

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : de TECHNOLOGIE

Département : Electronique

Domaine : Sciences et Techniques

Filière : Télécommunications

Spécialité : Systèmes des télécommunications

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

**Classification des mauvaises herbes par apprentissage
profond**

Présenté par : *Khiari Rayen et Soudani Aya*

Encadrant : *Harize Saliha*

Grade : *Pr*

Université : *Badji Mokhtar-Annaba*

Jury de Soutenance :

Boukari Karima	MCA	Badji Mokhtar -Annaba	Président
Harize Saliha	Pr	Badji Mokhtar -Annaba	Encadrant
Brik Fatima	MCA	Badji Mokhtar -Annaba	Examineur

Année Universitaire : 2023/2024

Remerciements

Nous tenons à exprimer notre reconnaissance particulière à **Allah**, qui nous a accordé la santé, la force et la patience pour accomplir ce travail. Sans sa grâce et Son inspiration, nous n'aurions pu réaliser quoi que ce soit.

Notre profonde gratitude et respect à notre encadrante , **Mme.Saliha harize** , pour son soutien, pour avoir accepté de bon gré de participer à ce mémoire, ainsi que pour ses efforts fournis, ses conseils utiles, sa gentillesse et ses appréciations.

Nos sincères remerciements au président et membres du jury, Dr **Brik Fatima** et Dr **Boukari**

Karima pour

avoir pris le temps de lire, de juger ce travail et pour leurs remarques pertinentes.

Nous tenons à exprimer notre profonde gratitude envers tous les enseignants du Département d'électronique à l'Université Badji Mokhtar Annaba.

Un merci particulier à nos parents pour nous avoir encouragés à suivre notre voie depuis bien des années. Merci à tous ceux que nous aurions oubliés d'adresser nos remerciements mais qui nous ont aidés dans notre projet de fin d'études.

Dédicaces Aya

Je rends grâce à Dieu Tout-Puissant pour la santé, la volonté, le courage et la patience qu'Il m'a accordés tout au long de ces années d'études.

Ce modeste travail est dédié spécialement

A ma chère maman ma raison de vivre, en témoignage de ma reconnaissance pour sa patience, son amour et ses sacrifices

A mon cher papa pour son amour et son dévouement,

«A vous, mes parents, je dis merci d'avoir fait de moi celui que je suis aujourd'hui.

Aucune dédicace ne pourra exprimer mes respects, mes considérations et ma grande admiration pour vous. Puisse ce travail vous témoigne mon affection et mon profond amour »

*À ma chère sœur **Zaineb Rihanna** et à mon cher frère **Asaad**, pour leur soutien inconditionnel, leur compréhension profonde et leurs encouragements constants. Votre amour m'a soutenu à chaque étape de ce parcours. Sans vous, rien de tout cela n'aurait été possible.*

*À mon binôme **Khiari Rayen**, pour son amitié et pour avoir été un pilier tout au long de ce parcours.*

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagné durant mon chemin d'études supérieures.

Soudani Aya.

Dédicaces Rayen

Louange à Dieu tout puissant, qui m'a permis de voir ce jour tant attendu

Je dédie ce succès d'abord à moi-même, puis à tous ceux qui ont œuvré avec moi pour

Accomplir ce parcours. Vous restez pour moi un soutien indéfectible.

À celui dont je porte le nom avec fierté, qui m'a soutenu sans limite et m'a donné sans rien attendre en retour. à celui qui m'a appris que la vie est un combat et que son arme est le savoir et la connaissance, à celui qui a implanté en moi les nobles valeurs morales. Mon premier soutien dans mon parcours, ma force et mon refuge après Dieu

*À ma fierté et à mon honneur (**Mon père**).*

*Aucun hommage ne pourrait être à la hauteur de tous les sacrifices dont elle a fait preuve, À celle dont les prières sont le secret de mon succès (**Ma mère**).*

*À mon pilier solide, à la sécurité de mes jours, à ceux sur lesquels j'ai appuyé mes épaules et qui sont devenus pour moi des sources d'où je puise (**Mon frère et Ma sœur**).*

*À mon binôme, **Soudani Aya**, pour son amitié, sa motivation et son travail acharné.*

À tous ceux qui ont été un soutien sur ce chemin, je vous dédie cette réalisation et le fruit de mon succès que j'ai tant désiré.

Aujourd'hui, j'ai accompli ses premiers fruits, espérant de Dieu Tout-Puissant qu'Il me rende utile de ce qu'Il m'a enseigné, qu'Il m'enseigne ce que j'ignore et qu'Il fasse de cela une preuve en ma faveur, non contre moi.

Khiari Rayen.

Résumé:

Actuellement, nous découvrons un nouveau concept technologique, l'intelligence artificielle, qui a un grand impact dans de nombreux domaines importants, y compris l'agriculture.

Les mauvaises herbes sont des plantes indésirables qui posent un problème aussi bien pour la qualité de la culture que la productivité.

Dans ce projet, nous avons développé un classificateur basé sur un réseau neuronal convolutif capable de reconnaître et d'identifier le type de mauvaises herbes avec une bonne précision.

Deux modèles CNN, GoogleNet et AlexNet, ont été utilisés pour comparer la précision de la classification. La performance des réseaux a été évaluée en utilisant les métriques de précision, rappel et le F1 score.

Mots clés: Mauvaises herbes, intelligence artificielle, CNN, classification, GoogleNet, AlexNet, Recall, précision, F1 score.

Abstract:

We are currently discovering a new technological concept, artificial intelligence, which is having a major impact in many important fields, including agriculture.

Weeds are undesirable plants that pose a problem for both crop quality and productivity.

In this project, we have developed a classifier based on a convolutional neural network capable of recognizing and identifying the type of weeds with good accuracy.

Two CNN models, GoogleNet and AlexNet, were used to compare classification accuracy.

Network performance was evaluated using the metrics of precision, recall and F1 score.

Keywords: Weeds, artificial intelligence, CNN, classification, GoogleNet, AlexNet, Recall, precision, F1 score.

ملخص:

نحن نكتشف حالياً مفهوماً تكنولوجياً جديداً، وهو الذكاء الاصطناعي، والذي له تأثير كبير في العديد من المجالات المهمة، بما في ذلك الزراعة.

الأعشاب الضارة هي نباتات غير مرغوب فيها تشكل مشكلة لجودة المحاصيل وإنتاجيتها. في هذا المشروع، قمنا بتطوير مصنف يعتمد على شبكة عصبية تلافيفية قادرة على التعرف على نوع الأعشاب الضارة وتحديدتها بدقة جيدة.

واستخدمنا نموذجين لشبكتين من الشبكات العصبية التلافيفية هما AlexNet و GoogleNet لمقارنة دقة التصنيف. تم تقييم أداء الشبكتين باستخدام مقاييس الدقة والاستدعاء ودرجة F1.

الكلمات المفتاحية: الأعشاب الضارة، الذكاء الاصطناعي، الذكاء الاصطناعي، شبكة سي إن إن، التصنيف، شبكة جوجل نت، شبكة أليكس نت، الاستدعاء، الدقة، نتيجة F1.

Liste des symboles

C

CIFAR: Canadian Institute for Advanced Research

CNN: Convolutional Neural Network.

Conv: Convolution

D

DL: deep Learning

F

FC: Fully connected.

I

IA: Intelligence Artificiel.

ILSVRC: ImageNet Large Scale Visual Recognition Challenge.

G

GAFAM: Google, Apple, Face book, Amazon, Microsoft

GP: Graphical Processing Unit

M

ML: Machine Learning

MLP: Multi layer Perception

MNIST: Modified National Institute of Standard and Technology.

R

RNN: Réseaux neuronaux récurrents.

RAG: Réseaux antagonistes génératifs.

R G B: Red Green Blue.

ReLU: Rectified Linear Unit.

Res-Net: Residual Network.

V

VGG: Visual Geometry Group

Liste des figures

Figure II.1 Histoire de l'intelligence artificielle.....	7
Figure II.2 Les domaines d'application de l'IA	9
Figure II.3 IA vs ML vs DL.	9
Figure II.4 Architecture de RNN.....	10
Figure II.5 Schéma du fonctionnement d'un RAG.	11
Figure II.6 Architecture d'un réseau CNN.....	12
Figure II.7 Exemple de filtre de taille (3*3).....	13
Figure II.8 Exemple de fonction ReLU.....	13
Figure II.9 Max pooling, average pooling.....	14
Figure II.10 Le principe de la couche entièrement connectée.....	15
Figure III.1 L'erreur top 5 en % des architectures gagnantes de la compétition ILSVRC.	16
Figure III.2 Architecture du réseau GoogleNet.....	17
Figure III.3 Architecture du réseau Alexnet.....	18
Figure III.4 Architecture du réseau ResNet-50.	19
Figure III.5 Architecture du réseau VGG16.....	20
Figure III.6 La base de données ImageNet.	21
Figure III.7 Quelques images de CIFAR-10.	22
Figure III.8 Un échantillon de MNIST.....	23
Figure IV.1 Etapes d'utilisation d'un réseau pré-entraîné.	25
Figure IV.2 Echantillon de la base de données.	27
Figure IV.3 Installation d'un réseau CNN sur MATLAB.	27
Figure IV.4 Résultat de la recherche d'AlexNet dans Add-Ons.	28
Figure IV.5 Installation d'AlexNet sur MATLAB.....	28
Figure IV.6 Affichage des caractéristiques du réseau installé AlexNet.....	28
Figure IV.7 Affichage des caractéristiques du réseau installé GoogleNet.....	29
Figure IV.8 Code pour chargement de la base de Données(AlexNet).	29
Figure IV.9 Code pour chargement de la base de données(GoogleNet).....	29
Figure IV.10 Affichage de la base de données dans Workspace.	30
Figure IV.11 Code de partition de la base de données.....	30
Figure IV.12 Partition de la base de données dans Workspace.	30
Figure IV.13 Analyse de l'architecture d'un réseau CNN.	31
Figure IV.14 Architecture d'AlexNet.	31

Figure IV.15 Architecture de GoogleNet.....	32
Figure IV.16 Informations sur la couche d'entrée d'AlexNet.	33
Figure IV.17 Informations sur la couche d'entrée de GoogleNet.....	33
Figure IV.18 Code pour redimensionner les données d'entraînement et de test.....	34
Figure IV.19 Extraction des dernières couches.....	34
Figure IV.20 Nombre de catégories dans la base de données.....	34
Figure IV.21 Modification du nom et des paramètres de la couche FutureLearner.....	35
Figure IV.22 Modification du nom de la couche de classification.	35
Figure IV.23 Remplacement des noms des couches du réseau.....	35
Figure IV.24 Code de vérification des modifications.	35
Figure IV.25 Affichage de la nouvelle architecture de GoogleNet.....	36
Figure IV.26 Paramètres d'entraînement.	36
Figure IV.27 Entraînement du réseau AlexNet.....	37
Figure IV.28 Classification des mauvaises herbes par le réseau AlexNet.....	38
Figure IV.29 Entraînement du réseau GoogleNet.....	39
Figure IV.30 Classification des mauvaises herbes par le réseau GoogleNet.....	40
Figure IV.31 Matrice de confusion.	42
Figure IV.32 Matrice de confusion multi-classes pour GoogleNet.	42
Figure IV.33 Performances de réseau GoogleNet par classe.....	44

Liste des tableaux

Tableau I.1 Quelques mauvaises herbes	4
Tableau III.1 Caractéristiques des principaux CNNs.....	20
Tableau IV.1 Différents types d'herbes nuisibles de notre base de données	26
Tableau IV.2 Comparaison entre AlexNet et GoogleNet	41
Tableau IV.3 Résultats TP, FN et FP	43
Tableau IV.4 Performances du réseau GoogleNet par classe	45

Sommaire

<i>Remerciements</i>	I
<i>Dédicaces Aya</i>	II
<i>Dédicaces Rayen</i>	III
<i>Résumé</i>	IV
Liste des symboles	VI
Liste des figures	VIII
Liste des tableaux	X
Introduction Générale	1

Chapitre I Les mauvaises Herbes

I.1 Introduction	3
I.2 Définition des mauvaises herbes	3
I.3 Caractéristiques des mauvaises herbes	3
I.4 Classification des mauvaises herbes	4
I.5 Avantages du deep learning dans la détection des mauvaises herbes	5
I.6 Conclusion	5

Chapitre II

Intelligence artificielle, Deep learning et Réseaux de neurones convolutifs

II.1 Introduction	6
II.2 L'intelligence artificielle	6
II.2.1 Définition	6
II.2.2 Histoire de l'IA	6
II.2.3 Les domaines d'application de l'IA	8
II.3 Le Deep Learning	9
II.3.1 Définition	9
II.3.2 Les principaux modèles de réseaux neuronaux	10
II.3.2.1 Réseaux neuronaux récurrents(RNN)	10
II.3.2.2 Réseaux antagonistes(RAG)	10
II.3.2.3 Réseaux de neurones convolutifs(CNNs)	10
II.3.2.4 Types d'Apprentissage	10

II.4 Architecture Globale des CNNs.....	12
II.4.1 La couche d'entrée.....	12
II.4.2 La couche de convolution.....	12
II.4.3 La couche Rectified Linear Unit (Relu).....	13
II.4.4 La couche Pooling.....	14
II.4.5 La couche entièrement connectée (fullyConnected).....	14
II.4.6 La couche Softmax/logistique.....	15
II.4.7 La couche de sortie (Output layer).....	15
II.5 Conclusion.....	15

Chapitre III

Les modèles des CNNs et les bases de données

III.1 Introduction.....	16
III.2 Les architectures des CNNs.....	16
III.2.1 GoogleNet.....	16
III.2.2 AlexNet.....	17
III.2.3 ResNet-50.....	18
III.2.4 VGG16.....	19
III.3 Les bases de données.....	21
III.3.1 ImageNet.....	21
III.3.2 CIFAR-10.....	22
III.3.3 MNIST.....	22
III.4 Conclusion.....	23

Chapitre IV

Simulations, résultats et discussions

IV.1 Introduction.....	24
IV.2 Langage de programmation.....	24
IV.3 Les différentes méthodes d'entraînement d'un CNN.....	25
IV.4 Méthodologie.....	26
IV.4.1 Création d'une base de données.....	26
IV.4.2 Installation du réseau sur MATLAB.....	27
IV.4.3 Chargement de la base de Données.....	29
IV.4.4 Partition de la base de données.....	30

IV.4.5 Analyse des réseaux AlexNet et GoogleNet.....	30
IV.4.6 Adaptation des couches	34
IV.4.7 Options d'entraînement	36
IV.4.8 Entraînement du réseau AlexNet	37
IV.4.9 Classification des mauvaises herbes par AlexNet	38
IV.4.10 Entraînement du réseau GoogleNet	39
IV.4.11 Classification des mauvaises herbes par GoogleNet	40
IV.6 Mesure des performances du réseau GoogleNet	41
IV.6.1 La matrice de confusion	41
IV.6.2 Définitions de la précision, du rappel (recall) et du F1 score.....	44
IV.7 Conclusion.....	46
Conclusion Générale.....	47
Références Bibliographiques	49

Introduction Générale

Introduction Générale:

En Algérie la Surface Agricole Totale (S.A.T) est de 42,4 millions d'hectares, représentant 18 % de la surface totale du pays. La Surface Agricole Utile ou S.A.U est de 8,458 millions d'hectares, représentant 20 % de la S.A.T. L'agriculture intervient pour environ 12 % dans le PIB. Le secteur fait vivre de façon directe et indirecte 21 % de la population nationale [1].

Les mauvaises herbes sont des plantes qui concurrencent la culture et peuvent réduire son rendement. Les adventices peuvent aussi constituer des foyers primaires pour certaines maladies et ravageurs. Leur contrôle est un facteur déterminant dans la production, mais aussi permet de limiter l'évolution du stock semencier dans le sol. La gestion des mauvaises herbes dans la culture du blé est un des principaux facteurs d'intensification de cette culture en Algérie et en Afrique du Nord. Les pertes occasionnées par ce fléau sur blé sont importantes : 25 à 50% de la production potentielle de la culture [2]. Par exemple, Les mauvaises herbes concurrencent le blé durant tout son cycle évolutif mais elles sont plus dangereuses durant la période juvénile de la culture (levée-tallage). La concurrence se fait pour l'espace, l'eau, les éléments nutritifs et la lumière.

La lutte chimique contre les mauvaises herbes est inévitable, mais le timing de l'intervention revêt une importance capitale. La quantité d'herbicides à utiliser dépend du degré d'infestation.

Au cours des dernières années, les systèmes d'intelligence artificielle ont connu une accélération de leur développement. D'importantes avancées ont été réalisées dans le domaine de la classification d'images. À la fin des années 80, Yann LeCun a développé un type particulier de réseau appelé "réseaux de neurones convolutionnels". Ces réseaux comprennent trois types principaux de couches: la couche convolutive, la couche de pooling et la couche entièrement connectée (Fully connected).

La raison de l'utilisation du deep learning pour la détection des mauvaises herbes est sa capacité à traiter des données complexes, à apprendre à partir d'exemples et à offrir une précision élevée dans des environnements variables. Cela contribue à une gestion plus efficace et durable des cultures agricoles.

C'est dans ce contexte que s'inscrit notre projet. Notre objectif est d'utiliser les réseaux de neurones convolutionnels (CNN) pour identifier des mauvaises herbes. Nous allons créer une base de données d'images de différents types de mauvaises herbes et l'utiliser dans l'entraînement de deux CNNs : AlexNet et GoogleNet. Nous allons commencer avec

AlexNet, qui est un réseau assez simple et évaluer sa performance à identifier et classer les mauvaises herbes. Si sa précision est jugée faible, nous utiliserons GoogleNet.

Notre mémoire est organisé en quatre chapitres:

- Dans le premier chapitre, nous allons définir les mauvaises herbes, présenter leurs caractéristiques et exposer les principales raisons d'utiliser les CNNs pour la détection des mauvaises herbes.
- Le deuxième chapitre présente des généralités et des notions de base sur l'intelligence artificielle « AI », le deep learning et les réseaux de neurones convolutifs.
- Le troisième chapitre est consacré à la description des architectures des réseaux CNNs (GoogleNet, ResNet , VGG ...) et les bases de données les plus populaires.
- Le quatrième chapitre expose en détails la partie expérimentale de notre travail où les simulations, les résultats et leurs analyses seront présentés.
- Le mémoire sera terminé par une conclusion générale.

Chapitre I

Les mauvaises

Herbes



I.1 Introduction:

Les mauvaises herbes représentent un défi majeur en agriculture, car elles entravent la croissance des cultures, diminuent la productivité et peuvent entraîner des pertes financières significatives.

Dans ce chapitre, nous allons examiner les caractéristiques et la classification des mauvaises herbes, ainsi que les avantages de l'apprentissage profond dans la détection de ces dernières.

I.2 Définition des mauvaises herbes:

Les mauvaises herbes se caractérisent souvent par leur croissance rapide, produisant une biomasse importante en peu de temps, une grande fécondité, un pouvoir germinatif élevé et une grande plasticité écologique. Elles font preuve de résistance à la sécheresse ainsi qu'à l'excès d'humidité [3].

Les mauvaises herbes impactent d'autres plantes et rivalisent pour des ressources vitales telles que l'eau, les nutriments et la lumière, ce qui affecte négativement la productivité agricole et la qualité des cultures. Parfois, les mauvaises herbes présentent un risque pour les animaux et les humains, pouvant causer des allergies, des empoisonnements et la transmission de maladies. Par conséquent, il est crucial de limiter la propagation des mauvaises herbes afin de préserver l'environnement naturel et de garantir la productivité agricole.

I.3 Caractéristiques des mauvaises herbes:

Les caractéristiques des mauvaises herbes peuvent être extraites en observant plusieurs facteurs:

- a. Aspect extérieur :** Les mauvaises herbes présentent une grande variété de formes et peuvent être identifiées en examinant leurs feuilles, leurs fleurs, leurs fruits et d'autres caractéristiques.
- b. Modèle de croissance :** Les mauvaises herbes se caractérisent par leur croissance et leur propagation rapides et étendues, certaines espèces se propageant par leurs graines.
- c. Couleur et odeur :** Les mauvaises herbes se distinguent par leurs couleurs ou leurs odeurs souvent différentes de celles des plantes naturelles. Elles peuvent dégager une odeur forte ou désagréable, ce qui facilite leur identification.
- d. Dommages causés :** Les mauvaises herbes peuvent être repérées en observant les dégâts qu'elles occasionnent aux autres plantes et aux champs agricoles. Il est possible

de les identifier en remarquant les symptômes qu'elles laissent sur les feuilles, les fruits, les racines et autres parties des plantes.

I.4 Classification des mauvaises herbes:

En Algérie, l'agriculture constitue un secteur économique vital. L'agriculture se compose d'une variété de cultures telles que les céréales, les fruits, les légumes, en particulier la culture des agrumes qui occupe une place importante dans l'économie agricole du pays. L'agriculture en Algérie est confrontée à plusieurs défis, notamment les mauvaises herbes. Parmi ces mauvaises herbes:

- **Pâquerette:** est une espèce de plante à fleurs de la famille des Astéracées [4].
- **Trèfle blanc:** est une espèce de plantes dicotylédones de la famille des Fabaceae originaire des régions tempérées de l'Ancien Monde. il est considéré comme une adventice dans certaines terres cultivées (notamment les vergers irrigués) et des jardins. C'est une plante très basse à tige rampante, reconnaissable à ses inflorescences blanches, parfois un peu rosées, au parfum assez agréable [5].
- **Lierre terrestre:** lierre terrestre est une espèce de plantes herbacées de la famille des lamiacées (famille des menthes notamment). Plante couvre-sol, elle est parfois appelée courroie de saint Jean ou courroie de terre [6].

Le tableau I.1 montre quelques herbes mauvaises de notre base de données.

Tableau I.1 Quelques mauvaises herbes.



Pâquerette.



Trèfle blanc.



Lierre terrestre.

I.5 Avantages du deep learning dans la détection des mauvaises herbes:

Le deep learning est souvent utilisé pour la détection des mauvaises herbes en raison de sa capacité à traiter de grandes quantités de données complexes, notamment des images. Nous pouvons citer les raisons suivantes:

1. **Traitement d'images complexe:** Le deep learning excelle dans le traitement des images complexes en extrayant automatiquement des caractéristiques pertinentes à partir des données. Cela permet de détecter les mauvaises herbes même dans des environnements variés et difficiles.
2. **Capacité d'apprentissage:** Les modèles de deep learning peuvent apprendre à reconnaître différents types de mauvaises herbes à partir d'images exemples, ce qui est très utile dans le domaine de l'agriculture pour différencier les mauvaises herbes des cultures.
3. **Adaptabilité:** Une fois entraînés, les modèles de deep learning peuvent s'adapter à de nouveaux types de mauvaises herbes ou à des variations dans les conditions d'éclairage et d'environnement.
4. **Précision:** Les modèles de deep learning peuvent offrir une précision élevée dans la détection des mauvaises herbes, ce qui permet une gestion plus efficace des cultures en réduisant la concurrence des mauvaises herbes.
5. **Automatisation:** En intégrant des modèles de deep learning dans des systèmes autonomes ou des outils agricoles, il est possible d'automatiser la détection et le contrôle des mauvaises herbes, ce qui peut réduire les coûts de main-d'œuvre et l'utilisation d'herbicides.

I.6 Conclusion:

Dans ce chapitre, nous avons présenté les caractéristiques de certaines mauvaises herbes retrouvées en Algérie. Nous avons aussi expliqué les raisons d'utiliser le deep learning pour la détection des mauvaises herbes.

Chapitre II

Intelligence artificielle, Deep learning et Réseaux de neurones convolutifs



II.1 Introduction:

L'intelligence artificielle (IA) a connu une expansion significative ces dernières années, infiltrant de plus en plus la vie quotidienne des utilisateurs d'appareils numériques. Pratiquement tous les domaines sont désormais touchés par cette avancée technologique.

Une branche avancée de l'IA, le deep learning, s'inspire du fonctionnement du cerveau humain pour permettre aux machines d'apprendre à partir de données. Les réseaux de neurones artificiels, en particulier, jouent un rôle crucial dans le deep learning.

Les réseaux convolutifs, également connus sous le nom de CNN, représentent une architecture spécifique de ces réseaux de neurones, principalement utilisée dans le domaine de la vision par ordinateur. Ces réseaux excellent dans le traitement de données organisées en grille, telles que les images, en utilisant des opérations de convolution pour extraire des caractéristiques et faciliter une reconnaissance précise.

Ce chapitre est alors consacré à la définition et la présentation de l'IA, le deep learning et les réseaux, en particulier, les CNNs.

II.2 L'intelligence artificielle:

II.2.1 Définition:

L'intelligence artificielle n'est pas une démarche aisée. Depuis ses débuts en tant que domaine de recherche spécifique au milieu du XXe siècle, elle a toujours constitué une frontière en constante redéfinition. L'intelligence artificielle, loin d'être un simple domaine de recherche bien défini, se présente plutôt comme un programme ambitieux: comprendre le fonctionnement de la cognition humaine et le reproduire; créer des processus cognitifs comparables à ceux de l'être humain [7].

II.2.2 Histoire de l'IA:

L'histoire de l'intelligence artificielle (figure II.1) peut être segmentée en plusieurs étapes, chacune caractérisée par des avancées majeures. Voici une synthèse succincte de ces étapes, accompagnée d'une brève description de chacune:

- Origines et Fondations (1930-1950) : Les bases de l'IA ont émergé dans les années 1930 avec les progrès en neurologie, montrant que les neurones du cerveau fonctionnent comme des circuits électriques. Ces découvertes ont conduit à la

cybernétique et à la théorie de l'information. En 1950, Alan Turing a posé les fondations en suggérant que les machines pouvaient reproduire l'intelligence humaine.

- Naissance de l'IA (1950-1956) : En 1956, l'IA est officiellement reconnue comme une discipline académique lors d'une conférence à Dartmouth College, dirigée par John McCarthy. Cette période marque le début des recherches systématiques dans le domaine.
- L'âge d'or de l'IA se manifeste en 1975, avec un développement rapide et des optimistes prévisions quant à la rivalité future entre les ordinateurs et l'intelligence humaine. Cependant, des résultats limités génèrent une première période d'hibernation, marquée par la stagnation des recherches et du financement.
- Le boom des années 1980 voit l'émergence des systèmes experts, populaires auprès des entreprises, attirant l'attention des chercheurs. La représentation des connaissances devient un axe majeur de recherche. Une deuxième période d'hibernation survient avec l'arrivée des ordinateurs personnels, freinant le développement du matériel nécessaire à l'IA.
- Ère de l'Apprentissage Profond et du Big Data (2011 - Présent) : Les dernières décennies ont vu l'émergence de l'apprentissage profond, alimenté par des ordinateurs puissants et des avancées dans le Big Data. Cela a ouvert la voie au développement de systèmes d'intelligence générale capables de résoudre une variété de problèmes, s'approchant davantage de l'intelligence humaine [8].

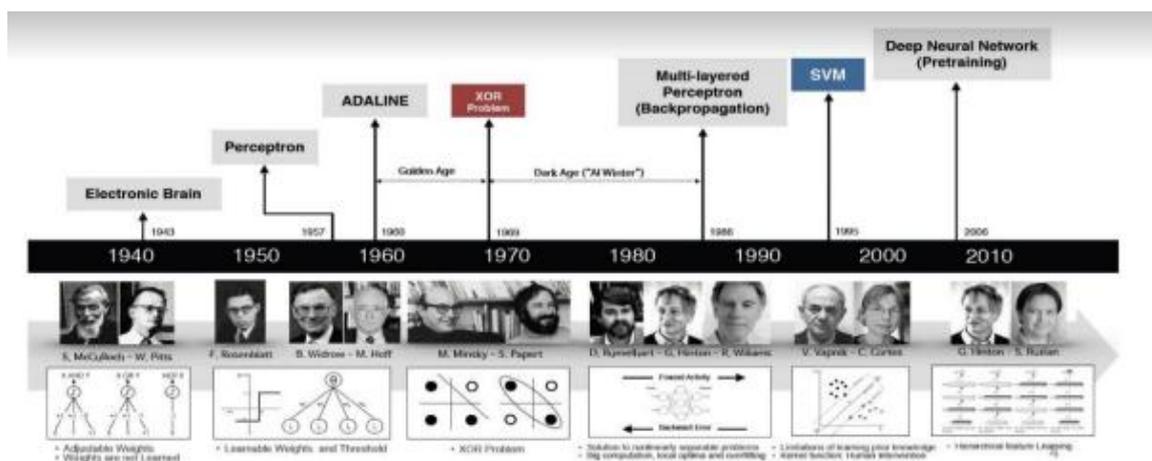


Figure I.1 Histoire de l'intelligence artificielle.

II.2.3 Les domaines d'application de l'IA:

Les domaines d'application de l'intelligence artificielle (IA) (figure II.2) englobent diverses disciplines, chacune exploitant des techniques spécifiques pour résoudre des problèmes particuliers. Parmi les principaux domaines d'application de l'IA, on trouve:

- a) Apprentissage automatique (Machine Learning) : Ce domaine de l'IA se concentre sur le développement de modèles et d'algorithmes capables d'apprendre à partir de données, permettant aux systèmes de prendre des décisions et d'effectuer des tâches sans être explicitement programmés.
- b) Systèmes experts : Les systèmes experts utilisent des règles et des bases de connaissances pour imiter le raisonnement humain dans un domaine spécifique. Ils sont souvent employés pour résoudre des problèmes complexes et prendre des décisions expertes.
- c) Traitement du langage naturel (Natural Language Processing - NLP) : Le NLP permet aux machines de comprendre, interpréter et générer un langage humain. Il est utilisé dans les applications de traitement de texte, la traduction automatique, les chatbots, et d'autres interfaces de communication homme-machine.
- d) Reconnaissance vocale (Speech Recognition) : Ce domaine se concentre sur la capacité des machines à interpréter et comprendre la parole humaine. Les applications incluent la commande vocale, la transcription automatique et les assistants vocaux.
- e) Planification et optimisation : L'IA est utilisée pour élaborer des plans et optimiser des processus dans divers contextes, tels que la logistique, la gestion de la chaîne d'approvisionnement, et la planification des ressources.
- f) Vision par ordinateur (Vision) : La vision par ordinateur permet aux machines de comprendre et d'interpréter des informations visuelles à partir d'images ou de vidéos. Elle est utilisée dans la reconnaissance d'objets, la détection de mouvement, et la réalité augmentée, entre autres applications.
- g) Robotique : L'IA joue un rôle crucial dans la robotique en permettant aux robots d'apprendre des tâches complexes, de s'adapter à leur environnement et d'interagir de manière intelligente avec les humains.

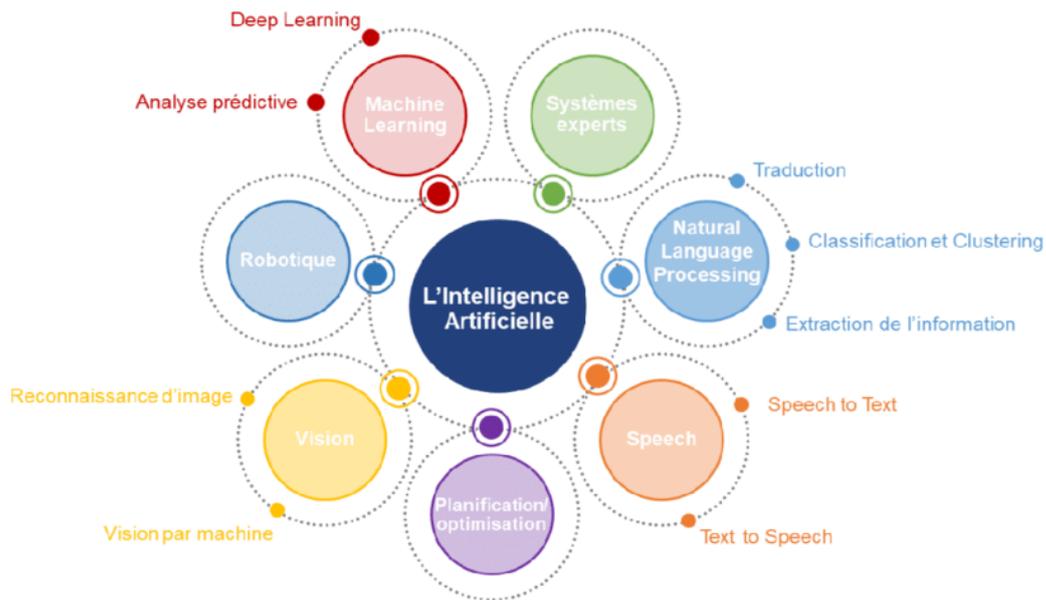


Figure II.2 Les domaines d'application de l'IA.

II.3 Le Deep Learning:

II.3.1 Définition:

Le deep learning (figure II.3) constitue une catégorie d'algorithmes d'apprentissage automatique qui exploite plusieurs couches pour extraire de manière progressive des caractéristiques de plus en plus abstraites à partir des données brutes en entrée. Dans le domaine du traitement d'images, par exemple, les couches inférieures peuvent détecter les contours, tandis que les couches supérieures peuvent identifier des concepts plus complexes, tels que des chiffres, des lettres ou des visages, qui sont pertinents pour un être humain [9].

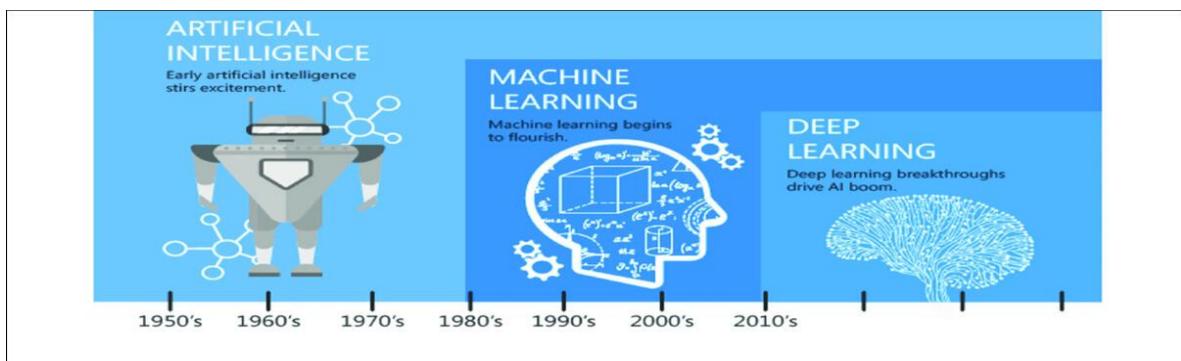


Figure II.3 IA vs ML vs DL.

II.3.2 Les principaux modèles de réseaux neuronaux:

II.3.2.1 Réseaux neuronaux récurrents (RNN):

Les réseaux de neurones récurrents (RNN) (figure II.4) représentent une catégorie importante de réseaux de neurones fréquemment employée dans le domaine de l'apprentissage en profondeur (Deep Learning). Ces RNN intègrent les sorties antérieures en tant qu'entrées supplémentaires [10].

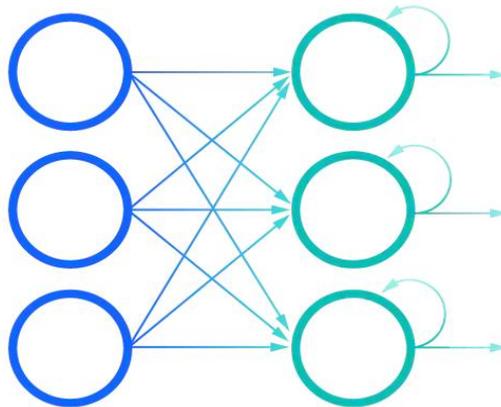


Figure II.4 Architecture de RNN.

II.3.2.2 Réseaux antagonistes génératifs (RAG):

Le réseau antagoniste génératif (RAG) (figure II.5) constitue un algorithme d'apprentissage profond non supervisé, orienté vers la génération d'images synthétiques hautement réalistes. Dans ce modèle, deux réseaux neuronaux profonds s'affrontent dans un contexte de théorie des jeux. D'une part, le générateur produit des échantillons, tels que des images, tandis que d'autre part, le discriminateur cherche à discerner si l'échantillon est authentique ou généré par le générateur. Les paramètres des deux réseaux sont ajustés en fonction des prédictions du discriminateur, permettant au générateur de produire des images de plus en plus convaincantes. Cette approche novatrice a été développée par Ian Goodfellow en 2014 lors de ses études au laboratoire MILA de l'Université de Montréal [11].

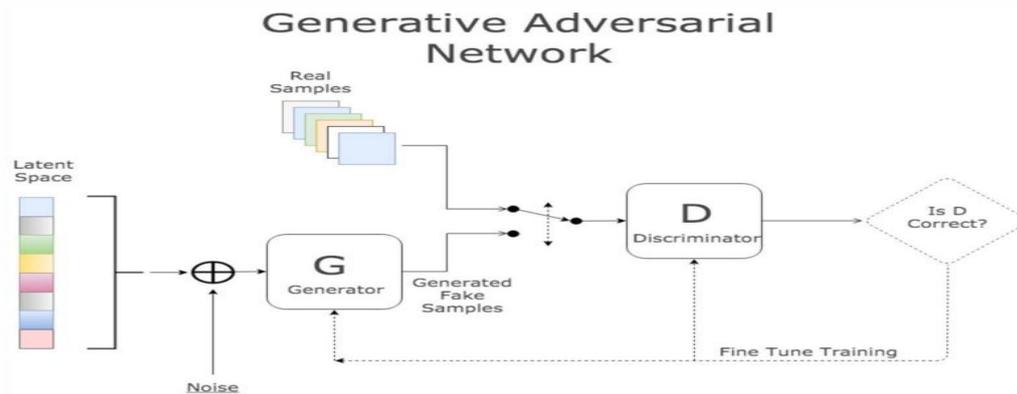


Figure II.5 Schéma du fonctionnement d'un RAG.

II.3.2.3 Réseaux de neurones convolutifs (CNNs):

Les réseaux neuronaux convolutifs représentent une catégorie de réseaux de neurones artificiels sans cycles, caractérisés par un schéma de connexion inspiré du cortex visuel des animaux. Les neurones de cette région cérébrale sont disposés de manière à correspondre à des zones se chevauchant lors du pavage du champ visuel. Ces réseaux tirent leur fonctionnement des processus biologiques et se composent d'une superposition multicouche de perceptrons, ayant pour objectif de prétraiter de petites quantités d'informations. Les applications des réseaux neuronaux convolutifs sont vastes et incluent la reconnaissance d'images et de vidéos, les systèmes de recommandation, ainsi que le traitement du langage naturel [12].

II.3.2.4 Types d'Apprentissage:

Apprentissage supervisé: Les données d'entraînement sont associées à des étiquettes ou des catégories fournies par des humains. L'algorithme apprend à faire correspondre correctement les entrées aux sorties en utilisant ces étiquettes. Par exemple, dans la classification d'images, chaque image est accompagnée d'une étiquette indiquant sa catégorie.

Apprentissage non supervisé: Les données d'entraînement ne sont pas étiquetées. L'algorithme cherche à découvrir des structures ou des modèles intrinsèques dans les données sans référence externe. Par exemple, dans le clustering, l'algorithme regroupe les données similaires sans connaître à l'avance les catégories.

II.4 Architecture Globale des CNNs:

Une architecture de réseau de neurone convolutif se compose d'une séquence de couches de traitement, chacune spécialisée dans l'apprentissage de différentes caractéristiques présentes dans une image (figure II.6) [13].

- La couche d'entrée (Input layer).
- La couche de convolution.
- La couche Rectified Liner Unit (Relu).
- La couche pooling (Pool).
- La couche entièrement connectée (FC).
- La couche Softmax /logistique.
- La couche de sortie (Output layer).

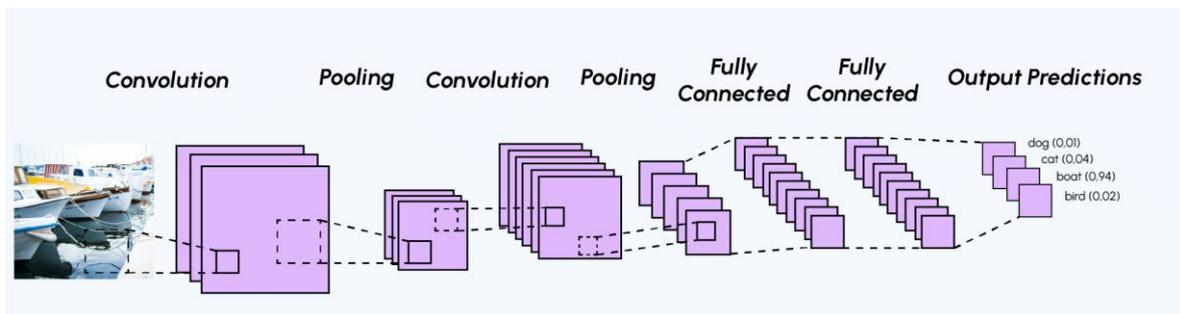


Figure II.6 Architecture d'un réseau CNN.

II.4.1 La couche d'entrée:

La couche d'entrée d'un CNN contient des données d'image. Si elle est en couleur (R.G.B) Les données d'image sont représentées par une matrice tridimensionnelle et si elles sont en noir et blanc, les données d'image sont représentées par une matrice d'une certaine taille [14].

II.4.2 La couche de convolution:

Le terme "convolution" est utilisé car les réseaux CNNs appliquent des opérations de convolution similaires à celles utilisées dans le traitement du signal ou des images. Dans la couche de convolution, un produit de convolution est appliqué entre l'image originale ou celle obtenue dans la couche précédente et un filtre dont les coefficients sont considérés comme des poids synaptiques [15].

La figure II .7 montre un exemple de filtre de taille (3*3).

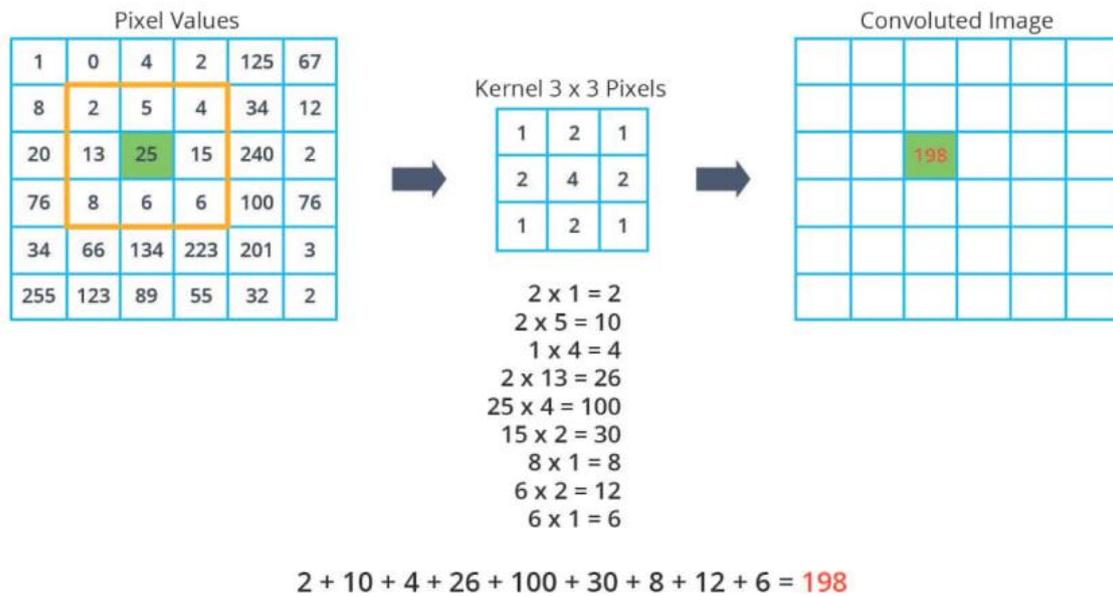


Figure II.7 Exemple de filtre de taille (3*3).

II.4.3 La couche Rectified Linear Unit (Relu):

ReLU (Rectified Linear Units) est une fonction non linéaire définie par $\text{Relu}(x) = \max(0, x)$. La couche ReLU remplace donc toutes les valeurs négatives en entrée par des zéros. Elle agit en tant que fonction d'activation (figure II.8) [16].

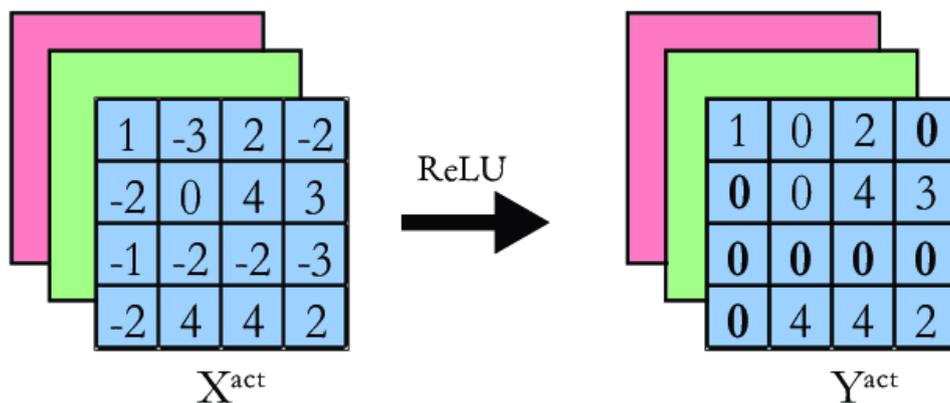


Figure II.8 Exemple de fonction ReLU.

II.4.4 La couche Pooling:

Le pooling est une technique cruciale dans les CNNs. Elle consiste à diviser l'image d'entrée en tuiles de pixels non superposées, puis à générer une sortie basée sur les valeurs de ces pixels. Cela réduit la taille spatiale de l'image intermédiaire, ainsi que le nombre de paramètres et de calculs requis dans le réseau. Le pooling est souvent inséré entre les couches convolutives pour prévenir le sur apprentissage et instaurer une invariance par translation [17].

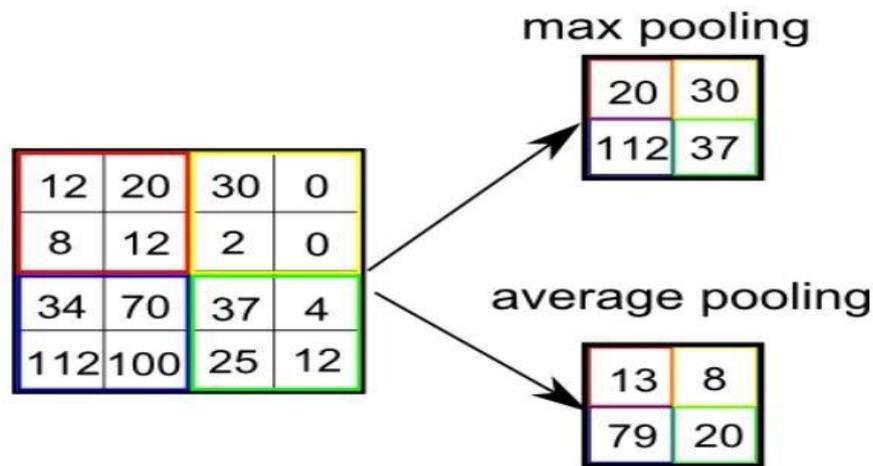


Figure II.9 Max pooling, average pooling.

II.4.5 La couche entièrement connectée (fullyConnected):

Elle prend en entrée un vecteur contenant les pixels aplatis de toutes les images qui ont été filtrées, corrigées et réduites par le pooling.

Dans le contexte de la classification d'images, le traitement des images, y compris l'extraction des caractéristiques et la segmentation en séquences d'entrée d'images RVB, est effectué automatiquement à travers les couches de convolution et de pooling des modèles.

L'application de la fonction softmax aux sorties de la couche entièrement connectée donne la probabilité d'appartenance à une classe [18].

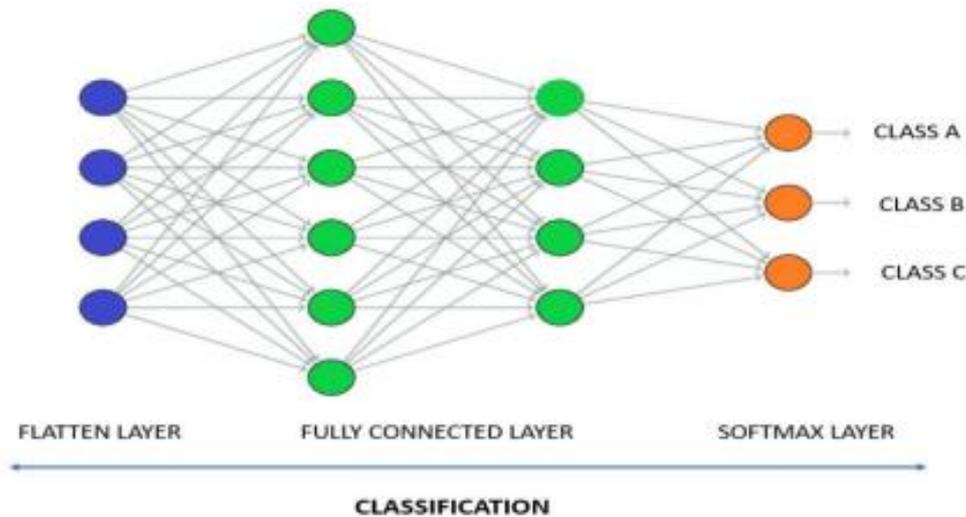


Figure II.10 Le principe de la couche entièrement connectée.

II.4.6 La couche Softmax/logistique:

La fonction Softmax est couramment employée dans la dernière couche d'un modèle de classification. Son rôle est de transformer un vecteur de valeurs réelles en un vecteur de probabilités [19].

II.4.7 La couche de sortie (Output layer):

La couche de sortie représente les prédictions finales du modèle, généralement sous la forme d'un encodage ou d'une représentation des étiquettes associées aux données en entrée.

II.5 Conclusion:

Dans ce premier chapitre, nous avons abordé de manière brève le domaine de l'intelligence artificielle, en présentant ses différentes catégories et ses divers champs d'application. En résumé, le domaine du deep learning se positionne comme crucial et prometteur dans notre quotidien, contribuant à résoudre de nombreuses problématiques difficiles et complexes auxquelles les êtres humains font face. Parmi ces applications, nous allons examiner le secteur agricole, plus précisément le traitement des mauvaises herbes, que ce soit en termes de détection ou de classification, en utilisant les techniques de deep learning.

Chapitre III

Les modèles des CNNs et les bases de données



III.Introduction:

Lorsqu'on parle de CNN, il est évident qu'une architecture spécifique et un ensemble de données adapté sont indispensables. Pour prendre une décision éclairée concernant le choix d'un réseau, il est primordial de bien appréhender sa structure architecturale.

Ce chapitre se concentre sur la présentation détaillée des architectures les plus renommées de CNN. De plus, nous discuterons des ensembles de données utilisés pour l'entraînement de ces réseaux.

III.2 Les architectures des CNNs:

L'histoire du développement et du succès des CNNs a débuté en 1998 avec la publication par Yann LeCun et ses co-auteurs du papier intitulé « Gradient-Based Learning Applied to Document Recognition » (Lecun et al. 1998). Vingt ans après cette avancée initiale, les CNNs ont démontré leur progression significative [20]. Plusieurs architectures ont été lancées à cette époque. Nous présenterons ici les architectures de classification des images les plus populaires et les plus efficaces.

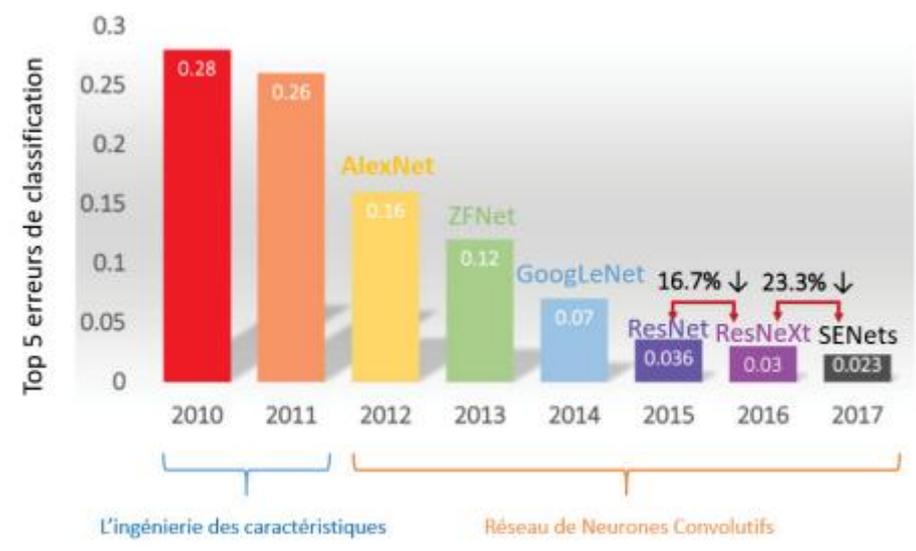


Figure III.1 L'erreur top 5 en % des architectures gagnantes de la compétition ILSVRC.

III.2.1 GoogleNet:

GoogleNet, aussi appelé Inception V1, représente une architecture de réseau neuronal convolutif (CNN) mise au point par Christian Szegedy et son équipe chez Google en 2014. Sa conception vise spécifiquement à classer les images provenant de l'ensemble de données

ImageNet, qui contient plus d'un million d'images réparties en 1000 catégories. GoogleNet a surpassé de manière significative les performances des architectures CNNs précédentes et a remporté haut la main le défi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2014. Cette architecture, composée de 22 couches de poids, a grandement réduit la marge d'erreur par rapport aux éditions précédentes du concours et par rapport à ses concurrents, en grande partie grâce à l'utilisation de sous-réseaux appelés "modules Inception", qui permettent une exploration plus approfondie du réseau. GoogleNet est particulièrement célèbre pour ses modules de démarrage, conçus pour réduire le nombre de paramètres et les coûts de calcul du réseau. En outre, il a introduit l'utilisation de classificateurs auxiliaires, des classificateurs supplémentaires formés en parallèle avec le classificateur principal afin d'améliorer les performances globales du système [21].

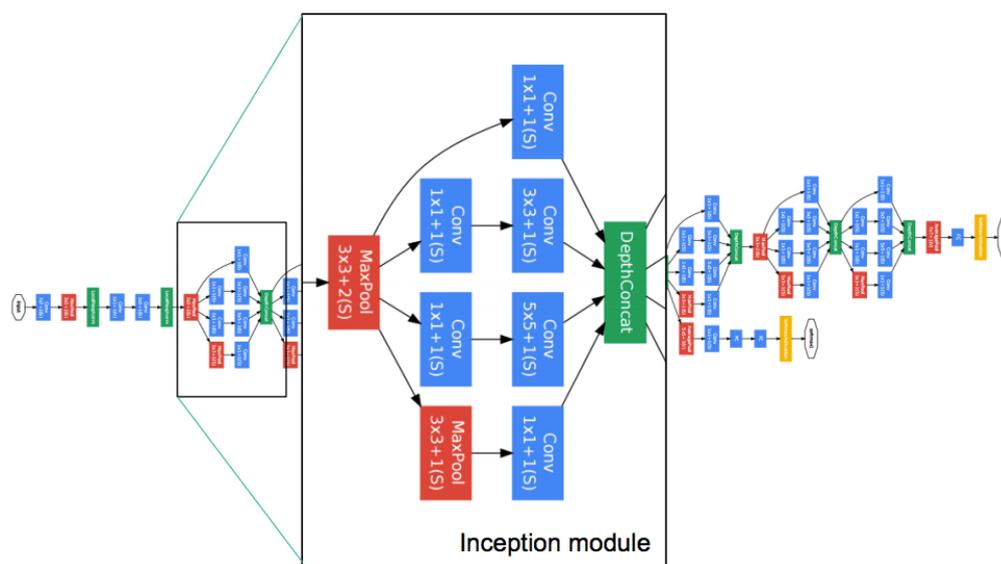


Figure III.2 Architecture du réseau GoogleNet.

III.2.2 AlexNet:

AlexNet, un réseau neuronal convolutif initialement conçu avec CUDA pour bénéficier du support GPU, a pris part au défi de reconnaissance visuelle à grande échelle ImageNet en 2012. En surpassant de loin ses prédécesseurs, AlexNet a remporté le défi en réduisant l'erreur de top 5 de 26% à seulement 15.3%. Cette architecture novatrice a été développée par le groupe Supervision, sous la direction d'Alex Krizhevsky, et a profondément marqué le domaine de l'apprentissage automatique, en particulier dans l'application de l'apprentissage en profondeur à la vision artificielle. Avec ses huit couches, dont cinq couches convolutionnelles

suivies de trois couches entièrement connectées, AlexNet a été formé simultanément sur deux GPU Nvidia GeForce GTX 580 pendant une période de 6 jours, ce qui explique sa division en deux pipelines. En utilisant la fonction d'activation ReLU non saturante, AlexNet a démontré une amélioration significative des performances d'entraînement par rapport aux fonctions Tanh et sigmoïde [22].

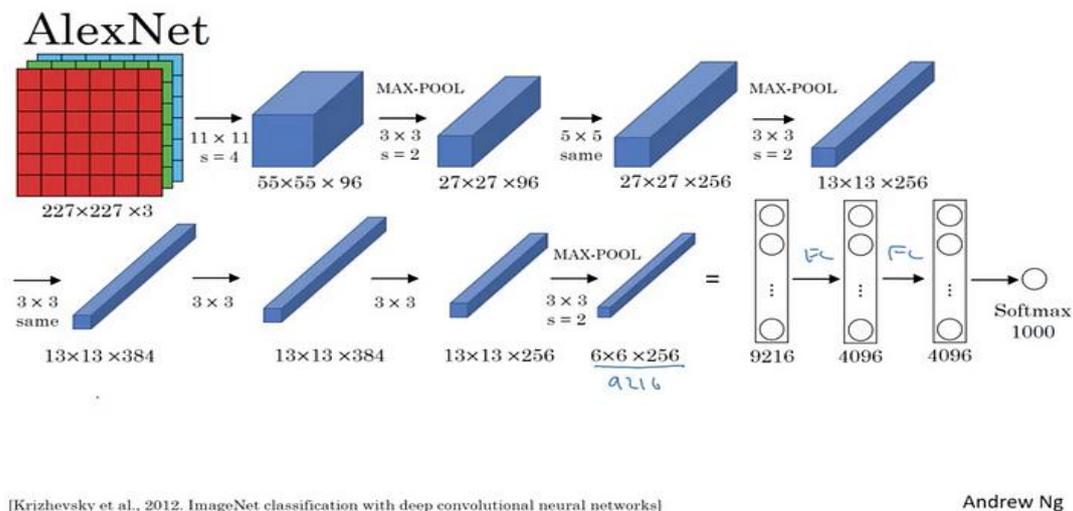


Figure III.3 Architecture du réseau AlexNet.

III.2.3 ResNet-50:

La Réseau Résiduel (Residual Network) développé par Kaiming He (et d'autres) a remporté le défi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2015 [23].

ResNet-50 est une architecture de réseau neuronal profond composée de plusieurs blocs résiduels. Ces blocs contiennent des couches de convolution, des couches de normalisation par lots et des connexions de raccourci. Les connexions de raccourci facilitent le flux des gradients pendant l'entraînement, résolvant ainsi le problème du gradient qui disparaît. L'architecture commence par une couche de convolution, suivie d'une normalisation par lots, d'une activation ReLU et d'un max pooling. Ensuite, les blocs résiduels sont répétés, chacun contenant des couches de convolution, de normalisation par lots et des connexions de raccourci. Les poids et les connexions de raccourci varient entre ces blocs. Cette répétition se poursuit jusqu'à ce que la dernière couche de sortie soit atteinte. La dernière couche de

ResNet-50 est une couche entièrement connectée avec 1000 unités, correspondant au nombre de classes dans l'ensemble de données ImageNet.

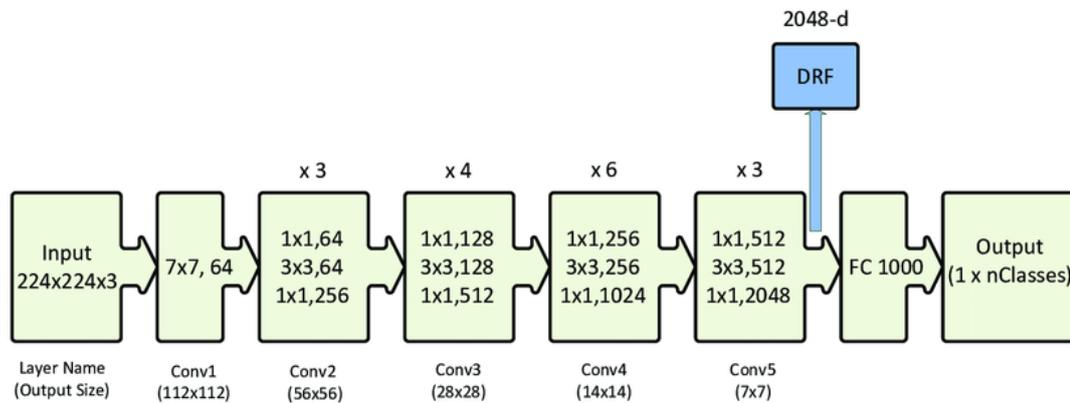


Figure III.4 Architecture du réseau ResNet-50.

III.2.4 VGG16:

VGG16 est un modèle d'architecture de réseau neuronal convolutif, développé par le Visual Geometry Group (VGG) de l'Université d'Oxford en 2014. Il est constitué de 16 couches, comprenant à la fois des couches convolutionnelles et entièrement connectées.

Cette architecture se compose de 13 couches convolutionnelles, suivies de 3 couches entièrement connectées. Les couches convolutionnelles sont spécifiquement conçues pour apprendre des représentations hiérarchiques des images d'entrée, en débutant par des caractéristiques simples telles que les bords et les textures, pour ensuite progresser graduellement vers des caractéristiques de plus haut niveau comme les parties et les formes des objets [24].

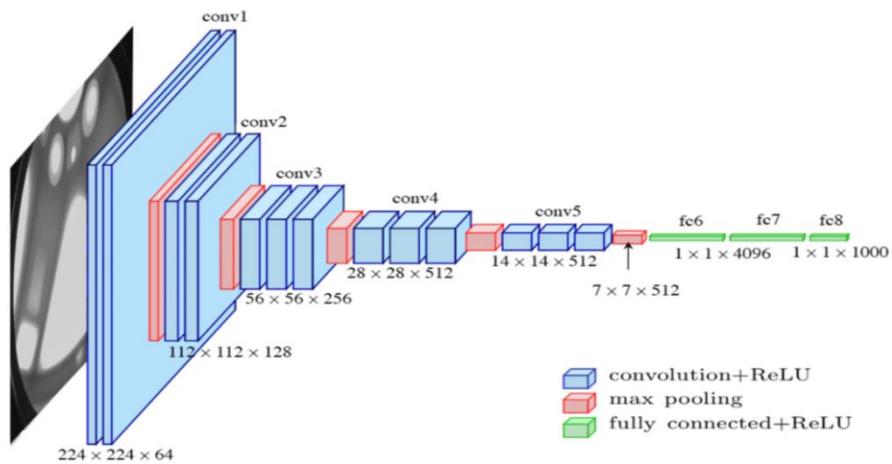


Figure III.5 Architecture du réseau VGG16.

✚ Le tableau suivant résume les différentes architectures que nous allons les détaillées par la suite :

Tableau III.1 Caractéristiques des principaux CNNs [25].

N°	Nom de L'architecture	Année	Contribution principale	Paramètres	Taux d'erreur
1	GoogleNet	2015	-Vainqueur du concours ILSVRC en 2014 - Introduction du bloc inception	4M	ImageNet : 6.7
2	AlexNet	2012	-Plus profond et plus large. -Remporte la compétition ILSVRC en 2012.	60M	ImageNet : 16.4
3	VGG	2014	-Réseau plus profond. -Utilisation des noyaux de petite taille.	138M	ImageNet : 7.3

4	ResNet	2015	-Remporte le concours ILSVRC 2015. -Residual Learning.	25.6M 1.7M	ImageNet : 3.6 CIFAR-10: 6.46
---	--------	------	---	-------------------	--------------------------------------

III.3 Les bases de données:

III.3.1 ImageNet:

ImageNet est une vaste base de données d'images annotées destinée à la recherche en reconnaissance visuelle des objets. Avec plus de 14 millions d'images, chaque image est annotée à l'aide de synsets WordNet, ce qui en fait une ressource inestimable pour l'entraînement de modèles d'apprentissage profond dans les domaines de la vision par ordinateur [26]. Un sous-ensemble d'ImageNet avec environ 1000 images dans chacune de ses 1000 catégories est utilisé par ILSVRC, 150000 photos de test, 50000 images de validation et images de formation.



Figure III.6 La base de données ImageNet.

III.3.2 CIFAR-10:

Le CIFAR-10 est un ensemble de données comprenant 60 000 images de dimensions 32×32 , réparties en 10 classes avec 6000 images pour chaque classe. Ces images sont divisées en cinq ensembles d'entraînement et un ensemble de test, chaque ensemble contenant 10 000 images. L'ensemble de test contient exactement 1000 images sélectionnées aléatoirement pour chaque classe. Les ensembles d'entraînement comprennent le reste des images avec des répartitions arbitraires, bien que certains ensembles puissent contenir plus d'images d'une classe que d'une autre. Au total, les ensembles d'entraînement contiennent précisément 5000 images de chaque classe [27].



Figure III.7 Quelques images de CIFAR-10.

III.3.3 MNIST:

La base de données MNIST (Modified ou Mixed National Institute of Standards and Technology), est une collection de chiffres manuscrits comprenant 60 000 exemples d'entraînement et 10 000 exemples de test. Elle est issue d'une base de données plus vaste, la NIST Spécial Database 3 et Special Database 1, contenant des images monochromes de chiffres manuscrits provenant respectivement d'employés du Bureau du recensement des États-Unis et de lycéens. Les chiffres sont normalisés en taille et centrés dans une image de 20×20 pixels, préservant leur aspect original en noir et blanc. La technique d'anti-aliasing utilisée dans le processus de normalisation donne lieu à des niveaux de gris dans les images résultantes. Enfin, les images sont centrées dans une image de 28×28 pixels en calculant le

centre de masse des pixels, assurant ainsi une disposition uniforme dans toute la base de données [28].

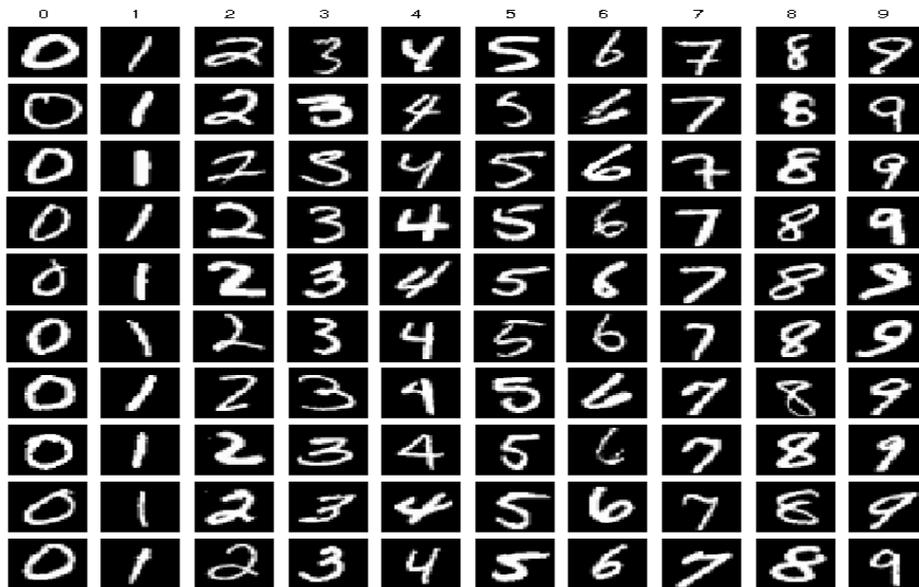


Figure III.8 Un échantillon de MNIST.

III.4 Conclusion:

Dans ce chapitre, nous avons présenté les architectures des CNNs les plus connus tels que GoogleNet, AlexNet, ResNet, VGG ainsi que les bases de données les plus utilisées dans l'entraînement des CNNs. Néanmoins, dans notre travail, nous allons créer notre propre base de données dont les détails seront présentés au chapitre suivant.

Chapitre IV

Simulations, résultats et discussions



IV.1 Introduction:

Dans ce chapitre, nous allons décrire en détails la méthodologie appliquée pour mener notre travail. Une base de données spécifique contenant 10 types d'herbes mauvaises, chaque type ayant 40 images, totalisant ainsi 400 images a été créée et utilisée. Au début, nous avons utilisé AlexNet en raison de sa simplicité et de sa structure comportant 25 couches. Cependant, les résultats obtenus avec AlexNet n'ont pas atteint la précision souhaitée. Nous avons donc travaillé avec GoogleNet pour améliorer les performances. Le but de cette étude est d'atteindre la plus grande précision possible dans la classification des mauvaises herbes.

Pour ce faire, les mêmes options d'entraînement seront utilisées pour les deux réseaux et la comparaison se fera, d'abord, en termes de précision et de temps d'entraînement du réseau, puis, nous évaluerons les performances des modèles en utilisant des métriques telles que la précision, le rappel (Recall) et le F1 score.

IV.2 Langage de programmation:

Les simulations et tests ont été menés en utilisant l'environnement de développement Matlab version R2021a sur un PC avec les caractéristiques ci-dessous:

Processor: Intel(R) Celeron(R) N4000 CPU @ 1.10GHz 1.10 GHz

RAM: 4.00 Go.

Carter graphique: intégrée.

OS: Windows 10 64Bits.

MATLAB (MATrix LABoratory) représente un langage de programmation de quatrième génération et un environnement d'analyse numérique. Conçu par la société THE MathWorks, cet outil se démarque par sa puissance, autorisant les calculs matriciels complexes, la création de graphiques, l'élaboration d'algorithmes ainsi que la conception d'Interfaces Utilisateurs (IU). En outre, il offre la possibilité aux développeurs d'interagir avec des programmes développés dans différents langages.

IV.3 Les différentes méthodes d'entraînement d'un CNN:

Il existe différentes façons d'entraîner un réseau de neurones convolutionnel pour la classification d'images:

- **Entraînement à partir de zéro:** Cette méthode est très difficile et nécessite beaucoup d'expérience. Elle consiste à initialiser les poids du réseau de manière aléatoire et à les optimiser progressivement en utilisant un algorithme d'optimisation comme la descente de gradient. Cette méthode nécessite aussi l'utilisation d'une très large base de données [16].
- **Transfert d'apprentissage:** Cette méthode utilise un réseau pré-entraîné sur un grand jeu de données et le réutilise pour une nouvelle tâche. Cela permet de gagner du temps et d'améliorer les performances du réseau [29].
- **La segmentation sémantique, "semantic segmentation",** consiste à utiliser un réseau pour reconnaître et classer les différents objets présents dans une image. C'est l'une des avancées récentes importantes dans le domaine des CNN [16].

Nous avons choisi la deuxième option en raison de notre manque d'expérience dans le domaine, ainsi que de la taille relativement petite de la base de données que nous avons constituée. L'autre raison est que les ressources hardware disponibles sont très limitées puisque un simple PC est utilisé.

La figure IV.1 montre les étapes d'utilisation d'un réseau pré-entraîné.

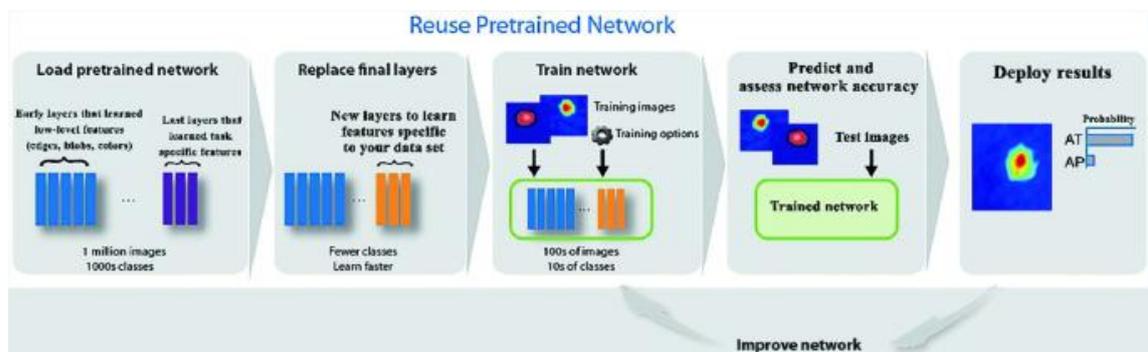


Figure IV.1 Etapes d'utilisation d'un réseau pré-entraîné.

IV.4 Méthodologie:

Pour réaliser les simulations, nous avons procédé comme suit:

IV.4.1 Création d'une base de données:

Nous avons collecté une variété d'images de mauvaises herbes provenant de différentes sources tels que Google Images. Ces images sont au format .JPG présentent des dimensions allant de 185×185 jusqu'à 1920×1281 pixels. Elles sont organisées dans un dossier principal avec des sous-dossiers pour chaque type d'herbe nuisible, totalisant 10 catégories, chacune avec 40 images représentatives de l'herbe correspondante. Ces données variées seront utilisées pour l'entraînement et le test de nos modèles de classification, utilisant d'abord AlexNet puis GoogleNet. Bien que nos images présentent des dimensions différentes, notre programme les prépare efficacement en redimensionnant automatiquement leurs tailles pour correspondre à la dimension d'entrée requise par les deux réseaux neuronaux que nous utilisons.

Le tableau IV.1 montre les types d'herbes utilisés, alors que la figure IV.2 présente quelques échantillons de notre base de données.

Tableau IV.1 Différents types d'herbes nuisibles de notre base de données.

Amaranthus 	Convolvulus arvensis 	Galium tricorne 	Lierre terrestre 	Oxalis 
Pâquerette 	Scandix spp 	Serpolet 	Silène enflé 	Trèfle blanc 



Figure IV.2 Echantillon de la base de données.

IV.4.2 Installation du réseau sur MATLAB:

- Allez dans MATLAB.
- Cliquez sur "Add-Ons".

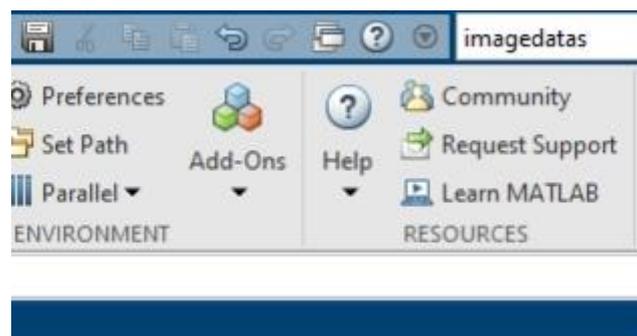


Figure IV.3 :2 Installation d'un réseau CNN sur MATLAB.

- Comme nous avons opté pour AlexNet et GoogleNet, nous les avons téléchargés et installés.
 - Premièrement nous avons installé AlexNet, tels que décrit dans les étapes :

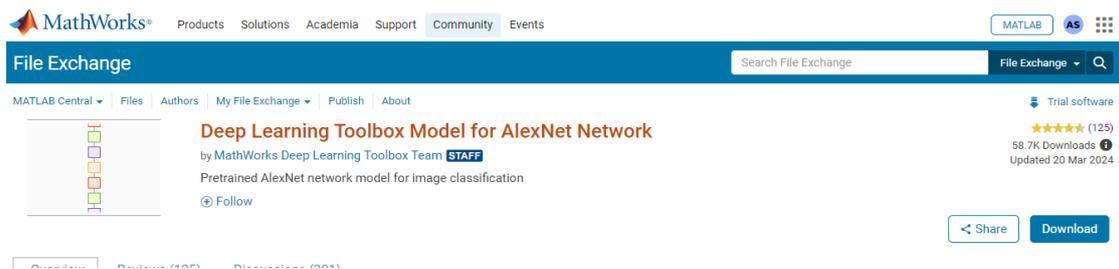


Figure IV.4 Résultat de la recherche d'AlexNet dans Add-Ons.



Figure IV.5 Installation d'AlexNet sur MATLAB.

Les deux étapes précédentes ont été refaites pour installer GoogleNet.

- Vérification de l'installation d'AlexNet et GoogleNet sur MATLAB :

Pour vérifier les caractéristiques principales d'un réseau pré-entraîné en utilisant la variable **'net'** dans MATLAB, il suffit de taper simplement **'net'** dans la fenêtre de commande MATLAB. Cela affichera un résumé des informations principales sur le réseau, y compris son architecture et ses propriétés.

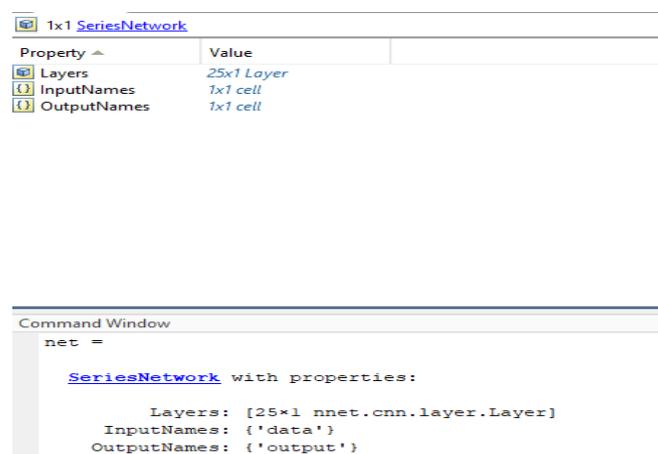


Figure IV.6 Affichage des caractéristiques du réseau installé AlexNet.

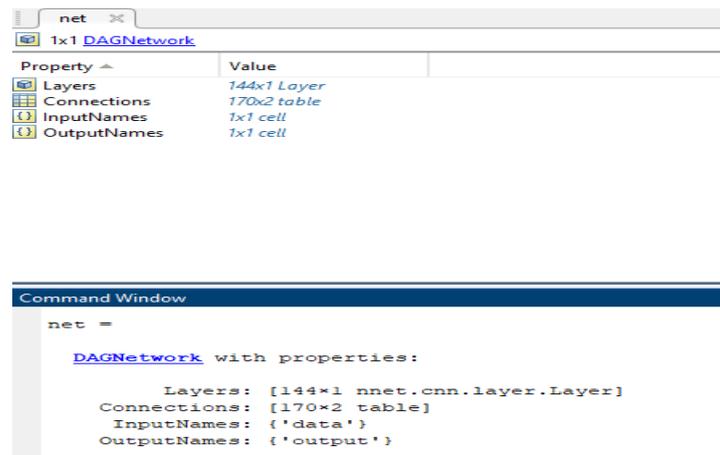


Figure IV.7 Affichage des caractéristiques du réseau installé GoogleNet.

IV.4.3 Chargement de la base de Données:

Après avoir installé le réseau sur MATLAB et examiné ses caractéristiques, la prochaine étape consiste à charger la base de données. Pour ce faire, nous utilisons le navigateur pour placer notre base de données dans le répertoire courant, appelé "Current Folder". Ensuite, nous utilisons la fonction "imageDatastore()" de MATLAB, qui permet de gérer une grande collection d'images. Cette fonction prend en paramètre le chemin complet vers le dossier contenant les images. Elle offre également la possibilité d'inclure les sous-dossiers et de spécifier les étiquettes des images.

- ✓ Le code suivant montre comment « imageDatastore() » est utilisée:

Pour AlexNet:

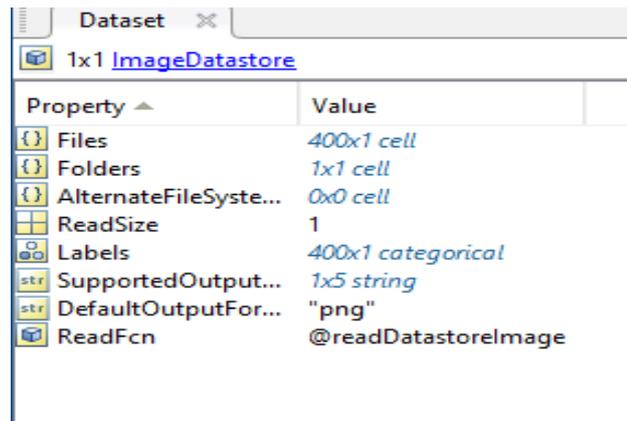
```
imds = imageDatastore('C:\Users\ayasd\Desktop\weed classification using alex-net\Dataset', ...
    'IncludeSubfolders', true, 'LabelSource', 'foldernames');
```

Figure IV.8 Code pour Chargement de la base de Données (AlexNet).

Pour GoogleNet:

```
1 - Dataset = imageDatastore('C:\Users\ayasd\Desktop\weed classification using google-net\Dataset', 'FileExtensions', {' .jpg'}, ...
2     'IncludeSubfolders', true, 'LabelSource', 'foldernames');
```

Figure IV.9 Code pour chargement de la base de données (GoogleNet).



Property	Value
Files	400x1 cell
Folders	1x1 cell
AlternateFileSyste...	0x0 cell
ReadSize	1
Labels	400x1 categorical
SupportedOutput...	1x5 string
DefaultOutputFor...	"png"
ReadFcn	@readDatastoreImage

Figure IV.10 Affichage de la base de données dans Workspace.

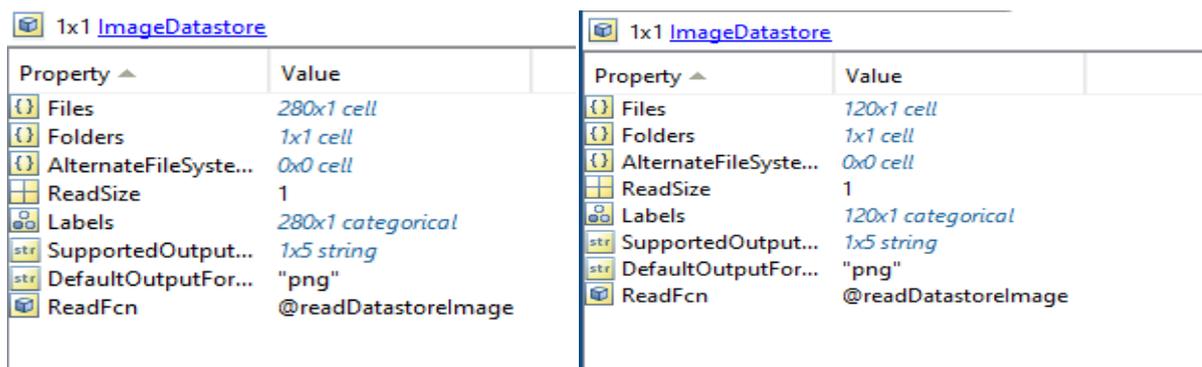
IV.4.4 Partition de la base de données:

Les images de la base de données ont été séparées en deux ensembles distincts: un ensemble d'entraînement composé de 70% des images et un second ensemble contenant les 30% restants.

Afin de réaliser cette division, on a utilisé le code suivant:

```
4 - [Training_Dataset, Validation_Dataset] = splitEachLabel(Dataset, 0.7);
```

Figure IV.11 Code de partition de la base de données.



Property	Value	Property	Value
Files	280x1 cell	Files	120x1 cell
Folders	1x1 cell	Folders	1x1 cell
AlternateFileSyste...	0x0 cell	AlternateFileSyste...	0x0 cell
ReadSize	1	ReadSize	1
Labels	280x1 categorical	Labels	120x1 categorical
SupportedOutput...	1x5 string	SupportedOutput...	1x5 string
DefaultOutputFor...	"png"	DefaultOutputFor...	"png"
ReadFcn	@readDatastoreImage	ReadFcn	@readDatastoreImage

Figure IV.12 Partition de la base de données dans Workspace.

IV.4.5 Analyse des réseaux AlexNet et GoogleNet:

Pour examiner un réseau de neurones convolutionnel dans MATLAB, on utilise la fonction "analyzeNetwork" telle qu'indiqué par la figure IV.13. Cette fonction fournit des détails sur

les différentes couches du réseau, ce qui est crucial pour comprendre comment il traite les données (voir figures IV.14 et 15). En examinant ces informations, on peut identifier les couches qui doivent être ajustées ou modifiées pour adapter le réseau à une tâche spécifique.

```

8 % Analyse de l'architecture du réseau
9 - analyzeNetwork(net)

```

Figure IV.13 Analyse de l'architecture d'un réseau CNN.

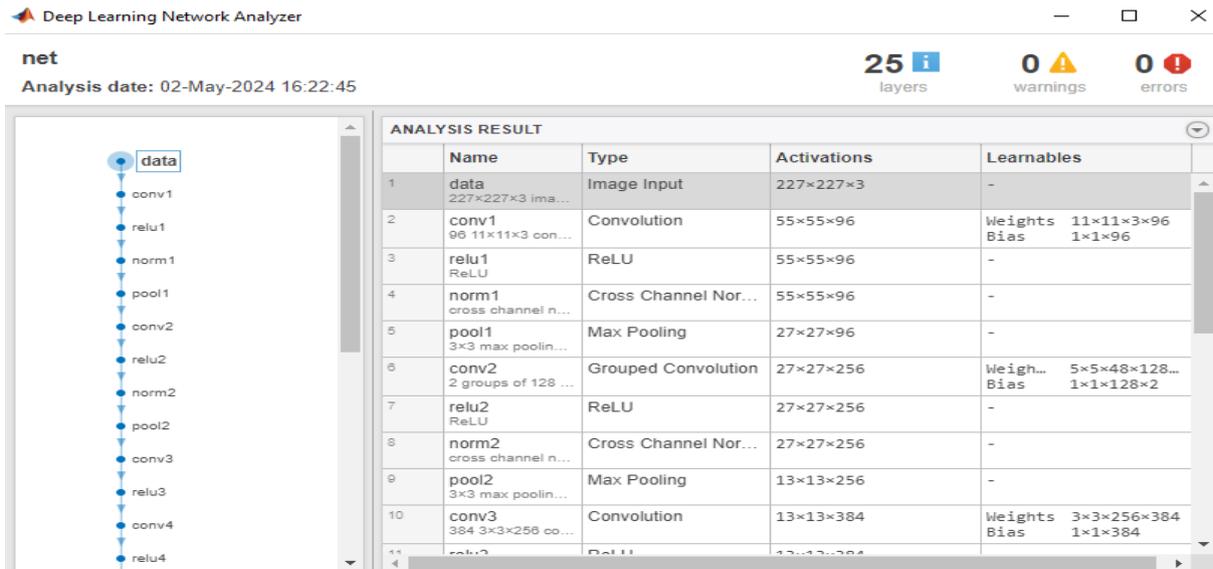


Figure IV.14 Architecture d'AlexNet.

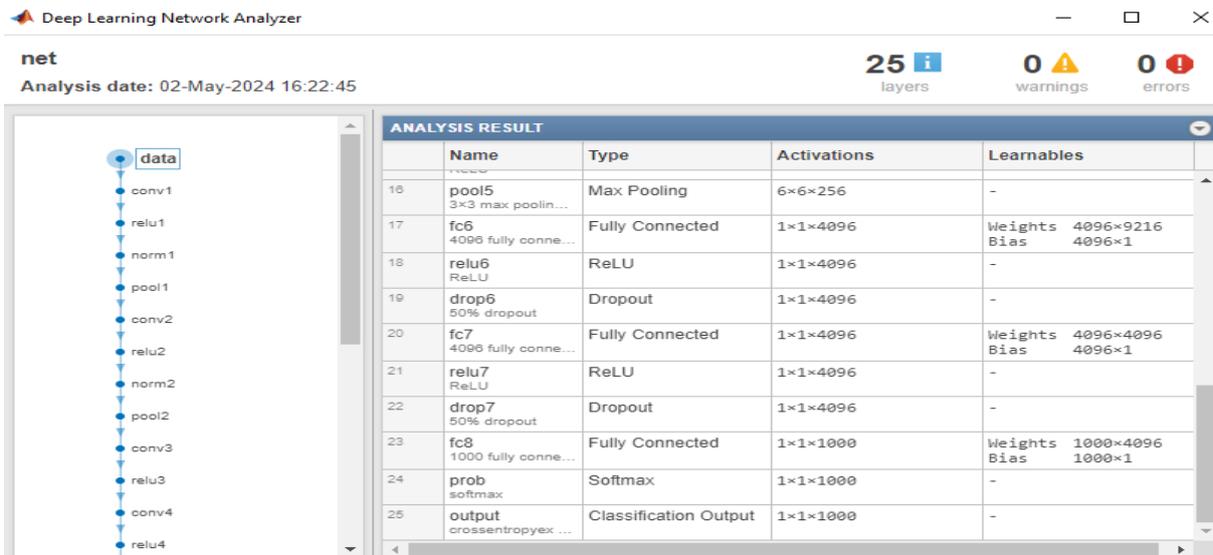


Figure IV.14 Architecture d'AlexNet (suite).

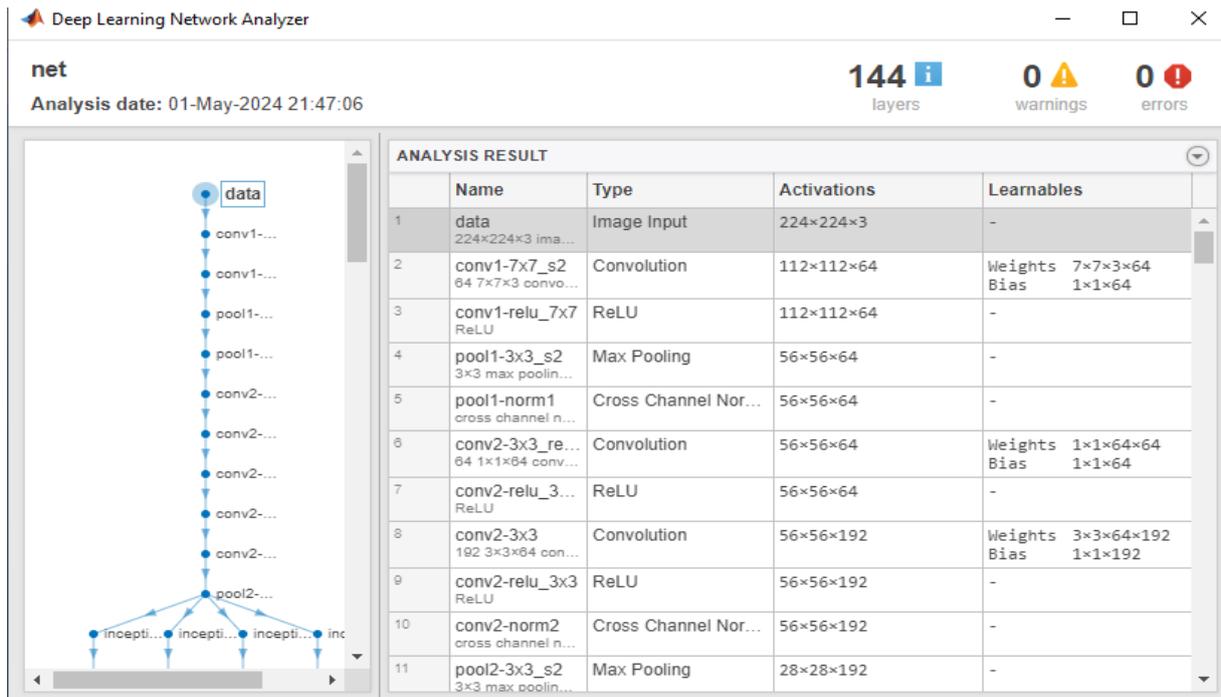


Figure IV.15 Architecture de GoogleNet.

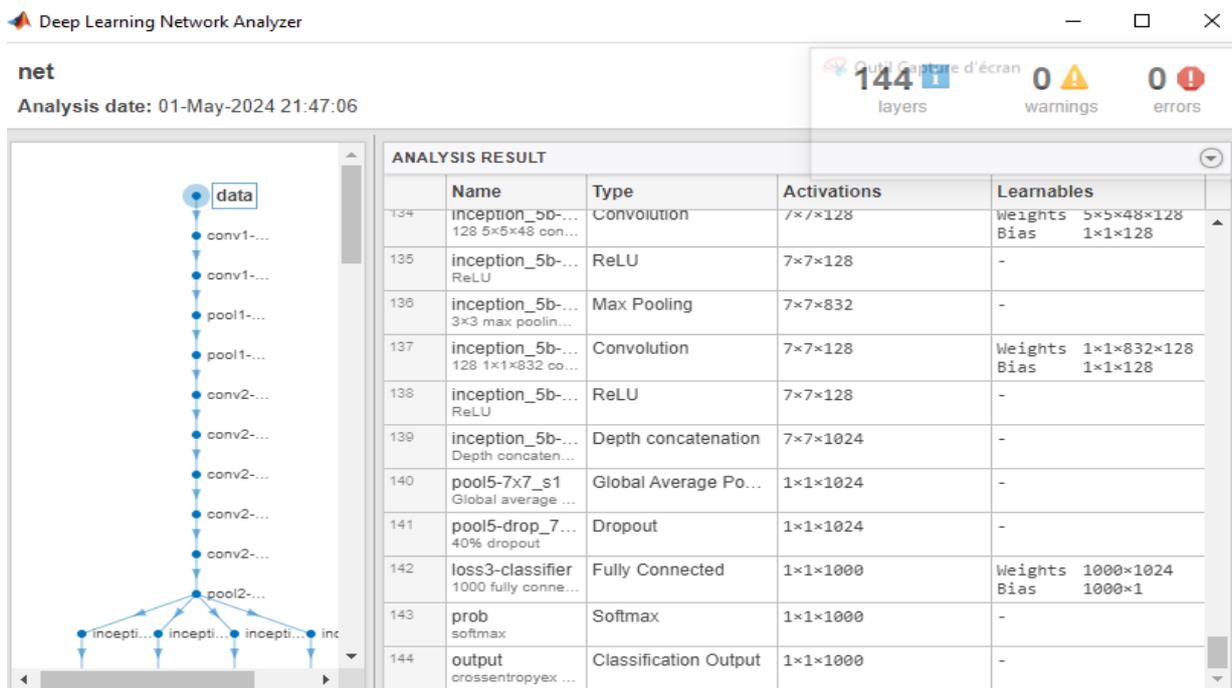


Figure IV.15 Architecture de GoogleNet (suite).

Pour déterminer les informations et la taille de la couche d'entrée d'un réseau CNN on peut, soit utiliser la commande **net.Layers (1)** pour y accéder (figures IV.16 et 17).

```
Command Window

ans =

    ImageInputLayer with properties:

        Name: 'data'
        InputSize: [227 227 3]

    Hyperparameters
        DataAugmentation: 'none'
        Normalization: 'zerocenter'
        NormalizationDimension: 'auto'
        Mean: [227×227×3 single]
```

Figure IV.16 Informations sur la couche d'entrée d'AlexNet.

```
Command Window

>> net.Layers(1)

ans =

    ImageInputLayer with properties:

        Name: 'data'
        InputSize: [224 224 3]

    Hyperparameters
        DataAugmentation: 'none'
        Normalization: 'zerocenter'
        NormalizationDimension: 'auto'
        Mean: [224×224×3 single]
```

Figure IV.17 Informations sur la couche d'entrée de GoogleNet.

- Après avoir segmenté notre base de données en ensembles distincts pour l'entraînement et le test, respectivement composés de 280 images pour l'entraînement et 120 pour le test, nous avons pris des mesures spécifiques pour adapter les images à chaque modèle que nous utilisons. Pour AlexNet, nous avons redimensionné les images à une taille de 227×227 pixels, tandis que pour GoogleNet, elle est de 224×224 pixels. Ces ajustements sont cruciaux pour garantir que les données en entrée soient conformes aux exigences de chaque architecture de modèle. Ainsi, nous utiliserons les variables "**Resized_datasetTrain**" et "**Resized_datasetTest**" pour représenter respectivement

l'ensemble d'entraînement et l'ensemble de validation, après cette opération de redimensionnement.

```
13 % Redimensionnement des images d'entraînement et de validation
14 - Resized_Training_Image = augmentedImageDatastore(Input_Layer_Size, Training_Dataset);
15 - Resized_Validation_Image = augmentedImageDatastore(Input_Layer_Size, Validation_Dataset);
16
```

Figure IV.18 Code pour redimensionner les données d'entraînement et de test.

IV.4.6 Adaptation des couches:

Pour accéder à la couche entièrement connectée qui extrait les caractéristiques à partir des données d'entrée et à la couche de classification où chaque neurone de sortie représente la probabilité d'appartenance à une classe spécifique, nous nous référons simplement aux deux dernières couches du réseau CNN.

Pour accéder à ces deux dernières couches, nous utilisons ce code:

```
17 - Feature_Learner = net.Layers(142);
18 - Output_Classifier = net.Layers(144);
19
```

Figure IV.19 Extraction des dernières couches.

Maintenant ce code permet de déterminer le nombre de classes dans l'ensemble de données d'entraînement. En utilisant la fonction "**categories**", nous récupérons les différentes classes présentes dans les données d'entraînement. Ensuite, en appliquant la fonction "**numel**", nous calculons le nombre d'éléments dans le tableau retourné par "**categories(datasetTrain.Labels)**". Ainsi, "**numel(categories(datasetTrain.Labels))**" nous donne le nombre de classes différentes présentes dans les libellés des données d'entraînement. Ce nombre de classes est essentiel pour définir la taille de la dernière couche du réseau neuronal, qui sera une couche de sortie avec une unité pour chaque classe.

```
21 - Number_of_Classes = numel(categories(Training_Dataset.Labels));
```

Figure IV.20 Nombre de catégories dans la base de données.

Après avoir déterminé le nombre de classes dans l'ensemble de données, Nous définissons nos propres couches en commençant par la couche entièrement connectée, stockée dans la variable "**Feature_Learner**":

```
24 - New_Feature_Learner = fullyConnectedLayer(Number_of_Classes, ...
25     'Name', 'Weed_Feature_Learner', ...
26     'WeightLearnRateFactor', 10, ...
27     'BiasLearnRateFactor', 10);
```

Figure IV.21 Modification du nom et des paramètres de la couche FutureLearner.

On va nommer la nouvelle couche "**Weed_Classifier**":

```
30 - New_Classifier_Layer = classificationLayer('Name', 'Weed_Classifier');
31
```

Figure IV.22 Modification du nom da la couche de classification.

Maintenant, pour remplacer les couches existantes par nos nouvelles couches redéfinies, nous utilisons la fonction "replaceLayer()". Cette fonction prend en argument le graphe de couches, le nom des couches existantes et les nouvelles couches. Le processus est similaire à celui que nous avons utilisé pour remplacer la couche d'apprentissage, comme l'indique le code suivant:

```
33 - New_Layer_Graph = layerGraph(net);
34 - New_Layer_Graph = replaceLayer(New_Layer_Graph, Feature_Learner.Name, New_Feature_Learner);
35 - New_Layer_Graph = replaceLayer(New_Layer_Graph, Output_Classifier.Name, New_Classifier_Layer);
36
```

Figure IV.23 Remplacement des noms des couches du réseau.

Après avoir remplacé les couches de sortie par nos propres couches, nous sommes prêts à démarrer l'entraînement. Avant cela, nous devons nous assurer que nos modifications ont été effectuées correctement en analysant à nouveau le réseau modifié.

```
- analyzeNetwork(New_Layer_Graph)
```

Figure IV.24 Code de vérification des modifications.

Weed_Featur... 10 fully connect...	Fully Connected	1×1×10	Weights 10×1024 Bias 10×1
prob softmax	Softmax	1×1×10	-
Weed_Classi... crossentropyex	Classification Output	1×1×10	-

Figure IV.25 Affichage de la nouvelle architecture de GoogleNet.

IV.4.7 Options d'entraînement:

Avant de commencer l'entraînement d'un réseau pour la classification des images, nous devons définir les "options d'entraînement", qui sont des paramètres importants affectant les performances du réseau, tels que la précision de reconnaissance et le temps d'entraînement.

Parmi les paramètres les plus importants, on trouve:

Batch: Un groupe de données d'entraînement traitées simultanément par le réseau neuronal lors d'une itération.

Epoch: Une passe complète à travers l'ensemble des données d'entraînement lors de l'apprentissage du modèle.

Itération: Une seule mise à jour des poids du réseau effectuée après avoir traité un batch de données.

Taux d'apprentissage (Learning Rate): Un paramètre contrôlant la taille des pas lors de la mise à jour des poids du réseau en fonction du gradient de la fonction de perte [29].

```
% Définition des options d'entraînement
Size_of_Minibatch = 5;
Validation_Frequency = floor(numel(Resized_Training_Image.Files) / Size_of_Minibatch);
Training_Options = trainingOptions('sgdm', ...
    'MiniBatchSize', Size_of_Minibatch, ...
    'MaxEpochs', 6, ...
    'InitialLearnRate', 3e-4, ...
    'Shuffle', 'every-epoch', ...
    'ValidationData', Resized_Validation_Image, ...
    'ValidationFrequency', Validation_Frequency, ...
    'Verbose', false, ...
    'Plots', 'training-progress');
```

Figure IV.26 Paramètres d'entraînement.

La figure IV.26 indique que les valeurs des paramètres sont:

Batch = 5.

Epoch = 6.

Itération = 276 itérations, ce qui donnera 46 itérations par epoch pour AlexNet et 330 itérations pour GoogleNet, ce qui donnera 55 itérations par epoch.

Taux d'apprentissage: Fixé à $3e-4$

IV.4.8 Entraînement du réseau AlexNet:

La figure IV.27 montre les courbes d'entraînement du réseau AlexNet (précision et perte en fonction du nombre d'epochs).

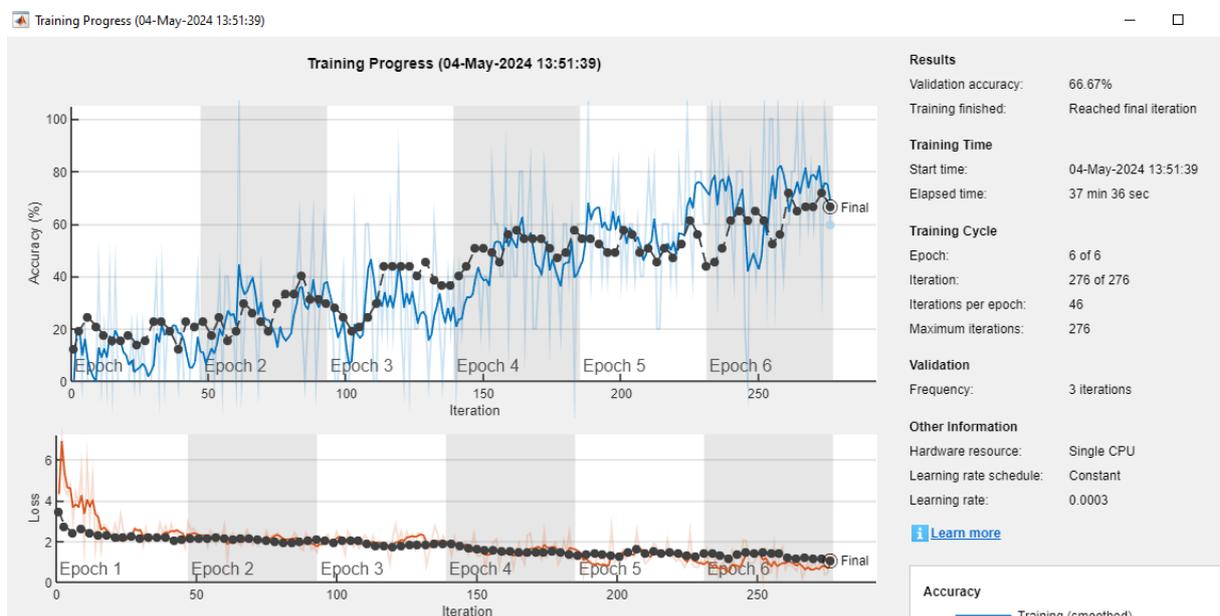


Figure IV.27 Entraînement du réseau AlexNet.

Analyse:

La courbe d'entraînement d'AlexNet est un graphique qui montre l'évolution des métriques de performance (comme la précision et la perte) du modèle AlexNet pendant le processus d'entraînement.

On remarque Que:

- ✓ **Précision croissante mais n'atteignant pas 100% (Courbe en bleu):**

La précision croissante sur l'ensemble de validation est encourageante car elle indique que le modèle apprend et s'améliore au fil du temps.

✓ **Perte presque constante à 2 (Courbe en rouge):**

Une perte presque constante à un niveau relativement élevé, comme 2, peut suggérer que le modèle n'a pas encore convergé vers son optimum global ou qu'il y a des problèmes de convergence.

En résumé, l'analyse de la courbe d'entraînement est cruciale pour évaluer la performance et la convergence d'un réseau CNN. Avec l'utilisation d'AlexNet, nous observons une précision de validation de 66.67% après 6 epoch d'entraînement. Le processus d'entraînement a duré 37 minutes et 36 secondes. Cette précision, bien qu'encourageante pour une première évaluation, suggère la nécessité de poursuivre l'optimisation du modèle pour atteindre des performances plus élevées.

IV.4.9 Classification des mauvaises herbes par AlexNet:

Malgré les performances moyennes du réseau, nous avons procédé au test de ses performances en utilisant des images des 30% initiaux dédiés aux tests ainsi que d'autres images n'appartenant pas à la base de données.

➤ Voici quelques résultats du test indiqués ci dessous:

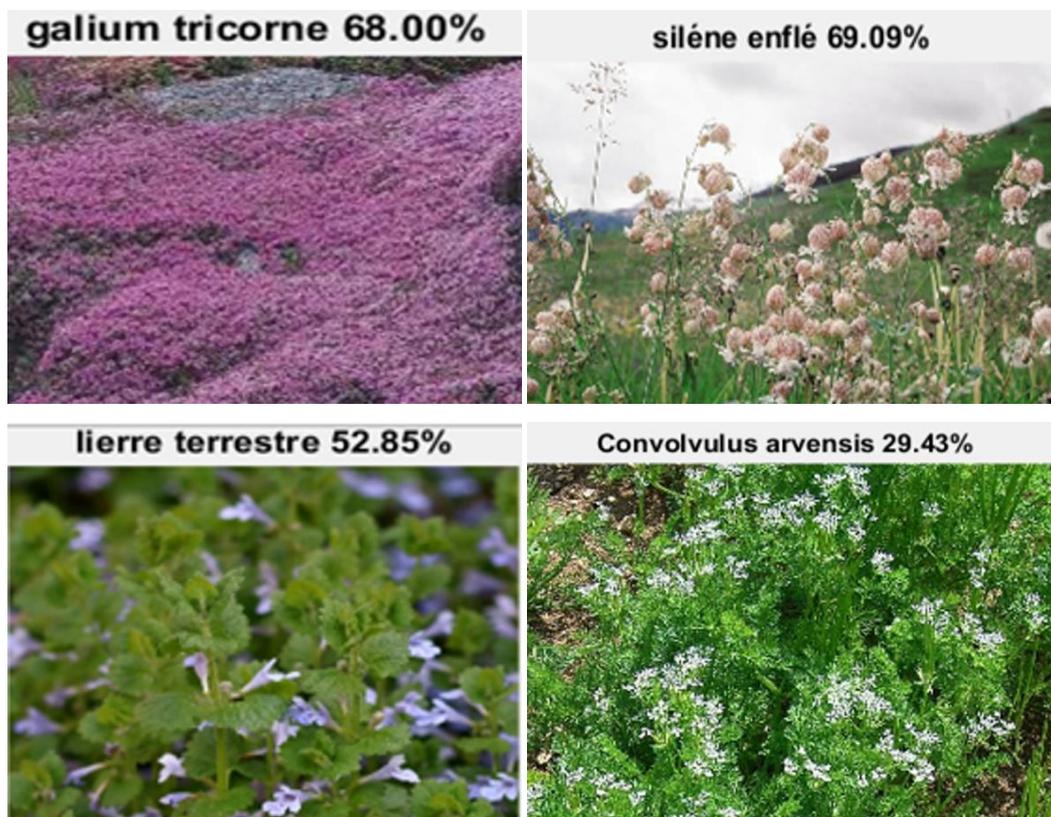


Figure IV.28 Classification des mauvaises herbes par le réseau AlexNet.

Analyse:

Les résultats d'AlexNet montrent une grande variation dans sa capacité à classer correctement les images. Avec une précision allant de 29.43% à 69.09%, le modèle ne semble pas donner des résultats précis et constants. Cela signifie qu'il peut avoir du mal à identifier correctement les objets dans les images. Pour améliorer cette situation, nous envisageons d'essayer GoogleNet, espérant obtenir de meilleurs résultats.

IV.4.10 Entraînement du réseau GoogleNet:

Nous avons utilisé les mêmes paramètres que ceux employés avec AlexNet. Dans le cas du modèle GoogleNet, la précision de validation atteinte à la sixième époque est de 91.67%, avec un temps d'entraînement de 35 minutes et 6 secondes, comme illustré sur les courbes d'entraînement et de perte.

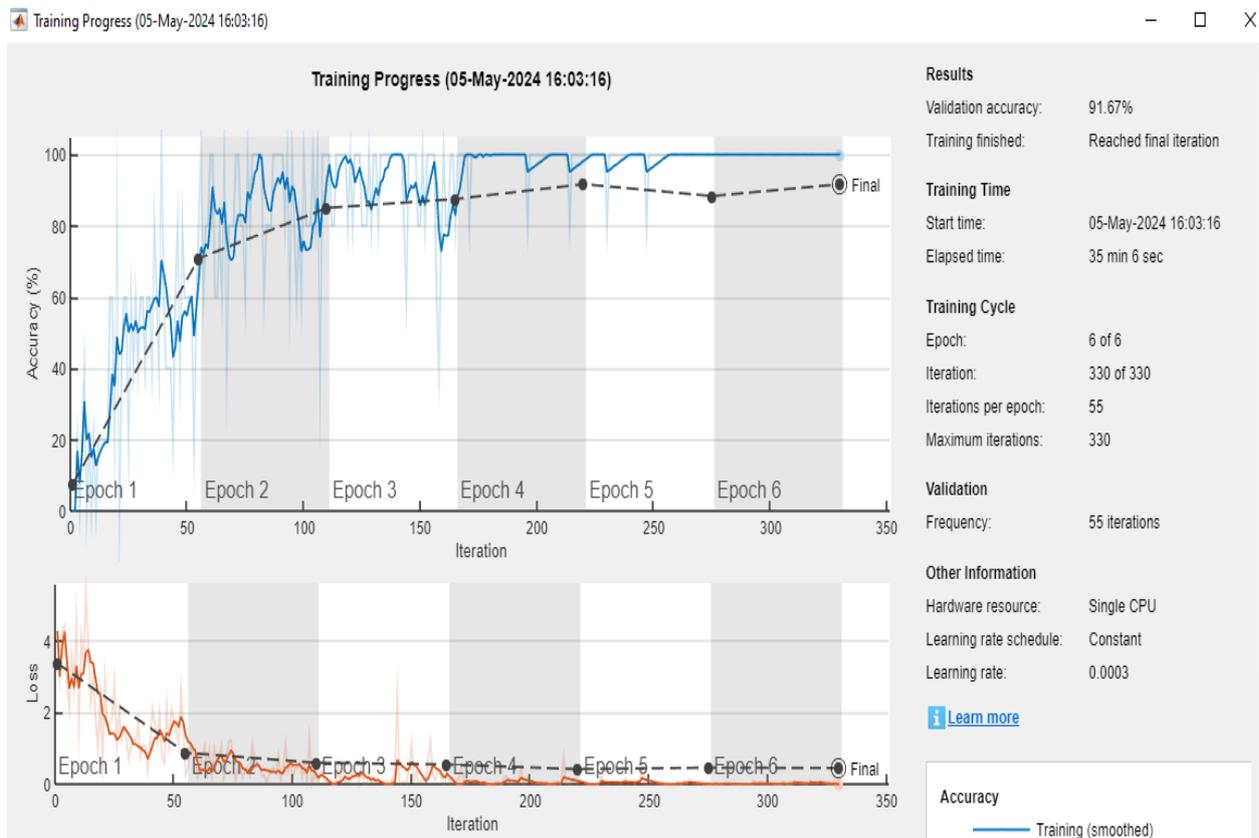


Figure IV.29 Entraînement du réseau GoogleNet.

Analyse:✓ **Précision croissante atteignant 100 % (Courbe en bleu):**

Atteindre une précision de 100 % sur l'ensemble de validation est un signe très positif, indiquant que le modèle a appris à classer correctement toutes les données de validation.

✓ **Perte presque nulle décroissante vers 0 (Courbe en rouge):**

Une perte décroissante vers 0 indique que le modèle s'améliore constamment et que son erreur diminue progressivement jusqu'à atteindre un niveau très faible, ce qui est souhaitable pour une performance optimale.

IV.4.11 Classification des mauvaises herbes par GoogleNet:

La figure IV.30 montre quelques résultats de la classification des images d'herbes mauvaises et la précision.



Figure IV.30 Classification des mauvaises herbes par le réseau GoogleNet.

Analyse:

Les résultats de GoogleNet sont excellents. Avec une précision allant de 97.55% à 100%, le modèle a montré une performance remarquable dans la classification des images. Ces résultats sont cohérents et fiables, ce qui montre que le modèle est très précis dans ses prédictions.

IV.5 Comparaison entre les deux réseaux CNNs:

Le tableau suivant présente une comparaison entre les deux réseaux CNNs:

Tableau IV.2 Comparaison entre AlexNet et GoogleNet.

Réseau CNN	Précision (%)	Temps d'entraînement (minutes: secondes)	Base de données
Alex-Net	66,67	37:36	400 images
Google-Net	91,67	35:06	400 images

Ces résultats confirment que GoogleNet est plus précis et très performant dans la classification des mauvaises herbes. Nous continuerons donc à étudier ses performances avec les métriques de performances largement utilisées comme la précision, le rappel (recall) et le F1 score.

IV.6 Mesure des performances du réseau GoogleNet:**IV.6.1 La matrice de confusion:**

Une matrice de confusion est un tableau qui permet de visualiser les performances d'un modèle de classification. Elle permet de calculer plusieurs métriques importantes notamment la précision, le F1 score et le rappel (recall) [30]. Elle se compose de quatre éléments:

TP (True Positive): Nombre d'échantillons/cas positifs correctement prédits positifs.

FN (False Negative): Nombre d'échantillons/cas positifs incorrectement prédits négatifs.

FP (False Positive): Nombre d'échantillons/cas négatifs incorrectement prédits positifs.

TN (True Negative): Nombre d'échantillons/cas négatifs correctement prédits négatifs.

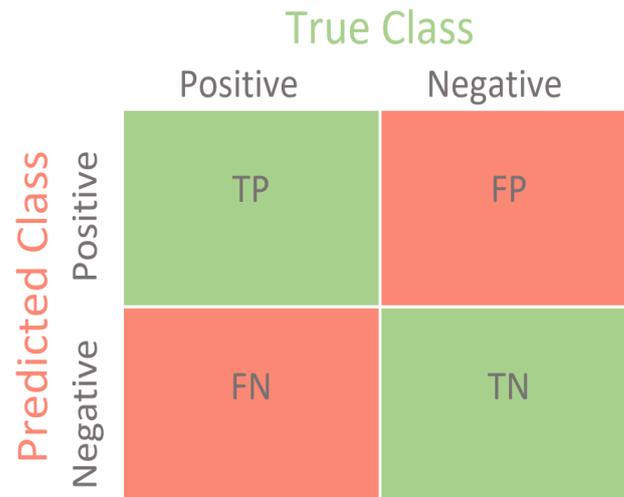


Figure IV.31 Matrice de confusion.

La matrice de confusion est, en général, binaire. Dans notre cas, elle est multi-classes telle que représenté par la figure IV.32.

Output Class	Amaranthus	Convolvulus arvensis	Pâquerette	Scandix spp	galium tricolore	lierre terrestre	oxalis	serpolet	silène enflé	trefle blanc	Accuracy
Amaranthus	7 5.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
Convolvulus arvensis	0 0.0%	11 9.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.8%	1 0.8%	0 0.0%	91.7%
Pâquerette	0 0.0%	0 0.0%	11 9.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
Scandix spp	0 0.0%	0 0.0%	0 0.0%	10 8.3%	2 1.7%	0 0.0%	0 0.0%	0 0.8%	1 0.8%	0 0.0%	76.9%
galium tricolore	1 0.8%	0 0.0%	0 0.0%	2 1.7%	10 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	76.9%
lierre terrestre	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.8%	11 9.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	91.7%
oxalis	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 9.2%	0 0.0%	0 0.0%	0 0.0%	100%
serpolet	1 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.8%	1 0.8%	12 10.0%	0 0.0%	0 0.0%	80.0%
silène enflé	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 9.2%	11 9.2%	0 0.0%	100%
trefle blanc	0 0.0%	0 0.0%	0 0.0%	1 0.8%	1 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	13 10.8%	86.7%
	77.8%	100%	100%	76.9%	71.4%	91.7%	91.7%	100%	84.6%	100%	89.2%
	22.2%	0.0%	0.0%	23.1%	28.6%	8.3%	8.3%	0.0%	15.4%	0.0%	10.8%
	Amaranthus	Convolvulus arvensis	Pâquerette	Scandix spp	galium tricolore	lierre terrestre	oxalis	serpolet	silène enflé	trefle blanc	
	Target Class										

Figure IV.32 Matrice de confusion multi-classes pour GoogleNet.

Analyse:

Les résultats de la matrice de confusion, comprenant les valeurs de TP (True Positive), FN (False Negative) et FP (False Positive), sont résumés dans le tableau ci-dessous:

Tableau IV.3 Résultats TP, FN et FP.

Classe	TP	FN	FP
Amaranthus	7	2	0
Convolvulus arvensis	11	0	1
Pâquerette	11	0	0
Scandix spp	10	3	3
Galium tricorne	10	4	3
Lierre terrestre	11	1	1
Oxalis	11	1	0
Serpolet	12	0	3
Silène enflé	11	2	0
Trefle blanc	13	0	2

Dans ce qui suit, nous analysons les deux cas extrêmes:

- ✓ Pour Pâquerette: Le modèle présente une performance parfaite pour la détection de l'herbe Pâquerette, avec une précision et un rappel de 100%. Le réseau n'a commis aucune erreur ni omission (FN = 0) ni de fausse détection (FP = 0). Cela suggère que le modèle est extrêmement fiable pour cette catégorie spécifique.
- ✓ Pour Galium tricorne: Le modèle présente des performances acceptables pour la détection de Galium tricorne, avec une précision de 71.4%. Cependant, les erreurs de classification (4 FN et 3 FP) indiquent qu'il reste des améliorations à apporter pour rendre le modèle plus précis et fiable pour cette catégorie spécifique.

IV.6.2 Définitions de la précision, du rappel (recall) et du F1 score:

- **Précision** : La précision mesure la proportion de prédictions correctes parmi les prédictions positives. En d'autres termes, elle indique la proportion de résultats positifs qui sont réellement positifs.

$$\text{Précision} = \frac{TP}{TP+FP} \quad (1)$$

- **Rappel (recall)** : Le rappel mesure la proportion de résultats positifs réels qui ont été correctement prédits. En d'autres termes, il indique la proportion de résultats positifs qui n'ont pas été manqués.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

- **F1 score** : Le score F1 est une mesure combinée de la précision et du rappel. Il est calculé comme la moyenne harmonique de la précision et du rappel. Le score F1 est une mesure utile pour évaluer les performances d'un modèle de classification lorsque les classes sont déséquilibrées [31].

$$\text{F1 score} = 2 \times \frac{(\text{Précision} \times \text{Rappel})}{(\text{Précision} + \text{Rappel})} \quad (3)$$

- ✚ Les métriques obtenues pour chaque classe sont résumées dans la figure suivante:

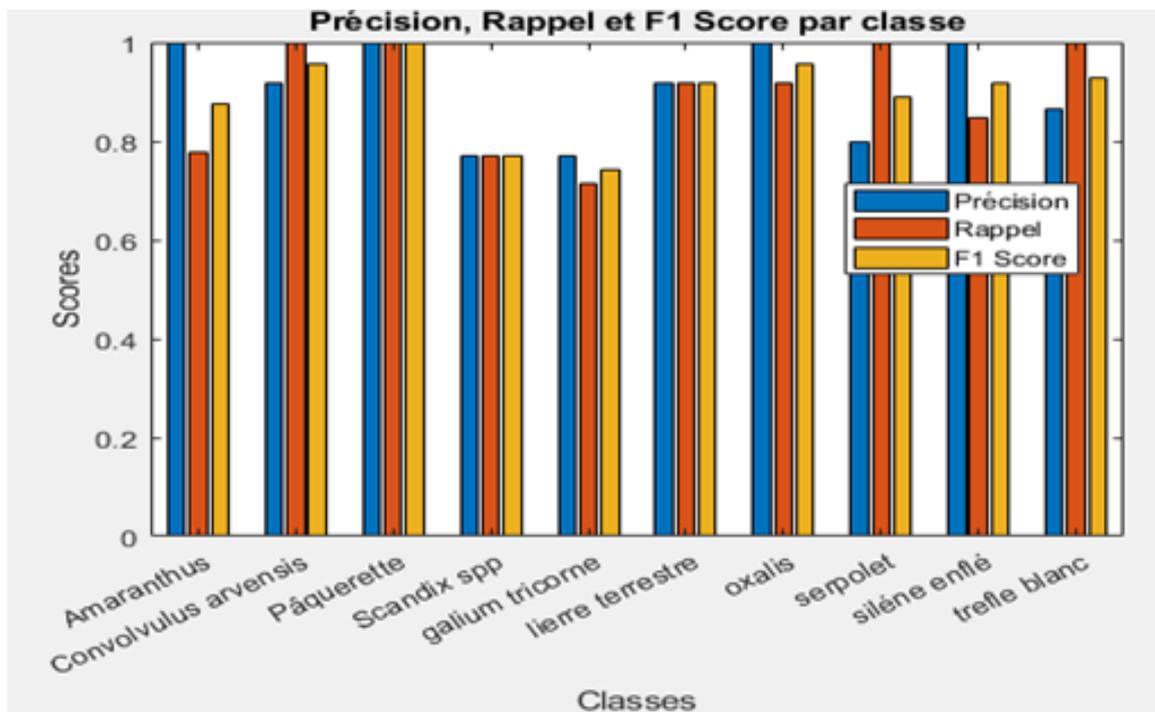


Figure IV.33 Performances du réseau GoogleNet par classe.

- ✚ Les résultats de la précision, du rappel et du F1 score pour GoogleNet, pour chaque classe, sont récapitulés dans le tableau ci-dessous :

Tableau IV.4 Performances du réseau GoogleNet par classe.

Classe	Précision	Rappel	F1 score
Amaranthus	1.00	0.78	0.88
Convolvulus arvensis	0.92	1.00	0.96
Pâquerette	1.00	1.00	1.00
Scandix spp	0.77	0.77	0.77
Galium tricorne	0.77	0.71	0.74
Lierre terrestre	0.92	0.92	0.92
Oxalis	1.00	0.92	0.96
Serpolet	0.80	1.00	0.89
Silène enflé	1.00	0.85	0.92
Trefle blanc	0.87	1.00	0.93

- ✚ Les métriques globales ont été calculées en utilisant la moyenne des True Positives (TP), des False Negatives (FN), et des False Positives (FP) pour chaque classe.

$$\text{TPmoyen} = \frac{\sum \text{TP}}{10} = 10.7$$

$$\text{FN moyen} = \frac{\sum \text{FN}}{10} = 1.3$$

$$\text{FP moyen} = \frac{\sum \text{FP}}{10} = 1.3$$

$$\text{Precision (moyenne)} = \frac{\text{TP moyen}}{(\text{TP moyen} + \text{FP moyen})} = 89.16\%$$

$$\text{Recall (moyen)} = \frac{\text{TP moyen}}{\text{TP moyen} + \text{FN moyen}} = 100\%$$

$$\text{F1 score (moyen)} = 2 \times \frac{(\text{Précision moyenne} \times \text{Recall moyen})}{(\text{Précision moyenne} + \text{Recall moyen})} = \mathbf{94.27\%}$$

Analyse:

En examinant les résultats de GoogleNet, nous constatons une précision moyenne de 89.16%, accompagnée d'un rappel moyen de 100% et d'un score F1 moyen de 94.27%. Ces résultats indiquent que GoogleNet a une performance robuste et cohérente dans la classification des types d'herbes nuisibles. La haute précision montre que le modèle a bien distingué les différentes classes d'herbes nuisibles, tandis que le rappel élevé indique qu'il a correctement identifié la totalité des instances de chaque classe. Le score F1, qui est également élevé, confirme que GoogleNet a maintenu un bon équilibre entre précision et rappel. En somme, ces résultats suggèrent que GoogleNet a des performances globales supérieures dans cette tâche de classification, offrant une bonne fiabilité et exactitude dans ses prédictions.

IV.7 Conclusion:

En conclusion, cette partie pratique a mis en évidence une différence significative des performances entre les deux modèles de réseaux neuronaux convolutifs, AlexNet et GoogleNet. Entraînés et testés sur notre ensemble de données spécifique, les résultats ont clairement montré que GoogleNet surpasse AlexNet en termes de précision. L'utilisation de la matrice de confusion ainsi que des mesures de scores telles F1 score et le rappel a renforcé cette conclusion. Par conséquent, nous pouvons affirmer que GoogleNet est bien adapté à la tâche de classification des mauvaises herbes.

Conclusion Générale

Conclusion Générale:

En conclusion, ce projet de classification des herbes mauvaises en utilisant des CNNs a été une expérience enrichissante. En passant d'AlexNet à GoogLeNet, nous avons pu constater une amélioration significative des résultats, démontrant ainsi l'importance du choix du modèle dans ce domaine. Toutefois, l'exploration de différents modèles, si le temps le permet, peut ouvrir de nouvelles perspectives pour affiner davantage la précision de la classification mais malheureusement nous n'avons pas eu cette opportunité.

L'idée d'incorporer des photos réelles d'herbes mauvaises disponibles en Algérie dans notre base de données spécifique constitue un pas important vers une application plus pertinente et pratique. En envisageant d'étendre ce projet à la détection et à l'élimination des herbes indésirables, avec l'utilisation d'un robot autonome, nous ouvrons la voie à une solution complète et automatisée pour ce problème.

En outre, l'intégration de technologies de vision par ordinateur avec des drones pour surveiller et cartographier les champs pourrait révolutionner la manière dont les cultures sont gérées. Les drones pourraient fournir des images aériennes à haute résolution, permettant une détection précoce et précise des mauvaises herbes, ce qui améliorerait encore l'efficacité du système envisagé.

De plus, une application mobile pourrait transformer la gestion quotidienne des champs. Les agriculteurs recevraient des alertes instantanées sur leur smartphones lorsqu'il y a des mauvaises herbes et des suggestions d'actions correctives à prendre. Par exemple, l'application pourrait recommander l'application de pesticides spécifiques ou des techniques de désherbage mécanique. Cela permettrait aux agriculteurs de réagir rapidement et efficacement, améliorant ainsi la santé et le rendement de leurs cultures.

En intégrant l'Internet des objets (IoT), des capteurs intelligents placés dans les champs collecteraient des données sur les conditions du sol et de l'environnement. Ces informations, combinées avec notre modèle de CNN, permettraient de détecter les mauvaises herbes de manière plus précise et de fournir des recommandations personnalisées aux agriculteurs.

Par ailleurs, l'amélioration du produit agricole grâce à ces technologies est un aspect essentiel. Une meilleure détection et gestion des mauvaises herbes conduit à une utilisation plus efficace des ressources et à une réduction de l'utilisation de pesticides. Cela peut non seulement augmenter les rendements des cultures, mais aussi améliorer la qualité des produits agricoles en minimisant les résidus de produits chimiques. En fin de compte, cela conduit à des

pratiques agricoles plus durables et respectueuses de l'environnement, tout en offrant aux consommateurs des produits plus sains.

En somme, ce projet s'avère être le début d'une exploration plus vaste et ambitieuse dans le domaine de l'agriculture, l'environnement et la santé publique. Les pistes de développement évoquées, telles que l'acquisition de données réelles, l'intégration de drones, la création d'applications mobiles et l'utilisation de capteurs IoT, offrent un aperçu prometteur de ce que le futur pourrait réserver. Cette combinaison de technologies pourrait mener à des solutions agricoles de précision, réduisant l'usage de produits chimiques, augmentant les rendements, améliorant la qualité des produits agricoles et renforçant la durabilité environnementale.

Références Bibliographiques

- [1] https://www.google.com/url?q=https://www.cropscience.bayer.dz/fr-dz/apropos-de-nous/l-agriculture-en-algerie.html&sa=D&source=docs&ust=1714592305886388&usg=AOvVaw3XFpNQp_7doGiGN5srWtH
- [2] https://www.google.com/url?q=https://agricchem.dz/19/la-gestion-des-mauvaises-herbes-un-facteur-d-amelioration-de-la-production-du-ble-en-algerie&sa=D&source=docs&ust=1714592305887273&usg=AOvVaw0c05nZx_R21QIWZH_ZMgiSQ
- [3] <https://www.cairn.info/revue-chimeres-2014-1-page-168.htm&wt.src=pdf>
- [4] <https://fr.wikipedia.org/wiki/P%C3%A2querette>
- [5] https://fr.wikipedia.org/wiki/Tr%C3%A8fle_blanc
- [6] https://fr.wikipedia.org/wiki/Lierre_terrestre
- [7] <https://books.google.com/books?hl=ar&lr=&id=Q7IUDwAAQBAJ&oi=fnd&pg=PP1&dq=%27intelligence+artificielles+&ots=0IDgQ9TAZT&sig=F4y0CW1cqf7kZYTEysI0kELG1gE>
- [8] <https://praedictia.com/page/lintelligence-artificielle/lhistoire-de-lia.html>
- [9] https://en.wikipedia.org/wiki/Deep_learning
- [10] <https://datavalue-consulting.com/deep-learning-reseaux-neurones-recurrents-rnn/>
- [11] https://datafranca.org/wiki/R%C3%A9seau_antagoniste_g%C3%A9n%C3%A9ratif
- [12] https://fr.wikipedia.org/wiki/R%C3%A9seau_neuronal_convolutif
- [13] https://fr.wikipedia.org/wiki/R%C3%A9seau_neuronal_convolutif
- [14] Sari Yasmine et Boubtana Zahra Rayene, Memoire Master « Analyse des paramètres affectant la précision de la reconnaissance d'images par deep learning dans un réseau neuronal convolutif» UNIVERSITE BADJI MOKHTAR – ANNABA , 2022.
- [15] https://dspace.ummo.dz/bitstream/handle/ummo/7631/HouacineNoura_KhelifaNadia.pdf
- [16] LIDOUGHI Rayene et BELHARAZEM Nasrine, Memoire Master «Impact de la complexité de CNN sur la précision de la reconnaissance des images» UNIVERSITE BADJI MOKHTAR – ANNABA , 2023.

- [17] <https://www.ummtto.dz/dspace/bitstream/handle/ummtto/12260/BouadiDyhia.pdf?sequence=1>
- [18] https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn?fbclid=IwAR1ALeQJ5Qk4FTOmI-ftqwhq2WhiTWbCLZi5GcP4k77Fem_S7je-zQpN0FM
- [19] [https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explicationsimple/#:~:text=sigmoid\(x\),Softmax,chaque%20vecteur%20est%20trait%C3%A9%20ind%C3%A9pendamment](https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explicationsimple/#:~:text=sigmoid(x),Softmax,chaque%20vecteur%20est%20trait%C3%A9%20ind%C3%A9pendamment)
- [20] [KAZI TANI (Ep. DALI YOUCEF) Lamia Fatiha « Conception et Réalisation d'un Outil Semi-Automatique Dédié à l'Analyse et la Recherche du Contenu Vidéo du Web », Université 2020. ,Oran ,Algérie , احمد بن بلة, Oran
- [21] <https://depot-e.uqtr.ca/id/eprint/9839/1/eprint9839.pdf>
- [22] <https://dspace.univ-ouargla.dz/jspui/bitstream/123456789/35571/1/RAHMOUNI-HEDDAOUI.pdf>
- [23] <http://dspace.univ-jijel.dz:8080/xmlui/bitstream/handle/123456789/5688/M-ELE.SY.E-2020-06.pdf?sequence=1>
- [24] <https://dspace.univ-ouargla.dz/jspui/bitstream/123456789/34311/1/MAHCENE-BETTAYEB.pdf>
- [25] <https://medium.com/@adbib.mohamed0/larchitecture-cnn-6fd06b6f9b11>
- [26] <https://docs.ultralytics.com/fr/datasets/classify/imagenet/>
- [27] <https://datascience.eu/fr/vision-par-ordinateur/lensemble-de-donnees-cifar-10/>
- [28] https://paperswithcode.com/dataset/mnist?fbclid=IwAR1uhPYAkRZBsoVcJ2O72y0QhOrFGkodPSFjdHUDb37wVb-BL_RNj77kfy4
- [29] chrome extension://efaidnbmnnnibpcajpcgclefindmkaj/https://www.mathworks.com/content/dam/mathworks/ebook/gated/deep-learning-practical-examples-ebook.pdf
- [30] <https://datascientest.com/matrice-de-confusion>
- [31] <https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec>