

People's Democratic Republic of Algeria  
Ministry of Higher Education and Scientific Research  
University of 8 Mai 1945 Guelma



Faculty of Mathematics, Computer Science and Material Sciences  
Department of Computer Science  
Domiciliation laboratory of Information and Communication Sciences and Technologies

## Thesis

Submitted in Candidacy for the Degree of *Doctorate in Third Cycle*

Field: Computer Science Stream: Mathematics and Computer Science  
Speciality: Informatics Systems

Presented by:  
**Imane BOUACIDA**

### *Title*

**Smart Farming Solutions: Automated Crop and Plantation Disease  
Detection**

Defended on: 16/10/2024

Before the jury composed of:

Full name	Rank	University	
Mr Zineddine KOUAHLA	Professor	Univ. of 8 Mai 1945, Guelma	President
Mr Brahim FAROU	Professor	Univ. of 8 Mai 1945, Guelma	Supervisor
Mr Hamid SERIDI	Professor	Univ. of 8 Mai 1945, Guelma	Co-supervisor
Mr Abdelkrim BOURAMOUL	Professor	Univ. of Abdelhamid Mehri, Constantine	Examiner
Mr Khaled HALIMI	MCA	Univ. of 8 Mai 1945, Guelma	Examiner
Mr Ali KHEBIZI	MCA	Univ. of 8 Mai 1945, Guelma	Invited

Academic year: 2023/2024

---

# ACKNOWLEDGEMENTS

First and foremost, I express my deepest gratitude to God for granting me the strength, knowledge, ability, and opportunity to embark on this research journey, persevere through its challenges, and successfully complete it.

I extend my sincere gratitude to my supervisor, Professor Brahim FAROU. Words cannot fully convey the depth of my appreciation for your invaluable guidance throughout this journey. Your mentorship has been instrumental in shaping my academic endeavors and make my first steps in the realm of scientific research. I am deeply thankful for the opportunity to work under your tutelage and for the knowledge and insights you have imparted to me. Your support and assistance have been invaluable in reaching this milestone. Thank you for everything.

I am deeply indebted to my co-supervisor, Professor Hamid SERIDI, whose consistent support, encouragement, and guidance have been invaluable to me over the years. I extend my heartfelt gratitude to him for his numerous insightful remarks and appreciations, which have been of utmost relevance.

I am deeply grateful to Professor Zineddine KOUAHLA, the head of the computer science department at Guelma University, for generously sharing his vast experience and consistently encouraging my progress. I extend my infinite thanks to him and express my deepest gratitude.

I extend my heartfelt gratitude to Professor Zineddine KOUAHLA, Doctor Khaled HALIMI, and Doctor Ali KHEBIZI from the University of Guelma, and Professor Abdelkrim BOURAMOUL from the University of Constantine for graciously accepting the responsibility of examining this work and participating in the defense jury.

I am grateful to the computer science department teachers of Guelma University for their invaluable assistance. My sincere appreciation goes to the LabSTIC laboratory researchers for their collaboration, which have provided a stimulating academic environment. I deeply appreciate the support provided by laboratory engineer Miss Madiha KHAROUBI.

My sincere gratitude goes to my parents, family, and friends for their steadfast support, patience, and understanding during this journey. Their unwavering encouragement has been a continual source of strength and motivation for me.

## ملخص

العمل المقدم في أطروحة الدكتوراه هذه يركز على تطوير أنظمة الكشف والتعرف على الأمراض التي تؤثر على المحاصيل الزراعية والمزارع، استناداً إلى التعلم العميق والتعلم الآلي. تقدم أنظمة الكشف والتعرف المقترحة في الأدبيات حلولاً دقيقة للتعرف المبكر والإدارة الفعالة. ومع ذلك، فإنها لا تزال تواجه عدة تحديات وتظل محدودة في استخداماتها. التحدي الأول هو نقص قوتها وتعميمها. يقتصر استخدامها على أنواع المحاصيل والأمراض التي واجهتها أثناء عملية التعلم، مما يشير لمشاكل عندما يتم مواجهة أنواع جديدة من المحاصيل والأمراض غير الموجودة في مرحلة التدريب. التحدي الثاني هو أن معظم هذه الأنظمة مصممة لاكتشاف مرض واحد في كل مرة ولا تعالج مشكلة الكشف المتعدد عن الأمراض. التحدي الآخر هو أن معظم الأنظمة المقترحة تكتشف الأمراض من الأوراق، وهو أمر شائع لأن الأوراق غالباً ما تكون المكان الأول الذي تظهر فيه الأمراض في النباتات. ومع ذلك، فإن بعض الأمراض تصيب فروع الأشجار ولا تظهر على الأوراق. للتغلب على هذه التحديات، قدمنا ثلاثة أنظمة للتعرف على أمراض النبات. النظام الأول مبني على التعلم العميق، قادر على التمييز بين الأوراق السليمة والمصابة بالأمراض، بغض النظر عن نوع المحصول والمرض، حتى إذا لم يتم مواجهتهما خلال مرحلة التدريب. الفكرة الرئيسية هي إعطاء الأولوية لتحديد المناطق الصغيرة في الأوراق التي تعاني من الأمراض بدلاً من الاعتماد فقط على المظهر العام للورقة المصابة. بالإضافة إلى ذلك، تشمل تقييم معدل انتشار المرض عبر الورقة بأكملها. لضمان تصنيف فعال، نستخدم تصميم نموذج Small Inception، الذي يتمتع بقدرة على معالجة المناطق الصغيرة دون التأثير على الأداء. النظام المقترح الثاني هو نموذج مبني على التعلم العميق مصمم لاكتشاف والتعرف على عدة أمراض في نفس الوقت من أي نوع من المحاصيل، بما في ذلك تلك التي لم تواجه خلال عملية التدريب. تتيح طريقتنا التعرف المستقل على أعراض كل مرض في المناطق الصغيرة للأوراق، بغض النظر عن وجود أمراض أخرى على نفس الورقة وبغض النظر عن نوع المحصول. يتم ذلك من خلال طريقة العزل، التي تعزل كل منطقة تحتوي على أعراض مرض محددة وتقضي على تأثير خصائص المحصول، بالتزامن مع تصميم نموذج Small Inception. بالإضافة إلى ذلك، تتيح طريقتنا حساب معدل انتشار كل مرض على الورقة وتحديد النطاق الكلي لجميع الأمراض الموجودة على الورقة. النظام الثالث هو نهج مبني على التعلم الآلي مصمم لاكتشاف وتجزئة الأمراض الموجودة على الورقة. الأشجار، مستهدفاً بشكل خاص مرض *Nectria cinnabarina* على فروع أشجار التفاح. يستخدم هذا النظام تقنيات معالجة الصور لتحسين جودة الصورة ودقتها، مما يسهل مهمة مصنف نموذج الخليط الغوسي. يتعلم هذا المصنف التوزيع الاحتمالي للون المرض ويشيئ قناع التجزئة، الذي يتم استخدامه لتحديد المناطق المصابة على الفرع. تظهر النتائج المتحصل عليها من التجارب

على الأنظمة الثلاثة فعالية الطرق المقترحة في تحسين دقة اكتشاف أمراض النباتات. علاوة على ذلك، تفوقت على الأساليب الحالية من خلال تحديد الأمراض بنجاح عبر مختلف أنواع المحاصيل، واكتشاف عدة أمراض في نفس الوقت من نفس الورقة، وتحديد وتجزئة الأمراض بدقة من الفروع.

**الكلمات المفتاحية:** الزراعة، أمراض النبات، اكتشاف الأمراض، الذكاء الاصطناعي، التعلم العميق، التعلم الآلي.



---

# RÉSUMÉ

Le travail présenté dans cette thèse de doctorat se concentre sur le développement de systèmes de détection et de reconnaissance des maladies qui affectent les cultures et les plantations, basés sur l'apprentissage profond et l'apprentissage automatique. Les systèmes de détection et de reconnaissance proposés dans la littérature offrent des solutions précises pour l'identification précoce et la gestion efficace. Cependant, ils rencontrent encore plusieurs défis et sont limités dans leurs utilisations. Le premier défi est leur manque de robustesse et de généralisation. Leur utilisation est limitée aux types de cultures et de maladies rencontrés lors du processus d'apprentissage, ce qui pose des problèmes lorsqu'ils sont confrontés à de nouveaux types de cultures et de maladies non rencontrés lors de la phase d'entraînement. Le deuxième défi est que la plupart de ces systèmes sont conçus pour détecter une seule maladie à la fois et ne traitent pas le problème de la détection simultanée de plusieurs maladies. Un autre défi est que la plupart des systèmes proposés détectent les maladies à partir des feuilles, ce qui est courant car les feuilles sont souvent le premier endroit où les maladies apparaissent dans les plantes. Cependant, certaines maladies infectent les branches des arbres et n'apparaissent pas sur les feuilles. Pour relever ces défis, nous avons introduit trois systèmes de reconnaissance des maladies des plantes. Le premier système est basé sur l'apprentissage profond, capable de distinguer entre les feuilles saines et malades, quel que soit le type de culture et de maladie, même si le système ne les a pas rencontrés lors de la phase d'entraînement. L'idée principale est de donner la priorité à l'identification des petites régions foliaires malades plutôt que de se fier uniquement à l'apparence globale de la feuille malade. De plus, cela inclut l'évaluation du taux de prévalence de la maladie sur toute la feuille. Pour assurer une classification efficace, nous utilisons une architecture de modèle Small Inception, qui est capable de traiter de petites régions sans compromettre les performances. Le deuxième système proposé est un modèle basé sur l'apprentissage profond conçu pour détecter et reconnaître plusieurs maladies simultanément à partir de n'importe quel type de culture, y compris ceux qui n'ont pas été rencontrés lors du processus de formation. Notre méthode permet la reconnaissance indépendante des symptômes de chaque maladie dans les petites régions foliaires, indépendamment de la présence d'autres maladies sur la même feuille et indépendamment du type de culture. Cela est accompli grâce à la méthode d'isolement, qui isole chaque région contenant des symptômes spécifiques de maladie et élimine l'influence des caractéristiques de la

culture, en conjonction avec l'architecture du modèle Small Inception. De plus, notre approche permet le calcul du taux de prévalence de chaque maladie sur la feuille et la détermination de l'étendue globale de toutes les maladies présentes sur la feuille. Le troisième système est une approche basée sur l'apprentissage automatique conçue pour détecter et segmenter les maladies des branches d'arbres, ciblant spécifiquement la maladie *Nectria cinnabarina* sur les branches de pommiers. Ce système utilise des techniques de traitement d'image pour améliorer la qualité et la précision de l'image, facilitant ainsi la tâche du classificateur de modèle de mélange gaussien. Ce classificateur apprend la distribution de probabilité de la couleur de la maladie et génère un masque de segmentation, qui est ensuite utilisé pour identifier les zones malades sur la branche. Les résultats obtenus à partir des expériences sur les trois systèmes démontrent l'efficacité des méthodes proposées pour améliorer la précision de la détection des maladies des plantes. De plus, ils ont surpassé les méthodes existantes en identifiant avec succès les maladies à travers différents types de cultures, en détectant plusieurs maladies simultanément à partir de la même feuille, et en identifiant et en segmentant avec précision les maladies sur les branches.

**Mots-clés:** Agriculture, Maladies des Plantes, Détection de Maladies, Intelligence Artificielle, Apprentissage Profond, Apprentissage Automatique.

---

# ABSTRACT

The work presented in this Ph.D. thesis focuses on developing detection and recognition systems for diseases that affect agricultural crops and plantations based on deep learning and machine learning. The proposed detection and recognition systems in the literature offer precise solutions for early identification and effective management. However, they still face several challenges and they are still limited in their uses. The first challenge is their lack of robustness and generalization. Their use is limited to the types of crops and diseases encountered during the learning process, leading to problems when faced with new types of crops and diseases not seen in the training phase. The second challenge is that the majority of these systems are designed to detect only one disease at a time and do not address the problem of simultaneous multi-disease detection. Another challenge is that most proposed systems detect diseases from leaves, which is common because leaves are often the first place where diseases appear in plants. However, some diseases infect the tree branches and do not appear on the leaves. To tackle these challenges, we have introduced three plant disease recognition systems. The first system is based on deep learning, capable of distinguishing between healthy and diseased leaves regardless of the crop type and disease, even if the system hasn't encountered them during the training phase. The primary idea is to prioritize the identification of diseased small leaf regions rather than solely relying on the overall appearance of the diseased leaf. Moreover, it includes assessing the disease's prevalence rate across the entire leaf. To ensure efficient classification, we employ a Small Inception model architecture, which is adept at processing small regions without compromising performance. The second proposed system is a deep learning-based model designed to detect and recognize multiple diseases simultaneously from any crop type, including those not encountered during the training process. Our method enables the independent recognition of each disease's symptoms within small leaf regions, regardless of the presence of other diseases on the same leaf and irrespective of the crop type. This is accomplished through the isolation method, which isolates each region containing specific disease symptoms and eliminates the influence of crop characteristics, in conjunction with the Small Inception model architecture. Additionally, our approach enables the calculation of the prevalence rate of each disease on the leaf and the determination of the overall extent of all diseases present on the leaf. The third system is a machine learning-based approach designed for detecting and segmenting diseases from tree branches, specifically targeting *Nectria*

cinnabarina disease on apple tree branches. This system utilizes image processing techniques to enhance image quality and precision, thus facilitating the task for the Gaussian Mixture Model classifier. This classifier learns the probability distribution of the disease color and generates a segmentation mask, which is then utilized to identify the diseased areas on the branch. The results obtained from the experiments on the three systems demonstrate the effectiveness of the proposed methods in enhancing the accuracy of plant disease detection. Moreover, they outperformed existing methods by successfully identifying diseases across various crop types, detecting multiple diseases simultaneously from the same leaf, and accurately identifying and segmenting diseases from branches.

**Key-words:** Agriculture, Plant Diseases, Disease Detection, Artificial Intelligence, Deep learning, Machine learning.

# Contents

<b>ملخص</b>	<b>iii</b>
<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Abbreviations</b>	<b>xvii</b>
<b>Introduction</b>	<b>1</b>
General context and issues . . . . .	1
Objectives . . . . .	2
Scientific Contributions . . . . .	2
Thesis Roadmap . . . . .	3
<b>I Backgrounds, Preliminaries, Basic Concepts and Literature Review</b>	<b>5</b>
<b>1 Evolution of Artificial Intelligence Paradigms</b>	<b>6</b>
1.1 Introduction . . . . .	7
1.2 Artificial Intelligence (AI) . . . . .	8
1.2.1 Historical background and evolution of AI . . . . .	8
1.2.2 AI definition . . . . .	9
1.2.3 AI applications . . . . .	11
1.2.3.1 AI in agriculture . . . . .	11
1.2.3.2 AI in healthcare . . . . .	12
1.2.3.3 AI in energy . . . . .	13
1.3 Machine learning . . . . .	14
1.3.1 Introduction to machine learning . . . . .	14
1.3.2 Types of machine learning . . . . .	15
1.3.2.1 Supervised learning . . . . .	15
1.3.2.2 Unsupervised learning . . . . .	15
1.3.2.3 Semi-supervised learning . . . . .	16
1.3.2.4 Reinforcement learning . . . . .	16
1.3.3 Popular machine learning algorithms . . . . .	17
1.3.3.1 Support Vector Machine (SVM) . . . . .	17

1.3.3.2	K-Nearest Neighbor (KNN) . . . . .	17
1.3.3.3	Decision Tree (DT) . . . . .	18
1.3.3.4	Random Forest (RF) . . . . .	18
1.3.3.5	K-means . . . . .	19
1.3.3.6	Fuzzy C-Means (FCM) . . . . .	19
1.3.3.7	Naïve Bayes (NB) . . . . .	20
1.3.3.8	Logistic Regression (LR) . . . . .	21
1.3.3.9	Gaussian Mixture Model (GMM) . . . . .	21
1.4	Deep learning . . . . .	23
1.4.1	Introduction to deep learning . . . . .	23
1.4.2	Deep learning basics . . . . .	24
1.4.2.1	Artificial Neural Network (ANN) . . . . .	24
1.4.2.2	Deep learning network . . . . .	27
1.4.3	Popular deep learning networks . . . . .	27
1.4.3.1	Convolutional Neural Network (CNN) . . . . .	27
1.4.3.2	Generative Adversarial Network (GAN) . . . . .	29
1.4.3.3	Autoencoders (AE) . . . . .	31
1.4.3.4	Transformer . . . . .	31
1.4.3.5	Long Short Term Memory Network (LSTM) . . . . .	34
1.5	Challenges and issues of machine learning and deep learning paradigms . . . . .	35
1.6	Conclusion . . . . .	36
<b>2</b>	<b>Plant Disease Detection with Artificial Intelligence</b>	<b>37</b>
2.1	Introduction . . . . .	38
2.2	Plant pathology . . . . .	39
2.3	Plant disease datasets . . . . .	45
2.3.1	PlantVillage . . . . .	45
2.3.2	Plant Pathology Challenge . . . . .	45
2.3.3	AI Challenger . . . . .	45
2.3.4	Plant Disease Symptoms (PDDB) . . . . .	46
2.4	Plant disease detection and classification techniques . . . . .	46
2.4.1	Plant disease detection and classification with machine learning techniques	46
2.4.1.1	Techniques for detecting a specific disease in a particular crop . . . . .	46
2.4.1.2	Techniques for detecting multiple diseases in a particular crop . . . . .	48
2.4.1.3	Techniques for detecting multiple diseases in multiple crops . . . . .	50
2.4.2	Plant disease detection and classification with deep learning techniques . . . . .	52
2.4.2.1	Techniques for detecting a specific disease in a particular crop . . . . .	53
2.4.2.2	Techniques for detecting multiple diseases in a particular crop . . . . .	54
2.4.2.3	Techniques for detecting multiple diseases in multiple crops . . . . .	58
2.5	Challenges and issues . . . . .	61
2.6	Conclusion . . . . .	61

<b>II</b>	<b>Proposed Plant Diseases Detection Systems</b>	<b>62</b>
<b>3</b>	<b>Deep Learning Approach for Cross-Crop Plant Disease Detection</b>	<b>63</b>
3.1	Introduction . . . . .	64
3.1.1	Research questions and hypotheses . . . . .	64
3.2	Materials and methods . . . . .	66
3.2.1	Dataset . . . . .	66
3.2.2	The proposed approach . . . . .	67
3.2.2.1	Generalization of the disease detection process to any crop type	68
3.2.2.2	Generalization of the disease detection process to any disease type	69
3.2.2.3	Detecting and determining the extent of the disease . . . . .	72
3.2.3	CNN Network . . . . .	73
3.2.4	Experimental setup . . . . .	73
3.3	Results and discussion . . . . .	74
3.3.1	Evaluation of the proposed methods . . . . .	74
3.3.2	Evaluation on new crop and new disease . . . . .	77
3.3.3	Comparison with other state-of-the-art methods . . . . .	78
3.3.4	Discussion . . . . .	79
3.4	Conclusion . . . . .	81
<b>4</b>	<b>Deep Learning Approach for Simultaneous Multi-Disease Detection on the Same Leaf</b>	<b>82</b>
4.1	Introduction . . . . .	83
4.1.1	Research questions and hypotheses . . . . .	83
4.2	Materials and methods . . . . .	84
4.2.1	Dataset . . . . .	84
4.2.2	A generalized method for simultaneous multi-disease recognition from the same leaf . . . . .	85
4.2.3	Proposed system . . . . .	85
4.2.3.1	Isolation technique . . . . .	85
4.2.3.2	The new PlantVillage dataset . . . . .	88
4.2.3.3	Classification and assessment of diseases extent . . . . .	92
4.2.4	CNN networks . . . . .	92
4.2.5	Experimental setup . . . . .	93
4.3	Results and discussion . . . . .	94
4.3.1	Results of the comparative experiments evaluating the performance of CNN models . . . . .	94
4.3.2	Results of imbalanced dataset handling and performance analysis of the proposed method . . . . .	97
4.3.2.1	Imbalanced dataset handling results . . . . .	97
4.3.2.2	Performance analysis of the proposed method . . . . .	97
4.3.3	Discussion . . . . .	100

4.4	Conclusion . . . . .	101
<b>5</b>	<b>Intelligence System for Detecting Diseases in Apple Tree Branches</b>	<b>102</b>
5.1	Introduction . . . . .	103
5.1.1	Research questions and hypotheses . . . . .	103
5.2	Materials and methods . . . . .	104
5.2.1	Dataset . . . . .	104
5.2.2	Proposed system . . . . .	105
5.2.2.1	Acquisition . . . . .	105
5.2.2.2	Preprocessing . . . . .	105
5.2.2.2.1	Conversion of color space . . . . .	106
5.2.2.2.2	Contrast enhancement . . . . .	106
5.2.2.2.3	Noise reduction . . . . .	107
5.2.2.2.4	Threshold and morphological operations . . . . .	107
5.2.2.2.5	Operation "AND" bit-by-bit . . . . .	108
5.2.2.3	Gaussian Mixture Model . . . . .	108
5.2.2.3.1	Initialization . . . . .	108
5.2.2.3.2	Parameters update . . . . .	108
5.2.2.4	Creation of the mask . . . . .	109
5.2.2.5	Detection . . . . .	110
5.2.3	Experimental setup . . . . .	111
5.3	Results and discussion . . . . .	111
5.3.1	Qualitative results . . . . .	111
5.3.2	Quantitative results . . . . .	117
5.4	Conclusion . . . . .	117
	<b>Conclusion and Perspectives</b>	<b>119</b>
	<b>Bibliography</b>	<b>121</b>
	<b>Author's publication</b>	<b>135</b>



# List of Figures

1.1	Supervised learning (from [77]). . . . .	15
1.2	Unsupervised learning (from [77]). . . . .	16
1.3	Semi-supervised learning (from [77]). . . . .	16
1.4	Reinforcement learning (from [77]). . . . .	17
1.5	Neuron structure: (a) biological neuron, (b) artificial neuron (from [88]). . . . .	25
1.6	A structure of an ANN (from [38]). . . . .	25
1.7	ANN architectures: (a) feedforward architecture, (b) feedback architecture (from [37]). . . . .	26
1.8	A structure of a DNN (from [37]). . . . .	27
1.9	A structure of a CNN (from [136]). . . . .	28
1.10	A convolution operation process (from [107]). . . . .	29
1.11	Pooling operations: (a) max pooling operation, (b) average pooling operation (from [107]). . . . .	29
1.12	A GAN structure (from [140]). . . . .	30
1.13	An AE structure (from [11]). . . . .	32
1.14	A transformer structure (from [72]). . . . .	33
1.15	A LSTM structure (from [92]). . . . .	35
2.1	Types of plant diseases and their respective causal factors. . . . .	40
2.2	Bar chart of accuracy for machine learning-based methods for detecting a specific diseases in a particular crop. . . . .	48
2.3	Bar chart of accuracy for machine learning-based methods for detecting multiple diseases in a particular crop. . . . .	51
2.4	Bar chart of accuracy for machine learning-based methods for detecting multiple diseases in multiple crops. . . . .	52
2.5	Bar chart of accuracy for deep learning-based methods for detecting a specific disease in a particular crop. . . . .	54
2.6	Bar chart of accuracy for deep learning-based methods for detecting multiple diseases in a particular crop. . . . .	58
2.7	Bar chart of accuracy for deep learning-based methods for detecting multiple diseases in multiple crops. . . . .	60
3.1	Examples of PlantVillage dataset images: (a) images with background, (b) images segmented from the background. . . . .	66
3.2	Flowchart of the proposed model-based CNN. . . . .	67

3.3	Image splitting examples: (a) infected grape leaf image, (b) healthy strawberry leaf image. . . . .	69
3.4	Data labeling examples: (a) infected grape leaf image, (b) healthy strawberry leaf image. . . . .	70
3.5	The Small Inception model architecture. . . . .	73
3.6	Training set loss and accuracy progression . . . . .	75
3.7	The new PlantVillage dataset confusion matrix. . . . .	76
3.8	Data examples: (a) healthy tomato leaf image, (b) Powdery mildew-infected squash leaf image, (c) Huanglongbing (Citrus greening) infected orange leaf image. . . . .	76
3.9	PDDDB dataset examples: (a) cotton leaf (b) soybean leaf (c) coffee leaf. . . . .	77
3.10	PDDDB dataset confusion matrix. . . . .	78
3.11	Dataset samples: Examples. . . . .	80
4.1	The examples of disease symptoms include: (a) Black rot infection on a potato leaf, (b) Black rot infection on a tomato leaf, (c) Bacterial spot infection on a peach leaf, and (d) Bacterial spot infection on a pepper leaf. . . . .	85
4.2	Flowchart of the proposed classification model based CNN . . . . .	86
4.3	Image splitting examples : (a) Black rot infected grape leaf, (b) Black rot infected apple leaf. . . . .	88
4.4	Data labeling examples : (a) Black rot infected grape leaf, (b) Black rot infected apple leaf. . . . .	89
4.5	Flowchart of the prediction process and calculation of prevalence rate and extent of the diseases in the leaf. . . . .	93
4.6	Evolution of loss and accuracy on the training set for Small Inception . . . . .	95
4.7	Evolution of loss and accuracy on the training set for MiniVGGNet . . . . .	95
4.8	Evolution of loss and accuracy on the training set for LeNet5 . . . . .	95
4.9	Comparison of confusion matrices from the Small Inception CNN: unbalanced (top values) vs. balanced (bottom values) . . . . .	98
5.1	General diagram of the disease spot detection process. . . . .	105
5.2	The proposed preprocessing process. . . . .	106
5.3	GMM stages. . . . .	110

# List of Tables

2.1	Summary of the common plant diseases, their pathogens, and symptoms. . . . .	40
2.1	Summary of the common plant diseases, their pathogens, and symptoms. . . . .	41
2.1	Summary of the common plant diseases, their pathogens, and symptoms. . . . .	42
2.1	Summary of the common plant diseases, their pathogens, and symptoms. . . . .	43
2.1	Summary of the common plant diseases, their pathogens, and symptoms. . . . .	44
2.1	Summary of the common plant diseases, their pathogens, and symptoms. . . . .	45
2.2	Comparative analysis of machine learning-based methods for detecting a specific disease in a particular crop. . . . .	47
2.2	Comparative analysis of machine learning-based methods for detecting a specific disease in a particular crop. . . . .	48
2.3	Comparative analysis of machine learning-based methods for detecting multiple diseases in a particular crop. . . . .	50
2.4	Comparative analysis of machine learning-based methods for detecting multiple diseases in multiple crops. . . . .	52
2.5	Comparative analysis of deep learning-based methods for detecting a specific disease in a particular crop. . . . .	54
2.6	Comparative analysis of deep learning-based methods for detecting multiple diseases in a particular crop. . . . .	57
2.7	Comparative analysis of deep learning-based methods for detecting multiple diseases in multiple crops. . . . .	60
3.1	The new PlantVillage dataset details. . . . .	71
3.1	The new PlantVillage dataset details. . . . .	72
3.2	Hyperparameters configuration . . . . .	74
3.3	Performance on training and testing Sets . . . . .	75
3.4	Performance on testing set . . . . .	77
3.5	A comparative study with state-of-the-art methods using the PlantVillage dataset	79
4.1	The PlantVillage dataset class details. . . . .	89
4.1	The PlantVillage dataset class details. . . . .	90
4.2	The new PlantVillage dataset details. . . . .	91
4.2	The new PlantVillage dataset details. . . . .	92
4.3	Hyperparameters configuration . . . . .	93
4.4	Training performances of the three CNN models . . . . .	94
4.5	Testing performances of the three CNN models . . . . .	96

---

4.6	Accuracies obtained for individual disease classes using the employed CNNs. . . .	96
5.1	Number of images per background variant . . . . .	104
5.2	Number of images per disease variant . . . . .	104
5.3	System parameters . . . . .	111
5.4	Results of <i>Nectria cinnabarina</i> disease detection in apple tree branches for selected test images . . . . .	112
5.4	Results of <i>Nectria cinnabarina</i> disease detection in apple tree branches for selected test images . . . . .	113
5.4	Results of <i>Nectria cinnabarina</i> disease detection in apple tree branches for selected test images . . . . .	114
5.4	Results of <i>Nectria cinnabarina</i> disease detection in apple tree branches for selected test images . . . . .	115
5.4	Results of <i>Nectria cinnabarina</i> disease detection in apple tree branches for selected test images . . . . .	116
5.5	Results in terms of F1-score, recall, and precision for selected test images . . . .	117

---

# ABBREVIATIONS

AE	Autoencoders
AHE	Adaptive Histogram Equalization
AI	Artificial Intelligence
ANN	Artificial Neural Network
AWS	Amazon Web Services
BP	Back Propagation
BPNN	Back Propagation Neural Network
BPTT	Back Propagation Through Time
CAE	Convolutional Autoencoder
CBAM	Convolutional Block Attention Module
CCF	Cross Central Filter
CFCM	Cognitive Fuzzy C-Means
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
C-SVC	Cost-Support Vector Classification
DLQP	Directional Local Quinary Patterns
DNN	Deep Neural Network
DT	Decision Tree
ELU	Exponential Linear Unit
EM	Expectation-Maximization
FAO	Food and Agriculture Organization of the United Nations

---

FCM	Fuzzy C-Means
FPN	Feature Pyramid Network
GAN	Generative Adversarial Network
GLCM	Gray Level Cooccurrence Matrix
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
GUI	Graphical User Interface
IoT	Internet of Things
KNN	K-Nearest Neighbor
LBP	Local Binary Pattern
LBPH	Local Binary Pattern Histogram
LR	Logistic Regression
LSTM	Long Short Term Memory Network
LTP	Local Ternary Pattern
LTriDP	Local Tri-Directional Pattern
MDC	Minimum Distance Classifier
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NAG	Nesterov's Accelerated Gradient
NB	Naive Bayes
PCA	Principal Component Analysis
RBF	Radial Basis Function
RBN	Radial Basis Network
RDF	Resource Description Framework
RELU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
SIFT	Scale Invariant Feature Transformation
SSD	Single-Shot multibox Detector
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
UK	United Kingdom

WT	Wavelet Transform
WWW	World Wide Web

---

# INTRODUCTION

## General context and issues

Agriculture holds paramount significance as the primary source of global food security and as an important economic sector in every country. Its development is crucial not only for meeting the increasing demands of a growing population but also for generating substantial profits from surplus production. The agricultural sector faces numerous threats that can hinder the optimal production of agricultural products, consequently disrupting the food supply chain and heightening the risks of food shortages. Additionally, economic dependency resulting from these losses impacts countries' economies. Among these threats, plant diseases stand out as the most serious threat. Early detection of plant diseases is crucial for effectively combating them. Early detection poses a significant challenge, as it demands meticulous monitoring of crops in the field, collecting diagnostic data, and extensive expertise in disease identification. Consequently, this process is time-consuming. With the emergence of modern technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), remote sensing, and others, the terms "smart agriculture" and "precision agriculture" have come to the forefront. These terms encompass the application of these advanced technologies in agriculture to enhance reliability, efficiency, and automation in farming practices, including the prediction and early detection of diseases. Two primary methods for automatic disease prediction have emerged. The first method involves the use of sensors distributed throughout the field, which collect information on environmental conditions and physiological changes in plants associated with disease emergence. While this method can provide detailed data, it is considered ineffective due to its high cost, especially for large fields where a significant number of sensors would be required. The second method involves predicting diseases through image analysis. In this approach, drones or surveillance cameras capture images of the plants in the field. These images are then analyzed to identify any defects or signs of disease in the plants. This method is more cost-effective and scalable, making it a practical alternative for large-scale agricultural operations.

To analyze the images taken from the field, the machine learning method was used in the beginning. With the advent of deep learning, and considering its superiority over machine learning, it became the ideal solution for identifying plants from images. Consequently, many methods were proposed in the literature that rely on deep learning, but there are many challenges facing the optimal use of this technology. The main challenge is the lack of data. Creating a



comprehensive dataset that includes all types of plants and diseases poses significant difficulty due to the vast number of plant species and diseases worldwide. Capturing data for each presents a formidable challenge, compounded by the fact that some diseases spread widely, making timely capture challenging. Furthermore, capturing multiple infections simultaneously may take several years, if not proving impossible. This explains the limited availability in the proposed methods, as most of them typically address the lack of data through the use of data augmentation and transfer learning. However, they still lack robustness, generalization, and multi-disease detection, as they are often specific to a particular combination of crop types or diseases within the training dataset. Therefore, it is imperative to develop solutions capable of simultaneously identifying any type of disease, any type of plant, and any number of infections, which serves as this thesis's main topic. Another topic in our thesis is the detection of diseases in tree branches, a subject not previously addressed in the literature.

## Objectives

This thesis will design novel AI-based systems to recognize plant disease detections. These systems must employ a new perspective for recognizing disease infections. This perspective must enable the long-term objective of addressing the following core issues:

- ❖ As it is impossible to collect a dataset that can be used to detect all types of diseases from all types of crops, the first solution is crucial: to have a system capable of distinguishing between infected and healthy leaves regardless of the type of crop and disease.
- ❖ As it is impossible to recognize all types of diseases that can infect crops, it is essential to determine the most common and well-known diseases that threaten agricultural plants. It is important to have a system capable of detecting multiple infections of these diseases in the same leaf simultaneously, regardless of their crop type.
- ❖ For determining the amount of chemical spraying, it is important to calculate the prevalence rate of each disease and assess the overall extent of all diseases present on the leaf.
- ❖ To ensure the health of all parts of plants in the field, it is important to have a system capable of detecting diseases that can infect the branches without affecting the leaves.

## Scientific Contributions

In this Ph.D. thesis, several contributions are made toward achieving the desired systems. The major contribution can be summarized as follows:

- ✍ **First contribution:** Our literature review on methods for detecting plant diseases has led us to propose a new taxonomy based on AI techniques, specifically machine learning and deep learning while considering the range of crops and diseases covered. This taxonomy categorizes disease infections into three main categories: unique disease infection from a unique crop type, multiple disease infections from one crop type, and multiple disease infections from multiple crop types. This categorization scheme offers a thorough framework for comprehending and classifying the different methods used in agricultural settings for disease detection.

- ✍ **Second contribution:** We have proposed a new generalized method for identifying unhealthy leaves, regardless of crop and disease types. This method utilizes a deep learning model, specifically the Small Inception architecture, and focuses on small leaf pieces that lack crop-specific characteristics and features, rather than using the entire leaf. The model is trained to extract common features present in both healthy and diseased leaves, irrespective of plant or disease type. By prioritizing the recognition of these shared characteristics, the model can effectively differentiate between healthy and diseased leaves across all crop and disease types. The small leaf pieces are created by splitting the leaves into small patches, each limited to the characteristics of the crops. The healthy patches, which do not exhibit any disease symptoms, are used to extract the features that represent healthy leaves, while the unhealthy patches containing disease symptoms are used to extract the features of unhealthy leaves.
- ✍ **Third contribution:** This method involves the independent identification of each disease, regardless of the crop type, by isolating each disease type from others and eliminating the influence of crop type variation. Instead of using the entire leaf in the training process, we utilize small leaf pieces. These pieces isolate each disease-specific feature, rather than relying on entire images containing leaves with distinct diseases and crop characteristics. The leaf is split into small patches, each isolating a specific disease type and eliminating crop-specific characteristics and features. All healthy patches extracted from healthy leaves, as well as from the healthy regions of unhealthy leaves, are labeled as healthy. For the remaining unhealthy patches, each patch containing specific disease symptoms is labeled with the corresponding disease. This ensures the complete elimination of crop-specific influences during training by using patches extracted from various crop types to train Small Inception models to learn features specific to each disease type. This ensures that disease characteristics identified in one crop can be applied to others.
- ✍ **Fourth contribution:** We have introduced a novel approach to identifying plant diseases in tree branches, leveraging a standard Resource Description Framework (RDF) across multiple stages: acquisition, preprocessing, relevant information extraction, and detection. The acquisition step involves capturing images from the field using drones or surveillance cameras. Preprocessing encompasses the application of various techniques to enhance image quality and accuracy, thereby facilitating classification. Relevant information extraction involves training a machine learning model, Gaussian Mixture Model (GMM), to extract relevant features and create a segmentation mask based on the probability distribution of disease color. Finally, detection involves disease spot segmentation using the generated segmentation mask.

## Thesis Roadmap

This thesis is organized into two main parts. The first part consists of state-of-the-art topics divided into two chapters, while the second part comprises contributions organized across three chapters. The thesis concludes with a general conclusion.

### ❖ **Part I: Backgrounds, Preliminaries, Basic Concepts and Literature Review**

#### □ *Chapter 01: "Evolution of Artificial Intelligence Paradigms"*

This chapter reviews the fundamental concepts of AI. Initially, we explore the domain

of AI, its definitions, and its most significant applications. Subsequently, we introduce its core components, machine learning and deep learning, and delve into their definitions, main types, and renowned models. Additionally, we address the primary challenges encountered when employing these paradigms.

□ *Chapter 02: “Plant Disease Detection with Artificial Intelligence”*

This chapter covers the current state of plant disease detection and its development. It begins with an overview of plant pathologies, including descriptions of common agricultural diseases. Next, it presents the most popular datasets for plant disease detection. This is followed by a comprehensive analysis of the latest approaches and methods proposed in this field, focusing on the detection and classification of plant diseases using deep learning and machine learning techniques. Lastly, it discusses the challenges and issues faced by plant disease identification and classification systems presented in the literature.

❖ **Part II: Proposed Plant Diseases Detection Systems**

□ *Chapter 03: “Deep Learning Approach for Cross-Crop Plant Disease Detection”*

This chapter presents our architecture for the proposed generalized system for identifying unhealthy leaves. It outlines the methods and materials used to develop this system, detailing the dataset and explaining the steps involved in its creation. The chapter then presents and discusses the results obtained from the evaluation of the proposed system.

□ *Chapter 04: “Deep Learning Approach for Simultaneous Multi-Disease Detection on the Same Leaf”*

This chapter describes our architecture for multi-disease detection on the same leaf simultaneously. It presents the methods and materials used to develop this system, including the dataset and proposed methods. Finally, it discusses the results obtained from the evaluation of the proposed system.

□ *Chapter 05: “Intelligence System for Detecting Diseases in Apple Tree Branches”*

In this chapter, we introduce the proposed system for detecting diseases in tree branches. It presents the steps for creating this system, starting from the collection of the dataset to the architecture of the system. Finally, it discusses the results of the evaluation of this system.

## Part I

# Backgrounds, Preliminaries, Basic Concepts and Literature Review

---

---

# CHAPTER 1

---

## EVOLUTION OF ARTIFICIAL INTELLIGENCE PARADIGMS

---

## Chapter contents

---

1.1	Introduction . . . . .	7
1.2	Artificial Intelligence (AI) . . . . .	8
1.3	Machine learning . . . . .	14
1.4	Deep learning . . . . .	23
1.5	Challenges and issues of machine learning and deep learning paradigms . . . . .	35
1.6	Conclusion . . . . .	36

---

### 1.1 Introduction

Since the emergence of modern computing in the late 19th and early 20th centuries, humanity has delved into the idea of "artificial intelligence". AI has undergone significant evolution over the decades, transitioning from early conceptualizations to its present status as a defining element of the modern era. With the progression of technology and the growing abundance of data, AI has permeated numerous domains, including business, healthcare, transportation, entertainment, and agriculture. AI can be defined as the simulation of human intelligence through the creation of intelligent computers. Precisely, AI refers to the capacity of a computer system or machine to perform tasks such as learning, problem-solving, and logical reasoning, which are typically associated with human intelligence [82, 118].

The rise of AI coincides with the emergence of machine learning as a pivotal sub-field within the AI domain. Machine learning can be defined as the type of AI wherein computers are capable of independent thinking and learning [14]. It entails the creation of algorithms for data analysis, utilizing this data to recognize patterns, and ultimately making decisions or predictions based on it. A highly esteemed machine learning technique is the Artificial Neural Network (ANN), designed to mimic the structure and operations of the human brain. With interconnected nodes resembling neurons, an ANN possesses the capability to efficiently analyze data and extract valuable insights from given instances.

The field of machine learning has made remarkable advancements in sophisticated learning algorithms and efficient pre-processing techniques [74]. A noteworthy breakthrough is the evolution of ANNs into deeper and more complex architectures, known as deep learning, which provide superior learning capabilities [56, 93]. Deep learning is the term used to describe ANNs that have delicate multi-layers [2]. Deep learning demonstrates superior performance compared to traditional machine learning methods in numerous scenarios, particularly when dealing with large datasets. This stands as its primary advantage over traditional machine learning techniques [161, 188].

This chapter will delve into the core principles of AI, focusing on machine learning and deep learning as fundamental components. ANNs will be highlighted as the bridge between these concepts. In Section 1.2, we will explore the domain of AI and its diverse applications. Following that, Section 1.3 will introduce the machine learning paradigm, detailing its types and prominent algorithms. In Section 1.4, we will delve into deep learning, where ANNs serve as the foundation.

We will discuss various types of deep learning networks. Finally, in Section 1.5 we tackled the challenges faced in both machine learning and deep learning.

## 1.2 Artificial Intelligence (AI)

### 1.2.1 Historical background and evolution of AI

The historical background and evolution of AI trace back to ancient times, with the concept of creating modern computer in the late 19th and early 20th centuries. However, the mid-20th century marked the start of the modern AI era.

In 1943, the term "artificial intelligence" hadn't yet been coined, but the seeds of this revolutionary field were sown. It was during this time that the concept of ANNs emerged, spearheaded by Warren McCulloch and Walter Pitts, marking a significant milestone in the journey toward creating intelligent machines with the publication of their paper titled "A Logical Calculus of the Ideas Immanent in Nervous Activity" [3].

In 1950, Alan Turing published a seminal paper titled "Computing Machinery and Intelligence," where he introduced his proposal commonly known as "the turing test" which aimed to rationalize and question whether machines could exhibit human-like intelligence [60].

---

*"I propose to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. ... [But] Instead of attempting such a definition I shall replace the question by another... The new form of the problem can be described in terms of a game which we call the 'imitation game'."*

---

ALAN TURING, "COMPUTING MACHINERY AND INTELLIGENCE", 1950

The "imitation game" serves as an alternative name for the turing test. Turing's objective with this test was to provide a means of assessing whether a machine could exhibit intelligent behavior comparable to human thought processes [139]. An interrogator communicates with both a human and a machine via a text interface, attempting to discern which is which based solely on their responses. If the interrogator is unable to consistently differentiate between the human and the machine, the machine is deemed to have passed the test, demonstrating behavior that is indistinguishable from that of a human [170].

---

*"I believe that in about fifty years' time it will be possible to programme computers, with a storage capacity of about 10<sup>9</sup>, to make them play the imitation game so well that an average interrogator will not have more than 70 percent chance of making the right identification after five minutes of questioning. ... I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted."*

---

ALAN TURING, 1950

In 1955, american computer scientist and cognitive scientist John McCarthy coined the term "artificial intelligence" [95], and he proposed the creation of a conference dedicated to AI. This

proposal was officially announced by John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon on September 2, 1955 [53]. The 1956 Dartmouth summer research project on AI was a pivotal event in the field's history, where the term "artificial intelligence" was officially coined [113].

*"We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer."*

---

MCCARTHY, MARVIN MINSKY, NATHANIEL ROCHESTER AND CLAUDE SHANNON, 1955

From the official coinage of the concept of AI, this field has indeed witnessed several stages. From the period from 1956 to 1974, known as the era of early AI research, significant developments occurred that laid the groundwork for the field. Researchers focused on symbolic AI, developing programs capable of logical reasoning and problem-solving, creating the groundwork for modern AI. The period from the mid-1974 to the early 1980s is indeed known as the first AI winter. The field experienced periods of both enthusiasm and skepticism, leading to funding being stopped temporarily. This was due to challenges in scaling AI systems and limitations in computing power, which halted progress temporarily. In the period from 1980 to 1987, AI was back on the scene, and several achievements were made such as expert systems, machine learning, and natural language processing. From 1987 to 1993, the AI winter returned, named the second AI winter. During this time, many AI projects failed, and AI research declined due to reduced funding and skepticism. Between 1993 and 2011, the concept of AI agents evolved significantly with advancements in areas such as intentional systems and multi-agent systems. AI permeated various aspects of daily life, becoming increasingly integrated into technology, communication, and decision-making processes. Beginning in 2011, significant advancements in AI have catalyzed the emergence of big data and the concept of deep learning. From the year 2020, interest in AI shifted towards large language models, with several notable examples such as ChatGPT emerging onto the scene.

### 1.2.2 AI definition

Since the birth of the term "artificial intelligence", various definitions have been proposed by different scientists and researchers according to their knowledge and viewpoints. AI has many definitions over its long history, but there is no standard, unified definition so far. For this reason, here are some of the most frequently cited definitions in the literature proposed by eminent experts in the field:

In 1955, John McCarthy coined the term AI as:

*"The science and engineering of making intelligent machines."* (John McCarthy, [10])

Marvin Minsky, in 1968, defined AI as:



"*The science of making machines do things that would require intelligence if done by men.*" (Marvin Minsky, [115])

Nils J. Nilsson, in 1971, described AI as:

"*The goal of work in artificial intelligence is to build machines that perform tasks normally requiring human intelligence.*" (Nils J. Nilsson, [123])

In 1977, RJ Nelson explained AI as:

"*The use of computer programs and programming techniques to cast light on the principles of intelligence in general and human thought in particular.*" (RJ Nelson, [122])

Elaine Rich, in 1985, defined AI as:

"*The study of how to make computers do things at which, at the moment, people are better.*" (Elaine Rich, [149])

Ray Kurzweil, in 1990, characterized AI as:

"*The art of creating machines that perform functions that require intelligence when performed by people.*" (Ray Kurzweil, [182])

Russell and Norvig, in 1995, conceptualized AI as:

"*Systems that think like humans, systems that act like humans, systems that think rationally, systems that act rationally.*" (Marvin Minsky, [150])

Nils J. Nilsson, in 2010, defined AI as:

"*That activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment.*" (Nils J. Nilsson, [124]).

Additionally, he described AI as:

"*the study of agents that receive percepts from the environment and perform actions.*" (Nils J. Nilsson, [151])

In 2018, various experts offered their perspectives on AI. Elon Musk described it as:

"*A complex adaptive system that maximizes its expected utility in a broad class of environments.*" (Elon Musk, [182])

Andrew Ng defined it as:

"*A set of algorithms and intelligence to enable computers to learn, see, hear, speak, and make decisions like humans.*" (Andrew Ng , [182])

Judea Pearl characterized AI as:

"*A scientific discipline concerned with understanding the mechanisms underlying thought and intelligent behavior and their embodiment in machines.*" (Judea Pearl , [182])

Yann LeCun defined it as:

"*The ability of a machine to perceive its environment and take actions that maximize its chance of success in some goal.*" (Yann LeCun , [182])

In 2018, the United Kingdom (UK) Parliament adopted the definition outlined in the 2017 industrial strategy white paper:

*"Technologies with the ability to perform tasks that would otherwise require human intelligence."* (UK parliament, [65])

In 2021, the European Union published its AI proposal to regulate the use and transparency of AI, aiming to establish a single future-proof definition of AI:

*"Software developed with techniques and approaches that can generate outputs influencing the environments they interact with."* (European Commission, [91])

Lastly, according to the Oxford Dictionary, AI is:

*"The theory and development of computer systems able to perform tasks that normally require human intelligence."* (Oxford Dictionary, [43])

While the Merriam-Webster Dictionary defines it as:

*"A branch of computer science dealing with the simulation of intelligent behavior in computers."* (Merriam-Webster Dictionary, [43])

Based on all previously cited definitions of AI, we can offer the following summarized definition:

*"AI is a field of computer science that, as its name suggests, attempts to create intelligence for machines to simulate human intelligence. This involves making machines behave and perform tasks that typically require human intelligence, such as thinking, learning, making decisions, solving problems, and recognizing patterns."*

### 1.2.3 AI applications

AI has revolutionized all aspects of life. It is utilized in numerous fields, bringing transformative changes. In this section, we will present some of the most impressive AI applications.

#### 1.2.3.1 AI in agriculture

AI is utilized throughout all stages of agriculture, from the initial planting to eventual harvesting. By analyzing data and images acquired through the IoT technologies such as sensors, drones, and satellite imagery, AI enhances various agricultural practices [108]. The objective is to facilitate timely interventions and precise actions to maintain crop quality and optimize yields. The following are some important areas in which AI is making an impact in agriculture:

- ❖ *Crop monitoring / precision agriculture:* Farmers are able to track various aspects of their crops, including growth, productivity, and health, by utilizing data collected through IoT technologies and analyzed by AI-driven analytics. With this information, farmers can implement appropriate agricultural practices to enhance crop quality, increase yields, and optimize farming procedures. By consistently monitoring crop conditions and making data-driven decisions, farmers can achieve greater efficiency and productivity.

- ❖ *Disease detection*: Machine and deep learning algorithms analyze images of crops to identify signs of diseases. This allows farmers to be informed of potential threats in the field, enabling them to promptly investigate and take action, such as using chemical pesticides, to prevent crop loss.
- ❖ *Pest detection*: The data collected by IoT sensors and cameras positioned in the field or traps can be utilized to detect pests and insects. By accurately identifying the type of insect present, precision spraying of chemical pesticides can be implemented. This approach reduces environmental pollution and safeguards crop health.
- ❖ *Weed detection*: Farmers can utilize computer vision algorithms to identify and categorize weeds, allowing them to implement the most suitable weed management and control techniques.
- ❖ *Pesticide application*: AI technologies empower farmers to make well-informed decisions regarding the types and quantities of pesticides necessary to mitigate threats identified within the farm. AI algorithms can precisely evaluate the severity of the threat and propose appropriate pesticide remedies. Additionally, AI-driven drones and robots outfitted with precision spraying mechanisms can effectively administer pesticides to specific sections of the farm. This not only helps in minimizing crop damage but also reduces the environmental impact by optimizing pesticide usage.
- ❖ *Yield prediction*: Machine and deep learning models, trained on historical crop data and considering various environmental conditions and agricultural practices, can forecast future crop yields. Based on this predictive capability, farmers can strategize their marketing plans more effectively.
- ❖ *Water management / irrigation*: Using data collected by IoT sensors on soil moisture levels and weather forecasts, AI systems can analyze the current conditions and accurately calculate the amount of water needed for irrigation. Additionally, AI can automate the irrigation process, ensuring that crops receive the appropriate amount of water at the right time.
- ❖ *Harvesting*: Computer vision systems can alert farmers when crops are ready for harvest by detecting their maturity. Additionally, AI-powered robotic arms can automate the harvesting process, enhancing efficiency and reducing labor costs.

### 1.2.3.2 AI in healthcare

AI is making a revolutionary impact on the healthcare process, transforming various aspects of medical practice. By analyzing medical data and images gathered by systems through a network of medical device-connected sensors, AI enhances several healthcare practices. The objective is to facilitate timely interventions and precise actions to provide safe and effective health services to patients. Here are some important areas in which AI is making an impact in healthcare:

- *Drug discovery*: Through the analysis of chemical and biological data, AI can identify new molecules and predict their effects. By assessing drug compounds and structures, AI can anticipate patient responses to these drugs and forecast potential interactions. Moreover, AI evaluates the risks associated with these medications, offering insights crucial for decision-making in drug development.

- *Clinical trials:* AI is pivotal across all phases of clinical trials, spanning from protocol design to patient recruitment, data analysis, and regulatory compliance. By leveraging historical trial data and patient records, AI aids in protocol design and predicts suitable patient cohorts for trial participation. Through robust analysis of trial outcomes, AI enhances decision-making processes and refines trial strategies. Additionally, AI facilitates conducting trials via simulation methods, providing invaluable insights for optimizing trial protocols and ensuring their success.
- *Patient care:* AI is revolutionizing patient care, starting with disease diagnosis, customization of treatment plans, remote patient monitoring, and communication. Through the analysis of medical data such as ECGs and vital signs, along with medical images like X-rays, MRIs, and CT scans, AI enables disease detection. Additionally, treatment is tailored by analyzing the patient's historical data and vital signs, aiding in formulating treatment plans. Through the use of wearable devices equipped with AI, patients can be monitored remotely to assess their vitality and health. Moreover, the utilization of virtual assistants and chatbots enables remote communication with patients to access medical services remotely.

### 1.2.3.3 AI in energy

AI has significantly transformed and enhanced the energy sector, seamlessly integrating into every phase from initial production to storage, distribution, and consumption. Through comprehensive analysis of production, consumption, weather patterns, wind, and solar activity data, AI optimizes energy utilization, thereby boosting efficiency. Below, we will delve into the key applications of AI in the energy field:

- *Smart grid:* AI is pivotal in enhancing the performance of smart energy networks, as it enables monitoring of network performance for early detection and prediction of outages. Additionally, it facilitates effective management of energy demand by predicting demand patterns and analyzing consumer behavior. This aids in more efficient production planning and energy distribution.
- *Renewable energy:* AI systems are enhancing the efficiency and sustainability of renewable energy sources by accurately predicting energy production from sources like wind and solar. By analyzing data related to wind patterns and solar radiation, AI can determine the optimal times and locations for energy extraction, thereby maximizing renewable energy utilization.
- *Energy storage:* AI significantly enhances the interactive efficiency of energy storage systems. Through predictive analytics, AI accurately forecasts energy demand and efficiently manages the charging and discharging processes. Moreover, AI plays a crucial role in extending battery life cycles by predicting potential failures and implementing improved maintenance strategies.
- *Energy trending:* AI indeed plays a crucial role in the energy trading market by analyzing data related to energy trading. It accurately predicts energy demand, provides insights into energy prices, enables informed decisions regarding optimal buying and selling times, and facilitates the management of supply and demand operations.

## 1.3 Machine learning

### 1.3.1 Introduction to machine learning

The term "machine learning" was coined by Arthur Samuel in 1959 in his paper titled "Some Studies in Machine Learning Using the Game of Checkers" [157], which dealt with the use of self-learning and pattern recognition to teach a computer program to play checkers [125]. Over the years, several definitions of machine learning have been proposed. The following presents the most cited machine learning definitions in the research literature:

Arthur Samuel (1959):

*"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed."* (Arthur Samuel, [157])

Tom M. Mitchell (1997):

*"A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ ."* (Tom M. Mitchell, [116])

Ethem Alpaydin (2004):

*"Machine learning is programming computers to optimize a performance criterion using example data or past experience."* (Ethem Alpaydin, [12])

Daphne Koller and Nir Friedman (2009):

*"Machine learning explores the study and construction of algorithms that can learn from and make predictions or decisions based on data."* (Daphne Koller and Nir Friedman, [87])

Kevin P. Murphy (2012):

*"Machine learning is a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data or other outcomes of interest."* (Kevin P. Murphy, [119])

Pedro Domingos (2015):

*"Machine learning is about predicting the future based on the past."* (Pedro Domingos, [44])

Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016):

*"Machine learning is a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data or perform other kinds of decision making under uncertainty."* (Ian Goodfellow, Yoshua Bengio, and Aaron Courville, [56])

Based on all previously cited definitions of machine learning, we can propose the following summarized definition:

*"Machine learning is a subset of AI, aim to learn machines in a manner analogous to human learning processes. This involves training machines on a dataset and directing them to extract distinctive characteristics or features from it. This allows machines to perform predictions, make decisions, and recognize patterns in new data based on their learning experience."*

### 1.3.2 Types of machine learning

Machine learning includes many different types that are essential to the field. These main types include:

#### 1.3.2.1 Supervised learning

Supervised machine learning is a machine learning kind where the training process relies on a labeled dataset, which means a dataset containing target values or labeled answers for the output, along with examples for the input [74]. The goal of the machine learning here is to optimize the parameters to reduce the gap between the predicted label and the target label [77]. Two major categories can be distinguished in supervised learning:

- ❑ **Classification:** The predicted label by the machine learning algorithm is a categorical value. For example, if we have a dataset with three classes (A, B, C), supervised machine learning will learn how to differentiate between these classes. It will be trained to predict which one of these three classes a new data point belongs to based on the patterns it learned from the labeled training data.
- ❑ **Regression:** The predicted label by the machine learning algorithm is a continuous value. For example, the machine learning algorithm learns from a dataset where each data point has a continuous numeric value associated with it. It learns patterns in the data and is trained to predict a continuous numeric value for new data points, based on the relationships it learned from the training data.

Supervised machine learning is illustrated in Figure 1.1.

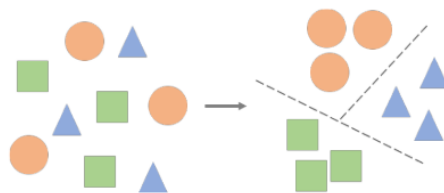


FIGURE 1.1: Supervised learning (from [77]).

#### 1.3.2.2 Unsupervised learning

Unsupervised machine learning is a type of machine learning where the training process relies on an unlabeled dataset, which means a dataset containing the input without the corresponding target values [153]. The goal of the learning system is to detect patterns and cluster the data without any previously assigned labels, based on the similarity within the training dataset [74, 77]. For example, let's consider a dataset with various data points. The algorithm's goal is to cluster these data points based on their similarity. This means that the algorithm will group together data points with similar characteristics, even though these groups are not predefined or labeled. Unsupervised machine learning is illustrated in Figure 1.2.

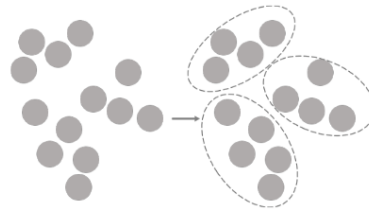


FIGURE 1.2: Unsupervised learning (from [77]).

### 1.3.2.3 Semi-supervised learning

Unsupervised and supervised learning are combined in semi-supervised machine learning, which is a hybrid approach. The training procedure in semi-supervised learning depends on a dataset with a large proportion of unlabeled data and a small proportion of labeled data [153]. The goal of machine learning here is for labeled data is to minimize the mistake between the target label and the predicted label, while for unlabeled data, it is to group them based on their similarities [77]. For instance, consider a dataset containing both labeled data points, categorized into three classes (A, B, C), and unlabeled data points. In semi-supervised learning, a small subset of labeled data is used to learn how to differentiate between classes. However, it also leverages the larger pool of unlabeled data to refine its understanding of the data and improve its ability to classify new instances within the known classes (A, B, C). Semi-supervised machine learning is illustrated in Figure 1.3.

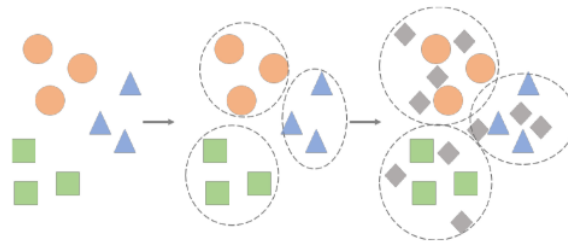


FIGURE 1.3: Semi-supervised learning (from [77]).

### 1.3.2.4 Reinforcement learning

Reinforcement learning is a machine learning kind where an agent, or learner, learns to make consecutive decisions to achieve an optimized final reward by performing actions and receiving rewards or penalties [153]. The external environment provides the input for reinforcement learning, and the action itself produces the output. The environment and critics determine the reward or penalty, and the parameters are modified to optimize rewards and reduce penalties [77]. Consider a robot that is learning how to get around a maze. The robot's sensors provide input, informing it about its current location and environment. The robot makes various actions, including forward, left, and right turns, as it navigates the maze. It gets input from the environment every time it moves: it gets rewarded if it approaches the exit and gets penalized if it strikes a wall or goes in the opposite direction. The robot eventually discovers the most effective way out of the maze after learning to optimize its route over time by maximizing rewards and minimizing penalties. Reinforcement machine learning is illustrated in Figure 1.4.

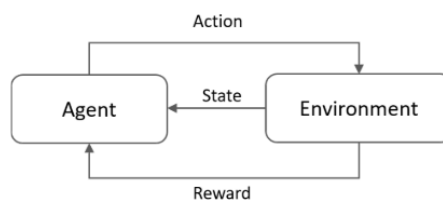


FIGURE 1.4: Reinforcement learning (from [77]).

### 1.3.3 Popular machine learning algorithms

#### 1.3.3.1 Support Vector Machine (SVM)

SVM [180] is an algorithm for supervised machine learning that can be applied to both regression and classification tasks, although it is particularly beneficial for classification. It is effective for both linear and non-linear data. The following outlines the key ideas behind the SVM algorithm:

- *Hyperplane*: In SVM, the algorithm aims to optimize the boundary that separates the classes. For linear data, this boundary is a dividing line, while for non-linear data, it corresponds to a separate space.
- *Support vectors*: Support vectors are data points closest to the hyperplane from each class, playing a crucial role in determining the hyperplane's position.
- *Margin*: It is the space between the hyperplane and the support vectors. The margin is calculated by finding the perpendicular distance between the hyperplane and the support vectors.
- *Kernel trick*: When dealing with non-linear data, SVM uses kernel functions, such as polynomial, sigmoid, Radial Basis Function (RBF), and linear, to transform the data into a higher-dimensional space, enabling linear separation.

The SVM algorithm aims to draw a hyperplane that best separates the data points in the feature space into different classes. This hyperplane guarantees the maximum possible margin between it and the nearest data points of each class [110].

#### 1.3.3.2 K-Nearest Neighbor (KNN)

KNN [6] is a simple and popular supervised machine learning algorithm that performs both classification and regression tasks. It is non-parametric and frequently referred to as "lazy learning". For each new data point, the KNN algorithm determines its class based on the distances to the  $k$ -nearest neighbors from the training data points [14]. The steps of the KNN proceed as follows:

- *K value selection*: Determining the value of  $K$ , the hyperparameter representing the number of nearest neighbors, should be the first step. Choosing this variable plays a crucial role in achieving optimal performance of the algorithm.



- *Distance calculation*: The distance between each new data point and all training data must be calculated using one of these distance metrics: Euclidean, Manhattan, or Minkowski distances.
- *K nearest neighbors determination*: Based on the calculated distances between the new data point and the training data, the K training data points with the smallest distances, which are the closest neighbors, are identified as the K nearest neighbors of the new data point.
- *Class label determination*: The class label that appears most frequently among the selected K nearest neighbors determine the class label for the new data point in the case of classification. In the case of regression, the average of the values of the K nearest neighbors is the value chosen for the new data point.

### 1.3.3.3 Decision Tree (DT)

DT is a well-known technique for non-parametric supervised learning. Both the classification and regression tasks employ DT learning techniques. There are several DT algorithms; the most well-known ones are CART [29], ID3 [144], and C4.5 [145]. The components of a DT include a root node, which is the first node of the tree and represents the data points. The decision nodes are the internal nodes that represent the decision attributes used for splitting the data points. The branches are the outputs of the decision nodes. The leaf nodes, which do not have branches, represent the class labels. A DT uses feature values to classify instances, moving them from the root node to a few leaf nodes [15]. The steps of the DT are as follows:

- *Decision attribute selection*: The best attributes that divide the dataset are chosen by minimizing impurity as much as possible. These attributes are selected using either Gini impurity or entropy.
- *Data splitting*: Using the chosen attributes, the dataset is divided into two subsets, with each subset representing a decision node.
- *Tree building*: By repeating the two previous processes for each decision node, the tree will be built until it reaches one of the chosen stopping conditions.
- *Class label determination*: Each leaf node is assigned a class label. In classification cases, the label corresponds to the majority class of the data points in that leaf node. In regression cases, it is the mean value of the data points within that leaf node.
- *Classification / regression with DT*: The resulting DT is used to predict the class label or regression value of new data points.

### 1.3.3.4 Random Forest (RF)

RF [28] is an ensemble learning technique applied to regression and classification tasks. A random subset of data is used to train multiple DTs using the bagging technique. To generate the final decision of the RF, the outputs from each DT in the ensemble are combined [15]. Consequently, compared to a single DT-based model, the RF learning model with multiple DTs is usually more accurate [162]. The workflow of RF algorithm is presented as follows:

- *Bootstrap sampling*: The training data is divided into a predefined number of subsets called bootstrap samples, with each subset used to train a DT.
- *Random feature selection (Feature randomness)*: The selected features available for the training are split into random subsets.
- *DT building / bagging*: Each subset of data is trained using a subset of features in each decision node to build a specific DT.
- *RF prediction*: New data points are predicted using all the built DTs. In classification tasks, the class label is determined using a voting technique, while in regression tasks, the mean of all DTs' predictions is calculated as the result of the RF.

### 1.3.3.5 K-means

K-means [109] is a common algorithm for cluster analysis in unsupervised machine learning. Its objective is to divide a set of data points into K clusters, where each data point is assigned to the cluster with the closest mean, acting as a representative of the cluster. The center of a cluster is defined by the mean of the observations within that cluster [15]. The K-means algorithm's execution depends on the parameter K selection. It determines how the algorithm operates, and different values for K can produce different clustering results [9]. The steps of the K-means algorithm are as follows:

- *K initialization*: The initial step in the K-means algorithm involves randomly assigning the value of K, which represents the number of clusters to be created from the training dataset. This entails determining the centroids, or cluster centers, for the initial clusters.
- *Cluster assignment*: Every data point from the training dataset is assigned to the nearest cluster, which means the cluster with the smallest distance between this data point and the centroid of that cluster. This distance is calculated using one of the following distance metrics: Euclidean distance, Manhattan distance, Chebyshev distance, or Mahalanobis distance.
- *Cluster update*: The centroids are recalculated by calculating the mean of all the data points in each cluster following the assignment of each dataset point to its corresponding cluster. The clusters are then updated using these new centroids.
- *Repetition of steps 2 and 3*: Until the centroids stabilize, indicating that the model has converged and the clusters won't change anymore, the two previous steps are repeated.
- *Prediction*: Each new data point will be assigned to a cluster by measuring the distance between it and the cluster centroids. The data point will be assigned to the nearest cluster.

### 1.3.3.6 Fuzzy C-Means (FCM)

FCM [46] is unsupervised clustering algorithm in d machine learning. This algorithm assigns each data point to several clusters with a membership degree, which is calculated based on the distance between each cluster center (centroids) and that data point [54]. The FCM algorithm objective is to minimize the following objective function  $J$ :

$$J = \sum_{i=1}^n \sum_{j=1}^K (u_{ij})^m \|x_i - C_j\|^2 \quad (1.1)$$

Where  $n$  is the number of the data points,  $K$  is the number of clusters,  $x_i$  is the  $i$ -th data point,  $C_j$  is the  $j$ -th cluster,  $u_{ij}$  is the membership degree of the data point  $x_i$  in the cluster  $C_j$ , and  $m$  is the fuzzy index ( $m > 1$ ).

The FCM algorithm consists of the following steps:

- *K initialization*: The initial step in the FCM algorithm is randomly assigning the number of clusters  $K$ . This step includes determining the centroids, or cluster centers, for the initial clusters.
- *Clusters membership calculation*: For each data point  $x_i$ , calculate their membership in all the cluster  $C$  using the following equation:

$$u_{ij} = \left( \sum_{k=1}^c \left( \frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (1.2)$$

Where  $u_{ij}$  is the membership of the data point  $x_i$  in the cluster  $C_j$ .

- *Cluster centroids update*: The cluster centroids are recalculated using the following equation:

$$v_j = \frac{\sum_{i=1}^n u_{ij}^m \cdot x_i}{\sum_{i=1}^n u_{ij}^m} \quad (1.3)$$

- *Repetition of steps 2 and 3*: Until the model has converged or the number of iterations is reached.

### 1.3.3.7 Naïve Bayes (NB)

NB is probabilistic supervised machine learning, It is based on the "Bayes" theorem, a principle of conditional probability, and operates under the "Naive" assumption, which assumes conditional independence between each pair of features or attributes [143]. It performs well and can be applied to many real-world scenarios, including document or text classification, spam filtering, and both binary and multi-class categories [160]. The key concepts of the NB algorithm are:

- *Bayes theorem*: The posterior probability of a class  $A$  given feature values  $B$  is calculated using the prior probability of class  $A$  given feature  $B$ , according to Bayes' theorem as follow:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (1.4)$$

Where  $P(A|B)$  is the posterior probability of class  $A$  given feature  $B$ ,  $P(B|A)$  is the probability of observing feature  $B$  given class  $A$ ,  $P(A)$  and  $P(B)$  are the prior probability of class  $A$ , and feature  $B$ .

- *Naive assumption*: The Naive assumption allows the calculation of the probability of class  $A$  given a set of features  $(B_1, B_2, \dots, B_n)$  as follows:

$$P(A | B_1, B_2, \dots, B_n) = P(A) \prod_{i=1}^n P(B_i | A) \quad (1.5)$$

Where  $P(A)$  is the probability of class  $A$ , and  $P(B_i | A)$  is the probability of feature  $B_i$  given class  $A$ .

- *Prediction*: The posterior maximum rule is used to predict the new data points  $x_{\text{new}}$  class:

$$\text{MAP} = \operatorname{argmax}_{A_k \in A} (P(A_k | x_{\text{new}})) \quad (1.6)$$

### 1.3.3.8 Logistic Regression (LR)

LR [30] is a probabilistic supervised machine learning model. Contrary to what its name suggests, it is a classification model used for binary and linear classification. It demonstrates exceptional performance with linearly separable classes [171]. The logistic function (Sigmoid function) is used in LR to estimate the probabilities [160]. The key concepts of the LR algorithm are:

- *linear combination*: Before calculating the probability using the logical function, the linear combination  $z$  of the features (independent variables) is calculated as follows:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (1.7)$$

Where  $\beta_0$  is the intercept term,  $\beta_1, \beta_2, \dots, \beta_n$  are the coefficients for the respective features  $x_1, x_2, \dots, x_n$ .

- *logistic function (Sigmoid function)*: The Sigmoid function  $g$  maps the real-valued result of the linear combination to the range  $[0, 1]$ , which means it gives the probability of the dependent variable  $z$  given the features  $x_1, x_2, \dots, x_n$ .

$$g(z) = \frac{1}{1 + \exp(-z)} \quad (1.8)$$

- *Prediction*: To classify a new data point, the probability of this data point based on its features is calculated using a logistic function. The class label is determined using a threshold (generally 0.5).

### 1.3.3.9 Gaussian Mixture Model (GMM)

A GMM is probabilistic model for cluster analysis in unsupervised machine learning. GMM assumes that data point are generated from a mixture of several Gaussian distributions [52, 134]. GMM are built based on two key concept, Gaussian distribution and mixture model. A Gaussian distribution or normal distribution is a continuous probability distribution with bell-shaped curve. The mixture model is a probabilistic model that assumes data generation from a mixture of several distributions. The mathematical representation of the GMM is as follows:

$$P(X) = \sum_{i=1}^K \omega_i \eta(X, \mu_i, \Sigma_i) \quad (1.9)$$

Where  $K$  represents the number of Gaussians,  $\omega_i$  denotes the mixture weight,  $\mu_i$  represents the average, and  $\Sigma_i$  indicates the covariance matrix, each of which is evaluated for the  $i$ -th Gaussian.  $\eta$  denotes the Gaussian probability density function, which is presented as follows:

$$\eta(X, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)} \quad (1.10)$$

Where  $D$  expresses the dimension of the vector  $X$ .

The GMM consists of the following phases:

- *Initialization*: First, determine the value of  $K$ , the hyperparameter representing the number of clusters or Gaussian components, and then initialize the parameters of each Gaussian  $\omega$ ,  $\mu$ , and  $\Sigma$  either randomly or using one of several initialization methods.
- *Parameters update*: There are several techniques available for estimating GMM parameters. The most popular and commonly used method is the Expectation-Maximization (EM) algorithm. The concept behind the EM algorithm is to iteratively estimate the parameters to maximize the posterior probability for each data point given each Gaussian component. The EM algorithm consists of two main steps:
  - *Expectation step (E-step)*: Calculating the posterior probability for each data point  $X_t$  in each cluster  $i$  as follows:

$$\gamma_{ti} = \frac{\omega_i \eta(X_t, \mu_i, \Sigma_i)}{\sum_{k=1}^K \omega_k \eta(X_t, \mu_k, \Sigma_k)} \quad (1.11)$$

- *Maximization step (M-step)*: Update the Gaussian parameters to maximize the posterior probabilities. using the following equations:

$$\bar{\omega}_i = \frac{1}{T} \sum_{t=1}^T \gamma_{ti} \quad (1.12)$$

$$\bar{\mu}_i = \frac{\sum_{t=1}^T \gamma_{ti} X_t}{\sum_{t=1}^T \gamma_{ti}} \quad (1.13)$$

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^T \gamma_{ti} X_t^2}{\sum_{t=1}^T \gamma_{ti}} - \bar{\mu}_i^2 \quad (1.14)$$

Where  $T$  present the number of the data points in the dataset.

The steps of the parameter update phase (E-step and M-step) are repeated until the model converges and the parameters stabilize.

## 1.4 Deep learning

### 1.4.1 Introduction to deep learning

The term "deep learning" was coined by Rina Dechter in 1986 in her paper titled "Learning while searching in constraint-satisfaction-problems" [40]. The paper discusses the concept of learning while searching in constraint-satisfaction problems. While "deep learning" gained popularity in the mid-2000s, it was defined as neural networks with multiple layers. Over the years, several definitions of deep learning have been proposed. Below are some of the most cited definitions in machine learning research:

Geoffrey Hinton (2007):

*"Deep learning is a type of machine learning in which a model learns to perform classification tasks directly from images, text, or sound."* (Geoffrey Hinton, [63])

LeCun Yann, Bengio Yoshua, and Hinton Geoffrey (2015):

*"Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection, and many other domains such as drug discovery and genomics."* (LeCun Yann, Bengio Yoshua, and Hinton Geoffrey, [93])

Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016):

*"Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts and representations, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones"* (Ian Goodfellow, Yoshua Bengio, and Aaron Courville, [56])

Francois Chollet (2018):

*"Deep learning is a specific subfield of machine learning: a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations."* (Francois Chollet, [34])

John D Kelleher (2019):

*"Deep learning is the subfield of artificial intelligence that focuses on creating large neural network models that are capable of making accurate data-driven decisions."* (John D Kelleher, [81])

M Arif Wani, Farooq Ahmad Bhat, Saduf Afzal, and Asif Iqbal Khan (2020):

*"Deep learning refers to the architectures which contain multiple hidden layers (deep networks) to learn different features with multiple levels of abstraction. Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher level learned features defined in terms of lower level features."* (M Arif Wani, Farooq Ahmad Bhat, Saduf Afzal, and Asif Iqbal Khan, [186])

Based on all previously cited definitions of deep learning, we can propose the following summarized definition:

"Deep learning is a subset of machine learning and AI, aims to simulate the intricate workings of human neural networks. It focuses on training deep neural networks to learn and automatically extract distinct characteristics and features from datasets. This enables deep learning models to perform predictions, make decisions, and recognize patterns based on the learned experience."

## 1.4.2 Deep learning basics

### 1.4.2.1 Artificial Neural Network (ANN)

ANN is computational technique employed in machine learning. They serve as effective tools for showcasing acquired knowledge and leveraging it to enhance the output responses of complex systems [33]. ANN aim to simulate the neural system found in the human brain. It emulates the brain's process of analyzing data and information to make decisions through the utilization of a mathematical model [37, 88]. Through this simulation, neural networks allow machines to identify patterns and make decisions in a way that resembles human thought processes. Similar to the human brain's composition of neural networks, ANNs also comprise network of artificial neurons (nodes). Below, we elucidate the fundamental principles and mechanics of ANNs.

- **Artificial neuron:** An artificial neuron serves as the fundamental unit within every ANN, functioning as a basic mathematical model that emulates biological neurons. The biological neuron receives data through its dendrites, processes it within the soma, and transmits output through its axon. The artificial neuron receives various data inputs, such as data points, weights, and biases. It processes these inputs using two primary functions: a summation function and a transfer function, also known as an activation function. Ultimately, the neuron generates output(s) according to the outcomes of these computational steps [88]. Figure 1.5 gives an illustration of the structure of biological and artificial neurons. An artificial neuron can be represented mathematically as follows:

$$Y = f \left( \sum_{i=1}^n w_i x_i + b \right) \quad (1.15)$$

Where  $Y$  is the output,  $X$  are the inputs,  $w$  are the weights,  $b$  is the bias, and  $f()$  is the transfer function (activation function).

Weights determine the significance of input features by assigning them different degrees of influence on the network's output. Bias enables the system to adapt to variations and shifts within the input data, facilitating the learning of more complex patterns. The activation function introduces non-linear behavior to a neuron's output, thereby determining whether the neuron is activated or not, essentially deciding whether it produces an output or remains inactive. The commonly used activation functions are Sigmoid, Hyperbolic tangent (tanh), REctified Linear Unit (ReLU), and Softmax. The mathematical equations for each of these functions are as follows:

- Sigmoid function:

$$0 < \sigma(x) = \frac{1}{1 + e^{-x}} < 1 \quad (1.16)$$

- Hyperbolic tangent (tanh) function:

$$-1 < \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} < 1 \quad (1.17)$$

- Rectified Linear Unit (ReLU) function:

$$\text{ReLU}(x) = \max(0, x) \quad (1.18)$$

- Softmax function:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (1.19)$$

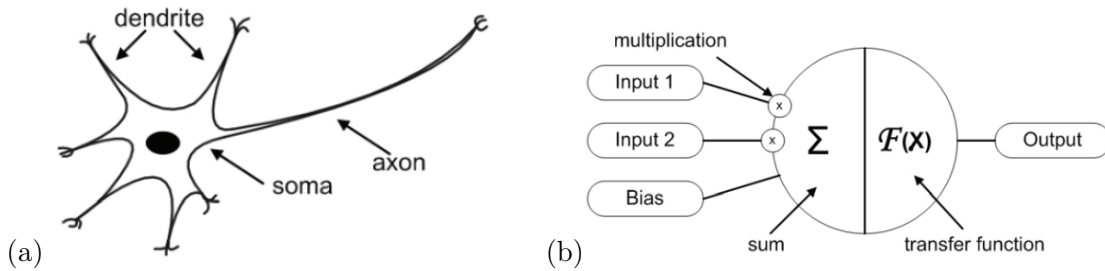


FIGURE 1.5: Neuron structure: (a) biological neuron, (b) artificial neuron (from [88]).

- **ANN structure:** An ANN is composed of numerous neurons interconnected through links, with each neuron possessing its own bias and specialized activation function. These connections between neurons are characterized by weights. Neurons are organized into layers, with the primary types being the input layer, hidden layer, and output layer. The input layer comprises neurons (nodes) that receive input data and transmit it to the hidden layer. The hidden layer is the intermediary layer responsible for processing data points and extracting relevant features from them. The output layer represents the final stage of an ANN, where it generates the predicted output. Its primary role is to produce the output based on the transformations carried out by the hidden layer. Figure 1.6 illustrates the structure of an ANN.

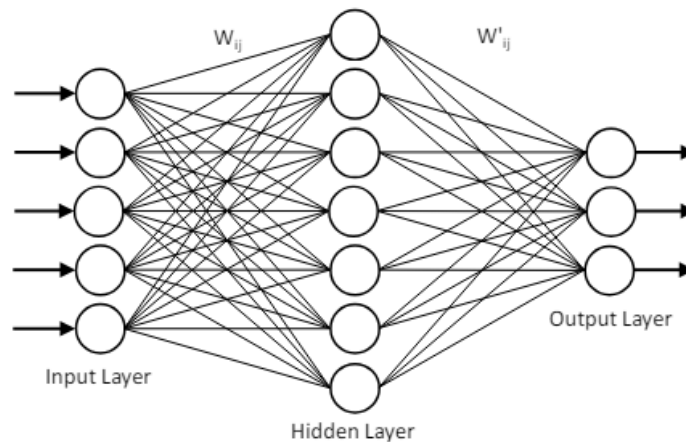


FIGURE 1.6: A structure of an ANN (from [38]).

In the context of ANNs based on the topology of connection, feedforward and feedback architectures are two fundamental types of network structures; they are as follows:

- **Feedforward architecture:** ANNs are classified as feedforward when information flows in one direction, from the input layer through the hidden layer to the output



layer. This directional flow excludes feedback loops, resulting in the absence of memory to preserve previous results and network states. Commonly, feedforward neural networks find applications in image classification, object recognition, and various regression problems. The architecture of feedforward ANN is shown in Figure 1.7a.

- **Feedback architecture:** ANN is classified as feedback, also known as Recurrent Neural Network (RNN), where information flows backward through connections between the network’s neurons, creating loops between them. RNNs possess memory capabilities that store both previous and next states, enabling the modeling of dynamic temporal behavior. Common applications include language modeling, speech recognition, time series prediction, and machine translation. The architecture of feedback ANN is shown in Figure 1.7b.

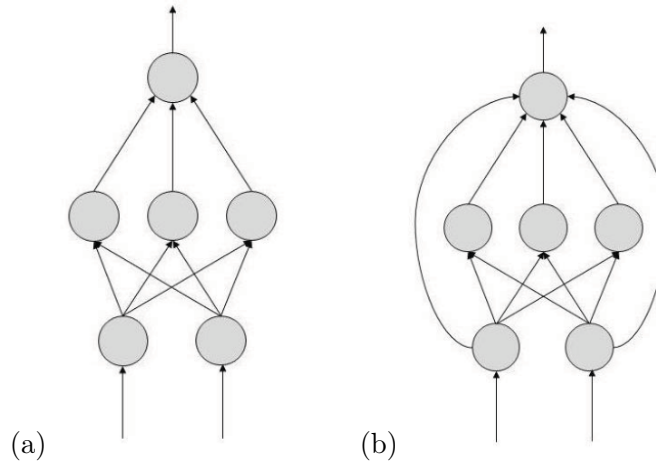


FIGURE 1.7: ANN architectures: (a) feedforward architecture, (b) feedback architecture (from [37]).

- **ANN learning:** There are several types of learning algorithms for ANNs. The common ones are the previously mentioned types: supervised, unsupervised, semi-supervised, and reinforcement learning. The common method used to train ANNs is back propagation, a learning strategy that reduces the error between the expected and actual outputs by adapting the weights and biases of the network. This consists of the following steps:
  - *Model initialization:* The weights and biases are initialized either randomly or using one of several initialization methods.
  - *Forward pass:* The dataset, or a subset of it known as a batch, passes through the ANN. This technique involves splitting the dataset into smaller sets, and the ANN is trained on each batch. This technique enhances the performance of the ANN and makes the training faster. During the forward pass, the data propagates through all the layers to calculate the outputs. Each neuron processes the data point using two primary functions: a summation function and an activation function. This process is repeated for all neurons until reaching the last neuron of the output layer, where the final output is calculated.
  - *Error calculation:* Error is calculated using the loss function, which estimates the difference between the actual output values and the predicted values. There are several loss functions based on the learning type. The commonly used ones are: Cross-Entropy Loss, Binary Cross-Entropy Loss, Categorical Cross-Entropy Loss, Mean Squared Error, Mean Absolute Error, and Huber Loss.

- *Backward pass (backward propagation)*: The gradients, which represent the loss values of each weight and bias, are propagated backward through the ANN layers to adjust and update the weights and bias for minimizing the loss function. This process is performed using optimization algorithms such as Gradient Descent, Stochastic Gradient Descent, or Adam.
- *Iteration*: The process repeats from the forward pass to the backward pass until the model converges and the error is not further minimized, or until the stopping criteria are achieved. These criteria can be defined by a predetermined number of epochs, a certain number of times passing the whole dataset through the ANN, or by using techniques such as early stopping to prevent overfitting.

In RNNs, Back Propagation Through Time (BPTT) is used for learning. This technique involves the forward pass, where data points propagate through the network and outputs are calculated at each time step. Then, during the backward pass, gradients are calculated at each time step, propagating backward through time.

### 1.4.2.2 Deep learning network

Deep learning network or Deep Neural Network (DNN) describe ANN that has multiple layers [2]. In other words, a deeper ANN consists of multiple hidden layers, facilitating the acquisition of data representations with diverse levels of abstraction [93]. Deep learning networks have proven highly effective in analyzing large datasets, demonstrating notable success in fields such as speech recognition, computer vision, pattern recognition, recommendation systems, and natural language processing [105]. The architecture of a DNN is shown in Figure 1.8.

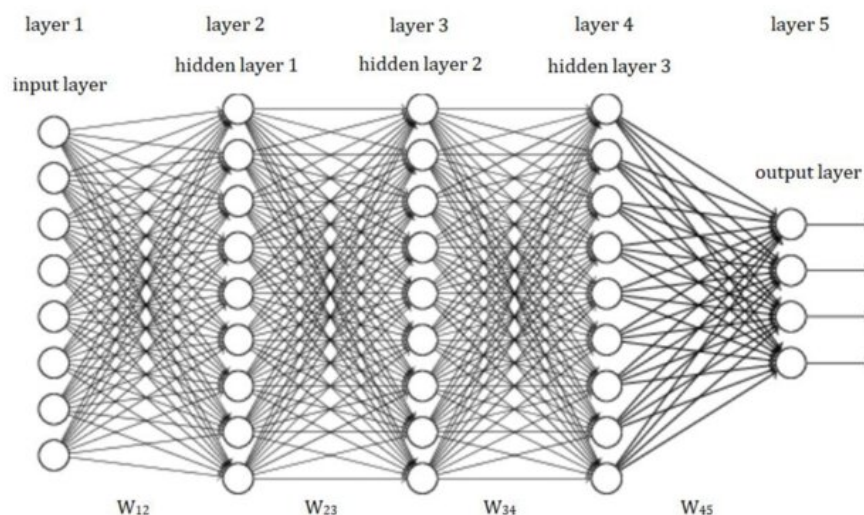


FIGURE 1.8: A structure of a DNN (from [37]).

## 1.4.3 Popular deep learning networks

### 1.4.3.1 Convolutional Neural Network (CNN)

CNNs [94] are a type of feedforward ANN that excel at extracting complex and hierarchical features. They find extensive use in image recognition and video processing due to their ability

to preserve spatial relationships between pixels in an image. This entails processing groups of pixels, termed patches, together. Each patch is directed to a specific node in the next layer, thereby retaining the patch's original location. This differs from traditional ANNs, which process individual pixels separately and do not preserve spatial relationships [64, 89, 93]. A CNN comprises multiple deep layers, with initial layers focusing on extracting primary features such as edges and corners, while subsequent layers specialize in extracting more intricate features and recognizing objects. The main layers in a CNN are the convolutional layer, pooling layer, and fully connected layer, as illustrated in Figure 1.9.

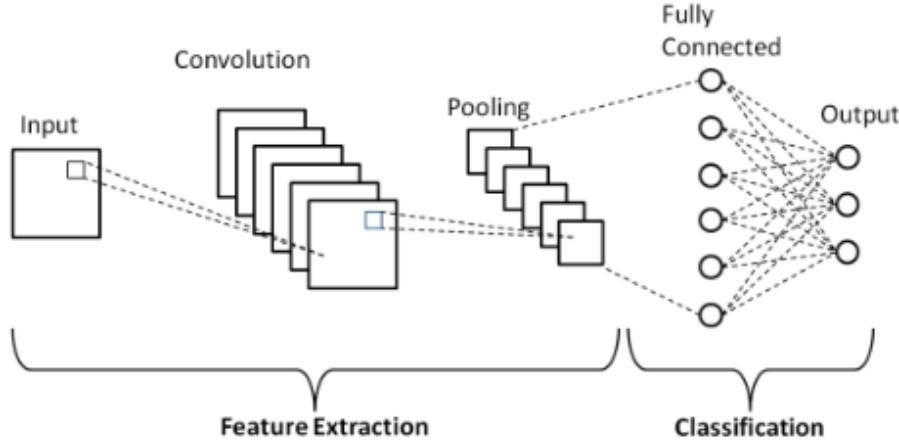


FIGURE 1.9: A structure of a CNN (from [136]).

The details of the CNN layers are as follows:

- **Convolutional layer:** It serves as the core component of the CNN and is responsible for feature extraction using kernels or filters, which are small weight matrices. Each filter is designed to detect a specific feature within the image. Convolutional layers convolve filters over the image with a step size known as stride. At each position, the filter weights are multiplied with the pixel values of the image using element-wise matrix multiplication, resulting in a new stacked matrix known as the feature map. This map demonstrates the presence of specific features at particular points in the image, as described by the following equation:

$$\gamma_{ij} = f\left(\sum_{m=0}^{h-1} \sum_{n=0}^{w-1} I(i+m, j+n) \cdot F(m, n)\right) \quad (1.20)$$

Where  $I$  is the input image with dimensions  $H \times W$ ,  $F$  is the filter with dimensions  $h \times w$ ,  $f()$  is the activation function, and  $\gamma_{ij}$  is the resulting feature map at position (pixel index)  $(i, j)$  with dimensions  $(H - h + 1) \times (W - w + 1)$ .

The process of convolution operation is shown in Figure 1.10.

- **Pooling layer:** The pooling layer downsamples the feature map to reduce its dimension, retaining only critical information. The pooling layer moves across the feature map using a kernel with a defined stride size ( $s$ ), called a pooling window, much like the convolutional layer. There are two types of pooling operations: max pooling and average pooling. In max pooling, the operation computes the maximum value within each window, representing the activated feature in that region (see Figure 1.11a, and equation 1.21). Conversely, average

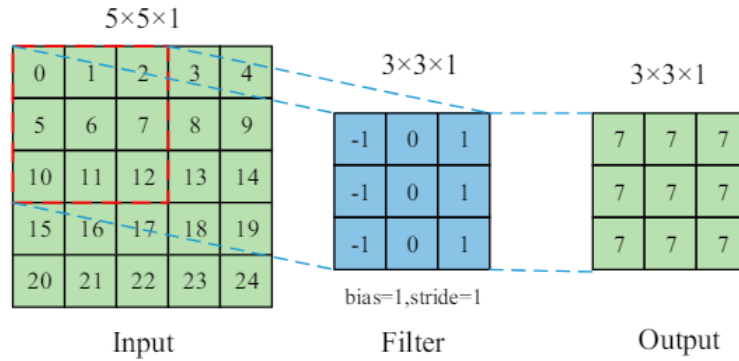


FIGURE 1.10: A convolution operation process (from [107]).

pooling calculates the average value of the window's contents, representing the combined features within that region (see Figure 1.11b, and equation 1.22).

$$\gamma_{ij} = \max_{m,n} (I(i \cdot s + m, j \cdot s + n)) \quad (1.21)$$

$$\gamma_{ij} = k \times k \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} I(i \cdot s + m, j \cdot s + n) \quad (1.22)$$

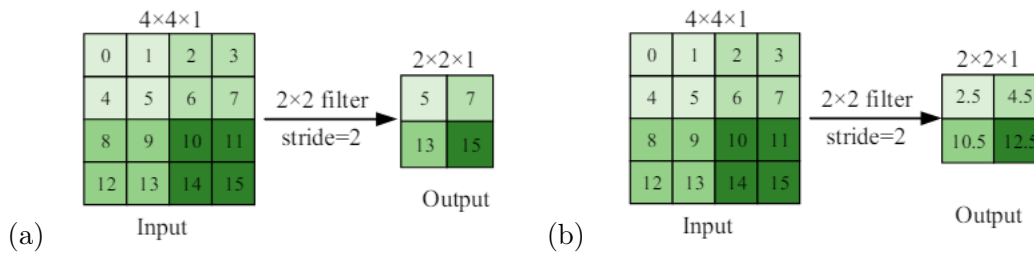


FIGURE 1.11: Pooling operations: (a) max pooling operation, (b) average pooling operation (from [107]).

- **Fully connected layer:** After traversing through numerous convolutional and pooling layers of CNNs, the fully connected layer emerges, serving as the final decision-maker based on the feature maps extracted from the preceding layers. In this layer, each node is intricately connected to every node of the previous layer, facilitating the simultaneous consideration of all features. This comprehensive connectivity enables the extraction of complex relationships among the features.

CNNs are widely used in computer vision tasks, and several architectures have been proposed over the years for tasks such as classification, object detection, and segmentation. Some of the most famous CNN architectures include: LeNet-5 [94], AlexNet [89], Inception [173], VGGNet [167], ResNet [62], Faster R-CNN [147], Mask R-CNN [61], etc.

### 1.4.3.2 Generative Adversarial Network (GAN)

A GAN [57] takes a set of noise inputs and generates new data samples that resemble the training dataset by learning the underlying probability distribution of the provided data. GAN composed

from two ANN, generator and discriminator. In a non-cooperative zero-sum game, both networks are trained simultaneously until the discriminator can no longer distinguish between real and generated samples (see Figure 1.12). The gain of one network corresponds to the loss of the other [74]. The following are the key concepts of GANs:

- **Generator network:** The generator ( $G$ ) in a GAN is an ANN that produces new data similar to the existing data in the training set from a set of noise. The aim is to make it difficult to differentiate between the newly produced data and the real data. Its goal is to reduce the probability of the discriminator correctly identifying the generated data as fake. The mapping from noise ( $z$ ) to new generated data samples ( $\hat{x}$ ) is represented by the following mathematical equation:

$$\hat{x} = G(z) \quad (1.23)$$

- **Discriminator network:** The discriminator ( $D$ ) is an ANN tasked with distinguishing between real data and data generated by the generator. It receives both real and generated data as inputs, classifying them as either real or fake. Its goal is to minimize the classification error between real and fake data. The mapping of the data samples to a probability is represented by the following mathematical equation:

$$D(x) = f(Wx + b) \quad (1.24)$$

Where  $x$  represents the inputs,  $W$  denotes the weight,  $b$  stands for the bias, and  $f()$  is the activation function.

- **Adversarial loss function:** During the training of the GAN, the objective is to minimize the adversarial loss function, which quantifies the discrepancy between the generated data and the real data. This function comprises both the generator loss and discriminator loss, as follows:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (1.25)$$

$x$  represents real samples,  $z$  represents random noise input to the generator,  $G(z)$  are generated samples from the generator.  $D(x)$  is the calculated probability distribution of real data from the discriminator, and  $D(G(z))$  is the calculated probability distribution of fake data from the discriminator.  $p_{\text{data}}(x)$  is the true data distribution, and  $p_z(z)$  is the noise distribution.

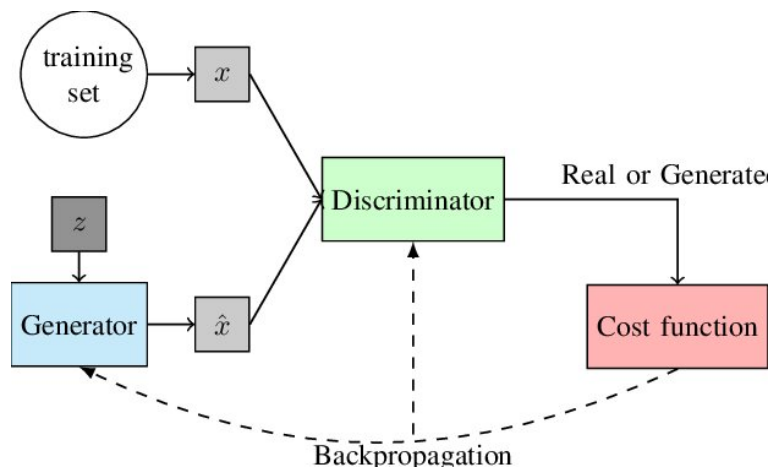


FIGURE 1.12: A GAN structure (from [140]).

### 1.4.3.3 Autoencoders (AE)

An AE is a feedforward ANN for unsupervised learning tasks. The network is driven to ignore unnecessary noise and retain important information from the input data [56]. It encodes the data in the first stage and then, based on the obtained encoded data, it tries to reconstruct the original copy of the data. It is primarily used for anomaly detection because of its capacity to measure reconstruction errors, which are thought to be substantially greater for anomalous instances than for typical cases [132]. The AE consists of three components: the encoder, the code, and the decoder, as illustrated in Figure 1.13. The details of these components and the key concepts of AE are as follows:

- **Encoder:** The encoder in an autoencoder is a neural network responsible for encoding the input data into a code or latent space representation (a lower-dimensional representation) to reduce the dimensionality of the data and extract only the important features, thus removing noise. This process is mathematically represented as follows:

$$Y = f_e(W_e x + b_e) \quad (1.26)$$

Where  $x \in \mathbb{R}^n$  is the input data,  $y \in \mathbb{R}^{n'}$  is the latent representation,  $f_e()$  is the encoder activation function,  $W_e$  is the encoder weight matrix, and  $b_e$  is the encoder bias.

- **Decoder:** The decoder in an autoencoder is a neural network responsible for decoding the code or latent space representation and reconstructing data that resembles the input data as closely as possible. This process is mathematically represented as follows:

$$\hat{x} = f_d(W_d y + b_d) \quad (1.27)$$

Where  $\hat{x} \in \mathbb{R}^n$  is the reconstructed input data,  $y \in \mathbb{R}^{n'}$  is the latent representation,  $f_d()$  is the decoder activation function,  $W_d$  is the decoder weight matrix, and  $b_d$  is the decoder bias.

- **Reconstruction loss:** The objective of the AE during the training process is to minimize the error between the original data and the reconstructed data, called the reconstruction error. This minimization is achieved using the following objective function:

$$\min_{W_e, b_e, W_d, b_d} L(X, \hat{x}) \quad (1.28)$$

$L()$  is the loss function, commonly used in AE is the Mean Squared Error (MSE) loss function.

### 1.4.3.4 Transformer

A transformer [181] is an ANN that recognizes contextual relationships within sequential data using a self-attention mechanism, enabling parallel processing [72]. It is primarily applied in natural language processing and is now used across various domains where data can be represented as sequences, such as computer vision and speech recognition. The main mechanisms used in transformers and their basic architecture are presented as follows:

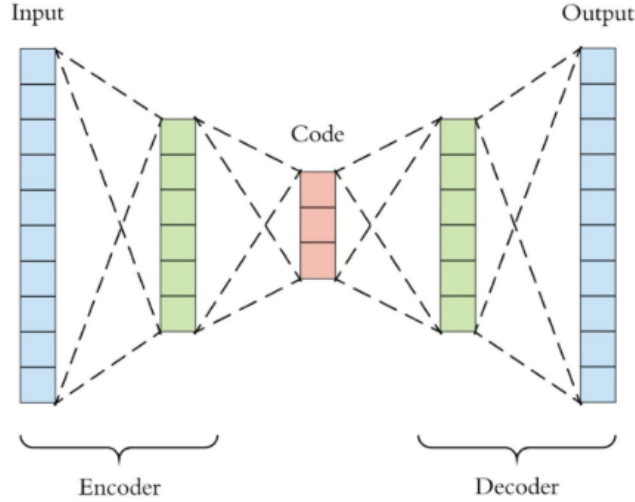


FIGURE 1.13: An AE structure (from [11]).

- **Input embedding:** Using the embedding matrix, which is either initially randomly initialized and then updated during the training process or pre-trained, to convert each token in the input sequence into a numerical representation (vector representation) is known as token embedding.
- **Positional encodings:** Adding positional information of each token in the sequence as a continuous vector to the input embedding. Token embedding and positional encoding are summed to produce the final input embedding. This ensures that both the meaning of the tokens and their placement in the sequence are incorporated into the input embedding.
- **Attention mechanism:** The attention mechanism involves analyzing the relationships between tokens in the sequence and their connections, where each token is given weight based on its relationship with the other tokens in the sequence. This mechanism is implemented through the Scaled Dot Product Attention, which is composed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1.29)$$

Where  $Q$ ,  $K$ , and  $V$  are linear transformations of the input embedding, each of them is presented as follows:

- $Query(Q)$  is the vector that represents the current token used to calculate the attention weight.
- $key(K)$  is the vector that represents tokens in the sequence which will be compared to the query.
- $value(V)$  is the vector that provides information about the token.

These values are calculated as follows:

$$Q = X \cdot W_q \quad (1.30)$$

$$K = X \cdot W_k \quad (1.31)$$

$$V = X \cdot W_v \quad (1.32)$$



Where  $X$  is the input embedding,  $w_q$ ,  $w_k$ , and  $w_v$  are learned weight matrices that transform input query, key, and value vectors respectively in transformer models.

- **Multi-head attention:** The mechanism enables the model to attend to various parts of the input sequence simultaneously and assess their significance. This is accomplished through multiple attention heads, each computing a distinct attention distribution as follow:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (1.33)$$

The output of the multi-head attention mechanism is obtained as follow:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O \quad (1.34)$$

Where  $h$  is the number of attention heads, and  $w_0$  is the output weight matrix used to linearly transform the outputs.

- **Transformer architecture:** The Transformer is composed of two primary modules, the encoder and the decoder, as shown in Figure 1.14. The encoder module, which is composed of a multi-head attention layer and a feedforward neural network, is used to encode the information in the input sequence by capturing the relationships between the tokens in the sequence. The decoder module is composed of a masked multi-head attention layer, a multi-head attention layer, and a feedforward network. It is used to decode the representation encoded by the encoder and generate the output sentence by capturing the relationships between the tokens from the perspective of the encoder presentation.

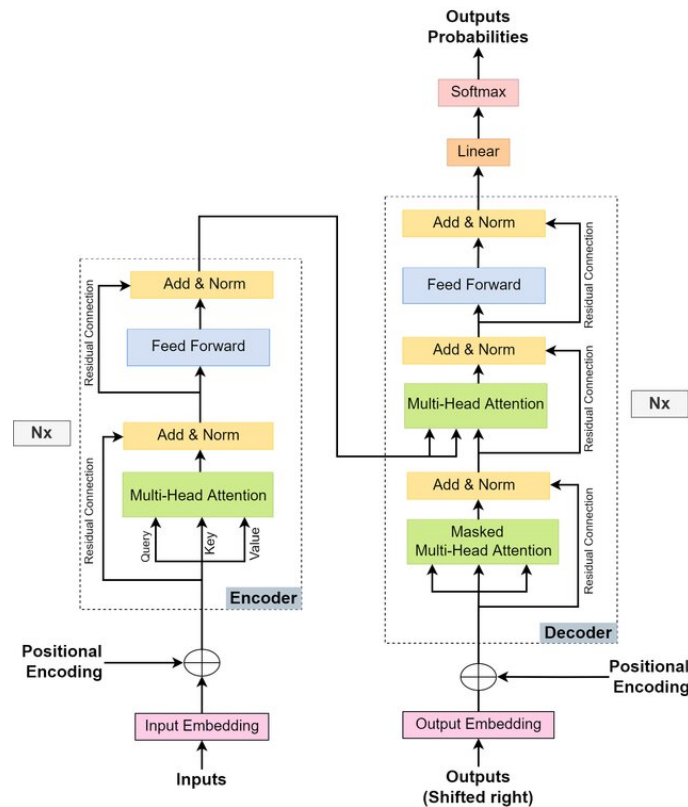


FIGURE 1.14: A transformer structure (from [72]).



### 1.4.3.5 Long Short Term Memory Network (LSTM)

LSTM [66] is a type of RNN that has the ability to extract long-term relationships in data that is in the form of a series by using a memory cell. Its main goal is to eliminate the problem found in traditional RNNs, represented by vanishing gradients, by allowing gradients to flow through unchanged [166]. This issue arises when the gradients that are backpropagated through time to update the network weights get smaller, making it harder for the network to learn long-term dependencies. LSTMs have been extensively employed in a variety of tasks, including time series prediction, speech recognition, and natural language processing. As presented in Figure 1.15, the LSTM is composed of memory cells and gates. These components are used to selectively retain and update information over time as follows:

- **Memory cell:** The memory cell is the key component of the LSTM that allows it to memorize long-term information. This component has two states:

- **Cell state** ( $C_t$ ): Is the long-term memory of the network that store the information over time steps. The cell state is updates as follow:

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \sigma_c(W_c x_t + W_h h_{t-1} + b_c) \quad (1.35)$$

- **Hidden state** ( $h_t$ ): The short-term memory of the network stores the information or output of the LSTMs at this time step and needs to be forwarded to the next time step. The hidden state is updates as follow:

$$h_t = o_t \otimes \tanh(c_t) \quad (1.36)$$

- **Gates:** The gates are the components responsible for controlling the flow of information in an LSTM. There are three types of gates:

- **Input gate** ( $i_t$ ): Determines the information that must be stored in the cell state. The following equation provides the input gate:

$$i_t = \sigma(W_i x_t + W_i h_{t-1} + b_i) \quad (1.37)$$

- **Output gate** ( $o_t$ ): Determines the information that must be outputted from the cell state. The following equation provides the output gate:

$$o_t = \sigma(W_o x_t + W_o h_{t-1} + b_o) \quad (1.38)$$

- **Forget gate** ( $f_t$ ): Determines the information that must be forgotten from the cell state. The following equation provides the forget gate:

$$f_t = \sigma(W_f x_t + W_f h_{t-1} + b_f) \quad (1.39)$$

Where  $x_t$  is the input at time step  $t$ ,  $h_{t-1}$  is the previous hidden state,  $\sigma$  is the sigmoid function,  $\tanh$  is the hyperbolic tangent function,  $\otimes$  is the element-wise multiplication,  $W$  are the input weight matrices,  $w$  is the recurrent output weight matrix, and  $b$  are the biases.

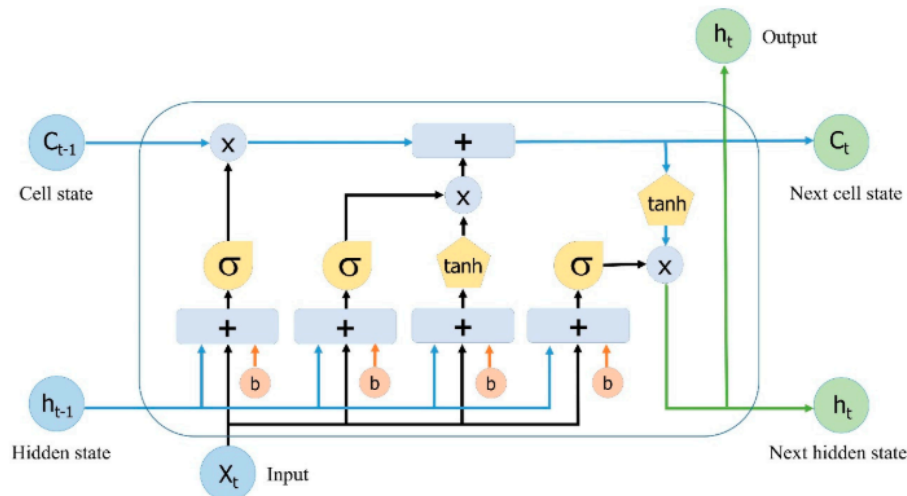


FIGURE 1.15: A LSTM structure (from [92]).

## 1.5 Challenges and issues of machine learning and deep learning paradigms

Despite the promising results achieved by machine learning and deep learning approaches, making them hot topics in many applications, they still face numerous challenges and obstacles that hinder their optimal and effective utilization. The quality and quantity of data play pivotal roles in the outcomes of machine learning and deep learning. A sufficient amount of data gives the model a wide range of patterns and relationships, leading to a deeper comprehension of the underlying structures in the dataset and helping the model adapt adeptly to new unseen data. Furthermore, data quality is crucial; high-quality data promotes more accurate learning, facilitating the development of robust models. In contrast, poor-quality data can compromise model performance. As a result, the generalization and robustness of machine learning and deep learning models depend on both the quantity and quality of data. However, the process of collecting a dataset of optimal quantity and quality is challenging, necessitating significant investments of time, resources, and availability.

The second obstacle, feature extraction, presents a formidable challenge in both machine learning and deep learning approaches. This critical task, especially for machine learning, relies on human expertise, rendering it susceptible to errors and subjectivity. Furthermore, when dealing with extensive datasets, the complexity of feature extraction is amplified, necessitating considerable experience and careful selection of preprocessing techniques to ensure optimal results. Within deep learning, models autonomously learn to extract features during training. This further complicates the process, with the model's internal workings concealed within a black box framework. This lack of transparency not only makes interpretation more difficult but also hinders understanding of the variables influencing program outputs.

Another issue pertains to the computational resources required for computations. Training machine learning and deep learning models on extensive datasets demands substantial computing power, often necessitating the utilization of high-performance computing systems, graphics processing units, or cloud computing platforms. These resources come with significant costs. The complexity of selecting appropriate training parameters further adds to the challenges. This task necessitates conducting multiple experiments to determine the optimal configuration, demanding a wealth of experience and expertise. Moreover, achieving proficiency in parameter selection

requires a considerable investment of time, involving thorough experimentation and analysis to understand the intricate dynamics of the model's performance under varying conditions. Despite ongoing efforts to address these challenges and the development of numerous methods and theories aimed at overcoming them, they continue to pose obstacles to achieving optimal results in machine learning and deep learning.

## 1.6 Conclusion

In this chapter, we have addressed various aspects within the scope of our thesis. We began by discussing the concept of AI, briefly exploring its historical emergence from its inception to the present day. We provided an overview of AI's evolution and presented definitions of AI from renowned scientists. Finally, we proposed our own definition and showcased some AI applications. In the subsequent section, we delved into machine learning paradigms. Starting with the definition of machine learning according to scientific literature, we then proposed our own definition. We discussed different types of machine learning and highlighted famous machine learning algorithms. Similarly, we explored deep learning paradigms, discussing definitions, types, and notable algorithms. Finally, we addressed the challenges encountered in both machine learning and deep learning.

---

---

## CHAPTER 2

---

# PLANT DISEASE DETECTION WITH ARTIFICIAL INTELLIGENCE

---

## Chapter contents

---

2.1	Introduction . . . . .	38
2.2	Plant pathology . . . . .	39
2.3	Plant disease datasets . . . . .	45
2.4	Plant disease detection and classification techniques . . . . .	46
2.5	Challenges and issues . . . . .	61
2.6	Conclusion . . . . .	61

---

## 2.1 Introduction

Plant diseases are considered one of the biggest threats facing global food security, imperiling the livelihoods of millions of humans [26, 27]. These diseases destroy agricultural crops, resulting in significant losses in agricultural production. The Food and Agriculture Organization of the United Nations (FAO) reports that each year plant diseases cause 20% to 40% of crop yield losses. Consequently, these agricultural losses disrupt food supply chains and increase the risk of widespread food shortages on a global scale.

Plant diseases can arise from various factors, with pathogens and environmental changes being particularly significant [41]. Pathogens invade plants and disrupt their life cycles, leading to disease development. Furthermore, alterations in environmental conditions can create conducive environments for diseases. These diseases typically exhibit symptoms, which manifest as distinct visual patterns on the plant. These patterns serve as crucial diagnostic signs, facilitating the identification and management of plant diseases [99].

Detecting diseases early is vital for effectively managing them and preserving plant health. Timely detection helps minimize losses and protects crops from extensive damage. However, traditional disease detection methods, reliant mainly on human diagnosis, often prove inadequate due to limited expertise and time constraints [99]. Moreover, the frequency of inspections and data collection methods may be insufficient, necessitating a more reliable approach to diagnosis. To address this challenge, modern technologies have emerged as automated solutions for identifying plant diseases [146].

The automated solution was developed using two subcategories of artificial intelligence, deep learning and machine learning. Plant disease detection using machine learning and deep learning has become popular due to its promising outcomes in precisely identifying plant diseases from images. Numerous research works have suggested deep learning and machine learning-based techniques for the identification and diagnosis of plant diseases. However, deep learning-based methods have demonstrated superiority over machine learning-based approaches, particularly in image recognition tasks. This superiority is attributed to their reliance on automatic feature extraction rather than human feature selection [104]. Despite their effectiveness, deep learning models often face challenges due to the scarcity of datasets. While deep learning and machine learning techniques have revolutionized the plant disease detection domain, they still encounter obstacles that hinder their optimal utilization. This chapter provided an overview

of the application of deep learning and machine learning methods to plant disease detection. It discussed the various applications of machine learning and deep learning in automating disease detection processes. Additionally, the chapter delved into the current status of research in this field, offering insights into recent advancements, methodologies, and trends.

The chapter is structured as follows: Section 2.2 provides an overview of plant pathologies, including disease types and symptoms, along with illustrations of common agricultural diseases. Section 2.3 illustrated the most used dataset in the field of plant disease detection. Section 2.4 offers an overview of the latest advancements in the detection and classification of plant diseases using machine learning and deep learning techniques. Consequently, it offers a comprehensive analysis of the latest approaches and methods proposed in this field. In Section 2.5, an analysis and discussion of the challenges and issues faced by proposed plant disease identification and classification systems in the literature are presented.

## 2.2 Plant pathology

Throughout their life cycles, plants are vulnerable to various diseases. These diseases can affect plants at different stages of their life cycles, from germination to maturity and beyond. Several factors, as summarized by [183] and illustrated in Figure 2.1, can cause plant diseases, which can be grouped into two different types. Abiotic diseases are non-infectious conditions that impact plants due to physiological factors or alterations in environmental conditions. Nutrient deficiencies or toxicities, water-related disorders, temperature extremes, light-related issues, soil-related complications, and air pollution are among the primary contributors. These factors not only directly affect plant health but also create environments favorable for the development of diseases. Biotic diseases are those caused by infectious factors, known as pathogens, such as bacteria, fungi, viruses, mites, mold, protozoa, and nematodes. These pathogens infect plants and disrupt their normal growth and function. These diseases are typically spread from one plant to another through various means, such as wind, water, insects, animals, or soil. Biotic diseases pose a greater risk due to their transmissible nature, as they can spread from one plant to another, leading to severe damage in agricultural crops and natural ecosystems. In contrast, abiotic diseases, being non-transmissible, are primarily caused by environmental factors and are often preventable through proper management practices. Consequently, researchers have chosen to exclusively focus on biotic diseases, recognizing their significant impact on plant health and agricultural productivity.

Biotic disease infection results in numerous external and internal symptoms that manifest in the appearance of the plant, both externally and internally. External symptoms are often the first indicators of disease and may include visible signs such as wilting, discoloration, lesions, spots, or deformities on the leaves, stems, flowers, or fruits of diseased plants [99]. These external symptoms vary depending on the specific disease and its impact on the plant's physiology. It is noteworthy that the initial external symptoms typically manifest on the leaves of the plants in nearly all cases [86]. On the other hand, internal symptoms consist of tissue rotting, cellular structure degradation, or changes in physiological processes like water or nutrient uptake. These internal and external symptoms have the potential to impair a plant's regular operations and ultimately cause it to deteriorate or die.

The most prominent biotic plant diseases in the agricultural domain, such as fungal infections, bacterial diseases, and viral infections [17], are extensively discussed in the research literature and addressed in our thesis. These diseases, which have significant implications for agricultural productivity, are presented in detail in Table 2.1. The table offers detailed information

on the pathogens associated with each disease, as well as the symptoms observed in the leaves. Additionally, image examples for each disease are included to ensure comprehensive visual representation. It is noteworthy that all images utilized in the table are sourced from the PlantVillage dataset [70].

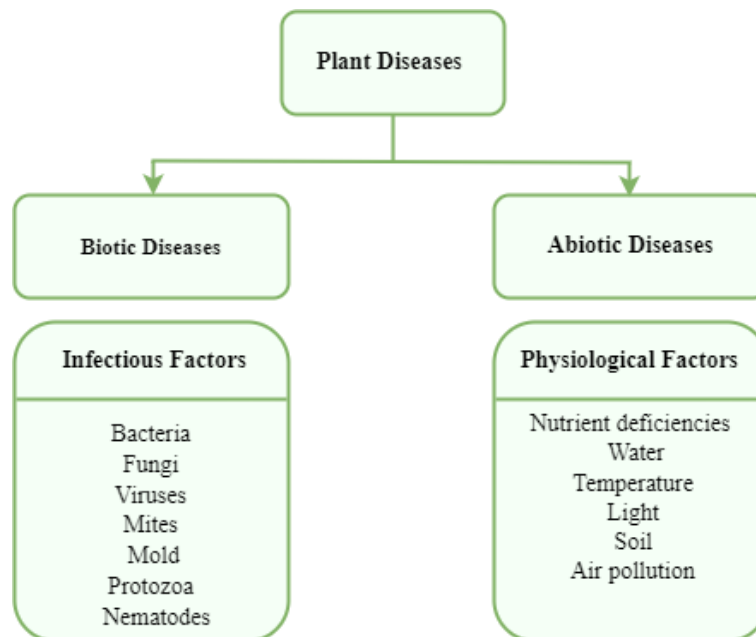


FIGURE 2.1: Types of plant diseases and their respective causal factors.

TABLE 2.1: Summary of the common plant diseases, their pathogens, and symptoms.



<i>Disease name</i>	<i>Pathogens</i>	<i>Disease Exemple</i>	<i>Disease symptoms</i>
Bacterial spot	Bacteria		<ul style="list-style-type: none"> <li>- Water-soaked spots: Small, water-soaked spots appear on the leaves, initially light green to yellowish, which may turn brownish-red or black;</li> <li>- Leaf yellowing: Surrounding the spots, affected leaf sections may turn yellow or chlorotic discoloration.</li> </ul>
Black rot	Fungi		<ul style="list-style-type: none"> <li>- Yellowing at the margins of the leaves: At first, the symptoms appear as leaf margin yellowing;</li> <li>- V-shaped spots: V-shaped spots emerge as the yellowing extends from the edges towards the middle of the leaf, resulting in spots resembling water-soaked areas that subsequently darken to brown or black;</li> <li>- Vein discoloration: The veins within the spots may become discolored, appearing dark brown or black.</li> </ul>

TABLE 2.1: Summary of the common plant diseases, their pathogens, and symptoms.





<i>Disease name</i>	<i>Pathogens</i>	<i>Disease Exemple</i>	<i>Disease symptoms</i>
Cedar apple rust	Fungi		<ul style="list-style-type: none"> <li>- Yellow spots: Small yellow spots initially appear on the leaves;</li> <li>- Orange spots: The yellow spots develop into orange-colored lesions with raised bumps or pustules as the disease progresses;</li> <li>- Leaf curling and distortion: Infected leaves may exhibit curling or distortion, particularly around the areas of lesion formation.</li> </ul>
Cercospora leaf spot gray leaf spot	Fungi		<ul style="list-style-type: none"> <li>- Gray to brown spots: Initially, small gray necrotic spots appear on the leaves. As the disease progresses, these spots enlarge and develop into larger rectangular lesions, turning brown in color;</li> <li>- Leaf yellowing: Leaf yellowing may occur around the affected spots. Furthermore, the impacted regions of the leaves might undergo necrosis, transitioning to a brown or black color before eventually desiccating.</li> </ul>
Common rust	Fungi		<ul style="list-style-type: none"> <li>- Orange to reddish-brown spots: Initially, small round to elongated spots appear on the leaf surface, which are orange to reddish-brown in color;</li> <li>- Leaf curling and yellowing: Infected leaves may exhibit curling or distortion, particularly around the areas where spots form. Additionally, the affected areas of the leaves may turn yellow.</li> </ul>
Early blight	Fungi		<ul style="list-style-type: none"> <li>- Circular to irregular spots: Spots appear on the leaves, initially presenting as small, dark, dry, papery flecks. These spots gradually expand to form brown-black, circular-to-oval areas on the leaves;</li> <li>- Leaf browning or yellowing and curling: Affected leaf tissue may display browning or yellowing, while leaves may also curl, common symptoms of early blight infection.</li> </ul>



TABLE 2.1: Summary of the common plant diseases, their pathogens, and symptoms.





<i>Disease name</i>	<i>Pathogens</i>	<i>Disease Example</i>	<i>Disease symptoms</i>
Esca (black measles)	Fungi		<ul style="list-style-type: none"> <li>- Necrotic spots: The leaves initially have tiny, dark brown spots that grow over time to become larger necrotic areas on the leaf surface;</li> <li>- Interveinal chlorosis and vein discoloration: Affected leaves show signs of interveinal chlorosis, which is characterized by discoloration of the veins within the affected areas as well as yellowing of the tissue between the leaf veins;</li> <li>- Leaf Curling: Curling, deformation, or other atypical growth patterns can be exhibited in infected leaves.</li> </ul>
Huanglongbing (Citrus greening)	Bacteria		<ul style="list-style-type: none"> <li>- Leaf yellowing: The leaves exhibit yellowing, which initiates from the tips and edges and progressively spreads across the leaf, resulting in a mottled appearance;</li> <li>- Vein discoloration: Discoloration of the veins within the affected leaves is evident, typically presenting as dark green or brown streaks.</li> </ul>
Late blight	Mold		<ul style="list-style-type: none"> <li>- Dark spots: Dark water-soaked spots may initially be small and irregular in shape. As the disease progresses, the lesions may change color, turning from dark brown to black, surrounded by a light green halo;</li> <li>- Leaf yellowing, wilting, and curling: Surrounding leaf tissue may exhibit yellowing, wilting, and curling, common symptoms of late blight infection.</li> </ul>
Leaf mold	Fungi		<ul style="list-style-type: none"> <li>- Light green to yellow spots: Small, light green to yellow spots emerge on the upper surface of the leaves. These spots may progress into tan or pale patches as the disease advances;</li> <li>- Leaf curling: Infected leaves may display curling or distortion, particularly around the areas of the spots.</li> </ul>

TABLE 2.1: Summary of the common plant diseases, their pathogens, and symptoms.





<i>Disease name</i>	<i>Pathogens</i>	<i>Disease Exemple</i>	<i>Disease symptoms</i>
Leaf blight (Isariopsis leaf spot)	Fungi		<ul style="list-style-type: none"> <li>- Irregular spots: Initially, irregular spots develop on the leaves, varying in color from dull brown to black as the disease progresses. These spots may gradually enlarge, forming larger necrotic areas on the leaf surface;</li> <li>- Lesion margins: The spots may have a yellow halo surrounding them and frequently have well-defined margins;</li> <li>- Leaf yellowing and curling: Infected leaves may exhibit yellowing, light green, or browning, and may also curl, especially around the areas of the lesions.</li> </ul>
Leaf scorch	Fungi		<ul style="list-style-type: none"> <li>- Marginal browning or yellowing: The leaves' tips or margins become brown or yellow, starting from the outer edges and progressing inward. The browning or yellowing may be accompanied by a dry, crispy texture and extend into the leaf between the veins;</li> <li>- Leaf curling or wilted appearance: Infected leaves may exhibit curling or a wilted appearance.</li> </ul>
North leaf blight	Fungi		<ul style="list-style-type: none"> <li>- Light brown Spots: Elongated lesions develop on the leaves, initially appearing as small, light brown spots. Over time, these spots gradually enlarge, with the lesions darkening to brown or tan. A dark green to black border may encircle the lesions, while the center of the lesion may become grayish-white;</li> <li>- Leaf Blighting and Curling: Blighting of the leaves occurs, characterized by large areas of necrosis. Additionally, affected leaves may exhibit curling or rolling, particularly around the areas where lesions have formed.</li> </ul>
Powdery mildew	Fungi		<ul style="list-style-type: none"> <li>- White powdery spots: White to grayish powdery patches or spots spread on the surfaces of leaves. Initially localized, these patches have the potential to spread and cover more of the plant;</li> <li>- Leaf curling or distortion: Infected leaves may exhibit curling, twisting, or other forms of distortion, and leaf edges curl upwards as the disease progresses.</li> </ul>

TABLE 2.1: Summary of the common plant diseases, their pathogens, and symptoms.







<i>Disease name</i>	<i>Pathogens</i>	<i>Disease Exemple</i>	<i>Disease symptoms</i>
Septoria leaf spot	Fungi		<ul style="list-style-type: none"> <li>- Circular spots: Circular water-soaked spots appear on the leaves, initially small and brownish, the spots may develop gray centers with darker margins;</li> <li>- Leaf yellowing or browning and necrosis: Surrounding the spots, the affected areas of the leaves may turn yellow or brown and eventually necrotic.</li> </ul>
Spider mites two spotted spider mite	Mite		<ul style="list-style-type: none"> <li>- Tiny spots: Fine, stippled spots may appear on the upper surface of leaves, ranging in color from white to yellow;</li> <li>- Yellowing or browning: Leaves may begin to yellow or brown, particularly on the upper surface;</li> <li>- Leaf curling or curling edges: Infected leaves may exhibit curling or rolling, especially around the edges.</li> </ul>
Target spot	Fungi		<ul style="list-style-type: none"> <li>- Circular spots: Circular lesions or spots appear on the leaves, typically with concentric rings of alternating light and dark colors, resembling a target;</li> <li>- Leaf yellowing: Surrounding the lesions, the affected areas of the leaves may turn yellow. The lesions may enlarge and coalesce, leading to extensive damage on the leaf surface.</li> </ul>
Yellow leaf curl virus	Virus		<ul style="list-style-type: none"> <li>- Leaf yellowing: Leaves display yellowing, beginning from the veins and extending towards the edges;</li> <li>- Leaf curling: Infected leaves may curl upwards or downwards, with their edges curling towards the center of the leaf.</li> </ul>
Mosaic virus	Virus		<ul style="list-style-type: none"> <li>- Mosaic patterns: Leaves develop irregular patterns of yellow to light and dark green areas, resembling a mosaic;</li> <li>- Leaf curling: Infected leaves may exhibit curling upwards or downwards, with the leaf edges curling towards the center of the leaf.</li> </ul>

TABLE 2.1: Summary of the common plant diseases, their pathogens, and symptoms.

<i>Disease name</i>	<i>Pathogens</i>	<i>Disease Exemple</i>	<i>Disease symptoms</i>
Scab	Fungi		<ul style="list-style-type: none"> <li>- Spots: Circular to irregularly shaped spots appear on the leaves, initially small and water-soaked, and later turn olive-green to black;</li> <li>- Leaf curling: Infected leaves may show signs of curling or distortion, especially around the areas of lesion formation.</li> </ul>

## 2.3 Plant disease datasets

In this section, we will outline the common disease datasets frequently utilized by researchers in the literature.

### 2.3.1 PlantVillage

The PlantVillage dataset [70] is an open dataset hosted on the PlantVillage online platform <sup>1</sup>, specifically designed for the recognition of plant leaf diseases. It comprises a total of 54,305 images, encompassing 14 crop types and 20 disease types, separated over 38 classes: 26 diseased leaf classes, and 12 healthy leaf classes. These classes represent a wide range of plant diseases, including fungi, bacterial, mold, viral, and mite diseases. The images were captured under controlled laboratory conditions with a uniform background, each sized at  $256 \times 256$  pixels.

### 2.3.2 Plant Pathology Challenge

The Plant Pathology Challenge dataset [174] is an openly accessible dataset hosted on the Kaggle online platform <sup>2</sup>. It consists of a collection of images utilized in a competition or challenge related to plant pathology, specifically focusing on foliar diseases of apples. The dataset comprises a total of 3651 images, covering two disease types: apple scab and cedar apple, separated over 3 classes: two diseased leaf classes, and healthy leaf classes. These images were taken under real-world conditions.

### 2.3.3 AI Challenger

The AI Challenger 2018 dataset, utilized in the Crop Disease Recognition Competition of the 2018 Artificial Intelligence Challenger Competition, is publicly accessible on GitHub <sup>3</sup>. This dataset comprises images of 10 crops and 27 diseases, totaling 36,379 images classified into

<sup>1</sup>PlantVillage: <https://www.plantvillage.org>, Accessed May. 2024

<sup>2</sup>Plant Pathology Challenge: <https://www.kaggle.com/c/plant-pathology-2020-fgvc7>, Accessed May. 2024

<sup>3</sup>AI challenger: [https://github.com/AIChallenger/AI\\_Challenger\\_2018](https://github.com/AIChallenger/AI_Challenger_2018), Accessed May. 2024

61 classes. These classes encompass a wide range of plant diseases, such as bacterial, mold, viral, and mite diseases. It's worth noting that the images in this dataset were captured under controlled laboratory conditions.

### 2.3.4 Plant Disease Symptoms (PDDDB)

The Image Database Plant Disease Symptoms (PDDDB) [21], available at Embrapa website <sup>4</sup>, is a comprehensive plant disease database containing 2326 images. The dataset images represent 171 diseases and other disorders affecting 21 plant species. Within the dataset, 715 images were captured in the field, while 1611 were obtained under controlled conditions. The dataset encompasses a wide range of plant diseases, including diseases caused by fungi, viruses, pests, and bacteria, as well as other issues such as phytotoxicity, algae, nutritional deficiencies, and senescence.

## 2.4 Plant disease detection and classification techniques

In light of their success in various fields, extensive research has investigated the utilization of machine learning and deep learning techniques for recognizing plant and crop diseases. The main objective is to develop methodologies and algorithms that can autonomously detect and classify diseases in agricultural settings, leveraging images of crop leaves or other crop features [41]. Notably, leaves frequently exhibit the initial symptoms of diseases, prompting a significant portion of research to concentrate on disease detection through leaf images. Traditional machine learning techniques, including feature extraction and classification, entail the extraction of features from images such as shape, texture, and color. These features are then utilized for training the classifier, enabling it to distinguish healthy from unhealthy crop leaves [164]. deep learning techniques, like CNNs, distinguish themselves by training the network to distinguish the intrinsic features of images, enabling the detection of subtle disease symptoms that traditional image processing methods might overlook [164]. Based on our comprehensive study, we categorize plant disease detection techniques into two main categories: plant disease detection and classification with machine learning, and deep learning. Each category can be further subdivided into three subcategories: techniques for detecting a specific disease in a particular crop, techniques for detecting multiple diseases in a specific crop, and techniques for detecting multiple diseases in multiple crops.

### 2.4.1 Plant disease detection and classification with machine learning techniques

This section offers a comprehensive overview of state-of-the-art approaches in plant disease detection using machine learning techniques.

#### 2.4.1.1 Techniques for detecting a specific disease in a particular crop

The studies presented in this section primarily focus on employing machine learning techniques to identify a specific disease in a particular crop. Table 2.2 outlines the latest research in

---

<sup>4</sup>Plant Disease Symptoms (PDDDB): <https://www.digipathos-rep.cnptia.embrapa.br/>, Accessed May. 2024

recognizing a specific disease across a particular crop, while Figure 2.2 presents their obtained accuracies. The study presented in [24] aims to combat powdery mildew disease in tomato plants through a hybrid approach combining SVM and LR algorithms. The method involves utilizing SVM and the Adaptive Sampling based Noise Reduction method to reduce data noise, followed by employing the LR classifier for disease prediction. Comparative analysis of the results highlights the superior performance of the proposed hybrid approach over individual SVM and LR algorithms. The paper [178] focuses on automating the detection and classification of sugarcane leaf scorch diseases through image processing techniques, including acquisition, filtering, and segmentation using color-based K-means segmentation. Additionally, mean Energy Entropy Contrast features are extracted, and a KNN classifier is employed for the classification of these extracted features. In [131], a novel method is introduced for detecting diseases and estimating their stage in cotton plants. The proposed approach utilizes two cascaded KNN classifiers: the first classifier segments the leaf from the background based on local statistical features, while the second classifier, trained on hue and luminance from the HSV color space, identifies the disease and determines its stage. The algorithm demonstrates high generalization and can be effectively applied to detect various diseases. The study described in [111] introduces an innovative and effective method for detecting and classifying diseases in wheat leaves. By employing FCM clustering on carefully selected features extracted from images of diseased wheat leaves, the system achieves accurate identification of plant diseases. Additionally, the paper outlines an efficient approach for selecting the feature set, which relies on analyzing inter and intra-class variances. Overall, the research demonstrates the effectiveness of the FCM Clustering Algorithm as a classifier for disease identification in plant leaves. The paper [42] concentrates on identifying citrus huanglongbing (HLB) through image processing and Cost-Support Vector Classification (C-SVC). It includes extracting texture and color space histograms from images, followed by feature modeling and HLB presence recognition with C-SVC. Experimental outcomes underscore the method's efficacy, achieving precise HLB recognition with low costs and computational complexity. In [192], an automated method for detecting citrus canker from leaf images captured in the field is introduced. It employs a hierarchical detection strategy, utilizing an improved AdaBoost algorithm, to segment lesion leaf images from their backgrounds. Subsequently, both color and local texture features are extracted using Local Binary Pattern Histogram (LBPH) to capture spatial properties within lesion zones. The study evaluates various classification techniques, including Radial Basis Network (RBN), SVM, and KNN. Results indicate superior classification accuracy compared to alternative methods, with performance comparable to human experts' classification. In the study by [4], the focus was on automating computer vision-based diagnosis of Cassava Mosaic Disease. Classification was based on shape and color features, using normalized histograms, Scale Invariant Feature Transformation (SIFT), and Speeded Up Robust Features (SURF). Several standard classification methods were applied, including NB, two-layer Multilayer Perceptron (MLP) networks, SVM, KNN, and divergence-based learning vector quantization. The results showed that near-perfect classification was possible for leaf images captured under ideal conditions.

TABLE 2.2: Comparative analysis of machine learning-based methods for detecting a specific disease in a particular crop.

<i>Ref</i>	<i>Crop</i>	<i>Disease</i>	<i>Dataset</i>	<i>Machine Learning technique</i>
[24]	Tomato	Powdery mildew	Tomato Powdery Mildew Disease	Hybrid of SVM and LR
[178]	Sugarcane	leaf scorch	Self	KNN
[131]	Cotton	Grey Mildew	Self	KNN

TABLE 2.2: Comparative analysis of machine learning-based methods for detecting a specific disease in a particular crop.

<i>Ref</i>	<i>Crop</i>	<i>Disease</i>	<i>Dataset</i>	<i>Machine Learning technique</i>
[111]	Wheat	Rust	Self	Fuzzy
[42]	Citrus	Citrus huanglongbing	Self	C-SVC
[195]	Citrus	Citrus canker	Self	RBN, SVM, KNN
[4]	Cassava	Mosiac	Self	NB, SVM, MLP, KNN

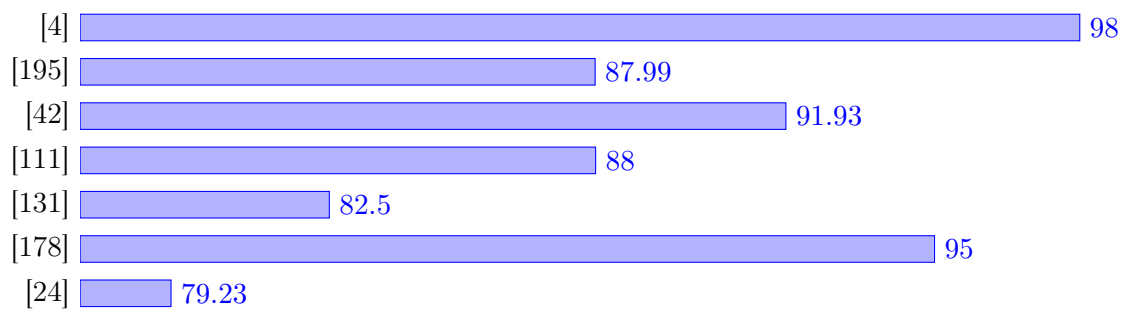


FIGURE 2.2: Bar chart of accuracy for machine learning-based methods for detecting a specific diseases in a particular crop.

#### 2.4.1.2 Techniques for detecting multiple diseases in a particular crop

This section encompasses relevant papers that address multiple diseases within a single crop used machine learning techniques. Table 2.3 highlights the recent studies in the recognition of multiple diseases in a particular crop, while Figure 2.3 presents their obtained accuracies. The authors of [114] introduced a hybrid model based on Back Propagation Neural Network (BPNN) and DT for detecting coffee diseases. They utilized K-means segmentation techniques along with Gray Level Cooccurrence Matrix (GLCM), Statistical, and Color features extraction methods. In [90], an approach is devised to identify and prevent the spread of turmeric diseases. A dataset of varied leaf images is compiled and processed through k-means image segmentation. Subsequently, the leaf images undergo textural analysis with GLCM and are classified using an SVM classifier. Prior to classification, the attributes of the images are prioritized using an information gain algorithm. Additionally, a Graphical User Interface (GUI) is created to visualize the various phases of the image processing algorithm and detect turmeric leaf diseases. The study presented in [138] explores supervised machine learning methods including NB, DT, KNN, SVM, and RF for detecting diseases in maize. These classification techniques are evaluated and compared to identify the most accurate model for predicting plant diseases. Among them, the RF algorithm demonstrates superior performance compared to other classification techniques. The study detailed in [19] introduces a computer vision framework aimed at identifying and classifying plant diseases. This system utilizes Local Tri-directional Pattern (LTriDP) to both reduce the dimensions of each class and extract discriminative information from plant leaf images. Classification tasks are performed using multiclass SVM. Experimental evaluations have demonstrated that the proposed framework surpasses other methods that rely on commonly used feature extractors. The objective of the study conducted by [126] is to streamline the process of detecting and

categorizing grape leaf diseases utilizing the SVM classification method. Initially, the segmentation technique involves identifying the diseased region through K-means clustering, followed by extracting texture and color features. Subsequently, a classification method is utilized to discern the type of disease present. The proposed methodology showcases effective detection and classification of the analyzed diseases. In [158], a comparative experiment was conducted to determine the most effective classifier for detecting plant diseases, particularly those affecting *Jatropha curcas* plants. DT, KNN, and a modified version of KNN were employed. The findings indicated that the adapted KNN method surpassed the performance of other techniques. The research outlined in [79] presents an innovative approach to diagnosing and categorizing rice diseases. A specialized algorithm is created to extract diverse characteristics, including shape and color, from the affected areas of the plants. Subsequently, these extracted features are combined for each disease and subjected to classification utilizing both Minimum Distance Classifier (MDC) and KNN classifier techniques. Experimental outcomes suggest that the MDC method excels over the KNN method regarding classification accuracy. The paper [133] presents an automated approach for identifying rice leaf diseases through image processing methodologies. The technique involves identifying and segmenting the affected areas using k-means segmentation. Color texture features obtained from each segmented region are subsequently utilized as inputs for both SVM and KNN classifiers. In this study, the SVM classifier demonstrates superior performance compared to the KNN classifier. In [67], a method is introduced for detecting and classifying citrus leaf diseases. The diseased segment is segmented using the KNN classifier, and texture features such as GLCM, Mean, Standard Deviation, Energy, Contrast, Homogeneity, and Correlation are extracted for classification with the KNN classifier. This approach effectively identifies and distinguishes the targeted diseases. An innovative system introduced in [73] revolutionizes the detection and classification of soybean diseases. The proposed system integrates multiclass SVM and KNN, coupled with preprocessing techniques, to pinpoint the Region of Interest and utilizes K-means clustering for color-based segmentation. Subsequently, color and texture features are extracted from segmented diseased leaf areas using RGB color space and GLCM. The study underscores the system's efficacy in accurately identifying and categorizing soybean diseases. In [71], the authors introduced an approach based on image processing and machine learning for disease recognition from leaf images. They utilized a segmentation approach and extracted color and textural features using GLCM. The SVM classifier was employed for classifying the extracted features, showing successful disease classification. The paper [159] introduces an Android app designed to address the detection and management of diseases impacting cotton leaves, in addition to soil quality monitoring. It proposes an SVM-based regression system for identification and classification tasks. Disease segmentation is achieved through color transformation and thresholding, while feature extraction involves utilizing color moments for color features and Gabor filters for texture features. Classification is then performed using SVM. Furthermore, the app provides farmers with information regarding the identified disease and its corresponding remedies, as well as soil quality information. The system described in [58] aims to automate the identification and categorization of plant diseases through the application of machine learning and image processing methodologies. Feature extraction encompasses both local and global features. Multiple machine learning algorithms, such as LR, RF, KNN, DT, and SVM, are assessed to identify the most suitable algorithm. The study concludes that the RF algorithm is particularly effective in this context. In [168], a methodology is proposed for both detecting and classifying diseases in potato plants. The segmentation of images is achieved using the K-means method, while feature extraction relies on the GLCM. For classification, the multiclass SVM method is employed. The proposed methodology demonstrates promising detection results. In the study [175], a web application is developed for detecting sugarcane leaf diseases using image processing techniques. The process begins with collecting leaf images, followed by preprocessing steps such as Adaptive Histogram Equalization (AHE) and segmentation using



the k-means clustering algorithm, followed by statistical feature extraction using GLCM and Principal Component Analysis (PCA) methods. Detection and classification are carried out using SVM, achieving good accuracy. The system also provides necessary control measures based on the classification results.

TABLE 2.3: Comparative analysis of machine learning-based methods for detecting multiple diseases in a particular crop.

<i>Ref</i>	<i>Crop</i>	<i>Disease</i>	<i>Dataset</i>	<i>Machine Learning technique</i>
[114]	Cofee	3 Diseases	Self	DT with BPNN
[90]	Turmeric	Leaf Spot, Leaf Blotch	Self	K-Means, GLCM, SVM
[129]	Maize	3 Diseases	PlantVillage	SVM, NB, KNN, DT, RF
[19]	Tomato	3 Diseases	PlantVillage	SVM
[126]	Grape	Downy, Powdery	Self	SVM
[158]	Jatropha curcas	9 Diseases	Self	KNN
[79]	Rice	5 Diseases	Self	KNN, MDC
[133]	Pady	Spot, leaf blast	Self	SVM, KNN
[67]	Citrus	5 Diseases	Arkansas, Reddit-plant	KNN
[73]	Soybean	3 Diseases	Self	KNN, SVM
[71]	Potato	Early blight, late blight	PlantVillage	SVM
[159]	Cotton	5 Diseases	Self	SVM
[58]	Tomato	7 Diseases	PlantVillage	SVM
[168]	Potato	Early blight, late blight	PlantVillage	Multiclass SVM
[175]	Sugarcane	4 Diseases	Self	SVM

### 2.4.1.3 Techniques for detecting multiple diseases in multiple crops

The studies examined in this section primarily concentrated on utilizing machine learning techniques for diagnosing various diseases across multiple crops. Table 2.4 provides a summary of the latest studies concerning the recognition of multiple diseases across multiple crops, while Figure 2.4 presents their obtained accuracies. The authors of [169] introduced an image segmentation algorithm aimed at automating the detection and classification of plant leaf diseases. They employed a genetic algorithm for image segmentation, which played a crucial role in disease detection. Furthermore, SVM was utilized for disease classification, with color and texture features extracted using methods such as color co-occurrence matrix, local homogeneity, contrast, cluster shade, energy, and cluster prominence. The study [8] explores the utilization of Directional Local Quinary Patterns (DLQP) as a novel feature descriptor for detecting plant leaf diseases, paired with SVM as a classifier. DLQP captures directional edge information by calculating grey-level variations between neighboring pixels in four directions. Comparative analysis with established algorithms for plant disease phenotyping, such as Local Ternary Pattern (LTP) and

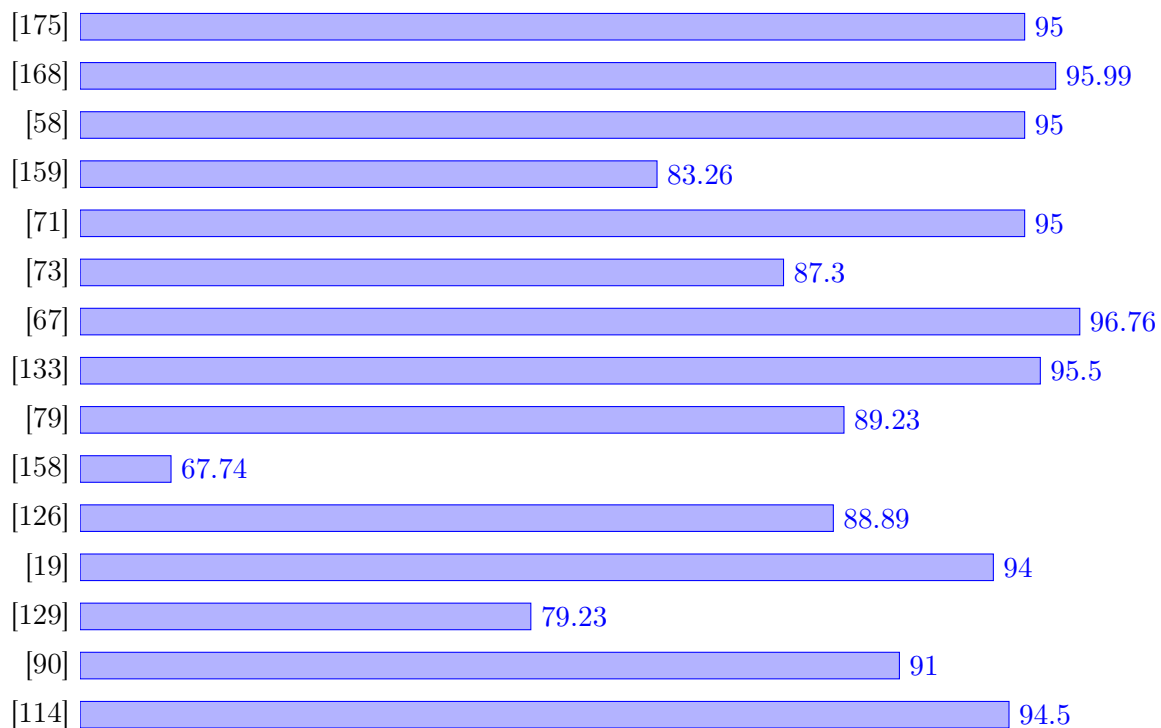


FIGURE 2.3: Bar chart of accuracy for machine learning-based methods for detecting multiple diseases in a particular crop.

LBP, additionally validates the effectiveness of the proposed method. The paper [154] presents a novel approach, the Hybrid RF Multiclass SVM (HRF-MCSVM), for detecting plant foliar diseases. The system enhances computation accuracy by preprocessing and segmenting image features using Spatial FCM before classification, followed by multi-class SVM classification. The efficiency of the proposed HRF-MCSVM method is evaluated through comparison with existing techniques. In [156], the authors introduce the Cross Central Filter (CCF) technique for noise removal in images. They employ the Cognitive Fuzzy C-Means (CFCM) algorithm to identify objects, distinguishing suspicious regions from normal ones. The evaluation demonstrates superior performance compared to alternative filters and segmentation techniques. The authors of [18] developed a processing scheme comprising four primary stages. Firstly, they create a color transformation framework for the input RGB image. Then, they eliminate green pixels by masking them using a particular threshold value, followed by segmentation. Texture and color features are then extracted. Lastly, the extracted features are fed into the MDC and SVM classifiers. The efficacy of the proposed algorithm is notable for its successful detection and classification of the investigated diseases, particularly leveraging the SVM classifier. The presented approach in [130] illustrates an automated method for identifying crop diseases from leaf sample images of different crop species. This method involves image segmentation using GMM and feature extraction using LBP. Classification is performed using a One Class SVM Classifier dedicated to each plant's health condition. When tested on new crops, the suggested methodology exhibits very high generalization behavior. In [141], the focus is on processing images of plant diseases. Color and texture features are extracted from sample images of plant diseases, and algorithms for this purpose are developed. These algorithms utilize a reduced feature set approach for recognizing and classifying plant disease images. The extracted features are then

employed to train SVM and ANN classifiers. Results suggest that the SVM classifier is more effective for identifying and classifying plant diseases in agriculture and horticulture crops. The authors of [128] presented a plant disease detection approach based on machine learning. Initially, the model segments images utilizing K-means clustering and HSV-dependent classification to identify infected sections of the leaf. Feature extraction is conducted using GLCM. Experimental comparisons were carried out for several classification models, including RF, DT, KNN, and SVM. The proposed methodology, especially when utilizing RF, exhibits high effectiveness in the detection and classification of plant diseases.

TABLE 2.4: Comparative analysis of machine learning-based methods for detecting multiple diseases in multiple crops.

<i>Ref</i>	<i>Crop</i>	<i>Disease</i>	<i>Dataset</i>	<i>Machine Learning technique</i>
[169]	4 Plant species	5 Diseases	Self	SVM with the proposed model
[8]	3 Crops	6 Diseases	PlantVillage	DLQP with SVM
[154]	14 Crops 38 Diseases	PlantVillage	HRF-MCSVM	
[156]	Apple, rice	9 Diseases	Self	FCM
[18]	9 Crops	16 Diseases	Self	SVM
[130]	18 Plant species	4 Diseases	Self	One-class SVM
[141]	11 Plant species	20 Diseases	Self	SVM
[128]	Pepper bell and tomato	3 Diseases	Self	RF, DT, KNN, and SVM

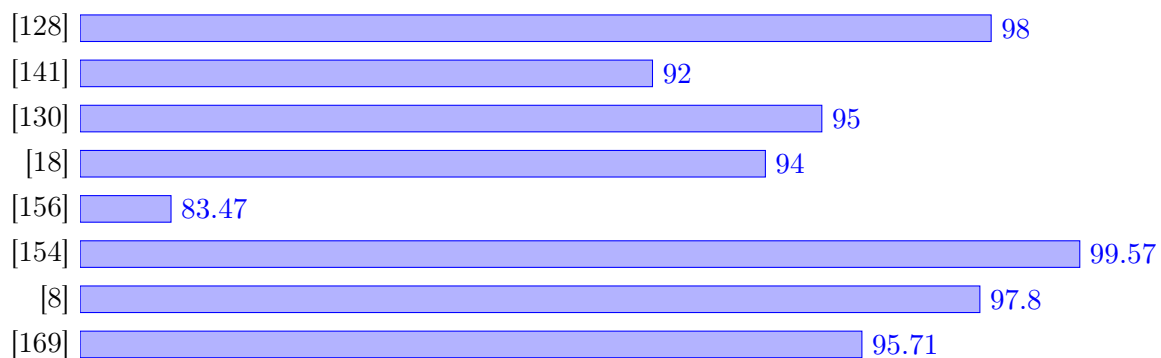


FIGURE 2.4: Bar chart of accuracy for machine learning-based methods for detecting multiple diseases in multiple crops.

## 2.4.2 Plant disease detection and classification with deep learning techniques

This section provides an in-depth summary of the methods proposed for detecting plant diseases using deep learning techniques.

### 2.4.2.1 Techniques for detecting a specific disease in a particular crop

The studies examined in this section primarily concentrated on utilizing deep learning techniques for diagnosing a specific disease in a particular crop. Table 2.5 provides a summary of recent studies focusing on the recognition of specific diseases in particular crops, while Figure 2.5 presents their obtained accuracies. In order to automatically detect northern leaf blight lesions in maize plants from field-acquired images with high reliability, a new approach was proposed by the authors of [39]. Their method utilizes a computational pipeline of CNNs to tackle challenges such as limited data and irregularities in images of plants grown in fields. A final CNN is fed with heat maps derived from the predictions of multiple CNNs trained to classify individual regions of an image for northern leaf blight lesions. This process allows for the classification of the entire image. The authors of [194] suggest an automatic method to identify cherry crop leaves infected by powdery mildew disease. Using GoogLeNet network with transfer learning techniques, the CNN achieves the best precision performance compared to traditional methods like SVM, KNN, and BP neural network. The study [98] addresses the significant losses caused by diseases in *Ginkgo biloba*, emphasizing the importance of automatically identifying disease degrees to mitigate losses. The authors employed CNNs, specifically VGGNet-16 and Inception V3, to classify different degrees of ginkgo leaf disease using two datasets: field-acquired and laboratory-acquired. Depending on the dataset, the authors observed different behaviors. The authors of [102] propose a novel method for rice blast recognition using CNN. Through comparative experiments, they demonstrate the effectiveness of CNN-based high-level feature extraction over traditional handcrafted features such as local LBPH and Haar-WT (Wavelet Transform). Quantitative evaluations show that CNNs with Softmax and CNNs with SVM perform similarly, both outperforming LBPH with SVM and Haar-WT with SVM classifiers. In [143] the authors address *Fusarium head blight* in wheat. They focused on creating a method that uses deep learning and image processing techniques to correctly identify diseased areas in color images of wheat spikes. They utilized a pre-trained Mask RCNN network with transfer learning for spike detection. Additionally, they applied a new color feature to obtain grayscale images of spikes, and a modified region-growing algorithm effectively segmented and detected a wide variety of disease forms and sizes. To address the challenge of early disease detection, the paper [22] proposes a new hybrid model combining Convolutional Autoencoder (CAE) and CNN for automatic detection of Bacterial Spot disease in peaches. The proposed system outperforms existing approaches while requiring fewer parameters, thus reducing training time and improving efficiency in disease detection. The work proposed in [191] deployed CNN models along with an imaging method developed for detecting bacteriosis from peach leaf images. The imaging method quantifies the disease-affected area and applies an adaptive operation to a suitable color image channel. Gray-level slicing is utilized for the segmentation and identification of bacterial lesions disease in peach crops. The study compares the outcomes of the CNN method and the imaging method, finding that various deep learning algorithms produce the best results. The paper [193] employs a CNN to recognize peach leaf disease caused by *Xanthomonas campestris*. Transfer learning is utilized to fine-tune AlexNet, with feature visualization demonstrating CNN's excellent ability to learn features autonomously. Comparative experiments were conducted, contrasting the CNN's performance with other classification methods including SVM, KNN, and Back Propagation (BP). CNN demonstrates superiority over these methods in recognizing unhealthy peach leaves.

TABLE 2.5: Comparative analysis of deep learning-based methods for detecting a specific disease in a particular crop.

<i>Ref</i>	<i>Crop</i>	<i>Disease</i>	<i>Dataset</i>	<i>Deep Learning model</i>
[39]	Corn	Northern leaf blight	Self	CNN (AlexNet, GoogLeNet Inception)
[194]	Cheery	Powdery mildew	PlantVillage	CNN (GoogLeNet)
[98]	Ginkgo biloba	Ginkgo leaf	Self	CNN (VGGNet-16, InceptionV3)
[102]	Rice	Rice blast	Self	New CNN
[143]	Wheat	Fusarium Head Blight	Self	CNN (Mask-RCNN)
[22]	Peach	Bacterial spot	PlantVillage	new CNN, new CAE
[191]	Peach	Bacteriosis	PlantVillage, Land Grant Universities	CNN (AlexNet, VGGNet, YOLO-v3)
[193]	Peach	Xanthomonas campestris	PlantVillage	CNN (AlexNet)

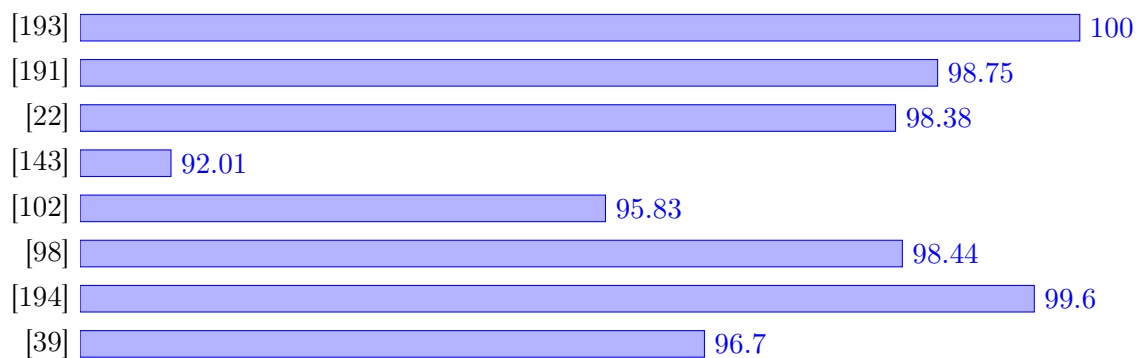


FIGURE 2.5: Bar chart of accuracy for deep learning-based methods for detecting a specific disease in a particular crop.

#### 2.4.2.2 Techniques for detecting multiple diseases in a particular crop

This section includes relevant papers that address multiple diseases within a single crop using deep learning techniques. Table 2.6 provides an overview of the latest studies in the recognition of multiple diseases in a specific crop, while Figure 2.6 presents their obtained accuracies. The paper [25] introduces a low-cost, stable, and high-precision method for identifying apple leaf diseases, employing the MobileNet model. It validates the method through experiments, comparing its efficiency and precision with established CNN models such as ResNet152 and InceptionV3. The study [104] introduced a novel CNN architecture for the detection of apple leaf diseases. The model comprised a concatenation of an AlexNet network and an Inception network. In this configuration, the Inception network substituted the fully connected layers of the AlexNet model. Optimization of network parameters was achieved by employing the Nesterov's Accelerated Gradient (NAG) algorithm. In [174], four apple leaf diseases are classified by using the ResNet50 CNN network, which has been pre-trained on ImageNet, and a Plant Pathology Challenge dataset. There were two experiments carried out: One focused on the classification

of a single disease, while the other involved a combination of multiple diseases. The paper [76] introduces a deep learning approach based on enhanced CNNs for real-time detection, named INAR-SSD. This is accomplished by integrating the GoogLeNet Inception module and incorporating Rainbow concatenation to improve the performance of small diseased object detection and multi-scale disease object detection. In [97], the authors adapted the Faster R-CNN to enhance the detection of tiny targets associated with balsam pear leaf diseases in field conditions. This was achieved by enlarging the size of the regional proposal frame and integrating the Feature Pyramid Network (FPN) based on ResNet50. The results showed that the model incorporating the FPN outperformed the original model. In [48], a deep learning approach is presented for identifying and classifying bean leaf diseases utilizing the MobileNet model. The method's goal is to effectively classify bean leaf diseases while establishing an optimized network architecture, encompassing hyperparameters and optimization methods. The findings indicate that the MobileNet model achieves good classification performance. The work performed by [106] presented a study on camellia leaf diseases, comparing two training approaches: training from scratch and transfer learning. The results demonstrated that transfer learning significantly enhances the convergence speed and classification performance of the models. The paper [16] presents an end-to-end deep learning model designed to differentiate between unhealthy and healthy corn leaves. Two pre-trained CNNs, EfficientNetB0 and DenseNet121, are employed to extract deep features from corn leaf images. These features are then combined using concatenation to create a more sophisticated feature set, enhancing the model's ability to learn from the data. Comparative analysis with other pre-trained CNN models, including ResNet152 and InceptionV3, underscores the effectiveness of the proposed model. To achieve image recognition of corn leaf diseases in complex field backgrounds with limited samples, the authors of [189] propose a CNN model (VGG16) based on transfer learning. In [196], the authors introduce a Global Pooling Dilated Convolutional Neural Network (GPDCNN) to tackle the problems of excessive parameters in the AlexNet model and the constraint of a single feature scale, particularly for plant disease identification. The work [198] introduced a novel approach to address challenges encountered in traditional CNNs when segmenting cucumber leaf lesions in images. The proposed method aimed to mitigate issues such as prolonged training duration, inadequate segmentation outcomes, and susceptibility to variations in illumination and background. Important changes included replacing the Exponential Linear Unit (ELU) activation function with the RELU activation function, employing batch normalization to enhance model stability during training, and replacing the softmax layer of the CNN with SVM classifier. To identify various degrees of grape diseases, the authors of [187] developed a multi-scale ResNet CNN based on ResNet18. This entailed modifying the first convolution layer to include a combination of multiple convolution kernels and integrating the SENet module into ResNet18. The proposed model has the highest average accuracy when compared to ResNet model. In [50], the authors introduced a model for recognizing maize leaf diseases with similar spots amidst complex real-world backgrounds. This was accomplished by modifying Faster R-CNN, which involved incorporating a batch normalization processing layer and introducing a center cost function. The improved model exhibited effectiveness in terms of both recognition accuracy and detection time compared to other CNN architectures employed. An efficient method for identifying four distinct rice leaf diseases was presented in the work [75]. The method involved the mean shift algorithm to segment disease spots, followed by shape feature extraction through artificial calculation and color feature extraction via CNN. Ultimately, the SVM classifier was employed to identify the diseases. The findings demonstrated that the combination of CNN and SVM produced superior results. In the work [7], the authors employed CNN techniques to classify and identify tomato leaf diseases. Four CNN architectures (VGG-19, VGG-16, ResNet, and Inception V3) were taken into consideration. Feature extraction and parameter tuning were employed for disease identification and classification using two datasets acquired from the field and laboratory. In both datasets,

Inception V3 was found to be the most effective model. The authors of [137] developed a mobile application presented on heat maps for identifying wheat diseases in real conditions. To improve the ResNet50 network, they replaced the softmax layer with a sigmoid activation function and replaced the first  $7 \times 7$  convolutional layer with two  $3 \times 3$  convolutions. For the detection of tomato leaf diseases at different stages, the authors of [59] introduced a multi-receptive field recognition model known as Multi-Scale AlexNet, which is derived from AlexNet. This model involved the removal of the local response normalization layer from the AlexNet network, adaptation of its fully connected layers, and integration of a multi-scale convolution kernel to extract features. The authors in [148] propose a new model called the deconvolution-guided VGG network for identifying tomato leaf diseases and segmenting disease spots. The proposed model is designed to address various issues that can arise in images, such as shadows, low light conditions, and occlusions. In [69], a technique for identifying early and late fine-grained tomato diseases was outlined. The authors introduced a novel CNN model that utilizes the ARNet architecture in conjunction with attention and residuals to extract features from fine regions. The findings of the investigation indicated that, when compared to existing models like VGG16, the ARNet exhibited superior classification performance. For the recognition of tomato leaf diseases, the authors in [5] developed a new CNN architecture consisting of three convolutional and three max-pooling layers followed by two fully connected layers. This novel CNN model outperforms pre-trained VGG16, MobileNet, and Inception V3. The work presented in [152] aims to differentiate between images of unhealthy and healthy tomato leaves to identify tomato leaf diseases using two pre-trained CNNs: Inception V3 and Inception ResNet V2. Various dropout rates were tested during the investigation. The main findings indicated that the Inception V3 model achieved optimal performance with a dropout rate of 50%, while the Inception ResNet V2 model performed best with a dropout rate of 15%. In [121], an approach was proposed for timely localization and recognition of tomato diseases, employing a robust deep learning based approach, namely ResNet-34-based Faster R-CNN. ResNet-34, along with the Convolutional Block Attention Module (CBAM), was implemented as a feature extraction component within Faster R-CNN to identify significant features within the tomato leaf images. For the detection of diseases in tomato plants, the authors of [163] proposed a deep learning approach using a combination of InceptionNet for classification and semantic segmentation techniques, namely U-Net and Modified U-Net. InceptionNet was utilized for the classification task employing supervised learning, while for segmentation, U-Net and Modified U-Net semantic segmentation models were employed. InceptionNet, combined with the modified U-Net, showcased exceptional accuracy. In [35], two CNN architectures were employed for the classification of segmented tomato leaf images. The EfficientNet model was employed for the classification task, while a modified U-Net architecture was tasked with segmenting the tomato leaves. The results underscored the superiority of deeper networks trained on segmented images, resulting in enhanced disease classification accuracy. In order to achieve real-time detection of tomato diseases, the study in [142] proposed a tomato disease recognition and classification method based on an improved MobileNetV3. By leveraging pretrained MobileNetV3 and transfer learning, the approach fine-tunes MobileNetV3 to mitigate model overfitting due to limited data. The results demonstrate a significant enhancement in tomato disease detection efficiency and a reduction in the time required for disease detection. The authors of [135] proposed the Dense Inception MobileNet-V2 Parallel Convolutional Block Attention Module Network (DIMPCNET). The network was specifically created to address issues related to the identification of tomato diseases, such as complex backgrounds, subtle distinctions between tomato diseases, and significant differences even among the same disease.

TABLE 2.6: Comparative analysis of deep learning-based methods for detecting multiple diseases in a particular crop.

<i>Ref</i>	<i>Crop</i>	<i>Disease</i>	<i>Dataset</i>	<i>Deep Learning model</i>
[25]	Apple	Alternaria leaf blotch, rust	Self	CNN (MobileNet, ResNet152, InceptionV3)
[104]	Apple	4 Diseases	Self	CNN (AlexNet, GoogLeNet Inception)
[174]	Apple	4 Diseases	Plant Pathology Challenge, self	CNN (ResNet50)
[76]	Apple	5 Diseases	Self	INAR-SSD
[97]	Balsam pear	5 Diseases	Self	R-CNN (Improved Fater R-CNN)
[48]	Bean	3 Diseases	iBean	CNN (MobileNet, Mobilev2)
[106]	Camellia oleifera	4 Diseases	Self	CNN(AlexNet)
[16]	Corn	4 Diseases	PlantVillage	CNN (EfficientNetB0, DenseNet121)
[189]	Corn	Leaf blight, rust	Self	CNN(VGG-16)
[196]	Cucumber	6 Diseases	Self	CNN(GPDCNN)
[198]	Cucumber	6 Diseases	Self	Full CNNk
[187]	Grape	8 Diseases	AI challenger, Self	CNN (Multi-Scale ResNet)
[50]	Maize	9 Diseases	Self	R-CNN (Improved Faster R-CNN)
[75]	Rice	4 Diseases	Self	New CNN, SVM
[7]	Tomato	6 Diseases	Self	CNN (VGG-16, VGG-19, ResNet, InceptionV3)
[137]	Wheat	3 Diseases	Self	CNN (ResNet50)
[59]	Tomato	7 Diseases	PlantVillage, Self	CNN (Improved Multi-Scale AlexNet)
[148]	Tomato	10 Diseases	PlantVillage	CNN (Deconvolution Guided VG-Net)
[69]	Tomato	5 Diseases	AI challenger	CNN (New CNN ARNet based)
[5]	Tomato	9 Diseases	PlantVillage	New CNN
[152]	Tomato	3 Diseases	PlantVillage, self	CNN (Inception V3, Inception ResNet V2)
[121]	Tomato	14 Diseases	PlantVillage	CNN (ResNet-34-based Faster-RCNN)
[163]	Tomato	6 Diseases	PlantVillage	CNN (Inception Net)
[35]	Tomato	10 Diseases	PlantVillage	CNN (EfficientNet)
[142]	Tomato	10 Diseases	PlantVillage	CNN (MobileNetV3)
[135]	Tomato	5 Diseases	Self	CNN (PCBAM)



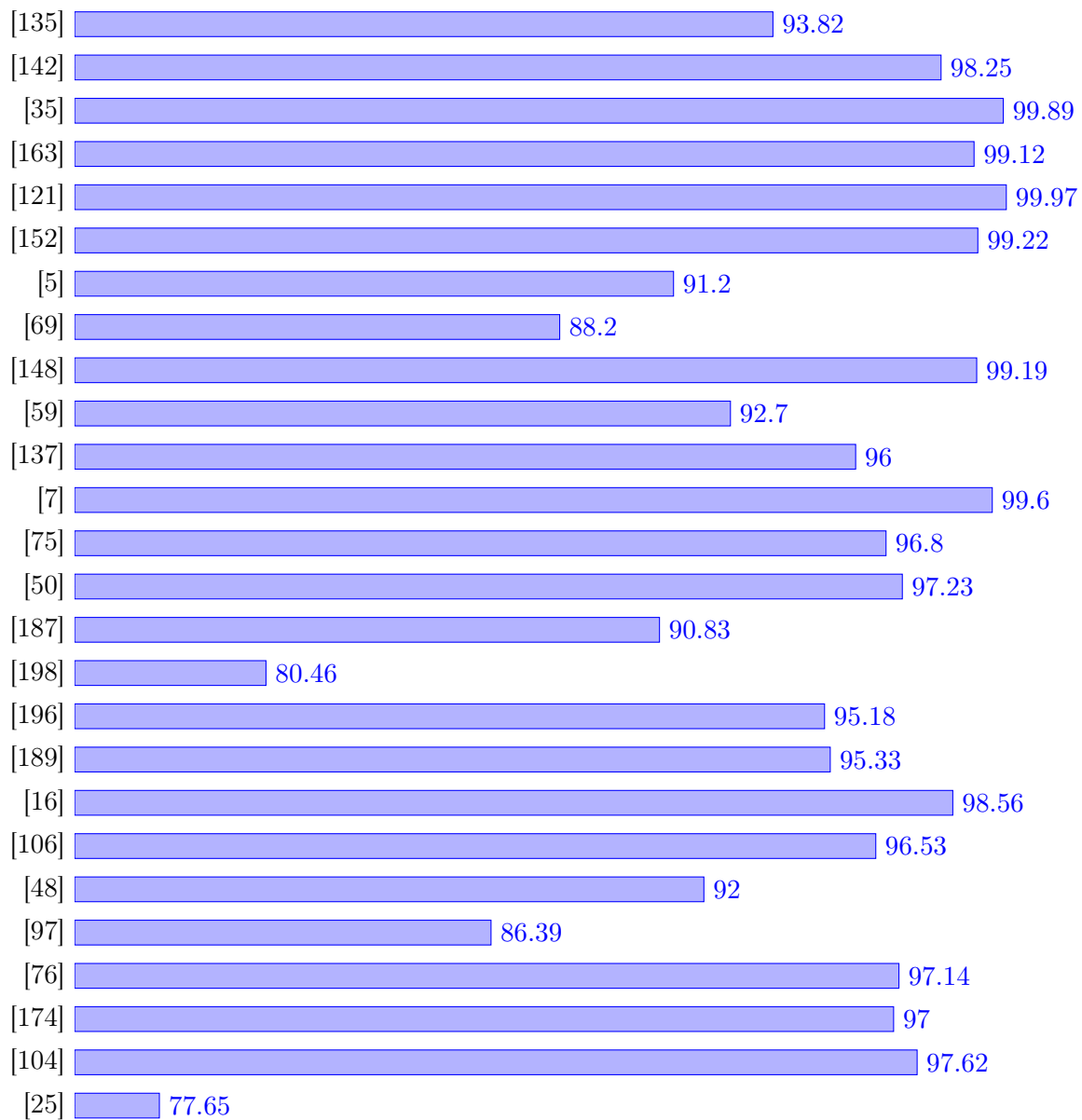


FIGURE 2.6: Bar chart of accuracy for deep learning-based methods for detecting multiple diseases in a particular crop.

### 2.4.2.3 Techniques for detecting multiple diseases in multiple crops

The works reviewed in this section predominantly focused on the application of deep learning techniques for diagnosing various diseases across multiple crops. Table 2.7 provides an overview of the latest studies in the recognition of multiple diseases in multiple crops, while Figure 2.7 presents their obtained accuracies. The study in [83] introduced a real-time plant detection model based on CNN to optimize hyper-parameters for classifying and detecting healthy and unhealthy

leaf parts. Comparative experiments demonstrated the effectiveness of YOLOv5 compared with EfficientDET and FasterRCNN. Results showcased the model's capacity to rapidly and precisely identify even minor disease patches on plant leaves. The authors of [32] focus on transferring pre-trained models, specifically, VGGNet trained on large datasets like ImageNet, to the task of identifying plant leaf diseases, creating INC-VGGN. The comparative experimental results validate the effectiveness and efficiency of the proposed approach compared to other CNN models, including DenseNet-201, ResNet-50, Inception V3, and VGGNet-19. The study presented in [185] introduces an enhanced Multi-scale ResNet model for lightweight disease recognition. This model tackles the challenge of deploying CNN on hardware with limited resources, by refining network architecture, feature extraction, and complexity reduction. The proposed model achieves notable accuracy while substantially reducing model size and training parameters. The proposed method in [197] addresses the challenge of lower segmentation accuracy in traditional CNN for crop disease leaf image segmentation by introducing a cascade CNN approach. The network consists of two components: a regional disease spot segmentation network based on the encoder-decoder model and a regional disease spot detection network based on VGG16. The results illustrated the effectiveness of the proposed approach in segmenting crop diseases in diverse environments. In [138], three CNN architectures RESNET-MC-1, RESNET-MC-2, and RESNET-MC-3 are introduced to combine contextual non-image meta-data such as crop information with image-based CNN. This integration facilitates the simplification of disease classification complexity while maintaining high accuracy across various environmental conditions. In [84], a DeepLens Classification and Detection Model (DCDM) is introduced to overcome limitations such as limited scalability, inefficiency in realistic use, and reliance on sophisticated hardware. It employs automatic detection and classification of leaf diseases by leveraging scalable transfer learning on Amazon Web Services (AWS) SageMaker, and subsequently importing it into AWS DeepLens for real-time functional usability. In [127], the proposed method involves applying pixel-based operations to leaf images to enhance their information. Subsequently, feature extraction is carried out, followed by image segmentation, and ultimately, disease classification based on the extracted patterns from the diseased leaves. CNNs are utilized for disease classification. The study outlined in [49] introduces a deep learning-based method for detecting crop leaf diseases utilizing CNN-based pre-trained models. Specifically, the research concentrates on fine-tuning the hyperparameters of established pre-trained models, such as DenseNet-121, ResNet-50, VGG-16, and Inception V4. Comparative analysis validates the effectiveness of the proposed approach. In [96], the authors presented a disease recognition method that emphasizes disease identification irrespective of crop type, using common disease names. Three methods were used to train the system: pre-training with ImageNet, pre-training with PlantCLEF2015, and training from scratch. The results showcased the model's ability to identify diseases across all crops, including those not included in the experiment, thereby underscoring the efficacy of pre-trained models from plant datasets. The work proposed in [20] suggests focusing on individual spots and lesions rather than entire leaves, thereby increasing data variability without necessitating additional images and enabling the identification of multiple diseases on the same leaf. After isolating these lesions and spots from the overall leaf image, tests were conducted to classify the lesions and identify any diseases. The findings suggest that, with sufficient data, deep learning techniques prove effective in disease detection and classification. The study [80] introduces a depthwise separable convolution architecture for plant disease recognition, featuring two variants: modified MobileNet and reduced MobileNet. Comparative experiments were conducted to assess their performance against MobileNet, AlexNet, and VGG. The compact size of these models makes them ideal for real-time crop diagnosis on resource-constrained mobile devices.

TABLE 2.7: Comparative analysis of deep learning-based methods for detecting multiple diseases in multiple crops.

<i>Ref</i>	<i>Crop</i>	<i>Disease</i>	<i>Dataset</i>	<i>Deep Learning model</i>
[83]	Bell pepper, potato	2 Diseases	PlantVillage, self	CNN (YOLOv5)
[32]	Rice, corn	9 Diseases	PlantVillage, self	CNN (INC-VGGN)
[185]	4 Crops	15 Diseases	PlantVillage, AI Challenge, self	CNN (Multi-scale ResNet)
[197]	3 Crops	9 Diseases	Self	Regional disease detection network (RD-net)
[138]	5 Crops	17 Diseases	Self	CNN (RESNET-MC-1, RESNET-MC-2, RESNETMC-3)
[84]	6 Crops	25 Diseases	PlantVillage	CNN (DeepLens)
[127]	14 Crops	38 Diseases	PlantVillage	CNN (naïve network)
[49]	14 Crops	38 Diseases	PlantVillage	CNN (DenseNet-121, ResNet-50, VGG-16, Inception V4)
[96]	14 Plant species	26 Diseases	PlantVillage, Bing	IPM, CNN (GoogLeNet, VGG-16, InceptionV3)
[20]	14 Plant species	79 Diseases	PDDDB	CNN (GoogLeNet)
[80]	24 Plants	55 Diseases	PlantVillage	CNN (Modified MobileNet, Reduced MobileNet)

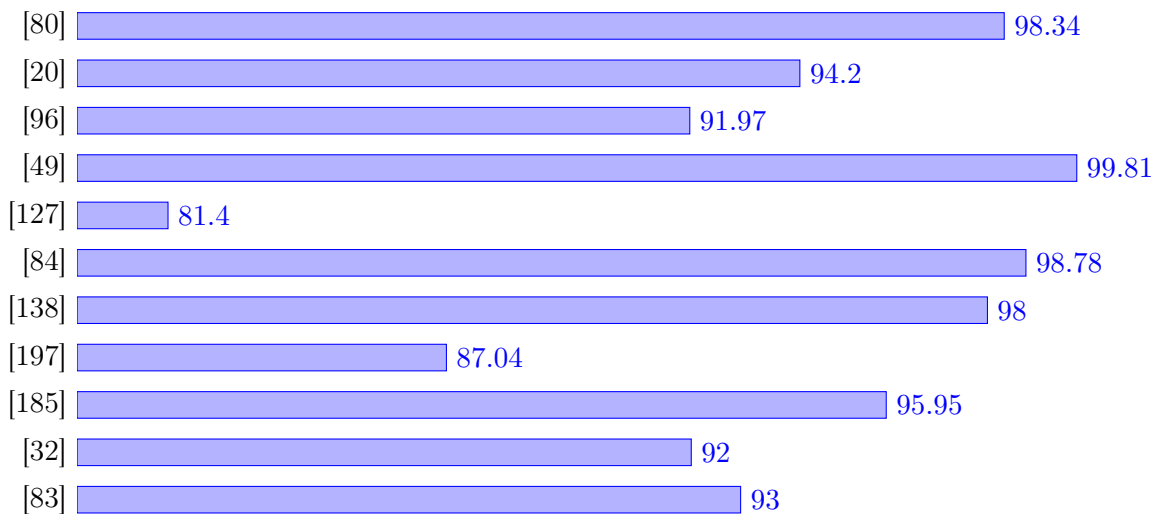


FIGURE 2.7: Bar chart of accuracy for deep learning-based methods for detecting multiple diseases in multiple crops.

## 2.5 Challenges and issues

From our comparative review of various research works on plant leaf disease detection and classification with both deep learning and machine learning techniques, several key observations emerge. Both deep learning and machine learning techniques have proven effective in these tasks, with deep learning-based methods generally outperforming machine learning-based approaches. This superiority is primarily attributed to deep learning's reliance on automatic feature extraction, contrasting with machine learning's dependence on human feature selection. However, the success of deep learning methods is heavily contingent on the availability of large training datasets. Among machine learning classifiers, SVM is widely used and considered highly effective. Conversely, CNN is the most commonly utilized deep learning architecture for plant disease detection, with models like Inception or GoogLeNet demonstrating notable effectiveness.

The majority of the aforementioned research works yield promising results, underscoring the efficacy of machine learning and deep learning in plant disease recognition. However, a common limitation across much of this prior work is a lack of robustness and generalization. Many methods are tailored to specific crops or diseases present in the dataset used for each study, limiting their applicability across diverse agricultural contexts. Furthermore, the issue of data scarcity is often addressed through techniques such as data augmentation or transfer learning, rather than tackling the root problem of insufficient data availability. Indeed, there is a noticeable gap in concurrently addressing multi-disease infections occurring within the same leaf. While many studies concentrate on detecting and classifying single diseases, the consideration of scenarios involving simultaneous disease infections is scarce. Even among the few studies that touch upon this issue, the focus tends to be narrow, typically concentrating on a single crop or a limited set of crops. As a result, these approaches often lack the necessary generalization and robustness required for practical application in real-world agricultural settings.

The challenges stemming from the lack of generalized models that can handle all crop and disease types, the variety of plant species, and the wide range of disease symptoms present in various plant and crop species present significant obstacles in creating a universal model. Deep learning and machine learning models frequently face challenges in generalizing across various plant types and diseases that were not seen in the training dataset. Additionally, the lack of labeled datasets for plant leaf diseases hinders training and may result in biased models. Additionally, because environmental conditions are dynamic, there is variability in the visual characteristics of diseased and healthy leaves, making it difficult for models to distinguish between them.

## 2.6 Conclusion

In this chapter, we've conducted a comprehensive review of the application of machine learning and deep learning techniques in plant disease detection. We delved into the field of plant pathology, exploring various types of diseases that affect agriculture and explaining the symptoms associated with each type. Then we illustrated the commonly used datasets in the plant disease field. Additionally, we introduced a taxonomy aimed at structuring the application of machine learning and deep learning in automatic disease detection. This taxonomy categorizes the works of both machine learning and deep learning into three subcategories: the detection of a singular disease from a single crop, the identification of multi-diseases within a particular crop, and the detection of multi-diseases across multiple crops. Through a comparative analysis of the proposed methodologies, We've addressed the challenges and limitations encountered in the realm of plant disease detection and classification.

## Part II

# Proposed Plant Diseases Detection Systems

---

---

## CHAPTER 3

---

# DEEP LEARNING APPROACH FOR CROSS-CROP PLANT DISEASE DETECTION

---

## Chapter contents

---

3.1	Introduction . . . . .	64
3.2	Materials and methods . . . . .	66
3.3	Results and discussion . . . . .	74
3.4	Conclusion . . . . .	81

---

### 3.1 Introduction

In this chapter, we will propose a generalized system capable of recognizing diseased and healthy leaves across different crops while also determining the extent of the disease in the leaves. The main objectives of our proposed approach are to generalize the disease detection process to all crop types, generalize the process to all disease types, and incorporate the calculation of disease extent. Several issues need to be addressed to design this system for identifying unhealthy leaves across different crops.

#### 3.1.1 Research questions and hypotheses

RQ1. *What is the chosen approach for our system: deep learning or machine learning?*

Plant disease detection using machine learning and deep learning techniques is a rapidly evolving field with promising outcomes [165]. Deep learning methods, which rely on automatic feature extraction instead of human feature selection, have demonstrated greater effectiveness compared to other machine learning techniques, especially in image recognition [104]. Several research studies have proposed deep learning-based techniques for identifying and diagnosing plant diseases. However, many obstacles prevent the better use of this technology.

RQ2. *How to generalize the disease detection process across different types of diseases and crops?*

To develop a generalized deep-learning model capable of detecting diseases across different disease types and crop types, a comprehensive dataset encompassing all types of crops and diseases is essential. This requirement is underscored by the findings in work [192], which demonstrated that the model's learning process is influenced not only by the characteristics of the disease but also by those of the crop. This suggests that the model's performance is influenced by both disease-related features and crop-specific attributes as it undergoes the learning process. Additionally, it implies that the model's ability to accurately detect diseases relies on its understanding of the unique features associated with both the disease and the crop. Furthermore, this indicates that the features extracted from a particular crop and disease cannot be generalized to other crops and diseases. Moreover, a system trained to detect a specific disease within a particular crop may not effectively identify the same disease in a different crop type. This limitation emphasizes the need for a diverse dataset encompassing various crop types and diseases to effectively train the model.

Unfortunately, the absence of such a dataset presents a considerable challenge, as constructing one is a difficult task if not an impossible one. With millions of plant species and crops in the world, along with millions of diseases that can impact them, gathering comprehensive data requires significant time and effort. These challenges directly impact the performance of deep learning-based systems due to a lack of data. The system's capacity to generalize learned patterns in plant disease recognition is hindered by the absence of access to sufficient datasets.

To resolve this challenge, the emphasis should not only be on acquiring a comprehensive dataset. Rather, the training model should focus on extracting common features present in both healthy and diseased leaves, irrespective of plant species or disease type. By prioritizing the recognition of these shared characteristics, the model can effectively differentiate between healthy and diseased leaves across a wide range of plant species and diseases.

**RQ3. *How to prevent the deep learning model from extracting crop-specific features during the learning process?***

Preventing a deep learning model from extracting crop-specific features during training is crucial for its generalization across diverse crop types. Our proposed approach involves training the model on smaller leaf pieces devoid of crop-specific features, instead of entire images containing leaves with distinct crop characteristics. These leaf pieces, extracted from the leaves, serve to mitigate the influence of specific crop structures and characteristics.

This novel approach involves splitting the image of a leaf into smaller patches, thereby facilitating the extraction of leaf pieces that lack crop-specific characteristics and features. By training the model on these extracted leaf pieces, we aim to ensure that it learns features that are applicable across different crop types and agricultural contexts. This methodology not only prevents the model from learning crop-specific features but also promotes the acquisition of more generalized representations of leaf features.

Furthermore, our proposed methodology advocates for training deep-learning models on small leaf patches extracted from diverse crop types. This approach ensures the complete elimination of crop-specific influences during training, thereby enabling the extracted disease features from a specific crop to extend to other crops.

**RQ4. *How to train the deep learning model to focus on extracting the common features of healthy and diseased leaves?***

Training the deep learning model to focus on extracting the common features of healthy and diseased leaves is crucial for its generalization across diverse disease types. Our approach emphasizes recognizing the disease itself rather than solely relying on the visual appearance of the diseased leaf. It particularly emphasizes recognizing the healthiness or infection in small pieces of the leaf. Our suggestion involves using two distinct sets of patches: one comprising patches from healthy leaves and another from infected leaves. By employing healthy leaf patches, the model can extract common features indicative of leaf health, thus learning to recognize the generalized characteristics of a healthy leaf. Similarly, utilizing infected leaf patches enables the model to capture distinguishing features associated with diseased or unhealthy leaves.

This strategy allows the model to differentiate between healthy and unhealthy leaves based on their shared and contrasting features, rather than becoming specialized in the characteristics of specific diseases. By training on these distinct sets of patches, the model can learn to generalize its extracted features of healthy and unhealthy leaves to any disease type.



**RQ5. *How to calculate the extent of disease in the leaves?***

Our proposed system incorporates a feature that allows for the calculation of the disease's extent on the leaf, providing valuable insights for further analysis and decision-making. Splitting the leaf into small patches enables the determination of the percentage of healthy and infected parts within the leaf. This allows for the quantification of the extent to which the leaf is affected by the disease, offering a quantitative measure of disease severity.

Based on the research questions, we established criteria for the proposed system. The system should be based on the deep learning approach, as this strategy has demonstrated its effectiveness in the context of plant disease detection (RQ1). It should generalize the disease detection process across different types of diseases and crops, with the training model focusing on extracting common features present in both healthy and diseased leaves, regardless of plant species or disease type (RQ2). To prevent the deep learning model from extracting crop-specific features during the learning process, we need to divide the leaf into small patches and utilize them in the training process (RQ3). To train the deep learning model to focus on extracting the common features of healthy and diseased leaves, we need to use all the extracted healthy patches to represent the healthy class and the unhealthy patches to represent the unhealthy class (RQ4). By counting the unhealthy patches extracted from the leaf, we can calculate the extent of disease in the leaf (RQ5).

The remainder of the chapter is organized as follows. Section 3.2 presents the proposed methods and the materials used. Section 3.3 presents the analysis of the obtained results. Finally, section 3.4 concludes our chapter.

## 3.2 Materials and methods

### 3.2.1 Dataset

In our study, we chose the PlantVillage dataset (For more details, please see Section 2.3.1). Notably selected for its diversity, comprehensiveness, annotations, benchmark status, accessibility, and relevance to real-world agricultural challenges, this dataset is widely utilized in research. We utilized an updated version of the PlantVillage dataset, as detailed in [117], where the leaves are segmented from the background. Figure 3.1 shows examples of images with and without backgrounds.

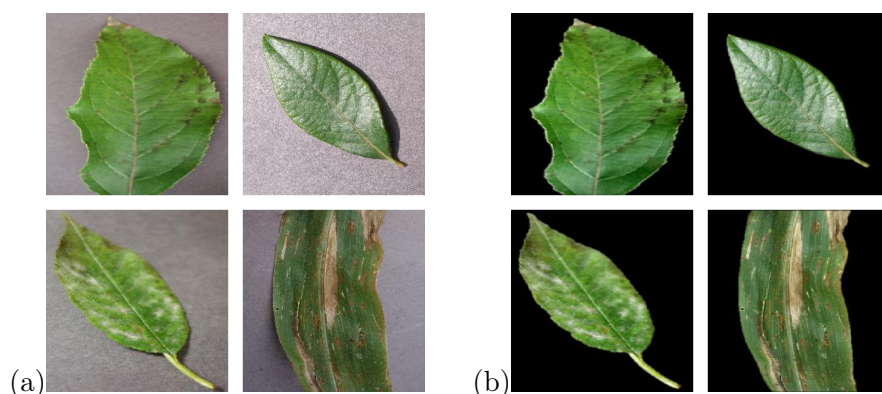


FIGURE 3.1: Examples of PlantVillage dataset images: (a) images with background, (b) images segmented from the background.

### 3.2.2 The proposed approach

This section presented a detailed explanation of the methodology used to develop a generalized plant disease classification system. The initial phase of the proposed system involves applying the designated methods to the dataset, followed by training the model with the updated dataset. Subsequently, utilizing the trained model, the system is capable of identifying the presence of infections in the leaves without necessitating specific crop or disease-type information. Additionally, the system provides an assessment of the disease's severity within the leaf. Figure 3.2 summarizes the workflow of the proposed system and shows the main phases that are involved in its functioning.

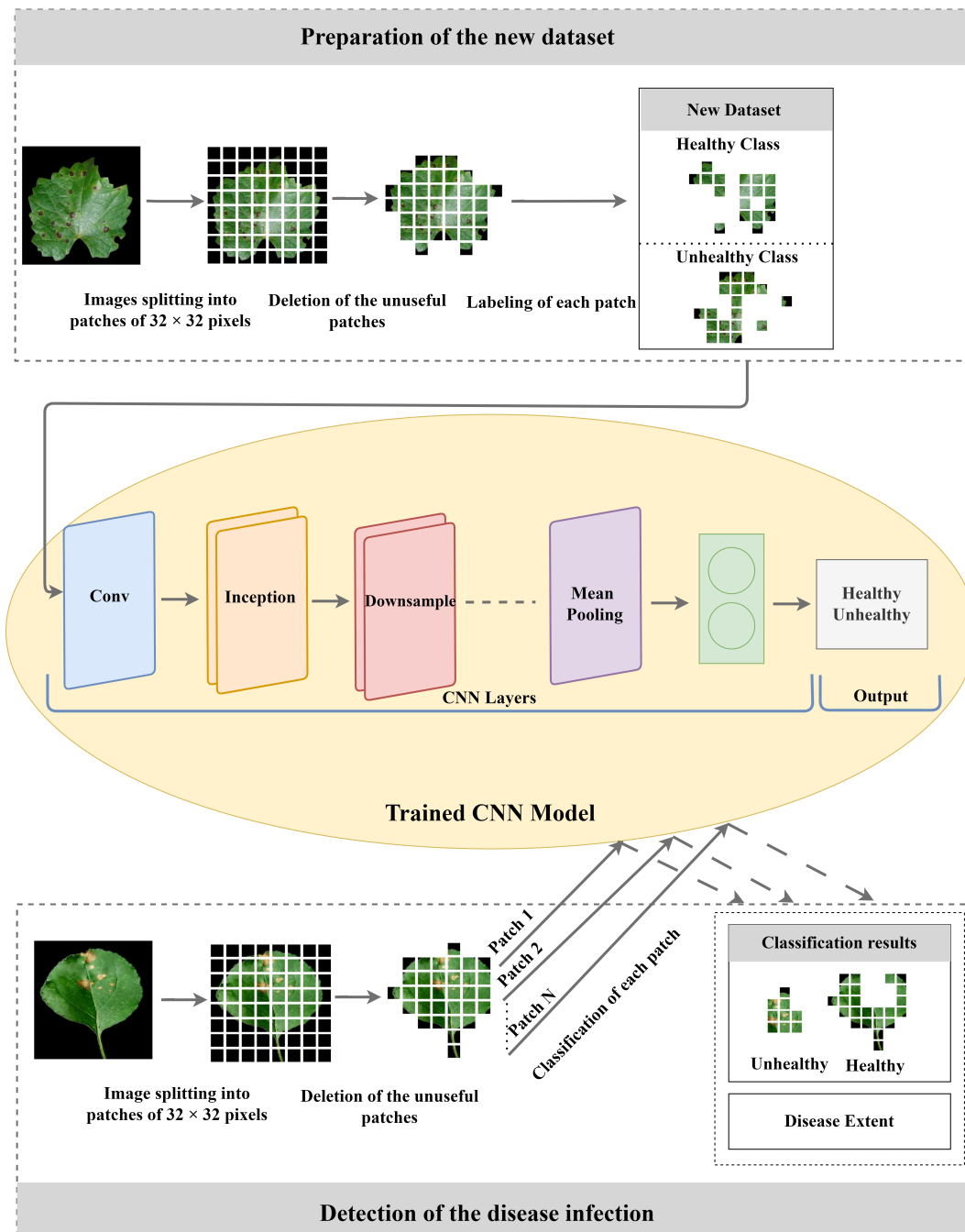


FIGURE 3.2: Flowchart of the proposed model-based CNN.

### 3.2.2.1 Generalization of the disease detection process to any crop type

To generalize the process of disease detection across any crop type, our aim is to ensure that the deep learning model, during the training process, is exclusively influenced by disease-related features and remains unaffected by crop-related features. To achieve this objective, we propose a novel method. Our approach involves splitting the leaf into small patches representing small leaf pieces, which do not contain any specific leaf type information, and utilizing them to detect the presence of the disease, rather than relying on the entire leaf in the detection process. These small patches effectively eliminate features related to crop type, ensuring the complete elimination of influences from crop type and structure. This approach guarantees that the disease features extracted from a specific crop can be applied to other crops. By processing small leaf pieces without any crop type information, we ensure the generalization of the disease detection process to any crop type, even those not seen during the training process.

To implement the proposed method, the image ( $I$ ) is split into square patches ( $P$ ) with dimensions of  $32 \times 32$  pixels. This particular size was selected to align with the input size of smaller CNN architectures. In cases where the image size is not a multiple of 32, the image is resized to the nearest multiple of 32. This ensures that the system can process images of any size, as demonstrated in Equations 3.1 and 3.2.

$$I = \{P_1, P_2, \dots, P_{N\_p}\} \quad (3.1)$$

$$N\_p = (H/h) \times (W/w) \quad (3.2)$$

Where  $N\_p$  is the number of patches extracted from one image,  $H$  is the image width,  $W$  is the image height,  $h$  is the proposed patches width, and  $w$  is the proposed patches height.

Each image in the PlantVillage dataset, initially sized at  $256 \times 256$  pixels, must be divided into a set of patches with dimensions of  $32 \times 32$  pixels. Dividing the image width (height) by the patch width (height) results in a quotient of 8, indicating that each image is split into a grid of 8 rows and 8 columns, thus creating a total of 64 patches. Equation 3.3 and 3.4 offer a mathematical representation of the procedure for this splitting process.

$$\begin{aligned} N\_p &= (256/32) \times (256/32) \\ N\_p &= 8 \times 8 \\ N\_p &= 64 \end{aligned} \quad (3.3)$$

$$\begin{aligned} I &= \sum_{i=1}^8 \sum_{j=1}^8 P_{ij} \\ I &= \{P_{11}, P_{12}, \dots, P_{88}\} \end{aligned} \quad (3.4)$$

Some criteria were applied to the patches following image splitting to remove noise from the dataset and retain only the useful ones. To determine whether a patch contains noise and should be eliminated, we calculated the percentage of black pixels in each patch based on the RGB color of each pixel, as described in Equation 3.5.

$$B = \sum_{i=1}^N \delta(R_i = 0 \wedge G_i = 0 \wedge B_i = 0) \quad (3.5)$$

$$B\_p = \frac{B}{TotalNumberofPixels} \times 100$$

Where  $B$  is the black pixel number in the patch,  $B\_p$  is the black pixel percentage of the patch,  $N$  is the total image pixels number of,  $R_i$ ,  $G_i$ , and  $B_i$  are the red, green, and blue components of the  $i$ -th pixel, respectively.  $\delta()$  is the counter delta function, which is equal to 1 if the condition inside is true and 0 otherwise.

As illustrated in Equation 3.6, patches with a black pixel percentage of 100% were eliminated because they do not contain any leaf pieces and consist entirely of black pixels. For the remaining patches, those with a black pixel percentage exceeding 50% were also eliminated. This threshold was chosen to closely align with the characteristics of the PlantVillage dataset, particularly the estimated black pixel percentage of the original image, which is approximately 50%. This approach ensures that the results closely resemble the characteristics of the original image, thus maintaining consistency with the dataset. Figure 3.3 presents the detailed processes of image splitting and the removal of unuseful patches.

$$I = \{P \mid P \in I, \text{ and black pixels percentage } B\_p \leq \text{selected black pixels percentage}\} \quad (3.6)$$

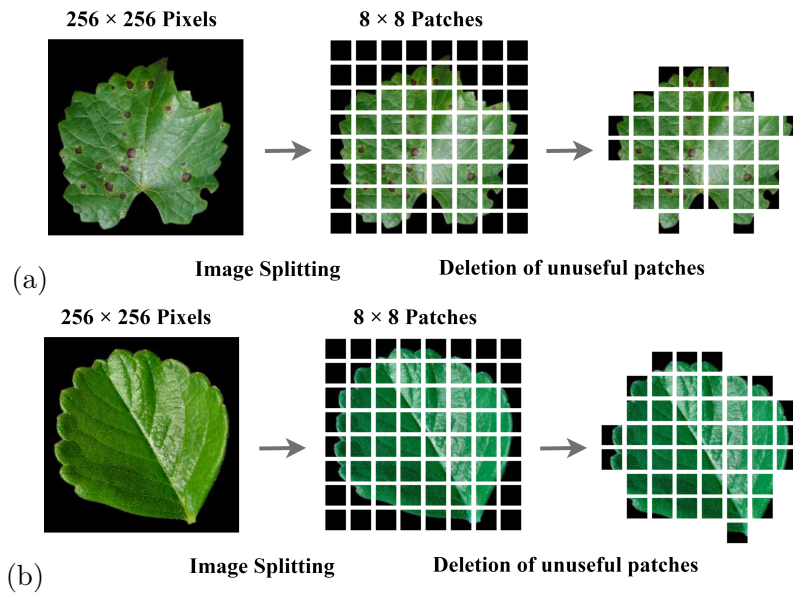


FIGURE 3.3: Image splitting examples: (a) infected grape leaf image, (b) healthy strawberry leaf image.

### 3.2.2.2 Generalization of the disease detection process to any disease type

Because of the limited data available for numerous disease types, detecting the presence of these diseases can be challenging. Initially, the focus should be on detecting the presence of infection without necessarily identifying the specific disease type. To detect the presence of infection

without determining the disease type, the system must be trained to differentiate between unhealthy and healthy leaves regardless of the specific disease type, even if not encountered during the learning process. This entails extracting features of both healthy and unhealthy leaves without considering any disease type. To achieve this, all healthy patches extracted from various crop types were grouped into one healthy class, and all unhealthy patches from different disease types were grouped into one unhealthy class. This approach enables the generalization of disease detection for any disease type.

All split patches from the PlantVillage dataset were labeled as either healthy or unhealthy based on specific criteria established through visual inspection. The labeling criteria were as follows: patches labeled as 'healthy' must contain healthy leaf pieces without any disease signs or symptoms. These healthy patches could be extracted from either healthy leaves or from unhealthy leaves that still have healthy parts, thereby benefiting from the presence of healthy patches within unhealthy leaves. Conversely, patches labeled as 'unhealthy' must contain leaf pieces presenting signs or symptoms of disease. Equation 3.7 and Figure 3.4 provide a detailed explanation of the labeling process.

$$I = \{P \mid P \in I, \text{ and healthy}\} \cup \{P \mid P \in I, \text{ and infected}\} \quad (3.7)$$

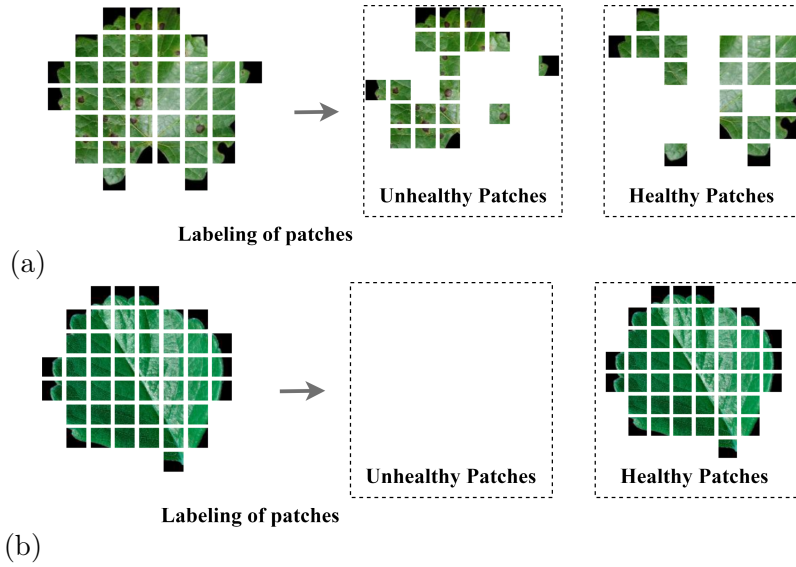


FIGURE 3.4: Data labeling examples: (a) infected grape leaf image, (b) healthy strawberry leaf image.

At the end of this process, we have a new PlantVillage dataset labeled into two classes: healthy (H) and unhealthy (U), where:

$$H = \sum_{c=1}^{N_c} \sum_{i=1}^{N_{i_c}} \{P \mid P \in I_i, \text{ and healthy}\} \quad (3.8)$$

$$U = \sum_{c=1}^{N_c} \sum_{i=1}^{N_{i_c}} \{P \mid P \in I_i, \text{ and unhealthy}\} \quad (3.9)$$

Where  $N_c$  is the dataset classes number,  $N_{i_c}$  is the class C images number.

The new dataset consists of 1860316 samples: 846162 healthy and 1014154 unhealthy. The details of the new dataset are illustrated in Table 3.1. As observed from the table, the classes Huanglongbing (Citrus greening), Leaf blight (Isariopsis Leaf spot), and powdery mildew do not generate unhealthy patches. This is because these diseases affect all regions of the leaves. Therefore, all extracted patches from these images were labeled as unhealthy.

TABLE 3.1: The new PlantVillage dataset details.

<i>Disease name</i>	<i>Crop name</i>	<i>Number of samples</i>	<i>Number of patches</i>	<i>Number of unhealthy patches</i>	<i>Number of healthy patches</i>
Healthy	Apple	1645	48693	0	48693
Healthy	Blueberry	1502	38080	0	38080
Healthy	Cherry (including sour)	854	28465	0	28465
Healthy	Corn (maize)	1162	74030	0	74030
Healthy	Grape	423	16136	0	16136
Healthy	Peach	360	7081	0	7081
Healthy	Pepper, bell	1478	54876	0	54876
Healthy	Potato	152	5505	0	5505
Healthy	Raspberry	371	12723	0	12723
Healthy	Soybean	5090	190448	0	190448
Healthy	Strawberry	456	16904	0	16904
Healthy	Tomato	1591	48173	0	48173
Bacterial spot	Peach	2297	60889	40187	20702
Bacterial spot	Pepper, bell	997	36250	19097	17153
Bacterial spot	Tomato	2127	72774	38222	34552
Black rot	Apple	621	20671	4992	15679
Black rot	Grape	1180	41174	18757	22417
Cedar apple rust	Apple	275	8104	5831	2273
Cercospora leaf spot gray leaf spot	Corn (maize)	513	25138	22738	2400
Common rust	Corn (maize)	1192	59798	56887	2911
Early blight	Potato	1000	37306	19097	17153
Early blight	Tomato	1000	30999	16850	14149
Esca (black measles)	Grape	1384	47236	27737	19499
Huanglongbing (Citrus greening)	Orange	5507	194626	194626	0

TABLE 3.1: The new PlantVillage dataset details.

<i>Disease name</i>	<i>Crop name</i>	<i>Number of samples</i>	<i>Number of patches</i>	<i>Number of unhealthy patches</i>	<i>Number of healthy patches</i>
Late bligh	Potato	1000	33730	16242	17488
Late blight	Tomato	1909	49721	28769	20952
Leaf mold	Tomato	952	22370	14596	7774
Leaf blight (Isariopsis leaf spot)	Grape	1075	43106	43106	0
Leaf scorch	Strawberry	1109	41062	39752	1310
North leaf blight	Corn (maize)	985	46769	40987	5782
Powdery mildew	Cherry (including sour)	105	35822	35822	0
Powdery mildew	Squash	1835	83457	83457	0
Septoria leaf spot	Tomato	1771	51360	34377	16983
Spider mites two spotted spider mite	Tomato	1667	41933	15110	26823
Target spot	Tomato	1404	44800	6086	38714
Tomato yellow leaf curl virus	Tomato	5357	155739	155739	0
Tomato mosaic virus	Tomato	373	7384	5209	2175
Scab	Apple	630	22984	10998	11986
Total	Total	54305	1860316	1014154	846162

### 3.2.2.3 Detecting and determining the extent of the disease

To identify the presence of disease in a leaf, we propose splitting the leaf image into non-overlapping patches of  $32 \times 32$  pixels, as outlined in Section 3.2.2.1. Subsequently, all unusable patches are eliminated using the same method presented in Section 3.2.2.1.

The classifier is then supplied with each retained patch to predict the healthiness or infection status of the corresponding section of the leaf. By predicting each leaf patch separately from the whole leaf, we can estimate the percentage of unhealthy parts of the leaves, thereby determining the extent of disease on the leaf. The number of unhealthy patches is the first step in calculating the prevalence rate of the disease ( $P$ ). Then, as Equation 3.10 illustrates, the percentage of these patches in relation to all the patches that make up the complete leaf is computed.

$$P = \frac{P_U * 100}{P_H + P_U} \quad (3.10)$$

Where  $P_U$  is the unhealthy patches number and  $P_H$  is the healthy patches number

### 3.2.3 CNN Network

The success of the GoogLeNet Inception model in the domain of plant disease detection influenced our decision to choose it for classifying the extracted leaf patches. Considering the small size of these patches, we opted for the smaller Inception model. The Small Inception architecture is an adapted version of the GoogLeNet Inception architecture, It was specially designed to fit the small input sizes. It features an adapted version of the Inception module, as shown in Figure 3.5. Consisting of three modules - the Conv module, the Inception module, and the Downsample module - each module's specifics are depicted in the accompanying figure.

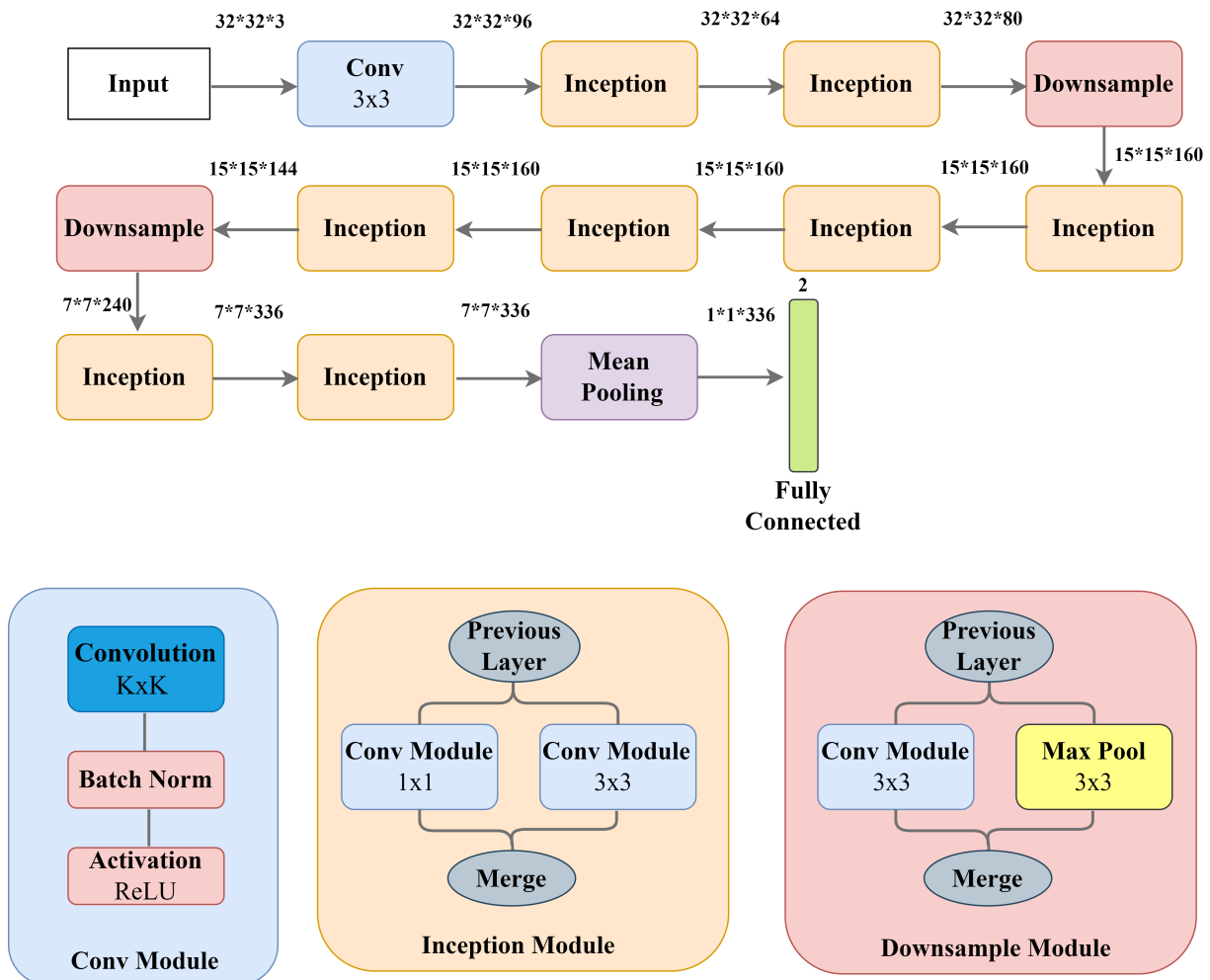


FIGURE 3.5: The Small Inception model architecture.

### 3.2.4 Experimental setup

The mechanism employed to train the model was training from scratch. After conducting numerous tests and trying various combinations of hyperparameter values, we determined the optimal combinations empirically. These selections were based on experimental observations, aiming to achieve the best performance on the proposed dataset while maintaining the system's ability to generalize to other datasets and avoiding overfitting. The Table 3.2 outlines the hyperparameters



explorer, alongside the selected values. The optimization technique utilized was adaptive moment estimation (ADAM) [86], employing its default parameters. The model was implemented in Python using the TensorFlow library, and the experiments were conducted on Google Colab using the Graphics Processing Unit (GPU).

TABLE 3.2: Hyperparameters configuration

<i>Hyperparameter</i>	<i>Values explored</i>	<i>Selected values</i>
<i>Batch size</i>	16, 32, 64, 128, 256	32
<i>Epochs</i>	1,2,...,1000	300
<i>Learning rate</i>	0.00005, 0.00002, 0.00001, 0.0005, 0.0002, 0.0001, 0.005, 0.002, 0.001, 0.01	0.001

Two sets of data were extracted from the dataset: a training set that included 80% of the images from the healthy class (676929 images) and 80% of the images from the unhealthy class (811323 images); the other set was used for testing and validation and contained 20% of the images from the unhealthy class (202831 images) and 20% of the images from our healthy class (169233 images).

### 3.3 Results and discussion

This section evaluates the proposed techniques for identifying disease infection without specifying the specific disease type and regardless of the crop type. Subsequently, we demonstrate the ability of the developed deep learning model to extend infection detection techniques to new diseases and crops that were not encountered during the learning phase. Finally, we compare the obtained results with those of the highest-ranked methods on the PlantVillage dataset to properly contextualize our work relative to other state-of-the-art techniques.

Since most papers testing their approaches on PlantVillage use classification performance metrics such as accuracy, loss, and confusion matrix, we employed them to evaluate our proposed system as well. Using the same evaluation criteria as previous studies conducted on the PlantVillage dataset ensures consistency and allows for meaningful comparisons of the performance of our proposed system with existing state-of-the-art methods. Furthermore, in order to address unbalanced datasets and carry out a more thorough examination of the model's functionality, we employed diverse evaluation metrics, including precision, recall, and the F1 score.

#### 3.3.1 Evaluation of the proposed methods

The results for the accuracy and loss function on both the testing and training sets are summarized in Table 3.3. Furthermore, Figure 3.6 illustrates the progression of the training loss function and accuracy across the training epochs. Finally, Figure 3.7 presents the confusion matrix of the proposed model. As depicted in Figure 3.6, the model's accuracy experiences rapid progress during the initial training epochs. However, after a certain number of epochs, this progression stabilizes, ultimately reaching an optimal performance of 94.66% after 300 epochs. The loss function exhibits an inverse curve, gradually decreasing at the beginning of the training epochs, then stabilizing, and ultimately reaching the lowest possible value.

TABLE 3.3: Performance on training and testing Sets

#	<i>Accuracy</i>	<i>Loss</i>
<b>Train</b>	0.9466	0.1336
<b>Test</b>	0.9404	0.1462

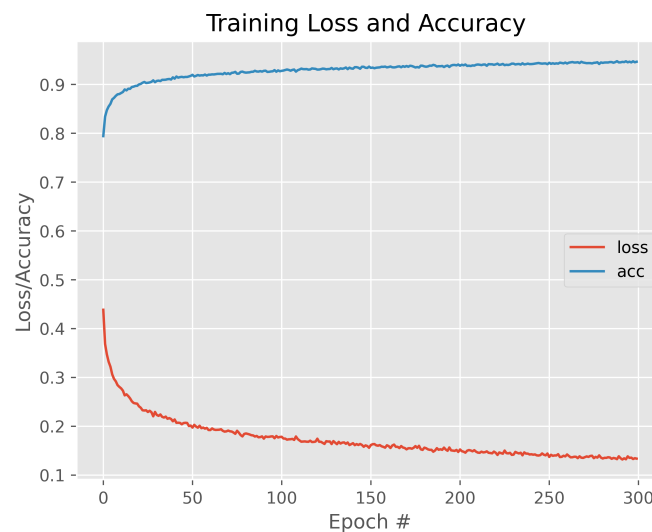


FIGURE 3.6: Training set loss and accuracy progression

The CNN achieved a testing classification performance of 94.04%. the recognition accuracy of the unhealthy class is better than that of the healthy class. This performance was distributed among the two classes as follows: 94.15% of samples from the healthy class were correctly predicted, while the error rate was 5.85%. Similarly, in the unhealthy class, 93.95% of samples were correctly predicted, with an error rate of 6.05%. For the purpose of identifying classification failures made by the model, each class, healthy and unhealthy, underwent careful analysis. The findings were as follows: mislabeling errors in the healthy class were primarily attributed to patches extracted from tomato crops. Conversely, patches originally from diseases such as Powdery mildew, Huanglongbing (Citrus greening), Toamto yellow leaf curl virus, Spider mites, two-spotted spider mites, and Bacterial spot were the main contributors to classification errors in the unhealthy class.

The misclassification observed in the model's predictions can be attributed to several factors. In particular, the mislabeling error observed in the healthy class can be explained by the notable similarity between healthy tomato leaves and squash leaves affected by powdery mildew disease, as depicted in Figure 3.8. This visual resemblance between the appearance of tomato leaves and the symptoms of powdery mildew in squash can lead to patches extracted from tomato crops closely resembling those from the squash powdery mildew class. The confusion results from The similarity between the epidermal hairs, or specialized tissues, present on tomato leaves and the symptoms of powdery mildew leads to confusion. These epidermal hairs can sometimes resemble the powdery appearance associated with mildew infection, leading to misclassification. Hence, both epidermal hairs and powdery mildew can contribute to a white or powdery appearance on the leaf surface, posing a challenge for accurate classification. On the other hand, the misclassification observed in the unhealthy class was influenced by two main factors: Firstly, powdery

Confusion matrix

Predicted	Healthy	159326 42.82%	12263 3.30%	171589 92.85% 7.15%
	Unhealthy	9907 2.66%	190568 51.22%	200475 95.06% 4.94%
sum_col		169233 94.15% 5.85%	202831 93.95% 6.05%	372064 94.04% 5.96%
		Healthy	Unhealthy	sum_lin
		Actual		

FIGURE 3.7: The new PlantVillage dataset confusion matrix.

mildew causes white powdery spots to develop on the leaves. However, patches containing only a small number of these white powder spots were not classified as unhealthy. This was mainly because of the difficulty in detecting this limited proportion of white powder spots. Secondly, patches originally taken from disease classes characterized by symptoms involving shoots appearing between light green and yellow, as illustrated in Figure 3.8, specifically those containing only small shoots of these disease symptoms, led to confusion in their classification by the CNN model. This was because it was very challenging for the model to recognize these small shoots.

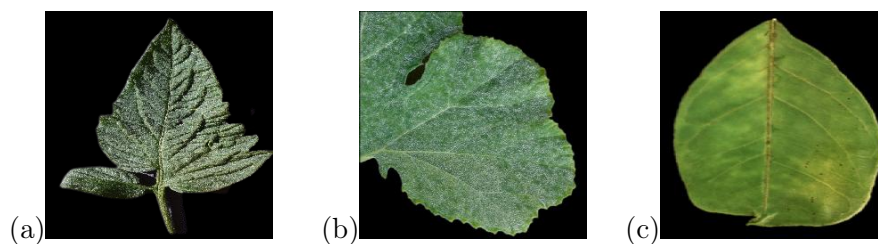


FIGURE 3.8: Data examples: (a) healthy tomato leaf image, (b) Powdery mildew-infected squash leaf image, (c) Huanglongbing (Citrus greening) infected orange leaf image.

As presented in Table 3.1, the newly created PlantVillage dataset exhibits some level of imbalance. To thoroughly assess the impact of this imbalance and delve deeper into the model's performance, particularly in distinguishing between diseased and healthy plants, we employed three critical metrics: precision, recall, and F1 score. Table 3.4 displays the results for these metrics, thus results can be explained as follows: Firstly, as indicated in the table, the precision, recall, and F1 score results demonstrate a balanced performance of the classification model. This conclusion is drawn from the close alignment of the three performance metrics. Secondly, a precision of 0.9505 indicates that 95.05% of the instances predicted as positive by the model

are actually correct. With a recall of 0.9395, the model can accurately identify approximately 93.95% of actual positive cases. An F1 score of 0.9450 indicates a robust balance between precision and recall, which indicates better model performance. In conclusion, all these findings collectively demonstrate that the model’s performance remains consistent and robust, even in the presence of imbalances in the dataset.

TABLE 3.4: Performance on testing set

#	<i>Precision</i>	<i>Recall</i>	<i>F1 score</i>
<i>Test</i>	0.9505	0.9395	0.9450

### 3.3.2 Evaluation on new crop and new disease

Testing the proposed model on additional unknown datasets is essential to demonstrate its generalizability. This includes assessing its performance on novel crops and diseases that are not included in the PlantVillage dataset. To evaluate the model’s generalizability, we considered three unseen situations: Firstly, a crop type not included in the training data infected with a disease type absent in the training dataset, such as cotton infected with Soreshin disease. Secondly, we assessed a situation where a crop seen during training was infected with a disease not encountered in the training process, like soybean affected by southern blight disease. Lastly, we investigated a situation where a new crop, unseen by the classification model, was infected with a disease present in the training dataset, such as soybean infected with southern blight disease. The new crop type and disease were selected from the challenging dataset PDDB (For more details, please see Section 2.3.4). From the cotton images, we randomly selected 19 images, while 62 images were chosen from the soybean images, and 27 images from coffee plants, resulting in a total of 108 images for evaluation. After applying the same preprocessing steps as those used for the PlantVillage dataset, we created a new test dataset from these images. This dataset contains 1619 patches in total, with 495 patches classified as healthy and 1124 patches classified as unhealthy. Examples of PDDB dataset images are presented in Figure 3.9.

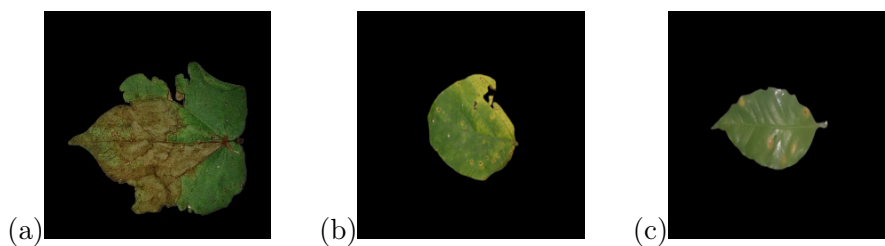


FIGURE 3.9: PDDB dataset examples: (a) cotton leaf (b) soybean leaf (c) coffee leaf.

The results of the model evaluation on the newly created dataset were illustrated in Figure 3.10. The model successfully achieved an accuracy of 97.22%. Notably, The model performed better in the unhealthy class compared to the healthy class, with recognition accuracies of 99.55% and 91.91%, respectively. In investigating the system’s classification failures, we carefully analyzed the misclassified patches from the PDDB dataset. In comparison to the correctly classified patches, these elements were identified as the primary causes of misclassification: Firstly, the mislabeling errors in the healthy class were attributed to the challenge in distinguishing healthy patches with light green colors, as they visually resembled symptoms of diseases characterized

by shoots appearing between light green and yellow. Additionally, patches containing only small disease symptom spots were the most challenging for classification in the unhealthy class.

Confusion matrix

Predicted	Healthy	455 28.10%	5 0.31%	460 <b>98.91%</b> <b>1.09%</b>
	Unhealthy	40 2.47%	1119 69.12%	1159 <b>96.55%</b> <b>3.45%</b>
	sum_col	495 <b>91.92%</b> <b>8.08%</b>	1124 <b>99.56%</b> <b>0.44%</b>	1619 <b>97.22%</b> <b>2.78%</b>
		Healthy	Unhealthy	sum_lin
		Actual		

FIGURE 3.10: PDDB dataset confusion matrix.

### 3.3.3 Comparison with other state-of-the-art methods

Table 3.5 provides a comparison between our system’s results and those of the highest-ranked state-of-the-art methods on the PlantVillage dataset, considering various comparison criteria. While our proposed method may not have achieved the highest accuracy score, it demonstrates robust and strong performance, showcasing its effectiveness compared to state-of-the-art methods. Our approach encompasses the entirety of the PlantVillage dataset, ensuring comprehensive coverage and broadening its applicability. This is unlike other state-of-the-art methods, which focus solely on specific portions of the dataset. The standout feature of this system is its remarkable capability to identify leaf infections without considering the specific crop or disease type, making it a comprehensive solution suitable for multiple crops and diseases. This adaptability allows the system to recognize new types of crops and diseases that were not part of the training dataset, making it adaptable to novel crop and disease types. Unlike the proposed method, other state-of-the-art methods are specialized in the specific set of crops or diseases encountered during the training process. As a result, they do not possess the ability to serve as multi-crop and multi-disease solutions. This limitation means that they are unable to recognize other types of crops and diseases not included in the training dataset. In comparison, the suggested approach demonstrates a greater degree of adaptability and robustness in handling a wide variety of crops and diseases.

TABLE 3.5: A comparative study with state-of-the-art methods using the PlantVillage dataset

<i>Ref</i>	<i>All dataset</i>	<i>Multi-crop</i>	<i>Multi-disease</i>	<i>Generalized to new crop</i>	<i>Generalized to new disease</i>	<i>Accuracy (mean)</i>
[194]	×	×	×	×	×	99.6%
[22]	×	×	×	×	×	98.38%
[191]	×	×	×	×	×	98.75%
[193]	×	×	×	×	×	100 %
[185]	✓	×	×	×	×	95.95%
[148]	×	×	×	×	×	93.25%
[59]	×	×	×	×	×	92.7%
[5]	×	×	×	×	×	91.2%
[96]	✓	✓	×	✓	×	91.97%
[32]	×	×	×	×	×	92%
[80]	✓	×	×	×	×	98.34%
[121]	×	×	×	×	×	99.97%
[84]	×	×	×	×	×	98.78%
[49]	✓	×	×	×	×	99.81%
[152]	×	×	×	×	×	99.22%
[16]	×	×	×	×	×	98.56%
[35]	×	×	×	×	×	99.89%
[163]	×	×	×	×	×	99.12%
[127]	✓	×	×	×	×	90.40%
[83]	×	×	×	×	×	93%
[142]	×	×	×	×	×	98.25%
<i>Proposed</i>	✓	✓	✓	✓	✓	94,04%

### 3.3.4 Discussion

There are several deductions that can be made from the aforementioned findings. Firstly, The primary reason behind most of the failures in classifying healthy patches is the significant similarity between certain disease symptoms and the natural characteristics of some crop leaves. It poses a challenge for the system to distinguish between these symptoms and characteristics, especially in discerning the light green shoots in the unhealthy patches resulting from infection with some diseases and the natural light green color present in some healthy patches. Concerning the unhealthy class, the majority of errors occurred in classifying patches containing small disease spots, as the system struggled to identify these minor abnormalities.

Secondly, assessing the trained CNN model on a new dataset containing previously unseen disease and crop types illustrates the system's capability to identify novel diseases and crops not encountered during training. This highlights the system's generalization capacity to accommodate any crop or disease type, demonstrating its robustness and adaptability to diverse agricultural scenarios.

Furthermore, our system has attained a prominent status among other state-of-the-art methods that employ the PlantVillage dataset. What distinguishes our program from others is its ability to tackle any disease found on diverse types of crop leaves, thus making it a versatile solution capable of handling multiple crops and diseases.

In conclusion, despite the dataset's diversity, the proposed system performs well. There is a notable diversity among patches within the same class, with each class containing various crop types and disease types. Each crop type's leaves display unique shapes and textures, while noticeable variations exist in the vein patterns of leaves. Moreover, diseases exhibit distinct symptoms, with significant differences observed among them. Some symptoms manifest as spots, while others cause changes in the color and texture of the images. Figure 3.11 illustrates the diversity present in the dataset. The proposed system provides an adaptable and highly beneficial solution for agriculture. Its ability to detect disease infections in crops, regardless of the specific disease or crop type, makes it applicable across various crops and diseases. This versatility is particularly advantageous in real-world farming scenarios characterized by diverse crops and potential threats from various diseases. In such situations, the system aids in the timely identification and efficient management of diseases.



FIGURE 3.11: Dataset samples: Examples.

### 3.4 Conclusion

This chapter presents a new, generalized system to address the plant disease recognition problem based on a novel approach. This approach relies on a modified version of a deep convolutional network designed for a small input size, known as the Small Inception. The proposed system demonstrates the capability to distinguish between unhealthy and healthy leaves, irrespective of leaf crop type or disease. This achievement is facilitated by an additional preprocessing step introduced before the classification process, which involves processing patches of the leaf rather than the entire leaf. The splitting of images into small patches helps remove the classifier's reliance on the leaf structure while also enhancing the dataset with real data. Furthermore, the system exhibits the ability to generalize the detection process of infections across various leaf and disease types, even those not encountered during the classifier's training phase.

The proposed system also surpasses other state-of-the-art techniques, which often rely on specialized methods for particular crops or diseases. In contrast, our system is a generalized approach that utilizes the same model for different crop and disease types. This distinction highlights the versatility and effectiveness of our approach compared to methods that are limited to specific applications.



---

---

## CHAPTER 4

---

DEEP LEARNING APPROACH FOR  
SIMULTANEOUS MULTI-DISEASE  
DETECTION ON THE SAME LEAF

---

## Chapter contents

---

4.1	Introduction . . . . .	83
4.2	Materials and methods . . . . .	84
4.3	Results and discussion . . . . .	94
4.4	Conclusion . . . . .	101

---

## 4.1 Introduction

As mentioned in the previous chapter, the deep learning approach is the preferred method in plant disease detection. Building upon the approach outlined previously, we propose a new generalized system for detecting multiple diseases from the same leaf, along with determining the extent of all diseases present on the leaf. The main objectives of our proposed approach are to detect the presence of multiple diseases in the leaf simultaneously, identify each disease type, generalize the process of multi-disease detection to all crop types, calculate the prevalence rate of each disease, and assess the overall extent of all diseases present on the leaf. To design such a system, several questions need to be addressed.

### 4.1.1 Research questions and hypotheses

RQ1. *How to detect the presence of multiple diseases in a leaf simultaneously from any crop type?*

Sometimes, the issue of plant disease infection exacerbates and poses a greater threat when a plant is infected with multiple diseases at the same time. To automatically recognize this multi-disease infection based on deep learning, it is necessary to have a dataset encompassing all possible infection scenarios, including diverse plant types and diseases. As demonstrated in the previous chapter and proven by research [192], the model's training can be influenced by the crop disease type, and the extracted features may not be extendable to other types of diseases and crops. Therefore, the model for recognizing multi-disease infections needs to be trained on such cases. However, obtaining images of various diseases affecting different crops is very challenging, but it is not impossible because there are millions of plants in the world that can be infected with millions of diseases, so it requires a lot of effort, time, money, and luck. Consequently, the challenge of simultaneous multi-disease detection is not yet adequately addressed.

In response to these challenges, we propose a method to identify each disease separately from others present on the same leaf, regardless of the crop type. Our approach involves isolating each disease instance from others and from the crop type. By isolating each disease separately and eliminating the influence of crop type extension, our solution enhances the generalizability and applicability of the model across diverse agricultural contexts.

RQ2. *How to isolate each disease type from other diseases affecting the same leaf and from the crop type?*

To ensure the effective isolation of each disease type from other diseases and crop types, the approach involves training the model on small leaf pieces. These pieces isolate each disease-specific feature, rather than relying on entire images containing leaves with distinct diseases and crop characteristics. These pieces extracted from the leaves serve to mitigate the influence of specific crop structures and characteristics while also isolating each disease symptom.

This novel approach involves splitting the image of a leaf into smaller patches, thereby facilitating the extraction of leaf pieces that isolate each disease type and eliminate crop-specific characteristics and features. By training the model on these isolated leaf pieces, we aim to ensure that it learns features that are specific to each disease type and applicable across different crop types and agricultural contexts. This approach prevents the model from being influenced by specific crop characteristics and promotes the acquisition of more generalized disease detection capabilities.

Furthermore, by training deep-learning models to learn features specific to each disease type from patches extracted from diverse crop types, we guarantee the complete elimination of crop-specific influences during training. This ensures that the disease features extracted from a specific crop can extend to other crops.

**RQ3. *How to calculate the prevalence rate of each disease and assess the overall extent of all diseases present on the leaf?***

Our proposed system integrates a feature enabling the calculation of the prevalence rate of each disease on the leaf, thereby assessing the overall extent of all diseases present. This provides valuable insights for further analysis and decision-making. Dividing the leaf into small patches allows the determination of the percentage of healthy and disease-infected parts within the leaf, facilitating the quantification of each disease's prevalence rate and its impact on the leaf.

According to the research questions, we have defined criteria for the proposed system. The system to detect the presence of multiple diseases in a leaf simultaneously, regardless of the crop type, involves recognizing each disease type separately from others and disregarding the crop type (RQ1). To achieve this, we need to isolate each disease type from others affecting the same leaf and from the crop type. This can be accomplished by splitting the leaf image into small patches and using all patches extracted from the same disease to extract the features of that disease, along with using healthy patches extracted from all crop types to extract the features of healthy leaves. Subsequently, the deep learning model should be trained on these patches (RQ2). By counting disease-specific patches, we can calculate the prevalence rate of each disease, thereby determining the extent of all diseases present in the leaf (RQ3).

The structure of the remaining chapter is as follows. Section 4.2 outlines the proposed methods and the materials employed. Section 4.3 provides an analysis of the results obtained. Finally, Section 4.4 offers the conclusion of our chapter.

## 4.2 Materials and methods

### 4.2.1 Dataset

The PlantVillage dataset was utilized in this contribution. For more details, please see Sections 3.2.1 and 2.3.1.

### 4.2.2 A generalized method for simultaneous multi-disease recognition from the same leaf

Several factors contribute to plants becoming infected with multiple diseases simultaneously during their life cycle. Each disease exhibits its own set of external and internal symptoms, which typically manifest in the internal or external characteristics of the plant. These symptoms can result from exposure to various pathogens such as bacteria, fungi, microscopic animals, and viruses [96]. Leaves are commonly the first part of the plant where external symptoms appear [47]. The symptoms of diseases are often distinctive, aiding in the differentiation of specific diseases from others. However, these distinctive symptoms of the disease affect all plants in the same way, causing the same apparent symptoms. As depicted in Figure 4.1, examples of the infection of two diseases, Black rot, and Bacterial spot, on two different crops, potato, and pepper leaves, produce similar symptoms.

Detecting infections with multiple diseases simultaneously is a complex and challenging task. Since each disease exhibits its unique set of symptoms, our approach involves identifying the symptoms of each disease individually, independent of other diseases affecting the leaf. Furthermore, considering that each disease affects all plants uniformly, we propose identifying them solely based on their symptoms, irrespective of the crop type. This lays the foundation for our proposed method, which aims to simultaneously identify multiple diseases from the same leaf. We suggest detecting each disease's symptoms independently, without consideration of the crop type.

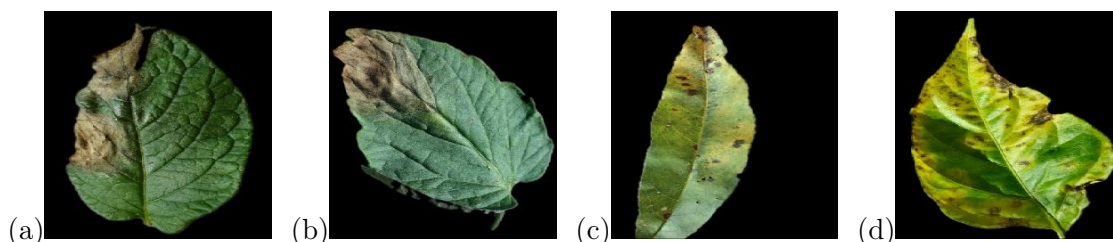


FIGURE 4.1: The examples of disease symptoms include: (a) Black rot infection on a potato leaf, (b) Black rot infection on a tomato leaf, (c) Bacterial spot infection on a peach leaf, and (d) Bacterial spot infection on a pepper leaf.

### 4.2.3 Proposed system

We have introduced a system designed to identify infections with multiple diseases simultaneously, regardless of the crop type, based on the method mentioned above. We implemented the proposed method in a real-world scenario and developed an approach to isolate each disease independently from others, without considering the crop type. After applying this approach to the dataset, we trained the CNN model with the new dataset. Upon completion of the training process, the resulting system can detect and classify diseases while providing information about the extent of the disease in the leaves. The details of the proposed approach and model are presented as follows. Figure 4.2 provides a general presentation of the proposed model.

#### 4.2.3.1 Isolation technique

This technique involves dividing each leaf into smaller patches and processing them individually. Each patch will undergo separate prediction, and the results of all the patches will be combined

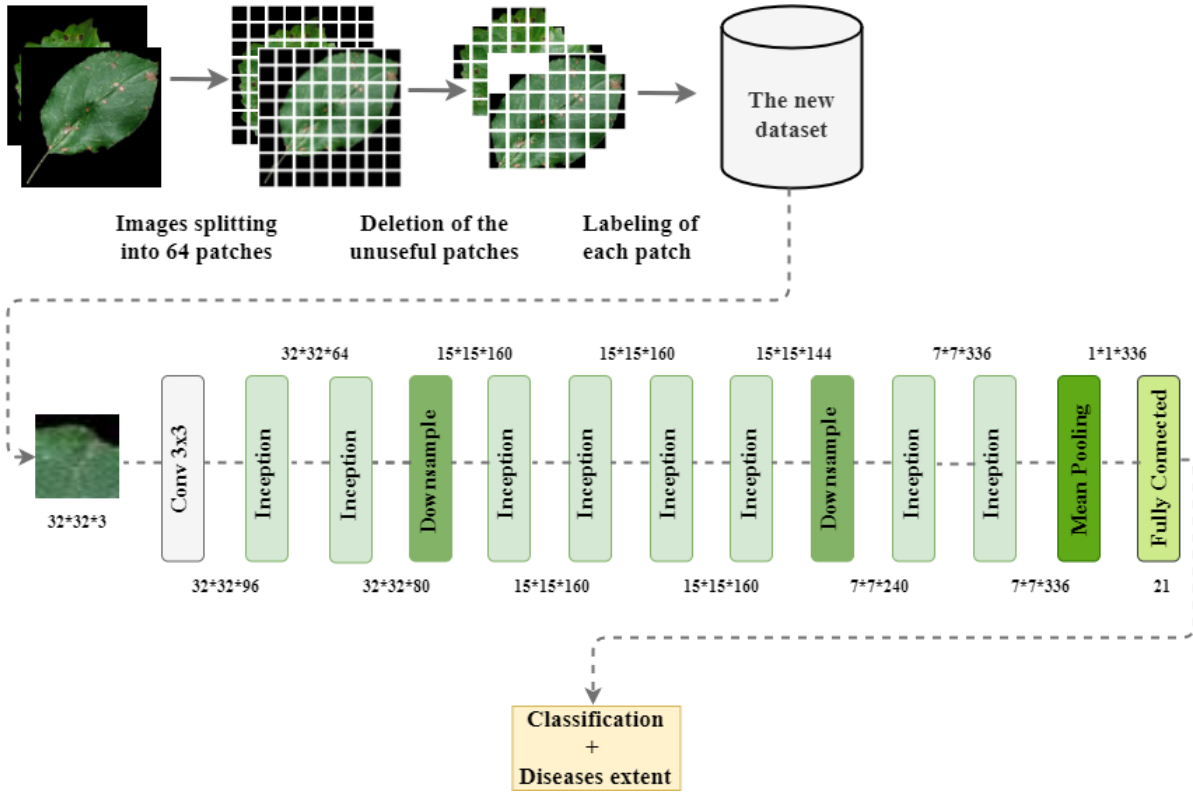


FIGURE 4.2: Flowchart of the proposed classification model based CNN

to determine the set of diseases present in the leaf or the overall health status of the leaf. This approach is employed instead of analyzing the entire leaf to detect the set of diseases infecting it. These patches ensure that each disease is effectively and separately isolated from others. Furthermore, these patches do not contain any information about the crop type, eliminating the characteristics related to the crop type. This allows for the generalization of the multi-disease detection process to any crop type. During the learning process, the CNN model extracts features related to both the disease and the crop. However, by processing patches that do not contain any crop type information, we focus solely on the disease. This ensures that only features relevant to the disease are extracted, and these features can be extended to new types of crops, as demonstrated in [96].

To implement the proposed method, each leaf image ( $I_D$ ) from class ( $D$ ) in the dataset is divided into equal square patches ( $P$ ) sized as  $32 \times 32$  pixels, as demonstrated in Equations 4.1 and 4.2. The criteria for choosing this size are to make the patches as small as possible while ensuring that each patch can represent one specific disease and to maintain compatibility with smaller CNN architectures. If the image dimensions are not a multiple of 32, a modification is made to allow the system to process images of any size. The resizing of the image is done to the closest multiple of 32.

$$I_D = \{P_1, P_2, \dots, P_{N_p}\} \quad (4.1)$$

$$N_p = (H/h) \times (W/w) \quad (4.2)$$

Where  $N_p$  is the number of patches extracted from one image,  $H$  is the image width,  $W$  is the image height,  $h$  is the proposed patches width, and  $W$  is the proposed patches height.

Every image in the PlantVillage dataset, which was originally sized at  $256 \times 256$  pixels, needs to be split into a set of patches that are  $32 \times 32$  pixels in size. A quotient of 8 is obtained by dividing the image width (height) by the patch width (height). This indicates that each image is divided into a grid consisting of 8 rows and 8 columns, resulting in a total of 64 patches. A mathematical depiction of the process for this splitting is provided by Equations 4.3 and 4.4.

$$\begin{aligned} N_p &= (256/32) \times (256/32) \\ N_p &= 8 \times 8 \\ N_p &= 64 \end{aligned} \tag{4.3}$$

$$\begin{aligned} I_D &= \sum_{i=1}^8 \sum_{j=1}^8 P_{ij} \\ I_D &= \{P_{11}, P_{12}, \dots, P_{88}\} \end{aligned} \tag{4.4}$$

After the image is split, noise may be present in the dataset. To remove this noise, we calculate the percentage of black pixels in each patch based on the RGB color of each pixel, as described in Equation 4.5. Using this percentage, we determine whether or not to remove the patch as follows: First, completely black patches (100% black pixel percentage) are removed because they don't contain any leaf parts. Second, we establish a pixel percentage threshold to closely match the attributes of the PlantVillage dataset, especially the approximate black pixel percentage in the original image, which is around 50%. Patches with a black pixel percentage exceeding this threshold are also eliminated, as illustrated in Equation 4.6. This strategy ensures that the outcomes closely resemble the attributes of the original image, thereby maintaining consistency with the dataset. Figure 4.3 illustrates the detailed processes of image splitting and the removal of unnecessary patches from two leaves, grape and apple, infected with the Black rot disease.

$$\begin{aligned} B &= \sum_{i=1}^N \delta(R_i = 0 \wedge G_i = 0 \wedge B_i = 0) \\ B_P &= \frac{B}{TotalNumberofPixels} \times 100 \end{aligned} \tag{4.5}$$

Where  $B$  is the black pixel number in the patch,  $B_P$  is the black pixel percentage of the patch,  $N$  is the total image pixels number of,  $R_i$ ,  $G_i$ , and  $B_i$  are the red, green, and blue components of the  $i$ -th pixel, respectively.  $\delta()$  is the counter delta function, which is equal to 1 if the condition inside is true and 0 otherwise.

$$I_D = \{P_D \mid P_D \in I_D, \text{ and black pixels percentage } B_P \leq \text{selected black pixels percentage}\} \tag{4.6}$$

All split patches from the PlantVillage dataset underwent visual inspection to determine their class label. The labeling criteria were as follows: patches originally split from healthy leaves were labeled as healthy. Patches split from each disease-type leaf contained two types of patches: patches containing healthy leaf pieces without any signs or symptoms of the disease type, and

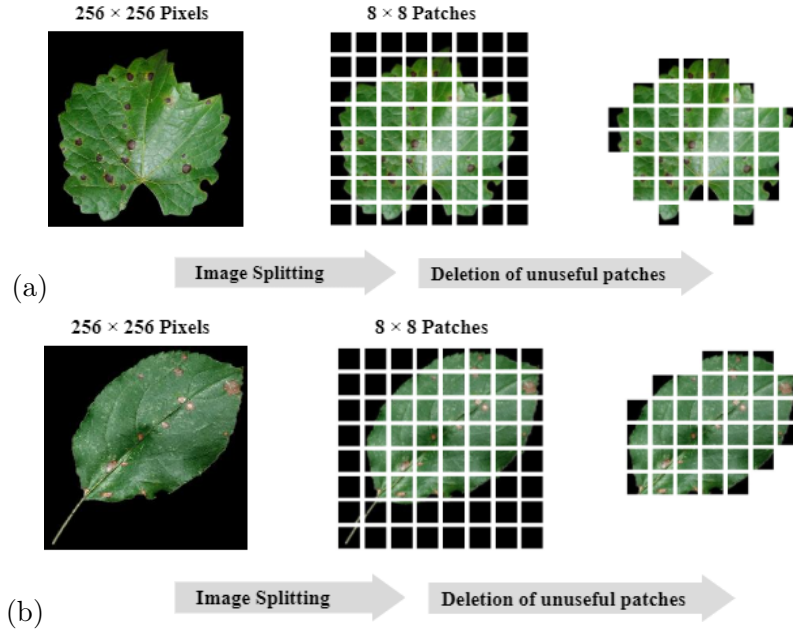


FIGURE 4.3: Image splitting examples : (a) Black rot infected grape leaf, (b) Black rot infected apple leaf.

patches containing infected leaf pieces presenting signs or symptoms of that disease type, as presented in Equation 4.7. The healthy patches were labeled as healthy, while the infected patches were labeled with the respective disease type of the original image from which the patch was extracted, as illustrated in Equations 4.8 and 4.9. An example of labeling of split patches extracted from two leaves, grape and apple leaves infected with the black rot disease, is presented in Figure 4.4. The infected patches extracted from the two leaves are identified as belonging to the black rot class, while the healthy patches extracted from both leaves are designated as the healthy class.

$$I_D = \{P_H \mid P_H \in I_D, \text{ and healthy}\} \cup \{P_D \mid P_D \in I_D, \text{ and Infected with diseas D}\} \quad (4.7)$$

$$H = \sum_{c=1}^{N\_c} \sum_{i=1}^{N\_i\_c} \{P_H \mid P_H \in I_{Di}, \text{ and healthy}\} \quad (4.8)$$

$$D = \sum_{c=1}^{N\_c} \sum_{i=1}^{N\_i\_c} \{P_D \mid P_D \in I_{Di}, \text{ and infected with disease D}\} \quad (4.9)$$

Where  $N\_c$  is the number of dataset classes,  $N\_i\_c$  is the number of class C images.

#### 4.2.3.2 The new PlantVillage dataset

Table 4.1 presents the outcomes of each class in the PlantVillage dataset after splitting all the images and removing all unnecessary patches.

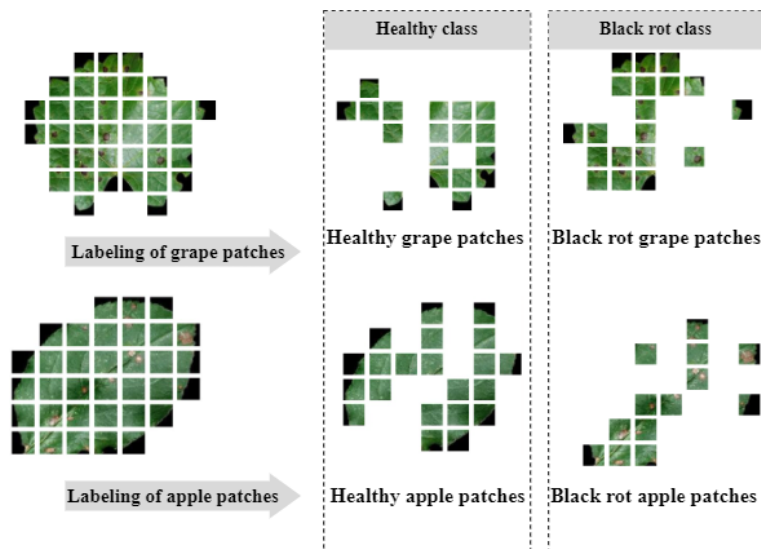


FIGURE 4.4: Data labeling examples : (a) Black rot infected grape leaf, (b) Black rot infected apple leaf.

TABLE 4.1: The PlantVillage dataset class details.

<i>Disease name</i>	<i>Crop name</i>	<i>Number of samples</i>	<i>Number of patches</i>	<i>Number of un-healthy patches</i>	<i>Number of healthy patches</i>
Bacterial spot	Peach	2297	60889	40187	20702
Bacterial spot	Pepper, bell	997	36250	19097	17153
Bacterial spot	Tomato	2127	72774	38222	34552
Black rot	Apple	621	20671	4992	15679
Black rot	Grape	1180	41174	18757	22417
Cedar apple rust	Apple	275	8104	5831	2273
Cercospora leaf spot gray leaf spot	Corn (maize)	513	25138	22738	2400
Common rust	Corn (maize)	1192	59798	56887	2911
Early blight	Potato	1000	37306	19097	17153
Early blight	Tomato	1000	30999	16850	14149
Esca (black measles)	Grape	1384	47236	27737	19499
Haunglonbing (Citrus greening)	Orange	5507	194626	194626	0
Healthy	Apple	1645	48693	0	48693
Healthy	Blueberry	1502	38080	0	38080
Healthy	Cherry (including sour)	854	28465	0	28465
Healthy	Corn (maize)	1162	74030	0	74030
Healthy	Grape	423	16136	0	16136



TABLE 4.1: The PlantVillage dataset class details.

<i>Disease name</i>	<i>Crop name</i>	<i>Number of samples</i>	<i>Number of patches</i>	<i>Number of un-healthy patches</i>	<i>Number of healthy patches</i>
Healthy	Peach	360	7081	0	7081
Healthy	Pepper, bell	1478	54876	0	54876
Healthy	Potato	152	5505	0	5505
Healthy	Raspberry	371	12723	0	12723
Healthy	Soybean	5090	190448	0	190448
Healthy	Strawberry	456	16904	0	16904
Healthy	Tomato	1591	48173	0	48173
Late bligh	Potato	1000	33730	16242	17488
Late blight	Tomato	1909	49721	28769	20952
Leaf mold	Tomato	952	22370	14596	7774
Leaf blight (Isariopsis leaf spot)	Grape	1075	43106	43106	0
Leaf scorch	Strawberry	1109	41062	39752	1310
North leaf blight	Corn (maize)	985	46769	40987	5782
Powdery mildew	Cherry (including sour)	105	35822	35822	0
Powdery mildew	Squash	1835	83457	83457	0
Septoria leaf spot	Tomato	1771	51360	34377	16983
Spider mites two spotted spider mite	Tomato	1667	41933	15110	26823
Target spot	Tomato	1404	44800	6086	38714
Tomato yellow leaf curl virus	Tomato	5357	155739	155739	0
Tomato mosaic virus	Tomato	373	7384	5209	2175
Scab	Apple	630	22984	10998	11986
Total	Total	54305	1860316	1014154	846162

The Table 4.2 illustrates the distribution details of the patches in the newly generated dataset after labeling. The dataset comprises a total of 1860316 patches, categorized into 21 classes, including 20 disease classes, each representing a specific disease type, and one healthy class. Among these, 8416162 patches belong to the healthy class, while the remaining 1014154 patches are distributed among the disease classes.

TABLE 4.2: The new PlantVillage dataset details.

<i>Class</i>	<i>Disease name</i>	<i>Crop name</i>	<i>Number of patches</i>	<i>Total number of class patches</i>
C0	Bacterial spot	Peach	40187	97506
		Pepper, bell	19097	
		Tomato	38222	
C1	Black rot	Apple	4992	23749
		Grape	18757	
C2	Cedar apple rust	Apple	5831	5831
C3	Cercospora leaf spot gray leaf spot	Corn (maize)	22738	22738
C4	Common rust	Corn (maize)	56887	56887
C5	Early blight	Potato	37980	54830
		Tomato	16850	
C6	Esca (black measles)	Grape	27737	27737
C7	Haunglonbing (citrus greening)	Orange	194626	194626
C8	Healthy	Apple	78631	846162
		Blueberry	38080	
		Cherry (including sour)	28465	
		Corn (maize)	85123	
		Grape	58052	
		Peach	27783	
		Pepper, bell	72029	
		Potato	26319	
		Raspberry	12723	
		Soybean	190448	
C9	Late blight	Potato	16242	45011
		Tomato	28769	
C10	Leaf mold	Tomato	14596	14596
C11	Leaf blight (isariopsis leaf spot)	Grape	43106	43106
C12	Leaf scorch	Strawberry	39752	39752
C13	North leaf blight	Corn (maize)	40987	40987

TABLE 4.2: The new PlantVillage dataset details.

<i>Class</i>	<i>Disease name</i>	<i>Crop name</i>	<i>Number of patches</i>	<i>Total number of class patches</i>
C14	Powdery mildew	Cherry (including sour)	35822	119279
		Squash	83457	
C15	Septoria leaf spot	Tomato	34377	34377
C16	Spider mites two spotted spider mite	Tomato	15110	15110
C17	Target spot	Tomato	6086	6086
C18	Tomato Yellow leaf curl virus	Tomato	155739	155739
C19	Tomato mosaic virus	Tomato	5209	5209
C20	Scab	Apple	10998	10998
Total			1860316	1860316

#### 4.2.3.3 Classification and assessment of diseases extent

We propose processing the image to detect diseases, as outlined in Section 4.2.3.1. The image is split into non-overlapping patches of size  $32 \times 32$  pixels, and all the unusful patches are eliminated. Each patch is predicted separately, and after predicting all the patches, the set of disease classes that infect the processed image is defined, as presented in Figure 4.5. Predicting each leaf patch separately allows us to determine the number of patches presenting each disease type, effectively determining the prevalence rate of each disease type and thus assessing the extent of all diseases affecting the leaf. As depicted in Equation 4.10, the prevalence rate ( $P_{Dr}$ ) for each disease ( $D$ ) is calculated by dividing the number of patches infected with the disease ( $D$ ) by the total number of patches. The overall extent of all diseases in the leaf ( $E$ ) is then determined by summing up the prevalence rates of all diseases infecting the leaf.

$$P_{Dr} = \frac{P_D * 100}{P_H + P_U} \quad (4.10)$$

$$E = \sum_{i=1}^n P_{Dr_i}$$

Where  $P_D$  is the number of patches infected with the disease type  $D$ ,  $P_U$  is the number of all infected patches and  $P_H$  is the number of healthy patches,  $i$  is the number of diseases infecting the leaf.

#### 4.2.4 CNN networks

The small Inception architecture was chosen as the CNN architecture in our proposed model. The detailed architecture of the model and the rationale for choosing this structure are outlined in Section 3.2.3. Two other CNN architectures, MiniVGGNet and LeNet5, were selected to

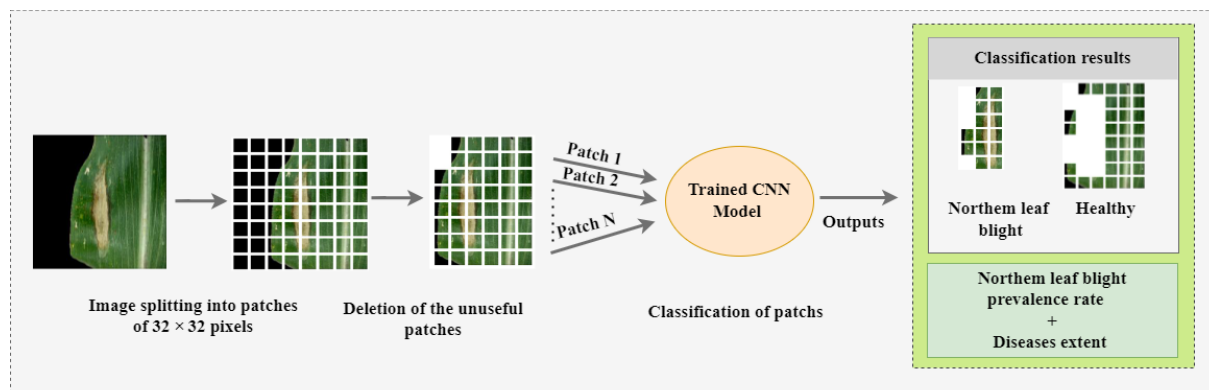


FIGURE 4.5: Flowchart of the prediction process and calculation of prevalence rate and extent of the diseases in the leaf.

perform two additional experiments with the new dataset. This enabled a comparison of the results obtained with these two CNN models with the results of the small Inception architecture, thereby evaluating the effectiveness of the smaller Inception architecture.

#### 4.2.5 Experimental setup

The three models were trained from scratch using a mechanism that involved extensive experimentation with various hyperparameter combinations. Through empirical testing, we identified the optimal parameter values for each model, aiming to achieve the best performance on our dataset while ensuring generalizability and preventing overfitting. Table 4.3 provides an overview of the explored hyperparameters and the selected values for the three models. We employed the ADAM optimization technique, utilizing its default parameters for all three models. Implementation was done in Python using the TensorFlow library, and experiments were conducted on Google Colab with GPU support.

TABLE 4.3: Hyperparameters configuration

<i>Hyperparameter</i>	<i>Values explored</i>	<i>Small Inception selected values</i>	<i>MiniVGGNet Selected values</i>	<i>LeNet5 selected values</i>
<i>Batch size</i>	16, 32, 64, 128, 256	32	32	32
<i>Epochs</i>	1,2,...,1000	451	505	200
<i>Learning rate</i>	0.00005, 0.00002, 0.00001, 0.0005, 0.0002, 0.0001, 0.005, 0.002, 0.001, 0.01	0.001	0.00002	0.0001

The dataset was split into two sets as follows: 80% of the dataset (1,488,261 images) was allocated for the training set, while the remaining 20% (372,055 images) was reserved for the testing and validation set.

## 4.3 Results and discussion

This section evaluates the proposed techniques, focusing on the generalization of multi-disease detection from the same leaf. Firstly, the results of the comparative experiment were assessed to demonstrate the performance of the employed CNN architectures. The evaluation was based on several classification performance metrics, including accuracy, loss function, and individual class accuracy. Secondly, to address imbalanced datasets and conduct a more comprehensive assessment of the model's performance, confusion matrices and individual class accuracy were utilized. Finally, the effectiveness of the model's performance was analyzed based on the evaluation process.

### 4.3.1 Results of the comparative experiments evaluating the performance of CNN models

Table 4.4 presents the training outcomes of the chosen CNNs. Figures 4.6, 4.7, and 4.8 illustrate the evolution of accuracy and loss function on the training set for all the different models. Initially, all CNN models exhibit rapid progress during the training phase. However, as the number of epochs increases, the convergence rate slows down, eventually leading to optimal performance for each model. Similarly, the loss function initially decreases quickly, followed by a gradual decline until reaching the minimum.

Although the Small Inception model does not demonstrate the fastest convergence, it outperforms the other CNNs, achieving an accuracy exceeding 0.9181. Conversely, MiniVGGNet model fails to surpass the accuracy threshold of 0.8338 and exhibits slower convergence. LeNet5 model, while converging rapidly, achieves the lowest recognition accuracy, peaking at 0.8306.

In terms of the loss function, the Small Inception model outperforms both MiniVGGNet and LeNet5 models. Specifically, the Small Inception model achieves a minimum loss function value of 0.2385, while MiniVGGNet and LeNet5 models reach higher loss function results of 0.5177 and 0.5223, respectively.

TABLE 4.4: Training performances of the three CNN models

<i>CNN model</i>	<i>Accuracy</i>	<i>Loss</i>
<i>Small Inception</i>	0.9181	0.2385
<i>MiniVGGNet</i>	0.8338	0.5177
<i>LeNet5</i>	0.8306	0.5223

The classification performance of the three CNN models on the testing set is displayed in Table 4.5. It shows that while the LeNet5 and MiniVGGNet models perform similarly, the Small Inception model significantly outperforms them. The Small Inception model achieved a test accuracy above 0.9057 and minimized the loss function to 0.2954. In contrast, the accuracy of the MiniVGGNet model was 0.8024 with a minimized loss of 0.6006, while LeNet5 achieved an accuracy of 0.8072 with a minimized loss of 0.6099.

Table 4.6 presents a comprehensive overview of the overall accuracies achieved for each disease class across all three CNN architectures. Small Inception stands out as the top performer, exhibiting the highest accuracies across all disease classes. The robust performance of Small

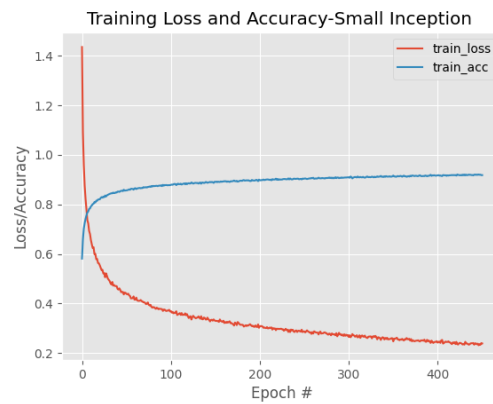


FIGURE 4.6: Evolution of loss and accuracy on the training set for Small Inception

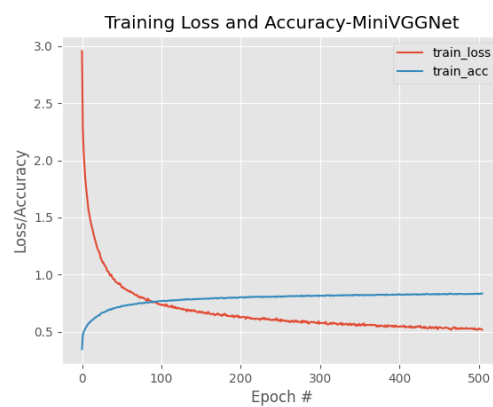


FIGURE 4.7: Evolution of loss and accuracy on the training set for MiniVGGNet

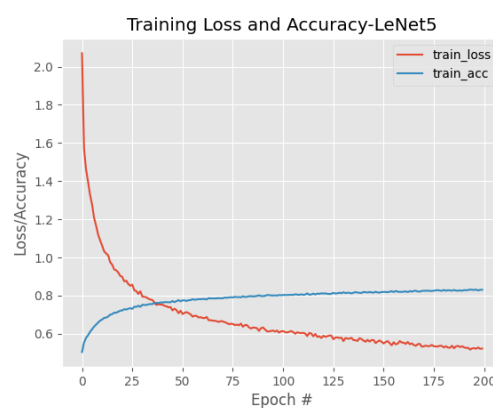


FIGURE 4.8: Evolution of loss and accuracy on the training set for LeNet5

Inception is particularly notable, as it demonstrates notable disparities when compared to the accuracies achieved by the other two CNN architectures. These results demonstrate the effectiveness and superior discriminative capabilities of the Small Inception model in accurately classifying instances across a diverse range of disease classes within the dataset.

In summary, the evaluation results highlight the Small Inception model as the top performer among the three CNN models. Its higher accuracy and superior performance in identifying

multi-disease infections on the same leaf validate its effectiveness in our proposed system.

TABLE 4.5: Testing performances of the three CNN models

<i>CNN model</i>	<i>Accuracy</i>	<i>Loss</i>
<i>Small Inception</i>	0.9057	0.2954
<i>MiniVGGNet</i>	0.8024	0.6006
<i>LeNet5</i>	0.8072	0.6099

TABLE 4.6: Accuracies obtained for individual disease classes using the employed CNNs.

<i>Class</i>	<i>Disease name</i>	<i>Small Inception</i>	<i>MiniVGGNet</i>	<i>LeNet5</i>	<i>Small Inception with class weight</i>
C0	Bacterial spot	80.87	64.33	61.39	84.56
C1	Black rot	73.97	56.00	36.87	78.73
C2	Cedar apple rust	74.37	39.67	42.45	86.97
C3	Cercospora leaf spot Gray leaf spot	67.04	37.40	55.24	73.30
C4	Common rust	93.21	85.53	86.80	92.07
C5	Early blight	76.97	48.21	50.63	75.46
C6	Esca (Black Measles)	82.82	70.58	73.80	89.52
C7	Haunglongbing (Citrus greening)	93.72	84.86	74.80	91.20
C8	Healthy	96.03	94.25	95.03	91.87
C9	Late blight	67.54	48.11	52.42	72.38
C10	Leaf Mold	73.25	39.89	41.55	83.56
C11	Leaf blight (Isariopsis Leaf Spot)	89.13	75.55	77.33	86.47
C12	Leaf scorch	91.18	73.77	78.44	96.05
C13	Northern Leaf Blight	78.65	69.33	54.30	80.65
C14	Powdery mildew	93.04	67.89	74.82	92.88
C15	Septoria leaf spot	71.75	57.09	57.48	80.64
C16	Spider mites Two spotted spider mite	73.36	32.46	35.27	74.32
C17	Target Spot	45.89	20.11	18.73	59.60
C18	Tomato Yellow Leaf Curl Virus	93.54	81.04	78.82	91.89
C19	Tomato mosaic virus	73.89	42.51	41.97	79.84
C20	Scab	77.45	40.04	43.11	85.81

### 4.3.2 Results of imbalanced dataset handling and performance analysis of the proposed method

To analyze the performance of the proposed method in recognizing disease symptoms from patches, each class's results and failures have been analyzed separately. Based on the individual class accuracies of the Small Inception model presented in Table 4.6 and the top values of the confusion matrix presented in Figure 4.9, it is evident that the results are biased for some classes, indicating the clear impact of class imbalance. In the following sections, we delve into the selected solution to address the dataset's imbalance and thoroughly examine the final outcomes of the proposed system, focusing on its performance.

#### 4.3.2.1 Imbalanced dataset handling results

To mitigate the challenge of the dataset imbalanced, a widely employed algorithmic technique involves incorporating class weights during model training. This strategy is presented by assigning specific weights to each of the 21 classes represented in the dataset. Classes with a larger proportion of images are assigned lower weights, while classes with a smaller representation are assigned higher weights. Consequently, the CNN model prioritizes the accurate prediction of instances from the minority classes, thereby addressing the imbalance in the dataset. The Small Inception model was trained with class weights, following the same rigorous experimental setup and methodology. To implement class weights effectively during model training, the `class_weight` module from `sklearn.utils` library was employed. This module facilitates the computation of class weights based on the distribution of class labels in the dataset, ensuring a balanced representation during training.

As depicted in the confusion matrix presented in Figure 4.9, the utilization of class weights is observed to have a good impact on the performance of the model. By assigning specific weights to each class, the model demonstrates a significant improvement in its ability to recognize and classify the symptoms associated with various diseases. This approach effectively addresses the challenge of severe class imbalance, allowing the model to allocate more attention to the accurate prediction of instances from minority classes. Consequently, the model's overall performance is notably enhanced, resulting in more reliable and robust disease recognition and classification.

#### 4.3.2.2 Performance analysis of the proposed method

To thoroughly assess the performance of the new system (Small Inception with class weights), we conducted a detailed analysis of the results and failures for each class. This involved studying the misclassification errors derived from the confusion matrix (bottom values in Figure 4.9). The analysis uncovered the following findings:

**Bacterial spot class (C0):** The presence of Bacterial spot disease manifests as spots on the leaves, initially appearing as light green and potentially transitioning to yellow or brown hues over time. In some cases, patches containing small, light green, and yellow spots are misclassified as healthy or mistaken for another disease exhibiting similar symptoms.

**Black rot class (C1):** The manifestation of brown spots is a common symptom of black rot disease. However, these spots can be mistaken for those induced by Esca (Black Measles) disease due to their similar appearance. This resemblance occasionally results in challenges when



True	C0	15773 16491	53 85	7 52	7 27	14 24	397 391	4 19	323 153	1968 952	166 252	37 53	13 24	29 128	30 60	112 132	215 331	12 7	35 64	281 203	0 2	26 52
	C1	282 197	3514 3740	32 44	0 2	4 1	39 30	193 310	2 2	553 231	25 42	2 7	11 14	20 61	5 6	8 13	25 33	2 1	15 5	4 1	0 0	14 10
	C2	46 25	12 2	868 1015	7 5	8 4	11 6	1 0	15 2	70 25	47 24	2 1	0 0	3 2	19 7	10 13	2 2	0 0	0 0	4 6	0 0	42 28
	C3	50 11	0 1	5 5	3049 3334	107 75	4 7	0 0	7 5	119 30	20 15	0 2	2 2	17 19	1144 1030	21 11	1 1	0 0	0 0	2 0	0 0	0 0
	C4	36 25	1 3	10 25	125 214	10606 10476	6 5	1 0	3 0	172 60	12 16	0 0	0 0	3 15	399 532	3 5	0 2	0 0	0 0	0 0	0 0	1 0
	C5	608 558	7 13	7 29	0 14	3 1	8441 8275	5 11	61 26	875 466	447 681	90 152	15 23	19 110	7 14	55 70	186 356	13 18	33 53	81 71	0 1	13 24
	C6	63 37	272 187	1 1	0 1	1 1	28 18	4595 4967	3 3	443 140	21 30	3 7	31 19	46 106	0 1	5 8	18 11	9 4	3 3	3 2	0 0	3 2
	C7	109 439	0 2	2 24	0 9	0 2	17 53	0 0	36485 35501	1867 2121	38 123	16 37	7 11	0 1	5 24	90 149	1 22	4 9	0 4	282 366	0 1	3 28
	C8	771 2051	99 385	18 189	51 160	142 203	352 722	113 444	1180 1455	162527 155487	341 850	161 678	34 78	49 252	166 531	883 1622	161 524	710 1126	56 273	1173 1444	81 230	165 529
	C9	293 246	3 9	10 57	3 22	34 27	640 478	4 10	115 47	1068 581	6081 6517	177 228	18 13	18 73	35 109	116 115	125 245	20 16	12 20	189 113	0 1	42 76
	C10	39 45	0 0	2 7	0 0	0 0	44 28	0 0	18 9	312 107	70 81	2139 2440	4 5	3 5	0 3	13 22	34 37	36 14	0 1	195 100	0 0	11 16
	C11	51 41	26 21	1 2	0 1	0 0	97 77	5 20	36 11	386 255	40 73	14 11	7685 7456	112 519	2 2	39 37	36 36	50 28	0 2	35 18	2 0	4 10
	C12	143 33	15 8	4 4	4 10	8 2	94 34	16 16	0 2	186 49	45 34	3 11	61 11	7250 7637	16 13	20 13	55 52	6 1	2 1	6 1	0 0	17 19
	C13	50 26	0 0	3 16	797 977	386 319	9 6	0 0	20 11	398 147	39 29	0 5	0 0	12 21	6448 6612	19 11	4 3	4 3	0 0	4 5	0 0	5 7
	C14	99 126	0 3	7 36	0 22	26 15	35 50	2 0	78 51	1088 913	73 161	10 37	5 8	5 19	6 18	22197 22159	7 21	4 5	1 1	183 134	0 1	31 75
	C15	454 271	6 9	3 10	0 0	0 3	339 213	2 8	33 23	564 180	223 189	42 88	12 17	28 83	1 10	13 23	4934 5545	55 23	60 89	76 56	11 6	20 30
	C16	10 14	0 0	0 0	0 0	0 0	11 14	0 1	1 1	665 503	6 23	11 36	2 5	0 1	0 0	5 9	7 28	2217 2246	12 41	69 86	5 11	1 3
	C17	128 110	3 7	0 0	0 1	0 0	50 51	4 1	6 0	281 119	32 47	10 13	0 1	0 2	0 0	0 2	85 102	45 15	559 726	18 14	0 0	0 4
	C18	164 303	0 2	0 16	0 1	0 0	38 70	0 2	395 345	1115 1090	56 141	47 206	3 12	0 7	2 7	74 147	16 41	75 73	1 0	29138 28624	18 38	6 23
	C19	1 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	201 147	0 0	0 1	1 0	0 0	0 0	0 2	1 10	15 6	0 0	53 44	770 832	0 0
	C20	48 38	9 6	12 22	1 2	0 0	20 7	1 1	9 2	278 126	33 30	4 3	2 1	6 16	4 3	43 32	10 12	2 1	0 0	14 10	0 0	1704 1888
		C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
		Predicted																				

FIGURE 4.9: Comparison of confusion matrices from the Small Inception CNN: unbalanced (top values) vs. balanced (bottom values)

distinguishing between the two diseases, particularly when dealing with patches that exhibit smaller spots.

***Cedar apple rus class (C2)***: Patches showing small spots and initial signs of discoloration, characteristic of these diseases, were incorrectly categorized as other diseases sharing similar symptoms, such as scab, healthy, late blight, and bacterial spot.

***Cercospora leaf spot Gray leaf spot (C3)***: The notable resemblance among the symptoms seen in the patches exhibiting small disease spots affected by cercospora leaf spot, gray leaf spot, and northern leaf blight, results in the misclassification of patches initially identified as cercospora leaf spot or gray leaf spot, which are erroneously labeled as northern leaf blight.

**Common rust class (C4):** The occurrence of small brown spots, a shared symptom between common rust and northern leaf blight diseases, resulted in misidentifying common rust patches with these spots as northern leaf blight infections.

**Early blight class (C5):** The symptoms characteristic of early blight disease involve the appearance of lesions on leaves, accompanied by browning or yellowing of the affected leaf tissue, along with leaf curling. These manifestations resemble the symptoms of late blight disease, potentially causing confusion between the two conditions. Furthermore, the occurrence of small yellowing areas within the patches can exacerbate the misclassification, labeling them as healthy.

**Esca (Black Measles) class (C6):** Similarly to black rot disease, some patches with small spots were misclassified as black rot due to the similarity of the brown spots caused by both diseases.

**Huanglongbing (Citrus greening) class (C7):** The yellowing of leaves is a prominent symptom of huanglongbing, which can be difficult to distinguish from the normal variation in leaf color. This ambiguity surrounding the yellow blotchy mottling of leaves has led to uncertainty and misidentification between healthy patches and those impacted by huanglongbing disease.

**Healthy class (C8):** Misinterpretation of patches displaying a light green hue is the most frequent misclassification error encountered in the healthy class. This confusion stems from the fact that several diseases manifest symptoms such as leaf yellowing, and in certain instances, the early stages of yellowing may resemble a light green color.

**Late blight class (C9):** As previously discussed, the symptoms of both early blight and late blight diseases exhibit remarkable similarity, which can lead to identical misclassification errors.

**Leaf Mold class (C10):** Leaf mold typically manifests with initial symptoms of leaf yellowing. Instances where this yellowing is not distinctly visible or apparent may have been misclassified as healthy.

**Leaf blight (Isariopsis leaf spot) class (C11):** The presence of leaf wilting, accompanied by browning and yellowing of leaves, may result in the misclassification of certain patches as leaf scorch disease. This misidentification arises due to the similarity between the symptoms observed in these patches and those characteristic of leaf scorch disease, leading to confusion and failure in classification.

**Leaf scorch class (C12):** The existence of small yellowing patches can indeed lead to misclassifying Septoria leaf spot as a healthy condition. This misinterpretation may arise because the small yellow patches might not be clearly distinguishable from patches affected by Septoria leaf spot and those considered healthy.

**Northern Leaf Blight class (C13):** The model encountered similar challenges when classifying patches affected by northern leaf blight and cercospora leaf spot gray leaf spot disease. In several instances, the model failed to differentiate between the two diseases and misclassified several patches as cercospora leaf spot gray leaf spot disease.

**Powdery mildew class (C14):** Powdery mildew results in the development of white powdery spots on the leaves. However, patches with relatively small and indistinct symptoms were misclassified. Extracting disease characteristics from these patches proved challenging, leading to their classification as healthy.

**Septoria leaf spot class (C15):** The resemblance in symptoms between Septoria leaf spot and late blight or early blight, including the presence of lesions on the leaves and the browning or yellowing of affected tissue, results in the misclassification of Septoria leaf spot patches as late blight or early blight.

**Spider mites Two spotted spider mites class (C16):** are recognized by the appearance of small yellow or white spots on the leaves, which can lead to leaf yellowing and browning. However, patches containing only small spots are often categorized as healthy due to the difficulty in recognizing these subtle symptoms.

**Target Spot class (C17):** The quality of the image dataset proved crucial in distinguishing this disease class. Leaves captured in the image dataset often exhibited unclear symptoms, particularly those with small spots. This contributed to the system's failure in recognizing a significant number of patches in this class.

**Tomato Yellow Leaf Curl Virus class (C18):** This disease may occasionally cause subtle changes that are difficult to identify, like yellowing. This can make it difficult for the model to recognize and categorize those patches as Tomato Yellow Leaf Curl Virus infected. Because of this, there may be misclassifications in which the model considers certain patches as healthy.

**Tomato mosaic virus class (C19):** The initial yellowing symptoms of tomato mosaic virus infection can sometimes appear similar to light green, leading to the misclassification of the patches as healthy.

**Scab class (C20):** The appearance of small, olive-green to black lesions or spots on the leaves is one of the symptoms of this disease. However, some of the patches that contain small olive-green spots are misclassified as healthy.

### 4.3.3 Discussion

Numerous deductions can be made from the aforementioned findings. First, the results of the comparative experiments highlight the superiority of the Small Inception model in terms of classification performance among the three CNN architectures evaluated (Small Inception, MiniVG-GNet, and LeNet5). Specifically, the accuracy achieved by the Small Inception model surpassed that of the other models. Additionally, the model exhibited exceptional proficiency in accurately classifying the patches, as evidenced by its ability to minimize the loss function to the lowest possible value. As a result, the strong performance of the Small Inception architecture validates our choice and underscores its suitability for our proposed system.

Second, the incorporation of class weights played a fundamental role in enhancing the models performance. Assigning weights to each class according to its relative imbalance within the dataset enabled the model to learn more efficiently and differentiate between different classes. This adaptive strategy effectively addressed the challenges arising from severe class imbalance, ultimately leading to more precise and dependable classification outcomes.

The detailed examination of the failures made by the Small Inception model with class weights aimed to analyze the misclassification errors for each individual class. Through this analysis, we deduced that A significant cause of misclassification by the classifier is attributed to patches exhibiting small spots and lesions associated with disease infection. The classifier encountered challenges in effectively recognizing and precisely classifying such patches. Another significant factor contributing to misclassification was the classifier's difficulty in distinguishing between healthy

patches and those displaying early stages of yellowing symptoms caused by various diseases. This challenge arose because both healthy patches and those affected by the diseases exhibited a similar light green coloration. Furthermore, some diseases exhibit overlapping symptoms, such as similar patterns of spots, lesions, or discoloration. This similarity between symptoms added complexity for the classifier in accurately distinguishing between different diseases.

In conclusion, despite facing challenges and limitations, the proposed agricultural solution remains adaptable and highly advantageous. Its capacity to detect multiple disease infections within a single leaf, regardless of the crop type, is especially beneficial in practical farming settings characterized by diverse crops and simultaneous disease threats. In such contexts, the system plays a crucial role in promptly identifying diseases and effectively managing them.

## 4.4 Conclusion

This chapter presents an innovative approach utilizing CNN to identify multiple diseases on the same leaf across different crop types. The method autonomously identifies the symptoms of each disease without interference from other diseases affecting the same leaf. To achieve this, the leaf is split into numerous patches, and each patch is analyzed individually. This segmentation process allows the model to accurately classify each disease based on its corresponding leaf patch. By adopting this approach, the proposed method effectively overcomes the limitations of traditional methods, which attempt to identify multiple diseases from the entire leaf.

To validate the effectiveness of our approach, we conducted experiments with three distinct CNN models: Small Inception, MiniVGGNet, and LeNet5, using the PlantVillage dataset. Results clearly demonstrated the superiority of the Small Inception architecture over the other two CNNs. The Small Inception model achieved the highest accuracy and minimized the loss function, showcasing its proficiency in recognizing leaf patches. Furthermore, we implemented class weights to address the significant class imbalance in the dataset. This technique significantly enhanced the model's ability to discern and classify different classes, leading to improved overall performance.

Overall, our proposed method, which integrates leaf splitting, the Small Inception architecture, and class weights, successfully identifies individual disease symptoms within leaf patches. It offers the advantage of generalizing disease detection across various leaf crop types and disease categories, even those not present in the training data.

---

---

## CHAPTER 5

---

# INTELLIGENCE SYSTEM FOR DETECTING DISEASES IN APPLE TREE BRANCHES

---

## Chapter contents

---

5.1	Introduction . . . . .	103
5.2	Materials and methods . . . . .	104
5.3	Results and discussion . . . . .	111
5.4	Conclusion . . . . .	117

---

## 5.1 Introduction

In this chapter, we propose an intelligent system based on AI to detect diseases on apple tree branches. The main objectives of our proposed approach are to develop a system that can automatically detect and segment disease spots on apple tree branches using machine learning and image processing techniques. However, to design a system capable of identifying diseases on apple tree branches, several issues must be addressed.

### 5.1.1 Research questions and hypotheses

#### RQ1. *Why apple tree branches?*

Apple trees hold significant economic value globally, contributing significantly to the agricultural and food industry. Apples rank among the most popular and widely consumed fruits in numerous countries, making them highly sought after in both local and international markets. However, apple trees are susceptible to various diseases, which can impede their growth and productivity. Early and accurate detection of these diseases is crucial for implementing rapid and effective measures to prevent their spread and minimize losses for farmers. Traditionally, agricultural experts manually detect diseases affecting apple crops, a process that is time-consuming, costly, and prone to human errors. In such circumstances, AI emerges as a viable solution. Leveraging advancements in machine learning, image processing, and deep learning algorithms, it becomes feasible to develop AI systems capable of automatically and accurately detecting disease apple plant.

While many proposed disease detection systems primarily focus on leaf diseases [13, 31, 36, 55, 68, 85, 101, 103, 179, 190] because leaves are typically the first site where plant diseases appear, others concentrate on detecting fruit diseases [1, 23, 45, 78, 100, 112, 155, 172, 176, 177]. However, it's worth noting that diseases affecting branches also present significant challenges. Despite this, research on branch diseases remains relatively scarce. To address this gap, our work focuses on detecting common apple branch diseases, such as *Nectria cinnabarina*, characterized by the development of orange cankers on branches.

#### RQ2. *What is the chosen machine learning approach for our system?*

The GMM is proposed because a single-model disease model encounters difficulties when confronted with issues like image acquisition noise, changes in lighting conditions, and the existence of multiple surfaces for a given pixel, all at once. Put differently, a unimodal model would face challenges in accurately depicting the intricate and varied features of images. This deficiency underscores the requirement for a more advanced approach, such

as the multimodal GMM [184], which has the capability to more effectively capture the complex features of disease-related variations in images.

Based on the research questions, we have established criteria for the proposed system. The system must focus on identifying diseases from trees branches rather than solely from the leaves, with a specialization in apple trees due to their significant economic importance (RQ1). It should be built upon a machine learning approach, specifically utilizing the GMM model, given its capability to capture the multifaceted nature of disease-related variations in images (RQ2). The remaining chapters are organized as follows: Section 5.2 describes the materials and proposed system. The results obtained are analyzed in Section 5.3. Section 5.4 concludes our chapter.

## 5.2 Materials and methods

### 5.2.1 Dataset

The dataset utilized in this contribution was generated by us. This decision was driven by the absence of a suitable dataset comprising a significant number of images presenting diseased branches of apple trees. The dataset was collected from various internet sources, comprising 500 images depicting different cases of apple branches infected with *Nectria cenabrina* disease. The images displayed diversity in both the color shades of the disease spots and the backgrounds. The disease spots exhibited various color shades, including light yellow, orange, dark orange, light red, and dark red. Meanwhile, the backgrounds varied, ranging from blank or single-color backgrounds to those containing leaves and branches. The Tables 5.1 and 5.2 provide a summary of the distinctions observed among the images within our dataset.

TABLE 5.1: Number of images per background variant

<i>Image Type</i>	<i>Total</i>	<i>Train</i>	<i>Test</i>
<i>Trees and leaves Background</i>	116	80	36
<i>Empty background or other</i>	384	264	120
<i>Total</i>	500	344	156

TABLE 5.2: Number of images per disease variant

<i>Image Type</i>	<i>Total</i>	<i>Train</i>	<i>Test</i>
<i>Dark orange</i>	230	170	60
<i>Light orange</i>	100	60	40
<i>Dark red</i>	60	40	20
<i>Light red</i>	80	50	30
<i>Yellow</i>	30	24	6
<i>Total</i>	500	344	156

### 5.2.2 Proposed system

The proposed system for detecting *Nectria cenabrina* from images of apple tree branches begins with image acquisition, followed by preprocessing to enhance image quality and precision, thus facilitating classification. Subsequently, relevant information extraction employs classification based entirely on GMM, allowing the generation of the segmentation mask by determining the probability distribution of the disease color. The final step involves disease spot segmentation based on the created segmentation mask. Figure 5.1 illustrates the proposed system.

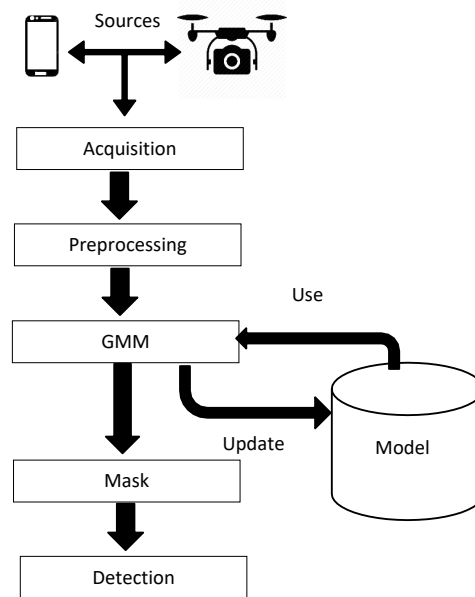


FIGURE 5.1: General diagram of the disease spot detection process.

#### 5.2.2.1 Acquisition

The input data for the proposed system comprises RGB images captured using either a smartphone or a camera affixed to a surveillance drone. Whether acquired from the ground using a smartphone or remotely via a surveillance drone, these images facilitate the study and diagnosis of diseases in apple tree branches, providing a comprehensive and detailed perspective of the situation.

#### 5.2.2.2 Preprocessing

During the preprocessing phase, each image undergoes a series of operations, as illustrated in Figure 5.2, aimed at enhancing the images and uncovering essential information required for detecting elements of interest. Following the application of these preprocessing operations, the images are prepared for analysis and detection, thereby maximizing the precision and reliability of our system. These operations encompass filtering techniques, contrast enhancement, noise reduction, among other advanced methods. Through the integration of these preprocessing steps, we achieve images that are ready for analysis and detection, thereby maximizing the accuracy and reliability of our system.



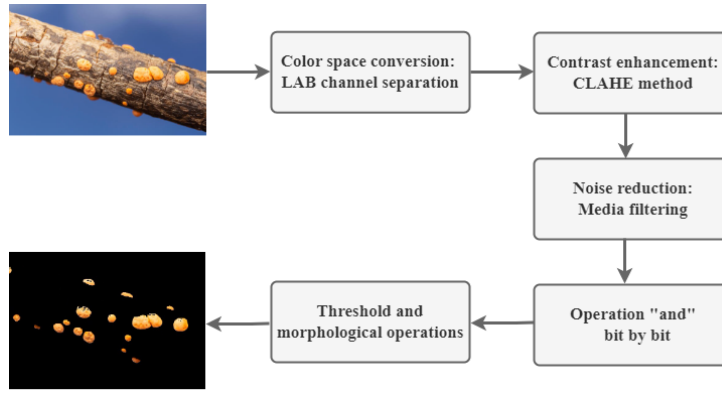


FIGURE 5.2: The proposed preprocessing process.

The preprocessing phase consists of the following sequence of operations:

#### 5.2.2.2.1 Conversion of color space

The images color spaces were converted from RGB to LAB to represent colors in a more perceptually uniform manner. The LAB color space comprises three components: L for luminosity, A for the green-magenta component, and B for the blue-yellow component. This conversion enables the extraction of relevant color features by segregating luminosity and color information. LAB is less susceptible to lighting variations compared to RGB [51], thereby enhancing the robustness of image processing. Therefore, this conversion is effective for tasks disease detection. The conversion of images from the RGB color space to the LAB color space is accomplished using the following formula:

$$\begin{aligned}
 L &= 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \\
 a &= (R - L) \cdot 0,5 + 128 \\
 b &= (B - L) \cdot 0,5 + 128
 \end{aligned}
 \tag{5.1}$$

Where  $R$ ,  $G$ , and  $B$  denote the red, green, and blue RGB image color channels, and  $L$ ,  $A$ , and  $B$ , represent the LAB image corresponding channels.

#### 5.2.2.2.2 Contrast enhancement

The contrast of the image was enhanced using Contrast Limited Adaptive Histogram Equalization (CLAHE) [120] applied to the luminance channel L within the LAB color space. This technique aims to improve image quality by increasing the local contrast while minimizing noise. The CLAHE algorithm works by dividing the image into small regions, called tiles, and equalizing the histogram of each tile adaptively. This means that histogram equalization is performed locally on each region, taking into account the specific characteristics of that region. By limiting the enhanced contrast to a predefined level, the CLAHE algorithm preserves the structural details of the image while avoiding excessive noise amplification. The following pseudo-code outlines the general steps of the contrast enhancement:

1. Divide the image into fixed-size tiles.
2. Compute the histogram of each tile.
3. Equalize the histogram using an adaptive transfer function for each tile.
4. Perform bilinear interpolation between values of neighboring tiles to avoid edge artifacts.

5. Reassemble the equalized tiles to obtain the final image with enhanced contrast.

Applying CLAHE to the L channel of the LAB color space enhances the visual quality of the image by highlighting details and increasing the perception of local luminance variations. This preprocessing step helps improve the accuracy of disease detection by highlighting important features in the interest regions.

#### **5.2.2.2.3 Noise reduction**

The noise in the "a" and "b" channels of the LAB image was reduced through median filtering. This approach entails substituting pixel values with the medians of adjacent pixels, thus diminishing unwanted fluctuations. By estimating the current pixel's value relative to neighboring pixels, this nonlinear method eliminates outliers and sudden variations, thereby conserving image contours and vital details. The following pseudo-code outlines the general steps of the median filtering:

1. Traverse each pixel of the image in the "a" and "b" channels.
2. Each pixel, extracts the neighborhood around the pixel using a defined window size.
3. Sort the values of neighboring pixels in ascending order.
4. Select the median value from the sorted values.
5. Replace the value of the pixel being processed with the obtained median value.
6. Repeat the above steps for all pixels in the image.

Applying median filtering to LAB images aids in noise reduction, preserving crucial color and texture details while improving overall image quality. This procedure substantially enhances disease detection accuracy by mitigating undesired variations.

#### **5.2.2.2.4 Threshold and morphological operations**

Thresholding is applied to the LAB image's B channel to generate a binary image. In this image, pixels surpassing a specific threshold are identified as probable blue areas, indicating the potential presence of *Nectria cinnabarina* disease. The following pseudo-code outlines the general steps of the threshold operation:

1. Traverse each pixel of the image in the B channel.
2. Compare the pixel value with a predefined threshold.
3. If the pixel value is greater than the threshold, assign the value 1 (blue area) to that pixel in the binary image; otherwise, assign the value 0 (non-blue area).
4. Repeat the above steps for all pixels in the image.

After generating the binary image, morphological operations are employed to improve detection quality. The closing operation connects disjointed blue areas, resulting in more robust shapes, while the opening operation eliminates small gaps within the blue regions, leading to smoother and more uniform shapes. The following pseudo-code outlines the typical procedures involved in morphological operations:

1. Apply morphological closing to the binary image.
2. Use a structuring element (e.g., a circular-shaped kernel) to expand and connect blue areas.
3. Repeat this operation for a predefined number of iterations.
4. Apply morphological opening to the resulting image from closing.
5. Use a structuring element to remove small holes within the blue areas.
6. Repeat this operation for a predefined number of iterations.

#### 5.2.2.2.5 Operation "AND" bit-by-bit

To emphasize the region of interest, the "AND" bit-by-bit operation is executed between the binary image and the original image. This process retains only the binary blue pixels while masking out the rest of the processed image. Below are the general steps of the "AND" bit-by-bit operation outlined in pseudo-code:

1. Traverse each pixel of the processed image and the binary image.
2. Perform a "AND" bit-by-bit operation between the pixel values corresponding to both images.
3. Assign the resulting value to that pixel in a new image that will serve as a mask.
4. Repeat the above steps for all pixels in both images.

The "AND" bit-by-bit operation generates a masked image that accentuates blue pixel areas present in the binary image, thereby facilitating the detection of *Nectria cinnabarina*. This approach improves the accuracy of disease detection by prioritizing pertinent regions.

#### 5.2.2.3 Gaussian Mixture Model

Figure 5.3 illustrates the stages involved in the GMM. The following steps outline the procedures involved in implementing the GMM model:

##### 5.2.2.3.1 Initialization

First, the intensity of each pixel ( $X_t$ ) in the LAB color space is utilized for its description. Subsequently, the probability distribution of the pixel's value is computed. At the onset of the disease detection process, the distributions of  $K$  Gaussians are initialized with low weight, high variance, and a predetermined mean for each pixel. We utilized multiple images for Gaussian mixture learning to acquire a stable model.

##### 5.2.2.3.2 Parameters update

Model updating was done using the K-Means approximation algorithm for possible real-time applications. Equation 5.2 illustrates how each value of a new pixel is evaluated using the  $k$  Gaussian distributions to observe if the distance between the pixel and each Gaussian is less than the distribution's standard deviation.

$$\frac{|P_t - \mu_i|}{\sigma_i} < 2.5 \quad (5.2)$$

In the event that none of the distributions meet the requirements of Equation 5.2, the pixel is assigned to the branch instead of the disease. The mean, variance, and weight of the current pixel are substituted for the parameters of the least probable Gaussian using Equations 5.3, 5.4, and 5.5.

$$\mu_k = X_t \quad (5.3)$$

$$\sigma_k^2 = \text{a large initial variance} \quad (5.4)$$

$$\omega_k = \text{low weight} \quad (5.5)$$

The following equation is used to update the weights of all Gaussian distributions:

$$\omega_{k,t} = (1 - \alpha) \cdot \omega_{k,t-1} + \alpha M_{k,t} \quad (5.6)$$

where  $\alpha$  is the learning coefficient that controls the model's rate of adaptation. For the Gaussian corresponding to the disease,  $M_{k,t}$  is 1; for other Gaussian, it is 0. A normalization step follows weight updating to guarantee that the total of the weights is always equal to 1.

The other parameters of the distributions, as specified in Equation 5.2, are updated using Equations 5.7, 5.8, and 5.9.

$$\mu_{k,t} = (1 - \varphi_k) \cdot \mu_{k,t-1} + \varphi_k \cdot P_t \quad (5.7)$$

$$\sigma_{k,t}^2 = (1 - \varphi_k) \cdot \sigma_{k,t-1}^2 + \varphi_k (P_t - \mu_{k,t})^T (P_t - \mu_{k,t}) \quad (5.8)$$

Where:

$$\varphi_t = \alpha \cdot \eta (P_t | \mu_k, \sigma_k) \quad (5.9)$$

#### 5.2.2.4 Creation of the mask

Based on the value  $w_{k,t}/\sigma_{k,t}$ , the distributions are sorted to ascertain if  $X_t$  represents a disease. Since the disease color tends to stay relatively constant, this order is based on the assumption that a disease pixel corresponds to a high weight with low variance. The disease is represented by the first  $B$  distributions that satisfy Equation 5.10:

$$B = \text{arg min}_b \sum_{k=1}^b \omega_{k,t} > T \quad (5.10)$$

The threshold  $T$  denotes the lowest percentage of the overall weight that is assigned to the disease model. When a low  $T$  value is selected, the disease turns unimodal. When a scene has both dynamic and static areas, the model's use of a single threshold  $T$  results in poor classification. In a dynamic environment, a higher threshold improves classification, but it also leads to inaccurate disease detections.

### 5.2.2.5 Detection

The binary masks obtained for each component are resized to align with the original image dimensions. Subsequently, these resized binary masks are utilized to mask the original image through a logical AND operation. This process yields a masked image that accentuates areas affected by *Nectria cinnabarina*. The resulting masked image can be visually inspected and analyzed to evaluate the detection results.

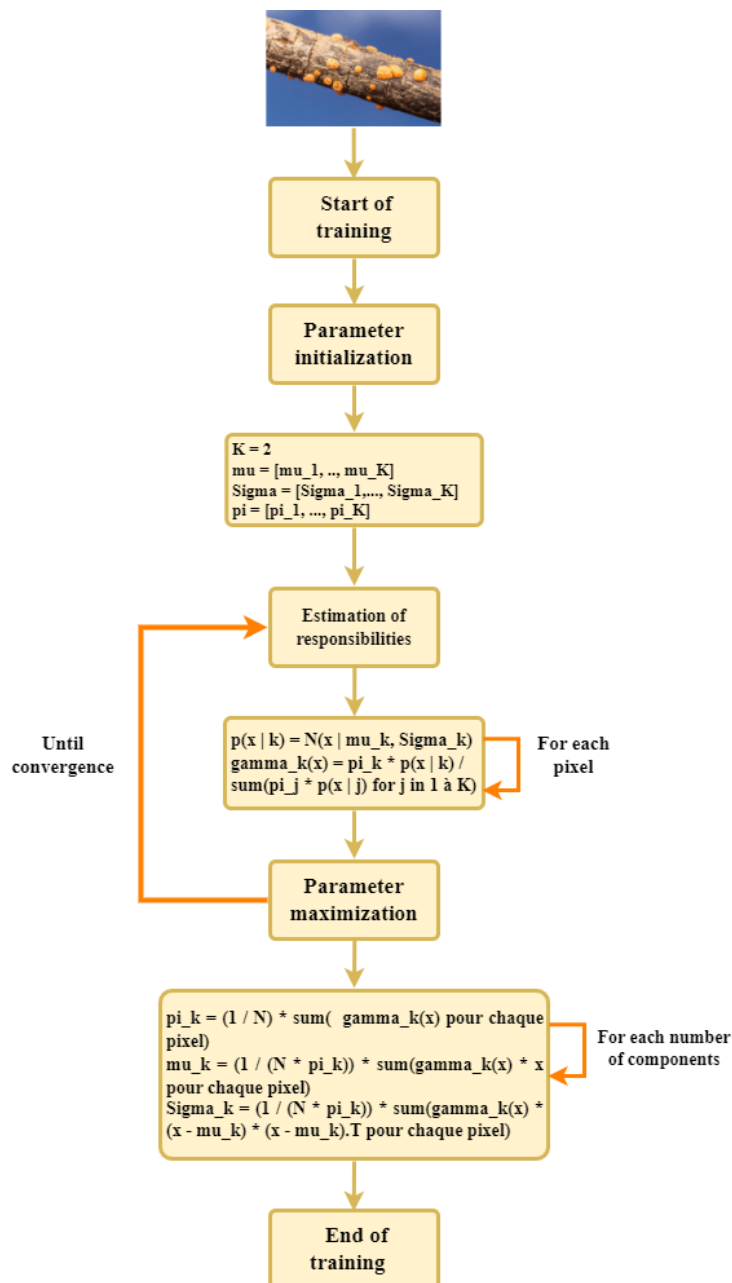


FIGURE 5.3: GMM stages.

### 5.2.3 Experimental setup

We conducted extensive testing and empirically adjusted the model configuration parameters to achieve the best possible results. After thorough analysis, we determined the values presented in Table 5.3 to optimize the proposed system.

TABLE 5.3: System parameters

<i>Parameters</i>	<i>Values</i>
<i>The number of components of the mixture</i>	2
<i>The type of covariance</i>	full
<i>The convergence threshold</i>	1e-3
<i>Non-negative regularization added to the covariance diagonal</i>	1e-6
<i>The number of initializations to perform</i>	1
<i>The method used to initialize the weights, means, and accuracies</i>	Kmeans
<i>Number of iterations performed before the next print</i>	10

For the parameters of the other methods, we used the default values from the Python APIs. The experiments were performed on a laptop computer running Windows 10, equipped with an i7 7700HQ processor operating at 2.80 GHz and 8 GB of DDR4 RAM. We implemented the algorithms using the Python language in both Jupyter and Google Colab development environments.

## 5.3 Results and discussion

### 5.3.1 Qualitative results

In this section, we will highlight the qualitative and visual aspects of the results achieved by the proposed system, as detailed in Table 5.4.

TABLE 5.4: Results of *Nectria cinnabarina* disease detection in apple tree branches for selected test images





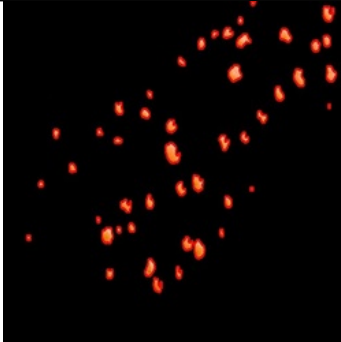


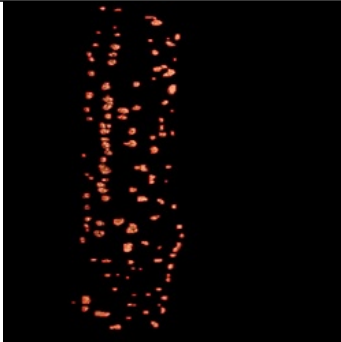


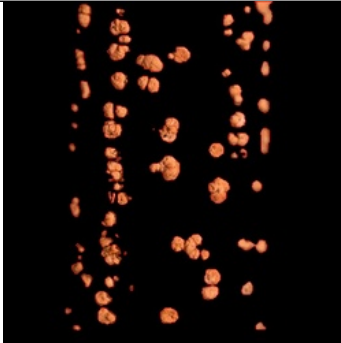

#	Original image	Ground truth	Detection
1			
2			
3			
4			

TABLE 5.4: Results of *Nectria cinnabarina* disease detection in apple tree branches for selected test images


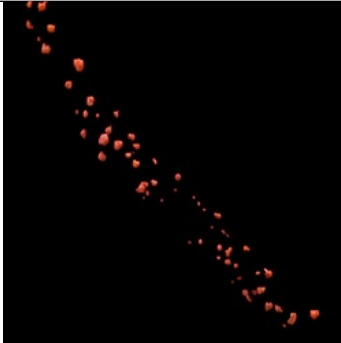



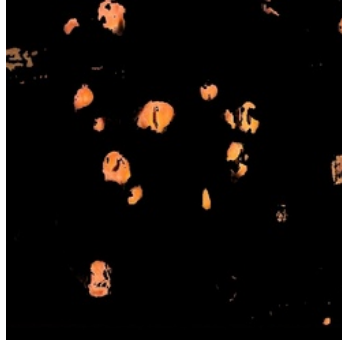

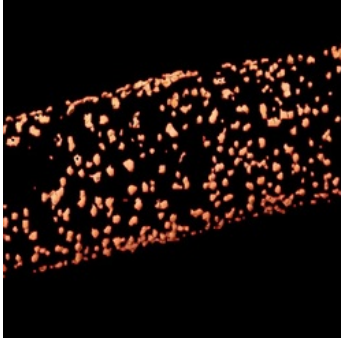
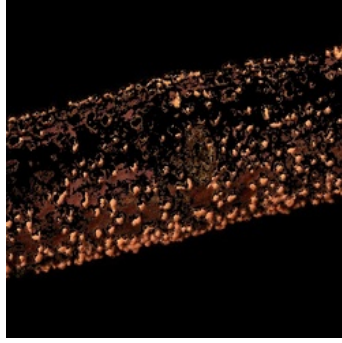
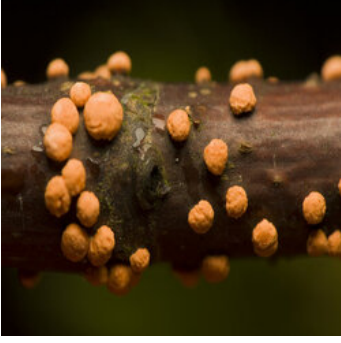

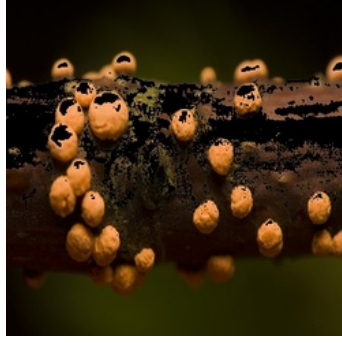
#	<i>Original image</i>	<i>Ground truth</i>	<i>Detection</i>
5			
6			
7			
8			



TABLE 5.4: Results of *Nectria cinnabarina* disease detection in apple tree branches for selected test images



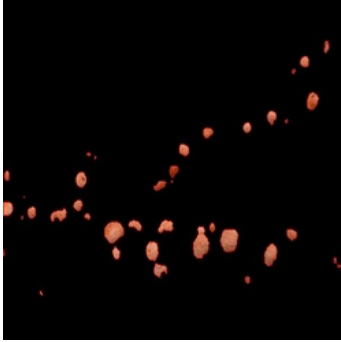
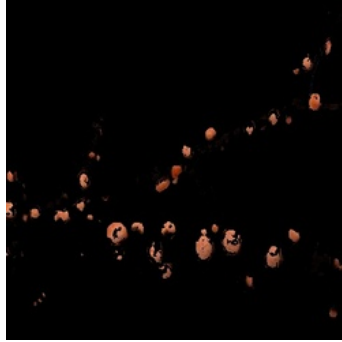

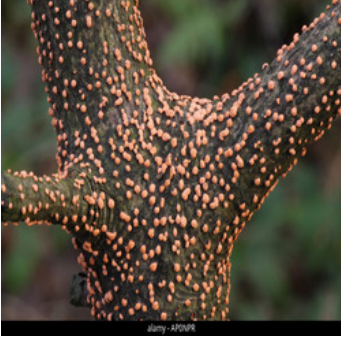
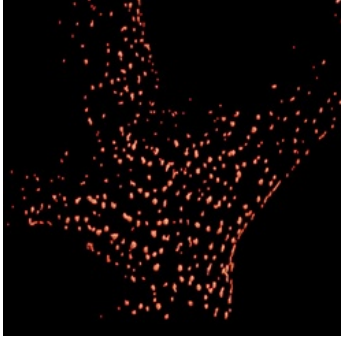
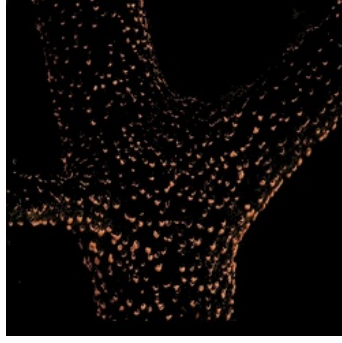
#	Original image	Ground truth	Detection
9			
10			
11			
12			

TABLE 5.4: Results of *Nectria cinnabarina* disease detection in apple tree branches for selected test images


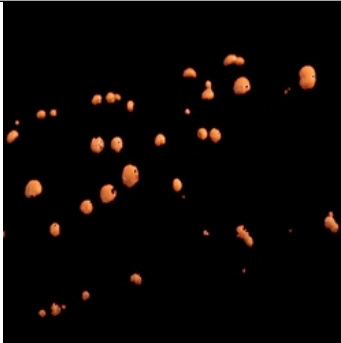
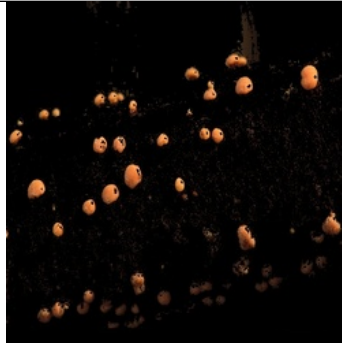




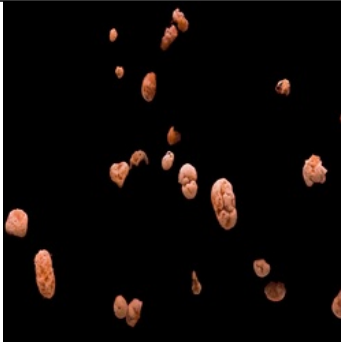


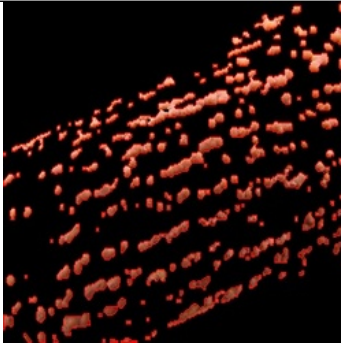
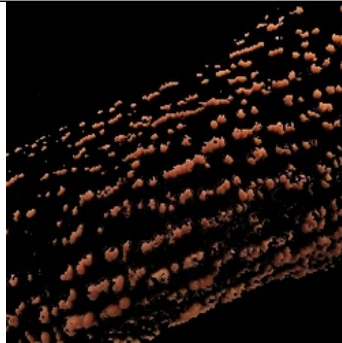

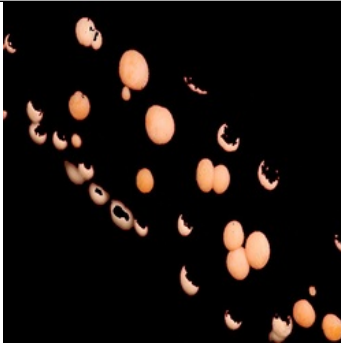



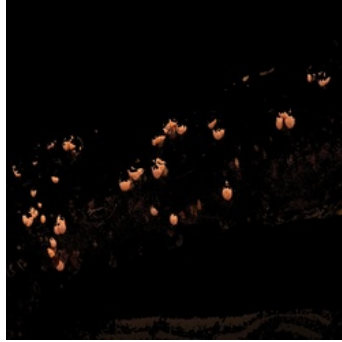
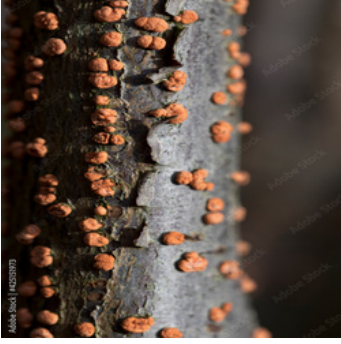

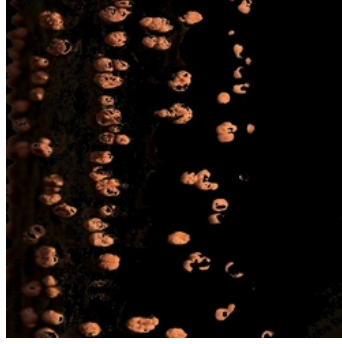
#	Original image	Ground truth	Detection
13			
14			
15			
16			

TABLE 5.4: Results of *Nectria cinnabarina* disease detection in apple tree branches for selected test images

#	<i>Original image</i>	<i>Ground truth</i>	<i>Detection</i>
17			
18			
19			

Certainly, the system effectively identified the disease in all test images, showcasing the robustness of the applied method. This proficiency stems from its capability to memorize the probability distribution of disease-representing pixels. Nevertheless, we observed instances where the system detected pixels beyond the diseased region, particularly especially in images presenting abundant vegetation.

This observation may be explained by a reduction in color distinctions when transitioning from the RGB image representation to the LAB image representation. In the RGB representation, when aiming to detect the color orange, we modify the values of the Red and Green channels, maintaining the Blue channel at zero. A common set of RGB values for generating orange is Red = 255, Green = 165, Blue = 0. Hence, it's valid to observe the presence of green hues within the detected orange color.

This could potentially cause The detection system could encounter confusion because certain plant parts might share color values similar to the color orange. This could clarify why pixels



outside the affected area are detected, particularly in areas with vegetation. It is imperative to take this limitation into account when assessing and interpreting the detection system's results.

### 5.3.2 Quantitative results

In this section, we will present the results in terms of F-score, recall, and precision to evaluate the performance of our apple tree branch disease detection system in Table 5.5. These metrics will allow us to quantify the effectiveness of our approach and identify areas that may require potential improvements.

TABLE 5.5: Results in terms of F1-score, recall, and precision for selected test images

#	<i>F1-score</i>	<i>Recall</i>	<i>Precision</i>
1	0.7990421455938698	0.7845387002727359	0.8140919293451742
2	0.7801719325667077	0.8515720204728248	0.7198187062216729
3	0.650347705146036	0.5089246843709185	0.9006163328197226
4	0.9366577906882918	0.893679267072138	0.9839789777061965
5	0.7502691065662002	0.6619183285849952	0.8658385093167702
6	0.8581730769230769	0.8521068576149821	0.8643262873638141
7	0.6115045526810232	0.46205089002948563	0.9038666951506088
8	0.7705454545454545	0.6440729483282674	0.9588235294117647
9	0.7088399605652317	0.7060556464811784	0.7116463213460904
10	0.7140992167101827	0.5723753051970701	0.9491035280508965
11	0.2976092333058532	0.1791563275434243	0.878345498783455
12	0.7486553578816715	0.6263412945655936	0.9303341902313624
13	0.6421428571428572	0.5461725394896719	0.7790294627383015
14	0.6818613485280152	0.7223340040241448	0.64568345323741
15	0.7759965022225461	0.8534220227600577	0.7114510956707643
16	0.37337461300309593	0.23399755524942276	0.923359620243473
17	0.6354403025391681	0.4762714609653385	0.9543979227523531
18	0.7845248685175648	0.6551202856621576	0.9776351614157915
19	0.7160828333655893	0.6441504178272981	0.8061002178649237
<i>Average (all test images)</i>	0.6792	0.6410	0.8018

## 5.4 Conclusion

This chapter delved into the utilization of AI for detecting diseases on apple tree branches. An intelligent AI-driven system was introduced, employing image analysis and machine learning

techniques to automatically identify disease symptoms on apple tree branches.

The outcomes from our system demonstrate its effectiveness in detecting diseases affecting apple trees, highlighting its potential to enhance fruit tree health and optimize agricultural yield. Through the integration of image processing and machine learning techniques, we developed a system adept at accurately identifying disease symptoms. This empowers farmers to receive timely alerts, enabling them to take proactive measures to prevent disease spread and minimize crop losses and fostering agricultural sustainability.

---

## CONCLUSION AND PERSPECTIVES

In our thesis, we employed AI techniques for detecting plant diseases, with a particular focus on utilizing machine learning and deep learning methods to identify plant diseases affecting leaves and branches depicted in crop and plantation images. Following this conclusion, our aim is to provide an overview of the contributions made in this thesis and to suggest potential future directions.

The first part of this thesis was dedicated to the state of the art on different AI techniques. This study allowed us to categorize AI techniques into machine learning and deep learning as fundamental components. ANNs were highlighted as the bridge between these concepts. The first chapter addressed this connection, while the second chapter provided an overview of plant pathology and recent developments in the field of plant disease through deep learning and machine learning methods. Subsequently, this examination of recent methodologies proposed in this area allowed us to extract the problems and challenges encountered by existing plant disease identification and classification systems in the literature.

The second part of this thesis introduces the proposed systems for plant disease detection, which aim to address the challenges and issues identified during the comparative study conducted in the first part. This section is divided into three chapters.

Based on the literature reviews, we have identified that the proposed systems lack robustness and generalization due to inadequate datasets. To address these challenges, in the third chapter, we proposed a plant disease detection system capable of recognizing healthy and unhealthy leaves regardless of crop pieces and disease type. The aim of this system is to generalize plant disease detection across all plant pieces and disease types. This is achieved by splitting the images into small patches, each containing a small leaf piece. This step removes crop characteristics and emphasizes disease-related features from the patches. This allows the system to utilize the extracted features from the training process, initially extracted on very specific types of crops and diseases, to be extended to all leaves and disease types, even those not seen during the training phase. Additionally, by predicting each leaf patch separately from the whole leaf, we can estimate the percentage of unhealthy parts of the leaves, thereby determining the extent of disease on the leaf. To demonstrate the effectiveness of the proposed system, it was evaluated using a set of seen crop and disease types during the training process, as well as a set of unseen crop and disease types during the training process. The results demonstrated the performance of our system in both sets, outperforming state-of-the-art methods.

The second challenge identified is that plants can become infected with multiple diseases simultaneously during their life cycle. In the fourth chapter, we propose a system capable of recognizing these simultaneous infections across different plant types. This is achieved by isolating each disease symptom separately from others infecting the same leaf and minimizing crop-related characteristics. The key to this capability lies in splitting the leaf into small patches, each containing individual disease symptoms and devoid of crop characteristics. This allows the system to extract features specific to each disease and extend these features to recognize multi-disease occurrences across all crop types, even those not seen during the training phase. Additionally, predicting each leaf patch separately enables the determination of the number of patches infected with a specific disease, effectively determining the prevalence rate of each disease type and assessing the extent of all disease types affecting the leaf. Several experiments were conducted to improve the performance of the proposed system. The results of these experiments were then used to evaluate the system's performance, demonstrating the effectiveness of the proposed approach in identifying individual disease symptoms within leaf patches rather than using the entire leaf.

Another issue not addressed in the literature is the detection of diseases on tree branches, which we tackled in the fifth chapter by proposing a system capable of detecting such diseases. This system employs a combination of image processing and machine learning techniques to automatically identify disease symptoms on tree branches. The image processing techniques include color space conversion, contrast enhancement, various filtering techniques, thresholding, and morphological operations such as closing and opening operations. Additionally, the "AND" bit-by-bit operation is performed between the binary image and the original image, enhancing the images and revealing essential information necessary for detecting elements of interest. In terms of machine learning, we utilized a GMM to calculate the probability distribution, enabling the creation of a segmentation mask used for the detection and segmentation of disease spots. The model was trained and evaluated using images of tree branches infected with *Nectria cinnabarina* disease, and the results of the evaluation demonstrated its adeptness at accurately identifying disease symptoms.

However, certain challenges hinder the system's optimal performance. One of these challenges is its inability to adapt to environmental conditions in real-world scenarios. Therefore, our future research aims to ensure the model's performance quality in real-world situations and enhance its adaptability to various environmental conditions. Additionally, for the deep learning approach for cross-crop plant disease detection and the deep learning approach for simultaneous multi-disease detection on the same leaf, instead of employing our current splitting method, a sliding window approach has the potential to generate multiple patches, thereby enhancing the system's disease detection capabilities. Indeed, for the intelligent system for detecting diseases in apple tree branches, future endeavors are essential to further enhance the performance and accuracy of disease detection, advancing the system's robustness and expanding its capability to detect a broader range of diseases on a wide range of tree branches. This includes exploring new datasets and conducting comparisons with deep learning algorithms.

---

# BIBLIOGRAPHY

- [1] ABD EL-AZIZ, A. A., DARWISH, A., OLIVA, D., AND HASSANIEN, A. E. Machine learning for apple fruit diseases classification system. In *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)* (2020), Springer, pp. 16–25.
- [2] ABIODUN, O. I., JANTAN, A., OMOLARA, A. E., DADA, K. V., MOHAMED, N. A., AND ARSHAD, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 4, 11 (2018).
- [3] ABRAHAM, T. H. (physio) logical circuits: The intellectual origins of the mcculloch–pitts neural networks. *Journal of the History of the Behavioral Sciences* 38, 1 (2002), 3–25.
- [4] ADUWO, J. R., MWEBAZE, E., AND QUINN, J. A. Automated vision-based diagnosis of cassava mosaic disease. In *ICDM (Workshops)* (2010), pp. 114–122.
- [5] AGARWAL, M., SINGH, A., ARJARIA, S., SINHA, A., AND GUPTA, S. Toled: Tomato leaf disease detection using convolution neural network. *Procedia Computer Science* 167 (2020), 293–301.
- [6] AHA, D. W., KIBLER, D., AND ALBERT, M. K. Instance-based learning algorithms. *Machine learning* 6 (1991), 37–66.
- [7] AHMAD, I., HAMID, M., YOUSAF, S., SHAH, S. T., AND AHMAD, M. O. Optimizing pretrained convolutional neural networks for tomato leaf disease detection. *Complexity* 2020 (2020), 1–6.
- [8] AHMAD, W., SHAH, S., AND IRTAZA, A. Plants disease phenotyping using quinary patterns as texture descriptor. *KSII Transactions on Internet and Information Systems (TIIS)* 14, 8 (2020), 3312–3327.
- [9] AHMED, M., SERAJ, R., AND ISLAM, S. M. S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics* 9, 8 (2020), 1295.
- [10] AKSHAR JOSHI, G. M. Artificial intelligence. In *ICWET 10: Proceedings of the International Conference and Workshop on Emerging Trends in Technology* (10), p. 1023.



- [11] ALKHAYRAT, M., ALJNIDI, M., AND ALJOUMAA, K. A comparative dimensionality reduction study in telecom customer segmentation using deep learning and pca. *Journal of Big Data* 7, 1 (2020), 9.
- [12] ALPAYDIN, E. *Introduction to machine learning*. MIT press, 2004.
- [13] ALQETHAMI, S., ALMTANNI, B., ALZHRANI, W., AND ALGHAMDI, M. Disease detection in apple leaves using image processing techniques. *Engineering, Technology & Applied Science Research* 12, 2 (2022), 8335–8341.
- [14] ALZUBI, J., NAYYAR, A., AND KUMAR, A. Machine learning from theory to algorithms: an overview. In *Journal of physics: conference series* (2018), vol. 1142, IOP Publishing, p. 012012.
- [15] ALZUBI, J., NAYYAR, A., AND KUMAR, A. Machine learning from theory to algorithms: an overview. In *Journal of physics: conference series* (2018), vol. 1142, IOP Publishing, p. 012012.
- [16] AMIN, H., DARWISH, A., HASSANIEN, A. E., AND SOLIMAN, M. End-to-end deep learning model for corn leaf disease classification. *IEEE Access* 10 (2022), 31103–31115.
- [17] ANNABEL, L. S. P., ANNAPOORANI, T., AND DEEPALAKSHMI, P. Machine learning for plant leaf disease detection and classification—a review. In *2019 international conference on communication and signal processing (ICCSP)* (2019), IEEE, pp. 0538–0542.
- [18] ARIVAZHAGAN, S., SHEBIAH, R. N., ANANTHI, S., AND VARTHINI, S. V. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. *Agricultural Engineering International: CIGR Journal* 15, 1 (2013), 211–217.
- [19] AZIZ, S., BASHIR, M., MUGHAL, O., KHAN, M. U., AND KHAN, A. Image pattern classification for plant disease identification using local tri-directional features. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (2019), IEEE, pp. 0973–0978.
- [20] BARBEDO, J. G. A. Plant disease identification from individual lesions and spots using deep learning. *Biosystems engineering* 180 (2019), 96–107.
- [21] BARBEDO, J. G. A., KOENIGKAN, L. V., HALFELD-VIEIRA, B. A., COSTA, R. V., NECHET, K. L., GODOY, C. V., JUNIOR, M. L., PATRICIO, F. R. A., TALAMINI, V., CHITARRA, L. G., ET AL. Annotated plant pathology databases for image-based detection and recognition of diseases. *IEEE Latin America Transactions* 16, 6 (2018), 1749–1757.
- [22] BEDI, P., AND GOLE, P. Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artificial Intelligence in Agriculture* 5 (2021), 90–101.
- [23] BENLACHMI, Y., EL AIREJ, A., AND HASNAOUI, M. L. Fruits disease classification using machine learning techniques. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)* 10, 4 (2022), 917–929.
- [24] BHATIA, A., CHUG, A., AND SINGH, A. P. Hybrid svm-lr classifier for powdery mildew disease prediction in tomato plant. In *2020 7th International conference on signal processing and integrated networks (SPIN)* (2020), IEEE, pp. 218–223.
- [25] BI, C., WANG, J., DUAN, Y., FU, B., KANG, J.-R., AND SHI, Y. Mobilenet based apple leaf diseases identification. *Mobile Networks and Applications* 27 (2020), 172–180.

- [26] BOUACIDA, I., FAROU, B., DJAKHDJAKHA, L., SERIDI, H., AND KURULAY, M. Innovative deep learning approach for cross-crop plant disease detection: A generalized method for identifying unhealthy leaves. *Information Processing in Agriculture* (2024).
- [27] BOUACIDA, I., FAROU, B., SERIDI, H., AND TOUAHRI, M. A. An artificial intelligence-based system for detecting diseases in apple tree branches. In *2023 International Conference on Electrical Engineering and Advanced Technology (ICEEAT)* (2023), vol. 1, pp. 1–7.
- [28] BREIMAN, L. Random forests mach learn 45 (1): 5–32, 2001.
- [29] BREIMAN, L., FRIEDMAN, J., OLSHEN, R., AND STONE, C. Classification and regression trees—crc press. *Boca Raton, Florida* (1984).
- [30] CESSIE, S. L., AND HOUWELINGEN, J. V. Ridge estimators in logistic regression. *Journal of the Royal Statistical Society Series C: Applied Statistics* 41, 1 (1992), 191–201.
- [31] CHAKRABORTY, S., PAUL, S., AND RAHAT-UZ ZAMAN, M. Prediction of apple leaf diseases using multiclass support vector machine. In *2021 2Nd international conference on robotics, electrical and signal processing techniques (ICREST)* (2021), IEEE, pp. 147–151.
- [32] CHEN, J., CHEN, J., ZHANG, D., SUN, Y., AND NANEHKARAN, Y. Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture* 173 (2020), 105393.
- [33] CHEN, M., CHALLITA, U., SAAD, W., YIN, C., AND DEBBAH, M. Artificial neural networks-based machine learning for wireless networks: A tutorial. *IEEE Communications Surveys & Tutorials* 21, 4 (2019), 3039–3071.
- [34] CHOLLET, F. *Deep learning with Python*. Simon and Schuster, 2018.
- [35] CHOWDHURY, M. E., RAHMAN, T., KHANDAKAR, A., AYARI, M. A., KHAN, A. U., KHAN, M. S., AL-EMADI, N., REAZ, M. B. I., ISLAM, M. T., AND ALI, S. H. M. Automatic and reliable leaf disease detection using deep learning techniques. *AgriEngineering* 3, 2 (2021), 294–312.
- [36] CHUANLEI, Z., SHANWEN, Z., JUCHENG, Y., YANCUI, S., AND JIA, C. Apple leaf disease identification using genetic algorithm and correlation based feature selection method. *International Journal of Agricultural and Biological Engineering* 10, 2, 74–83.
- [37] DASTRES, R., AND SOORI, M. Artificial neural network systems. *International Journal of Imaging and Robotics (IJIR)* 21, 2 (2021), 13–25.
- [38] DE LIMA, E. S., AND FEIJÓ, B. Artificial intelligence in human-robot interaction. *Emotional Design in Human-Robot Interaction: Theory, Methods and Applications* (2019), 187–199.
- [39] DECHANT, C., WIESNER-HANKS, T., CHEN, S., STEWART, E. L., YOSINSKI, J., GORE, M. A., NELSON, R. J., AND LIPSON, H. Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology* 107, 11 (2017), 1426–1432.
- [40] DECHTER, R. Learning while searching in constraint-satisfaction problems.
- [41] DEMILIE, W. B. Plant disease detection and classification techniques: a comparative study of the performances. *Journal of Big Data* 11, 1 (2024), 5.

- [42] DENG, X., LAN, Y., HONG, T., AND CHEN, J. Citrus greening detection using visible spectrum imaging and c-svc. *Computers and Electronics in Agriculture* 130 (2016), 177–183.
- [43] DICTIONARY, O. Artificial intelligence. *Dostupno na: [https://www.lexico.com/definition/artificial\\_intelligence](https://www.lexico.com/definition/artificial_intelligence), (7. 7. 2020)* (2023).
- [44] DOMINGOS, P. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books, 2015.
- [45] DUBEY, S. R., AND JALAL, A. S. Detection and classification of apple fruit diseases using complete local binary patterns. In *2012 Third International Conference on Computer and Communication Technology* (2012), IEEE, pp. 346–351.
- [46] DUNN, J. C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.
- [47] EBRAHIMI, M., KHOSHTAGHAZA, M. H., MINAEI, S., AND JAMSHIDI, B. Vision-based pest detection based on svm classification method. *Computers and Electronics in Agriculture* 137 (2017), 52–58.
- [48] ELFATIMI, E., ERYIGIT, R., AND ELFATIMI, L. Beans leaf diseases classification using mobilenet models. *IEEE Access* 10 (2022), 9471–9482.
- [49] EUNICE, J., POPESCU, D. E., CHOWDARY, M. K., AND HEMANTH, J. Deep learning-based leaf disease detection in crops using images for agricultural applications. *Agronomy* 12, 10 (2022), 2395.
- [50] FAN, X., ZHOU, J., AND XU, Y. Recognition of field maize leaf diseases based on improved regional convolutional neural network. *Journal of South China Agricultural University* 41, 6 (2020), 82–91.
- [51] FAROU, B., SERIDI, H., AND AKDAG, H. Using gaussian mixture models and hsv color space for background subtraction. In *CONTROL ENGINEERING TECHNOLOGIES CONFERENCE* (2013), p. 135.
- [52] FAROU, B., SERIDI, H., AND AKDAG, H. Improved gaussian mixture model with background spotter for the extraction of moving objects. *Int. Arab J. Inf. Technol.* 13, 6A (2016), 807–816.
- [53] FERNÁNDEZ-CABALLERO, A. Contribution of fuzziness and uncertainty to modern artificial intelligence. *Fuzzy Sets Syst.* 160 (2009), 129.
- [54] GHOSH, S., AND DUBEY, S. K. Comparative analysis of k-means and fuzzy c-means algorithms. *International Journal of Advanced Computer Science and Applications* 4, 4 (2013).
- [55] GONG, X., AND ZHANG, S. A high-precision detection method of apple leaf diseases using improved faster r-cnn. *Agriculture* 13, 2 (2023), 240.
- [56] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. *Deep learning*. MIT press, 2016.
- [57] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., AND BENGIO, Y. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

- [58] GOVARDHAN, M., AND VEENA, M. Diagnosis of tomato plant diseases using random forest. In *2019 Global Conference for Advancement in Technology (GCAT)* (2019), IEEE, pp. 1–5.
- [59] GUO, X., FAN, T., AND SHU, X. Tomato leaf diseases recognition based on improved multi-scale alexnet. *Transactions of the Chinese Society of Agricultural Engineering* 35, 13 (2019), 162–169.
- [60] HAREL, D., AND MARRON, A. The human-or-machine matter: Turing-inspired reflections on an everyday issue (brief summary of paper). *Bridging the Gap Between AI and Reality*, 7.
- [61] HE, K., GKIOXARI, G., DOLLÁR, P., AND GIRSHICK, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2961–2969.
- [62] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.
- [63] HINTON, G. E. Learning multiple layers of representation. *Trends in cognitive sciences* 11, 10 (2007), 428–434.
- [64] HINTON, G. E., AND SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.
- [65] HM GOVERNMENT. *Industrial Strategy: Building a Britain fit for the future*. 2017.
- [66] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [67] HOSSAIN, E., HOSSAIN, M. F., AND RAHAMAN, M. A. A color and texture based approach for the detection and classification of plant leaf disease using knn classifier. In *2019 international conference on electrical, computer and communication engineering (ECCE)* (2019), IEEE, pp. 1–6.
- [68] HOU, J., YANG, C., HE, Y., AND HOU, B. Detecting diseases in apple tree leaves using fpn–isresnet–faster rcnn. *European Journal of Remote Sensing* 56, 1 (2023), 2186955.
- [69] HU, Z., YANG, H., HUANG, J., AND XIE, Q. Fine-grained tomato disease recognition based on attention residual mechanism. *Journal of South China Agricultural University* 40, 6 (2019), 124–132.
- [70] HUGHES, D. P., AND SALATHE, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060* (2015).
- [71] ISLAM, M., DINH, A., WAHID, K., AND BHOWMIK, P. Detection of potato diseases using image segmentation and multiclass support vector machine. In *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)* (2017), IEEE, pp. 1–4.
- [72] ISLAM, S., ELMEKKI, H., ELSEBAI, A., BENTAHAR, J., DRAWEL, N., RJOUB, G., AND PEDRYCZ, W. A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems with Applications* (2023), 122666.

- [73] JADHAV, S. B., UDUP, V. R., AND PATIL, S. B. Soybean leaf disease detection and severity measurement using multiclass svm and knn classifier. *International Journal of Electrical and Computer Engineering* 9, 5 (2019), 4092.
- [74] JANIESCH, C., ZSCHECH, P., AND HEINRICH, K. Machine learning and deep learning. *Electronic Markets* 31, 3 (2021), 685–695.
- [75] JIANG, F., LU, Y., CHEN, Y., CAI, D., AND LI, G. Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Computers and Electronics in Agriculture* 179 (2020), 105824.
- [76] JIANG, P., CHEN, Y., LIU, B., HE, D., AND LIANG, C. Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access* 7 (2019), 59069–59080.
- [77] JO, T. *Machine learning foundations*. Springer, 2021.
- [78] JOLLY, P., AND RAMAN, S. Analyzing surface defects in apples using gabor features. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (2016), IEEE, pp. 178–185.
- [79] JOSHI, A. A., AND JADHAV, B. Monitoring and controlling rice diseases using image processing techniques. In *2016 International Conference on Computing, Analytics and Security Trends (CAST)* (2016), IEEE, pp. 471–476.
- [80] KAMAL, K., YIN, Z., WU, M., AND WU, Z. Depthwise separable convolution architectures for plant disease classification. *Computers and Electronics in Agriculture* 165 (2019), 104948.
- [81] KELLEHER, J. D. *Deep learning*. MIT press, 2019.
- [82] KENCHAKKANAVAR, A. Y. Exploring the artificial intelligence tools: Realizing the advantages in education and research. *Journal of Advances in Library and Information Science* 12, 4 (2023), 218–224.
- [83] KHALID, M., SARFRAZ, M. S., IQBAL, U., AFTAB, M. U., NIEDBALA, G., AND RAUF, H. T. Real-time plant health detection using deep convolutional neural networks. *Agriculture* 13, 2 (2023), 510.
- [84] KHAN, A., NAWAZ, U., ULHAQ, A., AND ROBINSON, R. W. Real-time plant health assessment via implementing cloud-based scalable transfer learning on aws deeplens. *Plos one* 15, 12 (2020), e0243243.
- [85] KHAN, A. I., QUADRI, S., AND BANDAY, S. Deep learning for apple diseases: classification and identification. *International Journal of Computational Intelligence Studies* 10, 1 (2021), 1–12.
- [86] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [87] KOLLER, D., AND FRIEDMAN, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [88] KRENKER, A., BEŠTER, J., AND KOS, A. Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech* (2011), 1–18.

- [89] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25* (2012).
- [90] KURICHETI, G., AND SUPRIYA, P. Computer vision based turmeric leaf disease detection and classification: a step to smart agriculture. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (2019), IEEE, pp. 545–549.
- [91] KURZWEIL, R., RICHTER, R., KURZWEIL, R., AND SCHNEIDER, M. L. *The age of intelligent machines*, vol. 580. MIT press Cambridge, 1990.
- [92] LE, X.-H., HO, H. V., LEE, G., AND JUNG, S. Application of long short-term memory (lstm) neural network for flood forecasting. *Water 11*, 7 (2019), 1387.
- [93] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep learning. *nature 521*, 7553 (2015), 436–444.
- [94] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.
- [95] LEE, N., AND LEE, N. Artificial intelligence and data mining. *Counterterrorism and Cybersecurity: Total Information Awareness* (2015), 323–341.
- [96] LEE, S. H., GOËAU, H., BONNET, P., AND JOLY, A. New perspectives on plant disease characterization based on deep learning. *Computers and Electronics in Agriculture 170* (2020), 105220.
- [97] LI, J., LIN, L., TIAN, K., AND ALAA, A. A. Detection of leaf diseases of balsam pear in the field based on improved faster r-cnn. *Transactions of the Chinese Society of Agricultural Engineering 36*, 12 (2020), 179–185.
- [98] LI, K., LIN, J., LIU, J., AND ZHAO, Y. Using deep learning for image-based different degrees of ginkgo leaf disease classification. *Information 11*, 2 (2020), 95.
- [99] LI, L., ZHANG, S., AND WANG, B. Plant disease detection and classification by deep learning—a review. *IEEE Access 9* (2021), 56683–56698.
- [100] LI, T., FANG, W., ZHAO, G., GAO, F., WU, Z., LI, R., FU, L., AND DHUPIA, J. An improved binocular localization method for apple based on fruit detection using deep learning. *Information Processing in Agriculture 10*, 2 (2023), 276–287.
- [101] LI, X., AND RAI, L. Apple leaf disease identification and classification using resnet models. In *2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT)* (2020), IEEE, pp. 738–742.
- [102] LIANG, W.-J., ZHANG, H., ZHANG, G.-F., AND CAO, H.-X. Rice blast disease recognition using a deep convolutional neural network. *Scientific reports 9*, 1 (2019), 1–10.
- [103] LINGQING, F., YUJING, L., HUA, Y., ZONGWEI, J., JIAXIONG, G., ZHU, H., AND YIMING, H. Detection of apple leaf diseases target based on improved yolov7. *INMATEH-Agricultural Engineering 72*, 1 (2024).
- [104] LIU, B., ZHANG, Y., HE, D., AND LI, Y. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry 10*, 1 (2018), 11.

- [105] LIU, W., WANG, Z., LIU, X., ZENG, N., LIU, Y., AND ALSAADI, F. E. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11–26.
- [106] LONG, M., OUYANG, C., LIU, H., AND FU, Q. Image recognition of camellia oleifera diseases based on convolutional neural network & transfer learning. *Transactions of the Chinese Society of Agricultural Engineering* 34 (2018), 194–201.
- [107] LU, J., TAN, L., AND JIANG, H. Review on convolutional neural network (cnn) applied to plant leaf disease classification. *Agriculture* 11, 8 (2021), 707.
- [108] LYNDIA, D., BRAHIM, F., HAMID, S., AND HAMADOUN, C. Towards a semantic structure for classifying iot agriculture sensor datasets: An approach based on machine learning and web semantic technologies. *Journal of King Saud University-Computer and Information Sciences* 35, 8 (2023), 101700.
- [109] MACQUEEN, J., ET AL. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (1967), vol. 1, Oakland, CA, USA, pp. 281–297.
- [110] MAHESH, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]* 9, 1 (2020), 381–386.
- [111] MAJUMDAR, D., GHOSH, A., KOLE, D. K., CHAKRABORTY, A., AND MAJUMDER, D. D. Application of fuzzy c-means clustering method to classify wheat leaf images based on the presence of rust disease. In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014: Volume 1* (2015), Springer, pp. 277–284.
- [112] MASPARUDIN, M., FITRI, I., AND SUMIJAN, S. Development of apple fruit classification system using convolutional neural network (cnn) mobilenet architecture on android platform. *Sistemasi: Jurnal Sistem Informasi* 13, 1 (2024), 230–243.
- [113] MCCARTHY, J., MINSKY, M., ROCHESTER, N., AND SHANNON, C. E. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine* 27 (2006), 12–14.
- [114] MENGISTU, A. D., MENGISTU, S. G., AND MELESEW, D. An automatic coffee plant diseases identification using hybrid approaches of image processing and decision tree. *Indonesian Journal of Electrical Engineering and Computer Science* 9, 3 (2018), 806–811.
- [115] MINSKY, M. *Semantic information processing* cambridge, 1968.
- [116] MITCHELL, T. *Machine learning* mcgraw-hill international.
- [117] MOHANTY, S. P., HUGHES, D. P., AND SALATHÉ, M. Using deep learning for image-based plant disease detection. *Frontiers in plant science* 7 (2016), 1419.
- [118] MORANDÍN-AHUERMA, F. What is artificial intelligence? *International Journal of Research Publication and Reviews* (2022).
- [119] MURPHY, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [120] MUSA, P., AL RAFI, F., AND LAMSANI, M. A review: Contrast-limited adaptive histogram equalization (clahe) methods to help the application of face recognition. In *2018 third international conference on informatics and computing (ICIC)* (2018), IEEE, pp. 1–6.

- [121] NAWAZ, M., NAZIR, T., JAVED, A., MASOOD, M., RASHID, J., KIM, J., AND HUSSAIN, A. A robust deep learning approach for tomato plant leaf disease localization and classification. *Scientific reports* 12, 1 (2022), 18568.
- [122] NELSON, R. Artificial intelligence and natural man, 1977.
- [123] NILSSON, N. J. Problem-solving methods in. *Artificial Intelligence* 5 (1971).
- [124] NILSSON, N. J. *The quest for artificial intelligence: A History of Ideas and Achievements*. Cambridge University Press, Cambridge, UK, 2010.
- [125] NOCK, R. Algorithms, neural networks and other machine learning techniques. *Closer to the Machine: Technical Social and Legal Aspects of AI* (2019).
- [126] PADOL, P. B., AND YADAV, A. A. Svm classifier based grape leaf disease detection. In *2016 Conference on advances in signal processing (CASP)* (2016), IEEE, pp. 175–179.
- [127] PANCHAL, A. V., PATEL, S. C., BAGYALAKSHMI, K., KUMAR, P., KHAN, I. R., AND SONI, M. Image-based plant diseases detection using deep learning. *Materials Today: Proceedings* 80 (2023), 3500–3506.
- [128] PANCHAL, P., RAMAN, V. C., AND MANTRI, S. Plant diseases detection and classification using machine learning models. In *2019 4th international conference on computational systems and information technology for sustainable solution (CSITSS)* (2019), IEEE, pp. 1–6.
- [129] PANIGRAHI, K. P., DAS, H., SAHOO, A. K., AND MOHARANA, S. C. Maize leaf disease detection and classification using machine learning algorithms. In *Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019* (2020), Springer, pp. 659–669.
- [130] PANTAZI, X. E., MOSHOU, D., AND TAMOURIDOU, A. A. Automated leaf disease detection in different crop species through image features analysis and one class classifiers. *Computers and electronics in agriculture* 156 (2019), 96–104.
- [131] PARIKH, A., RAVAL, M. S., PARMAR, C., AND CHAUDHARY, S. Disease detection and severity estimation in cotton plant from unconstrained images. In *2016 IEEE international conference on data science and advanced analytics (DSAA)* (2016), IEEE, pp. 594–601.
- [132] PAULA, E. L., LADEIRA, M., CARVALHO, R. N., AND MARZAGAO, T. Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. In *2016 15th IEEE international conference on machine learning and applications (icmla)* (2016), IEEE, pp. 954–960.
- [133] PAVITHRA, S., PRIYADHARSHINI, A., PRAVEENA, V., AND MONIKA, T. Paddy leaf disease detection using svm classifier. *International Journal of communication and computer Technologies* 3, 1 (2015), 16–20.
- [134] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND ÉDOUARD DUCHESNAY. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830.
- [135] PENG, D., LI, W., ZHAO, H., ZHOU, G., AND CAI, C. Recognition of tomato leaf diseases based on dimpcnet. *Agronom* 13, 7 (2023), 1812.



- [136] PHUNG, V. H., AND RHEE, E. J. A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. *Applied Sciences* 9, 21 (2019), 4500.
- [137] PICON, A., ALVAREZ-GILA, A., SEITZ, M., ORTIZ-BARREDO, A., ECHAZARRA, J., AND JOHANNES, A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Computers and Electronics in Agriculture* 161 (2019), 280–290.
- [138] PICON, A., SEITZ, M., ALVAREZ-GILA, A., MOHNKE, P., ORTIZ-BARREDO, A., AND ECHAZARRA, J. Crop conditional convolutional neural networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. *Computers and Electronics in Agriculture* 167 (2019), 105093.
- [139] PINAR SAYGIN, A., CICEKLI, I., AND AKMAN, V. Turing test: 50 years later. *Minds and machines* 10, 4 (2000), 463–518.
- [140] PONTI, M. A., RIBEIRO, L. S. F., NAZARE, T. S., BUI, T., AND COLLOMOSSE, J. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In *2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)* (2017), IEEE, pp. 17–41.
- [141] PUJARI, D., YAKKUNDIMATH, R., AND BYADGI, A. S. Svm and ann based classification of plant diseases using feature reduction technique. *IJIMAI* 3, 7 (2016), 6–14.
- [142] QIAOLI, Z., LI, M., LIYING, C., AND HELONG, Y. Identification of tomato leaf diseases based on improved lightweight convolutional neural networks mobilenetv3. *Smart agriculture* 4, 1 (2022), 47.
- [143] QIU, R., YANG, C., MOGHIMI, A., ZHANG, M., STEFFENSON, B. J., AND HIRSCH, C. D. Detection of fusarium head blight in wheat using a deep neural network and color imaging. *Remote Sensing* 11, 22 (2019), 2658.
- [144] QUINLAN, J. R. Induction of decision trees. *Machine learning* 1 (1986), 81–106.
- [145] QUINLAN, J. R. *C4. 5: programs for machine learning*. 1993.
- [146] RAMCHARAN, A., BARANOWSKI, K., MCCLOSKEY, P., AHMED, B., LEGG, J., AND HUGHES, D. P. Deep learning for image-based cassava disease detection. *Frontiers in plant science* 8 (2017), 1852.
- [147] REN, S., HE, K., GIRSHICK, R., AND SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).
- [148] REN, S., JIA, F., GU, X., YUAN, P., XUE, W., AND XU, H. Recognition and segmentation model of tomato leaf diseases based on deconvolution-guiding. *Transactions of the Chinese Society of Agricultural Engineering* 36, 12 (2020), 186–195.
- [149] RICH, E. Artificial intelligence, 1983.
- [150] RUSSELL, S. J., AND NORVIG, P. *Artificial intelligence: a modern approach*. Simon Schuster, 1995.
- [151] RUSSELL, S. J., AND NORVIG, P. *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, Upper Saddle River, N.J., 2010.

- [152] SAEED, A., ABDEL-AZIZ, A., MOSSAD, A., ABDELHAMID, M. A., ALKHALED, A. Y., AND MAYHOUB, M. Smart detection of tomato leaf diseases using transfer learning-based convolutional neural networks. *Agriculture* 13, 1 (2023), 139.
- [153] SAH, S. Machine learning: a review of learning types.
- [154] SAHU, S. K., AND PANDEY, M. An optimal hybrid multiclass svm for plant leaf disease detection using spatial fuzzy c-means model. *Expert Systems with Applications* 214 (2023), 118989.
- [155] SAMAJPATI, B. J., AND DEGADWALA, S. D. Hybrid approach for apple fruit diseases detection and classification using random forest classifier. In *2016 International conference on communication and signal processing (ICCSP)* (2016), IEEE, pp. 1015–1019.
- [156] SAMPATHKUMAR, S., AND RAJESWARI, R. An automated crop and plant disease identification scheme using cognitive fuzzy c-means algorithm. *IETE Journal of Research* 68, 5 (2022), 3786–3797.
- [157] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of research and development* 3, 3 (1959), 210–229.
- [158] SARAGIH, T. H., FAJRI, D. M. N., AND RAKHMANDASARI, A. Comparative study of decision tree, k-nearest neighbor, and modified k-nearest neighbor on jatropha curcas plant disease identification. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control* (2020), 55–60.
- [159] SARANGDHAR, A. A., AND PAWAR, V. Machine learning regression technique for cotton leaf disease detection and controlling using iot. In *2017 international conference of electronics, communication and aerospace technology (ICECA)* (2017), vol. 2, IEEE, pp. 449–454.
- [160] SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. *SN computer science* 2, 3 (2021), 160.
- [161] SARKER, I. H., KAYES, A., BADSHA, S., ALQAHTANI, H., WATTERS, P., AND NG, A. Cybersecurity data science: an overview from machine learning perspective. *Journal of Big data* 7 (2020), 1–29.
- [162] SARKER, I. H., KAYES, A., AND WATTERS, P. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *Journal of Big Data* 6, 1 (2019), 1–28.
- [163] SHOAIB, M., HUSSAIN, T., SHAH, B., ULLAH, I., SHAH, S. M., ALI, F., AND PARK, S. H. Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. *Frontiers in Plant Science* 13 (2022), 1031748.
- [164] SHOAIB, M., SHAH, B., EI-SAPPAGH, S., ALI, A., ULLAH, A., ALENEZI, F., GECHEV, T., HUSSAIN, T., AND ALI, F. An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science* 14 (2023), 1158933.
- [165] SHOAIB, M., SHAH, B., EI-SAPPAGH, S., ALI, A., ULLAH, A., ALENEZI, F., GECHEV, T., HUSSAIN, T., AND ALI, F. An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science* 14 (2023), 1158933.
- [166] SHRESTHA, A., AND MAHMOOD, A. Review of deep learning algorithms and architectures. *IEEE access* 7 (2019), 53040–53065.

- [167] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [168] SINGH, A., AND KAUR, H. Potato plant leaves disease detection and classification using machine learning methodologies. In *IOP Conference Series: Materials Science and Engineering* (2021), vol. 1022, IOP Publishing, p. 012121.
- [169] SINGH, V., AND MISRA, A. K. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information processing in Agriculture 4*, 1 (2017), 41–49.
- [170] STRATHEARN, C., AND MA, M. The multimodal turing test for realistic humanoid robots with embodied artificial intelligence. In *Lifelike Computing Systems: 8th Edition in the Evolution of the Series of Autonomously Learning and Optimizing Systems (SAOS)* (2021).
- [171] SUBASI, A. Chapter 3 - machine learning techniques. In *Practical Machine Learning for Data Analysis Using Python*, A. Subasi, Ed. Academic Press, 2020, pp. 91–202.
- [172] SUN, F., WANG, Y., LAN, P., ZHANG, X., CHEN, X., AND WANG, Z. Identification of apple fruit diseases using improved yolov5s and transfer learning. *Transactions of the Chinese Society of Agricultural Engineering 38*, 171–179.
- [173] SZEGEDY, C., LIU, W., JIA, Y., Sermanet, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCHE, V., AND RABINOVICH, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1–9.
- [174] THAPA, R., SNAVELY, N., BELONGIE, S., AND KHAN, A. The plant pathology 2020 challenge dataset to classify foliar disease of apples. *arXiv preprint arXiv:2004.11958* (2020).
- [175] THILAGAVATHI, K., KAVITHA, K., PRABA, R. D., ARINA, S., AND SAHANA, R. Detection of diseases in sugarcane using image processing techniques. *Bioscience Biotechnology Research Communications, Special Issue*, 11 (2020), 109–115.
- [176] TIAN, Y., LI, E., LIANG, Z., TAN, M., AND HE, X. Diagnosis of typical apple diseases: a deep learning method based on multi-scale dense classification network. *Frontiers in Plant Science 12* (2021), 698474.
- [177] TIWARI, R., AND CHAHANDE, M. Apple fruit disease detection and classification using k-means clustering method. In *Advances in Intelligent Computing and Communication: Proceedings of ICAC 2020* (2021), Springer, pp. 71–84.
- [178] UMAPATHY EAGANATHAN, J. S., LACKOSE, V., AND BENJAMIN, F. J. Identification of sugarcane leaf scorch disease using k-means clustering segmentation and knn based classification. *International Journal of Advances in Computer Science and Technology (IJACST) 3*, 12 (2014), 11–16.
- [179] UPADHYAY, N., AND GUPTA, N. Diagnosis of fungi affected apple crop disease using improved resnext deep learning model. *Multimedia Tools and Applications* (2024), 1–20.
- [180] VAPNIK, V. N. Pattern recognition using generalized portrait method. *Automation and remote control 24*, 6 (1963), 774–780.
- [181] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).

- [182] VERMA, M. Artificial intelligence role in modern science: Aims, merits, risks and its applications. *Artificial Intelligence* 7, 5 (2023).
- [183] VISHNOI, V. K., KUMAR, K., AND KUMAR, B. Plant disease detection using computational intelligence and image processing. *Journal of Plant Diseases and Protection* 128 (2021), 19–53.
- [184] WAFI, N., HAMID, S., AND NADJIB, K. M. A new process for selecting the best background representatives based on gmm. *International Journal of Informatics and Applied Mathematics* 1, 1 (2019), 35–46.
- [185] WANG, C., ZHOU, J., WU, H., TENG, G., ZHAO, C., AND LI, J. Identification of vegetable leaf diseases based on improved multi-scale resnet. *Transactions of the Chinese Society of Agricultural Engineering* 36, 20 (2020), 209–217.
- [186] WANI, M. A., BHAT, F. A., AFZAL, S., AND KHAN, A. I. *Advances in deep learning*. Springer, 2020.
- [187] XIN, H., SHUQIN, L., AND BIN, L. Identification of grape leaf diseases based on multi-scale residual neural network. *Comput. Eng.* 47, 5 (2021), 285–291.
- [188] XIN, Y., KONG, L., LIU, Z., CHEN, Y., LI, Y., ZHU, H., GAO, M., HOU, H., AND WANG, C. Machine learning and deep learning methods for cybersecurity. *Ieee access* 6 (2018), 35365–35381.
- [189] XU, J., SHAO, M., WANG, Y., AND HAN, W. Recognition of corn leaf spot and rust based on transfer learning with convolutional neural network. *Transactions of the Chinese Society of Agricultural Machinery* 51, 2 (2020), 230–236.
- [190] YADAV, D., YADAV, A. K., ET AL. A novel convolutional neural network based model for recognition and classification of apple leaf diseases. *Traitement du Signal* 37, 6 (2020).
- [191] YADAV, S., SENGAR, N., SINGH, A., SINGH, A., AND DUTTA, M. K. Identification of disease using deep learning and evaluation of bacteriosis in peach leaf. *Ecological Informatics* 61 (2021), 101247.
- [192] YOSUKE, T., AND FUMIO, O. How convolutional neural networks diagnose plant disease. *Plant Phenomics* (2019).
- [193] ZHANG, K., XU, Z., DONG, S., CEN, C., AND WU, Q. Identification of peach leaf disease infected by xanthomonas campestris with deep learning. *Engineering in agriculture, environment and food* 12, 4 (2019), 388–396.
- [194] ZHANG, K., ZHANG, L., AND WU, Q. Identification of cherry leaf disease infected by podosphaera pannosa via convolutional neural network. *International Journal of Agricultural and Environmental Information Systems (IJAEIS)* 10, 2 (2019), 98–110.
- [195] ZHANG, M., AND MENG, Q. Automatic citrus canker detection from leaf images captured in field. *Pattern Recognition Letters* 32, 15 (2011), 2036–2046.
- [196] ZHANG, S., ZHANG, S., ZHANG, C., WANG, X., AND SHI, Y. Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Computers and Electronics in Agriculture* 162 (2019), 422–430.

- 
- [197] ZHEN, W., SHANWEN, Z., AND BAOPING, Z. Crop diseases leaf segmentation method based on cascade convolutional neural network. *Computer Engineering and Applications Journal* 56, 15 (2020), 242–250.
- [198] ZHEN, W., SHANWEN, Z., AND XIANFENG, W. Method for segmentation of cucumber leaf lesions based on improved full convolution neural network. *Jiangsu J. Agricult. Sci.* 35, 5 (2019), 1054–1060.

---

# AUTHOR'S PUBLICATION

## ❖ International publications

- ❑ **Imane Bouacida**, Brahim Farou, Lynda Djakhdjakha, Hamid Seridi, and Muhammet Kurulay, "Innovative deep learning approach for cross-crop plant disease detection: A generalized method for identifying unhealthy leaves", Information Processing in Agriculture, 2024, DOI: 10.1016/j.inpa.2024.03.002.

## ❖ International Communications

- ❑ Lynda Djakhdjakha, Brahim Farou, **Imane Bouacida**, and Aya Meguellatni, "A Semantic Framework for Monitoring Wheat Grain Warehouses", The 13th International Conference on Research in computing at Feminine RIF, Constantine, 2024.
- ❑ **Imane Bouacida**, Brahim Farou, Hamid Seridi and Mohamed anis Touahri, "An Artificial Intelligence-Based System for Detecting Diseases in Apple Tree Branches," 2023 International Conference on Electrical Engineering and Advanced Technology (ICEEAT), Batna, Algeria, 2023, pp. 1-7, Doi: 10.1109/ICEEAT60471.2023.10425986.
- ❑ **Imane Bouacida**, Brahim Farou, and Djakhdjakha Lynda, "Advancements in Sustainable Agriculture: An Overview of Pest Detection Using Deep Learning", The 6th International Hybrid Conference on Informatics and Applied Mathematics IAM'23, Guelma, Algeria, 2023.

## ❖ National Communications

- ❑ **Imane Bouacida**, Brahim Farou, Muhammet Kurulay, Hamid Seridi, and Aissa Klai, "Generating Synthetic Data for Plant Disease Datasets", 5th Conference on Informatics and Applied Mathematics IAM'22, Guelma, Algeria, 2022.

- **Imane Bouacida**, Djakhdjakha Lynda, and Hamid Seridi, "Internet of Things in Agriculture: Smart Farming and Precision Agriculture Overview", 4th Conference on Informatics and Applied Mathematics IAM'21, Guelma, Algeria, 2021.