

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'enseignement Supérieur et de la Recherche Scientifique

Université M'Hamed Bougara-Boumerdes
Faculté des Hydrocarbures et de la Chimie



Département Transport et Equipements des Hydrocarbures

Mémoire de fin d'études
En vue de l'obtention du diplôme de

Master

Domaine : Sciences et Technologies

Filière : Hydrocarbure

Option : Génie mécanique-Mécanique des chantiers pétroliers

Présenté par :

GRINE Mohamed

HAMEK Badis

THEME

**Identification de la qualité des roues de pompes
submergées en utilisant la méthode des réseaux de
neurones convolutifs CNN**

Soutenu publiquement le 23 / 06 / 2024

Devant le jury :

	Nom et Prénom	Grade	Université/Affiliation
Président de jury	BENBRIK Abderrahmane	Professeur	UMBB
Examinateur	HALIMI Djamel	MA/A	UMBB
Examinateur	HARHOUT Riad	MC/B	UMBB
Encadreur	SABRI Khier	MC/B	UMBB

Remerciements

*En premier lieu, nous tenons à remercier **ALLAH**, le tout puissant de nous avoir donné la force pour accomplir ce travail.*

*Nous tenons à exprimer notre reconnaissance à **M. SABRI Khier** pour avoir acceptée de nous encadrer, son aide et ses encouragements au cours de ce travail. Nos remerciements s'adressent, également, aux membres du jury, d'avoir accepté d'examiner et d'évaluer le travail.*

Le grand remerciement vient aussi aux membres de familles et tous ceux qui, d'une manière ou d'une autre, ont contribué à la réussite de ce travail et qui n'ont pas pu être cités ici.

Nous remercions enfin nos enseignants, responsables et personnels du département de transport et équipements des hydrocarbures, qui ont déployé des efforts inlassables tout au long de notre parcours universitaire.

Dédicaces

Je dédie ce travail à

*Mes chers parents,
Mes frères, ma sœur*

*Et toute ma famille
Mes amis et mes collègues*

*Tous mes enseignants tout au long de mes études, À Tous ceux qui ont
participé de près ou de loin à la réalisation de ce travail.*

Badis.

Dédicaces

C'est avec grand plaisir que je dédie ce modeste travail

A ma très chère mère,

*Quoi que je fasse ou que je dise, je ne saurai point te remercier
comme il se doit. Ta présence à mes côtés a toujours été ma source de
force pour affronter les différents obstacles.*

A mon très cher père,

*Tu as toujours été à mes côtés pour me soutenir et m'encourager. Que
ce travail traduit ma gratitude et mon affection.*

*A mes très chères sœurs, **Amira, Nour-elhouda et Hadjer**, puisse
Dieu vous donner santé, bonheur, courage et surtout réussite.*

*À tous les membres de ma famille et tous mes amis, je dédie ce travail
à tous ceux qui ont participé à ma réussite.*

Mohamed.

Résumé

La présence de défauts dans les processus de fabrication des roues de pompes submergées en acier est inévitable, et pour y remédier, les usines ont toujours eu recours au contrôle humain, qui consiste à vérifier chaque roue produite à l'aide de méthodes manuelles qui ne sont pas toujours très précises. Ce constat nous a incités à rechercher de meilleurs moyens de diminuer les coûts de perte de production en faisant appel à l'IA et à l'apprentissage profond (Deep Learning). Ce travail de mémoire prévoit la construction d'un modèle de classification basé sur des techniques d'apprentissage profond (deep learning). Pour ce faire, nous appliquerons les modèles des réseaux de neurones convolutifs, notamment le modèle CNN simple, le modèle VGG16 et l'apprentissage par transfert, mais également la méthode Grad-Cam permettant de reconnaître et de classer correctement les roues en bon état et les roues présentant des défauts. Au cours de notre étude, nous démontrerons la capacité des modèles CNN à classifier les roues de pompes en se basant sur le contenu de leurs images crues, qui fait l'objet de la présente étude.

Mots-clés: défauts des roues, intelligence artificielle, réseaux de neurones convolutifs.

Abstract

It is inevitable that manufacturing defects will occur in steel submerged pump impellers. To address this, factories have traditionally relied on manual quality control, whereby each impeller produced is checked using methods that are not always highly accurate. This prompted us to investigate alternative methods of reducing production loss costs using AI and Deep Learning techniques. The objective of this thesis is to develop a classification model based on deep learning methods. To this end, we shall apply convolutional neural network models, in particular the simple CNN model, the VGG16 model and transfer learning, as well as the Grad-Cam method for correctly recognizing and classifying both impellers in good condition and those with defects. In the current study, we will demonstrate the ability of CNN models to classify pump impellers based on the raw image content, which is the focus of the present study.

Keywords: impellers defects, artificial intelligence, convolutional neural network.

ملخص

لا يمكن تجنب عيوب التصنيع، وللقضاء عليها لظالما استخدمت المصانع الرقابة البشرية، والتي تتضمن فحص كل جزء يتم إنتاجه باستخدام طرق يدوية ليست دقيقة للغاية دائماً. وقد دفعنا هذا الأمر إلى البحث عن طرق أفضل لتقليل تكاليف خسائر الإنتاج من خلال إشراك الذكاء الاصطناعي والتعلم العميق. الهدف الرئيسي لهذه الأطروحة هو بناء نموذج تصنيف يعتمد على تقنيات التعلم العميق. لذلك سوف نستخدم نماذج الشبكة العصبية التلافيفية (CNN) لتحديد وتصنيف الدفاعات السليمة والمعيبة في المضخات الغاطسة بشكل صحيح. سنوضح في عملنا فعالية نماذج شبكة CNN في تصنيف دفاعات المضخات من صورها الخام، وهو موضوع الجزء التجريبي الخاص بنا.

الكلمات المفتاحية: عيوب الدفاعات، الذكاء الاصطناعي، الشبكة العصبية التلافيفية.

Table des matières

Liste des figures

List des tableaux

Acronymes

Introduction générale

Introduction générale	1
-----------------------------	---

CHAPITRE I: Roues de pompes submersibles : procédés et défauts de moulage.

1	Introduction.....	3
2	Pompes submersibles.....	3
3	Procédé de moulage.....	4
4	Défauts de moulage	5
4.1	Porosités.....	5
4.2	Retrait (Retassure).....	6
4.2.1	Origines et causes de défauts.....	6
4.3	Déchirures chaudes.....	7
4.4	Fermeture à froid.....	7
4.5	Évier	8
4.6	Fusion	8
4.7	Pénétration du métal.....	9
4.8	Inadéquation (Décalages)	9
5	Conclusion	10

CHAPITRE II : Apprentissage profond et traitement d'images.

1	Introduction.....	12
2	Apprentissage machine (Machine Learning).....	13
2.1	Fonctionnement.....	13
2.2	Apprentissage supervisé	13
2.3	Apprentissage non supervisé.....	14
2.4	Apprentissage par renforcement	14
3	Apprentissage profond (Deep Learning)	15
3.1	Fonctionnement.....	15
4	Transfert d'apprentissage	16

4.1	Fonctionnement.....	16
5	Les réseaux de neurones convolutifs (CNN)	17
6	La vision d'ordinateur	17
7	La couche convolutive.....	18
8	Les filtres (kernels).....	19
9	Padding	20
10	Le pas (stride).....	20
11	La couche du pooling	21
11.1	Le maxpooling	21
11.2	Le averagepooling	22
11.3	Le global pooling	22
12	Les fonctions d'activation.....	22
12.1	La fonction ReLu:	23
12.2	La fonction Sigmoidé	24
12.3	La fonction Softmax	24
13	Flattening	25
14	La couche entièrement connectée (FC)	25
15	Output layer.....	26
16	Les paramètres d'un modèle CNN	26
16.1	La normalisation.....	26
16.2	La régularisation.....	27
16.3	L'optimisation.....	27
16.3.1	L'optimiseur Adam.....	27
16.3.2	L'optimiseur SGD.....	28
16.3.3	L'optimiseur ADMM.....	28
16.3.4	L'optimiseur RMSProp.....	29
16.3.5	Le SGD avec Momentum.....	29
16.3.6	AdaGrad	30
17	Les problèmes de l'apprentissage profond:	30
17.1	Le sur-apprentissage :.....	31
17.2	Le sous-apprentissage :.....	31
18	Les métriques de performances.....	33
18.1	La matrice de confusion.....	33
19	Architectures de réseaux de neurones convolutifs	34
19.1	LeNet-5.....	34

19.2	AlexNet.....	34
19.3	VGGNet.....	35
19.4	GoogLeNet.....	36
19.5	Resnet	37
20	Conclusion	38

CHAPITRE III : Identification et classification des défauts constatés sur les roues de pompes en acier.

1	Introduction.....	40
2	Configuration du matériel utilisée :	40
3	Outils et bibliothèques utiles.....	40
3.1	Python.....	40
3.2	Colaboratory	40
3.3	Les bibliothèques.....	41
3.3.1	Matplotlib :	41
3.3.2	TensorFlow.....	41
3.3.3	Keras	41
4	Base de données	41
5	Installation de détection des défauts.....	42
6	Architectures des modèles utilisés	43
6.1	Architecture du modèle CNN simple	43
6.1.1	Résultats obtenus du premier model.....	44
6.1.2	Matrice de confusion :.....	45
6.1.3	Test de prédiction sur les images.....	46
6.2	Architecture du model VGG16	46
6.2.1	Résultats obtenus et Interprétations.	48
7	. Méthode Grad-Cam	50
7.1	Principe de fonctionnement	50
8	Comparaison des résultats	52
9	Conclusion	52

Conclusion générale

Conclusion générale.....	54
Références	55

Liste des figures

Chapitre I

FIGURE I. 1. POMPE SUBMERSIBLE.	3
FIGURE I. 2. IMPULSEUR.	4
FIGURE I. 3. MICROCAVITES.....	5
FIGURE I. 4. DEFAUTS DE RETRAIT.	6
FIGURE I. 5. TYPOLOGIE DES RETRAITS.	7
FIGURE I. 6. DECHIRURES CHAUDES.	7
FIGURE I. 7. FERMETURE A FROID.....	8
FIGURE I. 8. EVIER.....	8
FIGURE I. 9. FUSION	9
FIGURE I. 10. INADEQUATION DU MOULE.....	10

Chapitre II

FIGURE II. 1. L'IA, AM ET L'AP.....	12
FIGURE II. 2. APPRENTISSAGE SUPERVISE VS NON-SUPERVISE.	14
FIGURE II. 3. APPRENTISSAGE PAR RENFORCEMENT.	15
FIGURE II. 4. ARCHITECTURES D'UN RESEAU DE NEURONES CONVOLUTIFS.	17
FIGURE II. 5. COMMENT L'ORDINATEUR VOIT LES IMAGES RGB.	18
FIGURE II. 6. CONVOLUTION PAR FILTRAGE.	19
FIGURE II. 7. APPLICATION DES FILTRES (KERNELS).....	19
FIGURE II. 8. LE PADDING.....	20
FIGURE II. 9. LA METHODE DE STRIDE.	21
FIGURE II. 10. MAXPOOLING.	22
FIGURE II. 11. AVERAGEPOOLING.	22
FIGURE II. 12. LA FONCTION RELU.	23
FIGURE II. 13. LA FONCTION RELU.....	23
FIGURE II. 14. LA FONCTION SIGMOÏDE.	24
FIGURE II. 15. LA FONCTION SOFTMAX.	25
FIGURE II. 16. FLATTENING.	25
FIGURE II. 17. DROPOUT.	27
FIGURE II. 18. UNDERFITTING VS OVERFITTING.	31
FIGURE II. 19. ARCHITECTURE DE LENET-5.....	34
FIGURE II. 20. ARCHITECTURE D'ALEXNET.	35
FIGURE II. 21. ARCHITECTURE DE VGGNET.....	36
FIGURE II. 22. ARCHITECTURE DE GOOGLNET.....	37
FIGURE II. 23. ARCHITECTURE DE RESNET.....	37

Chapitre III

FIGURE III. 1. ROUES NON DEFECTUEUSES.	42
FIGURE III. 2. ROUES DEFECTUEUSES.....	42
FIGURE III. 3. INSTALLATION DE DETECTION DES DEFAUTS.....	43
FIGURE III. 4. ARCHITECTURE DU MODELE CNN SIMPLE.	44
FIGURE III. 5. PRECISION ET PERTE DU MODEL.	45
FIGURE III. 6. MATRICE DE CONFUSION DU MODEL CNN.	45
FIGURE III. 7. IMAGES PREDITES ET LEURS LIBELLES.	46
FIGURE III. 8. LES IMAGES INCORRECTEMENT CLASSES.	46
FIGURE III. 9. ARCHITECTURE D'UN MODEL VGG16.	47
FIGURE III. 10. CONFIGURATION DU MODELE VGG16.....	48
FIGURE III. 11. PRECISION ET PERTE DU MODELE VGG16.....	49
FIGURE III. 12. MATRICE DE CONFUSION DU MODELE VGG16.	49
FIGURE III. 13. PRINCIPE DE FONCTIONNEMENT DU GRAD-CAM.	50

List des tableaux

Chapitre II

TABLEAU II. 1. METHODES POUR EVITER L'OVERFITTING ET L'UNDERFITTING.	32
TABLEAU II. 2. LA MATRICE DE CONFUSION.	33

Chapitre III

TABLEAU III. 1. CARACTERISTIQUES TECHNIQUES DU PC.	40
TABLEAU III. 2. VISUALISATIONS GRAD-CAM DU MODEL VGG16.	51
TABLEAU III. 3. RESULTATS COMPAREES.	52

Acronymes

CNN : Convolutional Neural Network.

RNN : Réseau de neurones récurrent.

ReLU : Unités Rectifié linéaires.

AI : Artificielle Intelligence.

ML : Machine learning (l'apprentissage automatique).

DL : Deep learning (Apprentissage profond).

SGD : Stochastique Gradient Descente.

RMSProp : RootMean Square Propagation (propagation quadratique moyenne).

Adam: Adaptative du moment.

ADMM: Alternating Direction Method of Multipliers.

RGB: red, green, blue.

VGG: Visual Geometry Group.

GradCAM: Gradient-weighted Class Activation Mapping.

AdaGrad: Adaptive Gradient Algorithm.

ResNet: Residual Network.

RAM: Random Access Memory.

GPU : Graphics processing unit.

INTRODUCTION GENERALE

Introduction générale

L'industrie de la fabrication de pompes submergées joue un rôle important dans de nombreux secteurs, tels que le pétrole et le gaz. Ces pompes dépendent fortement de la qualité et de la fiabilité de leurs composants, en particulier les roues, qui peuvent présenter différents types de défauts, notamment des fissures, des déformations, de l'érosion ou du moulage.

Ces défauts peuvent réduire l'efficacité des pompes, entraîner des pannes coûteuses et compromettre la sécurité opérationnelle. Par conséquent, il est crucial de détecter ces défauts le plus tôt possible, idéalement durant la phase de fabrication pour assurer la qualité, réduire les coûts et optimiser la production.

Le présent travail de mémoire porte sur l'application des réseaux de neurones convolutifs pour la classification des défauts présents sur les roues de pompes moulées. Elle a pour objectif principal de concevoir et d'entraîner un modèle CNN simple, et un autre modèle VGG16 en utilisant le transfert d'apprentissage (transfer learning), qui sont capables de classier automatiquement les défauts figurant sur des images de roues moulées, mais aussi de comprendre pourquoi nos modèles ont été conduit à prendre une décision de classification par la méthode de Grad-Cam. Cette recherche devrait permettre d'améliorer considérablement la précision de la classification, ainsi que la réduction du temps d'inspection tout en normalisant les procédés de contrôle de la qualité.

Pour ce faire, le présent mémoire est divisé en trois chapitres :

Au cours du premier chapitre, nous aborderons les défauts qui peuvent apparaître lors du moulage, leurs causes et les solutions envisageables pour les éviter. Les concepts d'intelligence artificielle, d'apprentissage machine ainsi que d'apprentissage profond seront présentés dans le deuxième chapitre, tout en mentionnant les relations entre les trois. Ensuite, nous présenterons les réseaux de neurones convolutifs (CNN), avec leurs caractéristiques en plus de leurs architectures. Quant au dernier chapitre, son contenu sera consacré aux aspects de mise en application, à savoir la classification d'images de roues à l'aide d'un réseau de neurones convolutifs simple CNN, et l'apprentissage par transfert en appliquant un modèle VGG16. Ainsi que l'utilisation du Grad-Cam pour visualiser ce que les réseaux CNN regardent réellement.

**CHAPITRE I: Roues de pompes
submersibles : procédés et défauts de moulage.**

1 Introduction

Une pompe submersible est dotée de roues qui sont l'un des principaux éléments garantissant sa fiabilité et ses bonnes performances. Elles sont souvent fabriquées par moulage de métaux, ce qui implique de verser dans un moule un métal en fusion afin de lui donner la forme souhaitée. Néanmoins, ce procédé peut entraîner une série de défauts qui affectent les performances et la durée de vie des roues de pompes.

2 Pompes submersibles

Une pompe submersible est une pompe conçue pour être immergée dans le liquide qu'elle va pomper. Parmi les pompes submersibles utilisées dans l'industrie pétrolière et gazière, on trouve souvent des pompes électriques submersibles (ESP). Elles permettent notamment la récupération de grandes quantités de fluides issus de puits profonds, de manière économique. Typiquement, ces pompes intègrent un moteur hermétique couplé au corps de la pompe. Le carter étanche, rempli généralement d'huile, enveloppe le moteur et l'empêche d'être court-circuité par des liquides, ce qui l'endommagerait [1].



Figure I. 1. Pompe submersible.

Les principaux éléments d'une pompe submersible sont les suivants :

- **Moteur** : représente l'élément principal qui entraîne les impulseurs (roues tournantes).
- **Impulseur** : Cette partie permet de générer la force centrifuge servant à aspirer le fluide de l'entrée à la sortie, ce qui assure l'action de pompage.



Figure I. 2. Impulseur.

- **Diffuseur** : Le diffuseur est un composant fixe qui convertit progressivement le flux rapide et faible pression de l'impulseur en un flux à haute pression en augmentant la surface de flux.
- **Arbre de rotation** : L'élément qui transmet le mouvement de rotation du moteur à la roue (impulseur).

3 Procédé de moulage

Les roues de pompes submersibles sont généralement fabriquées par moulage du métal. Ce procédé exige un certain nombre d'opérations fastidieuses [2].

- **Préparation du moule** : Il s'agit de la première étape de la fabrication d'une roue.
- **Choix du matériau** : En raison de leur durabilité et de leur résistance à la corrosion, l'acier inoxydable et le plastique sont des matériaux très répandus.
- **Le moulage** : Ce procédé consiste à verser du métal en fusion dans un moule conçu à cet effet, en vue d'obtenir la forme souhaitée de la roue.
- **Équilibrage** : Il est essentiel d'équilibrer les roues et d'éviter tout type de vibration pouvant compromettre les performances de la pompe et l'endommager.
- **Contrôle de qualité** : Dès leur fabrication, les roues subissent une série de contrôles de qualité visant à garantir qu'elles satisfont aux exigences en matière de performances.

Le contrôle de la qualité intervient à un stade déterminant de la chaîne de fabrication des roues. Il permet de détecter les défauts de fabrication, de garantir la fiabilité et de réduire les risques, tout en contribuant à la satisfaction des clients comme au succès de l'entreprise. Cependant, cette opération est trop coûteuse ; au quotidien, le personnel est chargé d'inspecter

les roues pour déceler les défauts, dans des conditions d'efficacité assez limitées et qui peuvent imposer des charges financières accrues à l'entreprise. De ce fait, des algorithmes de type CNN issus de l'apprentissage profond permettent désormais de détecter en temps réel les défauts de fabrication de ces roues.

4 Défauts de moulage

Nous illustrerons notamment les défauts suivants : les défauts de retrait(Retassure), porosités, ceux liés à la métallurgie, celles liées au moulage des métaux, et les défauts dus aux matériaux de moulage et aux formes :

4.1 Porosités

Ces défauts de moulage en coquille sont en général difficiles à identifier. Les pores sont des petits trous, cavités ou poches d'air dans le métal. Leur apparition est due à l'emprisonnement de gaz dans le métal pendant le refroidissement. Cela peut avoir pour conséquence de réduire la résistance du composant produit [2].

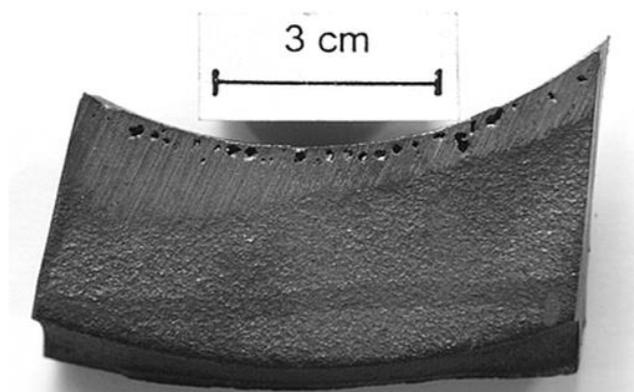


Figure I. 3. Microcavités.

Il existe trois types de porosités [3]:

- **Trous d'épingle (pinholes)** : Il s'agit de petits trous qui se forment à la surface d'une pièce fabriquée par moulage sous pression.
- **Soufflures (blowholes)** : Ils se présentent comme de gros trous à l'intérieur de la pièce moulée.
- **Trous ouverts (open holes)** : C'est un type de soufflure qui apparaît à la surface de la pièce. Ceux-ci proviennent de la présence d'air piégé au cours du moulage de la pièce métallique.

4.2 Retrait (Retassure)

La contraction et le refroidissement du métal fondu se traduit par la formation de cavités ou de vides dans le métal. Cela risque de fragiliser la pièce. Des taux de refroidissement adaptés et la maîtrise d'autres paramètres de moulage permettent d'éviter ce type de défauts [2].

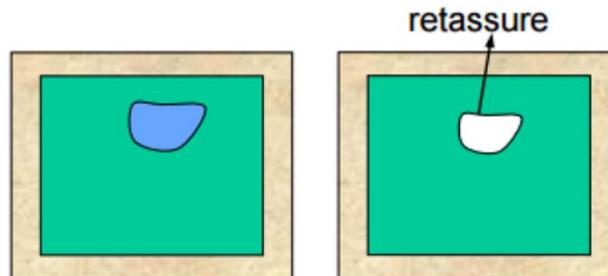


Figure I. 4. Défauts de retrait.

- **Retrait ouverts** : De tels défauts peuvent apparaître à la suite de contraintes internes induites par un refroidissement hétérogène du matériau au cours du processus de solidification. Par ailleurs, la contraction du matériau pendant le refroidissement donne lieu à un retrait ouvert qui prend la forme d'une fissure superficielle à la surface de la pièce moulée.
- **Retrait fermés** : Ce sont les trous internes dus à la contraction hétérogène du métal au cours de la solidification, plutôt que les défauts de retrait ouverts.

4.2.1 Origines et causes de défauts

La présence de défauts de retrait dans la masse au cours du moulage tient à un certain nombre de facteurs [5], dont :

- 1) Un mauvais design du moule.
- 2) Les contraintes non uniformes subies au cours de la solidification de la pièce.
- 3) Taille et emplacement insuffisants dans les fours de refroidissement.
- 4) La composition chimique du métal fondu.
- 5) Température du moulage imprécise (haute ou basse).

En guise de prévention, il est important de contrôler les paramètres de moulage tels que la température, la configuration du moule et l'alimentation avec du métal liquide afin

d'assurer une solidification homogène.

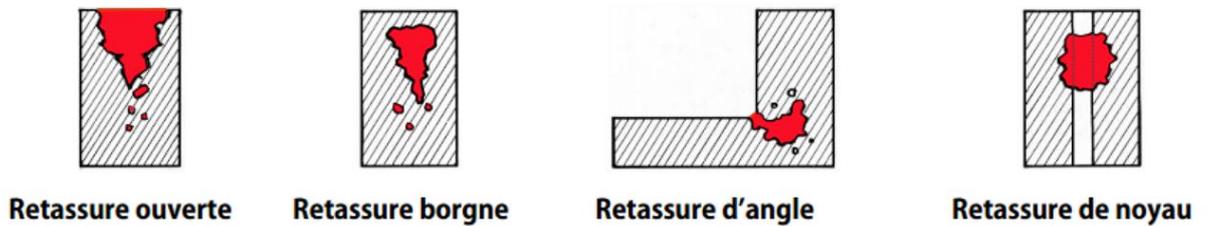


Figure I. 5. Typologie des retraits.

4.3 Déchirures chaudes

Elles se présentent sous la forme de fissures superficielles issues de contraintes de tension internes induites par la solidification du métal à haute température. On parle de fissures "chaudes" car, contrairement aux fractures "froides", qui apparaissent après refroidissement, elles prennent naissance à chaud.

Du fait des pressions thermiques extrêmes subies par le moule sous pression, ce défaut est très fréquent [3].



Figure I. 6. Déchirures chaudes.

4.4 Fermeture à froid

On parle alors de fermeture à froid quand deux flux de métal ne sont pas fondus simultanément. Il peut s'agir de vides ou de joints non soudés, qui peuvent apparaître dans une pièce moulée [3].

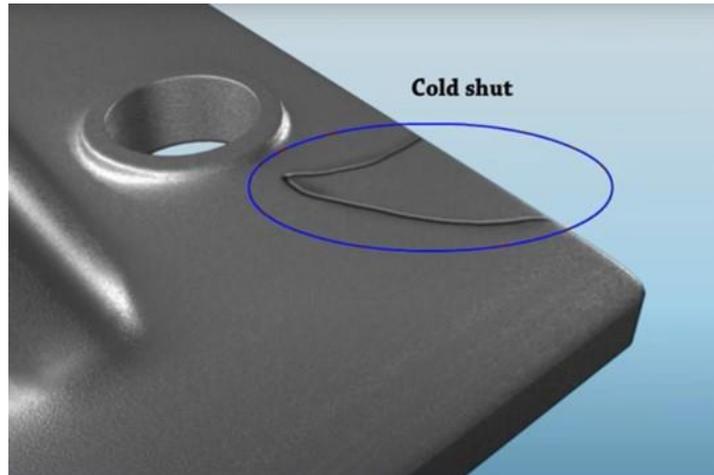


Figure I. 7. Fermeture à froid.

4.5 Évier

Un évier est un impact de pression sur la surface de coulée qui ne copie pas précisément la conception du moule. Des puits apparaissent sur la surface de coulée lorsqu'il y a une cavité sous la surface. Ces dépressions se produisent le long des zones à parois épaisses sur la surface métallique. Étant donné que les éviers réfléchissent la lumière, vous pouvez facilement identifier ces défauts de surface de coulée par des inspections visuelles [3].



Figure I. 8. Evier.

4.6 Fusion

La fusion est un autre défaut de moulage au sable qui se produit lorsque les grains de sable fusionnent avec le métal liquide qui coule. Il en résulte une fine couche de croûte dure et vitreuse fermement attachée au moulage. La principale cause de ce défaut est la faible réfractarité du sable. En raison de la faible réfractarité du sable, il est incapable de résister à la température élevée du métal liquide. Ainsi, un défaut de fusion se produit. [3].



Figure I. 9. Fusion

4.7 Pénétration du métal

Cela se produit dans des situations où le métal en fusion pénètre dans des ouvertures ou des interstices dans le sable de moulage. La principale raison en est que la granulométrie du sable de moulage est trop grossière, empêchant le bon écoulement du métal liquide. Par conséquent, vous obtiendrez une surface de coulée rugueuse et irrégulière, diminuant son esthétique. Vous pouvez facilement voir les effets de la pénétration du métal à l'œil nu. [3].

4.8 Inadéquation (Décalages)

Sont des défauts de moulage qui se présentent en raison d'un mauvais alignement des deux parties, inférieure et supérieure, du moule [3].

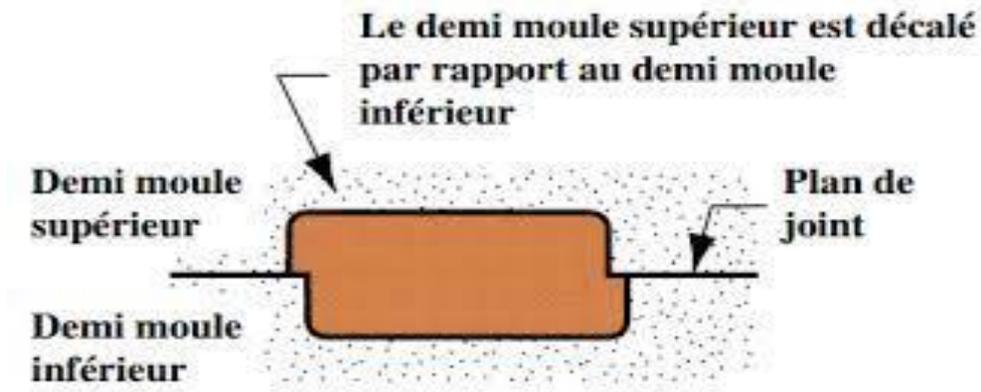


Figure I. 10. Inadéquation du moule.

Il existe deux catégories de désalignement :

- **Le déplacement du moule** : qui prend souvent la forme d'un déplacement horizontal, est causé par le mauvais alignement du moule [3].
- **Le déplacement du noyau** : un déplacement vertical provoqué par un noyau mal aligné [3].

Les goupilles desserrées, celles qui sont mal placées ou le mauvais alignement des deux moitiés du moule en sont la cause principale. Pour remédier à ce problème, il convient d'utiliser les boîtiers de moulage et les goupilles de verrouillage appropriés, et faire en sorte que la matrice soit correctement montée et ajustée [3].

5 Conclusion

Le moulage des métaux se révèle être une opération complexe propice à l'apparition de différents types de défauts de surface. Dès lors que ces défauts sont mieux compris, il est plus facile de contrôler la qualité de la coulée. Les professionnels expérimentés du moulage sont en mesure de prévenir ou de traiter ces défauts, qui peuvent s'avérer normaux dans certaines conditions. Dans cette optique, les fabricants ont besoin d'intégrer des algorithmes d'intelligence artificielle pour contrôler la qualité des pièces moulées dans leur production.

CHAPITRE II : Apprentissage profond et traitement d'images.

1 Introduction

L'intelligence artificielle, ou IA, est le terme utilisé pour décrire la façon dont les machines, en particulier les systèmes informatiques, peuvent simuler les processus de l'intelligence humaine. Ces processus comprennent le raisonnement (l'utilisation de règles pour parvenir à des approximations ou à des conclusions fermes), l'autocorrection et l'apprentissage (l'acquisition de connaissances et de règles pour les appliquer). L'intelligence artificielle (IA) englobe plusieurs approches telles que l'apprentissage machine, qui entraîne des algorithmes sur des données pour générer des prédictions ou des jugements, et l'apprentissage profond, qui est un sous-ensemble de l'apprentissage machine qui utilise des réseaux de neurones multicouches pour évaluer les données. L'IA est utilisée pour automatiser les opérations, fournir des informations et accroître l'efficacité dans divers secteurs, notamment la santé, les services bancaires, le transport, les industries pétrolières et gazières.

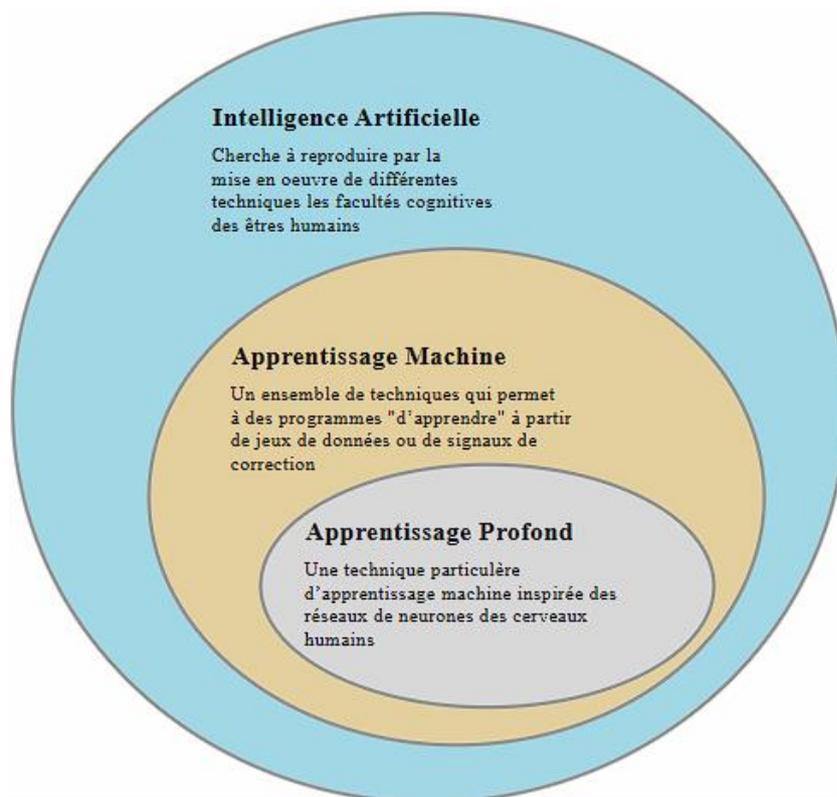


Figure II. 1. L'IA, AM et l'AP.

2 Apprentissage machine (Machine Learning)

Une sous-partie de l'intelligence artificielle (IA) baptisée "Machine Learning" (ML) met l'accent sur la création de modèles capables d'apprendre des données. Des logiciels peuvent devenir plus puissants au fil du temps grâce à la grande variété d'approches que l'apprentissage machine intègre. Les algorithmes utilisés dans l'apprentissage machine (ML) sont entraînés à identifier des modèles et des corrélations de données. Des applications récentes utilisant l'apprentissage machine (ML), notamment ChatGPT, Dall-E 2 et GitHub Copilot, exploitent la base de données disponibles pour établir des prédictions, catégoriser des informations, classer des données, diminuer la dimensionnalité et même contribuer à la mise au point de nouveaux matériaux [5].

2.1 Fonctionnement

La première étape du processus d'apprentissage automatique consiste à insérer des données d'entraînement dans l'algorithme choisi. L'algorithme final de Machine Learning (ML) est développé à l'aide de données d'entraînement, qui peuvent être connues ou inconnues. Le type de données d'apprentissage introduites a un effet sur l'algorithme. Pour vérifier que l'algorithme de Machine Learning (ML) fonctionne correctement, de nouvelles données d'entrée sont introduites. Ensuite, les résultats et les prévisions sont comparés les uns aux autres. L'algorithme est entraîné à plusieurs reprises jusqu'à ce qu'on obtienne le résultat souhaité si les prévisions et les résultats ne correspondent pas. En conséquence, l'algorithme de Machine Learning (ML) peut s'entraîner en continu et générer la meilleure réponse, devenant progressivement plus précis au fil du temps. Le Machine Learning (ML) étant très complexe, on peut le classer en deux catégories : l'apprentissage supervisé et l'apprentissage non supervisé. Chacune d'entre elles utilise différents types de données et a un objectif et une action spécifiques qui donnent des résultats. L'apprentissage supervisé représente environ 70 % de l'apprentissage automatique, tandis que l'apprentissage non supervisé représente les 10 à 20 % restants. L'apprentissage par renforcement occupe l'espace restant [5].

2.2 Apprentissage supervisé

En apprentissage supervisé, les données d'entraînement sont des données identifiées ou marquées. Du fait que les données sont connues, le processus d'apprentissage est supervisé, cela signifie qu'il est guidé pour obtenir des résultats. L'algorithme de machine Learning traite les données entrantes, qui sont ensuite utilisées pour former le modèle. Une fois que le modèle

a été formé à l'aide des données connues, il est possible de l'alimenter avec des données inconnues afin d'obtenir un résultat différent [5].

2.3 Apprentissage non supervisé

Les données d'entraînement en apprentissage non supervisé sont non identifiées et non étiquetées, ce qui indique que personne n'a examiné les données précédemment. L'appellation « non supervisé » vient de l'impossibilité de diriger les données d'entrée de l'algorithme en absence des données connues. Le modèle est formé à l'aide de ces données, qui sont introduites dans le processus de machine Learning. Pour cela, ces algorithmes apprennent à trouver des schémas ou des groupes dans les données, avec très peu d'intervention humaine. En termes mathématiques, l'apprentissage non supervisé implique l'observation de plusieurs occurrences d'un vecteur X and l'apprentissage de la probabilité de distribution $p(X)$ pour ces occurrences [5].

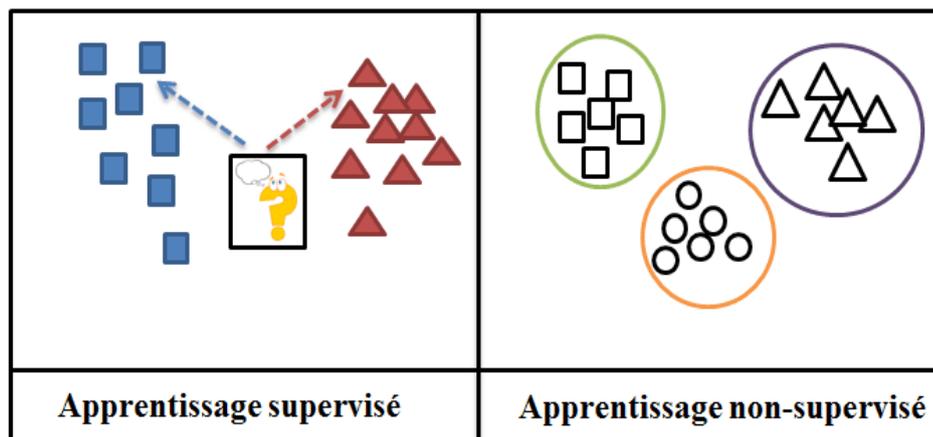


Figure II. 2. Apprentissage supervisé vs non-supervisé.

2.4 Apprentissage par renforcement

À la manière des formes conventionnelles d'analyse des données, l'algorithme recueille ici des données par essais et erreurs avant de déterminer l'activité qui rapporte le plus de récompenses. L'apprentissage par renforcement se compose de trois éléments principaux : les actions, l'environnement et l'agent. Un agent prend des décisions qui optimisent la récompense attendue sur une certaine période de temps, c'est l'apprentissage par renforcement [5].

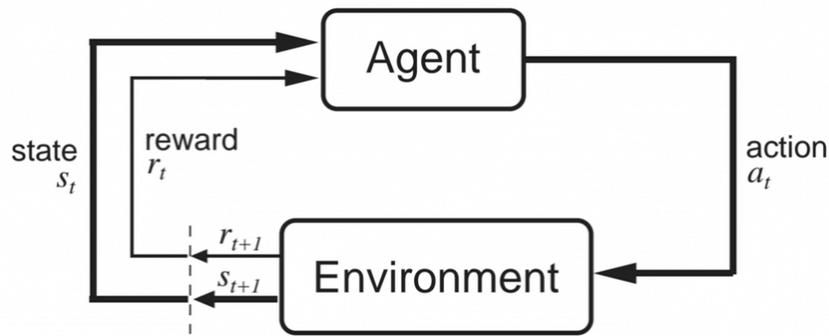


Figure II. 3. Apprentissage par renforcement.

3 Apprentissage profond (Deep Learning)

Un domaine de apprentissage machine appelé « apprentissage profond » s'appuie des réseaux de neurones artificiels. Il est capable de reconnaître des liens et des modèles complexes dans les données. Il n'est pas nécessaire de tout programmer explicitement dans le Deep Learning. En raison de la disponibilité d'énormes ensembles de données et des progrès de la puissance de calcul, le Deep Learning a gagné en popularité ces dernières années. Ces réseaux de neurones, qui sont conçus pour apprendre à partir de grandes quantités des données, s'inspirent de la composition et du fonctionnement des neurones du cerveau humain [6].

3.1 Fonctionnement

Les réseaux de neurones artificiels, également connus sous le nom de réseaux de neurones de Deep Learning, utilisent un mélange de poids, de biais et d'entrées de données pour simuler le fonctionnement du cerveau humain. Ensemble, ces composants permettent une identification, une classification et une description précises des éléments dans les données. Réseaux de neurones profond sont constitués de plusieurs couches de nœuds reliés entre elles, chacune s'appuie sur la précédente afin d'améliorer et d'optimiser la classification ou la prédiction. La propagation vers l'avant est le terme utilisé pour décrire cette évolution du réseau basée sur le calcul. Les couches visibles d'un « deep neural network » sont les couches d'entrée et de sortie. La prédiction ou la classification finale est produite au niveau de la couche de sortie après que le modèle d'apprentissage profond a déjà traité les données de la couche d'entrée. Mais les techniques d'apprentissage profond sont assez compliquées, et différents types de réseaux de neurones sont utilisés pour différents types de problèmes ou d'ensembles de données, notamment [6]:

- Les réseaux de neurones artificiels (ANN)
- Les réseaux de neurones convolutifs (CNN)
- Les réseaux de neurones récurrents (RNN)

4 Transfert d'apprentissage

Le « Transfer Learning » est le processus d'application de la formation préalable d'un modèle d'apprentissage profond à une nouvelle tâche connexe. Par exemple, vous pouvez utiliser les informations que le modèle a appris au cours de la formation pour identifier d'autres choses comme des lunettes de soleil si vous avez formé un classificateur de base pour déterminer si une photo comporte un sac à dos. Essentiellement, l'objectif de transfert d'apprentissage est d'utiliser les connaissances acquises dans un travail pour améliorer la généralisation dans un autre travail. Nous déplaçons les poids appris par un réseau à la « tâche A » vers une nouvelle tâche appelée « tâche B ». À cause de l'énorme puissance de calcul qu'il requiert, le transfert d'apprentissage est généralement utilisé dans les applications de vision artificielle et de traitement du langage naturel, telles que l'analyse des sentiments [7].

4.1 Fonctionnement

Le processus général de « Transfer Learning » est comme suit :

Modèle pré-entraîné : Commencez par un modèle qui a déjà fait l'objet d'une formation approfondie sur un ensemble de données important pour un travail particulier. Ce modèle a été régulièrement entraîné sur de grands ensembles de données et a trouvé des caractéristiques et des modèles communs qui s'appliquent à de nombreuses tâches comparables [7].

Modèle de base : Le modèle pré-entraîné est appelé modèle de base. Il se compose de couches qui ont appris les représentations hiérarchiques des caractéristiques en utilisant les données d'entrée [7].

Couches de transfert : Identifier un groupe de couches dans le modèle pré-entraîné qui enregistre des données générales pertinentes à la fois pour le nouveau travail et pour le travail précédent. Ces couches sont généralement situées vers le haut du réseau, car elles sont susceptibles d'apprendre des informations de bas niveau [7].

Ajustement : Réentraînement des couches sélectionnées avec l'ensemble de données du nouveau défi. Ce processus est ce que nous appelons la mise au point. L'objectif est de préserver les informations de pré-entraînement tout en permettant au modèle d'ajuster ses paramètres afin

de mieux répondre aux exigences de la mission en cours [7].

5 Les réseaux de neurones convolutifs (CNN)

Les réseaux de neurones convolutifs, ou CNN, est un type de réseau neuronal spécialisé conçu pour fonctionner avec des données d'images bidimensionnel, même s'il peut être utilisé avec des données unidimensionnelles et tridimensionnelles. Ces réseaux sont capables d'apprendre à extraire des caractéristiques locales, autrement dit des structures qui se répètent dans l'image. Le cœur du réseau neuronal convolutif est la « couche convolutive ». Celle-ci effectue une opération appelée « convolution » [7].

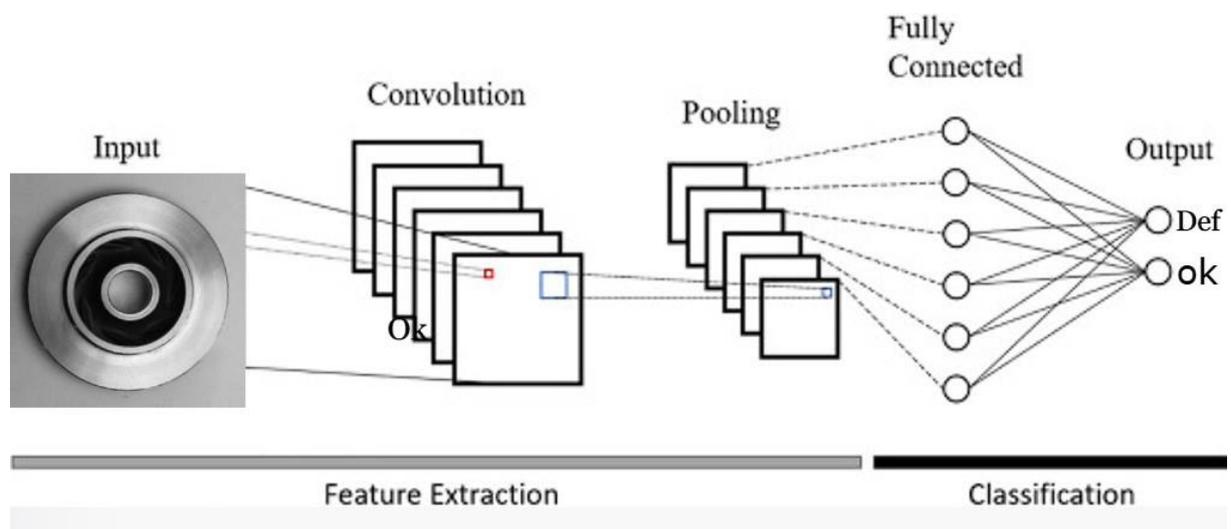


Figure II. 4. Architectures d'un réseau de neurones convolutifs.

6 La vision d'ordinateur

Le nombre moyen de neurones dans le cortex visuel primaire humain adulte dans chaque hémisphère a été estimé à environ 140 millions. Le cortex visuel a de petites parties de cellules qui sont sensibles à des places spécifiques du champ visuel. Certaines cellules neuronales individuelles du cerveau ne répondent que dans la présence des bords d'une certaine orientation. Par exemple, certains neurones s'activent lorsqu'ils sont exposés à des bords verticaux et d'autres lorsque les bords horizontaux ou diagonaux sont affichés.

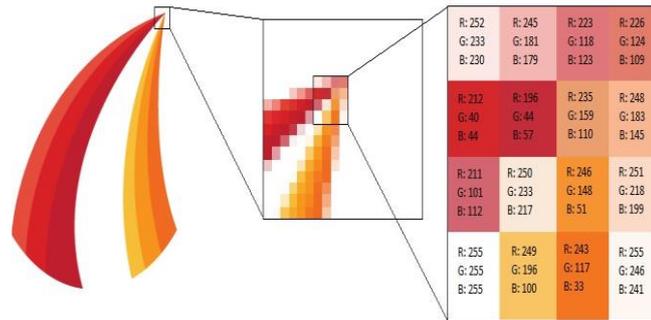


Figure II. 5. Comment l'ordinateur voit les images RGB.

En premier lieu, L'ordinateur voit l'image comme une collection d'intensités de couleur appelée un pixel, dans le cas d'une image colorée RGB, chaque pixel aura trois valeurs Rouge, Vert et Bleu, en conséquence, l'image sera une matrice avec trois dimensions Largeur, Hauteur et Profondeur (profondeur = 3 dans les images RGB).

En second lieu, il est préférable d'utiliser le réseau de neurones artificiels pour la classification des images au lieu de la méthode classique de neuraux networks (Multi layer perception MLP), dans le CNN une petite partie des neurones d'une couche est connectée à la couche qui la précède, contrairement au MLP qui est un network complètement connecté.

7 La couche convolutive

Les réseaux neuronaux à convolution sont particulièrement bien adaptés à la reconnaissance d'images, car la convolution est une méthode mathématique simple qui est souvent utilisée dans le traitement des images. Mathématiquement, la convolution représente une opération qui combine deux fonctions pour en produire une troisième, reflétant comment l'une influence l'autre lorsqu'elles se chevauchent et se déplacent. Concrètement, cela signifie que pour chaque valeur de x , vous prenez une fenêtre glissante de (x) , la superposez sur (x) , multipliez les valeurs des deux fonctions à chaque point où elles se chevauchent, puis intégrez (ou sommez) ces produits pour obtenir la valeur de $(f * g)(x)$.

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau) \cdot g(x - \tau) d\tau \dots \dots \dots 7.1$$

Cette équation représente l'intégrale de la multiplication des deux fonctions $f(\tau)$ et $g(x - \tau)$, où τ est une variable d'intégration. Le filtrage est similaire à la convolution. Une image pouvant être considérée comme un tableau, nous déterminons la taille d'une fenêtre qui entourera l'ensemble de l'image. La fenêtre sera placée en haut à gauche de l'image au début de

la convolution. À partir de là, elle se déplacera vers la droite d'un nombre prédéterminé de carrés (appelé pas), puis d'un pas vers le bas lorsqu'elle atteindra la fin de l'image, et ainsi de suite, jusqu'à ce que le filtre ait traversé toute l'image [8]:

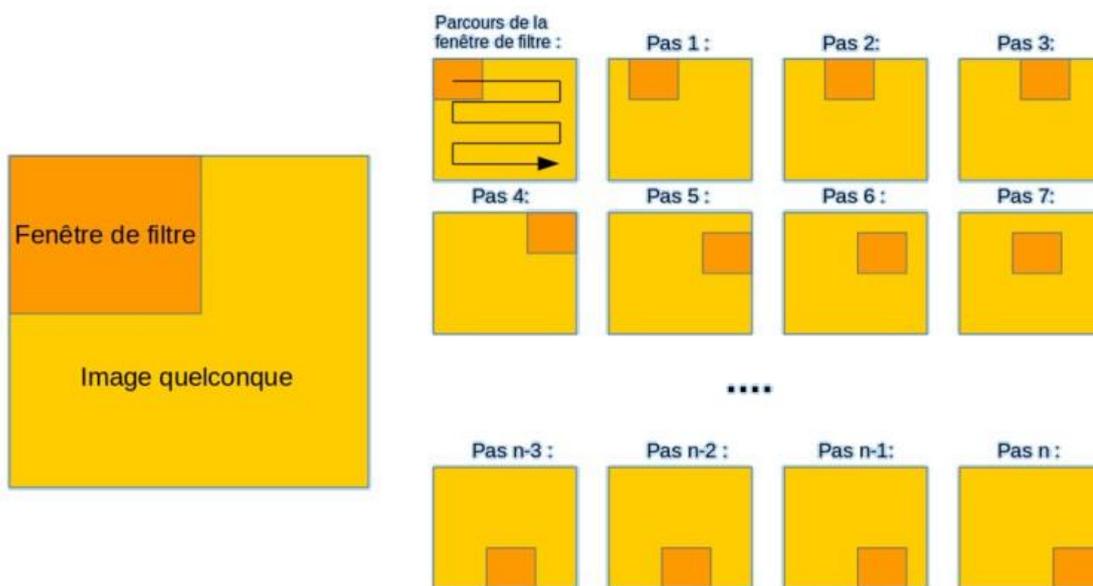


Figure II. 6. Convolution par filtrage.

8 Les filtres (kernels)

Les filtres appliqués dans la convolution sont des petites matrices des poids (weights) d'une forme cubique qui glisse sur les différentes parties de l'image et fait la multiplication avec chacune pour nous donner un output d'un seul pixel [8].

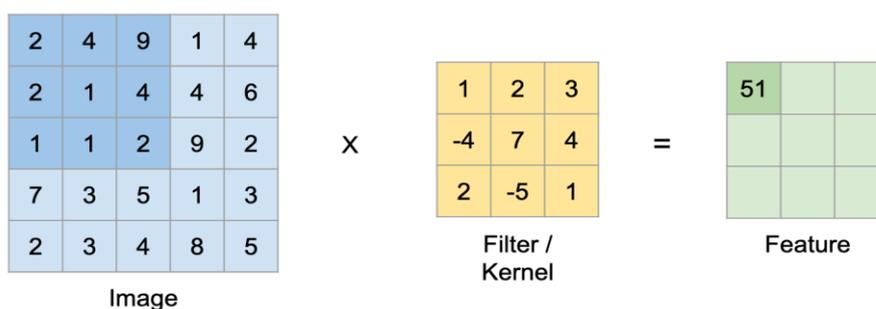


Figure II. 7. Application des filtres (kernels).

9 Padding

Dans les premières couches de notre réseau on veut conserver autant d'informations sur le volume d'entrée original pour qu'on puisse extraire toutes fonctionnalités. Pour faire passer le filtre sur toute l'image il faut remplir les zones autour avec deux pixels nuls. Durant la convolution, la taille de la map de la sortie est déterminée par la taille des données d'entrée, la taille du filtre et le stride. Si on applique un kernel sur la matrice d'entrée, la matrice de sortie sera plus petite que celle de l'input. Cela peut causer des pertes d'informations dans les bordures de l'image. Pour extraire autant d'informations on applique le padding. Le padding est une technique utilisée pour garder les dimensions de l'image après la convolution, il s'agit d'ajouter des pixels nuls autour des bordures de nos inputs. Nombre des pixels ajoutés dépend de la taille du kernel et le résultat qu'on souhaite obtenir de cette opération [9].

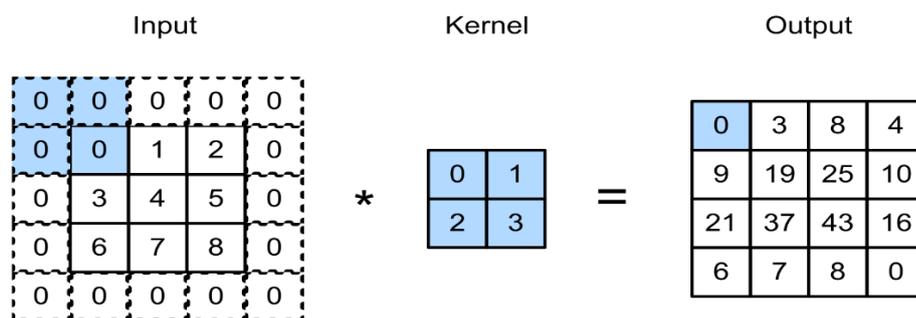


Figure II. 8. Le padding.

10 Le pas (stride)

Stride est un paramètre qui contrôle le glissement des filtres sur l'image d'entrée. Dans la majorité des cas le filtre glisse par un élément à la fois, mais des fois il peut se déplacer par plusieurs éléments pour avoir un output plus simple. Cette méthode est bénéfique lors le kernel est d'une taille importante et il peut capturer beaucoup d'informations des images [9].

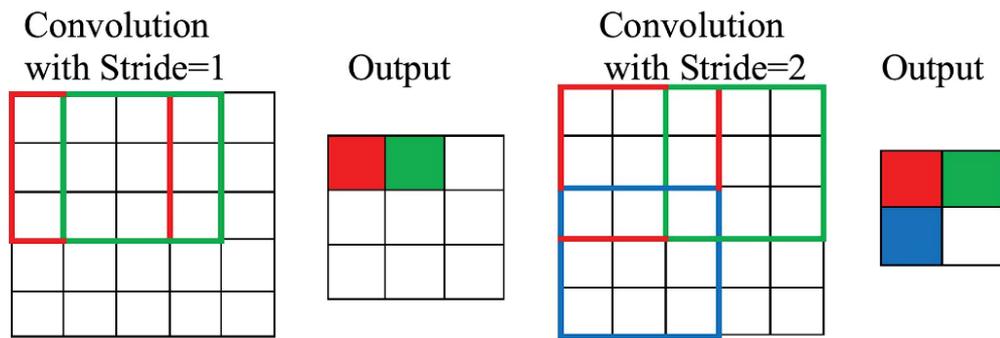


Figure II. 9. La méthode de stride.

11 La couche du pooling

La couche du pooling est souvent placée entre deux couches de convolutions est-elle est connecté à un certain nombre de neurones des sorties des couches précédentes. Le pooling aussi appelé sous-échantillonnage est une opération qui consiste à réduire la taille des images quand elle est trop large tout en préservant ses caractéristiques importantes. La couche de pooling 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. Il existe plusieurs types de pooling mais le max et l'average sont les plus populaires où les valeurs maximales et moyennes sont prises respectivement [9].

11.1 Le maxpooling

Le max pooling est une opération qui sélectionne la valeur maximale de la partie de la carte des caractéristiques couverte par le filtre. Donc, la sortie après cette couche de pooling contient les éléments les plus grands de la carte des caractéristiques précédente.

Hauteur de sortie [17]. :

$$\mathbf{H\ out} = \left\lfloor \frac{\mathbf{H\ in} - \mathbf{K} + 2\mathbf{P}}{\mathbf{s}} \right\rfloor + 1 \dots\dots\dots 11.1$$

Largeur de sortie :

$$\mathbf{W\ out} = \left\lfloor \frac{\mathbf{W\ in} - \mathbf{K} + 2\mathbf{P}}{\mathbf{s}} \right\rfloor + 1 \dots\dots\dots 11.2$$

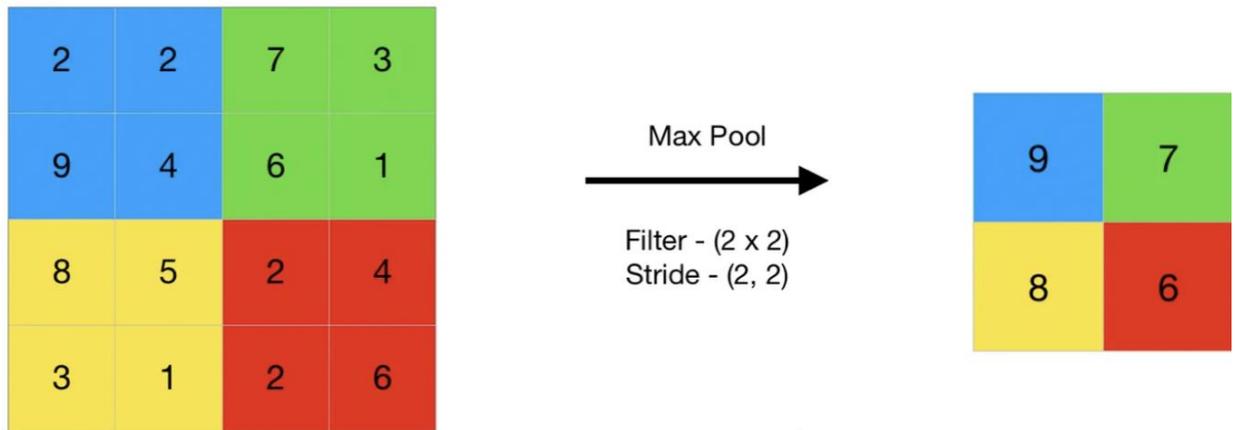


Figure II. 10. Maxpooling.

11.2 Le averagepooling

L'average pooling est une opération qui calcule la moyenne des éléments présents dans une partie de la carte des caractéristiques couverte par le filtre [17].

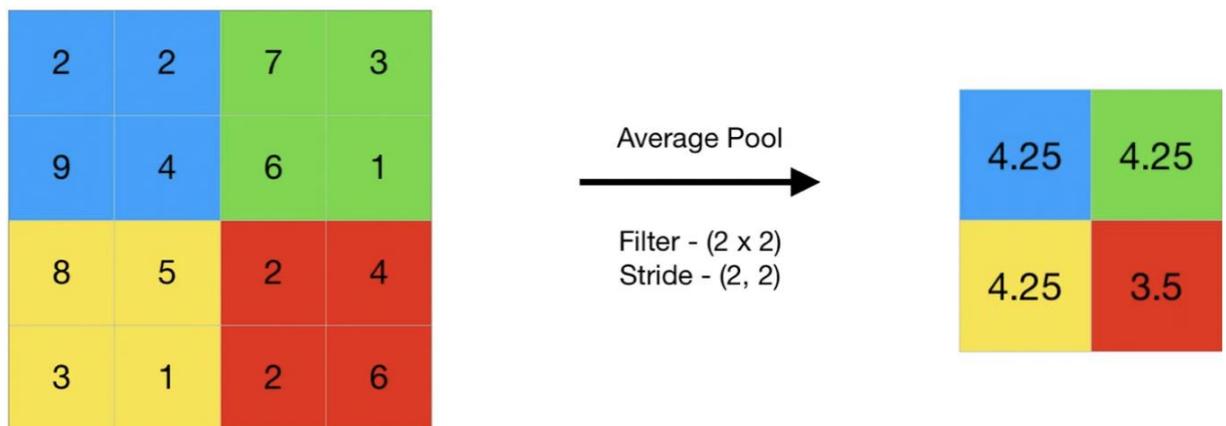


Figure II. 11. Averagepooling.

11.3 Le global pooling

Global pooling est une opération qui change la dimension du 3d en 1D, donc il a un seul effet sur chaque carte de caractéristiques, soit avec le max-pool ou l'average-pool. Il est généralement utilisé à la fin de la convolution pour avoir une forme où on peut appliquer la couche dense pour éviter l'utilisation du Flattening.

12 Les fonctions d'activation

La fonction d'activation est une fonction mathématique appliquée à un signal de sortie d'un neurone artificiel. Le terme de fonction d'activation vient de l'équivalent biologique «

potentiel d'activation », un seuil de stimulation qui, une fois atteint, déclenche une réponse du neurone. La fonction d'activation est une fonction non linéaire [10].

12.1 La fonction ReLu

Le ReLu (Rectified Linear Units) est une fonction réelle non-linéaire définie par :

$$f(x) = \max(0, x) \dots\dots\dots 12.1$$

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

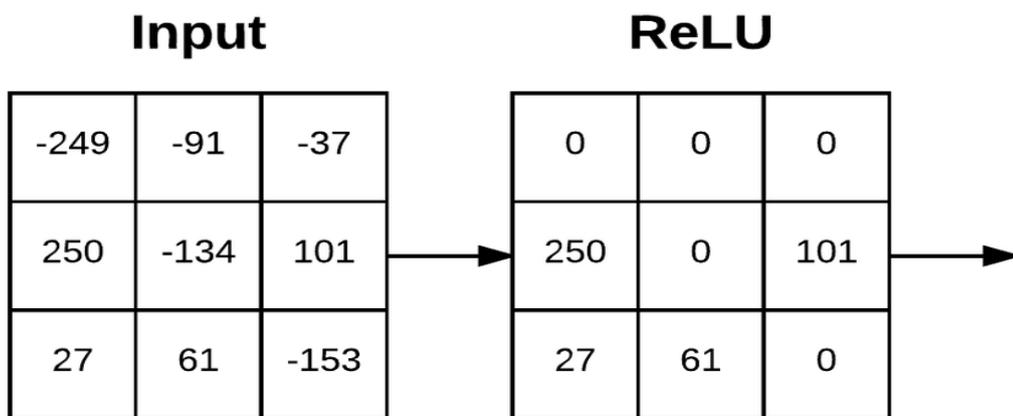


Figure II. 12. La fonction ReLu.

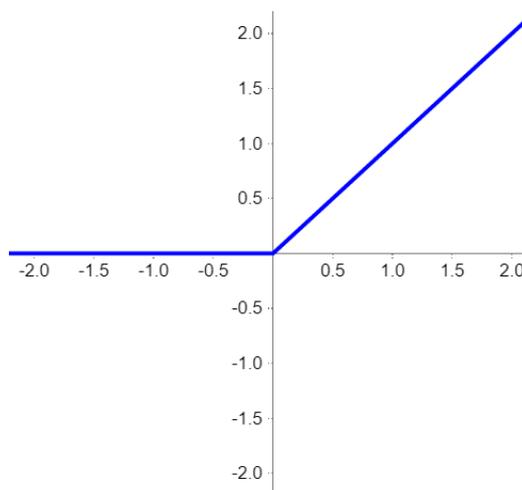


Figure II. 13. La fonction ReLU.

12.2 La fonction Sigmoidé

La fonction Sigmoidé est la fonction d'activation utilisée en dernière couche d'un réseau de neurones construit pour effectuer une tâche de classification binaire. La fonction d'activation sigmoïde permet d'obtenir un résultat ambivalent, donnant une indication sur deux classes à la fois, Elle donne une valeur entre 0 et 1 [10].

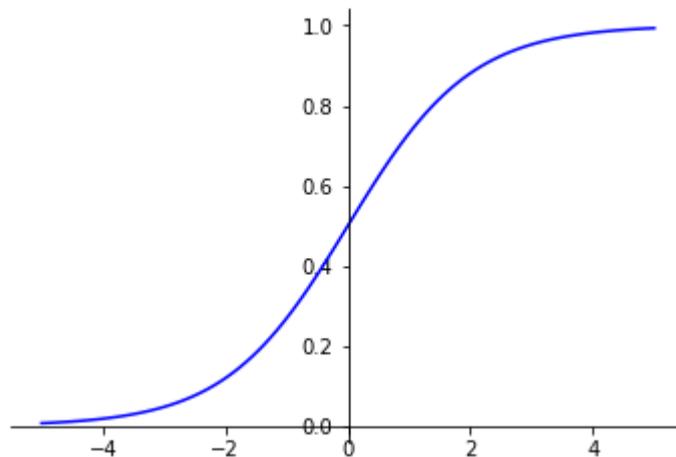


Figure II. 14. La fonction Sigmoidé.

12.3 La fonction Softmax

Elle est utilisée pour dériver la distribution de probabilité d'un ensemble de nombres dans un vecteur d'entrée. La sortie d'une fonction d'activation Softmax est un vecteur dans lequel ses ensembles de valeurs représente la probabilité d'occurrence d'une classe ou d'un événement. Les valeurs dans le vecteur s'additionnent toutes à 1.

On peut écrire ça fonction comme suit [10]:

$$\mathbf{Softmax}(z) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

.....12.2

Pour $i=1, 2, 3, \dots, n$

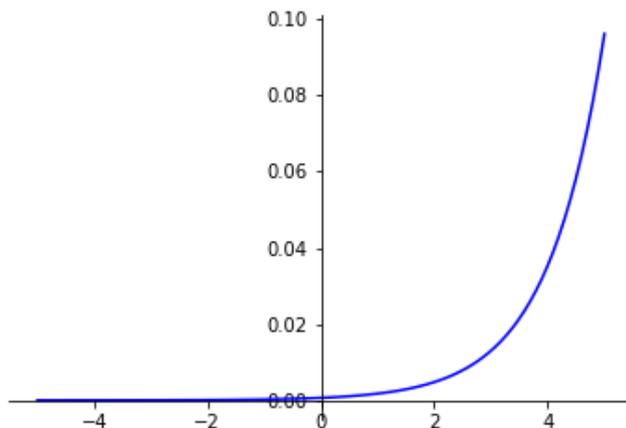


Figure II. 15. La fonction softmax.

13 Flattening

Une fois que l'image est convoluée et max regroupés, la procédure de Flattening (Aplatissement) consiste à tenir essentiellement les éléments dans une carte d'entités groupés et les transformés sous forme vectorielle (Code CNN). Cela devient la couche d'entrée pour la couche entièrement connectée [12].

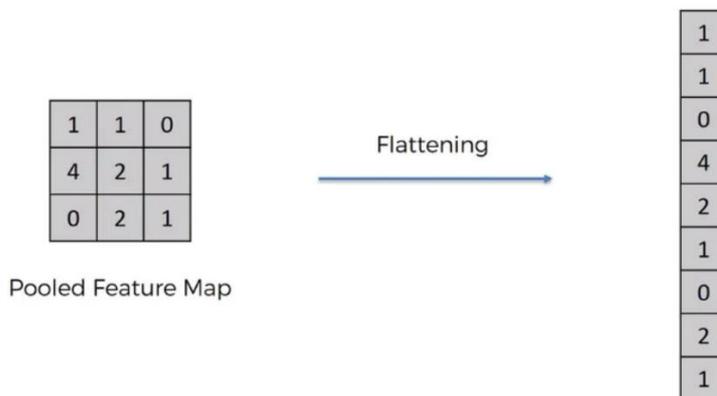


Figure II. 16. Flattening.

14 La couche entièrement connectée (FC)

Après plusieurs couches de convolution et de sous-échantillonnage, le raisonnement de haut niveau dans le réseau neuronal se fait via la couche entièrement connectée

Cette couche a des connexions vers toutes les sorties de la couche précédente, et elle est considérée comme la dernière dans un CNN.

La dernière couche fully-connected permet de classifier l'image en entrée du réseau :

elle renvoie un vecteur de taille N , où N est le nombre de classes dans notre problème de classification d'images. Chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe [12].

15 Output layer

La couche de sortie est chargée de générer les prédictions ou les sorties du réseau. La conception spécifique de la couche de sortie peut varier en fonction de la tâche pour laquelle le CNN est conçu, comme la classification d'images, la détection d'objets... etc.

16 Les paramètres d'un modèle CNN

16.1 La normalisation

La normalisation sert à modifier les valeurs des colonnes numériques dans la base de données à une échelle commune, sans fausser les différences dans les plages de valeurs. La normalisation des données accélère l'apprentissage et conduit à une convergence plus rapide.

L'équation de la Normalisation de l'entrée :

$$\hat{x}_{i,k} = \frac{x_{i,k} - \mu_k}{\sqrt{\sigma_k^2 + \epsilon}} \dots\dots\dots 16.1$$

Tel que :

μ_k et le Moyenne du mini-batch

$$\mu_k = \frac{1}{m} \sum x_{i,k} \dots\dots\dots 16.2$$

σ_k Variance du mini-batch

$$\sigma_k^2 = \frac{1}{m} \sum (x_{i,k} - \mu_k)^2 \dots\dots\dots 16.3$$

$\hat{x}_{i,k}$ est l'activation de l'élément ii du mini-batch pour le canal kk .

μ_k est la moyenne des activations pour le canal kk dans le mini-batch.

σ_k^2 est la variance des activations pour le canal kk dans le mini-batch.

ϵ est la petite constante ajoutée pour éviter la division par zéro.

16.2 La régularisation

Est une technique clé en machine Learning qui a pour rôle de limiter le sur-apprentissage (overfitting) et à contrôler l'erreur de l'apprentissage. Il existe de nombreuses formes de régularisation, qui dépendent de l'objectif recherché et des hypothèses fixées sur le problème. Pour les réseaux de neurones, les méthodes de régularisation les plus populaires sont le Dropout (les poids – paramètres du réseau de neurones – sont remplacés par zéro de manière aléatoire pendant l'entraînement), l'Early Stopping (l'apprentissage s'arrête plus tôt pour favoriser les modèles simples) [13].

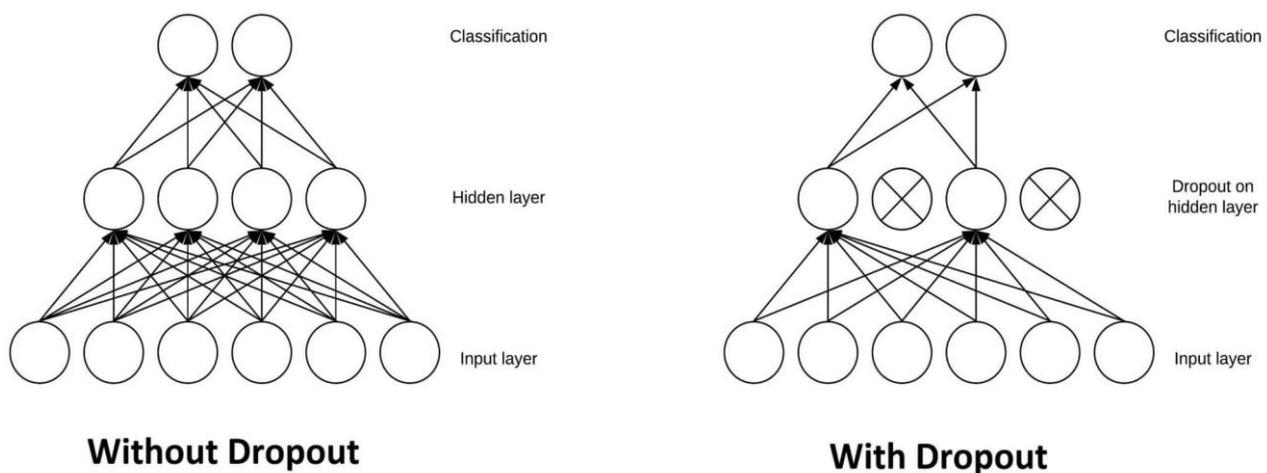


Figure II. 17. Dropout.

16.3 L'optimisation

L'optimisation est une étape essentielle pour entraîner efficacement un CNN à apprendre les caractéristiques des données d'entrée (par exemple, des images) et pour faire des prédictions précises. Elle se fait par l'utilisation des algorithmes d'optimisation, cet ajustement des paramètres peut fournir la meilleure valeur pour le classificateur, pour bien améliorer la vitesse de convergence, la performance du modèle et sa précision, ainsi que la réduction des pertes et du risque de débordement du modèle.

Les optimisateurs les plus utilisés, [14]:

16.3.1 L'optimiseur Adam

Adam est une technique d'optimisation qui peut être utilisée pour mettre à jour les poids du réseau de manière itérative sur la base des données d'apprentissage, à la place du processus

traditionnel de descente de gradient stochastique. Voici l'équation de mise à jour des poids dans l'optimiseur Adam [14]:

$$\begin{aligned}
 \mathbf{m}_t &= \beta_1 \cdot \mathbf{m}_{t-1} + (1 - \beta_1) \cdot \mathbf{g}_t \\
 \mathbf{v}_t &= \beta_2 \cdot \mathbf{v}_{t-1} + (1 - \beta_2) \cdot \mathbf{g}_t^2 \\
 \hat{\mathbf{m}}_t &= \frac{\mathbf{m}_t}{1 - \beta_1^t} \\
 \hat{\mathbf{v}}_t &= \frac{\mathbf{v}_t}{1 - \beta_2^t} \\
 \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\hat{\mathbf{v}}_t + \epsilon}} \cdot \hat{\mathbf{m}}_t
 \end{aligned}
 \dots\dots\dots 16.4$$

- \mathbf{m}_t est une estimation du premier moment du gradient,
- \mathbf{v}_t est une estimation du deuxième moment non centré du gradient,
- β_1 et β_2 sont les coefficients d'atténuation pour les moments du gradient,
- η est le taux d'apprentissage,
- ϵ est un petit terme ajouté pour des raisons de stabilité numérique,
- \mathbf{g}_t est le gradient à l'itération t ,
- $\hat{\mathbf{m}}_t$ et $\hat{\mathbf{v}}_t$ sont les estimations corrigées des moments du gradient.

16.3.2 L'optimiseur SGD

L'optimiseur de descente de gradient stochastique, ou SGD en abrégé, est l'un des optimiseurs les plus basiques utilisés dans la formation des réseaux neuronaux Deep Learning. Voici l'équation de mise à jour des poids pour l'optimiseur SGD [14]:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \cdot \nabla J(\boldsymbol{\theta}_t) \dots\dots\dots 16.5$$

- $\boldsymbol{\theta}_t$ sont les poids du modèle à l'itération t ,
- η est le taux d'apprentissage,
- $\nabla J(\boldsymbol{\theta}_t)$ est le gradient de la fonction de perte J par rapport aux poids $\boldsymbol{\theta}_t$ à l'itération t .

16.3.3 L'optimiseur ADMM

Les problèmes d'optimisation convexe peuvent être résolus à l'aide de la technique d'optimisation ADMM (Alternating Direction Method of Multipliers), qui décompose le problème principal en sous-problèmes plus petits et plus faciles à gérer. L'équation générale de l'optimiseur ADMM peut être exprimée comme suit [14]:

$$\begin{aligned}
 x^{k+1} &= \mathop{\text{arg min}}_x \left(f(x) + \frac{\rho}{2} \|Ax + Bz^k - c + u^k\|^2 \right) \\
 z^{k+1} &= \mathop{\text{arg min}}_z \left(g(z) + \frac{\rho}{2} \|Ax^{k+1} + Bz - c + v^k\|^2 \right) \dots\dots\dots 16.6 \\
 u^{k+1} &= u^k + Ax^{k+1} + Bz^{k+1} - c \\
 v^{k+1} &= v^k + Ax^{k+1} + Bz^{k+1} - c
 \end{aligned}$$

x^{k+1} et z^{k+1} sont les variables mises à jour à l'itération $k + 1$.

u^k et v^k sont les variables auxiliaires à l'itération k ,

ρ est un paramètre de pénalité,

$f(x)$ et $g(z)$ sont les fonctions objectives,

$Ax + Bz = c$ est la contrainte linéaire.

16.3.4 L'optimiseur RMSProp

Pour normaliser le taux d'apprentissage, l'optimiseur RMSprop utilise une moyenne mobile exponentielle du carré des gradients. Les poids de l'optimiseur RMSprop peuvent être mis à jour à l'aide de l'équation suivante [14]:

$$\begin{aligned}
 E[g^2]_t &= \beta E[g^2]_{t-1} + (1 - \beta)(\nabla J(\theta_t))^2 \\
 \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \nabla J(\theta_t) \dots\dots\dots 16.7
 \end{aligned}$$

θ_t est le vecteur des paramètres du modèle à l'itération t ,

$\nabla J(\theta_t)$ est le gradient de la fonction de perte J par rapport à θ_t ,

$E[g^2]_t$ est une estimation exponentielle pondérée des carrés des gradients,

β est le taux d'atténuation pour la moyenne exponentielle,

η est le taux d'apprentissage,

ϵ est un terme de régularisation pour éviter la division par zéro.

16.3.5 Le SGD avec Momentum

Une version de l'optimiseur SGD appelée optimiseur SGD avec momentum comprend un élément de momentum pour accélérer la convergence dans les directions à gradient persistant. La formule suivante peut être utilisée pour mettre à jour les poids dans le SGD à l'aide de l'optimiseur Momentum [14]:

$$\begin{aligned} \mathbf{v}_{t+1} &= \beta \cdot \mathbf{v}_t + (1 - \beta) \cdot \nabla J(\boldsymbol{\theta}_t) \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \eta \cdot \mathbf{v}_{t+1} \end{aligned} \dots\dots\dots 16.8$$

Dans ces équations :

$\boldsymbol{\theta}_t$ est le vecteur des paramètres du modèle à l'itération t ,

$\nabla J(\boldsymbol{\theta}_t)$ est le gradient de la fonction de perte J par rapport à $\boldsymbol{\theta}_t$,

\mathbf{v}_t est la vitesse (momentum) à l'itération t ,

β est le coefficient de momentum, généralement proche de 1,

η est le taux d'apprentissage.

16.3.6 AdaGrad

En divisant le taux d'apprentissage initial par une approximation de la somme des carrés des gradients précédents, l'optimiseur AdaGrad (Adaptive Gradient Algorithm) modifie le taux d'apprentissage de chaque paramètre. Le réajustement des poids de l'optimiseur AdaGrad peuvent être effectuées à l'aide de l'équation suivante [14]:

$$\begin{aligned} \mathbf{G}_{t+1} &= \mathbf{G}_t + (\nabla J(\boldsymbol{\theta}_t))^2 \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\mathbf{G}_{t+1} + \epsilon}} \cdot \nabla J(\boldsymbol{\theta}_t) \end{aligned} \dots\dots\dots 16.9$$

Dans ces équations :

$\boldsymbol{\theta}_t$ est le vecteur des paramètres du modèle à l'itération t ,

$\nabla J(\boldsymbol{\theta}_t)$ est le gradient de la fonction de perte J par rapport à $\boldsymbol{\theta}_t$,

\mathbf{G}_t est une somme cumulative des carrés des gradients jusqu'à l'itération t ,

η est le taux d'apprentissage initial,

ϵ est un terme de régularisation pour éviter la division par zéro.

17 Les problèmes de l'apprentissage profond:

L'Overfitting (sur-apprentissage), et l'Underfitting (sous-apprentissage) sont les causes principales des mauvaises performances des modèles prédictifs générés par les algorithmes de Machine Learning [15].

17.1 Le sur-apprentissage :

Cela représente un modèle qui a appris par cœur ses données d'entraînement, qui fonctionne donc bien sur le jeu d'entraînement mais pas de validation. Par conséquent, le modèle prédictif capturera tous les « aspects » et détails qui caractérisent les données du training set. Dans ce sens, il capturera toutes les fluctuations et variations aléatoires des données du training set. Quand un tel événement se produit, le modèle prédictif pourra donner de très bonnes prédictions sur les données du Training Set (les données qu'il a déjà vues et auxquelles il s'y est adapté), mais il prédira mal sur des données qu'il n'a pas encore vues lors de sa phase d'apprentissage. On dit que la fonction prédictive se généralise mal. Et que le modèle souffre d'Overfitting [15].

17.2 Le sous-apprentissage :

Ce cas-ci représente un modèle qui n'arrive pas à déduire des informations du jeu de données. Il n'apprend donc pas assez et réalise de mauvaise prédiction sur le jeu d'entraînement, donc il s'adapte mal au training set.

Autrement dit, le modèle prédictif n'arrive même pas à capturer les corrélations dues données. Par conséquent, le coût d'erreur en phase d'apprentissage reste grand. Bien évidemment, le modèle prédictif ne se généralisera pas bien non plus sur les données qu'il n'a pas encore vu. Finalement, le modèle ne sera pas viable car les erreurs de prédictions seront grandes. Dans ce cas de figure, on dit que le modèle souffre d'Underfitting [15].

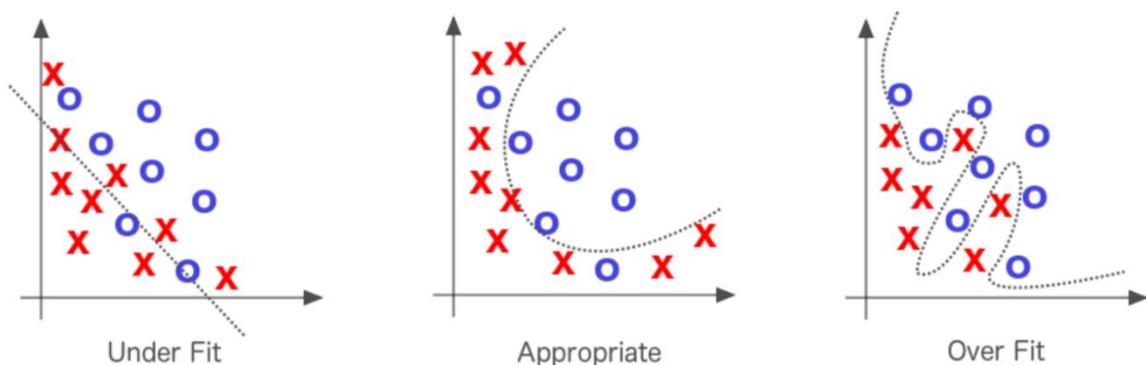


Figure II. 18. Underfitting vs Overfitting.

Méthodes pour éviter l'overfitting et l'underfitting, [15]:

Tableau II. 1. Méthodes pour éviter l'Overfitting et l'Underfitting.

Overfitting	Underfitting
<p>-Dropout : ignorer aléatoirement des neurones pendant l'entraînement pour empêcher le modèle de devenir trop dépendant de certains neurones.</p> <p>-Élargir la base de données : utiliser plus de données d'entraînement ou appliquer des techniques d'augmentation des données pour créer plus de variations.</p> <p>-Simplifier le modèle : réduire le nombre de couches ou de neurones pour limiter la capacité du modèle à mémoriser les données.</p> <p>-Early stopping : arrêter l'entraînement lorsque la performance sur le set de validation commence à se dégrader.</p>	<p>-Ajouter plus de couches ou de neurones au réseau de neurones pour augmenter sa capacité à apprendre des fonctionnalités complexes.</p> <p>-Augmenter le nombre d'époques d'entraînement pour permettre au modèle de mieux apprendre les données.</p> <p>-Choisir ou concevoir des fonctions de perte qui pénalisent correctement les erreurs du modèle.</p> <p>-Utiliser plus de données d'entraînement ou appliquer des techniques d'augmentation des données pour fournir au modèle plus de variations et d'exemples à apprendre.</p>

18 Les métriques de performances

18.1 La matrice de confusion

Cette matrice a pour but de visualisé les résultats ou les performances d'un model CNN ou un modèle de classification, dans une forme d'un tableau come le suivant :

Tableau II. 2. La matrice de confusion.

Prédit Positif	Prédit Négatif	
Vrai Positifs (VP)	Faux Négatifs (FN)	Réel Positif
Faux Positifs (FP)	Vrai Négatifs (VN)	Réel Négatif

- **Vrai Positifs (VP)** : Le nombre des pièces correctement classés comme appartenant à la classe positive.
- **Faux Positifs (FP)** : Le nombre des pièces incorrectement classés comme appartenant à la classe positive.

Taux de bonne classification (Accuracy) est l'indicateur le plus naturel et le plus évident permettant d'évaluer les performances d'un système de classification. Cette valeur simple à calculer, détecte la fréquence à laquelle notre modèle est correct

$$\text{Accuracy} = \frac{VP+VN}{VP+VN+FP+FN} \dots\dots\dots 18.1$$

Précision est définie par le nombre de vrais résultats positifs divisé par le nombre total de résultats positifs prédits par le classificateur

Rappel (Recall) est défini comme le rapport des vrais positifs (VP) sur le nombre total des éléments réels positifs (VP + FN).

$$\text{Recall} = \frac{VP}{VP+FN} \dots\dots\dots 18.2$$

F1 score est une mesure de la performance d'un modèle de classification, telle qu'un réseau de neurones convolutionnel (CNN). Il combine la précision et le rappel (ou sensibilité) en une seule métrique. Le F1-score est particulièrement utile lorsque les classes sont déséquilibrées, car il prend en compte à la fois les faux positifs et les faux négatifs. Le F1-score

est défini comme la moyenne harmonique de la précision et du rappel. La formule mathématique est la suivante:

$$F1\text{-score} = 2 \times \frac{Précision \times Rappel}{Précision + Rappel} \dots\dots\dots 18.3$$

19 Architectures de réseaux de neurones convolutifs

Les architectures de réseaux de neurones convolutifs obtiennent souvent de grandes performances dans le domaine de l'apprentissage profond. Leur facilité d'utilisation les rend très utilisées. Il en existe plusieurs, nous allons en exposer les plus populaires :

19.1 LeNet-5

LeNet-5 est une architecture de réseau neuronal convolutif (CNN) développée par Yann LeCun en 1989. Et il est considéré comme la première mise en œuvre réussie d'un CONVNET pour la classification d'images. Il a 2 couches convolutionnelles (CONV) et 3 couches Fully Connected (FC) (d'où 5 est le nombre de couches CONV et FC dont ils disposent), et il a environ 60000 paramètres. Il a été appliqué par plusieurs banques pour reconnaître et classer les chiffres manuscrits de l'ensemble de données, tels que les chèques, mais a également été appliqué à d'autres tâches de classification d'images. Malgré son âge, LeNet-5 reste un modèle populaire et influent dans le domaine de l'apprentissage en profondeur, et son architecture de base a inspiré de nombreux CNNs ultérieurs [16].

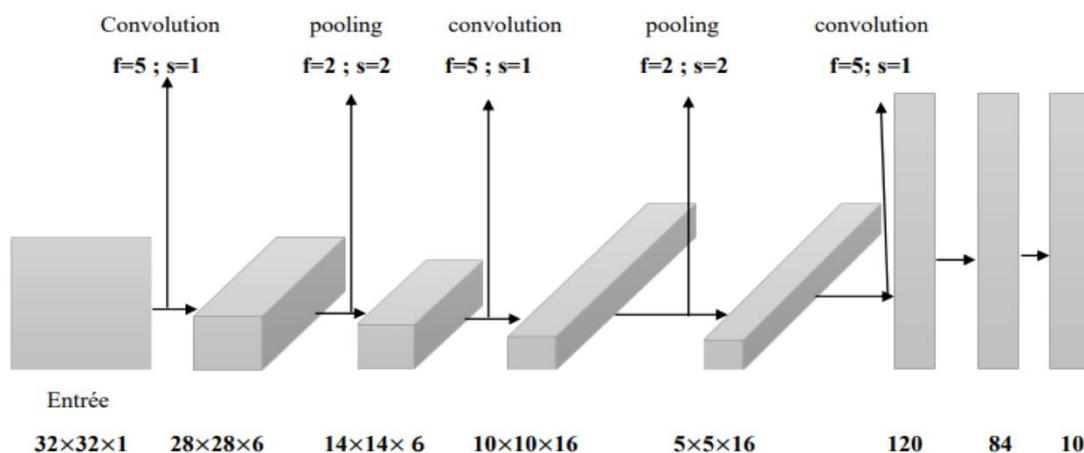


Figure II. 19. Architecture de LeNet-5.

19.2 AlexNet

AlexNet est une architecture de réseau de neurones convolutif (CNN) créée par Krizhevsky Ilya Sutskever, and Geoffrey Hinton en 2012, cet événement marque un tournant dans l'apprentissage profond. Il a obtenu des performances meilleures que les architectures CNN précédentes et a remporté le défi de reconnaissance visuelle à grande échelle ImageNet (ImageNet Large Scale Visual Recognition Challenge nommé ILSVRC) en 2012 avec un résultat top-5 avec 15.3% d'erreurs. Le deuxième meilleur résultat avait le taux d'erreur de 26,2%, qui représente une énorme amélioration des performances. Il est composé de 5 couches de convolutions combinées avec des couches Maxpooling, 3 couches entièrement connectées et 2 couches dropout. La fonction d'activation utilisée dans toutes les couches est ReLu. Et la fonction d'activation utilisée dans la couche de sortie est Softmax. Cette architecture a environ 60 millions paramètres. Dans l'ensemble, AlexNet est une contribution significative au domaine de l'apprentissage profond, et son succès sur l'ensemble de données ImageNet a contribué à faire des CNNs un outil puissant pour les tâches de classification d'images [16].

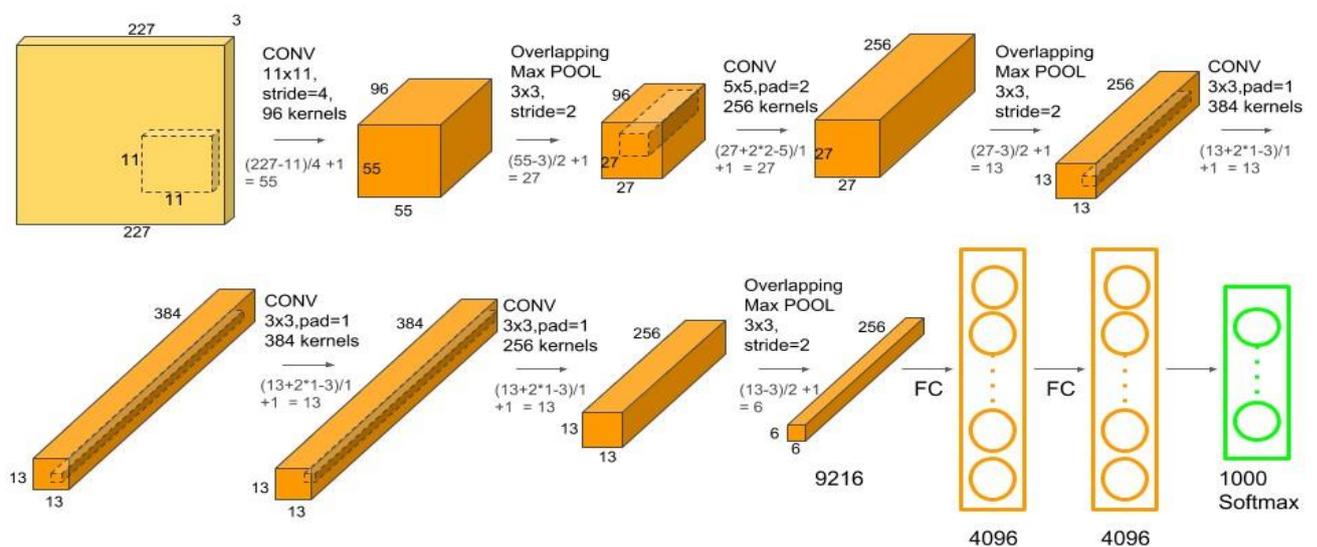


Figure II. 20. Architecture d'AlexNet.

19.3 VGGNet

VGGNet est une architecture de réseaux de neurones convolutifs proposée par l'équipe de Visual Geometry Group (VGG) en 2014 et ont inventé le VGG16, et il a obtenu la deuxième

place lors de ILSVRC dans la même année. Il est connu par sa simplicité avec deux ou trois couches de convolutions, puis une couche de pooling, s'ensuit une série similaire alternante de deux ou trois couches de convolutions et une de pooling jusqu'à avoir 16 couches de convolutions, et finir avec plusieurs couches complètement connectées successives, avec 138 millions de paramètres. Ils ont également conçu un autre modèle plus profond, VGG-19. Cela représentait 7,3% du taux d'erreur dans le top 5 lors la compétition ILSVRC. [16].

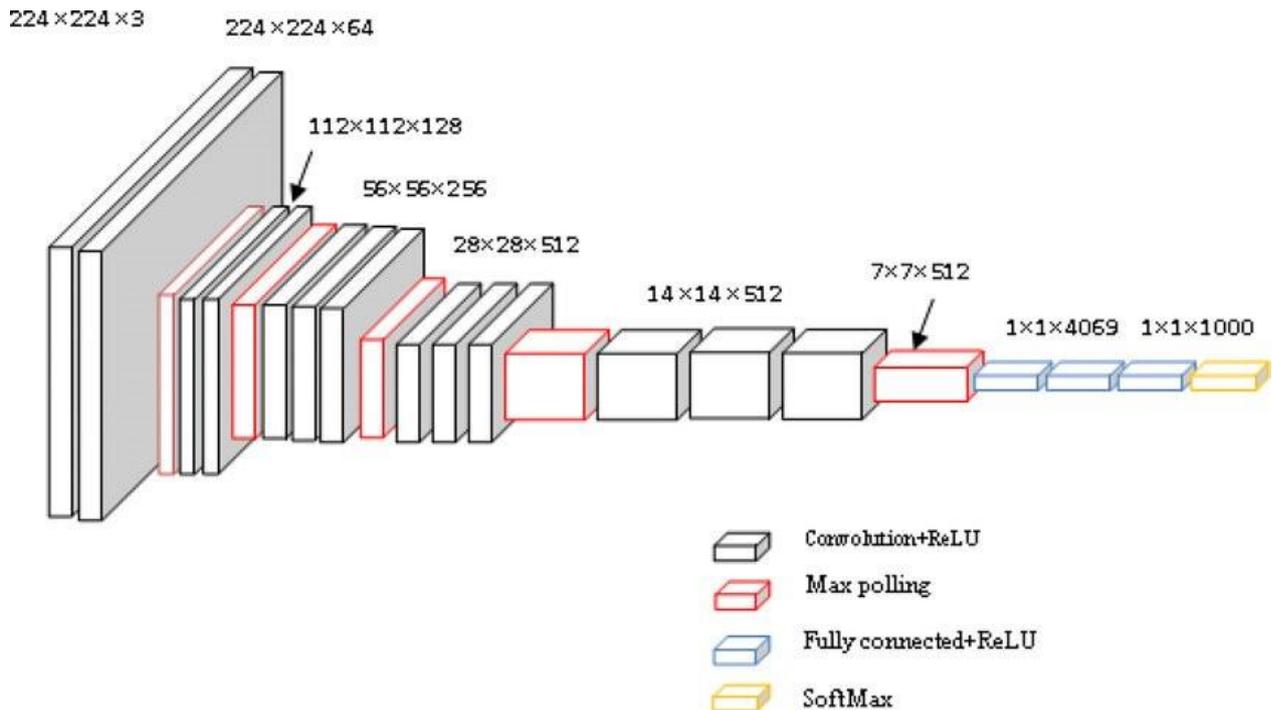


Figure II. 21. Architecture de VGGNet.

19.4 GoogLeNet

GoogLeNet aussi connu comme Inception v1 est une architecture CNN développée par des chercheurs de google en 2014 qui a remporté la première place lors du ILSVRC 2014 avec un taux d'erreur de 6.7%, 0.6% plus précis que VGG. Il a joué un rôle significatif dans le développement et l'avancement du domaine de la classification des images, et il est largement utilisé aujourd'hui. Ce modèle a introduit un nouveau module appelé Inception Module v1, composé de calculs parallèles de convolutions et de maxpooling qui peuvent aller plus profondément dans le réseau. Il a aussi utilisé pour la première fois des convolutions 1×1 qui ont réduit les dimensions des inputs avant d'appliquer des grandes convolutions. Depuis qu'il a été lancé, GoogLeNet a eu un impact marquant qui a inspiré le développement d'autres modèles comme le V2, et le v3 [16].

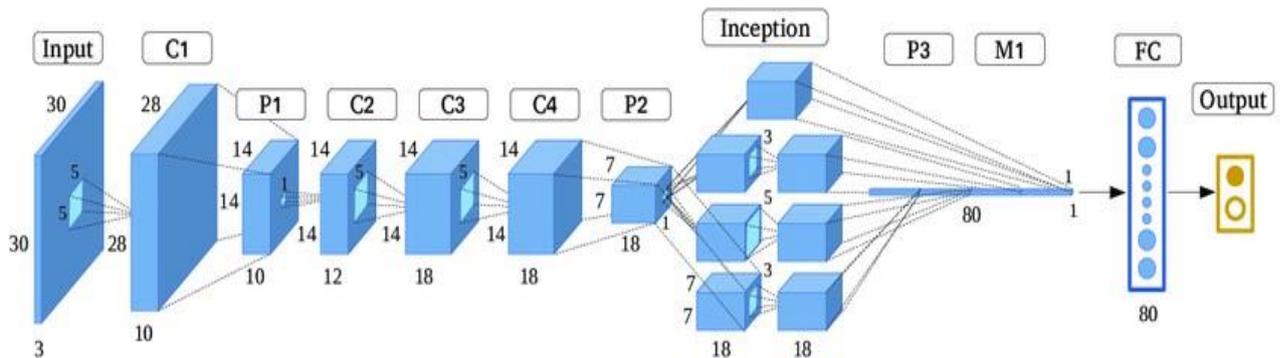


Figure II. 22. Architecture de GoogleNet.

19.5 Resnet

ResNet (Residual Network) est une architecture de réseau neuronal convolutif (CNN) développée par Kaiming He en 2015. Conçu pour classer les images du jeu de données ImageNet. Il a obtenu des performances nettement meilleures que les architectures CNN précédentes et a remporté le ImageNet Large Scale Visual Recognition Challenge ILSVRC en 2015 avec un taux d'erreurs de 3,57%. ResNet est connu pour son utilisation de connexions résiduelles, qui sont des connexions raccourcies qui sautent une ou plusieurs couches et permettent au réseau d'apprendre le mappage résiduel entre l'entrée et la sortie d'une couche. Cela lui permet de former des réseaux très profonds, avec jusqu'à 152 couches, sans souffrir du problème de gradient de fuite qui peut survenir dans les réseaux très profonds [16].

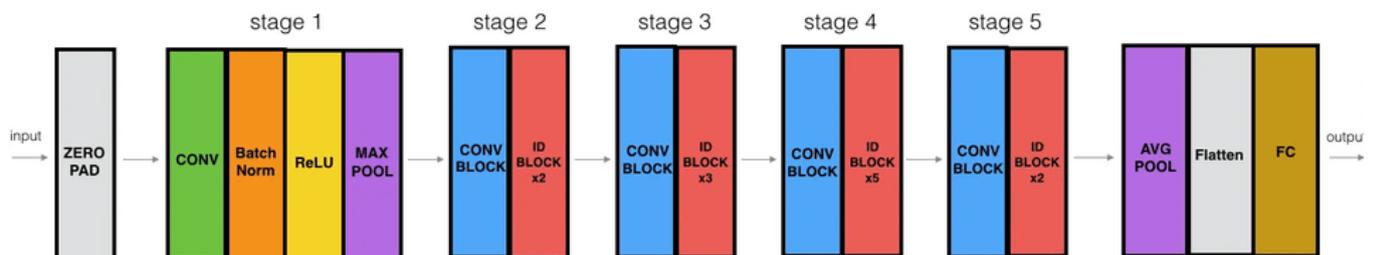


Figure II. 23. Architecture de ResNet.

20 Conclusion

Dans ce chapitre, on a présenté les bases du réseau de neurones Convolutifs CNN, commençant par son architecture fondamentale qui commence par la convolution en passant par ses différents types de couches et leurs fonctionnements, après on a parlé des méthodes pour améliorer l'apprentissage du CNN en utilisant la normalisation, la régularisation, et l'optimisation. Ensuite, on a présenté les architectures de CNN les plus connues et utilisées pour arriver à la meilleure précision possible lors de l'apprentissage d'un modèle.

CHAPITRE III : Identification et classification des défauts constatés sur les roues de pompes en acier.

1 Introduction

Dans ce chapitre, Nous allons illustrer et mettre en place l'architecture de deux modèles CNN, celui que nous avons élaboré et l'autre que nous avons adapté au moyen de l'apprentissage par transfert pour la classification d'images numériques, dans notre cas les images des roues des pompes en acier. Nous chercherons à construire un modèle capable d'extraire automatiquement les caractéristiques les plus pertinentes à partir des images brutes et de réaliser des prédictions précises.

2 Configuration du matériel utilisée :

Tableau III. 1. Caractéristiques techniques du PC.

CPU	Apple M2 Pro chip
La RAM	08 GB DDR4
GPU	Apple M2 pro graphique chip

3 Outils et bibliothèques utiles

3.1 Python

Python est un langage de programmation informatique, Contrairement à HTML, CSS ou JavaScript, son usage n'est donc pas limité au développement web. Il peut être utilisé pour tout type de programmation et de développement logiciel. Il est spécialement utilisé pour le développement des logiciels et des programmes pour PC, ainsi que dans le développement d'applications en ligne et mobiles. Il peut écrire des instructions de système informatique. Et dans les domaines de l'apprentissage automatique et l'apprentissage profond Python est le plus utilisé grâce à ces avantages tel que la facilité d'exécution et ça large gamme des bibliothèques (Numpy, Pandas, ScikitLearn, SciPy), ça capacité de traiter des larges données. C'est souvent le langage préféré pour la science des données [11].

3.2 Colaboratory

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

de note-book Jupyter qui ne nécessite aucune configuration et permet d'accéder sans frais à des ressources informatiques, dont des GPU [11].

3.3 Les bibliothèques

Dans le domaine de Data-science, une bibliothèque (Library) un ensemble de fonctions, de classes d'objets et de constantes qui permettent de travailler sur un thème particulier. Si ces modules appartiennent au pack originel du langage, ce sont des modules intégrés. Dans notre projet on a utilisé nombreux des bibliothèques comme suit, [11]:

3.3.1 Matplotlib :

Est un module de dataviz très agréable ça fonctionnement est de créer des graphes qui modélisent les données sur lesquelles tu travailles. Matplotlib s'utilise en synergie avec Numpy ou Pandas. Matplotlib offre une large variété de types de graphes qui s'adaptent à tous les besoins : histogrammes, boîtes à moustache, courbes, scatter plots, camemberts [11].

3.3.2 TensorFlow

TensorFlow Le modèle d'apprentissage profond de Google et disponible en open source. Il s'agit d'une bibliothèque calcul numérique et de Machine Learning compatible avec le langage Python. Il optimise les opérations de collecte de données, de développement de modèles d'apprentissage automatique, d'élaboration de prévisions et d'amélioration des résultats à l'avenir [11].

3.3.3 Keras

Keras est une bibliothèque Open Source de réseau neuronal écrite en Python. Fonctionnant à partir de structures telles que TensorFlow et Theano, Créée par François Chollet, développeur chez Google, Keras présente l'avantage d'être rapide, simple d'utilisation et modulaire. Keras est l'une des bibliothèques les plus populaires à l'heure actuelle dans le domaine de Deep Learning. Les conceptions complexes et la création de couches de réseaux neuronaux sont rendues incroyablement simples [11].

4 Base de données

La base de données comprend une série de photos de roues d'une pompe submersible, toutes en vue de dessus. Au total, l'ensemble de données contient 7348 images. Celles-ci ont toutes une taille de (300*300) pixels sur fond gris. Ce jeu de données est divisé en trois groupes, à savoir le premier contenant les images d'entraînement avec 5307 photos, le deuxième avec un

nombre de 1326 images de validation tout comme le troisième groupe de test avec 715 images. Tout compte fait, les images sont classées en deux catégories :

- Roues-ok (figure III. 1) (3137 images)
- Roues-def (figure III. 2) (4211 images)

Il convient enfin de noter que toutes ces données ont été fournies par PILOT TECHNOCAST, Shapar, Rajkot.

- Roues non défectueuses.

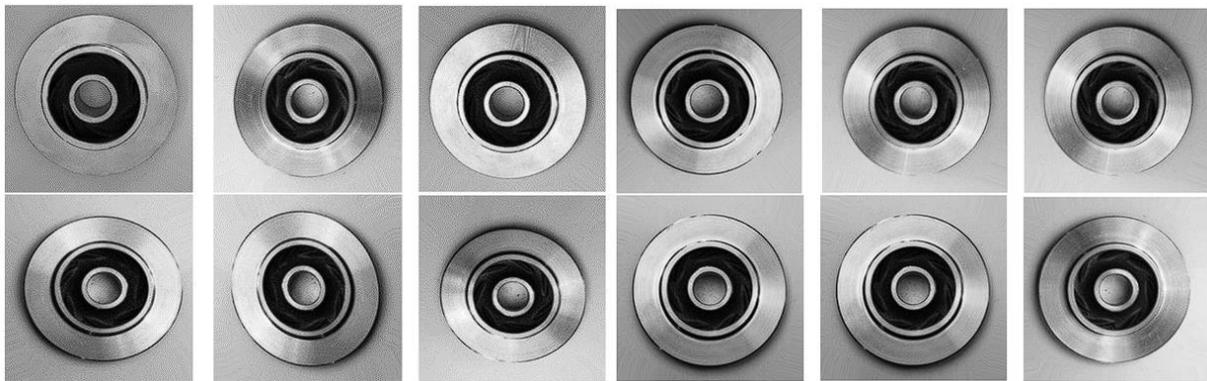


Figure III. 1. Roues non défectueuses.

- Roues défectueuses

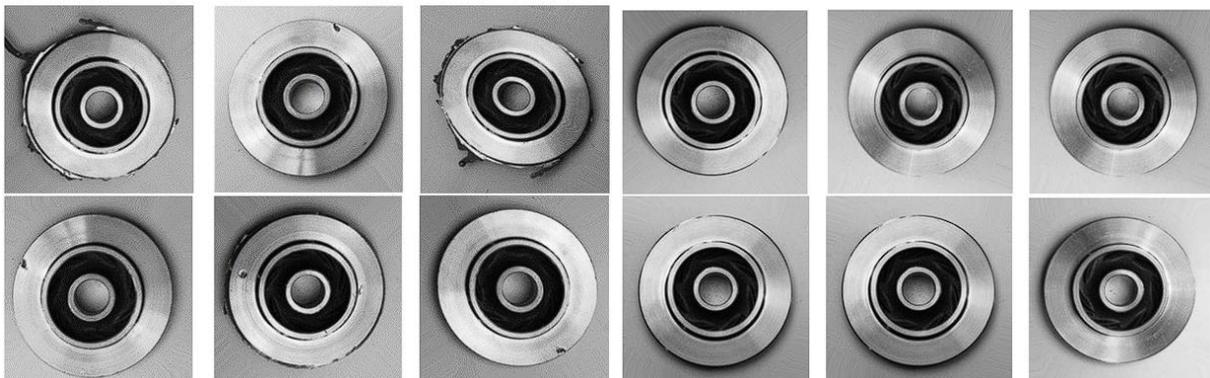


Figure III. 2. Roues défectueuses.

5 Installation de détection des défauts

La prise de ces images exige un éclairage stable, au moyen d'appareils spéciaux. De même, pour pouvoir identifier les défauts avec précision, des caméras à haute résolution dotées d'une vitesse de prise de vue supérieure deviennent indispensables comme on peut voir sur la

figure III. 3. Pour améliorer les performances du système, nous devons également prêter attention à d'autres facteurs tels que la taille de la lentille, la focalisation et son orientation.

Préparation du Système :

- 1) Sélection d'une caméra.
- 2) Installation de l'éclairage
- 3) Acquisition des images
- 4) Traitement et analyse des images :

Appliquer des algorithmes de détection de formes pour repérer les défauts. Cela peut inclure la détection de variations de texture, de couleur ou de forme.

- 5) Validation et rapports
- 6) Actions correctives :

Élimination des produits défectueux.

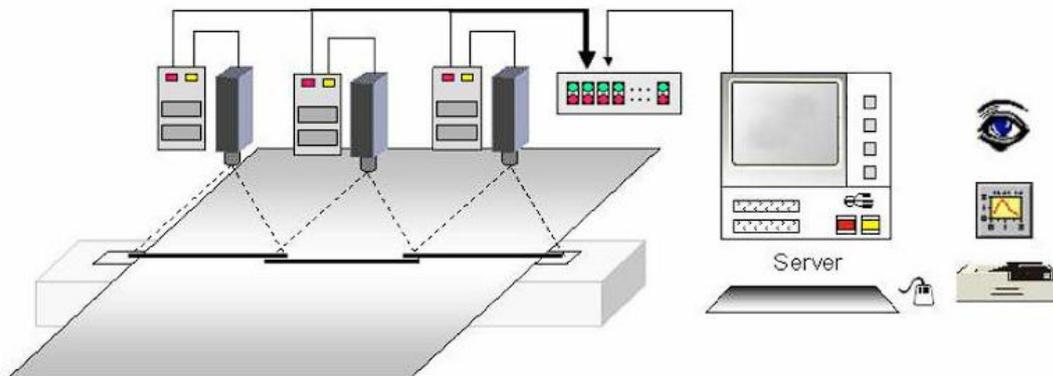


Figure III. 3. Installation de détection des défauts.

6 Architectures des modèles utilisés

Pour ce faire, nous avons utilisé deux modèles, le premier étant un modèle CNN simple et le deuxième du type VGG16 pré-entraîné on utilisant le transfert d'apprentissage.

6.1 Architecture du modèle CNN simple

Ce modèle est composé de deux couches de convolution et deux couches de maxpooling,

une couche de flattening et 2 couches dense.

Après avoir importé les bibliothèques dont nous avons besoin pour notre modèle, on a procédé à un import de nos données. Celles-ci ont été divisées en 3 catégories, avant d'être introduites dans les couches de convolution. L'image passe d'abord par la première couche de convolution, l'image en entrée est de taille 300*300. Cette couche est composée de 32 filtres de taille 3*3, avec un stride de 2. Chacune de nos couches de convolution est suivie d'une fonction d'activation ReLU, Nous appliquons ensuite la méthode Maxpooling pour réduire la taille de l'image et garder les paramètres de calcul les plus importants. Après cette convolution des nouveaux feature maps seront créés. Les feature maps qui sont obtenus auparavant elles sont introduites en entrée de la deuxième couche de convolution qui est composée de 64 filtres, une fonction d'activation RELU est appliquée sur la couche de convolution, ensuite on applique Maxpooling autrefois. À la sortie de cette couche, nous aurons 64 feature maps de taille 18*18 (figure III. 3).

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 149, 149, 32)	320
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 36, 36, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 18, 18, 64)	0
flatten (Flatten)	(None, 20736)	0
dense (Dense)	(None, 128)	2654336
dense_1 (Dense)	(None, 1)	129
=====		
Total params: 2673281 (10.20 MB)		
Trainable params: 2673281 (10.20 MB)		
Non-trainable params: 0 (0.00 Byte)		

Figure III. 4. Configuration du modèle CNN simple.

6.1.1 Résultats obtenus du premier modèle

Pour visualiser les résultats obtenus pour Le premier modèle, la figure III.5 illustre les résultats en termes de précision et de perte (loss).

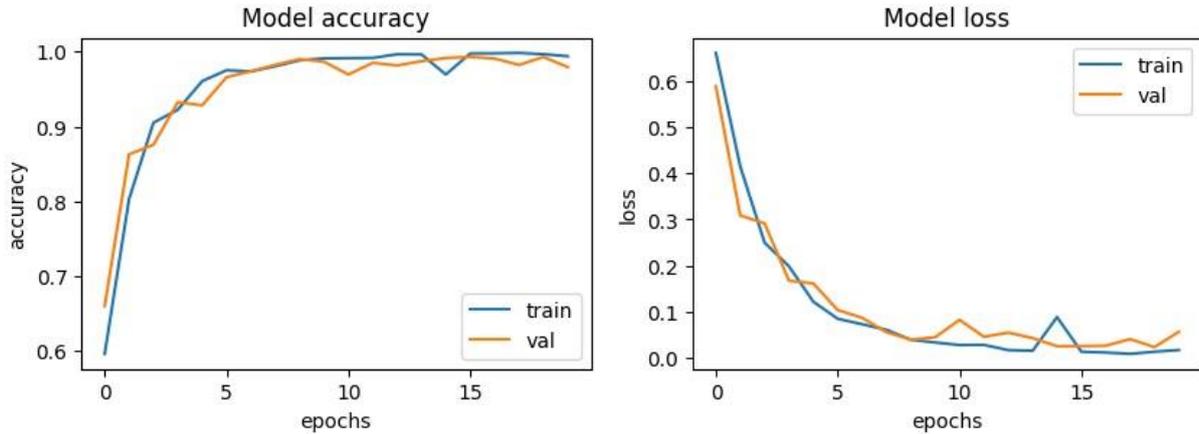


Figure III. 5. Précision et perte du model.

Il ressort de la figure III. 5 que la précision de notre modèle dépend du nombre d'itérations, ce qui signifie qu'à mesure que le nombre d'époques augmente, les informations tirées de nos données deviennent de plus en plus importantes. En même temps, dans la mesure où la perte d'apprentissage varie progressivement en fonction de la précision de l'apprentissage, on observe une baisse de cette perte avec l'augmentation du nombre d'itérations.

6.1.2 Matrice de confusion :

La matrice de confusion permet d'évaluer la performance de notre modèle, puisqu'elle reflète les métriques du Vrai positif, Vrai négatif, Faux positif et Faux négatif. La figure III.6 illustre la position de ces métriques pour chaque classe. On constate aussi que le premier modèle a correctement classé la plupart des images mais il a incorrectement classé juste 13 images.

Actual	Defective	441	12
	OK	1	261
		Defective	OK
		Predicted	

Figure III. 6. Matrice de confusion du model CNN.

6.1.3 Test de prédiction sur les images

La figure III.7 et la figure III.8 montrent les résultats et les probabilités de quelques images de notre base de données. On peut voir que notre modèle a classé correctement la majorité des images avec une prédiction très élevée de 99 % à 100 %, ce qui est un résultat remarquable. On a aussi montré les 13 images que notre modèle a incorrectement classées avec des prédictions.

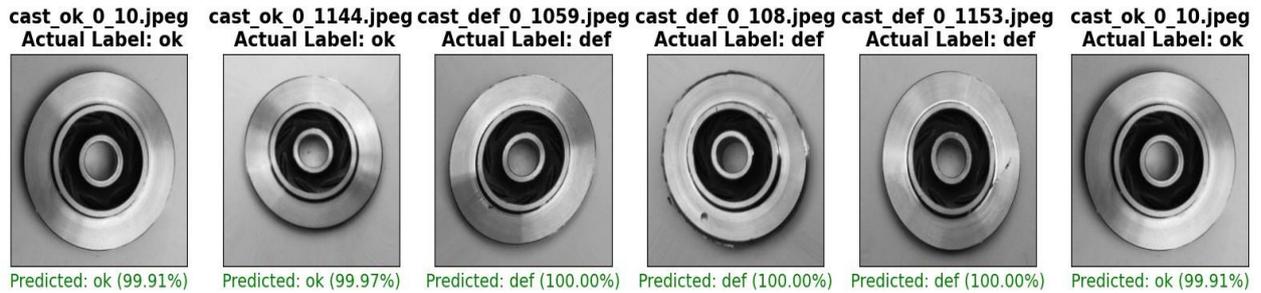


Figure III. 7. Images prédites et leurs libellés.

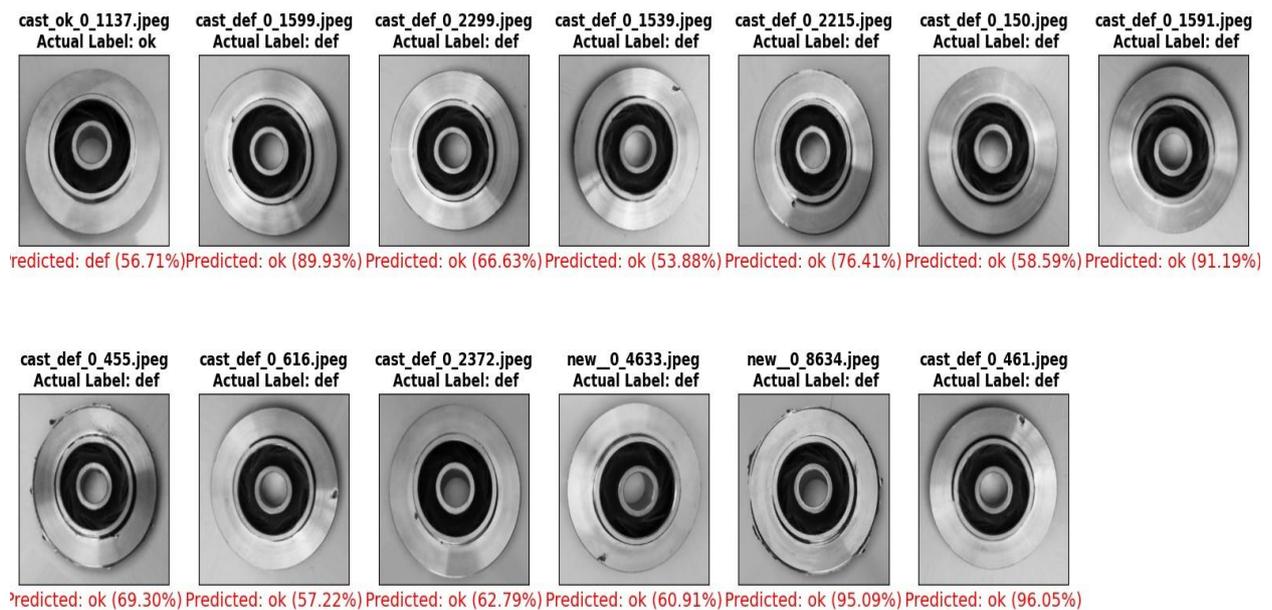


Figure III. 8. Les images incorrectement classés.

6.2 Architecture du model VGG16

Le modèle VGG16 est un réseau de neurones convolutif créé par Karen Simonyan and Andrew Zisserman dans le groupe de géométrie visuelle « Visual Geometry Group » de l'Université d'Oxford. Le modèle VGG examine la profondeur des couches avec un filtre convolutionnel de très petite taille (3×3) pour tenir compte des images à grande dimension.

Les chercheurs ont publié une série de modèles VGG avec différentes longueurs de couches, de 11 à 19.

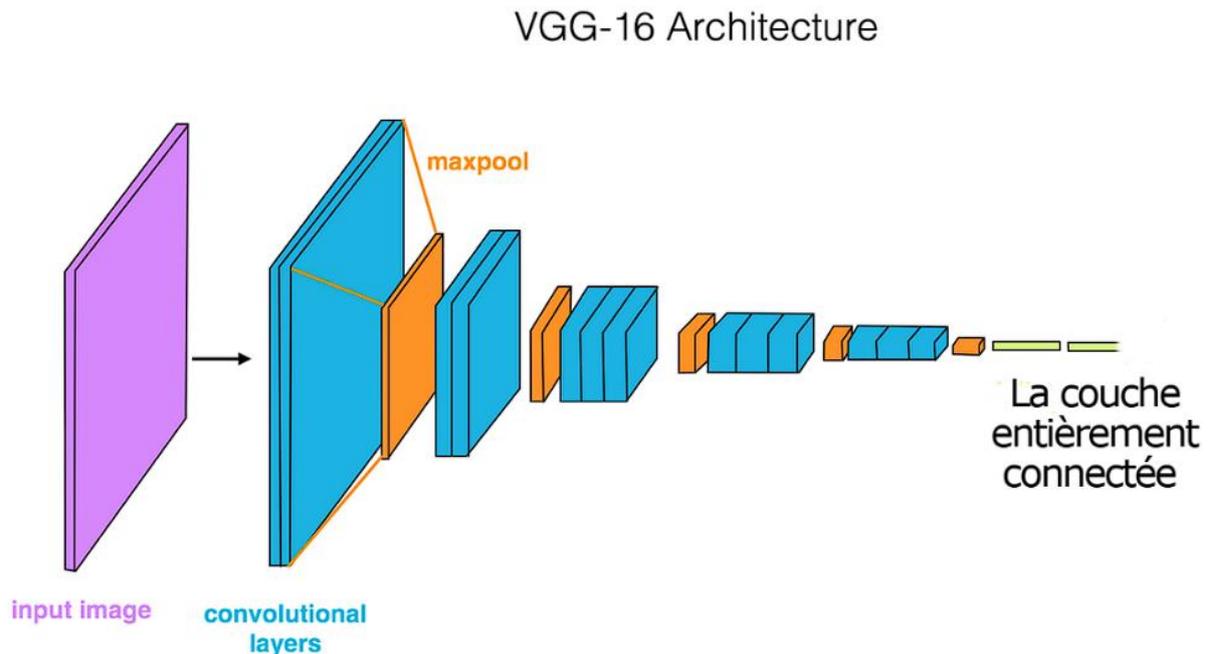


Figure III. 9. Architecture d'un model VGG16.

Un model VGG-16 se compose en général de 13 couches convolutives et 5 couches de maxpooling et 2 couches entièrement connectées comme la figure III.10. Ces couches sont réparties en blocs, chaque bloc contenant plusieurs couches convolutives avec une fonction d'activation ReLu suivies d'une couche de max-pooling.

Tout d'abord, nous avons une couche d'entrée, dont les dimensions d'entrée doivent être (224.224.3), puis deux couches convolutives consécutives avec 64 filtres chacune et une taille de filtre de 3×3 avec un «padding » pour maintenir les dimensions et ReLu comme une fonction d'activation, puis une couche de max-pooling avec une taille de mise en commun de 2×2 et un pas de 2. Ensuite, deux couches convolutives consécutives avec 128 filtres chacune et une taille de filtre de 3×3 avec un «padding », après, une couche de Max-pooling avec une taille de mise en commun de 2×2 et un pas de 2. Par la suite, on ajoute trois couches convolutives avec 256 filtres de taille 3×3 puis la couche de max-pooling avec une taille de mise en commun de 2×2 et un pas de 2. En suit, trois autres couches convolutives avec 512 filtres de taille 3×3 puis la couche de max-pooling avec une taille de mise en commun de 2×2 , on ajoute le dernier bloc qu'il est totalement comme le précédant. A la fin, flatten du «feature map » de sortie ($7 \times 7 \times 512$) en un vecteur de taille 25088.

Dans la couche entièrement connectée on trouve deux couches entièrement connectées :

La première couche a 512 neurones où la fonction d'activation utilisée est le ReLU.

La deuxième couche est un sigmoïde qui permet de calculer la distribution de probabilité des 2 classes.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_4 (Flatten)	(None, 25088)	0
dense_8 (Dense)	(None, 512)	12845568
dense_9 (Dense)	(None, 1)	513
=====		
Total params: 27560769 (105.14 MB)		
Trainable params: 12846081 (49.00 MB)		
Non-trainable params: 14714688 (56.13 MB)		

Figure III. 10. Configuration du modèle VGG16.

6.2.1 Résultats obtenus et Interprétations

On constate dans figure III.11 que la précision de l'apprentissage et de la validation augmente en fonction du nombre d'époques, reflétant ainsi le fait que le modèle recueille de plus en plus d'informations à chaque époque. En cas de diminution de la précision, nous aurons

besoin de plus d'informations pour que notre modèle apprenne, et nous devons donc augmenter le nombre d'époques, et vice versa. Par ailleurs, on remarque que la perte d'apprentissage et de validation décroît avec le nombre d'époques.

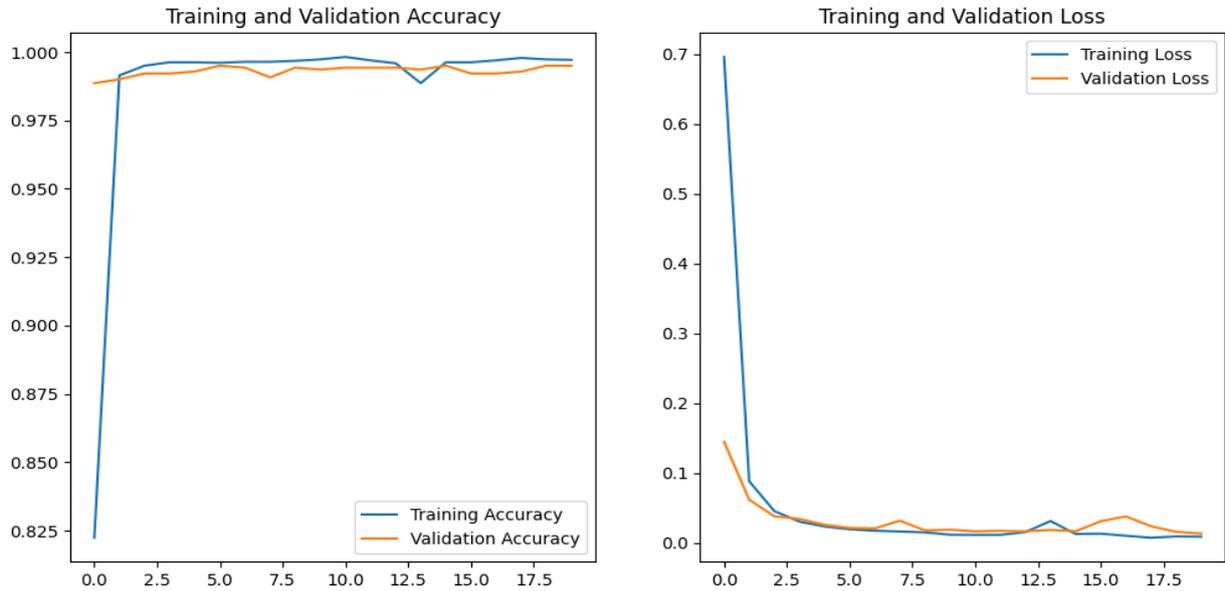


Figure III. 11. Précision et perte du modèle VGG16

La figure III.11 illustre la position de ces métriques pour chaque classe. On constate aussi que le premier modèle a correctement classé la plupart des images mais il a incorrectement classé juste 5 images.

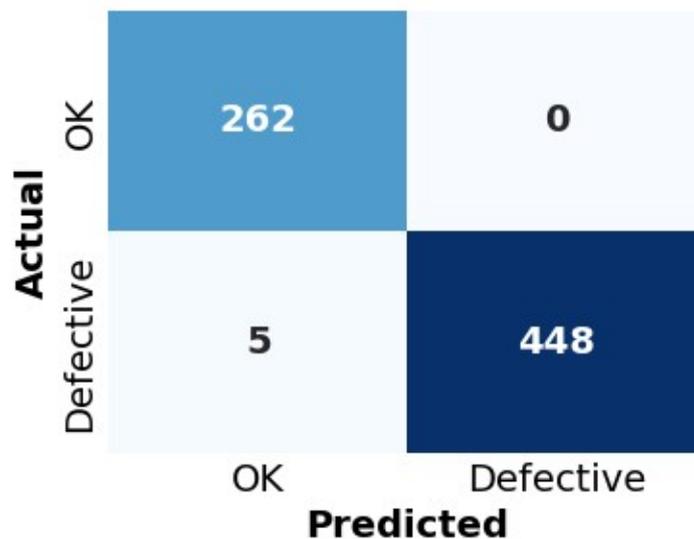


Figure III. 12. Matrice de confusion du modèle VGG16.

7 Méthode Grad-Cam

Cette technique de visualisation permet de montrer la manière dont un réseau de neurones convolutifs est adapté pour prendre une décision en matière de classification [17].

7.1 Principe de fonctionnement

La technique Grad-Cam vise à repérer les parties de l'image qui ont permis à un réseau de neurones convolutifs d'aboutir à la décision finale. Elle consiste à produire des cartes thermiques représentant des classes d'activation sur les images accueillies en entrée. On associe à chaque classe d'activation une classe de sortie spécifique. Ces classes sont utilisées pour marquer l'importance de chaque pixel par rapport à la classe concernée, en augmentant ou en diminuant l'intensité du pixel [17].

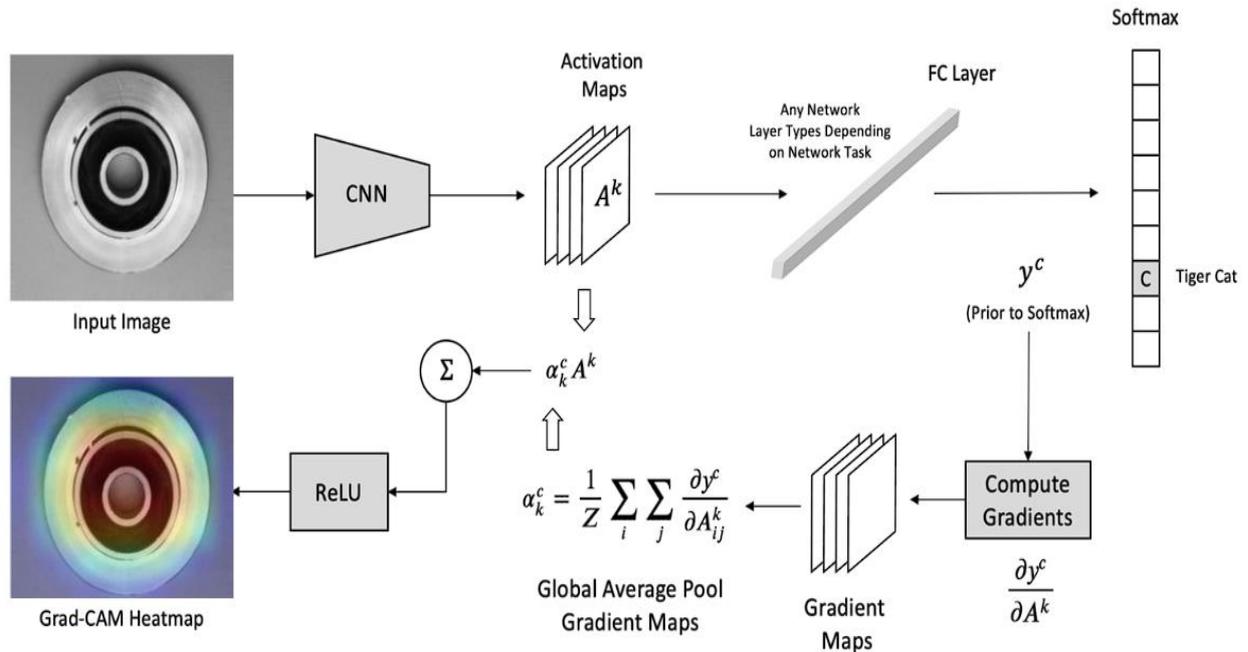
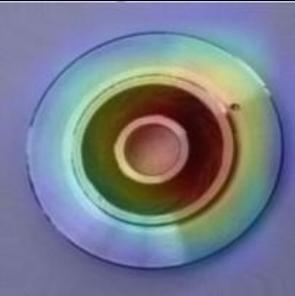


Figure III. 13. Principe de fonctionnement du Grad-Cam.

Les résultats de l'utilisation de la méthode Grad-Cam sur le deuxième modèle VGG16 ont été très satisfaisants (Tableau III. 2). Nous pouvons voir que notre algorithme Grad-Cam intensifie la carte thermique sur les défauts des roues. Dès lors, nous pouvons affirmer que notre algorithme CNN a bien reconnu les défauts de façon à faire la distinction entre les roues en bon état et celles présentant des défauts.

Tableau III. 2. Visualisations Grad-Cam du model VGG16.

Image d'entrée	La vision du Grad-Cam
	
	
	

8 Comparaison des résultats

Au vu du tableau ci-dessous, les différents résultats obtenus pour les deux modèles sont les suivants :

Tableau III. 3. Résultats comparées.

	Architecture Utilisé			Nombre d'époques	Précision obtenu Sur la base de test (%)	Perte (%)	Temps d'exécution (second)
	Couche de Convolution	Couche de Pooling	La couche entièrement connectée				
CNN	02	02	02	25	98.20	01.80	367
VGG16	13	05	02	20	99.30	00.70	1695

Sur le tableau III.1 figurent l'architecture utilisée dans chaque modèle, le nombre d'époques, la précision et la perte, tout comme le temps d'exécution. Dans notre cas, le temps d'exécution est très réduit grâce à Google colab pro, qui nous a permis d'utiliser un GPU très puissant (T4 GPU). En constate que les deux modèle on fournit des résultats relativement similaire présentait les meilleurs résultats en raison du nombre de couches de convolution, mais le temps d'exécution était relativement important.

9 Conclusion

Ce chapitre a présenté l'implémentation et l'évaluation de deux modèles CNN pour la classification des roues de pompes submersibles, le premier modèle et le deuxième modèle VGG16, ainsi que l'utilisation de la méthode Grad-CAM pour VGG16. Le premier modèle utilisé est un CNN simple, il a atteint une précision de classification de 98.20 %, et une perte de 01.80%, un résultat qui prouve que notre modèle a une grande capacité de détecter les défauts dans les images des roues de pompes avec précision. Le deuxième modèle, VGG16, doté d'un transfert d'apprentissage a aussi montré de sa part des résultats remarquables pour la détection des défauts avec une précision de 99.3%, et une perte de 0.7%. Ces résultats prometteurs mettent en évidence le potentiel des approches basées sur les réseaux de neurones convolutifs pour la surveillance et le contrôle de la qualité des pièces fabriquées.

CONCLUSION GENERALE

Conclusion générale

Les roues des pompes submersibles sont la partie la plus importante mais aussi la plus complexe à fabriquer. Leur fabrication repose en effet sur le moulage du métal, qui est un procédé très délicat. Cette particularité peut donner lieu à de nombreux types de défauts, de sorte que la qualité s'en trouve dégradée.

Souhaitant impérativement limiter les pertes et les coûts, nous avons appliqué aux roues des pompes submersibles deux algorithmes de classification à base de réseaux de neurones convolutifs. Un premier modèle de classification d'images CNN simple nous a permis d'obtenir des résultats satisfaisants, avec une précision d'environ 98,20 % pour un taux de perte de 01.80 %.

Par la suite, le modèle pré-entraînée VGG16 que nous avons appliqué a atteint une précision de 99,30 % et un taux de perte de 0,70%. Au bout du compte, la comparaison des deux modèles a révélé que le modèle VGG16 avait une profondeur et une complexité accrues, et qu'il donnait donc des meilleurs résultats en terme de précision.

Par ailleurs, une application de la méthode GRAD-CAM a été faite pour mieux comprendre comment le deuxième modèle CNN VGG16 prend une certaine décision afin de faire sa prédiction. Les résultats obtenus étaient satisfaisants et nous ont montré les parties de l'image qui ont conduit le modèle à sa décision finale.

De ce fait, le recours aux réseaux de neurones convolutifs qui permettent de classer les roues de pompes submergées au cours du processus de fabrication par moulage est une avancée significative vers l'amélioration de la qualité des produits. En exploitant au mieux les capacités des modèles CNN, les industriels peuvent garantir une qualité et une fiabilité optimales, diminuer les coûts de fabrication et satisfaire davantage les clients.

Comme perspective, le travail réalisé peut être amélioré en ajoutant des modèles de détection comme YOLO qui est un outil très efficace pour détecter des objets.

On recommande d'intégrer les deux modèles que nous avons abordé dans ce mémoire dans le processus de fabrication par moulage pour contribuer à l'adoption de cette technologie dans le secteur.

Références

- [1] PENG, Long, et al. Predictive approach to perform fault detection in electrical submersible pump systems. *ACS omega*, 2021, 6.12: 8104-8111.
- [2] Souetre, M. année universitaire 2022/2023. Simulation numérique de la fonderie d'aciers: détermination et optimisation des paramètres de calage pour les aciers faiblement alliés coulés en moule sable à vert. Doctoral dissertation, Université HESAM.
- [3] Article «17 types de défauts de moulage : comprendre leurs causes et leurs remèdes»
Site web : 17 types de défauts de moulage : comprendre leurs causes et leurs remèdes (rapiddirect.com)
- [4] EL NAQA, Issam; MURPHY, Martin J. *What is machine learning?* Springer International Publishing, 2015.
- [5] BI, Qifang, et al. What is machine learning? A primer for the epidemiologist. *American journal of epidemiology*, 2019, 188.12: 2222-2239.
- [6] SCHMIDHUBER, Jürgen. Deep learning. *Scholarpedia*, 2015, 10.11: 32832.
- [7] PAN, Sinno Jialin. Transfer Learning. *Data Classification: Algorithms and Applications*, 2014, 21: 537-570.
- [8] bWU, Jianxin. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 2017, 5.23: 495.
- [9] MAZIR Melissa AMRIOU Hanane année universitaire 2021/2022 .Conventionnel neural network pour la détection port du masque. Université abderrahmane mira bejaia
- [10] Article «Fonctions d'activation et réseaux de neurones. » Picsellia. (1er septembre 2023).
Site web : Fonctions d'activation et réseaux de neurones — Picsellia
- [11] Mr.Mokri Mohammed Zakaria Hanane année universitaire 2016/2017. Classification des images avec les réseaux de neurones convolutionnels, Université Abou Bakr Belkaid Tlemcen.
- [12] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8, 1-74.
- [13] Article «Regularization in Neural Networks». Bala Priya C. (Jun 30, 2023)
Site web: Regularization in Neural Networks | Pinecone

[14] Article « Various Optimization Algorithms For Training Neural Network». Sanket Doshi. (Jan 13, 2019). Site web: Various Optimization Algorithms for Training Neural Network | by Sanket Doshi | Towards Data Science

[15] HAMLAOUI Mohammed Yazid année universitaire 2020/2021. Classification du COVID'19 en utilisant CNN dans les images X-ray. Université de 8 Mai 1945 Guelma.

[16] Ahmed HIMEUR Rabah IDOUI année universitaire 2019/2020. Reconnaissance d'images par les réseaux de neurones convolutionnels (CNN). Université Mohammed Seddik BenYahia Jijel.

[17] Raphael Kassel. Qu'est-ce que la méthode Grad-CAM ? 2021.

Site web : Qu'est-ce que la méthode Grad-CAM ? (datascientest.com)