

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE MOHAMED SEDDIK BENYAHIA JIJEL

Faculté des sciences et de la technologie

Département d'Automatique

N° :/2022

MEMOIRE DE MASTER

DOMAINE : Sciences et Technologies

FILIERE: Automatique

SPÉCIALITÉ : Automatique et Informatique Industrielle

Thème

**CLASSIFICATION DES IMAGES PAR APPRENTISSAGE
PROFOND (DEEP LEARNING)**

Présenté par : KHATTRA Sidi Mohammed

Encadré par : Dr BIAD Souad

BOULFOUS Oussama

Date de soutenance : 14/07/2022

Jury de Soutenance

Président : D. Boutana

Grade Professeur

Univ MSB jijel

Encadreur : S. Biad

Grade MCB

Univ MSB jijel

Examinatrice : I. Bouraoui

Grade MCA

Univ MSB jijel

Promotion :2021 /2022



Remerciements

À l'issue de notre travail nous tenons à remercier en premier lieu Dieu le tout puissant.

Nos remerciements les plus sincères vont à : Dr Biad souad pour sa généreuse disponibilité et pour sa grande professionnalité, elle a su mettre à disposition sa connaissance pour nous permettre d'avancer dans notre recherche, nous avons énormément appris.

Mes vifs remerciements vont aux membres de jury pour avoir accepté de juger notre présent travail.

Enfin toute personne qui a participé de près ou de loin à l'accomplissement de ce mémoire soit remerciée et les enseignants qui ont participé à nos formations soient sincèrement remerciés.

...

Merci



Dédicace

À ma chère mère Idounha

À mon père Mohamed mahmoud

Dont le mérite, les sacrifices m'ont permis de vivre ce jour

À mes adorables frères « Sid 'Ahmed, Khalil wezir »

À chères sœurs « verida, zeinel, aminettou »

À mon ami : mostafa salihi

Sidi Mohammed Khattrra





Dédicace

*Je dédie ce modeste travail à celle qui ma
guidée, le symbole de la tendresse ,qui s'est
sacrifiée pour mon bonheur et ma réussite, à ma Mère.
À mon Père, l'école de mon enfance, qui a été mon ombre
durant toutes les années d'études, et qui a veillé tout au long
de ma vie à m'encourager à m'aider.*

Ainsi que mes grands-parents que j'aime très fort,

Surtout à ma grand-mère (que dieu l'accueil dans son vaste paradis).

À mes adorables frères « med-amine, omar, yasser, diaa eddine »

et mes chères sœurs « fella, nour-elhouda, assma » et mes

oncles, cousins, tantes surtout nour-eddine et fatiha.

À nihel, zakaria, layan, amina, doha, abboud,

khalil, chocho, rokia pour son aide précieuse.

À mes amis : walid, hakim, wahid, idris,

soufiane mekhlouf, hachem, hicham,

fares, haroun, yazid ...

Boulfous Oussama



TABLE DES MATIERES

INTRODUCTION GENERALE

INTRODUCTION GENERALE	1
-----------------------------	---

CHAPITRE I : APPRENTISSAGE PROFOND (DEEP LEARNING)

CHAPITRE I :	2
I.1 INTRODUCTION	2
I.2 APPRENTISSAGE AUTOMATIQUE	3
I.3 RESEAUX DE NEURONES ARTIFICIELS	5
I.3.1 Neurone biologique	5
I.3.2 Le neurone formel	6
I.3.3 Le Perceptron	7
I.3.4 Perceptron Multicouches (PMC)	8
I.3.5 Rétropropagation	9
I.4 APPRENTISSAGE PROFOND	10
I.4.1 Domaines d'application de l'apprentissage profond	11
I.5 RESEAUX DE NEURONES CONVOLUTIONNELS	11
I.5.1 Couche de convolution	13
I.5.2 Fonction d'activation	14
I.5.3 Couche de sous-échantillonnage (pooling layer)	14
I.5.4 Couches entièrement connectées	15
I.6 HYPERPARAMETRES D'UN RESEAU DL	16
I.7 CONCLUSION	17

CHAPITRE II : APPRENTISSAGE PAR TRANSFERT (TRANSFER LEARNING)

CHAPITRE II :	18
II.1 INTRODUCTION	18
II.2 APPRENTISSAGE PAR TRANSFERT	18
II.3 PRINCIPE DE BASE	19
II.3.1 Types d'apprentissage par transfert	19
II.4 APPRENTISSAGE PAR TRANSFERT PAR ADAPTATION DES RESEAUX CNN PRE-ENTRAINES	21
II.5 RESEAUX CNN PRE-ENTRAINES	21
II.5.1 AlexNet	22
II.5.2 Google Net/Inception	22
II.5.3 ResNet-50	23
II.5.4 VGG	24
II.6 CONCLUSION	25

CHAPITRE III : CLASSIFICATION D'IMAGE

CHAPITRE III :	26
III.1 INTRODUCTION	26
III.2 CLASSIFICATION D'IMAGE	26
III.2.1 Motivations de la classification des images	26
III.2.2 Apprentissage par transfert et classification	26
III.2.2.1 Fine Tuning	27
III.2.2.2 Extraction de caractéristique et SVM	28
III.3 LES PARAMETRES D'EVALUATION D'UNE CLASSIFICATION	28
III.3.1 Précision	28
III.3.2 Rappel	29
III.3.3 La courbe ROC	29
III.3.4 Matrice de confusion	29
III.4 LES RESULTATS DE LA SIMULATION	30
III.4.1 Bases de données utilisées	30
III.4.1.1 Base 1 (GC10-DET)	31
III.4.1.2 Base 2 (NEU-DET)	32
III.4.1.3 Base 3 (GC10-DET)	32
III. 5. Résultats de simulation	32
III. 5. 1. Base 1 (GC10-DET)	33
III. 5. 2 Base 2 (NEU-DET)	39
III. 5. 3. Base 3 (Casting data)	45
III.5 CONCLUSION	50

CONCLUSION GENERALE ET PERSPECTIVE

CONCLUSION GENERALE	52
----------------------------	-----------

LISTE DES FIGURES

FIGURE I.1 : relation entre IA, ML, DL-----	3
FIGURE I.2 : chaine de traitement par apprentissage automatique -----	4
FIGURE I.3 : neurone biologique -----	5
FIGURE I.4 : structure d'un neurone biologique contre un neurone artificiel-----	6
FIGURE I.5 : perceptron (a) une seule sortie, (b) k sorties -----	8
FIGURE I.6 : perceptron multicouches (PMC) -----	9
FIGURE I.7 : apprentissage des poids du PMC -----	9
FIGURE I.8 : difference entre apprentissage machine et apprentissage profond -----	10
FIGURE I.9 : principe de fonctionnement de l'apprentissage profond -----	11
FIGURE I.10 : architecture d'un reseau CNN-----	12
FIGURE I.11 : carte de caracteristiques apres le filtre de convolution -----	14
FIGURE I.12 : couche de sous-échantillonnage -----	15
FIGURE II.1 : Comparaison entre approche classique et TL-----	18
FIGURE II.2 : Illustration de l'apprentissage par transfert inductif -----	20
FIGURE II.3 : Differents types de transfert learning -----	20
FIGURE II.4 : Adaptation du transfert learning par les reseaux pre-entraines -----	21
FIGURE II.5 : Architecture alexnet -----	22
FIGURE II.6 : Architecture googlenet -----	23
FIGURE II.7 : Architecture resnet-50-----	24
FIGURE II.8 : La structure du reseau vgg16 -----	24
FIGURE III.1 : Architecture de fine tuning -----	27
FIGURE III.2 : Algorithme de SVM-----	28
FIGURE III.3 : Courbe roc-----	29
FIGURE III.4 : Matrice de confusion pour un modele 3 classes -----	30
FIGURE III.5 : Courbe de précision et sa perte pour la base 1 a l'aide d'alexnet -----	33
FIGURE III.6 : Taux d'erreur et de précision par alexnet -----	33

FIGURE III.7 : Matrice de confusion d'après Alexnet)	34
FIGURE III.8 : Courbe de précision et sa perte de la base 1 à l'aide de GoogleNet	34
FIGURE III.9 : Taux d'erreur et de précision par GoogleNet	35
FIGURE III.10 : Matrice de confusion d'après (googlenet)	35
FIGURE III.11 : Courbe précision et sa perte pour de la base 1 a l'aide de resnet50	36
FIGURE III.12 : Taux d'erreur et de précision par resnet50	36
FIGURE III.13 : Courbe de précision et sa perte pour la base 1 à l'aide de vgg19	37
FIGURE III.14 : Taux d'erreur et de précision par vgg19	37
FIGURE III.15 : Matrice de confusion d'après vgg19	38
FIGURE III.16 : Courbe de précision et l'erreur de la base 2 à l'aide de resnet50	40
FIGURE III.17 : Taux d'erreur et de précision par resnet50	40
FIGURE III.18 : Matrice de confusion d'après resnet50	41
FIGURE III.19 : Courbe de précision et l'erreur de la base 2 a l'aide d'alexnet	41
FIGURE III.20 : Taux d'erreur et de précision par alexnet	42
FIGURE III.21 : Matrice de confusion d'après alexnet	42
FIGURE III.22 : Courbe de précision et l'erreur à l'aide de googlenet	43
FIGURE III.23 : Taux d'erreur et de précision par googlenet	43
FIGURE III.24 : Courbe de précision et l'erreur a l'aide de vgg19	44
FIGURE III.25 : Taux d'erreur et de précision par vgg19	44
FIGURE III.26 : Courbe de précision et l'erreur de la base 3 à l'aide de vgg19	46
FIGURE III.27 : Taux d'erreur et de précision de la base 3 a l'aide de vgg19	47
FIGURE III.28 : Courbe de précision et l'erreur de la base 3 à l'aide de resnet50	47
FIGURE III.29 : Taux d'erreur et de précision de la base 3 à l'aide de resnet50	48
FIGURE III.30 : Courbe de précision et l'erreur de la base 3 à l'aide de googlenet	48
FIGURE III.31 : Taux d'erreur et précision de la base 3 à l'aide de googlenet	49
FIGURE III.32 : Courbe de précision et l'erreur de la base 3 à l'aide d'alexnet	49
FIGURE III.33 : Taux d'erreur et de précision de la base 3 à l'aide d'alexnet	50

LISTE DES TABLEAUX

TABLEAU III.1 : Matrice de confusion pour un modele 3 classes -----	22
TABLEAU III.2 : Resultats de précision de la base 1 par SVM-----	24
TABLEAU III.3 : Resultats de précision de la base 1 par finne tuning -----	24
TABLEAU III.4 : résultats de précision par extraction et SVM -----	31
TABLEAU III.5 : Resultats de précision de la base 2 par finne tuning -----	31
TABLEAU III.6 : Resultats de précision de la base 3 par SVM.-----	38
TABLEAU III.7 : Resultats de précision de la base 3 par finne tuning. -----	38

LISTE DES ABRÉVIATIONS

ANN: Artificiel Neural Network.

DL: Deep learning.

IA: Intelligence artificielle.

TL: Transfert learning.

LBP: Local binary pattern.

HOG: Histogram Oriented gradient.

KNN: K Nearest Neighbors.

SVM : Support vector machine.

PMC : Perceptron Multicouches.

GPU : Unités de traitement graphique.

ReLU : Unité de rectification linéaire.

MLP : Le perceptron multicouche.

CONV : convolution.

POOL: Pooling.

2D: Two-Dimension.

CNN : Convolutionnel Neural Network.

TP : prédiction positive correcte.

TN : prédiction négative correcte.

FP : prédiction positive incorrecte.

FN: prédiction négative incorrecte

ROC: Receiver Operating Characteristic.

INTRODUCTION GÉNÉRALE

Introduction Générale

L'Apprentissage Profond (**Deep Learning- DL**) est un sous-domaine de l'Apprentissage Automatique (**Machine Learning- ML**) qui est lui-même un sous domaine de l'Intelligence Artificielle. Dans ce type d'approche, en observant les données, la machine est capable d'apprendre par elle-même et de représenter les connaissances de manière structurée. Par contre, elle se contente d'exécuter à la lettre des règles prédéterminées pour l'apprentissage automatique.

L'apprentissage profond s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. L'utilisation de ce type de réseaux n'est pas récente et remonte à la fin des années 50 [1]. Les travaux de Yan LeCun [2] dans les années 80 ont contribué aux développements d'un réseau particulier s'appelant réseaux de neurones convolutionnel. (Convolutionel Neural Network **CNN**). Ces derniers sont une forme particulière du réseau neuronal multicouche dont l'architecture de ses connexions est inspirée de celle du cortex visuel des mammifères où les valeurs des filtres sont ajustées au contenu d'intérêt. Les CNN ont été à la base de plusieurs applications aux débuts des années 90, comme la reconnaissance de chèques bancaires aux Etats Unis. Malgré leurs succès, les réseaux convolutionnels et les réseaux neuronaux plus généralement ont été délaissés par la communauté scientifique avant 2012. Depuis cette année et avec l'utilisation des GPU (Graphical Process Unit) à un prix moins coûteux et avec la disponibilité de grande base de données, l'apprentissage profond a envahi le monde de l'Intelligence Artificielle.

L'apprentissage profond regroupe actuellement les méthodes les plus efficaces et les plus performantes appliquées pour résoudre les problèmes liés aux ML. Ce nouveau paradigme a considérablement amélioré les résultats dans de nombreux domaines tels que la vision par ordinateur, la reconnaissance de la parole, la traduction automatique, la reconnaissance et classification d'images.

Notre étude s'inscrit dans le cadre de la classification des images en utilisant l'apprentissage profond. Pour se faire nous avons utilisé trois bases de données disponibles sur la plateforme Kaggle [3]. Nous avons utilisé l'approche d'apprentissage par transfert qui constitue une branche du DL. Cette dernière représente une solution au manque du grand nombre de données dont le DL exige.

Pour ce faire, nous avons structuré notre manuscrit en trois chapitres :

INTRODUCTION GÉNÉRALE

- le premier chapitre intitulé « *Apprentissage Profond* » présente le principe de base de l'apprentissage automatique suivi d'une description détaillée de l'apprentissage profond, de l'idée de base jusqu'à sa forme actuelle ainsi que toutes les notions qu'en sont liées.

- le deuxième chapitre décrit le principe de base de l'apprentissage par transfert ainsi que ses différents types.

- le troisième chapitre expose les principaux résultats obtenus de la classification par deux types d'approches d'apprentissage par transfert qui sont l'apprentissage fin (Fine tuning) et approche par extraction de caractéristique plus le classifieur SVM (Singular Vector Machine).

Une conclusion générale résume les résultats obtenus en cours de ce travail et perspectives envisagées.

**CHAPITRE I : *APPRENTISSAGE
PROFOND (DEEP LEARNING)***

I.1 Introduction

L'apprentissage profond ou Deep Learning (DL) est une branche de l'apprentissage automatique ou Machine Learning (ML) qui est lui-même engendré du monde de l'Intelligence Artificielle (Figure 1.1). C'est un domaine de recherche très récents avec une popularité fulgurante. Ceci est due à ses performances impressionnantes sur plusieurs problèmes longtemps considérés comme très difficiles ainsi que par le très grand nombre de techniques facilitant grandement son utilisation.

Ce chapitre, introduit premièrement l'apprentissage automatique avec ces concepts de base et sont nécessaire pour présenter en détail l'apprentissage profond. On commence tout d'abord par le neurone artificiel (unité de base) et on arrive aux réseaux profonds en passant par quelques éléments de mathématique sur lesquels ces techniques se basent.

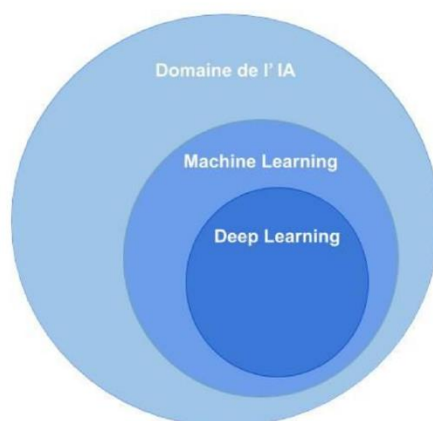


Figure I. 1 : Relation entre IA, ML, DL

I.2 Apprentissage Automatique

L'apprentissage automatique (*Machine Learning- ML*) est utilisé pour résoudre les problèmes qui sont difficiles à formuler. Le but est d'apprendre à un système à effectuer des tâches de façon autonome à partir des données. En effet, la contribution humaine est remplacée par la machine qui va apprendre à reconnaître automatiquement des modèles dans une grande quantité de données d'entraînement et qui seront à la base par la suite pour résoudre le problème posé ; reconnaissance et détection d'objet, classification....

Dans ce cas, la chaîne de traitement du ML se base sur deux grandes étapes ; l'extraction, la sélection des caractéristiques et la classification.



Figure I. 2 : Chaîne de traitement par apprentissage automatique

-**La première étape :** c'est l'étape d'extraction de caractéristiques qui consiste à calculer des informations représentatives de bords, contours et motifs de texture. Pour ce faire, plusieurs outils sont utilisés, on peut citer :

- Gradients : le gradient est calculé pour chaque pixel, afin de caractériser les contours dans une image.
- Local binary pattern (LBP) [4] : un codage binaire est utilisé pour caractériser les motifs répétitifs dans une image.
- Histogram Oriented gradient (HOG) [5] : Les gradients sont calculés pour chaque pixel dans l'image, puis ils sont regroupés selon l'orientation des angles. Ces groupes d'orientations de gradients sont représentés par des histogrammes. Ces méthodes sont invariantes au changement de luminosité ou d'ombre.
- Réseaux de neurones artificiels : ensemble de neurones formels interconnectés permettant la résolution de problèmes complexes tels que la reconnaissance des formes ou le traitement du langage naturel, grâce à l'ajustement des coefficients de pondération dans la phase d'apprentissage.

La deuxième étape : elle consiste à la classification, les méthodes de classification sont appliquées sur les caractéristiques extraites de l'étape précédente pour former un modèle représentatif et prédire la solution du problème posé. Différents classificateurs sont utilisés, on cite le KNN (K Nearest Neighbors), SVM (Support Vector Machine) [6]. Ces classificateurs sont utilisés pour projeter les caractéristiques dans un espace plus développé afin de trouver la bonne séparation entre les différentes classes.

Pour utiliser l'apprentissage machine, deux éléments sont essentiels : un modèle et un jeu de données. Le modèle est construit par les deux étapes déjà citées. Le jeu de données contient les données qui seront utilisées pour l'entraînement. Le plus souvent, ces données sont annotées, donc on parle de l'apprentissage supervisé mais certains jeux de données ne comportent pas d'annotation et permettent de faire de l'entraînement dit non-supervisé

Les chaînes de traitement à base de ML sont très performantes. Cependant, leur majeur inconvénient est le manque de généralisation du modèle d'apprentissage. En effet, les techniques de classification fonctionnent dans un contexte bien précis. Cependant, les solutions à base de Deep Learning sont capables de produire un modèle plus généralisé, capable de s'adapter à divers scénarios [7].

I.3 Réseaux de Neurones Artificiels

Les réseaux de neurones appartiennent aux algorithmes d'apprentissage automatique, fortement inspiré du fonctionnement du système nerveux biologique humain. Les premiers chercheurs innovateurs de ce domaine sont Warren McCulloch et Walter Pitts en 1943 [8] (invention de l'unité logique de seuil) et le perceptron de Frank Rosenblatt en 1957 [9].

Dans cette partie, nous allons décrire les différents concepts qui vont nous permettre par la suite d'aborder l'objectif du sujet, qui est le Deep Learning.

I.3.1 Neurone biologique

Les neurones sont les unités ou les cellules de base du cerveau humain. Un petit morceau de cerveau de la taille d'un grain de riz contient plus de 10 000 neurones, chacun formant en moyenne 6 000 connexions avec d'autres neurones. Fondamentalement, les neurones sont optimisés pour recevoir des informations d'autres neurones, traiter ces informations de manière unique et envoyer leurs résultats à d'autres cellules. La Figure 1.3, résume ce processus. :

- Les **dendrites** : c'est par eux que se fait la réception des signaux
- Le corps de la cellule ou le **noyau** : c'est l'unité de traitement
- L'**axone** : c'est la partie où passent les messages accumulés dans le corps de la cellule.
- Les **synapses** : sont des points de connexion par où passent les signaux de la cellule des terminaisons de l'axone vers les dendrites d'un autre neurone.

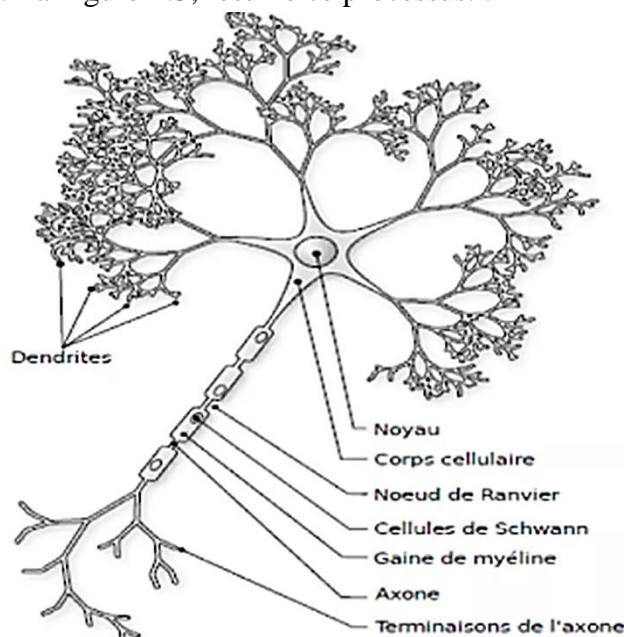


Figure I. 3 : Neurone biologique

I.3.2 Le neurone formel

Les réseaux de neurones artificiels (ANN) sont une classe d'algorithmes d'apprentissage automatique, fortement inspiré du fonctionnement des systèmes nerveux biologiques. L'ANN se compose d'un grand nombre de petites unités interconnectées qui s'appellent **Neurones artificiels ou formels**. Leur fonctionnement est inspiré de son homologue biologique (Figure I. 4). Un neurone artificiel ou formel est alimenté par un vecteur d'entrée X de dimension d , $X = (X_1, X_2, \dots, X_d)$ dont un poids de connexion W_i (weight) est associé à chaque attribut du vecteur. Les attributs en entrée ainsi que leurs poids de connexion d'un neurone formel correspondent respectivement aux dendrites et synapses sur son homologue biologique.

La fonction de combinaison d'un neurone est une fonction d'accumulation des différentes entrées de ce neurone émettant comme résultat leur somme pondérée.

Enfin, la somme calculée sera envoyée à une fonction d'activation pour générer la valeur de la sortie du neurone. La fonction d'activation reliée à la sortie traduit le noyau et l'axone sur la structure biologique d'un neurone

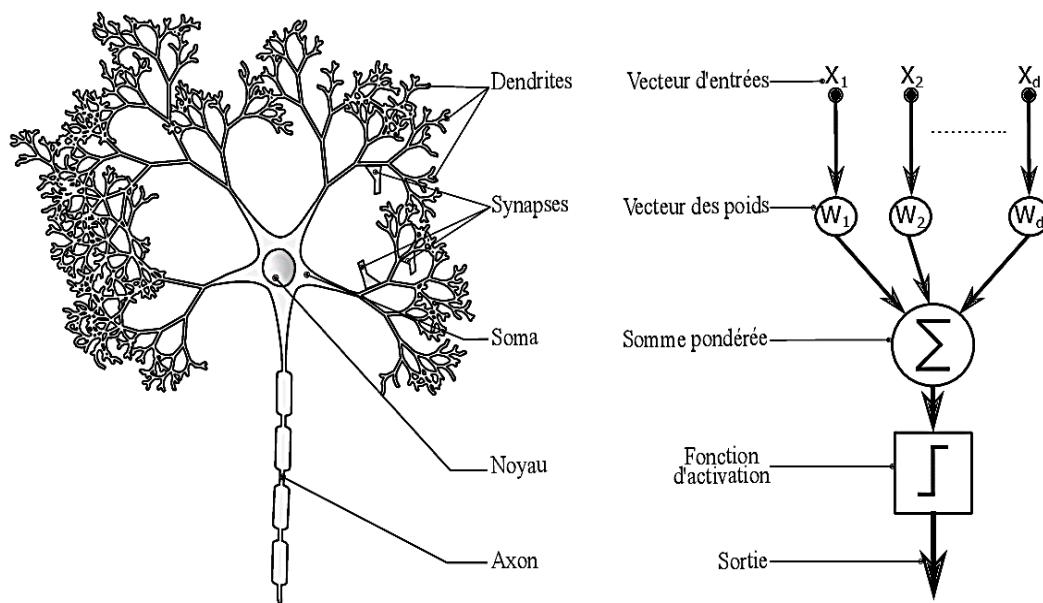


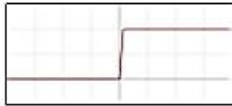
Figure I. 4 : Structure d'un neurone biologique contre un neurone artificiel

- **Fonction d'activation :** c'est une fonction non linéaire qui opère une transformation d'une combinaison affine des signaux d'entrée x_i , et de variables constantes b_i , appelé le biais du neurone. Cette fonction F représente la sortie du neurone et elle est déterminée par un vecteur de poids W_i pour chaque entrée x_i . Sa forme générale est donnée par l'équation (1) :

$$y = F\left(\sum_{i=0}^N W_i \cdot X_i + b\right) \quad (1)$$

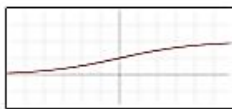
Différentes formes existent pour la fonction d'activation, les plus utilisées sont donnés comme ci-dessous :

Fonction "Marche/Heaviside"



$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$$

Fonction Sigmoide



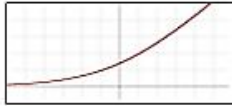
$$f(x) = \frac{1}{1 + e^{-x}}$$

Fonction "Unité de rectification linéaire" (ReLU)



$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$

Fonction "Unité de rectification linéaire douce" (SoftPlus)



$$f(x) = \ln(1 + e^x)$$

La fonction ReLU est la plus simple et la plus utilisée. Cependant, il est important de préciser que la fonction doit être suffisamment continue et dérivable pour que nous puissions calculer le gradient. De plus, si la fonction est la solution d'une équation différentielle de la forme (équation 2), cela va permettre de simplifier des calculs.

$$f'(x) = F(f) \quad (I. 2)$$

I.3.3 Le Perceptron

Le Perceptron est le modèle de réseau neuronal monocouche le plus simple. Il se compose de la couche d'entrée et de sortie. Il est défini comme la plus petite unité d'apprentissage des réseaux de neurones artificiels couche inventée par Frank Rosenblatt en 1957[10]. La figure I. 5 illustre son principe de fonctionnement dont tous les neurones d'entrées sont liés à tous neurones de sorties. Il permet de résoudre des problèmes linéaires comme les

fonctions OU, ET logique chose qui constitua une limitation importante. C'est un réseau de type Feedforward, c'est-à-dire que l'information ne se propage que dans un sens, de la couche d'entrée vers la couche de sortie. Son algorithme d'apprentissage pour les poids w_i est extrêmement simple à programmer est basé sur l'équation :

$$w_i = w_i + \alpha(y_d - y)x_i \quad (I.3)$$

Avec : α paramètre d'apprentissage ou Learning rate, y_d la sortie désirée, y la sortie calculée. Cependant, sa capacité de généralisation à de nouvelles situations (non vues durant l'apprentissage) est mauvaise du fait des limitations induites par sa structure à une seule couche [11].

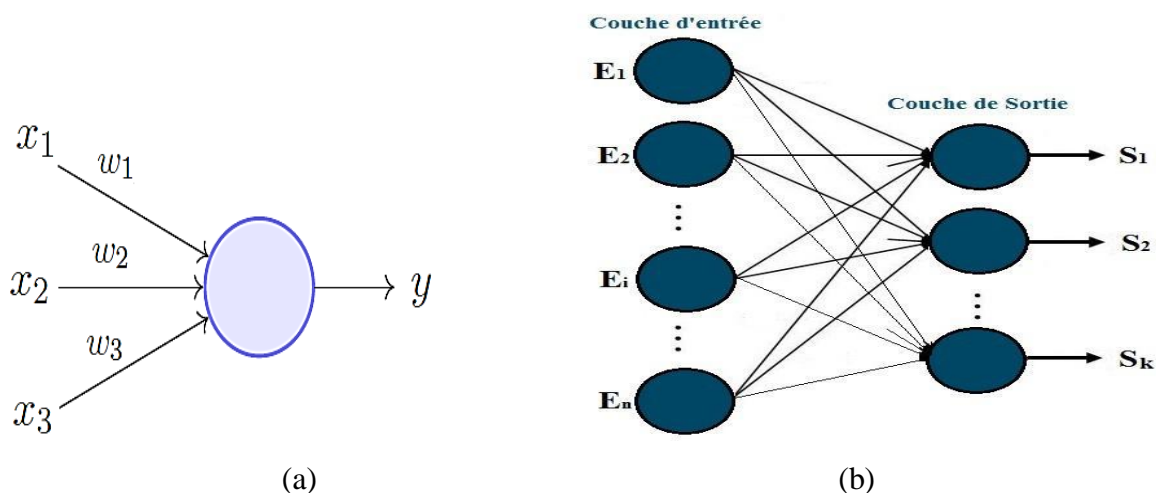


Figure I. 5 : Perceptron (a) une seule sortie, (b) k sorties

I.3.4 Perceptron Multicouches (PMC)

Dans le Perceptron Multicouches PMC (figure I. 6), chaque neurone des couches cachées est connecté à tous les neurones de la couche qui la précède et celle qui la suit. L'idée du réseau PMC est ancienne (années soixante) par rapport à son exploitation. Ce retard est due à l'absence d'un algorithme d'apprentissage des poids pour les couches cachées. C'est jusqu'au 1985 que les limitations d'un apprentissage restreint à une seule couche de neurones (ou Perceptrons) soient levées. C'est les travaux de Yann LeCun [2] qui ont mené à l'apprentissage par rétropropagation (Backpropagation) utilisant l'algorithme de la descente du gradient.

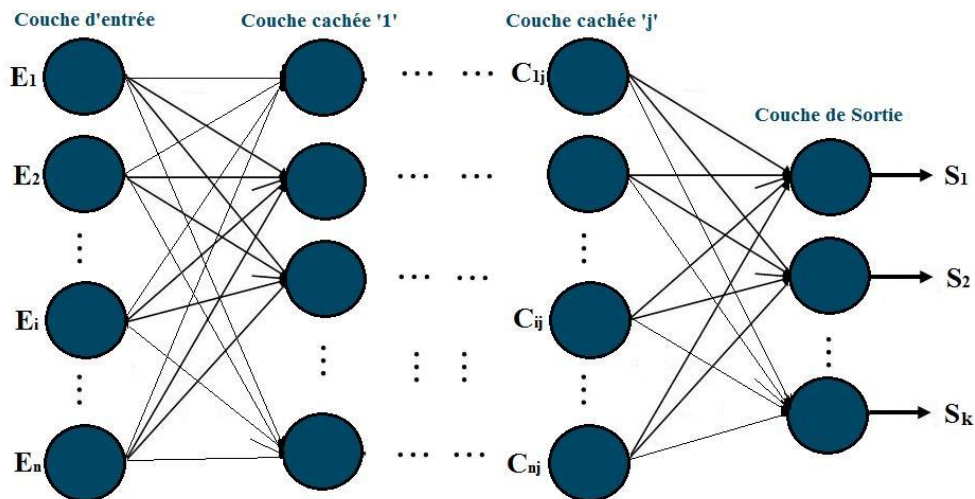


Figure I. 6 : Perceptron Multicouches (PMC)

I.3.5 Rétropropagation

Technique qui permet de mettre à jour les poids des différentes couches du réseau très efficacement. Elle est utilisée en conjonction avec une méthode d'optimisation de descente de gradient. Fondamentalement, la rétropropagation n'est qu'une approche qui permet de calculer les dérivées d'une fonction coût de l'erreur complexe afin d'utiliser ces dérivées pour apprendre et ajuster les coefficients de pondération du réseau PMC (figure I.7). L'algorithme peut être résumé comme suit :

1. Propagation du réseau de l'entrée jusqu'à la sortie
2. Calcul de l'erreur en sortie
3. Rétropropagation de l'erreur jusqu'aux entrées à condition que la fonction d'activation soit dérivable car la rétro-propager l'erreur est basée sur sa dérivée.

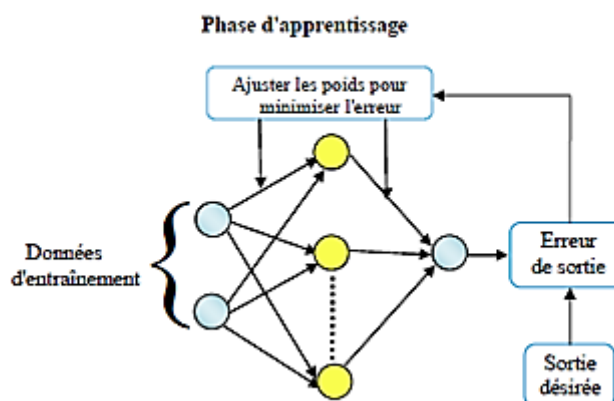


Figure I. 7 : apprentissage des poids du PMC

I.4 Apprentissage profond

Apprentissage profond peut être considéré comme la famille élargie de l'apprentissage automatique (ML), qui permet de faire apprendre à un système une tâche précise en observant un grand nombre d'exemples. La différence entre ces deux types d'apprentissage est que le DL va être entraîné pour sortir lui-même les éléments influents dans la prédiction souhaitée. Par contre l'apprentissage automatique prévoit des algorithmes pour faire sortir ces éléments (features), la figure I. 8 illustre cette différence. En plus, la nature des données traitée par les types d'apprentissage est différente. On trouve en ML des données quantitatives et structurées (des valeurs numériques), lorsque ceux apprentissage profond des données non-structurées, comme le son, le texte, l'image.

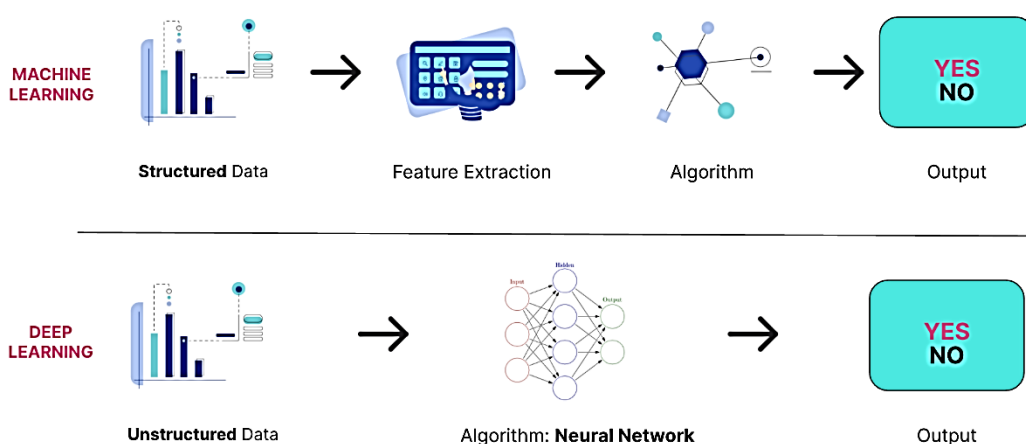


Figure I. 8 : Différence entre apprentissage machine et apprentissage profond [12]

En effet l'apprentissage profond est basé sur le réseau de neurones multicouches PMC dont son nom vient du fait que plus un réseau a un nombre élevé de couches, plus il est dit profond. Les PMC existent depuis les années 1980, mais au cours des dernières années, la croissance de la puissance des calculs surtout avec l'avènement des unités de traitement graphique (GPU) et la disponibilité de grandes bases de données ont accru la popularité du réseau de neurones profond.

La figure I. 8 présente en détail l'architecture de base d'un réseau DL convolutionnel dont nous allons l'utiliser au cours de ce travail. Ce type de réseaux est spécifique au traitement d'image. Il faut noter qu'il existe d'autres architectures DL comme les réseaux récurrents...

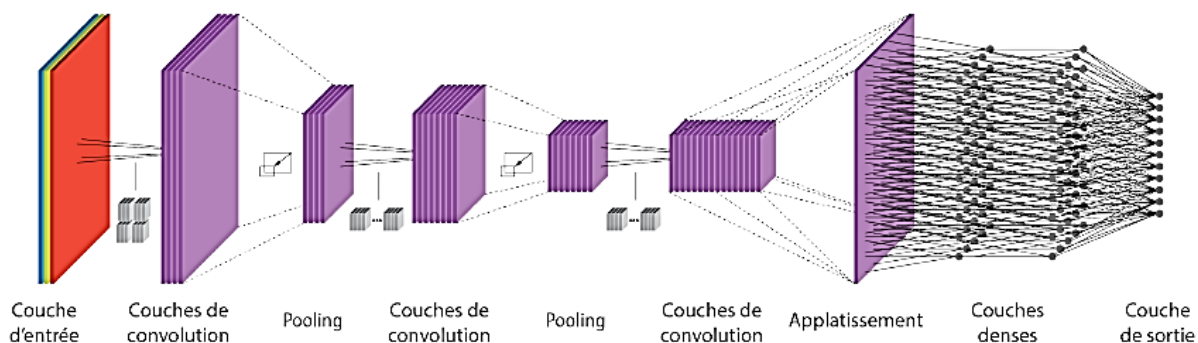


Figure I. 9 : Principe de fonctionnement de l'apprentissage profond

I.4.1 Domaines d'application de l'apprentissage profond

L'apprentissage profond a permis d'obtenir des résultats impressionnants dans différents domaines :

- Reconnaissance d'image,
- Traduction automatique,
- Voiture autonome,
- Diagnostic médical,
- Recommandations personnalisées,
- Modération automatique des réseaux sociaux,
- Prédiction financière et trading automatisé,
- Identification de pièces défectueuses,
- Détection de malwares ou de fraudes,
- Exploration spatiale,
- Robots intelligents.

La liste des applications ne peut être clôturer car il existe encore beaucoup d'exemple plus originaux les uns que les autres du faite que le DL tente toujours de simuler le modèle de réflexion humaine. Pour conclure ; il faut retenir c'est que le Deep Learning permet de faire apprendre à un ordinateur une tâche précise en observant un grand nombre d'exemples.

I.5 Réseaux de neurones convolutionnels

Les réseaux de neurones convolutionnels sont les plus impressionnants des réseaux de neurones artificiels. Ils sont principalement utilisés pour résoudre des tâches liées à la reconnaissance, la détection de formes, et la classification comme le cas de notre travail. Le

CNN représente l'élément clé de l'apprentissage profond. Son idée de base a été inspirée de la caractéristique du cortex visuel animal connue sous le nom de champ récepteur [13]. Les champs récepteurs agissent comme détecteurs sensibles à certains types de stimuli, par exemple les bords. Dans le traitement d'image, mêmes effets de détections peuvent être générés par filtrage convolutif.

L'entrée du réseau CNN est l'ensemble des images de la base de données à traiter. Chaque image est représentée sous la forme d'une matrice de pixels à 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La sortie du réseau dépend de l'application visée. Pour le cas de notre travail qui est la classification de différentes bases de données c'est les classes prédites pour chaque image.

Le réseau CNN est construit par la succession d'une couche de convolution et d'une couche d'agrégation de l'information (Pooling) et se termine par une couche de neurones totalement connectés.

L'architecture du réseau comporte deux parties bien distinctes :

- La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN. (Figure I. 10)

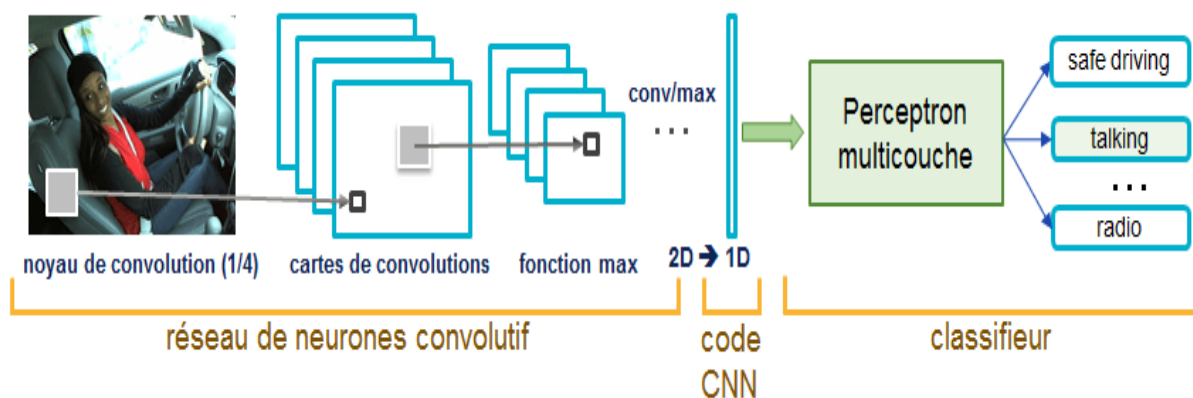


Figure I. 10 : architecture d'un réseau CNN [14]

- la deuxième partie est constituée de couche entièrement connectées ou perceptron multicouche (Fully Connected). Cette partie joue le rôle du classifieur qui permet de combiner les caractéristiques du code CNN pour classer l'image.

Enfin la sortie du réseau est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

I.5.1 Couche de convolution

La couche convolutionnelle joue un rôle important dans le fonctionnement de CNN. Les paramètres de cette couche se concentrent sur de petites fenêtres $n \times n$ connues sous le nom de noyau, kernels ou des filtres. La couche convolutive transforme chaque filtre à travers la dimension spatiale de l'entrée pour produire une carte d'objet 2D (feature map) ou carte d'activation après que la fonction d'activation a été calculée.

Mathématiquement parlant, l'opération de convolution est définie comme une intégrale représentant la quantité de chevauchement de l'image I avec une fonction K. Etant donné, considérant que l'image est définie comme une fonction discrète, l'opération de convolution discrète entre une image I et une matrice de filtre K est définie par :

$$F(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i-m, j-n)K(m, n) \quad (I.4)$$

La carte des caractéristiques est définie comme la matrice de sortie résultante de l'opération de convolution $I[n] * K[n]$. C'est -à dire le résultat de l'alignement du filtre K avec chaque sous-image de I à tour de rôle (même taille que K) centré sur le pixel central de la fenêtre de coordonnées x, y. La figure I. 9 illustre le principe de fonctionnement de l'opération de convolution avec deux masques de taille 3*3 et par la suite l'obtention de deux cartes de caractéristiques.

La couche de convolution possède des hyper-paramètres qui définissent le modèle, spécialement dans on doit définir :

- **taille de filtre** : La taille de filtre peut être n'importe quelle taille $n \times n$ avec $n > 2$ et n doit être impaire (3x3, 5x5, 7x7, etc.) ;
- **nombre de filtre par couche** : Il peut être (32, 64, 128, 256, etc.) ;
- **pas** : C'est le nombre de pixels à déplacer en appliquant le filtre sur l'image en entrée.
- **Zero Padding** : C'est le nombre de pixels (de valeur zéro) à ajouter à l'image en entrée pour contrôler la taille de sortie de convolution ;

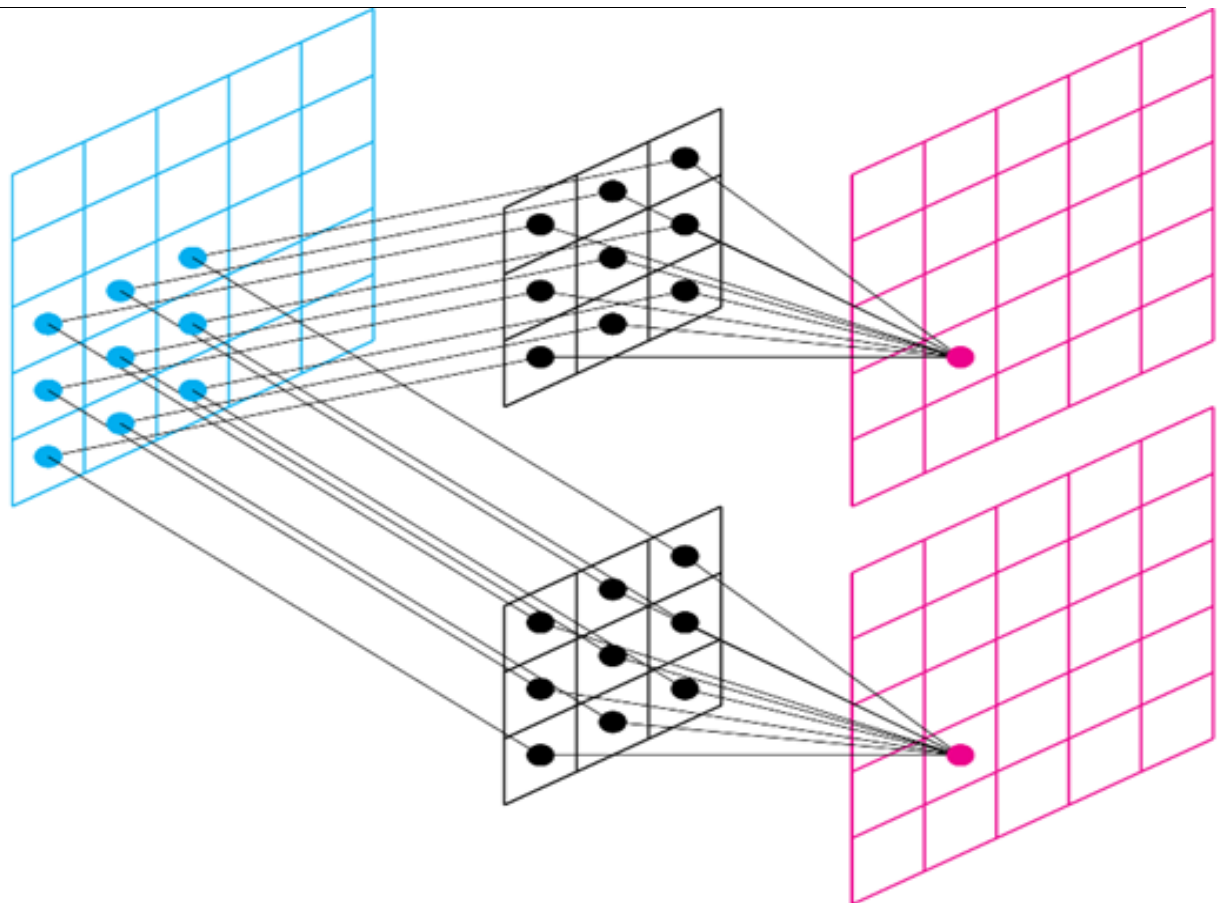


Image ou carte d'entrée

2 masques convolutionnels

2 cartes de caractéristiques de sortie

Figure I. 11 : Carte de caractéristiques après le filtre de convolution

I.5.2 Fonction d'activation

La sortie de chaque couche convolutionnelle est envoyée à une couche de fonction d'activation. La couche de fonction d'activation se compose d'une fonction d'activation qui prend la carte de caractéristiques produite par la couche de convolution et génère la carte d'activation comme sortie. La fonction d'activation est utilisée pour transformer le niveau d'activation d'un neurone en un signal de sortie. De nombreuses fonctions d'activation sont utilisées, En pratique, on utilise le ReLU (vue dans la section I. 3. 2) comme activation pour les couches convolutionnelles cachées car cette dernière accélère le processus d'entraînement.

I.5.3 Couche de sous-échantillonnage (pooling layer)

Les couches de sous-échantillonnage permettent de diminuer le nombre de paramètres et la complexité de calcul du modèle. Cette diminution est obtenue à travers la réduction de la taille des cartes d'activation par des fonctions de mise en commun maximale ou moyenne. La méthode de mise en commun la plus utilisée est la méthode de sous-échantillonnage par valeur moyenne (max-pooling). Cette méthode génère la valeur maximale d'un voisinage

rectangulaire de la carte d'activation. Il en est de même pour la moyenne sauf qu'il produit la moyenne. Son principe est représenté sur la figure (1. 10) avec la foulée 2 (la foulée est un paramètre qui représente le nombre de pixels de la fenêtre).

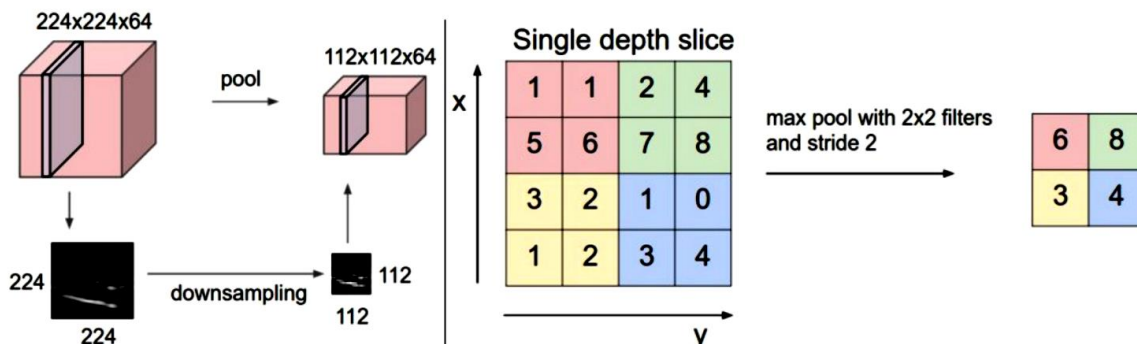


Figure I. 12 : couche de sous-échantillonnage

I.5.4 Couches entièrement connectées

Les couches entièrement connectées représentent la partie de classification où chaque neurone est connecté à chaque neurone de la couche suivante et chaque valeur contribue à prédire à quel point une valeur correspond à une classe particulière. La fonction d'activation utilisée dans la couche de sortie destinée à donner le résultat de la classification est différente à celle utilisée dans les couches cachés (ReLU). C'est la fonction Softmax qui est utilisée pour un problème de classification multiclassés comme le cas de notre travail. Elle est donnée par l'équation suivante :

$$\sigma(\vec{z}_i)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \tag{I.5}$$

Avec :

- σ = softmax
- \vec{z} = input vector
- e^{z_i} = standard exponential function for input vector
- K = number of classes in the multi-class classifier
- e^{z_j} = standard exponential function for output vector
- e^{z_j} = standard exponential function for output vector

I. 5. 5 Entraînement du réseau CNN

L'entraînement d'un CNN consiste alors à optimiser les coefficients du réseau, à partir d'une initialisation aléatoire, pour minimiser l'erreur de classification en sortie. Les deux parties des CNN sont entraînées simultanément : on apprend à la fois les coefficients des

noyaux de convolutions pour extraire des caractéristiques pertinentes, et la bonne combinaison de ces caractéristiques.

En pratique, les coefficients du réseau sont modifiés de façon à corriger les erreurs de classification rencontrées, selon la méthode de descente de gradient comme ça été déjà mentionné dans la section (1.3.4). Ces gradients sont rétropropagés dans le réseau depuis la couche de sortie.

L'entraînement du réseau se fait par batch consiste à rétropropager l'erreur de classification par groupes d'images. Cette méthode est plus rapide qu'en calculant l'erreur sur tout le jeu d'entraînement à chaque itération. Elle est plus stable qu'en travaillant image par image, car les gradients d'erreurs ont moins de variance. À noter qu'un nombre trop important d'images par batch peut engendrer des problèmes de mémoire lors de l'exécution du code.

I.6 Hyperparamètres d'un réseau DL

Le fonctionnement d'un réseau DL nécessite plusieurs termes qu'il faut les bien définir pour bien comprendre son mécanisme de fonctionnement. Le **Learning rate**, **Batch** et **Epoch** sont des hyper-paramètres dans le **DL** avec des valeurs entières utilisées par le modèle d'apprentissage. Ces valeurs ne sont pas découvertes par un processus d'apprentissage car elles ne sont pas des paramètres de modèle internes mais elles doivent être spécifiées pour le processus lors de la formation de l'algorithme d'apprentissage sur l'ensemble de données de formation :

- **Learning rate** : (le taux d'apprentissage) c'est un hyperparamètre qui contrôle le degré de modification du modèle en réponse à l'erreur estimée chaque fois que les poids du modèle sont mis à jour. Le choix du taux d'apprentissage est difficile car une valeur trop petite peut entraîner un long processus d'entraînement qui pourrait rester bloqué, tandis qu'une valeur trop grande peut entraîner l'apprentissage trop rapide d'un ensemble de poids sous-optimal ou un processus d'entraînement instable.

- **Epoch** : dans l'apprentissage automatique, un transit complet des données d'apprentissage à travers l'algorithme est appelé une époque. Le nombre d'époque est un hyperparamètre critique pour l'algorithme. Il spécifie le nombre d'époques ou de passages complets de l'ensemble de données d'apprentissage tout au long du processus d'apprentissage ou d'apprentissage de l'algorithme. Les paramètres du modèle interne de l'ensemble de données sont mis à jour à chaque époque.

- **Batch** : Le lot est le nombre d'échantillons ou d'exemples d'apprentissage dans une itération. Plus la taille du lot est élevée, plus nous avons besoin d'espace mémoire.

- **Itérations** : les itérations sont le nombre de lots ou d'étapes à travers des paquets partitionnés des données d'apprentissage, nécessaires pour terminer une époque.

Pour conclure, soit un ensemble d'entraînement de 1 million d'images au total, qui représente un gros ensemble de données pour les alimenter toutes à la fois sur le réseau. En plus, lors de l'entraînement du réseau, la taille des données ne cesse d'augmenter en mémoire. Nous allons donc diviser l'ensemble de données en parties ou batch. Pour une taille du batch égale à 50K, cela signifie que le réseau a besoin de 20 itérations ($1M/50K$) pour terminer une époque.

I.7 Conclusion

Dans ce premier chapitre nous avons présenté une description théorique sur l'apprentissage automatique, relatif à l'apprentissage profond en détaillant son architecture basée sur les réseaux convolutifs.

Dans le chapitre qui suit nous allons présenter l'apprentissage par transfert ; technique basée sur l'apprentissage profond et qui sera utilisée dans le cadre du travail proposé dans ce mémoire.

CHAPITRE II :
APPRENTISSAGE PAR
TRANSFERT (TRANSFER
LEARNING)

II.1 Introduction

La popularité de l'apprentissage par transfert ou *Transfer Learning* (TL) dans le domaine de l'apprentissage automatique a explosé ces dernières années. L'une des principales raisons de son succès est la capacité d'utiliser les connaissances acquises lors de l'exécution d'une tâche pour résoudre d'autres problèmes complètement différents. Dans ce chapitre nous allons détailler le scénario de transfert Learning, présenter les différentes architectures de réseaux de neurones convolutifs pré-entraînés (Convolutional Neural Network- CNN-) dont il est basé. En plus, nous allons examiner les fondements de la classification par TL qui constitue l'objectif de l'étude abordée dans ce mémoire

II.2 Apprentissage par transfert (TL)

Apprentissage par transfert désigne l'ensemble des méthodes permettant de transférer les connaissances acquises lors de la résolution d'un problème donné vers un autre problème. En d'autres termes, l'apprentissage par transfert repose sur une idée simple, celle de ré-exploiter les connaissances acquises dans d'autres configurations (**sources**) pour la résolution d'un problème particulier (**cible**).

Avec le développement de l'apprentissage profond (DL), le TL a connu un grand succès. En effet, les modèles utilisés dans le domaine du DL nécessitent généralement des temps de calcul très élevés et beaucoup de ressources. Cependant, en utilisant des modèles pré-entraînés comme point de départ, l'apprentissage par transfert permet de développer rapidement des modèles performants et de résoudre efficacement des problèmes complexes en vision par ordinateur. (La figure II. 1) illustre une comparaison entre l'approche d'apprentissage classique et l'approche par TL.

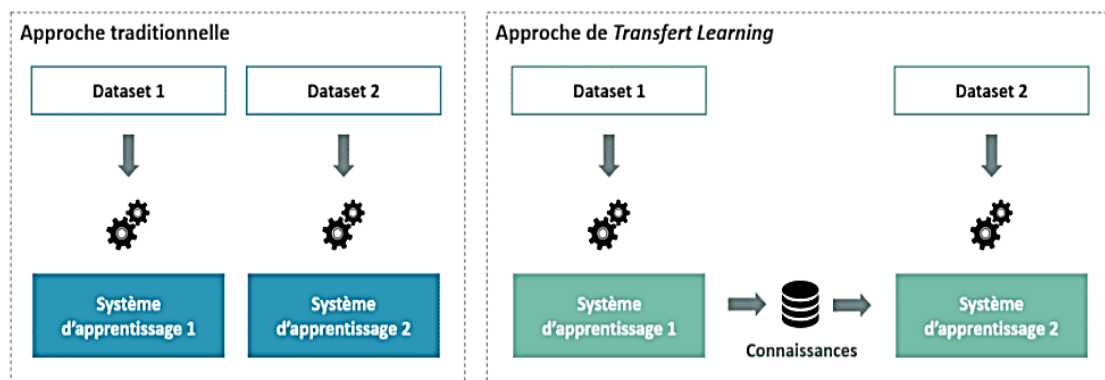


Figure II. 1 : comparaison entre approche classique et TL

II.3 Principe de base

De nombreux réseaux de neurones profonds entraînés sur des images naturelles présentent un phénomène curieux en commun : sur la première couche, ils apprennent des fonctionnalités similaires aux filtres de Gabor et à la reconnaissance des couleurs [15]. Ces fonctionnalités des premières couches ne semblent pas spécifiques à un ensemble de données ou à une tâche en particulier, mais sont générales dans le sens qu'elles sont applicables à de nombreux ensembles de données et tâches. Étant donné que la recherche de ces fonctionnalités standard sur la première couche semble se produire indépendamment de la fonction de coût exacte et de l'ensemble de données d'image naturelle, alors ces fonctionnalités sont dites **fonctionnalités générales**. Par contre, dans les dernières couches, avec une couche de sortie Softmax à N dimensions qui a été entraînée avec succès vers un objectif de classification supervisée, chaque unité de sortie sera spécifique à une classe particulière. Dans ce cas ces fonctionnalités de la dernière couche sont dites **spécifiques**.

En pratique, très peu de personnes forment un réseau convolutif complet à partir de zéro car il est relativement rare d'avoir un ensemble de données de taille suffisante (des métadonnées). Au lieu de cela, il est courant de préformer le réseau convolutionnel sur un très grand ensemble de données (par exemple, ImageNet, qui contient 1,2 million d'images avec 1000 catégories), puis l'utiliser soit comme initialisation, soit comme extracteur de caractéristiques fixes pour la tâche d'intérêt. Pour le cas de notre travail, nous avons tenté d'utiliser ces deux approches sur différentes bases de données (voir chapitres résultats).

II.3.1 Types d'apprentissage par transfert

En fonction à la fois de la taille du nouvel ensemble de données et de la similitude du nouvel ensemble de données avec l'ensemble de données d'origine, l'approche d'utilisation de l'apprentissage par transfert sera différente. Dans ce contexte, on peut distinguer plusieurs approches selon ce que l'on souhaite transférer, quand et comment réaliser le transfert. Globalement, nous pouvons distinguer trois types de TL

a- Apprentissage par transfert inductif : dans cette configuration, les domaines source et cible sont les mêmes (même données), mais les tâches source et cible sont différentes mais proches. L'idée consiste alors à utiliser les modèles existants pour réduire de manière avantageuse le champ d'application des modèles possibles comme illustré dans la figure ci-dessous. Par exemple, il est possible d'utiliser un modèle entraîné pour la détection d'animaux sur des images pour construire un modèle capable d'identifier des chiens.

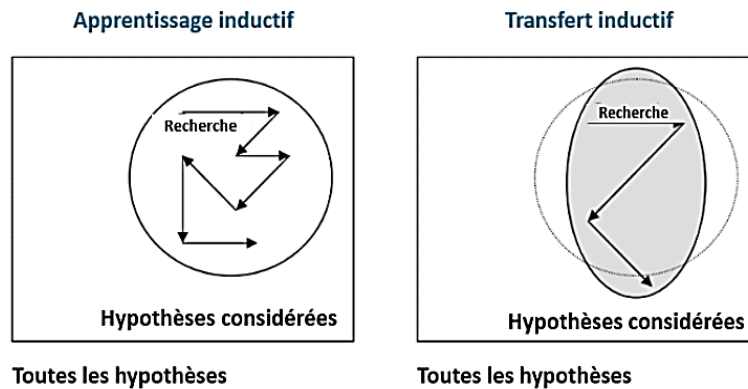


Figure II. 2 : Illustration de l'apprentissage par transfert inductif

b- Apprentissage par transfert non supervisé : dans ce type, les données des deux domaines (cible et source) ne sont pas labellisées. Par contre, comme dans le cas de l'apprentissage par transfert inductif, les domaines source et cible sont similaires, mais les tâches sont différentes. En effet, il est souvent plus facile d'obtenir de grandes quantités de données non labellisées, à partir de bases de données et de sources sur le web par exemple, que des données labellisées. C'est pourquoi l'idée d'utiliser l'apprentissage non supervisé en combinaison avec le Transfer Learning suscite un grand intérêt.

c- Apprentissage par transfert traductif : dans cette configuration, les tâches sources et cibles sont similaires, mais les domaines correspondants sont différents soit en termes de données ou de distributions de probabilités (nombres de classes).

La figure II. 3, représente les différents types de Transfert Learning. Le travail effectué dans le cadre de ce mémoire peut être classé dans le cadre de Transfert traductif ; les tâches sont similaires (classification) mais les données sources et cibles sont différentes soit en nature de données, soit en quantité de données.

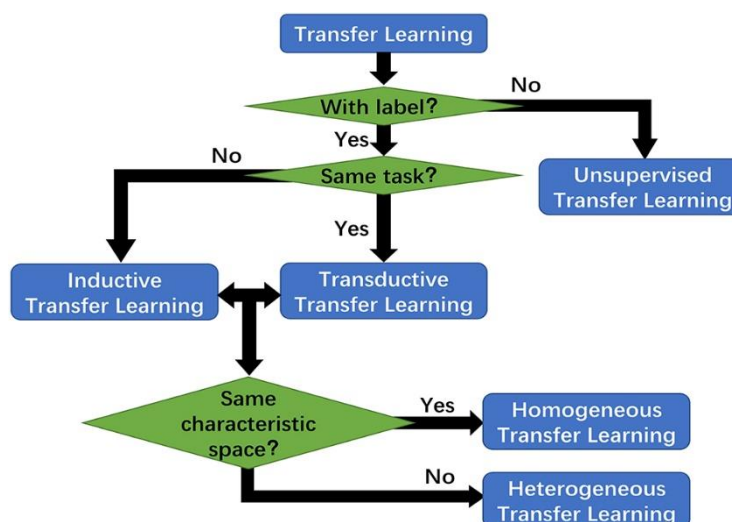


Figure II. 3 : Différents types d'apprentissage par transfert[16]

II.4 Apprentissage par transfert par adaptation des réseaux CNN Pré-entraînés

L'avantage majeur du TL est qu'il est possible d'exploiter la puissance des CNN sans être un expert du domaine, avec du matériel accessible et une quantité raisonnable de données annotées. Toute la complexité de création de CNN peut être évitée en adaptant des réseaux pré-entraînés disponibles publiquement. En d'autres termes, on va exploiter la connaissance acquise sur un problème de classification général pour l'appliquer de nouveau à un problème de classification particulier.

Cette "connaissance" sur la classification d'images contenue dans un réseau peut être exploitée de deux façons :

- **Extracteur automatique de caractéristiques** des images exploite uniquement la partie convolutive d'un réseau pré-entraîné matérialisé par le code CNN. Ce code sera ensuite utilisé comme entrée d'un classifieur classique comme les SVM (Support Vector Machine) utilisé dans ce travail et présenté dans le chapitre suivant.
- **Fine Tuning** : prend les paramètres du réseau comme une initialisation du modèle, qui est ensuite **pré-entraîné** plus finement pour traiter le nouveau problème de classification.

Le schéma de la figure II. 4, illustre les scénarios d'adaptation des réseaux pré-entraînés dans les deux cas pour le transfert learning.

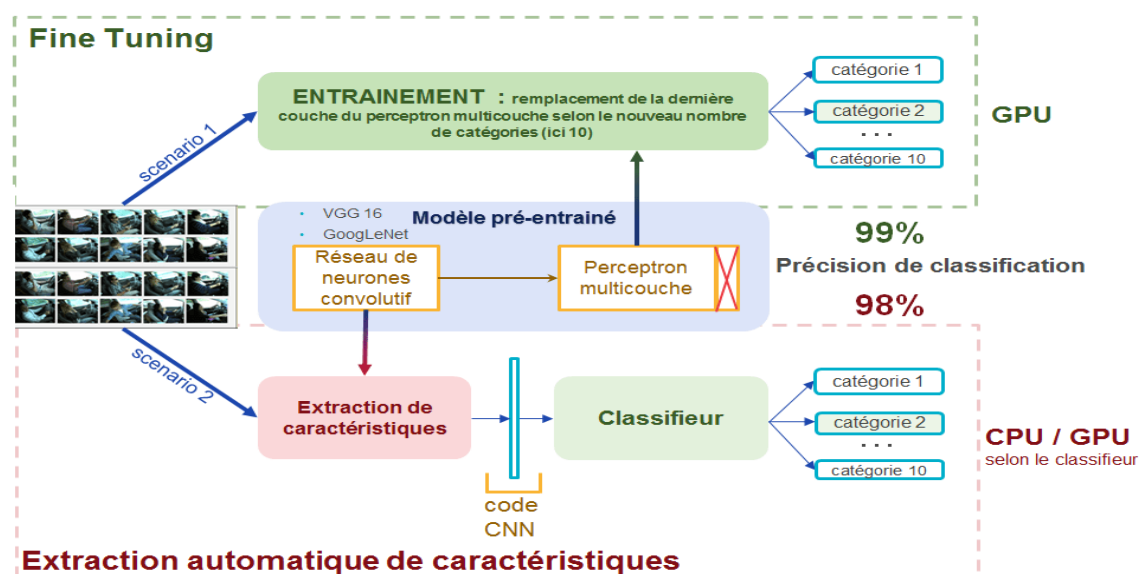


Figure II. 4 : adaptation du TL par les réseaux pré-entraînés [17]

II.5 Réseaux CNN pré-entraînés

La performance des réseaux de neurones profonds dépend de la quantité de données d'entraînement et de la puissance de calcul. Ainsi, la plupart des réseaux profonds utilisés en DL sont effectués sur de grandes ensembles de données accessibles au public. La base de données la plus célèbre dans ce domaine est la base Imagenet qui contient 1000 classes et comporte plus de 1 million d'images. Dans ce travail on s'est concentré sur 4 types de réseaux pré-entraînés : Alexnet, Googlenet, Resnet 50 et VGG. Dans les sections suivantes, nous allons examiner les architectures de ces réseaux.

II.5.1 AlexNet

Alexnet a été proposé par Krizhevsky et al [18]. AlexNet est le nom de l'architecture d'un réseau profond de neurones à convolutions qui a remporté l'épreuve ImageNet en 2012. Conçu par une équipe de l'Université de Toronto dirigée par Geoffrey Hinton, dont faisaient partie Alex Krizhevsky, architecte principal qui lui a donné son nom. En 2012 ce nouveau réseau a largement surperformé tous ses concurrents précédents et a remporté le défi en réduisant l'erreur de top 5 de 26% à 15,3%. AlexNet a eu un grand impact dans le domaine de l'apprentissage automatique, en particulier dans l'application de l'apprentissage en profondeur à la vision artificielle. Alex Net contenait huit couches ; les cinq premières étaient des couches convolutionnelles et les trois dernières étaient des couches entièrement connectées il a parallèlement deux lignes CNN formées sur deux GPU avec des connexions croisées et Il a utilisé la fonction d'activation non saturante de Relu, qui a montré une amélioration des performances d'entraînement par rapport au sigmoïde (voir figure II. 5).

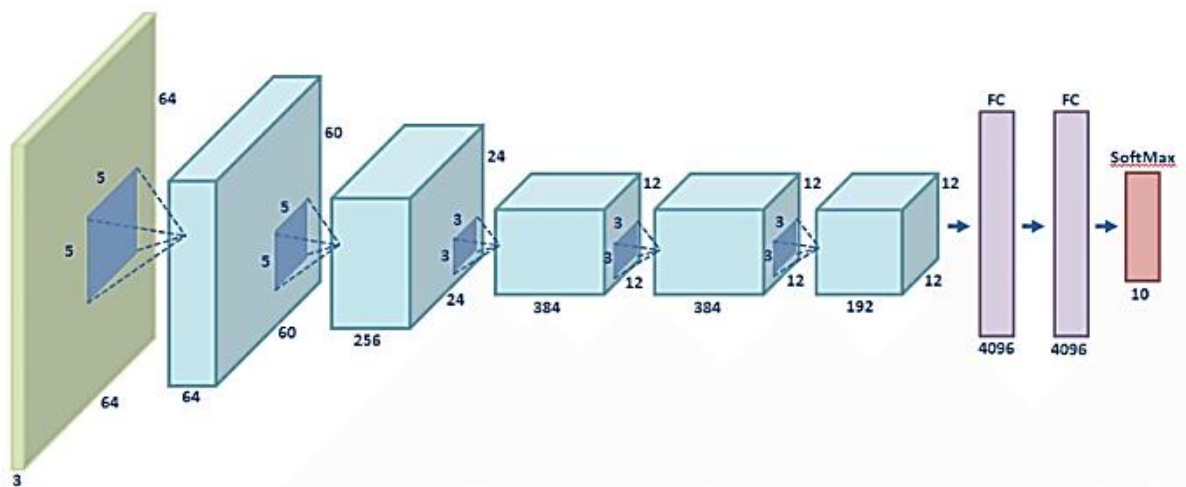


Figure II. 5 : Architecture AlexNet

II.5.2 Google Net/Inception

Google Net/Inception-v1 a été développé par les chercheurs de Google en 2014 et qui a remporté la compétition ILSVRC 2014 [19]. Google Net a 22 couches de profondeur avec près de 12 fois moins de paramètres (plus rapide, moins précis qu'Alex Net). Il a réduit le nombre de paramètres de 60 millions (Alex Net) à 4 millions. L'idée était de créer des modèles qui peuvent également être utilisés sur les smartphones (conserve un budget de calcul d'environ 1,5 milliard de multiplications par prédiction) Couche émissive : L'idée d'une couche émissive est de couvrir une plus grande surface tout en maintenant une résolution fine pour les petites informations d'image. L'idée est donc de combiner les différentes tailles en parallèle, de la taille la plus précise (1*1) à la taille la plus grande (5*5). Le moyen le plus simple d'améliorer les performances d'apprentissage en profondeur consiste à utiliser plus de couches et plus de données (voir Figure II. 6).

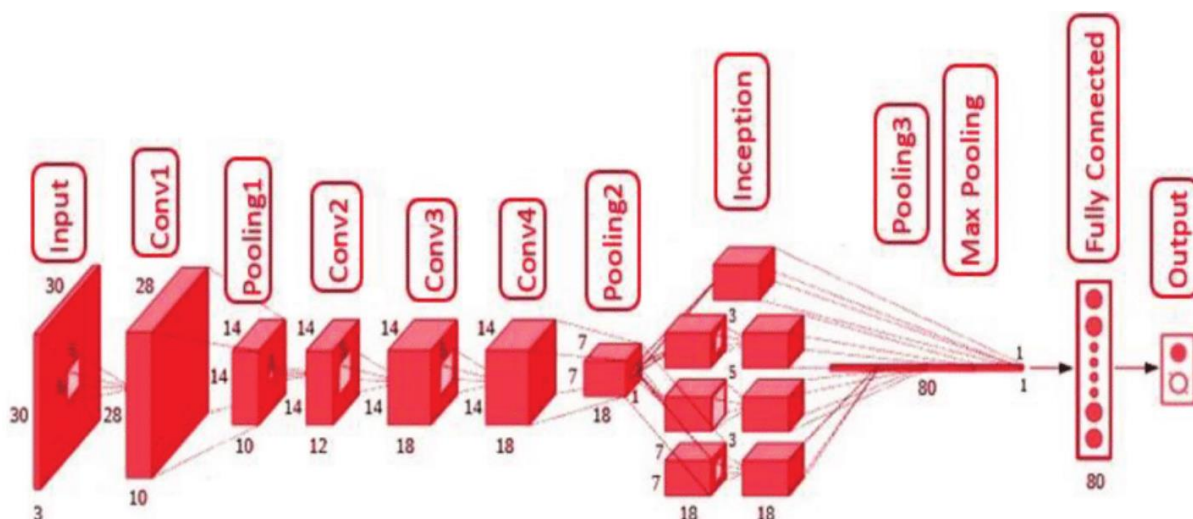


Figure II. 6 : Architecture GoogLeNet

II.5.3 ResNet-50

Les réseaux de neurones résiduels ResNet [20] introduisent une architecture innovante avec "saut de connexion" et normalisation par lots. Ces connexions, également connues sous le nom d'unités de déclenchement ou d'unités de déclenchement récurrentes, présentent une forte ressemblance avec les récents éléments réussis appliqués dans les RNN (Réseaux de Neurone Résiduel). En utilisant cette technique, ils ont pu former un réseau de neurone avec 152 couches tout en étant moins complexe que d'autres réseaux comme VGGNet (voir section suivante). Il atteint un taux d'erreur parmi les cinq premiers de 3,57 %, surpassant les performances humaines sur cet ensemble de données (voir Figure II. 7).

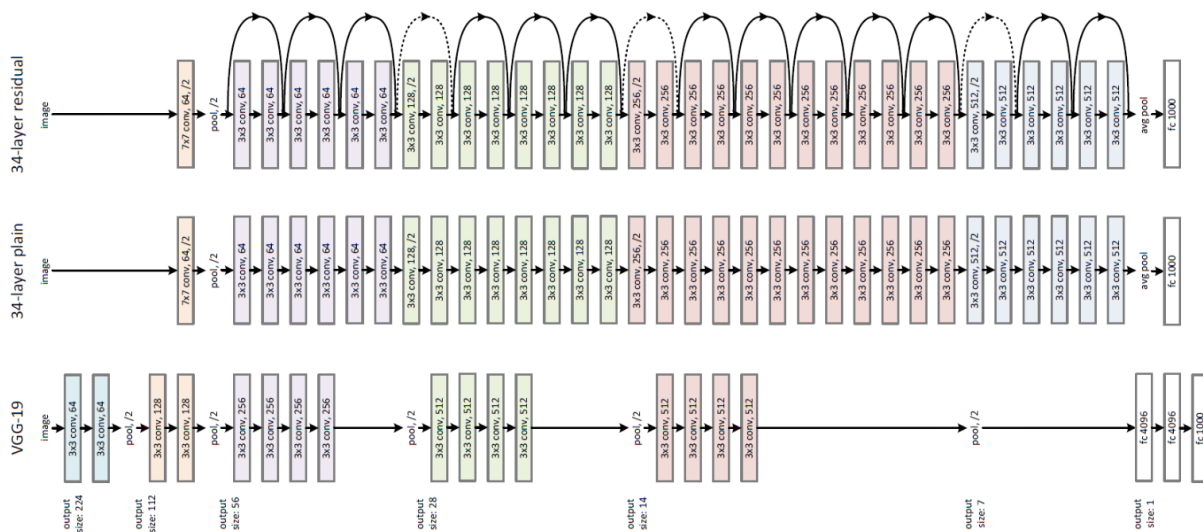


Figure II. 7 : Architecture ResNet-50

II.5.4 VGG16

Le VGG16 est un réseau de neurones [21] performant pour le défi de reconnaissance visuelle à grande échelle d'Image Net. Il a remporté la première place dans la tâche de localisation d'images et la deuxième place dans la tâche de classification d'images. VGG veut donner la réponse de "comment concevoir la structure du réseau" parmi les nombreux choix. VGG montre également que la profondeur du réseau joue un rôle important, et le réseau plus profond donne les meilleurs résultats. Un inconvénient de VGG est que ce réseau est généralement de grande taille, il contient environ 160 couches (voir la figure II. 8).

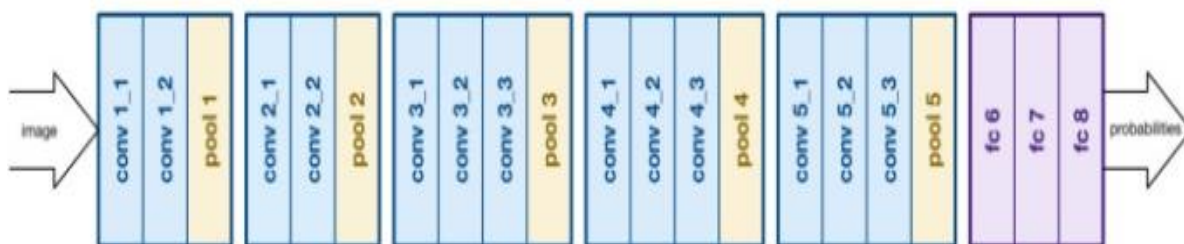


Figure II. 8 : La structure du réseau VGG16

II.6 Conclusion

Dans ce chapitre nous avons présenté la technique d'apprentissage par transfert très utilisé en apprentissage profond, son principe de base ainsi que ces caractéristiques et ces performances. En plus nous avons cité certaines architectures des réseaux pré-entraînés dont le TL est basé. Le chapitre suivant contiendra les résultats d'évaluation de la classification du système proposé à base d'apprentissage par transfert.

CHAPITRE III :
CLASSIFICATION D'IMAGE

III.1 Introduction

Après avoir détailler les différents outils utilisés pour la conception d'un système d'évaluation de la classification des bases de données choisies, ce chapitre met en lumière et présente les différents résultats obtenus. Mais avant de passer à cette phase, nous mettrons en évidence la classification des images en donnant les critères d'évaluation utilisés.

III.2 Classification d'image

La classification d'images est un processus de la vision par ordinateur, et notamment l'étape la plus importante de cette technologie. Elle consiste à apprendre un mappage des images à la classe Étiquettes ou labels. Les images d'entrée peuvent être représentées par des vecteurs de caractéristiques pré-extraits ou directement les pixels de l'image comme le cas de notre travail.

III.2.1 Motivations de la classification des images

La classification des images consiste à répartir les images aux différentes classes selon des critères préalablement. Classifier une image, lui fait correspondre une classe, et marquer sa relation avec d'autres images. Généralement, reconnaître une image est une tâche aisée pour un être humain, les connaissances acquises lui permettant de s'adapter aux variations résultant de conditions différentes d'acquisition. Il lui est par exemple relativement simple de reconnaître un objet dans plusieurs orientations partiellement cachées par une autre de près ou de loin et selon divers éclairages. Cependant, l'évolution technologique en matière d'acquisition d'images (caméras, capteurs, microscopes) et le stockage génèrent de riches bases de données d'informations et multiplient les domaines d'applications, il devient difficile pour l'humain d'analyser le grand nombre de images. Néanmoins, cette tâche n'est pas forcément facile pour un programme informatique pour lequel une image est un ensemble de valeurs numériques.

Le but de la classification d'images donc est de développer un système capable d'attribuer automatiquement une classe à une image. Les applications de ce type de système sont nombreuses est vont de l'analyse des documents, de la médecine au domaine militaire.

III.2.2 Apprentissage par transfert et classification

L'apprentissage par transfert est un outil précieux de l'apprentissage profond qui permet d'accéder aux connaissances acquises à partir de modèles déjà entraînés sur des millions

d'images étiquetées. L'exécution de ces modèles peut prendre des semaines, même lorsque plusieurs GPU sont utilisés, donc ceci est impossible sur un PC ordinaire.

Dans ce travail, l'utilisation de l'apprentissage par transfert pour la classification des images est réalisée par deux approches comme ça été déjà cité au chapitre II : par extraction de caractéristiques à partir du code CNN suivi par un classifieur SVM et par Fine Tuning.

III.2.2.1 Fine Tuning

Le Fine Tuning consiste à apporter de petits ajustements à un processus pour obtenir le résultat ou les performances souhaités. Le réglage du modèle d'apprentissage automatique, c'est comme faire tourner les commutateurs et les boutons du téléviseur jusqu'à ce que vous obteniez un signal plus clair.

Le Fine Tuning peut être vu comme un moyen d'optimisation [22]. Nous optimisons le réseau pour obtenir des résultats optimaux. Nous pouvons changer le nombre de couches utilisées, le nombre de filtres, le taux d'apprentissage et nous avons de nombreux paramètres du modèle à optimiser.

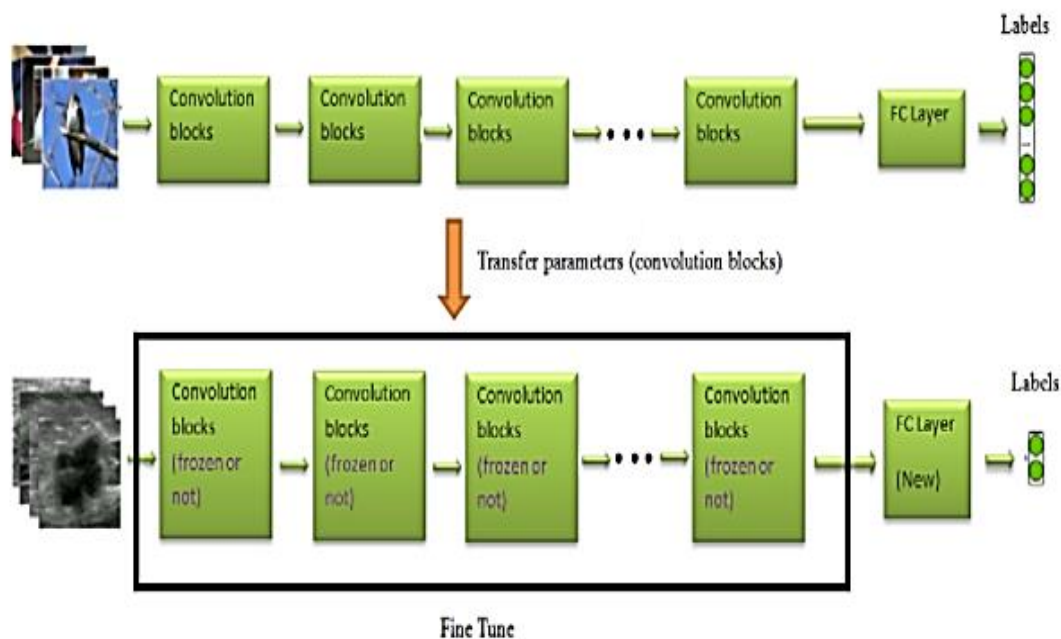


Figure III. 1 : Architecture de fine tuning

Le Fine tuning de l'apprentissage en profondeur utilise les poids de l'algorithme d'apprentissage en profondeur précédent pour programmer un autre processus d'apprentissage en profondeur similaire. Les poids sont utilisés pour connecter chaque neurone d'une couche à chaque neurone de la couche suivante du réseau de neurones.

III.2.2.2 Extraction de caractéristique et SVM

Le réseau pré-entraîné est utilisé ici comme moyen d'extraction de caractéristique. Une fois le code CNN ou vecteur de caractéristique est obtenu on utilise le classifieur SVM [23]. L'objectif de l'algorithme SVM est de trouver un hyperplan dans un espace à N dimensions (N le nombre d'entités) qui classe distinctement les points de données. Il existe de nombreux hyperplans possibles qui pourraient être choisis. Notre objectif est de trouver un écart à marge maximale (vérifier la figure), c'est-à-dire la distance maximale entre les points de données des deux classes. Maximiser la distance de la marge fournit un certain renforcement afin que les futurs points de données puissent être classés avec plus de confiance.

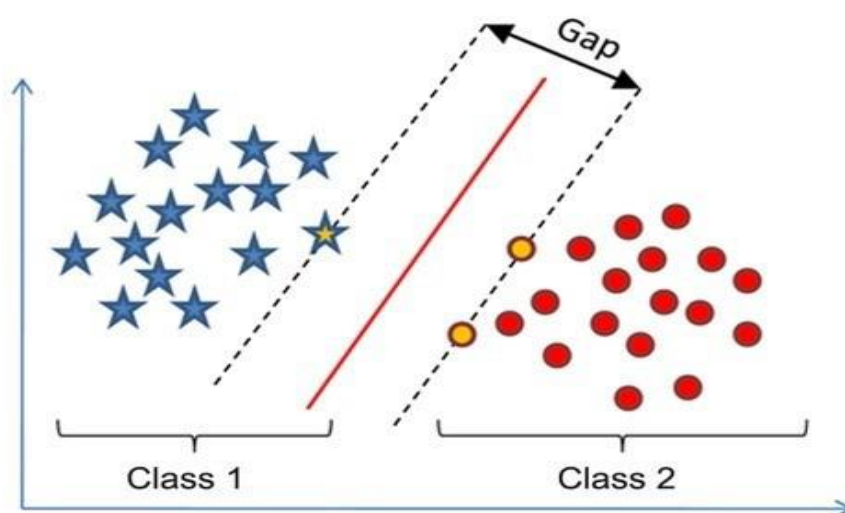


Figure III. 2 : algorithme de SVM

III.3 Les paramètres d'évaluation d'une classification

Pour l'évaluation d'un modèle de classification à l'aide de CNN, on définit les paramètres suivants :

- Vrai positif 'True positive TP : prédiction positive correcte.
- True negative TN (Vrai négatif) : prédiction négative correcte.
- False positive FP (Faux positif) : prédiction positive incorrecte.
- False negative (Faux négatif) FN : prédiction négative incorrecte.

III.3.1 Précision

La mesure la plus couramment utilisée pour évaluer les performances d'un modèle prédictif de classification est la précision. En règle générale, la précision d'un modèle prédictif

est bonne (précision supérieure à 90 %). La précision de classification peut être obtenue à partir de cette matrice comme suit :

$$Precision = \frac{VP + VN}{VP + VN + FN + FP}$$

III.3.2 Rappel

Le rappel permet d'obtenir la proportion de résultats positifs réels qui a été identifiée correctement.

$$Rappel = \frac{VP}{VP + FN}$$

III.3.3 La courbe ROC

Le graphe de La courbe ROC (Receiver Operating Characteristic) indique les performances du modèle de classification. On dessine cette courbe à l'aide de deux paramètres ; taux de vrais positifs (TVP) en fonction du taux de faux positifs (TFP).

- Le taux de vrais positifs (TVP) :

$$TVP = \frac{VP}{VP + FN}$$

- Le taux de faux positifs (TFP) :

$$TFP = \frac{FP}{FP + VN}$$

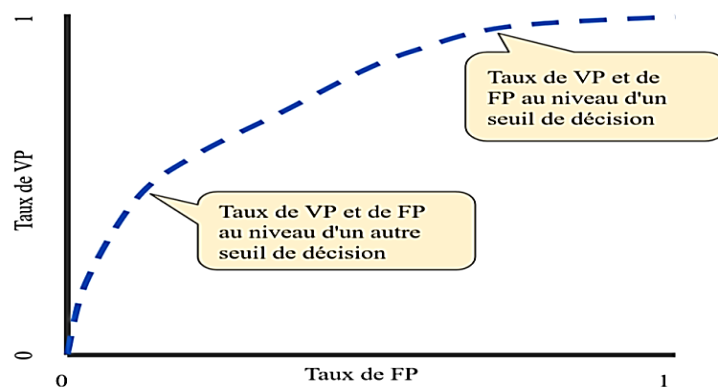


Figure III. 3 : courbe ROC

III.3.4 Matrice de confusion

La matrice de confusion montre des résultats de prédiction pour un problème particulier de classification. Elle compare les données réelles pour une variable cible à celles prédites par un modèle. Les prédictions justes et fausses sont révélées et réparties par classe, ce qui permet

de les comparer avec des valeurs définies. Également connue sous le nom de tableau de contingence, la matrice de confusion permet d'évaluer les performances d'un modèle de classification. Elle montre donc à quel point un certain modèle peut être confus lorsqu'il fait des prédictions. Sur la figure III. 4, on montre un modèle d'apprentissage pour classification en 3 classes.

Confusion matrix		Reality		
		Class A	Class B	Class C
Prediction	Class A	True Positive A : TPA	False Negative B : FNB False Positive A : FPA	False Negative C : FNC False Positive A : FPA
	Class B	False Positive B : FPB False Negative A : FNA	True Positive B : TPB	False Negative C : FNC False Positive B : FPB
	Class C	False Positive C : FPC False Negative A : FPA	False Positive C : FPC False Negative B : FNB	True Positive C : TPC

Figure III. 4 : Matrice de confusion pour un modèle 3 classes

III.4 Les résultats de la simulation

Dans ce projet, nous avons effectué nos simulations avec le logiciel Matlab version (R2002a) sur un ordinateur portable avec les caractéristiques suivantes :

- Processeur Intel i7-1165G7
- Vitesse d'horloge : 3.4 GHz
- Capacité Mémoire (RAM) : 8Go
- Carte Graphique : Intel iris XE Graphics
- Capacité disque dur : 512 Go SSD
- Système d'exploitation : Windows 11 Famille 64-bits

III.4.1 Bases de données utilisées

Dans notre travail nous avons utilisé trois bases de données téléchargées à partir du site Kaggle [24]. Les images contenues dans les bases de données utilisés sont de différentes tailles. Pour pouvoir les utilisées comme données d'entrées par les réseaux pré entraînés du système proposé, il nous a fallu les redimensionner. La dimension adaptée dans les deux approches est 227*227*3. Une description détaillée de ces bases est donnée comme suit :

III.4.1.1 Base 1 (GC10-DET)

Ensembles de données sur les défauts de surface métalliques. Ces défauts collectés se trouvent à la surface de la tôle d'acier. La base comprend 3570 images à niveaux de gris regroupées en dix types de défauts de surface. Ces derniers sont : Poinçonnage (Pu), ligne de soudure (Wl), espace en croissant (Cg), tache d'eau (Ws), tache d'huile (Os), tache de soie (Ss), inclusion (In), fosse roulée (Rp), pli (Cr), la taille (Wf).

- Poinçonnage : Dans la ligne de production de la bande, la bande d'acier doit être poinçonnée selon la spécification du produit ; une défaillance mécanique peut entraîner un poinçonnement indésirable, entraînant un poinçonnement défauts.
- Ligne de soudure : Lors du changement de feuillard, il faut souder les deux spires du feuillard, et la ligne de soudure est produite. À proprement parler, ce n'est pas un défaut, mais il doit être automatiquement détecté et suivi pour être contourné lors des coupes ultérieures.
- Écart en croissant : dans la production de bandes d'acier, la coupe entraîne parfois des défauts, tout comme la moitié un cercle.
- Point d'eau : Un point d'eau est produit par séchage en production. Sous différents produits et processus, les exigences pour ce défaut sont différentes. Cependant, parce que les points d'eau sont généralement avec un faible contraste, et sont similaires à d'autres défauts tels que les taches d'huile, ils sont généralement détectés par erreur.
- Tache d'huile : Une tache d'huile est généralement causée par la contamination du lubrifiant mécanique, qui affecte l'apparence du produit.
- Tache de soie : une plaque locale ou continue en forme de vague sur une surface de bande qui peut apparaître sur la partie supérieure et les surfaces inférieures, et la densité est inégale dans toute la direction de la longueur de la bande. En général, la raison principale réside dans la température inégale du rouleau et la pression inégale.
- Inclusion : L'inclusion est un défaut typique des défauts de surface métalliques, montrant généralement de petites taches, forme d'écailles de poisson, forme de bande, bloc distribution irrégulière dans la bande de la surface supérieure et inférieure (global ou local), et s'accompagne souvent de surfaces rugueuses grêlées. Certaines inclusions sont lâche et facile à tomber et certains sont pressés dans la plaque.
- Fosse laminée : les fosses laminées sont des renflements périodiques ou des piqûres à la surface d'une plaque d'acier qui sont ponctuées, floconneux ou en forme de bande. Ils sont

répartis sur toute la longueur ou la section de la bande, principalement causés par dommages au rouleau de travail ou au rouleau de tension.

III.4.1.2 Base 2 (NEU-DET)

La base 2 Contient six types de défauts de surface typiques de la bande d'acier laminée à chaud. On trouve : l'échelle laminée (RS), les patches (Pa), les craquelures (Cr), la surface piquée (PS), inclusion (In) et rayures (Sc). La base de données comprend 1 800 images en niveaux de gris : 300 échantillons de six types différents de défauts de surface typiques.

III.4.1.3 Base 3 (GC10-DET)

La troisième base de données utilisée concerne un produit de fabrication de moulage. Le moulage est un processus de fabrication dans lequel un matériau liquide est généralement versé dans un moule, qui contient une cavité creuse de la forme souhaitée, puis on le laisse se solidifier. La raison de la collecte de ces données est l'inspection des défauts de la coulée. Ce dernier est une irrégularité indésirable dans un processus de coulée de métal. Il existe de nombreux types de défauts dans la coulée, comme les trous de soufflage, les trous d'épingle, les bavures, les défauts de retrait, les défauts de matériau de moule, les défauts de métal coulé, les défauts métallurgiques, etc. Les défauts sont des choses indésirables dans l'industrie du moulage. L'ensemble de données contient au total 7348 données d'image. Celles-ci ont toutes les tailles d'images en niveaux de gris (300*300) pixels. Également la taille des images utilisées est de 512x512 en niveaux de gris. Cet ensemble de données est sans augmentation.

Pour capturer ces images, il faut un éclairage stable, pour cela nous avons fait un arrangement spécial. il y a principalement deux types : défectueux et d'accord. En créant un modèle de classification, nous avons déjà divisé les données pour la formation et les tests en deux dossiers. Les dossiers train et test contiennent les sous-dossiers deffront et okfront.

- Train : défaut de face ou **def front** 3758 image et sans défaut ou **ok front** 2875 images.
- Test : défaut de face ou **def front** 453 et sans défaut ou **ok front** ont 262 images.

III. 5. Résultats de simulation

Différentes simulations ont été réalisées afin de tester les performances des approches proposées sur les trois bases de données indiquées ci-dessus. Pour ce faire nous avons utilisé les réseaux pré entraînés déjà cité aux chapitre II, à savoir Alexnet, Googlenet, VGG19 et Resnet50. Les différents résultats obtenus sont présentés en détail et par base dans les sections suivantes.

III. 5. 1. Base 1 (GC10-DET)

Les tests de précisions effectués sur Base1 pour les deux approches (extraction de caractéristique plus SVM et Fine Tuning) avec différents réseaux sont présentés dans tableau III. 1 et III. 2 respectivement

Réseau Convolutionnel	Précision	Durée de traitement
ResNet 50	81.77%	9min
GoogleNet	80.47%	7min
Vgg19	81.25%	12min 30s
AlexNet	79.43%	5min

Tableau III. 1 : résultats de précision de la base 1 par SVM

Réseau Convolutionnel	Précision	Durée d'exécution
ResNet 50	97.92%	253min 47sec
GoogleNet	95.83%	100min 35sec
Vgg19	99.54%	1057min 20sec
AlexNet	98.38%	54min 52sec

Tableau III. 2 : résultats de précision de la base 1 par fine tuning

D'après les valeurs indiquées sur les deux tableaux, on peut constater que la méthode du Fine Tuning est plus précise que celle utilisant les SVM. Par contre pour la durée d'exécution c'est l'inverse. Cette remarque peut être généralisée pour les différentes bases de données.

Le meilleur résultat en terme de précision est obtenu pour le réseau VGG19, par contre en temps de calcul c'est le plus mauvais. En considérant les deux paramètres en même temps on peut constater que AlexNet est le plus perforant.

Les courbes de précisions et de perte, les matrices de confusion ainsi que des échantillons d'images de la Base1 obtenus avec les différents réseaux, sont illustrés dans les figures numérotées de III. 5 à III. 13.

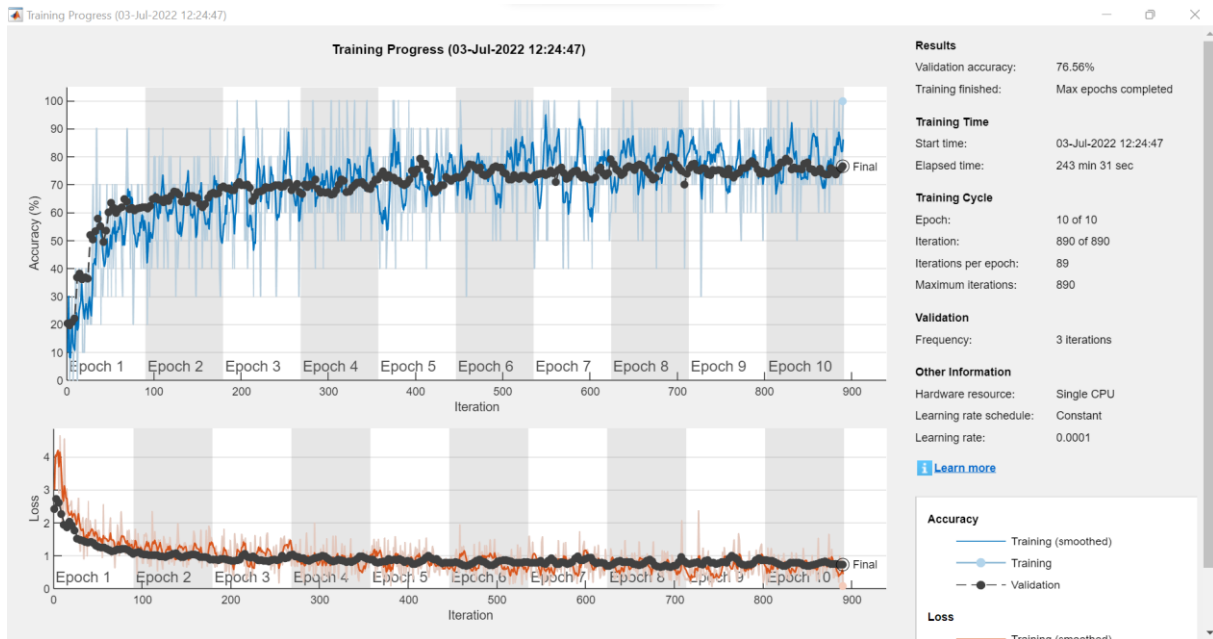


Figure III. 5 : courbe de précision et sa perte de la base 1 à l'aide d'AlexNet

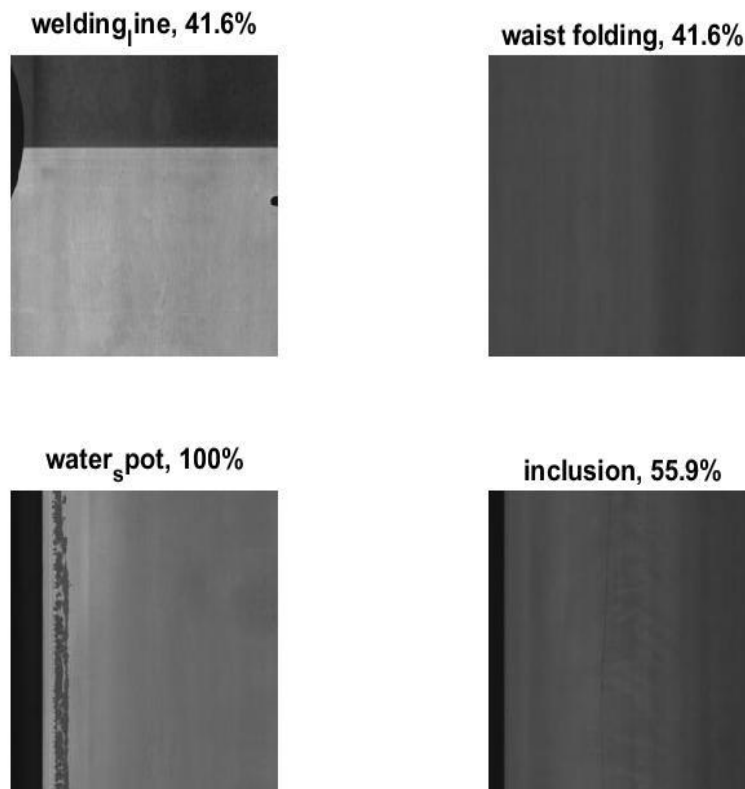


Figure III. 6 : Taux d'erreur et de précision par Alexnet

1	9			1	1	1	1	2		1
2	2	34				1		1	1	6
3			18	21				1	5	
4			4	40					1	
5		1		4	38	1		1		
6	1			1		6			1	
7							42		3	
8	1			1				42		
9	1			1					43	
10	4	4			11	2		2		22
	1	2	3	4	5	6	7	8	9	10

Figure III. 7 : matrice de confusion d'après Alexnet

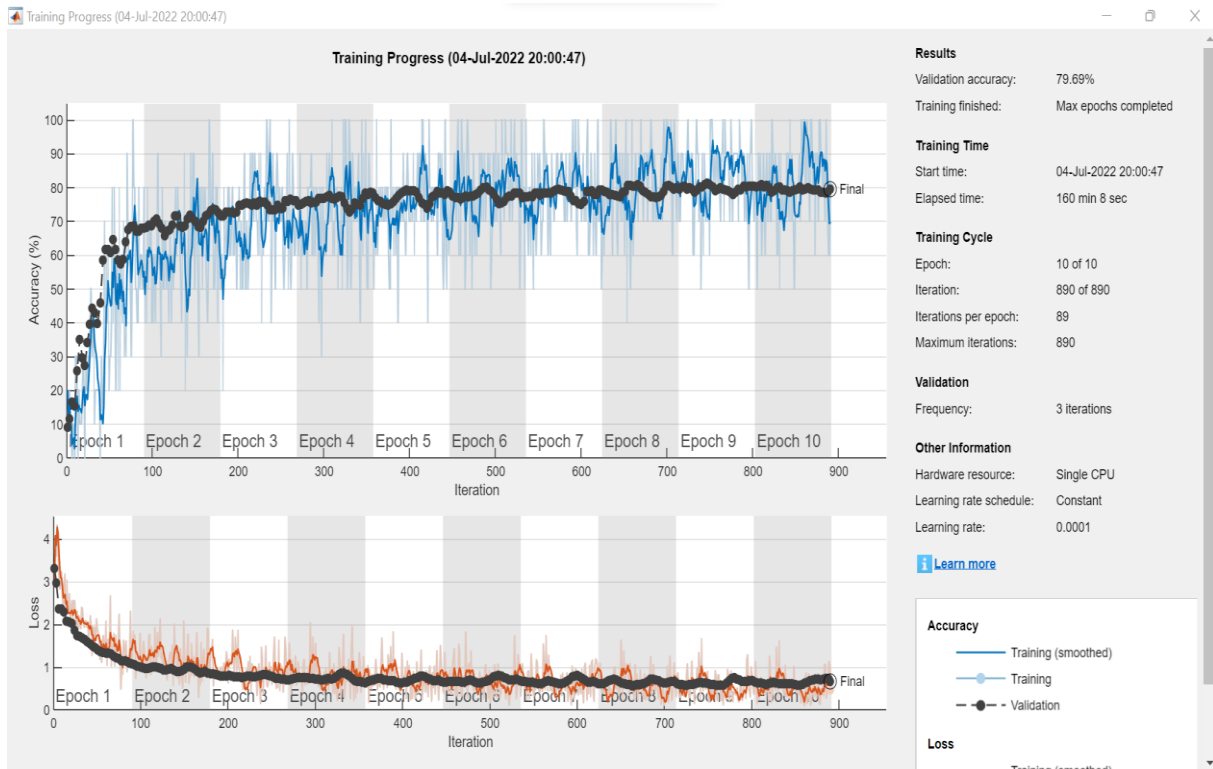


Figure III. 8 : courbe de précision et sa perte de la base 1 à l'aide de GoogleNet

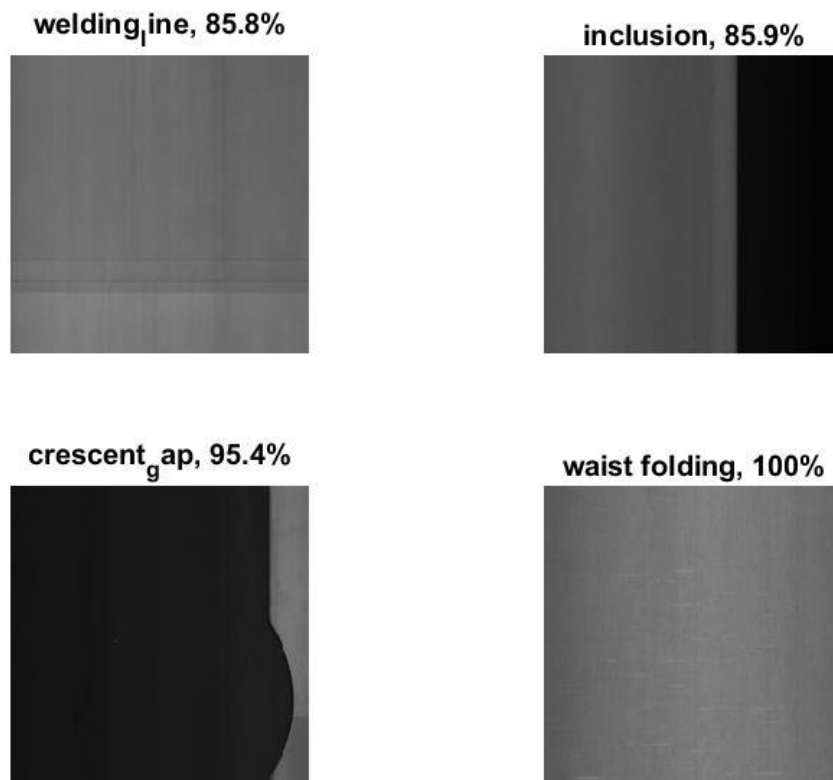


Figure III. 9 : Taux d'erreur et de précision par GoogleNet

1	8		1		1		1	5		
2		39		1	1					4
3	1		34					5	5	
4			7	32		2		1	3	
5		1		1	41					2
6			1	1		2		4		1
7							45			
8					1			43		
9								1	44	
10	4	3			17			2	1	18
	1	2	3	4	5	6	7	8	9	10

Predicted Class

Figure III. 10 : matrice de confusion d'après (GoogleNet)

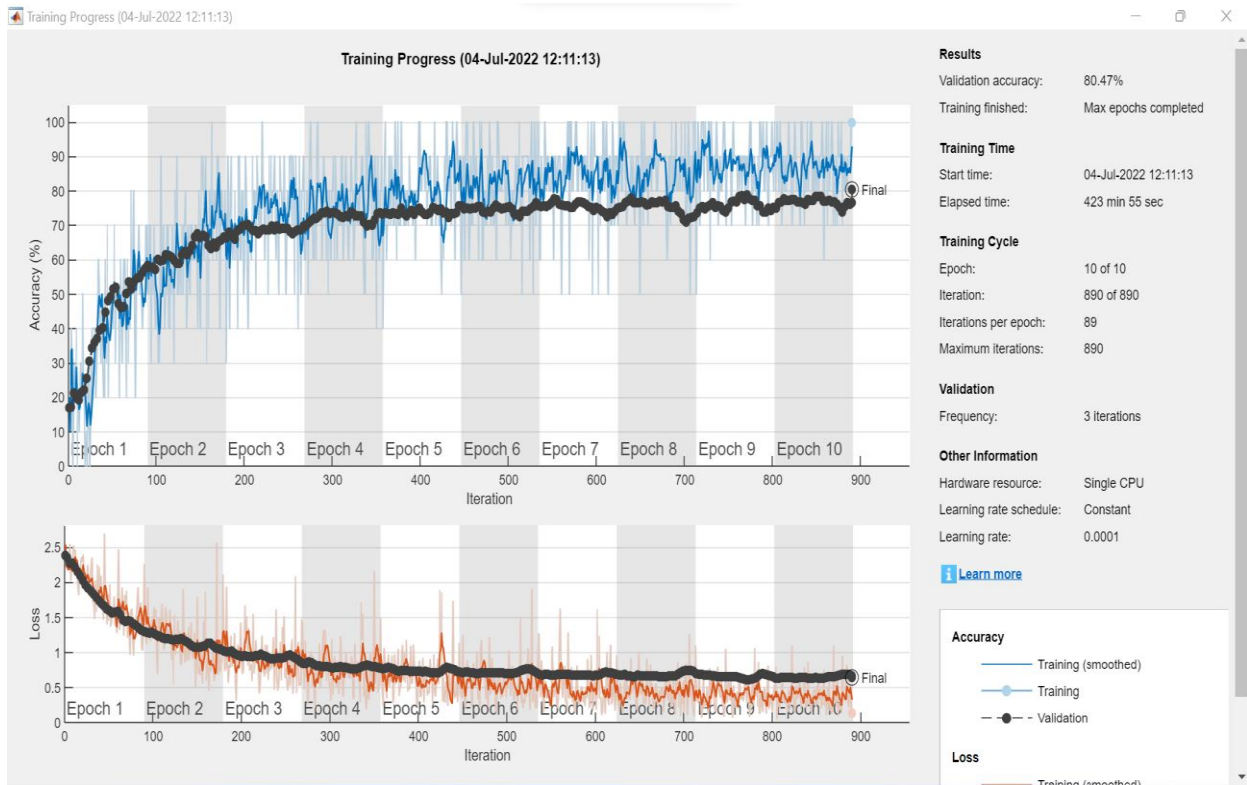


Figure III. 11 : courbe de précision et sa perte de la base 1 à l'aide ResNet50

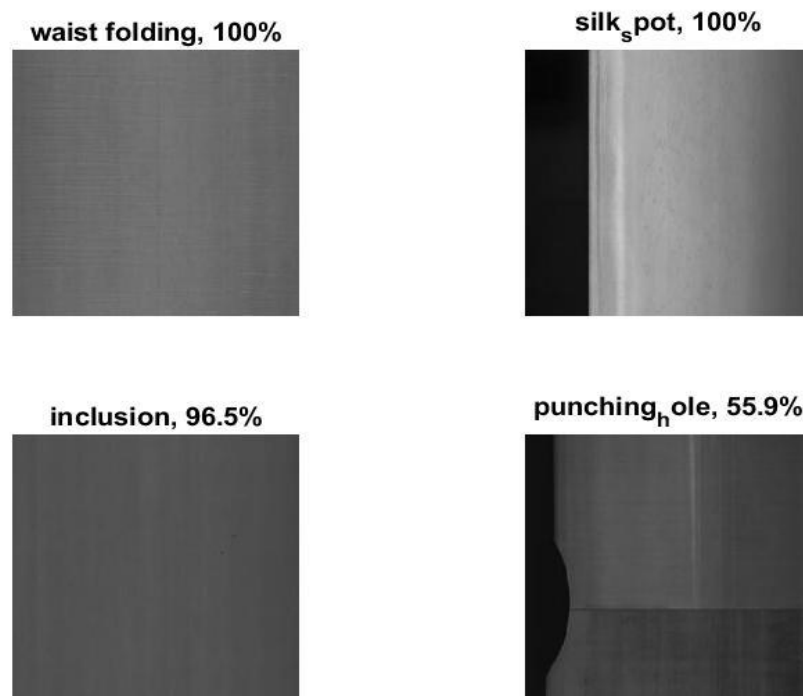


Figure III. 12 : Taux d'erreur et de précision à l'aide de ResNet50

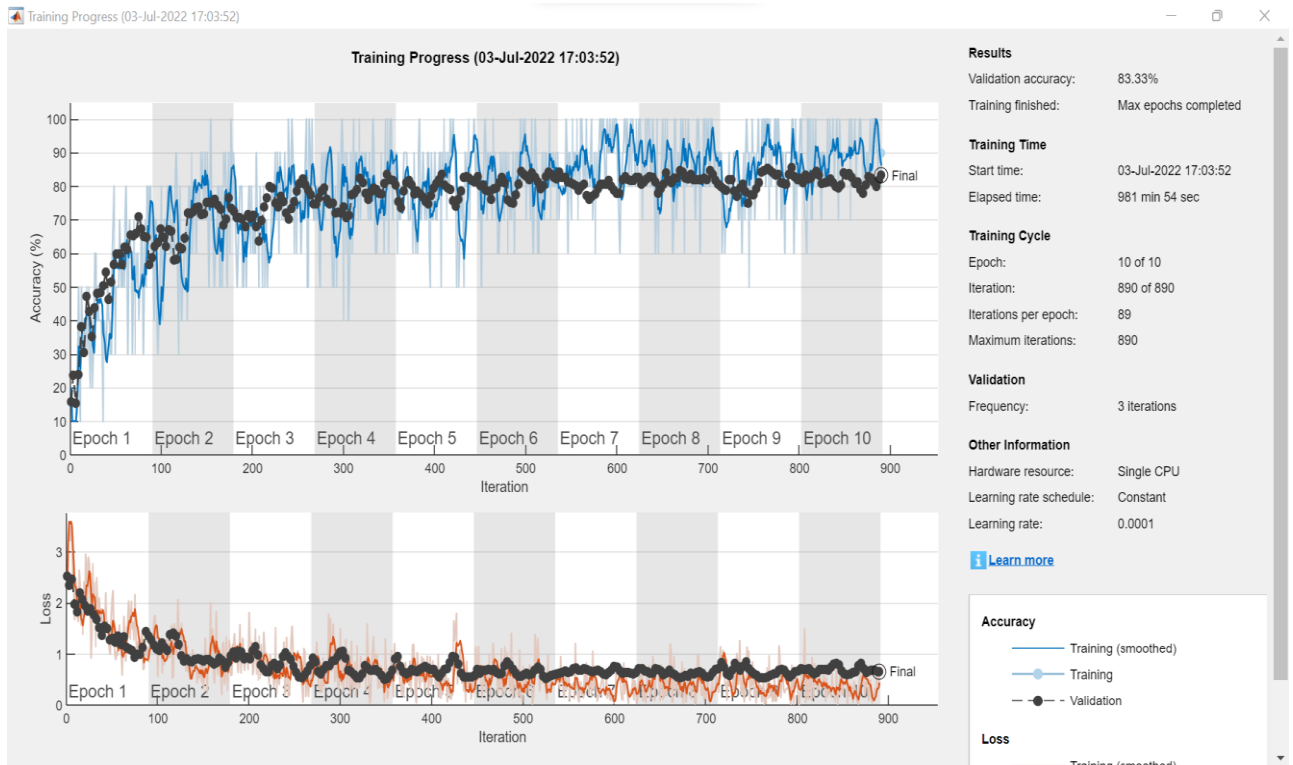


Figure III. 13 : courbe de précision et sa perte de la base 1 à l'aide Vgg19

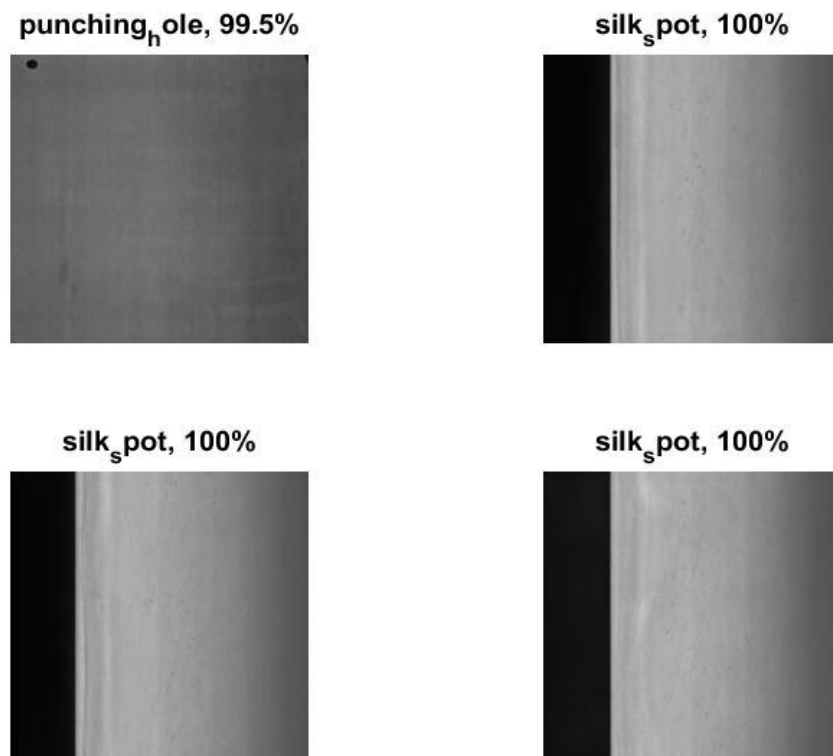


Figure III. 14 : Taux d'erreur et de précision par Vgg19

1	11							4	1	
2		40								5
3		1	41		2	1				
4	2		9	29		3		1	1	
5					42					3
6			1		2	6				
7							44		1	
8		1			1			42		
9			2				1		42	
10	2	6			14					23
	1	2	3	4	5	6	7	8	9	10

Figure III. 15 : matrice de confusion d'après Vgg19

Discutions :

- D'après les Figures III.5, III. 8 , III.11 et III. 13 des réseaux : Alexnet, GoogleNet, ResNet50 et VGG19 respectivement, la précision de l'apprentissage des tests augmente avec les époques (Epoch). Cela indique que le modèle apprend plus d'informations à chaque époque. Si la précision est réduite alors nous aurons besoin de plus d'informations pour que notre modèle apprenne, nous devons donc augmenter le nombre d'époques.
- La matrice de confusion montre des résultats détaillés sur les taux de classifications des différents échantillons de la base de données. Elle illustre les taux réels de classification des échantillons par rapport à leurs prédictions.

III. 5. 2 Base 2 (NEU-DET)

Les mêmes tests effectués sur Base 1 sont appliqués sur base 2. Les résultats obtenus sont présentés sur les tableaux III. 3 et III. 4.

Réseau Convolutionnel	Précision	Durée de traitement
ResNet 50	100%	7min 50sec
GoogleNet	99.31%	4min 50sec
Vgg19	99.54%	11min
AlexNet	99.07%	3min 20sec

Tableau III. 3 : résultats de précision par extraction et SVM

Réseau Convolutionnel	Précision	Durée de traitement
ResNet 50	97.92%	253min47sec
GoogleNet	95.83%	100min35sec
Vgg19	99.54%	1057min 20sec
AlexNet	98.38%	54min52sec

Tableau III. 4 : résultats de précision de la base 2 par fine tuning

Pour cette base de données, les meilleurs résultats -en terme de précisions ou en terme de temps d'exécution- sont obtenus avec la première approche ; celle utilisant l'extraction de caractéristiques plus les SVM.

Le réseau AlexNet présente le meilleur compromis précision temps de calcul.

Les courbes de précisions et de perte, les matrices de confusion ainsi que des échantillons d'images de la Base1 obtenus avec les différents réseaux, sont illustrés dans les figures numérotées de III. 16 jusqu'à III. 24.

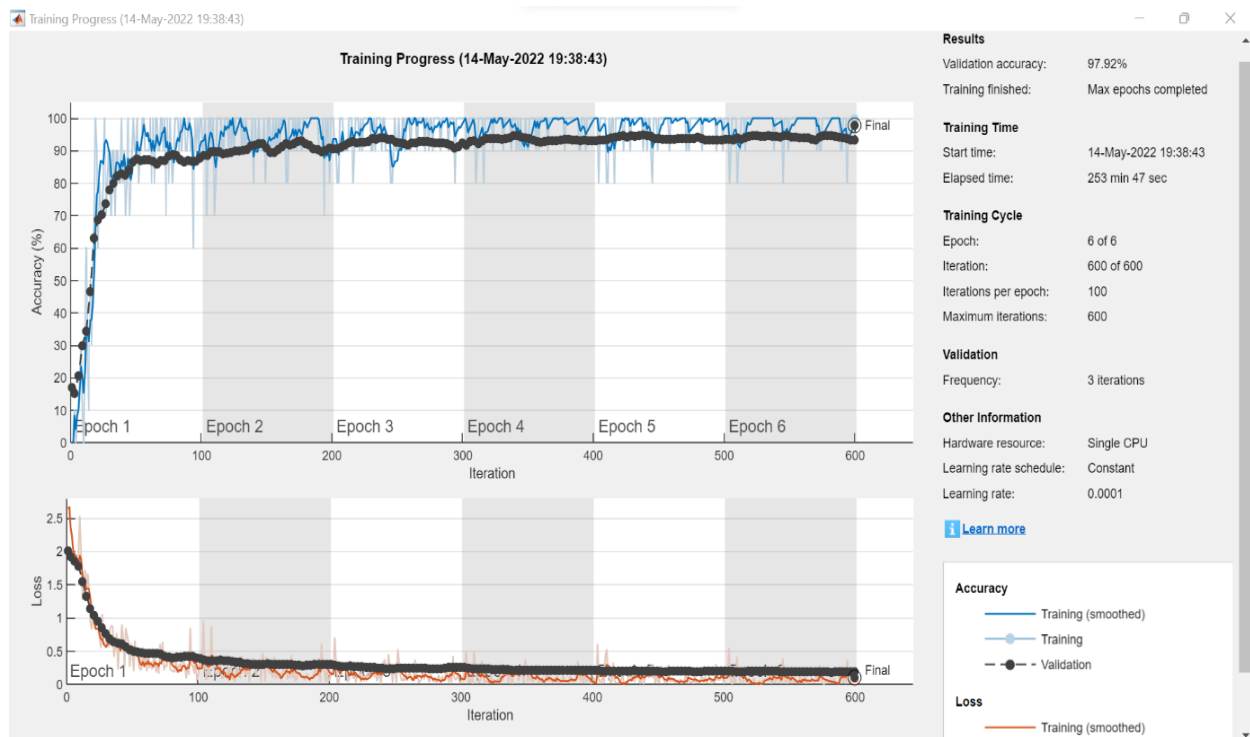


Figure III. 16 : courbe de précision et sa perte de la base 2 à l'aide de ResNet50

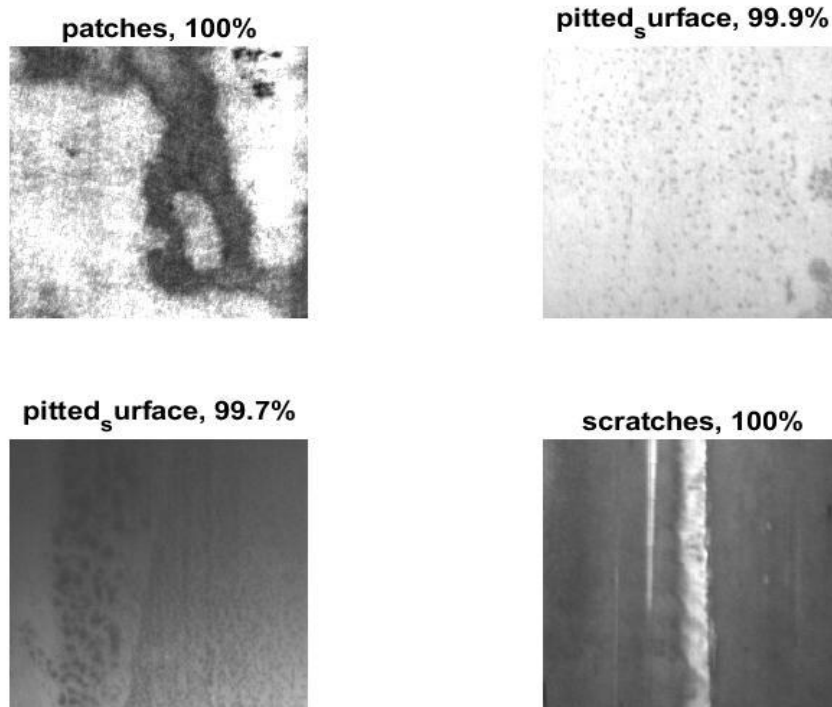


Figure III. 17 : Taux d'erreur et de précision par ResNet50

1	72					
2		65				7
3			72			
4				72		
5					72	
6		1				71
	1	2	3	4	5	6
	Predicted Class					

Figure III. 18 : matrice de confusion d'après ResNet50

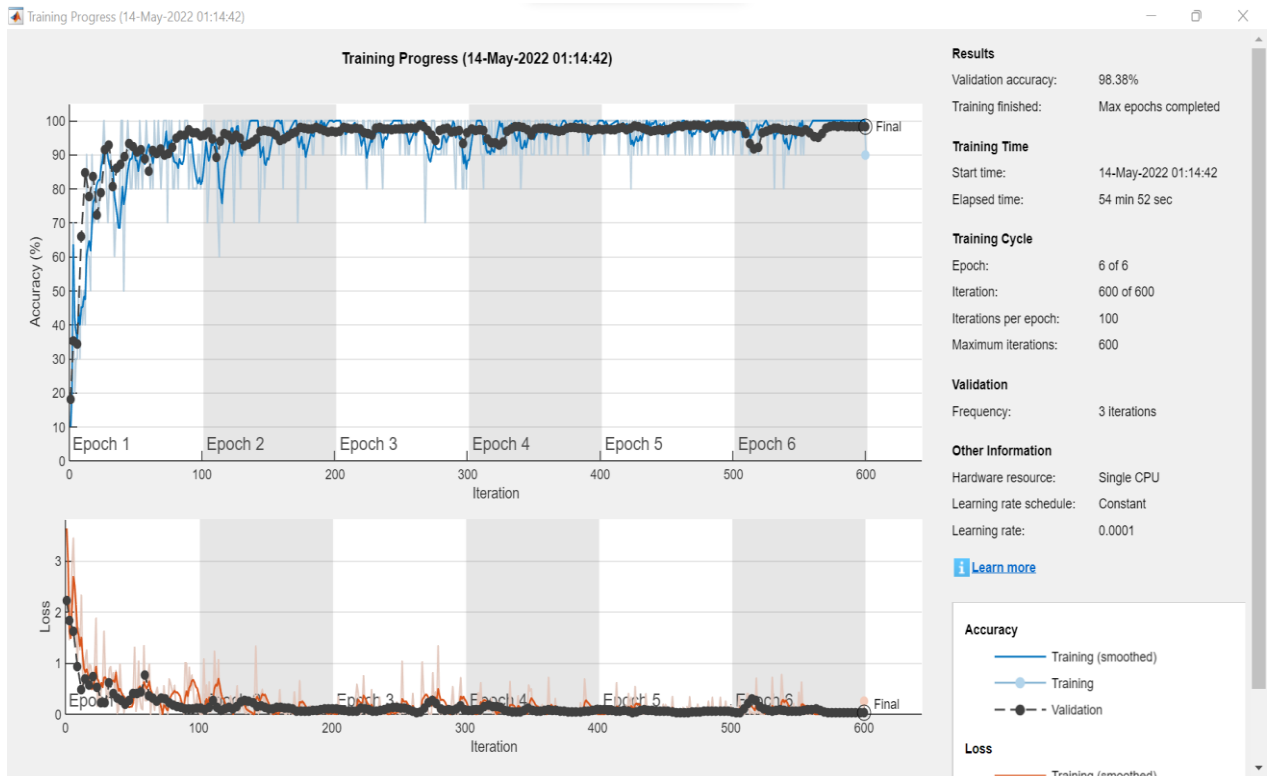


Figure III. 19 : courbe de précision et sa perte de la base 2 à l'aide d'AlexNet

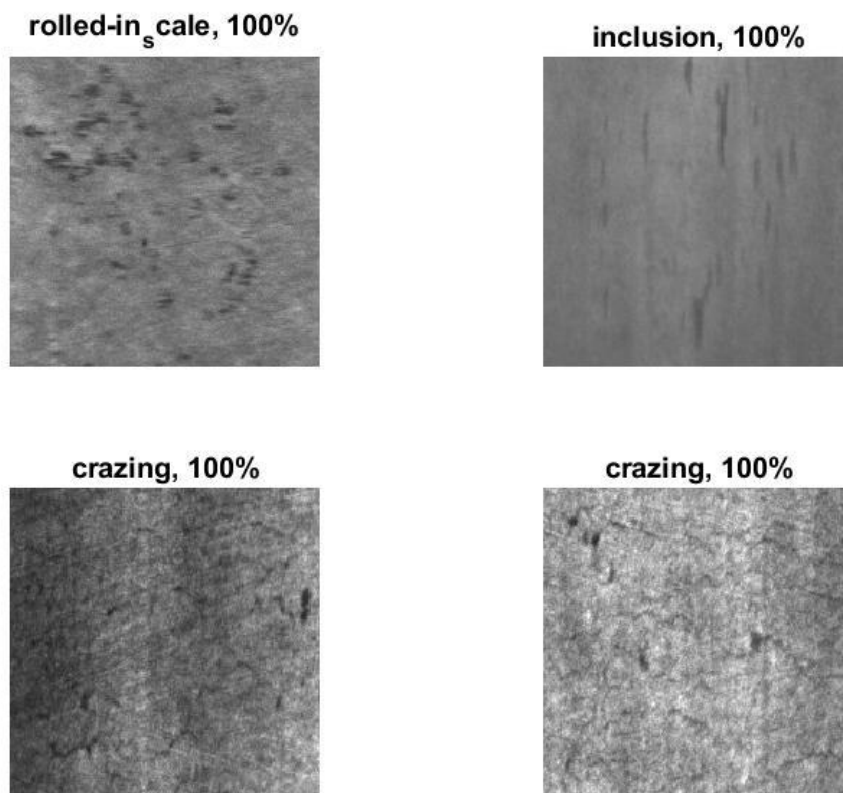


Figure III. 20 : Taux d'erreur et de précision par AlexNet

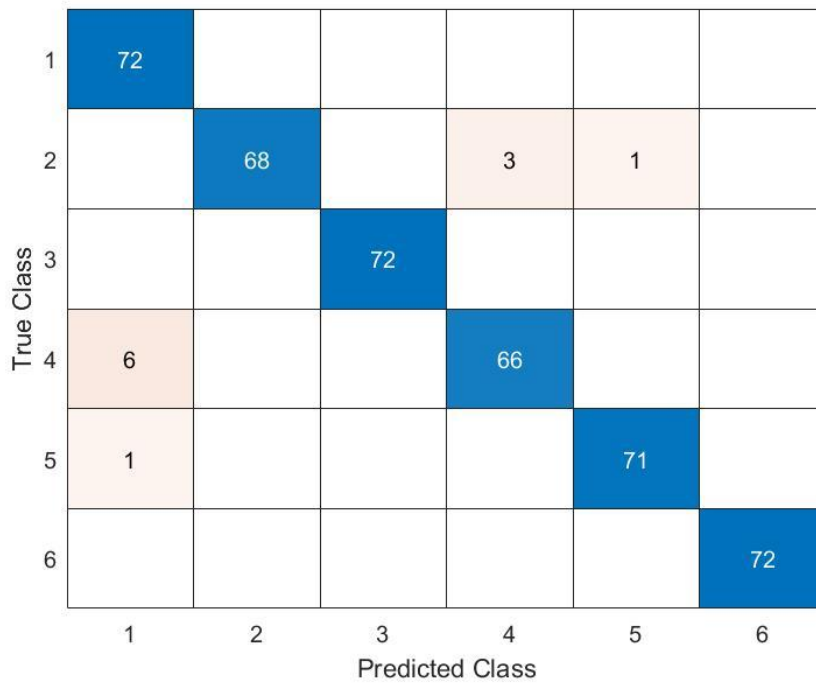


Figure III. 21 : matrice de confusion d'après AlexNet

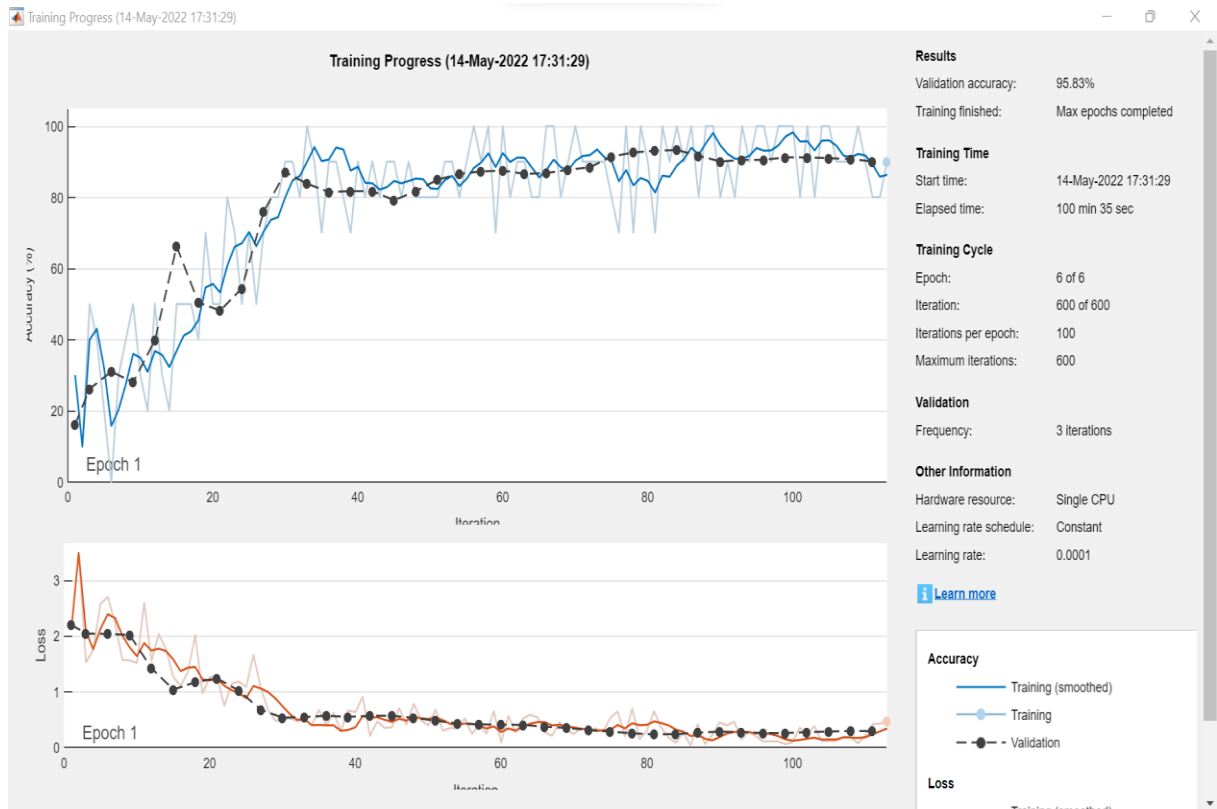


Figure III. 22 : courbe de précision et sa perte de la base 2 à l'aide GoogleNet

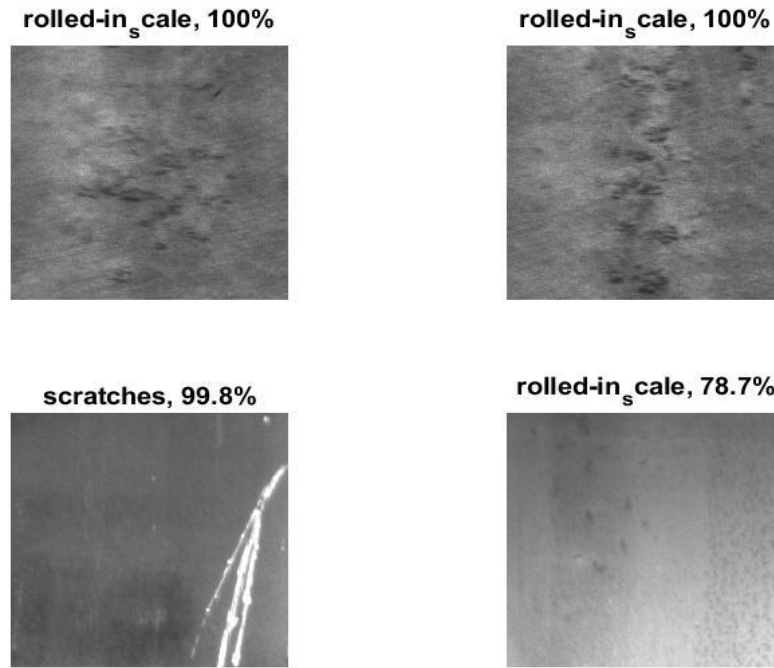


Figure III. 23 : Taux d'erreur et de précision par GoogleNet.

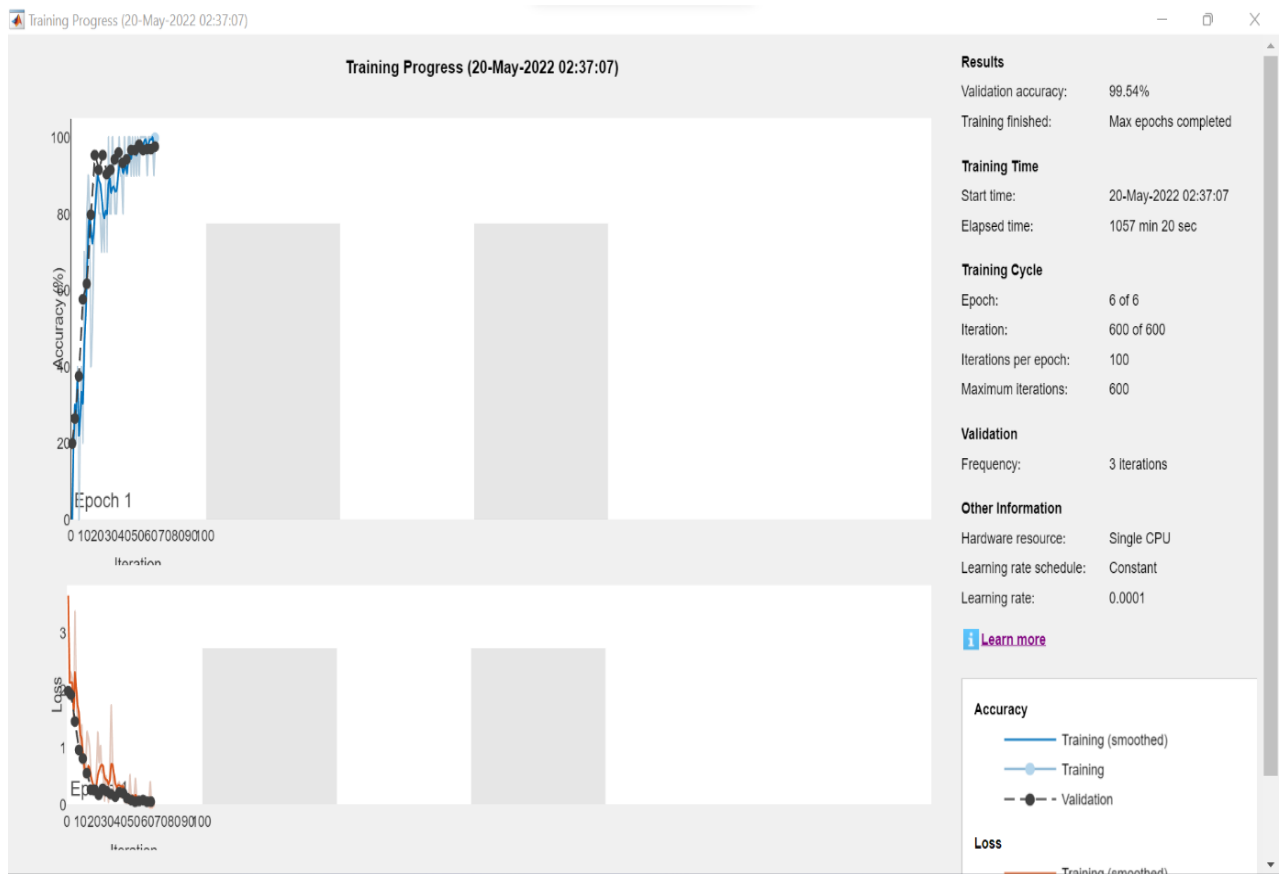


Figure III. 24 : courbe de précision et sa perte de la base 2 à l'aide de Vgg19

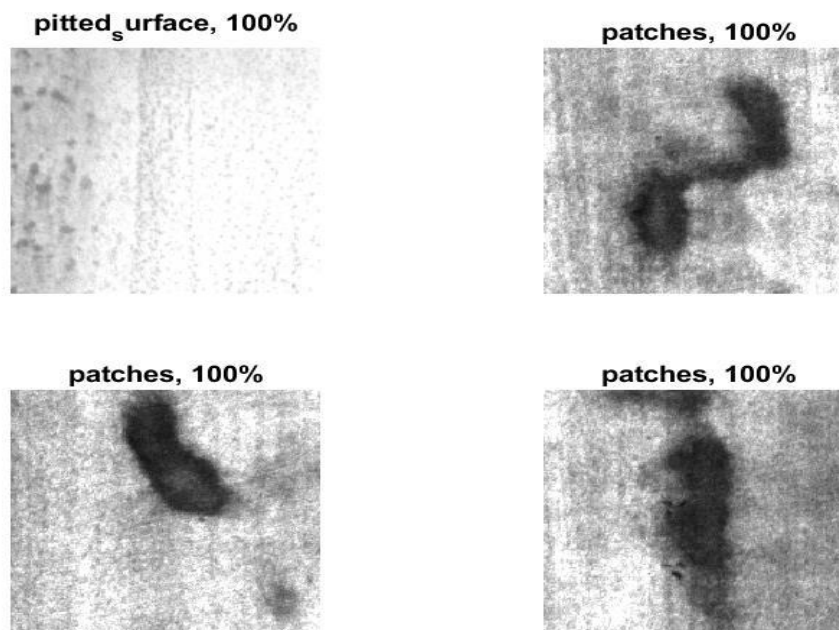


Figure III. 25 : Taux d'erreur et de précision par Vgg19

Les matrices de confusions retrouvées de base 2 sont quasi-diagonales ; la majorité des échantillons ont été correctement classés.

III. 5. 3. Base 3 (Casting data)

Les mêmes tests effectués sur Base 1 et 2 sont appliqués sur base 3. Les résultats obtenus sont présentés sur les tableaux III. 5 et III. 6.

ConvNet	Accuracy	Durée de traitement
ResNet 50	99.50%	19min17sec
GoogleNet	99.20%	15min 33sec
Vgg19	99.45%	23min 45sec
AlexNet	99.50%	9min 39sec

Tableau III. 5 : résultats de précision de la base 3 par SVM

ConvNet	Accuracy	Temps d'exécution
ResNet 50	99.45%	5993min
GoogleNet	97.29%	2063min
Vgg19	99.70%	12352min 59sec
AlexNet	99.60%	919min 35sec

Tableau III.6 : résultats de précision de la base 3 par fine tuning

Les résultats de simulation de la base 3 de données sont presque similaires pour les deux approches et pour les réseaux utilisés.

Le réseau Alexnet présente toujours le meilleur compromis précision temps de calcul.

Les figures III.26 jusqu'aux III. 33 illustres les courbes de précision, ainsi que des échantillons d'images de la base 3 de données

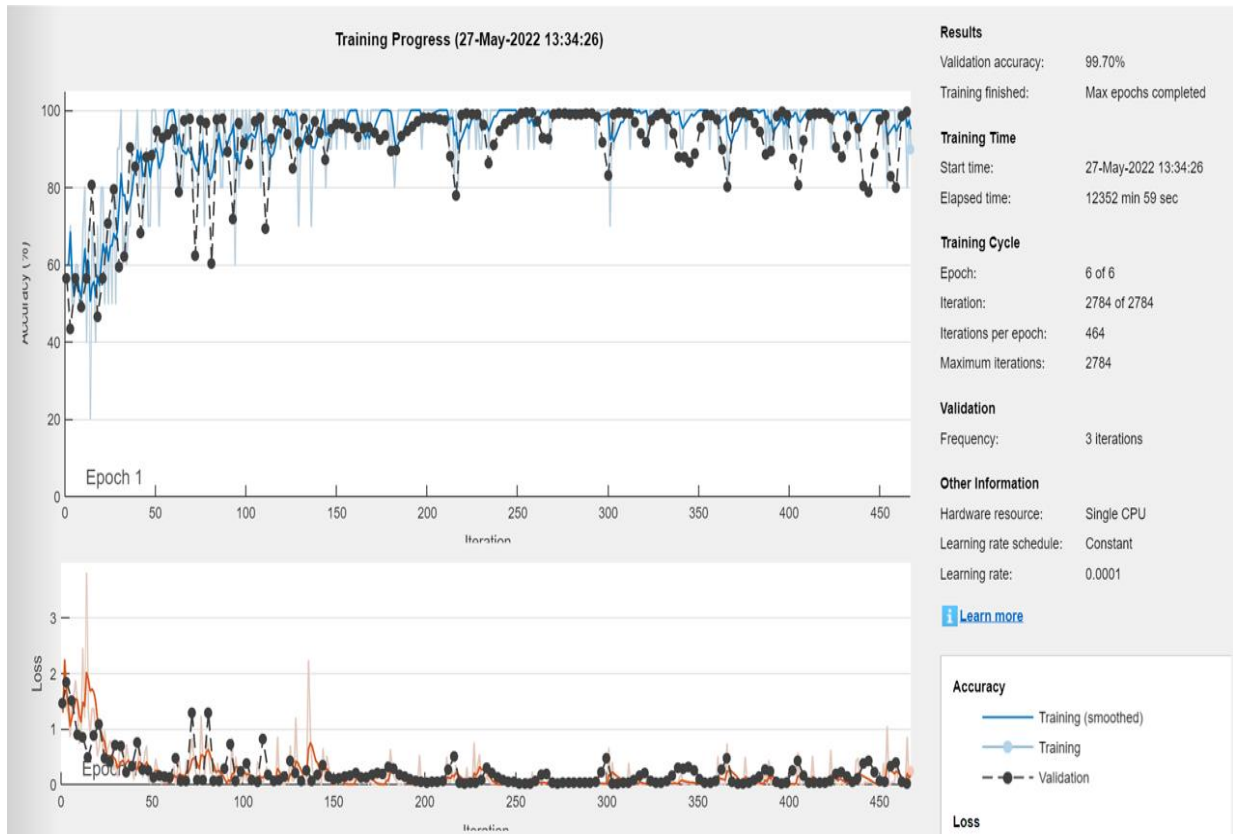


Figure III. 26 : courbe de précision et sa perte de la base 3 à l'aide de vgg19

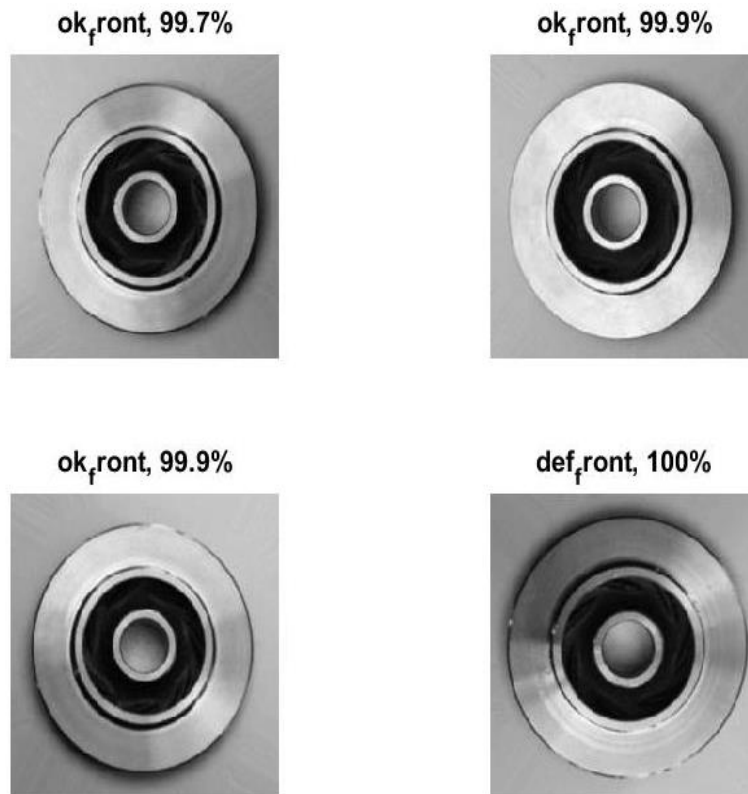


Figure III.27: Taux d'erreur et de précision par vgg19

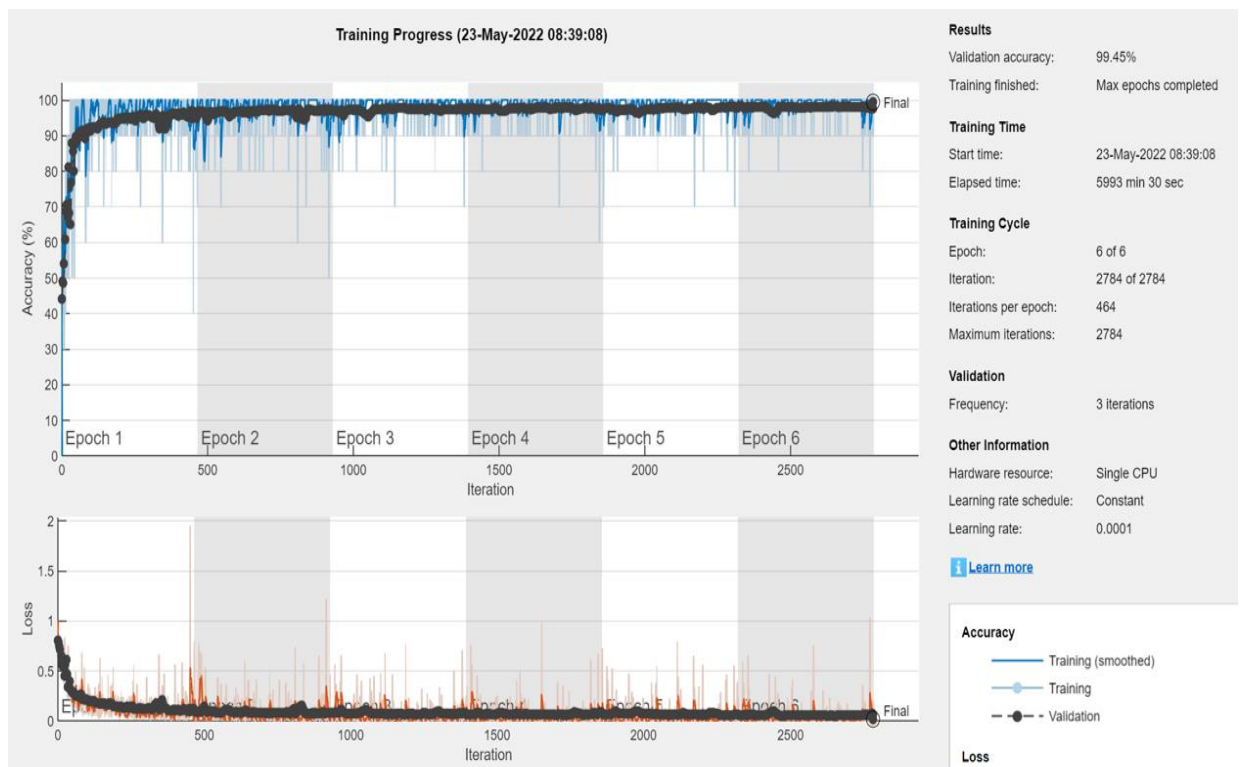


Figure III. 28 : courbe de précision et sa perte de la base 3 à l'aide de ResNet50

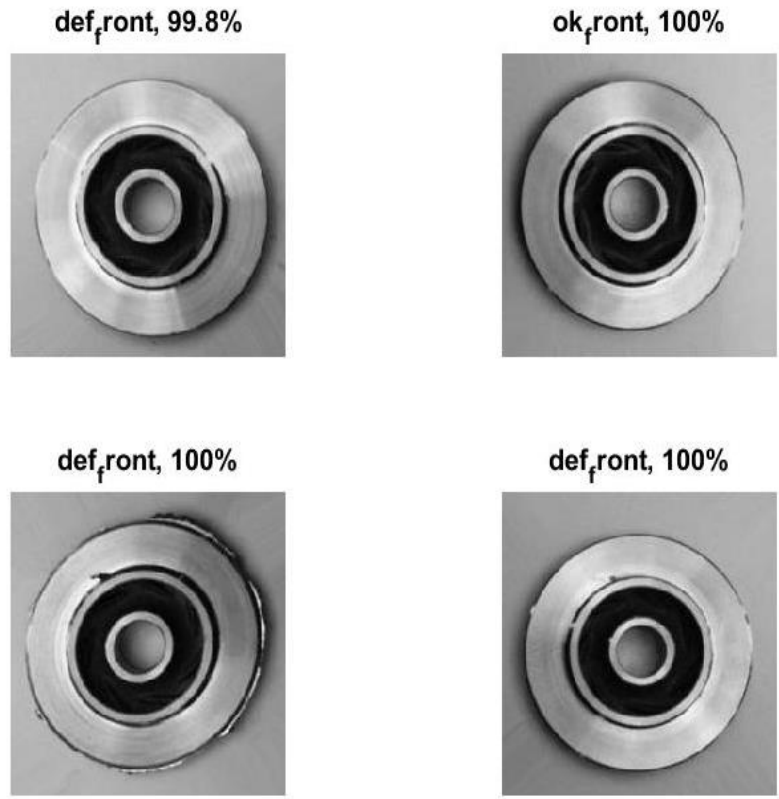


Figure III.29 : Taux d'erreur et de précision par Resnet50

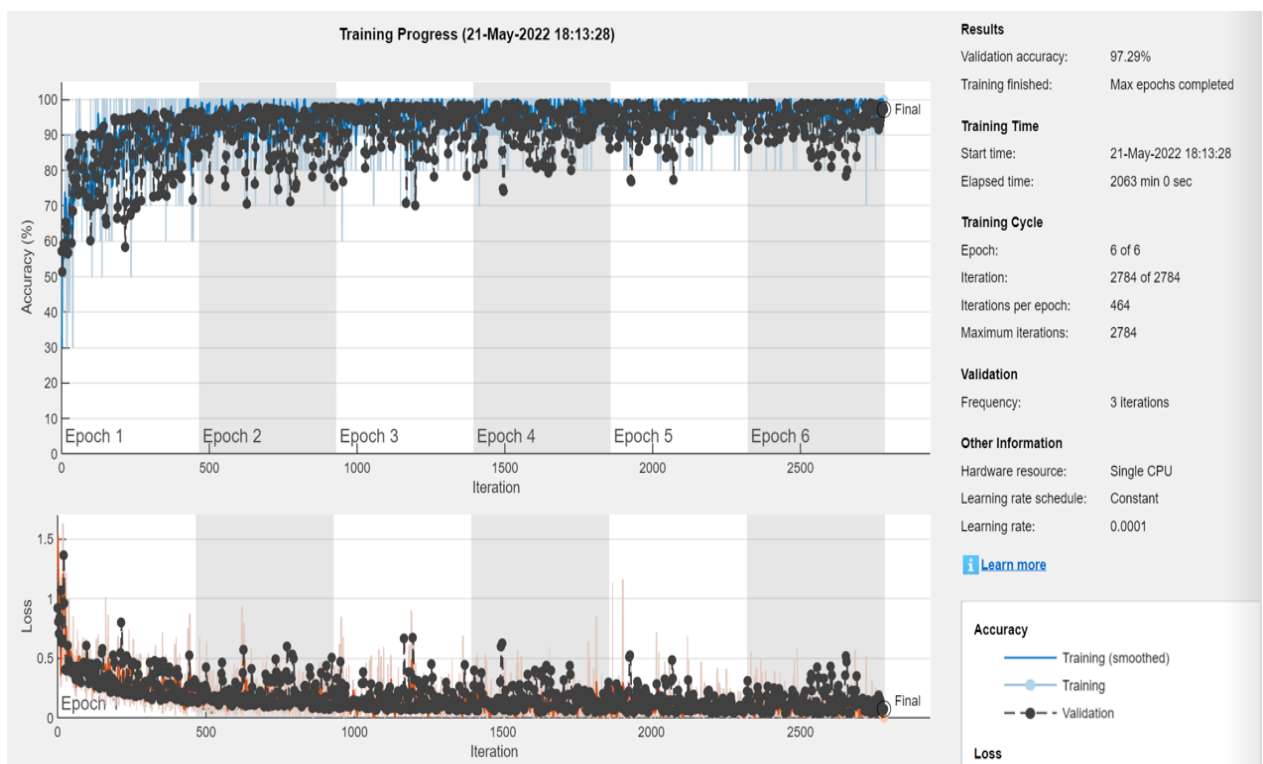


Figure III. 30 : courbe de précision et sa perte de la base 3 à l'aide de googleNet

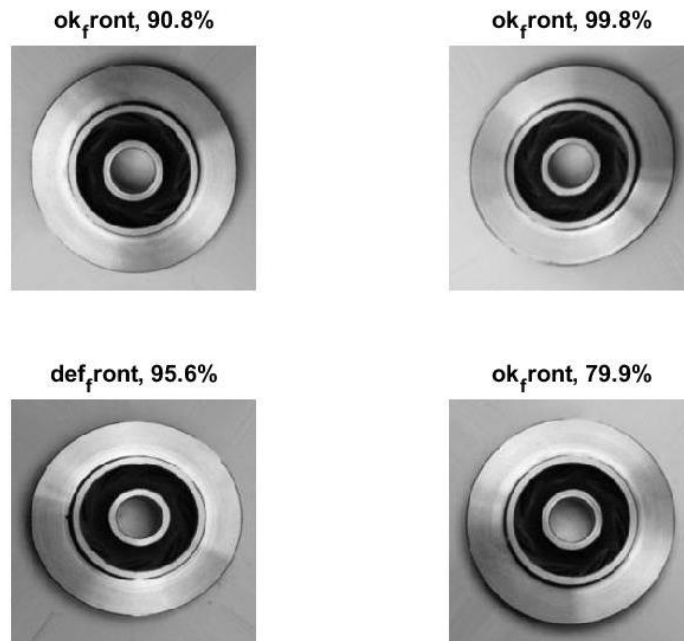


Figure III. 31 : Taux d'erreur et de précision par googleNet

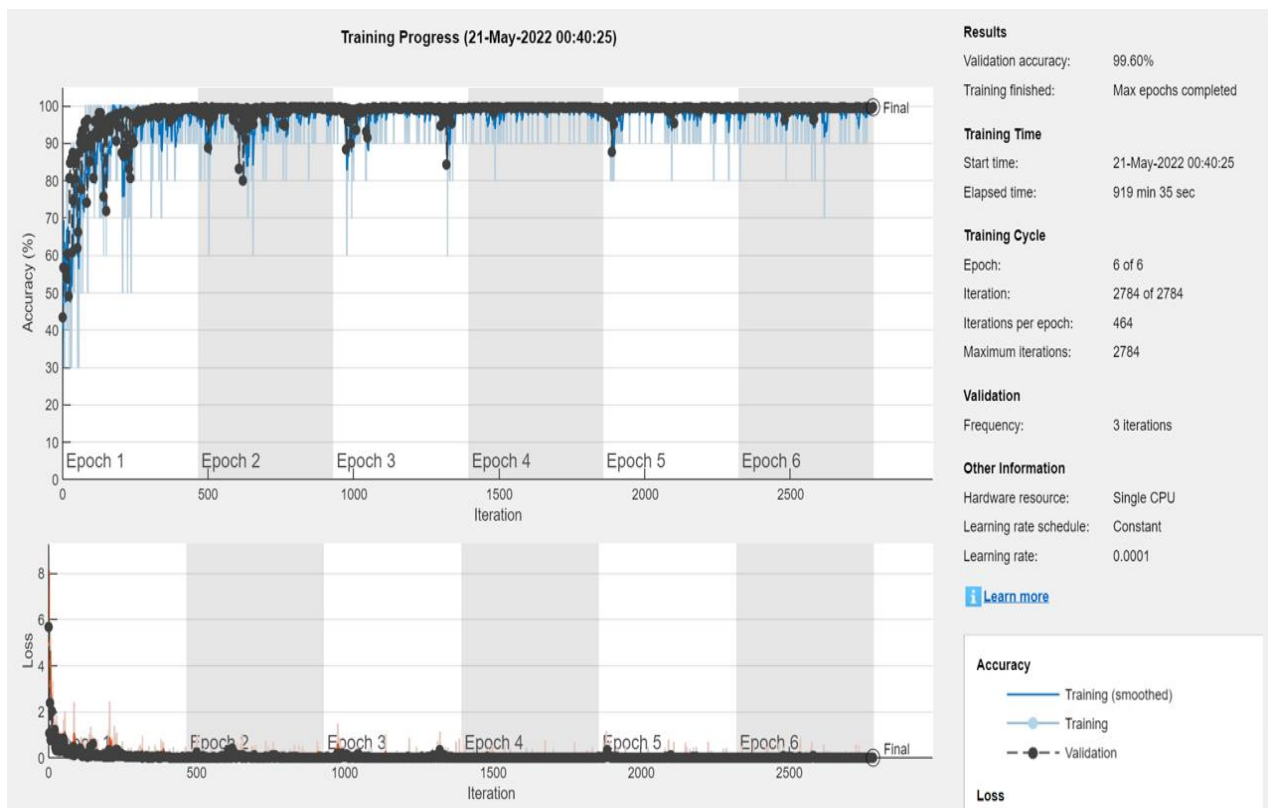


Figure III. 32 : courbe de précision et sa perte de la base 3 à l'aide d'AlexNet

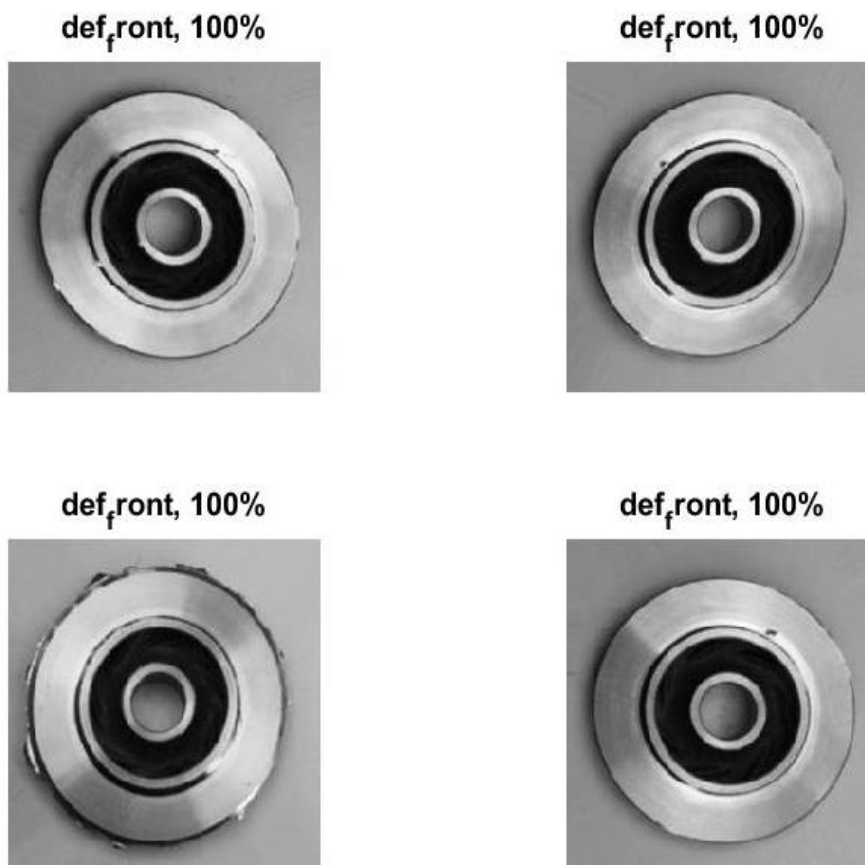


Figure III. 33 Taux d'erreur et de précision par AlexNet

III.5 Conclusion

Dans ce chapitre, nous avons élaboré un système de classification des images en utilisant trois bases de données différentes et deux approches d'Apprentissage par Transfert. Les résultats obtenus sont satisfaisants de la fait que les valeurs de la précision obtenues sont supérieures à 90% pour la plus parts des tests effectués. L'approche basée sur l'extraction de caractéristique présente l'avantage en termes du temps de calcul qui est inférieur par rapport à l'autre approche basée sur le Fine Tuning.

***CONCLUSION GÉNÉRALE ET
PERSPECTIVES***

Conclusion Générale

Au cours des dernières années, principalement en raison des progrès de l'apprentissage en profondeur, la classification d'images a progressé à un rythme spectaculaire.

Dans ce travail, on a utilisé la technique d'apprentissage par transfert issue de l'apprentissage profond pour faire la classification de différentes bases de données. Cette dernière représente une solution aux problèmes liés à l'apprentissage profond ou DL comme la nécessité d'avoir une très large base de données ou des moyens de calculs très performant.

Nous avons commencé par la présentation des notions de base de l'apprentissage Profond basé sur les réseaux convolutionnels. Puis on a détaillé les approches utilisées de l'apprentissage par transfert. : apprentissage fin (Fine Tuning) et extraction de caractéristiques plus le classifieur (SVM). La première consiste à utiliser un réseau déjà entraîné sur une large base de données avec une autre base de données de taille plus petite. La deuxième consiste à remplacer le classifieur utilisé en apprentissage profond par le classifieur SVM.

Les résultats obtenus par les deux méthodes sont satisfaisants de point de vue précision. La deuxième approche présente un meilleur temps d'exécution par rapport à la première pour les trois bases de données utilisées.

En perspective à ce travail, il serait très utile d'implémenter un système pratique de vision industriel à base de données réelles pour bénéficier des performances de l'apprentissage profond.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] : Zemouri, Ryad. *Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance*. Diss. Université de Franche-Comté, 2003.

[2] : Le Cun, Yann, and Françoise Fogelman-Soulié. "Modèles connexionnistes de l'apprentissage." *Intellectica* 2.1 (1987): 114-143

[3] : <https://www.kaggle.com>

[4] : Zhang, Guangcheng, et al. "Boosting local binary pattern (LBP)-based face recognition." *Chinese Conference on Biometric Recognition*. Springer, Berlin, Heidelberg, 2004.

[5] : Déniz, Oscar, et al. "Face recognition using histograms of oriented gradients." *Pattern recognition letters* 32.12 (2011): 1598-1603.

[6] : Jakkula, Vikramaditya. "Tutorial on support vector machine (svm)." *School of EECS, Washington State University* 37.2.5 (2006): 3.

[7] : Ingolia, Nicholas T., Liana F. Lareau, and Jonathan S. Weissman. "Ribosome profiling of mouse embryonic stem cells reveals the complexity and dynamics of mammalian proteomes." *Cell* 147.4 (2011): 789-802.

[8] : McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." *The bulletin of mathematical biophysics* 5.4 (1943) : 115-133.

[9] : Rosenblatt, Frank. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[10] : Gsell, M., and R. Rosenblatt. "III. Vaginal cytology in prolonged pregnancy." *Bulletin de la Federation des societes de gynecologie et obstetrique de langue francaise* 9.1 (1957): 122-129.

[11] : Lepage, Richard. *Reconnaissance d'algues toxiques par vision artificielle et réseau de neurones*. Université du Québec à Chicoutimi, 2004.

[12] : <https://www.jedha.co/formation-ia/vraie-difference-machine-learning-deep-learning>.

[13] : Bachatene, Lyes. "Plasticité du cortex visuel :« homéodynamie » des connexions neuronales et modèle d'effets d'antidépresseurs." (2016).

[14] : <https://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicité/>

[15] : Doublet, Julien, Marinette Revenu, and Olivier Lepetit. "Reconnaissance biométrique sans contact de la main intégrant des informations de forme et de texture." CORESA 2006. 2006.

[16] : <https://www.frontiersin.org/articles/10.3389/fgene.2019.00214/full>.

[17] : <https://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicité/>.

[18]: OUHAME, S., and Mr AMEUR. "Classification automatique des séquences vidéo : État de l'art."

[19] : CHOUINI, MOHAMMED EL MOUNSIF. "Détection d'objets basé Faster R-CNN." (2020).

[20] : Fawaz, Hassan Ismail, et al. "Augmentation de données pour la classification de séries temporelles par réseaux de neurones profonds résiduels." *EGC*. 2019.

[21]: Ding, Xiaohan, et al. "Repvgg: Making vgg-style convnets great again." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[22]: Mohtat, Davoud, and Katalin Susztak. "Fine tuning gene expression: the epigenome." *Seminars in nephrology*. Vol. 30. No. 5. WB Saunders, 2010.

[23]: Rohith Gandhi. Support vector machine—introduction to machine learning algorithms. <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.

[24] : <https://www.kaggle.com/datasets/zhangyunsheng/defects-class-and-location>

Résumé : La classification d'images permet d'élaborer un système capable d'affecter un classement automatique à ces images. Ainsi, ce système remplace une tâche d'expertise qui peut s'avérer coûteuse pour un volume important de données images. Récemment l'apprentissage profond est l'approche la plus utilisée dans ce domaine. Elle permet de modéliser les données à partir de grands ensembles de données apprises. Dans notre mémoire on a utilisé l'apprentissage profond basé sur les réseaux de neurones convolutifs (CNNs) pour faire la classification de différentes bases d'images.

ملخص: يتيح تصنيف الصور إمكانية تطوير نظام قادر على تعيين تصنيف تلقائي لهذه الصور. وبالتالي، فإن هذا النظام يحل محل مهمة الخبرة التي يمكن أن تكون مكلفة لحجم كبير من بيانات الصورة. يعد التعلم العميق مؤخرًا هو النهج الأكثر استخدامًا في هذا المجال. يجعل من الممكن نمذجة البيانات من مجموعات كبيرة من البيانات المكتسبة. في أطروحتنا، استخدمنا التعلم العميق على أساس الشبكات العصبية التلافيفية (CNNs) لتصنيف قواعد بيانات الصور المختلفة.

Abstract: Image classification allows the development of a system capable of assigning an automatic classification to these images. Thus, this system replaces an expert task that can be costly for a large volume of image data. Recently, deep learning is the most widely used approach in this field. It allows to model data from large learned datasets. In our thesis, we used deep learning based on convolutional neural networks (CNNs) to classify different image databases.