



Université A. Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Fin de Cycle

EN VUE DE L'OBTENTION DU DIPLOME DE MASTER RECHERCHE EN INFORMATIQUE

Option : Système d'information avancée

Thème

Détection et classification automatique des
maladies des plantes à partir des images de
feuilles

Réalisé par :

MASSIOUN Hakima

Soutenu le 04/Juillet/2024, Devant le jury composé de :

Président : Dr F. BOUCHEBBAH M.C.B - Université de Béjaïa.

Examineur : Dr M. MOKTEFI M.C.B - Université de Béjaïa.

Encadrant : Dr M. KHAMMARI M.C.A - Université de Béjaïa.

Promotion : 2023/2024

Remerciements

En premier lieu, je remercie Allah le tout puissant de m'avoir donné la volonté, le courage et la patience qui m'ont permis de mener à bien ce travail.

Je tiens à exprimer mes vifs remerciements à mes encadrants Mr.KHAMMARI Mohammed et Mr.BELABBACI El Ouanas, d'avoir dirigé mon travail durant la période d'élaboration de ce mémoire.

Leurs orientations, leurs encouragements et précieux conseils ont été d'une grande aide. qu'ils trouvent ici toute ma reconnaissance et ma vive gratitude pour les efforts qu'ils ont fournis afin de guider ce travail à terme.

Je remercie également les membres de jury pour l'honneur qu'ils ont fait en acceptant de juger notre travail.

Je tiens à remercier toutes les personnes qui m'ont aidée de près ou de loin à réaliser ce travail, votre aide est dans ces pages.

Dedicaces

À ma mère qui a toujours été une source d'encouragement et de motivation pour moi sans son amour constant et ses conseils précieux, je n'aurais pas pu atteindre cet objectif.

À mon très chers père, grâce à toi, j'ai appris la valeur du travail et de la responsabilité. Ton soutien a été une lumière dans tout mon parcours.

À mes chers frères Mahmoud et Mhenni, à mes chères soeurs Siham et Selma, merci pour votre soutien morale, votre confiance et votre précieux conseils qui m'ont aidé dans les moments difficiles.

Je vous souhaite tous le bonheur du monde, et beaucoup de succès dans vos vies. Sans vous, ma vie n'aurait aucun goût. Vous êtes les étoiles qui illuminent mon ciel.

À tous ceux qui sont chers, aux personnes qui m'ont aidé et encouragé de près ou de loin, qui étaient toujours à mes cotées et qui m'ont accompagné durant mon chemin d'étude.

Et à toutes ma famille et mes amies.

Hakima

Abstract

The role of agricultural development is very important in the economy of a country. However, the emergence of several plant diseases constitutes a major barrier to the growth rate and quality of crops. The precise detection and classification of crop leaf diseases is a complex and lengthy activity due to the appearance of low contrast information in the input samples. In addition, changes in the size, location, structure of the sick part of the crop and the existence of a noise and blur effect in the input images further complicate the classification task. To solve these problems, a robust approach based on machine learning and computer vision techniques is proposed. More specifically, we have introduced a system that consists of 4 stages : pre-treatment, characteristics extraction with a combination of 3 methods (LPQ, LBP, VGG16), Dimensions reduction with (PCA, LDA) and finally a final classification using Euclidean distance, allowing a binary classification in the first time (sick and healthy) and in the second time a classification of deferent types of culture disease.

Keywords : Computer vision, Detection and classification of plant diseases, LPQ, LBP, VGG16, PCA, LDA, Euclidean Distance.

Résumé

Le rôle du développement agricole est très important dans l'économie d'un pays. Cependant, l'apparition de plusieurs maladies des plantes constitue un frein majeur au taux de croissance et à la qualité des cultures. La détection et la classification exactes des maladies des feuilles des cultures sont une activité complexe et longue en raison de l'apparition d'informations à faible contraste dans les échantillons d'entrée. De plus, les modifications de la taille, de l'emplacement, de la structure de la partie malade de la culture et l'existence d'un effet de bruit et de flou dans les images d'entrée compliquent encore davantage la tâche de classification. Pour résoudre ces problèmes, une approche robuste basée sur les techniques d'apprentissage automatique et de la vision par ordinateur est proposée. Plus précisément, nous avons introduit un système qui compose de 4 étapes : pré-traitement, extraction des caractéristiques avec une combinaisons de 3 méthodes (LPQ, LBP, VGG16), Réduction des dimensions avec (PCA, LDA) et enfin une classification finale en utilisant la distance euclidienne, permettant une classification binaire dans un premier temps (malade et saine) et dans un deuxième temps une classification de déferents types de maladie de la culture.

Mots clés : Vision par ordinateur, Détection et classification des maladies des plantes, LPQ, LBP, VGG16, PCA, LDA, Distance euclidienne .

Table des matières

Liste des Figures	v
Liste des Tableaux	vi
Liste des abréviations	vii
Introduction générale	1
1 État de l'art sur les systèmes de détection et classification automatiques des maladies des plantes	3
1.1 Introduction	3
1.2 Les maladies des plantes	4
1.2.1 Types des maladies des plantes	4
1.2.2 Méthodes pour la détection des maladies des plantes	6
1.3 Systèmes de diagnostic des maladies des plantes	7
1.4 Étapes de détection des maladies des plantes	9
1.4.1 Acquisition de l'image :	9
1.4.2 Pré-traitement de l'image :	9
1.4.3 La Segmentation d'image :	10
1.4.4 Extraction des caractéristiques :	10
1.4.4.1 Méthodes classiques	11
1.4.4.2 Méthodes basées sur l'apprentissage profond	11
1.4.5 Classification :	11
1.5 Travaux connexes	11
1.6 Conclusion	14

2 Aspects théoriques sur les méthodes et techniques utilisées	16
2.1 Introduction	16
2.2 Méthodes de Segmentation	16
2.3 Méthodes d'extraction des caractéristiques	19
2.4 Réduction de dimension	22
2.4.1 Méthodes de réduction de dimension	22
2.5 Apprentissage profond (Deep learning)	24
2.5.1 Réseaux de neurones artificiels	24
2.5.2 Réseaux de neurones convolutifs	25
2.5.2.1 Couche d'entrée (Input layer)	26
2.5.2.2 Couche de convolution (Convolutional Layer)	26
2.5.2.3 Couche de pooling (Pooling layer)	27
2.5.2.4 Couche d'Aplatissement (Flatten layer)	28
2.5.2.5 Couche entièrement connectée (Fully-connected layer)	29
2.5.2.6 Paramètres d'un CNN	29
2.5.2.7 Exemples d'architectures des CNN les plus célèbres :	31
2.6 Algorithmes de classification	32
2.7 Conclusion	35
3 Approche proposée	36
3.1 Introduction	36
3.2 Description de l'Architecture	36
3.2.1 Pré-traitement	38
3.2.1.1 Suppression de bruit	38
3.2.1.2 Segmentation et Réduction des Dimensions	38
3.2.1.3 Redimensionnement	39
3.2.1.4 Normalisation	39
3.2.1.5 Augmentation des données	39
3.2.2 Extraction des caractéristiques	40
3.2.2.1 Extraction des caractéristiques avec LPQ	40
3.2.2.2 Extraction des caractéristiques avec LBP	41
3.2.2.3 Extraction des caractéristiques avec VGG16	43
3.2.3 Réduction de dimension	45
3.2.3.1 Réduction de dimension avec PCA	45

3.2.3.2 Réduction de dimension avec LDA	46
3.2.4 Fusion	47
3.2.5 Comparaison et prise de décision :	47
3.2.5.1 Distance euclidienne	48
3.3 Conclusion	48
4 Méthodologie et résultats	49
4.1 Introduction	49
4.2 Matériels, Environnement et outils de développement	49
4.2.1 Matériel utilisé	49
4.2.2 Environnements et outils de développement	49
4.3 Base de donnée	53
4.4 Métriques d'évaluation	55
4.4.1 Matrice de confusion(Confusion Matrix)	55
4.4.2 Précision de la classification (Classification Accuracy)	56
4.4.3 Précision	56
4.4.4 Recall ou Sensibilité ou Rappel	57
4.4.5 F1-score	57
4.4.6 ROC	57
4.5 Protocole expérimental	57
4.5.1 Méthodologie	57
4.5.2 Répartition des données	58
4.6 Performances atteintes	59
4.6.1 Modèle 01	59
4.6.2 Modèle 02	60
4.7 Comparaison avec l'état de l'art	62
4.8 Interprétation et discussion des résultats	63
4.9 Conclusion	64
Conclusion générale et perspectives	65
Bibliographie	67

Table des figures

1.1	Facteurs des maladies agricoles [54].	4
1.2	Maladies bactériennes et virales sur les feuilles [43].	5
1.3	Maladie fongique sur les feuilles [43].	6
1.4	Types de maladies des plantes causées par des bactéries, des virus et des champignons [76].	6
1.5	Méthodes de détection des maladies des plantes [77].	7
1.6	Processus de détection des maladies des plantes	9
1.7	Exemples des images avant et après la segmentation [81].	10
2.1	Image segmentée par l'approche contour avec filtrage Canny	17
2.2	Organigramme de l'ensemble des étapes nécessaire à la construction du descripteur LPQ	20
2.3	Illustration de calcul d'un LBP de base [53].	20
2.4	Illustration du principe de séparation optimale des classes par le LDA [34].	25
2.5	Architecture générale d'un CNN [31].	26
2.6	Les 3 dimension de l'entrée d'un CNN [26].	26
2.7	Opération de convolution [49].	27
2.8	Les 3 types de l'opération de pooling [27].	28
2.9	Mécanisme d'aplatissement [5].	28
2.10	Cartes de caractéristiques [66].	29
2.11	Principe du stride [18].	30
2.12	Diagramme de bloc de K-Voisin le plus proche [80].	33
2.13	Classification par SVM : Hyperplan Optimal et Vecteurs de Support [36].	34
2.14	Illustration d'un arbre de décision de William Koerhsen [6].	35

3.1 Schéma récapitulatif de notre système	37
3.2 Exemple de filtre gaussien appliquée sur les images	38
3.3 Exemple de segmentation et réduction de dimension	39
3.4 Exemples de transformation faites sur les images	40
3.5 Étapes de fonctionnement du descripteur LPQ [41].	41
3.6 Exemple de LPQ sur une image	41
3.7 Illustration de calcul de d'un LBP de base [33].	42
3.8 Exemple de LBP sur une image	42
3.9 Architecture du modèle VGG16 [49].	43
3.10 Extraction des caractéristiques avec VGG16 [58].	45
4.1 Logo Google Colab [8].	50
4.2 Logo Spyder [7].	50
4.3 Logo Python [21].	51
4.4 Logo OpenCV [14].	51
4.5 Logo NumPy [13].	52
4.6 Logo Matplotlib [16].	52
4.7 Logo TensorFlow [22].	52
4.8 Logo TensorFlow [12].	53
4.9 Scikit-learn-logo [20].	53
4.10 Exemple de notre base de données	55
4.11 Matrice de confusion [17].	56
4.12 Méthode adopté pour la détection et classification des maladies des plantes	58

Liste des tableaux

1.1 Résumé des articles sur l'utilisation de DL et ML dans la détection et classification des maladies des plantes	14
4.1 Description de la Base de données d'apprentissage 01	58
4.2 Description de la Base de données d'apprentissage 02	59
4.3 Résultats de la classification binaire sur les données de test	59
4.4 Résultats de la classification par type de maladie sur les données de test	60
4.5 Précision, recall et F1-score par classe du modèle 01	61
4.6 Précision, recall et F1-score par classe du modèle 02	62
4.7 Comparaison avec l'état de l'art en terme l'accuracy	63

Liste des abréviations

ACP	<i>Analyse en composantes principales</i>
API	<i>Application Programming Interface</i>
BSIF	<i>Binarized Statistical Image Features</i>
CNN	<i>Convolutional neural network</i>
DL	<i>Deep learning</i>
ED	<i>Euclidean Distance</i>
FCN	<i>Fully connected network</i>
HOG	<i>Histograms of Oriented Gradient</i>
IA	<i>Intelligence artificielle</i>
KNN	<i>K-Nearest Neighbors</i>
LBP	<i>Local Binary Patterns</i>
LDA	<i>Linear Discriminant Analysis</i>
LPQ	<i>Local Phase Quantization</i>
ML	<i>Machine learning</i>
UAV	<i>Unmanned Aerial Vehicles</i>
ReLu	<i>Fonction Unité Linéaire Rectifiée</i>
RF	<i>Random Forest</i>

Liste des abréviations

RGB	<i>Red Green Bleu</i>
SIFT	<i>Scale Invariant Feature</i>
SVM	<i>Support Vector Machine</i>

Introduction générale

L'augmentation rapide de la population mondiale a fait de l'agriculture un domaine vital pour l'humanité. La demande de nourriture ne cesse de croître et les agriculteurs sont confrontés à des défis qui mettent en péril leurs cultures. La principale menace qui affecte l'industrie agricole est les attaques d'organismes vivants, ainsi que les maladies. Tous les types de cultures, y compris celles qui sont consommées par les humains ou le bétail (comme les céréales, les légumineuses et les fruits), sont affectés par les maladies [39]. Ces dernières doivent être éliminées instantanément pour éviter les infections secondaires. Pour éliminer immédiatement les plantes infectées, la détection de la plante malade est la tâche la plus importante.

Traditionnellement, la détection des maladies des plantes repose sur des inspections visuelles réalisées par des experts. Bien que cette méthode soit efficace, elle est souvent limitée par le temps, les ressources, et la subjectivité humaine. Avec les avancées technologiques, il devient possible de surmonter ces limitations en utilisant des systèmes automatisés. En effet, la vision par ordinateur et l'apprentissage automatique offrent des méthodes rapides et précises pour détecter et classifier les maladies des plantes à partir d'images numériques.

Les drones agricoles, ou UAV (Unmanned Aerial Vehicles), jouent un rôle crucial dans cette révolution. Équipés de capteurs avancés et de caméras multispectrales, ils peuvent survoler les champs et capturer des données détaillées sur la santé des cultures. Ces données incluent des images haute résolution qui révèlent des anomalies invisibles à l'œil nu, telles que les stress hydriques, les variations de sol, et les premiers signes de maladies [28].

L'intégration de ces technologies permet non seulement une surveillance continue et précise des cultures, mais également une optimisation des ressources agricoles. En identifiant rapidement les zones affectées et en fournissant des diagnostics précis, les systèmes automatisés permettent aux agriculteurs de prendre des décisions éclairées et de mettre en œuvre des stratégies de gestion efficaces.

Pour présenter correctement notre modeste travail, nous avons divisé notre mémoire en

quatre chapitres principaux comme suit :

Le premier chapitre est consacré aux généralités sur les maladies des plantes, étapes de traitements des systèmes de détection et classification des maladies des plantes, ainsi nous élaborons l'état de l'art qui représente les travaux connexes les plus importants que nous synthétiserons, nous présenterons ceci dans un tableau qui contiendra les grandes lignes de chaque document.

Le deuxième chapitre, on présente un aperçu théorique sur les méthodes et techniques utilisée dans le domaine de détection et classification des maladies des plantes.

Le troisième chapitre, on vas présenter notre approche proposée, en expliquant en détaille les techniques et méthodes utilisée.

Le quatrième chapitre, nous présentons les outils de programmation ainsi que les logiciels choisis, et les résultats d'exécution de notre système.

Enfin, nous terminons ce mémoire par une conclusion et quelques perspectives pour un travail futur.

État de l'art sur les systèmes de détection et classification automatiques des maladies des plantes

1.1 Introduction

La santé des plantes est cruciale pour le succès de la production agricole. Cependant, elles sont souvent menacées par une variété de maladies et de parasites qui peuvent avoir un impact significatif sur le rendement des cultures. Ces maladies et ravageurs peuvent causer des dommages considérables aux cultures, entraînant des pertes économiques importantes pour les agriculteurs [52]. La détection précoce est la pierre angulaire d'une prévention et d'un contrôle efficaces des maladies des plantes, et joue un rôle essentiel dans la gestion et la prise de décision de la production agricole.

L'utilisation des feuilles des plantes dans l'agriculture est cruciale pour diagnostiquer diverses maladies, car la compréhension des symptômes de celles-ci est souvent essentielle pour résoudre des problèmes. Les cultures végétales sont confrontées à de nombreux problèmes de feuille telle que la couleur, la forme et la taille en raison des agents pathogènes, de l'alimentation des insectes et de la nutrition des plantes [48]. La méthode traditionnelle de diagnostic utilise une vérification visuelle sur le terrain par un spécialiste, mais cette méthode manuelle a des limites en termes de précision et de disponibilité de personnes qualifiées. Pour résoudre ces problèmes, il est de plus en plus nécessaire de faire des moyens automatisés de détecter et de catégoriser les maladies végétales à partir d'images. Cela donnera des solutions de gestion des cultures plus rapides et précises [25].

Dans ce chapitre, nous allons présenter des généralités sur les maladies des plantes, en décrivons les étapes de traitement qui constituent un système de détection et classification automatique des maladies des plantes. Ensuite nous élaborerons l'état de l'art qui représentera les travaux connexes les plus importants que nous synthétiserons.

1.2 Les maladies des plantes

Chez les plantes, la maladie est toute condition anormale qui endommage une plante et réduit sa productivité ou son utilité pour l'homme. Les maladies des plantes sont causées par des organismes vivants (biotique) qui attaquent et obtiennent leur nutrition de la plante qu'ils infectent, soit par certain facteur environnemental (abiotique), tel que le changement rapide de température, l'excès d'humidité, l'insuffisance des nutriments, le pH du sol et les conditions d'humidité élevée, voir la figure [1.1](#).

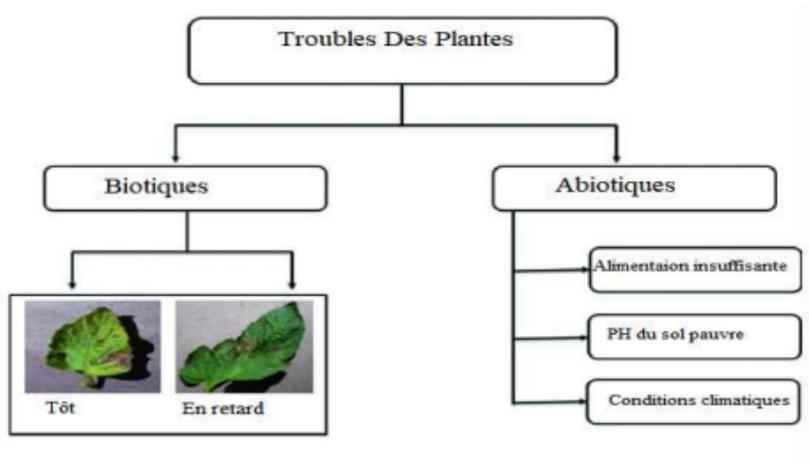


FIGURE 1.1 – Facteurs des maladies agricoles [\[54\]](#).

1.2.1 Types des maladies des plantes

La classification des maladies des plantes peut être effectuée de plusieurs manières en fonction de divers critères. Cependant, une classification courante se base sur l'agent pathogène responsable de la maladie, on distingue alors les types suivants :

1. **Maladies bactériennes** : sont des infections causées par environ 200 types de bactéries pathogènes qui peuvent entraîner des dommages importants aux cultures, et peuvent se propager par les insectes, les éclaboussures d'eau, ou d'autres plantes [\[59\]](#).

Symptômes : La maladie se caractérise par de petites taches vert pâle qui deviennent

rapidement visibles comme imbibées d'eau. Les lésions s'agrandissent puis apparaissent comme des taches sèches et mortes, comme illustré dans la figure 1.2 (a).

2. **Maladies virales** : elles sont des maladies incurables et généralisées causées par des virus (agents pathogènes les moins visibles mais les plus dangereux pour les plantes). Une fois infectée, la survie d'une plante devient pratiquement impossible c'est à dire une plante infectée reste malade toute sa vie (absence de système immunitaire), Les virus se propagent principalement à travers le sol et peuvent affecter différentes parties des plantes, y compris les feuilles, les tiges et les racines [43].

Symptômes : Les virus ne produisent aucun signe révélateur qui peut être facilement observé et souvent facilement confondus avec des carences en nutriments et des dommages causés par les herbicides. Les pucerons, les cicadelles, les aleurodes et les chrysomèles du concombre sont des vecteurs courants de cette maladie, par exemple le virus de la mosaïque, Recherchez des rayures ou des taches jaunes ou vertes sur le feuillage, comme le montre la figure 1.2 (b). Les feuilles peuvent être ridées, enroulées et la croissance peut être retardée.



(a) Bacterial leaf spot



(b) mosaic virus

FIGURE 1.2 – Maladies bactériennes et virales sur les feuilles [43].

3. **Maladies fongiques** : Causées par les champignons pathogènes. Selon les recherches, ce type de maladie des plantes détruit environ un tiers de toutes les cultures alimentaires chaque année [43].

Symptômes : le mildiou tardif causé par le champignon *Phytophthora infestans*, illustré dans la figure 1.3(a). Il apparaît d'abord sur les feuilles inférieures et plus anciennes sous forme de taches gris-vert imbibées d'eau. Lorsque la maladie fongique mûrit, ces taches foncent et un développement fongique blanc se forme sur les faces inférieures des feuilles. La brûlure tardive est causée par le champignon *Alternaria solani*, illustré dans la figure 1.3(b). Elle apparaît d'abord sur les feuilles inférieures et plus anciennes sous forme de petites taches brunes avec des anneaux concentriques formant un motif en œil de bœuf.

Lorsque la maladie mûrit, elle se propage vers l'extérieur sur la surface de la feuille, la faisant jaunir. Dans le mildiou, des taches jaunes à blanches apparaissent sur les surfaces supérieures des feuilles plus anciennes. Ces zones sont recouvertes de blanc à grisâtre sur les faces inférieures, comme illustré dans la figure 1.3(c).



FIGURE 1.3 – Maladie fongique sur les feuilles [43].

La figure 1.4 illustre les différents types de maladies des plantes causées par des bactéries, des virus et des champignons



FIGURE 1.4 – Types de maladies des plantes causées par des bactéries, des virus et des champignons [76].

1.2.2 Méthodes pour la détection des maladies des plantes

L'identification de la présence des maladies des plantes peut se faire de deux manières principales à l'aide de techniques moléculaires : l'ELISA et la PCR [77].

Dans l'ELISA, des anticorps spécifiques aux protéines des microbes sont générés par un animal suite à une injection d'antigène. Ces anticorps permettent de repérer la présence d'antigènes dans les échantillons végétaux en utilisant des colorants fluorescents ou des enzymes. En

revanche, la PCR consiste à extraire, purifier et amplifier l'ADN de l'agent pathogène avant de réaliser une électrophorèse sur gel pour confirmer sa présence. Bien que ces techniques aient révolutionné la détection des maladies des plantes, elles ne sont pas très fiables à l'heure actuelle, et moins pratiques pour le dépistage rapide d'un grand nombre d'échantillons en raison de leur processus laborieux [61].

les méthodes de détection des maladies des plantes sont classées en méthodes directes et indirectes comme illustré par la figure 1.5

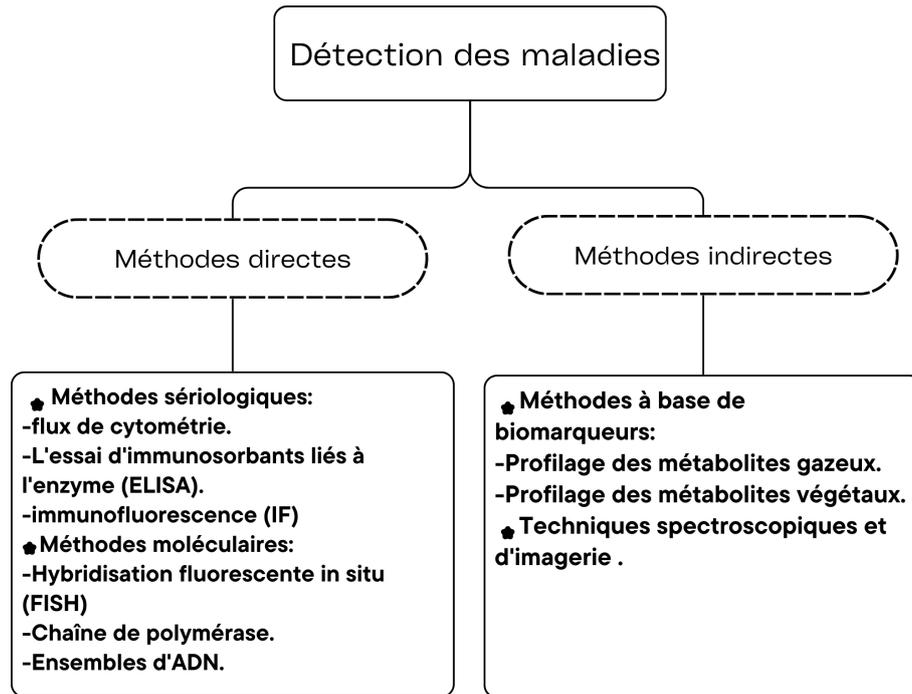


FIGURE 1.5 – Méthodes de détection des maladies des plantes [77]

Les progrès récents en matière de technologie agricole ont suscité une demande accrue de méthodes automatisées et non destructrices de détection des maladies végétales. Les techniques de spectroscopie et d'imagerie, telles que l'image fluorescente, l'imaging multispectral et l'infrarouge, sont étudiées pour leur capacité à détecter les maladies à un stade précoce, fournissant un outil pratique pour la surveillance des maladies sur une grande échelle en temps réel sur le terrain. Ces méthodes pourraient être intégrées dans les véhicules agricoles autonomes pour fournir des informations sur la santé des plantes et contrôler la propagation des maladies [77].

1.3 Systèmes de diagnostic des maladies des plantes

Parmi les définitions que nous avons trouvées dans la littérature, celle donnée par Esker [40] vient enrichir la compréhension des systèmes de détection et de classification des maladies des plantes, également connus sous le nom de systèmes d'alerte aux maladies des plantes. Ces

systèmes ont été développés pour aider les agriculteurs à prendre des décisions économiques éclairées sur la gestion des maladies. Ils évaluent les coûts et les avantages des actions nécessaires pour contrôler les maladies, permettant ainsi aux agriculteurs de déterminer quand et où agir pour maximiser leurs récoltes tout en réduisant leurs dépenses.

De plus, ces systèmes aident à déterminer quel matériel de propagation ou quelle semence acheter, ainsi que s'il est judicieux de planter une culture spécifique dans une région donnée. Leur principe fondamental est d'évaluer le risque de développement d'une maladie ou d'une augmentation de son intensité. En fournissant des informations sur la fréquence des maladies et leur impact potentiel, ces systèmes permettent aux agriculteurs de prendre des décisions éclairées concernant leurs pratiques agricoles et le choix des variétés de plantes les mieux adaptées à leur environnement.

Critères de Réussite des Systèmes de Détection et Classification des Maladies des Plantes

Dans le livre "Introduction aux maladies des plantes, identification et gestion" **Lucas et al** [83], estiment qu'un système de détection est efficace s'il possède les Critères essentielles suivantes :

1. **Fiabilité** : L'utilisation de données environnementales solides implique de préciser les facteurs climatiques tels que la température ou l'humidité.
2. **Simplicité** : Il est essentiel que le système propose une interface conviviale afin d'être facilement utilisé à grande échelle par les agriculteurs.
3. **Importance** : Il est essentiel que la maladie traitée soit d'une importance économique pour la culture et qu'elle soit plutôt sporadique, ce qui signifie que le traitement temporaire n'est pas possible.
4. **Utilité** : Il est essentiel que le système de diagnostic soit bénéfique, car il permet aux producteurs de se décharger de diverses tâches de surveillance de la culture en fournissant les recommandations nécessaires pour l'utilisation des produits chimiques et le contrôle. Des maladies à l'heure opportune.
5. **Disponibilité** : Il est nécessaire d'avoir des informations en temps réel sur l'élément d'interaction (types de plantes, variétés de plantes, données climatiques, etc.).
6. **Rentabilité** : Il est essentiel que le système de diagnostic soit accessible en ce qui concerne les coûts techniques et la gestion des maladies disponibles.

1.4 Étapes de détection des maladies des plantes

L'identification des maladies végétales est divisée en cinq étapes, comme illustrée par la figure 1.6 :

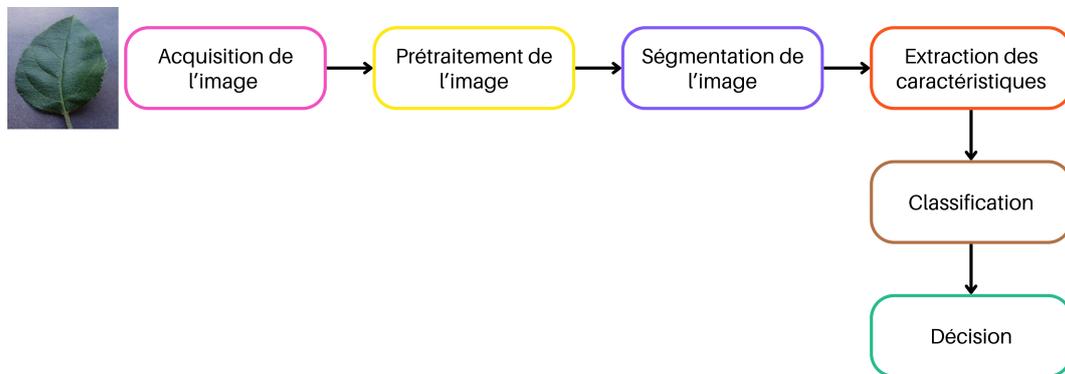


FIGURE 1.6 – Processus de détection des maladies des plantes

1.4.1 Acquisition de l'image :

L'acquisition d'images est la première étape de la détection des maladies des plantes. Cette étape consiste à capturer des images de plantes à l'aide de diverses techniques d'imagerie, telles que des appareils photo numériques, des smartphones ou des drones tout en fonction de la zone de contrôle et des objectifs de notre traitement.

1.4.2 Pré-traitement de l'image :

Le pré-traitement des images est une étape importante pour améliorer leur qualité avant de les utiliser dans des calculs ou des analyses [42]. Cette étape offre de nombreuses techniques dans lesquelles nous ajustons les images pour mieux analyser et détecter les caractéristiques pertinentes.

Parmi ces techniques, nous avons le débruitage, qui implique l'élimination d'objets indésirables présents dans l'image, et l'amélioration, qui vise à corriger des défauts tels que des distorsions ou des variations de contraste. Le redimensionnement permet d'ajuster la taille des images, tandis que l'augmentation des données implique la création de nouvelles données de formation en introduisant des variations dans les images existantes. Le recadrage est utilisé pour isoler les parties pertinentes de l'image, tandis que la conversion d'espace de couleur peut être nécessaire pour normaliser la représentation des couleurs. Enfin, le lissage est une technique qui aide à réduire les variations brusques de l'image et supprimer le bruit, ce qui facilite la détection des caractéristiques [76].

1.4.3 La Segmentation d'image :

Une fois le pré-traitement terminé, il est essentiel de procéder à la segmentation de l'image afin de séparer les éléments d'intérêt du reste de l'image, comme le fond ou les objets indésirables (voir la figure 1.7), ce qui facilite l'extraction des caractéristiques pertinentes pour une analyse ultérieure. La segmentation est une méthode utilisée dans le traitement d'images numériques pour séparer l'image en parties distinctes ou focalisées, généralement en se basant sur des caractéristiques similaires entre les pixels [42]. Il existe différentes techniques de segmentation d'images, telles que la segmentation par région, par seuil, etc. Nous les examinerons en détail dans le chapitre suivant.

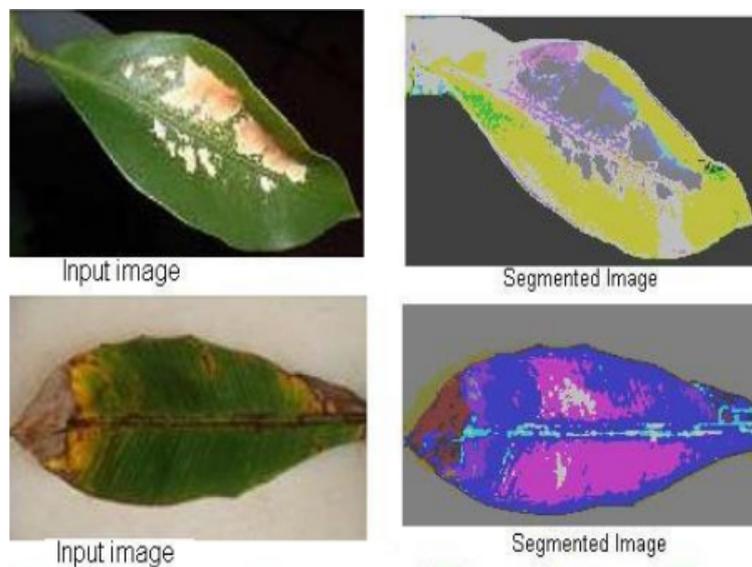


FIGURE 1.7 – Exemples des images avant et après la segmentation [81].

1.4.4 Extraction des caractéristiques :

L'extraction des caractéristiques est la principale étape pour la construction des modèles de détection et classification des maladies des plantes et vise à extraire les attributs pertinents caractérisant chaque maladie.

Les symptômes de maladies apparaissent dans différentes parties des plantes. Cependant, ce sont les feuilles qui sont les plus souvent utilisées pour détecter les maladies. Pour cela il existe plusieurs méthodes dans la littérature affectée à cette tâche, qu'on peut les classer en deux catégories : les méthodes classiques basées sur les descripteurs et les méthodes basées sur l'apprentissage profond.

1.4.4.1 Méthodes classiques

Parmi les descripteurs peuvent être utilisée pour extraire les caractéristiques, nous distinguons les descripteurs de textures et les descripteurs de gradient. En terme de texture, les descripteurs les plus couramment utilisés sont les motifs binaires locaux (LBP) et les matrices de co-occurrences dans les travaux de N.Kaur [51], tandis que pour les gradients, nous calculons principalement HOG (Histograms of Oriented Gradients) et SIFT (Scale Invariant Feature) [86].

1.4.4.2 Méthodes basées sur l'apprentissage profond

Des recherches récentes ont démontré l'efficacité des réseaux de neurones convolutifs pour extraire des caractéristiques des images. Ces méthodes sont discutées dans la section 1.5, qui donne un aperçu de l'état de l'art.

1.4.5 Classification :

La classification des images, plus précisément pour la détection et classification des maladies des plantes, repose sur l'utilisation des vecteurs des caractéristique obtenues par l'étape précédente. Cette étape consiste à analyser ces vecteurs pour attribuer chaque image à une catégorie spécifique, reflétant l'état de santé de la plante. Plusieurs travaux de recherche traitent cette étape avec différents classifieurs. Souvent ils utilisent des classifieurs par apprentissage automatique tels que des SVM, le Random Forest, KNN ou encore des réseaux de neurones. Un aperçu rapide de l'état de l'art sera présenté dans la section suivante.

1.5 Travaux connexes

L'identification des maladies des plantes à partir des images de feuilles a joué un rôle essentiel dans la recherche en matière d'agriculture durable, avec de nombreuses approches mises en œuvre pour résoudre ce problème que an vas voir ci-dessous :

Srivastava et al [82] proposent une approche qui utilise des modèles d'apprentissage par transfert modifiés pour améliorer la précision de la détection et de la classification des maladies. Ils utilisent cinq modèles d'apprentissage profond modifiés différents (Vgg16, InceptionV3, MobileNetV2, Xception et DenseNet121). Les modèles sont entraînés sur deux ensembles de données : l'ensemble de données de Mendeley (4590 images de 22 classes) et l'ensemble de données de cerises (2052 images de 2 classes). Les auteurs ont utilisé des techniques d'augmentation

de données, telles que des retournements horizontaux et des rotations, pour prévenir le surajustement et ont divisé chaque ensemble de données en 70 % pour l'entraînement et 30 % pour les tests. Pour évaluer la qualité et la quantité des modèles proposés, une matrice de qualité a été utilisée, et le résultat montre que pour l'ensemble de données1, MobileNetV2 surpasse les autres modèles existants avec une précision de 98,9 %. Pour l'ensemble de données Cherry, DenseNet121 surpasse les autres modèles existants avec une précision de 99,9 %.

Kaya et al [52] présentent une nouvelle approche basée sur l'apprentissage profond pour la détection des maladies végétales. En fusionnant des images RGB et segmentées, l'architecture de réseau de neurones convolutifs multi-têtes basée sur DenseNet présente une méthode précise et robuste pour reconnaître et diagnostiquer différentes maladies végétales. Pour évaluer l'efficacité de ce type d'approches, le modèle a été testé sur le jeux de données PlantVillage, composé de 54183 images appartenant à 38 classes. La technique de validation croisée de cinq plis a été appliquée pour évaluer les performances de modèles. Une précision moyenne, un rappel, une précision ainsi qu'un score F1 supérieur à 98 % est indiqué par les résultats expérimentaux.

Sunil et al [44] proposent une approche basée sur la vision par ordinateur et l'apprentissage automatique. Cette dernière comporte plusieurs étapes. Premièrement, des échantillons de feuilles de tomate représentant six troubles différents. Les tailles de ces échantillons sont redimensionnés et l'égalisation de l'histogramme est utilisée pour modifier la qualité des images. Ensuite, la segmentation basée sur le cluster K-means est appliquée à l'espace de données de partitionnement dans les cellules de Voronoi, en extrayant ainsi les contours des échantillons de feuilles. Différentes techniques d'extraction de caractéristiques telles que la transformation discrète de la vaillette et la matrice de co-occurrence de niveau gris, sont utilisées pour extraire des informations pertinentes à partir d'échantillons de feuilles, l'analyse des composants principaux (PCA) pour réduire la dimensionnalité des données tout en conservant les informations les plus importantes. Enfin, ces caractéristiques extraites sont classées en utilisant des approches d'apprentissage automatique telles que Support Vector Machines (SVM), Convolutional Neural Networks (CNN) et K-Nearest Neighbors (K-NN). Les modèles SVM, K-NN et CNN ont été évalués et ont montré des précisions respectives de 88 %, 97 % et 99,6 % pour la détection des échantillons de tomate atteints de maladies.

Ajra et al [24] suggèrent d'utiliser le traitement d'images en combinaison avec deux modèles de réseaux neuronaux convolutionnels bien reconnus : AlexNet et ResNet-50. La méthodologie proposée implique d'abord l'utilisation de cette technique sur les ensembles de données sur les feuilles de pommes de terre et de tomates disponibles à Kaggle pour étudier les symptômes des feuilles malades. Ensuite, un processus d'extraction et de classification des caractéristiques

est effectué sur les images de datasets pour la détection des maladies des feuilles en utilisant AlexNet et ResNet-50 comme modèles tout en appliquant le traitement de l'image. Les résultats expérimentaux indiquent l'efficacité de l'approche proposée ; avec des taux de précision globaux de 97% et 96,1% pour ResNet-50, et de 96,5% et 95,3% pour AlexNet, respectivement, pour la classification des feuilles saines-malades et des maladies des feuilles.

Audre Arlene et al[30] propose une approche novatrice pour classifier les maladies des feuilles de tomate. L'objectif principal est d'améliorer la précision de la classification tout en réduisant le temps de calcul. Les images de feuilles de tomate sont collectées et séparées en cinq catégories : sain, bactérien, mosaïque, septoria et courbure jaune, qui sont divisées en deux parties : 60% pour l'entraînement et 40% pour les tests. Le dossier d'entraînement contient 300 images et le dossier de test en contient 200. L'approche utilise des techniques d'extraction des caractéristiques multiples, incluant les histogrammes de couleur, les moments de Hu, les caractéristiques de Haralick et le motif binaire local (LBP), combinées avec des algorithmes de classification par forêt aléatoire et arbre de décision. Après avoir collecté des images de feuilles de tomate et extrait ces caractéristiques, les auteurs ont entraîné et testé les classificateurs. Les résultats montrent que la forêt aléatoire a atteint une précision de 94%, tandis que l'arbre de décision a atteint une précision de 90%. Cette méthode permet une classification plus précise des maladies, contribuant à la réduction des pertes agricoles en assurant une détection précoce et efficace des maladies des plantes.

Ganatra et al[42] présentent une méthode de détection et de classification des maladies des feuilles de plantes en utilisant des techniques de traitement d'image et d'apprentissage automatique. Ils ont utilisé le dataset Plant Village contenant 14 956 images réparties en 38 classes de maladies. Après le pré-traitement pour améliorer leur qualité, une segmentation par la méthode de seuillage d'Otsu est réalisée pour isoler les parties des feuilles qui sont pertinentes. Diverses caractéristiques sont extraites des feuilles malades, divisées en six catégories : caractéristiques des moments de couleur, caractéristiques de texture, caractéristiques de forme, caractéristiques des veines, transformation de Gabor, et moments de Zernike. Quatre algorithmes de classification ont été évalués : random forest, machine à vecteurs de support, k-plus proches voisins et réseau de neurones artificiels. Selon les résultats, le random forest a donné les meilleurs résultats avec une précision de 73,38%, le SVM était deuxième avec 67,27%. Par conséquent, cette méthode de détection est relativement rapide et a un niveau élevé de précision. Cela contribue à la détection rapide des maladies des plantes et, donc, à la réduction des pertes agricoles.

Zaki et al[90] utilisent le réseau de neurones convolutifs MobileNet V2, qui est une architecture d'apprentissage en profondeur compacte, pour détecter trois types de maladies des feuilles

Chapitre 1. État de l'art sur les systèmes de détection et classification automatiques des maladies des plantes

de tomates. Ils ont entraîné le réseau à l'aide de 4 671 images appartenant à 4 classes, provenant de l'ensemble de données PlantVillage. L'algorithme développé est capable de détecter les maladies des feuilles de tomates avec une précision de plus de 90%.

Voici un tableaux qui résume les articles analysée :

Auteurs	Type de plante	Nombre de classes	Nombre d'images	Techniques	Précision	Année
Srivastava et al [82]	Mendelej, Cerise	22 , 9	4590 et 2052	VGG16, InceptionV3, MobileNetV2, Xception, DenseNet121	98,10%, 98,9%, 98.40%, 96.90%, 99,9%	2023
Kaya et al [52]	14 plantes distinctes	38	54183	DenseNet	98%	2023
Sunil et al [44]	tomate	6	tomate	SVM,CNN,K-NN	88%, 97%, 99,6%	2022
Ajra et al [24]	pommes de terre, tomates	4	4000	AlexNet, ResNet-50.	96,5%,97%	2020
Audre Arlene et al[30]	tomates	5	500	Random forêt, Arbre de décision	94%,98%	2020
Ganatra et al[42]	Non spécifié	38	14956	Random Forest, SVM, K-NN, ANN	72.88%, 66.99%, 63.18%, 65.60%	2020
Zaki et al[90]	tomate	4	4671	MobileNet V2	90%	2020

TABLE 1.1 – Résumé des articles sur l'utilisation de DL et ML dans la détection et classification des maladies des plantes

1.6 Conclusion

Dans ce chapitre, nous avons abordé les concepts généraux liés aux maladies des plantes, en examinant leurs différentes causes et types, ainsi que les méthodes de détection couramment utilisées. Ensuite, nous avons exploré les systèmes de détection et de classification des maladies des plantes, décrivant les étapes impliquées dans leur mise en œuvre. Nous avons également présenté les diverses techniques de génération de caractéristiques et de classifieurs étudiées pour améliorer la détection des maladies végétales.

Chapitre 1. État de l'art sur les systèmes de détection et classification automatiques des maladies des plantes

Enfin, nous avons dressé un état de l'art des systèmes automatisés de détection et de classification des maladies des plantes, en mettant en lumière les avancées les plus récentes dans ce domaine. Dans le chapitre suivant, nous détaillerons les méthodes et techniques employées pour le développement des systèmes de détection et de classification des maladies automatiques des plantes.

Aspects théoriques sur les méthodes et techniques utilisées

2.1 Introduction

Le développement des systèmes pour détecter et classer les maladies des plantes ont beaucoup progressé récemment. Cela est dû aux progrès dans les technologies informatiques et de l'imagerie. Ces systèmes cherchent à rendre automatique le diagnostic des maladies des plantes en utilisant des méthodes de vision par ordinateur, d'apprentissage automatique et de traitement d'images.

Dans ce chapitre, nous présentons les aspects théoriques des méthodes et techniques utilisée dans ce contexte. Nous explorons les approches de segmentation pour isoler l'objet d'intérêt, les techniques d'extraction des caractéristiques et les méthodes de réduction de dimension pour simplifier les données. De plus, l'apprentissage profond, notamment les réseaux de neurones convolutifs (CNN), qui joue un rôle très important dans ce domaine.

2.2 Méthodes de Segmentation

De nombreuses techniques de segmentation d'images sont disponibles dans la littérature, que on peut les regroupés en trois catégories : Segmentation basée par contours, Segmentation basée par régions et Segmentation basée par Pixels, [69] [60].

Segmentation par contours :

De façon générale, un contour est défini comme étant la frontière entre deux régions. La détection du contour est donc équivalente à la détection de la discontinuité entre les régions en cherchant les zones de variations significatives d'intensité lumineuse ou de couleur dans l'image. Comme illustré par la figure 2.1. Il existe plusieurs méthodes de segmentation basées sur l'approche contour et qu'on peut regrouper en trois catégories : les méthodes dérivatives, par filtrage optimal et les contours actifs [73].



FIGURE 2.1 – Image segmentée par l'approche contour avec filtrage Canny

Segmentation par pixels :

Le principe est de grouper les pixels en fonction de leurs caractéristiques sans prendre en considération leur position dans l'image. Cela permet de créer des catégories de pixels ; les pixels proches, faisant partie d'une même catégorie, forment ainsi des zones. Parmi les méthodes pixelaires, on peut identifier les techniques de seuillage et les techniques de classification (clustering). [60].

1. **techniques de classification par pixels :** La classification des pixels est une opération de base d'analyse de données. Le but est de regrouper les pixels en classes caractérisées par un ensemble d'attributs. Des classes sont créées de telle sorte que les pixels de la même classe soient aussi similaires que possible et que les pixels de deux classes différentes soient aussi différents que possible. Ensuite, Une région est alors formée de pixels connexes appartenant à une même classe [73]. Il existe deux manières de classification des pixels qui sont la classification supervisée et la classification non supervisée ou (clustering). Dans la classification supervisée, on utilise des pixels prototypes classés. Cela exige une référence ou l'aide d'un expert pour étiqueter ces prototypes. Elle est habituellement utilisée pour tester la technique de classification ou les attributs utilisés.

Cependant, la classification non supervisée (ou clustering) est utilisée lorsqu'on ne sait pas si les pixels appartiennent à certaines classes spécifiques. Les techniques de clustering regroupent les pixels en fonction de leurs similitudes sans utiliser d'étiquettes pré-définies. Parmi eux figurent des algorithmes tels que K-means, FCM (Fuzzy C-means), et EM (Estimation-Maximisation), chacun ayant une approche particulière pour le regroupement des données en clusters significatifs.

2. **Technique de seuillage** : L'opération dite de « seuillage simple » consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur S (appelé seuil, en anglais treshold) et à la valeur maximale les pixels ayant une valeur supérieure, comme montré par la formule (2). Ainsi le résultat du seuillage est une image binaire contenant les pixels noirs et blanc. C'est la raison pour laquelle le terme de binarisation est parfois employé. Le seuillage permet de mettre en évidence des formes ou des objets dans une image. Toutefois, la difficulté réside dans le choix du seuil à adopter (50).

La segmentation par seuillage comporte moins de calculs que d'autres techniques ce qui implique sa facilité de mise en œuvre et son efficacité dans les systèmes en temps réel. L'inconvénient de cette technique de segmentation n'est pas adapté aux images complexes (43).

$$I_{seg}(i, j) = \begin{cases} 1 & (\text{blanc}) & I(i, j) \geq S \\ 0 & (\text{noir}) & I(i, j) < S \end{cases} \quad [37]. \text{ Il existe plusieurs approches de seuillage, notamment le seuillage}$$

3. **seuillage manuel** : Implique de définir manuellement le le seuil de luminosité (ou de couleur) en séparant les pixels en deux classes : celle qui est en dessous ou égale à la valeur du seuil et celle qui est au-dessus. Ce type de seuillage nécessite souvent une intervention humaine pour ajuster le seuil en fonction des besoins spécifiques de l'image.
4. **seuillage par intervalle** : Le seuillage par intervalle consiste à diviser l'image en plusieurs classes en spécifiant plusieurs seuils.
5. **seuillage automatique** : Le seuillage automatique utilise des algorithmes pour déterminer le seuil optimal sans nécessiter d'intervention manuelle. Ces algorithmes analysent l'histogramme de l'image (distribution des valeurs de pixels) pour trouver un seuil qui minimise une certaine fonction objectif (comme l'entropie, l'écart-type, etc.). Des méthodes populaires de seuillage automatique incluent la méthode d'Otsu,

qui trouve un seuil optimal pour la segmentation en deux classes en maximisant la variance inter-classe, et minimisant la variance intra-classe.

Segmentation par région :

L'approche région de segmentation d'image implique de séparer l'image en différentes régions. Les pixels voisins sont groupés en zones différentes en fonction d'un critère d'homogénéité ou de similarité spécifique. Il peut s'agir, par exemple, du niveau de gris, de la couleur, de la texture...etc.

On répète le processus de groupement jusqu'à ce que tous les pixels de l'image soient intégrés dans des régions spécifiques. L'objectif de cette méthode est donc de diviser l'image en fonction des caractéristiques propres aux régions [64].

2.3 Méthodes d'extraction des caractéristiques

Dans le domaine de l'extraction des caractéristiques, il existe plusieurs méthodes que nous allons examiner dans la suite de cette section.

Quantification de la phase locale (LPQ) :

Le descripteur de la phase locale aussi appelée Local Phase Quantization (LPQ) a été proposé par Ojansivu et al [67], Il permet d'améliorer la classification de textures afin d'être robuste aux artéfacts générés par le flou présent dans une image [41]. Le descripteur LPQ est construit de façon à ne garder dans une image que l'information locale invariante à un certain type de flou qui sera symétrique, qui causé généralement par le mouvement linéaire et hors du foyer du capteur [67].

Le LPQ est basée sur le calcul de la transformée de Fourier a court terme (STFT : short term Fourier transform) sur un carré N_x de $M * M$ voisins pour chaque pixel x dans l'image de texture $f(x)$ définie par l'équation (2.1). A chaque pixel, le coefficient local de Fourier est calculé pour quatre points de fréquence. Ensuite, les signes de la partie réelle et imaginaire de chaque coefficient sont quantifiés 'a l'aide d'un quantificateur scalaire binaire, afin de calculer l'information de phase. Le coefficient binaire de huit bits qui en résulte est ensuite représenté sous forme d'entiers par codage binaire. Au final un vecteur de caractéristiques a 256 dimensions

est obtenue [41] comme illustrée par la figure 2.2.

$$F(u, x) = \sum_{v \in x} f(x, y) e^{2\pi j u^T y} = W_u f_x \tag{2.1}$$

Ou u est la fréquence, et x est une fonction de fenêtre définissant le voisinage N .

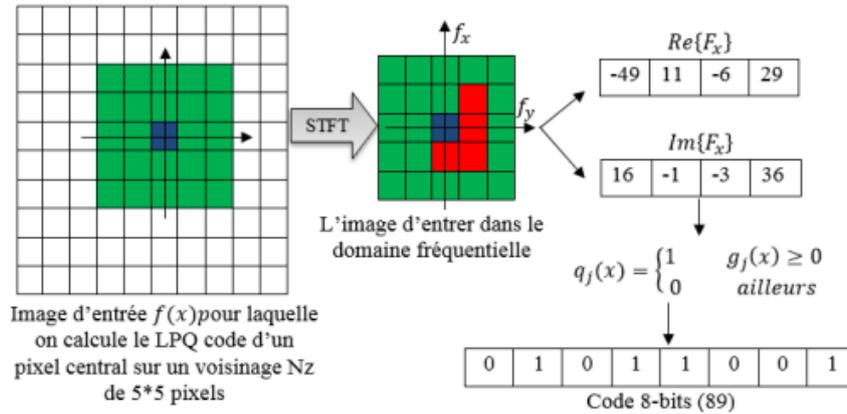


FIGURE 2.2 – Organigramme de l'ensemble des étapes nécessaires à la construction du descripteur LPQ

Motifs binaires locaux (LBP) :

L'opérateur LBP, qui signifie « Local Binary Patterns », a été proposée pour la première fois par Ojala et al en 1995 [45]. Il s'agit d'une méthode très efficace pour décrire les textures.

- LBP est initialement défini dans un voisinage de huit pixels, et la valeur de gris du pixel central est définie comme seuil. Tous les voisins dont la valeur est supérieure ou égale à la valeur du pixel central reçoivent la valeur 1, sinon ils sont définis sur 0. Les valeurs après le seuillage (à savoir 0 ou 1) seront respectivement multipliées par le poids de pixel correspondant, et leur résultat additif est la valeur LBP [53]. Le principe de calcul de LBP d'origine (8 voisins) est illustré à la figure 2.3

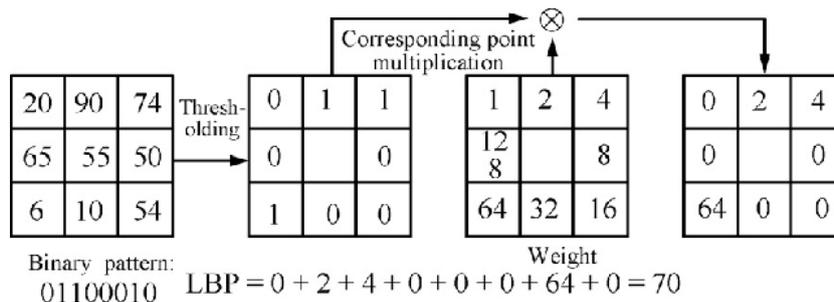


FIGURE 2.3 – Illustration de calcul d'un LBP de base [53].

Le LBP de base est défini par :

$$LBP(x_c, y_c) = \sum_{p=0}^{p-1} S(g_p - g_c) * 2^p \quad (2.2)$$

Où :

g_c est le niveau de gris du pixel central de coordonnées (x_c, y_c) .

g_p ($p = 0, 1, \dots, 7$) est le niveau de gris de chaque pixel voisin.

Avec : $S(x)$ une fonction définie comme suit :

$$S(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon.} \end{cases}$$

caractéristiques des images statistiques binarisées (BSIF) :

Contrairement à LBP et LPQ qui permettent de calculer les statistiques d'étiquettes dans les voisinages des pixels locaux, le descripteur local dit BSIF (Binarized Statistical Image Features), récemment proposé par Kannlaand Rahtu en 2012, offre une alternative. Utilise manuellement un ensemble préétabli de filtres linéaires et binarise les réponses du filtre [67].

Le principe est de calculer une chaîne binaire pour les pixels d'une image d'entrée. La valeur de code d'un pixel est considérée comme un descripteur local du motif d'intensité d'image dans l'environnement du pixel plus loin.

Les histogrammes des valeurs de code des pixels permettent de caractériser les propriétés de texture au sein des sous régions d'images.

La valeur de chaque élément (bit) dans la chaîne de code binaire est calculée en binarisation de la réponse d'un filtre linéaire avec un seuil à zéro. Chaque bit est associé à un filtre différent et la longueur souhaitée de la chaîne de bits détermine le nombre de filtres utilisés. L'ensemble de filtres est appris à partir d'un ensemble d'apprentissage de correctifs d'image naturelle en maximisant l'indépendance statistique des réponses de filtre. Par conséquent, les propriétés statistiques des correctifs d'image naturels déterminent les descripteurs et par conséquent, les appelons caractéristiques d'image statistique binarisées (BSIF). Supposant une image d'entre X de taille $l * l$ et un filtre linéaire W_i de la même taille la réponse de filtre est donnée par :

$$S_i = \sum_{u,v} W_i(u, v) X(u, v) = w_i x \quad (2.3)$$

D'où la notation de vecteur est introduite dans la dernière étape, à savoir, les vecteurs w et x contiennent les pixels de W_i b_i est calculé par la proposition ci-dessous :

$$b_i = \begin{cases} 1 & \text{si } S_i > 0 \\ 0 & \text{si } S_i \leq 0 \end{cases}$$

Le descripteur BSIF possède deux paramètres qui sont : la taille du filtre et la longueur n de la chaîne binaire.

2.4 Réduction de dimension

Augmenter la dimensionnalité des données fait référence aux défis rencontrés lors de l'analyse de multiples variables. Lorsque nous manipulons de vastes ensembles de données, il est souvent nécessaire de réduire leur complexité. L'objectif de cette réduction est de représenter les données dans un espace à moindre dimension tout en préservant certaines caractéristiques clés. Diverses techniques peuvent être employées à cet effet, comme l'utilisation non supervisée de l'Analyse en Composantes Principales (ACP) et la méthode supervisée d'Analyse Discriminante Linéaire (LDA), que nous définirons plus en détails dans la suite de cette section. Ces approches sont appliquées dans divers domaines tels que l'extraction d'informations, les bases de données, la statistique, la reconnaissance des formes, le traitement du langage naturel, la visualisation des données et le machine learning. La réduction de dimensionnalité s'avère particulièrement utile pour explorer les jeux de données contenant des images [74].

2.4.1 Méthodes de réduction de dimension

Parmi les méthodes couramment utilisées on trouve :

Analyse en Composantes Principales (PCA) :

L'algorithme ACP, en anglais PCA (Principal Component Analysis) est né des travaux de Matthew et al en 1991 [85], est une méthode de réduction de dimensionnalité non supervisée. L'ACP cherche à réduire la dimension des données en trouvant quelques combinaisons linéaires orthogonales (les composantes principales CP, ou axes principaux) des variables d'origine ayant la plus grande variance. Selon le nombre de variables d'origine, il y a autant de CP. Les premiers CP expliquent la majeure partie de la variance, de sorte que le reste peut être négligé avec une

perte minimale d'information, pour de nombreux ensembles de données [70].

PCA est une méthode rapide, facile et répandue pour identifier les modèles, étant l'une des techniques les plus performantes et elle est la méthode la plus efficace pour reconstruire une base de dimension réduite [68]. L'ACP peut nous aider à améliorer les performances à un coût très faible pour la précision du modèle. Parmi les autres avantages de l'ACP, citons la réduction du bruit dans les données, la sélection des caractéristiques (dans une certaine mesure) et la possibilité de produire des caractéristiques indépendantes et non corrélées des données.

Les étapes à suivre pour effectuer une Analyse en Composantes Principales sur des données sont les suivantes :

1. Obtenir les données de l'image : Supposons $x_1, x_2 \dots x_M$ sont représentés par des vecteurs $N \times 1$.
2. Calculer la moyenne du vecteur :

$$\bar{x} = \frac{1}{M} \sum_{i=0}^M x_i \quad (2.4)$$

3. Centrer les données : en soustrayons l'image moyenne de chaque vecteur image, par la formule (2.5) :

$$\phi = x_i - \bar{x} \quad (2.5)$$

4. Calculer la matrice de covariance : des données, cette matrice capture les relations linéaires entre les différentes dimensions des vecteurs (2.6).

$$\sum = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (2.6)$$

5. On calcule ensuite les vecteurs propres V et les valeurs propres D de la matrice C. Les valeurs propres de la matrice C représentent le taux de variation le long de l'axe du vecteur propre associé. La matrice de transformation de PCA est donc les R premiers vecteurs propres ordonnés par ordre décroissant des valeurs propres correspondantes

Analyse discriminante linéaire (LDA) :

Linear Discriminant Analysis ou LDA a été proposée par R. Fischer en 1936 [88], est une technique supervisée de réduction de la dimensionnalité basée sur une projection linéaire de l'espace de haute dimension vers un espace de faible dimension en maximisant la dispersion entre les classes et en minimisant la dispersion intra-classe. Contrairement à la PCA qui ne tient

pas compte des différences entre les classes, l'algorithme LDA assure une véritable séparation des classes. LDA est également connu sous le nom de discriminant linéaire de Fisher, Il est principalement utilisé comme technique de réduction de dimension des vecteurs de caractéristiques sans perte d'informations avant la classification [74]. LDA définit deux mesures pour tous les échantillons de toutes les classes :

1. Matrice de dispersion intra-classe :

$$S_w = \sum_{j=1}^c \sum_{i=1}^N (x_i^j - \mu_j)(x_i^j - \mu_j)^T \quad (2.7)$$

Où c est le nombre de classes, x_i^j est le i ème échantillon de la classe j , μ_j est la moyenne de la classe j et N le nombre d'échantillons de la classe j .

2. Matrice de dispersion inter-classe :

$$S_b = \sum_{j=1}^c (\mu_j - \mu)(\mu_j - \mu)^T \quad (2.8)$$

Où μ représente la moyenne de toutes les classes.

L'objectif de LDA, est de trouver une projection des données optimale qui maximise le rapport entre la dispersion inter-classes S_b et la dispersion intra-classes S_w voir l'équation (2.9). Cela revient à maximiser les variances des données appartenant à des classes différentes tout en minimisant les variances des données appartenant à la même classe.

$$W_{(opt)} = \operatorname{argmax} \frac{W S_b W^T}{W S_w W^T} \quad (2.9)$$

Dans la figure 2.4 Trois distributions 3D sont projetées sur deux sous-espaces 2D décrits par les vecteurs $W1$ et $W2$. Puisque le LDA essaye de trouver la plus grande séparation parmi les classes, on voit bien que $W1$ est ici le vecteur optimal.

2.5 Apprentissage profond (Deep learning)

2.5.1 Réseaux de neurones artificiels

Ces dix dernières années, l'utilisation des réseaux de neurones artificiels (ANN) s'est développée dans de nombreux domaines (économie, écologie et environnement, biologie et médecine,

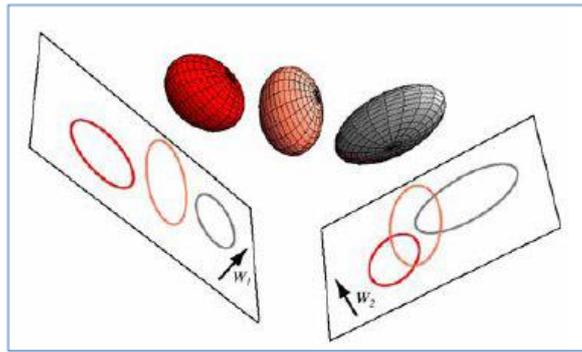


FIGURE 2.4 – Illustration du principe de séparation optimale des classes par le LDA [34].

etc.). Ils sont principalement utilisés pour résoudre des problèmes de classification, de prédiction, de classification, d'optimisation, de reconnaissance de formes et de mémoire associative [11].

Les ANN, développés comme modèles mathématiques de systèmes nerveux biologiques, ont suscité un intérêt après l'introduction de neurones simplifiés par McCulloch et Pitts en 1943 [29].

2.5.2 Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN) sont une forme particulière de réseau neuronal multi-couche dont l'architecture des connexions est inspirée de celle du cortex visuel des mammifères [19]. Dans le domaine de deep learning, le CNN est l'algorithme le plus célèbre et le plus couramment utilisé. Le principal avantage du CNN par rapport à ses prédécesseurs est qu'il identifie automatiquement les caractéristiques pertinentes sans aucune supervision humaine. Les CNN ont été largement appliqués dans un éventail de domaines différents, notamment la vision par ordinateur, le traitement de la parole, la reconnaissance faciale, etc. La structure des CNN a été inspirée par les neurones du cerveau humain et animal, similaire à un réseau neuronal classique. Plus précisément, dans le cerveau d'un chat, une séquence complexe de cellules forme le cortex visuel, cette séquence est simulée par le CNN. Goodfellow et al. ont identifié trois avantages clés du CNN : les représentations équivalentes, les interactions clairsemées et le partage des paramètres [27].

CNN est une construction mathématique qui est généralement composée de trois types de couches (ou blocs de construction) : la convolution, la mise en commun et les couches entièrement connectées. Les deux premières, les couches de convolution et de regroupement, effectuent l'extraction des caractéristiques, tandis que la troisième, une couche entièrement connectée, mappe les entités extraites dans la sortie finale, telle que la classification [89]. Un

exemple d'architecture CNN pour la classification d'images est illustré à la figure 2.5.

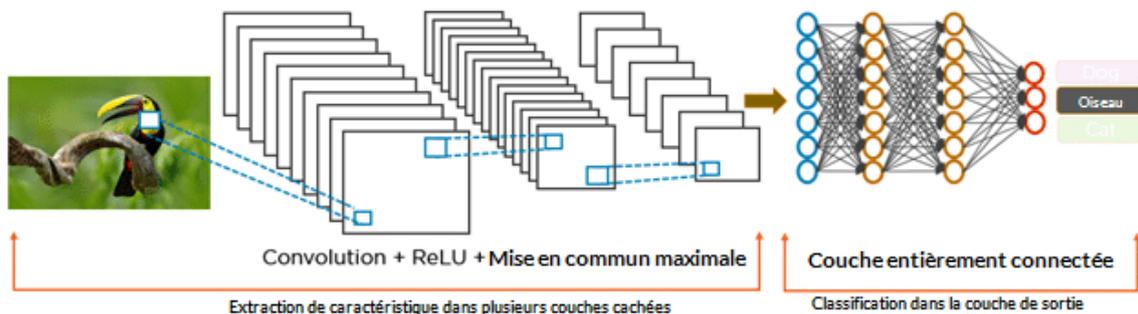


FIGURE 2.5 – Architecture générale d'un CNN [31].

2.5.2.1 Couche d'entrée (Input layer)

La couche d'entrée (couche la plus à gauche) représente l'image d'entrée dans le CNN, est organisée en trois dimensions comme illustrée par la figure 2.6 : hauteur, largeur et profondeur, ou $m \times m \times r$, où la hauteur (m) est égale à la largeur, et La profondeur (r) est également appelée numéro de canal. Par exemple, dans une image RVB, la profondeur (r) est égale à trois canaux, correspondant respectivement aux canaux rouge, vert et bleu.

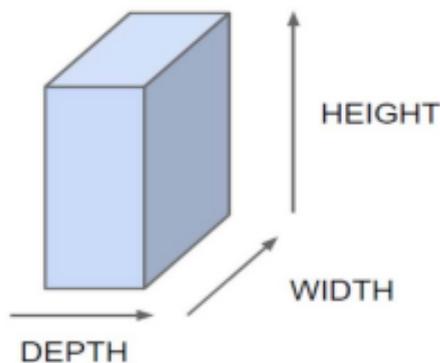


FIGURE 2.6 – Les 3 dimension de l'entrée d'un CNN [26].

2.5.2.2 Couche de convolution (Convolutional Layer)

Dans l'architecture CNN, le composant le plus important est la couche convolutive. Il s'agit d'un ensemble de filtres convolutifs (appelés noyaux). L'image d'entrée, exprimée sous forme de métriques à N dimensions, est convoluée avec ces filtres pour générer la carte d'entités en sortie.

Le format vectoriel est l'entrée du réseau neuronal traditionnel, tandis que l'image multi-canal est l'entrée du CNN. Par exemple, le format à un seul canal est celui de l'image en

niveaux de gris, tandis que le format d'image RGB est à trois canaux. Pour comprendre l'opération convolutive, prenons un exemple d'une image en niveaux de gris de $5 \times 5 \times 1$ avec un noyau de poids aléatoires initialisé de taille $3 \times 3 \times 1$. Tout d'abord, le noyau glisse sur toute l'image horizontalement et verticalement. De plus, le produit scalaire entre l'image d'entrée et le noyau est déterminé, où leurs valeurs correspondantes sont multipliées puis sommées pour créer une seule valeur scalaire, calculée simultanément. Tout le processus est ensuite répété jusqu'à ce qu'aucun autre glissement ne soit possible. Notez que les valeurs du produit scalaire calculées représentent la carte des caractéristiques de la sortie. La figure 2.7 illustre l'exemple de l'opération de convolution.

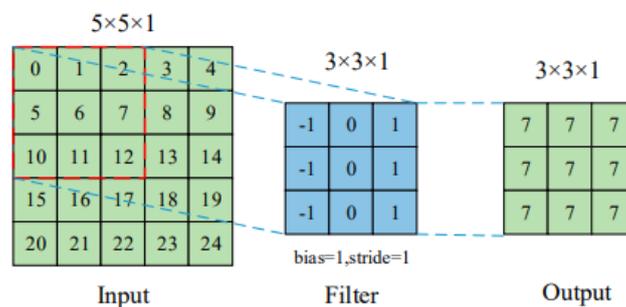


FIGURE 2.7 – Opération de convolution [49]

2.5.2.3 Couche de pooling (Pooling layer)

La couche de pooling : La tâche principale de la couche de pooling est le sous-échantillonnage des cartes des caractéristiques. Ces cartes sont générées en suivant les opérations de convolution. En d'autres termes, cette approche réduit la taille des cartes des caractéristiques de grande taille pour créer des cartes de caractéristiques plus petites. Parallèlement, elle conserve la majeure partie des informations dominantes (ou des caractéristiques) à chaque étape de la phase de pooling. De manière similaire à l'opération de convolution, à la fois le pas et le noyau sont initialement attribués une taille avant que l'opération de pooling ne soit exécutée. Plusieurs types de méthodes de pooling sont disponibles pour être utilisés dans différentes couches de pooling. Ces méthodes incluent le pooling arborescent, le pooling filtré, le pooling moyen, le pooling minimal, le pooling maximal, le pooling moyen global (GAP) et le pooling maximal global. Les méthodes de pooling les plus familières et les plus fréquemment utilisées sont le pooling maximal, minimal et GAP. La figure 2.8 illustre ces trois opérations de pooling.

Parfois, les performances globales du CNN sont diminuées en conséquence ; cela représente la principale lacune de la couche de pooling, car cette couche aide le CNN à déterminer si une

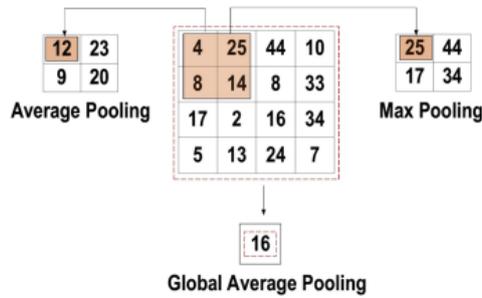


FIGURE 2.8 – Les 3 types de l’opération de pooling [27]

certainne caractéristique est présente dans l’image d’entrée donnée, mais se concentre exclusivement sur l’identification de l’emplacement correct de cette caractéristique. Ainsi, le modèle CNN passe à côté des informations pertinentes.

2.5.2.4 Couche d’Aplatissement (Flatten layer)

Dernière étape de la partie « extraction des informations », l’aplatissement consiste simplement à mettre bout à bout toutes les images (matrices) que nous avons pour en faire un (long) vecteur. Les pixels (en réalité ce ne sont plus des images ou des pixels, mais des matrices de nombres, donc les pixels sont ces nombres) sont récupérés ligne par ligne et ajoutés au vecteur final [5], comme le montre la figure 2.9.

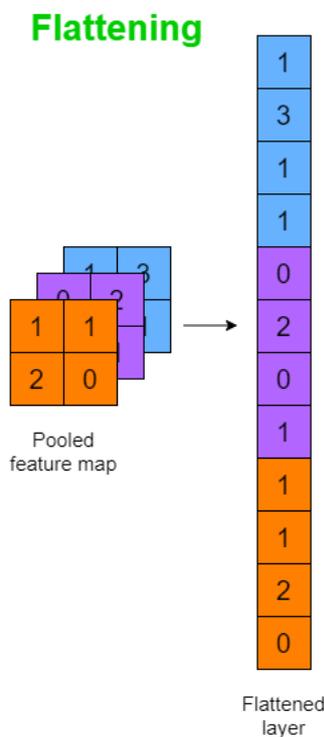


FIGURE 2.9 – Mécanisme d’aplatissement [5]

2.5.2.5 Couche entièrement connectée (Fully-connected layer)

Cette couche est généralement située à la fin de chaque architecture CNN. À l'intérieur de cette couche, chaque neurone est connecté à tous les neurones de la couche précédente, ce qu'on appelle l'approche totalement connectée (FC). Elle est utilisée comme classifieur CNN. Elle suit la méthode de base du réseau neuronal à couches multiples perceptron, car il s'agit d'un type de réseau neuronal à propagation avant. L'entrée de la couche FC provient de la dernière couche de pooling ou de convolution. Cette entrée est sous forme de vecteur, qui est créé à partir des cartes des caractéristiques après aplatissement. La sortie de la couche FC représente la sortie finale du CNN.

2.5.2.6 Paramètres d'un CNN

En réalité, les CNN proposent davantage d'options qui offrent de nombreuses possibilités pour diminuer encore plus les paramètres et, en même temps, diminuer certains des effets secondaires. Dans ce qui suit, nous allons examiner ces options.

- **Filtre** : La convolution consiste à glisser une série de filtres ou de noyaux de convolution sur l'image d'entrée. La sélection des filtres, de leur taille et de leur nombre (également appelé profondeur) est une tâche expérimentale. La « profondeur de la couche convolutive » fait référence au nombre de noyaux utilisés, d'où vient le nom « apprentissage en profondeur ». En utilisant un numéro de filtre de q , l'image est convoluée q fois. Ce processus crée q matrices filtrées appelée «carte des caractéristiques». Un exemple de filtrage est présenté dans la figure 2.10. Pour la plupart des architectures CNN prédéfinies, il existe toujours une sélection par défaut spécifique à l'application pour laquelle le modèle a été construit [57].

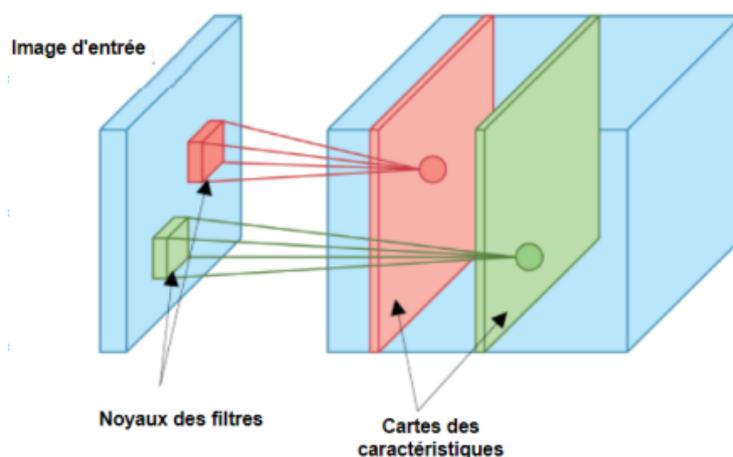


FIGURE 2.10 – Cartes de caractéristiques [66].

- **Stride** : Le stride est la quantité de décalage du filtre à chaque étape lors de la convolution. Il détermine le nombre de pixels par lesquels la fenêtre de convolution se déplace après chaque opération. Un stride plus grand réduit la taille de l'image en sortie, tandis qu'un stride plus petit conserve plus d'informations spatiales, comme l'indique la figure 2.11 [26].

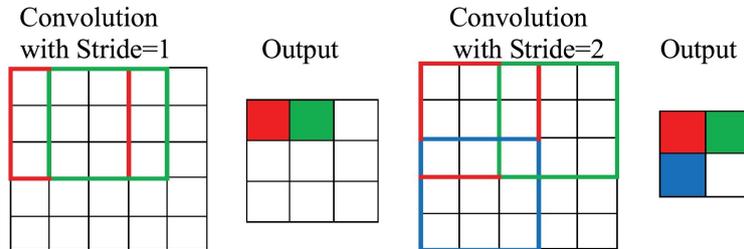


FIGURE 2.11 – Principe du stride [18].

- **Zéro padding** : Les inconvénients de l'étape de convolution est la perte d'information qui pourrait exister sur le bord de l'image. Parce qu'ils ne sont capturés que lorsque le filtre glisse, ils n'ont jamais la chance d'être vus, et la réduction de la taille de l'image après la convolution. Une méthode très simple mais efficace pour résoudre le problème est d'utiliser le zéro padding (ajouter des zéros aux contours de l'image, en tenant compte de la taille de filtre) [26].

— **Les fonctions d'activation** :

1. **Sigmoid** : L'entrée de cette fonction d'activation est constituée de nombres réels, tandis que la sortie est limitée entre zéro et un [27]. La courbe de la fonction sigmoïde est en forme de S et peut être représentée mathématiquement par l'équation (2.10).

$$F(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

2. **Tanh** : Il est similaire à la fonction sigmoïde, car son entrée est constituée de nombres réels, mais la sortie est limitée entre -1 et 1 [27]. Sa représentation mathématique est donnée par l'équation (2.11).

$$F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.11)$$

3. **ReLU** : La fonction la plus couramment utilisée dans le contexte des CNN. Elle convertit toutes les valeurs d'entrée en nombres positifs. La principale bénéfice de ReLU par rapport aux autres est une charge computationnelle réduite [27]. Sa re-

présentation mathématique est donnée par l'équation (2.12)

$$F(x) = \max(0, x) \quad (2.12)$$

4. **Softmax** : Souvent employée dans la couche finale du réseau, cette fonction agit comme une normalisation qui fournit une estimation de la probabilité qu'une classe soit correcte [27]. Elle calcule la valeur de probabilité comme suit :

$$F(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}, \quad i = 1, 2, \dots, K \quad (2.13)$$

où $\mathbf{x} = (x_1, x_2, \dots, x_K)$ est un vecteur de nombres réels et K est le nombre de classes.

2.5.2.7 Exemples d'architectures des CNN les plus célèbres :

Au cours des dernières années, de nombreuses architectures de CNN ont été développées, jouant un rôle crucial dans l'amélioration des performances des applications variées. Des évolutions majeures ont été apportées à ces architectures, notamment des réorganisations structurelles, des techniques de régularisation et d'optimisation des paramètres. La profondeur du réseau est devenue un élément central des améliorations apportées aux CNN. dans ce qui suit on vas cites quelques modelés CNN les plus célèbres.

AlexNet :

AlexNet a été l'un des premiers modèles qui popularisé les CNN dans la vision par ordinateur, développée par Alex Krizhevsky, Ilya Sutskever et Geoff Hinton en 2012. Le réseau avait une architecture plus profonde, plus grande et se composait de huit couches : cinq couches convolutifs empilées les unes sur les autres (auparavant, il était commun de ne disposer que d'une seule couche convolutifs toujours immédiatement suivie d'une couche de pooling), deux couches cachées entièrement connectées et une couche de sortie entièrement connectée [55].

GoogleNet (Inception) :

GoogLeNet, également connu sous le nom d'Inception, est développé par l'équipe de recherche Google en 2014, il utilise des modules Inception, des blocs de convolutions en parallèle avec différentes tailles de filtres (1x1, 3x3, 5x5) et des couches de réduction dimensionnelle intermédiaires, appelées "bottleneck layers", qui réduisent le nombre de canaux avant d'appliquer

des convolutions plus coûteuses en termes de calcul [84].

ResNet :

ResNet a été introduit pour la première fois en 2015, où il a également remporté le concours ILSVRC avec un taux d'erreur de 3,57 %. Le taux de précision élevé de ResNet peut être principalement attribué à l'introduction de couches résiduelles qui permettent de concevoir le réseau plus profondément par rapport aux architectures de réseau populaires précédentes. La couche résiduelle ou également connue sous le nom de cartographie d'identité atténue le problème de la diminution du gradient lors de l'entraînement d'un réseau profond, où la couche précédente est alimentée par les couches suivantes. L'idée était de surmonter la réduction des caractéristiques d'entrée par rapport à la fonction d'apprentissage originale qui ne produit aucune caractéristique [47].

MobileNet V2 :

MobileNet est une architecture de deep learning qui se concentre sur la plate-forme mobile où les ressources de calcul sont limitées. Une version améliorée, appelée MobileNet V2, est ensuite introduite par Google avec de légères modifications par rapport à la version originale. La base du réseau reste toujours la même, qui est la convolution séparable. La version 2 de MobileNet, précédemment entraînée sur des ensembles de données ImageNet, a été utilisée pour extraire les caractéristiques d'image des fruits dans [75].

2.6 Algorithmes de classification

Les algorithmes d'apprentissage automatique les plus couramment utilisés pour la classification des maladies des plantes mentionnés dans la littérature sont :

Classifieur de K-plus proche voisin(KNN) :

Le classificateur KNN est un modèle qui permet de catégoriser des données inconnues en les associant à des données connues en utilisant des mesures de similarité. Cet algorithme fonctionne en déterminant la classe majoritaire parmi les k voisins les plus proches d'une instance inconnue. Il s'agit d'une méthode simple basée sur le principe du vote majoritaire parmi les voisins les plus proches [87].

KNN est adapté aux problèmes de classification et peut également être utilisé pour des tâches de régression. En pratique, il est souvent privilégié dans l'industrie pour les problèmes de classification [80]. Les mesures de similarité utilisées pour déterminer les voisins les plus proches peuvent être basées sur différentes distances telles que la distance euclidienne voir l'équation 2.14.

$$E(i, j) = \sqrt{\sum_{k=1}^n (i_k - j_k)^2} \quad (2.14)$$

La figure 2.12 représente un exemple de classification par K plus proches voisins

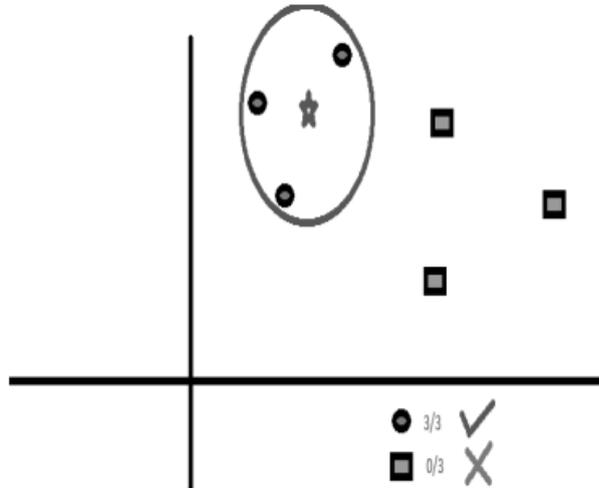


FIGURE 2.12 – Diagramme de bloc de K-Voisin le plus proche [80].

La machine à vecteur du support(SVM) :

Le SVM à été utilisée pour résoudre divers problèmes de régression et de classification [36]. Le principe de fonctionnement de SVM est de trouver la meilleure fonction de classification qui distingue les membres de deux classes dans les données d'apprentissage. Pour des données linéairement séparables, cette fonction de classification linéaire correspond à un hyperplan de séparation $f(x)$, qui passe entre les deux classes et les sépare. Ce dernier peut être représenté par l'équation suivante :

$$f(x) = w \cdot x + b \quad (2.15)$$

où w est le vecteur de poids et b est le biais. Une nouvelle instance x_n peut être classée en testant simplement le signe de $f(x_n)$:

- x_n appartient à la classe positive si $f(x_n) > 0$
- x_n appartient à la classe négative si $f(x_n) < 0$

Les SVM garantissent que la meilleure fonction de ce type est trouvée en maximisant la marge γ , définie géométriquement comme la distance la plus courte entre les points de données les plus proches (vecteurs de support) et l'hyperplan. La marge est donnée par :

$$\gamma = \frac{2}{\|w\|} \quad (2.16)$$

Le problème d'optimisation des SVM consiste à minimiser $\|w\|^2$ tout en respectant les contraintes de classification pour les vecteurs de support.

La figure 2.13 représente un exemple de problème séparable dans un espace à 2 dimensions.

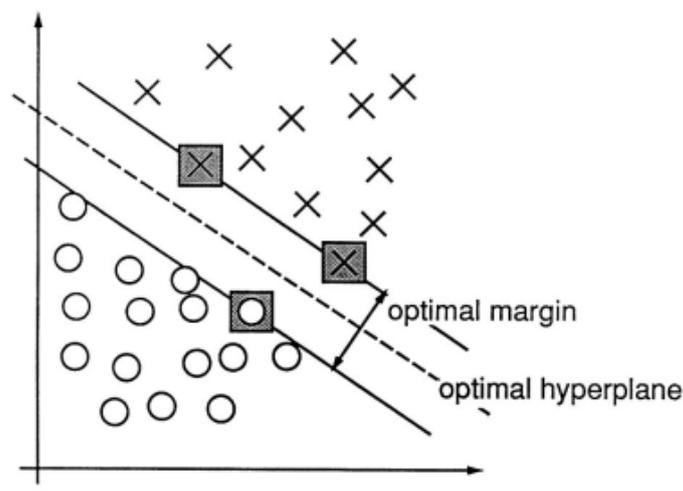


FIGURE 2.13 – Classification par SVM : Hyperplan Optimal et Vecteurs de Support [36].

Les arbres de décisions, Random Forest

Le Random Forest est un algorithme d'apprentissage machine basé sur les arbres de décision. [71]. Un arbre de décision est considéré comme une méthode d'apprentissage automatique qui prend des décisions en organisant un ensemble de règles hiérarchiques. Il crée un modèle de prédiction où chaque nœud est utilisé pour représenter un attribut de caractéristiques des données, chaque branche indique la décision basée sur cet attribut, et chaque feuille est utilisée pour représenter la prédiction ou la décision finale. En présence de nouvelles données, elles peuvent être envoyées à l'arbre en passant de nœud en suite de nœuds jusqu'à ce qu'ils atteignent la feuille où la prise de décision a lieu.

La figure 2.14 représente un arbre de décision d'apprentissage automatique pour la prévision de la température.

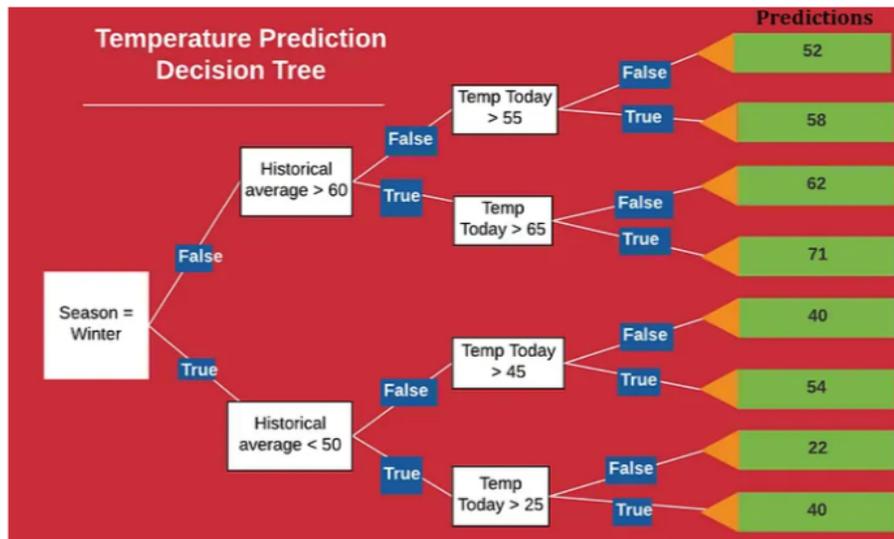


FIGURE 2.14 – Illustration d’un arbre de décision de Williom Koerhsen [6].

Random Forest (RF) traduit en français les forêts aléatoires introduits par Breiman en 2001, qui a défini les forêts aléatoires comme une famille des méthodes d’ensemble [32], c’est une combinaison de plusieurs arbres de décisions, chacun formé sur un sous-ensemble aléatoire des données d’entraînement, puis en agrégeant leurs prédictions pour obtenir une prédiction finale plus robuste et moins sujette au sur-ajustement.

2.7 Conclusion

Dans ce chapitre, nous avons introduit les aspects théoriques des méthodes et techniques utilisées pour la détection et classification des maladies des plantes à partir d’images de feuilles. nous avons exploré l’évolution des réseaux neuronaux artificiels, en soulignant les réseaux neuro-convolutionnaires et leurs principaux paramètres. Nous avons également examiné des architectures pré-entraînées, qui servent de modèles CNN pour la classification des maladies végétales.

Dans le chapitre suivant, nous détaillerons notre approche utilisée pour développer le système de diagnostic des maladies végétales.

Approche proposée

3.1 Introduction

La détection et la classification automatiques des maladies des plantes d’après les images des feuilles posent un grand défi dans les domaines de la vision par ordinateur et de l’agriculture de précision. En raison de l’importance accrue de la sécurité alimentaire et de la nécessité de minimiser les pertes de rendement, on doit concevoir des systèmes intelligents qui peuvent identifier les maladies des plantes de manière rapide et précise.

Les maladies des plantes se manifestent par divers symptômes sur les feuilles, tels que les taches, les décolorations ,et les déformations. Ces symptômes varient considérablement en fonction de la maladie, des espèces végétales et des conditions environnementales, ce qui complique la tâche de détection et de la classification. Le système présenté dans ce chapitre utilise des techniques avancées de traitement d’images et de l’apprentissage automatique. L’approche proposée combine des méthodes d’extraction des caractéristiques, de réduction de dimension et de classification pour analyser les images de feuilles et détecter la plante malade de la saine, et identifie la maladie spécifique.

La figure [3.1](#) illustre un schéma général de notre approche, mettant en évidence les étapes clés de la collecte des données à la classification finale.

3.2 Description de l’Architecture

L’architecture de notre système est conçues en deux partie distinctes mais complémentaires. Un modèle pour la détection de la maladie à pour objectif de déterminer si une plante est saine ou malade, et un modèle d’identification de la maladie spécifique parmi plusieurs classes prédéfinies. Les deux sous-systèmes passent par 6 étapes : collection des données, pré-traitement,

extraction des caractéristiques, réduction de dimension, fusion et classification finale, en vas les voir en détaille dans la suite de ce chapitre.

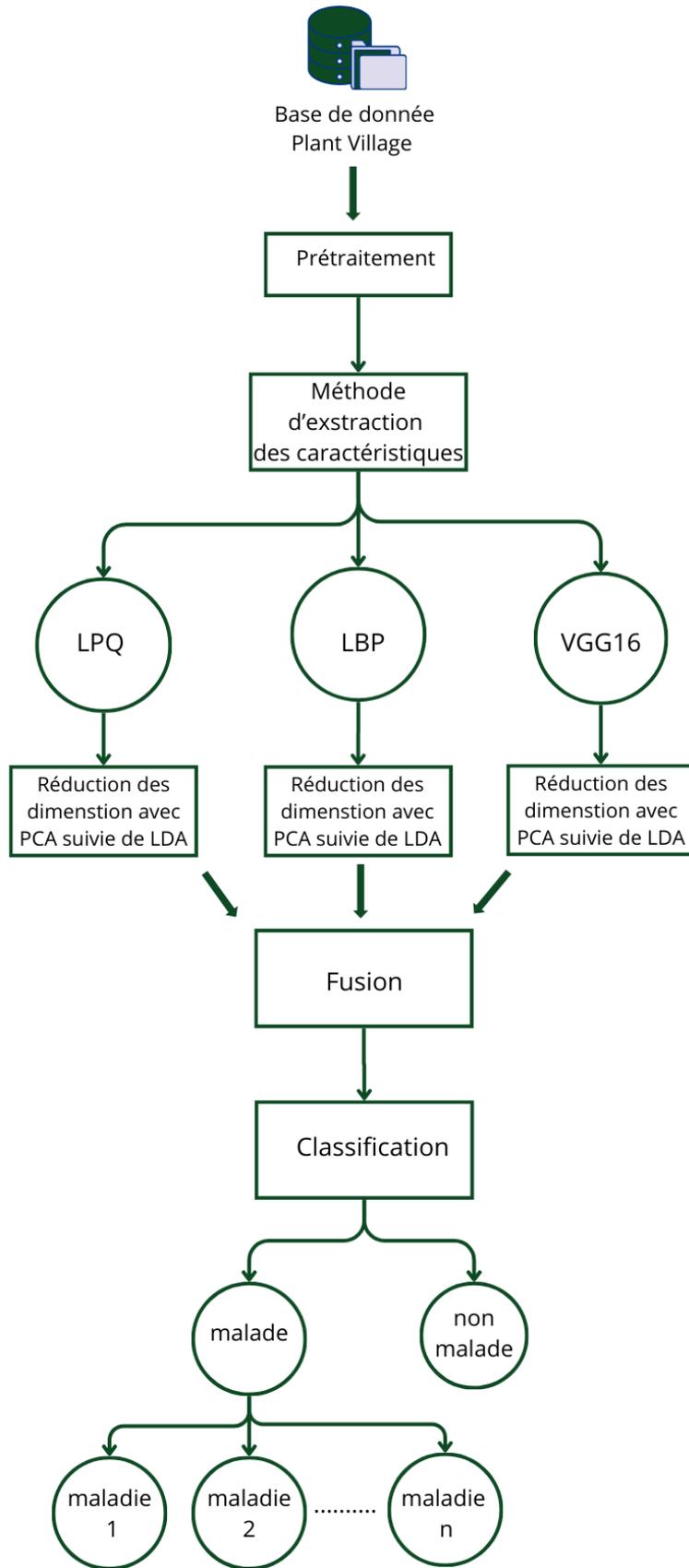


FIGURE 3.1 – Schéma récapitulatif de notre système

3.2.1 Pré-traitement

Le pré-traitement des images est une étape importante pour améliorer la qualité des données et préparer les images pour l'extraction des caractéristiques et la classification. De ce fait 5 pré-traitement en été réaliser sur notre jeux de donnée à savoir :

3.2.1.1 Suppression de bruit

En traitement d'images, il est souvent nécessaire d'éliminer le bruit qui peut dégrader la qualité de l'image et rendre son interprétation visuelle difficile [65]. Pour remédier à ce problème, des techniques de filtrage sont utilisées pour supprimer ces perturbations tout en préservant les informations importantes de l'image, comme les contours, les détails et les textures. Dans ce travail, nous avons utilisé un filtre gaussien de taille 5X5.

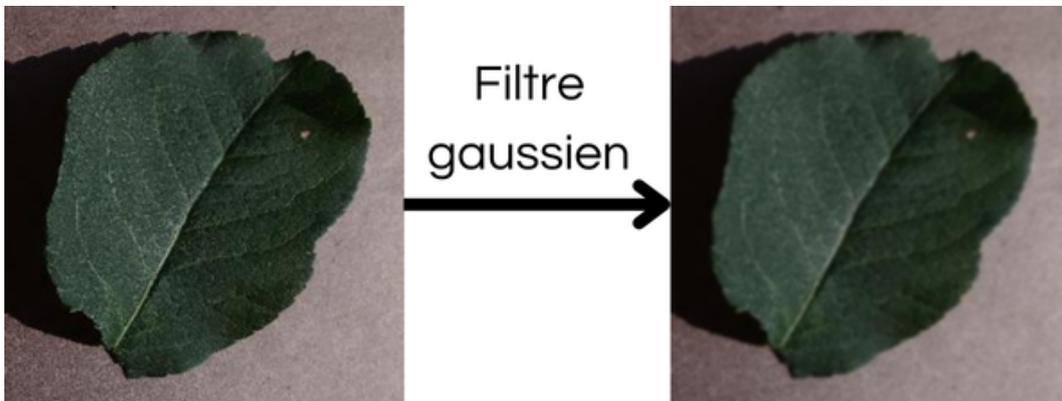


FIGURE 3.2 – Exemple de filtre gaussien appliqué sur les images

3.2.1.2 Segmentation et Réduction des Dimensions

La segmentation est une méthode curiale dans diverses applications de la vision par ordinateur, comme la reconnaissance d'objets, la détection de contours, et la classification d'images. Elle consiste à séparer et isoler l'objet d'intérêt du fond, ce qui facilite le traitement sur les images. Il existe plusieurs techniques de segmentation telle que la segmentation par région, par contour et par seuillage. Dans notre travail, nous avons opté pour l'utilisation de la segmentation par seuillage (seuil) en raison de sa simplicité de calcul et d'implémentation, plus précisément la méthode d'Otsu. la figure 3.3 (b) illustre un exemple de segmentation par seuillage avec la méthode d'Otsu, après avoir appliqué les opérateurs d'ouvertures suivie d'une itération de la dilatation.

Méthode otsu :

Elle permet d'effectuer la segmentation de niveaux de gris en une image binaire. Le principe de cette méthode est de trouver le seuil qui minimise la variance intra-classe pondérée σ_w^2 (la variance à l'intérieure des classes) ou de manière équivalente, qui maximise la variance inter-classe (variance entre les classes) des pixels de l'image.

Après segmentation, nous avons déterminé les dimensions de l'objet segmenté et les avons appliquées à l'image originale. L'objectif de cette approche était de supprimer autant de pixels que possible qui ne contiennent pas d'informations pertinentes, en conservant uniquement les parties essentielles de l'image pour l'analyse ultérieure comme illustrée par la figure ci-dessous.

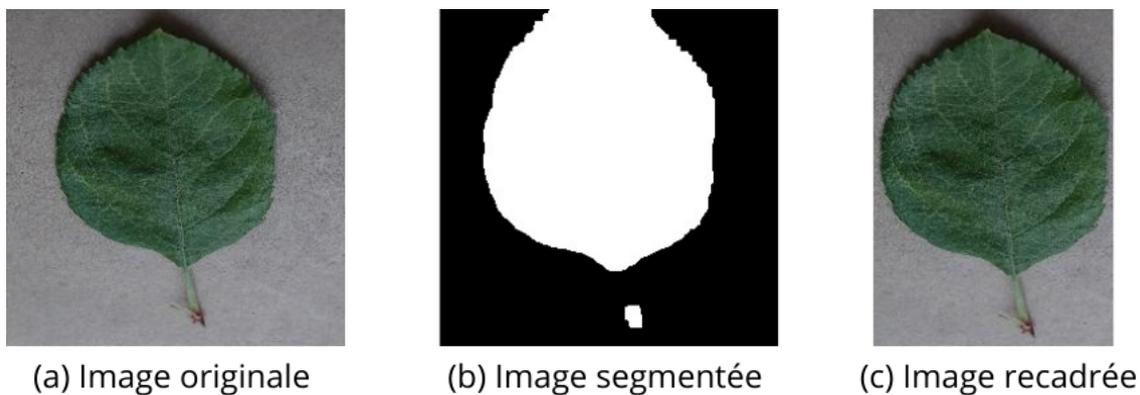


FIGURE 3.3 – Exemple de segmentation et réduction de dimension

3.2.1.3 Redimensionnement

Après la segmentation et réduction des dimensions, nous avons obtenu des images de différentes tailles. Pour uniformiser ces images, nous les avons toutes redimensionnées à une taille standard de 224x224 pixels en fonction du modèle utilisé pour l'extraction des caractéristiques.

3.2.1.4 Normalisation

La normalisation de l'image, consiste à ajuste les valeurs des pixels de manière à ce qu'elles aient une distribution commune (entre 0 et 1).

3.2.1.5 Augmentation des données

L'augmentation est une technique très utile lors d'entraînement d'un modèle, qui permet de générer de nouvelles images d'entraînement à partir d'images existantes en appliquant des transformation telle que le zoom, la rotation, l'inversion...etc.

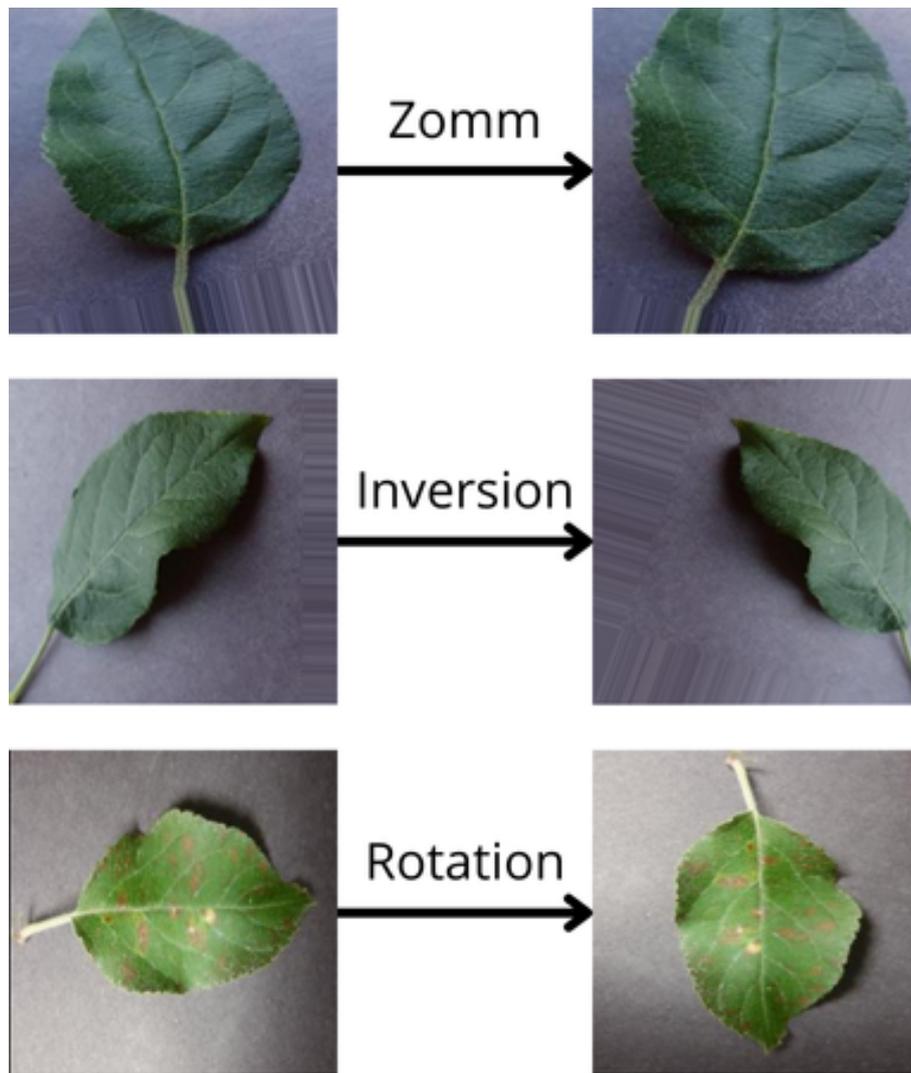


FIGURE 3.4 – Exemples de transformation faites sur les images

3.2.2 Extraction des caractéristiques

Après les pré-traitements faites sur les images de feuilles, trois (3) méthodes d'extraction des caractéristiques LPQ, LBP et VGG16 sont appliquées pour obtenir les caractéristiques qui seront utilisée par la suite pour la classification.

3.2.2.1 Extraction des caractéristiques avec LPQ

Le LPQ est un algorithme qui se concentre sur l'analyse des phases locales des fréquences dans une image pour extraire des informations robustes aux flous à l'aide de la transformée de Fourier à court terme (STFT). La figure 3.5 illustre les étapes de fonctionnement de l'opérateur LPQ sur les images.

Lorsqu'on applique le descripteur LPQ (Local Phase Quantization) avec un carré de taille 4x4 sur une image, le résultat est illustré dans la figure 3.6 :

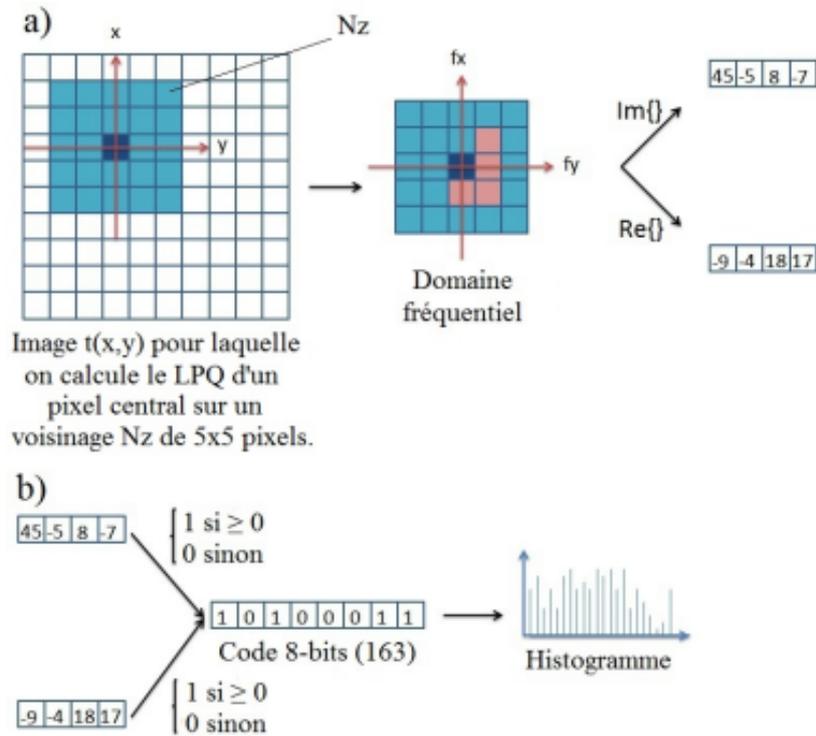


FIGURE 3.5 – Étapes de fonctionnement du descripteur LPQ [41].

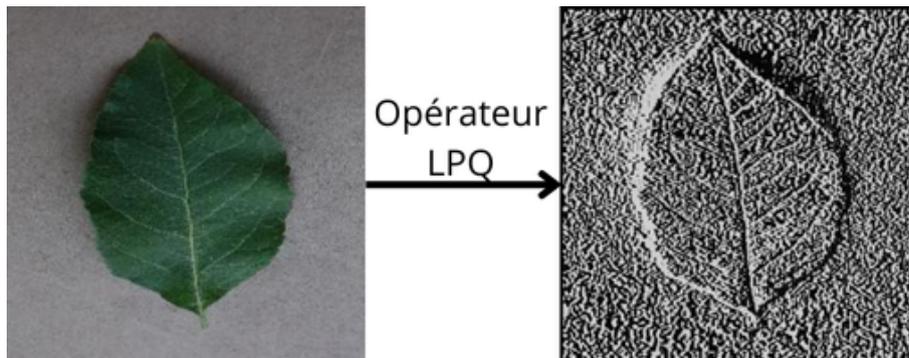


FIGURE 3.6 – Exemple de LPQ sur une image

3.2.2.2 Extraction des caractéristiques avec LBP

Le LBP [45] est une méthode simple et efficace pour capturer les motifs de texture locale dans les images.

Le concept de calcul de LBP de base est simple et peut se résumer comme suit :

1. considérer un voisinage dans une image autour d'un pixel centrale, typiquement on utilise un voisinage de 3×3 pixels.
2. Comparer les pixels de voisinage avec le pixel centrale, si l'intensité du pixel voisin est supérieure ou égale à celle du pixel centrale on attribue la valeur 1, sinon on attribue la valeur 0.

3. Les valeurs binaires obtenues, sont ensuite organisée en un motif autour du pixel central, avec des poids de puissance 2 attribuer à chaque position.
4. Les valeurs binaires obtenue dans l'étape 2 sont multipliée par les poids binaires correspondants.
5. Faire la somme des valeurs pondérées, afin d'obtenir un code LBP du pixel courant.
6. Répéter les étapes de 1 à 5 jusqu'à ce que tous les pixels de l'image soient traités.

On obtient au final une image ou l'intensité de chaque pixel compris entre 0 et 255.

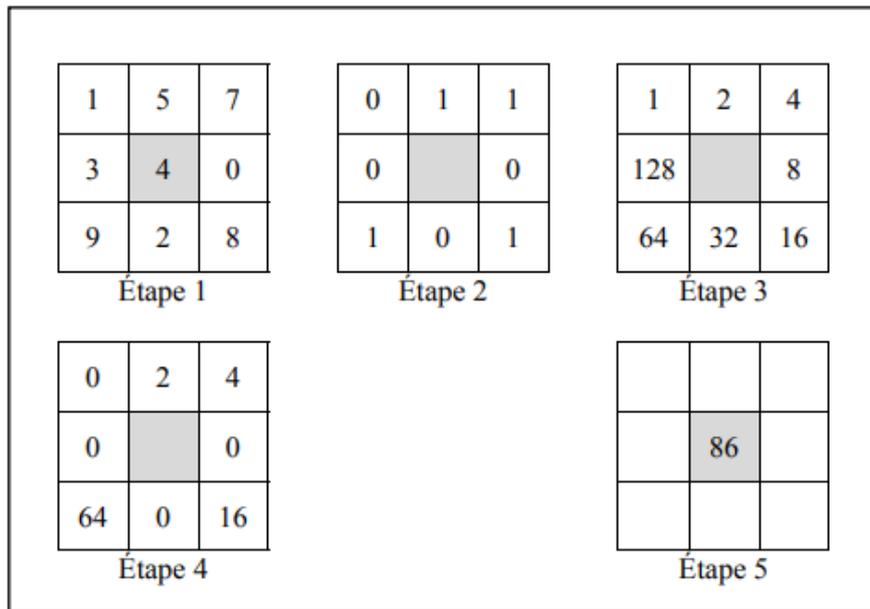


FIGURE 3.7 – Illustration de calcul de LBP de base [33].

Voici un exemple d'application de LBP sur une image :

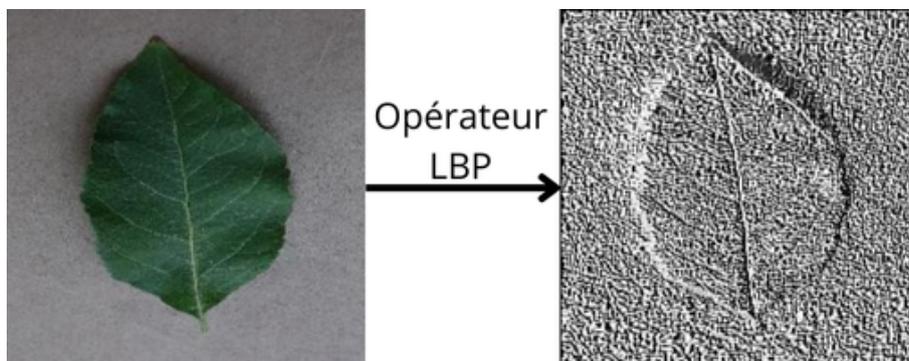


FIGURE 3.8 – Exemple de LBP sur une image

3.2.2.3 Extraction des caractéristiques avec VGG16

Dans ce travail, on s'intéresse à l'utilisation de modèles (VGG16). Il est important de comprendre les caractéristiques de cette architecture, telle que la taille d'entrée, la profondeur et la robustesse. Le modèle VGG-16 est une architecture de réseau de neurones convolutifs (CNN) développée par le Visual Geometry Group (VGG) de l'Université d'Oxford. Il se distingue par sa profondeur, composée de 16 couches, dont 13 couches convolutives et 3 couches entièrement connectées. Présenté en 2014 par Karen Simonyan et Andrew Zisserman, le VGG-16 a été conçu pour la compétition annuelle ImageNet Large Scale Visual Recognition Challenge (ILSVRC), où les équipes s'affrontent pour des tâches telles que la localisation d'objets et la classification d'images. Ce modèle a surpassé les autres en atteignant une précision de test de 92,7 % dans le top 5 sur le jeu de données ImageNet, qui comprend 14 millions d'images réparties en 1000 classes différentes [79].

l'architecture

Il existe en réalité deux algorithmes : VGG16 et VGG19. Dans ce mémoire, nous allons nous intéresser à l'architecture du premier illustrée par la figure 3.9. Si les deux architectures sont très proches et respectent la même logique, VGG19 a un nombre plus élevé de couches de convolution.

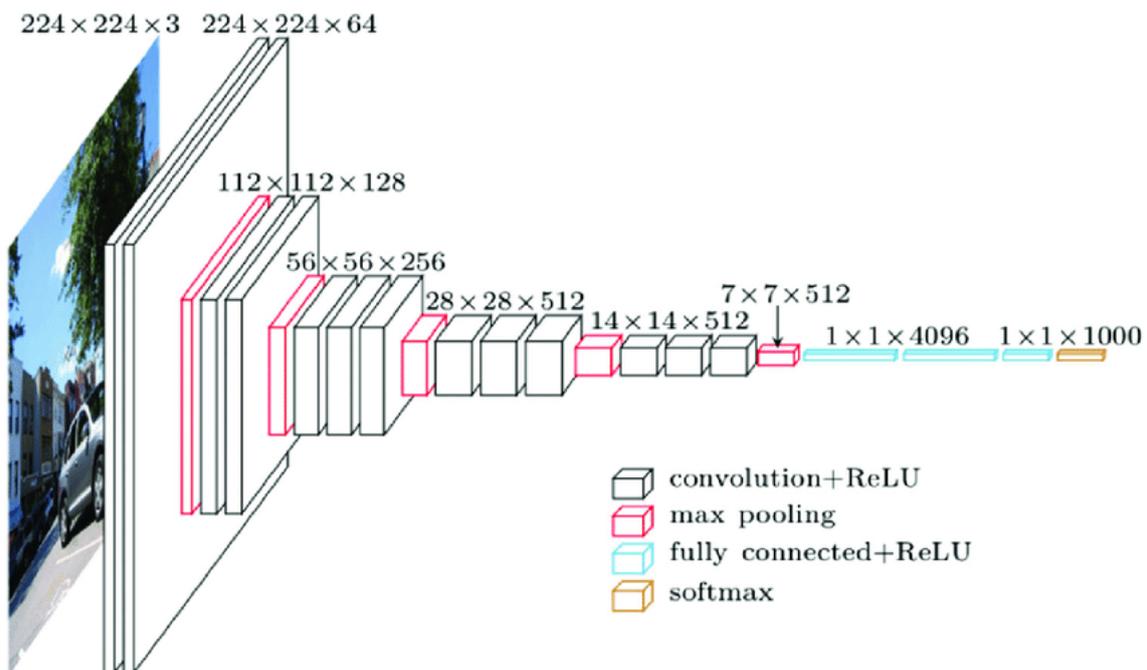


FIGURE 3.9 – Architecture du modèle VGG16 [49].

L'architecture VGG16 est une référence majeure dans le domaine de la vision par ordinateur.

Elle est composée d'une série de couches qui traitent les images à différentes étapes qui se peut illustré comme suit :

1. Tout d'abord, une image RGB de taille 224x224 est introduite.
2. Ensuite, dans les deux premières couches, des opérations de convolution avec 64 filtres de taille 3x3 extraient les caractéristiques initiales de l'image. Ensuite, une couche de max pooling est utilisée pour réduire de moitié la taille de l'image tout en préservant les caractéristiques importantes, avec une fenêtre de taille 2x2 et un stride de 2 fixé pour toutes les les opérations de pooling qui suivent.
3. Les deux ensembles suivants de convolutions utilisent respectivement 128 et 256 filtres, suivis d'un autre max pooling. Il y a également une couche de convolution supplémentaire, à travers laquelle les images sont encore réduites à l'aide de 512 filtres, suivie à nouveau d'une opération de max pooling.
4. Enfin, le dernier ensemble de trois couches de convolution utilise à nouveau 512 filtres, suivis d'une autre couche de max pooling.
5. Après cette étape, nous obtenons (7,7,512) cartes de caractéristiques, qui sont aplaties en un vecteur de caractéristiques de (1,25088), et seront ensuite alimentées dans deux couches entièrement connectées : une première couche avec 4096 neurones, puis une autre couche de dropout pour éviter le sur-ajustement.
6. Enfin, la couche de sortie est entièrement connectée avec 1000 neurones, correspondant aux catégories disponibles dans l'ensemble de données ImageNet.

Grâce à cette architecture, le modèle VGG16 a réussi à obtenir des performances exceptionnelles sur de nombreux types différents de tâches de classification d'images. Il est également couramment utilisé comme modèle de base pour le transfert d'apprentissage dans de nombreuses applications de vision par ordinateur, mais cela s'accompagne généralement d'un coût computationnel élevé car la complexité de l'architecture nécessite une puissance de calcul substantielle [79].

Cependant, dans notre cas, nous nous sommes arrêtés à l'étape d'aplatissement. Nous utilisons uniquement l'extraction des caractéristiques fournie par VGG16 jusqu'à cette étape, sans passer par les couches entièrement connectées et la couche de sortie softmax. Cette méthode nous permet de bénéficier de la capacité de VGG16 de traiter des caractéristiques visuelles complexes, tout en intégrant ces caractéristiques dans notre propre modèle de classification pour la détection des maladies des plantes.

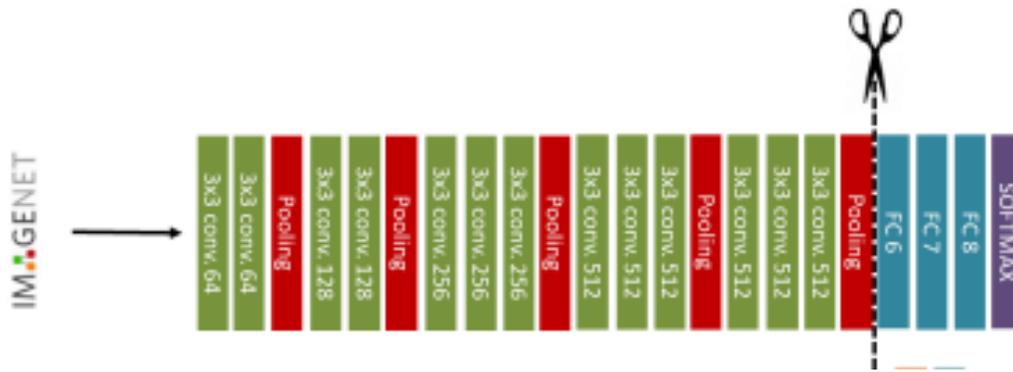


FIGURE 3.10 – Extraction des caractéristiques avec VGG16 [58].

3.2.3 Réduction de dimension

Une fois les caractéristiques sont extraite, nous avons utilisée une combinaison de techniques de réduction de dimension avec PCA (Analyse en Composantes Principales) et LDA (Analyse Discriminante Linéaire). Cette double réduction permet de diminuer la complexité des données tout en maximisant la séparabilité entre les différentes classes de maladies ainsi améliorer l'efficacité de notre système.

3.2.3.1 Réduction de dimension avec PCA

L'ACP est une technique statistique non supervisée et non paramétrique principalement utilisée pour la réduction de la dimensionnalité des grands ensembles de données tout en préservant autant d'informations que possible [46], il s'agit donc d'obtenir le résumé le plus pertinent possible des données initiales. Tous cela est réaliser par la projection des données dans un espace réduit en utilisant les vecteurs propres.

Les vecteurs propres sont appelée aussi les composantes principales (PC), sont des directions dans l'espace des variables d'origine dans lesquelles la variance des données est plus grande. Les premières composantes captures la majorité de la variance, tandis que les suivants en capturent de moins en moins.

voici les étapes de l'ACP :

- 1) **Standardisation des données** : Cela consiste à centrer (soustraire la moyenne) et à réduire (diviser par l'écart-type) les données, afin que toutes les variables aient la même échelle. L'objectif de cette étape est d'assurer que chaque variable contribue de manière égale à l'analyse, évitant ainsi que les variables ayant des plages de valeurs différentes dominant l'analyse et produisent des résultats biaisés.
- 2) **Calcul de la matrice de covariance** : cette étape consiste a calculer la matrice qui

montre comment les variables liées entre elle.

- 3) **Calcule des valeurs propres et vecteurs propres :** Les valeurs propres et les vecteurs propres de la matrice de covariance sont calculés. Les vecteurs propres représentent les directions des composantes principales, et les valeurs propres représentent l'importance de chaque composante.
- 4) **Sélection des composantes principales :** Les vecteurs propres (composantes principales) sont triés par ordre décroissant des valeurs propres. Les premières composantes principales expliquent la plus grande part de la variance totale des données.
- 5) **Projection des données :** Les données initiales sont projetées sur le nouvel espace formé par les vecteurs propres sélectionnés. Cela réduit la dimensionnalité des données tout en conservant la majorité de l'information

3.2.3.2 Réduction de dimension avec LDA

L'Analyse Discriminante Linéaire (LDA) est une technique statistique utilisée principalement pour la classification et la réduction de dimensionnalité. L'objectif de LDA est de trouver un sous-espace de dimension réduite dans lequel les différentes classes de données sont le plus séparées possible. Contrairement à l'ACP (Analyse en Composantes Principales) qui cherche à maximiser la variance totale des données, LDA se concentre sur la variance entre les classes et la variance intra-classe. Voici les étapes principales de la réduction de Dimensionnalité avec LDA :

- 1) **Standardisation des données :** La première étape consiste à standardiser les données pour que chaque caractéristique ait une moyenne de 0 et une variance de 1. Cela est crucial et aide à éviter que certaines caractéristiques ne dominent la variance simplement en raison de leur échelle numérique. car LDA est sensible aux échelles des caractéristiques.
- 2) **Calcul des Matrices de Dispersion :** Pour LDA, il faut calculer deux matrices de dispersion :
Matrice de Dispersion Intra-classe : Cette matrice mesure la dispersion des points de données au sein de chaque classe.
Matrice de dispersion inter-classe : Cette matrice mesure la dispersion des centres de classes entre eux.
- 3) **Calcul des vecteurs propres et valeurs propres :** Ensuite, les vecteurs propres et les valeurs propres de la matrice de dispersion entre-classes et de la matrice de dispersion intra-classe sont calculés.

- 4) **Sélection des composantes principale** : Les vecteurs propres sont classés par valeurs propres associées, et les k vecteurs propres les plus importants sont sélectionnés pour former une matrice de transformation.
- 5) **Projection des données** : Enfin, les données originales sont projetées sur le sous-espace défini par les vecteurs propres sélectionnés.

3.2.4 Fusion

Cette étape consiste à concaténer les caractéristiques réduites de LPQ, LBP et VGG16 pour chaque image après application de PCA et LDA, obtenant ainsi un vecteur unique de caractéristiques (chaque image est représentée par un vecteur de caractéristiques combiné), prêt pour la phase de classification.

Cette fusion permet de créer un vecteur de caractéristiques riche et diversifié pour chaque image, prêt à être utilisé dans l'étape de classification suivante. Le vecteur fusionné contient des informations détaillées sur les textures, les structures et les caractéristiques de haut niveau des images, augmentant ainsi la précision et la robustesse de la classification des maladies des plantes.

3.2.5 Comparaison et prise de décision :

Dans le contexte de la détection et de la classification des maladies des plantes, la comparaison et la prise de décision sont des étapes essentielles après l'extraction des caractéristiques des images de feuilles. Voici comment ces étapes se déroulent dans notre approche :

Comparaison

La comparaison consiste à évaluer la similarité entre les caractéristiques extraites d'une image de feuille et celles stockées dans une base de données de références. Pour chaque image de feuille, les caractéristiques extraites, représentées sous forme de vecteurs, sont comparées à celles présentes dans la base de données. Cette comparaison peut se faire en utilisant des mesures de similarité comme la distance euclidienne, qui permet de quantifier la différence entre les vecteurs de caractéristiques.

Prise de décision

Après la comparaison, la prise de décision permet de déterminer si la feuille analysée appartient à une plante saine ou malade, et, en cas de maladie, de quelle maladie spécifique

il s'agit. Un seuil de décision est défini pour établir la similitude minimale nécessaire pour qu'une correspondance soit considérée comme valide. En fonction des résultats obtenus lors de la comparaison et du seuil établi, le système prend une décision finale pour classer l'image de la feuille.

Pour notre système, nous avons choisi d'utiliser la distance euclidienne pour les étapes de comparaison et de prise de décision, permettant ainsi de distinguer efficacement entre les plantes saines et malades et d'identifier les types spécifiques de maladies.

3.2.5.1 Distance euclidienne

La distance euclidienne est une mesure de la distance entre deux points dans un espace euclidien. Elle est définie comme la longueur d'un segment de droite entre ces deux points. Cette distance peut être calculée à partir des coordonnées cartésiennes des points en utilisant le théorème de Pythagore [10]. Elle est utilisée pour comparer les caractéristiques extraites de deux feuilles en calculant la distance entre les deux vecteurs de caractéristiques. Plus la distance est petite, plus les feuilles sont similaires selon la formule suivante :

$$\text{Distance Euclidienne} = \sqrt{\sum_{i=1}^n (A_i - B_i)^2} \quad (3.1)$$

3.3 Conclusion

Dans ce chapitre, nous avons présenté une approche complète pour la détection et la classification des maladies des plantes à partir d'images de feuilles. Notre méthode intègre plusieurs étapes clés, allant de la collection des données à la prise de décision finale, en passant par des processus avancés de pré-traitement et d'extraction de caractéristiques.

Méthodologie et résultats

4.1 Introduction

Dans ce chapitre, nous allons examiner les performances de notre système de détection et classification des maladies des plantes afin de vérifier la validité de notre approche proposée mentionnée dans le chapitre précédent. Nous commencerons par décrire les matériels et outils de développement utilisés, ainsi que la base de donnée employé pour nos expériences. Ensuite, nous présenterons les résultats de notre protocole expérimental en mettant l'accent sur les performances de système selon divers métriques d'évaluation.

4.2 Matériels, Environnement et outils de développement

Dans cette section, nous donnons un bref aperçu du langage de programmation et des outils logiciels utilisés dans notre travail, à savoir Python, Keras, TensorFlow ainsi que du matériel utilisé.

4.2.1 Matériel utilisé

La configuration matérielle utilisée dans l'implémentation était un PC portable Acer DESKTOP-74T3LPR équipé d'un processeur Intel Core i3-5005U à 2.00 GHz et 4 Go de RAM, sous le système d'exploitation Windows 10 64 bits.

4.2.2 Environnements et outils de développement

Divers environnements et langages de programmation sont disponibles dans la littérature pour le développement de modèles de deep learning. Pour la mise en œuvre de notre système,

nous avons utilisé les outils suivants :

1. **Google Colaboratory** : Colab est un service de notebook Jupyter hébergé qui ne nécessite aucune configuration et offre un accès gratuit aux ressources informatiques telles que les GPU et les TPU. Colab est particulièrement adapté à l'apprentissage automatique, à la science des données et à l'éducation.



FIGURE 4.1 – Logo Google Colab [8].

2. **Spyder** : Est un environnement de développement pour python. Libre et multiplateforme, il intègre un certain nombre de bibliothèques pour des applications scientifiques (Matplotlib, NumPy, SciPy, IPython).



FIGURE 4.2 – Logo Spyder [7].

3. **Python** : Python est un langage de programmation interprété, de haut niveau et orienté objet avec une sémantique dynamique. La syntaxe de Python est simple et facile à apprendre, et l'accent mis sur la lisibilité réduit les coûts de maintenance du programme. Python prend en charge modules et packages, facilitant la modularisation du programme et la réutilisation du code. Python est un langage de programmation élégant, polyvalent et extrêmement populaire.



FIGURE 4.3 – Logo Python [21].

4. **OpenCV** : est une bibliothèque gratuite développée indépendamment par Intel et spécialisée dans le traitement d'images en temps réel. La bibliothèque OpenCV fournit un grand nombre de fonctionnalités très diverses qui permettent de créer des programmes à partir de données brutes jusqu'à la création d'interfaces graphiques de base. Fournit la plupart des opérations traditionnelles de traitement d'image et de vidéo.



FIGURE 4.4 – Logo OpenCV [14].

5. **Numpy** : NumPy est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux masqués et des matrices), et un assortiment de routines pour des opérations rapides sur les baies, y compris mathématiques, logiques, manipulation de formes, tri, sélection, E/S, transformées de Fourier discrètes, algèbre linéaire de base, statistiques de base opérations, simulation aléatoire et bien plus encore. Au cœur du package NumPy, se trouve l'objet ndarray. Ceci encapsule des tableaux à n dimensions de types de données homogènes, avec de nombreuses opérations sont effectuées dans le code compilé pour des raisons de performance [13].



FIGURE 4.5 – Logo NumPy [13].

6. **Matplotlib :**

Matplotlib a été créé par John D. Hunter, est une bibliothèque de traçage de graphes de bas niveau en python qui sert d'utilitaire de visualisation. Matplotlib est open source et nous pouvons l'utiliser librement [15].



FIGURE 4.6 – Logo Matplotlib [16].

7. **TensorFlow :** TensorFlow est une plate-forme open source d'apprentissage automatique créé à l'origine par des chercheurs de Google. Il s'agit d'une bibliothèque de mathématiques symboliques qui utilise le flux de données et la programmation différentiable pour effectuer diverses tâches axées sur la formation et l'inférence de réseaux neuronaux profonds. Elle permet aux développeurs de créer des applications d'apprentissage automatique en utilisant divers outils, bibliothèques et ressources communautaires [2].



FIGURE 4.7 – Logo TensorFlow [22].

8. **Keras :** Keras est une API de réseau de neurones écrite en langage Python. Il s'agit d'une bibliothèque Open Source, capable de s'exécuter sur TensorFlow. Elle permet de créer très facilement des couches pour les réseaux de neurones ou de mettre en place des architectures complexes [9].

9. **Scikit-Learn :** Scikit-learn est une bibliothèque Python gratuite pour l'apprentissage

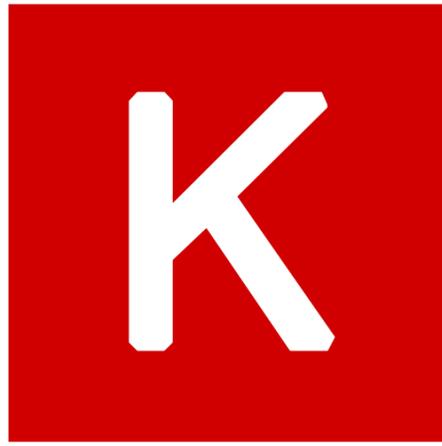


FIGURE 4.8 – Logo TensorFlow [12].

automatique. Il est développé par de nombreux contributeurs, notamment universitaires, par des instituts d'enseignement supérieur et de recherche français comme Inria [20].



FIGURE 4.9 – Scikit-learn-logo [20].

4.3 Base de donnée

Pour créer des modèles de classification d'images précis pour le diagnostic des maladies des plantes, nous avons utilisé des images de pommes saines et malades provenant du projet "PlantVillage". Ce projet a rassemblé des milliers d'images de plantes cultivées, disponibles gratuitement, couvrant 37 variétés de plantes.

Nous avons sélectionné les images de pommes et les avons réparties en quatre catégories : une pour les pommes saines et trois pour les pommes malades, avec 1500 images par catégorie. Les feuilles ont été isolées et placées sur des arrière-plans unis afin de mettre en évidence clairement les symptômes.

Les photos ont été prises avec une caméra terrestre RGB puis redimensionnées à 256×256 pixels. Cette étape de redimensionnement vise à diminuer le volume des données et à améliorer la réactivité de l'algorithme.

Les 3 maladies traitées dans ce travail sont :

- **La rouille du pommier de cèdre (Apple Cedar apple rust)** : La maladie de la rouille du pommier de cèdre est fongique, affecte surtout les pommiers et les cèdres. Ce champignon démarre son cycle sur les cèdres en créant des galles qui libèrent des spores au printemps, infectant ensuite les feuilles, fruits et branches des pommiers. Les signes incluent des taches jaunes ou oranges sur les feuilles des pommiers, des déformations des fruits et, en cas d'infection sévère, une chute prématurée des feuilles. Sur les cèdres, on remarque des galles brun-orange sur leurs branches. Pour contrôler cette maladie, il est conseillé de choisir des variétés de pommiers résistantes, d'éviter la proximité entre ces arbres et d'utiliser des fongicides adaptés au printemps [4].
- **Pourriture noire du pommier (Apple Black rot)** : La pourriture noire est une maladie répandue des pommiers, causée par le champignon *Botryosphaeria obtusa*. Celui-ci infecte un large éventail de feuillus, dont le pommier et le poirier. Les arbres infectés deviennent à leur tour une source d'infection pour les jeunes blocs fructifères. Le champignon noir provoque une diminution de la productivité et de la qualité des cultures en raison de son infection des branches, des tiges, des feuilles et des fruits. Les symptômes comprennent généralement des taches ocellées, qui se développent progressivement et forment des lésions de 4-5 mm de diamètre. La maladie apparaît souvent entre une et trois semaines suivant le calice, avec des symptômes ressemblant à une ocelle (tache évoquant un œil) [3].
- **Tavelure du pommier (Apple scab)** : Les taches foliaires causées par la tavelure du pommier sont rondes, de couleur vert olive, et peuvent atteindre jusqu'à $\frac{1}{2}$ pouce de diamètre. Ces taches ont une apparence veloutée avec des bordures frangées. En vieillissant, elles virent du brun foncé au noir, s'agrandissent et se regroupent souvent. Les taches apparaissent fréquemment le long des nervures des feuilles. Les feuilles sévèrement atteintes jaunissent et tombent prématurément, souvent dès le milieu de l'été. Les fruits infectés montrent des taches vert olive qui deviennent brunes et liégeuses avec le temps. Lorsque les fruits sont infectés très jeunes, ils peuvent se déformer et se fissurer en grandissant [1].

La figure [4.10] illustre la base de donnée utilisée.

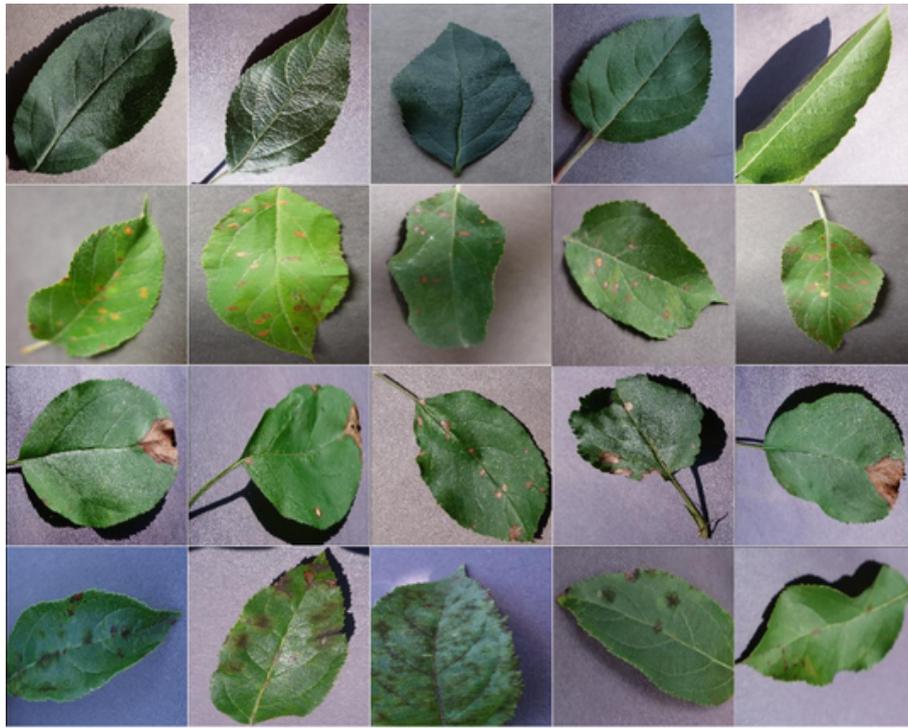


FIGURE 4.10 – Exemple de notre base de données

4.4 Métriques d'évaluation

Les mesures d'évaluation sont utilisées pour mesurer la qualité du modèle. L'un des sujets les plus importants de l'apprentissage profond est l'évaluation du modèle. Lors de la construction d'un modèle, il est très important de mesurer sa précision dans la prédiction du résultat attendu. L'incapacité d'évaluer correctement un modèle à l'aide de mesures d'évaluation appropriées peut entraîner des prédictions incorrectes, en particulier lorsque l'ensemble de données est déséquilibré. Par conséquent, nous proposons d'utiliser diverses mesures d'évaluation répertoriées ci-dessous :

4.4.1 Matrice de confusion(Confusion Matrix)

Une matrice de confusion ou une matrice d'erreur est un tableau qui montre le nombre de prédictions correctes et incorrectes faites par un modèle par rapport à la classification réelle de l'ensemble de données, et donc le nombre de prédictions correctes et incorrectes faites par le modèle concernant la confusion. (erreur) faite par le modèle Vous pouvez obtenir des informations. modèle. Cette matrice décrit les performances du modèle de classification sur des données de test dont les vraies valeurs sont connues. Il s'agit d'une matrice $n \times n$, où n est le nombre de classes, . La figure 3.3 montre la matrice de confusion pour un problème avec 2 classes. Comme vous pouvez le constater, les résultats de la matrice de confusion se répartissent en

quatre catégories principales : les vrais positifs (VP), les vrais négatifs (VN), les faux positifs (FP) et les faux négatifs (FN).

Confusion matrix		Reality	
		Negative : 0	Positive : 1
Prediction	Negative : 0	True Negative : TN	False Negative : FN
	Positive : 1	False Positive : FP	True Positive : TP

FIGURE 4.11 – Matrice de confusion [17]

- **Vrai positif (True Positive)** : Nombre d'échantillons qui sont réellement positifs et qui sont prédit positifs.
- **Vrai négatif (True Negative)** : Nombre d'échantillons qui sont réellement négatifs et qui sont prédit négatifs.
- **Faux positif (False Positive)** : Nombre d'échantillons qui sont en réalité négatifs mais prédits positifs.
- **Faux négatif (False Negative)** : Nombre d'échantillons qui sont en fait positifs mais prédits négatifs.

4.4.2 Précision de la classification (Classification Accuracy)

L'Accuracy est un critère d'évaluation couramment utilisé pour mesurer la performance des modèles de classification. Elle indique la proportion de prédictions correctes réalisées par le modèle par rapport au nombre total d'échantillons. La formule de calcul est le nombre de prédictions correctes divisé par le nombre total d'échantillons :

$$\text{Accuracy \%} = \left(\frac{TP + TN}{TP + TN + FP + FN} \right) \times 100 \quad (4.1)$$

4.4.3 Précision

La précision mesure le pourcentage de valeurs positives prédites correctement par rapport au nombre total de valeurs positives. L'équation peut être énoncée comme suit :

$$\text{Précision \%} = \left(\frac{TP}{TP + FP} \right) \times 100 \quad (4.2)$$

4.4.4 Recall ou Sensibilité ou Rappel

Le rappel (the recall), également appelé sensibilité ou taux de vrais positifs (*True Positive*), indique correctement la proportion de vrais positifs classés identifiés par le modèle (e.g., le pourcentage de patients correctement identifiés comme atteints par la maladie). C'est une mesure très pertinente lorsque les données sont déséquilibrées. Elle est calculée en utilisant l'équation suivant :

$$\text{Rappel \%} = \left(\frac{TP}{TP + FN} \right) \times 100 \quad (4.3)$$

4.4.5 F1-score

Il s'agit de la moyenne harmonique de la précision et du rappel d'un échantillon, le score F1 peut être défini comme suit :

$$\text{F1 - score \%} = \left(2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \right) \times 100 \quad (4.4)$$

4.4.6 ROC

Une courbe ROC (Receiver Operating Characteristic) est un graphique qui représente les performances d'un modèle de classification binaire en traçant le taux de vrais positifs (sensibilité) en fonction du taux de faux positifs (1- spécificité) pour tous les seuils de classification.

4.5 Protocole expérimental

Le développement de systèmes de diagnostic des maladies des plantes nécessite des protocoles expérimentaux très précis pour obtenir des résultats satisfaisants.

Dans cette section, nous décrivons le protocole expérimental adopté en terme de méthodologie, de répartition des données.

4.5.1 Méthodologie

Notre objectif principal est de développer éventuellement un modèle capable de détecter si une plante est malade ou non, et d'identifier la maladie spécifique parmi 3 classes de maladie.

Encore une fois, pour ce travail, nous avons pour plante de pomme. Nous cherchons à parvenir au diagnostic d'une feuille de la plante en utilisant à la fois une classification binaire

et une classification multi-classes pour identifier les maladies associées. ainsi avec cette méthode nous avons 2 modèles de classification.

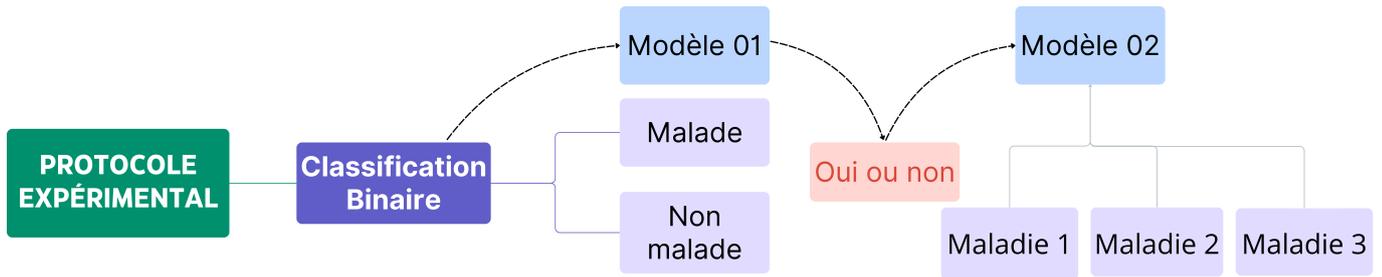


FIGURE 4.12 – Méthode adopté pour la détection et classification des maladies des plantes

4.5.2 Répartition des données

Nous avons utilisée deux configuration déférentes à partir de la même source de donnée PlantVillage qui se compose d’une classe saine et 3 classes de maladies. Notons que 80% des échantillons sont utilisés pour la phase d’entraînement du modèle, et 20% pour le test, dans les 2 configurations.

1. Pour le modèle de détection de la maladie, nous avons regroupée les classes malades en une seule classe. Ainsi, la base de donnée utilisée pour ce modèle contient un sous-dossier pour les feuilles malades et un autre pour les feuilles saines, comme illustrée dans le tableaux [4.1](#). Pour compenser le déséquilibre des classes, nous avons effectué une augmentation des données pour la classe saine afin d’assurer une meilleure performance du modèle.

	Entraînement	Test
Apple sick	3600	900
Apple healthy	3598	900
Totale	7198	1800

TABLE 4.1 – Description de la Base de données d’apprentissage 01

2. Pour le modèle d’identification des maladies spécifique, nous avons utilisé uniquement les classes malades de la base de données.

	Entrainement	Test
Apple Cedar apple rust	1200	300
Apple Black rot	1200	300
Apple scab	1200	300
Totale	3600	900

TABLE 4.2 – Description de la Base de données d'apprentissage 02

4.6 Performances atteintes

Dans un premier temps, nous allons évaluer l'efficacité de notre système en utilisant quatre méthodes qui sont : LPQ, LBP, VGG16 ainsi que la concaténation des trois méthodes.

4.6.1 Modèle 01

Rappelons, que dans ce premier modèle de classification nous allons détecter juste si une feuille de plante est malade ou bien non. L'évaluation des performances des modèles est principalement basée sur la précision (accuracy), le recall et le F1-score dans les différents ensemble de test.

Méthode d'extraction des caractéristiques	Précision(%)
LPQ	66%
LBP	84.44%
VGG16	99.56%
Fusion(LPQ+LBP+VGG16)	99.67%

TABLE 4.3 – Résultats de la classification binaire sur les données de test

Les résultats obtenus de la classification binaire sur les données de test montrent des performances variées en fonction de choix de la méthode d'extraction des caractéristiques utilisée. En remarque que la méthode basée sur le LBP donne des meilleurs résultats par rapport à celle de LPQ avec une précision de 84.44%, ce qui signifie que elle capture mieux les détails nécessaires pour distinguer la plante malade de la plante saine. D'autre part, la méthode VGG16 qui utilise un modèle de deep learning, a eu des résultats plus élevés par rapport au deux cités auparavant avec une précision de 99.56%, cela montre que les techniques avancées d'apprentissage automatique sont très efficaces pour analyser des images complexes des plantes. Enfin,

la concaténation des caractéristiques provenant des 3 méthodes précédentes nous a fournis le meilleurs résultat avec une précision de 99.67%, indiquant que combiner plusieurs techniques offre les meilleurs résultats

4.6.2 Modèle 02

Pour la deuxième partie de cette expérience qui concerne la classification des maladies par type de maladie, les résultats obtenus sont illustrés dans le tableau 4.4 :

Méthode d'extraction des caractéristiques	Précision(%)
LPQ	53.00%
LBP	69.44%
VGG16	99.44%
Fusion(LPQ+LBP+VGG16)	99.56%

TABLE 4.4 – Résultats de la classification par type de maladie sur les données de test

Nous remarquons, d'après les résultats que le LBP est encore performant que le LPQ avec une précision de 69,44%. d'autre par la méthode de VGG16 a eu des résultats plus élevée par apport à celles cité auparavant avec une précision 99.44%. Enfin la concaténation des trois méthodes nous a fournis les meilleures résultats en atteignant une précision de 99.56%, prouvant que combiner plusieurs méthodes d'extraction des caractéristiques peut améliorer la robustesse et la précision de la classification.

Afin de mieux apprécier les performances de nos modèles, nous présentons les métriques précision, recall et F-1 score par classe 4.5 et 4.6

LPQ			
	Précision(%)	Recall(%)	F1-score (%)
Apple sick	66%	66%	66%
Apple healthy	66%	66%	66%
LBP			
	Précision(%)	Recall(%)	F1-score (%)
Apple sick	88%	79%	84%
Apple healthy	81%	89%	85%
VGG16			
	Précision(%)	Recall(%)	F1-score (%)
Apple sick	99%	100%	100%
Apple healthy	100%	99%	100%
Fusion			
	Précision(%)	Recall(%)	F1-score (%)
Apple sick	100%	100%	100%
Apple healthy	100%	100%	100%

TABLE 4.5 – Précision, recall et F1-score par classe du modèle 01

LPQ			
	Précision(%)	Recall(%)	F1-score (%)
Apple Cedar apple rust	61%	74%	67%
Apple Black rot	38%	30%	33%
Apple scab	55%	55%	55%
LBP			
	Précision(%)	Recall(%)	F1-score (%)
Apple Cedar apple rust	70%	65%	67%
Apple Black rot	59%	62%	60%
Apple scab	80%	82%	81%
VGG16			
	Précision(%)	Recall(%)	F1-score (%)
Apple Cedar apple rust	100%	100%	100%
Apple Black rot	99%	99%	99%
Apple scab	99%	100%	99%
Fusion			
	Précision(%)	Recall(%)	F1-score (%)
Apple Cedar apple rust	100%	100%	100%
Apple Black rot	99%	99	99%
Apple scab	100%	99%	99%

TABLE 4.6 – Précision, recall et F1-score par classe du modèle 02

4.7 Comparaison avec l'état de l'art

Le tableau [4.7](#) représente une comparaison avec l'état de l'art en terme de l'accuracy.

Auteurs	Type de plante	Nombre de classes	Nombre d'images	Techniques	Précision
Srivastava et al [82]	Mendeley, Cerise	22 , 9	4590 et 2052	VGG16, InceptionV3, MobileNetV2, Xception, DenseNet121	98,10%, 98,9%, 98.40%, 96.90%, 99,9%
Kaya et al [52]	14 plantes distinctes	38	54183	DenseNet	98%
Sunil et al [44]	tomate	6	tomate	SVM,CNN,K-NN	88%, 97%, 99,6%
Ajra et al [24]	pommes de terre, tomates	4	4000	AlexNet, ResNet-50.	96,5%,97%
Audre Arlene et al[30]	tomates	5	500	Random forêt, Arbre de décision	94%,98%
Ganatra et al[42]	Non spécifié	38	14956	Random Forest, SVM, K-NN, ANN	72.88%, 66.99%, 63.18%, 65.60%
Notre approche (modèle 2)	pomme	3	6000	LPQ, LBP,VGG16, PCA, LDA, ED	99.56%

TABLE 4.7 – Comparaison avec l'état de l'art en terme l'accuracy

4.8 Interprétation et discussion des résultats

Notre premier modèle détecte si une plante est saine ou malade avec une précision de 99,67% en utilisant 2 classes et 8998 images. Ce dernier utilise des techniques d'extraction de caractéristiques telles que LPQ, LBP, VGG16 et technique de réduction de dimensions telles que PCA, LDA prise de décision avec ED. Cette combinaison de technique démontre une grande efficacité pour une classification binaire, surpassant les résultats des autres études qui se concentrent directement sur une classification multiclassées.

Notre deuxième modèle a été développée pour détecter trois classes de maladies chez les pommes, utilisant un total de 6000 images. Ce modèle, basé sur les mêmes techniques que le modèle 1, atteint une précision de 99,56%. Comparé à d'autres travaux, notre modèle surpasse plusieurs études, notamment [30] avec une précision de 94% à 98% utilisant des forêts aléatoires

et arbres de décision, et [44] avec des précisions de 88% à 99,67% en utilisant SVM, CNN et K-NN pour les maladies de la tomate. L'étude de [82], utilisant des réseaux de neurones profonds pour 22 et 9 classes, obtient des précisions de 98,10% à 99,9%. Enfin, [52] utilisant DenseNet pour 38 classes atteint une précision de 88%. En conclusion, notre approche combinant les techniques classiques et profondes s'avère très compétitive et efficace pour la classification des maladies des plantes.

4.9 Conclusion

Dans ce chapitre, nous avons présenté les tests établis pour évaluer le système de détection et de classification des maladies des plantes, fondé sur les techniques avancées de vision par ordinateur et de Deep Learning. Notre travail a porté sur l'étude de trois méthodes d'extraction des caractéristiques qui ont montré leur prouesse dans le domaine de la vision par ordinateur, à savoir LPQ, LBP, et VGG16. D'après les expériences et tests menés, nous pouvons dire que la concaténation des trois méthodes précédentes donne des résultats excellents, ce qui indique le succès de notre approche.

Conclusion générale et perspectives

La détection et la classification des maladies des plantes représentent un domaine de recherche crucial et dynamique dans l'agriculture moderne, avec des implications directes pour la productivité agricole, la durabilité environnementale et la sécurité alimentaire mondiale. Ce mémoire a exploré en profondeur les divers aspects de ce domaine, en mettant l'accent sur les méthodes et les technologies avancées utilisées pour la détection automatique des maladies des plantes.

Nous avons structuré notre étude autour de plusieurs étapes clés, à commencer par la compréhension des types de maladies des plantes et des méthodes traditionnelles de détection. La transition vers des systèmes automatisés a été ensuite examinée, en soulignant l'importance de l'acquisition d'images de haute qualité, le pré-traitement des images pour éliminer le bruit, la segmentation précise pour isoler les régions d'intérêt, l'extraction des caractéristiques pertinentes avec trois méthodes distinctes LPQ, LPB et VGG16, la réduction des dimension avec PCA suivie de LDA et enfin, la classification finale et la prise de décision à l'aide de la distance euclidienne.

Ce travail nous a permis d'acquérir de nouvelles connaissances et d'approfondir d'autres, que ce soit dans l'aspect théorique : les maladies des plantes, techniques et algorithmes utilisés, Deep Learning ou l'aspect pratique : Python, les diverses bibliothèques dédiées au traitement d'images et la vision artificielle, Google Colab...etc. Les résultats obtenus ont validé notre approche tout en ouvrant des perspectives intéressantes pour de futures recherches et améliorations.

Parmi les perspectives d'amélioration de notre système, nous citons :

- Inclusion plus de classes de maladies pour une détection plus précise.
- Inclusion d'autres types de cultures proposée par plante village au d'autres bases de données publiques.
- La validation des résultats sur d'autres bases de données pour garantir la généralisation

du modèle.

— Déploiement de système dans une application mobile.

Bibliographie

- [1] Apple scab of apples and crabapples. <https://extension.umn.edu/plant-diseases/apple-scab#:~:text=Apple%20scab%20is%20the%20most,olive%20green%20to%20brown%20spots./>. Consulté le 12/06/2024.
- [2] Architecture of tensorflow. <https://www.geeksforgeeks.org/architecture-of-tensorflow/>. Consulter le 12/06/2024.
- [3] Black rot. <https://www.ontario.ca/page/black-rot/>. Consulté le 12/06/2024.
- [4] Cedar apple rust (gymnosporangium juniperi-virginianae schwein). https://www.fs.usda.gov/wildflowers/plant-of-the-week/gymnosporangium_juniperi-virginianae.shtml. Consulté le 12/06/2024.
- [5] Convolutional neural network (cnn) architecture explained in plain english using simple diagrams. <https://towardsdatascience.com/convolutional-neural-network-cnn-architecture-explained-in-plain-english-using-simple-diagrams/>. Consulté le 11/05/2024.
- [6] Explication simple de la forêt aléatoire. <https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d>. Consulté le 29/06/2024.
- [7] File :spyder logo.svg. https://commons.wikimedia.org/wiki/File:Spyder_logo.svg. Consulter le 14/06/2024.
- [8] Google colab logo. <https://logowik.com/google-colaboratory-logo-vector-svg-pdf-ai-eps-clipart.html>. Consulter le 14/06/2024.
- [9] Keras : tout savoir sur l'api de deep learning. <https://datascientest.com/keras/>. Consulter le 14/06/2024.
- [10] La distance euclidienne. https://datafranca.org/wiki/Distance_euclidienne/. Consulter le 12/06/2024.

- [11] Les réseaux de neurones artificiels un outil de traitement de données prometteur pour l'anthropologie. <https://journals.openedition.org/bmsap/4463>. Consulté le 28/06/2024.
- [12] Logo de keras. https://commons.wikimedia.org/wiki/File:Keras_logo.svg. Consulter le 14/06/2024.
- [13] Logo numpy. <https://numpy.org/doc/stable/user/whatisnumpy.html/>. Consulté le 14/06/2024.
- [14] Logon opencv. <https://logodix.com/logo/1989934.png>. Consulter le 14/06/2024.
- [15] Matplotlib. https://www.w3schools.com/python/matplotlib_intro.asp/. Consulté le 14/06/2024.
- [16] Matplotlib 3.9.0 documentation. <https://matplotlib.org/stable/>. Consulter le 12/06/2024.
- [17] Matrice de confusion, la comprendre et l'utiliser. <https://kobia.fr/classification-metrics-matrice-de-confusion/>. Consulter le 14/06/2024.
- [18] Principe de stride. <https://www.analyticsvidhya.com/blog/2022/03/basics-of-cnn-in-deep-learning/>. Consulter le 29/06/2024.
- [19] Réseau de neurones convolutif. <https://dataanalyticspost.com/Lexique/reseau-de-neurones-convolutifs/>. Consulté le 11/05/2024.
- [20] Scikit-learn. <https://fr.wikipedia.org/wiki/Scikit-learn/>. Consulté le 14/06/2024.
- [21] Simplifying python context management with exitstack. <https://medium.com/@julianofischer/simplifying-python-context-management-with-exitstack-c604ae008278/>. Consulter le 14/06/2024.
- [22] Tensorflow. https://www.tensorflow.org/extras/tensorflow_brand_guidelines.pdf. Consulter le 12/06/2024.
- [23] Nisar Ahmad, Hafiz Muhammad Shahzad Asif, Gulshan Saleem, Muhammad Usman Yunus, Sadia Anwar, and Muhammad Rizwan Anjum. Leaf image-based plant disease identification using color and texture features. *Wireless Personal Communications*, 121 :1139 – 1168, 2021.
- [24] Husnul Ajra, Mst Khairun Nahar, Lipika Sarkar, and Md Shohidul Islam. Disease detection of plant leaf using image processing and cnn with preventive measures. In *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, pages 1–6. IEEE, 2020.

- [25] Waleed Albattah, Marriam Nawaz, Ali Javed, Momina Masood, and Saleh Albahli. A novel deep learning method for detection and classification of plant diseases. *Complex & Intelligent Systems*, pages 1–18, 2022.
- [26] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.
- [27] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning : concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8 :1–74, 2021.
- [28] Ayachi Amor Safa Labbi Manel Amirat Zahia, Atik Ben Amor Khouloud. *Développement d'un système automatisé de détection des lésions de la paume (Oryctes Agamemnon Burmeister) par drone*. PhD thesis, UniversitéEchahid Hamma Lakhderd'El-Oued FacultéTechnologie, 2023.
- [29] Dave Anderson and George McNeill. Artificial neural networks technology. *Kaman Sciences Corporation*, 258(6) :1–83, 1992.
- [30] Jagadeesh Basavaiah and Audre Arlene Anthony. Tomato leaf disease classification using multiple feature extraction techniques. *Wireless Personal Communications*, 115(1) :633–651, 2020.
- [31] Mohamed Benchohra and Sid Ali Mahmoudi. *Conception et mise en œuvre d'un système de contrôle d'accès pour les bâtiments intelligents basé sur la reconnaissance faciale et vocale conjointe*. PhD thesis, 10 2020.
- [32] Simon Bernard, Laurent Heutte, and Sebastien Adam. On the selection of decision trees in random forests. In *2009 International joint conference on neural networks*, pages 302–307. IEEE, 2009.
- [33] Jonathan Bouchard. *Méthodes de vision et d'intelligence artificielles pour la reconnaissance de spécimens coralliens*. PhD thesis, École de technologie supérieure, 2011.
- [34] Razika BOUZAR. *Utilisation de l'analyse discriminante linéaire pour un système biométrique FKP*. PhD thesis, ECOLE NATIONALE POLYTECHNIQUE, 2013.
- [35] Samir Chtita, M Bouachrine, and L Tahar. Modélisation de molécules organiques hétérocycliques biologiquement actives par des méthodes qsar/qspr. *Recherche de nouveaux médicaments. Chimie théorique et/ou physique. Faculté des sciences, Université Moulay Ismaïl, Meknès*, 2017.

- [36] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20 :273–297, 1995.
- [37] Benoît De Mezzo. *Reconnaissance d'objets par la génération d'hypothèses de modèles de forme appliquée à l'extraction des feuilles de plantes dans des scènes naturelles complexes*. PhD thesis, Doctorat Informatique, Université Montpellier II, 2004.
- [38] Youcef Djeriri. Les réseaux de neurones artificiels. *UDL-SBA, Sidi Bel Abbès*, 2017.
- [39] Clément Douarre. *Spectro-imagerie et apprentissage profond : application à la détection de maladies de plantes*. PhD thesis, Université Lumière Lyon 2, 2021.
- [40] Paul David Esker, Adam H. Sparks, L. H. Campbell, Z. Guo, Matthew N. Rouse, Sharad Silwal, Siti Marponga Tolos, Brandon Van Allen, and Karen A. Garrett. Ecology and epidemiology in r : Disease forecasting and validation. 2008.
- [41] Cécile Fiche. *Repousser les limites de l'identification faciale en contexte de vidéo-surveillance*. PhD thesis, Université de Grenoble, 2012.
- [42] Nilay Ganatra and Atul Patel. A multiclass plant leaf disease detection using image processing and machine learning techniques. *International Journal on Emerging Technologies*, 11(2) :1082–1086, 2020.
- [43] Kiran R Gavhale, Ujwalla Gawande, et al. An overview of the research on plant leaves disease detection using image processing techniques. *Iosr journal of computer engineering (iosr-jce)*, 16(1) :10–16, 2014.
- [44] Sunil S Harakannanavar, Jayashri M Rudagi, Veena I Puranikmath, Ayesha Siddiqua, and R Pramodhini. Plant leaf disease detection using computer vision and machine learning algorithms. *Global Transitions Proceedings*, 3(1) :305–310, 2022.
- [45] David Harwood, Timo Ojala, Matti Pietikäinen, Shalom Kelman, and Larry Davis. Texture classification by center-symmetric auto-correlation, using kullback discrimination of distributions. *Pattern Recognition Letters*, 16(1) :1–10, 1995.
- [46] Basna Mohammed Salih Hasan and Adnan Mohsin Abdulazeez. A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*, 2(1) :20–30, 2021.
- [47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [48] Sachin B Jagtap and Mr Shailesh M Hambarde. Agricultural plant leaf disease detection and diagnosis using image processing based on morphological feature extraction. *IOSR J. VLSI Signal Process*, 4(5) :24–30, 2014.
- [49] Andreas Kamilaris and Francesc X Prenafeta-Boldú. A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science*, 156(3) :312–322, 2018.
- [50] Maameri Karim and Mazouni Amirouche. *Segmentation multi-résolution d'images texturées par approche markovienne*. PhD thesis, Université Mouloud Mammeri, 2011.
- [51] Navneet Kaur et al. Plant leaf disease detection using ensemble classification and feature extraction. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(11) :2339–2352, 2021.
- [52] Yasin Kaya and Ercan Gürsoy. A novel multi-head cnn design to identify plant diseases using the fusion of rgb images. *Ecological Informatics*, 75 :101998, 2023.
- [53] Song Ke-Chen, YAN Yun-Hui, CHEN Wen-Hui, and Xu Zhang. Research and perspective on local binary pattern. *Acta Automatica Sinica*, 39(6) :730–744, 2013.
- [54] Saiqa Khan and Meera Narvekar. Disorder detection in tomato plant using deep learning. In *Advanced Computing Technologies and Applications : Proceedings of 2nd International Conference on Advanced Computing Technologies and Applications—ICACTA 2020*, pages 187–197. Springer, 2020.
- [55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [56] Akshay Kumar and M Vani. Image based tomato leaf disease detection. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE, 2019.
- [57] Yann LeCun, Larry Jackel, Leon Bottou, A Brunot, Corinna Cortes, John Denker, Harris Drucker, Isabelle Guyon, Urs Muller, Eduard Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pages 53–60. Perth, Australia, 1995.
- [58] Alessandro Leone, Andrea Caroppo, Andrea Manni, and Pietro Siciliano. Vision-based road rage detection framework in automotive safety applications. *Sensors*, 21(9) :2942, 2021.

- [59] Mohamed Loey, Ahmed ElSawy, and Mohamed Afify. Deep learning in plant diseases detection for agricultural crops : a survey. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, 11(2) :41–58, 2020.
- [60] AMINA MADJAOUÏ and FADIA FARES. *Segmentation des images par contours actifs : application sur les Images Satellitaires à Haute Résolutions*. PhD thesis, 2014.
- [61] Federico Martinelli, Riccardo Scalenghe, Salvatore Davino, Stefano Panno, Giuseppe Scuderi, Paolo Ruisi, Paolo Villa, Daniela Stroppiana, Mirco Boschetti, Luiz R Goulart, et al. Advanced methods of plant disease detection. a review. *Agronomy for Sustainable Development*, 35 :1–25, 2015.
- [62] H Mathlouthi, K Abderrahim, M Ltaïef, and S Telmoudi. Identification des systèmes non linéaires : approche multimodèle neuronale. In *Troisième Séminaire Tunisien d’Automatique*, pages 22–24, 2002.
- [63] KECHIR Mokrane and Kerchit Mohammed. *Segmentation d’image basée sur la technique de seuillage flou*. PhD thesis, Université Mouloud Mammeri, 2010.
- [64] Larbi Nacerdine. *Segmentation d’images avec le deep learning*. PhD thesis, Université Mouloud Mammeri, 2018.
- [65] Meni Babakidi Narcisse. Développement d’un filtre gaussien pour la réduction du bruit dans les images médicales [development of a gaussian filter for noise reduction in medical images]. *International Journal of Innovation and Applied Studies*, 36(2) :594–599, 2022.
- [66] ATCHI Abdel Malek NENNOUCHE Mohamed. *Fusion de caractéristiques pour la classification des différents niveaux de démence de la maladie d’Alzheimer*. PhD thesis, ECOLE NATIONALE POLYTECHNIQUE, 2022.
- [67] Ville Ojansivu and Janne Heikkilä. Blur insensitive texture classification using local phase quantization. In *Image and Signal Processing : 3rd International Conference, ICISP 2008. Cherbourg-Octeville, France, July 1-3, 2008. Proceedings 3*, pages 236–243. Springer, 2008.
- [68] Hanane OUAMANE. Identification de reconnaissance faciale avec des expressions. Master’s thesis, 2012.
- [69] Nikhil R Pal and Sankar K Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9) :1277–1294, 1993.
- [70] Yanwei Pang, Yuan Yuan, and Xuelong Li. Effective feature extraction in high-dimensional space. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(6) :1652–1656, 2008.

- [71] Herimanitra Ranaivoson. *Classification de la sévérité des bogues par l'utilisation de métriques tirées de l'historique GIT*. PhD thesis, Université du Québec à Trois-Rivières, 2019.
- [72] Bendjenna Riadh. *Protéger l'échange d'information via un système crypto-biométrique*. PhD thesis, Université laarbi tebessi tebessa, 2021.
- [73] Meziani Roza and Bounif Katia. *Segmentation d'images couleur par seuillage d'histogrammes*. PhD thesis, Université Mouloud Mammeri, 2017.
- [74] Deshmukh Sachin et al. Dimensionality reduction and classification through pca and lda. *International journal of computer Applications*, 122(17), 2015.
- [75] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2 : Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [76] S KRITHIKHA SANJU and DR BL VELAMMAL. An automated detection and classification of plant diseases from the leaves using image processing and machine learning techniques : A state-of-the-art review. *Annals of the Romanian Society for Cell Biology*, pages 15933–15950, 2021.
- [77] Sindhuja Sankaran, Ashish Mishra, Reza Ehsani, and Cristina Davis. A review of advanced techniques for detecting plant diseases. *Computers and electronics in agriculture*, 72(1) :1–13, 2010.
- [78] Adhao Asmita Sarangdhar and VR Pawar. Machine learning regression technique for cotton leaf disease detection and controlling using iot. In *2017 international conference of electronics, communication and aerospace technology (ICECA)*, volume 2, pages 449–454. IEEE, 2017.
- [79] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
- [80] Jaskaran Singh and Harpreet Kaur. A review on : Various techniques of plant leaf disease detection. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 232–238. IEEE, 2018.
- [81] Vijai Singh and Ak K Misra. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information processing in Agriculture*, 4(1) :41–49, 2017.
- [82] Meenakshi Srivastava and Jasraj Meena. Plant leaf disease detection and classification using modified transfer learning models. *Multimedia Tools and Applications*, pages 1–31, 2023.

- [83] Elwin L Stewart. Introduction to plant diseases : Identification and management, 1991.
- [84] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [85] M Turk. Pentland. eigenfaces for recognition. *K. Cogn. Neurosci*, 4 :72–86, 1991.
- [86] Vibhor Kumar Vishnoi, Krishan Kumar, and Brajesh Kumar. Plant disease detection using computational intelligence and image processing. *Journal of Plant Diseases and Protection*, 128 :19–53, 2021.
- [87] Javaid Ahmad Wani, Sparsh Sharma, Malik Muzamil, Suhaib Ahmed, Surbhi Sharma, and Saurabh Singh. Machine learning and deep learning based computational techniques in automatic agricultural diseases detection : Methodologies, applications, and challenges. *Archives of Computational methods in Engineering*, 29(1) :641–677, 2022.
- [88] Petros Xanthopoulos, Panos M Pardalos, Theodore B Trafalis, Petros Xanthopoulos, Panos M Pardalos, and Theodore B Trafalis. Linear discriminant analysis. *Robust data mining*, pages 27–33, 2013.
- [89] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks : an overview and application in radiology. *Insights into imaging*, 9 :611–629, 2018.
- [90] Siti Zulaikha Muhammad Zaki, Mohd Asyraf Zulkifley, Marzuraikah Mohd Stofa, Nor Azwan Mohammed Kamari, and Nur Ayuni Mohamed. Classification of tomato leaf diseases using mobilenet v2. *IAES International Journal of Artificial Intelligence*, 9(2) :290, 2020.