

Université Hassan II de Casablanca  
Faculté des Sciences Ben M’Sik

Projet de Fin d’Année

**Master Data science & big data**

**Sujet** :

**Artificial intelligence for medicinal plants detection**

Réalisé par :  
 MEFTAHI Jamil

Encadré par :  
 Pr. SAEL Nawal

Année universitaire 2021/2022

# Dédicace

*A mes chers parents,*

*Pour l’amour et le support qu’ils m’ont offert et surtout à ma Mère pour leurs prières.*

*A mes sœurs et frères.*

*A mes amis qui nous ont toujours offert l’amour, le soutien, et l’aide*

*Pour affronter la peur de ne pas réussir.*

# Remerciement

# Résumé

Les plantes jouent un rôle crucial dans la préservation de la vie et le maintien de la biodiversité sur terre en fournissant de l'air et de l'eau aux êtres vivants. Les plantes médicinales, l'une des principales catégories de plantes, servent de remède à de nombreuses maladies. Les méthodes de vision par ordinateur et de traitement d'images peuvent combler le manque de taxonomistes experts et les besoins potentiels en matière d'identification et de classification des plantes médicinales. L'objectif de ce projet est d'étudier les recherches effectuées dans ce domaine et d'extraire les méthodes utilisées, de les expliquer et de les simplifier, de les comparer en termes de précision et de facilité, et enfin de faire un travail expérimental en utilisant l'apprentissage profond CNN et la technologie de traitement des images pour se faire une idée générale de toutes les étapes de l'identification des plantes médicinales.

# Abstract

Plants play a crucial role in preserving life and maintaining biodiversity on earth by providing air and water to living things. Medicinal plants, one of the major categories of plants, serve as a remedy for many diseases. Computer vision and image processing methods can address the lack of expert taxonomists and potential needs in the identification and classification of medicinal plants. The objective of this project is to study the research done in this field and extract the methods used, explain and simplify them, compare them in terms of accuracy and ease, and finally do an experimental work using CNN deep learning and image processing technology to get a general idea of all the steps of medicinal plant identification.

# Tables des matières

[Dédicace 1](#_Toc116974880)

[Remerciement 1](#_Toc116974881)

[Résumé 1](#_Toc116974882)

[Abstract 1](#_Toc116974883)

[Tables des matières 1](#_Toc116974884)

[Liste des abréviations 3](#_Toc116974885)

[Listes des tableaux 4](#_Toc116974886)

[Listes des figures 5](#_Toc116974887)

[Introduction Générale 6](#_Toc116974888)

[**Chapitre 1** Contexte Générale 7](#_Toc116974889)

[I- Contexte et Problèmatique 8](#_Toc116974890)

[II- Objectif du travail 9](#_Toc116974891)

[**Chapitre 2** Etat de l’art 10](#_Toc116974892)

[I- Synthèses 11](#_Toc116974893)

[1- Automatic Recognition of Medicinal Plants using Machine Learning Techniques 11](#_Toc116974894)

[2- Recognition of leaves of different medicinal plant species using a robust image processing algorithm and artificial neural networks classifier 13](#_Toc116974895)

[3- A Deep Learning Approach for Classification of Medicinal Plants 16](#_Toc116974896)

[4- Identification of Medicinal Plant Leaves Using Convolutional Neural Network 20](#_Toc116974897)

[5- Factors influencing the use of Deep Learning for Medicinal Plants Recognition 22](#_Toc116974898)

[6- CNN-based Leaf Image Classification for Bangladeshi Medicinal Plant Recognition 26](#_Toc116974899)

[7- Medicinal Plant identification in the wild by using CNN 30](#_Toc116974900)

[8- Classification of Plant Leaves Using New Compact Convolutional Neural Network Models 34](#_Toc116974901)

[II- Tableau des synthèses 37](#_Toc116974902)

[III- Conclusion 40](#_Toc116974903)

[**Chapitre 3** Background techniques 41](#_Toc116974904)

[I- Machine learning 42](#_Toc116974905)

[1- SVM 42](#_Toc116974906)

[2- Random Forest 43](#_Toc116974907)

[3- Naïve Bayes 44](#_Toc116974908)

[4- KNN 45](#_Toc116974909)

[II- Deep learning 46](#_Toc116974910)

[1- Réseau de neurones artificiels (ANN) 46](#_Toc116974911)

[2- Artificial neural Architectures 47](#_Toc116974912)

[i. MLP: multilayer perceptron neural network 47](#_Toc116974913)

[ii. PNN: probabilistic neural network 47](#_Toc116974914)

[iii. CNN: convolutional neural network 49](#_Toc116974915)

[3- Fonctions d’activation 49](#_Toc116974916)

[i. Sigmoid 49](#_Toc116974917)

[ii. Softmax 50](#_Toc116974918)

[iii. ReLU 50](#_Toc116974919)

[4- Backpropagation 51](#_Toc116974920)

[II- Transfer Learning 51](#_Toc116974921)

[1- AlexNet 52](#_Toc116974922)

[2- VGG 53](#_Toc116974923)

[III- Préparation du dataset 55](#_Toc116974924)

[**Chapitre 4** Réalisation et mise en œuvre 57](#_Toc116974925)

[I- Méthodologies de recherche 59](#_Toc116974926)

[1- Dataset 59](#_Toc116974927)

[2- Prétraitement des données 59](#_Toc116974928)

[3- Approches proposées 59](#_Toc116974929)

[i. CNN 59](#_Toc116974930)

[ii. AlexNet 59](#_Toc116974931)

[iii. VGG16 59](#_Toc116974932)

[4- Création du modèle Sequentiel 59](#_Toc116974933)

[5- Training model 59](#_Toc116974934)

[6- Implémentation 59](#_Toc116974935)

[II- Résultats et discusion 59](#_Toc116974936)

[III- Outils de développements 59](#_Toc116974937)

[1- Environnement de développement 59](#_Toc116974938)

[2- Technologies de développement 60](#_Toc116974939)

[IV- Conclusion 61](#_Toc116974940)

[**Conclusion générale** 62](#_Toc116974941)

[Bibliographie 62](#_Toc116974942)

[Annexe 62](#_Toc116974943)

# Liste des abréviations

ML Machine learning

DL Deep learning

ANN Artificial neural network

CNN Convolutional neural network

SVM Support vector machine

MLP Multilayers perception

VGG Visual Geometry Group

PV PlantVillage

PNN Probabilistic neural network

TL Transfer Learning

# Listes des tableaux

[Tableau 1: Caractèristiques et méthodes d'extraction 12](file:///C:\Users\JAMIL\Desktop\PFA\PFA.docx#_Toc114611393)

[Tableau 2: Classificateur utilisés et ses précisions 13](file:///C:\Users\JAMIL\Desktop\PFA\PFA.docx#_Toc114611394)

[Tableau 3 Precision des model transfer learning 33](#_Toc114611395)

[Tableau 4 Résultats des modèles proposées 36](file:///C:\Users\JAMIL\Desktop\PFA\PFA.docx#_Toc114611396)

[Tableau 5 tableau des synthèses 37](#_Toc114611397)

# Listes des figures

[Figure 1: Etapes de traitment d'image 15](#_Toc114611408)

[Figure 2: L'architecture CNN d'AyurLeaf 17](#_Toc114611409)

[Figure 3 L'architecture de CNN 20](#_Toc114611410)

[Figure 4 Vue d'ensemble d'un CNN et de ses principaux composants 24](#_Toc114611411)

[Figure 5 Modèle CNN pour l'extraction de caractéristiques et la classification 25](#_Toc114611412)

[Figure 6 Procédure d'extraction des caractéristiques 27](#_Toc114611413)

[Figure 7 Structure de réseau du modèle proposé 28](#_Toc114611414)

[Figure 8: le fonctionnement de l'algorithme Random Forest 44](#_Toc114611415)

[Figure 9:Probabilistic neural network (PNN) structure. 48](#_Toc114611416)

[Figure 10. Fonction d'avtivation sigmoide 50](#_Toc114611417)

[Figure 11. unités linéraires réctifiés 51](#_Toc114611418)

[Figure 12. AlexNet architecture 53](#_Toc114611419)

[Figure 13. Configurations de VGG 54](#_Toc114611420)

# Introduction Générale

Aujourd'hui, il n'y a pas de domaine qui n'utilise pas l'IA, en effet, l'intelligence artificielle apporte des outils modernes et bouleverse les processus existants dans de nombreux domaines tels que la santé, l'hôtellerie, l'industrie et l'agriculture. Le défi que nous avons aujourd'hui est de développer les algorithmes d'apprentissage automatique et d'apprentissage profond qui existent encore, d'autant plus que maintenant nous avons un énorme volume de données que nous pouvons exploiter et du matériel très sophistiqué.

Actuellement, dans le secteur de l'agriculture, on s'intéresse au Machine Learning (ML) pour alimenter des algorithmes intelligents, la première étape est de collecter des données massives (Big Data) grâce à des capteurs plantés dans le sol ou installés sur des tracteurs, grâce à des caméras, ou grâce à la cartographie des sols réalisée avec des drones. Dans ce travail, On s’intéresse à détection et classification automatique les feuilles des plantes medicinales et à la conception et la mise en œuvre d'un système d'apprentissage automatique pour cela.

Dans ce travail on va traiter les axes suivants :

* **Etats de l’art :** Dans ce chapitre, on va présenter des synthèses basées sur des articles scientifiques afin d’élaborer les différentes techniques et les solutions existants
* **Backgroung techniques :** Ce chapitre a pour objectif d'étudier en profondeur les technologies qui font l'objet de notre étude : les modèles de Machine learning et Deep learning et en particulier les modèles de classification d'images.
* **Réalisation et mise en œuvre :** Ce chapitre sera consacré à la présentation des aspects technique et les différents outils et méthodes utilisés pour la réalisation de ce projet. La partie de réalisation est l'aboutissement des phases précédentes car c'est dans celle-ci qu'est réalisé le produit du projet pensé.

# **Chapitre 1** Contexte Générale

## Contexte et Problèmatique

Les plantes jouent un rôle crucial dans la préservation de la vie et le maintien de la biodiversité sur terre en fournissant de l'air et de l'eau aux êtres vivants. Les plantes médicinales, l'une des principales catégories de plantes, servent de remède à de nombreuses maladies. Les connaissances sur les plantes médicinales transmises par les générations doivent être préservées et protégées.

L'identification et la classification des plantes ont été réalisées au cours des dernières décennies à l'aide de méthodes classiques de traitement et de classification des images. Ces méthodes utilisent des caractéristiques basées sur la forme, la texture et la couleur pour effectuer la classification. Le rapport d'aspect, l'excentricité, l'aplatissement, l'asymétrie, l'énergie, la corrélation, la variance de la somme, l'entropie et la compacité sont quelques-unes de ces caractéristiques. Le temps de calcul excessif requis pour l'extraction de caractéristiques artisanales est le problème majeur associé à ces méthodes classiques. Aujourd'hui, toutes les méthodes classiques sont remplacées par des techniques d'apprentissage automatique.

Les méthodes de vision par ordinateur et de traitement d'images peuvent combler le manque de taxonomistes experts et les besoins potentiels en matière d'identification et de classification des plantes médicinales. La forme, la couleur et les textures sont les caractéristiques spatiales et morphologiques importantes utilisées par les chercheurs pour classer les plantes. Mais la couleur n'est pas une caractéristique judicieuse pour la classification puisqu'elle varie selon les saisons et que différents stades d'une même feuille auront une couleur différente. Les taxonomistes utilisent les variations des caractéristiques des feuilles comme un outil pour la classification des plantes médicinales. De nombreux travaux de recherche ont été réalisés dans ce domaine. Mais le taux élevé de similarité interclasse dans les caractéristiques de forme, de couleur et de texture.

Les technologies de vision par ordinateur, de reconnaissance des formes et de traitement des images fournissent des résultats prometteurs pour l'identification et la classification des plantes médicinales. L'identification d'une plante médicinale ayant les valeurs médicinales requises est l'un des principaux défis à relever. Même si la médecine par les plantes n'a pas d'effets secondaires, un traitement utilisant une plante médicinale mal identifiée peut coûter la vie à un patient. Par conséquent, un système entièrement automatisé pour classer correctement les plantes médicinales est inévitable à l'heure actuelle

## Objectif du travail

L'objectif de ce projet est d'étudier les recherches effectuées dans ce domaine et d'extraire les méthodes utilisées, de les expliquer et de les simplifier, de les comparer en termes de précision et de facilité, et enfin de faire un travail expérimental en utilisant l'apprentissage profond CNN et la technologie de traitement des images pour se faire une idée générale de toutes les étapes de l'identification des plantes médicinales.

# **Chapitre 2** Etat de l’art

Dans ce chapitre, on va présenter des synthèses basées sur des articles scientifiques afin d’élaborer les différentes techniques et les solutions existants.

## Synthèses

### Automatic Recognition of Medicinal Plants using Machine Learning Techniques

[Article 1](https://www.researchgate.net/publication/316601212_Automatic_Recognition_of_Medicinal_Plants_using_Machine_Learning_Techniques)

Dans cet article, des techniques de vision par ordinateur ont été utilisées pour extraire plusieurs caractéristiques basées sur la forme des feuilles de plantes médicinales. Des algorithmes de *machine learning* ont ensuite été utilisés pour classer les feuilles de 24 espèces végétales différentes dans les catégories appropriées.

Le dataset utilisé a été réalisé en suivant une méthodologie, nous la reprenons dans les étapes suivantes :

Le pétiole de chaque feuille a été retiré, puis placé un par un sur une feuille de papier blanc avant d'être photographié. La taille de chaque image était de 1024x600 pixels. Les images sont enregistrées au format jpeg.

1. Automatic pre-processing steps: pour supprimer l'ombre, l'image doit d'abord être convertie au format HSV, puis divisée en ses différents canaux de couleur. Pour réduire le bruit dans l'image, un filtre flou médian avec une taille de fenêtre de 25 est appliqué à l'image résultante. L'étape suivante consiste à effectuer une opération de seuillage qui convertira l'image en une image binaire ne comportant que deux valeurs : les pixels noirs et blancs. Cette opération est réalisée à l'aide de la méthode de seuillage d'Otsu. Une opération d'ouverture est ensuite effectuée sur les images. Il s'agit d'une opération d'érosion suivie d'une dilatation. L'érosion a pour effet de réduire la taille des pixels de premier plan (blancs) tandis que la dilatation les agrandit. Cette opération est importante pour débarrasser l'image de nombreux petits pixels bruyants, qui sont les artefacts de l'opération de seuillage.
2. Feature extraction: Un certain nombre de caractéristiques de base ont été extraites des images. Il s'agit de la longueur, de la largeur, de la surface de la boîte de délimitation, de la surface de la feuille, du périmètre de la feuille, de la surface de la coque, du périmètre de la coque, du nombre de sommets, des cartes de distance horizontale et verticale, de la carte radiale à 45° et des valeurs RVB originales de chaque pixel.
3. Derived features : En utilisant les caractéristiques de base qui sont extraites directement de l'image, un certain nombre de caractéristiques dérivées sont calculées. Les ratios sont plus appropriés pour la comparaison car ils sont indépendants de la taille réelle de l'image en pixels. Les ratios sont présentés ci-dessous dans le tableau :

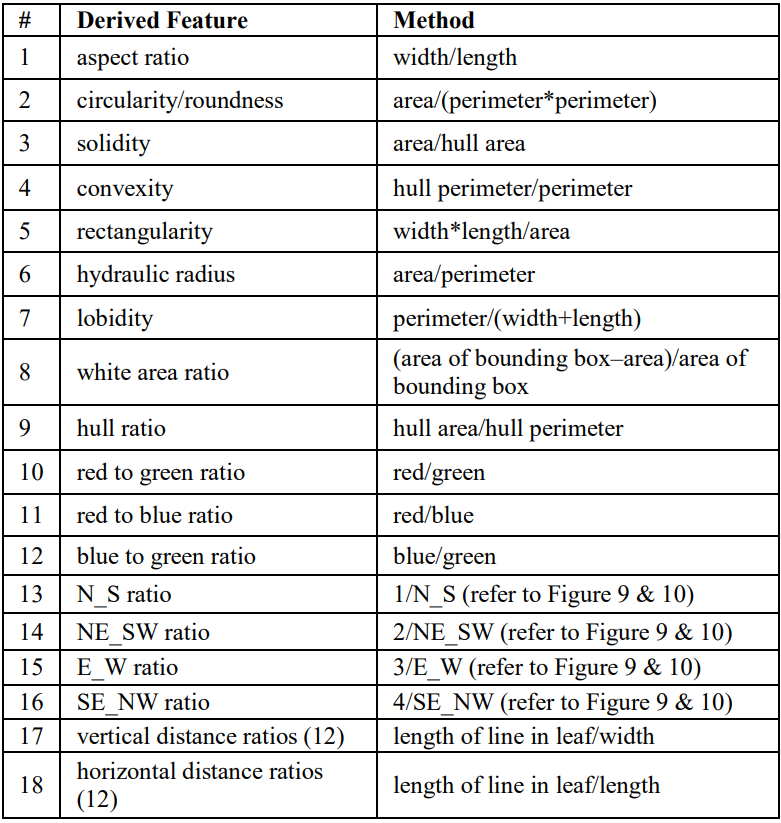


Tableau 1: Caractèristiques et méthodes d'extraction

Cinq classificateurs de *machine learning* différents ont été utilisés pour évaluer le taux de reconnaissance. Les résultats sont présentés dans le tableau. Le classificateur Random Forest obtient la meilleure performance avec une précision de 90,1%, c'est-à-dire que sur 720 feuilles, 649 feuilles ont été classées correctement alors que 71 ne l'ont pas été. Cette excellente performance indique la viabilité de telles approches assistées par ordinateur dans la classification de spécimens biologiques et son applicabilité potentielle dans la lutte contre la "crise taxonomique".

### Recognition of leaves of different medicinal plant species using a robust image processing algorithm and artificial neural networks classifier

Tableau 2: Classificateur utilisés et ses précisions

[Article 2](https://www.sciencedirect.com/science/article/abs/pii/S221478612100036X?via%3Dihub)

[PDF](https://sci-hub.se/10.1016/j.jarmap.2021.100327)

La classification ANNs cherche à classer une observation comme appartenant à une classe discrète en fonction des entrées. Les caractéristiques d'entrée (variables indépendantes) peuvent être de type catégorique ou numérique, mais nous avons besoin d'une caractéristique catégorique comme variable dépendante.

Dans cet article, les auteurs se concentrent sur un algorithme de traitement d'image robuste et un classificateur ANNs pour la reconnaissance des feuilles de différentes espèces de plantes médicinales. Nous allons reprendre les étapes pour atteindre cet objectif.

1. *Sample preparation*

Dans la présente étude, 40 sites ont été sélectionnés et 4 type(species) de plantes ont été collectés dans chaque site. En général, 6 groupes de plantes médicinales ont été rassemblés et étiquetés de A1 à A6.

1. *Image acquisition*

Les images prises avec un téléphone portable Galaxy avaient une taille de 1088 × 2220 pixels au format JPEG.

Une lentille externe, coaxiale à l'objectif du smartphone, a été utilisée pour améliorer la résolution des images des types végétaux et pour régler la vue de la chambre d'imagerie. Un bouclier d'ombrage composé de résine a été utilisé pour éliminer l'interférence réfléchissante créée par la paroi interne de la chambre.

1. *Image processing*

Un système de traitement d'images a été utilisé. Il consiste en une série de procédures qui sont utilisées pour amplifier la qualité de l'image, l'extraction des caractéristiques et la classification des caractéristiques. Le logiciel MATLAB a été utilisé pour appliquer le processus sur les images de plantes.

* 1. Image segmentation

La segmentation d'image est une étape très importante et facile à utiliser dans la technique de traitement d'image dans laquelle les images d'échantillon sont divisées en différents segments et seules les parties souhaitables des images sont traitées. À cet égard, différentes étapes ont été automatiquement exécutées par l'algorithme développé, notamment

a) la lecture des images de plantes,

b) la séparation des composantes R, G et B,

c) l'utilisation du canal B pour la détection des images cibles à partir du fond,

d) la conversion du canal B en image binaire,

e) l'inversion du canal B,

f) la suppression des bruits,

g) le remplissage des trous,

h) la suppression du fond des canaux R, G et B

i) la combinaison des images monochromes pour produire une image RGB sans fond.

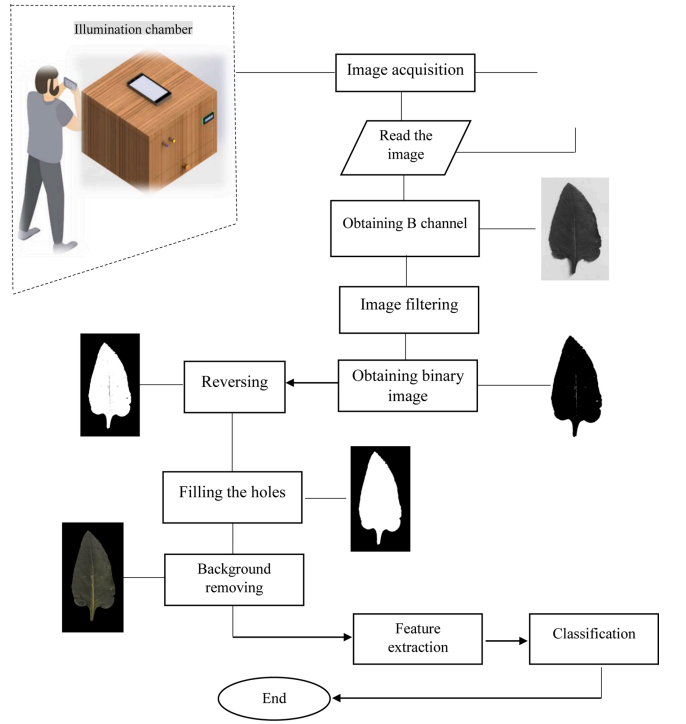


Figure 1: Etapes de traitment d'image

* 1. Feature extraction

Certaines caractéristiques de texture, de forme et de couleur ont été extraites en raison de la classification des attributs utiles. Avant d'extraire les caractéristiques, dans les différents espaces de couleur des images ont été automatiquement obtenus par l'algorithme de traitement d'image développé tels que I1I2I3, HIS, CrCgCb, NRNGNB, L\*a\*b\*, et le niveau de gris.[[1]](#footnote-1)

* 1. Medical plant classification

La classification des plantes médicinales étudiées a été effectuée en utilisant les caractéristiques efficaces des images des échantillons. Ainsi, la méthode des réseaux neuronaux artificiels (ANN) a été mise en œuvre pour classer les différents types d'échantillons de plantes médicinales.

Conclusion

Le meilleur modèle optimal avait une structure de 28- 10-6. Les résultats indiquent que le modèle peut classer suffisamment les différents types de plantes médicinales avec une précision de 100 %. Il convient de noter que les algorithmes d'extraction de texture, de forme et de couleur ont été largement utilisés avec succès dans les produits agricoles pour diverses tâches telles que la classification, la reconnaissance, etc. La principale contribution de cette recherche est liée à la conception et au développement d'algorithmes robustes pour l'extraction de la couleur, de la texture et de la forme. Par conséquent, l'intégration du système de vision artificielle proposé a un grand potentiel pour catégoriser et reconnaître divers produits agricoles en fonction des caractéristiques utiles de texture, de forme et de couleur.

### A Deep Learning Approach for Classification of Medicinal Plants

[Article 3](https://ieeexplore.ieee.org/document/8929394)

[PDF](https://sci-hub.se/10.1109/tencon.2019.8929394)

1. *MATERIALS AND METHODS*
   1. *AyurLeaf Dataset*

AyurLeaf est le jeu de données proposé qui consiste en 2400 images de feuilles de plantes médicinales que l'on trouve couramment dans le Kerala. Plus de 30 feuilles de 40 espèces végétales différentes ont été collectées et ont fait l'objet d'un échantillonnage. Les feuilles présentant de graves déformations ont été supprimées et 30 feuilles présentant une différence significative en termes de forme, de couleur et de taille ont été sélectionnées. Apparemment, 30 feuilles de chaque classe sont sélectionnées pour la suite du processus de numérisation. Les faces supérieure et inférieure de ces 30 feuilles sélectionnées sont scannées pour créer 60 images de feuilles par espèce. Seule la zone de la feuille est sélectionnée et recadrée à l'aide de l'éditeur d'images GIMP, puis chaque image est enregistrée au format jpg. Une convention d'appellation commune est utilisée pour étiqueter (label) chaque image, le nom de l'espèce végétale suivi d'un numéro de séquence unique. L'échantillonnage assure la nature diverse de l'ensemble de données au niveau des espèces végétales et aide le modèle à fournir des résultats de classification plus précis.

* 1. *AyurLeaf - Proposed CNN Architecture*

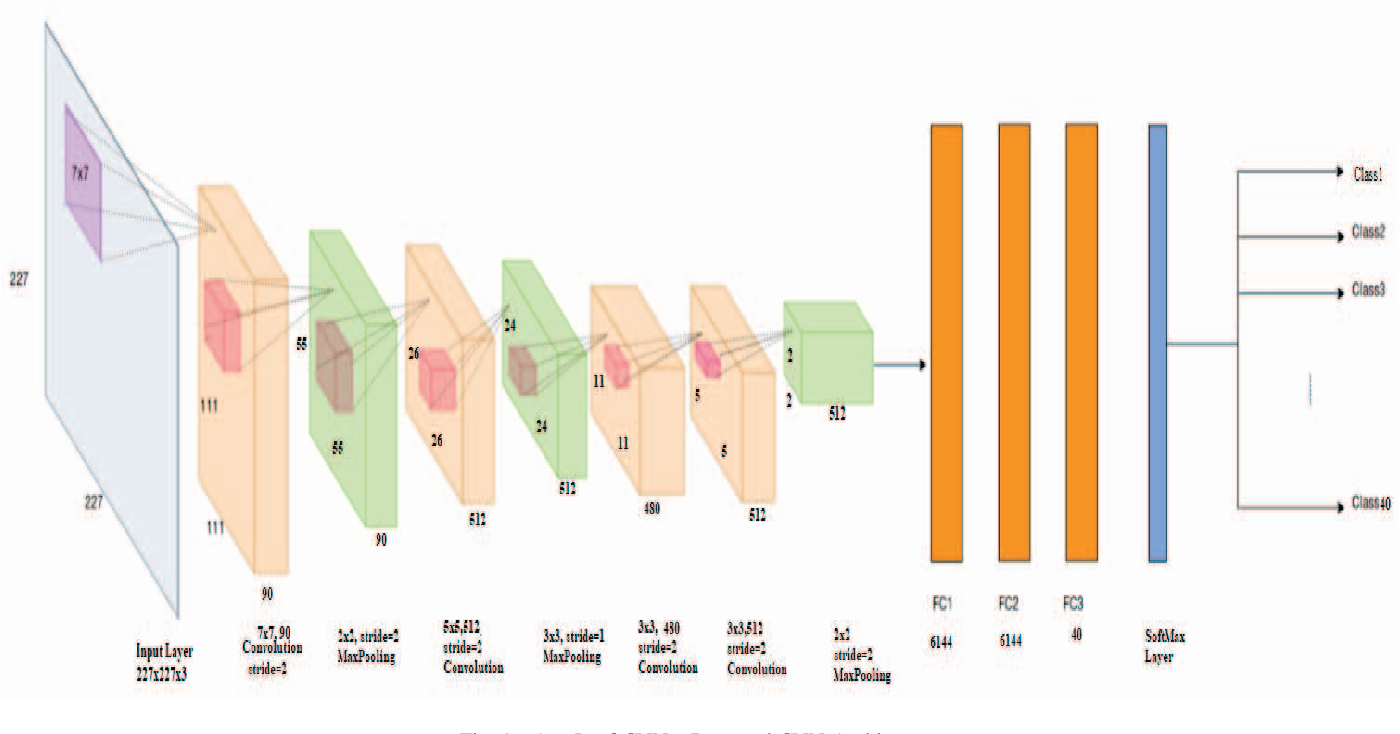


Figure 2: L'architecture CNN d'AyurLeaf

L'architecture CNN d'AyurLeaf est conçue et développée sur la base de l'architecture AlexNet. Le modèle abstrait de notre système proposé est donné dans la Fig.2.

La première couche est la couche d'entrée qui spécifie la dimension des images d'entrée. La deuxième couche, la couche de convolution, utilise 90 filtres (7x7) avec une taille de stride de 2. Cette couche est suivie d'une couche ReLU qui seuille la sortie, puis d'une couche de mise en commun maximale avec une taille de filtre de 2x2. Cette couche réduit exactement de moitié la taille de sa sortie. Cette couche de mise en commun maximale est suivie d'une deuxième couche de convolution qui opère sur 512 noyaux avec la dimension 5x5 et un stride de 2. La couche suivante est une couche ReLU suivie d'une couche de mise en commun maximale avec une taille de filtre 3x3 et un stride de 1*.*

Les deux couches suivantes sont deux couches de convolution dos à dos avec les configurations suivantes. Les deux utilisent des noyaux 3x3 avec un stride de 2 et le nombre de noyaux dans la première et la deuxième couche est respectivement de 480 et 512. Ces couches sont à nouveau suivies d'une couche ReLU et d'une couche de max pooling avec une taille de filtre de 2x2 et un stride de 2. La sortie de la dernière couche de mise en commun maximale est transmise à la première couche entièrement connectée qui se compose de 6144 neurones. La deuxième couche entièrement connectée est également composée de 6144 neurones. Et la troisième couche entièrement connectée comporte 40 neurones, ce qui est égal au nombre de classes de la plante médicinale à classer. Enfin, la sortie de la troisième couche entièrement connectée est transmise à la couche de classification softmax qui calcule les probabilités de classification pour chaque espèce.

Le modèle CNN proposé effectue les quatre étapes suivantes :

* + 1. *Image Acquisition*

Les échantillons de feuilles de plantes médicinales ont été collectés auprès de praticiens ayurvedic et dans les forêts tropicales du Kerala. Au moins 30 feuilles de 40 espèces différentes de plantes médicinales ont été collectées. Effectuez un échantillonnage manuel de base et retirez les feuilles gravement endommagées. Sélectionnez 30 feuilles de chaque espèce pour les scanner et 60 échantillons d'images par espèce de plante médicinale sont générés. Pour garantir la qualité des images de l'ensemble de données, les images de feuilles sont scannées à l'aide d'un scanner à plat en 1200 ou 600 dpi.

* + 1. *Image Pre-processing*

L'objectif de la phase de prétraitement est de transformer les images scannées en une dimension de 227x227x3 qui est l'un des formats acceptables des images d'entrée du CNN. Nous avons utilisé les images au format RGB et si les images ne sont pas dans le format requis, nous les avons converties dans le format spécifié 227x227x3. Puisque la dimension des images dans notre jeu de données varie, nous avons d'abord converti les images en une dimension NxN en effectuant un remplissage. Et enfin, nous avons redimensionné l'image rembourrée à la dimension 227x227x3.

* + 1. *Feature Extraction*

Un bon nombre de modèles CNN ont été conçus avec un nombre de couches, un nombre de filtres et une taille de filtre différents avec diverses options d'entraînement. Tous les modèles conçus sont entraînés et testés avec le jeu de données AyurLeaf et les résultats sont comparés à ceux des modèles CNN Alexnet et Dleaf. La performance de ces modèles dépend clairement de la qualité du jeu de données, du nombre de couches convolutionnelles, max-pooling, ReLU, des options de formation et du nombre de neurones dans les deux premières couches entièrement connectées. Enfin, le CNN Ayurleaf proposé est conçu.

AyurLeaf CNN a utilisé 80 % des échantillons d'images de l'ensemble des données pour l'apprentissage et les 20 % restants pour les tests. La précision du modèle dépend fortement du nombre d'images utilisées pour l'entraînement du modèle et du nombre d'itérations du CNN. Une fois le processus de formation terminé, les images de l'ensemble de test sont données pour obtenir la précision du modèle. Enfin, le modèle AyurLeaf fournit la précision de la formation et de la validation pendant la formation en utilisant le graphique de perte de précision, et la précision de la classification en utilisant la matrice de confusion.

1. *RESULTS AND DISCUSSION*

Une validation croisée cinq fois est effectuée sur l'ensemble de données AyurLeaf. La précision maximale obtenue pour une seule exécution sur cinq exécutions consécutives est de 98,46 %. Au total, 6000 itérations sont effectuées par l'algorithme sur le jeu de données.

Le modèle AyurLeaf CNN est testé et comparé aux modèles AlexNet, DLeaf et AlexNet affiné. Pour la formation et les tests par validation croisée, nous avons spécifiquement opté pour la validation croisée triple et la validation croisée quintuple. Notre modèle surpasse les modèles susmentionnés avec une précision de classification de 96,76 %.

*Conclusion*

AyurLeaf est un modèle de classification basé sur CNN qui a été entraîné et testé sur son propre jeu de données et sur le jeu de données DLeaf. Le classificateur SVM a obtenu la meilleure précision de classification sur le jeu de données AyurLeaf. Le modèle AyurLeaf peut être mis à niveau pour être adapté à la classification d'images de feuilles dans lesquelles une seule image contient des feuilles de plus d'une espèce végétale et plus d'une feuille de la même espèce végétale dans des orientations différentes.

### Identification of Medicinal Plant Leaves Using Convolutional Neural Network

[Article 4](https://iopscience.iop.org/article/10.1088/1742-6596/1845/1/012026)

Cette recherche vise à construire un système d'identification de neuf types de feuilles de plantes médicinales hypertensives à l'aide de CNN.

1. *Methods*
   1. *Data Acquition*

the leaves identified in this study consisted of nine medicinal leaves. There are 180 of training data used by grouped by type of leaf, with the number of each leaf 20 data.

* 1. *Convolutional Neural Network*

Le réseau neuronal convolutif (CNN) est l'une des méthodes de "Deep Learning" et de réseaux neuronaux les plus couramment utilisées pour analyser les données d'images visuelles. Les résultats des études utilisant les caractéristiques CNN avec différents classificateurs montrent la cohérence et l'excellence. L'architecture des CNNs est présentée dans la figure 3. Les CNN comportent deux couches, à savoir la couche d'apprentissage des caractéristiques et la couche de classification par rétropropagation.

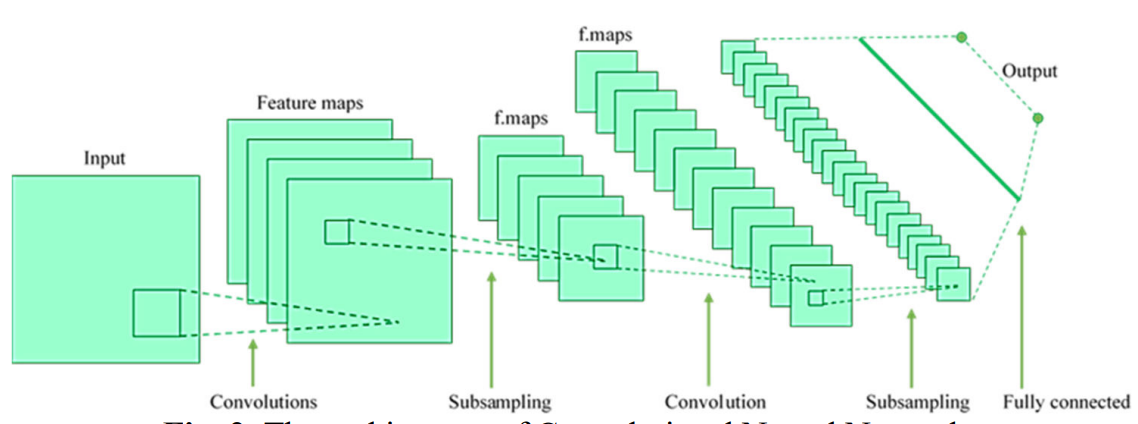


Figure 3 L'architecture de CNN

La couche convolutive est le principal élément constitutif du réseau neuronal convolutif. La couche convolutionnelle est constituée de neurones disposés de manière à former un filtre de longueur et de hauteur (pixels). Le bloc de filtre est décalé au-dessus de l'image et pour chaque position du produit, on prend le point situé entre le bloc de filtre et la partie de l'image couverte par le bloc. L'équation de la convolution peut être vue dans l'équation:

Cette couche modifie l'image à une profondeur différente en calculant le "Kernel" de l'image d'entrée, puis décale le "Kernel" en fonction de la valeur de l'enjambement de manière à produire une sortie ou communément appelée "Feature Map".

* 1. *Backpropagation*

La "backpropagation" est un algorithme d'apprentissage supervisé qui utilise l'architecture MLP et qui est utilisé pour la classification sur le CNN. Un autre avantage que possède ce ANN est sa capacité d'apprentissage (est adaptatif) et son immunité aux erreurs (tolérance aux pannes) ; avec ces avantages, l'ANN peut réaliser un système qui est résistant aux dommages (robuste) et qui fonctionne bien de manière constante. La backpropagation corrige le poids de la sortie d'information obtenue par feed-forward à la sortie devrait être. Ce processus est répété jusqu'à ce que l'erreur entre la sortie produite soit inférieure à la valeur spécifiée. Il y a trois étapes dans le processus de formation de cet algorithme, à savoir : la phase de feed-forward du modèle d'entrée de formation, la phase de back-ward des erreurs qui se produisent, et la phase de modification de la valeur du poids.

L'étape initiale consiste à initialiser le facteur de pondération en attribuant aléatoirement une petite valeur, puis à répéter la phase d'avance, la phase de recul, la phase de modification de la valeur du poids et à tester la condition d'arrêt jusqu'à ce que celle-ci soit remplie. Pour chaque paire d'apprentissage, effectuez la phase d'avance, la phase de retour et la phase de modification de la valeur de poids (calculez tous les changements de poids).

1. *Résultats et discusion*

Le programme va lire les pixels de l'image dans la matrice, puis extraire les paramètres de couleur. Puis il extrait les paramètres de forme. Les paramètres calculés seront comparés à toutes les feuilles des données de formation qui ont été précédemment identifiées dans la base de données, en utilisant une taille de différence minimale pour afficher les feuilles les plus appropriées. Ce logiciel utilise 180 données d'entraînement avec 20 types de données et est testé avec 50 données de feuilles composées de 45 feuilles qui correspondent aux données d'entraînement (5 feuilles de chaque type) et 5 feuilles qui ne sont pas reconnues. Le logiciel identifiera le nom de la feuille identifiée avec les informations relatives à la feuille, et affichera une notification indiquant que la feuille n'est pas reconnue si les données identifiées ne correspondent pas aux données d'entraînement.

Cette recherche a mis en œuvre avec succès la méthode du réseau neuronal convolutionnel pour extraire les caractéristiques des feuilles de plantes médicinales et les identifier en 9 classes de feuilles de plantes médicinales hypertendues sur la base de la valeur la plus proche entre les données de formation et les données de test.

### Factors influencing the use of Deep Learning for Medicinal Plants Recognition

[Article](https://iopscience.iop.org/article/10.1088/1742-6596/2089/1/012055/pdf)5

De nombreuses recherches ont été menées pour détecter les plantes médicinales à l'aide de techniques de classification d'images. Cependant, il existe différentes techniques proposées pour classer les objets ou les fleurs, et certaines d'entre elles utilisent des approches d'apprentissage profond.

1. Collection of Data Set

Les caractéristiques extraites pour l'identification sont la texture, la forme, la couleur, la physiologie ou la morphologie des feuilles. L'architecture CNN est appliquée au jeu de données pour entraîner et développer le système avec une grande précision.

Dans cette étude, les étapes suivantes sont utilisées pour la collecte des données :

1. Articuler le problème ; savoir ce que l'on veut prédire aide à décider des données utiles à collecter. L'exploration des données dans les catégories de la classification, du regroupement, de la régression et du classement aide à prendre la décision.
2. Établir un mécanisme de collecte des données : processus de collecte des données qui peut être automatisé ou manuel en fonction des besoins.
3. Formater les données : le format de fichier des images stockées doit être le même pour maintenir la cohérence.
4. Réduire la taille : les données doivent être collectées en fonction de l'objectif à atteindre, ce qui est essentiel pour notre ensemble de données.
5. Nettoyer complètement les données : les données manquantes, erronées ou moins représentatives sont supprimées pour rendre la prédiction plus précise.
6. Background Study and Methodology

Réseau neuronal convolutif (CNN) : Un réseau neuronal convolutif est une architecture d'apprentissage profond. La classification d'images est l'un des problèmes que peut résoudre un CNN. Il s'agit d'un réseau entraîné qui peut classer des images dans l'une des mille catégories prédéterminées. Un CNN peut être utilisé pour le traitement d'images, notamment pour la détection, la segmentation et la classification d'images. L'avantage important du CNN par rapport à un NN traditionnel est qu'il identifie automatiquement les caractéristiques significatives sans aucun contrôle de supervision. Un CNN est constitué de plusieurs couches qui transforment une entrée en sortie. La complexité des caractéristiques apprises augmente dans chaque couche cachée. Par exemple, les caractéristiques simples de détection sont apprises dans la première couche cachée, comme les bords, et la détection de formes plus complexes dans la dernière couche. Un modèle CNN comprend deux composants principaux : le composant d'extraction de caractéristiques et le composant de classification. Afin de comprendre l'architecture CNN, nous introduisons plusieurs concepts.

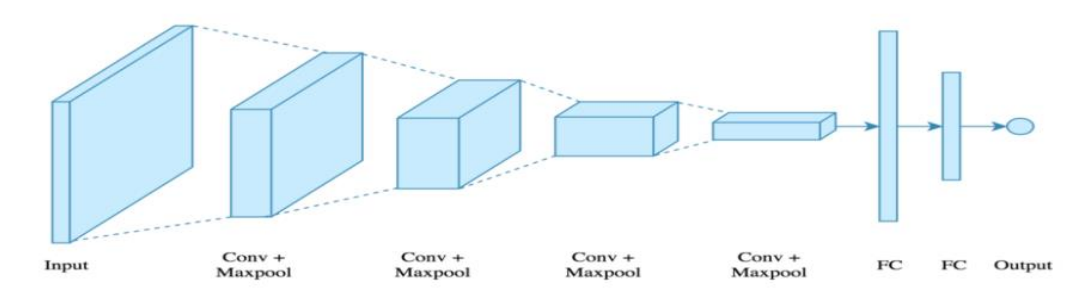


Figure 4 Vue d'ensemble d'un CNN et de ses principaux composants

Par rapport à un réseau neuronal typique dans lequel chaque neurone de la couche d'entrée est relié aux neurones de la couche cachée. Dans un CNN, nous avons des champs réceptifs locaux, un petit nombre de neurones de la couche d'entrée qui sont connectés aux neurones de la couche cachée. Les champs réceptifs locaux utilisent la convolution pour traduire une image en une carte de caractéristiques. La convolution peut effectuer l'opération mathématique de convolution en déplaçant un filtre sur l'image. À chaque région, une multiplication matricielle par éléments et une sommation du résultat sont effectuées. Un réseau neuronal peut réaliser cette opération en utilisant une fonction d'activation en transférant la somme pondérée de ses entrées à la couche suivante. Le CNN utilise la même fonction et applique la transformation à la sortie de chaque neurone en faisant passer le résultat de l'opération de convolution par une fonction d'activation.

la figure 3 montre l'extraction de caractéristiques et la classification pour une image de feuille.

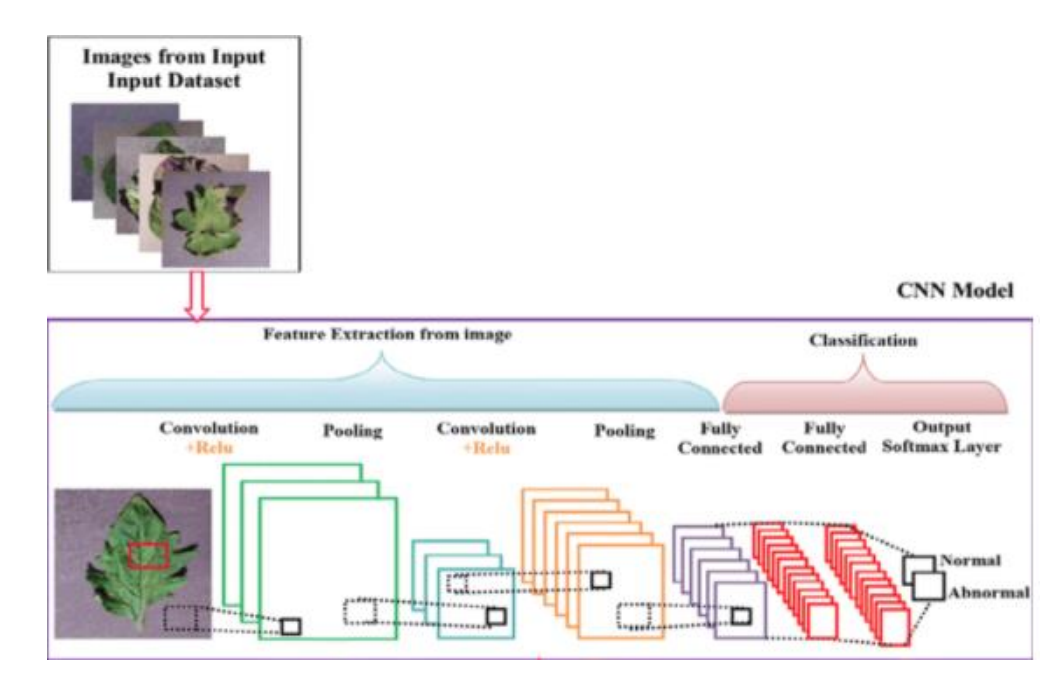


Figure 5 Modèle CNN pour l'extraction de caractéristiques et la classification

La conception architecturale se compose de cinq phases différentes. Chaque phase est responsable de sa propre tâche, depuis la mise en place du répertoire d'images jusqu'à la sortie finale du modèle.

1. Result and Discussion

Pour construire le modèle, le modèle InceptionV3model est utilisé comme modèle pré-entraîné/modèle de base. Ce modèle est disponible dans la bibliothèque d'applications keras. La construction du modèle comprend les étapes suivantes :

Tout d'abord, nous initialisons le modèle InceptionV3 comme modèle de base. Les paramètres sont entraînés sur "ImageNetdataset".

Ensuite, nous incluons une couche GAP (Global Average Pooling) au modèle pour réduire les dimensions spatiales d'un tenseur tridimensionnel, ce qui conduit finalement à minimiser l'overfitting. Elle réduit les dimensions h \* w \* d à des dimensions 1\*1 \* d en trouvant les valeurs moyennes de h et w.

Ensuite, nous ajoutons une couche entièrement connectée, FC, de 1024 nœuds cachés.

Enfin, la couche FC est connectée à notre couche de sortie pour prédire la sortie.

La couche FC prend principalement la sortie de GAP comme entrée et fournit la sortie à notre couche de sortie finale pour prédire la meilleure étiquette pour chaque image. Nous utilisons la fonction d'activation "Softmax", car le nombre de classes de sortie est supérieur à deux. Nous utilisons 65 couches dans le modèle et rendons 249 couches non entraînables.

Le résultat est l'une des parties les plus importantes de tout projet. Dans ce projet, nous avons atteint une précision de 100% sur le modèle InceptV3. Il faut 3 époques pour atteindre cette précision.

Conclusion

Ce modèle CNN n'est pas générique pour tous les différents types d'espèces végétales. Il est seulement limité à quatre classes différentes. Le modèle développé ne concerne que cinq espèces végétales différentes, les valeurs d'indice sont disposées dans l'ordre de [0, 1, 2, 3, 4] respectivement pour les étiquettes de classes citées ci-dessus et une précision de 96,67% est atteinte.

### CNN-based Leaf Image Classification for Bangladeshi Medicinal Plant Recognition

[Article 6](https://sci-hub.se/10.1109/etcce51779.2020.9350900)

Dans cette recherche, on a proposé un système automatisé pour la classification des plantes médicinales, qui aidera les gens à identifier rapidement les espèces végétales utiles. Un nouveau jeu de données de 10 plantes médicinales du Bangladesh est introduit, collecté dans différentes régions du pays, et certaines des images collectées à partir de différentes sources. Ensuite, un CNN à trois couches est utilisé pour extraire les caractéristiques de haut niveau pour la classification formée avec la technique d'augmentation des données.

1. DATA COLLECTION AND AUGMENTATION

Un nouveau jeu de données sur les plantes médicinales du Bangladesh est présenté. Il contient 37 693 images de 10 plantes médicinales différentes avec une classe par défaut. 39% des images ont été collectées auprès de différentes sources, et les 61% restants ont été prises dans différentes conditions à l'aide d'un téléphone portable. Comme le réseau de neurones convolutifs nécessite un grand nombre d'images d'entraînement, la technique d'augmentation peut apporter un effet massif dans cette petite quantité d'images. Ainsi on a appliqué des techniques d'augmentation telles que le retournement et différents angles de rotation.

1. NETWORK ARCHITECTURE

Dans ce travail, on construit un petit réseau de neurones convolutifs à partir de zéro pour classifier les classes exactes. Techniquement, dans un réseau neuronal convolutif, chaque image d'entrée passe par une série de couches convolutives avec différents noyaux. Ensuite, l'opération de convolution est effectuée par une multiplication matricielle par éléments et additionne le résultat qui devient une carte de caractéristiques. Ici, on a appliqué 128 types différents de filtres de poids pour produire la carte de caractéristiques. Comme on a utilisé différents filtres, on obtient différents types de cartes de caractéristiques. On empile donc toutes les cartes de caractéristiques et on les fait passer par une fonction d'activation pour les additionner et aussi pour avoir des effets de non-linéarité. La figure 6 décrit la procédure d'extraction de caractéristiques de base de l'architecture CNN.

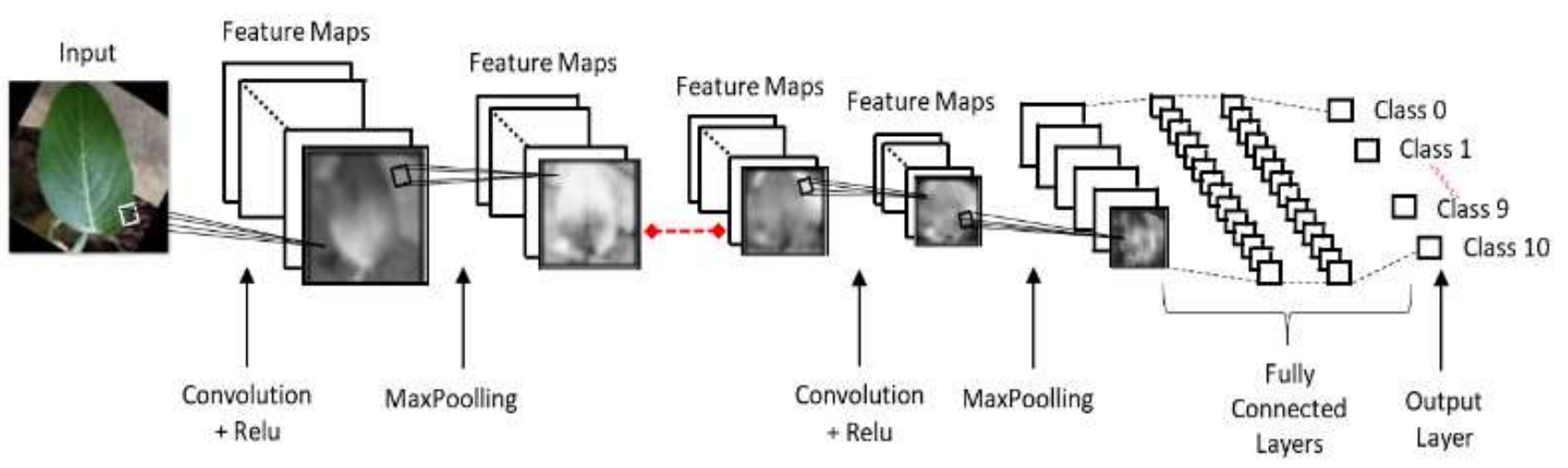


Figure 6 Procédure d'extraction des caractéristiques

L'architecture CNN proposée, telle qu'illustrée à la figure 7, est constituée de plusieurs couches, dans lesquelles sont placées séquentiellement les couches convolutionnelles, le bruit gaussien, la normalisation du lot, le Max Pooling et le Dropout. Ce schéma a été répété trois fois. Ensuite, une grande couche dense a été placée à l'extrémité de sortie du réseau pour tenter de mieux traduire le grand nombre de cartes de caractéristiques en valeurs de classe. L'image d'entrée était de 128 x 128 avec 3 canaux et a ensuite été alimentée par un CNN de 128 noyaux différents, chacun ayant une taille de filtre de 3x3, et une fonction d'activation (ReLU) a également été ajoutée pour l'effet de non-linéarité qui est défini comme suit :

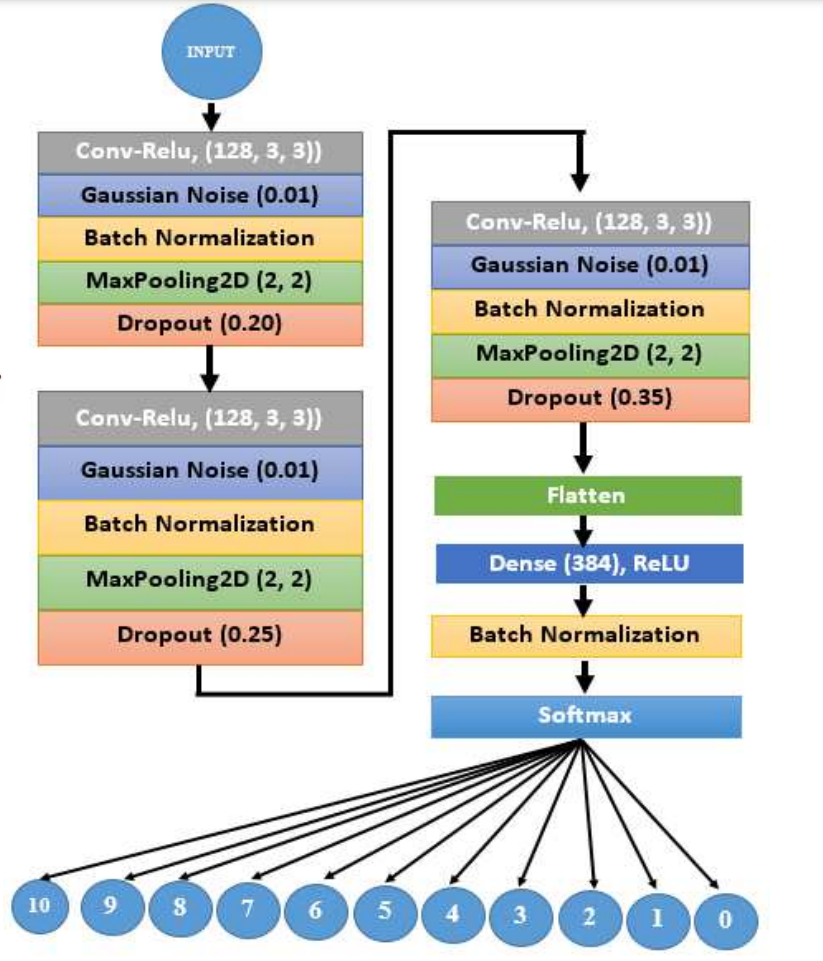


Figure 7 Structure de réseau du modèle proposé

1. EXPERIMENTAL RESULT

Dans l'expérience, les images ont été divisées en deux parties : le jeu de données des images d'entraînement et celui des images de test. Pour utiliser l'ensemble de données, nous devons d'abord traiter les images car les machines ne peuvent pas prendre les images telles quelles. Les données ont donc été prétraitées pour être adaptées aux étapes de formation à venir. Dans les techniques de prétraitement, nous avons d'abord redimensionné les images en 128\*128 pour établir une taille fixe. Ensuite, les images sont converties en tableau numpy. Après cela, chaque image a été normalisée en divisant chaque valeur de données par 255, qui est l'observation maximale, et en mettant les données à l'échelle [0, 1].

Après le prétraitement des données, elles ont été envoyées au réseau neuronal convolutionnel pour l'extraction de caractéristiques.

Nous avons utilisé la technique de double augmentation lors du prétraitement, et au cours de la période d'apprentissage, les autres méthodes d'augmentation telles que le décalage en largeur, le décalage en hauteur, la plage de zoom, etc. ont été appliquées. Le modèle mentionné a été formé en utilisant des taux d'apprentissage multiples à travers les époques, d'abord 0,01 pour 10 époques, puis 0,001 pour 165 époques et enfin 0,0001 pour 125 époques. Il a été entraîné avec 300 époques et une petite taille de lot de 64. L'ensemble de la phase d'entraînement s'est terminée en environ 7,5 heures, où chaque époch a pris près de 87 secondes. Pendant la période d'apprentissage, les poids ont été optimisés par backpropagation. La backpropagation est un processus dans lequel les poids des couches ont été correctement réglés par rapport à la fonction d'optimisation tout en effectuant des passages en avant et en arrière.

Le modèle proposé a obtenu une précision de 71,3 %, où la moyenne macro a diminué, mais la moyenne pondérée a augmenté.

L'entraînement a été effectué sur un environnement Google Colab doté d'un service GPU et de 25 Go de RAM. Le modèle d'apprentissage a été développé dans Keras, une bibliothèque Python pour l'apprentissage profond, en utilisant le backend Tensorflow et entraîné sur GPU.

Conclusion

Dans ce travail, on a introduit une architecture d'attention pour la tâche d'extraction de caractéristiques pour la classification des plantes et identifié les plantes médicinales à partir d'images de feuilles. on a collecté un jeu de données de plantes médicinales bangladaises, on les a prétraitées et on a classé les espèces végétales en utilisant une technique d'apprentissage profond. Dans notre article, nous avons décrit la méthodologie de l'architecture. on a analysé les résultats basés sur les ensembles d'entraînement et de test, et notre résultat est assez impressionnant, qui identifie correctement 71,3 % des feuilles. En outre, les résultats de l'architecture proposée sont prometteurs et comparables à d'autres méthodes existantes grâce à notre jeu de données, nous avons à la fois des images de feuilles simples et composées.

### Medicinal Plant identification in the wild by using CNN

[Article 7](https://sci-hub.se/10.1109/ictc49870.2020.9289480)

Le modèle CNN a connu un succès considérable, il joue un rôle principal dans la compréhension des différentes caractéristiques des images. Nous appliquons des modèles CNN, tels que Resnet50 et DenseNet, pour améliorer le taux de précision sur le jeu de données VNPlant-200.

1. METHOD

On applique les modèles CNN, des solutions avancées de vision par ordinateur et la méthode d'apprentissage par transfert pour classer les images de plantes médicinales vietnamiennes.

* 1. Convolutional Neural Network

Le Deep Learning (DL) a connu une croissance extrêmement rapide qui repose principalement sur des données d'entraînement énormes. La base de données la plus utilisée pour le DL était ImageNet[[2]](#footnote-2) avec 1,2 million d'images et 1 000 classes différentes.

AlexNet[[3]](#footnote-3), GoogLeNet[[4]](#footnote-4), VGG[[5]](#footnote-5). D'autres alternatives et des architectures avancées plus efficaces ont été proposées, notamment DenseNet[[6]](#footnote-6), FractalNet[[7]](#footnote-7), Inception units[[8]](#footnote-8), et Residual Networks[[9]](#footnote-9). Elle peut atteindre une meilleure performance que les méthodes conventionnelles.

* 1. State of the art models

**VGG16** : est une première architecture de neurones profonds après le succès d'Alexnet. L'équipe de VGG a empilé de nombreuses couches convolutionnelles et pleinement connectées et a obtenu de meilleures performances en utilisant le plus petit filtre initial des filtres convolutionnels 3×3. VGG16 obtient une précision parmi les cinq meilleures sur le jeu de données ImageNet[[10]](#footnote-10), qui comprend plus de 14 millions d'images de 1 000 classes.

**Inception** : l'idée du module est de concaténer plusieurs structures locales optimales avec une corrélation élevée analysée à partir de la couche précédente. Il utilise des opérateurs convolutifs de différentes tailles, tels que 1×1, 3×3, 5×5 et des techniques de fusion. Ils sont susceptibles d'être des types de présentation multi-échelle dans le schéma pyramidal. Pour la conception de l'Inception avec réduction, elle permet d'augmenter plusieurs nœuds à chaque couche sans affecter la couche de calcul suivante. Ce réseau est totalement étendu à 22 couches avec des paramètres pré-entraînés.

**Resnet** : a déclaré qu'à l'origine H(x) est une fonction de cartographie prédite qui apprend une cartographie de l'entrée à la sortie. Alternativement, définissons une autre cartographie F(x) = H(x) - x et donc à nouveau H(x) = F(x) + x. Maintenant H(x) - fonction résiduelle - est plus facile à optimiser en référence à l'entrée de la couche. Pour le modèle ResNet-50, il suffit de remplacer chaque bloc résiduel à deux couches par un bloc goulot d'étranglement à trois couches qui utilise des convolutions 1 × 1. Cela permet de réduire, puis de rétablir la profondeur de canal et la charge de calcul lors du calcul de la convolution 3 × 3

**DenseNet** : est un réseau proposé récemment pour l'identification visuelle d'objets. Il ressemble à Resnet en se composant de blocs denses et de couches de transition. Empilement de blocs denses - couches de transitionblocs denses - couches de transition. Avec le CNN traditionnel, si nous avons L couches, nous avons L connexions, et dans le densenet, nous avons L(L + 1) = 2 connexions.

**Xception[[11]](#footnote-11)** est proposé pour remplacer le module Inception en utilisant des convolutions séparables en profondeur. Ce réseau commence à séparer légèrement les deux en utilisant une convolution 1×1 pour projeter l'entrée originale dans des espaces d'entrée plus petits et en utilisant différents types de filtres pour transformer ces blocs de données 3D plus petits.

**MobileNet[[12]](#footnote-12)** : ces dernières années, l'intégration de l'intelligence artificielle dans les appareils mobiles a suscité un vif intérêt dans la communauté des chercheurs. Bien que le téléphone intelligent dispose d'une grande capacité de calcul et de stockage, il n'est pas comparable à un ordinateur. Le MobileNet a été introduit pour les appareils mobiles et intégrés qui permettent d'ajuster les hyperparamètres pour qu'ils soient adaptés aux applications spécifiques.

1. EXPERIMENTAL RESULTS
   1. Data Preparation

Le jeu de données VNPlant-200 contient un total de 20 000 images de plantes médicinales appartenant à 200 catégories. Toutes les images étiquetées de ce jeu de données ont été divisées en 50 : 10 : 40 correspondants aux ensembles de formation, de validation et de test. Les images de 256 × 256 pixels sont considérées comme augmentées de façon aléatoire pour chaque sous-ensemble de formation et de validation. Ensuite, les images sont retournées horizontalement puis recadrées à la taille de 224 × 224 pixels.

* 1. Experimental setup

Six modèles d'apprentissage profond, tels que VGG16, VGG19, Resnet50, InceptionV3, Densenet121, Xception et MobileNetV2 sont appliqués pour s'entraîner sur le jeu de données VNPlant-200. Tous les modèles disponibles dans Keras et Tensorflow 2.2 ont été initialisés en tant que modèles pré-entraînés pour apprendre de nouvelles caractéristiques de feuilles. Leurs architectures originales entraînées sur ImageNet ont été légèrement modifiées pour classifier 200 espèces. L'expérience et l'analyse sont réalisées sur un PC cadencé à un processeur i7, 16 Go de RAM et une mémoire GPU Gefore RTX 2060. En raison de la nécessité d'entraîner les modèles CNN à l'aide d'un grand nombre de données, nous avons adopté les techniques d'amélioration des données de Keras. Cette technique a permis de faire pivoter, de zoomer et de redimensionner les images actuelles en 224 × 24 × 3 dimensions. Cette technique est développée pour réduire l'overfitting des modèles d'apprentissage profond.

* 1. Results

L'objectif de cette recherche est de proposer une méthode basée sur CNN pour la classification de plantes à grande échelle. Les modèles expérimentaux sont VGG16, Resnet50, InceptionV3, Xception, DenseNet121 et MobileNetV2. Le modèle pré-entraîné sur le jeu de données ImageNet est appliqué avec les mêmes poids configurés par une couche entièrement connectée de 200 neurones. Les techniques d'optimisation Adam[[13]](#footnote-13) et Early Stopping[[14]](#footnote-14) sont appliquées avec un taux d'apprentissage initial fixé à lr = 0,0001. Pendant la phase de formation de six CNN profonds, nous avons eu besoin de 100 époques de formation pour obtenir une précision de validation satisfaisante. Les résultats sont présentés dans le tableau 2, qui montre que le modèle Xception a atteint 88,26% et surpasse les autres modèles.

Tableau 3 Precision des model transfer learning

|  |  |
| --- | --- |
| Models | Accuracy |
| VGG16 | 76 |
| InceptionV3 | 82.5 |
| MobileNetV2 | 87.92 |
| Resnet50 | 88 |
| Densenet121 | 88 |
| Xception | **88.26** |

Conclusion

L'identification des plantes joue un rôle majeur dans la recherche des plantes médicinales et de la botanique. Nous avons utilisé des modèles CNN d'apprentissage profond (VGG16, Resnet50, Inceptionv3, DenseNet121, Xception et MobileNetV2) et le VNPlant-200 est entraîné sur un modèle pré-entraîné. Nous avons classifié 200 couches de feuilles différentes et amélioré de manière significative les performances de classification. Bien que la performance du système soit assez bonne, nous pensons qu'elle pourrait être améliorée en ajoutant plus d'images et en ajoutant plus de couches.

### Classification of Plant Leaves Using New Compact Convolutional Neural Network Models

[Article 8](https://www.mdpi.com/2223-7747/11/1/24/htm)

Ce travail traite de la classification d'une plante à l'aide d'un modèle CNN compact récemment développé et d'AlexNet avec apprentissage par transfert. Les neuf classes appartenant à neuf espèces différentes d'images de plantes de la base de données PlantVillage sont utilisées pour la classification. De plus, les 32 classes de la base de données Flavia sont classées.

Les ensembles de données PV et Flavia sont augmentés séparément, et les images sont redimensionnées à la taille requise. La taille de l'image d'entrée pour les modèles proposés est de 256 × 256 × 3, et la taille de l'image d'entrée pour AlexNet est de 227 × 227 × 3. L'ensemble de données est ensuite divisé en 80-20% de données d'entraînement et de données de test. Les modèles proposés et AlexNet sont entraînés avec l'ensemble de données d'entraînement pour la classification des espèces végétales. Le modèle formé est utilisé pour la validation avec les données de test pour la prédiction de la nouvelle classe de données.

Le jeu de données PV comprend des images de feuilles saines et malades de 38 classes différentes. Le jeu de données comprend des catégories de feuilles saines et malades de neuf espèces végétales. On considère ici les espèces végétales à des fins de classification. Le jeu de données Flavia avec 32 classes a été utilisé pour la classification avec les modèles proposés. Le jeu de données Flavia est composé d'une classe variante de la feuille de la plante appartenant à différentes récoltes.

Le Pre-processing des données est essentiel pour maintenir l'uniformité et le bon fonctionnement de l'algorithme. L'apprentissage profond se comporte bien lorsque l'ensemble de données d'entrée est aussi large que possible et évite le surajustement. Les modifications les plus infimes, invisibles à l'œil humain, comme l'ajout de bruit et de flou aux images d'entrée, peuvent aider les CNN à apprendre des caractéristiques plus robustes. Dans ce travail, l'ensemble de données est augmenté d'un flou gaussien, d'un bruit de sel et de poivre avec une mise à l'échelle aléatoire de 0,95 à 1,05 dans une direction horizontale et verticale, et une rotation aléatoire dans la plage de -30° à 30° des images.

L'autre augmentation réalisée ici concerne la rotation et le retournement de l'ensemble de données. Dans l'augmentation de position, les images sont tournées de 45°, 135°, 225° et 315° et retournées horizontalement et verticalement. L'augmentation de la couleur de la saturation, de la teinte et du contraste est ajoutée. La saturation représente le degré de pureté ou de coloration d'une couleur. La teinte représente la couleur (bleu, rouge, vert, etc.) ; sa valeur varie de 0 à 360. L'égalisation de l'histogramme est effectuée pour évaluer la valeur de contraste dans l'augmentation de la couleur. L'égalisation de l'histogramme est connue pour améliorer la précision.

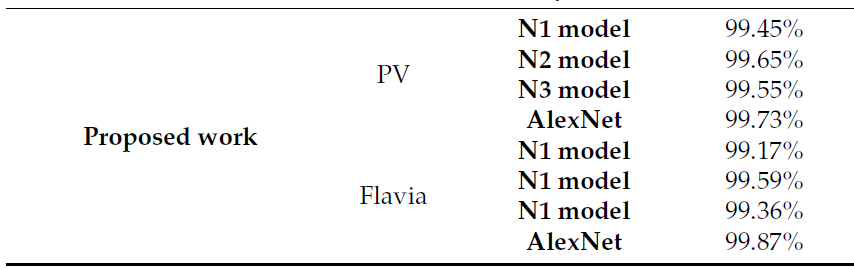
La classification des données de feuilles de plantes est réalisée sur un jeu de données de 38 400 images et un jeu de données augmenté de 336 000 images. Le réseau d'apprentissage profond utilisé dans ce travail est le modèle CNN 1 (modèle N1), le modèle CNN 2 (modèle N2), le modèle CNN 3 (modèle N3) et le modèle AlexNet avec apprentissage par transfert. Les images d'entrée sont redimensionnées à la taille 256 × 256 × 3 pour les modèles développés proposés, et les images sont redimensionnées à 227 × 227 × 3 pour le modèle AlexNet.

Tableau 4 Résultats des modèles proposées

## Tableau des synthèses

Tableau 5 tableau des synthèses

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Reference | | Objectives du travail | Dataset | | | | Classificateurs | Accuracy (%)  (Précision) |
| Auteurs | date | Nom | Size | N. de classes  (Species) |  |
| 1 | A. Begue, V. Kowlessur, F. Mahomoodally, U. Singh, S. Pudaruth | 2017 | classer les feuilles de 24 espèces végétales | Propre données | 720 images | 24 | * longueur * largeur * surface de la boîte englobante * surface de la feuille * périmètre de la feuille * surface de la coque * périmètre de la coque * nombre de sommets * cartes de distance horizontale et verticale * carte radiale à 45° et les valeurs RVB originales de chaque pixel. | Random Forest (numTrees=100) | 90.1 |
| Multilayer Perceptron Neural Network (Epochs=500) | 88.2 |
| Support Vector Machine (PolyKernel and c=4.0) | 87.4 |
| Naïve Bayes | 84.3 |
| k-Nearest Neighbour (k=1) | 82.5 |
| 2 | Rahim Azadnia , Kamran Kheiralipour | 2021 | la reconnaissance des feuilles de différentes espèces de plantes médicinales | Propre données | 960 images | 6 | * Texture * Forme * Couleur | ANN (28-10-6) | 100 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | Dileep M.R  Pournami P.N. |  | Classification des plantes Médicinales | AyurLeaf | 2400 images | 40 | * Texture * Forme * Couleur | CNN (Alexnet model) | | 94.87 |
| CNN (D-leaf model) | | 93.16 |
| CNN (AyurLeaf model) | | 95.06 |
| SVM (Ayurleaf model) | | 96.76 |
| 4 | Yuanita A. Putri, Esmeralda C. Djamal , Ridwan Ilyas | 2021 | Identification et classification des plantes Médicinales | Propre données | 180 images | 9 | - Forme  - Couleur | CNN (Backpropagation) | |  |
| 5 | Anchitaalagammai J V , Shantha Lakshmi Revathy J S , Kavitha S , Murali S | 2021 | Reconnaissance des plantes medicinales | Propre données | 58,280 images | 5 | la texture, la forme et la couleur des feuilles, physiologique ou morphologique | CNN(InceptionV3model) | | 96.67% |
| 6 | Raisa Akter, Md Imran Hosen | 2020 | Classification des plantes medicinales de Bangladesh | Propre données | 37,693 images | 10 | toutes les caractéristiques visuelles | CNN | | 71.3% |
| 7 | Trung Nguyen Quoc, Vinh Truong Hoang | 2020 | Identification et classification des plantes medicinales | VNPlant-200 | 20,000 images | 200 | toutes les caractéristiques visuelles | CNN | VGG16 | 76.00% |
| InceptionV3 | 82.50% |
| MobileNetV2 | 87.92% |
| Resnet50 | 88% |
| Densenet121 | 88% |
| Xception | 88.26% |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | Shivali Amit Wagle, R. Harikrishnan, Sawal Hamid Md Ali, Mohammad Faseehuddin | 2021 | Classification des feuilles des plantes medicinales | PlantVillage | 54303 | 38 | toutes les caractéristiques visuelles | PlantVillage | N1 model | 99.45% |
| N2 model | 99.65% |
| N3 model | 99.55% |
| AlexNet | 99.73% |
| Flavia | 1907 | 32 | Flavia | N1 model | 99.17% |
| N2 model | 99.59% |
| N3 model | 99.36% |
| AlexNet | 99.87% |

## Conclusion

Dans ces articles, nous trouvons quelques méthodes efficaces pour la classification des images de feuilles de plantes médicinales, en adoptant, les caractéristiques de forme et de texture des feuilles de la plante qui sont extraites après le prétraitement des images de feuilles avec plusieurs méthodes. Et en utilisant, divers modèles tels que la random forest, le réseau neuronal BP(BPNN), le réseau neuronal probabiliste(PNN), l'algorithme de classification K-voisin le plus proche et le classificateur de la machine à vecteur de support sont utilisés dans l'expérience de comparaison pour étudier leur efficacité de classification. Les résultats de l'expérience indiquent qu'il n'est pas possible d'obtenir un taux de reconnaissance raisonnablement bon en se basant uniquement sur les caractéristiques de forme ou de texture de la feuille. Le taux de reconnaissance est considérablement amélioré si les caractéristiques de forme et de texture sont utilisées ensemble dans le processus de classification. Dans le cas où les caractéristiques de forme et de texture sont utilisées comme base de la classification des images de feuilles, d'autres modèles CNN d'apprentissage profond (AlexNet, VGG16, Resnet50, Inceptionv3, DenseNet121, Xception et MobileNetV2) sont utilisés, qui sont des modèles pré-entraînés sur certains jeux de données tels que ImageNet , LeafSnap, PlantVillage and Flavia.

# **Chapitre 3** Background techniques

Ce chapitre a pour objectif d'étudier en profondeur les technologies qui font l'objet de notre étude : les modèles de Machine learning et Deep learning et en particulier les modèles de classification d'images.

## Machine learning

L'apprentissage automatique est un domaine de recherche consacré à la compréhension et à l'élaboration de méthodes qui "apprennent", c'est-à-dire des méthodes qui exploitent les données pour améliorer les performances d'un ensemble de tâches. Les algorithmes d'apprentissage automatique construisent un modèle sur la base d'un échantillon de données, appelé données d'apprentissage, afin de faire des prédictions ou de prendre des décisions sans être explicitement programmés pour le faire. Les algorithmes d'apprentissage automatique sont utilisés dans une grande variété d'applications, telles que la médecine, le filtrage du courrier électronique, la reconnaissance vocale et la vision par ordinateur, où il est difficile ou irréalisable de développer des algorithmes conventionnels pour effectuer les tâches nécessaires.[[15]](#footnote-15)

 La classification d’images fait référence aux technologies du ML qui étiquettent les lieux, les logos, les personnes, les objets, les bâtiments et plusieurs autres variables dans des images. Il existe de nombreux algorithmes de classification d’images pour la reconnaissance d’images, nous en aborderons quelques-uns dans les axes suivants.

### SVM

Dans le domaine de l'apprentissage automatique, les machines à vecteurs de support (SVM, également appelées réseaux à vecteurs de support [1]) sont des modèles d'apprentissage supervisé avec des algorithmes d'apprentissage associés qui analysent les données pour la classification et l'analyse de régression. Les SVM sont l'une des méthodes de prédiction les plus robustes. Étant donné un ensemble d'exemples d'apprentissage, chacun étant marqué comme appartenant à l'une des deux catégories, un algorithme d'apprentissage SVM construit un modèle qui affecte les nouveaux exemples à l'une ou l'autre catégorie, ce qui en fait un classificateur linéaire binaire non probabiliste. Le SVM fait correspondre les exemples d'apprentissage à des points dans l'espace de manière à maximiser la largeur de l'écart entre les deux catégories. Les nouveaux exemples sont ensuite mis en correspondance dans ce même espace et leur appartenance à une catégorie est prédite en fonction du côté de l'écart qu'ils occupent.

En plus de réaliser une classification linéaire, les SVM peuvent efficacement réaliser une classification non linéaire en utilisant ce que l'on appelle "kernel trick", en mappant implicitement leurs entrées dans des espaces de caractéristiques à haute dimension.

Lorsque les données ne sont pas étiquetées, l'apprentissage supervisé n'est pas possible, et une approche d'apprentissage non supervisé est nécessaire, qui tente de trouver un regroupement naturel des données en groupes, puis de faire correspondre les nouvelles données à ces groupes formés. L'algorithme de regroupement par vecteurs de support, créé par Hava Siegelmann et Vladimir Vapnik, applique les statistiques des vecteurs de support, développées dans l'algorithme des machines à vecteurs de support, pour classer les données non étiquetées.[[16]](#footnote-16)

### Random Forest

Random Forest est un algorithme populaire d'apprentissage automatique qui appartient à la technique d'apprentissage supervisé. Il peut être utilisé pour les problèmes de classification et de régression en apprentissage automatique. Il est basé sur le concept d'"ensemble learning", qui consiste à combiner plusieurs classificateurs pour résoudre un problème complexe et améliorer les performances du modèle.

Comme son nom l'indique, "Random Forest est un classificateur qui contient un certain nombre d'arbres de décision sur divers sous-ensembles de l'ensemble de données donné et prend la moyenne pour améliorer la précision prédictive de cet ensemble de données." Au lieu de s'appuyer sur un seul arbre de décision, la forêt aléatoire prend la prédiction de chaque arbre et, sur la base des votes majoritaires des prédictions, elle prédit la sortie finale.

Le plus grand nombre d'arbres dans la forêt permet d'obtenir une plus grande précision et d'éviter le problème du surajustement.[[17]](#footnote-17)

Le diagramme ci-dessous explique le fonctionnement de l'algorithme Random Forest :



Figure 8: le fonctionnement de l'algorithme Random Forest

### Naïve Bayes

L'algorithme de Naïve Bayes est un algorithme d'apprentissage supervisé, basé sur le théorème de Bayes et utilisé pour résoudre les problèmes de classification.

Il est principalement utilisé dans la classification de textes qui comprend un ensemble de données d'entraînement à haute dimension.

Le classificateur Naïve Bayes est l'un des algorithmes de classification les plus simples et les plus efficaces, qui permet de construire des modèles d'apprentissage automatique rapides, capables de faire des prédictions rapides.

Il s'agit d'un classificateur probabiliste, ce qui signifie qu'il prédit sur la base de la probabilité d'un objet.[[18]](#footnote-18)

***Théorème de Bayes :***

Le théorème de Bayes est également connu sous le nom de règle de Bayes ou de loi de Bayes, qui est utilisé pour déterminer la probabilité d'une hypothèse avec des connaissances préalables. Il dépend de la probabilité conditionnelle.

La formule du théorème de Bayes est donnée comme suit :

Naïve Bayes Classifier Algorithm

Algorithme de classification de Naïve Bayes

Où ,

* P(A|B) est la probabilité postérieure : Probabilité de l'hypothèse A sur l'événement observé B.
* P(B|A) est la probabilité de vraisemblance : Probabilité de la preuve étant donné que la probabilité d'une hypothèse est vraie.
* P(A) est Probabilité antérieure : Probabilité de l'hypothèse avant l'observation de la preuve.
* P(B) est la probabilité marginale : Probabilité de l'évidence.

### KNN

K-Nest Neighbour est l'un des algorithmes d'apprentissage automatique les plus simples, basé sur la technique d'apprentissage supervisé.

L'algorithme K-NN suppose la similarité entre le nouveau cas/données et les cas disponibles et place le nouveau cas dans la catégorie qui est la plus similaire aux catégories disponibles.

L'algorithme K-NN stocke toutes les données disponibles et classifie un nouveau point de données sur la base de la similarité. Cela signifie que lorsqu'une nouvelle donnée apparaît, elle peut être facilement classée dans une catégorie bien suivie en utilisant l'algorithme K-NN.

L'algorithme K-NN peut être utilisé aussi bien pour la régression que pour la classification, mais il est surtout utilisé pour les problèmes de classification.

L'algorithme K-NN est un algorithme non-paramétrique, ce qui signifie qu'il ne fait aucune hypothèse sur les données sous-jacentes.

Il est également appelé lazy learner algorithm parce qu'il n'apprend pas immédiatement à partir de l'ensemble d'apprentissage, mais il stocke l'ensemble de données et au moment de la classification, il effectue une action sur l'ensemble de données.

L'algorithme KNN, lors de la phase d'apprentissage, se contente de stocker l'ensemble de données et, lorsqu'il reçoit de nouvelles données, il les classe dans une catégorie qui est très similaire aux nouvelles données.

Le fonctionnement de K-NN peut être expliqué sur la base de l'algorithme ci-dessous :

* Étape 1 : Sélectionner le nombre K de voisins.
* Étape 2 : Calculer la distance euclidienne du nombre K de voisins.
* Étape 3 : Prendre les K voisins les plus proches selon la distance euclidienne calculée.
* Étape 4 : Parmi ces K voisins, compter le nombre de points de données dans chaque catégorie.
* Étape 5 : Attribuez les nouveaux points de données à la catégorie pour laquelle le nombre de voisins est maximal.
* Étape 6 : Notre modèle est prêt.[[19]](#footnote-19)

## Deep learning

Deep learning (également appelé apprentissage structuré profond) fait partie d'une famille plus large de méthodes d'apprentissage automatique basées sur des réseaux neuronaux artificiels avec apprentissage par représentation.

Les algorithmes de deep learning utilisent plusieurs couches pour extraire progressivement des caractéristiques de plus haut niveau de l'entrée initiale. Par exemple, dans le traitement des images, les couches inférieures peuvent identifier les bords, tandis que les couches supérieures peuvent identifier les concepts pertinents pour un être humain, tels que les chiffres, les lettres ou les visages.

### Réseau de neurones artificiels (ANN)

Les réseaux de neurones artificiels (ANN) ont été inspirés par le traitement de l'information et les nœuds de communication distribués dans les systèmes biologiques. Les ANN présentent diverses différences par rapport aux cerveaux biologiques. Plus précisément, les réseaux neuronaux artificiels ont tendance à être statiques et symboliques, alors que le cerveau biologique de la plupart des organismes vivants est dynamique (plastique) et analogique.

Un réseau neuronal artificiel est constitué d'une collection de neurones simulés. Chaque neurone est un nœud qui est connecté à d'autres nœuds par des liens qui correspondent aux connexions biologiques axon-synapse-dendrite. Chaque lien a un poids, qui détermine la force de l'influence d'un nœud sur un autre.

### Artificial neural Architectures

#### MLP: multilayer perceptron neural network

Un perceptron multicouche (MLP) est une classe de réseaux neuronaux artificiels (ANN) entièrement connectés. Le terme MLP est utilisé de manière ambiguë, parfois pour désigner n'importe quel RNA à action directe, parfois pour désigner strictement les réseaux composés de plusieurs couches de perceptrons (avec activation par seuil).

Un MLP est constitué d'au moins trois couches de nœuds : une couche d'entrée, une couche cachée et une couche de sortie. À l'exception des nœuds d'entrée, chaque nœud est un neurone qui utilise une fonction d'activation non linéaire. Le MLP utilise une technique d'apprentissage supervisé appelée rétropropagation pour la formation. Ses couches multiples et son activation non linéaire distinguent le MLP d'un perceptron linéaire. Il peut distinguer des données qui ne sont pas linéairement séparables.[[20]](#footnote-20)

#### PNN: probabilistic neural network

Un réseau neuronal probabiliste (PNN) est un réseau neuronal à anticipation, largement utilisé dans les problèmes de classification et de reconnaissance des formes. Dans l'algorithme PNN, la fonction de distribution de probabilité (PDF) de chaque classe est approximée par une fenêtre de Parzen (une fonction non-paramétrique). Ensuite, à l'aide de la PDF de chaque classe, la probabilité de classe d'une nouvelle donnée d'entrée est estimée et la règle de Bayes est ensuite utilisée pour attribuer la classe avec la probabilité postérieure la plus élevée aux nouvelles données d'entrée. Grâce à cette méthode, la probabilité d'une mauvaise classification est minimisée. Ce type d'ANN a été dérivé du réseau bayésien et d'un algorithme statistique appelé analyse discriminante de Kernel Fisher. Il a été introduit par D.F. Specht en 1966. Dans un réseau PNN, les opérations sont organisées en un réseau multicouche feedforward à quatre couches :

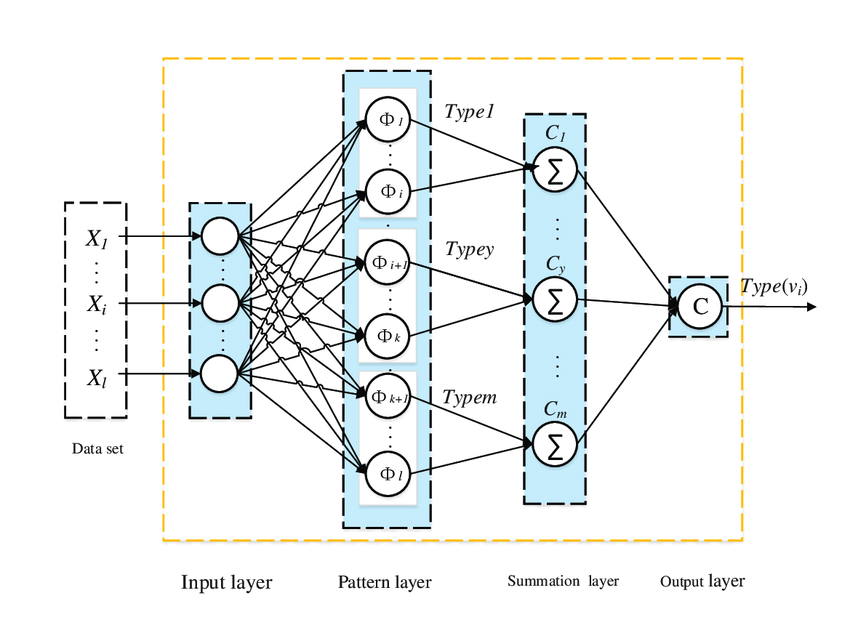


Figure 9:Probabilistic neural network (PNN) structure.

* Couche d'entrée (input layer)

Chaque neurone de la couche d'entrée représente une variable prédictive. Dans le cas de variables catégorielles, N-1 neurones sont utilisés lorsqu'il y a un nombre N de catégories. Il normalise l'étendue des valeurs en soustrayant la médiane et en divisant par l'écart interquartile. Ensuite, les neurones d'entrée transmettent les valeurs à chacun des neurones de la couche cachée.

* Couche de motif (pattern layer)

Cette couche contient un neurone pour chaque cas de l'ensemble de données d'apprentissage. Elle stocke les valeurs des variables prédicteurs pour le cas ainsi que la valeur cible. Un neurone caché calcule la distance euclidienne du cas test par rapport au point central du neurone, puis applique la fonction noyau de la fonction de base radiale en utilisant les valeurs sigma.

* Couche de sommation (summation layer)

Pour le PNN, il existe un neurone de modèle pour chaque catégorie de la variable cible. La catégorie cible réelle de chaque cas d'apprentissage est stockée dans chaque neurone caché ; la valeur pondérée qui sort d'un neurone caché n'est transmise qu'au neurone à motif qui correspond à la catégorie du neurone caché. Les neurones à motif ajoutent les valeurs de la classe qu'ils représentent.

* Couche de sortie (output layer)

La couche de sortie compare les votes pondérés pour chaque catégorie cible accumulés dans la couche des motifs et utilise le vote le plus important pour prédire la catégorie cible.

#### CNN: convolutional neural network

Le CNN est l'un des algorithmes de la branche de l'apprentissage automatique qui se base sur les réseaux neuronaux artificiels (ANN) ou son développement, à savoir le Deep Learning qui est un développement du Perceptron multicouche (MPL) pour traiter des données bidimensionnelles, l'une d'entre elles étant l'image. Le CNN est utilisé dans les données d'image pour détecter et reconnaître des objets dans une image, avec un apprentissage de type Backpropagation. Le fonctionnement du CNN est similaire à celui du MLP, mais dans le CNN, chaque neurone propagé sur le réseau a une forme bidimensionnelle, de sorte que les paramètres de poids et de fonctionnement linéaire du CNN sont différents.

### Fonctions d’activation

#### Sigmoid

La fonction sigmoïde est définie sous la forme mathématique suivante :

Elle prend une valeur réelle en entrée et la transforme en une valeur réelle de sortie comprise entre 0 et 1.



Figure 10. Fonction d'avtivation sigmoide

#### Softmax

La dernière fonction d’activation que nous examinons dans cette partie est la fonction d’activation softmax. Elle se trouve généralement dans la couche de sortie d’un réseau de neurones. Cette fonction est surtout utilisée dans le cas des tâches de multi-class classification. Elle force la sortie du r´eseau de neurones `a représenter la probabilité que l’entrée appartient à un classe définit. Pour obtenir cette probabilité, on utilise cette équation :

Avec i représente l’indice du neurone de sortie en cours de calcul, et j représente les index de tous les neurones du niveau. La variable z désigne le vecteur des neurones de sortie. Il est important de noter que l’activation softmax est calculée différemment des autres fonctions d’activation. Lorsque softmax est la fonction utilisée, la sortie d’un neurone dépend des autres neurones de sortie.

#### ReLU

La fonction d’activation nommée unité de rectification linéaire (ou ReLU) est souvent un peu plus efficace qu’une fonction lisse de type sigmoïde, tout en étant bien plus simple à calculer.

𝑓(𝑥)=𝑚𝑎𝑥(0,𝑥)

Cette fonction d’activation est non saturante et augmente considérablement la convergence du réseau. Mais la fonction ReLU n’est pas parfaite. Si la valeur d’entrée est négative, le neurone reste inactif, ainsi les poids ne sont pas mis `a jour et le réseau n’apprend pas.



Figure 11. unités linéraires réctifiés

### Backpropagation

backpropagation est une méthode utilisée pour ajuster les poids des connexions afin de compenser chaque erreur constatée pendant l'apprentissage. Le montant de l'erreur est effectivement réparti entre les connexions. Techniquement, la rétropropagation calcule le gradient (la dérivée) de la fonction de coût associée à un état donné par rapport aux poids. Les mises à jour des poids peuvent être effectuées par descente de gradient stochastique ou par d'autres méthodes, telles que les machines d'apprentissage extrêmes, les réseaux "No-prop", la formation sans retour en arrière, les réseaux "sans poids" et les réseaux neuronaux non connexionnistes.[[21]](#footnote-21)

## Transfer Learning

L'apprentissage par transfert (TL) est un problème de recherche en apprentissage machine (ML) qui se concentre sur le stockage des connaissances acquises lors de la résolution d'un problème et leur application à un problème différent mais connexe. Par exemple, les connaissances acquises lors de l'apprentissage de la reconnaissance des voitures peuvent être appliquées à la reconnaissance des camions.

### AlexNet

AlexNet est un modèle CNN pré-entraîné avec des images du jeu de données ImageNet. Il peut classifier 1000 espèces différentes. Il se compose de 5 couches convolutionnelles, de trois couches entièrement connectées et d'une couche de classification softmax. Chaque couche convolutive effectue une opération de convolution de base sur les images d'entrée en utilisant un nombre fixe de noyaux de dimensions différentes dans chaque couche. Les couches convolutionnelles de départ sont responsables de l'extraction des caractéristiques de bord et de couleur des images d'entrée. Ces couches génèrent les cartes de caractéristiques des images d'entrée qui leur sont données. La taille des cartes de caractéristiques dépend du nombre et de la taille du filtre appliqué dans une couche convolutive spécifique. La sortie de la dernière couche convolutionnelle est transmise comme entrée à la première couche entièrement connectée. La sortie de la dernière couche entièrement connectée est transmise à la couche softmax pour la classification et la couche softmax calcule les valeurs de probabilité pour chaque espèce. Il y a 4096 neurones dans les deux premières couches convolutionnelles et la dernière couche entièrement connectée est composée de 1000 neurones. La couche ReLU et la couche de mise en commun sont appliquées à ces cartes de caractéristiques pour que toutes les valeurs des caractéristiques restent positives et pour réduire la taille de l'ensemble des caractéristiques respectivement. Ces deux couches jouent un rôle clé entre toutes les deux couches convolutives.

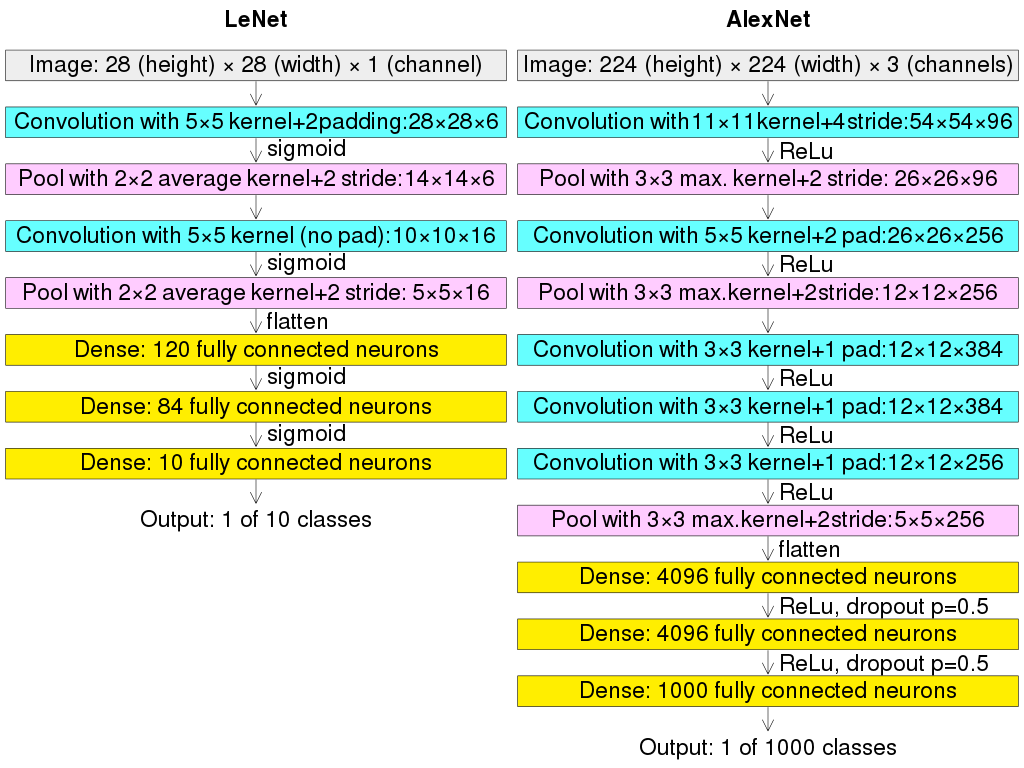


Figure 12. AlexNet architecture

AlexNet est sorti en 2012 et a constitué une avancée révolutionnaire ; il a amélioré les réseaux neuronaux convolutifs (CNN) traditionnels et est devenu l'un des meilleurs modèles de classification d'images... jusqu'à l'arrivée de VGG.

### VGG

Réseaux neuronaux VGG. Alors que les précédents dérivés d'AlexNet se concentraient sur la réduction de la taille des fenêtres et des pas dans la première couche convolutive, VGG aborde un autre aspect très important des CNN : la profondeur. Passons en revue l'architecture de VGG :

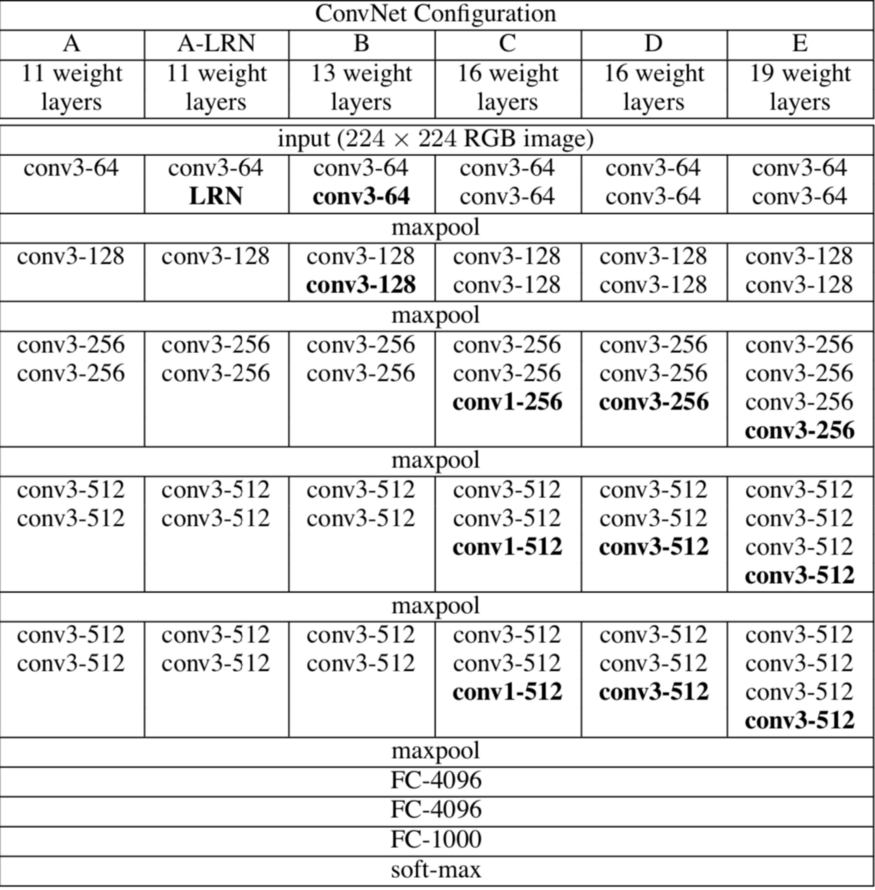


Figure 13. Configurations de VGG

* input VGG prend en entrée une image RVB de 224x224 pixels. Pour la compétition ImageNet, les auteurs ont recadré le patch central de 224x224 dans chaque image afin de conserver la taille de l'image d'entrée.
* Convolutional layers. Les couches convolutionnelles de VGG utilisent un champ réceptif très petit (3x3, la plus petite taille possible qui capture encore la gauche/droite et le haut/bas). Il y a aussi des filtres de convolution 1x1 qui agissent comme une transformation linéaire de l'entrée, qui est suivie par une unité ReLU. Le pas de convolution est fixé à 1 pixel afin que la résolution spatiale soit préservée après la convolution.
* Fully-Connected Layers. VGG a trois couches entièrement connectées : les deux premières ont 4096 canaux chacune et la troisième a 1000 canaux, 1 pour chaque classe.
* Hidden Layers. Toutes les couches cachées de VGG utilisent ReLU (une innovation importante d'AlexNet qui a réduit le temps de formation). VGG n'utilise généralement pas la normalisation locale de la réponse (LRN), car la LRN augmente la consommation de mémoire et le temps d'apprentissage sans amélioration particulière de la précision.

VGG16 est une première architecture de neurones profonds après le succès d'Alexnet. L'équipe de VGG a empilé de nombreuses couches convolutionnelles et pleinement connectées et a obtenu de meilleures performances en utilisant le plus petit filtre initial des filtres convolutionnels 3×3. VGG16 obtient une précision parmi les cinq meilleures sur le jeu de données ImageNet, qui comprend plus de 14 millions d'images de 1 000 classes.

## Préparation du dataset

* Collection des images

Il existe plusieurs méthodes pour collecter des images pour préparer un ensemble de données, l'une de ces méthodes consiste à supprimer les images du Web avec toutes les informations suffisantes sur l'image, nous pouvons également collecter des images en prenant des photos par un téléphone portable, le plus grand nombre d'images prises l'ensemble de données est meilleur. Toutes les images doivent être prises avec la même configuration, et la même extension (par exemple .jpg), après cela, nous stockons les images de la même espèce dans un répertoire et l'étiquetons avec une étiquette ( P0 , P1 ….. Pn).

* Prétraitement des images

Une fois que nous avons ces images capturés, nous pouvons effectuer des modifications telles que le redimensionnement, l'éclairage, la segmentation de l'image, l'amélioration de l'image, la réduction du bruit, les transformations géométriques et la sauvegarde de l'image, ce qui peut être fait avec MATLAB ou GIMP ou tout autre logiciel spécifique pour cela.

* Techniques d’augmentation du dataset

Comme les modèles de réseau de neurones nécessite un grand nombre d'images pour entraînement, la technique d'augmentation peut apporter un effet massif dans cette petite quantité d'images. Ainsi on peut appliquer des techniques d'augmentation telles que le retournement et différents angles de rotation.

# **Chapitre 4** Réalisation et mise en œuvre

Ce chapitre sera consacré à la présentation des aspects technique et les différents outils et méthodes utilisés pour la réalisation de ce projet. La partie de réalisation est l'aboutissement des phases précédentes car c'est dans celle-ci qu'est réalisé le produit du projet pensé.

## Méthodologies de recherche

### Dataset

<http://leafsnap.com/static/dataset/leafsnap-dataset.tar>

### Prétraitement des données

### Approches proposées

#### CNN

#### AlexNet

#### VGG16

### Création du modèle Sequentiel

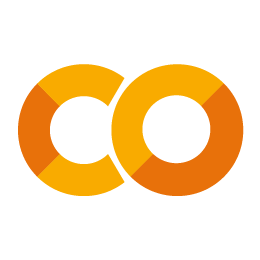
### Training model

### Implémentation

## Résultats et discusion

## Outils de développements

### Environnement de développement



**Colaboratory**, souvent raccourci en "**Colab**", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation.

### Technologies de développement



**Python** est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions



**TensorFlow** est une plate-forme Open Source de bout en bout dédiée au machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie.

**Keras** est une bibliothèque open source écrite en python. La bibliothèque Keras permet d'interagir avec les algorithmes de réseaux de neurones profonds et d'apprentissage automatique, notamment Tensorflow3, Theano, Microsoft Cognitive Toolkit4 ou PlaidML. 

Conçue pour permettre une expérimentation rapide avec les réseaux de neurones profonds, elle se concentre sur son ergonomie, sa modularité et ses capacites d’extension.

## Conclusion

# **Conclusion générale**

# Bibliographie

# Annexe

1. Jahanbakhshi, A., Kheiralipour, K., 2020. Evaluation of image processing technique and discriminant analysis methods in postharvest processing of carrot fruit. Food Science & Nutrition. [↑](#footnote-ref-1)
2. https://en.wikipedia.org/wiki/ImageNet [↑](#footnote-ref-2)
3. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, May 2017. [↑](#footnote-ref-3)
4. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–9, Boston, MA, USA, June 2015. IEEE. [↑](#footnote-ref-4)
5. Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs], April 2015. arXiv: 1409.1556 [↑](#footnote-ref-5)
6. Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, Honolulu, HI, July 2017. IEEE [↑](#footnote-ref-6)
7. Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. FractalNet: Ultra-Deep Neural Networks without Residuals. arXiv:1605.07648 [cs], May 2017. arXiv: 1605.07648. [↑](#footnote-ref-7)
8. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI’17, page 4278–4284. AAAI Press, 2017 [↑](#footnote-ref-8)
9. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. [↑](#footnote-ref-9)
10. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3):211–252, December 2015. [↑](#footnote-ref-10)
11. Francois Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1800–1807, Honolulu, HI, July 2017. IEEE. [↑](#footnote-ref-11)
12. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861 [cs], April 2017. arXiv: 1704.04861. [↑](#footnote-ref-12)
13. Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs], January 2017. arXiv: 1412.6980. [↑](#footnote-ref-13)
14. Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On Early Stopping in Gradient Descent Learning. Constructive Approximation, 26(2):289– 315, August 2007. [↑](#footnote-ref-14)
15. https://en.wikipedia.org/wiki/Machine\_learning [↑](#footnote-ref-15)
16. https://en.wikipedia.org/wiki/Support-vector\_machine [↑](#footnote-ref-16)
17. https://www.javatpoint.com/machine-learning-naive-bayes-classifier [↑](#footnote-ref-17)
18. https://www.javatpoint.com/machine-learning-naive-bayes-classifier [↑](#footnote-ref-18)
19. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning [↑](#footnote-ref-19)
20. https://en.wikipedia.org/wiki/Multilayer\_perceptron [↑](#footnote-ref-20)
21. https://en.wikipedia.org/wiki/Backpropagation [↑](#footnote-ref-21)