

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : TECHNOLOGIE

Département : ELECTRONIQUE

Domaine : SCIENCES ET TECHNIQUES

Filière : Télécommunications

Spécialité : Réseaux et télécommunications

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Analyse des paramètres affectant la précision de la reconnaissance d'images par deep learning dans un réseau neuronal convolutif

Présenté par : *SARI YASMINE & BOUBTANA ZAHRA RAYENE*

Encadrant : *HARIZE SALIHA* Grade *MCA* Université : *Badji-Mokhtar ANNABA*

Jury de Soutenance :

AMARA Fethi	MCB	Université Badji-Mokhtar ANNABA	Président
HARIZE Saliha	MCA	Université Badji-Mokhtar ANNABA	Encadrant
NAILI Zakaria	MAB	Université Badji-Mokhtar ANNABA	Examineur

Année Universitaire : 2021/2022

Remerciements

*M*erci. Nos vifs et sincères remerciements à Dieu « **Allah** », qui nous a donné la force, la volonté, la santé, le courage et la patience durant nos années d'études et afin d'accomplir ce travail.

*M*erci. Notre immense reconnaissance et notre profonde gratitude à notre encadreur, **Mme. Saliha Harize**, de nous avoir fait l'honneur de nous proposer ce sujet, d'avoir eu la patience et la détermination de nous guider pendant ces temps spéciaux, et nous avons eu l'honneur et le privilège de travailler sous son assistance et de profiter de ses qualités humaines, professionnelles et de sa grande expérience.

*M*erci. Chaleureusement les membres du jury pour avoir accepté de juger ce mémoire de fin d'études.

*M*erci à tous nos enseignants de l'université d'Annaba.

*M*erci infiniment à nos familles, en particulier nos parents, pour nous avoir toujours soutenu au cours de nos études. Qu'ils trouvent ici le fruit de leur patience et du soutien permanent qu'ils nous ont prodigué pour affronter tous les moments difficiles.

*M*erci énormément à nos camarades pour leur aide et leur conseils précieux.

Dédicaces

*Je dédie ce modeste travail avec grand amour, sincérité et fierté à : **Allah** qui m'a aidé au long de ma vie et surtout dans mes études.*

Je dédie ce modeste travail avec grand amour, sincérité et fierté à :

*A La plus proche de mon cœur ma très chère mère "**Boumaiza Fariha**" et mon très cher père "**Amar**", qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance.*

A mon soutien moral et source de joie et de bonheur mes sœurs soumaya, hiba el-rahemene et basmala.

*A ma chère tante "**Boumaiza djahida**"*

*A mon chère binôme **boubtana zahra rayen** et toute sa famille.*

Au département d'électronique enseignants et fonctionnaires, toute la promotion de télécommunication.

*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.
Merci à tous d'être dans ma vie.*

Sari Yasmine

Dédicaces

Je dédie ce modeste travail avec grand amour, sincérité et fierté à : Allah qui m'a aidé au long de ma vie et surtout dans mes études.

Je dédie ce modeste travail avec grand amour, sincérité et fierté à :

A la femme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celle qui s'est toujours sacrifiée pour me voir réussir, que Dieu te garde toujours avec moi, à toi ma mère « Sakta FATMA ».

A celui que j'aime beaucoup et qui m'a soutenu tout au long de ce projet : mon amie NESMA et bien sûr ma sœur RIHEM et mon frère ABDELWAHAB.

A toute ma famille, A ma copine YASMINE et sa gentille maman que j'aime, à mes douces amies AYMEN et MAISSA et à tous ceux qui ont contribué de près ou de loin pour rendre ce projet possible, je dis merci.

Tous mes très chers potes et amis, RIAD, RYM, NOUH, je dédie ce travail dont le grand plaisir leur revient en premier lieu pour leurs conseils, aides, et encouragements, et être à mes côtés dans ces années d'études.

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.

Merci à tous d'être dans ma vie.

Boubtana Rayene

Résumé:

L'intelligence artificielle, les réseaux de neurones convolutifs " CNN " et l'apprentissage profond " deep learning " ont connu des évolutions impressionnantes au point d'être utilisés dans plusieurs domaines tels que le diagnostic médical, les décisions judiciaires et même la composition musicale, mais et surtout dans la reconnaissance et identification des images très utiles en vidéo/télé-surveillance. Ce travail est consacré à l'utilisation d'un réseau CNN pré-entraîné par transfert d'apprentissage et son utilisation dans la classification des oiseaux. Les effets de certains hyper paramètres tels que le "batch size", le nombre d'époque et la valeur du taux d'apprentissage "learning rate " sur la précision et le temps d'entraînement sont analysés.

Mots clés: IA, CNN, deep learning, epoch, learning rate, classification d'images.

Abstract:

Artificial intelligence, convolutional neural networks "CNN" and deep learning have experienced impressive developments to the point of being used in several fields such as medical diagnosis, court decisions and even musical composition, but and above all in the recognition and identification of very useful images in video/remote surveillance. This work is devoted to the use of a CNN network pre-trained by transfer learning and its use in the classification of birds. The effects of certain hyper parameters such as the "batch size", the number of epochs and the value of the learning rate on the accuracy and the training time are analyzed.

Key words: AI, CNN, deep learning, epochs, learning rate, classification of images.

ملخص:

لقد شهد الذكاء الاصطناعي والشبكات العصبية التلافيفية "CNN" و"التعلم العميق" تطورات مثيرة للإعجاب لدرجة أنه يتم استخدامها في العديد من المجالات مثل التشخيص الطبي والقرارات القضائية وحتى التأليف الموسيقي، ولكن وقبل كل شيء في الاعتراف وتحديد الصور التي تكون مفيدة جداً في المراقبة بالفيديو / عن بعد تم تخصيص هذا العمل لاستخدام شبكة " CNN " المدربة مسبقاً عن طريق نقل التعلم واستخدامها في تصنيف الطيور يتم تحليل تأثيرات بعض المعلمات الفائقة مثل "حجم الدفعة"، وعدد الفترات وقيمة معدل التعلم على الدقة ووقت التدريب.

الكلمات المفتاحية: الذكاء الاصطناعي، الشبكات العصبية التلافيفية، التعلم العميق، الفترات، معدل التعلم، تصنيف الصور.

Liste des Symboles

C

CONV : Convolution.

CNN : Convolutional Neural Network.

I

IA : Intelligence Artificielle.

ILSVRC : ImageNet Large Scale Visual Recognition Challenge.

M

ML : Machine Learning.

MNIST : Modified National Institute of Standard and Technology.

R

ReLU : Rectified Linear Unit.

Re-Net : Residual Network.

R G B : Red Green Blue.

V

VGG : Visual Geometry Group.

W

WSN : Wireless Sensor Network.

Liste des Figures

Figure I.1 :	L'intelligence artificielle.....	3
Figure I.2 :	Machine Learning.....	4
Figure I.3 :	Schéma d'un réseau de neurone profond.....	4
Figure I.4 :	Exemple d'application d'un réseau de neurone profond.....	5
Figure I.5 :	La relation entre l'IA, ML et le Deep Learning.....	6
Figure I.6 :	Architecture d'un réseau de neurones convolutif.....	7
Figure I.7 :	Les couches d'un CNN.....	8
Figure I.8 :	Images RGB composées de trois plans.....	8
Figure I.9 :	Exemple de convolution avec un filtre 3×3.....	9
Figure I.10 :	Exemple de la fonction ReLu.....	9
Figure I.11 :	Réseau entièrement connecté.....	10
Figure II.1 :	L'erreur top-5 en % des architectures gagnantes de la compétition ILSVR.....	11
Figure II.2 :	Visualisation de l'architecture du réseau Alex-Net.....	12
Figure II.3 :	Visualisation de l'architecture du réseau Zef-Net.....	12
Figure II.4 :	Visualisation de l'architecture du réseau VGG-16.....	13
Figure II.5 :	Visualisation de l'architecture du réseau Google-Net.....	14
Figure II.6 :	Visualisation de l'architecture du réseau Res-Net.....	15
Figure II.7 :	Echantillons de la base MNIST.....	18
Figure II.8 :	Image-Net.....	19
Figure II.9 :	Les 10 catégories de CIFAR-10.....	19
Figure III.1:	Etapas d'utilisation d'un réseau pré-entraîmé.....	22
Figure III.2 :	Capture de liste d'outil HOME ENVIRONEMENT de MATLAB.....	24
Figure III.3 :	Résultat de la recherche de Google-Net dans Add-Ons.....	24
Figure III.4 :	Installation de Google-Net sur MATLAB.....	25
Figure III.5 :	Affichage des caractéristiques de Google-Net sur Workspace et command windows.....	25
Figure III.6 :	Code pour Chargement de la base de données.....	26

Figure III.7 :	L'affichage de dataset dans workspace.....	26
Figure III.8 :	Code de partition de la base de données.....	26
Figure III.9 :	Affichage de dataset dans wokspace (30% Validation).....	26
Figure III.10 :	Affichage de dataset dans workspace (70% Entraînement).....	26
Figure III.11 :	Affichage des couches du Google-Net.....	27
Figure III.12 :	Suite des couches de Google-Net.....	27
Figure III.13 :	Code pour afficher l'architecture de Google-Net.....	27
Figure III.14 :	Architecture du Google-Net.....	27
Figure III.15 :	Suite de l'architecture de Google-Net.....	28
Figure III.16 :	Informations sur la couche d'entrée de Google-Net.....	28
Figure III.17 :	Code d'affichage de la taille de couche d'entrée de Google-Net	28
Figure III.18 :	Extraction des dernières couches.....	29
Figure III.19 :	Nombre de catégories dans la base de données.....	29
Figure III.20 :	Modification du nom et des paramètres de la couche FeautreLearner.....	29
Figure III.21 :	Affichage de la structure des couches.....	29
Figure III.22 :	Modification du nom de la couche de classification.....	30
Figure III.23 :	Remplacement des couches de Google-Net.....	30
Figure III.24 :	Affichage de la nouvelle architecture de Google-Net.....	30
Figure III.25 :	Paramètres d'entraînement.....	31
Figure III.26 :	Vue d'ensemble de la classification des oiseaux à l'aide de Google-Net modifié.....	32
Figure III.27 :	Processus d'entraînement pour 2 Batch.....	37
Figure III.28 :	Processus d'entraînement pour 5 Batch.....	37
Figure III.29 :	Processus d'entraînement pour 10 Batch.....	38
Figure III.30 :	Processus d'entraînement pour 4 Epoch.....	39
Figure III.31 :	Processus d'entraînement pour 6 Epoch.....	40
Figure III.32 :	Processus d'entraînement pour 9 Epoch.....	40
Figure III.33 :	Processus d'entraînement avec Taux d'apprentissage égale $4e-6$	41
Figure III.34 :	Processus d'entraînement avec Taux d'apprentissage égale $3e-4$	42
Figure III.35 :	Processus d'entraînement avec Taux d'apprentissage égale $3e-3$	42

Liste des Tableaux

Tableau II.1 :	Les architecture de CNN.....	16
Tableau III.1 :	Les 10 oiseaux de la Base de données.....	23
Tableau III.2 :	Résultats des tests.....	33
Tableau III.3 :	Résultats des tests des images n'appartenant pas à la base.....	34
Tableau III.4 :	Résultats des tests des images avec rotation.....	35
Tableau III.5 :	Résultat des tests des images n'appartenant pas aux catégories de la base..	36
Tableau III.6 :	Effet du paramètre "Batch".....	38
Tableau III.7 :	Effet du paramètre "Epoch".....	41
Tableau III.8 :	Effet du paramètre "initial Learning Rate".....	43

Table des matières

Remerciements	I
Dédicaces Sari	II
Dédicaces Boubtana	III
Résumé	IV
Liste des symboles	V
Liste des figures	VI
Liste des tableaux	VIII
Introduction générale	01
Chapitre I : IA, ML, DL et réseau de neurones convolutifs.	
I.1. Introduction	03
I.2. Intelligence artificiel	03
I.3. Apprentissage automatique (Machine Learning)	03
I.4. L'apprentissage profond (Deep Learning)	04
I.4.1. Fonctionnement du deep learning	05
I.5. Réseau de neurones convolutifs (Convolution Neural Network ou CNN)	06
I.5.1. Architecture globale des CNN	07
I.5.1.1. Couche d'entrée du CNN	08
I.5.1.2. Couche de convolution	08
I.5.1.3. Couche de correction (ReLu : Rectified Linear Units)	09
I.5.1.4. Couche de pooling	09
I.5.1.5. Couche complètement connectée (Fully connected Layer)	10
I.5.1.6. Couche Softmax/logistique	10
I.5.1.7. Couche de sortie (Output layer)	10
I.6. Conclusion	10
Chapitre II : Architectures des CNN et Base de données.	
II.1 Introduction	11
II.2 Les architectures des CNN	11
II.2.1 Alex-Net	11
II.2.2 Zef-Net	12
II.2.3 VGG-16	13

II.2.4 Google-Net/Inception-V1.....	13
II.2.5 Res-Net.....	14
II.3 Les bases de données.....	18
II.3.1 MNIST.....	18
II.3.2 Image-Net.....	18
II.3.3 CIFAR-10.....	19
II.4 Conclusion.....	20
 Chapitre III : Méthodologie, résultats et discussion. 	
III.1 Introduction.....	21
III.2 Langage de programmation.....	21
III.3 Les Différents façons d’entraîner un réseau de CNN.....	21
III.4 Transférer l’apprentissage.....	22
III.5 Méthodologie.....	23
III.5.1 Création d’une base de données.....	23
III.5.2 Installation du réseau Google-Net sur MATLAB.....	24
III.5.3 Chargement de la base de données.....	25
III.5.4 Partition de la base de données.....	26
III.5.5 Adaptation des couches.....	29
III.5.6 Options d’entraînement.....	30
III.6 Schéma récapitulatif de la méthodologie.....	32
III.7 Tests avec différentes images.....	33
III.7.1 Tests sur des images d’oiseaux des catégories de la base.....	33
III.7.2 Tests sur des images n’appartenant pas à la base.....	34
III.7.3 Tests sur des images avec rotation.....	35
III.7.4 Tests sur des images n’appartenant pas aux catégories de la base	35
III.8 Analyse des effets des paramètres d’entraînement sur la Vitesse et la précision de reconnaissance.....	36
III.8.1 Batch.....	36
III.8.2 Itération.....	38
III.8.3 Epoch.....	38
III.8.4 Taux d’apprentissage.....	41
III.9 Conclusion.....	43
Conclusion générale.....	44

Références bibliographiques..... 46

Introduction générale

L'intelligence artificielle a connu une croissance monumentale en comblant le fossé entre les capacités des humains et des machines. Chercheurs et passionnés, travaillent sur de nombreux aspects du domaine pour réaliser des choses incroyables telles que la vision par ordinateur, la reconnaissance d'images, la détection d'objets, le diagnostic médical, les décisions juridiques et même la composition musicale. Le domaine de l'intelligence artificielle comprend l'apprentissage automatique, où les machines peuvent apprendre par expérience et acquérir des compétences sans intervention humaine et l'apprentissage profond est une branche de l'apprentissage automatique où les algorithmes de réseaux de neurones inspirés du cerveau humain apprennent à partir de grandes quantités de données. L'apprentissage en profondeur utilise des réseaux de neurones à plusieurs couches. Un réseau neuronal profond analyse les données avec des représentations apprises de la même manière qu'une personne considérerait un problème.

Les réseaux de neurones convolutifs "CNN " se distinguent des autres réseaux de neurones par leurs performances supérieures avec des entrées d'image, de parole ou de signal audio. Ils ont trois principaux types de couches, qui sont : Couche convolutive, couche de Pooling et couche entièrement connectée "Fully Connected".

Il n'est plus à démontrer l'importance de la faune et la flore dans la préservation des écosystèmes et de l'environnement. Ainsi, les développements technologiques en matière de télécommunication sont exploités dans la surveillance des lieux et des espèces animales menacées d'extinction. Des réseaux de capteurs sans fil "WSN " sont utilisés pour la capture des sons et des images et/ou des vidéos des animaux. Les CNN et le deep learning sont de plus en plus proposés pour la reconnaissance ou l'identification des objets ciblés par l'étude. L'avantage est la précision et la vitesse de reconnaissance.

L'objectif de ce travail est de faire les premiers pas dans ce domaine. Un réseau CNN est à utiliser pour la reconnaissance de différentes catégories d'oiseaux sédentaires ou migrateurs existants en Algérie. Ce sont d'excellents indicateurs de l'état de l'écosystème et des sentinelles de la nature. Les oiseaux sont de plus en plus menacés d'extinction surtout après les vagues d'incendies qui ont ravagé le pays. Le réseau choisi doit être analysé et adapté à l'étude. Des tests doivent être menés pour évaluer la précision de la reconnaissance. La deuxième phase

Introduction générale

consiste en une analyse des paramètres du CNN, dits "hyper- paramètres " afin d'évaluer leurs influences sur la précision de reconnaissance du réseau.

Le mémoire est composé de trois chapitres. Dans le premier chapitre, des généralités et des notions de base sur l'intelligence artificielle "AI", l'apprentissage machine "machine learning" et l'apprentissage profond "deep learning" sont présentées, avec plus de détails sur les fonctionnalités des couches de CNN.

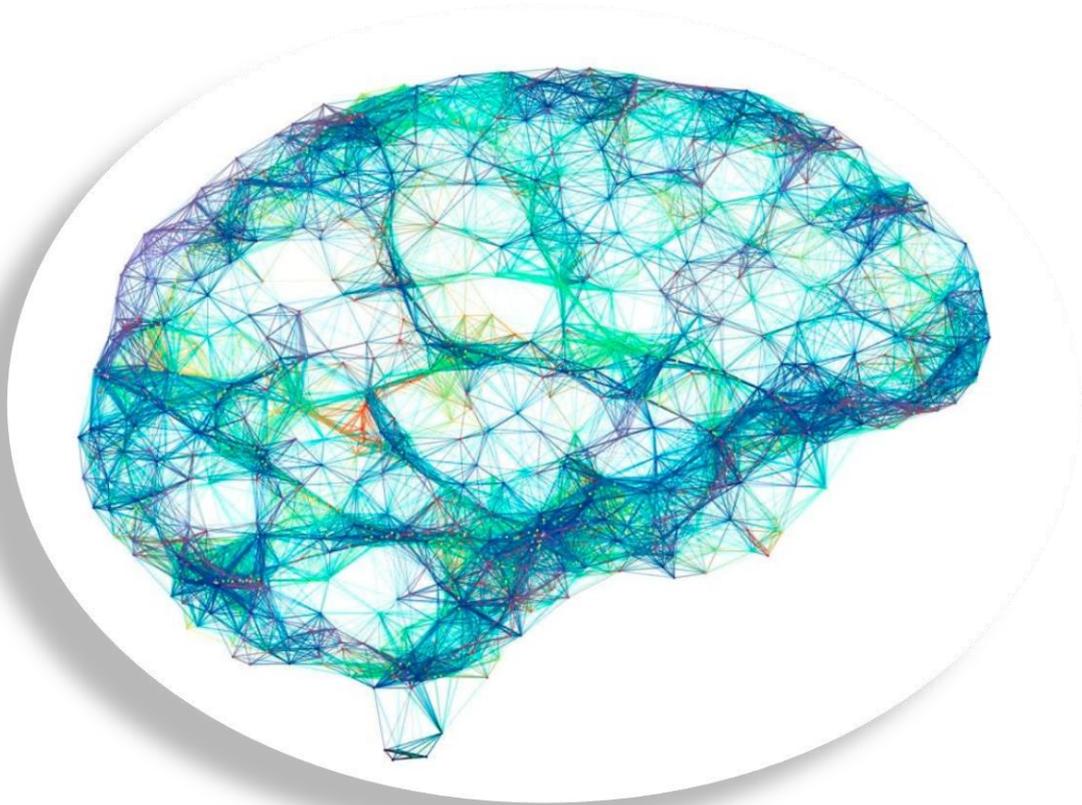
Le deuxième chapitre est consacré à une présentation des architectures des réseaux CNN les plus connus tels que Alex-Net, VGG-16 et Google-Net. Les bases de données utilisées dans l'entraînement de ces réseaux sont aussi exposées avec leurs historiques.

Dans le troisième chapitre, la partie expérimentale de notre travail est exposée en détails et les différents résultats analysés et discutés.

Le mémoire se termine par une conclusion générale et des perspectives.

Chapitre I :

IA, ML, DL, Réseaux de neurones Convolutifs.



I.1 Introduction :

Dans ce chapitre, nous donnons des définitions de l'intelligence artificielle "AI", de l'apprentissage machine "ML" et de l'apprentissage profond "Deep Learning". Le fonctionnement d'un réseau de neurones convolutifs "CNN" et les principales couches qui le composent sont aussi présentés.

I.2 Intelligence artificielle :

C'est un domaine informatique qui sert à créer des machines intelligentes et inspiré du cerveau humain, Il englobe l'apprentissage automatique et l'apprentissage profond mais il inclut aussi des approches qui n'impliquent aucun apprentissage.



Figure I.1 L'intelligence artificielle.

I.3 Apprentissage automatique (Machine Learning) :

Le Machine Learning ou apprentissage automatique est un domaine scientifique, et plus précisément une sous-catégorie de l'intelligence artificielle. Elle consiste à permettre des algorithmes découvrir des "patterns", à savoir des motifs récurrents, dans les ensembles de données. Ces données peuvent être des chiffres, des mots, des images, des statistiques

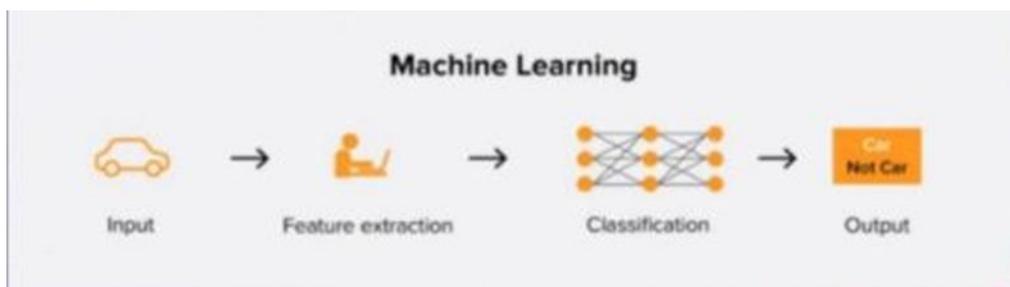


Figure I.2 Machine Learning.

I.4 L'apprentissage profond (Deep Learning) :

L'apprentissage profond ou "deep learning" fait beaucoup parler de lui ces dernières années. Et pour cause, ce sous ensemble de l'apprentissage machine "machine learning" s'est imposé de manière impressionnante dans plusieurs champs de recherche : reconnaissance faciale, synthèse vocale, traduction automatique, et bien d'autres [1].

Un modèle de Deep Learning a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minimale [2].

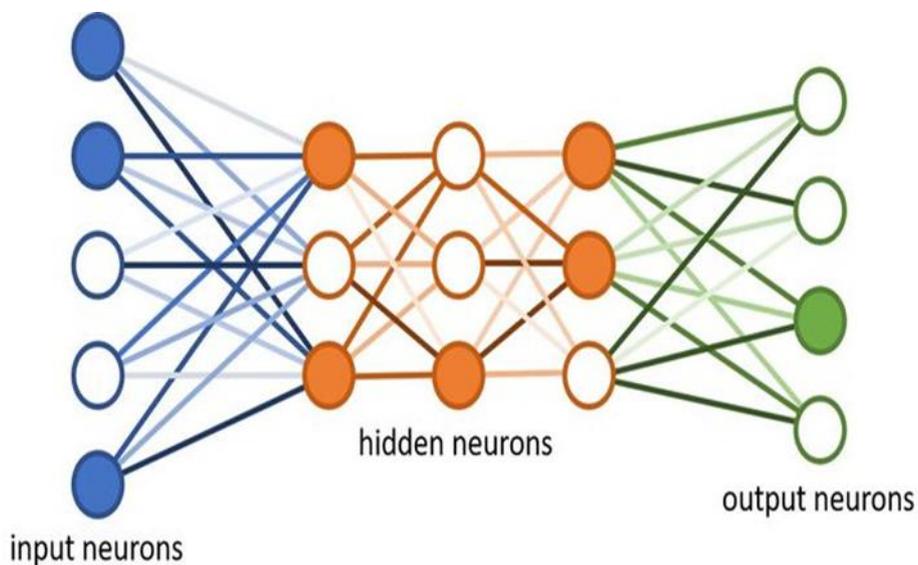


Figure I.3 Schéma d'un réseau de neurone profond.

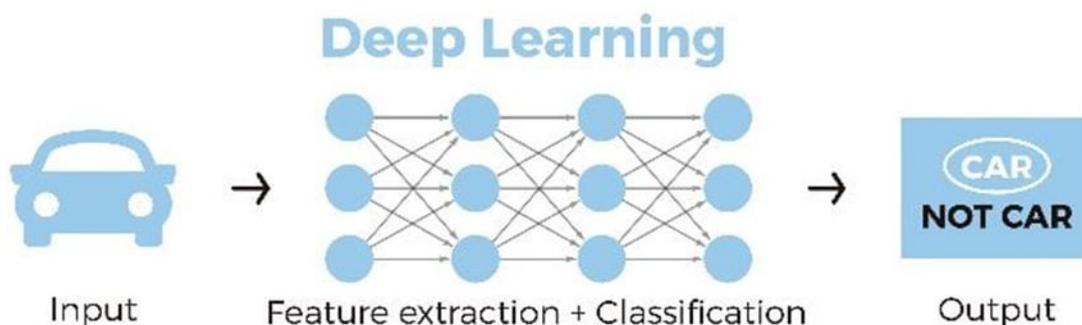


Figure I.4 Exemple d'application d'un réseau de neurone profond.

I.4.1. Fonctionnement du deep learning :

Les neurones sont regroupés en trois types de couches différents :

- **Couche d'entrée** : reçoit des données d'entrée et transmet les entées a la première couche masquée.
- **Couches cachées** : effectuent des calculs mathématiques sur nos entrées. L'un des défis de la création de réseaux neuronaux est de déterminer le nombre de couches cachées, ainsi que le nombre de neurones pour chaque couche.
- **Couche de sortie** : renvoie les données de sortie. Chaque connexion entre neurones est associée à un poids. Ce poids détermine l'importance de la valeur d'entrée. Les poids initiaux sont définis aléatoirement.

Chaque neurone a une fonction d'activation. Ces fonctions sont difficiles à comprendre sans raisonnement mathématique.

En termes simples, l'un de ses objectifs est de « normaliser » la sortie du neurone. Une fois qu'un ensemble de données d'entrée est passé à travers toutes les couches du réseau neuronal, il renvoie les données de sortie via la couche de sortie.

Pour former un réseau neuronal il faut :

- Un grand ensemble de données.
- Une grande quantité de puissance de calcul.

Pour former l'intelligence artificiel "IA", on lui donne les informations de l'ensemble de données et comparer ses résultats avec les résultats de l'ensemble de données. Puisque l'IA n'est toujours pas entraînée, ses sorties seront fausses.

Une fois que l'ensemble des données est parcouru, on peut créer une fonction qui montre à quel point les sorties de l'IA étaient erronées par rapport aux sorties réelles. Cette fonction s'appelle la fonction de coût. Elle que cette fonction soit nulle. C'est à ce moment que les sorties sont identiques aux sorties du jeu de données [3].

Voici une figure qui nous montre la relation entre les trois (3) concepts cités, Intelligence Artificielle "AI", Machine Learning "ML" et Deep Learning [4].

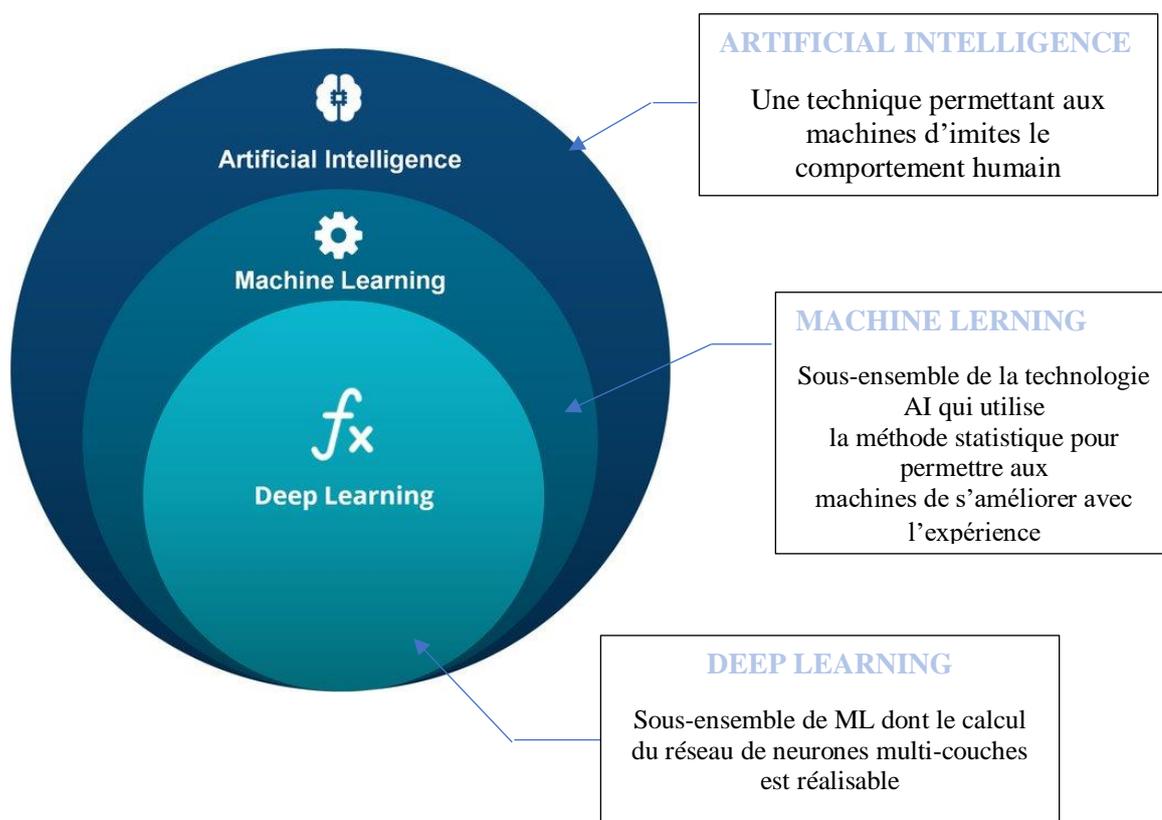


Figure I.5 La relation entre l'IA, ML et le Deep Learning.

I.5. Réseaux de neurones convolutifs (Convolution Neural Network ou CNN) :

Les réseaux de neurones convolutifs (CNN ou ConvNets) sont un groupe populaire de réseaux de neurones qui appartient à une famille plus large de méthodes intitulé par l'apprentissage en profondeur [5]. Ce qui s'est avéré efficace dans des domaines tels que la classification d'images et de vidéos et de la reconnaissance des visages, des objets, des feux de

circulation et des voitures autonomes. Récemment, les CNN ont été efficaces dans de nombreuses tâches de traitement du langage naturel (telles que la catégorisation des phrases) [6].

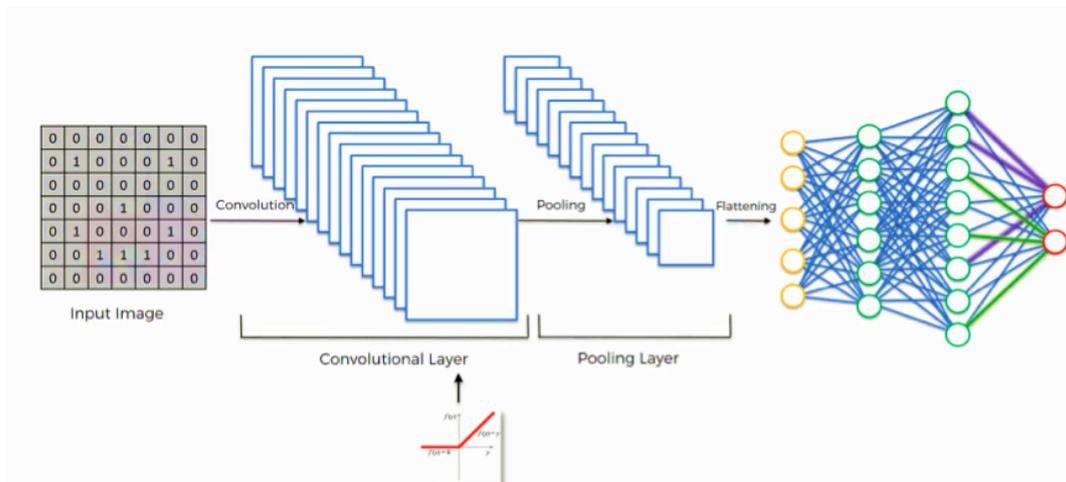


Figure I.6 Architecture d'un réseau de neurones convolutif.

Ce type de réseau est constitué de trois types de couches qui sont la couche de convolution liée souvent avec la fonction d'activation ReLu, la couche de pooling et la couche entièrement connectée. Ces couches effectuent des opérations sur les données dans le but d'apprendre des caractéristiques des données [7].

I.5.1. Architecture globale des CNN :

Les CNN ont plusieurs couches qui traitent et extraient les caractéristiques des données (voir figure I.7).

- Couche d'entrée (Input layer).
- Couche de convolution.
- Couche de correction (ReLU).
- Couche de Pooling.
- Couche entièrement connectée (Fully connected layer).
- Couche Softmax/logistique.
- Couche de sortie.

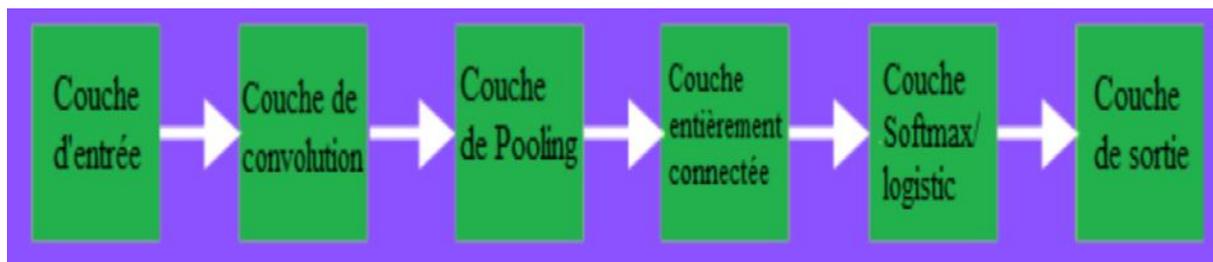


Figure I.7 Les couches d'un CNN.

I.5.1.1. Couche d'entrée du CNN :

La couche d'entrée d'un CNN contient des données d'image. Les données d'image sont représentées par une matrice tridimensionnelle si elle est en couleur (R.G.B) et les données d'image sont représentées par une matrice à une dimension si elles sont en noir et blanc.

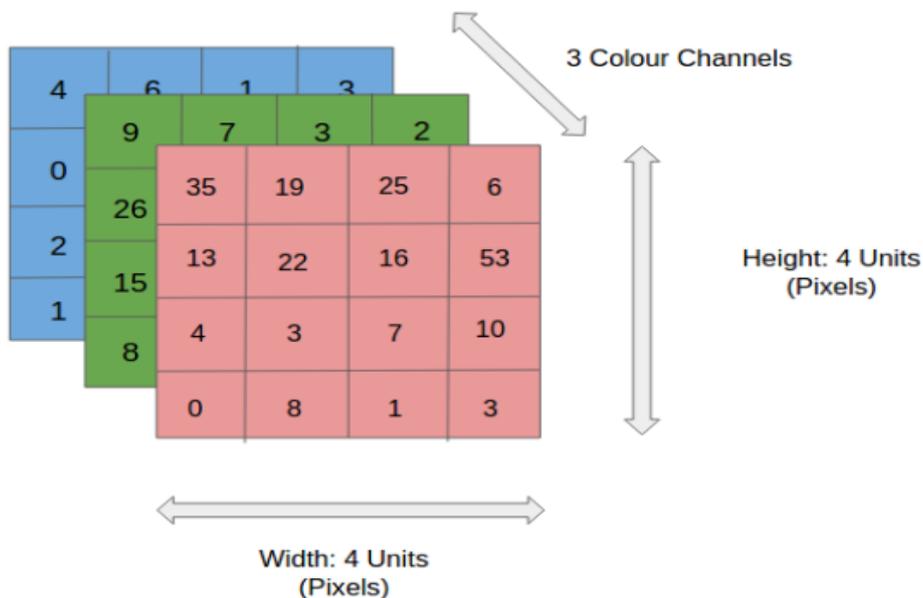


Figure I.8 Images RGB composées de trois plans.

I.5.1.2. Couche de convolution :

La convolution est la partie la plus importante de CNN, dont le rôle est d'analyser les images fournies en entrée et de détecter la présence d'un ensemble de caractéristiques (features) [8]. L'opération est basée sur un filtrage par convolution : le principe est de faire "glisser" le filtre sur l'image, et de calculer le produit de convolution entre le filtre et chaque portion de l'image balayée [9].

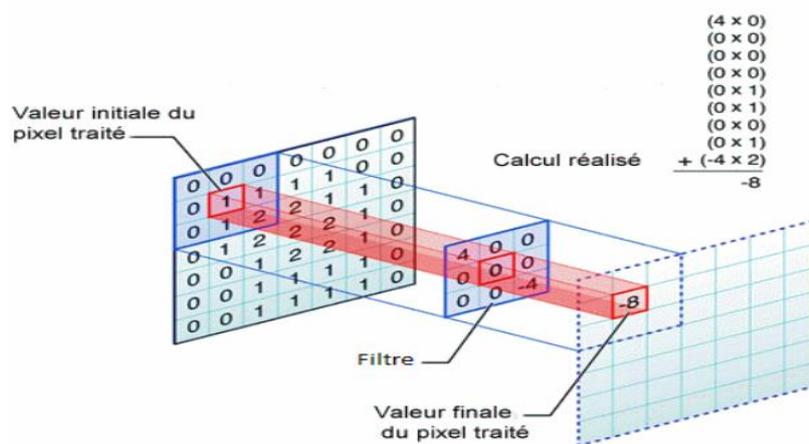


Figure I.9 Exemple de convolution avec un filtre 3 x 3.

I.5.1.3. Couche de correction (ReLU: Rectified Linear Units) :

Est une couche qui désigne la fonction réelle non-linéaire définie par :

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

La couche de correction ReLu remplace donc toutes les valeurs négatives reçues en entrées pas des zéros. Elle joue le rôle de fonction d'activation [10].

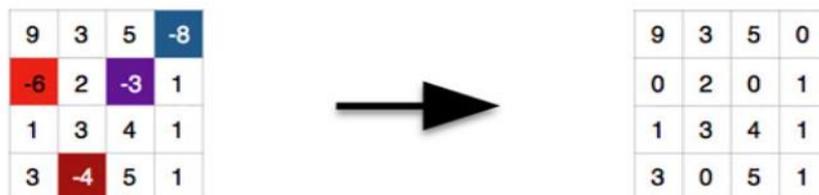


Figure I.10 Exemple de la fonction ReLu.

I.5.1.4. Couche de Pooling :

La couche de pooling est souvent placée entre deux couches de convolution. Elle reçoit en entrée les features maps obtenues de la couche de convolution et son rôle est de réduire la taille des images. Elle permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur apprentissage.

L'opération consiste à découper l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. En pratique, et dans la plupart du temps on utilise souvent des cellules carrées de petite taille pour ne pas prendre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille 2*2 pixels qui ne se chevauchent pas ou des

cellules de taille 3*3 pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc) [10].

I.5.1.5. Couche entièrement connectée (Fully connected layer):

Ces couches sont constituées toujours la dernière couche d'un réseau de neurones et sont entièrement connectées à tous les neurones de sorties. Après avoir reçu un vecteur en entrée, cette couche applique successivement une combinaison linéaire puis une fonction d'activation dans le but final est de classifier l'image d'entrée, elle renvoie enfin en sortie un vecteur de taille correspondant au nombre de classes dans lequel chaque composante représente la probabilité pour l'image d'entrée d'appartenir à une classe [8].

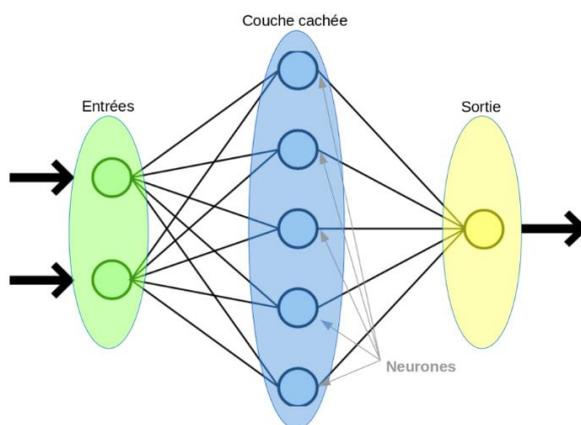


Figure I.11 Réseau entièrement connecté.

I.5.1.6. Couche Softmax/logistique :

La fonction Softmax est souvent utilisée dans la dernière couche d'un modèle de classification, elle permet de convertir un vecteur réel en un vecteur de probabilité [11].

I.5.1.7. Couche de sortie (Output layer) :

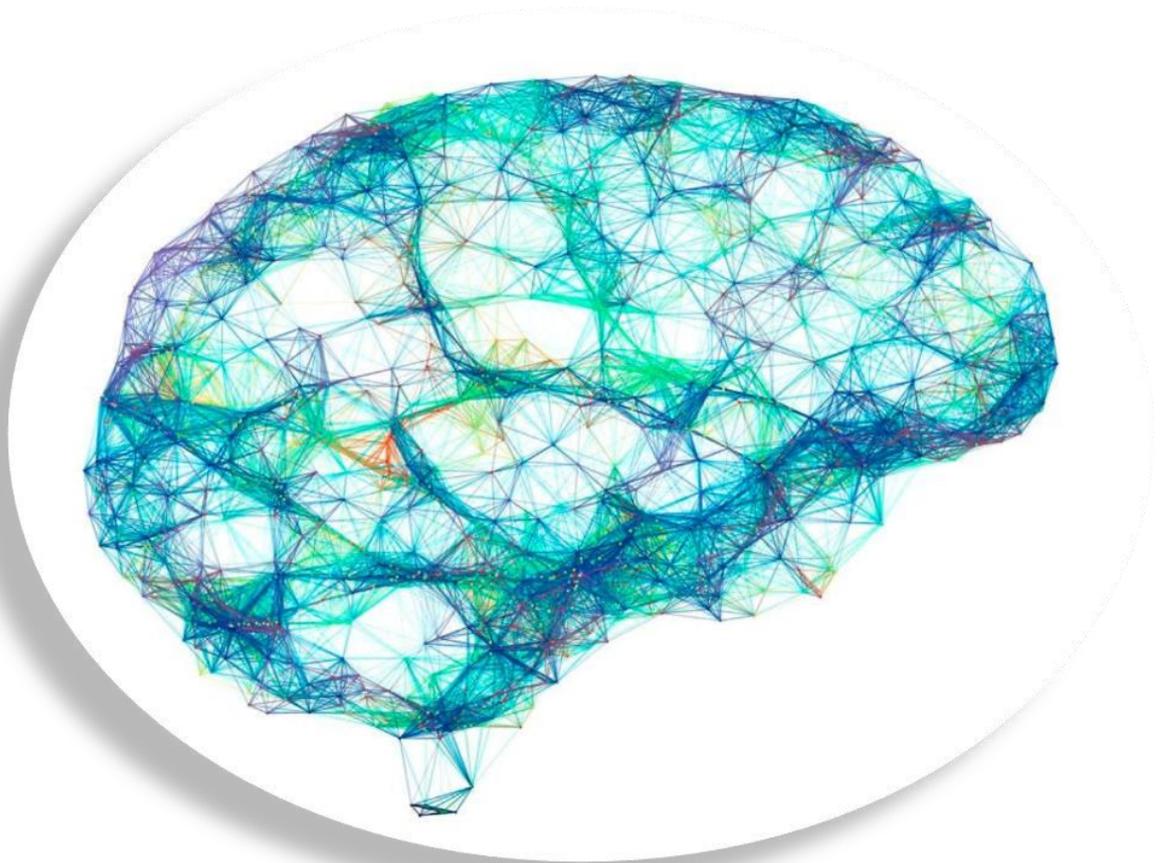
La couche de sortie contient l'étiquette qui se présente sous la forme d'un codage.

I.6 Conclusion :

Dans ce chapitre, on a couvert le concept d'apprentissage de profondeur au niveau fondamental spécifiquement tous de définition, architecture, etc. Avec un aperçu général de l'IA et ses sous-catégories, ainsi qu'une approche vers notre sujet principal le CNN détaillé pour le savoir.

Chapitre II :

Architectures des CNN et Base de données



II.1 Introduction :

Quand on parle de CNN, on sait qu'on a besoin d'une des architectures et d'un jeu de données, pour bien choisir il faut les connaître, pour cela voici quelques-unes des architectures et des jeux de données les plus utilisés avec lesquels travailler.

II.2 Les architectures des CNN :

Il est toujours évident que L'histoire de développement et le chemin de succès qui a pris les CNNs a commencée en 1998 lorsque Yann LeCun et ses co-auteurs ont publié le papier « Gradient-Based Learning Applied to Document Recognition » (Lecun et al., 1998). 20ans après cette innovation, les CNNs ont montré leur progrès [12]. Durant cette période, plusieurs architectures ont été lancées. Nous donnerons ici les architectures les plus utilisées et répondues dans le domaine de Classification des images.

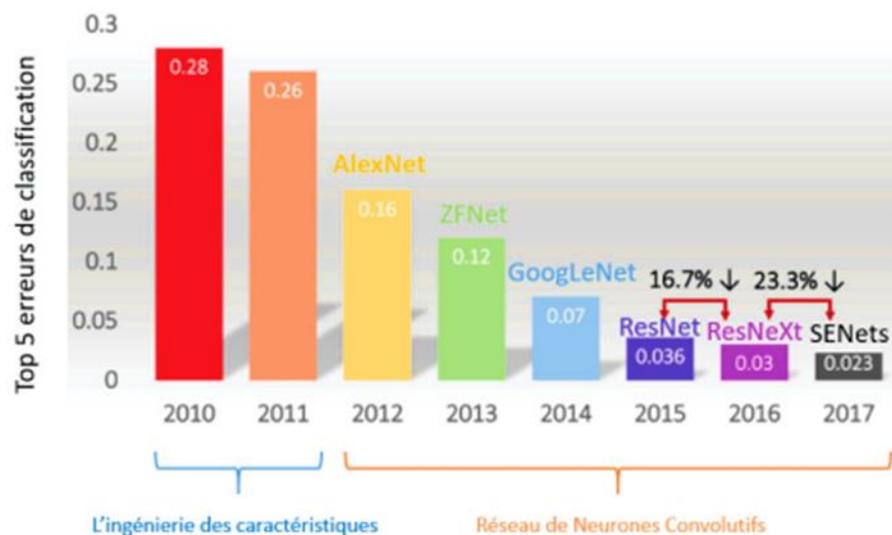


Figure II.1 L'erreur top-5 en % des architectures gagnantes de la compétition ILSVRC.

II.2.1 Alex-Net :

Alex-Net a été proposé par Alex Krizhevsky durant le défi Image-Net Large-Scale Visual Recognition Challenge (ILSVRC) de 2012 et remporte le premier prix [13]. Il résout le problème de la classification des images où l'entrée est une image de l'une des 1000 classes différentes (par exemple, chats, chiens, etc.) et la sortie est un vecteur de 1000 éléments.

Alex-Net est constitué de 5 couches convolutives, suivies de 3 couches entièrement connectées et utilise ReLu pour la partie non linéaire.

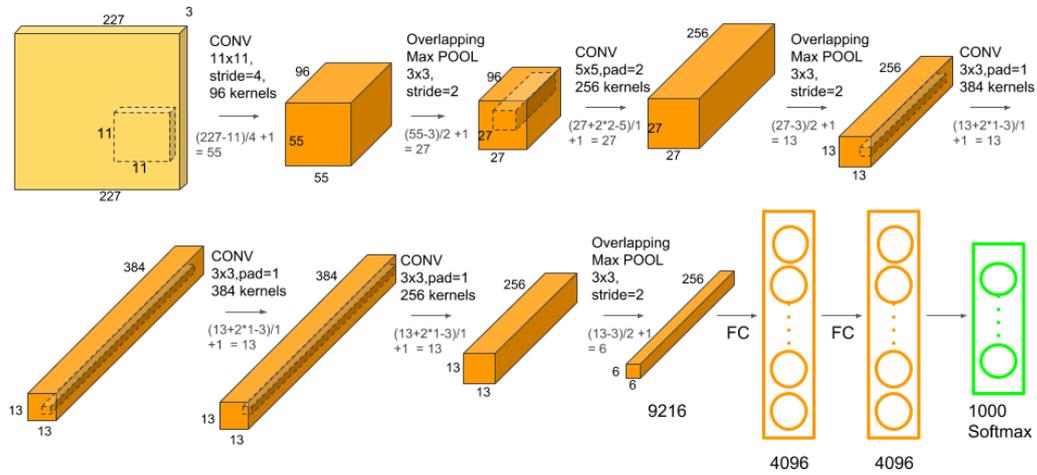


Figure II.2 Visualisation de l'architecture du réseau Alex-Net [14].

II.2.2 Zef-Net :

Le Zef-Net a été proposé par Zeiler et Fergus en 2013. Il n'est pas surprenant qu'il a Remporté l'ILSVRC en 2013 avec le top 5 des taux d'erreur de 11,2%. Ce modèle n'a été Entraîné que sur 1,3 million d'images, contre 15 millions d'images pour Alex-Net. Ce modèle a été entraîné sur le GPU Nvidia Geforce GTX 580 mais pendant 12 jours [15].

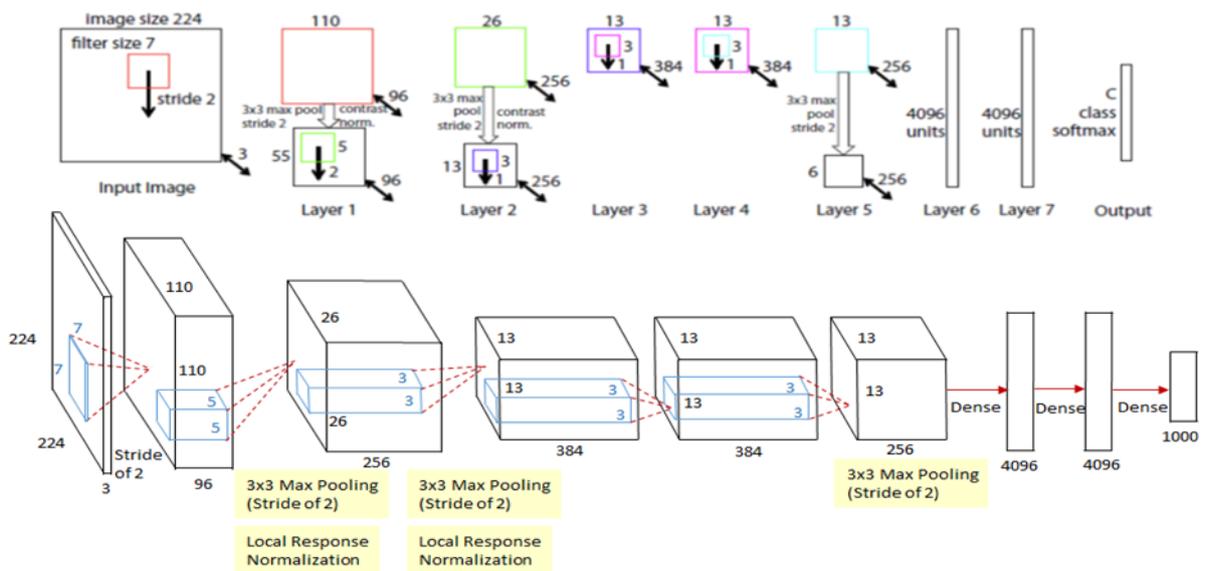


Figure II.3 Visualisation de l'architecture du réseau Zef-Net [16].

II.2.3 VGG-16 :

VGG-16 est un modèle atteint une précision de 92,7% dans le top 5 des tests dans Image-Net, qui est un ensemble des données de plus de 14 millions d'images appartenant à 1000 classes. C'était l'un des célèbres modèles soumis à la conférence ILSVRC- 2014 [17].

VGG-16 est constitué de 16 couches (13 couches enveloppantes et 3 couches entièrement connectées).

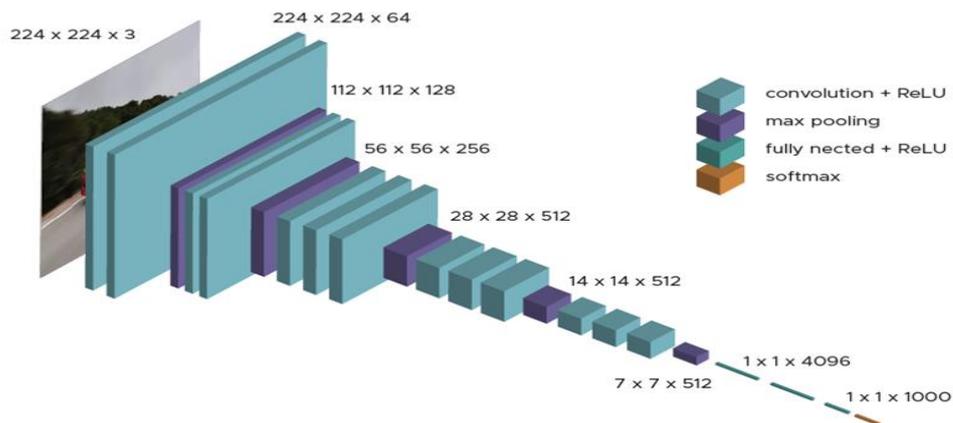


Figure II.4 Visualisation de l'architecture du réseau VGG-16[18].

II.2.4 Google-Net / Inception-V1 :

Google-Net/Inception-V1 (2014) le gagnant du concours ILSVRC 2014 était Google-Net (alias Inception V1) de Google. Il a atteint un taux d'erreur de 6,67 % dans le top 5. Le réseau utilise un réseau CNN inspiré par Le-Net. La principale caractéristique de cette architecture est la meilleure utilisation des ressources informatiques à l'intérieur du réseau.

L'architecture de Google-Net consistait en un réseau CNN profond de 22 couches, mais réduisait le nombre de paramètres de 60 millions (Alex-Net) à 4 millions [19].

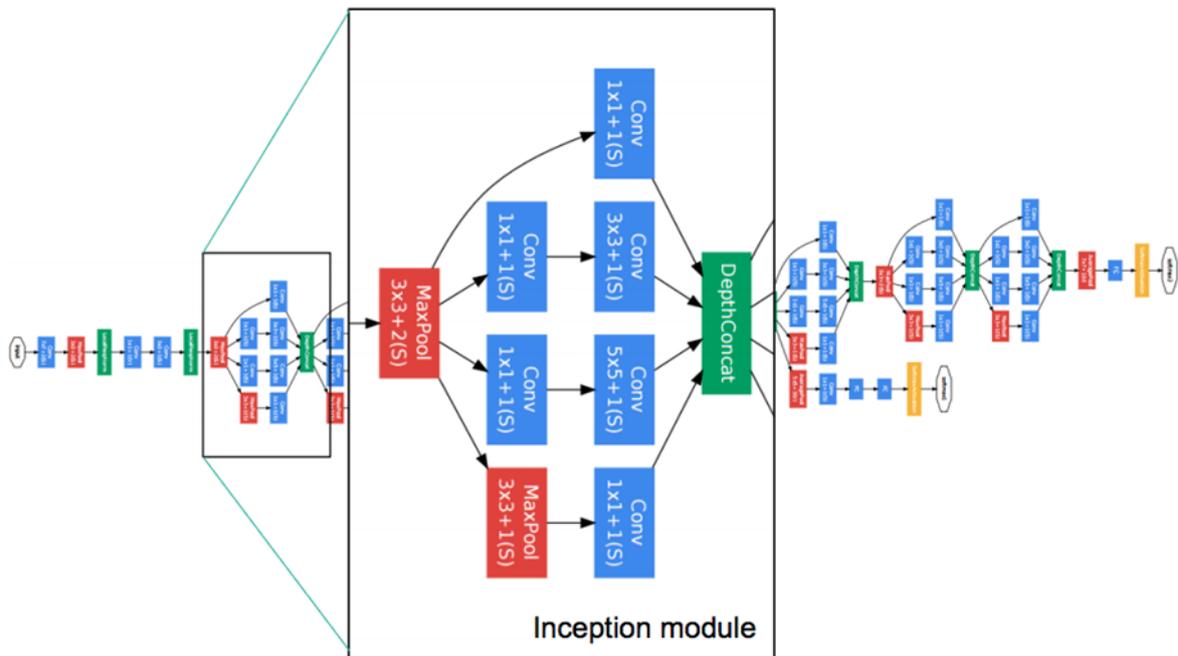


Figure II.5 Visualisation de l'architecture du réseau Google-Net [19].

II.2.5 Res-Net :

C'est l'un des réseaux de neurones profonds les plus puissants qui a obtenu des excellents résultats de performances dans le défi de classification ILSVRC 2015. Res-Net a réalisé d'excellentes performances de généralisation sur d'autres tâches de reconnaissance et a Remporté la première place sur la détection Image-Net, la localisation Image-Net, la détection COCO et la segmentation COCO dans les concours ILSVRC et COCO 2015.

Il existe de nombreuses variables de l'architecture Res-Net, c'est-à-dire le même concept mais avec un nombre de couches différent. Nous avons **Res-Net-18**, **Res-Net-34**, **Res-Net-50**, **Res-Net-101**, **Res-Net-110**, **Res-Net-152**, **Res-Net-164**, **Res-Net-1202**[19].

Tableau II.1 les architecture des CNN [21].

N°	Nom de l'architecture	Année	Contribution principale	Paramètres	Taux d'erreur	Profondeur
1	Le-Net	1998	- Première architecture CNN populaire	60.000	MNIST : 0.95	5
2	Alex-Net	2012	- Plus profond et plus large que Le-Net - Remporte la compétition ILSVRC en 2012. - Utilise Relu, dropout et chevauchement Pooling - GPU NVIDIA GTX 580	60 M	Image-Net : 16.4	8
3	Zef-Net	2014	- Visualisation des couches intermédiaires	60 M	Image-Net : 11.7	8
4	VGG	2014	- Réseau plus profond (environ deux fois plus profond qu'Alex-Net) - Utilise des noyaux de petite taille.	138 M	Image-Net : 7.3	19
5	Google-Net (Inception-V1)	2015	- Vainqueur du concours ILSVRC en 2014. - Introduire le bloc inception.	4 M	Image-Net : 6.7	22

			- Coût de calcul réduit. -Grande précision.			
10	Res-Net	2015	- Remporte le concours ILSVRC 2015. - Residual learning - Identity mapping-based skip connections	25.6 M 1.7 M	Image-Net : 3.6 CIFAR-10 : 6.43	152 110
14	Xception	2017	- Convolution en profondeur suivie d'une convolution par points convolution par points	22.8 M	Image-Net : 0.055	126

II.3 Les bases de données :

II.3.1 MNIST :

MNIST est un ensemble de données bien connu utilisé en vision par ordinateur qui a été construit par Yann Le Cun et Al. Ou elle constituée de 70 000 chiffres manuscrits (7000 images par chiffre de 0 à 9), dont 60 000 pour l'entraînement des données et 10 000 pour le test. Chaque image a 28 x 28 pixels et est traitée par les algorithmes d'apprentissage comme un vecteur de $784 = 28 \times 28$ variables. La figure II.7 est un échantillon de 64 images tirées au hasard sur la base MNIST [22].

Il permet de reconnaître les chiffres de la base test suite à l'entraînement des données [22].



Figure II.7 Echantillon de la base MNIST.

II.3.2 Image-Net :

Image-Net est un ensemble de données qui comporte de plus de 15 millions d'images étiquetées appartenant à environ 22 000 classes. ILSVRC utilise un sous-ensemble d'Image-Net avec environ 1000 images dans chacune des 1000 catégories. Au total, il existe environ 1,2 million d'images d'entraînement, 50 000 images de validation et 150 000 images de test [23].

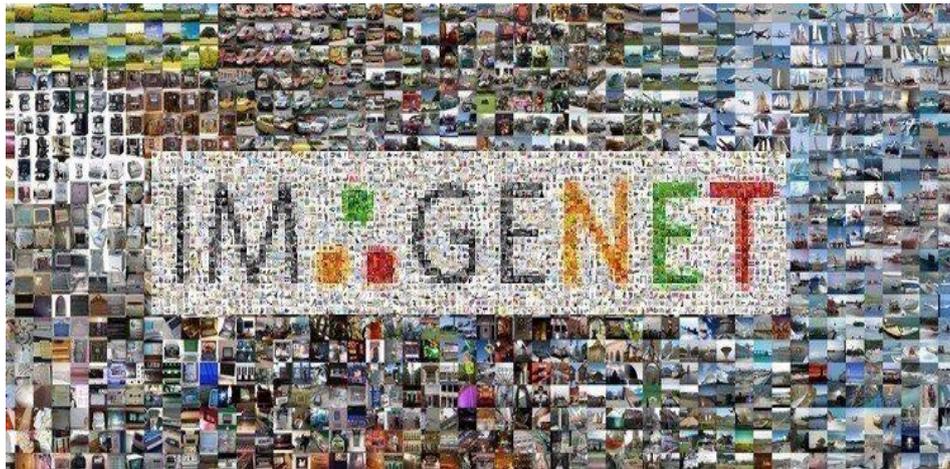


Figure II.8 : Image-Net.

II.3.3 CIFAR-10 :

L'ensemble de données CIFAR-10 (Institut canadien de recherches avancées, 10 classes) est un sous-ensemble de l'ensemble de données Tiny Images et se compose de 60 000 images couleur 32x32. Les images sont étiquetées avec l'une des 10 classes mutuellement exclusives : avion, automobile (mais pas camion ou camionnette), oiseau, chat, cerf, chien, grenouille, cheval, bateau et camion (mais pas camionnette). Il y a 6000 images par classe avec 5000 images de formation et 1000 images de test par classe [24].

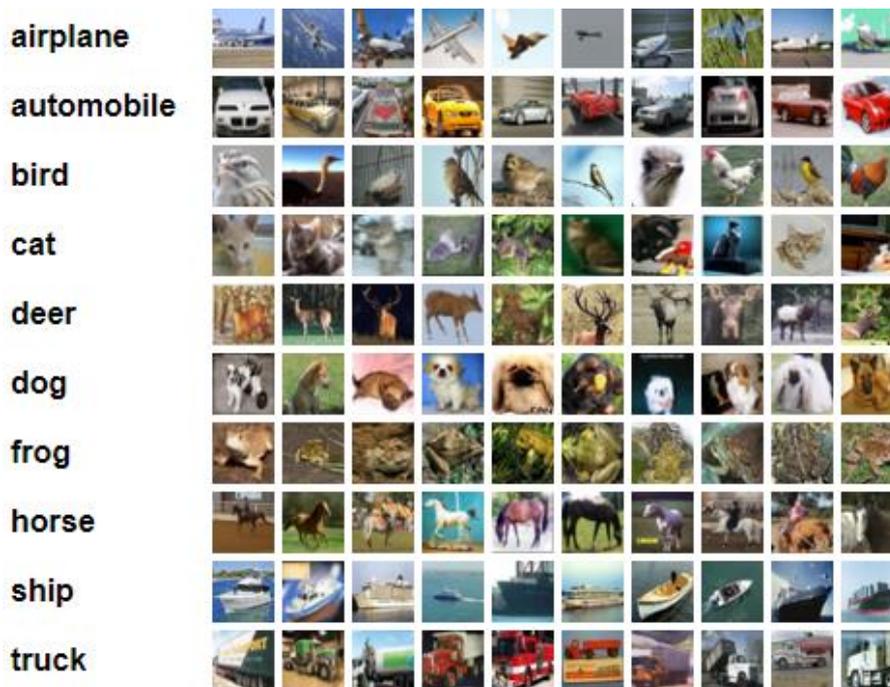


Figure II.9 Les 10 catégories de CIFAR-10 [15].

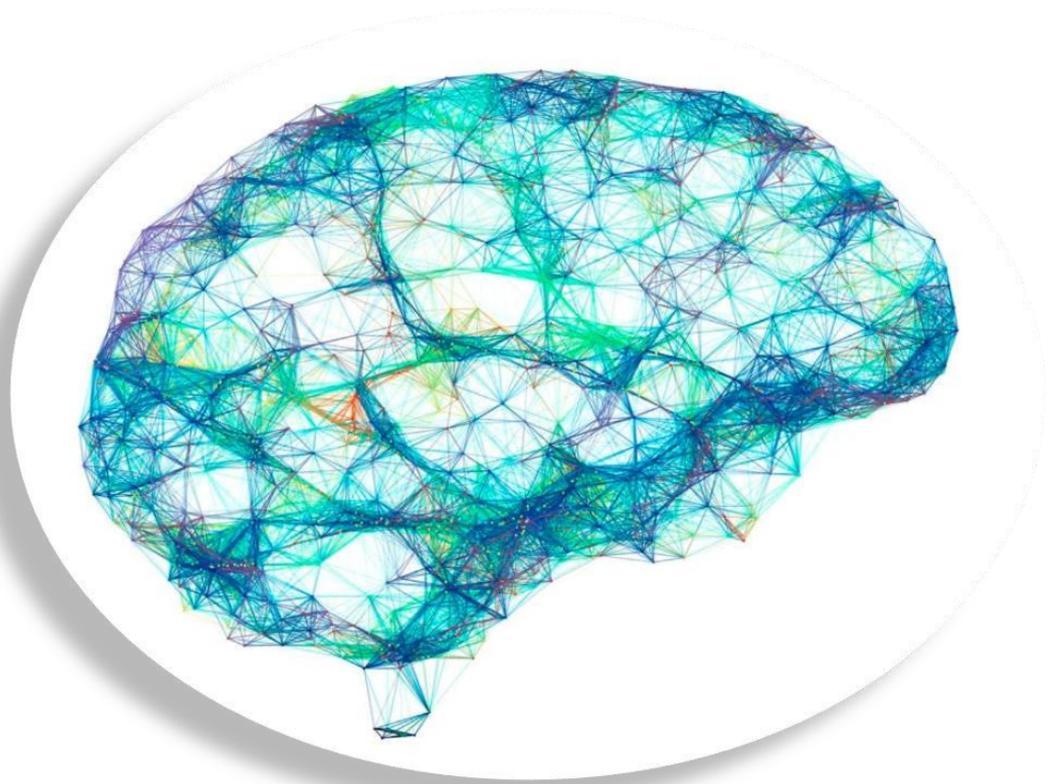
Les classes sont complètement mutuellement exclusives. Il n'y a pas de chevauchement entre les automobiles et les camions. "Automobile" comprend les berlines, les VUS, les choses de ce genre. "Camion" ne comprend que les gros camions. Ni l'un ni l'autre n'inclut les camionnettes [25].

II.4 Conclusion :

Dans ce chapitre on a concentré sur la présentation de l'architecture de CNN et les bases de données ainsi que donnant leur historique et caractéristique (couches) avec réorientation de chaque architecture.

Chapitre III :

Méthodologie, résultats et discussion



III.1 Introduction :

Les oiseaux jouent un rôle très important dans la protection de l'écosystème et sont parmi d'autres espèces végétales et animales menacés d'extinction. Les scientifiques cherchent à exploiter les progrès en télécommunications pour les surveiller par l'utilisation des réseaux de capteurs sans fil dans la forêt pour capturer des sons/images/vidéos des oiseaux et les transmettre pour analyse. Cette opération est assurée par des CNN.

Dans ce chapitre, nous allons prendre un réseau CNN pré entraîné « Google-Net », l'adapter et l'entraîner pour l'utiliser dans la reconnaissance des images de 10 catégories d'oiseaux existants en Algérie. La précision et le temps de traitement seront mesurés. Les hyper paramètres qui influencent les résultats seront aussi examinés.

III.2 Langage de programmation :

Les simulations et tests ont été menés en utilisant l'environnement de développement Matlab version **R2020b** sur un PC avec les caractéristiques ci-dessous :

Processor: intel® core™ i3-2350 CPU @ 2.30GHz 2.30GHz.

RAM: 4.00 Go.

Carter graphique : intégrée.

OS : Windows 7 64Bits.

MATALAB (**MA**Tri**X****LAB**oratory) est un langage de programmation de quatrième génération et un environnement d'analyse numérique. Développé par la société THE MathWorks, Il est très puissant, permet les calculs matriciels complexes, les tracés des courbes, le développement d'algorithmes et la création des IU (**I**nter**f**aces **U**tilisateurs). Il permet aussi aux développeurs d'interagir avec des programmes développés dans différents langages.

III.3 Les Différentes façons d'entraîner un réseau de CNN :

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

- ✚ Créer et entraîner un réseau en partant de zéro : ceci est très difficile et nécessite beaucoup d'expérience et l'utilisation d'une très grande base de données formée de dizaines ou des centaines de milliers d'images, voire même des millions.

- ✚ Utiliser un transfert d'apprentissage "Transfer learning" : dans ce cas, un CNN déjà entraîné est utilisé pour la classification d'images même si leur nombre est relativement petit. Néanmoins, il serait nécessaire d'adapter le CNN utilisé aux nouvelles données.
 - ✚ La segmentation sémantique "semantic segmentation" où un réseau est utilisé pour identifier différents objets dans une image. C'est l'une des dernières avancées des CNN.
- Nous avons opté pour la deuxième option en raison de notre expérience limitée dans le domaine ainsi que de la taille de la base de données que nous avons créée.

III.4 Transfert d'apprentissage :

Le but de transférer l'apprentissage est de prendre un CNN pré-entraîné et l'utiliser comme point de départ pour apprendre une nouvelle tâche. Il est couramment utilisé dans les applications d'apprentissage en profondeur. Au lieu de former un réseau à partir de zéro, l'utilisation d'un réseau avec l'apprentissage par transfert est généralement beaucoup plus rapide et plus facile.

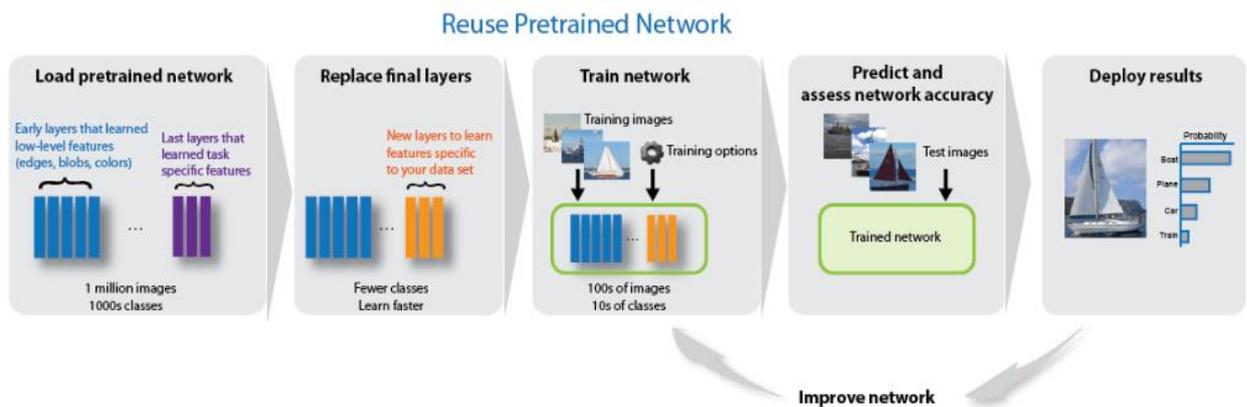


Figure III.1 Etapes d'utilisation d'un réseau pré-entraîné.

Dans la première phase, il est nécessaire d'importer un CNN. Nous avons choisi Google-Net dont les caractéristiques sont les suivantes :

- La caractéristique principale de cette architecture réseau est d'optimiser l'utilisation des ressources informatiques au sein du réseau.
- L'architecture Google-Net se compose de 22 couches, et une partie de ces couches sont un total de 9 modules de Inception.

- La couche d'entrée de l'architecture Google-Net a été conçue pour des images de la dimension 224×224.
- Le réseau Google-Net travaille avec 1000 catégories.

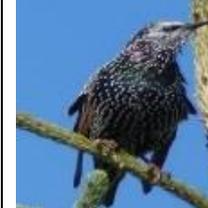
III.5 Méthodologie :

Pour mener les simulations, nous avons suivi les étapes suivantes :

III.5.1 Création d'une base de données :

Pour créer une base de données avec des images déjà existantes, nous avons cherché dans un site des images comme (Pinterest, weheartit,...etc) ou par Google Images. Les images choisies ont été enregistrées de format .jpg et de dimensions allant de 185×185 jusqu'à 1920×1281 dans un répertoire avec un sous-répertoire pour chaque catégorie d'images qui correspondent aux oiseaux sélectionnés pour ce travail. Pour notre base on a 10 catégories pour les 10 oiseaux suivants :

Tableau III.1 les 10 oiseaux de la Base de données.

				
Caille de blés سمان القمح	Chardonneret طائر الحسون	Cigogne Blanche اللقلق الأبيض	épervier d'Europe الباشق	Etourneau sansonnet زرزور
				
Hirondelle de fenêtre خطاف	Moineau عصفور	mouette rieuse نورس	Pigeon حمامة	Rouge Gorge أبو الحناء

III.5.2 Installation du réseau Google-Net sur MATLAB :

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

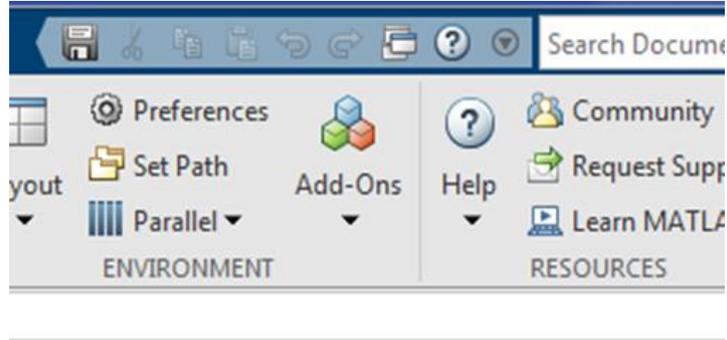


Figure III.2 Capture de liste d’outil HOME ENVIRONNEMENT de MATLAB.

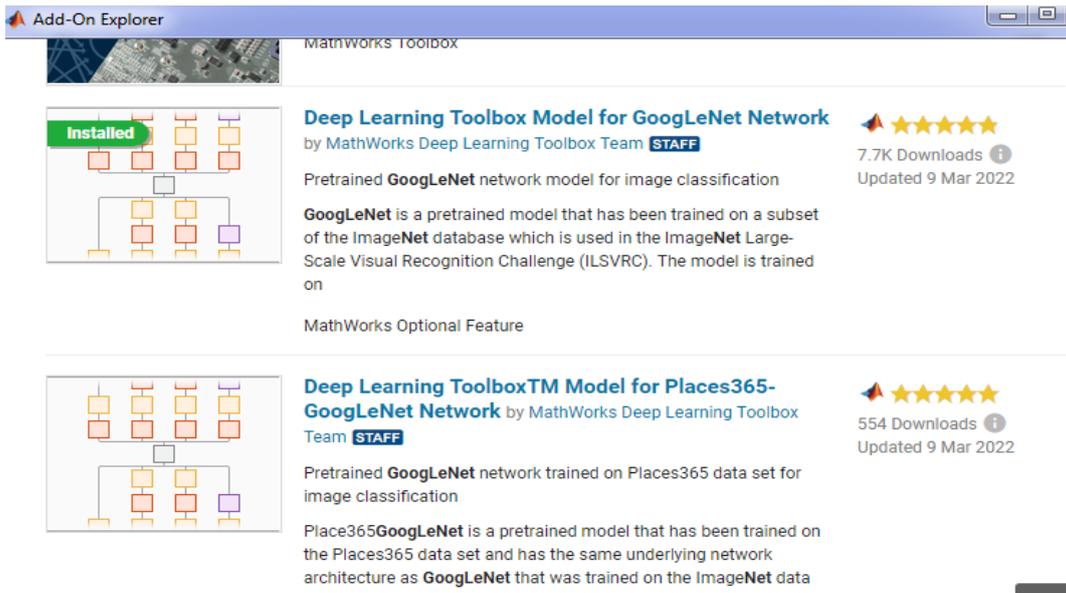


Figure III.3 Résultat de la recherche de Google-Net dans Add-Ons.

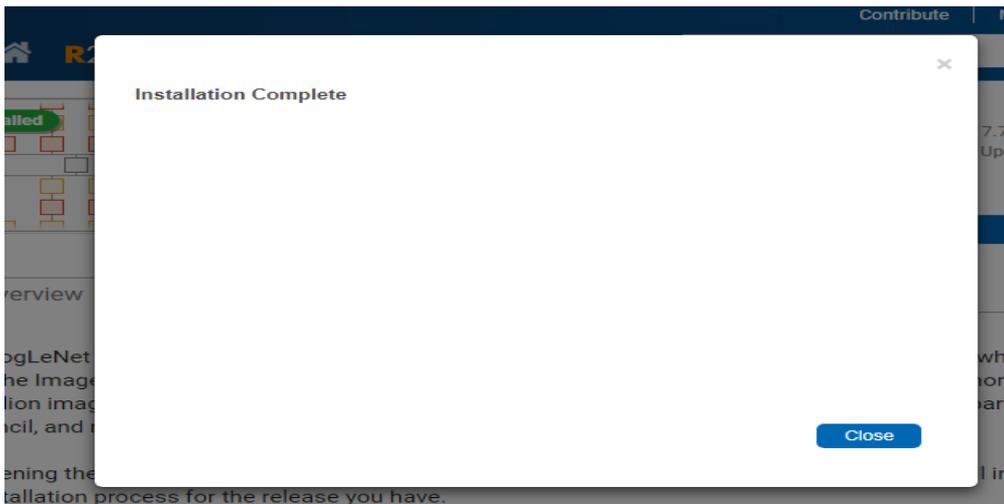


Figure III.4 Installation de Google-Net sur MATLAB.

- Vérification de l'installation de Google-Net sur MATLAB.

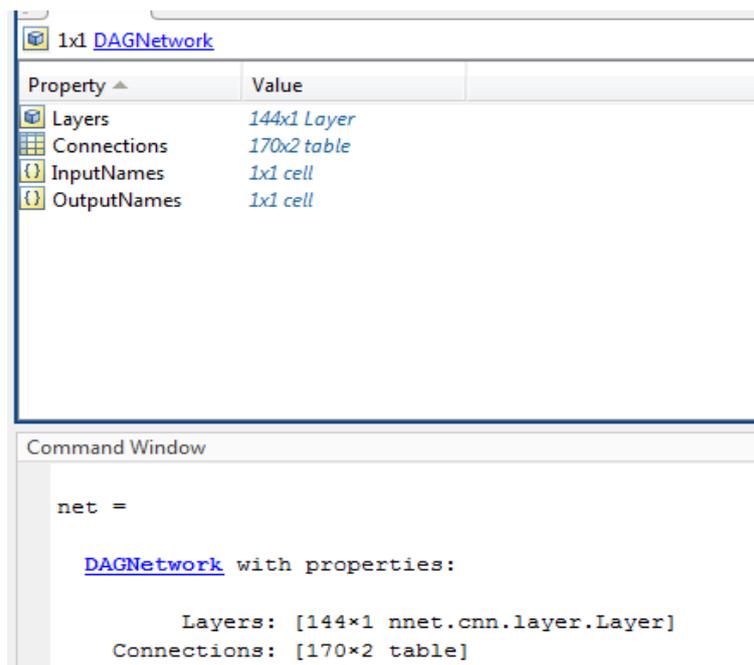


Figure III.5 Affichage des caractéristiques de Google-Net sur workspace et command Windows.

III.5.3 Chargement de la base de Données :

Après avoir installé le réseau Google-Net sur MATLAB, l'étape suivante consiste à charger la base de données. Dans MATLAB, la fonction recommandée pour gérer une grande collection d'images est « `imageDatastore()` ». Cette fonction nécessite comme paramètre, le chemin complet de l'emplacement du dossier d'images. Nous pouvons inclure les sous-dossiers et

spécifier les étiquettes des images à l'aide de cette fonction. Le code suivant montre comment « imageDatastore() » est utilisé:

```

2 - digitDatasetPath = fullfile('C:\Users\Zeid.Info\Desktop\Ma Base');
3 - dataset = imageDatastore(digitDatasetPath, ...
4     'IncludeSubfolders',true,'LabelSource','foldernames');
    
```

Figure III.6 Code pour Chargement de la base de Données.

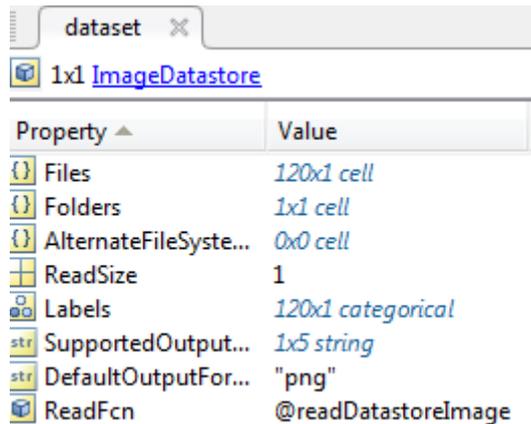


Figure III.7 L'affichage de dataset dans workspace.

III.5.4 Partition de la base de données :

Les images formant la base de données ont été partagées en deux sous-ensembles : un sous ensemble pour l'entraînement composé de 70% des images de la base de données et un deuxième contenant les 30% images restantes. Pour cela, on a utilisé le code suivant :

```

[Train_dataset,Valid_dataset] = splitEachLabel(dataset,0.7,'randomized');
    
```

Figure III.8 Code de partition de la base de données.

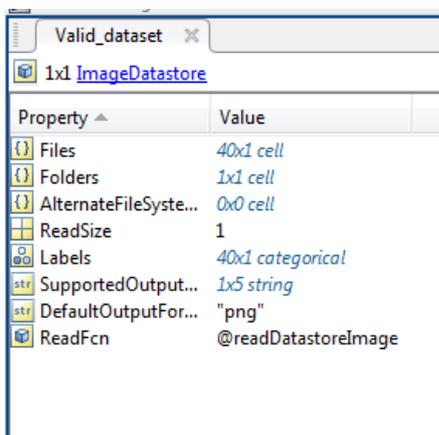


Figure III.9 Affichage de dataset dans workspace (30% Validation).

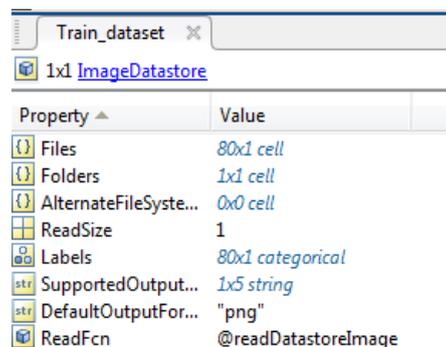


Figure III.10 Affichage de dataset dans workspace (70%Entrainement).

Après installation du réseau pré-entraîné dans MATLAB, on le charge dans la variable net :

```
Command Window
>> net.Layers

ans =

144x1 Layer array with layers:

   1  'data'                Image Input                224x224x3 images with 'zerocenter' normalization
   2  'conv1-7x7_s2'        Convolution                 64 7x7x3 convolutions with stride [2 2] and padding [3
   3  'conv1-relu_7x7'      ReLU                       ReLU
   4  'pool1-3x3_s2'        Max Pooling                 3x3 max pooling with stride [2 2] and padding [0 1 0]
```

Figure III.11 Affichage des couches du Google-Net.

```
Command Window
135 'inception_5b-relu_5x5'    ReLU                       ReLU
136 'inception_5b-pool'     Max Pooling                 3x3 max pooling with stride [1 1] and padding [1 1
137 'inception_5b-pool_proj' Convolution                 128 1x1x832 convolutions with stride [1 1] and paddin
138 'inception_5b-relu_pool_proj' ReLU                       ReLU
139 'inception_5b-output'    Depth concatenation         Depth concatenation of 4 inputs
140 'pool5-7x7_s1'          Global Average Pooling      Global average pooling
141 'pool5-drop_7x7_s1'     Dropout                     40% dropout
142 'birds Feature Learner'  Fully Connected             10 fully connected layer
143 'prob'                   Softmax                     softmax
144 'birds Classifier'      Classification Output       crossentropyex with 'Caille de blés' and 9 other class
```

Figure III.12 Suite des couches de Google-Net.

A l'aide du code suivant, on va visualiser la structure du réseau Google-Net. Ceci est très important pour que nous puissions modifier notre réseau pré-entraîné :

```
analyzeNetwork(net)
```

Figure III.13 Code pour afficher l'architecture de Google-Net.

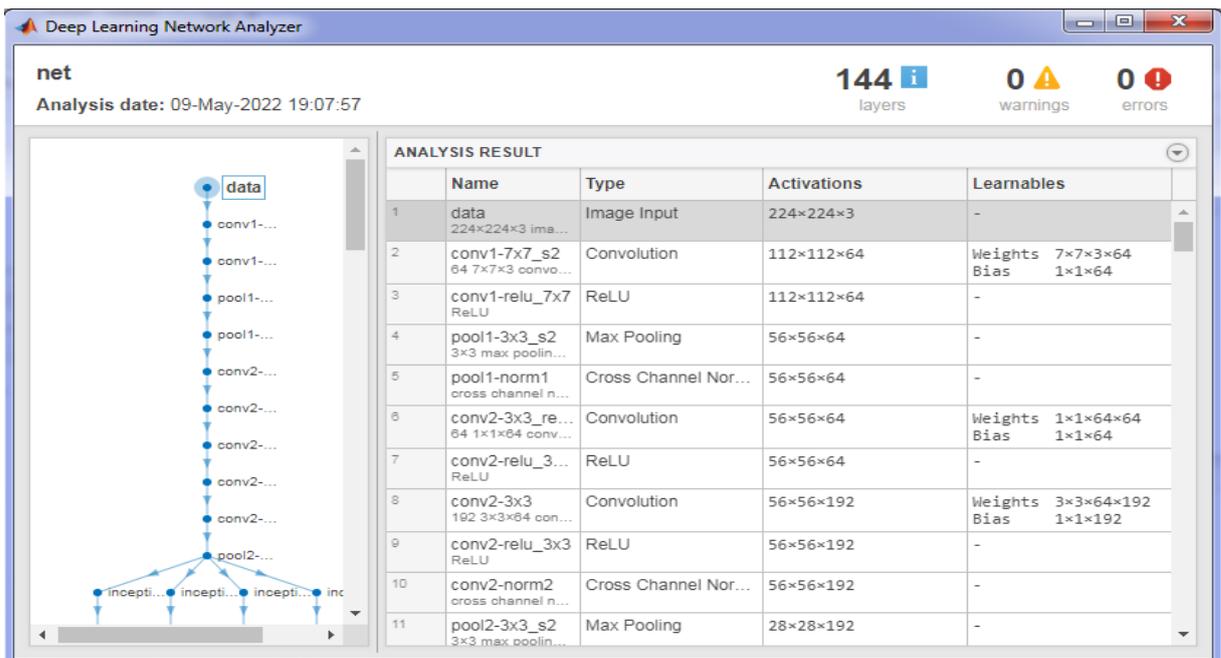


Figure III.14 Architecture du Google-Net.

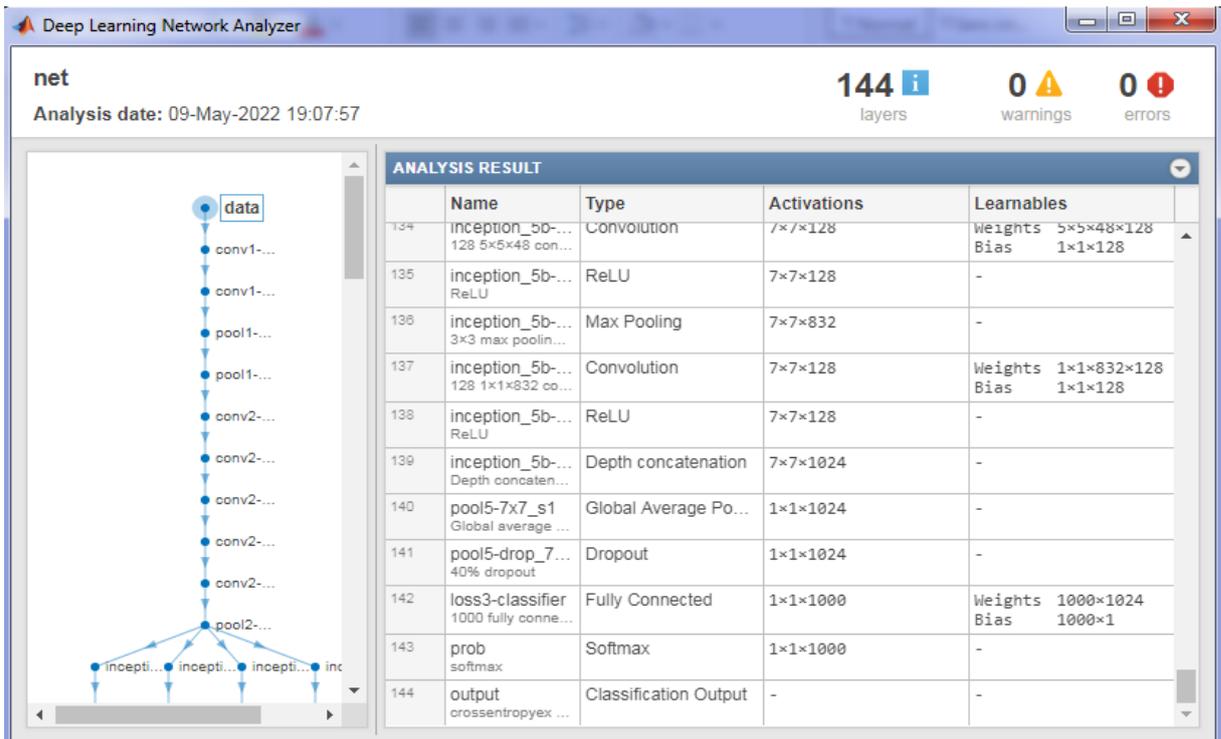


Figure III.15 Suite de l'architecture de Google-Net.

D'après ce code on peut avoir les informations sur la première couche du réseau :

```

Command Window
>> net.Layers(1)

ans =

    ImageInputLayer with properties:

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

    Hyperparameters
        DataAugmentation: 'none'
    
```

Figure III.16 Informations sur la couche d'entrée de Google-Net.

En utilisant le code suivant, on stocke la taille des images traitées par ce réseau dans une variable :

```

11 - input_layer_size = net.Layers(1).InputSize(1:2);
    
```

Figure III.17 Code d'affichage de la taille de couche d'entrée de Google-Net.

III.5.5 Adaptation des couches :

D'après la structure précédente du réseau, on remarque que la couche 142 est la couche entraînée avec des fonctionnalités spécifiques à la tâche et la couche 144 est responsable de la classification des objets.

On peut accéder aux couches 142 et 144 en utilisant les deux lignes suivantes :

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

Figure III.18 Extraction des dernières couches.

Maintenant, les couches que nous devons remplacer sont stockées dans les variables « Feature_Learner » et « Output_Classifier ».

À l'aide du code suivant, nous pouvons connaître le nombre de classes que nous avons dans l'ensemble de données d'entraînement :

```
19 - Number_of_Classes = numel(categories(Train_dataset.Labels));
```

Figure III.19 Nombre de catégories dans la base de données.

Il est le temps de définir nos propres couches. Tout d'abord, nous avons défini la couche entièrement connectée, qui est stockée dans la variable « Feature_Learner ». Nous utilisons le bloc de code suivant pour affiner une couche entièrement connectée :

```
20 - New_Feature_Learner = fullyConnectedLayer(Number_of_Classes, ...  
21     'Name', 'birds Feature Learner', ...  
22     'WeightLearnRateFactor', 10, ...  
23     'BiasLearnRateFactor', 10);
```

Figure III.20 Modification du nom et des paramètres de la couche FeatureLearner.

Nous devons remplacer aussi les couches de Google-Net qui sont entraînées à classer 1000 objets par nos propres couches. Nous pouvons utiliser le code suivant pour trouver des informations sur les couches spécifiques à la tâche :

```
28 - Layer_Graph = layerGraph(net);
```

Figure III.21 Affichage de la structure des couches.

Nous devons maintenant définir la couche de classification. Pour cela, nous utilisons une fonction nommée « classification Layer() » :

```
24 - New_Classifier_Layer = classificationLayer('Name', 'birds Classifier');
```

Figure III.22 Modification du nom de la couche de classification.

Le nom de notre nouvelle couche de classification est "birds Classifier".

Ainsi, les couches ont été définies. Il est temps de remplacer les couches existantes par nos couches redéfinies. Pour remplacer les couches, nous utilisons la fonction « replaceLayer () ». La fonction « replaceLayer () » nécessite la couche graphe, le nom du couches existantes et le nouveau couches comme arguments. La procédure est identique à la procédure de remplacement de la couche d'apprentissage. Voici le code utilisé pour remplacer la couche de sortie :

```
30 - New_Layer_Graph = replaceLayer(Layer_Graph, Feature_Learner.Name, New_Feature_Learner);
31 - New_Layer_Graph = replaceLayer(New_Layer_Graph, Output_Classifier.Name, New_Classifier_Layer);
```

Figure III.23 Remplacement des couches de Google-Net.

Ainsi, les couches de sortie ont été remplacées par nos couches. Nous pouvons commencer à former ces couches maintenant. Cependant, il est préférable de congeler certaines couches initiales avant de commencer le processus de formation.

142	birds Feature... 10 fully connect...	Fully Connected	1×1×10	Weights 10×1024 Bias 10×1
143	prob softmax	Softmax	1×1×10	-
144	birds Classifier crossentropyex	Classification Output	-	-

Figure III.24 Affichage de la nouvelle architecture de Google-Net.

III.5.6 Options d'entraînement :

L'entraînement d'un réseau neuronal multicouches nécessite la précision de plusieurs paramètres. Ils sont appelés « options d'entraînement » et peuvent influencer les résultats tels que la précision ou le temps d'entraînement. La fonction MATLAB « trainingOptions » est utilisée pour sélectionner ces options tels qu'il est illustré dans le code suivant :

```
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 III.25 Paramètres d'entraînement.

La méthodologie suivie est récapitulée par la figure suivante (III.26) :

III.6 Schéma récapitulatif de la méthodologie :

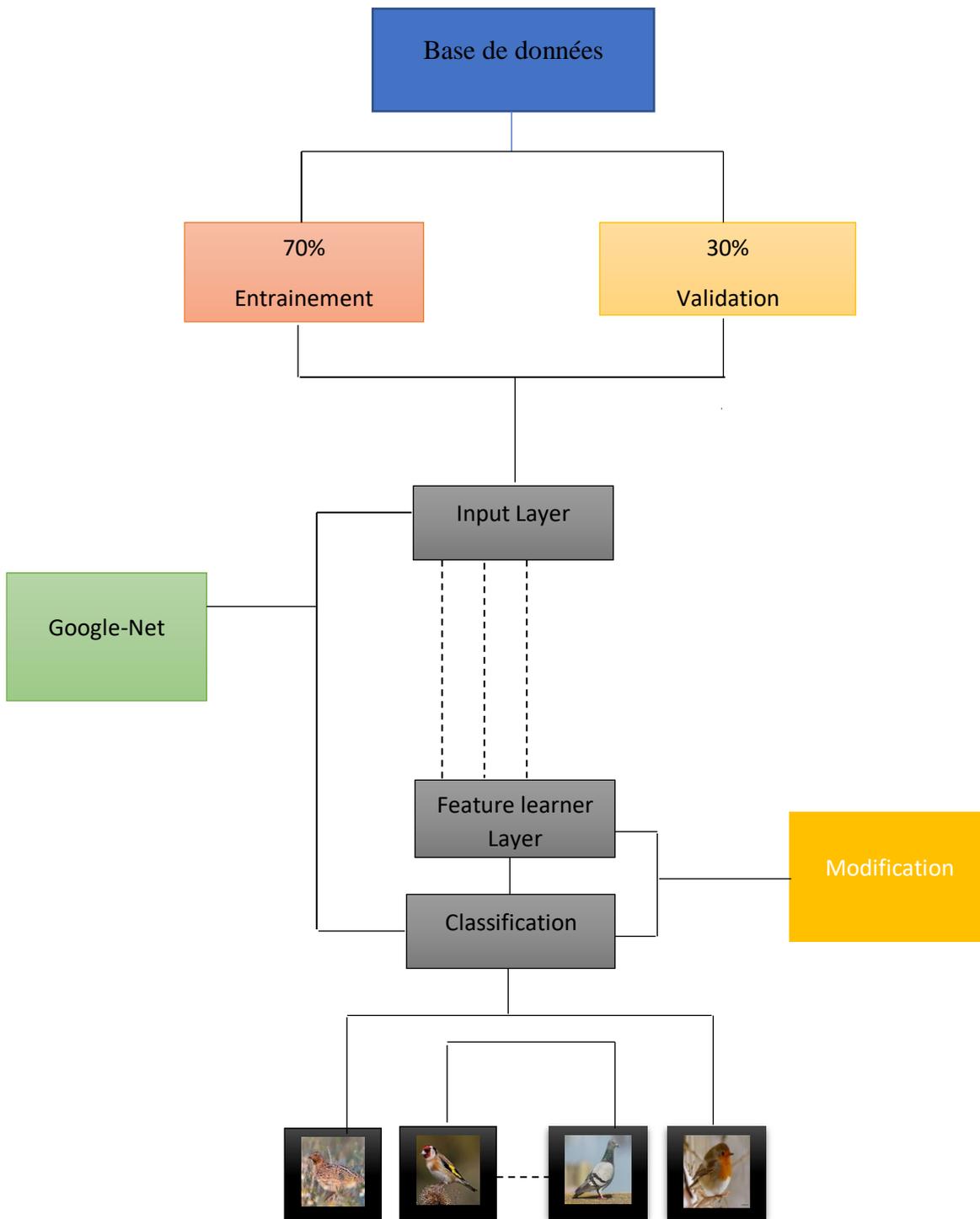


Figure III.26 Vue d'ensemble de la classification des oiseaux à l'aide de Google-Net modifié.

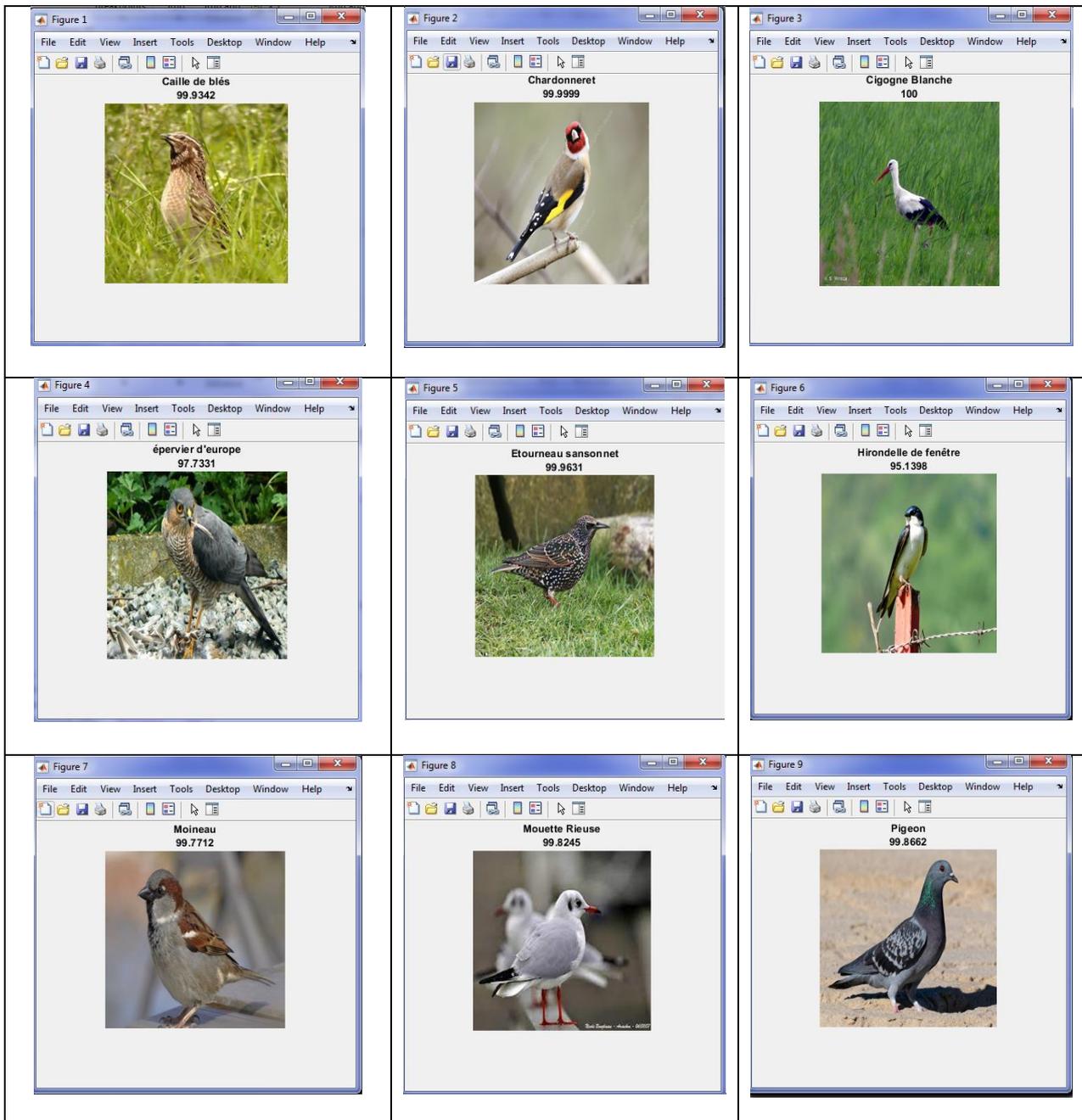
III.7 Tests avec différentes images :

III.7.1 Tests sur des images d'oiseaux des catégories de la base :

Pour tester la précision de reconnaissance de Google-Net avec notre base, on a pris pour chacune des 10 catégories une image complètement différente de celles qui sont dans notre base de données.

Les résultats sont indiqués dans le tableau ci-dessous.

Tableau III.2 Résultats des tests.





Analyse :

Du fait que nous avons entraîné le réseau Google-Net modifié sur notre base de données contenant une centaine d’images d’oiseaux, les résultats de la précision de reconnaissance ont été compris entre 99% et 100%. Donc le réseau a appris à reconnaître ces catégories d’oiseaux.

III.7.2 Tests sur des images n’appartenant pas à la base :

Dans une deuxième partie et pour mieux tester l’efficacité du réseau, on a pris des images complètement différentes des catégories des sujets de notre base de données : un oiseau différent, un animal différent, une partie (l’aile) d’un des oiseaux de notre base.

Tableau III.3 Résultat des tests des images n’appartenant pas à la base.

<p>Rouge Gorge 29.8193</p> 	<p>Pigeon 93.8873</p> 	<p>Caille de blés 99.9979</p> 	<p>Pigeon 28.9561</p> 
--	---	--	---

Analyse :

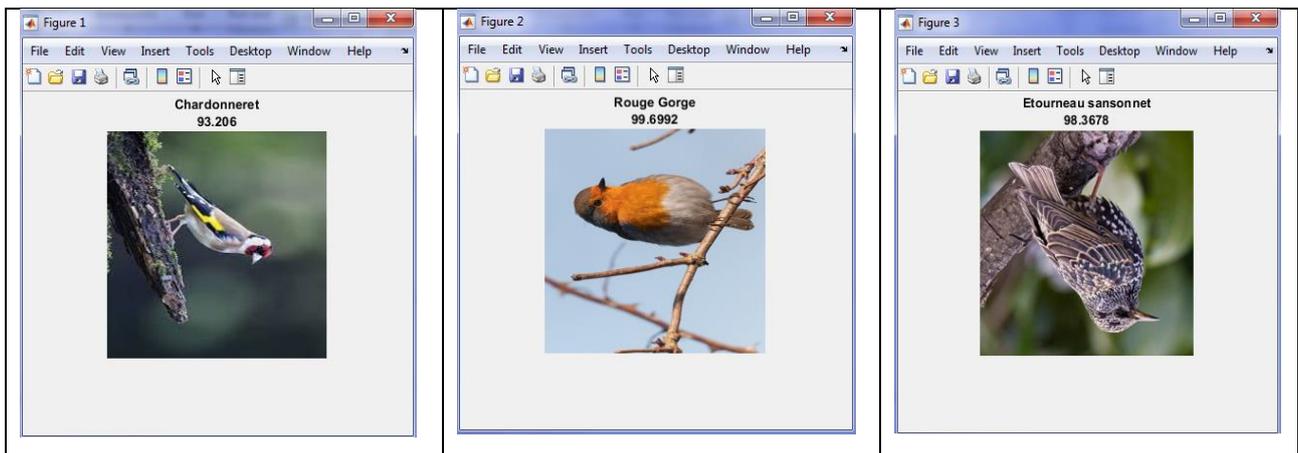
Le réseau Google-Net modifié a été entraîné avec les images de notre base de données. Quand on le teste avec l’image d’un animal différent comme un chat ou un objet comme un avion, la précision de reconnaissance devient très faible. Ceci signifie que le réseau ne reconnaît pas l’objet. Mais, il est obligatoirement classé dans une des catégories de la base.

Nous avons aussi testé le réseau avec les images d'un oiseau n'appartenant à aucune catégorie de la base de données et des images d'une partie de corps d'un oiseau. Le réseau les classe dans les catégories qui ont le plus de ressemblance avec les objets testés et donne un bon pourcentage précis.

III.7.3 Tests sur des images avec rotation :

Les oiseaux peuvent être photographiés dans différentes positions. Alors, pour tester notre réseau et sa précision de reconnaissance, nous avons pris trois images de notre base dans ces trois catégories (Chardonneret, Rouge Gorge, Etourneau sansonnet) et nous avons modifié avec une rotation de 90° à droite et à gauche et puis avec une rotation de 180°.

Tableau III.4 Résultat des tests des images avec rotation



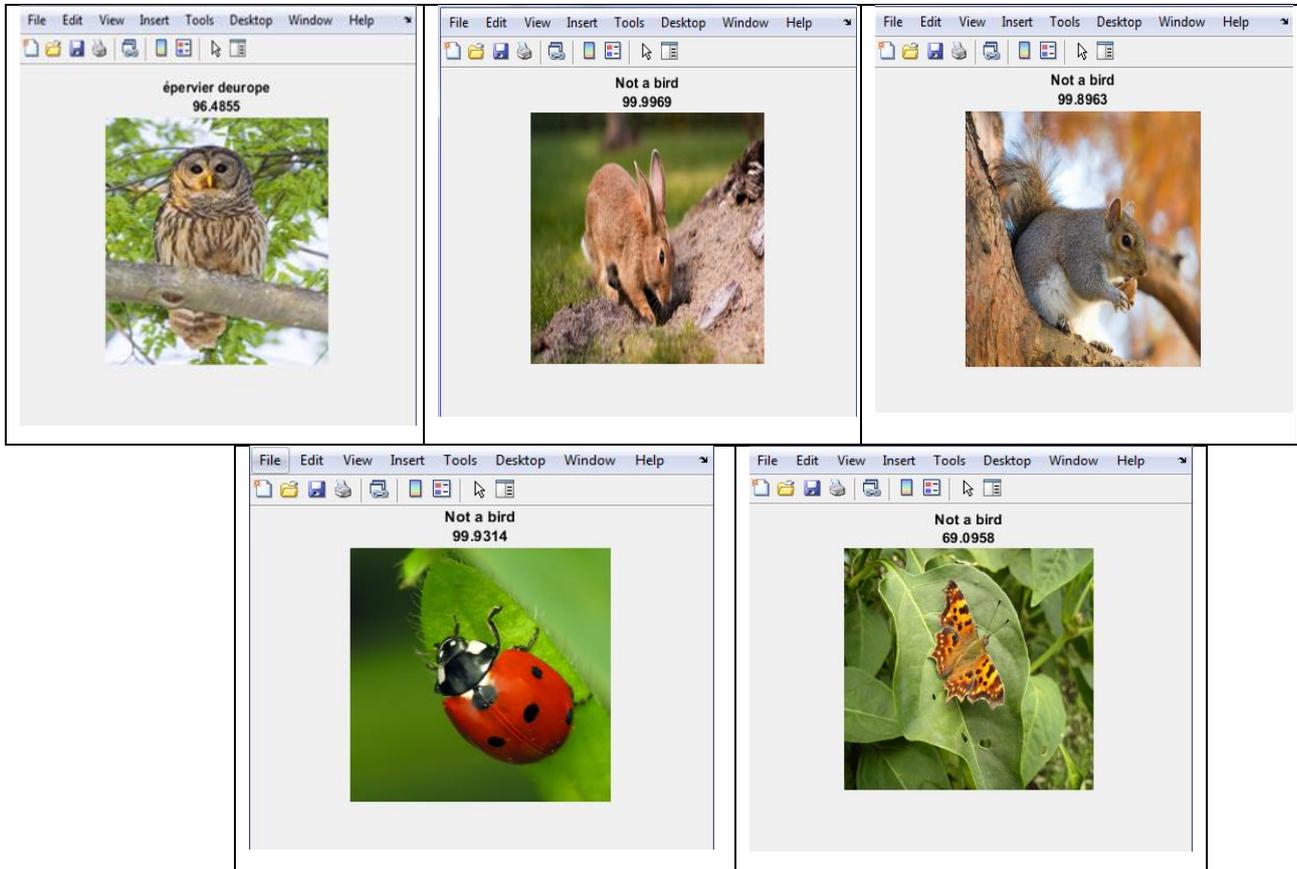
Analyse :

On remarque que le réseau a réussi à reconnaître les images de la base avec une bonne précision supérieure à 90% malgré qu'elles ont subi des rotations.

III.7.4 Tests sur des images n'appartenant pas aux catégories de la base :

Ayant constaté que Google-Net place toute image dans l'une des 10 catégories, nous avons ajouté une nouvelle catégorie dans notre base appelée "Not a bird" où sont placés des images d'autres animaux qui peuvent être présents sur les sites des oiseaux à surveiller. Nous procédons par la suite aux tests de la précision de reconnaissance de Google-Net modifié.

Tableau III.5 Résultats des tests des images n'appartenant pas aux catégories de la base.



Analyse :

On remarque que Google-Net a réussi à reconnaître les images et les catégoriser correctement. Si l'image ne représente pas un oiseau, elle est placée dans la catégorie "Not a bird" .

III.8 Analyse des effets des paramètres d'entraînement sur la vitesse et la précision de reconnaissance :

III.8.1 Batch :

La base donnée est généralement trop grande pour passer en une seule fois à travers le réseau. Elle est alors divisée en parties appelées « batch ».

La dimension du batch représente le nombre d'échantillons dans un batch.

Dans notre programme on va faire varier ce paramètre (2, 5, 10) pour évaluer la précision de reconnaissance et le temps nécessaire à l'entraînement :

Pour 2 Batch :

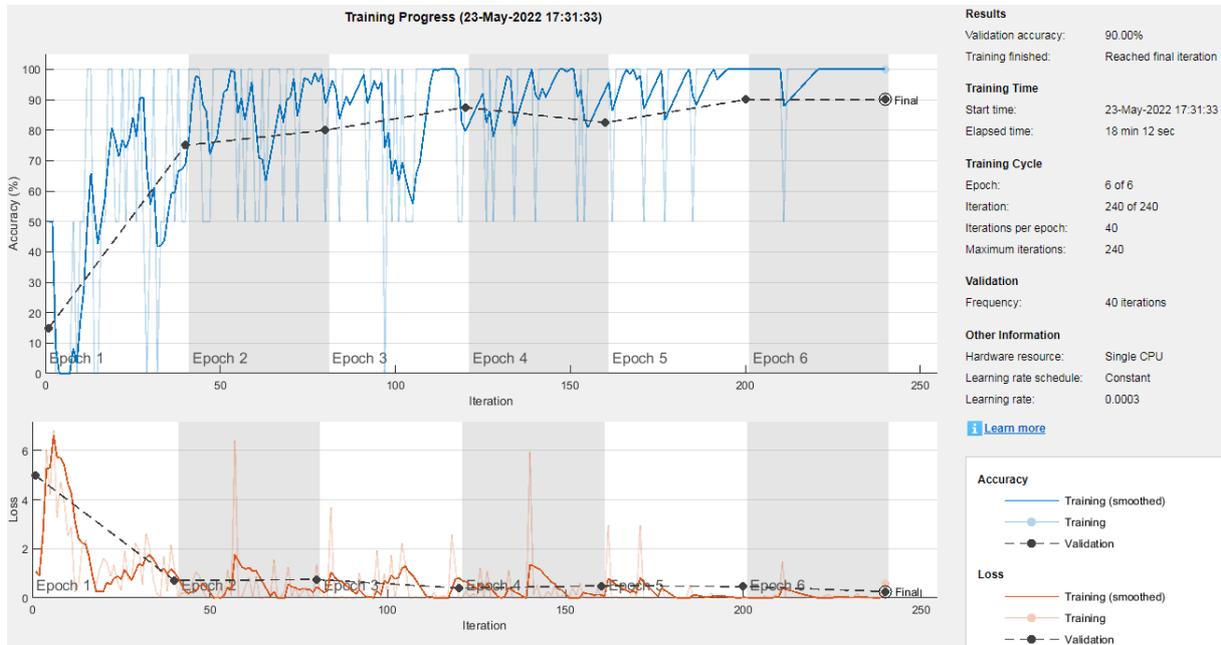


Figure III.27 Processus d'entraînement pour 2 Batch.

Pour 5 Batch qui est la valeur utilisée dans notre programme, on a ce résultat :

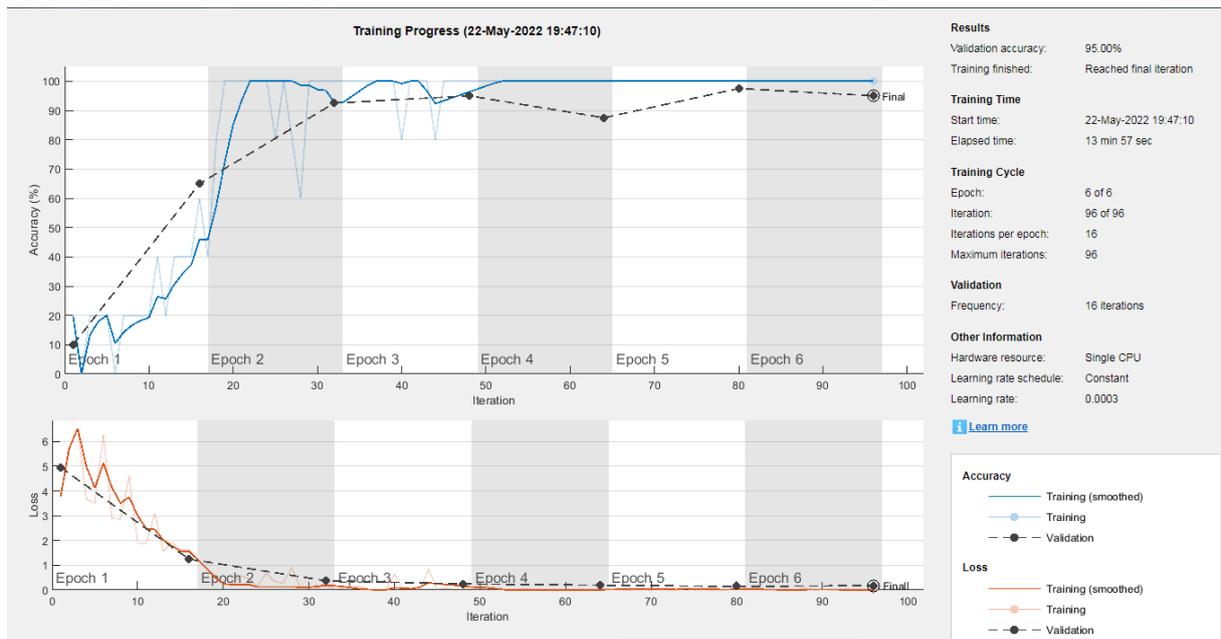


Figure III.28 Processus d'entraînement pour 5 Batch.

Pour 10 Batch on a ce résultat :

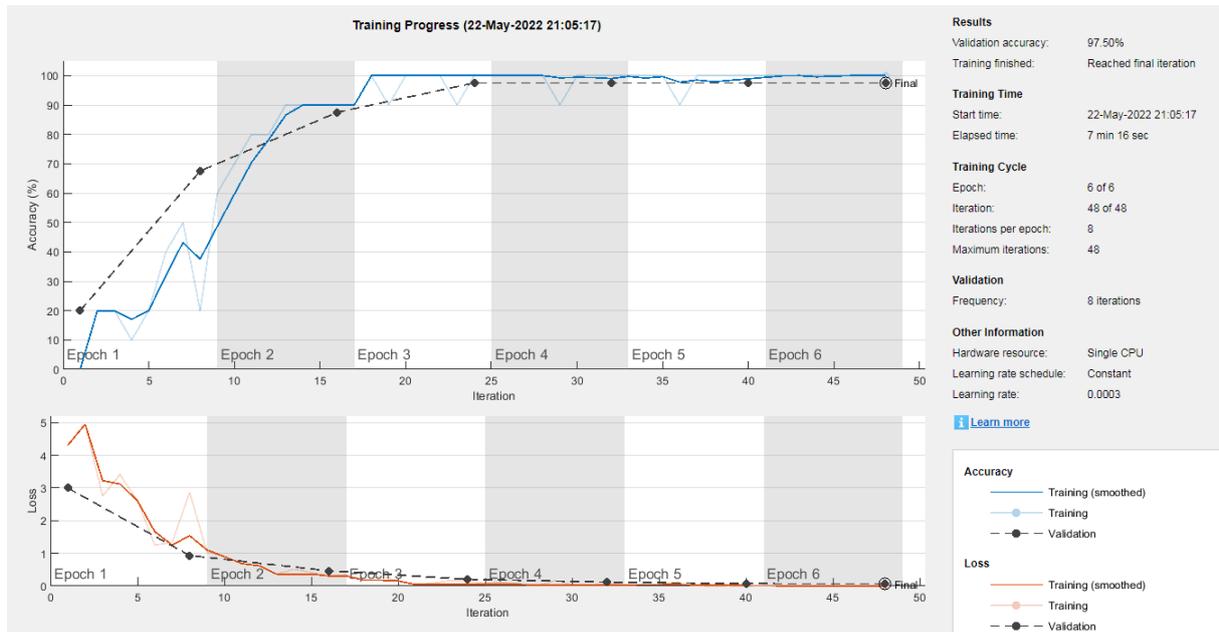


Figure III.27 Processus d’entrainement pour 10 Batch.

Tableau III.6 Effet du paramètre "Batch".

Size-of-Minibatch	Précision	Temps
2	90.00%	18 min 12 sec
5	95.00%	13 min 57 sec
10	97.50%	7 min 16 sec

Analyse :

On remarque que la précision augmente quand on augmente le nombre de batch car il est relié aux itérations. En augmentant le nombre de batch, la fréquence d’itération diminue, ce qui conduit à augmenter la précision et diminuer le temps d’entrainement.

III.8.2 Itérations :

Les itérations sont simplement le nombre de batch nécessaire pour compléter un epoch.

Si on a, par exemple, 2000 échantillons pour l’entrainement, on peut les diviser en des batch de 500, ce qui donnera 4 itérations pour un epoch.

III.8.3 Epoch :

Le nombre d’epoch est un hyper paramètre qui détermine le nombre de passage de l’ensemble des données réservé pour l’entrainement par l’algorithme d’entrainement.

Un epoch signifie que chaque objet « image » de la base de données de l’entraînement est passée une fois et a permis une modification/mise à jour des paramètres internes du CNN.

Mais un passage n’est pas suffisant. Il faut donc passer les données de l’entraînement plusieurs fois.

Les bases de données sont limitées et pour optimiser l’apprentissage, un algorithme itératif est utilisé **Gradient Descent pour la mise à jour des poids**.

Le nombre d’epoch reste empirique et dépend de la diversité des bases de données.

Dans nos tests, on a donné à ce paramètre les valeurs ce 4, 6, et 9 pour évaluer la précision et le temps :

Pour 4 epoch on a ce résultat :

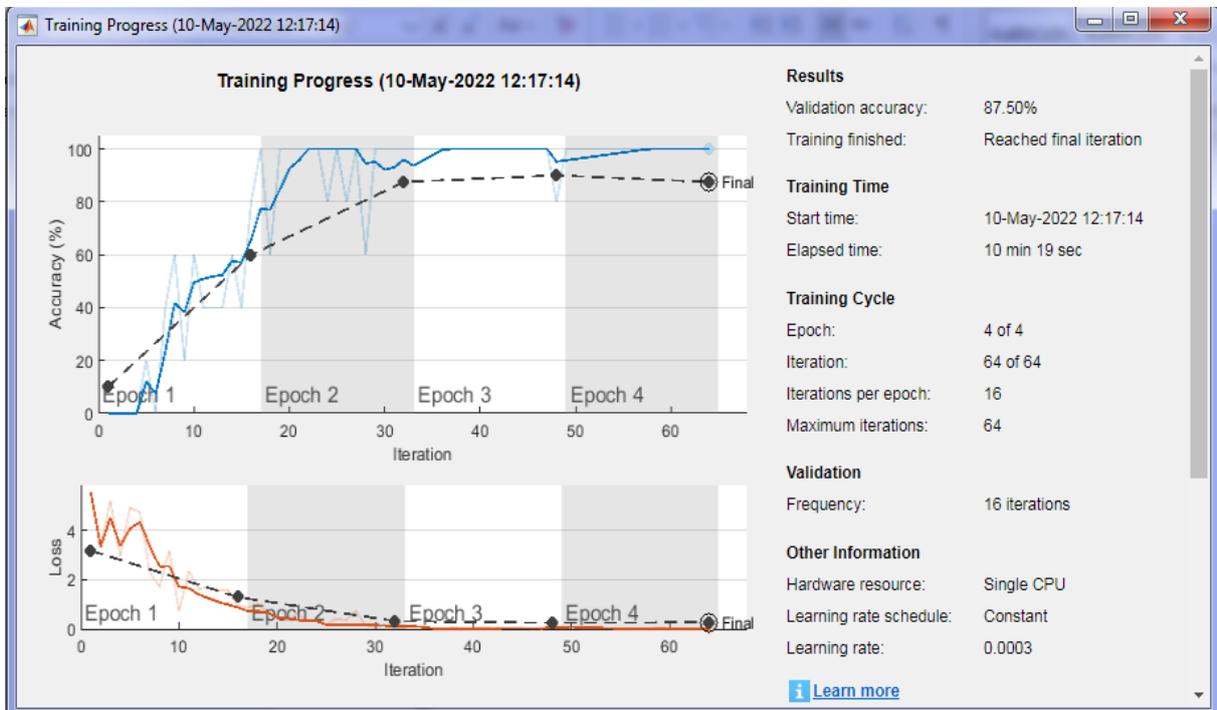


Figure III.30 Processus d’entraînement pour 4 Epoch.

Pour 6 Epoch qu’est le epoch principale de notre programme, on a ce résultat :

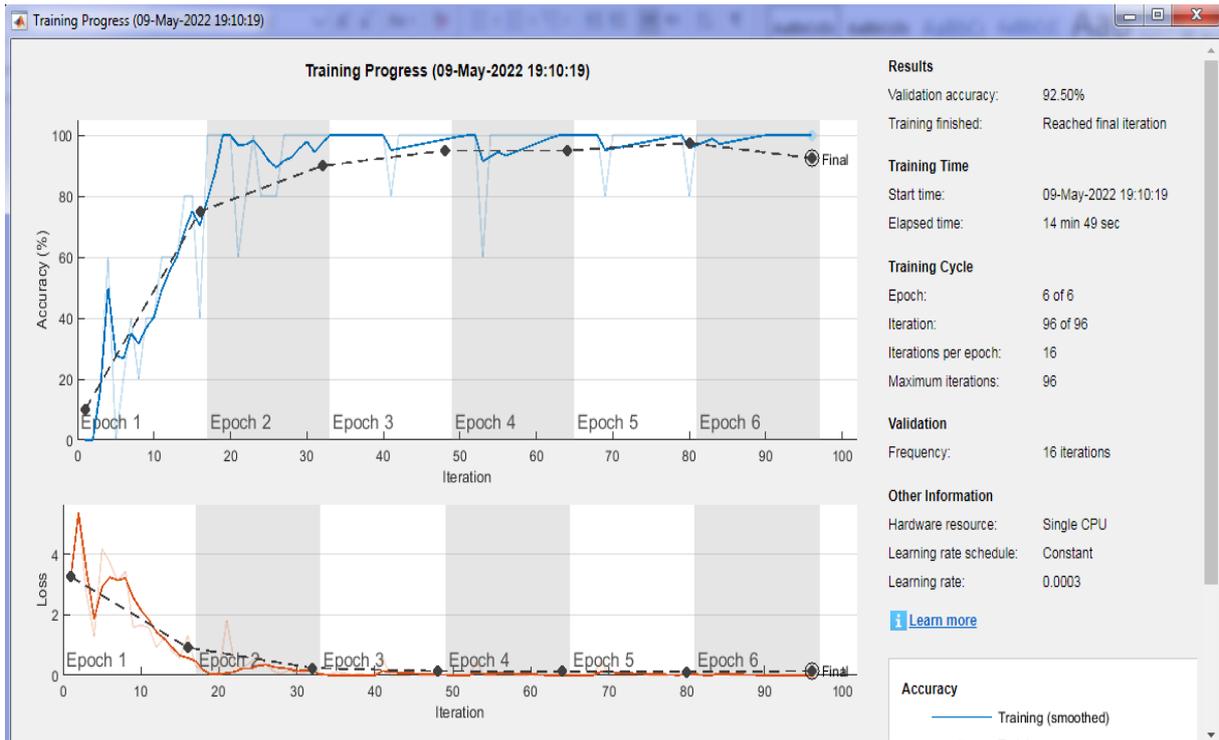


Figure III.31 Processus d’entrainement pour 6 Epoch.

Pour 9 Epoch on a ce resultat :

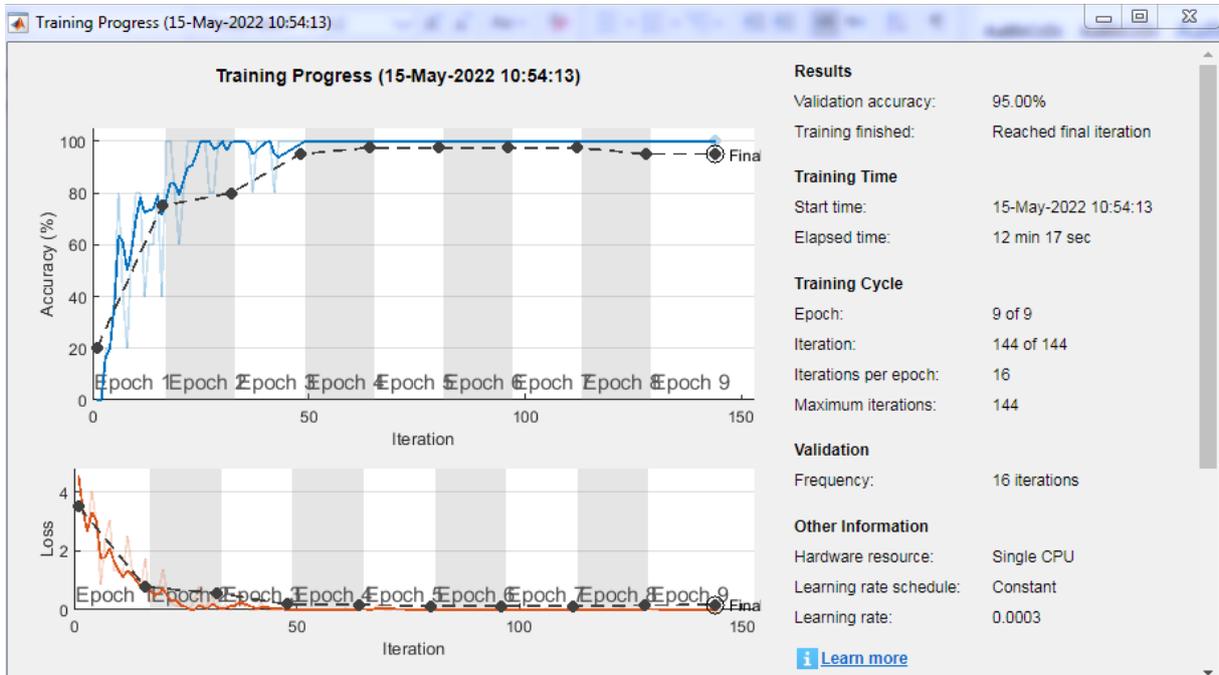


Figure III.32 Processus d’entrainement pour 9 Epoch.

Tableau III.7 Effet du paramètre "Epoch".

MaxEpochs	Précision	Temps
4	87.50%	10 min 19 sec
6	92.50%	14 min 49 sec
9	95.00%	12 min 17 sec

Analyse :

On remarque que quand on augmente le nombre d'époch, la précision augmente aussi. C'est tout à fait logique puisque les coefficients internes du réseau sont ajustés avec chaque epoch. Le temps d'entraînement devient normalement plus long mais n'est pas proportionnel au nombre d'époch.

III.8.4 Taux d'apprentissage :

Le taux d'apprentissage peut être l'hyperparamètre le plus important lors de la configuration d'un réseau de neurones, car il contrôle le degré de modification du modèle en réponse à l'erreur estimée à chaque fois que les poids du modèle sont mis à jour.

Dans notre programme on va faire varier ce paramétré (4e-6, 3e-4, 3e-3) pour évaluer la précision et le temps d'entraînement :

Pour 4e-6 on a ce résultant :

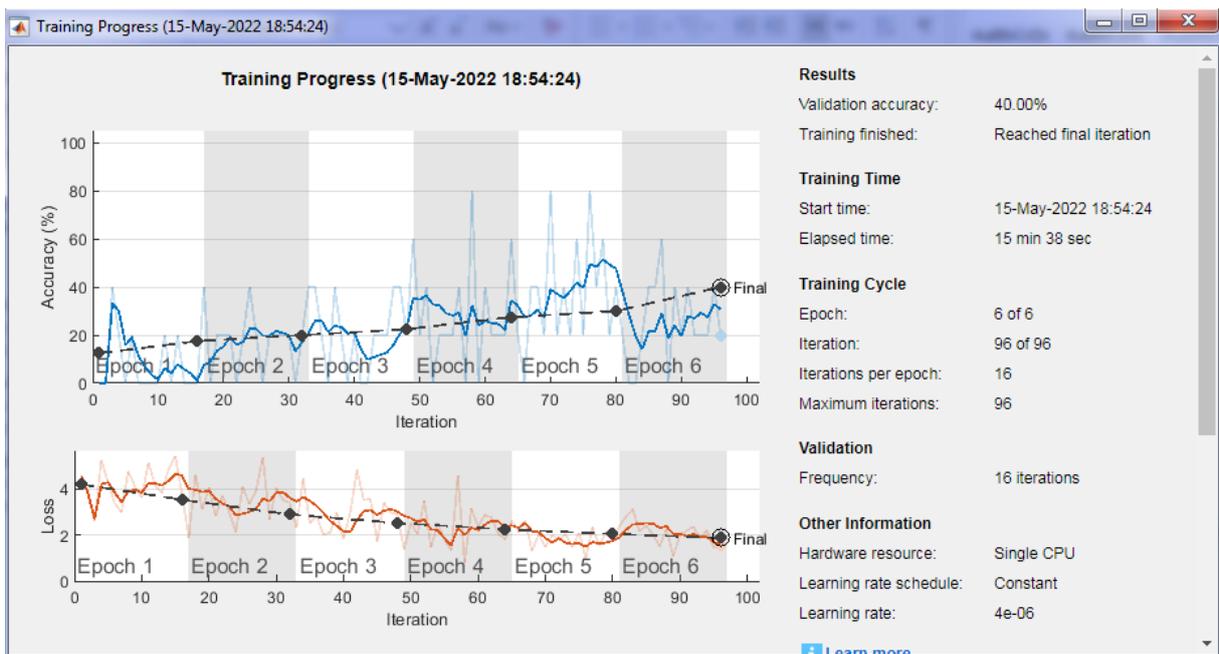


Figure III. 33 Processus d'entraînement avec Taux d'apprentissage 4e-6.

Pour 3e-4 :

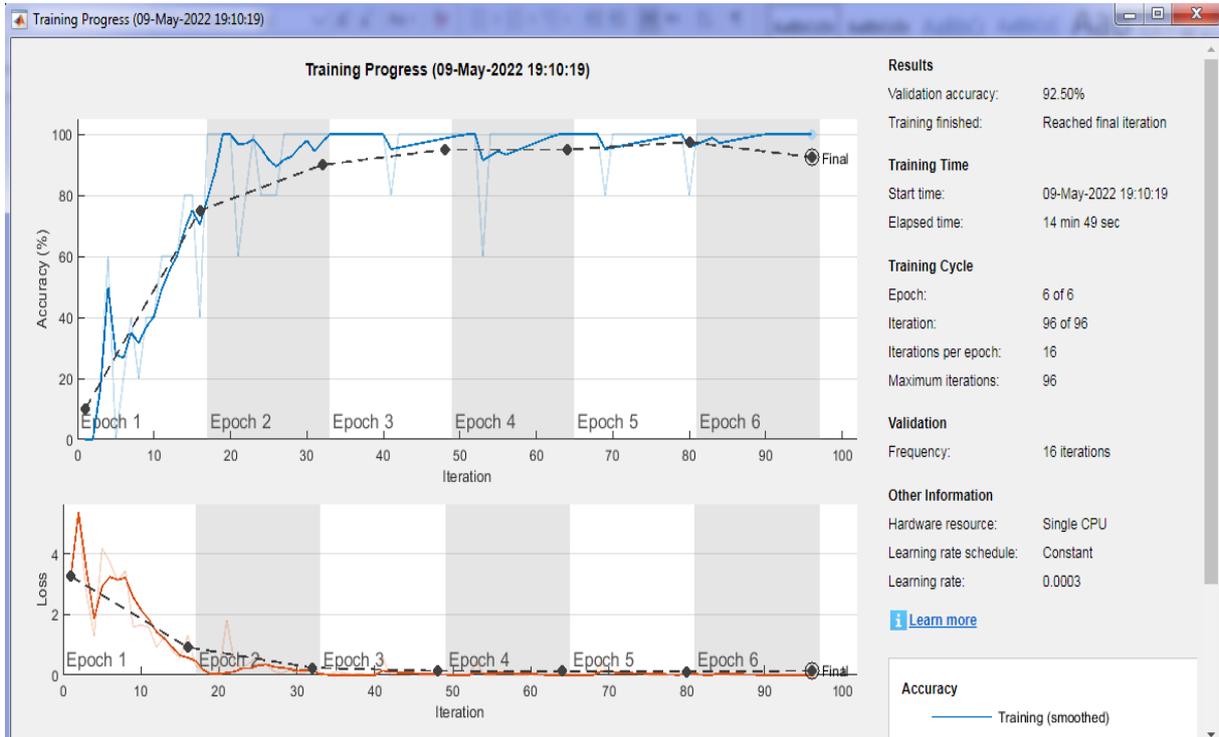


Figure III.34 Processus d'entraînement avec Taux d'apprentissage égale 3e-4.

Pour 3e-3 :

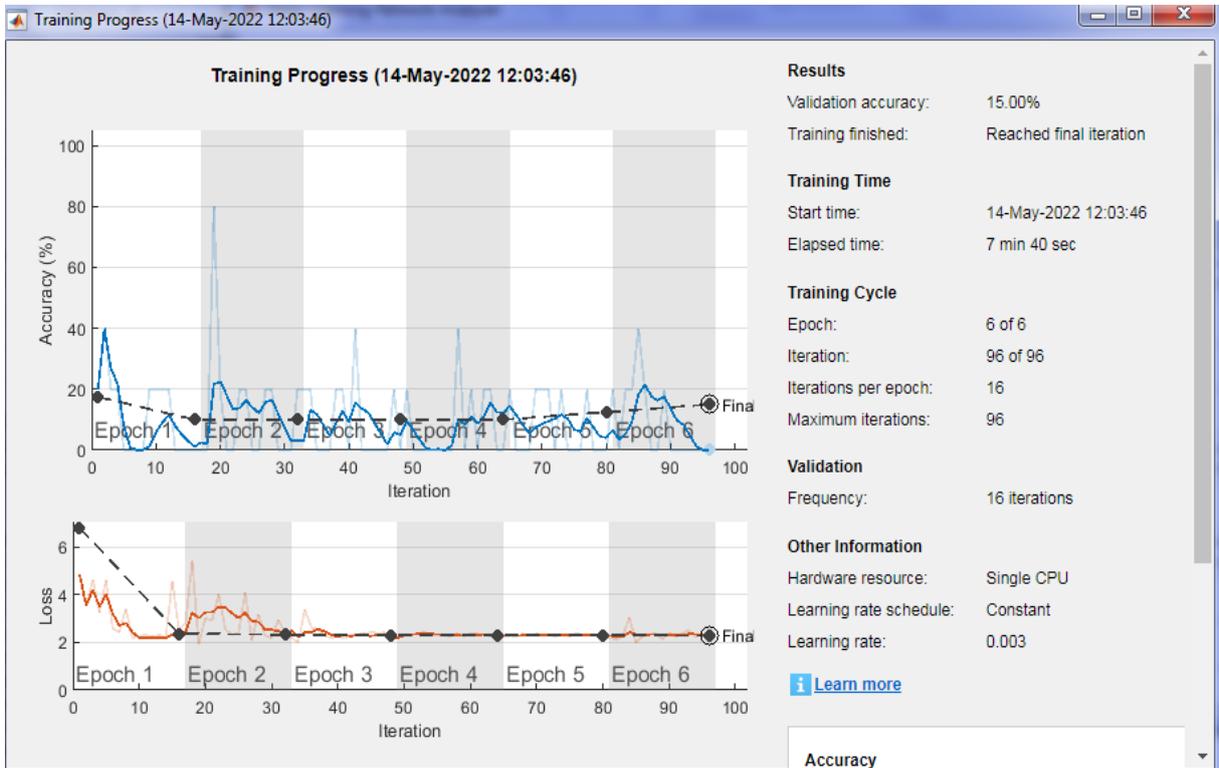


Figure III. 35 Processus d'entraînement avec Taux d'apprentissage égale 3e-3.

Tableau III.8 Effet du paramètre "initial Learning Rate".

InitialLearnRate	Précision	Temps
4e-6	40.00%	15 min 38 sec
3e-4	92.50%	14 min 49 sec
3e-3	15.00%	7 min 40 sec

Analyse :

Pour chaque réseau il y a un taux d'apprentissage qui l'aide à mieux analyser et apprendre à reconnaître les images. Donc d'après ces résultats on remarque que le taux d'apprentissage de Google-Net est $3e-4$ car n'importe quelle autre valeur donne une précision toujours inférieure à 50%.

III.9 Conclusion :

Dans ce chapitre on a présenté la partie pratique de notre travail où on a entraîné le réseau Google-Net sur notre base de données. Les tests effectués sur les paramètres influençant les résultats des simulations sont aussi exposés en détails. On a découvert que pour un bon résultat, c.à.d. une bonne précision, on doit avoir un grand nombre de batch avec un grand nombre d'epoch et une valeur appropriée du taux d'apprentissage.

Conclusion générale

Conclusion générale

Le travail présenté dans ce mémoire est lié à l'apprentissage en profondeur, où nous nous sommes concentrés sur les réseaux de neurones à (CNN) et leur utilisation dans la vidéo surveillance appliquée à la protection de l'environnement.

Tout d'abord, nous avons commencé à nous familiariser avec le domaine de l'IA et ses sous-catégories. Nous nous sommes concentré d'abord sur les couches qui composent un CNN et leurs fonctions. Puis, nous avons pris connaissance de l'histoire des réseaux CNN pour connaître leur évolution et leurs architectures. Enfin, nous avons travaillé sur l'utilisation d'un réseau CNN pré entraîné dans la classification de dix catégories d'oiseaux sédentaires et migrateurs existants en Algérie. Avec plus de travail dans ce sens, on pourrait contribuer à la protection de notre environnement en contribuant à des études sur les oiseaux, en particulier ceux menacés d'extinction. Nous avons utilisé Google Net et le transfert d'apprentissage après avoir créé une petite base de données des oiseaux ciblés.

Nous avons réussi à adapter le réseau CNN à l'application désirée après avoir appris à mener une analyse sur son architecture et les caractéristiques de certaines couches.

Nous avons mené des tests et mesuré la précision de reconnaissance du réseau et le temps du traitement. Nous avons aussi apporté des modifications à des copies des images testées qui faisaient partie ou non des catégories de la base de données afin de simuler les prises d'images réelles dans la nature où l'objet surveillé peut être photographié dans n'importe quelle position, en entier ou en partie, seul ou en compagnie d'autres espèces.

Des paramètres du réseau CNN, dits "hyper paramètres" ont une très grande importance dans la phase d'entraînement. Nous avons considéré la taille du batch, le nombre d'époque et le taux d'apprentissage et testé leurs influences sur la précision de classification et le temps du traitement. Nous avons alors conclu que tous ces paramètres ont une influence sur le temps et la précision.

Nous aurions aimé travailler sur les images compressées à différents taux et les utiliser pour estimer la précision de reconnaissance. En effet, les images capturées doivent toujours subir

Conclusion générale

une compression avant leur stockage ou transmission. Cette opération permet de réduire le débit, i.e. la bande passante ainsi que l'énergie nécessaire pour leur transmission. Malheureusement, nous n'avons pas beaucoup de temps ou mener cette partie du travail. Nous proposons ce volet dans les perspectives. Cette partie consisterait, en un premier temps, à compresser des images de la base de données à différents taux de compression et les utiliser dans les tests sans qu'elles soient dans la partie de l'entraînement. Puis, les inclure dans le sous-ensemble de l'entraînement et refaire des tests. Les précisions de reconnaissance dans les deux cas seront alors comparées.

Enfin, nous espérons que nous aurons l'occasion et la chance de travailler sur cette idée dans l'avenir et même développer un réseau CNN dédié à une application précise telle que la vidéo surveillance des oiseaux ou autres espèces menacées afin de protéger notre nature.

Références bibliographiques

[1]<https://www.datakeen.co/3-deep-learning-architectures-explained-in-human-language/>.

Consulté Mai 2022.

[2]Rebahi Ghediri Imane ,Semri KhawLa, «La Reconnaissance Des émotions de base Par Les .réseaux de neurones : Application de Deep Learning », Université L'arbi Ben M'Hidi , Algérie, Oum El Bouaghi .2020/2021.

[3] MERDOUD Kenz, BOUSBAIN Karim, « Détection de maladies par traitement d'image », UNIVERSITE AKLI MOAND OULHADJE, Algérie, Bouira.2017/2018.

[4] Dyhia BOUADI,Thanina IMARAZENE, « Classification d' images agricoles avec le Deep Learning », Université Mouloud MAMMARI, Tizi Ouzou.

[5]<https://www.sciencedirect.com/science/article/pii/B9780128157398000109?via%3Dihub>.

Consulté Mai 2022.

[6] DIALLO Nene Adama Dian, « La reconnaissance des expressions faciales » Mémoire de Fin d'études Master, Université 8 Mai 1945, Guelma, Algérie, Juillet 2019.

[7]<https://www.jeuxetredatascientist.fr/tag/machine-learning/#:~:text=Comme%20d'autres%20r%C3%A9seaux%20de,caract%C3%A9ristiques%20sp%C3%A9cifiques%20de%20ces%20donn%C3%A9es>.

Consulté Mai 2022.

[8][https://datascientest.com/convolutional-neural-network/#:~:text=Couche%20de%20convolution%20\(CONV\)%20%3A,quoi%20sert%20la%20convolution%20%3F](https://datascientest.com/convolutional-neural-network/#:~:text=Couche%20de%20convolution%20(CONV)%20%3A,quoi%20sert%20la%20convolution%20%3F).

Consulté Mars 2022.

[9] BRAIKIA_BOUCHOUICHA. “Étude et implémentation d'un réseau de neurones convolutif CNN sur FPGA“ master universite badji mokhtar annaba, 2020.

[10]<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn>.

Consulter à mai 2022.

[11][https://www.inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/#:~:text=sigmoid\(x\)-,Softmax,chaque%20vecteur%20est%20trait%C3%A9%20ind%C3%A9pendamment](https://www.inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/#:~:text=sigmoid(x)-,Softmax,chaque%20vecteur%20est%20trait%C3%A9%20ind%C3%A9pendamment).

Consulté Juin 2022.

Références bibliographiques

[12] KAZI TANI (Ep. DALI YUCEF) Lamia Fatiha « Conception et Réalisation d'un Outil Semi-Automatique Dédié à l'Analyse et la Recherche du Contenu Vidéo du Web », Université Oran احمد بن بلة, Algérie, Oran, 2020.

[13] <https://fr.acervolima.com/ml-premiers-pas-avec-alexnet-2/>. Consulté Avril 2022.

[14] <https://www.arabicprogrammer.com/article/7565444219/>. Consulté Mars 2022.

[15] Chouini Mohammed El Mounsif. « Détection d'objets basé Faster R-CNN »,

Université de 8 Mai 1945, Algérie, Guelma, octobre 2020.

[16] https://www.google.com/url?sa=i&url=https%3A%2F%2Fmedium.com%2Fcoinmonks%2Fpaper-review-of-zfnet-the-winner-of-ilsvlc-2013-image-classification-d1a5a0c45103&psig=AOvVaw1OATHM1NV4xD3qu9UqL88O&ust=1653692537003000&source=images&cd=vfe&ved=0CAwQjRxqFwoTCNDdmfqi_vcCFQAAAAAdAAAAABAJ
Consulté Juin 2022.

[17] <https://fr.acervolima.com/vgg-16-modele-cnn-2/> .Consulté Mai 2022.

[18] <https://datascientest.com/quest-ce-que-le-modele-vgg> .Consulté Mars 2022.

[19] LOUNIS Katia, MOUSSI Dahbia, « La Classification d'images d'insectes ravageurs en utilisant le Deep Learning », UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU, Algérie, Tizi Ouzou, 2019/2020.

[20] <https://datascience.stackexchange.com/questions/33022/how-to-interperet-resnet50-layer-types>. Consulté Mai 2022.

[21] HAMLAOUI Mohammed Yazid « Classification du COVID '19 en utilisant CNN dans les images X-ray », Université de 8 Mai 1945, Algérie, Guelma. Septembre 2021.

[22] <https://depot-e.uqtr.ca/id/eprint/9555/1/eprint9555.pdf>.

[23] https://www.linkedin.com/pulse/imagenet-classification-deep-convolutional-neural-zuluaga-g%C3%B3mez?trk=portfolio_article-card_title. Consulté Mai 2022.

[24] <https://paperswithcode.com/dataset/cifar-10>. Consulté Avril 2022.

Références bibliographiques

[25] <https://www.cs.toronto.edu/~kriz/cifar.html>. Consulté Avril 2022.