

République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté de Science et Technologie
Département D'électronique et des Télécommunications



Projet de Fin d'Etudes
Dans le cadre de l'arrêté ministériel 1275
« Un diplôme, une startup ou brevet d'invention »
Pour l'obtention du diplôme de Master en :
Filière : Électronique
Spécialité : Electronique Instrumentation

PLANT DOCTOR

Développement d'un système intelligent pour le diagnostic des maladies des plantes

Présenté Par :

1/MECIRDI YASSINE M2 Électronique Instrumentation
2/REMINI FAROUK M2 Électronique Instrumentation

Devant le jury composé de :

Pr BENCHERIF Kaddour	professeur	U.AinTémouchent	Président
Dr BADIR Benkrelifa Lahouaria	MCB	U.AinTémouchent	Examinateur
Dr SEKKAL Mansouria	MCB	U.AinTémouchent	Encadrante
Dr ABDELLAOUI Hadjira	MCB	U.AinTémouchent	Invitée d'honneur
Dr BEMMOUSSAT Chems Eddine	MCB	U.AinTémouchent	Représentant de l'incubateur
Dr BENOSMANE Morad	MCB	U.AinTémouchent	Représentant de l'incubateur
MR BOUKRAA Omar	Ingénieur Principal	Agriculture Ain temouchent	Partenaire socioéconomique

Année Universitaire :2022/2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

Nous exprimons notre profonde gratitude envers Dieu le Tout-Puissant pour nous avoir accordé la santé et la volonté nécessaires pour entreprendre et mener à bien ce projet de fin d'études.

Tout d'abord, nous souhaitons exprimer notre sincère reconnaissance envers le **Dr. SEKKAL Mansouria** pour son encadrement exceptionnel, sa patience, sa rigueur et sa disponibilité tout au long de la préparation de ce mémoire. Nous n'aurions pas pu réaliser ce travail sans son précieux soutien. Nous tenons également à remercier chaleureusement le **Dr. ABDELLAOUI Hadjira** pour son aide précieuse dans son domaine d'expertise.

Nous souhaitons également exprimer notre gratitude envers les membres du jury qui ont accepté d'évaluer notre travail. Leur expertise et leurs commentaires constructifs ont contribué à améliorer la qualité de notre mémoire.

Nous tenons à remercier tous nos professeurs qui nous ont enseigné au cours de ces années d'études. Leurs connaissances et leur dévouement ont été essentiels pour notre formation.

Enfin, nous adressons un grand merci à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet. Votre soutien et vos encouragements ont été précieux tout au long de ce parcours.

Encore une fois, merci à tous pour votre précieuse contribution à la réussite de ce projet de fin d'études.

DEDICACE

YASSINE MECIRDI.

- *En témoignage d'amour et d'affection, je suis très fière de dédicacer cet ouvrage simple à tous mes proches :*
- *Ma chère maman qui a contribué à ma réussite par son amour, son soutien inconditionnel, ses sacrifices et ses précieux conseils. Je suis reconnaissant pour toute son aide et sa présence continue dans ma vie.*
- *Mon cher père, qui peut retrouver ici avec fierté le résultat des sacrifices et des privations qu'il a consentis au fil des années pour m'aider à avancer dans la vie. Que Dieu le bénisse et que je continue à réussir et lui apporte bonheur et satisfaction.*
- *Mes quatre frères , que Dieu vous accorde la santé, le bonheur et la réussite dans tous vos efforts.*
- *Mes grand parents Alah Yarhamhom SLIMAN et AYACHI , aussi ma grand'mère et tout la famille*
- *En fin mes meilleurs amie KHIEREDIN et Nabil.*

DEDICACE

REMINI Farouk.

Je dédie ce travail à:

Ma mère : Pour ses sacrifices qu'ils font de moi ce que je suis aujourd'hui.

Mes soeurs et frères : Ma grande soeur ASSIA et mes deux petites soeurs NIHAD et ALAA et mes freres qui me soutiennent toujours Khalil Sidahmed et WALID.

RESUME

Notre projet consiste en la création d'une application intelligente basée sur des algorithmes avancés d'intelligence artificielle (IA) visant à détecter et diagnostiquer rapidement les maladies des plantes. Cette application utilise des modèles d'apprentissage profond, notamment des réseaux neuronaux convolutifs (CNN), pour analyser des images de plantes, classifier les maladies et fournir des recommandations de traitement précises. Les utilisateurs peuvent facilement télécharger des images pour une analyse en temps réel, recevoir des résultats instantanés et accéder à une vaste base de connaissances sur les maladies des plantes et leurs solutions. Ce projet de recherche a pour objectif de révolutionner la gestion de la santé des plantes en proposant un outil accessible aux passionnés d'agriculture et de jardinage, contribuant ainsi à améliorer le rendement des cultures et la vitalité des plantes.

Mots-clés : intelligence artificielle, apprentissage profond, Python, maladies des plantes, réseau neuronal convolutif (CNN).

Abstract

Our project involves the creation of an intelligent application based on advanced artificial intelligence (AI) algorithms for the rapid detection and diagnosis of plant diseases. The application uses deep learning models, including convolutional neural networks (CNNs), to analyse plant images, classify diseases and provide accurate treatment recommendations. Users can easily upload images for real-time analysis, receive instant results and access a vast knowledge base on plant diseases and their solutions. The aim of this research project is to revolutionise plant health management by providing an accessible tool for agriculture and gardening enthusiasts, helping to improve crop yields and plant vitality.

Keywords: artificial intelligence, deep learning, Python, plant diseases, convolutional neural network (CNN).

ملخص

يتمثل مشروعنا في إنشاء تطبيق ذكي يعتمد على خوارزميات الذكاء الاصطناعي المتقدمة (AI) التي تهدف إلى اكتشاف وتشخيص الأمراض النباتية بسرعة. اذ يستخدم هذا التطبيق نماذج التعلم العميق، بما في ذلك الشبكات العصبية التلافيفية (CNN)، لتحليل صور النباتات وتصنيف الأمراض وتقديم توصيات علاجية دقيقة. يمكن للمستخدمين بسهولة تحميل الصور لتحليلها في الوقت الفعلي، والحصول على نتائج فورية، والوصول إلى قاعدة معرفية واسعة حول أمراض النباتات وحلولها. يهدف هذا المشروع البحثي إلى إحداث ثورة في إدارة صحة النبات من خلال تقديم أداة يمكن الوصول إليها لعشاق الزراعة والبستنة، مما يساعد على تحسين إنتاجية المحاصيل وحيوية النبات.

الكلمات المفتاحية: الذكاء الاصطناعي، التعلم العميق، بايثون، أمراض النبات، الشبكة العصبية التلافيفية (CNN).

Table des matières

Liste des figures

Liste des tableaux

Introduction

Contexte.....	1
Problématique	1
Chapitre I : Les maladies des plants	
I. Introduction	3
II. Présentation de la feuille de plante :.....	3
II.1 Définition de la feuille de plante :.....	3
II.2 Fonctions de la feuille :	4
II.3 Anatomie de la feuille :	4
II.4 Les maladies des plantes les plus courantes dans l'Algérie :	5
II.4.1 Le mildiou :.....	5
II.4.2 La rouille :.....	6
II.4.3 La tavelure :	7
II.4.4 L'oïdium :.....	8
II.4.5 La cercosporiose :.....	9
II.4.6 La septoriose :	10
II.4.7 La tache angulaire :	11
II.4.8 Aculus olearius :	12
II.4.9 Oeil de paon :	12
II.4.10 L'araignée :	13
II.4.11 La bactériose :	14
II.4.12 La cladosporiose :.....	15
II.4.13 Alternaria solani :	16
II.4.14 Black rot :	17
III. Conclusion.....	18

Table des matières

Chapitre II : le deep learning et machine learning

I.	Introduction :	19
II.	L'apprentissage automatique (Machine Learning) :	19
II.1	Domain d'utilisation :	20
II.2	Les types d'apprentissage automatique :	21
II.2.1	L'apprentissage supervisé :	21
II.2.2	L'apprentissage non supervise :	22
II.2.3	L'apprentissage semi-supervise :	25
II.2.4	L'apprentissage par renforcement :	26
II.3	Les avantages et les inconvénients de l'apprentissage automatique :	27
II.3.1	Les avantages :	27
II.3.2	Les inconvénients :	28
III.	L'apprentissage profond (Deep Learning) :	29
III.2	L'architecture de Deep Learning :	29
III.2.1	Réseaux antagonistes génératifs (GAN) :	29
III.2.2	Principe du réseau antagoniste génératif (Generative Adverserials Networks GAN) :	30
III.2.1.1	Fonctions pertes (loss Function) du discriminateur et du générateur :	31
III.2.3	Autoencoder :	31
III.2.5	Réseaux de Neurones Convolutionnels (CNN) :	36
III.2.5.1	Les couches de réseaux de neurones Convolutionnels :	36
IV.	Conclusion :	39

Chapitre III : Le développement du model intelligent

I.	Introduction :	40
II.	Travail Réalisé :	40
II.1	Création de Les bases de donnes :	40
II.1.1	Fraise :	41
II.1.2	Pomme de terre :	41
II.1.2	Raisin :	42
II.1.3	Pomme :	42
II.1.1	Maïs :	43
II.1.5	Tomate :	44
II.2.1	Pêche :	45
II.1.7	Haricot :	46
III.	Architecture du modèle et processus d'entraînement :	46

Table des matières

III.1	Environnement de travail :	46
III.1.1	GPU :	47
III.1.2	TPU :	48
III.1	Langage utilisé :	48
III.2.1	Domaine d'utilisation :	49
III.2.2	Historiques :	49
IV.	Bibliothèques utilisées :	50
IV.1	TensorFlow :	50
IV.2	Keras :	50
IV.3	Matplotlib :	51
IV.4	CV2 :	51
IV.3	Numpy :	52
IV.	Pandas :	52
V.	La Définition du chemin d'accès à la base de données :	52
VI.	L'augmentation des données :	52
VII.	Les techniques d'augmentation des données que nous avons utilisées :	53
VIII.	Le redimensionnement des images :	55
IX.	La création de Réseau de neurones convolutifs (CNN).....	56
IX.1	Principe de fonctionnement de la Convolution :	56
IX.1.1	Fonctionnement du Dropout :	60
IX.1.2	Formule du Dropout :	60
X.	Explication et analyse de notre CNN (Convolutional neural network) :	65
X.	Les caractéristiques de notre réseau de neurones :	67
XI.	Résultats et discussion :	68
XI.1	Analyse de la progression d'entraînement du system :	68
XI.1.1	Résultats et Discussion : Analyse de la Progression d'Entraînement	68
XII.	Conclusion :	73
Chapitre IV : Présentation de l'application		
I.	Introduction	73
II.	Préparation de l'Environnement de Développement.....	74
II.1	Visual Studio:	74
II.2	Installation de Python et Visual Studio	74
A.	Installation de Python :	74
b.	Installation de Visual Studio :	75

Table des matières

III.	Installation des Bibliothèques Nécessaires.....	75
III.1	Bibliothèques Python Utilisées :	76
III.2	Installation des Bibliothèques :	76
IV.	Développement de l'Interface Utilisateur	77
IV.1	Introduction à l'Interface Utilisateur	78
IV.2	Utilisation de Tkinter.....	78
IV.3	Intégration de la Caméra	78
V.	Chargement et Utilisation des Modèles de Détection des Maladies	79
V.3	Chargement des Modèles	79
V.1	Prédiction des Maladies.....	79
VI.	Personnalisation de l'Interface Utilisateur	80
VI.1	Intégration d'un Logo Personnalisé	80
VI.1	Options de Choix de Plante.....	80
VI.3	Amélioration de l'Expérience Utilisateur	81
VII.	Affichage des Résultats et Améliorations Possibles.....	82
VII.1	Affichage des Résultats de Prédiction.....	82
VIII.	Conclusion :	83
I.	Bibliographie :	73
II.	Webographie	73
I.	Annexe :	Erreur ! Signet non défini.

Liste des figures

Figure 1.1 morphologie de la feuille.....	4
Figure 1.2 Anatomie de la feuille	5
Figure 1.3 Photos de Mildiou de tomate pris par camera	6
Figure 1.4 Photo de la rouille de pommier pris par camera.....	7
Figure 1.5 Photo de la tavelure de pommier pris par camera	8
Figure 1.6 Photo de l'oïdium de pommier pris par camera	9
Figure 1.7 Photo de la cercosporiose de maïs pris par camera	10
Figure 1.8. Photo de la vigne septoriose pris par camera	11
Figure 1.9. Photo de la tache angulaire de haricot pris par camera	11
Figure 1.10. Photo de Aculus oleariu d'olive pris par camera	12
Figure 1.11. Photo de Aculus oleariu d'olive pris par camera	13
Figure 1.12. Photo d'araignée tomate pris par camera	14
Figure 1.13. Photo de bactériose de tomate pris par camera	15
Figure 1.14. Photo de cladosporiose de tomate pris par camera.....	16
Figure 1.15. Photo zoomée de l'alternaria solanie de tomate pris par camera.....	17
Figure 1.16. Photo zoomée de l'alternaria solanie de tomate pris par camera.....	18
Figure .2.1. L'apprentissage automatique.....	21
Figure .2.2. L'apprentissage non supervisé	23
Figure .2.3. Partitionnement des données, ou clustering	24
Figure .2.4. Réduction de dimension	24
Figure .2.5. Un exemple d'apprentissage semi supervise	25
Figure .2.6. Principe de fonctionnent d'apprentissage par renforcement	26
Figure .2.7. Schéma bloc du GAN.....	31
Figure .2.8. Représentation linéaire d'une entrée de données 2D à l'aide de PCA	32
Figure .2.9. les nœuds de l'Autoencoder	33
Figure .2.10. Récurrent Neural Network	34
Figure .2.11. Architecture classique d'un réseau de neurones convolutif (CNN).....	36
Figure .2.12. Exemple de fonctionnent d'un filtre Convolutionnels 3x3	37
Figure .2.13. Échantillon de regroupement maximal avec filtre 2 x 2.....	38
Figure .2.14. Fully connected layer	38
Figure .3.1 Kaggle le guide des recherches en deep learning.....	47
Figure .3.2 La vitesse de traitement :GPU et TPU surperformant le CPU dans l'augmentation des données	48

Liste des figures

Figure .3.3 La déclaration des bibliothèques nécessaire.....	52
Figure .3.4 Définir le chemin pour le model du frais.....	52
Figure .3.5 L'augmentation des donnes	52
Figure .3.6 L'image original (sans effets)	53
Figure .3.7 L'image après effets appliqué	54
Figure .3.8 Redimensionnent les images de train	55
Figure .3.9 Redimensionnent les images de test.....	55
Figure .3.10 Le fonctionnement de CONV2D en utilisant une image 4x4 et un noyau 2x2.....	58
Figure .3.11 Fonctionnement simplifié de la technologie Maxpooling	59
Figure .3.12 L'effet d'application du Dropout sur le réseau neuronal	61
Figure .3.13 Transformation d'une matrice multidimensionnelle en un vecteur à une dimension	62
Figure .3.14 Les couches Dense	63
Figure .3.15 L'architecture de réseau de neurones convolutifs	65
Figure .3.16 L'évolution de la précision du modèle.....	68
Figure .3.17 Test sur les feuilles de raisin	71
Figure.3.18 Test sur les feuilles de Fraise.....	72
Figure.4.1. logo de python et visual studio	75
Figure.4.2. invite de commande (ou terminal).....	77
Figure 4.3. installation des bibliothèques sur l'invite de commande.....	77
Figure 4.4. Vérification des bibliothèques sur le mode interactif du python.....	77
Figure 4.5.intégration la fonctionnalité de capture d'image dans notre application.....	79
Figure 4.6. Chargement des modèles.....	79
Figure4.7. logo de notre application.....	81
Figure 4.8.Options de Choix de Plante.....	81
Figure 4.9.application plant doctor.....	82
Figure 4.10. affichage des résultats de prédiction.....	83

Liste des tableaux

Tableau 3.1 Paramètres de base de données fraise	41
Tableau 3.2 Paramètres de base de données pmme de terre	41
Tableau 3.3 Paramètres de base de données raisin	42
Tableau 3.4 Paramètres de base de données pomme	43
Tableau 3.5 Paramètres de base de données mais.....	44
Tableau 3.6 Paramètres de base de données tomate	45
Tableau 3.7 Paramètres de base de données peche.....	46
Tableau 3.8 Paramètres de base de données haricot	46
Tableau 3.9. Les caractéristiques de notre réseau de neurones pour différentes variétés végétales	68
Tableau 3.10. taux de reconnaissance pour chaque type de plante.....	71

Introduction

La sécurité alimentaire mondiale repose en grande partie sur la santé des cultures agricoles. Cependant, les maladies des plantes demeurent une menace persistante qui peut entraîner des pertes de récoltes considérables. Pour faire face à ce défi, il est essentiel de développer des méthodes de détection précoce et de diagnostic précis des maladies des plantes. Dans cette perspective, les avancées dans le domaine de l'apprentissage profond (deep learning) offrent des opportunités prometteuses pour automatiser la détection des maladies des plantes de manière efficace et précise.

Contexte

Les maladies des plantes sont un problème mondial qui affecte la production agricole, la sécurité alimentaire et l'économie agricole. La détection précoce des maladies est cruciale pour la mise en œuvre rapide de mesures de contrôle et de prévention. Cependant, la détection manuelle des symptômes de maladies des plantes peut être fastidieuse, sujette aux erreurs humaines et nécessite une expertise considérable. C'est là qu'intervient la technologie de l'apprentissage profond, qui permet de développer des systèmes de détection automatisée basés sur des modèles de réseaux de neurones profonds.

Problématique

La détection précise des maladies des plantes à l'aide de l'apprentissage profond présente des défis complexes liés à la variabilité des symptômes, aux conditions environnementales changeantes et à la diversité des espèces végétales. Cette recherche s'attaque à la problématique suivante : comment mettre en place un système de détection des maladies des plantes basé sur l'apprentissage profond qui puisse fournir des résultats précis et fiables dans des conditions réelles de terrain ?

Objectifs

L'objectif principal de ce mémoire est de concevoir, développer et évaluer un système de détection automatisée des maladies des plantes en utilisant des techniques avancées d'apprentissage profond.

Les objectifs spécifiques sont les suivants :

1. Réaliser une revue approfondie des maladies des plantes, de leurs symptômes et des méthodes de détection existantes.

2. Explorer les fondements théoriques de l'apprentissage profond et son application potentielle dans la détection des maladies des plantes.
3. Concevoir et mettre en œuvre un modèle de deep learning pour la détection des maladies des plantes.
4. Évaluer les performances du modèle sur des jeux de données réels.
5. Développer une application pratique basée sur Python et Visual Studio pour mettre en œuvre le modèle de détection.

Structure du Mémoire

Ce mémoire est structuré en quatre chapitres principaux :

- Le chapitre 1 offre un aperçu des maladies des plantes, de leurs impacts et de l'importance de la détection précoce.
- Le chapitre 2 explore en détail les concepts et les techniques de l'apprentissage profond.
- Le chapitre 3 décrit le travail réalisé, y compris la conception du modèle, la collecte de données, les expériences menées et les résultats obtenus.
- Le chapitre 4 présente l'application pratique développée pour la détection automatisée des maladies des plantes, en mettant en lumière les bénéfices potentiels de cette technologie.

Ce mémoire vise à contribuer à l'amélioration de la gestion des maladies des plantes en utilisant des approches innovantes basées sur l'apprentissage profond. En combinant des connaissances en agronomie, en informatique et en intelligence artificielle, cette recherche ouvre de nouvelles perspectives pour la protection des cultures et la sécurité alimentaire

Chapitre I
LES MALADIES DES PLANTS

I. Introduction :

Les maladies de plantes constituent l'un des principaux défis pour l'agriculture, l'environnement et l'économie en Algérie. Ces maladies peuvent être causées par différents agents pathogènes, tels que des bactéries, des champignons, des virus ou des nématodes, qui infectent les tissus des feuilles et peuvent entraîner des dommages considérables aux cultures. Les maladies des plantes peuvent se manifester par une grande variété de symptômes, tels que des taches, des nécroses, des décolorations, des malformations et des dessèchements, qui peuvent réduire la croissance et la productivité des plantes. En outre, ces maladies peuvent se propager rapidement et affecter de vastes étendues de cultures, entraînant des pertes économiques importantes pour les agriculteurs et des conséquences négatives pour la sécurité alimentaire et la santé humaine. C'est pourquoi la prévention, le diagnostic et le traitement des maladies des plantes sont devenus des enjeux majeurs pour la recherche agronomique et la gestion des ressources naturelles. Dans cette optique, cette mémoire de master se propose d'étudier la possibilité de détecter les différentes maladies des plantes à l'aide d'un système intelligent équipé avec une caméra.

II. Présentation de la feuille de plante :

II.1 Définition de la feuille de plante :

Les feuilles sont des organes végétaux verts plats attachés aux tiges ou aux branches. Elles sont responsables de la photosynthèse, c'est le processus par lequel la plante utilise l'énergie solaire pour convertir le dioxyde de carbone en oxygène et en sucres simples comme le glucose pour produire leur propre nourriture. Les feuilles sont composées du limbe (le prolongement plat de la feuille), du pétiole (la partie qui relie la feuille à la tige) et des nervures (les nervures qui transportent la sève dans la feuille).

Morphologie de la feuille :

La feuille présente une grande variabilité morphologique, avec des formes, des tailles, des textures et des couleurs différentes selon les espèces. Les feuilles peuvent être simples (composées d'une seule feuille) ou composées (composées de plusieurs folioles), et leur taille peut varier de quelques millimètres à plus d'un mètre de longueur. Les feuilles peuvent également avoir différentes

textures, allant du velu au lisse, et peuvent présenter différentes couleurs en fonction des pigments présents dans les cellules de la feuille. [8]

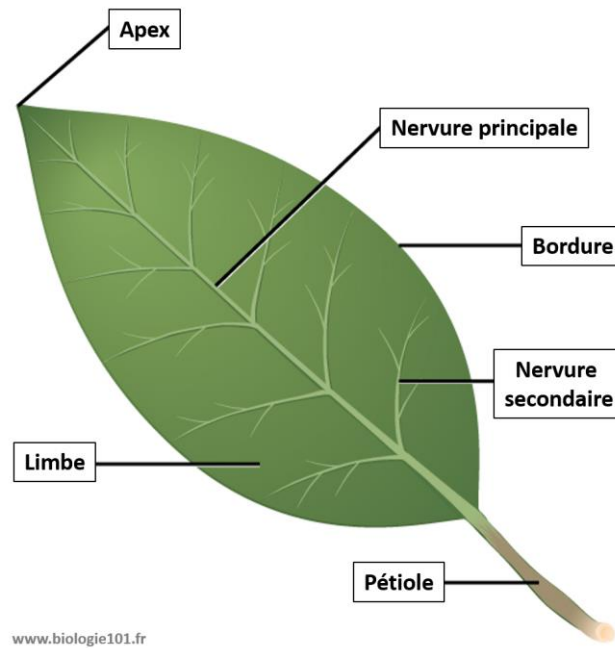


Figure 1.1 morphologie de la feuille. [W17]

II.2 Fonctions de la feuille :

La feuille est un organe important de la plante qui est responsable de la photosynthèse, processus par lequel la plante produit de la nourriture en utilisant de la lumière solaire, du dioxyde de carbone et de l'eau. La photosynthèse se produit dans les chloroplastes, de petits organites présents dans les cellules de la feuille. [W8]

Lors de la photosynthèse, la lumière du soleil est captée par les pigments chlorophylliens présents dans les chloroplastes. L'énergie lumineuse est ensuite utilisée pour convertir le dioxyde de carbone et l'eau en glucose, un sucre simple utilisé par la plante comme source d'énergie. L'oxygène est également produit comme sous-produit de ce processus. [W8]

En plus de la photosynthèse, la feuille est également impliquée dans la respiration, la régulation de l'eau et de la température, ainsi que la synthèse de certaines hormones végétales. Les stomates, petites structures trouvées sur la surface de la feuille, régulent la quantité d'eau qui est libérée de la plante et la quantité de dioxyde de carbone qui est absorbée pour la photosynthèse.

II.3 Anatomie de la feuille :

La feuille est composée de plusieurs types de tissus, chacun ayant des fonctions spécifiques. L'épiderme est la couche externe de la feuille, qui protège la plante contre les dommages physiques, la perte d'eau et les attaques de pathogènes. Le mésophylle est la couche interne de la feuille où se déroule la photosynthèse, et est composé de deux types de cellules : les cellules palissadiques, qui sont disposées en colonnes verticales, et les cellules spongieuses, qui sont plus lâches et permettent la circulation de l'air. Les nervures sont les veines qui transportent la sève dans la feuille, apportant de l'eau et des nutriments aux cellules de la feuille et permettant le transport des sucres produits par la photosynthèse vers le reste de la plante. Les stomates, quant à eux, sont des pores situés sur l'épiderme de la feuille, qui permettent les échanges gazeux entre la plante et l'atmosphère. [8]

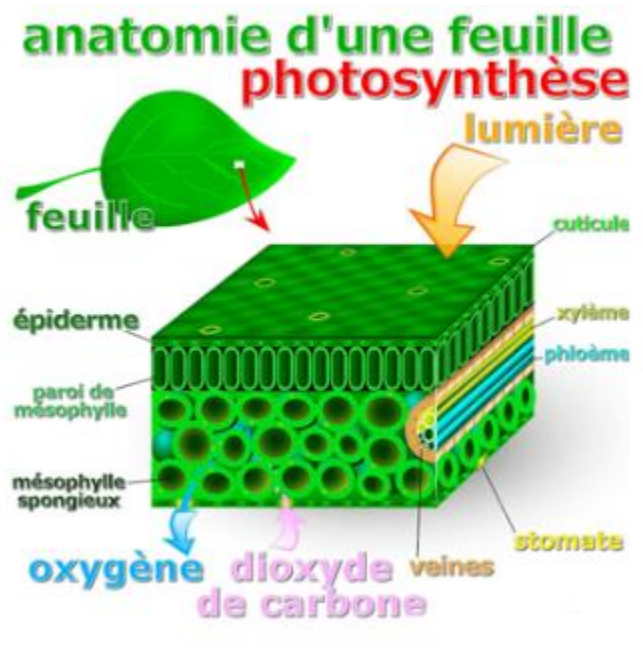


Figure 1.2 Anatomie de la feuille. [W16]

II.4 Les maladies des plantes les plus courantes dans l'Algérie :

Nous ne verrons que les maladies qui apparaissent sur la feuille de la plante pour que notre system intelligent peut les détecter à travers la photo de feuille pris par la caméra.

II.4.1 Le mildiou :

Le mildiou est une maladie cryptogamique qui affecte principalement les plantes cultivées telles que les tomates, les pommes de terre, les vignes et les fraisiers. Elle est causée par un champignon

appelé *Phytophthora infestans*. Les symptômes du mildiou comprennent des taches brunes ou violettes sur les feuilles, des taches d'eau sur les fruits et un flétrissement rapide des plantes.

Le mildiou peut se propager rapidement dans des conditions chaudes et humides, ce qui le rend particulièrement prévalent pendant les mois d'été. Les méthodes de prévention incluent l'utilisation de variétés résistantes, l'arrosage des plantes le matin plutôt que le soir, et la rotation des cultures pour éviter de planter des plantes sensibles dans la même zone chaque année. [9]



Figure 1.3 Photos de Mildiou de tomate pris par camera

II.4.2 La rouille :

La rouille des feuilles est une maladie fongique qui affecte la plante et se manifeste par des taches brunes, rougeâtres ou orange sur les feuilles. Cette maladie est causée par différents types de champignons appartenant au genre *Puccinia*, et peut affecter diverses plantes telles que les céréales, les arbres fruitiers, les légumes et les plantes ornementales.

Les symptômes de la rouille des feuilles comprennent des taches circulaires de couleur rouille sur les feuilles, souvent accompagnées de pustules de spores orangées ou brunâtres. Les feuilles touchées peuvent finir par tomber, affaiblissant la plante et réduisant sa capacité à produire des fruits ou des fleurs.

La rouille des feuilles se propage généralement par temps chaud et humide, et peut être évitée en maintenant une bonne circulation de l'air autour des plantes, en évitant l'arrosage excessif et en retirant régulièrement les feuilles infectées. Les méthodes de traitement comprennent l'utilisation de fongicides, l'élimination des parties de la plante infectées et la rotation des cultures pour éviter de planter des plantes sensibles dans la même zone chaque année.



Figure 1.4 Photo de la rouille de pommier pris par camera

II.4.3 La tavelure :

La maladie de Tavelures, également connue sous le nom de flétrissement vasculaire, est une maladie fongique qui affecte de nombreuses espèces de plantes, notamment les tomates, les bananes, les fraises, les pommes de terre et les roses. Cette maladie est causée par un champignon pathogène appelé *Fusarium oxysporum*, qui pénètre dans les racines de la plante et se propage dans les vaisseaux conducteurs de sève.

L'infection par le champignon conduit à une obstruction des vaisseaux conducteurs de sève, ce qui réduit la capacité de la plante à absorber l'eau et les nutriments. Cela entraîne un flétrissement progressif des feuilles et des tiges, qui peut être accompagné d'une décoloration brune ou noire des vaisseaux conducteurs.

Les plantes atteintes peuvent également présenter des taches foliaires, une pourriture des racines et une décoloration des fruits. La maladie de Tavelures peut se propager par les semences, les plantes infectées, les outils de jardinage contaminés et même les chaussures des personnes qui ont marché sur des sols infectés.

Les mesures de prévention de la maladie de Tavelures incluent l'utilisation de semences certifiées, la rotation des cultures, la désinfection des outils de jardinage et des surfaces de travail, et l'application de fongicides. Il est également important de maintenir une bonne hygiène de culture, en éliminant régulièrement les débris végétaux et en évitant les excès d'humidité. [9]



Figure 1.5 Photo de la tavelure de pommier pris par camera

II.4.4 L'oïdium :

La maladie d'oïdium, également appelée mildiou poudreux, est une maladie fongique qui affecte de nombreuses espèces de plantes, y compris les cultures agricoles et les plantes ornementales. Cette maladie est causée par plusieurs espèces de champignons du genre *Erysiphe*, qui se propagent par les spores aériennes.

Les symptômes de la maladie d'oïdium incluent une couverture blanche ou grisâtre poudreuse sur les feuilles, les tiges et les fleurs de la plante. Cette couverture peut entraîner une décoloration des feuilles, une torsion et un enroulement des feuilles, et une croissance ralentie de la plante. Les fruits peuvent également être affectés, présentant une décoloration et un aspect ratatiné.

La maladie d'oïdium peut être propagée par des facteurs tels que des températures chaudes et sèches, un manque de ventilation, un arrosage excessif, un stress de la plante et des facteurs environnementaux tels que l'humidité élevée. Les mesures de prévention de la maladie d'oïdium comprennent l'utilisation de plantes résistantes, la rotation des cultures, la suppression régulière des feuilles et des branches malades, la réduction de l'humidité, l'arrosage au niveau du sol et l'utilisation d'un fongicide si nécessaire.



Figure 1.6 Photo de l'oïdium de pommier pris par camera

II.4.5 La cercosporiose :

La cercosporiose est une maladie fongique qui affecte de nombreuses plantes, y compris les cultures agricoles telles que le soja, le maïs, le coton et le tabac, ainsi que les plantes ornementales telles que les bégonias et les chrysanthèmes. Cette maladie est causée par plusieurs espèces de champignons du genre *Cercospora*, qui se propagent par les spores aériennes et se développent dans des conditions humides et chaudes.

Les symptômes de la cercosporiose comprennent des taches circulaires ou irrégulières sur les feuilles, les tiges et les fleurs de la plante, qui peuvent varier en couleur de brun clair à brun foncé. Ces taches peuvent s'agrandir et se rejoindre pour couvrir une grande partie de la plante, entraînant une défoliation précoce et une réduction de la croissance de la plante. Les fruits peuvent également être affectés, présentant des taches brunâtres ou noirâtres.

La cercosporiose peut être propagée par des facteurs tels que des températures chaudes et humides, une humidité élevée, un arrosage excessif et des blessures sur la plante. Les mesures de prévention de la cercosporiose comprennent l'utilisation de plantes résistantes, la rotation des cultures, l'élimination régulière des débris végétaux, le maintien d'une bonne ventilation et d'une hygiène de culture, ainsi que l'utilisation de fongicides si nécessaire.



Figure 1 .7 Photo de la cercosporiose de maïs pris par camera

II.4.6 La septoriose :

La septoriose est une maladie fongique qui affecte les plantes, causée par le champignon *Septoria*. Cette maladie se manifeste par des taches foliaires qui se développent sur les feuilles et qui sont caractérisées par des zones brunâtres entourées d'un halo jaune. Les taches peuvent se rejoindre pour former de grandes zones nécrotiques, ce qui peut entraîner la mort de la feuille.

Les symptômes de la septoriose peuvent varier selon la plante infectée, mais les feuilles sont généralement les premières touchées. Les plantes touchées peuvent également montrer des signes de pourriture et de décoloration sur les fruits et les tiges.

La septoriose se propage par les spores fongiques qui sont transportées par l'air, l'eau ou les insectes. Les plantes sont plus susceptibles d'être infectées si elles sont maintenues dans des conditions de forte humidité ou de faible luminosité.

Le traitement de la septoriose peut être effectué par l'utilisation de fongicides ou d'autres produits chimiques, ainsi que par la modification des conditions environnementales pour réduire les risques d'infection. Cependant, la prévention est souvent la meilleure solution, en évitant de planter des plantes sensibles dans des zones de risque élevé et en veillant à la santé générale des plantes.



Figure 1.8. Photo de la vigne septoriose pris par camera

II.4.7 La tache angulaire :

La tache angulaire est une maladie fongique qui affecte les plantes, causée par le champignon *Pseudomonas syringae*. Elle se caractérise par des taches angulaires brun foncé ou noires qui se forment sur les feuilles des plantes, ainsi que sur les tiges et les fruits. Les taches ont souvent une apparence humide et peuvent être entourées d'une zone jaune ou brune.

Les plantes touchées peuvent également présenter des symptômes tels que la chute prématurée des feuilles, une croissance ralentie et une décoloration des tiges et des fruits.

La tache angulaire se propage par les gouttelettes d'eau qui se propagent par l'air ou par la pluie. Les plantes sont plus susceptibles d'être infectées si elles sont maintenues dans des conditions de forte humidité, de faible luminosité ou si elles subissent des blessures.

Le traitement de la tache angulaire peut être effectué par l'utilisation de fongicides ou d'autres produits chimiques, ainsi que par la modification des conditions environnementales pour réduire les risques d'infection. Cependant, la prévention est souvent la meilleure solution, en évitant de planter des plantes sensibles dans des zones de risque élevé et en veillant à la santé générale des plantes.



Figure 1.9. Photo de la tache angulaire de haricot pris par camera

II.4.8 *Aculus olearius* :

Aculus olearius est un acarien qui affecte les oliviers et provoque la maladie appelée l'ériophyose de l'olivier. Cette maladie se caractérise par des feuilles déformées, enroulées et rousses. Les olives peuvent également être affectées, avec des fissures et des taches noires.

Les acariens se nourrissent de la sève des feuilles de l'olivier, ce qui entraîne la déformation des feuilles et la réduction de la production de fruits. Les acariens se propagent facilement d'un arbre à l'autre, ce qui peut entraîner des épidémies dans les plantations d'oliviers.

Le traitement de l'ériophyose de l'olivier peut être effectué par l'utilisation de pesticides ou d'autres produits chimiques, ainsi que par la taille et l'élagage des arbres pour améliorer la circulation de l'air et réduire les risques d'infection. Cependant, la prévention est souvent la meilleure solution, en évitant de planter des oliviers dans des zones de risque élevé et en veillant à la santé générale des arbres.



Figure 1.10. Photo de *Aculus olearius* d'olive pris par camera

II.4.9 *Oeil de paon* :

L'œil de paon est une maladie fongique qui affecte les plantes, en particulier les arbres fruitiers tels que les pêchers, les nectarines, les prunes et les abricots. Elle est causée par le champignon *Taphrina deformans*.

Les symptômes de l'œil de paon incluent des taches circulaires et rouges sur les feuilles, qui finissent par se décolorer en jaune ou en blanc. Les feuilles touchées ont également une texture et une apparence froissée. Les fleurs peuvent être déformées ou ne pas se développer correctement, ce qui entraîne une diminution de la production de fruits.

La maladie se propage par des spores qui sont transportées par le vent et la pluie. Les plantes sont plus susceptibles d'être infectées si elles sont maintenues dans des conditions de forte humidité, de faible luminosité ou si elles subissent des blessures. [9]



Figure 1.11. Photo de Aculus olearii d'olive pris par camera

Le traitement de l'œil de paon peut être effectué par l'utilisation de fongicides ou d'autres produits chimiques, ainsi que par la taille et l'élagage des arbres pour améliorer la circulation de l'air et réduire les risques d'infection. Cependant, la prévention est souvent la meilleure solution, en évitant de planter des plantes sensibles dans des zones de risque élevé et en veillant à la santé générale des plantes.

II.4.10 L'araignée :

La maladie des araignées, également connue sous le nom d'infestation d'araignées rouges, est une maladie causée par les araignées rouges. Ce sont de petits acariens qui se nourrissent de la sève des plantes, en particulier des plantes d'intérieur telles que les orchidées, les hibiscus et les plantes tropicales.

Les araignées rouges laissent souvent des toiles d'araignée sur les plantes et causent des taches jaunes sur les feuilles. Les feuilles peuvent également se flétrir et tomber prématurément, ce qui peut entraîner une réduction de la production de fruits ou de fleurs.

Les araignées rouges se multiplient rapidement et peuvent infester rapidement une plante ou une culture entière. Ils se propagent facilement par le vent, les animaux et les outils de jardinage.

Le traitement de la maladie des araignées peut être effectué en utilisant des produits chimiques tels que des acaricides, ou en utilisant des méthodes plus naturelles telles que l'utilisation d'huiles essentielles ou l'introduction d'insectes prédateurs qui se nourrissent des araignées rouges. Il est

également important de maintenir un niveau d'humidité approprié pour les plantes et de les nettoyer régulièrement pour éliminer les toiles d'araignées et les araignées. [9]



Figure 1.12. Photo d'araignée tomate pris par camera

II.4.11 La bactériose :

Les bactérioses sont des maladies causées par des bactéries pathogènes qui peuvent affecter les plantes. Elles peuvent être transmises par le sol, les semences, les outils de jardinage ou les insectes porteurs de maladies. Les symptômes de la bactériose varient selon le type de plante et la bactérie qui l'infecte, mais peuvent inclure des taches brunes ou noires sur les feuilles, des taches molles sur les fruits et des tiges qui flétrissent. Les bactérioses sont souvent difficiles à contrôler, car il n'existe pas de traitement curatif pour les plantes infectées. Cependant, il est possible de limiter leur propagation en adoptant des pratiques de gestion appropriées. Il est important de retirer les plantes infectées et les parties de plantes touchées pour éviter que la maladie ne se propage à d'autres plantes. Les outils de jardinage doivent être désinfectés avant et après leur utilisation, et les semences doivent être achetées auprès de sources fiables pour éviter de propager des maladies. Il est également possible de prévenir les bactérioses en améliorant les conditions de croissance des plantes. Cela peut inclure la fourniture d'une bonne aération et d'un drainage adéquat du sol, ainsi que la fourniture d'un engrais équilibré pour maintenir une croissance saine des plantes. L'utilisation de variétés de plantes résistantes aux bactérioses peut également être efficace pour prévenir l'infection.



Figure 1.13. Photo de bactériose de tomate pris par camera

II.4.12 La cladosporiose :

La cladosporiose est une maladie fongique qui affecte les plantes. Elle est causée par des champignons du genre *Cladosporium*, qui se développent souvent dans des conditions chaudes et humides. Les plantes les plus couramment touchées par la cladosporiose sont les arbres fruitiers, les légumes, les plantes ornementales et les plantes d'intérieur.

Les symptômes de la cladosporiose peuvent varier selon le type de plante infectée, mais ils comprennent souvent des taches brunes ou noires sur les feuilles, les tiges et les fruits. Les taches peuvent s'étendre et fusionner pour couvrir de grandes parties de la plante. Les feuilles infectées peuvent également tomber prématurément, ce qui peut affecter la production de fruits ou de légumes.

La cladosporiose se propage facilement par les spores fongiques qui peuvent être transportées par le vent, la pluie ou les insectes. Les spores peuvent également survivre dans le sol et infecter les plantes saines la saison suivante.

Le traitement de la cladosporiose peut être effectué en utilisant des fongicides, qui peuvent être appliqués sur les plantes infectées ou en prévention pour éviter l'infection. Il est également important de maintenir un niveau d'humidité approprié pour les plantes, de les arroser au niveau du sol et de les tailler régulièrement pour éliminer les parties infectées. [9]



Figure 1.14. Photo de cladosporiose de tomate pris par camera

II.4.13 Alternaria solani :

L'alternariose est une maladie fongique causée par le champignon *Alternaria solani*. Cette maladie est courante dans les régions chaudes et humides du monde entier, où elle peut causer de graves pertes de récoltes de tomates.

Les symptômes de l'alternariose peuvent varier, mais ils commencent généralement par l'apparition de taches brunes ou noires sur les feuilles inférieures de la plante. Les taches peuvent s'étendre rapidement et finir par couvrir toute la feuille, qui peut finalement tomber prématurément. Les fruits peuvent également être affectés, présentant des taches brunâtres et des zones de pourriture.

L'*Alternaria solani* peut survivre dans le sol pendant de longues périodes et peut être transporté par le vent ou par la pluie. La maladie se propage rapidement dans des conditions chaudes et humides, surtout lorsque les plants de tomates sont étroitement espacés. Les pratiques culturales telles que la rotation des cultures, la désinfection du sol, l'élagage des feuilles infectées et la suppression des débris végétaux peuvent aider à réduire la gravité de l'alternariose.

Le traitement de l'alternariose peut être effectué en utilisant des fongicides, qui doivent être appliqués avant l'apparition de la maladie ou dès les premiers symptômes. Il est important de suivre les instructions du fabricant et de répéter l'application selon les besoins. [9]



Figure 1.15. Photo zoomée de l'alternaria solanie de tomate pris par camera

II.4.14 Black rot :

Le Black Rot est une maladie fongique qui affecte les plantes, notamment les raisins, les pommes, les pêches et les fraises. Elle est causée par le champignon *Guignardia bidwellii* et peut causer des pertes de rendement importantes dans les cultures affectées.

Les symptômes du Black Rot peuvent varier selon les plantes affectées, mais ils commencent généralement par des taches brunes sur les feuilles, qui se propagent rapidement et se transforment en lésions nécrotiques. Les fruits affectés peuvent également présenter des taches brunes qui peuvent devenir des ulcères et pourrir.

Le champignon *Guignardia bidwellii* peut survivre dans les débris de cultures et les tissus des plantes infectées. Il se propage par temps humide et par le vent. Les pratiques culturales telles que la taille appropriée des plantes, la suppression des débris végétaux et la pulvérisation de fongicides peuvent aider à réduire la gravité du Black Rot.

Le traitement du Black Rot peut être effectué en utilisant des fongicides, qui doivent être appliqués avant l'apparition de la maladie ou dès les premiers symptômes. Il est important de suivre les instructions du fabricant et de répéter l'application selon les besoins.



Figure 1.16. Photo zoomée de l'alternaria solanie de tomate pris par camera

III. Conclusion

En conclusion, la feuille de plante est un organe essentiel pour la survie et la croissance de la plante. Elle est responsable de la photosynthèse, processus de production de nourriture pour la plante, et joue également un rôle important dans la régulation de l'eau, de la température et de la synthèse de certaines hormones végétales. La forme et la taille des feuilles varient considérablement d'une espèce à l'autre, mais toutes ont en commun leur rôle crucial dans la physiologie de la plante. En plus de leur importance pour la plante elle-même, les feuilles ont également des implications importantes pour l'environnement, car elles contribuent à la production d'oxygène et à la régulation du dioxyde de carbone dans l'atmosphère. En somme, la feuille de plante est un organe fascinant et fondamental pour la vie sur Terre.

Chapitre II

DEEP LEARNING ET MACHINE LEARNING

I. Introduction :

L'apprentissage profond (Deep Learning) et l'apprentissage automatique (machine Learning) sont deux sous-domaines importants de l'intelligence artificielle (IA) qui sont étroitement liés, mais qui diffèrent dans leur approche de l'analyse des données. L'apprentissage automatique implique l'utilisation d'algorithmes pour analyser les données, identifier des modèles et prendre des décisions sur la base de ces données. Il existe depuis les années 1950 et a connu une croissance et une évolution significatives au cours des dernières décennies.

L'apprentissage profond, quant à lui, est une forme plus avancée d'apprentissage automatique qui implique l'utilisation de réseaux neuronaux artificiels modélisés sur la structure et la fonction du cerveau humain pour analyser des données complexes. Bien que l'apprentissage profond soit un sous-ensemble de l'apprentissage automatique, il diffère de l'apprentissage automatique traditionnel par sa capacité à apprendre automatiquement des représentations de caractéristiques à partir de données brutes, plutôt que de s'appuyer sur l'ingénierie manuelle des caractéristiques.

Malgré leurs différences, l'apprentissage profond et l'apprentissage automatique ont tous deux de nombreuses applications dans divers secteurs, notamment les soins de santé, la finance et la technologie. L'apprentissage automatique est souvent utilisé pour des tâches plus simples telles que la classification et la régression, tandis que l'apprentissage profond est généralement utilisé pour des tâches plus complexes telles que la reconnaissance d'images et de la parole, le traitement du langage naturel et la conduite autonome. [1]

II. L'apprentissage automatique (Machine Learning) :

L'apprentissage automatique est un domaine de l'informatique qui se concentre sur le développement d'algorithmes et de modèles capables d'apprendre des schémas et des idées à partir de données. Le processus d'apprentissage automatique consiste à introduire de grandes quantités de données dans un système informatique et à utiliser des méthodes statistiques pour extraire des modèles et des relations significatifs de ces données. Ces modèles et relations peuvent ensuite être utilisés pour faire des prédictions, identifier des anomalies ou effectuer d'autres tâches.

Il existe trois principaux types d'apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement. [2]

L'apprentissage automatique est un domaine de l'informatique qui se concentre sur le développement d'algorithmes et de modèles capables d'apprendre des schémas et des idées à partir

de données. L'apprentissage automatique est utilisé dans un large éventail d'applications, notamment la reconnaissance d'images et de la parole, le traitement du langage naturel, les systèmes de recommandation, la détection des fraudes et l'analyse prédictive. Par exemple, les algorithmes d'apprentissage automatique peuvent être utilisés pour classer automatiquement les courriels comme étant du spam ou non, ou pour recommander des films en fonction de l'historique de visionnage d'un utilisateur.

Dans l'ensemble, l'apprentissage automatique est un outil puissant qui peut aider les organisations et les individus à prendre de meilleures décisions, à automatiser des tâches répétitives et à obtenir des informations à partir de grandes quantités de données.

II.1 Domain d'utilisation :

- **Médecine et Santé** : Le machine learning aide à diagnostiquer des maladies à partir d'images médicales et à découvrir de nouveaux médicaments.
- **Finance** : Il est utilisé pour l'analyse de crédit, la prévision des marchés financiers et la détection des fraudes
- **Commerce électronique** : Le machine learning personnalise les recommandations de produits, optimise les prix et gère les stocks en ligne.
- **Automobile** : Il alimente la conduite autonome, la gestion de flottes de véhicules et la prévision de la demande pour les covoiturages.
- **Industrie** : Le machine learning assure la maintenance prédictive, le contrôle qualité automatisé et l'optimisation de la production industrielle.
- **Marketing et publicité** : Il personnalise les publicités en ligne, analyse les sentiments des clients et segmente la clientèle.
- **Energie** : Il optimise la demande énergétique, gère l'énergie intelligemment et optimise les réseaux électriques.
- **Langage naturel** : Le machine learning est essentiel pour la traduction automatique, le résumé de texte et les chatbots.
- **Vision par ordinateur** : Il permet la reconnaissance d'objets, le suivi d'objets et l'analyse d'images et de vidéos.
- **Education** : Il personnalise l'apprentissage, évalue les réponses des élèves et analyse la rétention des étudiants.

- **Sécurité** : Il détecte les intrusions informatiques, analyse les menaces de sécurité et identifie les activités suspectes.
- **L'agriculture** : Il gère l'irrigation, prédit les rendements agricoles et surveille les cultures.
- **Environnement** : Il prédit les catastrophes naturelles, surveille la qualité de l'air et de l'eau, et favorise la conservation de la faune.
- **Divertissement** : Le machine learning recommande du contenu, génère du contenu créatif et améliore les jeux vidéo.
- **Transport** : Il planifie les itinéraires, gère le trafic et facilite le partage de véhicules électriques.

II.2 Les types d'apprentissage automatique :

II.2.1 L'apprentissage supervisé :

Nous appelons l'apprentissage supervisé une branche de machines Apprentissage intéressé par des problèmes formalisables par : étant donné n observations $\{(x^i)\}$ est décrit dans un espace X, leurs étiquettes $\{y^i\}$ sont décrits dans un espace Y, on Supposons que des étiquettes puissent être obtenues à partir d'observations grâce à une fonction :

Fixe et inconnu : $y^i = \phi(x^i) + \epsilon_i$, où ϵ_i est un bruit aléatoire. il s'agit alors d'utilise les donne pour déterminer la fonction : $x \rightarrow y$ Ainsi, pour tout couple

$$(x^i, \phi(x^i)) \in x \times y, f(x^i \approx \phi(x^i))$$

L'espace définissant les données est généralement $X = R_p$. [1]

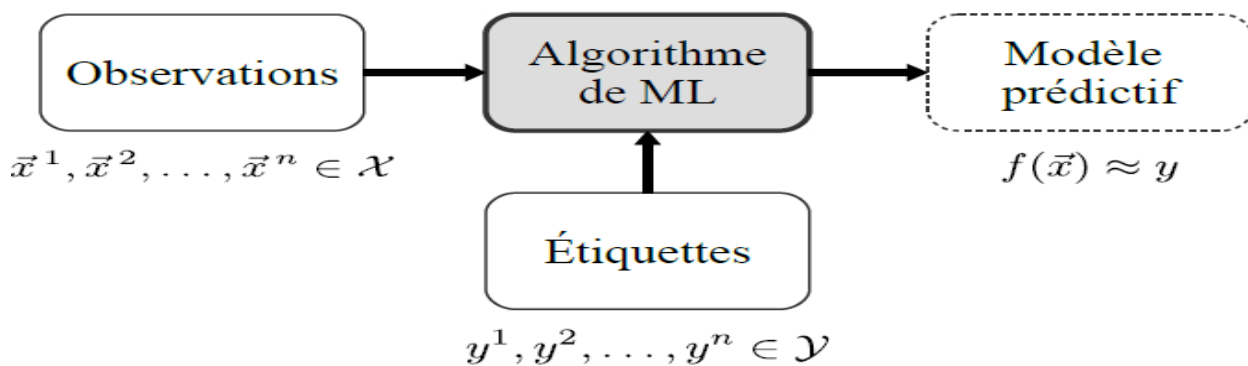


Figure .2.1. L'apprentissage automatique

- **Classification binaire** : Dans le cas où les étiquettes sont binaires, elles indiquent l'appartenance à une classe. On parle alors de classification binaire.

Un type d'apprentissage supervisé dans lequel l'espace des étiquettes est binaire, autrement dit $Y = \{0 ; 1\}$ est appelé une classification binaire. [1]

- **Classification multi-class** : Dans le cas où les étiquettes sont discrètes, et correspondent donc à plusieurs classes, on parle de classification multi-classe.

Un type d'apprentissage supervisé dans lequel l'espace des étiquettes est discret et fini, autrement dit $Y = \{1, 2, \dots, C\}$ est appelé un classification multi-classe. C'est le nombre de classes. [1]

- **Régression linéaire** : Le but de la régression linéaire est de représenter la variable de sortie y en fonction de la variable d'entrée x de manière linéaire, c'est-à-dire $y=ax+b$. Le modèle comporte donc deux paramètres, A et B , dont les valeurs optimales doivent être trouvées lors de la phase d'apprentissage. Plusieurs techniques existent pour estimer ces paramètres, les plus courantes étant les moindres carrés, la déviance et le maximum de vraisemblance. La régression linéaire est un algorithme d'apprentissage supervisé, alors nous avons N paires d'entrées-sorties pour former un ensemble de données $D=\{x_i, y_i\}_i \in [1, N]$ Ce sont ces données connues qui permettent d'estimer les paramètres du modèle. Ici nous utiliserons la méthode des moindres carrés, qui est plus courante que les autres méthodes. Dans le cas des moindres carrés, on cherche les paramètres A et B qui permettent de minimiser la fonction de coût $\sum_{i=1}^N (y_i - (ax_i + b))^2$ Cette fonction correspond à la somme des écarts au carré entre les valeurs prédites et attendues. Ces écarts à minimiser sont appelés résidus. [2]

II.2.2 L'apprentissage non supervise :

L'apprentissage non supervisé désigne la situation d'apprentissage automatique où les données ne sont pas étiquetées (par exemple, étiquetées "balle" ou "poisson"). Il s'agit donc de découvrir la structure derrière ces données non étiquetées. Étant donné que les données ne sont pas étiquetées, l'algorithme ne peut pas calculer de manière fiable la note de passage. Les méthodes non supervisées présentent ainsi une auto-organisation, capturant des modèles sous forme de densités de probabilité ou, dans le cas des réseaux de neurones, sous forme de combinaisons de poids de machine et de préférences de caractéristiques neuronales codées dans les activations.

L'apprentissage non supervisé c'est une branche de l'Apprentissage automatique s'intéressant aux problèmes qui peuvent être formalisés par : Étant donné n est décrit dans un espace X , qui est un

problème d'apprentissage d'une fonction sur X Vérifiez observations $\{x^i\}_{i=1,\dots,n}$ certaines propriétés. [1]

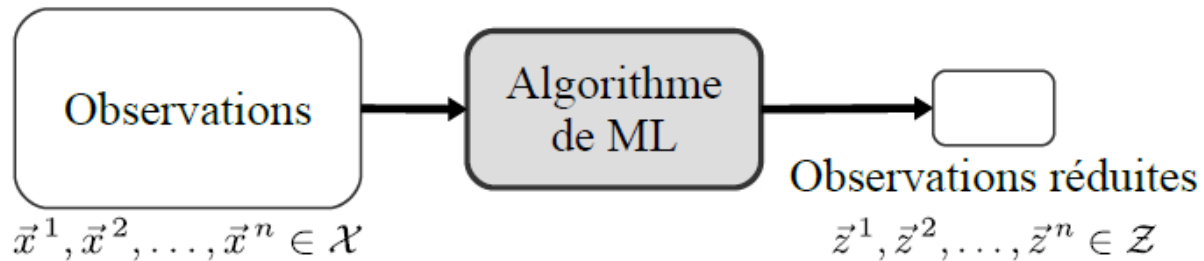


Figure.2.2. L'apprentissage non supervisé

- **Clustering** : Le partitionnement (clustering) est une méthode d'analyse statistique utilisée pour organiser les données brutes en îlots homogènes. Au sein de chaque cluster, les données sont regroupées en fonction de caractéristiques communes. Un ordonnanceur est un algorithme qui mesure la proximité entre chaque élément selon des critères définis. Le clustering est principalement utilisé pour segmenter ou classer des bases de données (comme le tri des données clients par âge, profession, lieu de résidence, etc. pour optimiser la gestion de la relation client) ou pour extraire des connaissances afin de tenter d'identifier des sous-ensembles de données difficiles à identifier avec l'œil nu. Dans les citations naturelles, le regroupement est utilisé pour construire les mots-clés d'un site et créer la base de sa structure sémantique basée sur l'intention de recherche collectée sur les pages de résultats des moteurs de recherche. [1]

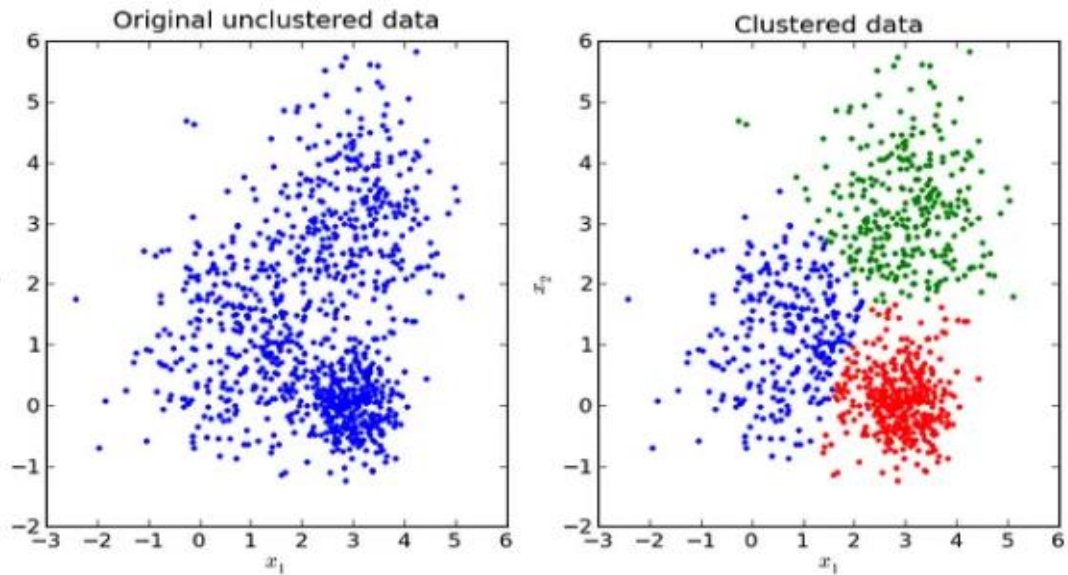


Figure.2.3. Partitionnement des données, ou clustering.

- **Réduction de dimension :** La réduction de la dimensionnalité est une autre classe importante de problèmes d'apprentissage non supervisé.

Il s'agit d'un problème de recherche de représentations de données dans un espace de dimensions inférieures à l'espace qu'elles représentent à l'origine (voir figure 1.4). Cela permet de réduire le temps de calcul et l'espace mémoire nécessaires pour stocker les données, et améliore souvent aussi les performances des algorithmes d'apprentissage supervisé qui sont ensuite entraînés sur ces données. [3]

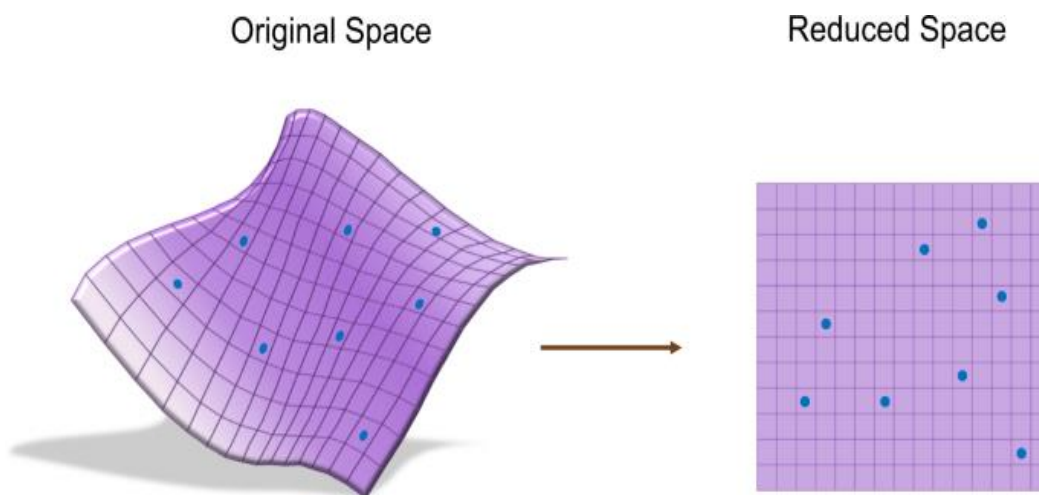


Figure.2.4. Réduction de dimension

II.2.3 L'apprentissage semi-supervise :

L'apprentissage semi-supervisé consiste à apprendre à partir de Ensemble de données partiellement étiquetées. Le premier avantage de cette approche est qu'elle évite d'avoir à étiqueter tous les exemples d'apprentissage, ce qui est pertinent lorsqu'il est facile d'accumuler des données mais que l'étiquetage nécessite un certain effort humain.

Prenons l'exemple de la classification des images : il est facile d'obtenir une base de données de centaines de milliers d'images, mais il faut beaucoup de travail pour étiqueter chaque image avec celle qui nous intéresse.

De plus, les étiquettes données par les humains peuvent reproduire des biais humains, qu'un algorithme entièrement supervisé reproduirait à son tour.

L'apprentissage semi-supervisé permet parfois d'éviter cet écueil. [3]

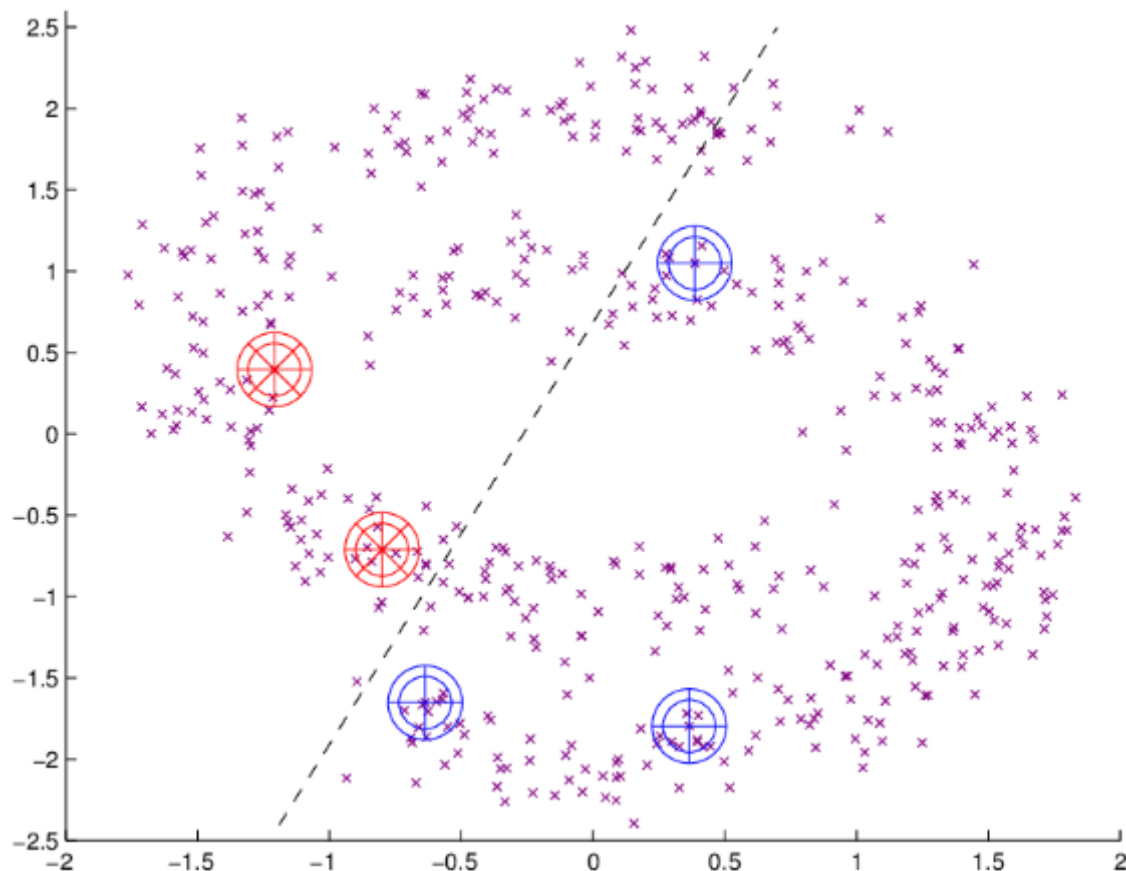


Figure.2.5. Un exemple d'apprentissage semi supervise

Sur cette figure, on peut voir la surface de décision apprise dans un but précis

Soit un classement de certains points marqués comme appartenant à la première ou à la deuxième classe et certains points marqués non.

II.2.4 L'apprentissage par renforcement :

L'apprentissage par renforcement (RL) est une méthode d'apprentissage axée sur la prise de décision, à partir d'un état s de l'environnement (s étant un état), l'agent choisit une action a en utilisant la politique π . L'apprentissage par renforcement vise à optimiser la politique d'action π d'un agent à travers un ensemble de récompenses positives et négatives.

L'apprentissage demandant un stupendous nombre d'expérimentation, il est très utile voir irremplaçable de disposer d'un environnement simulé. Les jeux vidéo étant un environnement simulé avec des récompenses (le score), ils sont un support classique pour l'apprentissage standard renforcement. [4]

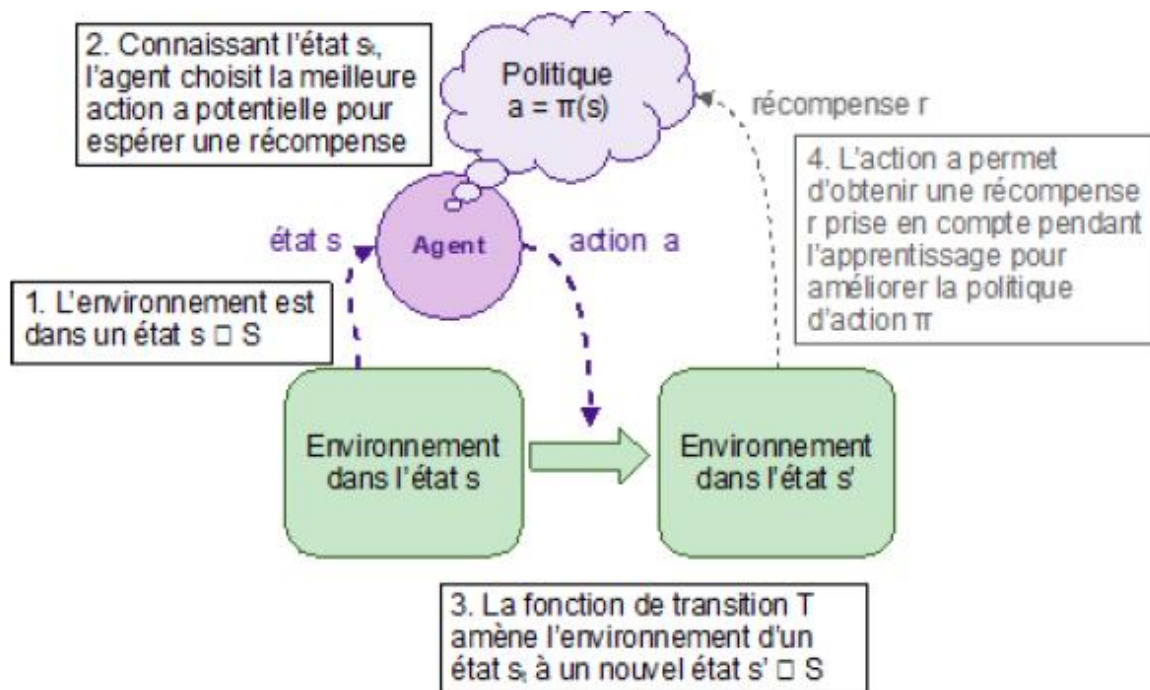


Figure.2.6. Principe de fonctionnement d'apprentissage par renforcement

II.3 Les avantages et les inconvénients de l'apprentissage automatique :

II.3.1 Les avantages :

- **Il est automatique** : dans l'apprentissage automatique, l'ensemble du processus d'interprétation et d'analyse des données est réalisé par l'ordinateur.
Aucune intervention humaine n'est nécessaire pour la prédiction ou l'interprétation des données. L'ensemble du processus d'apprentissage automatique consiste à ce que la machine commence à apprendre et à prédire l'algorithme ou le programme afin d'obtenir le meilleur résultat. L'un des exemples est celui de Google Home, qui détecte la voix et obtient en conséquence le résultat souhaité par l'utilisateur, et celui du logiciel antivirus, qui détecte le virus de l'ordinateur et le corrige. [W7]
- **Il est utilisé dans différents domaines** : L'apprentissage automatique est utilisé dans divers domaines tels que l'éducation, la médecine, l'ingénierie, etc.
D'une très petite application à des machines structurées très grandes et compliquées qui aident à la prédiction et à l'analyse des données. Il devient non seulement le fournisseur de soins de santé, mais il fournit également des services plus personnalisés au client potentiel. [W7]
- **Il peut traiter une grande variété de données** : Même dans un environnement incertain et dynamique, il peut traiter une grande variété de données. Il est multidimensionnel et multitâche. [W7]
- **Possibilités de progrès** : De la même manière que les humains s'améliorent après avoir acquis de l'expérience, l'apprentissage automatique s'améliore et devient plus précis et plus efficace dans son travail, cela a permis de prendre de meilleures décisions. [W7]
- **Il peut identifier des tendances et des modèles** : Par exemple, pour un site de réseau social comme Facebook, les gens surfent et naviguent sur plusieurs données et leur intérêt est enregistré, ce qui permet de comprendre le modèle et de leur montrer la même tendance ou une tendance similaire afin de maintenir leur intérêt pour la même application. L'apprentissage automatique permet ainsi d'identifier les tendances et les modèles. [W7]
- **Considéré comme le meilleur pour l'éducation** : L'apprentissage automatique est considéré comme le meilleur pour l'éducation, car l'éducation est dynamique et, de nos

jours, les classes intelligentes, l'apprentissage à distance et l'apprentissage en ligne pour les étudiants ont beaucoup augmenté. L'apprentissage automatique intelligent jouera le rôle d'enseignant et permettra aux étudiants de se tenir au courant du scénario actuel du monde. Il en va de même pour le shopping ou le commerce électronique : les gens ont besoin de se tenir au courant et on leur montre donc les tendances actuelles du monde.

II.3.2 Les inconvénients :

- **Le risque d'erreur ou de faute est plus élevé :** Bien que l'apprentissage automatique soit considéré comme plus précis, il est très vulnérable. Par exemple, un ensemble de programmes fournis à la machine peut être biaisé ou comporter des erreurs. Si le même programme est utilisé pour faire une autre prévision ou prédiction, il y aura alors une chaîne d'erreurs qui pourrait se former et qui, bien que reconnue, pourrait prendre du temps pour trouver la source de l'erreur. [W7]
- **Les besoins en données sont plus importants :** Plus une machine reçoit de données, plus elle devient précise et efficace, et plus elle a besoin de données pour améliorer ses prévisions ou sa prise de décision. Mais cela peut parfois s'avérer impossible. En outre, les données doivent être impartiales et de bonne qualité. Les exigences en matière de données sont parfois problématiques. [W7]
- **Ils prennent du temps et nécessitent davantage de ressources :** Il peut arriver que le processus d'apprentissage de la machine prenne beaucoup de temps, car l'efficacité et l'efficience ne peuvent être obtenues que par l'expérience, ce qui demande également du temps. Les ressources requises sont également plus importantes, par exemple des ordinateurs supplémentaires peuvent être nécessaires. [W7]
- **Imprecision de l'interprétation des données :** Comme nous l'avons déjà vu, une petite manipulation ou des données biaisées peuvent conduire à une longue chaîne d'erreurs et il y a donc également des risques d'inexactitude dans l'interprétation. Parfois, des données sans aucune erreur peuvent également être interprétées de manière inexacte par la machine, car les données fournies précédemment peuvent ne pas répondre à toutes les exigences de base de la machine.
- **Plus d'espace requis :** Comme l'interprétation nécessite plus de données, il faut plus d'espace pour les stocker, ce qui est l'une des faiblesses de l'apprentissage automatique. Plus de données signifie plus de connaissances ou de matériel à apprendre pour la machine,

ce qui nécessite beaucoup d'espace pour stocker ou gérer les données en vue d'une prise de décision ultérieure. [W7]

III. L'apprentissage profond (Deep Learning) :

Le Deep Learning est une branche de l'intelligence artificielle qui s'intéresse à l'apprentissage automatique à partir de données complexes et hétérogènes telles que des images, des sons ou des textes.

Cette approche est basée sur l'utilisation de réseaux de neurones artificiels profonds, capables d'apprendre des caractéristiques ou des représentations de ces données de manière automatique.

Il est utilisé dans de nombreuses applications telles que la reconnaissance d'images, la traduction automatique, la reconnaissance de la parole, l'analyse de sentiments, la prédiction de séries temporelles, etc. Il est particulièrement efficace pour traiter des données de grande dimensionnalité, non structurées et bruyantes.

Dans le processus d'apprentissage, les réseaux de neurones profonds sont entraînés sur des ensembles de données d'entraînement en ajustant les poids des connexions entre les neurones pour minimiser une fonction de coût qui mesure la différence entre les prédictions du réseau et les valeurs cibles correspondantes.

Le Deep Learning représente une avancée majeure dans le domaine de l'intelligence artificielle, permettant aux ordinateurs d'apprendre à résoudre des problèmes complexes de manière autonome en utilisant des données non structurées et des algorithmes d'apprentissage profond.

III.2 L'architecture de Deep Learning :

III.2.1 Réseaux antagonistes génératifs (GAN) :

Les réseaux de neurones antagonistes génératifs appartiennent à la famille des modèles génératifs qui visent à apprendre une probabilité distribution sur un ensemble de données d'apprentissage afin que le réseau puisse générer de manière aléatoire de nouveaux échantillons de données avec variation.

Les GAN sont constitués de deux sous-réseaux concurrents. Le premier réseau est un générateur réseau qui capture la distribution de l'entrée et génère de nouveaux exemples.

Le deuxième réseau est un réseau discriminatoire essayant de distinguer les exemples réels de ceux générés artificiellement., les deux réseaux sont formés ensemble dans un jeu à somme nulle non

coopératif où le gain d'un réseau est la perte d'un autre jusqu'à ce que le discriminateur ne peut plus faire la distinction entre les deux types d'échantillons.

Sur cette base, les GAN sont susceptibles de révolutionner les domaines dans lesquels de nouveaux contenus ou de nouvelles configurations de produits sont continuellement créés (par exemple, la composition d'art et de musique, design de mode), ou lorsque le contenu est converti d'une représentation à un autre (par exemple, du texte à l'image pour les descriptions de produits). En même temps, cependant, de telles approches constituent également de graves menaces avec des implications sociétales lorsqu'elles sont utilisées à des fins malveillantes.

Dans notamment, la génération de contenus « deepfake » sous la forme de discours abusifs et de nouvelles trompeuses à manipuler l'opinion publique ou fausser les marchés financiers est préoccupant. [5]

III.2.2 Principe du réseau antagoniste génératif (Generative Adverserials Networks GAN) :

Le Générative Adversarial Network (GAN en abrégé) est l'un des algorithmes d'apprentissage en profondeur les plus populaires développés ces dernières années.

Le rôle du réseau GAN est de générer des images ou d'autres données plus efficacement que le détecteur (Forger). Après avoir terminé sa culture, il peut le faire, par exemple, en utilisant cette technologie, il peut créer un faux visage.

Les GAN sont de plus en plus importants dans la recherche scientifique et l'analyse d'images médicales en raison de leur capacité à générer des images fidèles à la réalité (c'est pourquoi nous les utilisons pour la recherche) [5]

Les étapes de GAN :

- Le générateur prend un nombre aléatoire et renvoie une image.
- Envoyez les images générées au discriminateur avec des flux d'images tirées d'ensembles de données réels et réels.
- Le discriminateur prend à la fois des images réelles et fausses et renvoie la probabilité, un nombre compris entre 0 et 1, où 1 représente une prédiction réelle et 0 représente un faux.

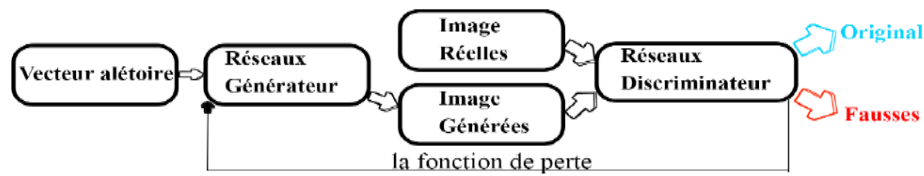


Figure.2.7. Schéma bloc du GAN

III.2.1.1 Fonctions pertes (loss Function) du discriminateur et du générateur :

On prend m une donnée d'apprentissage $\{x(1), x(2), x(m)\}$ et une distribution

Data p_{data} , Soit les m échantillons de bruit $\{z(1), z(2), z(m)\}$, et la distribution est p_z . (i) est

la sortie du générateur qui prend des échantillons de bruit en entrée. (i) est la sortie du

discriminateur pour l'entrée $x(i)$ et $D(G(z(i)))$ est la sortie du discriminateur pour l'entrée $G(z(i))$.

[5] Les fonctions de perte du discriminateur et du générateur sont définies comme :

$$(\mathbf{D}) = \frac{1}{2m} \sum_{i=1}^m [(D(x^i) - 1)^2] + \frac{1}{2m} \sum_{i=1}^m [D(G(z^i))]^2$$

$$J(\mathbf{G}) = \frac{1}{2m} \sum_{i=1}^m [(G(x^i) - 1)^2]$$

III.2.3 Autoencoder :

Autoencoder est un réseau de neurones qui utilise un algorithme non supervisé et apprend la représentation dans l'ensemble de données d'entrée pour réduire la dimensionnalité et recréer l'ensemble de données d'origine.

L'algorithme d'apprentissage est basé sur l'implémentation de la rétro propagation. Les auto-encodeurs étendent l'idée de l'analyse en composantes principales (ACP) comme montre sur figure

ci-dessus, une ACP transforme les données multidimensionnelles en une représentation linéaire.

La figure ci-dessus montre comment des données d'entrée 2D peuvent être réduites à un vecteur

linéaire à l'aide de l'ACP. Les auto-encodeurs, quant à eux, peuvent aller plus loin et produire une

représentation non linéaire. L'ACP détermine un ensemble de variables linéaires dans les

directions avec la plus grande variance. Les points de données d'entrée de dimension p sont

représentés sous la forme de directions orthogonales m , de sorte que $m \leq p$ et constituent un espace

dimensionnel inférieur (c'est-à-dire inférieur à m). Les points de données d'origine sont projetés

dans les directions principales, omettant ainsi les informations dans les directions orthogonales correspondantes. [5]

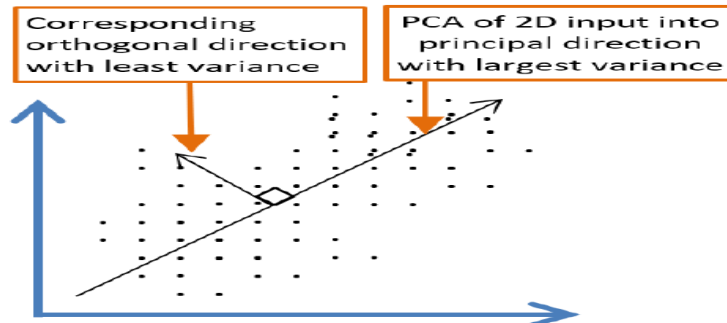


Figure.2.8. Représentation linéaire d'une entrée de données 2D à l'aide de PCA

L'ACP se concentre davantage sur les variances que sur les covariances et les corrélations et recherche la fonction linéaire avec la plus grande variance. Le but est de déterminer la direction avec le moins d'erreur quadratique moyenne, qui aurait alors le moins d'erreur de reconstruction. Les auto-encodeurs utilisent des blocs d'encodeur et de décodeur de couches cachées non linéaires pour généraliser l'ACP afin d'effectuer une réduction de dimensionnalité et une reconstruction éventuelle des données d'origine.

Il utilise couche par couche un pré-entraînement non supervisé et un réglage fin avec rétro propagation.

Malgré l'utilisation de la rétro propagation, qui est principalement utilisée dans la formation supervisée, les encodeurs automatiques sont considérés comme DNN non supervisés car ils régénèrent x^i l'entrée elle-même au lieu d'un ensemble différent de valeurs cibles y^i

C'est-à-dire

$$y^i = x^i$$

Hinton et al. Ont pu obtenir une reconstruction presque parfaite d'images de 784 pixels à l'aide d'un encodeur automatique, ce qui prouve qu'il est bien meilleur que l'ACP.

Lors de la réduction de la dimensionnalité, les auto-encodeurs proposent des représentations intéressantes du vecteur d'entrée dans la couche cachée.

Ceci est souvent attribué au plus petit nombre de nœuds dans la couche cachée ou à chaque deuxième couche des blocs à deux couches. [5]

Mais même s'il y a un plus grand nombre de nœuds dans la couche cachée, une contrainte de parcimonie peut être appliquée sur les unités cachées pour conserver des représentations de

dimension inférieure intéressantes des entrées. Pour atteindre la parcimonie, certains nœuds ne peuvent pas être déclenchés, c'est-à-dire que la sortie est définie sur une valeur proche de zéro.

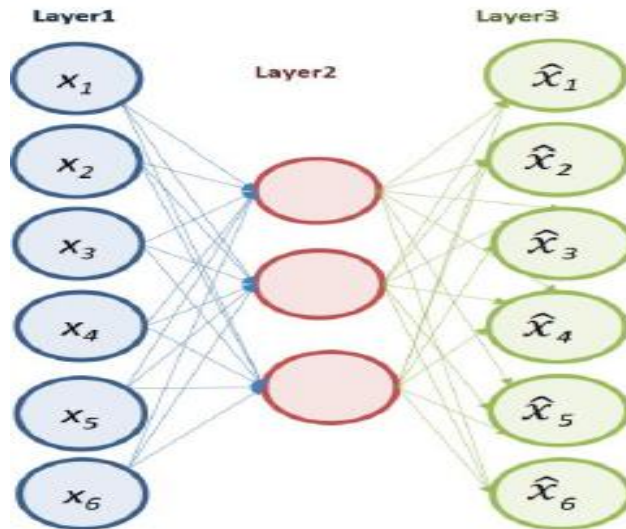


Figure.2.9.les nœuds de l'Autoencoder

Cette figure illustre une représentation simplifiée de la façon dont les auto-encodeurs peuvent réduire la dimension des données d'entrée et apprendre à les recréer dans la couche de sortie. Wang et al. A mis en œuvre avec succès un auto-encodeur profond avec des piles de blocs RBM pour obtenir une meilleure précision et efficacité de modélisation que la méthode de décomposition orthogonale (POD) appropriée pour la réduction de la dimensionnalité des systèmes de paramètres distribués (DPS). L'équation ci-dessous décrit la moyenne de la fonction d'activation a_j^i de l'unité j th de la 2em couche lorsque l'entrée x th active le neurone :

$$P = \frac{1}{m} \sum_{i=1}^m [a_j^i x^i]$$

III.2.4 Réseaux de neurone récurrent (RNN) :

Le réseau de neurones récurrent appartient à la classe des réseaux de neurones artificiels. Dans ces réseaux, il existe une connexion cyclique dirigée entre ses nœuds internes le long d'une séquence. Il présente le comportement temporel dynamique d'une séquence temporelle. Ces réseaux utilisent des états de mémoire interne pour traiter dans les séquences d'entrée. Dans les réseaux de neurones artificiels conventionnels, les valeurs d'entrée dans un vecteur d'entrée sont indépendantes les unes des autres et donc traitées indépendamment. Mais il existe de nombreuses tâches où la sortie dépend du calcul précédent dans un processus séquentiel. Les réseaux de

neurones récurrents sont appliqués à ce type de tâches où il y a un processus séquentiel sur les entrées. Ce réseau est appelé récurrent car il effectue la même tâche pour chaque élément de la séquence. La mémoire utilisée par le réseau stocke les informations sur les calculs précédents. Pratiquement, ces réseaux ne rappellent que les calculs de quelques étapes précédentes. [5]

Le fonctionnement du réseau de neurones récurrent peut être expliqué par l'architecture représentée dans la figure ci-dessous. [6]

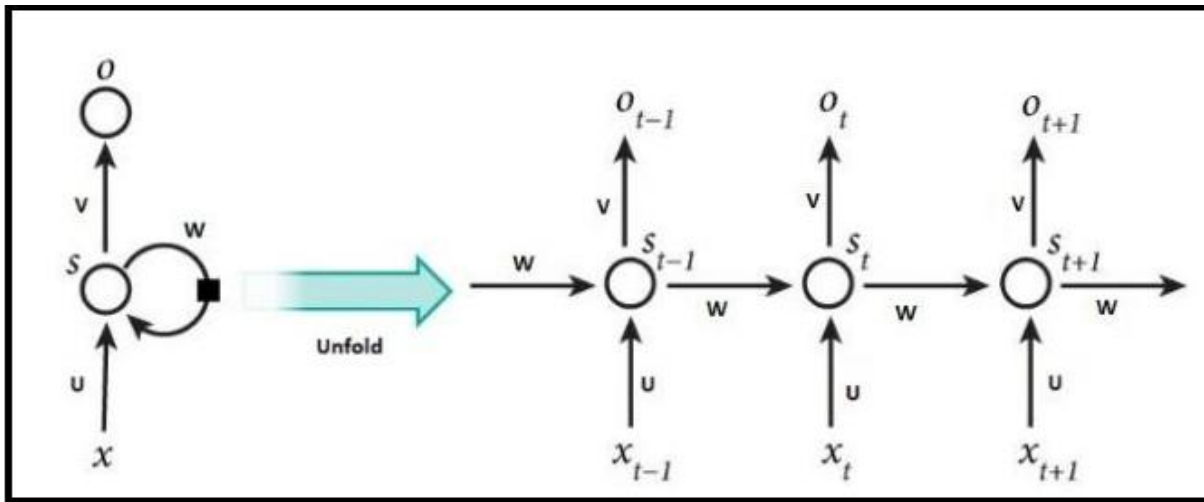


Figure.2.10. Récurrent Neural Network

La représentation ci-dessus montre un réseau neuronal récurrent déplié en un réseau complet pour traiter une séquence d'entrées. Ici, une couche fonctionne pour chaque valeur d'entrée dans la séquence. Si toutes les couches sont pliées ou combinées ensemble en une seule couche cachée, les poids et le biais restent les mêmes car une seule couche cachée est utilisée dans le réseau.

Soit x_t Une entrée du réseau à l'horodatage t et s_t l'état caché ou la mémoire à l'horodatage t . Ce s_t est calculé sur la base de l'état masqué à l'horodatage précédent et de l'entrée à l'horodatage actuel comme :

$$s_t = f(u \cdot x_t + w \cdot s_{t-1})$$

Où f est une fonction non linéaire qui peut être tanh ou ReLU. s_{t-1} est initialisé à zéro au premier horodatage. O_t C'est la sortie à l'horodatage t qui est calculée comme suit

$$O_t = f(v \cdot s_t)$$

Dans les équations ci-dessus, f est une fonction logistique qui peut être softmax ou une fonction exponentielle normalisée.

Contrairement aux autres réseaux de neurones profonds où différents paramètres tels que les poids et les biais sont utilisés à différentes couches cachées, dans le réseau de neurones récurrent, ces paramètres sont partagés entre tous les horodatages. Cela permet de réduire le nombre de paramètres lors de l'apprentissage. Dans certaines applications, une entrée est requise à chaque horodatage et une sortie est produite à chaque horodatage. Mais ce n'est pas nécessaire dans toutes les applications de réseau neuronal récurrent. Ceci en raison de l'utilisation de l'état caché qui capture des informations sur certaines séquences.

Il existe plusieurs extensions du réseau neuronal récurrent. Ils sont discutés brièvement comme indiqué ci-dessous :

- **Réseaux neuronaux récurrents bidirectionnels** : Ce réseau repose sur le concept selon lequel la sortie à l'instant t dépend non seulement des éléments précédents de la séquence, mais aussi des éléments futurs. Son architecture est telle que deux réseaux neuronaux récurrents sont empilés l'un sur l'autre. Sa sortie est calculée sur la base de l'état caché des deux réseaux
- **Réseaux neuronaux récurrents bidirectionnels profonds** : Ces réseaux sont similaires aux réseaux neuronaux récurrents bidirectionnels, à ceci près qu'ils comportent plusieurs couches par horodatage. Ils offrent l'avantage d'une plus grande capacité d'apprentissage, mais nécessitent une grande quantité de données d'apprentissage.
- **Réseaux à mémoire à long terme (LSTM)** : Cette variante du réseau neuronal récurrent est appliquée pour éviter le problème du gradient de Disparition dans l'apprentissage par rétropropagation. Dans ces réseaux, les unités de mémoire sont appelées cellules et sont très efficaces pour capturer les dépendances à long terme. L'état précédent s_{t-1} et l'entrée actuelle x_t sont pris en compte par les cellules, qui décident en interne quelle information sera stockée et quelle information sera effacée. Les réseaux neuronaux récurrents ont un grand nombre d'applications dans l'analyse prédictive. Ils ont été largement utilisés dans les prévisions du marché boursier pendant une longue période. Son application à la prédiction de séries temporelles a permis de généraliser les performances par rapport à d'autres modèles. Ces réseaux, après avoir combiné des poids dynamiques, ont été utilisés pour prédire la fiabilité des logiciels. Avec l'ajout de caractéristiques de corrélation spatiale, le réseau neuronal récurrent est utilisé pour prédire la vitesse du vent. Outre les applications importantes susmentionnées, les RNN présentent certaines limites. Le temps

d'apprentissage de ces réseaux est lent. Dans les RNN, le nombre de neurones cachés doit être fixé avant l'apprentissage. Lors du traitement d'un vocabulaire, la taille du contexte doit être faible.

III.2.5 Réseaux de Neurones Convolutionnels (CNN) :

Le réseau neuronal convolutif est une variante du perceptron multicouche. Il s'inspire du processus biologique de visualisation. Ce modèle est une composition de neurones, de poids pouvant être appris et de valeurs de biais. Il se compose d'une couche d'entrée, d'une couche de sortie et de plusieurs couches cachées entre la couche d'entrée et la couche de sortie. Les couches cachées du réseau sont composées de couches de convolution, de couches de mise en commun, de couches entièrement connectées et de couches de normalisation. Elles sont conçues de manière à ne nécessiter qu'un minimum de prétraitement, [6]

L'architecture d'un réseau neuronal convolutif est représentée dans la figure ci-dessous.

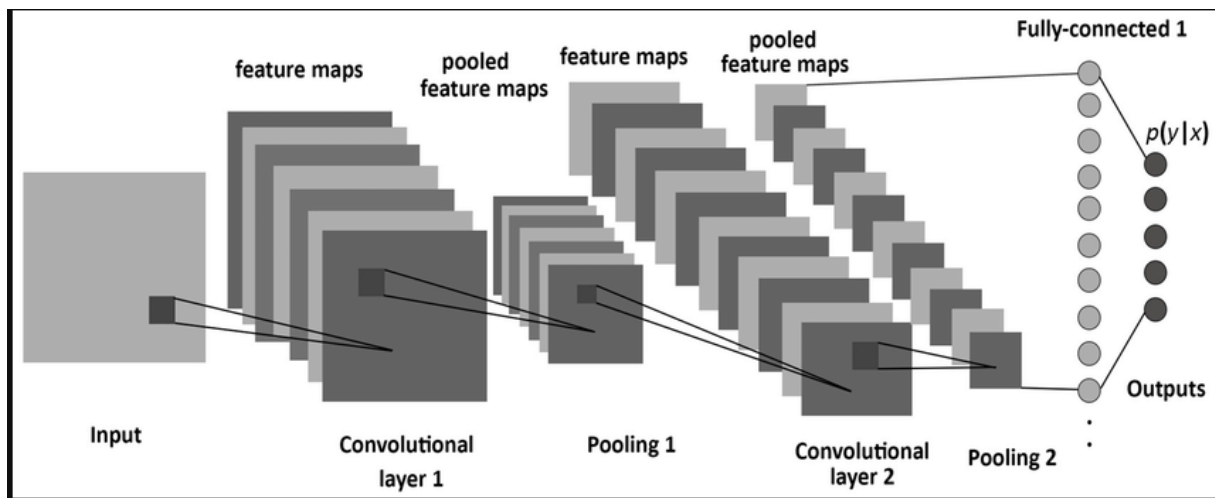


Figure.2.11. Architecture classique d'un réseau de neurones convolutif (CNN)

III.2.5.1 Les couches de réseaux de neurones Convolutionnels :

- **Couche d'entrée :** La couche d'entrée du CNN doit contenir des données d'image. Les données d'image sont représentées par une matrice tridimensionnelle, comme nous l'avons vu précédemment. Vous devez les remodeler en une seule colonne. Supposons que vous ayez une image de dimension $28 \times 28 = 784$, vous devez la convertir en 784×1 avant de

l'introduire dans l'entrée. Si vous avez "m" exemples d'apprentissage, la dimension de l'entrée sera (784, m). [7]

- Couche de convolution et filtre :** La couche convolutive est la sortie du produit du point entre les poids et les entrées de la couche précédente, couche précédente. Les neurones qui couvrent l'ensemble de l'entrée et recherchent une seule caractéristique sont appelés filtres. De nombreux filtres peuvent être utilisés sur cette couche convolutive. Les poids des filtres sont partagés entre les champs réceptifs. Ces filtres peuvent aider à déterminer caractéristiques des entrées. Ces filtres sont bidimensionnels. Les couches convolutives sont invariantes dans l'espace. Cela signifie qu'elles recherchent les mêmes caractéristiques à travers sur l'ensemble des vecteurs d'entrée. L'entrée de cette couche peut être constituée de plusieurs canaux. Par exemple, une image comporte trois canaux RVB (rouge, vert et bleu).

Exemple : une image comporte 3 canaux RVB (rouge, vert, bleu). Ces canaux sont transmis à la couche pour l'opération de mise en correspondance des caractéristiques, à la couche pour l'opération de mise en correspondance des caractéristiques.

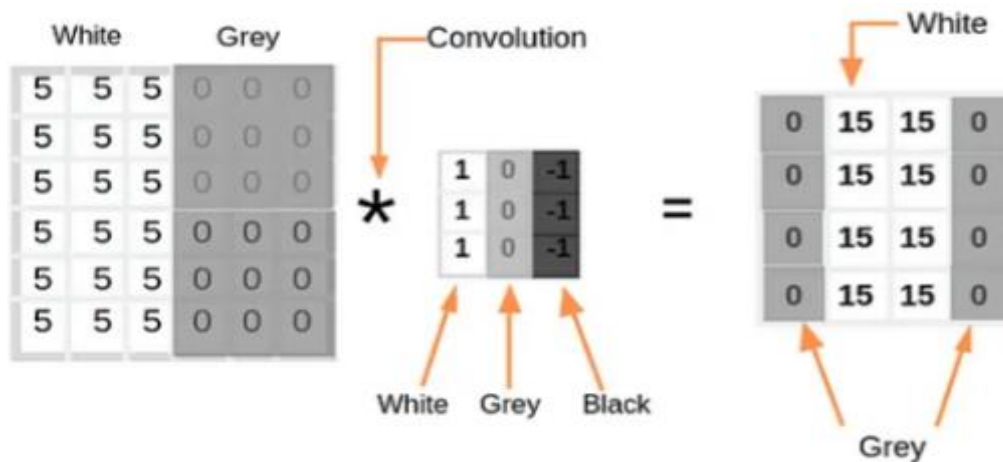


Figure.2.12. Exemple de fonctionnement d'un filtre Convolutionnels 3x3

- Couche de mise en commun (Pooling) :** La couche de mise en commun est le processus d'échantillonnage en aval de la couche convolutive. Il dépend du type de couche de mise en commun pour la sélection des éléments de la couche Convolutionnels. Dépend du type de couche de mise en commun pour la sélection des éléments de la couche convolutionnelle lors de l'échantillonnage descendant. Par exemple : La mise en commun maximale choisit le nombre maximal d'un groupe d'éléments.

Maximum d'un groupe d'éléments. [7]

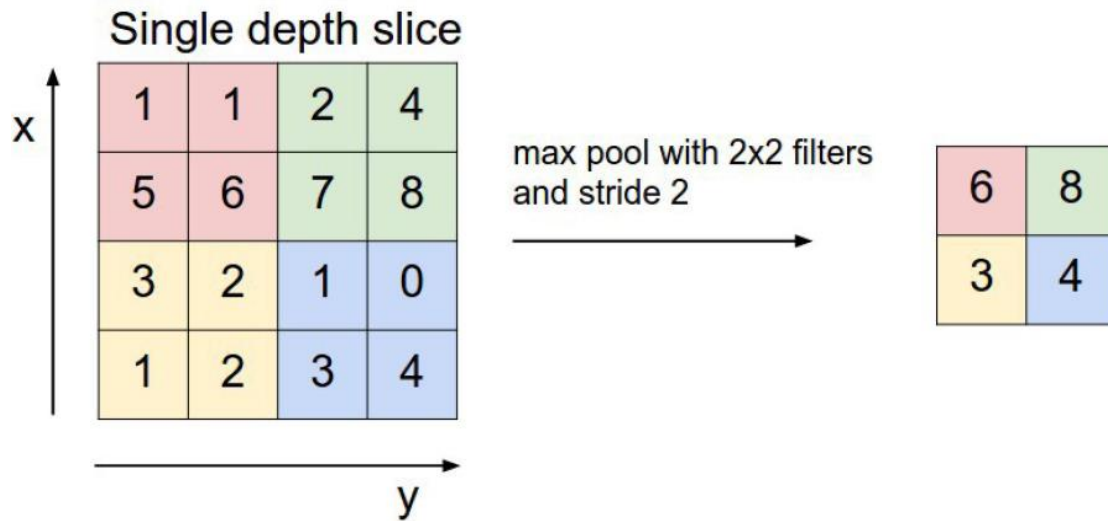


Figure.2.13. Échantillon de regroupement maximal avec filtre 2 x 2

- **Couche entièrement connecté (Fully Connected) :** Une couche entièrement connectée implique des poids, des Biais et Neurones. Il relie les neurones d'une couche aux neurones d'une autre couche. Il est utilisé pour classer les images entre différentes catégories par format. [7]

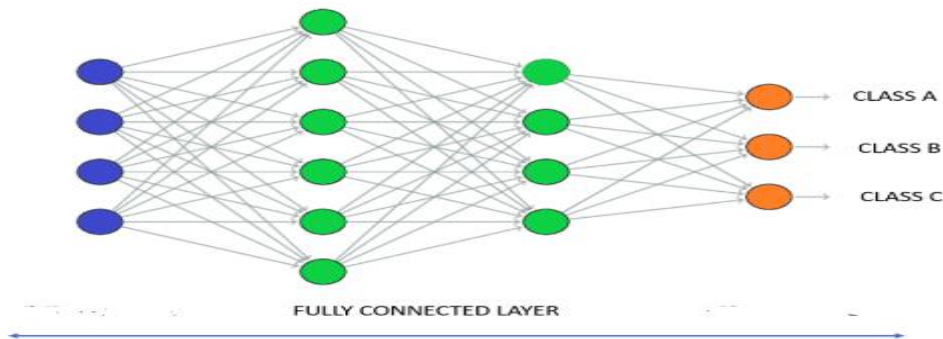


Figure2.14. Fully connected layer

- **Couche Logistique ou Softmax:** Softmax est utilisé dans les problèmes de classification. Cette fonction écrase la sortie de toutes les unités entre 0 et 1. La sortie de cette fonction est égale à une distribution de probabilité catégorielle. [7]
C'est-à-dire que cette fonction peut décrire la probabilité de la classe à partir de la sortie.
- **Couche de sortie (output layer) :** c'est la couche finale qui produit les prédictions. Il s'agit généralement d'une couche entièrement connectée qui associe les caractéristiques aux

prédictions de sortie. Le nombre de neurones dans la couche de sortie dépend de la tâche et la fonction d'activation utilisée dépend du problème à résoudre. La fonction de perte utilisée pour former le CNN est sélectionnée sur la base de la fonction d'activation de la couche de sortie.[7]

IV. Conclusion :

En conclusion, l'intelligence artificielle (IA) est un vaste domaine qui englobe une série de technologies, dont l'apprentissage automatique et l'apprentissage profond.

L'apprentissage automatique implique l'utilisation d'algorithmes pour analyser les données et en tirer des enseignements, tandis que l'apprentissage profond est un type spécifique d'apprentissage automatique qui utilise des réseaux neuronaux à couches multiples pour traiter et analyser des données complexes. Ensemble, ces technologies sont utilisées pour révolutionner des secteurs tels que les soins de santé, la finance et les transports, entre autres. Les applications de l'IA sont vastes et comprennent la reconnaissance vocale, le traitement du langage naturel, la reconnaissance d'images et de vidéos, et la conduite autonome. Au fur et à mesure que ces technologies se développent et s'améliorent, elles joueront sans aucun doute un rôle de plus en plus important dans la construction de notre avenir. ;

Chapitre III

***LE DEVELOPPEMENT DU MODEL
INTELLIGENT***

I. Introduction :

Dans ce chapitre, nous présentons le processus de développement et les résultats de nos modèles d'apprentissage profond utilisant les techniques CNN pour l'identification des maladies des plantes à partir d'images de feuilles.

Avec des pertes considérables infligées à l'agriculture en raison des maladies végétales. Ces pertes trouvent leur origine dans divers facteurs, tels que des diagnostics incorrects, les coûts élevés associés à l'expertise humaine requise, et la propagation rapide des maladies d'une culture à l'autre. Face à cette situation critique, l'urgence se fait sentir quant à la mise en place d'un système automatisé et intelligent, capable d'effectuer avec précision l'identification des maladies végétales. Afin de relever ces défis majeurs, nous avons élaboré une approche novatrice qui fait appel aux modèles d'apprentissage en profondeur, spécifiquement les Convolutional Neural Networks (CNN). Ces modèles sont employés pour analyser des images de feuilles et ainsi identifier les maladies qui les affectent. En tenant compte des particularités propres à chaque espèce végétale et aux schémas spécifiques de développement des maladies, nous avons opté pour une approche individualisée en développant des modèles distincts pour chaque type de plante sélectionné.

II. Travail Réalisé :**II.1 Création de Les bases de donnes :**

L'étape cruciale suivante dans le développement d'un modèle d'apprentissage profond, après la définition du problème, est la collecte et le prétraitement des données. Dans notre cas, nous avons méthodiquement sélectionné les espèces végétales en fonction de leur disponibilité dans la région. Notre priorité a été accordée aux espèces pour lesquelles un volume substantiel d'informations sur les maladies était accessible, dans le but d'assurer l'efficacité de l'apprentissage de notre modèle. Parallèlement, nous avons attaché une grande importance à l'acquisition d'un nombre conséquent d'images de feuilles de qualité supérieure, car ces images revêtent un rôle fondamental dans le processus d'entraînement de notre modèle en vue d'atteindre un taux de reconnaissance élevé. Pour ce faire, nous avons sélectionné avec soin huit espèces végétales spécifiques. Les espèces ont été choisies comme suit :

II.1.1 Fraise :

Nous avons travaillé avec un ensemble de 1565 images pour cette espèce, réparties en 1251 images pour l'entraînement et 314 images pour les tests. Pour une vue plus détaillée des images que nous avons utilisées, veuillez consulter le **Tableau 3.1** ci-dessous :

Les classes	Les tailles des images	Le nombre des images du train	Le nombre des images du test
Tache angulaire	256* 256	887	222
Bonne sante	256* 256	364	92

Tableau 3.1 Paramètres de base de données Fraise**II.1.2 Pomme de terre :**

Nous avons utilisé un total de 1937 images pour cette espèce végétale. Parmi ces images, 1550 ont été consacrées à l'entraînement de notre modèle, tandis que les 387 images restantes ont été réservées aux tests. Pour obtenir une vue plus approfondie des détails relatifs aux images utilisées, nous vous invitons à consulter le **Tableau 3.2** ci-dessous :

Les classes	Les tailles des images	Le nombre des images du train	Le nombre des images du test
Alternariose	256* 256	720	180
Mildiou	256* 256	720	180
Bonne sante	256* 256	110	27

Tableau 3.2 Paramètres de base de données pomme de terre

II.1.2 Raisin :

Nous avons exploité un ensemble de 11171 images pour cette variété de plante. Parmi ces images, 9276 ont été réservées à l'entraînement de notre modèle, tandis que les 1895 images restantes ont été destinées aux tests. Pour obtenir une vision plus détaillée des informations relatives aux images utilisées, veuillez-vous référer au **tableau 3.3** :

Les classes	Les tailles des images	Le nombre des images du train	Le nombre des images du test
Vigne black rot	256*256 / 224*224	2182	562
Vigne rougeot Parasitaire	256*256 / 224*224	2448	480
Vigne septoriose	256*256 / 224*224	2874	430
Bonne sante	256*256 / 224*224	1772	423

Tableau 3.3 Paramètres de base de données raisin**II.1.3 Pomme :**

Nous avons utilisé un total de 18939 images pour cette variété végétale. Parmi ces images, 16109 ont été consacrées à l'entraînement de notre modèle, tandis que les 2830 images restantes ont été réservées aux tests. Pour obtenir une vision plus détaillée des informations relatives aux images utilisées, nous vous prions de vous référer au **tableau 3.4** :

les classes	Les tailles des images	Le nombre des images du train	Le nombre des images du test
Oïdium	256*256 / 640*958	1035	149
Pommier black rot	256* 256 / 1365*2048 / 640*960	2805	300
Rouille de pommier	256*256 /360*360 /300*225/408*612	4380	926
Tavelure du pommier	256* 256/ 640*853 /	4525	885
Bonne sante	256* 256/ 640*853 / 531*800 /150*200 /	3664	570

Tableau 3.4 Paramètres de base de données pomme

II.1.1 Maïs :

Nous avons travaillé avec un total de 3852 images pour cette variété végétale. Parmi ces images, 3080 ont été attribuées à l'ensemble d'entraînement, tandis que les 772 images restantes ont été réservées pour l'ensemble de test. Pour obtenir une vue plus détaillée des informations relatives aux images que nous avons utilisées, veuillez-vous référer au **tableau 3.5** :

Les classes	Les tailles des images	Le nombre des images du train	Le nombre des images du test
Corn cercosporiose	256*256	410	103
Corn rouille commune	256*256	953	239
Corn septoriose	256*256	788	197
Bonne sante	256*256	929	233

Tableau 3.5 Paramètres de base de données maïs

II.1.5 Tomate :

Nous avons employé un total de 24976 images pour cette espèce végétale. Parmi celles-ci, 19832 images ont été allouées à l'ensemble d'entraînement, tandis que les 5144 images restantes ont été réservées à l'ensemble de test. Pour obtenir des informations plus détaillées sur les images que nous avons utilisées, veuillez consulter le **tableau 3.6**

Les classes	Les tailles des images	Le nombre des images du train	Le nombre des images du test
Moldiou	227*227 / 640*640 / 256*256	3113	792
Tomate septoriose	800*533 / 600*800 / 725*966	2882	746
Araignée	/ 343*500 / 350*750 /	1747	435

Bacteriose	600*900 / 300*168	2826	732
Cladosporiose	227*227 / 640*640 / 256*256 / 800*533 /	2754	739
Oidium	227*227 / 640*640 / 256*256 / 800*533 / 600*800 /	1004	252
Tomate alternaria solani	256*256 / 800*533 / 600*800 / 256*256 /	2455	643
Bonne sante	227*227 / 640*640 / 256*256	3051	

Tableau 3.6 Paramètres de base de données tomate

II.2.1 Pêche :

Nous avons travaillé avec un ensemble de 2392 images pour cette variété végétale. Parmi ces images, 1915 ont été attribuées à l'ensemble d'entraînement, tandis que les 477 images restantes ont été réservées pour l'ensemble de test. Pour obtenir des informations plus détaillées sur les images que nous avons utilisées, veuillez-vous référer au **tableau 3.7**

Les classes	Les tailles des images	Le nombre des images du train	Le nombre des images du test
Pêche bacteriose	256* 256	1655	413
Bonne sante	256* 256	260	64

Tableau 3.7 Paramètres de base de données pêche

II.1.7 Haricot :

Nous avons utilisé un total de 990 images pour cette variété végétale. Parmi ces images, 750 ont été affectées à l'ensemble d'entraînement, tandis que les 240 images restantes ont été réservées à l'ensemble de test. Pour obtenir des informations plus détaillées concernant les images que nous avons utilisées, nous vous invitons à vous référer au **tableau 3.8** suivant :

Les classes	Les tailles des images	Le nombre des images du train	Le nombre des images du test
Haricot rouille	256* 256	240	90
Tache angulaire	256* 256	255	75
Bonne sante	256* 256	255	75

Tableau 3.8 Paramètres de base de données haricot

III. Architecture du modèle et processus d'entraînement :

III.1 Environnement de travail :

Pour ce projet, nous avons choisi une plateforme Kaggle qui a joué un rôle essentiel dans le développement de notre modèle d'apprentissage profond, en fournissant une plateforme exceptionnelle pour la science des données collaborative.

Son vaste référentiel d'ensembles de données accessibles au public a rationalisé notre processus d'acquisition de données, ce qui nous a permis de nous concentrer sur les éléments essentiels de notre projet.

L'utilisation de l'environnement cloud de Kaggle, avec accès aux GPU et TPU, a considérablement accéléré l'entraînement de notre modèle et l'optimisation des hyper paramètres, ce qui nous a permis d'économiser du temps et des ressources.

La nature collaborative de la plateforme a favorisé des discussions engageantes et des forums de partage de connaissances, nous permettant partager des idées et d'apprendre d'une communauté dynamique de scientifiques et de chercheurs en données.

En outre, les riches ressources éducatives de Kaggle, y compris les tutoriels et les carnets de notes, nous ont permis d'approfondir notre compréhension des techniques avancées d'apprentissage profond.



Figure.3.1. Kaggle le guide des recherches en deep learning

III.1.1 GPU :

Un GPU (Graphics Processing Unit) est un circuit électronique spécialisé conçu pour accélérer le traitement des données visuelles et des calculs mathématiques complexes. Développés à l'origine pour gérer le rendu des images et des graphiques dans les jeux vidéo et les applications graphiques, les GPU ont évolué pour devenir de puissants processeurs parallèles capables d'effectuer des milliers d'opérations mathématiques simultanément.

Les GPU sont très efficaces dans les tâches qui impliquent des calculs répétitifs et parallèles, ce qui les rend idéaux pour certains types de charges de travail informatiques, notamment

l'apprentissage automatique, l'apprentissage profond, les simulations scientifiques et le minage de crypto-monnaies.

III.1.2 TPU :

TPU signifie Tensor Processing Unit, et il s'agit d'un accélérateur matériel spécialisé développé par Google, spécifiquement conçu pour les charges de travail d'apprentissage automatique. Plus précisément, les TPU sont optimisées pour accélérer les opérations liées aux calculs tensoriels, qui sont fondamentaux dans les modèles d'apprentissage profond.

Les TPU font partie de l'infrastructure de cloud computing de Google et sont disponibles sur Google Cloud Platform (GCP). Elles ont été introduites pour la première fois par Google en 2016 et ont depuis subi plusieurs générations d'améliorations, chacune offrant des performances et une efficacité accrue.

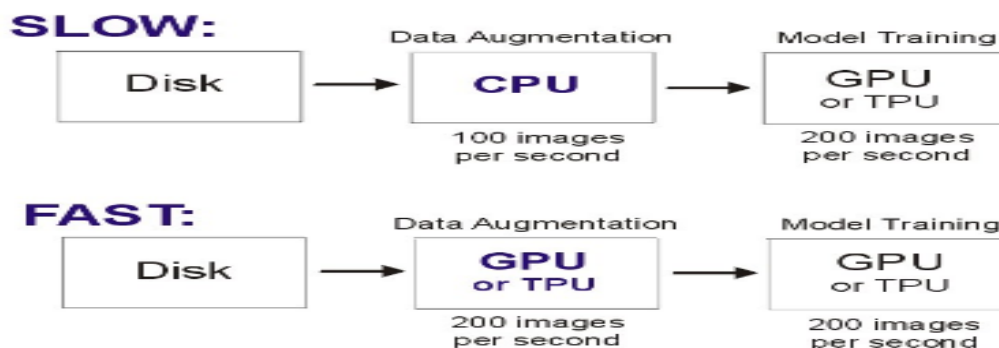


Figure.3.2. La vitesse de traitement : GPU et TPU surperformant le CPU dans l'augmentation des données

III.1 Langage utilisé :

Python : Python est un langage de programmation de haut niveau reconnu pour sa simplicité, sa lisibilité et sa polyvalence. Il a été développé à la fin des années 1980 par Guido van Rossum et a depuis gagné en popularité en raison de sa syntaxe conviviale ainsi que de sa riche bibliothèque standard.

Voici une brève présentation de Python, abordant son champ d'application, son histoire et les motifs qui justifient son utilisation :

III.2.1 Domaine d'utilisation :

Python est un langage de programmation généraliste qui peut être utilisé pour un large éventail d'applications, notamment le développement web, l'analyse de données, le calcul scientifique, l'intelligence artificielle, l'apprentissage automatique, l'automatisation, etc. Sa philosophie de conception met l'accent sur la lisibilité du code et vise à fournir une syntaxe simple et expressive qui permet aux développeurs d'écrire un code clair et concis.

III.2.2 Historiques :

Le développement de Python a débuté en décembre 1989 lorsque Guido van Rossum, un programmeur néerlandais, s'est lancé dans la création d'un nouveau langage de programmation pour succéder au langage ABC. Il a choisi de l'appeler Python en hommage au groupe comique britannique Monty Python, optant ainsi pour un nom court, unique et légèrement mystérieux. La conception de Python a été étroitement influencée par des langages tels qu'ABC, Modula-3, et d'autres, intégrant leurs meilleures caractéristiques tout en ajoutant de nouveaux concepts pour aboutir à un langage à la fois intuitif et performant. La première version publique de Python, Python 0.9.0, a vu le jour en février 1991, et depuis lors, les versions ultérieures ont continué à évoluer et à gagner en popularité au fil des ans.

Raisons du choix de Python :

- **Puissantes bibliothèques d'intelligence artificielle** : Python offre des bibliothèques de pointe pour l'intelligence artificielle, en particulier le deep learning, avec des frameworks tels que TensorFlow et Keras.
- **Lisibilité** : La syntaxe de Python est axée sur la lisibilité, ce qui facilite la compréhension et la rédaction du code. Sa structure claire et l'utilisation de l'indentation pour organiser les blocs de code améliorent la lisibilité et réduisent les erreurs potentielles.
- **Polyvalence** : Python propose une variété étendue de bibliothèques et de frameworks pour de nombreux domaines et objectifs. Il couvre le calcul scientifique (NumPy, SciPy), l'analyse de données (pandas), le développement web (Django, Flask), l'apprentissage automatique (TensorFlow, PyTorch), et bien plus encore. Cette polyvalence permet aux développeurs de travailler efficacement sur divers projets.
- **Productivité** : La simplicité et l'ergonomie de Python améliorent la productivité des développeurs. Ses abstractions de haut niveau et sa vaste bibliothèque standard fournissent des

modules prêts à l'emploi, réduisant la nécessité de coder tout depuis le début et accélérant le processus de développement.

- **Communauté et écosystème** : Python dispose d'une communauté dynamique et solidaire qui contribue activement à son développement. Avec une documentation complète, des tutoriels en ligne et des forums actifs, les développeurs peuvent apprendre plus facilement, demander de l'aide et collaborer sur des projets.

- **Intégration et interopérabilité** : Python excelle en intégration, permettant une interaction transparente avec d'autres langages et systèmes. Il peut être facilement intégré à des langages comme C/C++, Java, ce qui en fait un choix idéal pour étendre des applications existantes ou créer des interfaces avec d'autres systèmes.

En résumé, le choix de Python s'appuie sur ses bibliothèques d'IA de pointe, sa lisibilité, sa polyvalence, sa productivité, sa communauté active et son intégration facile, ce qui en fait un langage puissant et adapté à une grande variété de projets.

IV. Bibliothèques utilisées :

IV.1 TensorFlow :

TensorFlow est un puissant Framework qui permet de réaliser efficacement le calcul et l'optimisation des modèles d'apprentissage profond.

Il constitue un ensemble d'outils complet, responsable de fournir une large gamme d'outils et de ressources spécifiquement conçus pour les tâches d'apprentissage profond, incluant la création, l'entraînement et le déploiement des modèles. W[9]

IV.2 Keras :

Keras est une API d'apprentissage profond à code source ouvert écrite en Python. Elle a été développée pour fournir une interface conviviale et de haut niveau pour la construction et l'entraînement de modèles d'apprentissage profond. Keras a été conçu dans un souci de simplicité et de modularité, ce qui permet aux chercheurs et aux développeurs de créer rapidement des prototypes et d'expérimenter différentes architectures de réseaux neuronaux.

Au départ, Keras était un projet indépendant et pouvait être utilisé avec plusieurs bibliothèques dorsales, telles que Theano et Microsoft Cognitive Toolkit (CNTK). W[10]

Cependant, en 2017, Keras est devenu une partie officielle de TensorFlow, le populaire cadre d'apprentissage machine open-source développé par Google.

Depuis son intégration à TensorFlow, Keras a été inclus en tant qu'API de haut niveau de TensorFlow.

Cela signifie que Keras est désormais l'API recommandée pour la plupart des utilisateurs de TensorFlow, offrant une interface conviviale qui simplifie le processus de construction, d'entraînement et de déploiement des modèles d'apprentissage profond.

IV.2.1 Les principales caractéristiques de Keras :

- Convivialité et facilité d'utilisation : Keras fournit une API simple et intuitive, permettant aux utilisateurs de construire des modèles de réseaux neuronaux avec seulement quelques lignes de code.
- Modularité : Keras permet de personnaliser facilement les modèles, car il permet aux utilisateurs de construire des modèles en utilisant des couches prédéfinies ou de créer des couches personnalisées et de les combiner dans des architectures complexes.
- Extensibilité : Les utilisateurs peuvent étendre les fonctionnalités de Keras en définissant des fonctions de perte, des métriques et des optimiseurs personnalisés.

IV.3 Matplotlib :

Il s'agit d'une bibliothèque Python populaire permettant de créer des visualisations statiques, interactives et animées.

Elle est largement utilisée dans la science des données, la recherche scientifique, l'ingénierie et divers autres domaines pour générer des tracés et des graphiques et des diagrammes et des visualisations de haute qualité à partir de données.

IV.4 CV2 :

(OpenCV) est une bibliothèque open-source largement utilisée pour le traitement d'images et la vision par ordinateur. Elle offre des fonctionnalités avancées telles que la lecture, la manipulation et l'écriture d'images, la détection d'objets, la reconnaissance faciale, la segmentation d'image, et bien plus encore.

OpenCV est essentiel dans le domaine de l'apprentissage profond pour la préparation des données, le prétraitement des images et la réalisation de diverses opérations sur les images utilisées dans les modèles d'apprentissage profond.

IV.3 Numpy :

C'est une bibliothèque fondamentale pour le calcul scientifique en Python.

Elle fournit des structures de données efficaces pour représenter et manipuler des tableaux multidimensionnels, ainsi que des fonctions mathématiques pour effectuer des opérations sur ces tableaux.

Numpy est largement utilisée en apprentissage profond pour le traitement et la manipulation des données, notamment pour les opérations de calcul matriciel et l'optimisation des performances.

IV. Pandas :

La bibliothèque Pandas est essentielle en science des données offrant des structures de données performantes,

Telles que les Data Frames, pour la manipulation et l'analyse de données tabulaires. Pandas facilite le nettoyage, la transformation et l'exploration des données, jouant ainsi un rôle clé dans l'apprentissage profond et l'analyse de données.

```
from tensorflow import keras
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
from keras.layers import Conv2D, MaxPool2D, Dropout, Flatten, Dense, Activation, BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Figure.3.3. La déclaration des bibliothèques nécessaire.

V. La Définition du chemin d'accès à la base de données :

Dans cette section, nous établirons la définition du chemin (path) qui permet d'accéder à la base de données.

```
trainpath= '/kaggle/input/fraise/Fraise/Fraise/train/'
testpath= '/kaggle/input/fraise/Fraise/Fraise/test/'
randompath= '/kaggle/input/fraise/New folder/'
```

Figure.3.4. Définir le chemin pour le model du frais.

VI. L'augmentation des données :

L'augmentation des données est cruciale dans l'apprentissage profond pour faire face aux données d'entraînement limitées en élargissant artificiellement l'ensemble de données. Elle permet d'améliorer la généralisation du modèle et de réduire le surajustement en introduisant des variations grâce à des techniques telles que la rotation, la mise à l'échelle et le retournement. L'augmentation des données renforce la capacité du modèle à apprendre divers modèles et améliore ses performances sur des données inédites.

```
[8]: datagen = ImageDataGenerator(
      rotation_range=20,
      width_shift_range=0.1,
      height_shift_range=0.1,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      vertical_flip=True,
      featurewise_std_normalization=True,
      fill_mode='nearest')
```

Figure.3.5. L'augmentation des données.

VII. Les techniques d'augmentation des données que nous avons utilisées :

Rotation : La rotation d'une image permet au modèle d'apprendre à reconnaître des objets à partir de différents points de vue, améliorant ainsi sa capacité à s'adapter à des orientations et des angles différents.

Shift : Le déplacement horizontal ou vertical d'une image introduit des variations spatiales, ce qui permet au modèle de mieux comprendre les positions et les translations des objets dans différents contextes.

Shear : L'application de transformations de cisaillement à une image permet au modèle d'apprendre à reconnaître des objets de formes et de distorsions différentes, améliorant ainsi sa résistance aux déformations.

Zoom : le zoom avant ou arrière d'une image fournit les modèles différentes échelles et différents niveaux de détail, améliorant ainsi sa capacité à identifier des objets de tailles et de distances variées.

Flip : Le retournement horizontal ou vertical d'une image accroît la diversité de l'ensemble de données, ce qui permet au modèle d'apprendre à partir de versions miroir ou inversées d'objets, qui peuvent être présentes dans des scénarios du monde réel.

Feature std normalisation : La normalisation des caractéristiques, telle que la normalisation de la moyenne et la mise à l'échelle de l'écart type, permet de prétraiter les données d'entrée en les centrant autour de zéro et en les mettant à l'échelle dans une plage comparable. Cette Normalisation améliore la convergence et la stabilité du modèle pendant la formation.

Original Image

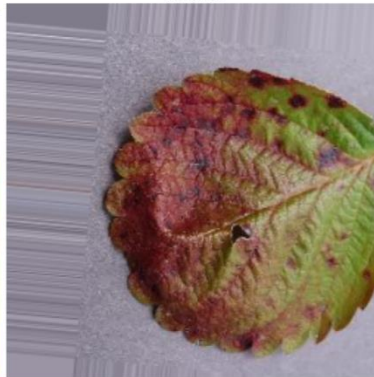


Figure .3.6.L'image original (sans effets)

Rotated



Sheared



Width Shift



Horizontally Flipped



Vertically Flipped



Height Shift

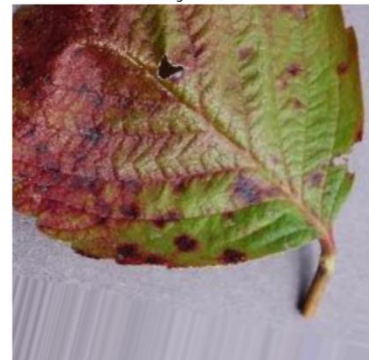


Figure .3.7.L'image après effets appliqué

VIII. Le redimensionnement des images :

Le redimensionnement des images est un processus essentiel dans le domaine de l'apprentissage profond appliqué à l'identification des maladies végétales. Les images de la base de données peuvent présenter des tailles variables, ce qui peut impacter négativement l'efficacité de l'apprentissage des modèles. Le redimensionnement vise à uniformiser les dimensions de ces images, offrant ainsi plusieurs avantages.

Tout d'abord, cela facilite l'utilisation des ressources informatiques en rendant les données compatibles avec les architectures de modèles d'apprentissage profond. En conséquence, le processus d'entraînement est plus rapide et plus efficace. De plus, le redimensionnement améliore la capacité d'apprentissage des modèles en minimisant les perturbations dues à des images de tailles différentes.

En uniformisant les dimensions, il devient plus facile de comparer et d'analyser les caractéristiques des images. Cela conduit à des modèles plus cohérents et performants, capables de généraliser avec précision.

En somme, le redimensionnement des images assure la cohérence et l'efficacité des modèles d'apprentissage profond, contribuant ainsi à la réussite de l'identification des maladies végétales à partir d'images de feuilles.

```
s=150
for folder in os.listdir(trainpath):
    files = gb.glob(pathname= str ( trainpath +folder + '/*'))
    for file in files :
        image = cv2.imread(file)
        image_array = cv2.resize(image)
        x_train.append(list(image_array))
        y_train.append(code)
```

Figure.3.8. Redimensionnent les images de train.

```
s=150
for folder in os.listdir(testpath):
    files = gb.glob(pathname= str ( testpath +folder + '/*'))
    for file in files :
        image = cv2.imread(file)
        image_array = cv2.resize(image)
        x_test.append(list(image_array))
        y_test.append(code)
```

Figure.3.9. Redimensionnent les images de test.

IX. La création de Réseau de neurones convolutifs (CNN)

Dans cette section, nous explorons en détail l'architecture du réseau neuronal de notre modèle d'apprentissage profond.

Nous explorons la disposition et la configuration des couches essentielles, à l'aide de différents paramètres, ce sont :

Conv2D : CONV2D est l'abréviation de "Convolutional 2D" en anglais, ce qui se traduit littéralement par "convolution en 2 dimensions". C'est un terme couramment utilisé dans le domaine du traitement d'images et de la vision par ordinateur, en particulier dans le contexte des réseaux de neurones convolutionnels (CNNs - Convolutional Neural Networks).

La convolution en 2D est une opération mathématique fondamentale utilisée pour le traitement d'images. Elle implique la superposition d'un petit noyau (ou filtre) sur l'image d'entrée pour produire une nouvelle image (appelée feature map) en appliquant un processus de multiplication et d'addition.

IX.1 Principe de fonctionnement de la Convolution :

La convolution est une opération fondamentale dans les réseaux de neurones convolutifs (CNN), utilisée pour extraire des caractéristiques importantes des images. Voici comment elle fonctionne :

- **Noyau (Filtre)** : Le noyau est une petite matrice, souvent de dimensions 3x3, contenant des valeurs de poids. Ces valeurs déterminent comment l'information sera extraite de l'image d'entrée.
- **Superposition** : Le noyau est superposé sur chaque région de l'image d'entrée. On place initialement le centre du noyau sur le premier pixel de l'image, puis on le fait glisser sur l'image en effectuant des multiplications élémentaires entre les valeurs du noyau et les valeurs des pixels sous le noyau.
- **Multiplication et Addition** : À chaque superposition, les valeurs des pixels de l'image d'entrée qui sont couverts par le noyau sont multipliées par les valeurs correspondantes du noyau. Ensuite, toutes ces valeurs sont additionnées pour obtenir une valeur unique dans la "feature map" (carte de caractéristiques).
- **Stride (Pas)** : On peut spécifier un pas (stride) pour le déplacement du noyau lors de la convolution. Un stride de 1 signifie que le noyau se déplace d'un pixel à la fois, tandis qu'un stride plus grand permet un déplacement plus rapide.

- **Padding (Bourrage) :** Pour éviter que la taille de la "feature map" ne diminue après chaque convolution, on peut ajouter un bourrage (padding) de zéros autour de l'image d'entrée.

En utilisant les notations :

- I pour l'image d'entrée,
- K pour le noyau (filtre) de convolution,
- O pour la "feature map" (image de sortie),
- S pour le pas (stride) de convolution, et P pour le bourrage (padding),

La convolution pour chaque position (x, y) de l'image de sortie O (feature map) est calculée comme suit :

$$I(x \times Sx + i - (P-1) / 2, y \times Sy + j - (Q-1) / 2)$$

$$(x \times Sx + i - (P-1) / 2, y \times Sy + j - (Q-1) / 2).$$

$$K(i, j)$$

$$(P-1) / 2 \text{ et } (Q-1) / 2$$

En résumé, la convolution avec ses étapes de superposition, multiplication et addition, ainsi que ses paramètres tels que le pas et le bourrage, joue un rôle essentiel dans l'extraction des caractéristiques des images, ce qui est crucial pour le succès des réseaux de neurones convolutifs.

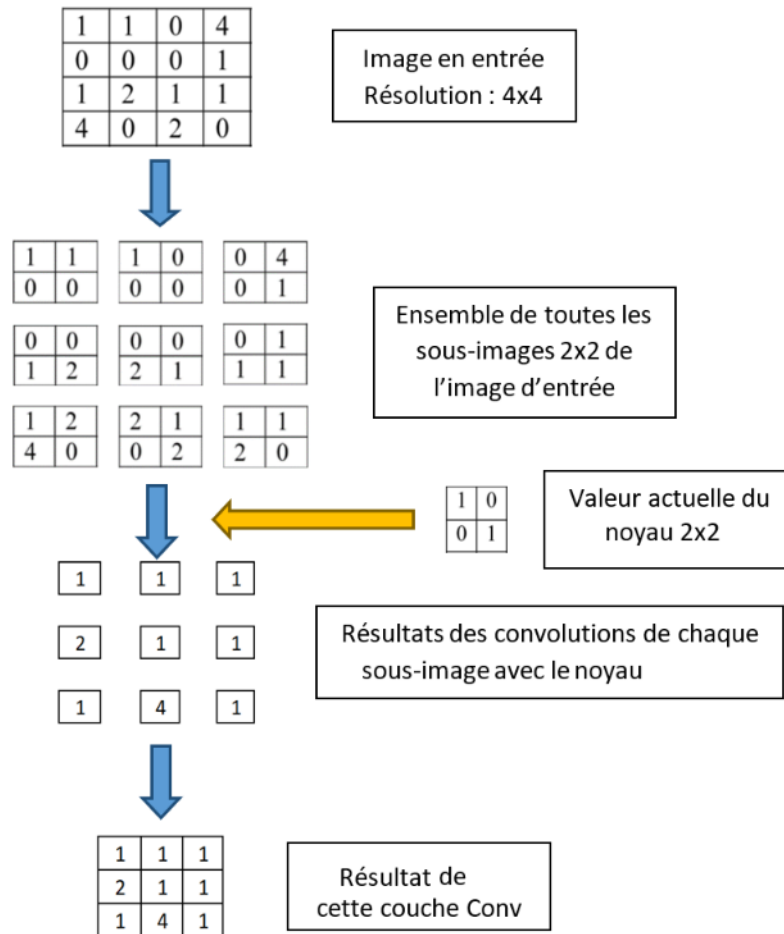


Figure.3.10. Le fonctionnement de CONV2D en utilisant une image 4x4 et un noyau 2x2.

[W11]

- **Maxpool2D :** Maxpool2D signifie "Max Pooling en 2 dimensions". Le Max Pooling est une opération très courante qui intervient après les couches de convolution dans un CNN. Son rôle est de réduire la dimension spatiale de la représentation des caractéristiques (feature map) tout en conservant les informations les plus importantes.
- **Principe de fonctionnement :**
 - Fenêtre de pooling : Le Max Pooling est effectué à l'aide d'une fenêtre (dans notre cas 2x2 et 4x4) qui se déplace sur la carte des fonctionnalités produite par la couche de convolution. La fenêtre se déplace généralement avec un pas (stride) de 2, ce qui réduit la taille de la représentation.

- Valeur maximale : Pour chaque position de la fenêtre, le Max Pooling sélectionne la valeur maximale présente dans la fenêtre. Cela signifie qu'une seule valeur est extraite de la fenêtre, qui correspond à la caractéristique la plus dominante dans cette zone.
- Sous-échantillonnage : En utilisant cette opération de sélection du maximum, le Max Pooling réduit la taille de la carte des fonctionnalités. Cela permet de réduire le nombre de paramètres du modèle et de simplifier l'apprentissage, tout en conservant les informations importantes sur les caractéristiques les plus saillantes de l'image.

L'objectif du Max Pooling est double : il permet d'obtenir une représentation plus compacte de l'image, ce qui réduit les exigences de mémoire et de calcul du modèle, tout en créant une invariance partielle aux petites translations de l'objet dans l'image. En d'autres termes, même si l'objet se déplace légèrement dans l'image, le Max Pooling garantit que la représentation des caractéristiques reste inchangée, car seule la valeur maximale est conservée.

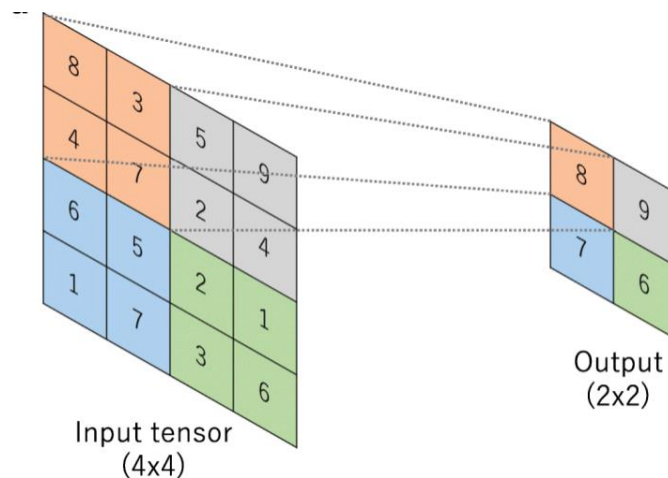


Figure.3.11. Fonctionnement simplifié de la technologie Maxpooling. [W12]

Le Dropout est une stratégie de régularisation largement adoptée pour améliorer les performances et la généralisation des réseaux de neurones, en particulier dans les structures profondes comme les CNN (réseaux de neurones convolutionnels) et les RNN (réseaux de neurones récurrents).

L'idée fondamentale derrière le Dropout est de désactiver aléatoirement certains neurones pendant l'apprentissage. Ces neurones désactivés sont temporairement ignorés lors de la propagation en avant et en arrière de l'erreur. Le Dropout est appliqué exclusivement durant l'apprentissage, et non lors de l'évaluation ou de l'utilisation du modèle.

IX.1.1 Fonctionnement du Dropout :

Pendant l'apprentissage : À chaque itération d'apprentissage, pour chaque couche cachée, chaque neurone a une probabilité fixe p (un hyperparamètre) d'être désactivé. Par exemple, on peut avoir $p = 0,5$.

Désactivation des neurones : Les neurones choisis pour le dropout ont leur sortie mise à zéro durant la propagation en avant. En d'autres termes, ils ne contribuent pas au transfert du signal aux couches suivantes.

Renforcement de la robustesse : Étant donné que certains neurones sont désactivés, les autres neurones doivent compenser leur travail, ce qui les amène à apprendre des caractéristiques plus robustes. Cela peut contribuer à améliorer la généralisation du modèle.

Le Dropout opère comme une forme d'échantillonnage aléatoire de sous-réseaux à chaque itération, introduisant de l'incertitude et empêchant une dépendance excessive à certains neurones spécifiques. Cette technique réduit le risque de surapprentissage en encourageant le réseau à apprendre des caractéristiques plus robustes, applicables à diverses parties du jeu de données.

Pendant l'évaluation : Lors de l'évaluation (prédictions), aucun neurone n'est désactivé. Cependant, les poids des neurones sont ajustés en prenant en compte la probabilité de désactivation utilisée pendant l'apprentissage.

IX.1.2 Formule du Dropout :

Durant l'apprentissage, pour chaque neurone i dans une couche cachée avec une probabilité de dropout p , l'opération de Dropout peut être exprimée comme suit :

$$\begin{aligned} r_i &\sim \text{Bernoulli}(p) \\ h_i' &= h_i * (1 - p) \quad \text{si } r_i = 1 \\ h_i' &= 0 \quad \quad \quad \text{si } r_i = 0 \end{aligned}$$

Ici, $\text{Bernoulli}(p)$ est une distribution de Bernoulli avec une probabilité p . Pour chaque neurone, une valeur binaire aléatoire est échantillonnée (0 ou 1) selon la probabilité p . Si le résultat est 1, le neurone est activé et sa sortie est réduite en multipliant par $(1 - p)$ pour maintenir la moyenne. Si le résultat est 0, le neurone est désactivé et sa sortie est mise à zéro.

En conclusion, le Dropout est une stratégie de régularisation puissante qui favorise la robustesse du modèle, réduisant ainsi le surapprentissage et améliorant la capacité de généralisation des réseaux de neurones profonds.

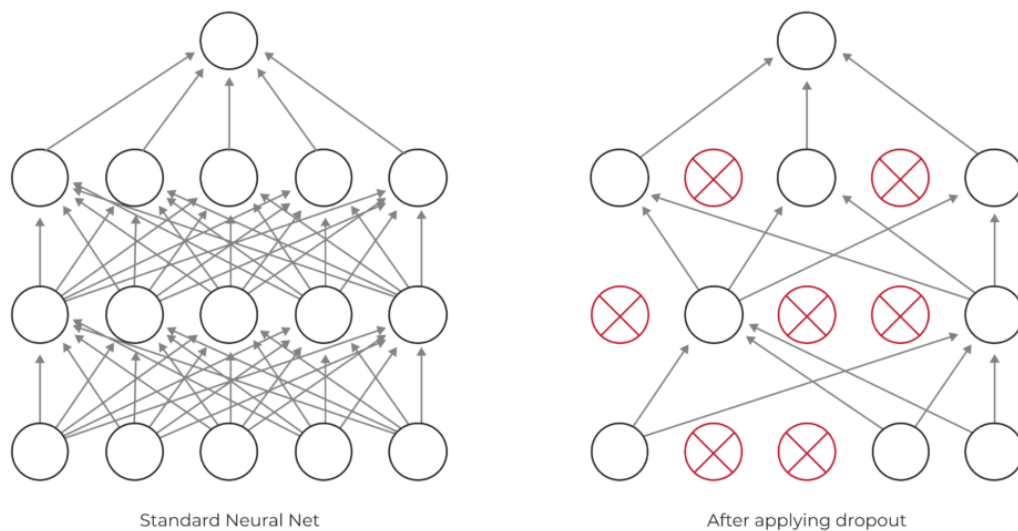


Figure.3.12. L'effet d'application du Dropout sur le réseau neuronal. [W13]

F

Flatten : dans les réseaux de neurones convolutionnels (CNNs) et les réseaux de neurones entièrement connectés (Fully Connected Networks). L'opération Flatten est utilisée pour transformer une matrice multidimensionnelle en un vecteur à une dimension.

Supposons que nous ayons une couche de sortie ou une couche intermédiaire dans notre réseau de neurones qui est représentée sous la forme d'un tenseur multidimensionnel, généralement de dimensions (batch_size, height, width, channels), où :

batch_size : Le nombre d'exemples dans un lot (batch) d'entraînement.

height : La hauteur des cartes de caractéristiques (feature maps) dans la couche.

width : La largeur des cartes de caractéristiques dans la couche.

Channels : Le nombre de canaux dans chaque carte de caractéristiques (par exemple, 3 pour les images RGB).

L'opération Flatten transforme ce tenseur en un vecteur unidimensionnel de longueur $\text{batch_size} * \text{height} * \text{width} * \text{channels}$.

▫ **Principe de fonctionnement :**

Si nous notons le tenseur d'entrée de la couche comme T, et le vecteur résultant après l'application de l'opération Flatten comme V, alors l'opération Flatten est définie comme suit :

$$V = \text{Flatten}(T)$$

Pour expliquer en termes simples, l'opération Flatten "aplatit" essentiellement le tenseur multidimensionnel en un seul vecteur en alignant les éléments de chaque canal dans une seule dimension. Cela permet d'utiliser les sorties de cette couche comme entrée dans une couche entièrement connectée (fully connected layer) ou dans d'autres couches qui nécessitent une entrée unidimensionnelle.

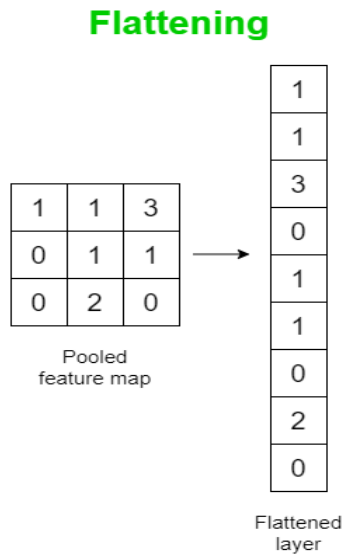


Figure.3.13. Transformation d'une matrice multidimensionnelle en un vecteur à une dimension. [W14]

Dense : Dense est un terme utilisé dans les réseaux de neurones profonds pour désigner des couches de réseau de neurones entièrement connectées. Dans une couche Dense, chaque neurone est connecté à chaque neurone de la couche précédente, formant une connexion dense (d'où le nom "Dense"). Chaque neurone d'une couche dense prend l'ensemble de sorties de la couche précédente (ou l'entrée réseau de la première couche dense), et effectue une combinaison linéaire de ces entrées pondérées par les poids du neurone, puis applique une fonction d'activation pour produire la sortie du neurone.

▫ **Principe de fonctionnement :**

Supposons que nous ayons une couche Dense avec n neurones. Soit x l'entrée du réseau, et les poids de chaque neurone de cette couche sont représentés par un vecteur de poids

' W_i ' Et un biais ' b_i ' pour chaque neurone (i allant de 1 à n). La sortie ' y_i ' du neurone i dans la couche Dense est calculée comme suit :

$$y_i = \text{activation}(W_i \cdot x + b_i)$$

Où :

- , $W_i \cdot x$ représente le produit scalaire entre le vecteur de poids ' W_i ' du neurone i et l'entrée x
- ' b_i ' est le biais du neurone i .
- Activation (\cdot) est la fonction d'activation appliquée au résultat de la combinaison linéaire. La fonction d'activation introduit de la non-linéarité dans le modèle, ce qui permet au réseau de neurones d'apprendre des relations plus complexes entre les données.

La sortie de la couche Dense est un vecteur de longueur n , contenant les sorties de chaque neurone de la couche. Ce processus est répété pour chaque couche Dense dans le réseau de neurones.

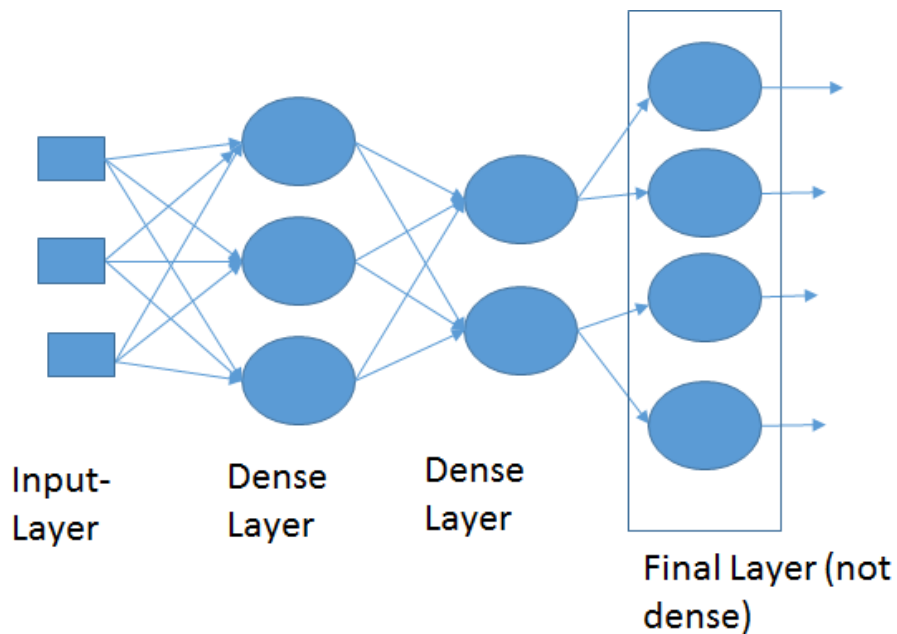


Figure.3.14. Les couches Dense [W15]

Activation : l'activation (ou fonction d'activation) fait référence à une fonction non linéaire appliquée à la sortie d'un neurone ou d'une couche de neurones dans un réseau de neurones. L'activation est un élément essentiel dans les réseaux de neurones car elle introduit de la non-

linéarité dans le modèle, permettant ainsi au réseau de neurones d'apprendre des relations plus complexes entre les données.

Lorsque l'on parle d'un neurone, la sortie du neurone est calculée comme une combinaison linéaire des entrées pondérées par les poids du neurone, et cette somme pondérée est ensuite passée à travers une fonction d'activation. Mathématiquement, pour un neurone avec des entrées x , des poids w et un biais b , la sortie y est donnée par :

$$y = \text{activation}(w \cdot x + b)$$

La fonction d'activation prend en entrée le résultat de la combinaison linéaire ($w \cdot x + b$) et produit la sortie du neurone après l'avoir transformée de manière non linéaire. Sans fonction d'activation, les couches de neurones successives dans un réseau de neurones seraient équivalentes à une seule couche, car la combinaison linéaire de couches linéaires est elle-même une fonction linéaire.

1. Activation ReLU (Rectified Linear Unit) : La fonction d'activation ReLU est l'une des plus simples et des plus populaires dans les réseaux de neurones. Elle est définie comme suit :

$$\text{ReLU}(x) = \max(0, x)$$

En d'autres termes, si la valeur d'entrée (x) est positive, la fonction ReLU renverra cette valeur. Si la valeur d'entrée est négative ou nulle, elle renverra simplement 0.

L'avantage principal de ReLU réside dans sa simplicité et son efficacité de calcul. Elle permet également de surmonter le problème du gradient disparu (vanishing gradient) dans les réseaux de neurones profonds, car sa dérivée est soit 1 (pour $x > 0$) soit 0 (pour $x \leq 0$). Ainsi, lors de la rétropropagation du gradient, elle ne réduit pas fortement le gradient comme cela peut se produire avec d'autres fonctions d'activation (comme la fonction sigmoïde).

2. Activation Softmax : La fonction d'activation Softmax est couramment utilisée pour les tâches de classification multiclasse dans lesquelles un modèle doit choisir entre plusieurs classes possibles. Elle prend un vecteur de scores en entrée et les convertit en probabilités normalisées.

Pour un vecteur d'entrée de taille n , la fonction Softmax est définie comme suit pour chaque élément i du vecteur :

$$\text{Softmax}(x_i) = \exp(x_i) / \sum(\exp(x_j)) \text{ for } j = 1 \text{ to } n$$

Ici, $\exp(x_i)$ représente l'exponentielle de l'élément x_i du vecteur d'entrée, et la somme est prise sur tous les éléments du vecteur d'entrée.

Ici, $\exp(x_i)$ représente l'exponentielle de l'élément x_i du vecteur d'entrée, et la somme est prise sur tous les éléments du vecteur d'entrée.

L'activation Softmax est utilisée pour obtenir des probabilités normalisées pour chaque classe, de sorte que la somme de toutes les probabilités soit égale à 1. Cela facilite l'interprétation du modèle pour une tâche de classification, car la classe avec la probabilité la plus élevée est choisie comme prédiction finale.

X. Explication et analyse de notre CNN (Convolutional neural network) :

L'architecture de réseau de neurones joue un rôle crucial dans la capture de modèles complexes et l'extraction de représentations significatives à partir des données d'entrée, ce qui permet au modèle d'apprendre et de faire des prédictions précises.

```
model = Sequential()
model.add(Conv2D(128, kernel_size=(3,3), padding='same', activation='relu', input_shape=(s,s,3)))
model.add(MaxPool2D(2,2))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPool2D(4,4))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(2, activation='softmax'))
model.summary()
```

Figure.3.15. L'architecture de réseau de neurones convolutifs.

- La 1ère couche : la première couche convolutive, qui se compose de 128 filtres de taille 3x3. Ces filtres glissent sur l'image d'entrée de 150x150 pixels avec trois canaux de couleur (RVB). La fonction d'activation utilisée dans cette couche est Relu qui introduit une non-linéarité dans le processus d'extraction des caractéristiques. En sortie de la première couche

convolutive, la couche MaxPooling2D effectue un max-pooling avec une taille de pool de 2x2, réduisant ainsi Les dimensions spatiales des cartes de caractéristiques. Cette étape permet de réduire la complexité de calcul et rend le réseau plus robuste aux variations de l'entrée. Après la mise en commun maximale, la couche de normalisation par lots normalise les activations de la couche précédente, ce qui stabilise et accélère le processus du training en réduisant le décalage interne des covariables. Cette étape de normalisation contribue à une optimisation plus douce pendant la rétropropagation. Pour améliorer encore la généralisabilité du réseau et empêcher l'ajustement excessif, une couche d'exclusion est appliquée, mettant aléatoirement 20 % des entrées à zéro pendant le training. Cette technique d'exclusion améliore la capacité du modèle à se généraliser et réduit sa dépendance à l'égard de caractéristiques spécifiques.

- 2eme couche : la deuxième couche convolutive est introduite avec 64 filtres de taille 3x3. Comme la première couche convolutive, cette couche utilise la fonction d'activation Relu pour capturer des motifs complexes dans les données d'entrée. Ensuite, une autre couche MaxPooling2D réduit encore les dimensions spatiales, ce qui permet d'extraire davantage de caractéristiques de haut niveau des représentations sous-échantillonnées. Pour maintenir la robustesse pendant le training, une autre couche BatchNormalization est appliquée. La normalisation par lots permet de maintenir les activations dans une fourchette stable, évitant ainsi le problème du gradient qui s'évanouit. En outre, une autre couche d'élimination avec un taux de 20 % est employée pour continuer à régulariser le modèle. Après les couches convolutives, une couche d'aplatissement est introduite pour convertir la sortie des couches convolutives en un tableau 1D, préparant ainsi les données pour les couches entièrement connectées (denses).
- 3eme couche : cette couche est constituée de 3 couches de dense sont : La première couche dense se compose de 128 neurones avec activation Relu. En tant que couche entièrement connectée, elle traite la représentation aplatie des caractéristiques et introduit la non-linéarité, ce qui permet au modèle d'apprendre des relations complexes entre les caractéristiques. La deuxième couche dense, avec 64 neurones et une activation Relu, traite davantage les caractéristiques extraites, contribuant ainsi à l'apprentissage de la représentation hiérarchique. La couche de sortie est la dernière couche du modèle, elle comporte deux neurones, représentant les deux classes. La fonction d'activation Softmax

est appliquée, ce qui produit des probabilités de classe, indiquant la probabilité que l'entrée appartienne à chaque classe.

X. Les caractéristiques de notre réseau de neurones :

Voici un aperçu des caractéristiques de notre réseau de neurones pour différentes variétés végétales :

	Entrée	Nombre de couches cachées	Sortie
Fraise	150x150x3	2	2
Pomme	150x150x3	2	5
Pomme de terre	150x150x3	2	3
Tomate	126x126x3	2	8
Raisin	150x150x3	2	4
Haricot	256x256x3	3	3

Pêche	150x150x3	2	2
Mais	150X150X3	2	4

Tableau 3.9. Les caractéristiques de notre réseau de neurones pour différentes variétés végétales

L'entrée (input_shape) est définie comme (s, s, 3) dans le programme, où 's' est une variable. Cela signifie que l'entrée du modèle est un tenseur tridimensionnel avec 's' lignes, 's' colonnes et 3 canaux. Les 3 canaux représentent les valeurs RVB (Rouge, Vert, Bleu) de l'image.

Pour le nombre de sorties, il s'agit du nombre de classes possibles pour déterminer la classification finale. Cela indique combien de catégories distinctes notre modèle est capable de prédire.

XI. Résultats et discussion :

XI.1 Analyse de la progression d'entraînement du system :

XI.1.1 Résultats et Discussion : Analyse de la Progression d'Entraînement

L'analyse de la progression de l'entraînement de notre modèle de Deep Learning pour la classification d'images de raisin est présentée ci-dessous :

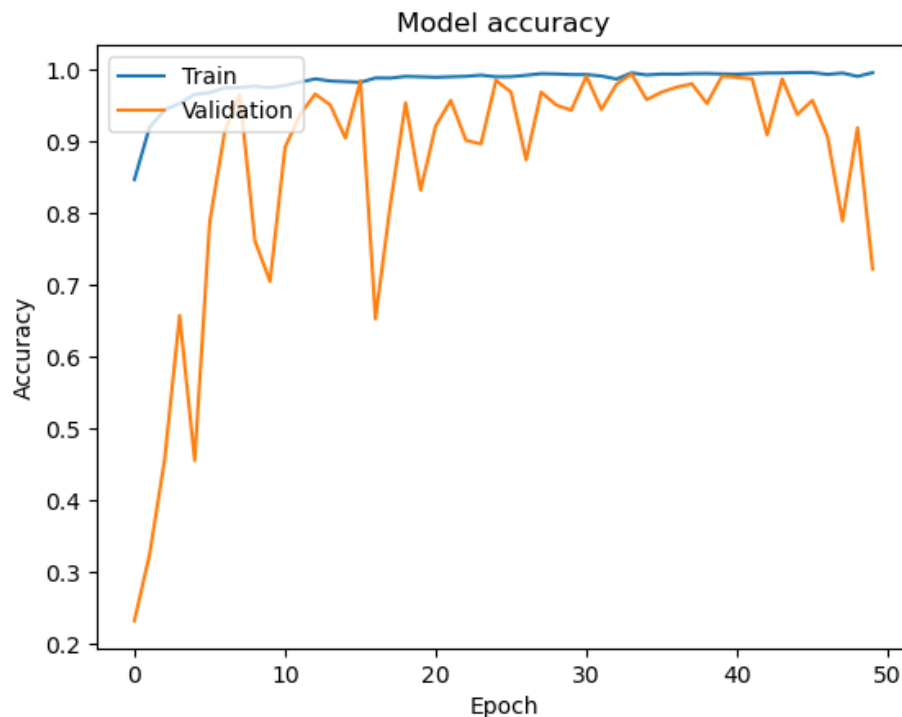


Figure.3.16. L'évolution de la précision du modèle

Époque 1/50 :

- La première époque marque le début de l'entraînement. À ce stade, le modèle commence avec des poids initialisés aléatoirement.
- Le temps d'exécution de cette époque est indiqué.
- Un avertissement de TensorFlow concernant la mise en page peut apparaître, indiquant un potentiel problème de dimensionnement des tenseurs. Cela peut ne pas avoir un impact majeur sur l'entraînement.

Époques 2 à 50 :

- Les époques suivantes illustrent la progression continue de l'entraînement.
- "Loss" (perte) représente la valeur de la fonction de perte, mesurant la divergence entre les prédictions du modèle et les vraies étiquettes. L'objectif est de minimiser cette perte.
- "Accuracy" (précision) quantifie la performance du modèle sur les données d'entraînement, exprimée en pourcentage.
- "Val_loss" (perte de validation) et "Val_accuracy" (précision de validation) sont les mêmes métriques, mais calculées sur l'ensemble de validation. L'ensemble de validation n'est pas

utilisé pour l'entraînement et permet d'évaluer la performance du modèle sur des données qu'il n'a pas encore rencontrées.

Cette sortie d'entraînement permet de suivre la progression de notre modèle à travers chaque époque. En observant les métriques de perte et de précision, nous pouvons évaluer la convergence de l'entraînement et déterminer si le modèle améliore sa capacité à faire des prédictions précises au fil des époques. La comparaison entre les métriques d'entraînement et de validation aide à surveiller tout surapprentissage potentiel, où le modèle pourrait performer de manière excessive sur les données d'entraînement mais moins bien sur de nouvelles données.

Cette analyse de la progression d'entraînement est essentielle pour comprendre comment notre modèle apprend au fil du temps et comment il généralise sur de nouvelles données. Cela nous permet de prendre des décisions informées concernant l'ajustement des hyperparamètres et la durée de l'entraînement pour obtenir les meilleurs résultats.

En résumé L'analyse de la progression d'entraînement de notre modèle pour la classification d'images de raisin révèle les tendances suivantes :

1. Les valeurs de "loss" (perte) diminuent au fil des époques, indiquant que le modèle améliore sa capacité à minimiser l'écart entre les prédictions et les vraies valeurs cibles.
2. Les valeurs de "accuracy" (précision) augmentent progressivement, ce qui suggère que le modèle apprend à mieux prédire les classes correctes au fur et à mesure de l'entraînement.
3. Les valeurs de "val_loss" (perte de validation) et "val_accuracy" (précision de validation) suivent des tendances similaires à celles de l'entraînement. Cependant, elles peuvent présenter des fluctuations occasionnelles.
4. Les fluctuations dans les valeurs de "val_loss" et "val_accuracy" peuvent indiquer un léger surapprentissage (overfitting), où le modèle peut avoir tendance à bien performer sur les données d'entraînement mais moins bien sur de nouvelles données.
5. Les baisses soudaines dans la précision de validation, comme observé à l'époque 24, peuvent être des points à surveiller. Ces baisses pourraient indiquer des problèmes temporaires ou des fluctuations liées à l'ensemble de validation.

XI.2 Évaluation des Performances de Reconnaissance

Le taux de reconnaissance pour chaque type de plante a été évalué à travers des expériences d'apprentissage profond. Les résultats ont été compilés dans le tableau ci-dessous, mettant en évidence la performance de chaque modèle de classification. Les taux de reconnaissance illustrent

la capacité de chaque modèle à prédire avec précision les classes correspondantes pour les différentes plantes. Ces taux de reconnaissance offrent un aperçu des forces et des défis rencontrés par chaque modèle dans la tâche de classification des maladies végétales.

Type de Plante	Taux de Reconnaissance
Fraise	99%
Pomme	93%
Pomme de terre	97%
Tomate	92%
Raisin	99%
Haricot	85%
Pêche	99%
Maïs	93%

Tableau 3.10. taux de reconnaissance pour chaque type de plante

Ce tableau offre une vue d'ensemble des performances relatives des modèles sur chaque type de plante. Les taux de reconnaissance démontrent les résultats obtenus en termes de prédiction précise des maladies végétales, reflétant les spécificités de chaque modèle dans sa capacité à généraliser et à distinguer entre les différentes classes.

Exemples Illustrant l'Efficacité de Notre Système

Après avoir procédé au calcul de reconnaissance pour chaque modèle, nous avons entrepris de mettre à l'épreuve notre système en l'appliquant à diverses images, tant saines que pathologiques. Les résultats obtenus ont souligné l'efficacité de notre approche dans la détection précise des maladies des plantes.

En intégrant la détection précise des maladies des plantes, notre système offre des solutions pratiques pour une gestion efficace. Une fois qu'une maladie est identifiée, le système fournit des recommandations personnalisées pour l'utilisation de produits chimiques ciblés, tout en

encourageant des solutions alternatives telles que l'introduction d'organismes auxiliaires et des pratiques agricoles biologiques. De plus, le système assure un suivi personnalisé en fournissant des conseils adaptés à la culture spécifique de chaque utilisateur. En combinant la détection proactive avec des solutions concrètes, notre système vise à être un atout complet pour les professionnels de l'agriculture, contribuant ainsi à une agriculture plus durable et résiliente.

Maladie: Maladie de la Tache des Feuilles (Tache Isariopsis)
Produits Chimiques: Fongicide E, Fongicide F



Maladie: Maladie de la Pourriture Noire
Produits Chimiques: Fongicide A, Fongicide B



Figure.3.17.Test sur les feuilles de raisin

Maladie: Plante Saine
Traitements: Pas de traitement requis



Maladie: Brûlure des Feuilles
Traitements: Gestion de l'arrosage, Paillis organique



Figure.3.18.Test sur les feuilles de Fraise

XII. Conclusion :

La progression de l'entraînement de notre modèle témoigne d'une amélioration graduelle au fil des époques. Dès le départ, le modèle démontre une précision raisonnable. Les valeurs de perte dans la phase de validation sont généralement légèrement supérieures à celles de l'entraînement, ce qui est conforme à une situation typique. Notamment, le modèle présente une tendance positive marquée, renforçant ainsi la fiabilité et la stabilité du système lors des tests. Cette confiance est solidifiée par les résultats probants que le système a générés pour diagnostiquer plusieurs maladies. Cependant, ces résultats ne marquent que le début de notre parcours. Dans le prochain chapitre, le Chapitre 4, nous envisageons de prendre cette recherche un pas plus loin en explorant les perspectives d'application pratique de notre travail, discutant des développements nécessaires pour transformer cette avancée en une application concrète.

Chapitre IV

Présentation de L'application

I. Introduction

Dans le domaine de l'agriculture et de la santé des plantes, la détection précoce des maladies est essentielle pour prévenir les pertes de récolte et assurer la sécurité alimentaire. Nous avons déjà développé un modèle de détection des maladies des plantes à l'aide de techniques d'apprentissage en profondeur, mais pour qu'il soit véritablement utile, il doit être accessible et utilisable par un large éventail d'utilisateurs, qu'ils soient agriculteurs, chercheurs ou passionnés de jardinage. C'est dans ce contexte que ce chapitre vise à transformer notre modèle de détection en une application conviviale, accessible via le web ou sur des appareils mobiles.

L'objectif de ce chapitre est de guider les lecteurs à travers le processus de conversion de notre modèle en une application pratique et conviviale, en utilisant les technologies Python et Visual Studio. Cette étape est cruciale pour rendre la puissance de la technologie de détection des maladies accessible à tous, sans nécessiter de compétences avancées en programmation ou en apprentissage automatique. Grâce à cette transformation, les utilisateurs pourront facilement télécharger l'application, capturer des images de leurs plantes et obtenir des résultats de détection en temps réel.

Il est essentiel de comprendre que la conversion du modèle en application va bien au-delà de la simple interface utilisateur. Cela implique la création d'une expérience utilisateur fluide et intuitive, ainsi que l'intégration de fonctionnalités telles que la prise de photos, la communication avec le serveur et l'affichage des résultats de détection. Cette transformation joue un rôle clé dans la mise en œuvre pratique du modèle et dans la résolution de problèmes réels auxquels sont confrontés les amateurs de jardinage et les professionnels de l'agriculture.

Tout au long de ce chapitre, nous allons explorer les différentes étapes nécessaires pour convertir notre modèle en une application conviviale. Nous aborderons les aspects techniques de la création d'une interface utilisateur, de l'intégration avec le modèle de détection des maladies et de la mise en œuvre de fonctionnalités avancées. En fin de compte, l'objectif est de fournir aux utilisateurs une solution complète et accessible pour la détection précoce des maladies des plantes, contribuant ainsi à la promotion d'une agriculture durable et d'une santé végétale optimale.

II. Préparation de l'Environnement de Développement

Pour commencer le processus de conversion de notre modèle de détection des maladies en une application conviviale, il est essentiel de mettre en place un environnement de développement approprié. Dans cette section, nous allons explorer les étapes pour installer les outils nécessaires, à savoir Python et Visual Studio, afin de pouvoir procéder à la transformation en toute fluidité.

II.1 Visual Studio:

Visual Studio est un environnement de développement intégré (IDE) créé par Microsoft. C'est un ensemble d'outils puissants destinés aux développeurs pour la création, la conception, le débogage et le déploiement d'applications logicielles. Il prend en charge une variété de langages de programmation, de plates-formes et de technologies, et offre des fonctionnalités telles que la coloration syntaxique, l'auto complétion, le débogage avancé, la gestion de version, et bien plus encore. Visual Studio facilite le développement d'applications pour diverses plates-formes, notamment les applications de bureau, les applications web, les applications mobiles, les jeux et les applications cloud.

II.2 Installation de Python et Visual Studio

La première étape consiste à installer Python, le langage de programmation qui sera au cœur de notre application, ainsi que Visual Studio, l'environnement de développement intégré (IDE) qui facilitera le processus de développement. Suivez ces étapes pour installer les deux composants :

A. Installation de Python :

1. Rendez-vous sur le site officiel de Python à l'adresse <https://www.python.org/downloads/>.
2. Téléchargez la dernière version stable de Python pour votre système d'exploitation (Windows, macOS, Linux).
3. Exécutez le fichier d'installation téléchargé.
4. Cochez l'option "Add Python to PATH" (Ajouter Python au PATH) pour faciliter l'accès à Python depuis l'invite de commande.
5. Choisissez "Install Now" (Installer maintenant) pour démarrer l'installation.
6. Une fois l'installation terminée, ouvrez une nouvelle fenêtre d'invite de commande et tapez "python --version" pour vérifier que Python a été installé correctement.

b. Installation de Visual Studio :

1. Rendez-vous sur le site officiel de Visual Studio à l'adresse <https://visualstudio.microsoft.com/fr/downloads/>.
2. Téléchargez la version de Visual Studio adaptée à vos besoins. Nous recommandons Visual Studio Code pour son interface légère et ses fonctionnalités adaptées au développement Python.
3. Exécutez le fichier d'installation téléchargé.
4. Suivez les instructions d'installation en choisissant les composants que vous souhaitez installer. Assurez-vous de sélectionner les extensions Python nécessaires pour faciliter le développement.

Une fois que nous avons installé Python et Visual Studio, nous mis en place l'environnement de développement nécessaire pour créer votre application conviviale de détection des maladies des plantes. Dans les sections suivantes, nous allons approfondir l'installation des bibliothèques Python nécessaires et la préparation de nos données pour la transformatio



Figure.4.1.logo de python et visual studio

III. Installation des Bibliothèques Nécessaires

Maintenant que nous avons mis en place l'environnement de développement, nous devons nous assurer que les bibliothèques Python requises sont correctement installées. Ces bibliothèques

joueront un rôle crucial dans la création de notre application conviviale de détection des maladies des plantes. Voici une présentation des principales bibliothèques utilisées et les instructions pour les installer à l'aide de l'outil de gestion de paquets pip.

Maintenant que nous avons mis en place l'environnement de développement, nous devons nous assurer que les bibliothèques Python requises sont correctement installées. Ces bibliothèques joueront un rôle crucial dans la création de notre application conviviale de détection des maladies des plantes. Voici une présentation des principales bibliothèques utilisées et les instructions pour les installer à l'aide de l'outil de gestion de paquets pip.

III.1 Bibliothèques Python Utilisées :

- **Tkinter** : Tkinter est la bibliothèque standard pour la création d'interfaces utilisateur graphiques (GUI) en Python. Elle sera utilisée pour concevoir la fenêtre de l'application et pour intégrer des éléments interactifs tels que des boutons et des étiquettes.
- **PIL (Python Imaging Library)** : PIL est utilisée pour le traitement des images en Python. Nous l'utiliserons pour charger, redimensionner et afficher les images capturées par les utilisateurs.
- **numpy** : numpy est une bibliothèque essentielle pour les calculs numériques en Python. Nous l'utiliserons pour effectuer des opérations mathématiques sur les images et les données du modèle.
- **tensorflow** : tensorflow est la bibliothèque d'apprentissage automatique qui nous permettra d'intégrer notre modèle de détection des maladies dans l'application.

III.2 Installation des Bibliothèques :

1. nous avons ouvrir notre invite de commande (ou terminal)

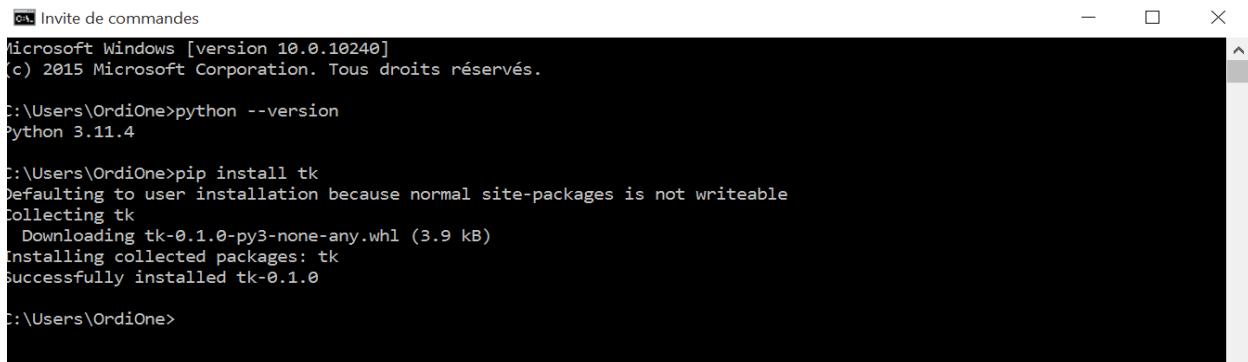


```
Invite de commandes
Microsoft Windows [version 10.0.10240]
(c) 2015 Microsoft Corporation. Tous droits réservés.

/Users/OrdiOne>
```

Figure.4.2. invite de commande (ou terminal)

2. Pour installer une bibliothèque, nous avons utilisé la commande "pip install" suivie du nom de la bibliothèque. Par exemple, pour installer Tkinter, nous avons tapé :



```
Invite de commandes
Microsoft Windows [version 10.0.10240]
(c) 2015 Microsoft Corporation. Tous droits réservés.

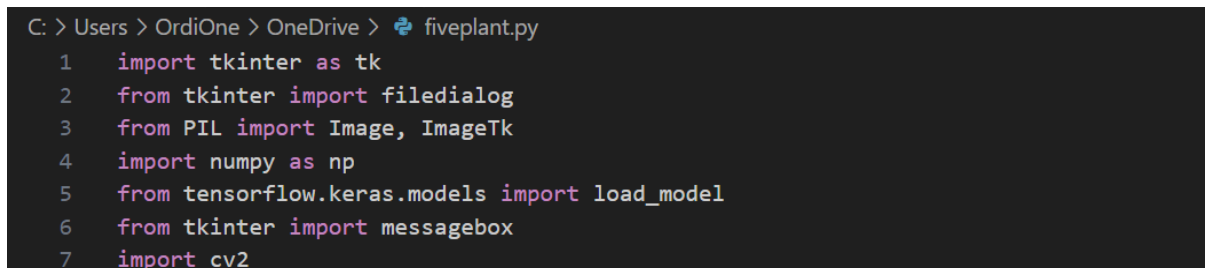
C:\Users\OrdiOne>python --version
Python 3.11.4

C:\Users\OrdiOne>pip install tk
Defaulting to user installation because normal site-packages is not writeable
Collecting tk
  Downloading tk-0.1.0-py3-none-any.whl (3.9 kB)
Installing collected packages: tk
Successfully installed tk-0.1.0

C:\Users\OrdiOne>
```

Figure 4.3. installation des bibliothèques sur l'invite de commande

Une fois l'installation terminée, Nous avons vérifié si les bibliothèques sont correctement installées en lançant Python en mode interactif et en important chaque bibliothèque pour vérifier l'absence d'erreurs



```
C: > Users > OrdiOne > OneDrive > fiveplant.py
1 import tkinter as tk
2 from tkinter import filedialog
3 from PIL import Image, ImageTk
4 import numpy as np
5 from tensorflow.keras.models import load_model
6 from tkinter import messagebox
7 import cv2
```

Figure 4.4. Vérification des bibliothèques sur le mode interactif du python

En installant ces bibliothèques, nous avons assuré d'avoir toutes les ressources nécessaires pour développer notre application conviviale.

IV. Développement de l'Interface Utilisateur

Dans cette section, nous allons plonger dans le développement de l'interface utilisateur pour notre application de détection des maladies des plantes. L'interface utilisateur sera la passerelle par laquelle les utilisateurs interagiront avec l'application, téléchargeront des images de plantes et recevront les résultats de la détection des maladies.

IV.1 Introduction à l'Interface Utilisateur

L'interface utilisateur de notre application joue un rôle crucial pour offrir aux utilisateurs une expérience conviviale et intuitive. Elle devrait permettre aux utilisateurs de télécharger facilement des images, de visualiser les résultats de la prédiction et d'interagir avec les fonctionnalités de l'application. Pour ce faire, nous allons utiliser la bibliothèque Tkinter, qui nous permettra de créer une interface graphique interactive.

IV.2 Utilisation de Tkinter

Tkinter est la bibliothèque standard de Python pour créer des interfaces utilisateur graphiques. Elle offre une variété d'éléments tels que des fenêtres, des boutons, des étiquettes, des champs de texte et plus encore, que nous pouvons utiliser pour construire l'interface de notre application. Nous allons créer des boutons pour différentes plantes, un bouton de téléchargement d'image, une zone pour afficher l'image téléchargée et un bouton de prédiction.

IV.3 Intégration de la Caméra

Pour permettre aux utilisateurs de capturer des images directement depuis leur caméra d'ordinateur, nous allons intégrer la fonctionnalité de capture d'image dans notre application. Cela permettra aux utilisateurs de prendre une photo de la plante qu'ils veulent analyser. Nous allons utiliser la bibliothèque OpenCV pour accéder à la caméra et capturer des images en temps réel. En intégrant cette fonctionnalité, nous offrirons une option pratique pour télécharger des images sans avoir à rechercher des fichiers sur l'ordinateur.

Dans la section suivante, nous aborderons l'intégration du modèle de détection des maladies dans l'application et expliquerons comment utiliser les résultats de la prédiction pour fournir des informations utiles aux utilisateurs.

```
# Fonction pour capturer une photo depuis la caméra
def capturer_photo():
    camera = cv2.VideoCapture(0) # Ouvre la caméra
    ret, frame = camera.read() # Capture une image
    if ret:
        cv2.imwrite("photo_capturee.jpg", frame) # Enregistre l'image capturée
        camera.release() # Ferme la caméra
        return "photo_capturee.jpg"
    else:
        camera.release()
        return None
```

Figure 4.5.intégration la fonctionnalité de capture d'image dans notre application

V. Chargement et Utilisation des Modèles de Détection des Maladies

Dans cette section, nous allons explorer la manière dont les modèles de détection des maladies, que nous avons préalablement entraînés à l'aide de TensorFlow, seront intégrés dans notre application. Nous allons découvrir comment utiliser ces modèles pour prédire les maladies à partir des images téléchargées ou capturées par les utilisateurs.

V.3 Chargement des Modèles

Le processus de chargement des modèles de détection des maladies est essentiel pour que notre application puisse effectuer des prédictions précises. Nous expliquerons comment charger les modèles TensorFlow préalablement entraînés en utilisant les chemins de fichiers appropriés. Chaque modèle spécifique à une plante (fraise, raisin, pomme, etc.) doit être chargé dans l'application pour effectuer les prédictions correspondantes.

```
# Chargement des modèles
chemin_du_modele_fraise = "c:\\Users\\OrdiOne\\OneDrive\\model_cnn_fraise.h5"
chemin_du_modele_raisin = "c:\\Users\\OrdiOne\\OneDrive\\model_cnn_raisin1.h5"
chemin_du_modele_pommes = "c:\\Users\\OrdiOne\\OneDrive\\model_cnn_apple1.h5"
chemin_du_modele_peches = "c:\\Users\\OrdiOne\\OneDrive\\model_cnn_peach.h5"
chemin_du_modele_mais = "c:\\Users\\OrdiOne\\OneDrive\\model_cnn_mais1.h5"
chemin_du_modele tomates = "c:\\Users\\OrdiOne\\OneDrive\\model_cnn_tomato_disease.h5"
chemin_du_modele_pommes_de_terre = "c:\\Users\\OrdiOne\\OneDrive\\model_cnn_potato_disease.h5"
```

Figure 4.6. Chargement des modèles

V.1 Prédiction des Maladies

Une fois que les modèles sont chargés, nous aborderons le processus de prédiction des maladies. Lorsqu'un utilisateur télécharge ou capture une image de plante, nous allons utiliser les modèles chargés pour effectuer des prédictions sur cette image. Nous expliquerons comment prétraiter l'image téléchargée de manière à la rendre compatible avec les entrées attendues par les modèles. Ensuite, nous montrerons comment utiliser les modèles pour effectuer des prédictions et obtenir les probabilités associées à chaque classe de maladie.

Nous expliquerons également comment mapper les résultats de la prédiction aux libellés des maladies pour fournir une information compréhensible aux utilisateurs. Par exemple, si le modèle prédit qu'une plante est atteinte de "Black Rot", nous montrerons comment afficher ce résultat de manière conviviale dans l'interface utilisateur.

Dans la section suivante, nous aborderons la manière dont nous pouvons rendre l'application encore plus conviviale en intégrant des fonctionnalités telles qu'un logo personnalisé et des options de choix de plante.

VI. Personnalisation de l'Interface Utilisateur

Dans cette section, nous allons explorer comment personnaliser davantage l'interface utilisateur de notre application de détection des maladies des plantes. Nous allons intégrer des fonctionnalités telles qu'un logo personnalisé et des options de choix de plante, ce qui rendra l'expérience utilisateur plus agréable et intuitive.

VI.1 Intégration d'un Logo Personnalisé

Pour ajouter une touche visuelle unique à notre application, nous allons intégrer un logo personnalisé. Un logo attrayant peut aider à identifier rapidement notre application et lui donner une apparence professionnelle. Nous expliquerons comment charger et afficher le logo à l'aide de la bibliothèque Pillow (PIL) pour la manipulation d'images.



Figure4.7. logo de notre application

VI.1 Options de Choix de Plante

Afin de rendre notre application polyvalente pour différents types de plantes, nous allons intégrer des boutons d'options de choix de plante. Les utilisateurs pourront sélectionner le type de plante (fraise, raisin, pomme, etc.) pour lequel ils souhaitent effectuer la détection des maladies. Nous

expliquerons comment créer ces boutons d'options et comment mémoriser la sélection de l'utilisateur.

Ensuite, nous montrerons comment lier ces options de choix de plante aux modèles appropriés que nous avons chargés précédemment. Par exemple, si l'utilisateur sélectionne "raisin", l'application utilisera le modèle de détection des maladies spécifique au raisin pour effectuer les prédictions.

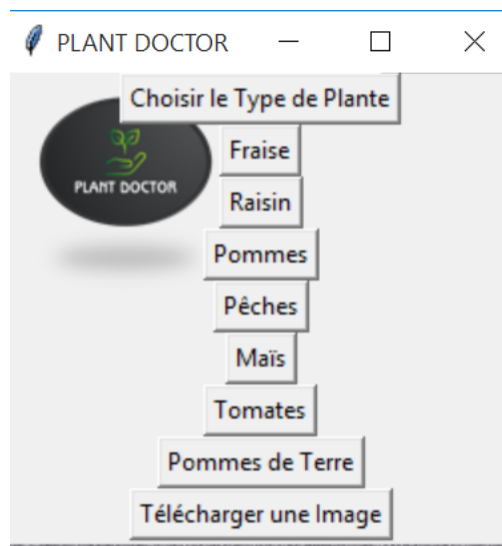


Figure 4.8.Options de Choix de Plante

VI.3 Amélioration de l'Expérience Utilisateur

En intégrant ces fonctionnalités de personnalisation, nous améliorons l'expérience utilisateur globale de l'application. Les utilisateurs auront plus de contrôle sur les types de plantes qu'ils souhaitent analyser et ils seront accueillis par un logo attrayant. Ces ajouts visent à rendre l'application plus conviviale et à encourager davantage d'utilisateurs à l'utiliser.

Dans la dernière section, nous aborderons les étapes nécessaires pour déployer notre application, que ce soit en tant qu'application de bureau ou en ligne, afin qu'elle soit accessible à un large public.

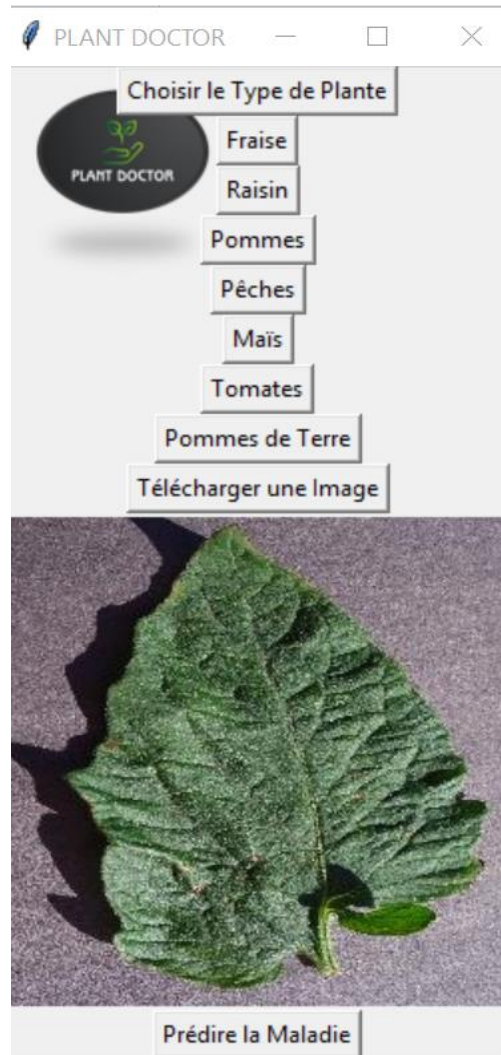


Figure 4.9.application plant doctor

VII. Affichage des Résultats et Améliorations Possibles

Dans cette section, nous allons aborder la manière dont nous pouvons afficher les résultats de prédiction dans l'interface utilisateur de notre application.

VII.1 Affichage des Résultats de Prédiction

Une fois que les modèles ont effectué leurs prédictions, nous allons montrer comment afficher ces résultats de manière claire et compréhensible dans l'interface utilisateur. Nous expliquerons comment traduire les résultats numériques des modèles en libellés de maladies compréhensibles par les utilisateurs. Par exemple, si le modèle prédit que la plante est atteinte de "Black Rot", nous

montrons comment afficher ce résultat dans une fenêtre pop-up ou une zone dédiée de l'interface utilisateur.

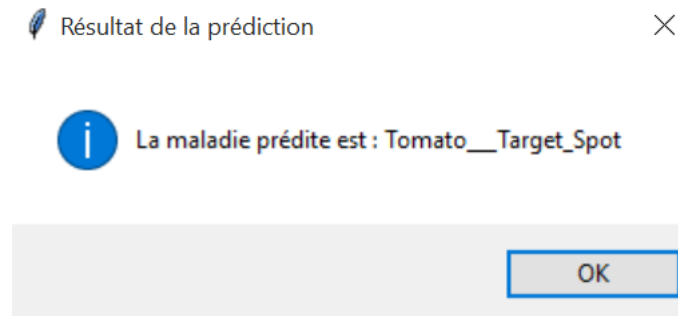


Figure 4.10. affichage des résultats de prédiction

VIII. Conclusion :

Dans ce chapitre, nous récapitulerons les principales étapes que nous avons suivies pour créer notre application de détection des maladies des plantes. Nous soulignerons l'importance de rendre les modèles accessibles aux utilisateurs grâce à une interface conviviale et nous encouragerons les lecteurs à poursuivre leur exploration dans le domaine de la détection des maladies des plantes et du développement d'applications innovantes.

En combinant nos connaissances en apprentissage automatique avec les compétences en développement d'applications, nous avons réussi à créer une solution qui peut avoir un impact positif sur l'agriculture et l'environnement.

Références bibliographiques

I. Bibliographie :

- [1] **Chloé-Agathe Azencott** : Introduction au Machine Learning
- [2] Introduction à l'apprentissage automatique Ludovic DE MATTEIS Steeven JANNY - Solal NATHAN – Wenqi SHU-QUARTIER
- [3] Apprentissage semi-supervisé par réduction de dimensionnalité non linéaire par **François Payette**
- [4] Introduction à l'apprentissage par renforcement **Anthony JUTON – Valentin NOËL – Rida LALI**
- [5] Review of Deep Learning Algorithms and Architectures par Ajay **Shrestha, Ausif Mahmood**
- [6] Deep Learning as a Frontier of Machine Learning : A Review Article in **International Journal of Computer Applications · July 2018**
 - [7] **LOUAM ABDELHAK BILAL** : Deep Learning basé sur les méthodes de réduction pour la reconnaissance de visage
 - [8] **Dr Benlaksira** : Anatomie des organes végétaux / Feuille Dr Benlaksira
 - [9] George N. Agrios, 2005 (la cinquième édition). Plant Pathology. Academic Press.

I. Webographie

- [W1] <https://www.nature.com/articles/s41524-020-0276-y>
- [W2] https://folia.unifr.ch/documents/315019/files/TB_Charbonnier_version_finale.pdf
- [W3] https://www.researchgate.net/figure/Apprentissage-semi-supervise-les-cinq-exemples-encercles-sont-les-seuls_fig2_238103615
- [W4] <https://datapeaker.com/fr/Big-Data/rnn-%C3%A0-partir-de-z%C3%A9ro-construire-le-mod%C3%A8le-rnn-en-python/>
- [W5] <https://www.aquaportail.com/dictionnaire/definition/3783/feuille-simple>
- [W6] <https://www.aquaportail.com/dictionnaire/definition/3775/feuille>
- [W7] <https://www.cisin.com/coffee-break/fr/enterprise/highlights-the-advantages-and-disadvantages-of-machine-learning.html>

-
- [W8] <https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/photosynthesis>
 - [W9] <https://github.com/tensorflow/tensorflow>
 - [W10] [Tutoriel : Introduction à Keras \(dataquest.io\)](https://dataquest.io/tutorials/python/introduction-to-keras/)
 - [W11] <https://perso.esiee.fr/~buzerl/IA/60%20reseauconv/reseauconv.html>
 - [W12] https://www.researchgate.net/figure/MaxPooling-operation-on-a-4-4-image-with-2-2-window-size-where-stride2-and_fig2_344197871
 - [W13] <https://aashay96.medium.com/experimentation-with-variational-dropout-do-subnetworks-exist-inside-a-neural-network-e482cbbea7dd>
 - [W14] <https://data-flair.training/blogs/keras-convolution-neural-network/>
 - [W15] <https://data-flair.training/blogs/keras-convolution-neural-network/>
 - [W16] <https://www.aquaportail.com/dictionnaire/definition/3775/feuille>
 - [W17] <https://www.biologie101.fr/biologie-vegetale/appareil-vegetatif-plante/systeme-caulinaire-feuille>

Annexe

Plant Doctor business model



الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة عين تموشنت بلحاج بوشعيب
حاضنة الأعمال عين تموشنت



ملحق نموذج العمل التجاري

البطاقة التقنية للمشروع **Fiche technique du projet**

مسيردي ياسين رميني فاروق	الاسم و اللقب Votreprénom et nom Your first and last Name
PLAT DOCTOR	الاسم التجاري للمشروع Intitulé de votre projet Title of your Project
0792728336-0797045656	رقم الهاتف Votrenuméro de téléphone Your phone number
mecirdiyassine46@gmail.com faroukfarouk599@gmail.com	البريد الالكتروني Votreadresse e-mail Your email address
ولاية عين تيموشنت	مقر مزاولة النشاط (الولاية- البلدية) Votrevilleou commune d'activité Your city or municipality of activity

طبيعة المشروع **Nature de projet**

المنتج ذو خدماتي (تطبيق)

The product has a service nature (app)



المشكلة المراد حلها وتكون مدعمة بالبيانات (إحصائيات إن وجدت)

- على الصعيد العالمي ، تشير التقديرات إلى أن حوالي 20-40 ٪ من غلة المحاصيل تُفقد كل عام بسبب الآفات والأمراض النباتية والحشرات الضارة.
- وفي البلدان النامية ، يمكن أن تكون خسائر المحاصيل أعلى ، مع تقديرات تتراوح من 30 إلى 50 ٪.
- بالإضافة إلى الخسائر المتعلقة بأمراض وعدوى ما قبل الحصاد ، هناك أيضًا خسائر ما بعد الحصاد بسبب الأمراض الفطرية والآفات وعوامل أخرى. وبحسب منظمة الفاو ، يمكن أن تصل هذه الخسائر إلى 20-30 ٪ من الإنتاج الزراعي في العالم



1- القيمة المقترحة:

- التعرف على أمراض النبات
- اقتراحات العلاجات
- سهولة الوصول إلى منتجات الأدوية النباتية
- توليد الإيرادات من خلال إعلانات الأدوية و المبيدات

:

- ما طبيعة هذا الحل للمشكلة هل هي قيم نوعية أو كمية؟ :
- بالحد من مخاطر الأمراض التي تصيب النباتات سواء كانت طفيلية أو فطرية.

- جعل التطبيق سهل الاستعمال.
- قيمة مبتكرة و فكرة جديدة غير متوفرة بالجزائر حاليا.

1/1- القيمة التي نقدمها للعميل:

إليك هذا الجدول الذي يوضح النباتات التي اخترناها، إضافة إلى الأمراض التي يمكن للنظام اكتشافها:

الأمراض % نسبة الإصابة

Classes	Maladies (% pourcentage d'infection)						
Raisin	Black Rot (10 à 30%)	Rougeot parasitaire (10 à 50 %)	Septoriose (10 à 30%)				
Pomme de terre	Alternariose (10 à 50 %)	Mildiou (20 à 90 %)					
Tomate	Mildiou (20 à 80 %)	Septoriose	Araignée	Bacterioses	Cladosporiose	Oidium (10 à 50 %)	Alternaria solani
Pomme	Oidium (10 à 50 %)	Black rot (10 à 30%)	rouille du pommier	Tavelure (50 à 90 %)			
Haricot	Rouille d'haricot	Tache angulaire					
Fraise	Tache angulaire						
Olives	Oeil de paon	Aculus olearius					
Maïs	Cercosporiose	Rouille commune (10 à 50 %)	Septoriose				
Pêche	Bacteriose						

2/1- ما هي المشاريع الأخرى التي استهدفت نفس المشكلة والتي جرى تنفيذها؟

في الوقت الحالي ليس هناك أي مشروع أو تطبيق بنفس الفكرة أو حتى بنفس الخصائص في الجزائر.

2- شرائح العملاء :

- المزارعون
- البستانيون عامة والمحترفون خاصة
- هواة الزراعة والبيستنة

3- العلاقات مع العملاء:



- تكون الخدمة الذاتية من خلال التطبيق
- التحديثات والتحسينات
- تحسينات بناء على تعليقات المستخدمين

4- القنوات:

1/4- الآليات والطرق لإعلام بمنتوجنا أو خدمتنا:

- متاجر تطبيقات الأجهزة المحمولة (iOS، Android) وأجهزة الكمبيوتر أيضاً
- موقع إلكتروني
- تسويق عبر وسائل التواصل الاجتماعية (... youtube , facebook , instagram)

2/4- قنوات التوزيع التي يفضلها العملاء:

في وقت الحالي يقضي الناس الكثير من وقت في مواقع تواصل الاجتماعي و هذه فرصة جيدة للتعريف بتطبيقنا



5- **Keypartners:** الشركات الرئيسية:

- من هم الشركاء الرئيسيون الذين يمكن مساعدتنا في الانتاج أو الخدمة أو في تسويقها أو توزيعها؟ (الشركاء الذين أضع معهم عقد).
- من هم الموردون الرئيسيين؟ (الذين يقدمون لنا: المواد الأولية + الآلات للإنتاج + برنامج لتقديم خدمة +...)

1/5- الشركاء الرئيسيون الذين يمكن مساعدتنا:

- معاهد البحوث الزراعية(المعهد الفلاحي عين تموشنت)
- موردي و مصنعي الأدوية الخاصة بالنباتات و الشجار :

ACI ALGERIE



MEDMAC ALGERIE



6- **Keyactivities:** الأنشطة الرئيسية:

الأنشطة الرئيسية:

- تطوير وصيانة التطبيق
- بناء وتحديث قاعدة بيانات أمراض النبات
- إقامة شراكات مع مصنعي للأدوية الزراعية



Key Resources

1- الموارد الرئيسية:

نقوم بتحديد فقط الموارد دون ذكر التكلفة.

1/7- الموارد المادية:

- قاعدة بيانات الأمراض النباتية
- خوارزميات التعرف على الصور
- اتفاقيات الشراكة مع الشركات المصنعة للأدوية
- فريق التطوير والخبرة

2/7- الموارد البشرية:

صنف المورد البشري	العدد
-------------------	-------

1	Web designer
1	Ingénieur en informatique
1	Ingénieur en agronomie
1	Délègue commercial



8- Cost structure



2- هيكل التكاليف:

■ 1/8: هيكل التكاليف structure Costs

50000DA-100000DA	تكاليف التعريف بالمنتج أو المؤسسة Frais d'établissement
-	تكاليف الحصول على العدادات (الماء- الكهرباء) Frais d'ouverture de compteurs (eaux-gaz-....)
-	تكاليف (التكوين- برامج الاعلام الالي المختصة) Logiciels, formations
-	Dépôt marque, brevet, modèle تكاليف براءة الاختراع و الحماية الصناعية و التجارية
-	Droits d'entrée تكاليف الحصول على تكنولوجيا او ترخيص استعمالها
-	Achat fonds de commerce ou parts شراء الأصول التجارية أو الأسهم
-	Droit au bail الحق في الإيجار
-	Caution ou dépôt de garantie وديعة أو وديعة تأمين
-	Frais de dossier رسوم إيداع الملفات
-	Frais de notaire ou d'avocat تكاليف الموثق-المحامي-.....
-	Enseigne et éléments de communication تكاليف التعريف بالعلامة و تكاليف قنوات الاتصال
-	Achat immobilier شراء العقارات
-	Travaux et aménagements الأعمال والتحسينات الاماكن
200000DA	Matériel الألات- المركبات- الاجهزة
-	Matériel de bureau تجهيزات المكتب
-	Stock de matières et produits تكاليف التخزين
-	trésorerie de départ التدفق النقدي (الصندوق) الذي تحتاجه في بداية المشروع.
-	

المجموع = 250000DA





3- مصادر الإيرادات : Revenue Streams9-

- إعلانات الشركات المصنعة للأدوية العشبية في التطبيق
- ميزات مميزة أو خيارات الاشتراك
- شراكات لتقاسم الأرباح