	
Convolution Filtre 3*3						Max pooling	
Dropout		Dropout		Dropout		Dropout	
Flatten		Flatten		Flatten		Flatten	
Dense		Dense		Dense		Dense	
Dropout		Dropout		Dropout		Dropout	
Dense		Dense		Dense		Dense	
Sofmax		Sofmax		Sofmax		Sofmax	
Précision (%)	**Temps d'exécution**	**Précision (%)**	**Temps d'exécution**	**Précision (%)**	**Temps d'exécution**	**Précision (%)**	**Temps d'exécution**
64,99	11min 44s	34,40	10min56s	97,06	10 min 9s	75	14min

D'après les résultats de tableau Ⅳ.3, nous pouvons déduire : quand nous utilisons un CNN avec des déférents ordres de couches de convolution, nous observons que l'architecture 3 a une meilleure précision (97,06%). Les performances du réseau augmentent peu en comparaison avec les temps d'exécution. Ceci nous permet de conclure qu'en effectuant plusieurs tests, nous devons trouver la structure optimale qui permet d'obtenir le meilleur compromis entre précision et temps d'exécution.

IV.3.3.2 Impact du nombre de couche de taille de filtre

Dans cette partie, nous voulons souligner l'effet de la taille du filtre sur les performances de réseau CNN. Pour cela, nous avons choisi de fournir des résultats de quatre tailles de filtres : 3*3, 5*5, 7*7 et 9*9.

De plus, nous avons gardé la meilleure architecture depuis les expériences suivantes. Les résultats obtenus avec les quatre nouvelles architectures sont présentés dans le tableau Ⅳ.4 ci-dessous.

Tableau Ⅳ.4 Influence de la taille de filtre.

Architecture 1		Architecture 2		Architecture 3		Architecture 4	
Convolution Filtre 3*3		Convolution Filtre 5*5		Convolution Filtre 7*7		Convolution Filtre 9*9	
Max pooling							
Convolution Filtre 3*3		Convolution Filtre 5*5		Convolution Filtre 7*7		Convolution Filtre 9*9	
Convolution Filtre 3*3		Convolution Filtre 5*5		Convolution Filtre 7*7		Convolution Filtre 9*9	
Max pooling							
Dropout		Dropout		Dropout		Dropout	
Flatten		Flatten		Flatten		Flatten	
Dense		Dense		Dense		Dense	
Dropout		Dropout		Dropout		Dropout	
Flatten		Flatten		Flatten		Flatten	
Dense		Dense		Dense		Dense	
Softmax		Softmax		Softmax		Softmax	
Précision (%)	**Temps d'exécution**	**Précision (%)**	**Temps d'exécution**	**Précision (%)**	**Temps d'exécution**	**Précision (%)**	**Temps d'exécution**
83	11 min	78,38	11min 14s	37	11 min 25s	64	11min44s

D'après les résultats de tableau ci-dessus, nous constatons que la meilleure taille de filtre est 3*3. Grâce à ces tests, nous pouvons logiquement conclure que les filtres à grande taille réduisent la précision du CNN.

IV.3.3.3 Impact du nombre d'itération

Dans cette partie, nous voulons souligner l'effet de nombre d'itération sur les performances de réseau CNN. Pour cela, nous avons choisi de fournir des résultats de déférentes itérations. Ils sont présentés dans le tableau Ⅳ.5 ci-dessous.

Tableau Ⅳ.5 Influence de nombre d'itérations.

	Itération			
	20	**25**	**30**	**40**
Précision (%)	97,06	83 ,40	76,20	87
Temps d'exécution	10min 9s	14 min 20s	16min 55s	20 min 37s

D'après les résultats, nous remarquons que le meilleur nombre époque est 20 qui permet d'obtenir une meilleure précision et un temps d'exécution réduit.

IV.3.4 Architecture du CNN la plus performante

Après avoir effectué divers tests précédents et les résultats obtenus, nous avons remarqué que l'architecture 3 c'est la meilleure.

Les différents résultats obtenus sur les modèles sont exprimés en termes de précision d'apprentissage, de validation, de test et erreur et enfin de temps d'exécution. Le temps d'exécution est trop couteux. Ceci revient à la grande dimension de la base ce qui nécessite l'utilisation d'un GPU au lieu d'un CPU. Nous avons remarqué que le modèle 3 a présenté les meilleurs résultats trouvés. Le nombre d'époque et de couches de convolution reflètent ces bons résultats, cependant le temps d'exécution était très couteux (tant que nous avons travaillé dans Google Colab tout dépond la connexion). Grâce au test du modèle 3 comme meilleur modèle, nous avons obtenu une très bonne précision de plus de 97 % des échantillons de classification après 20 époques. Celui-ci est présenté dans les figures sont ci-dessous.

Figure Ⅳ.6 présentation du modèle de réseau de neurones convolutifs. Les différentes étapes sont décrites dans le test.

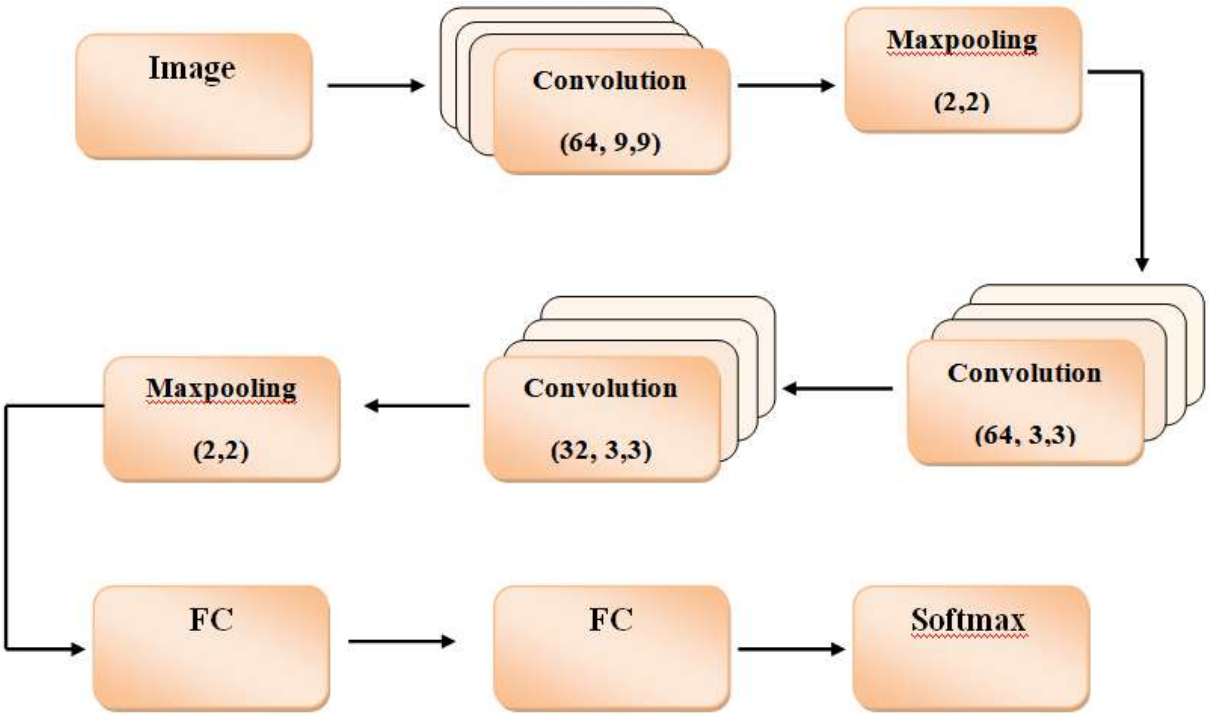


Figure Ⅳ.6 Présentation du modèle de réseau de neurones convolutifs.

IV.3.4.1 Résultats obtenus pour le modèle

Celui-ci est présenté dans les figures sont ci-dessous. L'étape d'exécution de ce modèle comme dans la figure Ⅳ.7.

```
[ ]
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 256, 256, 64)      15616
_________________________________________________________________
activation_12 (Activation)   (None, 256, 256, 64)      0
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 85, 85, 64)        0
_________________________________________________________________
dropout_7 (Dropout)          (None, 85, 85, 64)        0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 83, 83, 64)        36928
_________________________________________________________________
activation_13 (Activation)   (None, 83, 83, 64)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 83, 83, 32)        18464
_________________________________________________________________
activation_14 (Activation)   (None, 83, 83, 32)        0
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 41, 41, 32)        0
_________________________________________________________________
dropout_8 (Dropout)          (None, 41, 41, 32)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 53792)             0
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              55084032
_________________________________________________________________
activation_15 (Activation)   (None, 1024)              0
_________________________________________________________________
batch_normalization_7 (Batch (None, 1024)              4096
_________________________________________________________________
dropout_9 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_5 (Dense)              (None, 10)                10250
_________________________________________________________________
activation_16 (Activation)   (None, 10)                0
=================================================================
Total params: 55,169,386
Trainable params: 55,167,338
Non-trainable params: 2,048
```

Figure Ⅳ.7 Compilation de modèle 3.

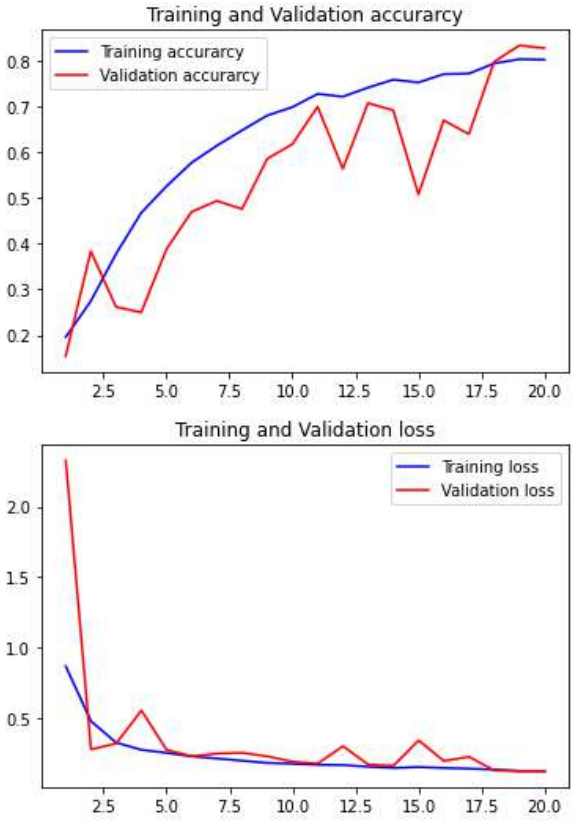


Figure Ⅳ.8 Précision et Erreur pour le modèle .

D'après la Figure III.8 nous avons remarqué que la précision de l'apprentissage et de test augmente avec le nombre d'époque, ceci reflète qu'à chaque époque le modèle apprenne plus d'informations. Si la précision est diminuée alors on aura besoin de plus d'information pour faire apprendre notre modèle et par conséquent on doit augmenter le nombre d'époque et vice versa.

```
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

```
[INFO] Calculating model accuracy
591/591 [==============================] - 1s 2ms/step
Test Accuracy: 97.0671203031556
```

Figure Ⅳ.9 Valeur de précision de modèle 3.

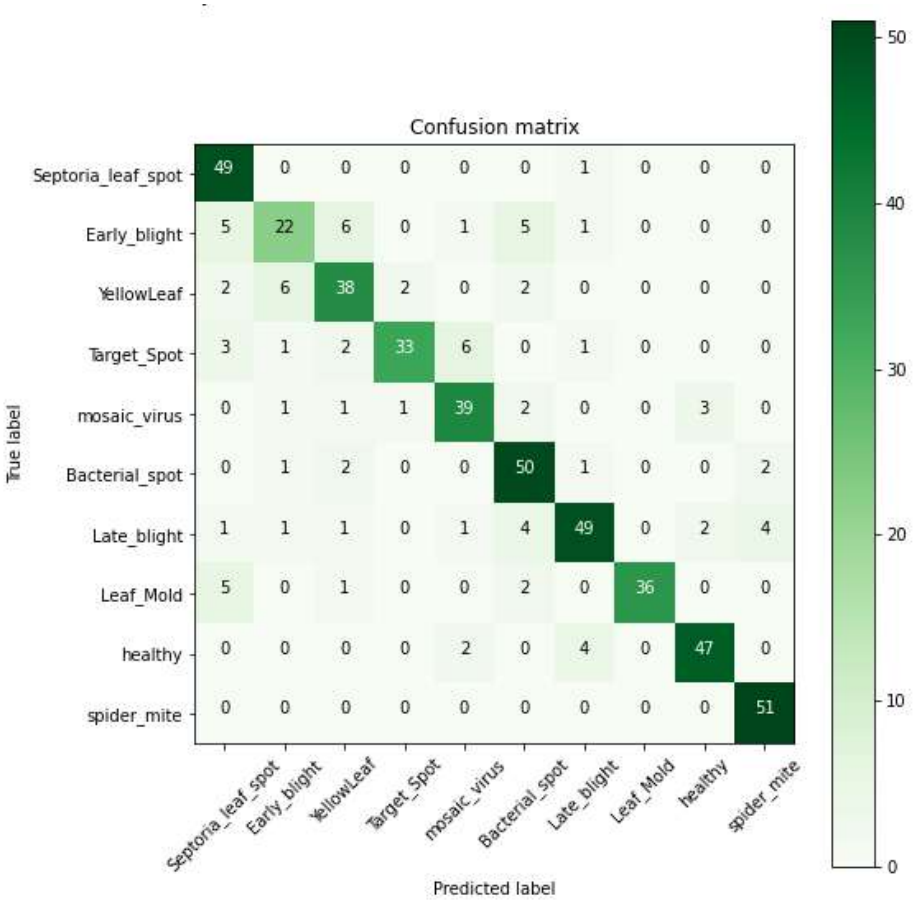


Figure Ⅳ.10 Matrice de confusion.

La matrice de confusion est utilisée pour évaluer les performances de notre modèle, car elle reflète la mesure vraie positive, vraie négative, faux positif et faux négatif. La figure Ⅳ.10 montre de près le placement de ces mesures pour chaque catégorie. La diagonale contient seulement les catégories qui sont bien classées que les autres positions.

Ⅳ.3.4.2 Exemples de tests

Après avoir obtenu un bon modèle, nous sommes passés à l'étape de test. De ce fait nous avons essayé le modèle en Spyder en deux cas. Dans le premier cas (Figure Ⅳ.11) nous avons testé des photos qui se trouvent déjà dans le dossier data de la prédiction ainsi que des photos prises d'internet (Figure Ⅳ.12). Ce teste nous a mené aux même résultats que Google Colab.

Dans cette partie, nous avons utilisé seulement environnement Spider pour faire les tests.

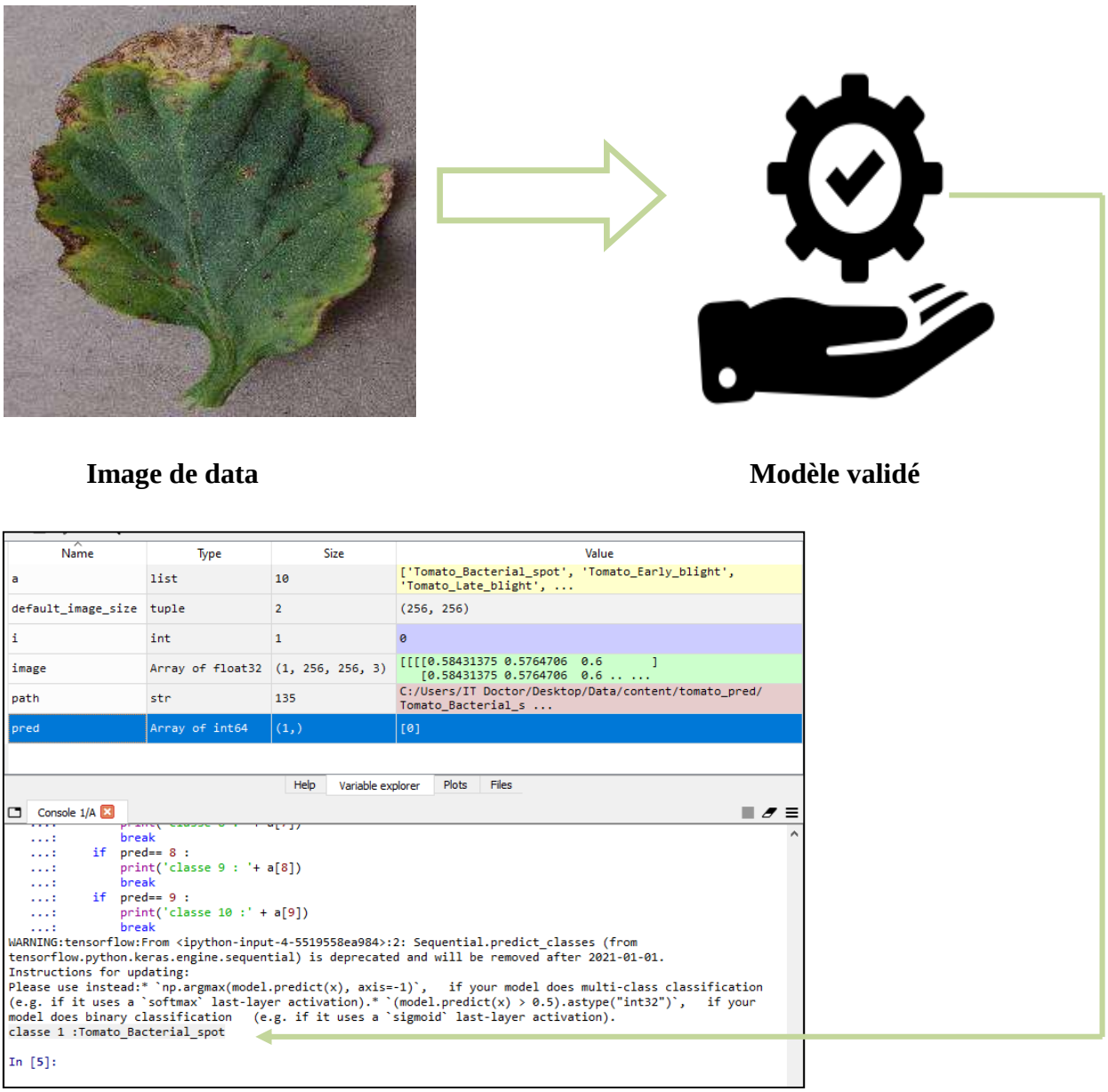


Figure Ⅳ.11 Exemple sur le premier cas.



Image prise internet



Modèle validé

Name	Type	Size	Value
a	list	10	['Tomato_Bacterial_spot', 'Tomato_Early_blight', 'Tomato_Late_blight', ...
default_image_size	tuple	2	(256, 256)
i	int	1	0
image	Array of float32	(1, 256, 256, 3)	[[[[0.01960784 0.5686275 0.28235295] [0.02745098 0.57254905 0.2901 ...
path	str	47	C:/Users/IT Doctor/Desktop/Data/late blight.jpg
pred	Array of int64	(1,)	[2]

Help | Variable explorer | Plots | Files

Console 1/A

```
    ...:            break
classe 3 :Tomato_Late_blight

In [11]:
```

Figure Ⅳ.12 Exemple photo prise d'internet.

IV.4 Implémentation hardware

La figure Ⅳ.13 qui représente la réalisation de hardware et la solution proposé. Pour tester notre système de reconnaissance et de classification des maladies des plantes développé un prototype à base de raspberry, une plante et caméra été utilisés.

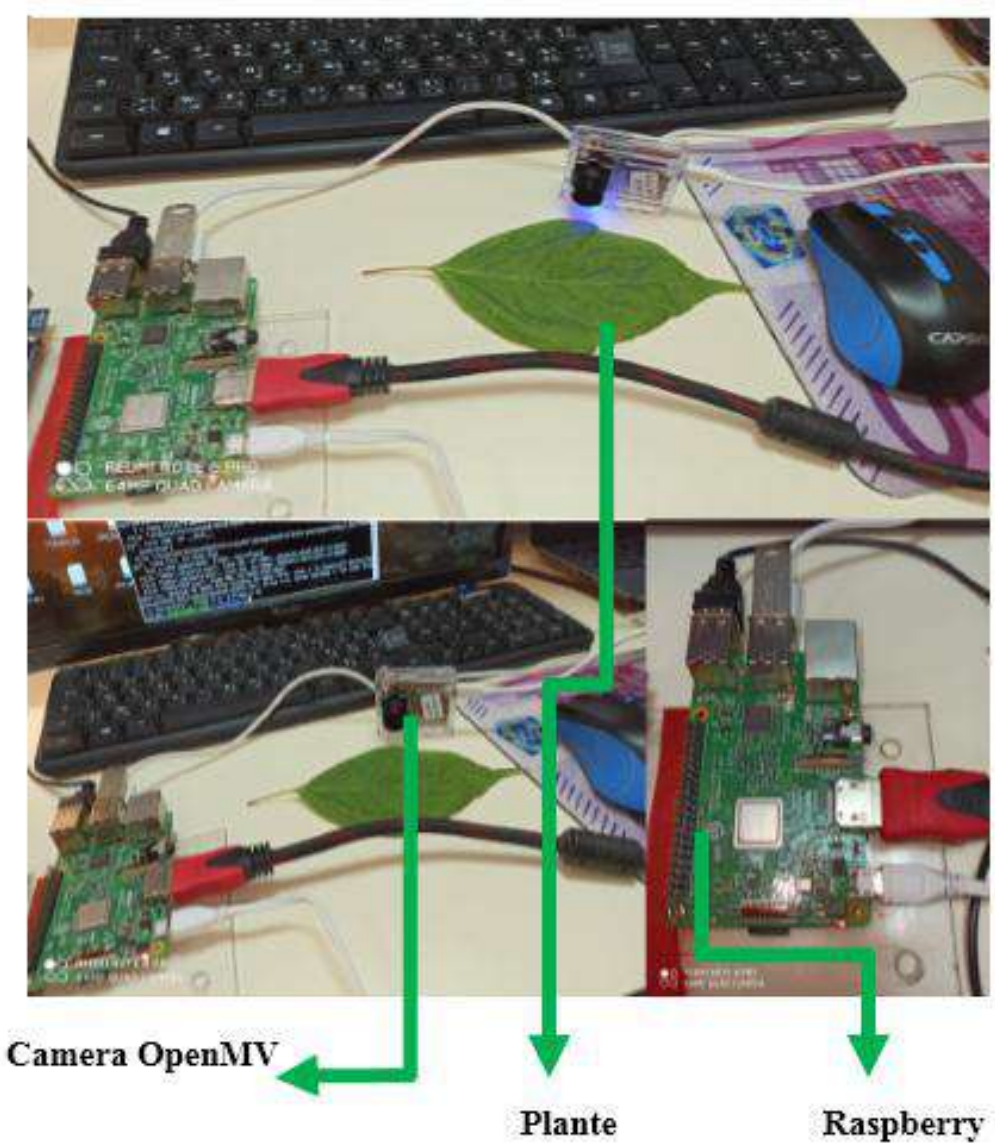


Figure Ⅳ.13 Réalisation hardware.

Pour vous illustrer les capacités de notre système intelligent à reconnaître et classifier les maladies des plantes, nous présentons ci-dessous quelque exemple que nous avons réalisé avec des images des feuilles.

Les figures Ⅳ.14, Ⅳ.15 et Ⅳ.16 ci-dessous représentent les différents tests (test1, test2 et test3) en ordre que nous avons effectués.

Pendant la programmation, une fenêtre apparait nous permettant d'en lire les images. Après nous mettons l'une de images en face de la camera. Suivie par le traitement de cette image qui donne le résultat de la catégorie de maladie de cette dernière.

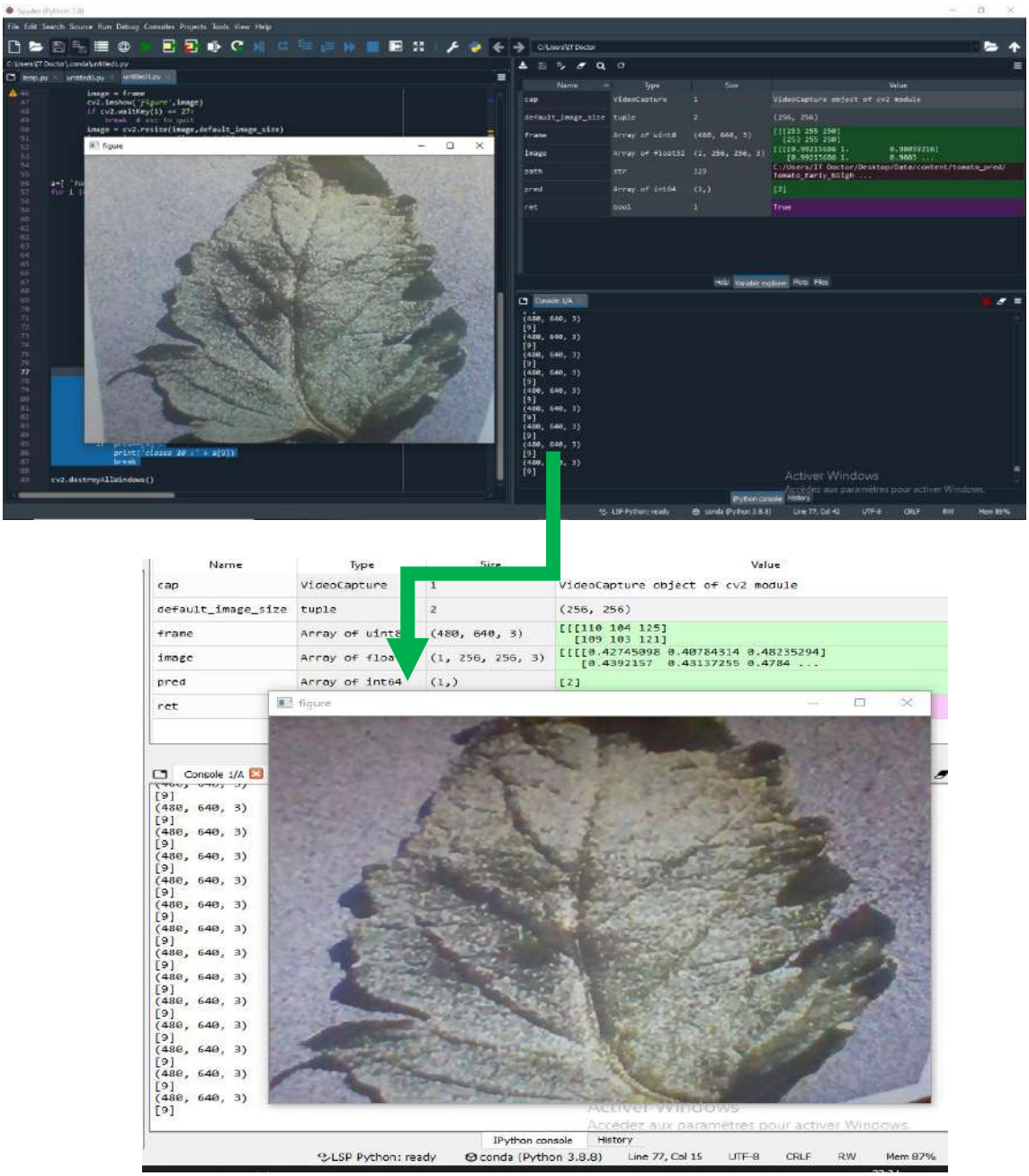


Figure Ⅳ.14 Résultat de test 1.

Dans la section des résultats, il y a un affichage ([9]) qui indique la classe correspondante, c'est tomate healthy. Donc le résultat est validé.

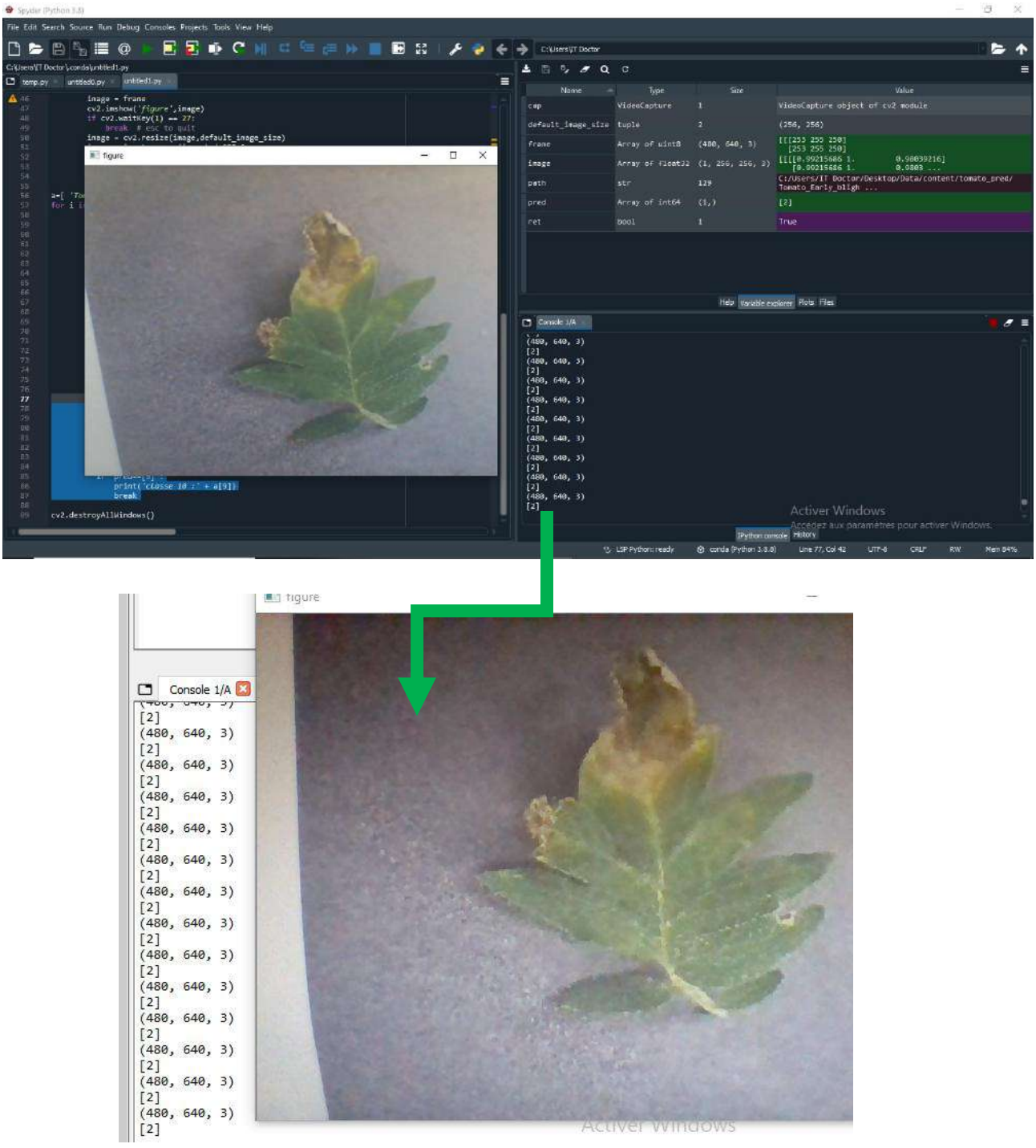


Figure Ⅳ.15 Résultat de test 2.

Dans la section des résultats, il y a un affichage ([2]) qui indique la classe correspondante, c'est tomate late_blight. Donc le résultat est validé.

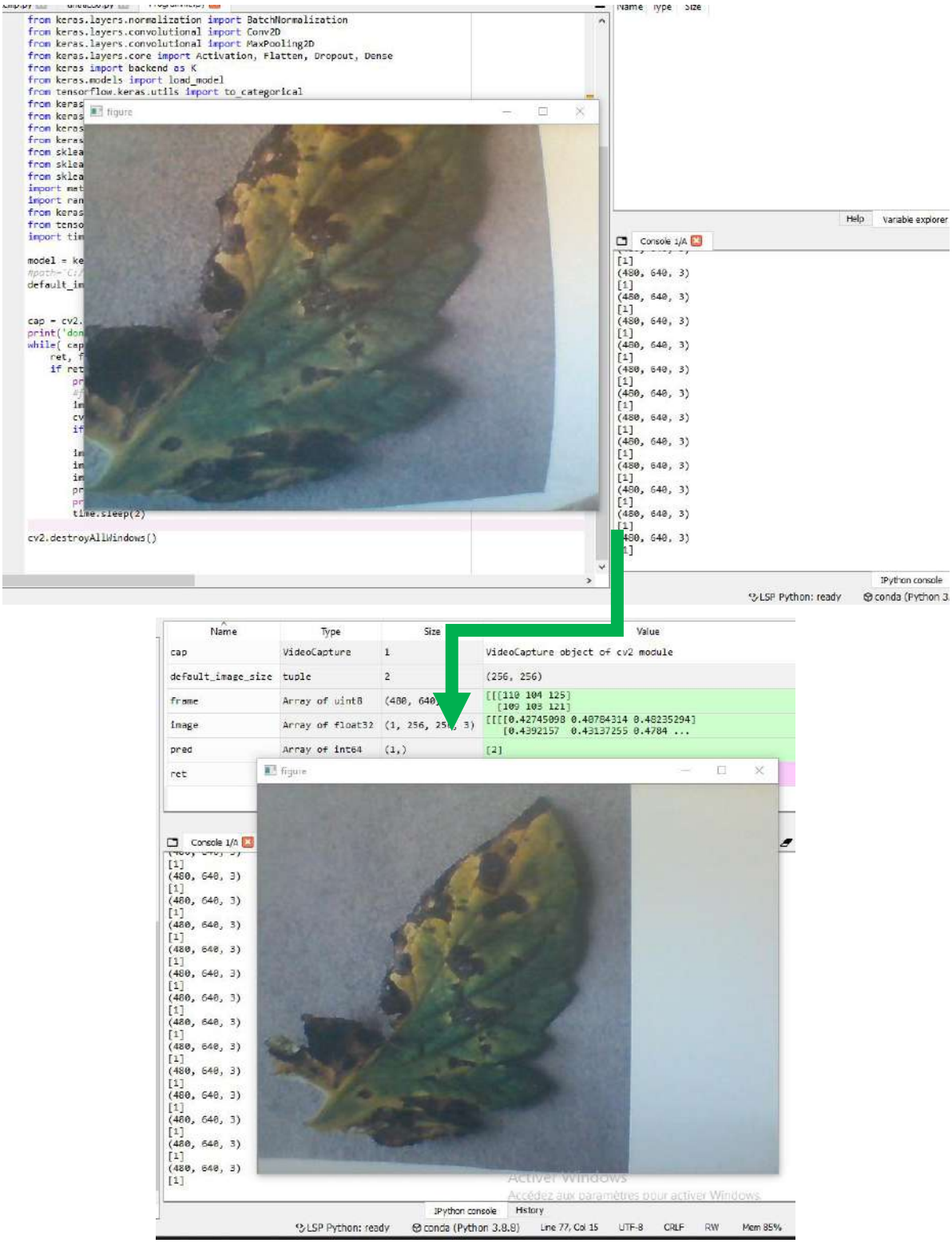


Figure Ⅳ.16 Résultat de test 3.

Dans la section des résultats, il y a un affichage ([2]) qui indique la classe correspondante, c'est tomate early_blight. Donc le résultat est encore validé.

IV.5 Conclusion

Dans ce chapitre nous avons présenté une approche de classification basée sur les réseaux de neurone convolutif. L'objectif que nous sommes fixés est d'obtenir les meilleures performances en termes de précision et de temps d'exécution. Nous avons utilisé plusieurs modèles avec différents architectures et nous avons montré les différents résultats obtenus en termes de précision. La comparaison des résultats trouvés a montré que le nombre d'époque, la taille du filtre et le nombre des couches de convolution, sont des facteurs importants pour l'obtention de meilleurs résultats.

Pour ce faire nous avons implémenté et évalué plusieurs configurations. Nous avons commencé par augmenter le nombre de couches de convolution. Nous avons déduit que la configuration avec 3 couches était la plus adéquate pour la réalisation de notre tâche. Ensuite nous avons également cherché la meilleure taille du filtre qui améliorerait les performances du système. Ainsi avec cette architecture et quelques autres modifications, nous avons obtenu un système intelligent de reconnaissance des maladies des plantes. Ses performances ont été démontrées avec la présentation par des tests.

Conclusion générale

Le projet est investi dans l'agriculture moderne. Notre objectif général c'est la conception et l'implémentation d'un système de la détection et la classification des maladies foliaires. Dans notre travail nous avons montré une approche d'apprentissage pour détecter et classer les maladies des plantes à partir des images des feuilles de la tomate. Plus de 16012 images nous avons utilisés pour le modèle.

Nous avons pris le plaisir de travailler sur ce thème, par ce qu'il consiste l'avenir du domaine économique algérien. Cette idée est un système qui peut aider à améliorer l'agriculture en facilitant les informations effectué par l'application développée.

Comme perspective, les travaux futurs devraient être axés sur :

- Un moyen plus efficace de visualiser les taches de maladie dans les plantes devrait être possible car il réduit les coûts en évitant l'application inutile de fongicide et pesticide herbicide.
- La gravité des maladies des plantes change avec le temps, de sorte que les modèles doivent être améliorés et modifiés pour leur permettre de détecter et de classer les maladies.
- Une étude plus approfondie est nécessaire pour comprendre les facteurs influençant la détection des maladies des plantes, tels que les catégories et la taille des ensembles de données, le taux d'apprentissage, l'éclairage, etc.
- Fournir les bonnes conditions pour prendre des décisions optimales. Il peut également être intégré dans une application mobile et fournit une solution peu coûteuse pour détecter les maladies des plantes en prenant seulement une photo des feuilles de la plante.

Références bibiographiques

[1] A. Abade, P. A. Ferreira, and F. de Barros Vidal, "Plant diseases recognition on images using convolutional neural networks: A systematic review," *Computers and Electronics in Agriculture*, vol. 185. Elsevier B.V., p. 106125, Jun. 01, 2021, doi: 10.1016/j.compag.2021.106125.

[2] "Promo-Cultures - Maladies des plantes." http://www.omafra.gov.on.ca/CropOp/fr/general_agronomics/pest_management/general_pests/diseases.html (accessed Jun. 15, 2021).

[3] A. Arora, N. Khan, S. Gupta, S. Singh, and M. M. Gupta, "Leaf Disease Identification using CNN and Raspberry PI," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, 2019.

[4] "Les avantages de la modernisation de l'agriculture." https://www.cultivonsnosracines.fr/les-avantages-de-la-modernisation-de-lagriculture/ (accessed Jun. 15, 2021).

[5] "Traditional agriculture: what it is, characteristics and performance | Gardening On." https://www.jardineriaon.com/en/agricultura-tradicional.html (accessed Jun. 26, 2021).

[6] "Traditional Agriculture and its impact on the environment." https://www.jagranjosh.com/general-knowledge/traditional-agriculture-and-its-impact-on-the-environment-1518096259-1 (accessed Jun. 15, 2021).

[7] "Le retour aux techniques traditionnelles dans la culture et l'agriculture - Regis Vansnick - 11 mai 2020." https://cursus.edu/articles/43729/le-retour-aux-techniques-traditionnelles-dans-la-culture-et-lagriculture (accessed Jun. 14, 2021).

[8] "L'agriculture de précision : de quoi s'agit-il et quels avantages offre-t-elle ?" https://www.mccormick.it/fr/lagriculture-de-precision/ (accessed Jun. 15, 2021).

[9] "Agriculture de précision • Les Horizons." https://leshorizons.net/cest-quoi-agriculture-de-precision/ (accessed Jun. 22, 2021).

[10] "Les nouvelles technologies au service de l'agriculture | Mutualia." https://www.mutualia.fr/agriculteur/infos/economie-et-societe/news/les-nouvelles-technologies-au-service-de-lagriculture (accessed May 31, 2021).

[11] "Les capteurs connectés pour l'agriculture – Blog techno." http://educagri27.fr/revuedepresse/index.php/2019/03/22/les-capteurs-connectes-pour-lagriculture/ (accessed Jun. 22, 2021).

[12] "De nouvelles technologies pour mieux détecter les maladies des plantes." https://theconversation.com/de-nouvelles-technologies-pour-mieux-detecter-les-maladies-des-plantes-145206 (accessed Jun. 15, 2021).

[13] “Mémoire : Intelligence Artificielle | Béhaviorisme | Sélection naturelle.” https://fr.scribd.com/document/321480908/Memoire-Intelligence-Artificielle (accessed May 28, 2021).

[14] “Qu’est-ce que l’intelligence artificielle (IA)? | Oracle Canada.” https://www.oracle.com/ca-fr/artificial-intelligence/what-is-ai/ (accessed Apr. 11, 2021).

[15] M. Boughaba and B. Boukhris, “L`apprentissage profond (Deep Learning) pour la classification et la recherche d’images par le contenu,” mémoire de master, KASDI MERBAH OUARGLA, 2017.

[16] “(No Title).” https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-Intro-ApprentStat.pdf (accessed Apr. 22, 2021).

[17] “Réseau de neurones artificiels : qu’est-ce que c’est et à quoi ça sert ?” https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition (accessed Apr. 17, 2021).

[18] “Qu’est-ce qu’un réseau neuronal artificiel ? - IONOS.” https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-quun-reseau-neuronal-artificiel/ (accessed Apr. 17, 2021).

[19] “Memoire Online - Identification et commande des systèmes non linéaires - LEMMOU Amira- BELLAKHDAR Khaoukha- LEDJEDEL Adila.” https://www.memoireonline.com/04/12/5750/m_Identification-et-commande-des-systemes-non-lineaires21.html (accessed Apr. 25, 2021).

[20] M. M. Zakaria, “Classification des images avec les réseaux de neurones convolutionnels,” mémoire de master, Université Abou Bakr Belkaid Tlemcen, 2017.

[21] “CS 230 - Pense-bête de réseaux de neurones convolutionnels.” https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels (accessed Apr. 19, 2021).

[22] T. Ikram, “Reconnaissance automatique des plaques d’immatriculation par réseau CNN,” mémoire de master, UNIVERSITE YAHIA FARES DE MEDEA, 2019.

[23] “Installer Anaconda pour python pour vos projets data - Stat4decision.” https://www.stat4decision.com/fr/installer-anaconda-data-python/ (accessed Jun. 24, 2021).

[24] “Getting Started — scikit-learn 0.24.2 documentation.” https://scikit-

learn.org/stable/getting_started.html (accessed Jun. 29, 2021).

[25] “Home — Spyder IDE.” https://www.spyder-ide.org/?fbclid=IwAR32HPkS_t7GslGeudjvxra9PvpfVNZxMpJGNsrV3eeC0I7e6WS1uS02XDc (accessed Jun. 24, 2021).

[26] “Keras: the Python deep learning API.” https://keras.io/ (accessed Jun. 24, 2021).

[27] “Google Colab : Le guide Ultime | Le Data Scientist.” https://ledatascientist.com/google-colab-le-guide-ultime/ (accessed Jun. 24, 2021).

[28] “Qu’est-ce qu’un Raspberry Pi ? Introduction au nano-ordinateur.” https://www.clubic.com/raspberry-pi/article-849782-1-raspberry-pi-introduction-nano-ordinateur.html (accessed Jun. 24, 2021).

[29] “OpenMV Cam M7 | OpenMV.” https://openmv.io/products/openmv-cam-m7 (accessed Jun. 26, 2021).