

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERRI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D' INFORMATIQUE
DEPARTEMENT D' AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Automatique

Spécialité : Automatique et Informatique
Industrielle

Présenté par

MADJENE Tassadit

Thème

Détection Des Défauts De Surface À Base Des Réseaux De Neurones Convolutifs

Mémoire soutenu publiquement le 03 / 10 / 2024 devant le jury composé de :

M DORBANE.F

Grade, Lieu d'exercice, Président

M HAMMOUCHE.K

Grade, Lieu d'exercice, Encadrant

M BEDOUHENE.S

Grade, Lieu d'exercice, Examineur

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERRI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D' INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Automatique

Spécialité : Automatique et Systèmes

Présenté par

IZERKHEF Sabrina

Thème

Détection Des Défauts De Surface À Base Des Réseaux De Neurones Convolutifs

Mémoire soutenu publiquement le 03 / 10 / 2024 devant le jury composé de :

M DORBANE. F

Grade, Lieu d'exercice, Président

M HAMMOUCHE.K

Grade, Lieu d'exercice, Encadrant

M BEDOUHENE.S

Grade, Lieu d'exercice, Examineur

Remerciements

Remerciements

Avant tout, je rends grâce à Allah, le Tout-Puissant, pour m'avoir donné la force, la patience et la persévérance nécessaires à la réalisation de ce mémoire. Sans Sa bénédiction, rien de cela n'aurait été possible.

Je remercie chaleureusement mon encadrant, HAMMOUCHE Kamal, pour ses conseils avisés et son accompagnement tout au long de cette aventure académique. Votre expertise a enrichi ce projet, et votre patience m'a permise de progresser avec confiance.

Un remerciement spécial à mes parents, pour leur amour inébranlable et leur soutien constant. Vous êtes ma plus grande source d'inspiration.

À ma grande sœur Liza, pour sa complicité et son écoute bienveillante, et à ma meilleure amie Lydia, pour les moments partagés et les précieux conseils.

À ma binôme de mémoire, Sabrina, qui a partagé avec moi chaque moment de doute et de joie. Merci pour ta joie de vivre et ces moments mémorables. Ce mémoire est notre bébé à toutes les deux.

Enfin, à une personne très spéciale, Didiw, pour tes mots doux, ta présence constante et tes encouragements qui m'ont accompagnée tout au long de ce parcours. Ton soutien a été une véritable source de motivation et de lumière dans les moments les plus difficiles. C'est en grande partie grâce à toi que j'ai pu aller jusqu'au bout.

M^{lle} MADJENE Tassadit

Remerciements

Je commence par exprimer ma profonde gratitude envers Allah pour m'avoir guidé et soutenu tout au long de cette aventure.

Je tiens à remercier mon promoteur, Mr Hammouche Kamal, pour son accompagnement, ses conseils et sa confiance. Votre générosité et votre bienveillance ont eu un impact significatif sur ma trajectoire professionnelle, et je suis honorée de pouvoir compter sur votre soutien.

À mes plus grands soutiens et sources d'inspiration, je dédie ce travail avec tout mon amour.

À ma mère, merci pour ton amour inconditionnel et ton dévouement. Tu as été ma lumière, toujours là pour me soutenir même dans le doute.

À mon père, je suis reconnaissante pour tes conseils et pour m'avoir appris l'importance du travail acharné. Tu es un modèle de réussite et m'inspires à poursuivre mes rêves.

À ma sœur Louiza et mes frères Mohand et Yacine, merci pour votre soutien et votre humour. Vous êtes ma source de joie. Je souhaite également bon courage à Yacine pour son examen de baccalauréat.

À ma binôme, Tassadit, merci pour notre collaboration enrichissante. Ta motivation et ton inspiration ont été précieuses tout au long de ce parcours.

Enfin, à ma meilleure amie Mitchi, qui a été ma bouée de sauvetage dans les moments difficiles et ma compagne de joie dans les victoires. Ton amitié est un trésor que je chérirai toujours.

M^{lle} IZERKHEF Sabrina

Résumé

Résumé

Ce mémoire porte sur la détection et la classification des défauts de surface dans plusieurs contextes industriels en utilisant des réseaux de neurones convolutifs (CNNs). Deux approches complémentaires ont été abordées. La première est la détection des défauts par classification à une classe où le modèle Fully Convolutional Data Description (FCDD), basé sur VGG16, a été employé. Cette méthode permet de détecter la présence ou l'absence d'une anomalie sans préciser sa nature, en se focalisant uniquement sur la distinction entre les images normales et celles présentant des défauts. La seconde approche concerne la reconnaissance des défauts par classification multiclassées, pour laquelle le modèle GoogLeNet a été utilisé. Cette méthode permet d'identifier précisément le type du défaut. Pour cela, nous avons exploité trois bases de données spécifiques à des secteurs variés : l'industrie pharmaceutique (Pill QC), l'agroalimentaire (Hazelnut), et l'industrie du bois (Wood). Les résultats obtenus avec les deux approches sont prometteurs. FCDD atteint près de 90 % de détection, tandis que GoogLeNet offre un taux de classification dépassant 96 %. Ces résultats soulignent l'importance des CNNs dans l'amélioration des processus de contrôle qualité, contribuant à une meilleure productivité et à une réduction des erreurs humaines.

Table des matières

Introduction Générale	1
------------------------------------	----------

Chapitre 1: Généralités Sur La Détection Des Défauts De Surface Basée Sur La Vision Artificielle

1	Introduction	4
2	Définition et classification des défauts de surface	4
2.1	Définition des défauts de surface	4
2.2	Détection des défauts de surface	5
2.3	Classification des défauts de surface	5
2.3.1	Origine	5
2.3.2	Nature	6
2.3.3	Taille	6
2.3.4	Impact sur la qualité	7
2.4	Domaine d'application	7
2.4.1	Industrie métallurgique	7
2.4.2	Industrie alimentaire	8
2.4.3	Industrie pharmaceutique	8
2.4.4	Industrie du verre	9
2.4.5	Industrie du plastique	9
2.4.6	Industrie du papier et de l'impression	10
2.4.7	Industrie du textile	10
2.4.8	Industrie automobile	11
2.4.9	Industrie électronique	11
2.5	Intérêts et avantages de la détection des défauts de surface	12
3	Méthodes traditionnelles de détection des défauts de surface	13
3.1	Inspection visuelle humaine	13
3.2	Techniques d'imagerie conventionnelles	13
3.2.1	Photographie	13

3.2.2	Microscopie optique	14
3.2.3	Caméras industrielles	14
3.2.4	Caméras thermiques	14
3.2.5	Caméras à ultrasons	14
3.3	Utilisation de méthodes de contrôle non destructif	14
3.3.1	Contrôle par ressuage (ou pénétrant)	15
3.3.2	Contrôle par magnétoscopie	15
3.3.3	Contrôle par courant de Foucault	16
3.3.4	Contrôle par radiographie (rayons X et gammagraphie)	17
3.3.5	Contrôle par ultrasons (US)	17
3.4	Limitations des méthodes traditionnelles	18
4	Détection automatisée des défauts de surface	19
5	Système de vision artificielle de détection des défauts de surface	19
5.1	Acquisition des données visuelles	20
5.1.1	Types d'éclairage	20
5.2	Prétraitement	24
5.2.1	Filtrage	25
5.2.2	Rehaussements de contraste (Egalisation d'histogramme)	30
5.3	Segmentation	32
5.4	Post traitement	38
5.4.1	Morphologie Mathématique	38
5.5	Extraction de caractéristiques	40
5.5.1	Caractéristiques basées sur la texture	40
5.5.2	Caractéristiques basées sur la forme	42
5.5.3	Caractéristiques basées sur la couleur :	43
5.6	Classification (décision) :	43
5.6.1	Machines à Vecteurs de Support (SVM)	43
5.6.2	Arbres de Décision et leurs Ensembles (Random Forest, Gradient Boosting)	44
5.6.3	Réseaux de neurones convolutifs (CNNs)	45
6	Avantages de l'utilisation de systèmes de vision artificiel pour la détection automatisée	45
7	Conclusion	46

Chapitre 2: Réseaux De Neurones Convolutifs (CNNs)

1	Introduction	48
2	Neurones biologiques/Neurones artificiels	49
2.1	Description d'un Neurone biologique	49

2.2	Neurones artificiels	50
3	Architecture des réseaux de neurones	55
3.1	Réseaux de neurones statiques (acycliques, non bouclés)	57
3.2	Réseaux de neurones dynamiques (ou récurrents, ou bouclés)	57
4	Apprentissage des réseaux de neurones	57
4.1	Apprentissage supervisé	58
4.2	Apprentissage non supervisé	58
4.3	Apprentissage par renforcement	58
4.4	Apprentissage semi-supervisé	59
4.5	Apprentissage auto-supervisé	59
4.6	Rétropropagation de l'erreur	59
5	Réseaux de Neurones Convolutifs (CNNs)	60
6	Architecture générale d'un CNN	61
6.1	Couche d'entrée	62
6.2	Couches de convolution	63
6.3	Couche d'activation	67
6.4	Couche de pooling	67
6.5	Couche Flatten	69
6.6	Couche entièrement connectée (FC)	69
6.7	Fonction d'activation Softmax	70
6.8	Couche de perte (Loss)	70
6.8.1	Fonction de perte :	70
6.8.2	Règle de la chaine	72
6.9	Techniques d'optimisation	72
6.9.1	Descente de Gradient (Gradient Descent)	72
6.9.2	Descente de Gradient par Mini-Lots (Mini-Batch Gradient Descent)	73
6.9.3	Descente de Gradient Stochastique (SGD)	73
6.9.4	Momentum	73
6.9.5	RMSProp (Root Mean Square Propagation)	73
6.9.6	Adam (Adaptive Moment Estimation)	74
7	Exemples de CNNs	74
8	Utilisation et entraînement des réseaux de neurones convolutifs	76
8.1	Prétraitement des données	76
8.2	Choix du modèle CNN	76
8.2.1	Création d'un nouveau modèle CNN	76
8.2.2	Transfert d'apprentissage	77
8.2.3	Transfert d'apprentissage avec réglage fin (fine-tuning)	77
8.3	Paramètres d'entraînement	77
8.4	Surapprentissage	78
8.4.1	Régularisation :	78

8.4.2	Dropout :	79
8.4.3	Arrêt anticipé (early stopping) :	79
9	Défis et limitations des réseaux de neurones convolutifs	79
10	Conclusion	80

Chapitre 3: Détection Des Défauts De Surface À L'aide Des CNNs

1	Introduction	82
2	Bases d'images traitées	82
2.1	Base Pill QC	82
2.2	Base Hazelnut	83
2.3	Base Wood	85
3	Approches utilisées	87
3.1	Classification à une classe	87
3.1.1	Architecture du FCDD	87
3.1.2	Fonctionnement du FCDD	90
3.1.3	Test et résultats	91
3.1.3.1	Répartition des Données	91
3.1.3.2	Métriques de Performance :	93
3.1.3.3	Résultats	94
3.2	Classification Multiclasses	99
3.2.1	Choix du réseau de neurones convolutif	99
3.2.2	Résultats de classification	100
4	Conclusion	104

Conclusion générale	107
--------------------------------------	------------

Bibliographie	109
--------------------------------	------------

Liste des tableaux

1.1	Comparaison des résultats entre le filtre de lissage et le filtre gaussien sur une image	29
2.1	Fonctions d'activation dans les réseaux de neurones avec leurs dérivées	53
2.2	Comparaison entre les neurones biologiques et artificiels	55
2.3	Comparaison entre les réseaux de neurones classiques et les réseaux de neurones convolutifs	61
3.1	Répartition des données pour l'évaluation du modèle FCDD	92
3.2	Matrice de confusion	93
3.3	Métriques de performance pour les trois bases de données	97
3.4	Répartition des données pour l'évaluation du modèle de la classification	101

Table des figures

1.1	Types de défauts de l'origine	5
1.2	Défauts liés à la nature	6
1.3	Défauts selon la taille	6
1.4	Défauts impactant la qualité	7
1.5	Exemple de défaut de soufflures dans l'industrie métallurgique	7
1.6	Exemple de défaut de production dans l'industrie alimentaire	8
1.7	Exemple de défaut des comprimés médicamenteux	8
1.8	Verre fissuré	9
1.9	Marque d'écoulement	9
1.10	Défaut de papier	10
1.11	Défauts de tâches sur un textile	10
1.12	Défaut de rayure sur un pare-brise d'un véhicule	11
1.13	Carte électronique endommagée	11
1.14	Inspection visuelle	13
1.15	Différentes caméras	14
1.16	Contrôle par ressuage	15
1.17	Contrôle par magnétoscopie	16
1.18	Contrôle par courant de Foucault	16
1.19	Contrôle par radiographie	17
1.20	Contrôle par ultrasons	18
1.21	Schéma d'un système de vision artificielle	19
1.22	Éclairage directionnel	20
1.23	Éclairage brillant	21
1.24	Éclairage diffus	21
1.25	Éclairage coaxial	22
1.26	Éclairage axial diffus	22
1.27	Lumière structurée	23
1.28	Lumière polarisée	23
1.29	Fond noir	24
1.30	Méthodes de prétraitement	25
1.31	Exemple d'une opération de convolution	26

1.32 Exemple de filtres moyeneurs	27
1.33 Application du filtre moyeneur	27
1.34 Exemples de filtres Gaussiens	28
1.35 Application d'un filtre gaussien	28
1.36 Application d'un filtre médian	30
1.37 Égalisation d'histogramme	31
1.38 Exemple de seuillage global appliqué à une image 5x5.	32
1.39 Image binarisé	33
1.40 Segmentation par régions.	34
1.41 Image segmentée par croissance des régions	35
1.42 Détection de contours à partir du gradient de l'image	37
1.43 Detection de contours à laide de l'opérateur Lplacien	37
1.44 Exemples d'éléments structurants.	38
1.45 Exemple d'érosion avec un élément structurant croix 3x3	39
1.46 Exemple de dilatation avec un élément structurant croix 3x3	39
1.47 Application des opérations morphologiques de base	40
1.48 Illustration d'une matrice de cooccurrence	41
1.49 Machines à Vecteurs de Support	44
1.50 Arbres de Décision et leurs Ensembles	44
1.51 Processus de Détection de Défauts de Surface Utilisant des Réseaux de Neu- rones Convolutifs	45
2.1 Neurone biologique	49
2.2 Structure d'un neurone artificiel	51
2.3 Les courbes des fonctions d'activation	54
2.4 Architecture d'un réseau de neurones	56
2.5 Réseau bouclé	57
2.6 Réseau non bouclé	57
2.7 Architecture des CNN	62
2.8 convolution d'une image en niveaux de gris	64
2.9 convolution d'une image en couleur	64
2.10 Illustration de la notion stride dans une couche de convolution	65
2.11 Illustration du zero padding	66
2.12 Illustration de Max et average pooling	68
2.13 Évolution des CNNs	75
3.1 Exemples d'images de la base Pill Qc	83
3.3 Exemples d'images de la base Hazelnut	85
3.5 Exemples d'images de la base Wood	87
3.6 Architecture du VGG16	88

3.7	Architecture du FCDD	88
3.8	Courbe ROC de la base Pill Qc (ligne diagonale rouge en pointillés indique la ligne de hasard)	95
3.9	Courbe ROC de la base Hazelnut (ligne diagonale rouge en pointillés indique la ligne de hasard)	95
3.10	Courbe ROC de la base Wood (ligne diagonale rouge en pointillés indique la ligne de hasard)	95
3.11	Matrice de confusion de la base Pill Qc	96
3.12	Matrice de confusion de la base Hazelnut	96
3.13	Matrice de confusion de la base Wood	97
3.16	Images de test classées correctement comme anomalie avec leurs carte de chaleur	98
3.17	Architecture de GoogleNet	99
3.18	Graphe de couches du modèle après modifications	100
3.19	Progression de l'accuracy et perte durant l'entraînement du CNN pour la base Pill Qc	102
3.20	Progression de l'accuracy et perte durant l'entraînement du CNN pour la base Hazelnut	102
3.21	Progression de l'accuracy et perte durant l'entraînement du CNN pour la base Wood	103
3.22	Matrices de confusion des trois bases tests	104

Introduction Générale

Introduction Générale

Dans le monde de l'industrie, il est très important de garantir la bonne qualité des produits. Il s'agit de la sécurité, du bon fonctionnement et de l'apparence des articles que nous utilisons quotidiennement, garantissant non seulement le respect des normes réglementaires, mais également la confiance des consommateurs. Un élément clé du contrôle qualité consiste à identifier les problèmes de surface, tels que les rayures ou les fissures, qui pourraient affecter la qualité du produit.

Traditionnellement, le contrôle qualité en milieu industriel se base fortement sur des méthodes manuelles, s'appuyant sur l'expérience et l'expertise humaines. Les inspecteurs qualifiés sont chargés d'inspecter visuellement les produits, composants ou matériaux à la recherche d'imperfections, de défauts de fabrication ou d'anomalies. Cette approche manuelle comporte plusieurs inconvénients. Premièrement, cela prend souvent du temps et nécessite une attention particulière à chaque détail. Les inspecteurs doivent inspecter chaque produit, ce qui est particulièrement exigeant dans le cas d'une production de masse. De plus, cela expose le processus au risque d'erreur humaine, car même les experts les plus qualifiés peuvent être sujets à une fatigue visuelle ou à des malentendus ou à des erreurs d'interprétation. En outre, cette approche est parfois limitée par la variabilité liée aux compétences individuelles. Différents inspecteurs peuvent avoir des normes de qualité légèrement différentes, ce qui entraîne des évaluations incohérentes. De plus, la subjectivité de l'évaluation visuelle peut faire en sorte que les résultats d'un inspecteur diffèrent de ceux d'un autre.

Avec les progrès rapides de la technologie, notamment l'émergence de l'intelligence artificielle et de la vision par ordinateur, les entreprises ont commencé à explorer des alternatives plus efficaces. Les défis posés par la complexité croissante des produits et la nécessité de réponses plus rapides conduisent à l'adoption d'approches automatisées. Les systèmes automatisés de contrôle qualité pilotés par des algorithmes d'apprentissage automatique et des réseaux de neurones sont apparus comme des solutions potentielles à ces défis.

L'évolution des processus de contrôle qualité depuis les méthodes manuelles traditionnelles vers des méthodes automatisées plus sophistiquées reflète une réponse nécessaire aux exigences croissantes de l'industrie moderne. Il est conçu pour accroître l'efficacité, réduire les

erreurs humaines et fournir une réponse plus rapide dans l'environnement dynamique de la production industrielle moderne.

C'est dans ce contexte que s'inscrit notre travail. Nous nous sommes principalement intéressé à la détection des défauts de surface à base des réseaux de neurones convolutifs (CNNs). Les CNNs sont des modèles d'apprentissage profond ayant montré une capacité extraordinaire à extraire des informations significatives à partir de données complexes, notamment visuelles. Cette capacité en fait des alliés précieux dans différents domaines et notamment celui de la détection automatisée des défauts, promettant d'augmenter l'efficacité, de réduire les erreurs humaines et de s'adapter à différents environnements. Nous avons scindé notre mémoire en trois chapitres :

Le premier chapitre donne un aperçu sur les différents défauts de surface. Il décrit les différentes méthodes conventionnelles utilisées pour détecter les défauts de surface. L'approche basée sur la vision artificielle est notamment détaillée.

Le deuxième chapitre est consacré aux réseaux de neurones convolutifs (CNNs). L'architecture de ces réseaux de neurones artificiels ainsi que leur fonctionnement sont présentés dans le but de montrer leur capacité d'apprentissage.

Le troisième chapitre est dédié à la détection des défauts de surface à l'aide des CNNs. Il décrit deux modèles de CNNs utilisés pour détecter les défauts de surface sur les produits industriels. L'un possède une architecture entièrement convolutive (FCDD : Fully Convolutional Data Description) permet de détecter la présence ou non d'une anomalie et le second conçu à base du modèle GoogleNet permet de reconnaître le type du défaut. Ces deux méthodes sont testées sur trois bases d'images appartenant à des domaines industriels : pharmaceutique, agroalimentaire et bois.

Chapitre 1 :
Généralités Sur La Détection Des Défauts
De Surface Basée Sur La Vision Artificielle

Chapitre 1

Généralités Sur La Détection Des Défauts De Surface Basée Sur La Vision Artificielle

1 Introduction

En moyenne, le coût d'un produit de mauvaise qualité pour les industries manufacturières est d'environ 20% de ventes totales [1]. L'assurance de la qualité des produits est un élément essentiel pour assurer la satisfaction des clients et demeurer compétitif sur le marché. Au cœur de cette quête de qualité, la détection des défauts de surface se positionne comme un maillon indispensable. Son rôle primordial réside dans son aptitude à repérer les imperfections susceptibles de compromettre la qualité et l'efficacité des produits. En effet, en identifiant ces défauts, qu'ils soient visibles ou non, cette démarche permet de garantir des produits finis exempts de tout problème, assurant ainsi la confiance des clients et la réputation de l'entreprise sur le marché concurrentiel.

Ce chapitre se concentre sur une étude approfondie de la détection des défauts de surface en utilisant la vision artificielle comme principal outil. Il examine en détail les bases de cette pratique, les méthodes classiques employées, ainsi que les progrès récents dans le domaine de la vision artificielle pour une détection automatisée et efficace des défauts.

2 Définition et classification des défauts de surface

2.1 Définition des défauts de surface

Les défauts de surface désignent toutes les anomalies, irrégularités ou imperfections perceptibles ou non à l'œil nu sur la surface d'un matériau, d'un produit ou d'un composant. Ces défauts peuvent prendre diverses formes telles que des fissures, des éraflures, des bosses, des tâches, ou encore des variations de texture. Leur présence peut altérer l'aspect esthétique,

la fonctionnalité ou même la sécurité du produit final.

2.2 Détection des défauts de surface

La détection des défauts de surface consiste en l'identification et la localisation de ces anomalies afin d'assurer la qualité et la conformité des produits. La détection des défauts permet de prévenir les non-conformités dès les premières étapes de la production, réduisant ainsi les coûts associés aux retouches et aux rebuts de produits défectueux, et garantissant la sécurité des produits et des processus en identifiant les anomalies pouvant compromettre leur fiabilité et leur fonctionnalité.

2.3 Classification des défauts de surface

Les défauts de surface peuvent être classés de différentes manières en fonction de divers critères tels que leur origine, leur nature, leur taille ou leur impact sur la qualité du matériau ou de l'objet [2] [3]. Voici quelques classifications courantes des défauts de surface :

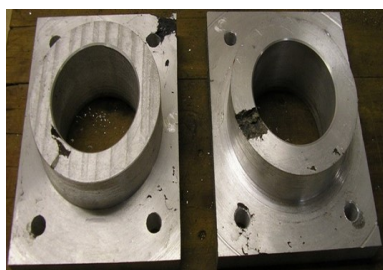
2.3.1 Origine

Selon l'origine du défaut on distingue deux types de défauts :

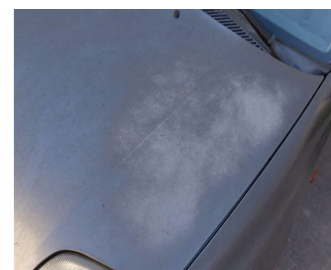
- Défauts de fabrication** : Ils surviennent pendant le processus de fabrication initial du matériau ou de l'objet. Par exemple, des bulles d'air emprisonnées dans le verre, des imperfections de moulage sur des pièces plastiques, ou des rayures sur une surface métallique causées par un processus d'usinage inadéquat.

- Défauts environnementaux** : Ces défauts sont extrinsèque à l'objet et causés par des influences extérieures telles que l'exposition à l'eau, à l'air, ou à d'autres agents corrosifs. Par exemple, la corrosion de métaux exposés à des environnements humides, la décoloration de surfaces peintes exposées à la lumière du soleil, ou l'usure de surfaces en plastique exposées à des produits chimiques.

La figure 1.1 illustre ces deux catégories de défauts.



(a) Défaut de fabrication



(b) Défaut environnemental

FIGURE 1.1 – Types de défauts de l'origine

2.3.2 Nature

On peut aussi classer les défauts de surface en fonction de leurs structures physiques ou chimiques :

•**Physiques** : Ces défauts altèrent directement l'aspect ou la structure physique de la surface visible. Par exemple, des bosses causées par un choc, des rayures dues à un frottement ou des fissures résultant d'une contrainte excessive.

•**Chimiques** : Ces défauts résultent de réactions chimiques altérant la composition de la surface. Par exemple, la formation de rouille sur des surfaces métalliques exposées à l'humidité ou à l'air, ou la décoloration causée par des réactions avec des produits chimiques corrosifs.

La figure 1.2 montre quelques exemples de ces deux types de défauts.



FIGURE 1.2 – Défauts liés à la nature

2.3.3 Taille

Il est également possible de distinguer les défauts selon leurs tailles :

•**Microscopiques** : Ces défauts sont si petits qu'ils ne sont pas visibles à l'œil nu. Par exemple, des microfissures dans un matériau ou des pores microscopiques dans une surface.

•**Macroscopiques** : Ceux-ci sont visibles sans l'aide d'instruments spéciaux. Par exemple, des éraflures profondes ou des bosses évidentes.

La figure 1.3 montre des exemples de ces deux types de défauts.



FIGURE 1.3 – Défauts selon la taille

2.3.4 Impact sur la qualité

En fonction de la sévérité du défaut on distingue deux types :

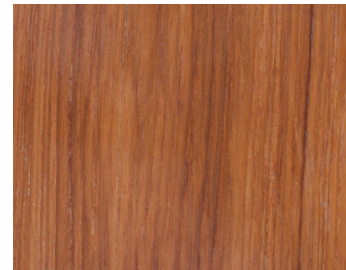
- **Critiques** : Ces défauts affectent sérieusement la qualité ou la fonctionnalité. Par exemple, des fissures dans une pièce structurale, des bulles d'air dans un produit scellé ou des trous dans un tuyau en PVC.

- **Non critiques** : Ces défauts sont moins graves et n'affectent pas fondamentalement la qualité. Par exemple, de petites égratignures sur un meuble ou des imperfections mineures dans la peinture.

La figure 1.4 illustre des exemples de ces deux types de défauts.



(a) Critique



(b) Non critique

FIGURE 1.4 – Défauts impactant la qualité

2.4 Domaine d'application

La détection des défauts de surface est une méthode pratique essentielle dans de nombreux domaines industriels. Parmi ces domaines, on trouve [4] :

2.4.1 Industrie métallurgique

Cette industrie traite la production, le traitement et la manipulation des métaux, couvrant une gamme étendue d'activités, de la production d'acier à la fabrication de pièces métalliques diverses. Les défauts de surface courants dans la métallurgie comprennent les soufflures, les inclusions et les piqûres, qui peuvent affaiblir la structure du métal et entraîner des défaillances potentiellement dangereuses (Fig.1.5).

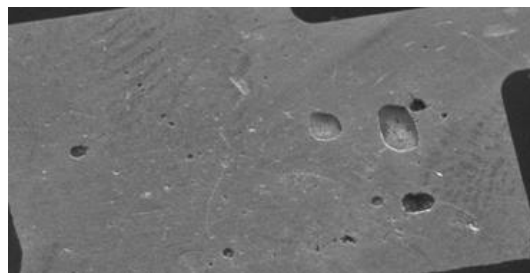


FIGURE 1.5 – Exemple de défaut de soufflures dans l'industrie métallurgique

2.4.2 Industrie alimentaire

Ce domaine se concentre sur la production, la transformation et la distribution d'aliments, garantissant la sécurité alimentaire, la qualité et la conformité réglementaire. Les défauts de surface dans ce domaine peuvent inclure les fissures ou les éclats dans les emballages, ainsi que les contaminations visibles sur les produits eux-mêmes, ce qui peut compromettre leur sécurité et leur qualité (Fig.1.6).



FIGURE 1.6 – Exemple de défaut de production dans l'industrie alimentaire

2.4.3 Industrie pharmaceutique

L'industrie pharmaceutique s'engage dans la recherche, le développement, la fabrication et la commercialisation de produits pharmaceutiques tels que les médicaments, les vaccins et les dispositifs médicaux. Les défauts de surface observés sur les comprimés, les capsules ou les emballages peuvent se manifester sous forme de fissures, d'éclats ou de marques d'usure, ce qui peut avoir un impact sur l'efficacité du médicament ou mettre en péril sa sécurité (Fig.1.7).



FIGURE 1.7 – Exemple de défaut des comprimés médicamenteux

2.4.4 Industrie du verre

Cette industrie englobe la fabrication et la transformation du verre, largement utilisé dans diverses applications telles que les fenêtres, les bouteilles, les écrans de smartphones et les composants optiques. Les défauts de surface du verre, tels que les bulles, les éraflures, les stries ou les inclusions, peuvent altérer à la fois l'aspect esthétique et la qualité structurelle du matériau (Fig.1.8).



FIGURE 1.8 – Verre fissuré

2.4.5 Industrie du plastique

Cette industrie fabrique une grande variété de produits en plastique, allant des emballages et des conteneurs aux pièces automobiles et électroniques. Les défauts de surface des pièces plastiques peuvent inclure des bulles, des fissures, des marques d'écoulement ou des déformations, qui peuvent compromettre leur fonctionnalité et leur apparence (Fig.1.9).



FIGURE 1.9 – Marque d'écoulement

2.4.6 Industrie du papier et de l'impression

Ce domaine comprend la fabrication de papier ainsi que l'impression de documents, de livres, de magazines, d'emballages et d'autres produits papetiers. Ces défauts peuvent inclure des plis, des tâches, des impuretés ou des irrégularités d'impression, qui peuvent affecter leur aspect visuel et leur qualité (Fig.1.10).



FIGURE 1.10 – Défaut de papier

2.4.7 Industrie du textile

Cette industrie concerne la production, la transformation et la fabrication de vêtements, allant des fibres naturelles comme le coton et la laine aux tissus synthétiques. Ces défauts peuvent inclure des nœuds, des fils cassés, des tâches ou des déchirures, qui peuvent affecter leur apparence et leur durabilité (Fig.1.11).



FIGURE 1.11 – Défauts de tâches sur un textile

2.4.8 Industrie automobile

Ce secteur englobe la conception, la fabrication, la vente et l'entretien des véhicules, y compris les voitures, les camions, les motos et les véhicules utilitaires. Les défauts de surface dans ce domaine comprennent les rayures, les bosses, les bulles de peinture, les marques d'usure ou les défauts de soudure, qui peuvent affecter l'esthétique et la sécurité des véhicules (Fig.1.12).



FIGURE 1.12 – Défaut de rayure sur un par brise d'un véhicule

2.4.9 Industrie électronique

Cette industrie couvre la conception, la fabrication et la distribution de composants électroniques, de dispositifs et d'équipements allant des smartphones et des ordinateurs aux systèmes embarqués et aux appareils médicaux. Les défauts de surface des composants électroniques peuvent inclure des joints soudés défectueux, des puces endommagées, des traces des doigts des tâches d'huile des particules de poussière (Fig.1.13).



FIGURE 1.13 – Carte électronique endommagée

2.5 Intérêts et avantages de la détection des défauts de surface

La détection des défauts de surface présente plusieurs intérêts et avantages significatifs dans divers domaines industriels :

1. **Assurance qualité** : La détection précoce des défauts de surface permet d'identifier les problèmes potentiels dès le début du processus de fabrication, ce qui garantit la production de produits conformes aux normes de qualité.
2. **Réduction des coûts** : En identifiant et en éliminant les défauts de surface dès le stade de la production, les fabricants évitent les coûts liés aux rejets de produits défectueux, aux réparations et aux rappels de produits, ce qui peut entraîner des économies importantes.
3. **Amélioration de la satisfaction client** : En garantissant la qualité et la fiabilité des produits, la détection des défauts de surface contribue à la satisfaction des clients en fournissant des produits conformes à leurs attentes en termes de fonctionnalité, d'apparence et de durabilité.
4. **Sécurité** : Dans des secteurs tels que l'aérospatiale, l'automobile et l'industrie pharmaceutique, la détection des défauts de surface contribue à garantir la sécurité des produits en identifiant et en éliminant les imperfections pouvant compromettre leur intégrité structurelle ou leur efficacité.
5. **Optimisation des processus de fabrication** : En identifiant les sources de défauts de surface, les fabricants peuvent mettre en œuvre des améliorations dans leurs processus de fabrication pour réduire la probabilité de défauts futurs, améliorant ainsi l'efficacité et la rentabilité de la production.
6. **Maintien de la réputation de la marque** : La détection des défauts de surface aide à prévenir les problèmes de qualité qui pourraient nuire à la réputation d'une entreprise. En garantissant des normes de qualité élevées, les fabricants renforcent la confiance des consommateurs dans leurs produits et leur marque.
7. **Conformité réglementaire** : Dans certaines industries réglementées, telles que l'industrie pharmaceutique et l'industrie alimentaire, la détection des défauts de surface est nécessaire pour se conformer aux normes de qualité et de sécurité établies par les autorités réglementaires.

3 Méthodes traditionnelles de détection des défauts de surface

3.1 Inspection visuelle humaine

L'inspection visuelle humaine, souvent désignée simplement comme l'examen visuel, est une méthode classique de détection des défauts de surface largement utilisée dans divers secteurs industriels. Elle repose sur l'observation directe et minutieuse d'un matériau ou d'un produit fini par un opérateur humain afin d'identifier tout défaut visible à l'œil nu (Fig.1.14).



FIGURE 1.14 – Inspection visuelle

Pour faciliter cette observation, des techniques d'imagerie sont exploitées.

3.2 Techniques d'imagerie conventionnelles

Les techniques d'imagerie conventionnelles sont des méthodes largement utilisées pour la détection des défauts de surface dans divers secteurs industriels. Elles se fondent sur l'utilisation d'appareils d'imagerie spécifiques pour capturer des images des surfaces des matériaux ou des produits finis, qui sont ensuite analysées visuellement par l'humain ou intégrés dans un système automatique pour identifier tout défaut ou anomalie. Voici quelques-unes de ces techniques :

3.2.1 Photographie

La photographie est l'une des méthodes les plus simples et courantes pour capturer des images de surface. Elle utilise des caméras standards pour capturer des images fixes qui peuvent ensuite être analysées pour détecter les défauts (Fig.1.15a).

3.2.2 Microscopie optique

La microscopie optique utilise des microscopes pour observer les surfaces à un niveau de détail très fin. Elle est particulièrement utile pour détecter les défauts minuscules ou subtils qui peuvent ne pas être visibles à l'œil nu (Fig.1.15b).

3.2.3 Caméras industrielles

Les caméras industrielles sont conçues spécifiquement pour des applications d'inspection et de surveillance dans des environnements industriels. Elles offrent souvent une résolution élevée, une sensibilité à la lumière ajustable et des fonctionnalités avancées telles que la vision nocturne (Fig.1.15c).

3.2.4 Caméras thermiques

Les caméras thermiques, également appelées caméras infrarouges, détectent les variations de température sur les surfaces pour identifier les anomalies thermiques. Elles sont utiles pour détecter les défauts tels que les points chauds ou les zones de refroidissement inégales (Fig.1.15d).

3.2.5 Caméras à ultrasons

Les caméras à ultrasons utilisent des ondes ultrasonores pour générer des images des surfaces internes ou externes des matériaux. Elles sont particulièrement utiles pour détecter les défauts subsurface tels que les fissures ou les inclusions (Fig.1.15e).



FIGURE 1.15 – Différentes caméras

3.3 Utilisation de méthodes de contrôle non destructif

Les méthodes de contrôle non destructif (CND) sont des techniques utilisées pour inspecter les matériaux ou les produits afin de détecter les défauts sans les endommager [5]. Ces

méthodes permettent d'identifier les imperfections à la surface ou à l'intérieur des matériaux. Nous allons maintenant explorer quelques-unes des principales méthodes de CND utilisées dans divers secteurs industriels [6] :

3.3.1 Contrôle par ressuage (ou pénétrant)

•**Principe** : Le contrôle par ressuage consiste à appliquer un liquide pénétrant sur la surface d'un matériau. Ce liquide pénètre dans les fissures, les pores ou les défauts de surface du matériau. Une fois que le liquide pénétrant a été laissé en contact avec la surface pendant un certain temps, l'excès de liquide est enlevé. Ensuite, un révélateur est appliqué, ce qui permet de faire ressortir les défauts en créant une indication visible sur la surface (Fig.1.16).

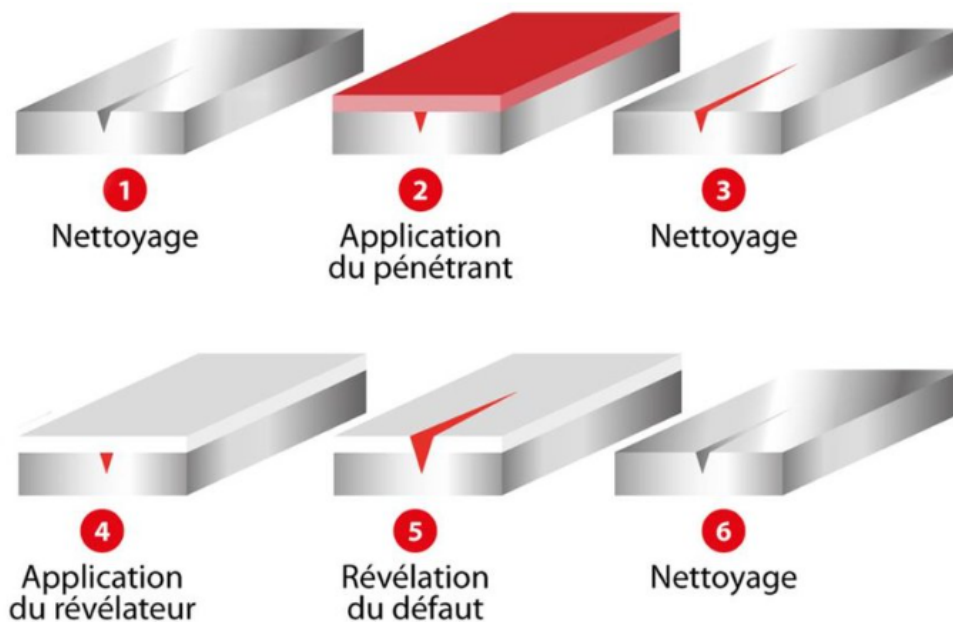


FIGURE 1.16 – Contrôle par ressuage

•**Applications** : Le contrôle par ressuage est principalement utilisé pour détecter les défauts de surface tels que les fissures, les porosités, les délaminations et les joints mal formés sur des matériaux métalliques, plastiques et céramiques.

3.3.2 Contrôle par magnétoscopie

•**Principe** : Il repose sur la détection des variations du champ magnétique causées par les défauts de surface du matériau.

Le matériau est magnétisé à l'aide d'un aimant ou d'un électroaimant. Des particules magnétiques sont ensuite appliquées à sa surface. Ces particules s'accumulent aux endroits où il y a des défauts de surface, mettant ainsi en évidence leur présence (Fig.1.17).

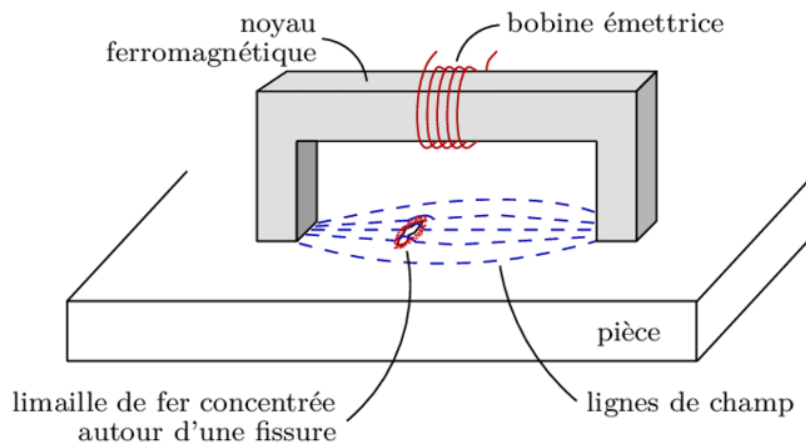


FIGURE 1.17 – Contrôle par magnétoscopie

•**Applications** : Cette méthode est particulièrement efficace pour détecter les fissures de surface dans les matériaux ferromagnétiques tels que l'acier.

3.3.3 Contrôle par courant de Foucault

•**Principe** : Le contrôle par courant de Foucault repose sur le principe de l'induction électromagnétique. Des courants électromagnétiques induits sont générés à la surface du matériau, créant un champ magnétique. Les variations dans ce champ magnétique dues aux défauts de surface sont détectées. Les variations de conductivité électrique de la surface du matériau, affectent le champ magnétique, ce qui permet de les détecter (Fig.1.18).

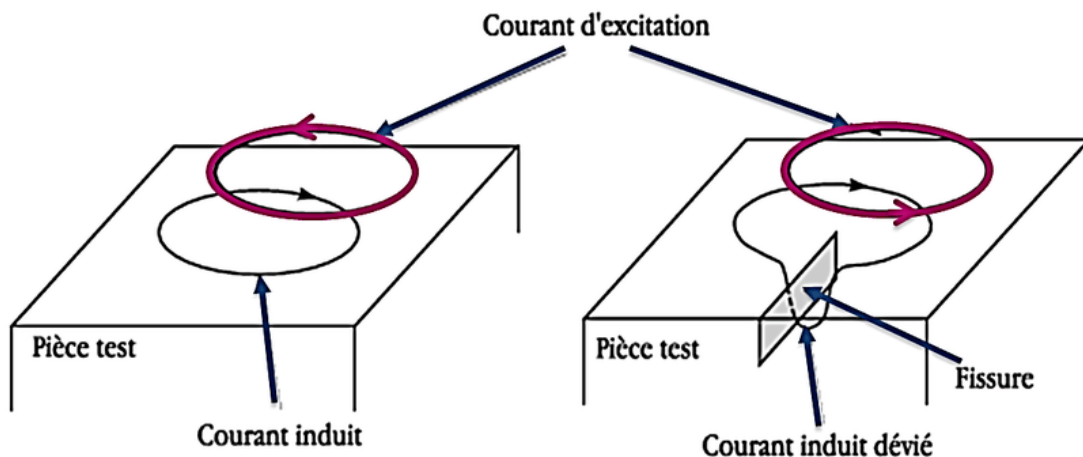


FIGURE 1.18 – Contrôle par courant de Foucault

•**Applications** : Cette méthode est efficace pour détecter une variété de défauts de surface dans les matériaux conducteurs tels que les métaux.

3.3.4 Contrôle par radiographie (rayons X et gammagraphie)

•**Principe** : Le contrôle par radiographie utilise des rayons X ou des rayons gamma pour créer des images des défauts de surface et internes dans les matériaux. Les rayons traversent le matériau et atteignent un film radiographique ou un capteur numérique. Les zones plus denses ou présentant des défauts absorbent d'avantage de rayons, ce qui crée des zones sombres sur l'image radiographique (Fig.1.19).

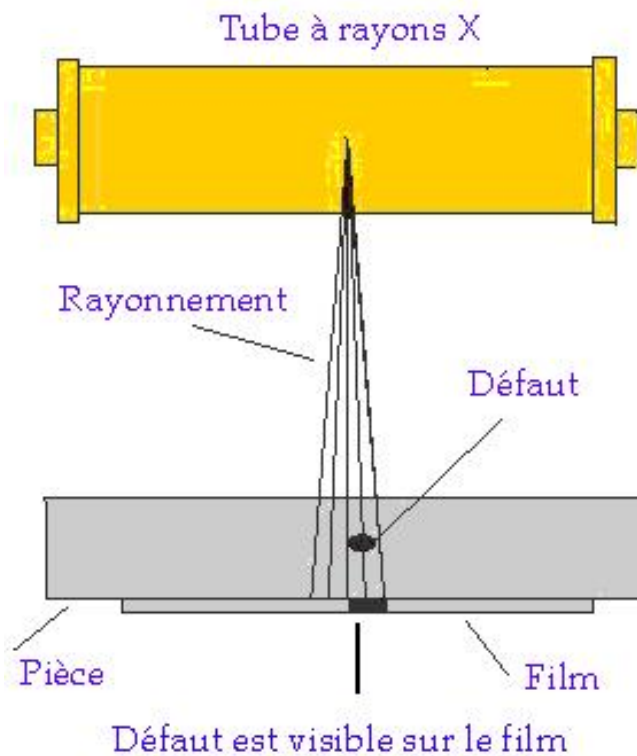


FIGURE 1.19 – Contrôle par radiographie

•**Applications** : Cette méthode est couramment utilisée pour détecter une large gamme de défauts de surface et internes tels que les fissures, les inclusions, les porosités et les vides dans les métaux, les composites et les soudures.

3.3.5 Contrôle par ultrasons (US)

•**Principe** : Le contrôle par ultrasons utilise des ondes sonores à haute fréquence qui se propagent à travers le matériau. Les variations dans les ondes captées sont analysées pour détecter les défauts de surface. Lorsque les ondes ultrasonores rencontrent un changement dans les propriétés du matériau, tel qu'un défaut de surface, une partie de l'énergie des ultrasons est réfléchiée ou diffusée (Fig.1.20).

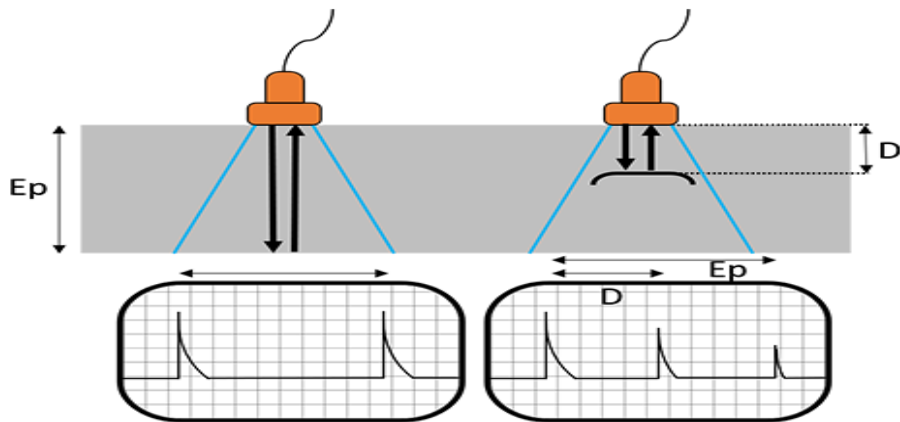


FIGURE 1.20 – Contrôle par ultrasons

•**Applications** : Cette méthode est largement utilisée pour inspecter une grande variété de matériaux, tels que les métaux, les composites et les plastiques, pour détecter les défauts de surface comme les fissures, les éclatements, les pores et les inclusions.

3.4 Limitations des méthodes traditionnelles

1. **Subjectivité** : Les résultats peuvent varier selon l'interprétation de l'opérateur, car la détection des défauts dépend de son jugement personnel.
2. **Dépendance à la formation** : La qualité des résultats dépend fortement du niveau de compétence et d'expérience des opérateurs. Un manque de formation peut conduire à des erreurs de détection.
3. **Temps de traitement** : L'inspection manuelle peut prendre beaucoup de temps, surtout pour des surfaces complexes ou des pièces volumineuses, ce qui peut entraîner des retards dans la production.
4. **Coûts élevés** : Les méthodes traditionnelles peuvent être coûteuses en raison de l'équipement spécialisé nécessaire, du temps de formation des opérateurs et des arrêts de production nécessaires pour les inspections.
5. **Limites de détection** : Certaines méthodes ont des limites dans la détection des défauts de petite taille, ce qui signifie qu'ils peuvent manquer des défauts plus petits.
Compte tenu de ces limitations, et afin de les surmonter, des méthodes plus avancées et automatisées ont été développées. Ces méthodes permettent d'améliorer la fiabilité et l'efficacité de la détection des défauts de surface.

4 Détection automatisée des défauts de surface

La détection automatisée des défauts de surface reflète une tendance croissante à utiliser des technologies avancées pour améliorer l'efficacité et la fiabilité des processus d'inspection industrielle [7]. Cette transition est motivée par plusieurs facteurs, notamment :

- La nécessité de garantir une qualité élevée des produits.
- Réduire les coûts de production.
- Améliorer la compétitivité sur le marché mondial.

Les systèmes de vision artificielle jouent un rôle crucial dans cette évolution, offrant des capacités avancées de détection et de classification des défauts de surface grâce à l'utilisation de techniques telles que l'apprentissage automatique et les réseaux de neurones. En automatisant la détection des défauts de surface, les entreprises peuvent accroître l'efficacité de leurs processus d'inspection, réduire les risques d'erreurs humaines et assurer une cohérence et une fiabilité accrues des résultats.

Cette évolution vers la détection automatisée des défauts de surface représente un progrès significatif dans le domaine de l'inspection industrielle, ouvrant la voie à des méthodes d'inspection plus rapides, plus précises et plus économiques.

5 Système de vision artificielle de détection des défauts de surface

Les méthodes automatiques de détection des défauts de surface reposent sur l'utilisation de système de vision artificielle assistés par des ordinateurs pour identifier et caractériser les défauts sur les matériaux et les produits. Un système de vision artificielle dédié à la détection des défauts de surface est schématisé sur la figure (1.21) :

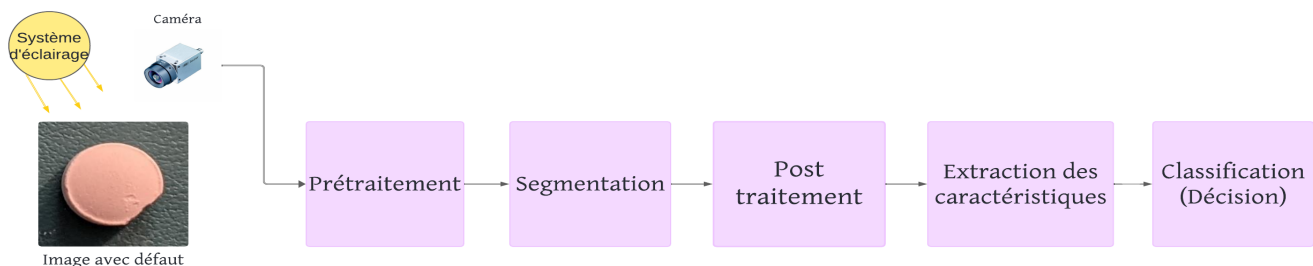


FIGURE 1.21 – Schéma d'un système de vision artificielle

5.1 Acquisition des données visuelles

Tout d'abord, les données visuelles sont acquises à l'aide de capteurs tels que des caméras, des scanners 3D, etc. Ces capteurs convertissent la lumière en données numériques que les ordinateurs peuvent traiter.

Cependant, la qualité des données visuelles obtenues dépend également d'un facteur souvent sous-estimé qui est l'éclairage. Un bon éclairage est essentiel pour garantir la qualité d'image souhaitée dans tout système d'imagerie. En effet, un éclairage approprié peut résoudre de nombreux problèmes de contraste et de résolution auxquels les utilisateurs peuvent être confrontés. Plutôt que de se concentrer uniquement sur l'amélioration des composants matériels tels que les détecteurs ou les objectifs d'imagerie, il est important de reconnaître l'impact significatif que l'éclairage peut avoir sur la qualité globale de l'image. Investir dans un éclairage adéquat peut souvent suffire à obtenir des résultats satisfaisants, sans avoir besoin de recourir à des solutions plus coûteuses.

5.1.1 Types d'éclairage

Les techniques d'éclairage sont des méthodes précises employées en vision artificielle pour illuminer les objets ou les surfaces. Leur but est de garantir des images de qualité supérieure pour des analyses précises, en mettant en évidence des caractéristiques essentielles comme les défauts de surface, les contours ou encore les textures. Voici quelques exemples de ces techniques couramment utilisées [8] :

Éclairage directionnel : L'éclairage directionnel est une méthode où la lumière est projetée sur un objet sous un angle spécifique. Cela crée des ombres et des reflets qui mettent en évidence certains défauts de surface comme les fissures, les bosses, les creux, la rugosité et les défauts de forme. C'est particulièrement utile pour détecter des détails subtils qui pourraient ne pas être visibles avec un éclairage uniforme (Fig.1.22).

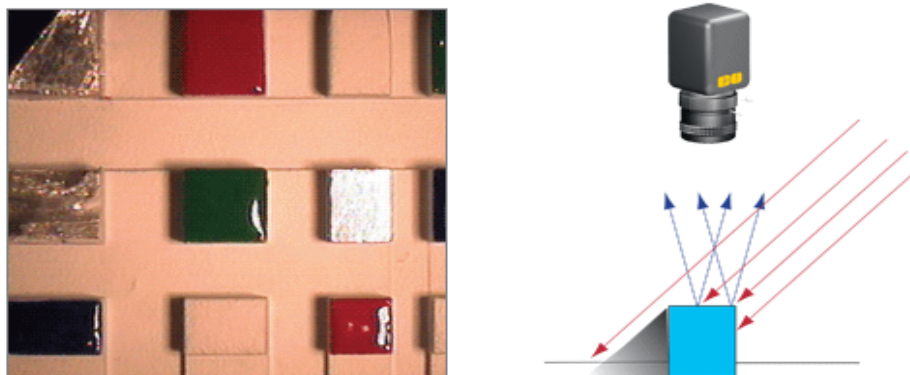


FIGURE 1.22 – Éclairage directionnel

Éclairage brillant : L'éclairage brillant est une technique qui consiste à utiliser une source lumineuse très intense pour mettre en évidence les détails minuscules et les imperfections sur une surface. Cette intensité lumineuse élevée permet de détecter des défauts tels que les rayures, les marques fines et d'autres petits problèmes qui seraient moins visibles avec un éclairage ordinaire (Fig.1.23).

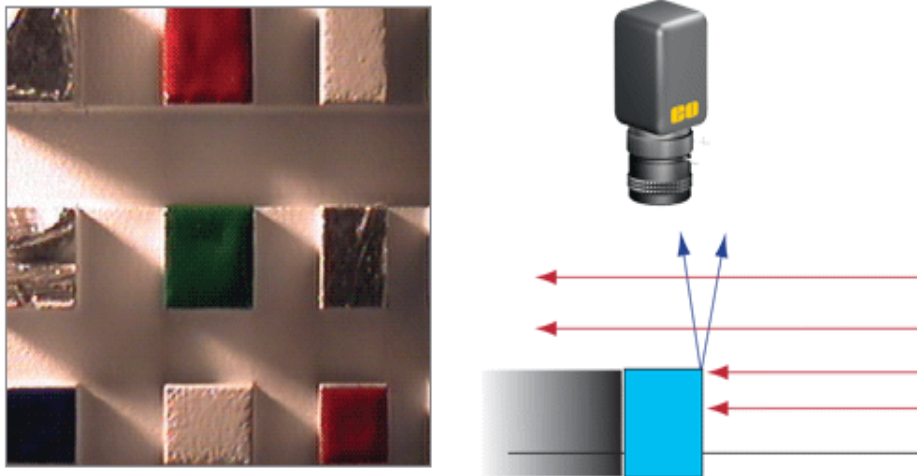


FIGURE 1.23 – Éclairage brillant

Éclairage diffus : L'éclairage diffus est une méthode où la lumière est répartie de manière uniforme sur une surface, évitant ainsi les ombres et les reflets forts. Cette technique permet d'observer les détails subtils et les textures sans créer de contrastes marqués. En d'autres termes, l'éclairage diffus crée une lumière douce et homogène qui est idéale pour l'inspection de surfaces lisses (Fig.1.24).

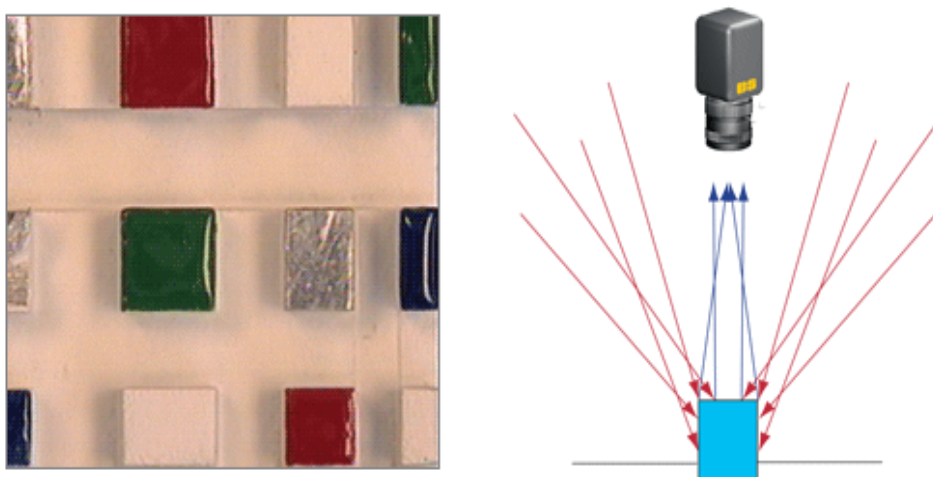


FIGURE 1.24 – Éclairage diffus

Éclairage coaxial : Dans ce type d'éclairage, la source lumineuse est placée directement sur l'axe optique de l'objectif de la caméra ou du système d'inspection. Cela signifie que la lumière est projetée dans la même direction que la vision de l'observateur, créant ainsi un éclairage uniforme et éliminant les ombres et les reflets indésirables. L'éclairage coaxial est utilisé pour inspecter les surfaces réfléchissantes et détecter des défauts subtils comme les rayures et les imperfections de peinture (Fig.1.25).

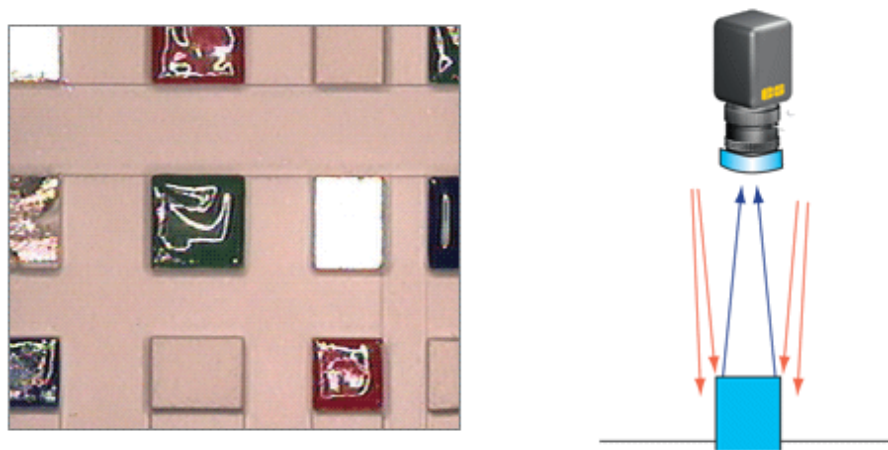


FIGURE 1.25 – Éclairage coaxial

Éclairage axial diffus : Pour réaliser cette technique, on utilise souvent un séparateur de faisceaux, parfois appelé miroir dichroïque. Ce composant réfléchit la lumière provenant d'une source vers l'objet à inspecter, tout en permettant à l'image réfléchi de passer à travers vers la caméra ou le système d'inspection (Fig.1.26).

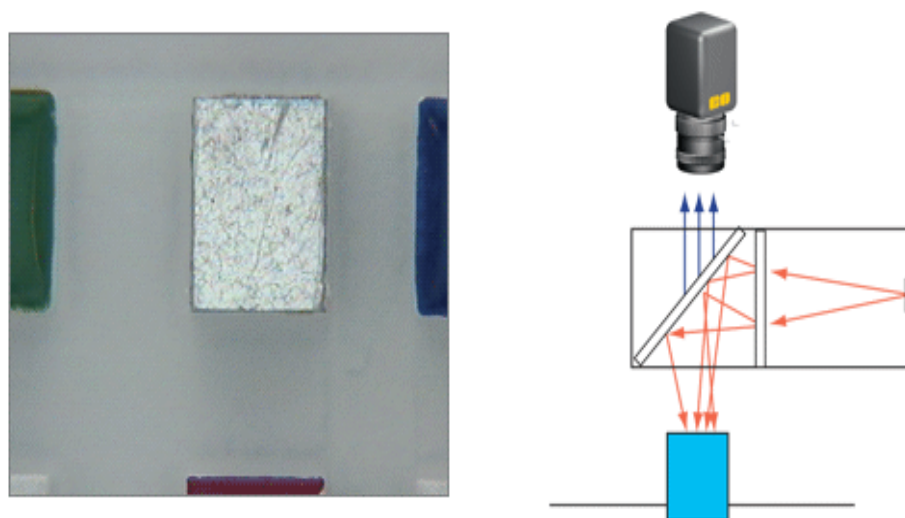


FIGURE 1.26 – Éclairage axial diffus

Lumière structurée : Elle consiste à projeter un motif lumineux prédéfini sur l'objet à inspecter, généralement sous la forme de lignes ou de grilles. Cette projection de motifs lumineux permet de créer des contrastes sur l'objet, facilitant ainsi la détection des contours, des déformations et des irrégularités de surface (Fig.1.27).

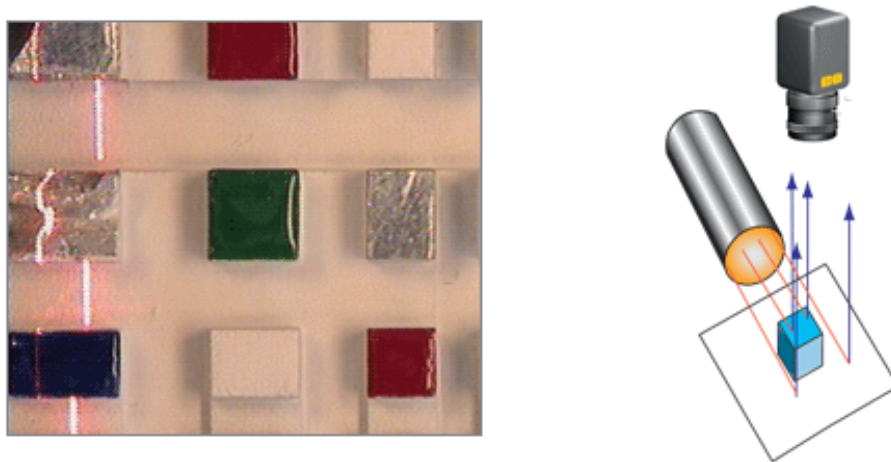


FIGURE 1.27 – Lumière structurée

Lumière polarisée : La lumière polarisée est une méthode d'éclairage utilisée pour réduire les reflets et améliorer la visibilité des détails de surface lors de l'inspection. Pour ce faire, on utilise des filtres polarisants sur la source lumineuse et sur la caméra. Ces filtres contrôlent la direction et l'orientation de la lumière qui atteint l'objet inspecté et qui est capturée par la caméra.

En filtrant les réflexions parasites et en alignant la polarisation de la lumière, on peut obtenir des images avec un meilleur contraste et moins de reflets, ce qui facilite la détection des défauts sur des surfaces réfléchissantes ou dans des environnements où les reflets peuvent être problématiques pour l'analyse visuelle (Fig.1.28).

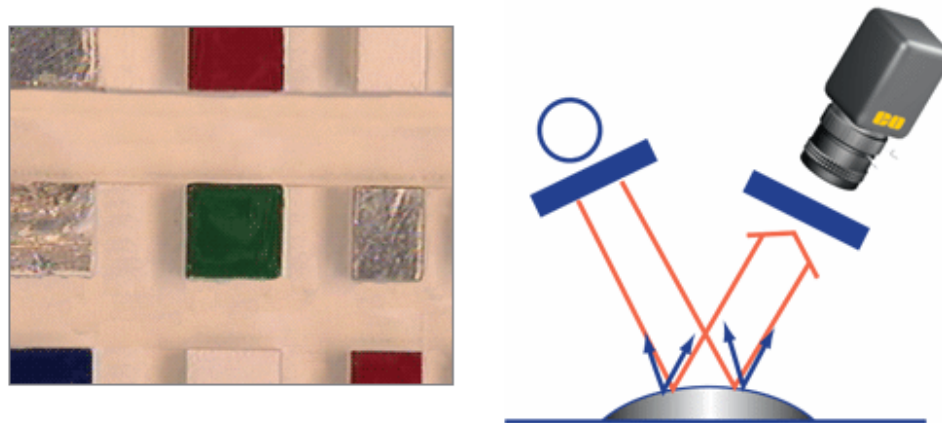


FIGURE 1.28 – Lumière polarisée

Fond noir : Le fond noir est une technique d'éclairage utilisée dans l'imagerie et l'inspection pour créer un contraste élevé entre l'objet à examiner et son environnement. Contrairement à un éclairage traditionnel où l'objet est illuminé par une lumière directe, l'éclairage de fond noir implique l'utilisation d'une source lumineuse placée derrière l'objet et d'un fond sombre ou noir. L'objectif de cette technique est de mettre en évidence les détails et les caractéristiques de l'objet en créant un fort contraste entre la luminosité de l'objet et l'obscurité du fond. Cela permet de visualiser plus clairement les contours, les textures et les défauts de surface de l'objet, ce qui facilite l'inspection et la détection des anomalies (Fig.1.29).

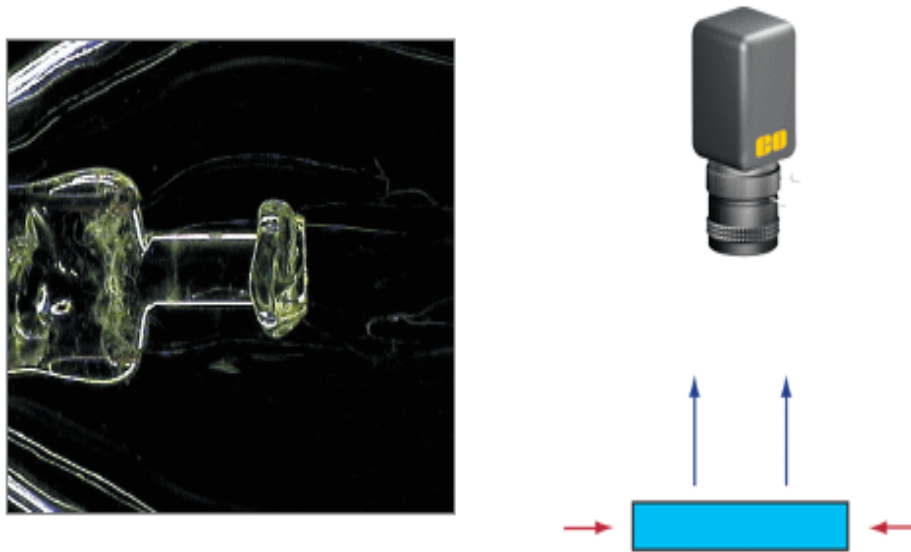


FIGURE 1.29 – Fond noir

5.2 Prétraitement

Le prétraitement des images est une étape fondamentale dans le processus d'analyse d'images de défauts de surface. Il vise à optimiser la qualité des données visuelles en éliminant les artefacts indésirables, en améliorant le contraste, et en mettant en évidence les caractéristiques pertinentes. Sans un prétraitement adéquat, la détection et la classification des défauts peuvent être compromises par des informations erronées ou des données bruitées. Lorsqu'il s'agit de traiter des images pour corriger des anomalies, il est crucial de sélectionner des méthodes de prétraitement adaptées à la nature spécifique des imperfections présentes dans ces images. On peut classer les méthodes de prétraitement en différentes catégories en fonction des types d'anomalies présentes dans l'image. On distingue deux types de prétraitement (Fig.1.30) :

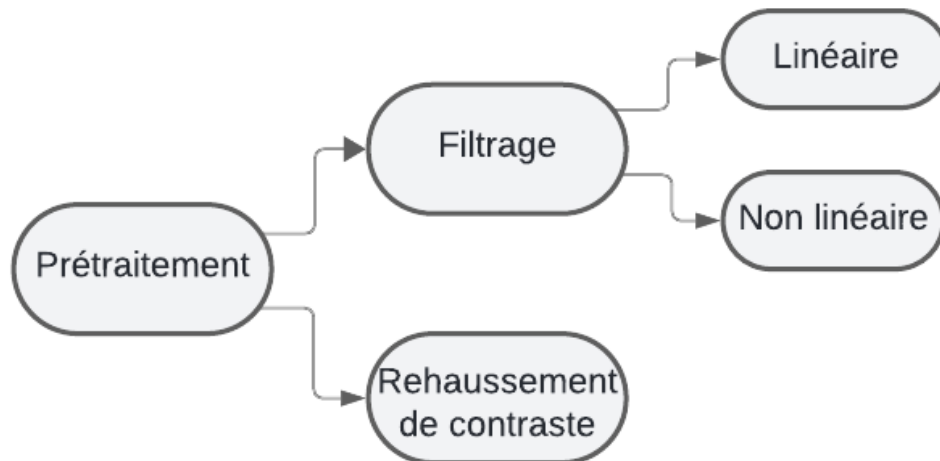


FIGURE 1.30 – Méthodes de prétraitement

5.2.1 Filtrage

Le filtrage d'images, est une opération mathématique appliquée à chaque pixel d'une image pour modifier ses propriétés. Dans la détection de défauts de surface, les filtres sont utilisés pour améliorer la qualité de l'image en réduisant le bruit, en accentuant les contours et en mettant en évidence les caractéristiques importantes pour faciliter la détection des défauts. En réduisant le bruit, les défauts deviennent plus visibles, tandis que l'accentuation des contours permet de les détecter plus facilement.

Les filtres préparent l'image pour une détection plus précise des défauts, elles sont de deux types :

Filtres linéaires

Les filtres linéaires appliquent des transformations linéaires aux pixels de l'image en utilisant des noyaux de convolution. Ce processus implique l'utilisation d'un noyau ou masque de convolution K qui passe sur chaque pixel de l'image et calcule une nouvelle valeur basée sur les valeurs des pixels voisins et les coefficients du noyau.

L'opération de convolution discrète est définie comme suit :

$$I'[i, j] = \sum_{m=-a}^a \sum_{n=-b}^b I[i - m, j - n] \cdot K[m, n] \quad (1.1)$$

où :

— $I'[i, j]$ est la nouvelle valeur du pixel à la position (i, j) dans l'image convoluée,

- $I[i - m, j - n]$ est la valeur du pixel de l'image d'origine à la position décalée de (m, n) par rapport au pixel (i, j) ,
- $K[m, n]$ est le coefficient du noyau à la position (m, n) ,
- a et b sont les demi-longueurs du noyau, déterminant la taille de la zone de voisinage pour chaque pixel, souvent $a=b$.

La convolution pour le pixel $(3, 3)$ d'une image I de taille 5×5 et un noyau K de taille 3×3 (Fig.1.31) est :

$$\begin{aligned}
 I'(3, 3) = & I(2, 2) \cdot K(-1, -1) + I(2, 3) \cdot K(0, -1) + I(2, 4) \cdot K(1, -1) \\
 & + I(3, 2) \cdot K(-1, 0) + I(3, 3) \cdot K(0, 0) + I(3, 4) \cdot K(1, 0) \\
 & + I(4, 2) \cdot K(-1, 1) + I(4, 3) \cdot K(0, 1) + I(4, 4) \cdot K(1, 1)
 \end{aligned} \tag{1.2}$$

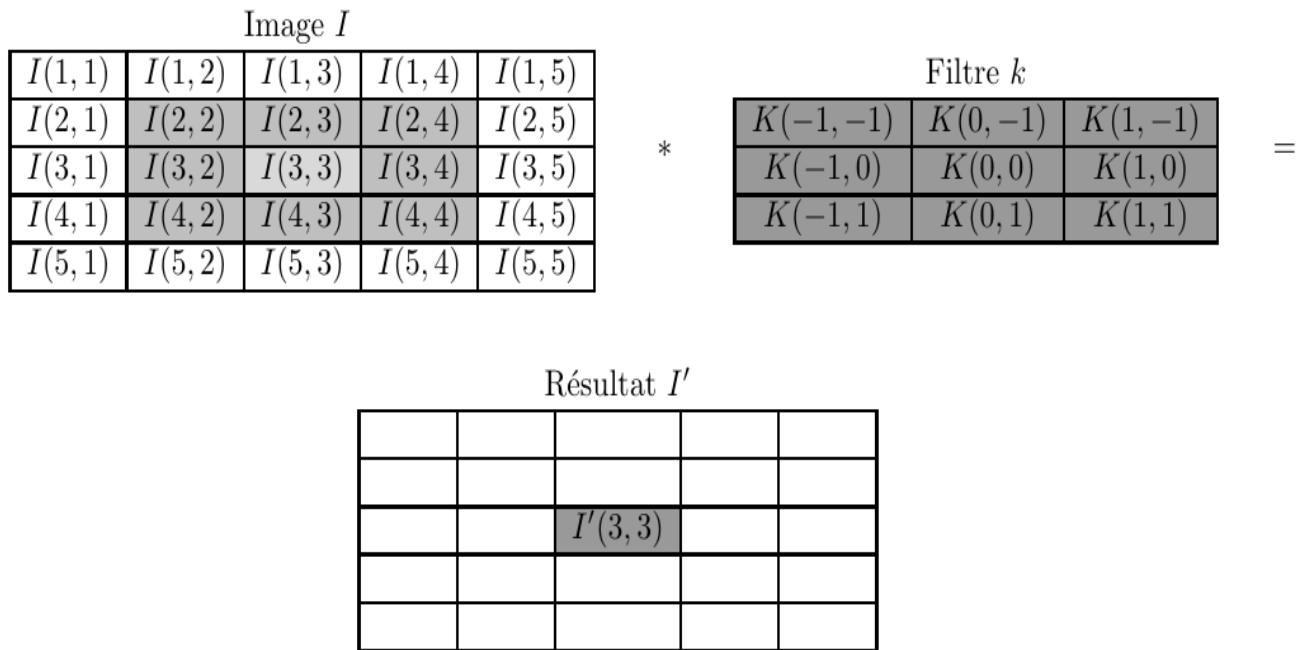


FIGURE 1.31 – Exemple d'une opération de convolution

Plusieurs filtres (noyau K) ont été proposés.

-Filtre de lissage (moyenueur) : Le filtre moyenneur également appelé filtre de lissage, est un type de filtre linéaire utilisé pour réduire le bruit et adoucir les contours. Il utilise un noyau K de taille $n \times n$ où n est généralement un nombre impair. Chaque élément du noyau est pondéré par $\frac{1}{n^2}$, de sorte que la somme de tous les éléments du noyau soit égale à 1 (Fig.1.32).

Filtre moyennneur 3 × 3 :

$$K = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filtre moyennneur 5 × 5 :

$$K = \frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

FIGURE 1.32 – Exemple de filtres moyennneurs

L'application d'un filtre moyennneur à tous les pixels de l'image, permet d'obtenir une nouvelle image où le bruit a été réduit et les contours sont plus lisses. Cependant, il peut également entraîner une perte de détails, notamment dans les zones de forte transition. Le filtre moyennneur agit comme un filtre passe-bas en atténuant les hautes fréquences de l'image (qui sont souvent associées au bruit) tout en conservant les basses fréquences (qui représentent les structures principales de l'image), comme illustré sur la figure (1.33)

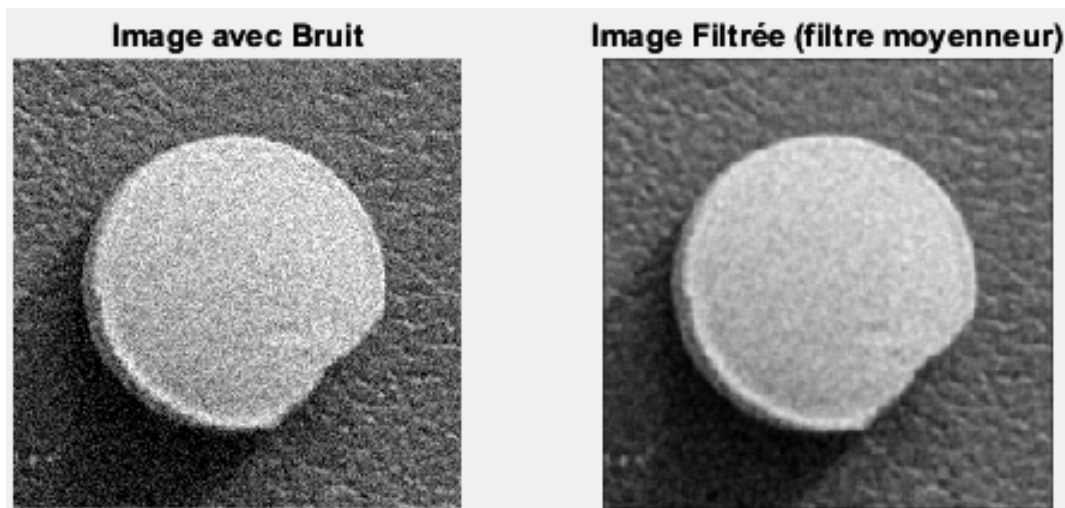


FIGURE 1.33 – Application du filtre moyennneur

- **Filtre gaussien** : Utilise une fonction gaussienne pour pondérer les pixels voisins, donnant plus de poids aux pixels proches du pixel central.

L'expression de la gaussienne en 2D est donnée par la relation suivante :

$$K(m, n) = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{m^2 + n^2}{2\sigma^2}\right) \quad (1.3)$$

où $k(m, n)$ est la valeur du noyau de Gaussien à la position (m, n) , et σ est l'écart-type du noyau qui contrôle le niveau de lissage.

La discrétisation de la fonction gaussienne est donnée par ces filtres gaussiens (Fig.1.34) :

Filtre 3×3 pour $\sigma = 0.8$:

$$K = \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Filtre 5×5 pour $\sigma = 1$:

$$K = \frac{1}{300} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 18 & 30 & 18 & 4 \\ 6 & 30 & 48 & 30 & 6 \\ 4 & 18 & 30 & 18 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

FIGURE 1.34 – Exemples de filtres Gaussiens

Un petit σ produit un filtre gaussien plus "étroit" ou "local". Cela signifie que chaque pixel dans l'image filtrée est principalement influencé par ses pixels voisins immédiats dans l'image d'origine. Cela peut aider à préserver les détails fins mais peut ne pas être aussi efficace pour supprimer le bruit sur de grandes étendues.

Un σ plus grand donne un filtre gaussien plus "large" ou "global". Cela signifie que chaque pixel dans l'image filtrée est influencé par une plus grande zone autour de lui dans l'image d'origine. Cela peut être efficace pour supprimer le bruit sur de plus grandes étendues, mais cela peut également entraîner une perte de détails fins.

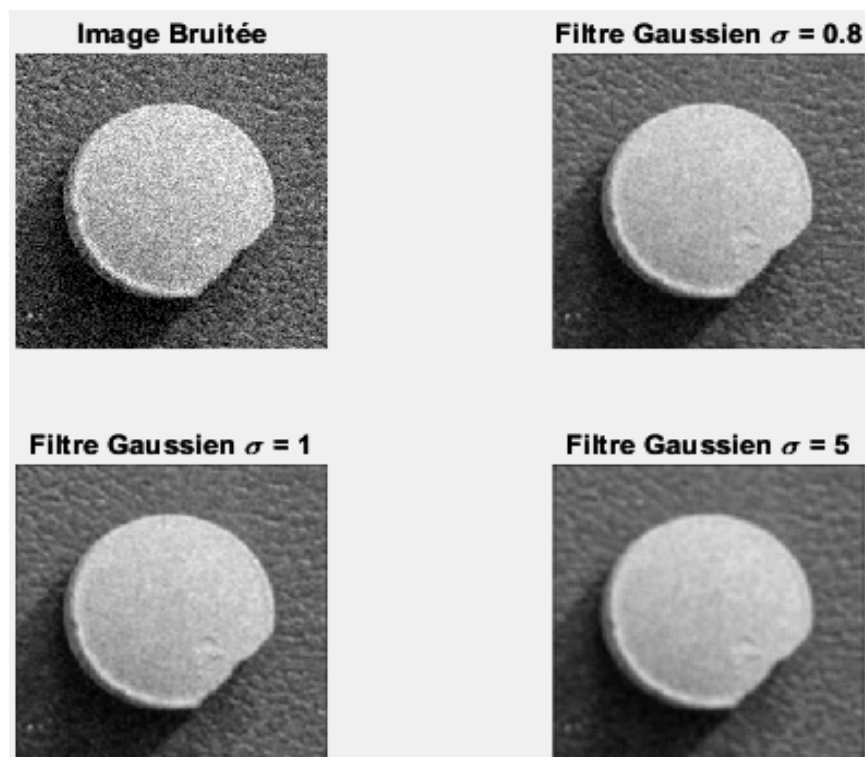


FIGURE 1.35 – Application d'un filtre gaussien

Le tableau (1.1) ci-dessous résume les effets des deux filtres sur une image, en soulignant leurs avantages et inconvénients respectifs.

Aspect	Filtre Moyenneur (Lissage)	Filtre Gaussien
Réduction de bruit	Efficace pour réduire le bruit aléatoire, mais peut pour supprimer des détails importants.	Réduit efficacement le bruit tout en conservant une partie des détails grâce à la pondération gaussienne.
Préservation des détails	Mauvaise, car tous les pixels sont moyennés de manière égale, ce qui tend à flouter l'image .	préserve les détails.
Effet sur les contours	Les contours sont floutés.	Les contours sont mieux préservés.
Effet général	Lissage uniforme, ce qui peut donner une apparence trop floue à l'image.	Lissage plus naturel et moins agressif, préservant mieux les structures importantes de l'image.

TABLE 1.1 – Comparaison des résultats entre le filtre de lissage et le filtre gaussien sur une image

Filtres non linéaires

Un filtre non linéaire ne se base pas sur des opérations linéaires simples comme la convolution, ils prennent en compte les relations complexes entre les pixels pour transformer une image. Cela signifie que sa sortie n'est pas directement proportionnelle à l'entrée. Les filtres non linéaires sont indispensables pour obtenir des images de haute qualité, notamment dans les applications nécessitant la réduction du bruit, la préservation des détails. Ils fournissent des solutions adaptatives et efficaces là où les filtres linéaires atteignent leurs limites. Parmi ces filtres, on trouve le filtre médian.

Celui ci remplace chaque pixel par la valeur médiane des pixels dans son voisinage. Ce type de filtre est particulièrement efficace pour éliminer le bruit impulsionnel (comme le bruit poivre et sel) tout en préservant les bords (Fig.1.36). L'exemple suivant illustre ce type de filtre.

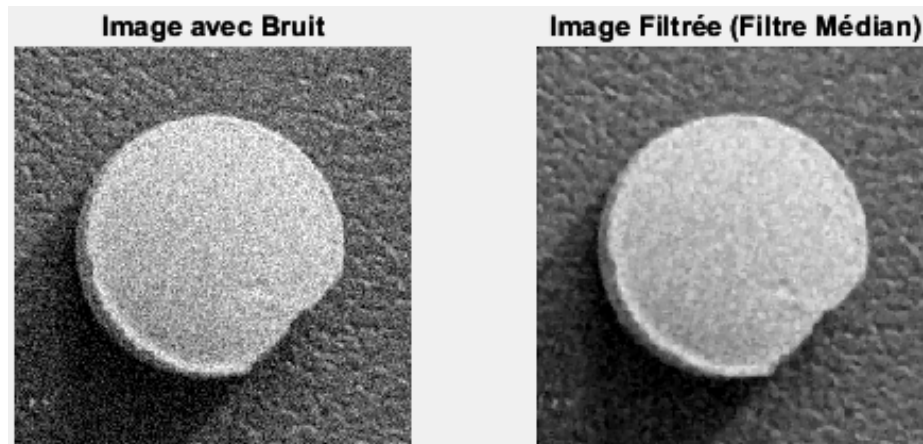
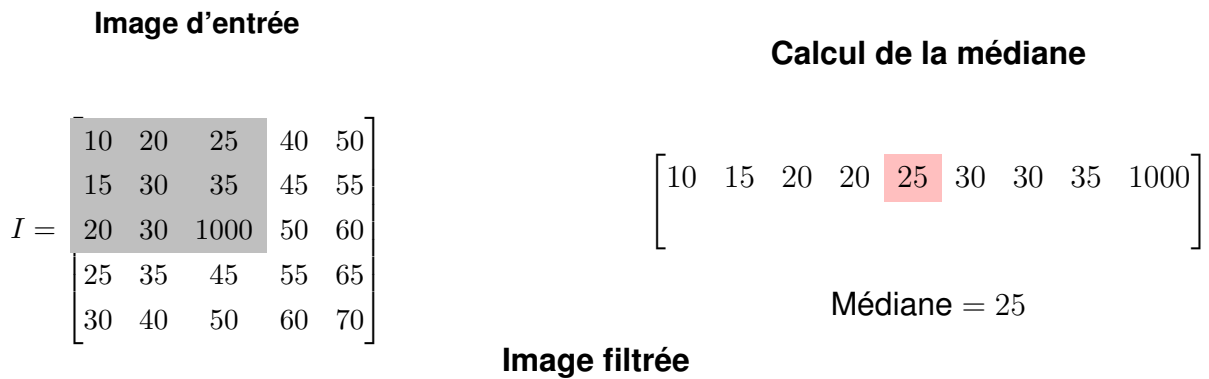


FIGURE 1.36 – Application d'un filtre médian

5.2.2 Rehaussements de contraste (Egalisation d'histogramme)

L'égalisation d'histogramme est une technique utilisée pour améliorer le contraste d'une image. Elle redistribue les valeurs d'intensité des pixels de manière à utiliser au maximum la gamme de valeurs possibles, ce qui permet de rendre les détails d'une image plus visibles.

Les étapes de l'égalisation d'histogramme sont les suivants :

1. Calcul de l'histogramme

Pour une image en niveaux de gris, l'histogramme $h(k)$ est défini comme :

$$h(k) = \text{nombre de pixels ayant le niveau de gris } k$$

avec $k = 0, 1, \dots, L - 1$. L est généralement égal à 256 pour une image codée en 8 bits.

2. Calcul de l'histogramme cumulé

L'histogramme cumulé $H(k)$ est défini comme :

$$H(k) = \sum_{j=0}^k h(j) \quad (1.4)$$

3. Normalisation de l'histogramme cumulé

La fonction de distribution cumulative normalisée (CDF) est donnée par :

$$CDF(k) = \frac{H(k)}{N} \quad (1.5)$$

où N est le nombre total de pixels.

4. Remappage des niveaux de gris

Les nouveaux niveaux de gris s_k sont obtenus par :

$$s_k = \lfloor (L - 1) \cdot CDF(k) \rfloor$$

où L est le nombre total de niveaux de gris possibles.

La figure (1.37) illustre cette opération d'égalisation d'histogramme.

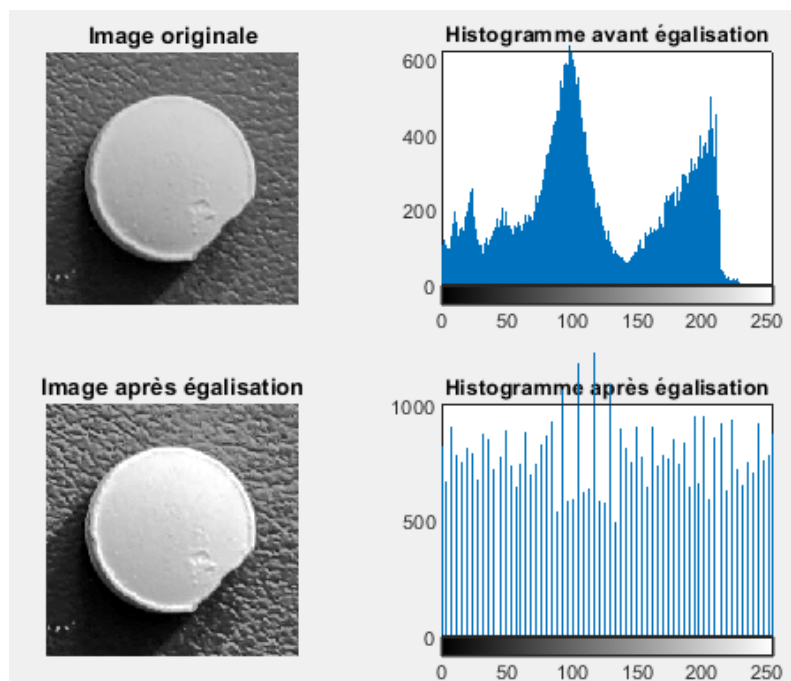


FIGURE 1.37 – Égalisation d'histogramme

5.3 Segmentation

La segmentation consiste à diviser une image en régions ou en objets significatifs. Dans le contexte de la détection des défauts de surface, la segmentation peut être utilisée pour isoler les zones potentiellement défectueuses des autres parties de l'image, facilitant ainsi leur analyse ultérieure.

Il existe plusieurs méthodes de segmentation, qui peuvent être classées en trois catégories principales [9] : La segmentation par seuillage, la segmentation par régions et la segmentation par contours.

Segmentation par seuillage

Le seuillage est la méthode la plus simple de segmentation. Elle consiste à convertir une image en une image binaire, en fonction d'un ou plusieurs seuils.

-Seuillage global (binarisation) : Utilise un seuil unique pour toute l'image. Chaque pixel de l'image est comparé à ce seuil, et il est assigné à l'une des deux valeurs binaires (généralement 0 ou 1) en fonction de cette comparaison.

Le seuillage global est défini par :

$$I_{bin}(x, y) = \begin{cases} 1 & \text{si } I(x, y) \geq T \\ 0 & \text{sinon} \end{cases} \quad (1.6)$$

où T est le seuil.

50	80	90	70	60	⇒	0	1	1	1	1
85	90	100	95	80		1	1	1	1	1
55	70	75	65	50		0	1	1	1	0
45	60	65	55	40		0	1	1	0	0
30	50	55	45	35		0	0	0	0	0
Image Originale					Image Binarisée ($T = 60$)					

FIGURE 1.38 – Exemple de seuillage global appliqué à une image 5x5.

La méthode de seuillage globale la plus connue est la méthode d'Otsu [10].

La figure (1.39) montre le résultat du seuillage sur une image en niveaux de gris.

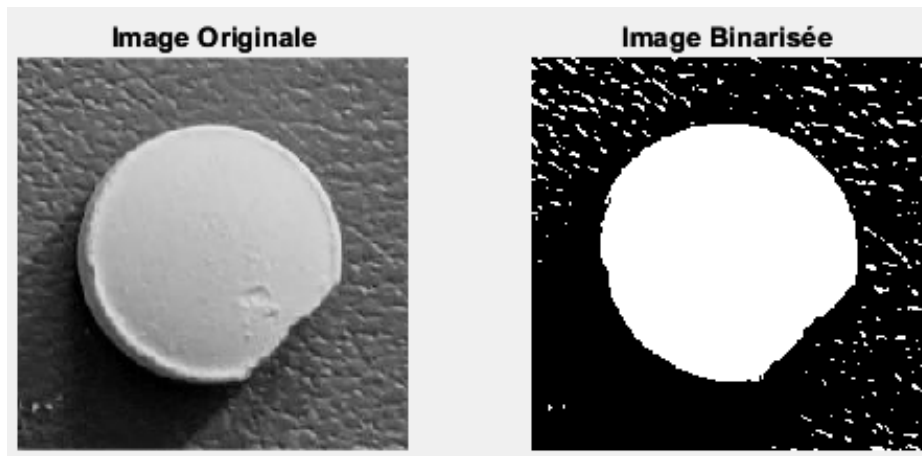


FIGURE 1.39 – Image binarisé

-Seuillage adaptatif : Il consiste à définir un seuil pour chaque pixel de l'image en fonction des valeurs des pixels environnants dans un bloc donné.

La formule pour binariser une image en utilisant le seuillage adaptatif est donnée par :

$$I_{\text{binaire}}(x, y) = \begin{cases} 1 & \text{si } I(x, y) \geq T_{\text{local}}(x, y) \\ 0 & \text{sinon} \end{cases} \quad (1.7)$$

où $T_{\text{local}}(x, y)$ est le seuil calculé localement pour la région autour du pixel (x, y) .

Plusieurs méthodes peuvent être utilisées pour le seuil local $T_{\text{local}}(x, y)$:

La méthode de la moyenne pondérée (Gaussian) est donnée par :

$$T_{\text{local}}(x, y) = \frac{1}{W} \sum_{(i,j) \in \mathcal{N}(x,y)} I(i, j) \cdot G(x - i, y - j) \quad (1.8)$$

où :

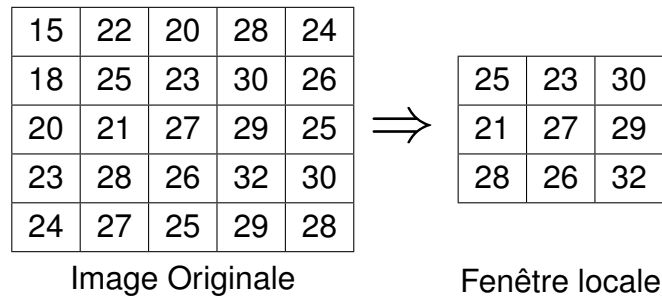
- $I(i, j)$ est l'intensité du pixel à la position (i, j) ,
- $\mathcal{N}(x, y)$ est la région locale centrée autour du pixel (x, y) ,
- $G(x - i, y - j)$ est la valeur de la fonction gaussienne centrée sur (x, y) pour le pixel (i, j) , et W est la somme des poids gaussiens.

La méthode de la médiane locale est donnée par :

$$T_{\text{local}}(x, y) = \text{Med}\{I(i, j) \mid (i, j) \in \mathcal{N}(x, y)\} \quad (1.9)$$

où $\text{Med}\{\cdot\}$ représente l'opération de médiane.

L'exemple ci-dessous montre le calcul du seuil locale par ces deux méthodes



Calcul du Seuil Local par Moyenne Locale :

$$T_{local} = \frac{(25+23+30+21+27+29+28+26+32)}{9} = 26.78.$$

Calcul du Seuil Local par Médiane Locale :

$$T_{local} = 21 \ 23 \ 25 \ 26 \ \underline{27} \ 28 \ 29 \ 30 \ 32 = 27$$

Segmentation par régions

La segmentation par régions est une technique utilisée pour diviser une image en régions homogènes en fonction de certains critères de similarité. Elle se base sur le principe que des pixels appartenant à la même région partagent des propriétés communes, telles que l'intensité, la couleur ou la texture. Cette technique peut être appliquée de différentes manières, notamment par la croissance de régions et la division et fusion de régions.

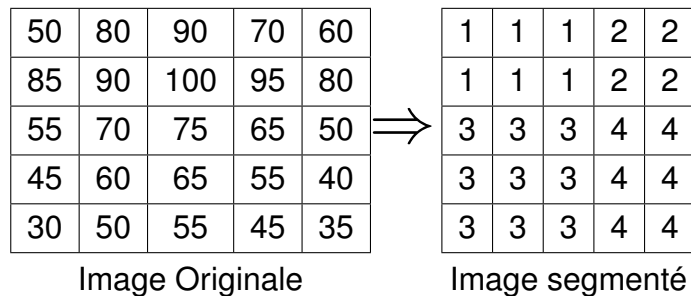


FIGURE 1.40 – Segmentation par régions.

- Croissance de régions (Region Growing) : Cette méthode commence par un ensemble de pixels "graines" (seed points) et ajoute des pixels voisins à la région s'ils sont similaires aux pixels de la région. Les étapes de cette méthode sont les suivantes :

1. **Sélection des graines :** Choisir un ou plusieurs pixels initiaux pour chaque région.

2. **Critère d'adhésion** : Définir une condition de similarité pour inclure un pixel voisin dans la région (par exemple, différence d'intensité inférieure à un certain seuil).
3. **Expansion des régions** : Ajouter les pixels voisins aux régions initiales tant que les critères de similarité sont satisfaits.

- **Division et fusion de régions (Split and Merge)** : Cette méthode divise l'image en régions jusqu'à ce que les régions soient homogènes, puis fusionne les régions adjacentes similaires. Les étapes principales de cette méthode sont :

1. **Division** : Diviser l'image en sous-régions (généralement en utilisant une méthode de quadtree).
Continuer à diviser les sous-régions jusqu'à ce qu'elles deviennent homogènes (satisfont un critère de similarité).
2. **Fusion** : Fusionner les régions voisines si la région fusionnée est homogène selon un critère de similarité.

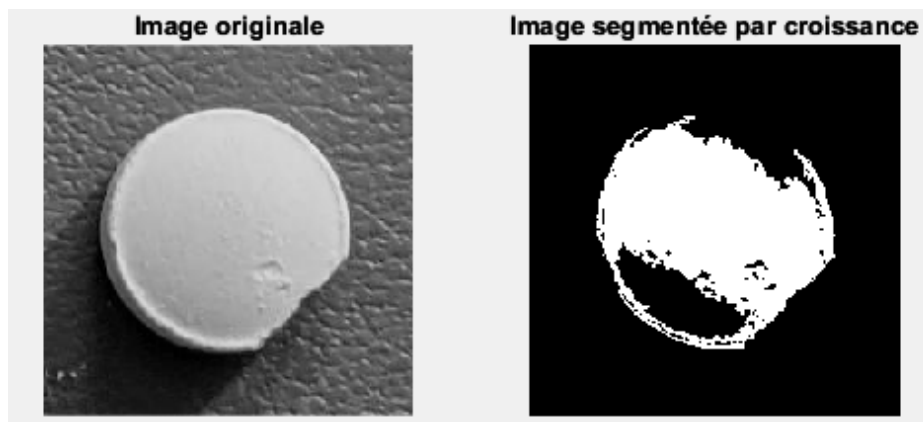


FIGURE 1.41 – Image segmentée par croissance des régions

Segmentation par Contours

La segmentation par contours est une méthode qui vise à délimiter les objets dans une image en détectant leurs contours. Contrairement à la segmentation par régions, qui se base sur l'homogénéité des pixels à l'intérieur des régions, la segmentation par contours se concentre sur la disimilarité ou les transitions abruptes de l'intensité des pixels. Ces variations peuvent être détectées en utilisant des opérateurs dérivés pour calculer les gradients ou les changements de gradient (laplaciens) de l'image.

- **Filtrage par Gradient** : Le gradient d'une image est une mesure de la variation d'intensité des pixels dans l'image. Il indique dans quelle direction et à quel point l'intensité change le plus fortement. Mathématiquement, le gradient est un vecteur constitué de deux composantes : la dérivée partielle en direction $x(G_x)$ et la dérivée partielle en direction $y(G_y)$.

Il existe plusieurs opérateurs pour calculer les gradients, les plus courants sont les opérateurs de Sobel, Prewitt et Roberts. Ces opérateurs sont définis par des matrices (ou masques) utilisées pour effectuer des convolutions avec l'image :

Opérateur de Sobel :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Opérateur de Prewitt :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Opérateur de Roberts :

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Une fois que les gradients G_x et G_y sont calculés pour chaque pixel, la magnitude du gradient G et la direction θ sont calculées comme suit :

La magnitude du gradient est donnée par :

$$G = \sqrt{G_x^2 + G_y^2} \quad (1.10)$$

La direction du gradient est donnée par :

$$\theta = \arctan \left(\frac{G_y}{G_x} \right) \quad (1.11)$$

Pour extraire les contours, on applique un seuil sur la magnitude des gradients. Les pixels où la magnitude du gradient dépasse ce seuil sont considérés comme faisant partie des contours. Cette étape peut être effectuée avec des techniques comme le seuillage simple ou adaptatif. La figure (1.42) montre l'effet du filtre de sobel sur une image.

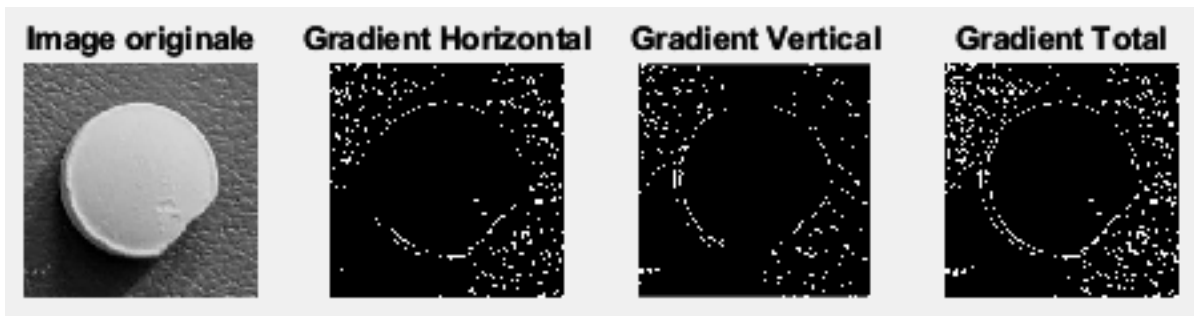


FIGURE 1.42 – Détection de contours à partir du gradient de l'image

- **Filtrage Laplacien** : Le filtrage Laplacien est une technique utilisée pour détecter les contours dans une image. Il se base sur la deuxième dérivée de l'image pour identifier les changements rapides d'intensité. Le filtrage Laplacien détecte les contours en calculant la somme des dérivées secondes de l'image. Il met en évidence les zones de changement rapide d'intensité dans toutes les directions. Contrairement aux gradients, le Laplacien n'est pas directionnel, ce qui permet de détecter les contours indépendamment de leur orientation. Le Laplacien est plus sensible aux variations locales d'intensité, ce qui peut rendre cette méthode plus efficace pour détecter des contours complexes et détaillés.

Le Laplacien d'une fonction $I(x, y)$ est défini comme :

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (1.12)$$

En pratique, le Laplacien est implémenté en utilisant une matrice de convolution telle que :

$$L_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ ou } L_2 = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

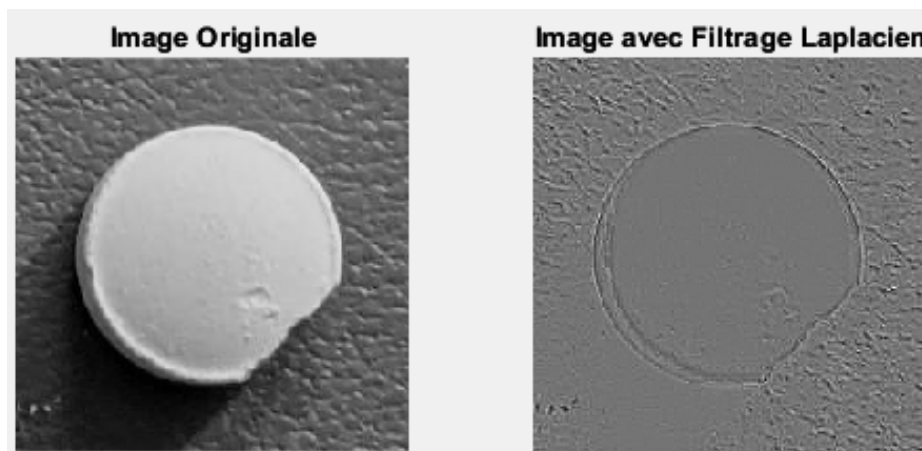


FIGURE 1.43 – Detection de contours à laide de l'opérateur Laplacien

5.4 Post traitement

La segmentation peut produire des images bruitées avec des objets détectés imparfaits (présence de trous, contours non fermés ou déformés,...). Pour améliorer la forme des objets détectés, on applique des opérations de post traitement basées sur la morphologie mathématique.

5.4.1 Morphologie Mathématique

La morphologie mathématique est une technique de traitement d'images qui traite les formes et les structures au sein de l'image. Elle est utilisée pour diverses opérations de post-traitement, telles que le lissage des contours, le remplissage des trous, et la suppression du bruit.

Éléments Structurants : La morphologie mathématique se base sur des opérations impliquant des éléments structurants. Ces derniers sont des matrices binaires dont la forme et la taille dépendent des objets que l'on souhaite détecter ou modifier dans une image. Les éléments structurants peuvent avoir différentes formes, telles que des carrés, des rectangles, des cercles, des croix, etc. La taille de l'élément structurant est également ajustable en fonction de la résolution de l'image et de la complexité des objets à traiter (Fig.1.44). Par exemple, pour détecter de petits détails, on utilise souvent des éléments structurants de petite taille, tandis que pour des objets plus grands, on peut utiliser des éléments structurants plus grands. Le principe de la morphologie mathématique est de comparer les parties de l'image avec les éléments structurants.

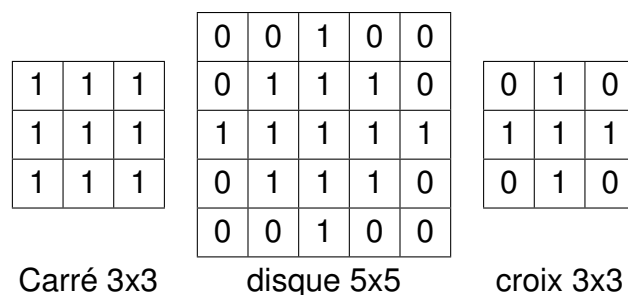


FIGURE 1.44 – Exemples d'éléments structurants.

Opérations morphologiques de base

- **Érosion** : L'érosion rétrécit les formes dans une image. Elle consiste à déplacer l'élément structurant E sur l'image et à remplacer chaque pixel de l'image I par la valeur minimale des pixels couverts par l'élément structurant (Fig.1.45) :

$$I \ominus E = \{z \mid (E)_z \subseteq I\} \quad (1.13)$$

$(E)_z$ représente la translation de E par z .

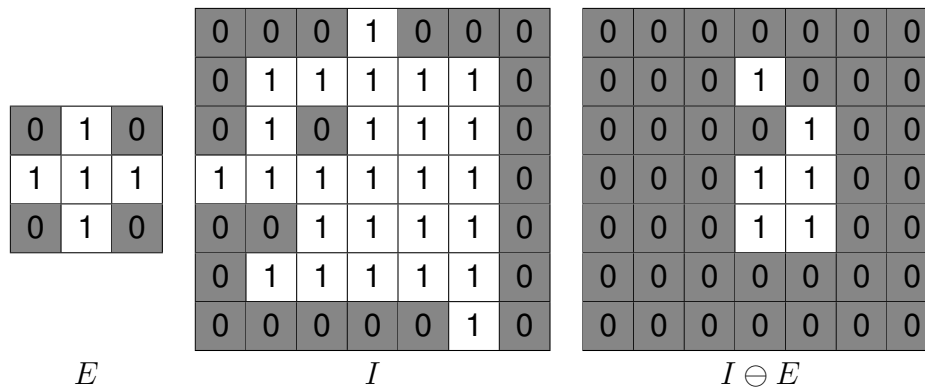


FIGURE 1.45 – Exemple d'érosion avec un élément structurant croix 3x3

- **Dilatation** : L'opération de dilatation élargit les formes dans une image. Elle consiste à déplacer l'élément structurant sur l'image et à remplacer chaque pixel de l'image par la valeur maximale des pixels couverts par l'élément structurant (Fig.1.46) :

$$I \oplus E = \{z \mid (E)_z \cap I \neq \emptyset\} \tag{1.14}$$

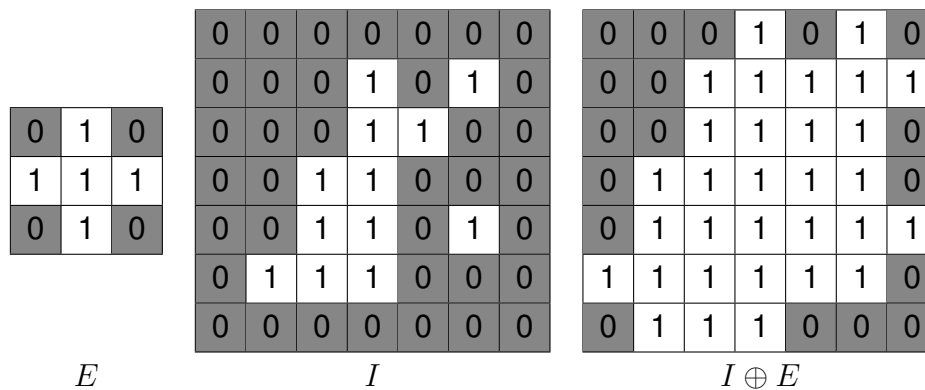


FIGURE 1.46 – Exemple de dilatation avec un élément structurant croix 3x3

- **Ouverture** : L'ouverture est une érosion suivie d'une dilatation, utile pour éliminer les petites structures, briser les connexions fines et lisser les contours (Fig.1.47) :

$$I \circ E = (I \ominus E) \oplus E \tag{1.15}$$

- **Fermeture** : La fermeture est une dilatation suivie d'une érosion, utilisée pour remplir les petits trous et fusionner les structures proches (Fig.1.47) :

$$I \bullet E = (I \oplus E) \ominus E \tag{1.16}$$

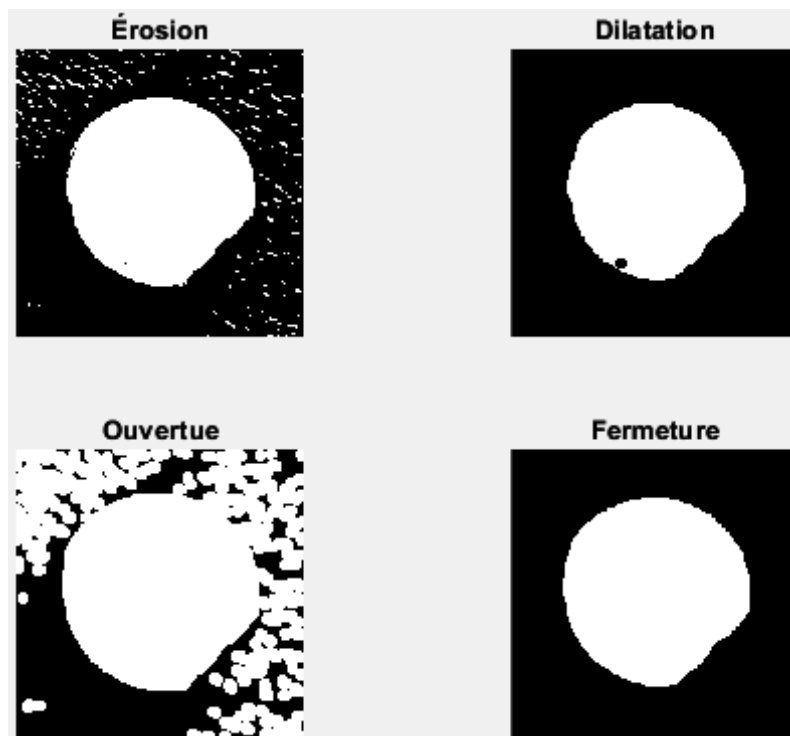


FIGURE 1.47 – Application des opérations morphologiques de base

5.5 Extraction de caractéristiques

L'objectif principal de l'extraction des caractéristiques dans la classification des défauts de surface est de caractériser les défauts par un ensemble d'attributs. Ces attributs doivent être très pertinents afin de permettre de différencier efficacement les différents types de défauts.

5.5.1 Caractéristiques basées sur la texture

Les caractéristiques de texture sont des descripteurs quantitatifs qui capturent les variations d'intensité et les motifs présents dans une image. Elles sont utilisées pour analyser la structure de la surface et pour différencier les régions homogènes des régions hétérogènes. La technique basée sur les matrices de cooccurrence reste la plus populaire. Elle est définie par la probabilité conjointe des niveaux de gris $p(i, j)$, égale à l'ensemble des pixels ayant un niveau de gris i séparés d'une distance d dans la direction θ des pixels ayant un niveau de gris j (Fig.1.48). Plusieurs attributs peuvent être déterminés à partir d'une matrice de cooccurrence.

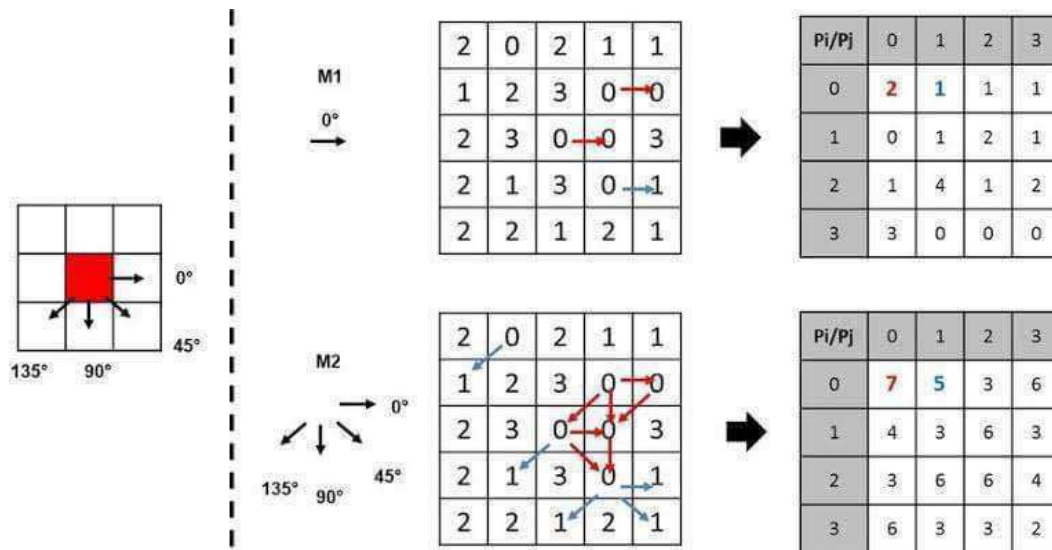


FIGURE 1.48 – Illustration d’une matrice de cooccurrence

Exemples d’attributs de Texture

- **Contraste** : Mesure la variation d’intensité entre un pixel et ses voisins. Des valeurs élevées indiquent de grandes différences d’intensité. Il permet de détecter les contours et les structures marquées dans l’image, utiles pour identifier les défauts nets et bien définis.

$$\text{Contraste} = \sum_{i,j} (i - j)^2 \cdot p(i, j) \tag{1.17}$$

- **Homogénéité** : Indique la similarité des niveaux de gris des pixels. Des valeurs élevées signifient que les pixels voisins ont des valeurs d’intensité similaires. Elle est utile pour identifier les surfaces uniformes et lisses, et aide à distinguer les zones homogènes des zones rugueuses ou texturées.

$$\text{Homogénéité} = \sum_{i,j} \frac{1}{1 + |i - j|} \cdot p(i, j) \tag{1.18}$$

- **Entropie** : Mesure le désordre ou la complexité d’une texture. Une entropie élevée signifie une grande variabilité des niveaux de gris. Elle est souvent associée à des défauts ou des anomalies irrégulières.

$$\text{Entropie} = - \sum_{i,j} p(i, j) \cdot \log(p(i, j)) \tag{1.19}$$

- **Énergie** : Représente l’uniformité de la texture. Une énergie élevée indique une texture homogène. Elle Permet d’identifier les régions homogènes et structurées, opposées aux régions texturées et hétérogènes, ce qui est crucial pour la segmentation des surfaces.

$$\text{Énergie} = \sum_{i,j} [p(i, j)]^2 \quad (1.20)$$

- **Corrélation** : Mesure à quel point un pixel est corrélé à ses voisins sur une certaine distance et orientation. Elle aide à détecter les motifs répétitifs et les alignements dans l'image, ce qui est utile pour la reconnaissance de structures et de motifs réguliers.

$$\text{Corrélation} = \frac{\sum_{i,j} (i \cdot j \cdot p(i, j)) - \mu_x \cdot \mu_y}{\sigma_x \cdot \sigma_y} \quad (1.21)$$

— μ_x, μ_y : moyennes des niveaux de gris des pixels.

— σ_x, σ_y : écarts-types.

5.5.2 Caractéristiques basées sur la forme

Les caractéristiques de forme sont des mesures quantitatives qui décrivent la géométrie et la structure des objets (défauts) dans une image. Elles capturent des informations telles que la taille, la compacité, la circularité, l'aspect ratio, les angles, les moments géométriques. Voici quelques exemples de caractéristiques de forme couramment utilisées :

- **Taille** : La taille d'un objet, généralement mesurée en pixels, représente sa dimension dans l'image.

- **Compacité** : La compacité mesure à quel point un objet est compact ou étalé. Elle est souvent définie comme le rapport entre l'aire de l'objet et le carré de son périmètre.

$$\text{Compacité} = \frac{\text{Aire}}{\text{Périmètre}^2} \quad (1.22)$$

- **Circularité** : La circularité mesure à quel point un objet ressemble à un cercle. Plus la circularité est proche de 1, plus l'objet est circulaire.

$$\text{Circularité} = \frac{4\pi \times \text{Aire}}{\text{Périmètre}^2} \quad (1.23)$$

- **Aspect Ratio** : L'aspect ratio est le rapport entre la longueur et la largeur d'un objet. Il indique la forme générale de l'objet, comme s'il est allongé ou comprimé.

$$\text{Aspect Ratio} = \frac{\text{Longueur}}{\text{Largeur}} \quad (1.24)$$

- **Moments géométriques** : Les moments géométriques captent des informations sur la distribution des pixels dans l'objet par rapport à son centre de masse. Ils sont utilisés pour calculer des caractéristiques telles que le centre de gravité, l'inertie, la forme des contours, etc.

5.5.3 Caractéristiques basées sur la couleur :

Les caractéristiques de couleur se réfèrent aux propriétés et aux attributs associés aux couleurs d'une image. Elles sont utilisées pour décrire la distribution, la variabilité et d'autres aspects liés à la couleur dans une image.

Exemples de caractéristiques basées sur la couleur :

- **Histogrammes de couleurs** : Ils représentent la répartition des niveaux de couleur dans une image, fournissant des informations sur la distribution des couleurs dominantes et leur fréquence.

- **Espaces colorimétriques** : Les espaces colorimétriques tels que RVB (rouge, vert, bleu), HSV (teinte, saturation, valeur), LAB, etc., fournissent des informations sur les composantes de couleur comme la luminosité, la saturation, la teinte, etc. et permettent la manipulation et l'analyse de la couleur dans différents contextes.

5.6 Classification (décision) :

Les systèmes de détection des défauts par vision artificielle utilisent des algorithmes d'apprentissage automatique, tels que :

5.6.1 Machines à Vecteurs de Support (SVM)

Les SVM (Support Vector Machines) sont des outils puissants en classification, fréquemment utilisés pour détecter les défauts de surface [11]. Leur fonction principale est de trouver un hyperplan de séparation optimal dans l'espace des caractéristiques des données de surface. Cet hyperplan agit comme une frontière de décision qui sépare efficacement les données de référence des données présentant des irrégularités, facilitant ainsi l'identification et la localisation des défauts sur la surface. En construisant cet hyperplan de manière appropriée lors de l'entraînement du modèle SVM, on peut obtenir une séparation nette entre les données de base et les données contenant des anomalies.

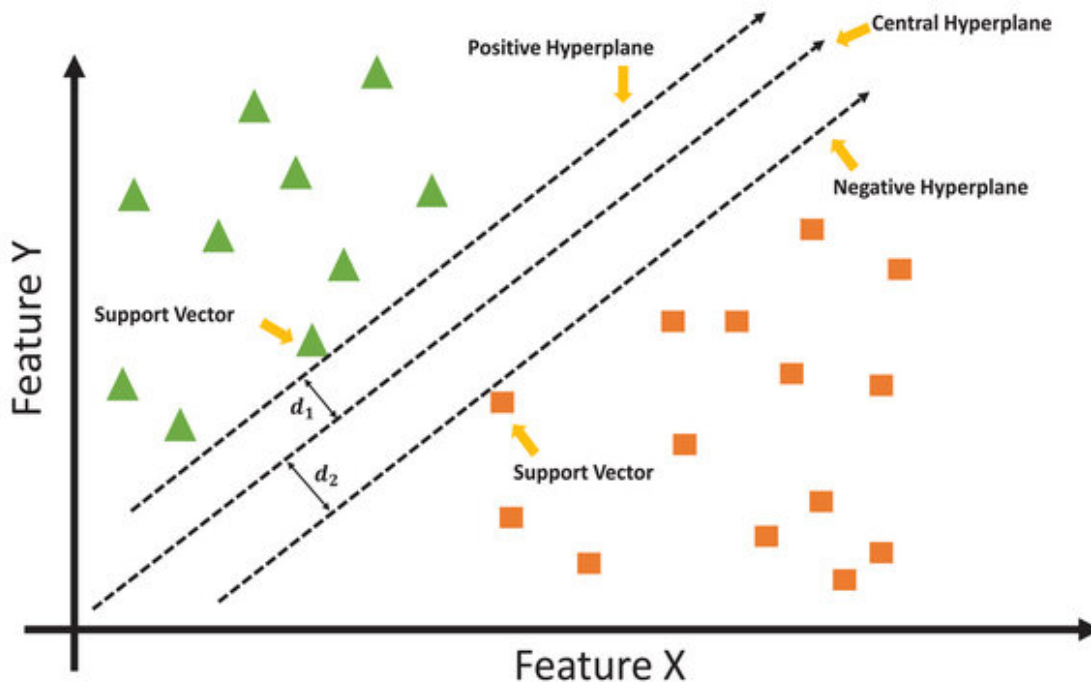


FIGURE 1.49 – Machines à Vecteurs de Support

5.6.2 Arbres de Décision et leurs Ensembles (Random Forest, Gradient Boosting)

Les arbres de décision et leurs ensembles sont également largement utilisés pour la détection de défauts de surface. Les arbres de décision individuels peuvent être utilisés pour classifier les données de surface en fonction de certaines caractéristiques, ce qui permet de détecter les anomalies [11]. Les méthodes d'ensemble telles que Random Forest ou Gradient Boosting améliorent la robustesse du modèle en combinant les prédictions de plusieurs arbres de décision. En utilisant ces ensembles, le modèle peut capturer des relations complexes entre les caractéristiques des données de surface, ce qui améliore sa capacité à détecter une variété de défauts avec une plus grande précision.

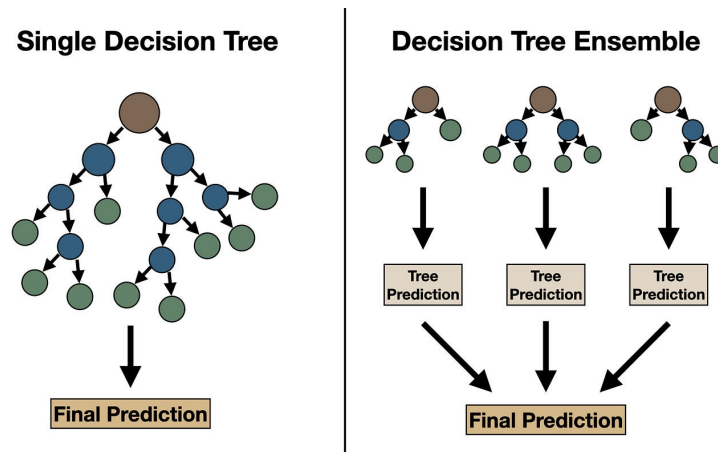


FIGURE 1.50 – Arbres de Décision et leurs Ensembles

5.6.3 Réseaux de neurones convolutifs (CNNs)

Les CNNs intègrent l'extraction des caractéristiques et la classification en un seul processus (fig1.51). Cela simplifie la détection des défauts et réduit le besoin d'intervention humaine. Les CNNs apprennent automatiquement à identifier les caractéristiques pertinentes, comme les bords et les textures, et à classifier les images en fonction de la présence ou non de défauts.

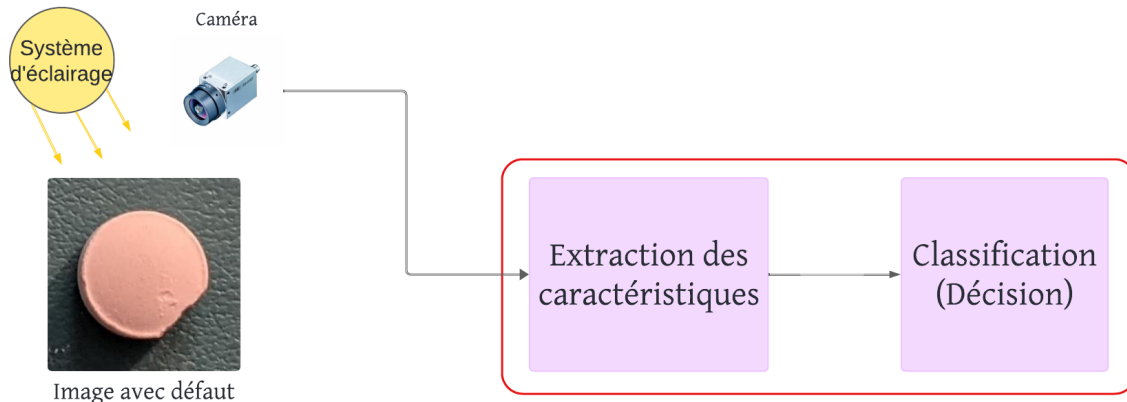


FIGURE 1.51 – Processus de Détection de Défauts de Surface Utilisant des Réseaux de Neurones Convolutifs

Enfin, en fonction des résultats de la classification, le système décidera si l'objet inspecté est défectueux ou non défectueux. Cette décision peut impliquer l'activation d'une alarme, l'éjection du produit ou la génération d'un rapport pour son analyse postérieure.

6 Avantages de l'utilisation de systèmes de vision artificiel pour la détection automatisée

1. **Précision accrue** : Les systèmes de vision artificielle utilisent des algorithmes avancés pour détecter les défauts avec une grande précision, réduisant ainsi les erreurs de détection.
2. **Rapidité** : Les processus d'inspection automatisés sont généralement plus rapides que les méthodes manuelles, ce qui permet d'augmenter la productivité et d'accélérer les opérations de contrôle qualité.
3. **Fiabilité** : En éliminant la dépendance à l'égard de l'observation humaine, les systèmes de vision artificielle offrent une fiabilité constante dans la détection des défauts, réduisant ainsi les risques d'erreur humaine.
4. **Adaptabilité** : Les systèmes de vision artificielle peuvent être programmés pour détecter une grande variété de défauts et peuvent être facilement adaptés à différents

environnements de production et types de matériaux.

5. **Réduction des coûts** : En automatisant le processus de détection des défauts, les entreprises peuvent réduire les coûts liés à la main-d'œuvre et augmenter l'efficacité opérationnelle.
6. **Analyse avancée des données** : Les systèmes de vision artificielle peuvent collecter et analyser de grandes quantités de données, ce qui permet une surveillance continue et une optimisation des processus de production.

7 Conclusion

En conclusion, ce chapitre a permis de présenter une vue d'ensemble des méthodes de détection des défauts de surface, en mettant en lumière tant les approches traditionnelles que les techniques modernes de vision artificielle. Nous avons commencé par définir les défauts de surface, leurs classifications et souligné leur importance pour assurer la qualité et la conformité dans divers secteurs industriels. Les méthodes traditionnelles de détection, comme l'inspection visuelle humaine et les techniques d'imagerie conventionnelles, bien qu'utiles, ont révélé des limitations importantes, notamment en termes de subjectivité, de coûts élevés et de faible efficacité pour des applications à grande échelle. Ce constat a ouvert la voie à l'adoption de solutions plus avancées, dont les systèmes automatisés de vision artificielle.

Nous avons ensuite décrit les différentes étapes de ces systèmes, depuis l'acquisition des données jusqu'à la classification des défauts de surface. Ces méthodes modernes offrent des avantages indéniables en matière de précision et de rapidité, contribuant ainsi à une meilleure fiabilité des processus de contrôle qualité.

Chapitre 2 :

Réseaux De Neurones Convolutifs (CNNs)

Chapitre 2

Réseaux De Neurones Convolutifs (CNNs)

1 Introduction

L'intelligence artificielle, notamment à travers l'utilisation de réseaux de neurones convolutifs (CNNs), a marqué un tournant décisif dans la manière dont les défauts de surface sont détectés et traités. Cette technologie révolutionnaire a permis d'automatiser et d'améliorer significativement les processus de contrôle qualité, offrant ainsi des solutions plus efficaces et précises dans une variété de domaines industriels.

Un réseau de neurones est une structure mathématique et informatique conçue pour imiter le fonctionnement du système nerveux biologique, notamment celui du cerveau humain. Ils sont largement employés dans divers domaines tels que la classification d'images, la reconnaissance vocale, la traduction automatique, la prédiction de séries temporelles, la recommandation de produits, et bien d'autres. Leur force réside dans leur capacité à modéliser des données complexes et non linéaires. Ainsi, ils sont devenus des outils incontournables dans de nombreux champs de l'intelligence artificielle et de l'apprentissage automatique.

Dans ce chapitre, nous aborderons les réseaux de neurones convolutifs (CNNs). Nous nous pencherons sur leur fonctionnement interne, en examinant comment les CNNs sont structurés et comment ils traitent les données visuelles. Mais avant cela, il est primordial de jeter un regard approfondi sur les réseaux de neurones dans leur ensemble. Cela implique de comprendre la structure et le fonctionnement des neurones, les connexions synaptiques qui les relient, ainsi que l'organisation en couches qui caractérise ces réseaux. En effet, en comprenant les bases importantes des réseaux de neurones, nous préparerons le terrain pour comprendre plus facilement comment les réseaux de neurones convolutifs fonctionnent.

2 Neurones biologiques/Neurones artificiels

2.1 Description d'un Neurone biologique

Un neurone biologique est composé de quatre parties principales, comme le montre la figure (2.1) :

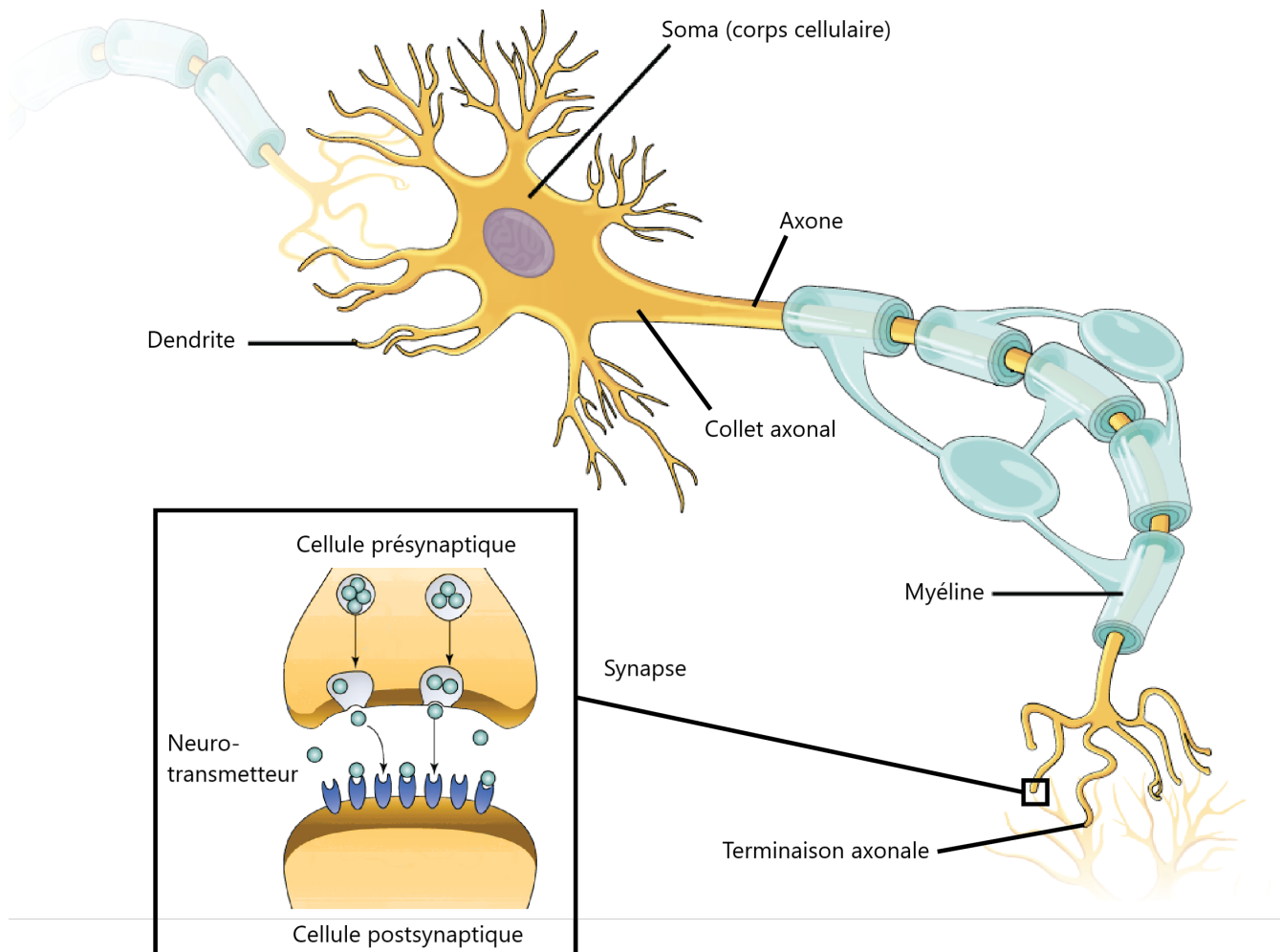


FIGURE 2.1 – Neurone biologique [12]

1. **Corps cellulaire** : Également appelé soma, est la partie principale d'un neurone où se trouve le noyau. C'est comme le cerveau de la cellule, où les signaux reçus des dendrites sont traités avant d'être transmis à d'autres neurones.
2. **Dendrites** : Les dendrites sont les branches qui sortent du corps cellulaire d'un neurone. Elles agissent comme des antennes pour capter les signaux provenant d'autres neurones ou des cellules sensorielles. En résumé, les dendrites reçoivent les informations et les transmettent au corps cellulaire pour traitement.

3. **Axone** : L'axone est le câble de communication d'un neurone. Il part du corps cellulaire du neurone et transmet les signaux électriques vers d'autres neurones ou vers les muscles et les glandes. En d'autres termes, l'axone agit comme une ligne de communication pour transmettre les informations à d'autres parties du corps.
4. **Synapses** : une synapse est la région d'interaction entre deux cellules nerveuses qui permet le passage d'un signal. Quand un neurone veut envoyer un message à un autre, il envoie un signal électrique jusqu'à la synapse, ce signal électrique se transforme en un signal chimique spécial appelé neurotransmetteur. Ce neurotransmetteur traverse ensuite la synapse et va activer le neurone suivant, lui transmettant ainsi le message.

Le processus de transmission de l'information entre les neurones se déroule en plusieurs étapes :

- Signal électrique** : Lorsqu'un neurone est activé, il produit un signal électrique appelé potentiel d'action. Ce potentiel d'action se propage le long de l'axone du neurone.

- Libération de neurotransmetteurs** : Arrivée à la synapse, le potentiel d'action déclenche la libération de neurotransmetteurs depuis la terminaison de l'axone du neurone présynaptique.

- Transmission chimique** : Les neurotransmetteurs se diffusent dans l'espace synaptique et se fixent aux récepteurs spécifiques sur la membrane du neurone postsynaptique.

- Déclenchement d'un nouveau potentiel d'action** : La liaison des neurotransmetteurs aux récepteurs déclenche une modification de la perméabilité de la membrane du neurone postsynaptique, ce qui peut provoquer la génération d'un nouveau potentiel d'action si le seuil d'excitation est atteint. Le "seuil d'excitation" est le niveau de stimulation nécessaire pour déclencher une réponse électrique dans un neurone. Chaque neurone a un seuil d'excitation spécifique, qui représente le niveau minimum de dépolarisation (La dépolarisation se produit lorsque la charge électrique à travers la membrane change, généralement en devenant moins négative à l'intérieur du neurone) nécessaire pour générer un potentiel d'action.

- Propagation du signal** : Si un nouveau potentiel d'action est généré dans le neurone postsynaptique, le processus se répète, et le signal est transmis à d'autres neurones.

2.2 Neurones artificiels

Les neurones biologiques ont été une source d'inspiration essentielle pour la création des réseaux de neurones artificiels. Les chercheurs ont observé le fonctionnement complexe des

neurones dans le cerveau humain et ont cherché à reproduire ces mécanismes dans des modèles informatiques. Les premiers modèles de neurones artificiels ont été proposés par Warren McCulloch et Walter Pitts en 1943 [13], qui ont mis en évidence les relations entre les entrées et les sorties des neurones biologiques. Parmi les principales caractéristiques des neurones biologiques reprises dans les modèles artificiels, on trouve :

Structure : Les réseaux de neurones artificiels sont conçus pour imiter la structure des neurones biologiques, avec des entrées, des poids synaptiques et des fonctions d'activation.

Transmission de l'information : Comme les neurones biologiques transmettent des signaux électriques et chimiques le long des axones et à travers les synapses, les réseaux de neurones artificiels utilisent des calculs mathématiques pour simuler ce processus.

Apprentissage : Les neurones biologiques peuvent s'adapter et changer leur comportement en fonction des expériences passées, un processus appelé plasticité synaptique. De même, les réseaux de neurones artificiels sont dotés de capacités d'apprentissage qui leur permettent de s'ajuster et d'améliorer leurs performances au fil du temps.

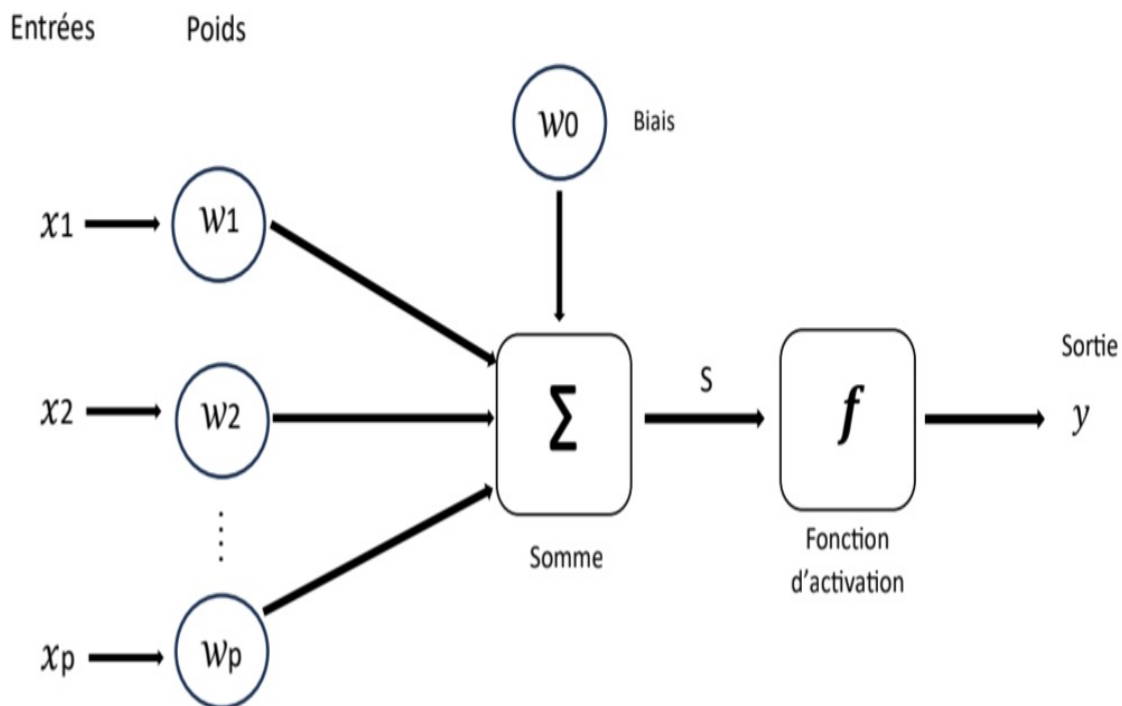


FIGURE 2.2 – Structure d'un neurone artificiel

Un neurone artificiel est donc un modèle simplifié d'un neurone biologique. Il est décrit dans la figure (2.2) et comprend :

Entrées : Un neurone artificiel reçoit en entrée un ensemble de caractéristiques ou de variables, représentées sous forme de vecteur $X = [x_1, x_2, \dots, x_p]$

Poids : Chaque entrée est associée à un poids correspondant w_1, w_2, \dots, w_p . Ces poids synaptiques représentent l'importance relative des différentes caractéristiques pour la sortie du neurone. Ils peuvent être ajustés pendant l'apprentissage pour améliorer les performances du modèle.

Biais : C'est un paramètre supplémentaire ajouté à chaque neurone, en plus des poids associés aux entrées. Le biais permet d'ajuster la sortie d'un neurone même lorsque toutes les entrées sont nulles. Il agit comme un terme constant qui permet de décaler la sortie du neurone, facilitant ainsi l'apprentissage et l'ajustement des prédictions du réseau de neurones.

Somme pondérée : Le neurone effectue une opération fondamentale qui consiste à calculer la somme des produits des entrées par leurs poids associés. Cette opération qui permet de donner plus d'importance à certaines entrées par rapport à d'autres, en fonction des poids assignés, est définie par :

$$s = \sum_{i=1}^p (x_i \cdot w_i) + w_0 \quad (2.1)$$

Fonction d'activation : Le neurone artificiel applique une fonction d'activation sur la somme pondérée afin de déterminer si le neurone doit être activé ou non. Cette fonction introduit de la non-linéarité dans le modèle, ce qui lui permet d'apprendre des relations plus complexes dans les données elle est exprimée comme suite :

$$y = f(s) \quad (2.2)$$

Le tableau (2.1) regroupe quelques fonctions d'activation les plus couramment utilisées [14][15].

Les allures de ces fonctions sont affichées sur la figure (2.3).

Fonction d'activation	Fonction mathématique	Description	Dérivée
Fonction d'activation linéaire	$f(x) = x$	Une fonction qui conserve la même valeur que l'entrée.	$f'(x) = 1$
Fonction d'activation sigmoïde	$f(x) = \frac{1}{1+e^{-x}}$	Une courbe en forme de S, produisant des valeurs dans l'intervalle $[0, 1]$.	$f'(x) = \frac{e^{-x}}{(1+e^{-x})^2}$
Tangente hyperbolique	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	Une courbe en forme de S, produisant des valeurs dans l'intervalle $[-1, 1]$, centrée autour de zéro.	$f'(x) = 1 - \left(\frac{e^x + e^{-x}}{e^x - e^{-x}}\right)^2$
Rectified Linear Unit (ReLU)	$f(x) = \max(0, x)$	Une fonction qui renvoie x si x est positif et 0 sinon.	$f'(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$
Leaky ReLU	$f(x) = \max(0.01x, x)$	Une variante de ReLU qui introduit une pente légèrement négative pour les valeurs négatives de x .	$f'(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0.01 & \text{sinon} \end{cases}$
ReLU paramétrique	$f(x) = \max(\alpha x, x)$	Une variante de ReLU où la pente pour les valeurs négatives peut être un paramètre (α).	$f'(x) = \begin{cases} 1 & \text{si } x > 0 \\ \alpha & \text{sinon} \end{cases}$
Unité exponentielle linéaire (ELU)	$f(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha(e^x - 1) & \text{sinon} \end{cases}$	Une fonction qui lisse les parties négatives de la fonction ReLU, avec un paramètre α pour contrôler la pente pour les valeurs négatives de x .	$f'(x) = \begin{cases} 1 & \text{si } x > 0 \\ \alpha e^x & \text{sinon} \end{cases}$

TABLE 2.1 – Fonctions d'activation dans les réseaux de neurones avec leurs dérivées

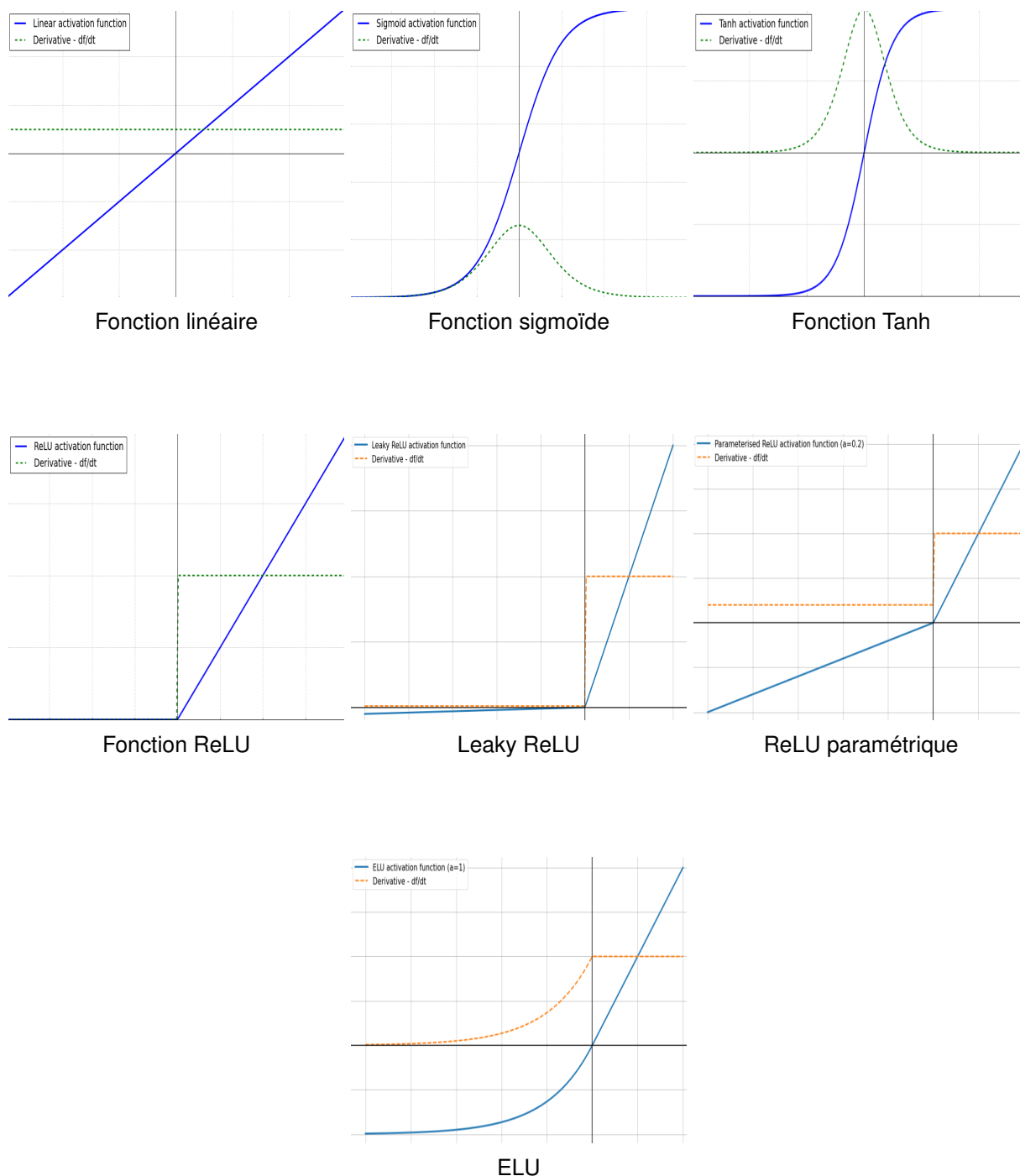


FIGURE 2.3 – Les courbes des fonctions d’activation

En résumé, le tableau 2.2 donne les similarités entre les neurones biologiques et neurones artificiels :

Caractéristique	Neurones Biologiques	Neurones Artificiels
Structure	Synapses. Axones. Dendrites. Soma.	Poids de connexions. Signal de sortie. Signal d'entrée. Fonction d'activation.
Fonctionnement	Utilisent des signaux électriques et chimiques pour communiquer entre eux.	Effectuent des calculs mathématiques sur les données qu'ils reçoivent en entrée.
Plasticité	Adaptation aux expériences.	Ajustement des connexions.
Taille et complexité	Très complexes.	Variables en fonction de la conception du réseau.
Durabilité	Vulnérables aux maladies et au vieillissement.	Sensibles aux pannes.
Vitesse de traitement	Relativement lent, les signaux électriques et chimiques dans le cerveau se propagent à des vitesses limitées.	Rapide, les neurones artificiels effectuent des calculs mathématiques à des vitesses élevées.

TABLE 2.2 – Comparaison entre les neurones biologiques et artificiels

3 Architecture des réseaux de neurones

Un réseau de neurones est une association de plusieurs neurones connectés entre eux. Les réseaux de neurones artificiels sont souvent organisés en plusieurs couches de neurones inter-connectés. Chaque couche est constituée d'un ensemble de neurones, qui sont reliés aux neurones de la couche précédente et/ou suivante formant ainsi un réseau de connexions. Les informations sont transmises à travers ces connexions. L'assemblage des neurones permet de multiplier leurs capacités d'apprentissage. Les couches sont généralement organisées en trois types (Fig.2.4) :

Couche d'entrée :

Cette première couche reçoit les données brutes en entrée dans le réseau. Chaque nœud de cette couche représente une caractéristique ou une variable des données d'entrée.

Couches cachées :

Elles sont appelées "cachées" parce qu'elles ne sont pas directement accessibles depuis l'extérieur du réseau, ce sont des couches intermédiaires dans un réseau de neurones. Chaque couche cachée analyse et extrait des caractéristiques des données d'entrée, les rendant plus compréhensibles pour la couche de sortie. Le nombre de couches cachées varie en fonction de la complexité du problème et des ressources disponibles.

Couche de sortie :

La couche de sortie est la dernière partie d'un réseau de neurones où les résultats finaux sont produits. En utilisant les informations traitées par les couches précédentes, la couche de sortie produit les prédictions finales du modèle, telles que les classes dans un problème de classification. En résumé, elle permet au réseau de générer des résultats basés sur les données d'entrée et les modèles appris.

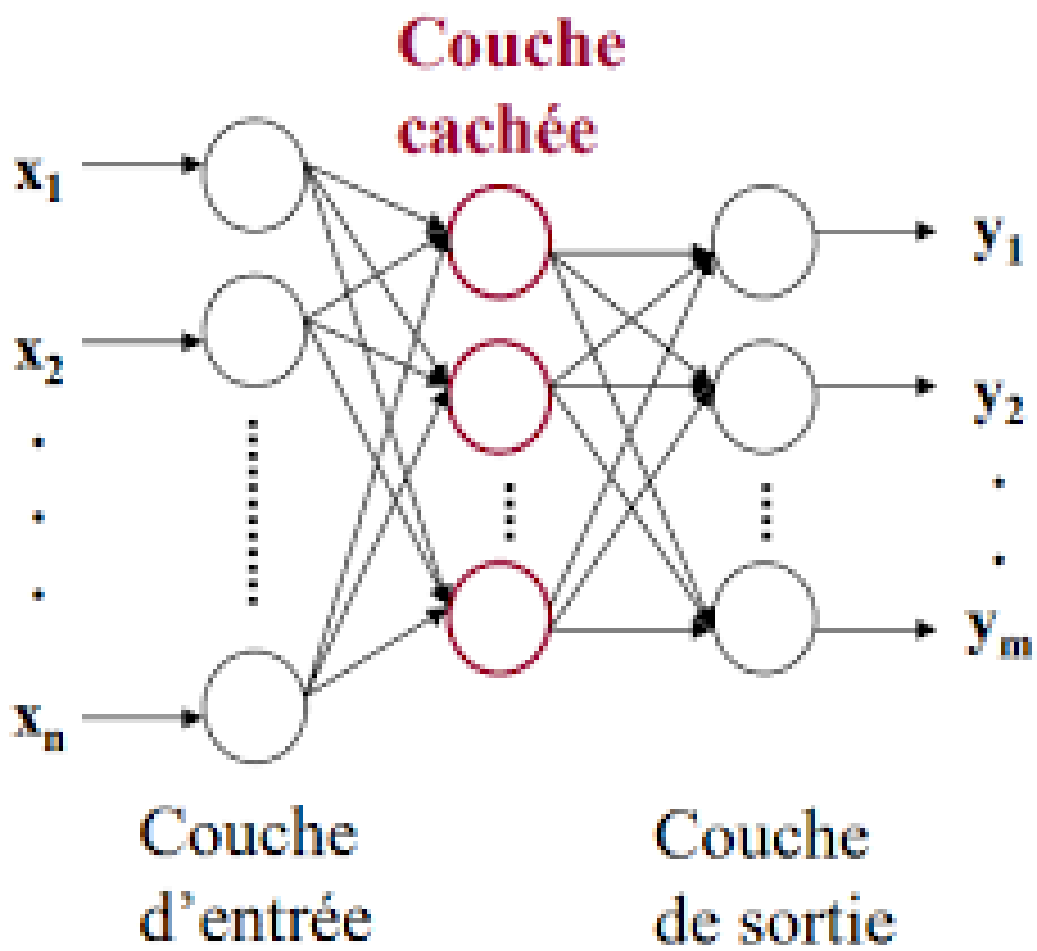


FIGURE 2.4 – Architecture d'un réseau de neurones

La manière dont les neurones sont connectés entre eux est d'une importance capitale. On peut classer les réseaux de neurones en deux grandes catégories en fonction de cette structure

[16][17] :

- Les réseaux de neurones dynamiques (ou récurrents, ou bouclés) (Fig.2.5).
- Les réseaux de neurones statiques (ou acycliques, ou non bouclés) (Fig.2.6).

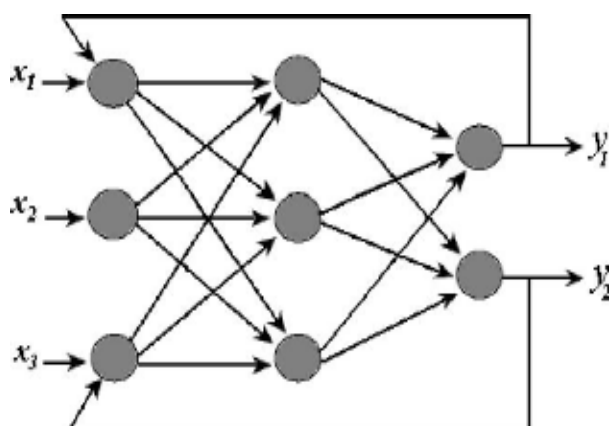


FIGURE 2.5 – Réseau bouclé

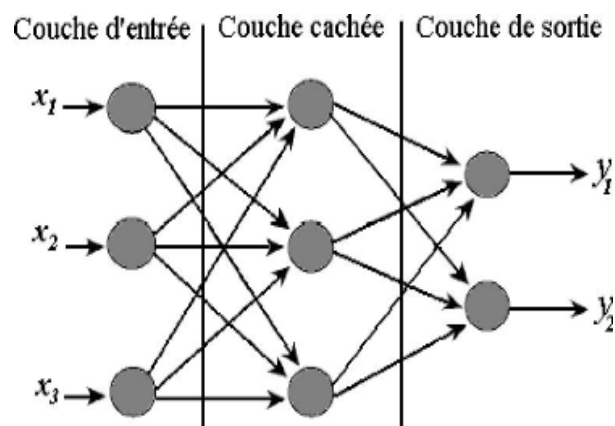


FIGURE 2.6 – Réseau non bouclé

3.1 Réseaux de neurones statiques (acycliques, non bouclés)

Les réseaux de neurones statiques sont des réseaux unidirectionnels chaque couche traite les données et les transmet à la suivante, sans retour en arrière. Les perceptrons multicouches (MLPs), les réseaux de neurones convolutifs (CNNs) sont des exemples de réseaux statiques utilisés dans des tâches telles que la classification d'images ou des défauts et la régression.

3.2 Réseaux de neurones dynamiques (ou récurrents, ou bouclés)

Ce sont des modèles qui permettent aux informations de circuler dans les deux sens. Cela leur permet de conserver une mémoire des états précédents. Ces réseaux sont utilisés dans des domaines tels que la reconnaissance vocale, et la traduction automatique. Les réseaux les plus connus de cette catégorie sont les réseaux de neurones récurrents (RNNs), les réseaux LSTMs (Long Short-Term Memory Networks), et les réseaux GRUs (Gated Recurrent Unit Networks).

4 Apprentissage des réseaux de neurones

L'apprentissage des réseaux de neurones est un processus par lequel un réseau de neurones ajuste ses paramètres internes (poids), en fonction des données qu'il reçoit, afin d'effectuer une tâche spécifique [15].

On distingue plusieurs types d'apprentissage :

4.1 Apprentissage supervisé

Dans l'apprentissage supervisé, le modèle est entraîné sur un ensemble de données étiquetées, où chaque exemple de données est associé à une étiquette qui indique la réponse attendue.

Le modèle apprend à partir de ces exemples en ajustant ses paramètres pour minimiser la différence entre ses prédictions et les étiquettes réelles.

Globalement, le processus comprend deux phases distinctes :

- **Phase d'apprentissage** : Pendant cette phase initiale, le système apprend à partir des données étiquetées pour déterminer un modèle. Cela implique l'analyse des données existantes pour extraire des motifs et des relations, afin de construire un modèle capable de faire des prédictions ou de prendre des décisions.

- **Phase de test** : Dans cette seconde phase, le système utilise le modèle entraîné pour prédire l'étiquette d'une nouvelle donnée. En se basant sur les informations acquises lors de la phase d'apprentissage, le système évalue la performance du modèle en prédisant les étiquettes correctes pour les nouvelles données.

Ce type d'apprentissage est largement utilisé dans des tâches telles que la classification, la régression et la détection d'anomalies.

4.2 Apprentissage non supervisé

Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé se fait sans étiquettes explicites. Le modèle doit découvrir des structures ou des motifs intrinsèques dans les données sans être guidé par des étiquettes. Les tâches courantes de l'apprentissage non supervisé comprennent le clustering de données.

4.3 Apprentissage par renforcement

Dans l'apprentissage par renforcement, le modèle apprend à travers l'interaction avec un environnement dynamique. Le modèle prend des actions dans l'environnement et reçoit des récompenses ou des pénalités en fonction de ses actions.

L'objectif est d'apprendre à prendre des décisions qui maximisent les récompenses cumulées sur le long terme.

Les applications de l'apprentissage par renforcement comprennent les jeux et la robotique.

4.4 Apprentissage semi-supervisé

L'apprentissage semi-supervisé combine des éléments de l'apprentissage supervisé et non supervisé. Il utilise à la fois des données étiquetées et non étiquetées pour entraîner le modèle, exploitant ainsi les avantages des deux approches.

Cette méthode est utile lorsque les données étiquetées sont rares ou coûteuses à obtenir.

4.5 Apprentissage auto-supervisé

L'apprentissage auto-supervisé est une approche où un modèle apprend à partir de données non étiquetées en formulant des tâches supervisées sur ces données.

Par exemple, dans la prédiction de la prochaine image dans une séquence, le modèle peut être formé pour prédire une image à partir des précédentes.

Cette méthode est utilisée pour pré-entraîner des modèles sur de grandes quantités de données non annotées avant d'effectuer un fine-tuning (réglage fin) sur des tâches spécifiques.

4.6 Rétropropagation de l'erreur

La rétropropagation de l'erreur est un algorithme utilisé dans l'apprentissage supervisé des réseaux de neurones artificiels. Son principe est de calculer les gradients des poids du réseau par rapport à une fonction de perte, puis de propager ces gradients en sens inverse à travers le réseau afin de mettre à jour les poids de manière à minimiser la fonction de perte.

Progressivement, la rétropropagation permet d'améliorer graduellement les performances du réseau en renforçant sa capacité à représenter de manière plus précise les relations entre les entrées et les sorties désirées.

Les étapes de la rétropropagation de l'erreur sont les suivantes [18] :

1. Propagation avant (Forward Pass) :

- Au cours de cette phase, les données sont transmises à travers le réseau de neurones, couche par couche, de l'entrée à la sortie.
- Pour chaque neurone dans chaque couche, les entrées pondérées sont calculées en multipliant les valeurs d'entrée par les poids correspondants, puis en ajoutant un biais.
- Ensuite, une fonction d'activation est appliquée à la somme pondérée pour calculer la sortie du neurone.

Ce processus est répété pour tous les neurones et pour toutes les couches jusqu'à ce que la sortie finale du réseau soit obtenue.

2. Calcul de l'Erreur :

- Une fois que la sortie du réseau est obtenue, l'erreur entre les prédictions du réseau et les vraies étiquettes est calculée à l'aide d'une fonction de perte. Cette fonction mesure à quel point les prédictions du réseau sont proches des vraies étiquettes.

3. **Rétropropagation (Backward Pass) :**

- L'erreur calculée est alors rétropropagée à travers le réseau en utilisant l'algorithme de rétropropagation du gradient.

- Pour chaque poids et chaque biais dans le réseau, on calcule la dérivée partielle de la fonction de perte par rapport à ce poids ou biais en utilisant la règle de la chaîne.

- Les poids et les biais sont ensuite ajustés dans la direction qui minimise l'erreur globale, en utilisant un algorithme d'optimisation.

Ce processus est répété sur plusieurs itérations (époques) jusqu'à ce que le modèle converge vers une solution où l'erreur est minimisée et les prédictions sont optimales.

5 Réseaux de Neurones Convolutifs (CNNs)

Aujourd'hui, nous sommes entourés d'une multitude d'images et de vidéos, que ce soit des scans médicaux, des vidéos de surveillance, des selfies sur les réseaux sociaux ou des diffusions en direct. Cette profusion de données visuelles crée une forte demande pour des ordinateurs capables de les comprendre et d'en tirer des informations utiles. Cependant, ces données sont souvent complexes et variées, ce qui rend difficile leur analyse avec les méthodes traditionnelles. Les techniques habituelles qui consistent à extraire manuellement les caractéristiques visuelles rencontrent leurs limites devant cette diversité de données réelles.

C'est dans ce contexte que les CNNs émergent comme une réponse prometteuse et novatrice. Motivée par la nécessité de développer des systèmes informatiques capables de comprendre et d'interpréter automatiquement les données visuelles, l'étude des CNNs vise à créer des modèles d'apprentissage automatique capables d'apprendre à extraire des informations pertinentes à partir de données brutes. Ils offrent un potentiel considérable pour résoudre des problèmes réels et complexes.

Le tableau (2.3) résume les principales différences entre les deux architectures de réseaux de neurones, mettant en évidence les caractéristiques distinctives des réseaux de neurones classiques par rapport aux réseaux de neurones convolutifs.

Caractéristique	Réseaux de Neurones Classiques	Réseaux de Neurones Convolutionnels (CNNs)
Structure des couches	Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante.	Ils utilisent des couches de convolution, de pooling et entièrement connectées. Chaque neurone dans une couche de convolution est connecté à une région locale de l'entrée, tandis que dans les couches entièrement connectées, chaque neurone est relié à tous les neurones de la couche précédente.
Partage des poids	Chaque poids entre les neurones est unique.	Les poids sont partagés dans les couches de convolution, appliquant les mêmes filtres à différentes régions de l'image.
Invariance aux translations	Ils ne sont pas intrinsèquement invariants aux translations, ce qui signifie que des motifs déplacés dans l'image peuvent ne pas être reconnus de la même manière.	Capables de détecter des motifs indépendamment de leur position grâce à la convolution
Opérations de base	Multiplications matricielles entre couches	Convolutions et Pooling

TABLE 2.3 – Comparaison entre les réseaux de neurones classiques et les réseaux de neurones convolutifs .

6 Architecture générale d'un CNN

Les réseaux de neurones convolutifs (CNNs) désignent une sous-catégorie de réseau de neurones, présentant toutes les caractéristiques d'un réseau de neurones cité précédemment. Cependant, ils sont spécialement conçus pour traiter des images en entrée, leur architecture est donc plus spécifique. Elle est composée de deux blocs principaux [19] :

•**Bloc d'extraction de caractéristiques (ou bloc de convolution)** : Ce bloc est crucial dans un CNN car il sert d'extracteur de caractéristiques. Il fonctionne en appliquant des opérations de filtrage par convolution sur l'image en entrée. La première couche filtre l'image avec plusieurs noyaux de convolution, ce qui produit des "feature maps" (carte de caractéristiques) contenant des caractéristiques importantes de l'image. Ces feature maps sont ensuite normalisées (par exemple avec une fonction d'activation ReLU) et redimensionnées. Ce processus peut être répété plusieurs fois en filtrant les feature maps obtenues avec de nouveaux noyaux, produisant ainsi de nouvelles feature maps à normaliser et redimensionner. Finalement, les valeurs des dernières feature maps sont concaténées dans un vecteur, qui constitue la sortie du premier bloc et l'entrée du second bloc.

•**Bloc de classification (ou bloc entièrement connecté)** : Dans ce bloc, les valeurs du vecteur en entrée sont transformées à l'aide de plusieurs combinaisons linéaires et fonctions d'activation pour produire un nouveau vecteur en sortie. Ce vecteur contient autant d'éléments qu'il y a de classes dans le problème de classification, où chaque élément représente la probabilité que l'image appartienne à une classe spécifique.

Les CNNs sont constitués d'une cascade de couches, chacune ayant des fonctions spécifiques dans la détection et l'identification des caractéristiques dans les données d'entrée. La figure (2.7) présente une vue d'ensemble de cette architecture [21] :

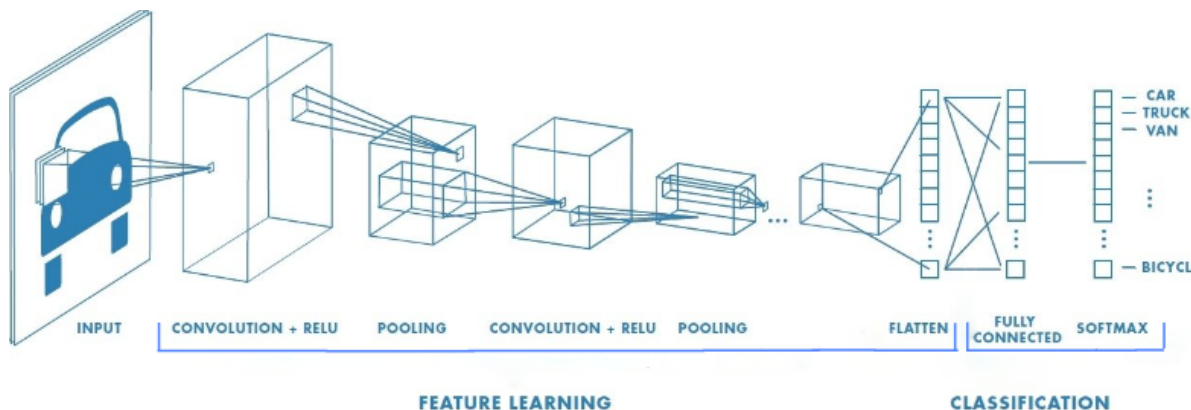


FIGURE 2.7 – Architecture des CNN

6.1 Couche d'entrée

Cette première couche reçoit les images à traiter. Les images sont typiquement représentées sous forme de matrices de pixels de dimension $W \times H \times D$ où W représente la largeur de l'image, H la hauteur de l'image et D le nombre de canaux de couleur (La profondeur des données d'entrée), (1 pour une image en niveaux de gris et 3 pour une image couleur RVB).

6.2 Couches de convolution

Chaque couche de convolution effectue des opérations de convolution entre les filtres (ou noyaux) et l'image d'entrée. Chaque filtre extrait des caractéristiques spécifiques de l'image, telles que les contours ou les textures.

L'opération de convolution est l'étape principale de la couche de convolution. Elle à consiste :

1. Un filtre est superposé sur une petite région de l'image d'entrée, en commençant par le coin supérieur gauche.

2. Pour chaque emplacement où le filtre est superposé, une multiplication élément par élément est effectuée entre les valeurs du filtre et les valeurs correspondantes de l'image d'entrée. Autrement dit, chaque élément du filtre est multiplié par le pixel correspondant dans l'image d'entrée.

3. Les produits de ces multiplications sont ensuite sommés pour obtenir une seule valeur. Cette valeur représente la contribution du filtre à cet emplacement spécifique de l'image.

4. Ce processus est répété pour chaque position où le filtre peut être superposé sur l'image d'entrée, parcourant ainsi toute l'image.

5. Les valeurs obtenues sont rassemblées pour former la carte de caractéristiques (feature maps) de sortie. Chaque valeur dans cette carte représente la réponse du filtre à un motif spécifique dans l'image d'entrée.

La formule de convolution pour une image en niveaux de gris (monochrome) et une image en couleur (RVB) dans le contexte des réseaux de neurones convolutifs (CNNs) est assez similaire, mais diffère légèrement en fonction du nombre de canaux de l'image.

Image en niveaux de gris : La convolution de l'image en niveau de gris I avec un filtre F pour produire la carte des caractéristiques $S = (I * F) + b$ peut être exprimée mathématiquement comme suit :

$$S(i, j) = \left(\sum_{m=0}^{F-1} \sum_{n=0}^{F-1} I(i - m, j - n) \cdot F(m, n) \right) + b \quad (2.3)$$

Où (i, j) sont les coordonnées du pixel dans l'image résultante, et (m, n) sont les indices du filtre de dimension F .

Dans cette formule, le biais b est ajouté après la somme pondérée des produits de convolution. Cela ajoute une constante à chaque pixel de la sortie, introduisant ainsi une non-linéarité dans le modèle et permettant au réseau de neurones de modéliser des décalages ou des tendances dans les données. La figure (2.8) décrit cette opération de convolution mais sans biais.

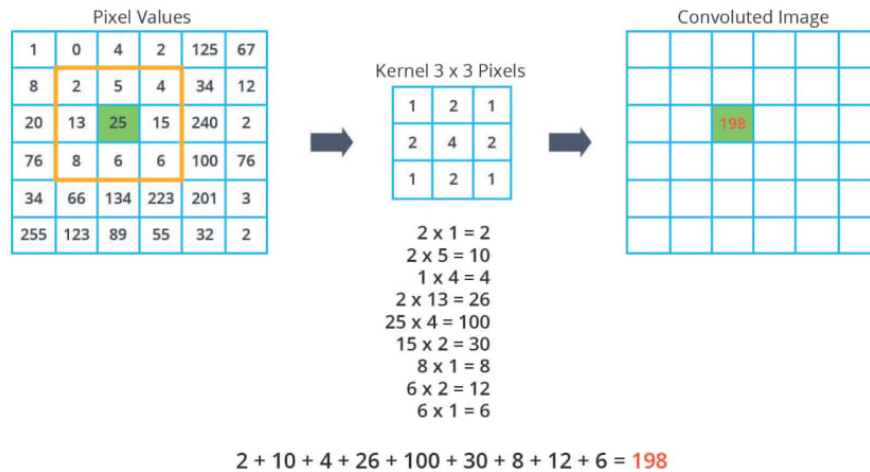


FIGURE 2.8 – convolution d’une image en niveaux de gris

Image en couleur (RVB) : Pour une image en couleur, chaque pixel est représenté par trois canaux de couleur : Rouge (R), Vert (V) et Bleu (B). Donc, chaque canal subit une convolution séparée avec le même filtre, et ensuite les résultats des convolutions des trois canaux sont généralement sommés pour obtenir la valeur finale du pixel dans la carte des caractéristiques de sortie (Fig.2.9).

$$(I * F + b)(i, j) = (I_r * F + b_r)(i, j) + (I_v * F + b_v)(i, j) + (I_b * F + b_b)(i, j) \quad (2.4)$$

où F est le filtre partagé pour tous les canaux, I_r , I_v et I_b représentent respectivement les canaux rouge, vert et bleu de l’image d’entrée, et b_r , b_v et b_b sont les termes de biais respectifs pour chaque canal de couleur.

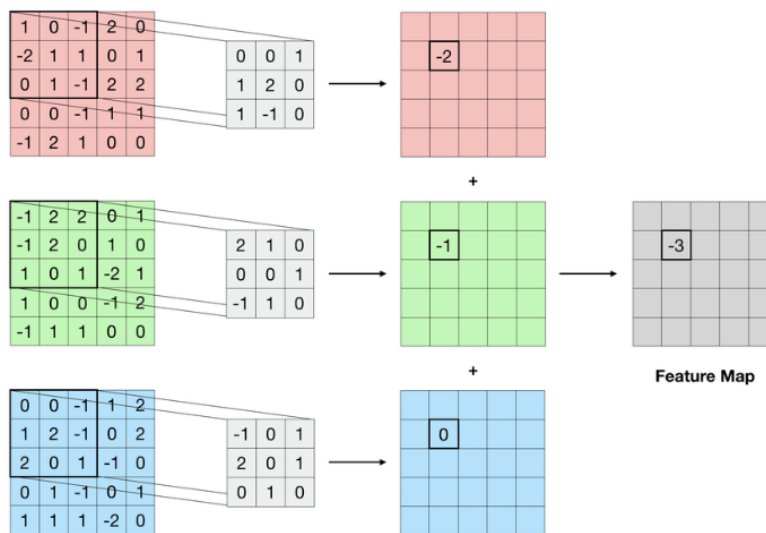


FIGURE 2.9 – convolution d’une image en couleur

Les hyperparamètres d'une couche de convolution

L'opération de convolution dépend de plusieurs paramètres :

- **Le nombre de filtres (ou kernels) (K)** : Il détermine le nombre de caractéristiques ou de motifs différents que la couche de convolution peut extraire de l'image en entrée.

- **La taille des filtres (F)** : Elle spécifie la taille spatiale des filtres de convolution, généralement représentée par une matrice carrée de dimensions $F \times F \times D$.

- **Stride(s) et padding (p)** : Ce sont des paramètres importants dans les CNNs, ils permettent de contrôler la taille de la sortie de la convolution et la manière dont les filtres se déplacent à travers l'image d'entrée.

- **Stride (s)** : Le stride (ou pas) est un paramètre qui définit le nombre de pixels de décalage à chaque fois que le filtre se déplace lors de l'opération de convolution (Fig.2.10). Plus précisément :

- Un stride de 1 signifie que le filtre se déplace d'un pixel à la fois.
- Un stride de 2 signifie que le filtre se déplace de deux pixels à chaque itération, et ainsi de suite.

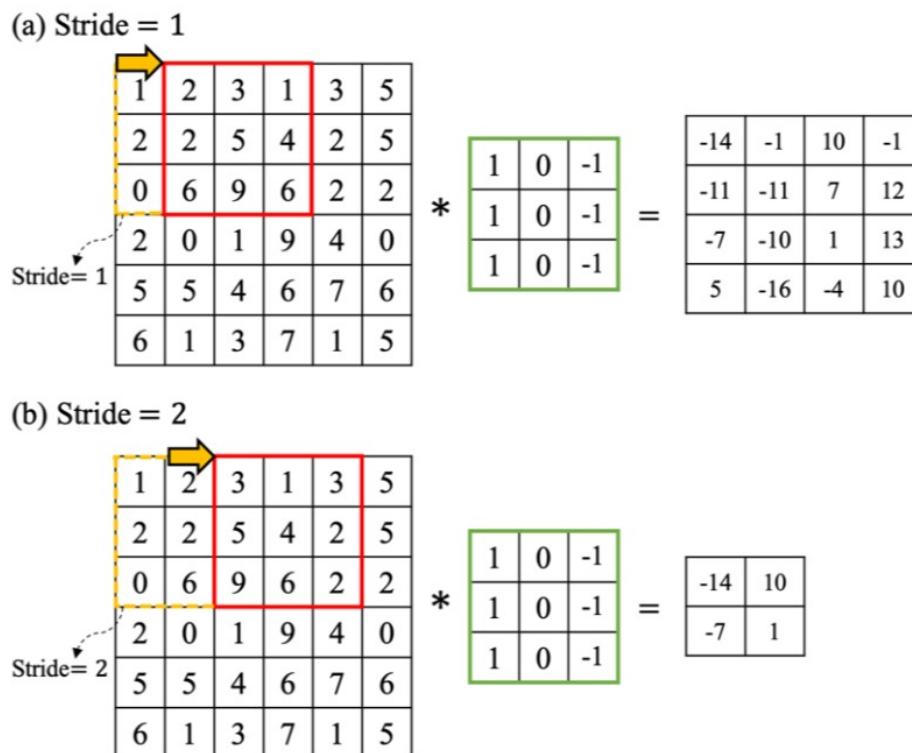


FIGURE 2.10 – Illustration de la notion stride dans une couche de convolution

En utilisant un stride plus grand que 1, la sortie de la convolution sera réduite en taille par rapport à l'image d'entrée. Cela peut être utile pour réduire la dimensionnalité de la sortie et

accélérer le calcul, en particulier dans les couches profondes des réseaux CNN.

✦ **Padding (P)** : Padding consiste à ajouter des pixels supplémentaires autour des bords de l'image d'entrée avant d'appliquer la convolution. Les types de padding sont :

Zero-padding (Rembourrage avec des zéros) : Il ajoute des pixels de valeur zéro autour des bords de l'image (Fig.2.11). sans nécessairement avoir un objectif précis sur la taille de la sortie.

padding same : C'est une application spécifique de zero padding qui vise à maintenir la taille de la sortie égale à celle de l'entrée, en ajoutant les zéros nécessaires autour de l'image.

Valid-padding (Rembourrage valide) : Ne pas ajouter de pixels supplémentaires. Cela signifie que les filtres ne seront appliqués que là où ils s'inscrivent entièrement dans l'image d'entrée, ce qui peut entraîner une réduction de la taille de la sortie.

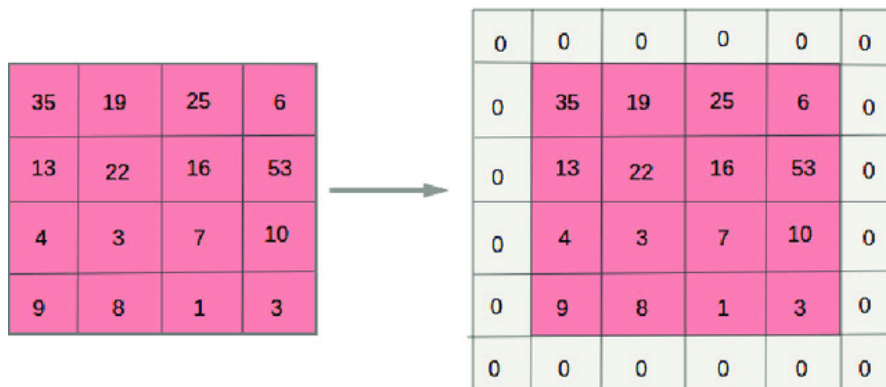


FIGURE 2.11 – Illustration du zero padding

Le padding est généralement utilisé pour deux raisons principales :

- Maintenir la taille spatiale de la sortie de convolution, ce qui est important pour préserver les informations sur les bords de l'image.
- Contrôler la taille de la sortie de la convolution, en particulier lorsque des filtres de grande taille sont utilisés ou lorsqu'il est nécessaire de superposer plusieurs couches de convolution sans réduire drastiquement la taille de la sortie.

Après une couche de convolution dans un CNN, la dimension de la sortie dépend des hyperparamètres de la couche de convolution. Si nous utilisons les notations précédentes :

- W, H, D sont les dimensions de l'image en entrée (largeur, hauteur, et nombre de canaux respectivement).
- K est le nombre de filtres de convolution.

- F est la taille spatiale des filtres.
- P est le zero-padding.
- S est le pas (stride).

Une couche de convolution, produit en sortie K cartes de caractéristiques $I_k(k = 1, \dots, K)$ de dimension $W_{\text{conv}} \times H_{\text{conv}} \times D_{\text{conv}}$ [19] [20].

$$W_{\text{conv}} = \frac{W - F + 2P}{S} + 1 \quad (2.5)$$

$$H_{\text{conv}} = \frac{H - F + 2P}{S} + 1 \quad (2.6)$$

$$D_{\text{conv}} = K \quad (2.7)$$

Notons que les premières couches de convolution détectent des caractéristiques de bas niveau Tels que des bords, des contours et des textures, tandis que les couches supérieures combinent ces caractéristiques pour détecter des motifs plus complexes et abstraits. Elles sont aussi appelées carte de caractéristiques.

6.3 Couche d'activation

Après chaque opération de convolution, une fonction d'activation non linéaire est appliquée sur chaque neurone de la couche de convolution pour introduire de la non-linéarité dans le modèle. La fonction ReLU (Rectified Linear Unit) est couramment utilisée pour cette tâche (Tableau.2.1).

6.4 Couche de pooling

Aussi appelée couche de sous-échantillonnage, cette couche réduit la dimension spatiale des cartes de caractéristiques en conservant les informations les plus importantes. Cela réduit la quantité de calculs et permet d'obtenir une certaine invariance aux translations. Les deux types principaux de couches de pooling sont le max pooling et l'average pooling.

Max Pooling : Dans une couche de max pooling, une fenêtre (généralement de taille 2×2) se déplace sur chaque carte de caractéristiques en entrée. À chaque position, le max pooling sélectionne la valeur maximale de la région couverte par la fenêtre et la transmet à la carte de caractéristiques de sortie (Fig.2.12). Cela permet de réduire la dimension spatiale de la carte tout en conservant les caractéristiques les plus significatives.

Mathématiquement, pour un max pooling avec une fenêtre de taille 2×2 , la valeur $S(i, j)$ de la carte de caractéristiques de sortie $I = I_k$ est donnée par :

$$S(i, j) = \max \{I(2i - 1, 2j - 1), I(2i - 1, 2j), I(2i, 2j - 1), I(2i, 2j)\} \quad (2.8)$$

où I est la carte de caractéristiques en entrée et S est la carte de caractéristiques de sortie.

Average Pooling : Contrairement au max pooling, l'average pooling prend la moyenne des valeurs dans la région couverte par la fenêtre (Fig.2.12). Cela peut être bénéfique pour réduire le suréchantillonnage dans les cartes de caractéristiques tout en maintenant les caractéristiques générales de l'image.

Mathématiquement, pour un average pooling avec une fenêtre de taille 2×2 , la valeur $S(i, j)$ de la carte de caractéristiques de sortie est donnée par :

$$S(i, j) = \frac{1}{4} (I(2i - 1, 2j - 1), I(2i - 1, 2j), I(2i, 2j - 1), I(2i, 2j)) \quad (2.9)$$

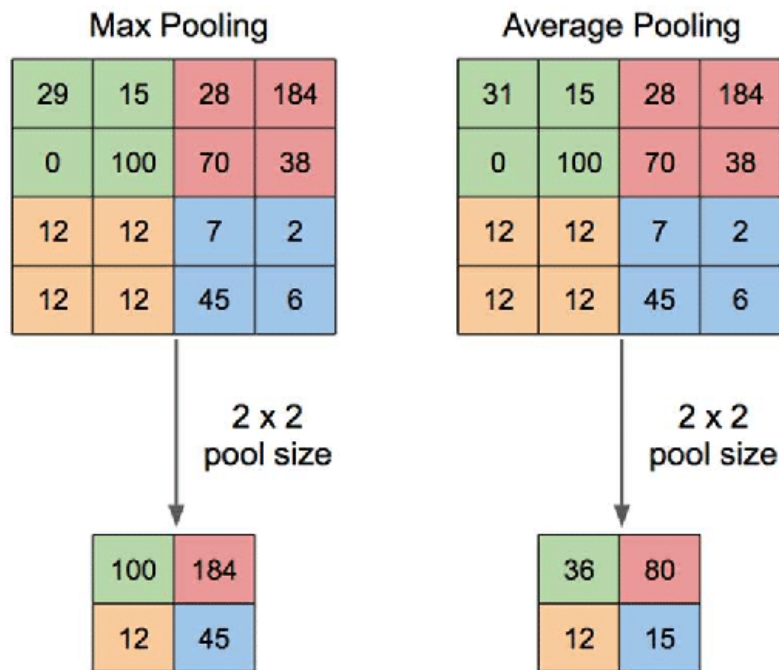


FIGURE 2.12 – Illustration de Max et average pooling

Les hyperparamètres de pooling

Une couche pooling est également définie par des paramètres comme :

•**Taille des cellules (F)** : C'est la taille spatiale des cellules utilisées pour effectuer le pooling. Habituellement, il s'agit d'une valeur comme 2x2, ce qui signifie que chaque cellule est une matrice de 2x2 pixels.

•**Pas (S)** : C'est le décalage ou le pas de déplacement des cellules lors de l'opération de pooling.

La dimension de la carte de caractéristique à la sortie du pooling dépend également des hyperparamètres de la couche de pooling [19] [20] :

$$W_{\text{pool}} = \left\lfloor \frac{W - F}{S} \right\rfloor + 1 \quad (2.10)$$

$$H_{\text{pool}} = \left\lfloor \frac{H - F}{S} \right\rfloor + 1 \quad (2.11)$$

Le nombre de canaux de sortie reste inchangé après la couche de pooling $D_{\text{pool}} = D_{\text{conv}}$.

6.5 Couche Flatten

La couche Flatten ou mise à plat réalise une opération d'aplatissement des données multidimensionnelles en un format unidimensionnel. Cette transformation est essentielle pour permettre au CNN de transmettre le résultat $2D$ des dernières couches de convolution et de pooling à des couches entièrement connectées $1D$. En transformant les données en un vecteur unidimensionnel, la couche Flatten rend les données compatibles avec les couches de neurones entièrement connectées.

6.6 Couche entièrement connectée (FC)

Ces couches entièrement connectées sont généralement placées à la fin du réseau, après plusieurs couches de convolution et de pooling. Elle joue le rôle du classifieur avec une architecture similaire à celle du MLP.

- Une fois que les caractéristiques sont aplaties, chaque neurone de la couche entièrement connectée est connecté à chaque élément de ce vecteur d'entrée. Chaque élément du vecteur d'entrée agit comme une entrée pour chaque neurone de la couche entièrement connectée.

- Comme dans une couche entièrement connectée traditionnelle, pour chaque neurone de la couche entièrement connectée, les valeurs d'entrée sont multipliées par des poids spécifiques à chaque connexion. Ces valeurs pondérées sont ensuite sommées et un biais est ajouté. Cette somme pondérée est ensuite passée à travers une fonction d'activation pour produire la sortie du neurone.

- Une fois que chaque neurone de la couche entièrement connectée a calculé sa sortie, ces sorties sont transmises aux neurones de la couche suivante, généralement la couche de sortie du réseau, où elles sont utilisées pour générer des prédictions.

6.7 Fonction d'activation Softmax

La fonction d'activation softmax est une méthode utilisée dans les réseaux de neurones pour convertir les scores de sortie en probabilités interprétables. Lorsqu'un réseau de neurones produit des scores bruts pour différentes classes, la fonction softmax est appliquée pour normaliser ces scores en probabilités.

Mathématiquement, la fonction softmax prend un vecteur de scores \mathbf{z} et produit un vecteur de probabilités \mathbf{p} , où chaque élément p_i représente la probabilité que l'entrée appartienne à la classe i .

La formule de la fonction softmax est définie comme suit :

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (2.12)$$

Pour chaque score z_i , la fonction softmax calcule l'exponentielle de ce score et le divise par la somme des exponentielles de tous les scores. Cela garantit que chaque probabilité p_i est comprise entre 0 et 1, et que la somme de toutes les probabilités est égale à 1, ce qui est essentiel pour une interprétation probabiliste correcte des prédictions du modèle.

6.8 Couche de perte (Loss)

La couche de perte est essentielle dans l'apprentissage des réseaux de neurones, car elle détermine comment mesurer la différence entre les prédictions du réseau et les étiquettes réelles. Placée à la fin du réseau, elle utilise diverses fonctions de perte adaptées aux différentes tâches. Cette couche guide l'optimisation du réseau en cherchant à réduire au maximum l'écart entre les prédictions et les vérités terrain.

6.8.1 Fonction de perte :

Également appelée fonction d'erreur ou fonction objectif, est une mesure de l'écart entre les prédictions d'un modèle et les valeurs réelles (ou étiquettes) des données d'entraînement [22].

Mathématiquement, la fonction de perte L est une fonction qui prend en entrée les prédictions du modèle \hat{y} et les vraies valeurs y (ou étiquettes) des données, et retourne un scalaire qui quantifie l'écart entre ces deux. Formellement, cela peut être exprimé comme suit :

$$L(\hat{y}, y)$$

où :

- \hat{y} représente les prédictions du modèle.
- y représente les vraies valeurs (ou étiquettes) des données.

La fonction de perte est conçue de manière à ce que lorsque les prédictions du modèle \hat{y} sont proches des vraies valeurs y , la valeur de la fonction de perte est faible, indiquant ainsi une bonne performance du modèle. En revanche, lorsque les prédictions du modèle s'éloignent des vraies valeurs, la valeur de la fonction de perte est élevée, ce qui indique une mauvaise performance du modèle.

Mathématiquement, une fonction de perte peut prendre différentes formes selon le type de problème que l'on cherche à résoudre (classification, régression, segmentation, etc.) et les propriétés des données. Par exemple :

- Pour les problèmes de classification binaire, une fonction de perte couramment utilisée est la fonction de perte logistique (log loss) définie comme :

$$L(\hat{y}, y) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \quad (2.13)$$

- Pour la classification de plusieurs classes, la fonction de perte la plus couramment utilisée est l'entropie croisée catégorielle (cross-entropy loss). Elle est adaptée pour évaluer la divergence entre les distributions de probabilité prédites par le modèle et les distributions réelles des étiquettes. Elle attribue des scores plus élevés lorsque les prédictions du modèle divergent davantage des étiquettes réelles, et des scores plus bas lorsque les prédictions sont proches des vérités de terrain.

La perte d'entropie croisée catégorielle est définie comme suit :

$$L(\hat{y}, y) = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^C y_{ij} \log(\hat{y}_{ij}) \quad (2.14)$$

où :

— n est le nombre d'échantillons dans l'ensemble de données,

— C est le nombre de classes,

— y_{ij} est une variable indicatrice qui vaut 1 si la classe de l'échantillon i est j et 0 sinon,

— p_{ij} est la probabilité prédite par le modèle que l'échantillon i appartienne à la classe j .

- Pour les problèmes de régression, l'erreur quadratique moyenne (mean squared error) est une fonction de perte fréquemment utilisée, définie comme :

$$L(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (2.15)$$

- Pour les problèmes de segmentation, la perte dice est une fonction de perte populaire,

définie comme :

$$L(\hat{y}, y) = 1 - \frac{2 \sum_{i=1}^n \hat{y}_i y_i}{\sum_{i=1}^n \hat{y}_i + \sum_{i=1}^n y_i} \quad (2.16)$$

Pour des tâches spécifiques ou des scénarios particuliers, d'autres fonctions de perte peuvent être utilisées.

6.8.2 Règle de la chaîne

La règle de la chaîne est un concept fondamental en calcul différentiel qui permet de calculer les dérivées des fonctions composées.

Dans le contexte des réseaux de neurones, la règle de la chaîne est utilisée pour calculer les gradients des fonctions de perte par rapport aux poids et aux biais du réseau lors de la rétropropagation du gradient. Voici la formulation mathématique correcte :

Supposons que nous ayons une fonction de perte L qui dépend des sorties du réseau de neurones y , qui elles-mêmes dépendent des poids et des biais θ du réseau. Alors, nous pouvons écrire $L = L(y(\theta))$, où y est une fonction des poids et des biais θ .

La règle de la chaîne dans ce contexte s'exprime comme suit :

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial \theta} \quad (2.17)$$

Où :

$\frac{\partial L}{\partial \theta}$ est le gradient de la fonction de perte par rapport aux poids et aux biais.

$\frac{\partial L}{\partial y}$ est le gradient de la fonction de perte par rapport aux sorties du réseau.

$\frac{\partial y}{\partial \theta}$ est le gradient des sorties du réseau par rapport aux poids et aux biais.

6.9 Techniques d'optimisation

Les techniques d'optimisation sont des méthodes utilisées pour ajuster les paramètres d'un modèle afin de minimiser une fonction de perte spécifique. Voici quelques-unes des techniques d'optimisation les plus couramment utilisées [23] [24] :

6.9.1 Descente de Gradient (Gradient Descent)

Cette méthode ajuste itérativement les paramètres du modèle en suivant la direction opposée au gradient de la fonction de perte, avec un taux d'apprentissage fixe.

Formule de mise à jour :

$$\theta_{t+1} = \theta_t - \alpha \cdot \nabla_{\theta} J(\theta_t) \quad (2.18)$$

où θ_t représente les paramètres à l'itération t , α est le taux d'apprentissage, et $\nabla_{\theta} J(\theta_t)$ est le gradient de la fonction de perte par rapport à θ_t .

6.9.2 Descente de Gradient par Mini-Lots (Mini-Batch Gradient Descent)

Variante de la descente de gradient où les mises à jour des paramètres sont effectuées sur des mini-lots de données plutôt que sur l'ensemble complet des données.

Formule de mise à jour :

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{1}{|B|} \sum_{(x_i, y_i) \in B} \nabla_{\theta} J(\theta_t; x_i, y_i) \quad (2.19)$$

où B est un mini-lot de données de taille $|B|$, x_i sont les données du lot B et y_i leurs étiquettes.

6.9.3 Descente de Gradient Stochastique (SGD)

Autre variante de la descente de gradient où les mises à jour des paramètres sont effectuées sur de petits sous-ensembles aléatoires des données d'entraînement.

Formule de mise à jour :

$$\theta_{t+1} = \theta_t - \alpha \cdot \nabla_{\theta} J(\theta_t; x_i, y_i) \quad (2.20)$$

où (x_i, y_i) est un exemple de données aléatoire.

6.9.4 Momentum

Cette méthode améliore la descente de gradient en ajoutant un terme de "momentum" pour lisser les mises à jour et accélérer la convergence.

Formule de mise à jour :

$$v_t = \beta \cdot v_{t-1} + (1 - \beta) \cdot \nabla_{\theta} J(\theta_t) \quad (2.21)$$

$$\theta_{t+1} = \theta_t - \alpha \cdot v_t \quad (2.22)$$

où v_t est la vitesse à l'itération t , β est le coefficient de friction, et α est le taux d'apprentissage.

6.9.5 RMSProp (Root Mean Square Propagation)

RMSProp ajuste les taux d'apprentissage pour chaque paramètre en utilisant une moyenne mobile des carrés des gradients récents.

Formule de mise à jour :

$$v_t = \beta \cdot v_{t-1} + (1 - \beta) \cdot g_t^2 \quad (2.23)$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_t + \epsilon}} \cdot g_t \quad (2.24)$$

où g_t est le gradient à l'itération t , β est le facteur de décroissance, et ϵ est une petite constante pour éviter la division par zéro.

6.9.6 Adam (Adaptive Moment Estimation)

Adam combine les avantages de Momentum et RMSProp en utilisant des moyennes mobiles des gradients et des carrés des gradients.

Formule de mise à jour :

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (2.25)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (2.26)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.27)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.28)$$

$$\theta_{t+1} = \theta_t - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (2.29)$$

où β_1 et β_2 sont les coefficients pour les moyennes mobiles, et ϵ est une petite constante.

7 Exemples de CNNs

Plusieurs architectures de réseaux de neurones convolutifs (CNNs) qui ont été développées pour des tâches spécifiques en vision par ordinateur. Voici quelques-unes des architectures les plus populaires [25] :

LeNet-5 : Développé par Yann LeCun et son équipe en 1998, c'est l'une des premières architectures de CNN. LeNet-5 était principalement utilisé pour la reconnaissance de caractères dans des chèques bancaires, mais il a également été appliqué à d'autres tâches de classification d'images.

AlexNet : Développé par Alex Krizhevsky, Ilya Sutskever et Geoffrey Hinton, AlexNet a remporté la compétition ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2012. Il a été le premier à utiliser des couches de convolution profondes et a marqué une avancée significative dans le domaine de la vision par ordinateur.

VGGNet : Développé par le Visual Geometry Group (VGG) à l'Université d'Oxford, VGGNet est connu pour sa structure simple avec des couches de convolution de petite taille (3x3) et des couches de pooling de 2x2. Il est célèbre pour son architecture profonde et a été finaliste de la compétition ILSVRC en 2014.

GoogLeNet (Inception) : Développé par l'équipe de recherche de Google, GoogLeNet est célèbre pour son architecture complexe basée sur des modules d'inception. Il utilise des filtres

de taille variable dans une même couche de convolution et a remporté la compétition ILSVRC en 2014.

ResNet : Introduit par Kaiming He et al. en 2015, ResNet est connu pour son architecture résiduelle qui utilise des connexions résiduelles pour surmonter le problème de la disparition du gradient dans les réseaux très profonds. Il a atteint des performances impressionnantes sur diverses tâches de classification d'images.

InceptionResNetV2 : Cette architecture combine les idées d'Inception et de ResNet. Elle utilise des modules d'inception pour extraire les caractéristiques de l'image et intègre des connexions résiduelles pour améliorer la propagation du gradient.

MobileNet : Développé par Google, MobileNet est conçu pour les applications mobiles et les périphériques embarqués. Il utilise des convolutions profondes avec un faible nombre de paramètres pour réduire la complexité et la taille du modèle tout en maintenant de bonnes performances.

EfficientNet : Cette famille d'architectures, introduite par Mingxing Tan et Quoc Le de Google, vise à améliorer l'efficacité des modèles CNN en optimisant conjointement la profondeur, la largeur et la résolution des réseaux.

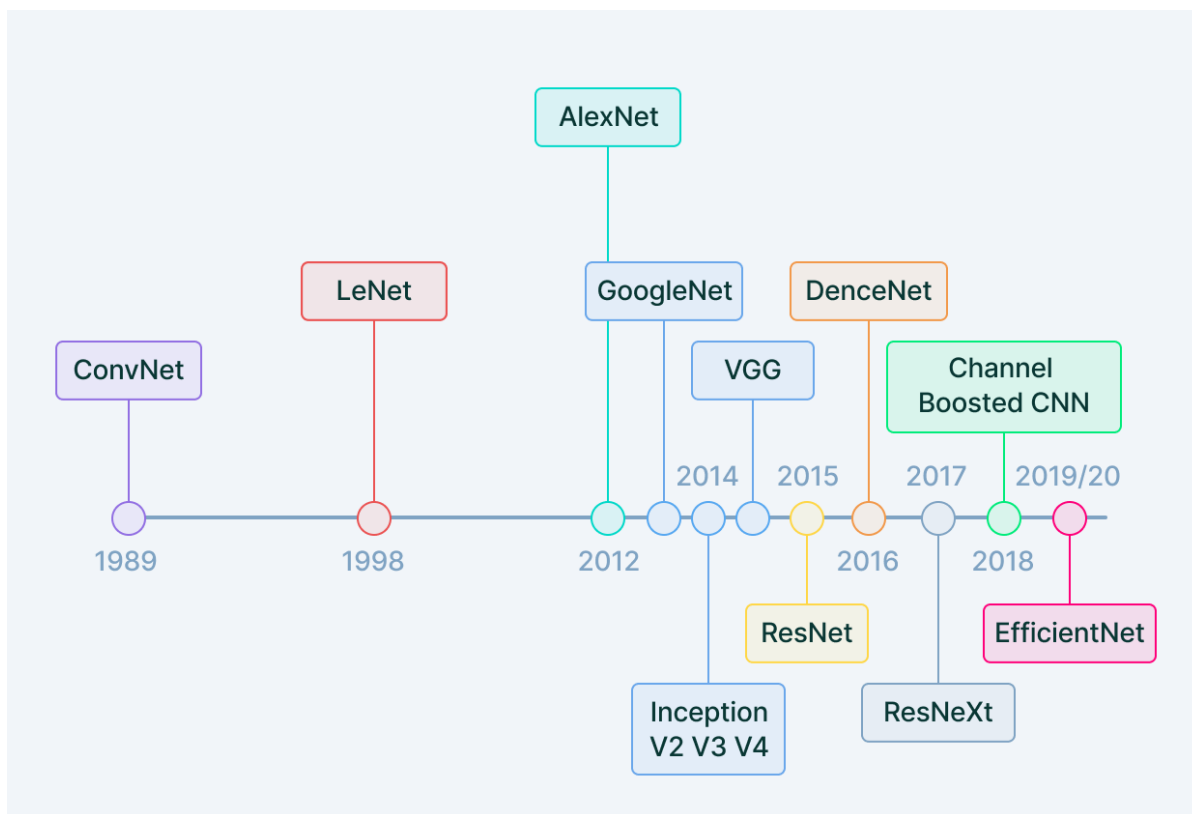


FIGURE 2.13 – Évolution des CNNs

8 Utilisation et entraînement des réseaux de neurones convolutifs

8.1 Prétraitement des données

Le prétraitement des données fait référence à un ensemble d'opérations effectuées sur les données avant de les utiliser pour l'entraînement d'un modèle d'apprentissage automatique. Ces opérations sont conçues pour rendre les données plus appropriées et plus utiles pour le modèle, améliorant leur qualité de résultats, leur cohérence et leur facilité d'utilisation.

- Redimensionnement des images** : Ajuster toutes les images à une taille uniforme pour assurer la compatibilité.

- Normalisation des valeurs de pixels** : Mettre à l'échelle les valeurs de pixels pour qu'elles se situent dans une plage spécifique, généralement entre 0 et 1 [26][27].

- Augmentation des données** : Appliquer des transformations aléatoires telles que des rotations, des zooms et des translations pour augmenter la taille et la variabilité de l'ensemble de données, aider le modèle à généraliser mieux [28].

8.2 Choix du modèle CNN

Le choix d'un modèle CNN dépend des besoins spécifiques et des ressources disponibles, voici trois options courantes [29] :

8.2.1 Création d'un nouveau modèle CNN

Cette approche consiste à concevoir et entraîner un réseau de neurones convolutif (CNN) à partir de zéro. L'architecture du réseau est entièrement définie en fonction des exigences spécifiques de la tâche à accomplir. Cette méthode permet une grande flexibilité dans la conception du réseau, en adaptant des paramètres tels que le nombre de couches, la taille des filtres et la fonction d'activation pour répondre à des besoins précis. Cependant, la création d'un nouveau modèle nécessite une grande quantité de données annotées pour éviter le surapprentissage et obtenir des performances satisfaisantes. De plus, la phase d'entraînement est souvent coûteuse en termes de temps et de ressources computationnelles, car l'algorithme doit apprendre l'ensemble des caractéristiques à partir de données brutes, sans bénéficier de connaissances préalables. Cette approche est donc recommandée lorsque des données suffisantes sont disponibles et que les architectures existantes ne conviennent pas à la tâche.

8.2.2 Transfert d'apprentissage

Le transfert d'apprentissage repose sur l'utilisation de modèles pré-entraînés, c'est-à-dire des réseaux qui ont été initialement entraînés sur de grands ensembles de données génériques, comme ImageNet. Cette méthode est particulièrement utile dans les cas où les données disponibles pour la tâche spécifique sont limitées. Plutôt que de former un modèle à partir de zéro, on utilise les poids et les connexions déjà appris par un modèle performant, et on applique ce modèle à une nouvelle tâche. L'idée est que les premières couches d'un réseau CNN capturent des caractéristiques générales, comme les bords ou les textures, qui peuvent être pertinentes pour de nombreuses tâches de vision par ordinateur. Par conséquent, les couches finales du réseau sont remplacées ou ajustées pour s'adapter aux nouvelles classes cibles. Cette approche permet de réduire le temps d'entraînement et d'obtenir de meilleures performances, même avec un petit jeu de données.

8.2.3 Transfert d'apprentissage avec réglage fin (fine-tuning)

Le fine-tuning est une extension du transfert d'apprentissage, dans laquelle non seulement les dernières couches d'un modèle pré-entraîné sont adaptées à une nouvelle tâche, mais l'ensemble des couches peut être réentraîné sur les données spécifiques de cette tâche. Cela permet d'ajuster finement les poids du modèle pour mieux capturer les particularités de la nouvelle tâche, tout en bénéficiant des connaissances accumulées lors de l'entraînement initial. Cette méthode est particulièrement efficace lorsque les données spécifiques à la tâche sont suffisamment différentes du domaine d'entraînement initial. Le fine-tuning nécessite une plus grande capacité de calcul par rapport au transfert d'apprentissage classique, car les couches profondes du modèle sont également mises à jour, mais il permet souvent d'obtenir des modèles plus performants et mieux adaptés aux nouvelles données.

8.3 Paramètres d'entraînement

Les paramètres d'entraînement, également connus sous le nom d'hyperparamètres, sont des variables qui influencent le processus d'apprentissage. Contrairement aux paramètres du modèle, comme les poids et les biais, qui sont appris à partir des données pendant l'entraînement, les hyperparamètres doivent être définis avant le début de l'entraînement et restent constants pendant celui-ci. Voici quelques-uns des principaux paramètres d'entraînement pour les CNN :

- **Taux d'apprentissage (learning rate)** : Détermine la taille des pas effectués lors de la mise à jour des poids du modèle pendant l'entraînement. Un taux d'apprentissage trop élevé peut entraîner une divergence ou une instabilité, tandis qu'un taux trop faible peut ralentir la

convergence.

- **Taille du lot (batch size)** : Détermine le nombre d'exemples utilisés dans chaque itération de l'entraînement. Un plus grand lot peut accélérer l'entraînement en exploitant des calculs parallèles, mais il nécessite également plus de mémoire.

- **Nombre d'itérations (epochs)** : Indique le nombre de fois que l'ensemble de données complet est passé à travers le modèle pendant l'entraînement. Contrôle la durée totale de l'entraînement et peut aider à éviter le surapprentissage en s'arrêtant lorsque les performances sur un ensemble de validation commencent à se détériorer.

- **Taille et nombre des filtres de convolution** : Plus le nombre de filtres est élevé, plus le réseau peut détecter de motifs différents. Les filtres plus grands capturent des motifs plus gros, tandis que les plus petits sont sensibles aux détails fins.

- **Taille de la fenêtre de pooling** : Une fenêtre plus grande réduit la dimension de l'image en conservant les caractéristiques importantes. Une fenêtre plus petite préserve les détails fins.

- **Stratégie de padding et de stride** : Le padding Same maintient la taille de l'entrée dans la sortie, tandis que le padding Valid réduit la taille de la sortie. Un stride plus élevé réduit la taille de la sortie, mais peut entraîner une perte d'informations spatiales.

8.4 Surapprentissage

Le surapprentissage (ou overfitting en anglais) se produit lorsqu'un modèle d'apprentissage profond apprend à partir des données d'entraînement avec une précision trop élevée, à tel point qu'il commence à capturer le bruit ou les fluctuations aléatoires présentes dans les données d'entraînement. En conséquence, le modèle devient trop spécifique aux données d'entraînement et ne généralise pas bien à de nouvelles données. Pour faire face à ce problème plusieurs stratégies sont proposées :

8.4.1 Régularisation :

Concrètement, la régularisation vise à limiter la complexité du modèle en ajoutant une pénalité à la fonction de coût (ou fonction de perte) du modèle pendant l'entraînement. Cette pénalité favorise des modèles plus simples et plus lisses, ce qui réduit le risque de surapprentissage. Les techniques de régularisation les plus courantes sont :

- **Régularisation L1 (Lasso) et L2 (Ridge)** :

La régularisation L1 (également appelée régularisation de Lasso) ajoute la somme des valeurs absolues des poids, tandis que la régularisation L2 (également appelée régularisation de Ridge) ajoute la somme des carrés des poids.

Les deux techniques encouragent les poids du modèle à rester petits, mais la régularisation L1 peut également entraîner une sélection automatique des caractéristiques en forçant certains

poids à devenir exactement zéro.

Mathématiquement, la régularisation L1 et L2 sont définie comme suit [30] :

$$\text{Régularisation L1} = \lambda \sum_{i=1}^n |w_i| \quad (2.30)$$

$$\text{Régularisation L2} = \lambda \sum_{i=1}^n w_i^2 \quad (2.31)$$

où :

- λ est le coefficient de régularisation qui contrôle l'importance de la régularisation.
- w_i sont les poids du modèle.
- n est le nombre total de poids dans le modèle.

8.4.2 Dropout :

Le concept du Dropout consiste à simuler l'entraînement de plusieurs sous-réseaux pendant chaque itération de l'entraînement en désactivant aléatoirement un pourcentage de neurones. Cette technique contraint le réseau à apprendre des caractéristiques plus robustes en évitant de trop dépendre de certains neurones ou combinaisons de neurones spécifiques.

8.4.3 Arrêt anticipé (early stopping) :

L'arrêt anticipé est une technique simple mais efficace qui consiste à arrêter l'entraînement du modèle lorsque la performance sur un ensemble de validation commence à se détériorer, indiquant que le modèle commence à surapprendre.

9 Défis et limitations des réseaux de neurones convolutifs

Bien que les réseaux de neurones convolutifs (CNNs) soient extrêmement puissants et polyvalents, ils présentent également des limites et des défis. Voici quelques-uns des principaux :

1. Exigences en matière de données :

Les CNNs nécessitent souvent de grandes quantités de données d'entraînement pour obtenir des performances optimales, ce qui peut être difficile à obtenir dans certains domaines, notamment en médecine ou dans les secteurs où les données sont rares.

2. Surapprentissage (overfitting) :

Comme de nombreux modèles d'apprentissage profond, les CNNs sont susceptibles de surapprendre sur les données d'entraînement, ce qui signifie qu'ils peuvent avoir du mal à généraliser à de nouvelles données.

3. **Interprétabilité :**

Les CNNs sont souvent considérés comme des "boîtes noires" en raison du grand nombre de paramètres qu'ils utilisent et de la complexité de leur architecture. Comprendre pourquoi un modèle CNN prend une décision particulière peut être difficile, ce qui peut être un obstacle dans les applications où l'interprétabilité est importante, comme en médecine.

4. **Taille du modèle et puissance de calcul :**

Les CNNs, en particulier les architectures les plus avancées, peuvent être très volumineux et nécessiter des ressources informatiques importantes pour l'entraînement et l'inférence. Cela peut rendre l'utilisation des CNNs coûteuse en termes de temps et de puissance de calcul.

5. **Invariance spatiale et translationnelle :**

Bien que les CNNs soient capables de détecter des motifs spatiaux dans les images, ils peuvent avoir du mal à généraliser efficacement à des transformations spatiales ou translationnelles, telles que la rotation ou le changement d'échelle. Cela peut nécessiter l'utilisation de techniques supplémentaires pour augmenter la robustesse du modèle.

10 Conclusion

Nous avons exploré dans ce chapitre les concepts fondamentaux des réseaux de neurones et leur évolution vers les réseaux de neurones convolutifs (CNNs). Nous avons vu comment les réseaux de neurones se sont inspirés du fonctionnement du cerveau humain pour apprendre à partir de données et effectuer des tâches complexes telles que la reconnaissance d'images, la classification et la segmentation. Les CNNs, en particulier, se sont révélés être des outils puissants dans le domaine de la vision par ordinateur, avec leur capacité à détecter automatiquement des motifs et des caractéristiques dans les données visuelles.

Nous avons examiné les composants clés des CNNs, tels que les couches de convolution, de pooling et entièrement connectées, ainsi que les fonctions d'activation comme ReLU et softmax.

Enfin, nous avons souligné des défis tels que l'interprétabilité des modèles, le surapprentissage et les exigences en matière de données. Néanmoins, avec les avancées continues dans le domaine de l'apprentissage profond et de l'intelligence artificielle, les CNNs continuent de progresser et de trouver de nouvelles applications dans divers domaines, ouvrant la voie à un avenir où les machines sont capables d'apprendre, de raisonner et d'interagir de manière plus intelligente avec le monde qui les entoure.

Chapitre 3 :
Détection Des Défauts De Surface À L'aide
Des CNNs

Chapitre 3

Détection Des Défauts De Surface À L'aide Des CNNs

1 Introduction

Dans ce chapitre, nous nous focalisons sur l'application des réseaux de neurones convolutifs (CNNs) à la détection des défauts de surface. Pour ce faire, nous adopterons deux approches complémentaires. La première est la détection des défauts par classification à une classe, permettant de détecter la présence ou non d'une anomalie. La seconde est la reconnaissance des défauts par classification Multiclasses, permet d'identifier le type du défaut. L'efficacité de chaque approche est évaluée sur trois bases d'images.

2 Bases d'images traitées

Nous avons utilisé trois bases de données, chacune contenant des images spécifiques à des contextes industriels différents, permettant ainsi une évaluation approfondie des capacités du modèle à identifier des anomalies dans diverses conditions. Voici une description détaillée de ces bases :

2.1 Base Pill QC

La base Pill QC est une collection de 330 images couleur, utilisée pour évaluer le contrôle qualité des pilules pharmaceutiques. Cette base de données est subdivisée en trois catégories distinctes (Fig.3.1) :

1. **Classe 1 « Normal »** : Cette classe comprend 149 images de pilules sans défauts. Ces images servent de base pour l'apprentissage des caractéristiques des pilules intactes.

2. **Classe 2 « Chip »** : Cette classe contient 43 images de pilules présentant des défauts d'écaillage ou de fragmentation visibles sur leur surface (éclats ou fissures). Ces défauts structurels permettent au modèle d'apprendre à identifier les pilules endommagées.
3. **Classe 3 « Dirt »** : Cette classe est constituée de 138 images de pilules présentant des saletés ou des impuretés sur leur surface. Ces défauts de surface permettent au modèle de distinguer les pilules propres de celles contaminées.

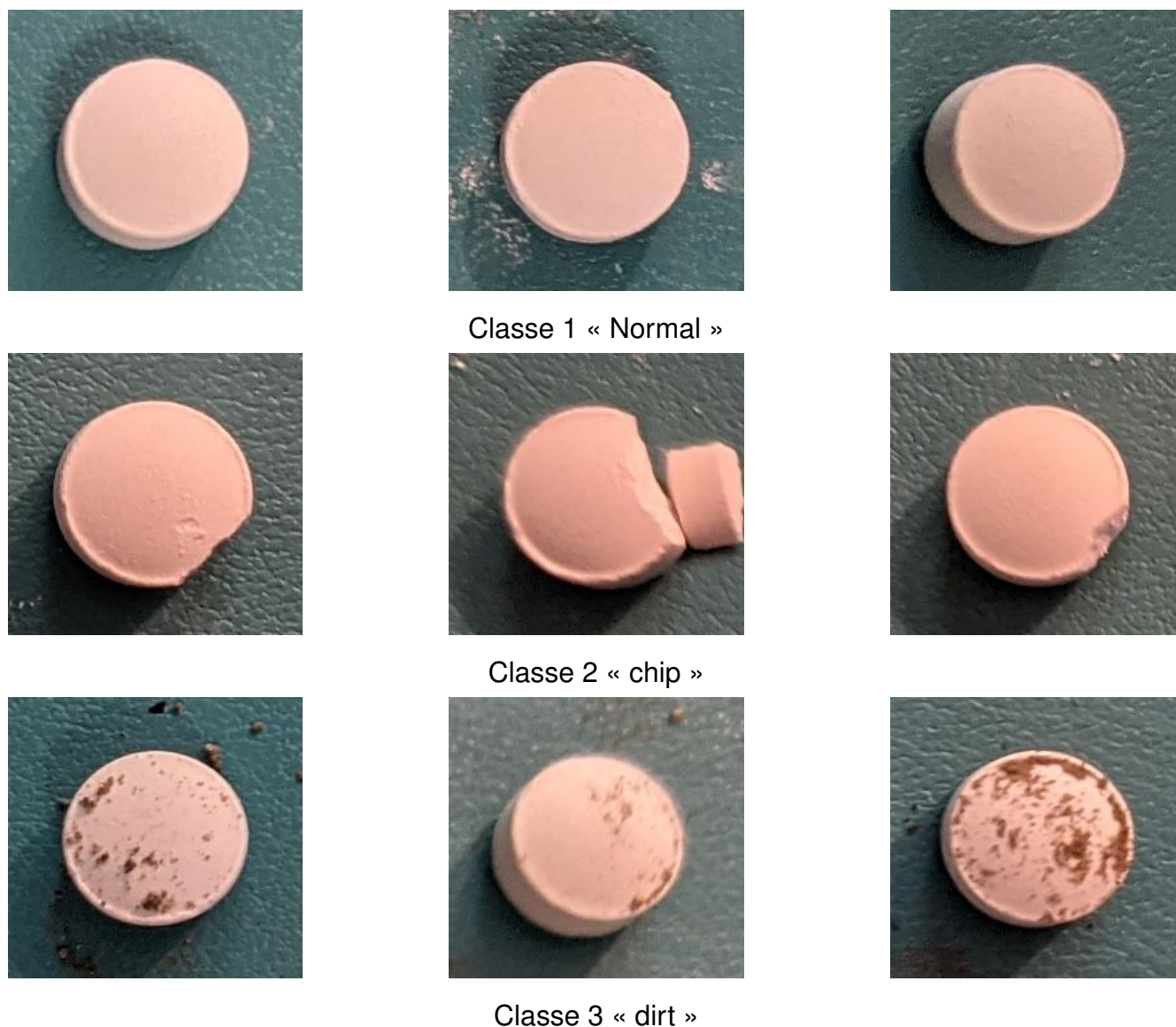


FIGURE 3.1 – Exemples d'images de la base Pill Qc

2.2 Base Hazelnut

L'ensemble de données Hazelnut comprend un total de 15 770 images représentant 1577 noisettes différentes, réparties en trois catégories (classes) distinctes (Fig.3.3). Chaque noi-

sette est photographiée sous différents angles avec 10 photos par noisette, offrant une vue variée de leurs caractéristiques. Les images sont toutes en couleur avec une résolution de 224x224 pixels. Les catégories sont les suivantes :

1. **Classe 1 « quality_nuts » (Noisette de qualité)** : Cette classe regroupe 5350 images de noisettes présentant des caractéristiques idéales et sans défauts visibles. Les noisettes de cette classe apparaissent en parfait état, avec une surface lisse, une couleur uniforme et une forme régulière. Ces noisettes sont exemptes de tout dommage ou altération, correspondant aux normes élevées de qualité.
2. **Classe 2 « damaged_nuts » (Noisette endommagée)** : Cette classe inclut 5190 images de noisettes présentant des signes visibles de défauts. Les noisettes endommagées peuvent présenter des fissures, des éraflures, des taches ou des décolorations, généralement causées par des conditions de croissance défavorables ou un stockage inapproprié.
3. **Classe 3 « nuts_kernel » (Coque de noisette)** : Composée de 5230 images, cette classe représente un type de défaut où les images montrent des noisettes avec des coques partiellement ou complètement cassées, ainsi que des noyaux gravement endommagés et non comestibles. Les noyaux sont souvent troués ou fortement détériorés, ce qui indique des anomalies significatives.



Classe 1 « quality_nuts »



Classe 2 « damaged_nuts »



Classe 3 « nuts_kernel»

FIGURE 3.3 – Exemples d'images de la base Hazelnut

2.3 Base Wood

La base de données Wood fait partie de l'ensemble MVTEC, spécialement conçue pour la détection d'anomalies dans des contextes industriels. Elle comprend un total de 307 images en couleur de surfaces de bois, toutes dimensionnées à 1024 x1024 pixels et réparties en six classes (Fig.3.5). Le bois étant un matériau largement utilisé dans divers secteurs industriels, la qualité de ses surfaces est essentielle pour assurer la durabilité et l'esthétique des produits finis.

1. **Classe 1 « Good »** : Cette classe représente 247 images de bois sans aucun défaut visible. Les surfaces de bois sont intactes, sans imperfections ni marques indésirables, servant de référence pour les autres classes.
2. **Classe 2 « Color »** : Cette classe regroupe 8 images où les défauts sont liés à des variations anormales de couleur du bois. Cela peut inclure des taches, des décolorations ou des irrégularités dans les teintes naturelles du bois.
3. **Classe 3 « Scratch »** : Cette classe contient 21 images de bois présentant des rayures superficielles ou profondes, qui altèrent la surface du matériau. Ces rayures peuvent être causées par des outils, des frottements ou d'autres interactions mécaniques.
4. **Classe 4 « Hole »** : Les 10 images dans cette classe montrent des trous ou des perforations sur les surfaces de bois. Ces trous peuvent être le résultat de défauts naturels, d'insectes ou de dommages physiques.
5. **Classe 5 « Combined »** : Cette classe regroupe 11 images présentant plusieurs types de défauts combinés, tels que des rayures et des variations de couleur, ou encore des trous. C'est une classe plus complexe, où plusieurs types de défauts coexistent sur une même surface.
6. **Classe 6 « Liquid »** : Cette classe inclut 10 images où la surface du bois est affectée par des liquides, tels que des taches d'eau, de produits chimiques ou d'autres substances, provoquant des altérations visibles sur la texture ou la couleur du bois.



Classe 1 « Good »



Classe 2 « Color »



Classe 3 « Scratch »



Classe 4 « Hole »



FIGURE 3.5 – Exemples d’images de la base Wood

3 Approches utilisées

Nous avons exploité deux modèles de CNNs pour classer les défauts de surface :

3.1 Classification à une classe

La classification à une classe ou One-Class Classification, est basée sur l’apprentissage des caractéristiques des données normales, aussi appelées classe nominale. Contrairement aux méthodes traditionnelles qui nécessitent des exemples de chaque classes, cette approche fonctionne sans ou avec très peu d’exemples d’anomalies. L’idée est que le modèle apprenne à reconnaître ce qui est normal et puisse ensuite identifier les échantillons qui diffèrent significativement de ces données normales, les considérant ainsi comme anormaux. Cette méthode est particulièrement utile lorsque les exemples d’anomalies sont rares ou difficiles à obtenir. Dans le cadre de ce mémoire, nous nous sommes intéressé à une méthode de classification à une classe basée sur les CNNs appelée Fully Convolutional Data Description (FCDD).

3.1.1 Architecture du FCDD

FCDD est un modèle de CNN développé par Liznerski et Al en 2021 pour détecter les anomalies dans les données visuelles [31]. Il est caractérisé par son architecture entièrement

convolutive (FCN : Fully Convolutional Network)(Fig.3.7). Contrairement aux architectures CNNs traditionnelles, FCDD n'utilise pas de couches entièrement connectées, permettant ainsi de conserver les informations spatiales des images tout au long du réseau.

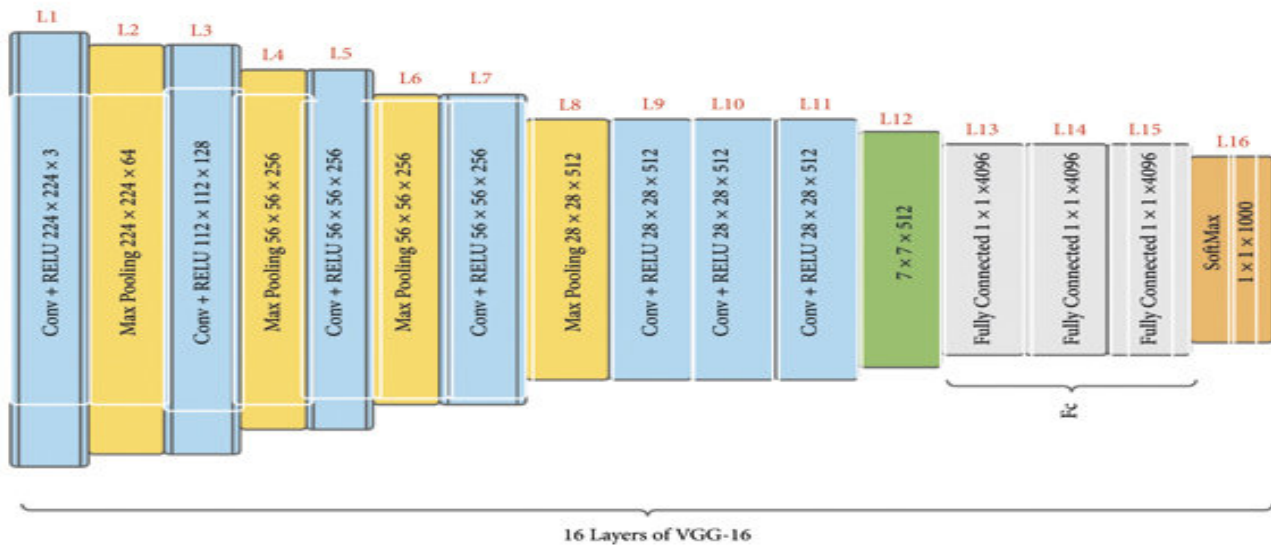


FIGURE 3.6 – Architecture du VGG16

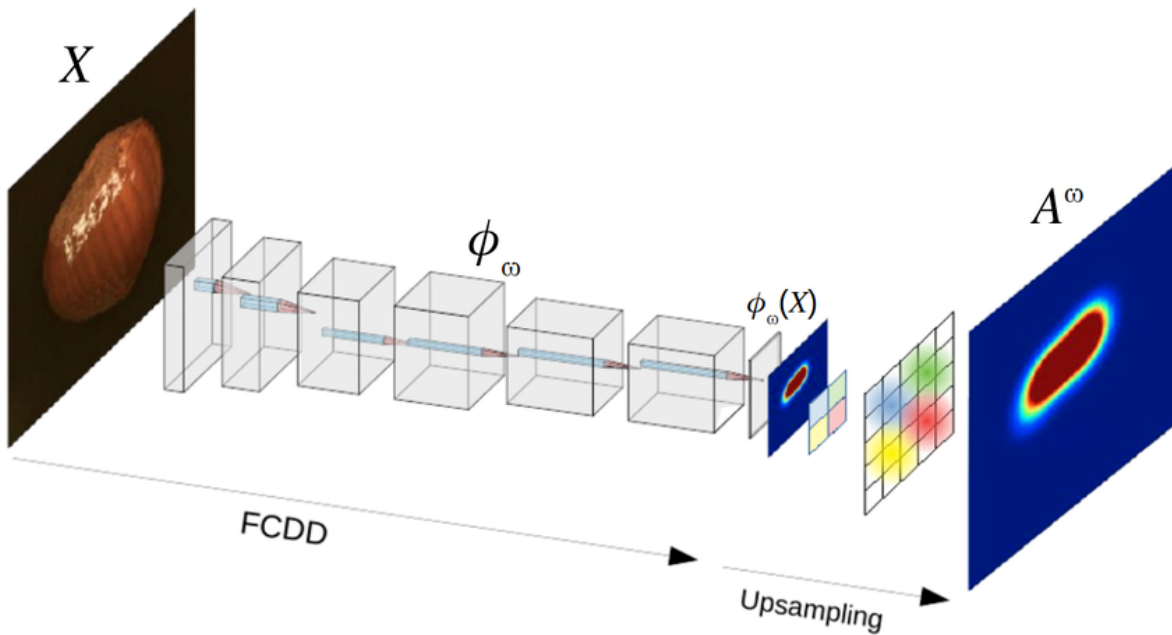


FIGURE 3.7 – Architecture du FCDD

Le FCDD est composé d'un réseau de neurones convolutif préentraîné. Ce réseau, déjà entraîné sur des ensembles de données vastes et diversifiés (ImageNet), possède une capacité avancée à extraire des caractéristiques visuelles complexes. Cette base solide de connaissances permet de raccourcir le processus d'apprentissage et d'améliorer les performances sur

des tâches spécifiques en exploitant les caractéristiques préalablement acquises. En particulier, les réseaux préentraînés offrent une meilleure généralisation, ce qui est crucial lorsque les données d'entraînement disponibles sont limitées. Parmi les modèles pré-entraînés disponibles, nous avons opté pour VGG16, réputée pour ses performances robustes et sa capacité à extraire des caractéristiques complexes des images. VGG16 se compose de 16 couches, incluant 13 couches convolutives et 3 couches entièrement connectées (Fig.3.6). Sa profondeur combinée à sa simplicité le rend capable de capturer des caractéristiques détaillées tout en étant relativement facile à implémenter. Pour adapter le modèle VGG16 à la détection des défauts de surface, plusieurs ajustements ont été apportés à son architecture d'origine :

- Les couches convolutives préentraînées sont gelées pour préserver leurs poids acquis, stabilisant ainsi les caractéristiques extraites lors de l'entraînement sur nos données spécifiques.

- Les couches entièrement connectées traditionnelles à la fin du VGG16 sont remplacées par deux couches de convolutions de 512 filtres de taille 3x3 chacune, utilisant un padding de type 'same'. Ces modifications permettent au VGG16 de devenir un modèle entièrement convolutif. Ces couches sont suivies d'une couche de normalisation par lot pour stabiliser l'apprentissage et d'une fonction d'activation ReLU pour introduire de la non-linéarité.

- Pour générer une carte de score d'anomalie, une couche de convolution finale avec un seul filtre est utilisée. Ce filtre calcule la distance entre les caractéristiques extraites et celles des données normales. Ensuite, une transformation mathématique est appliquée aux valeurs des filtres pour obtenir le score d'anomalie.

- Une couche de redimensionnement (interpolation bilinéaire) augmente la résolution spatiale des cartes de score, facilitant la visualisation des anomalies.

- Une couche de pooling global agrège les caractéristiques spatiales en une seule valeur par canal, simplifiant ainsi la représentation des anomalies détectées.

- Enfin, une couche spécifique de calcul de perte (loss layer) adaptée à la tâche de détection d'anomalies pour optimiser le réseau.

Pour une base d'apprentissage constituée de n échantillons X_1, X_2, \dots, X_n munis de leurs étiquettes y_1, y_2, \dots, y_n , où $y_i = 1$ dénote une anomalie et $y_i = 0$ indique la classe normale.

Le FCDD peut être représenté par une fonction $\phi(X_i, W) : \mathbb{R}^{h \times w \times c} \rightarrow \mathbb{R}^{u \times v}$ dépendant des

pois synaptiques W du réseau, qui a pour but de transformer une image couleur X_i de dimension $h \times w \times c$ (h : nombre de lignes, w : nombre de colonnes et c : nombre de couleurs) en une image de dimension $(u \times v)$.

L'apprentissage du FCDD a pour objectif de déterminer les poids synaptiques W en minimisant la fonction objective suivante :

$$\min_W \frac{1}{n} \sum_{i=1}^n \left((1 - y_i) \cdot \frac{1}{u \cdot v} \|A(X_i)\|_1 - y_i \cdot \log \left(1 - \exp \left(-\frac{1}{u \cdot v} \|A(X_i)\|_1 \right) \right) \right) \quad (3.1)$$

où

$$\|A(X_i)\|_1 = \sqrt{\phi(X_i; W)^2 + 1} - 1 \quad (3.2)$$

3.1.2 Fonctionnement du FCDD

Le processus de détection des défauts de surface à l'aide du FCDD se déroule en deux phases :

Phase 1 : Entraînement du Modèle

1. **Extraction de Caractéristiques (Feature Extraction)** : Le modèle est entraîné avec un ensemble de données ne contenant que des exemples normaux, sans défauts ni anomalies. Les images passent à travers le réseau Fully Convolutional Network (FCN), où des caractéristiques pertinentes sont extraites à chaque couche convolutive.
2. **Exposition aux Anomalies Auxiliaires (Auxiliary Outlier Exposure)** : Lors de l'entraînement d'un modèle de détection d'anomalies, il peut être difficile de bien définir ce qui constitue une anomalie si l'on se base uniquement sur des données normales. L'idée de l'Auxiliary Outlier Exposure (OE) est d'introduire des échantillons supplémentaires qui sont clairement des anomalies, mais qui proviennent d'un autre ensemble de données, différent des données normales utilisées pour l'entraînement principal.
3. **Projection dans l'Espace Latent** : Ces caractéristiques sont projetées dans un espace latent de dimensions réduites. Cet espace est conçu de manière à ce que les points de données similaires (c'est-à-dire qui partagent des caractéristiques similaires) soient regroupés près les uns des autres, tandis que les points dissimilaires soient éloignés. Le modèle apprend à modéliser la distribution des caractéristiques des données normales, créant ainsi un espace de caractéristiques normal. Cet espace représente les caractéristiques des données normales et sert de référence pour la détection des anomalies.

Phase 2 : Détection des Défauts

1. **Analyse des Nouvelles Données** : Les nouvelles données (images) passent à travers les mêmes couches convolutives du réseau FCN pour extraire leurs caractéristiques.

- 2. Comparaison avec le Modèle Normal** : Les caractéristiques des nouvelles données sont projetées dans le même espace latent que celui utilisé pendant l'entraînement. La distance entre les caractéristiques des nouvelles données et la distribution des caractéristiques normales est calculée. Une distance significative indique une anomalie potentielle.
- 3. Calcul du Score d'Anomalie (Anomaly Score Layer)** : Au bout du réseau, une couche spécifique produit un score d'anomalie en évaluant la distance entre les caractéristiques extraites et celles attendues pour des données normales. Les scores élevés indiquent des anomalies, tandis que les scores bas indiquent des données normales. Après la projection des caractéristiques, les distances calculées sont agrégées, souvent par moyenne ou somme, puis normalisées à l'aide de transformations mathématiques pour ajuster les valeurs dans une plage spécifique, facilitant ainsi leur interprétation et comparaison.
- 4. Upsampling et Génération de Heatmaps** : Les scores d'anomalie sont visualisés sous forme de cartes de chaleur (heatmaps), où les régions avec des scores élevés sont mises en évidence avec des couleurs chaudes (rouge) et les régions normales avec des couleurs froides (bleu). Les heatmaps sont des visualisations qui montrent l'emplacement des anomalies dans les données d'entrée en mettant en évidence les zones qui sont considérées comme anormales par le modèle. La sortie brute du réseau est une carte d'anomalie de basse résolution. Pour produire des heatmaps de haute résolution, les scores d'anomalie sont suréchantillonnés (augmentés en résolution) à la taille originale de l'image d'entrée par interpolation bilinéaire, puis un noyau gaussien est appliqué pour lisser d'avantage la heatmap. Le paramètre σ du noyau gaussien contrôle le degré de lissage. Un σ plus élevé entraîne une carte plus floue, tandis qu'un σ plus bas conserve plus de détails locaux.

3.1.3 Test et résultats

Dans cette section, nous présenterons les résultats de détection de défauts de surface obtenus par le modèle FCDD dans les ensembles de données Pill QC, Hazelnut et Wood.

3.1.3.1 Répartition des Données

Pour évaluer les performances de notre modèle FCDD dans la détection de défauts de surface, nous avons structuré les ensembles de données en deux classes principales : « Normal » et « Anomaly ». En nous concentrant sur une seule classe d'anomalie, notre objectif est d'optimiser le modèle pour détecter toute déviation significative par rapport aux données normales, ce qui est essentiel pour identifier de nouvelles anomalies non observées lors de

l'entraînement. Cette stratégie simplifie la tâche de détection tout en augmentant la robustesse du modèle.

Pour les bases Pill QC et Hazelnut, nous avons réparti 70 images normales dans la base d'entraînement et 30 dans la base de validation, représentant respectivement la classe « Normal » et « quality_nuts ». Les images restantes ont été affectées à la base de test.

Concernant les anomalies, seulement 4 images (2 par classe) contenant des défauts de surface ont été utilisées pour l'entraînement. Cette approche, bien que limitée en termes de données anormales, permet au modèle de se concentrer sur la distinction des caractéristiques des images sans défauts. Nous avons également utilisé 30 images anormales (15 par classe) pour la validation. Ces anomalies, correspondent aux classes « Chip » et « Dirt » pour Pill Qc , « damaged_nuts » et « nuts_kernel » pour Hazelnut, les classes de chacune de ces bases ont été regroupées en une seule classe « Anomaly ». Les images restantes d'anomalies ont été réservées à la base de test.

De même, pour la base de données Wood, la structure des données d'entraînement reste identique à celle de Pill QC et Hazelnut, avec 70 images de la classe « Good » (normales) pour l'entraînement et 30 pour la validation. Concernant les anomalies, nous avons utilisé 10 images (2 par classe) pour l'entraînement et 25 images (5 par classe) pour la validation. Les classes « Color », « Scratch », « Hole », « Combined » et « Liquid » ont été aussi regroupées en une seule classe « Anomaly ». Les images restantes des anomalies de la base Wood ont été réservées à la base de test, suivant le même principe que pour Pill QC et Hazelnut. Comme mentionné dans le tableau(3.1) ci-dessous .

Base de Données	Résolution d'Image	Classe	Base d'Entraînement	Base de Validation	Base de Test
Pill QC	225x225	Normal	70	30	49
		Anomaly ("chip" et "dirt")	4	30	147
Hazelnut	224x224	Normal ("quality_nuts")	70	30	5250
		Anomaly ("damaged_nuts" et "nuts_kernel")	4	30	10 386
Wood	1024x1024	Normal ("good")	70	30	147
		Anomaly ("color", "scratch", "hole", "combined" et "liquid")	10	25	25

TABLE 3.1 – Répartition des données pour l'évaluation du modèle FCDD

3.1.3.2 Métriques de Performance :

Différentes métriques de performance sont utilisées pour évaluer l'efficacité du modèle FCDD à détecter les anomalies de surfaces.

• **Matrice de Confusion** : Une matrice de confusion est un tableau de dimension $n \times n$, où n représente le nombre de classes (Tab.3.2). Elle est utilisée pour évaluer la performance d'un modèle de classification en comparant les prédictions du modèle avec les résultats réels. Chaque ligne de la matrice représente les instances de la classe réelle, tandis que chaque colonne représente les instances de la classe prédite.

	Classe 1 prédiction	Classe 2 prédiction	...	Classe N prédiction
Classe 1 Réel	TP_1	FP_{12}	...	FP_{1N}
Classe 2 Réel	FN_{21}	TP_2	...	FP_{2N}
⋮	⋮	⋮	⋮	⋮
Classe N Réel	FN_{N1}	FN_{N2}	...	TP_N

TABLE 3.2 – Matrice de confusion

- VP_i : Vrais positifs pour la classe i (prédits correctement comme appartenant à la classe i).
- FP_{ij} : Faux positifs pour la classe i , où i est la classe prédite, mais la classe réelle est j .
- VN_i : Vrais négatifs pour la classe i (prédits correctement comme n'appartenant pas à la classe i).
- FN_{ji} : Faux négatifs pour la classe i , où la classe réelle est i , mais le modèle a prédit j .

En analysant les valeurs de la matrice de confusion, pour une classe i nous pouvons mieux comprendre les types d'erreurs commises par le modèle. A partir de cette matrice, plusieurs métriques résumant ces informations peuvent être déduites [33].

• **Exactitude (Accuracy)** : L'exactitude mesure le nombre total de prédictions correctes par rapport au nombre total d'échantillons.

$$\text{Exactitude} = \frac{\sum_{i=1}^n TP_i}{\text{Total des instances}} \quad (3.3)$$

• **Précision (Precision)** : La précision mesure la proportion d'échantillons positifs correctement classés parmi tous les échantillons positifs prédits par le modèle.

$$\text{Précision}_i = \frac{TP_i}{TP_i + \sum_{j \neq i} FP_{ij}} \quad (3.4)$$

• **Rappel (Recall)** : Le rappel mesure la proportion d'échantillons positifs correctement classés parmi tous les échantillons positifs réels dans l'ensemble de données.

$$\text{Rappel}_i = \frac{TP_i}{TP_i + \sum_{j \neq i} FN_{ji}} \quad (3.5)$$

• **F1-score** : Le F-score est une moyenne pondérée de la précision et du rappel. Il est calculé comme suit :

$$\text{F1-Score}_i = \frac{2 \times \text{Précision}_i \times \text{Rappel}_i}{\text{Précision}_i + \text{Rappel}_i} \quad (3.6)$$

• **Courbe ROC** : La courbe ROC (Receiver Operating Characteristic) est un graphique qui illustre la performance d'un modèle de classification binaire. Elle trace le taux de vrais positifs (True Positive Rate, TPR) contre le taux de faux positifs (False Positive Rate, FPR) à différents seuils de décision. Une courbe ROC plus proche du coin supérieur gauche indique une meilleure performance du modèle.

• **Taux de vrais positifs (TPR) ou sensibilité** : Il mesure la proportion de vrais positifs parmi les positifs réels :

$$\text{TPR} = \frac{TP}{TP + FN} \quad (3.7)$$

• **Taux de faux positifs (FPR)** : Il mesure la proportion de faux positifs parmi les négatifs réels :

$$\text{FPR} = \frac{FP}{FP + TN} \quad (3.8)$$

• **AUC (Area Under the Curve)** : La surface sous la courbe ROC. Plus l'AUC est proche de 1, meilleure est la performance du modèle. Un modèle parfait atteindrait une AUC de 1, tandis qu'un modèle ayant une performance équivalente au hasard aurait une AUC de 0,5.

3.1.3.3 Résultats

Les figures (3.8), (3.9) et (3.10) affichent les courbes ROC obtenues par FCDD sur les trois bases de données.

Pour la base Pill Qc, le modèle atteint une performance optimale avec une AUC de 1, indiquant une séparation parfaite des classes.

Pour la deuxième base Hazelnut, l'AUC est de 0.94, ce qui montre une très bonne distinction entre les classes, bien que quelques erreurs subsistent.

Pour la troisième base Wood, le modèle atteint presque la perfection avec une AUC de 0.99, ce qui indique une excellente performance dans la détection des anomalies.

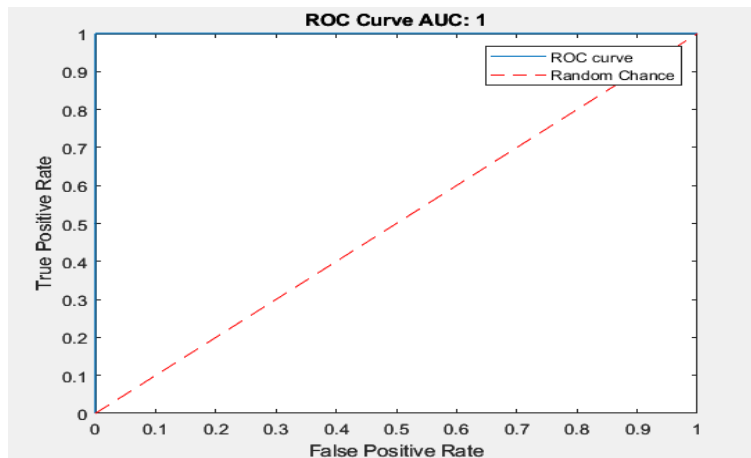


FIGURE 3.8 – Courbe ROC de la base Pill Qc (ligne diagonale rouge en pointillés indique la ligne de hasard)

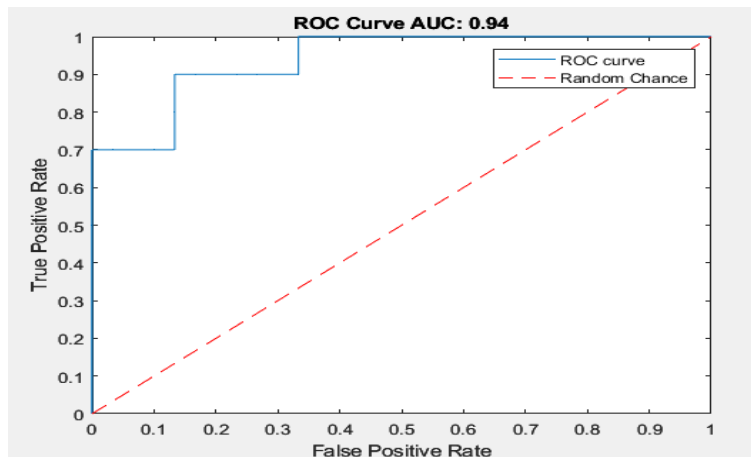


FIGURE 3.9 – Courbe ROC de la base Hazelnut (ligne diagonale rouge en pointillés indique la ligne de hasard)

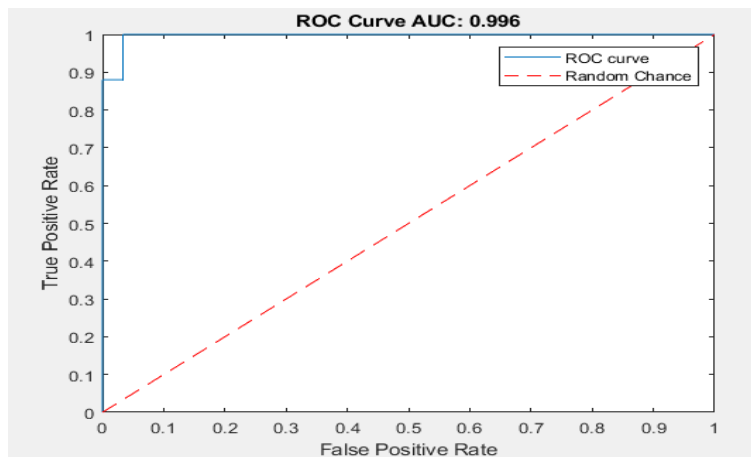


FIGURE 3.10 – Courbe ROC de la base Wood (ligne diagonale rouge en pointillés indique la ligne de hasard)

Les figures (3.11), (3.12) et (3.13) présentent les matrices de confusion obtenus par FCDD sur les données de test pour les trois bases.

Pour la première base Pill Qc, 147 images d'anomalies ont été correctement classées comme « Anomaly », tandis que 48 images normales ont été identifiées comme « Normal », 1 faux positif et sans faux négatifs.

Pour la seconde base Hazelnut, 8719 images d'anomalies ont été correctement classées, avec 4338 images normales identifiées comme « Normal ». 912 faux positifs et 1667 faux négatifs ont été observés.

Concernant la troisième base Wood, 25 images d'anomalies ont été correctement classées comme « Anomaly », tandis que 143 images normales ont été identifiées comme « Normal ». 4 faux positifs et sans faux négatifs ont été enregistrés.

Ces résultats démontrent l'efficacité du modèle dans la détection des anomalies sur les trois ensembles de données.

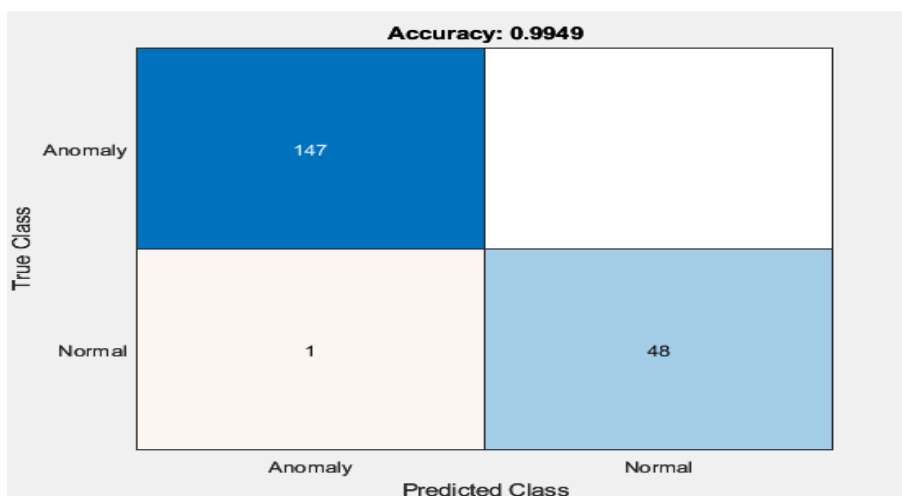


FIGURE 3.11 – Matrice de confusion de la base Pill Qc

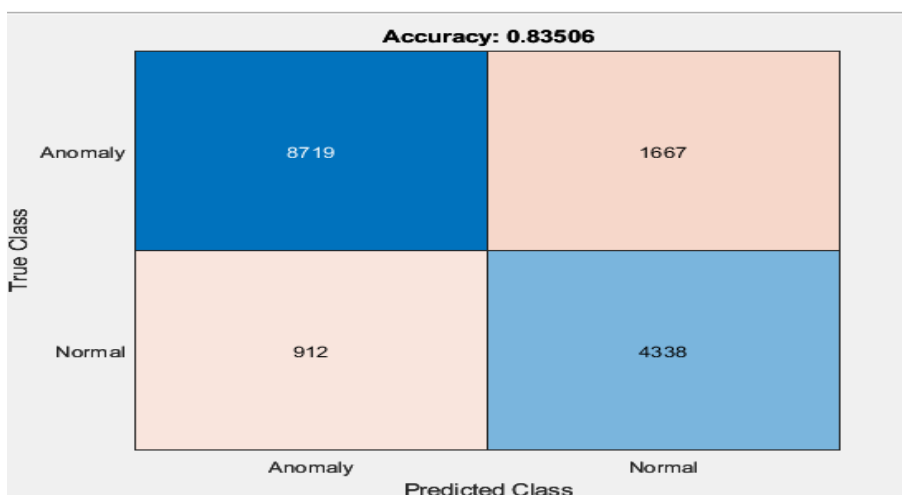


FIGURE 3.12 – Matrice de confusion de la base Hazelnut

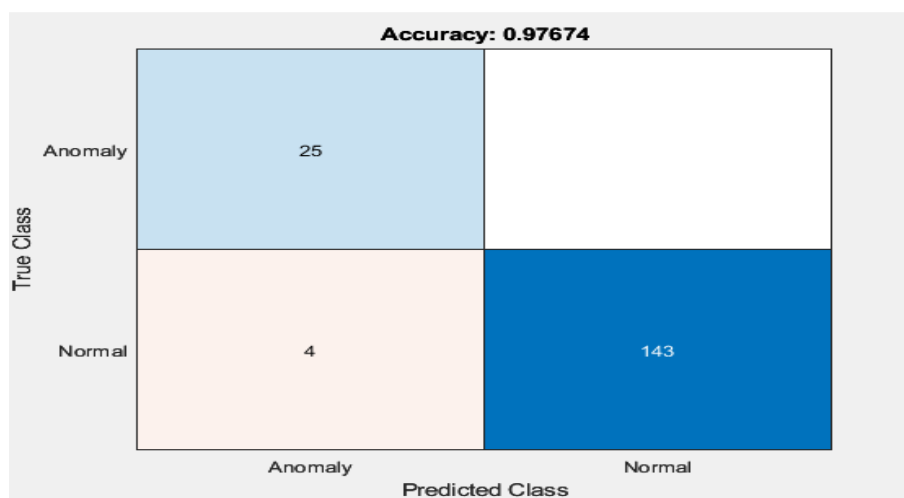


FIGURE 3.13 – Matrice de confusion de la base Wood

Le tableau (3.3) regroupe les métriques de performance (précision, recall, F1 score et accuracy) obtenues par FCDD sur les trois ensembles de données Pill Qc, Hazelnut et Wood. L'accuracy qui mesure le taux de prédictions correctes reste très élevé pour Pill Qc et Wood (>90%) et de (>80%) pour Hazelnut. La précision, recall et F1 score confirment la qualité des détections d'anomalies.

Métrique	Base Pill Qc	Base Hazelnut	Base Wood
Précision	0.99	0.94	1
Recall	0.93	0.87	0.96
F1 Score	0.96	0.90	0.98
Accuracy	0.99	0.84	0.98

TABLE 3.3 – Métriques de performance pour les trois bases de données

La figure (3.16) présente une image correctement identifiée comme anomalie (vrai positif), accompagnée de sa carte de chaleur correspondante. Ces cartes mettent en évidence les régions des images qui ont le plus contribué à la décision du modèle dans l'identification des anomalies. Ensemble, ces visualisations illustrent comment le modèle a discerné les anomalies dans les trois ensembles de données, soulignant les zones pertinentes ayant influencé les classifications.

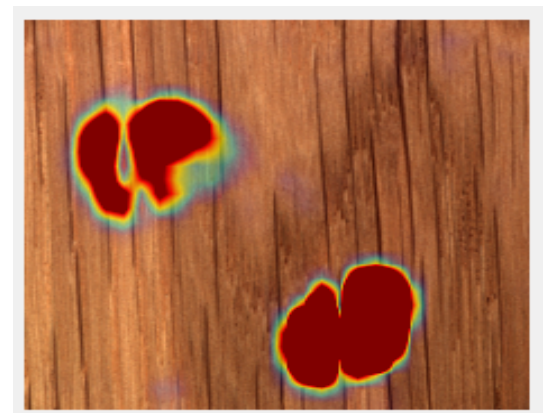
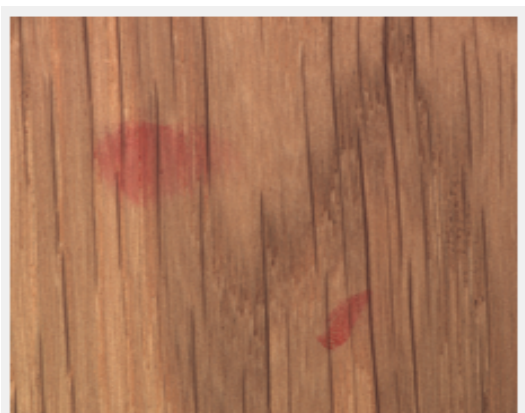
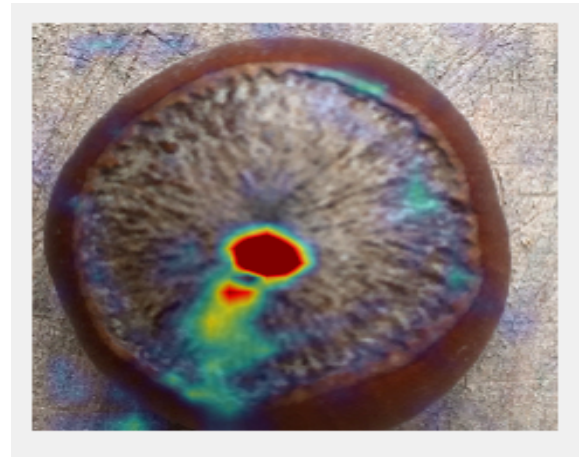
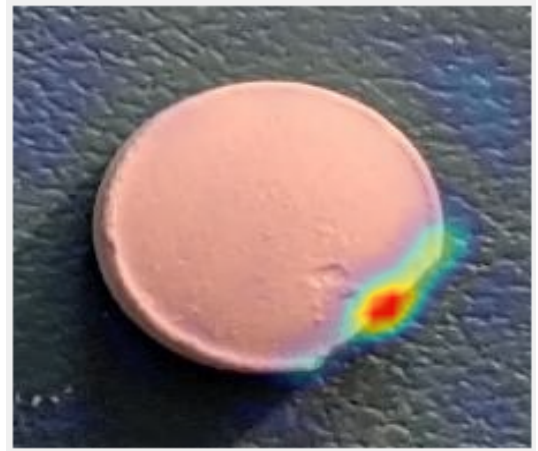


FIGURE 3.16 – Images de test classées correctement comme anomalie avec leurs carte de chaleur

3.2 Classification Multiclasses

Dans cette deuxième approche, nous abordons la reconnaissance des défauts de surface en utilisant une méthode de classification multiclassées. Contrairement à la méthode FCDD (Fully Convolutional Data Description), qui se concentre sur la classification d'une seule classe, cette approche vise à identifier et classer les défauts de surface en plusieurs catégories distinctes. L'objectif de la classification multiclassées est de distinguer entre différents types de défauts de surface dans les images. Plutôt que de déterminer simplement si une image présente un défaut ou non, cette méthode vise à identifier le type spécifique de défaut parmi plusieurs classes possibles.

3.2.1 Choix du réseau de neurones convolutif

Dans cette approche, nous avons utilisé le modèle pré-entraîné GoogleNet, également connu sous le nom d'Inception V1. Celui-ci est un réseau de neurones convolutif (CNN) développé par l'équipe de recherche de Google. Il se distingue par son architecture innovante, intégrant des blocs Inception qui permettent de capturer des caractéristiques à différentes échelles (Fig.3.17). Chaque bloc Inception combine des convolutions 1x1, 3x3, 5x5 et du max-pooling en parallèle, offrant ainsi une efficacité computationnelle supérieure. GoogLeNet, avec ses 22 couches, utilise des convolutions 1x1 pour réduire la dimensionnalité et le nombre de paramètres.

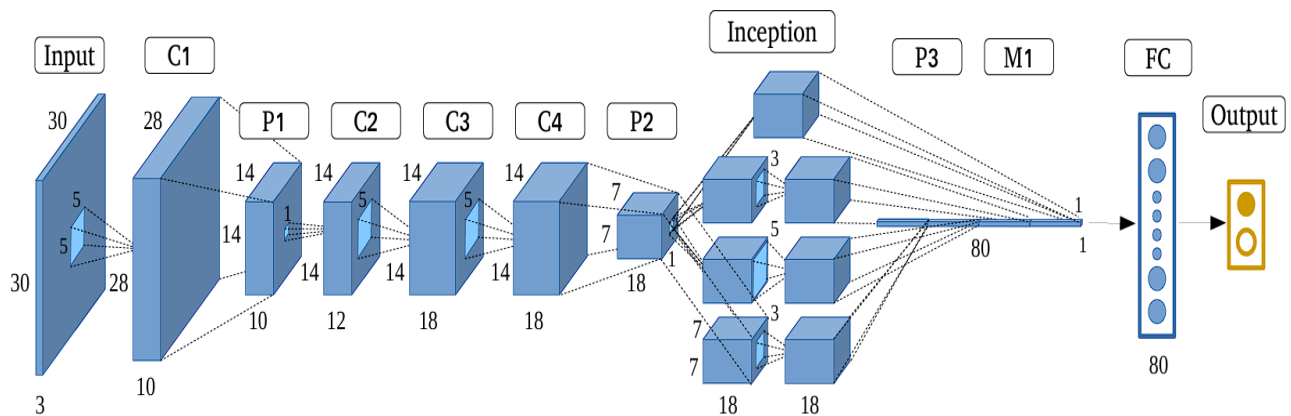


FIGURE 3.17 – Architecture de GoogleNet

Pour adapter GoogleNet à notre tâche de classification de défauts de surface, nous avons effectué plusieurs modifications. Tout d'abord, nous avons remplacé la couche entièrement connectée finale et la couche de sortie de classification, initialement configurées pour traiter les 1000 classes du dataset ImageNet. La nouvelle couche entièrement connectée a été adap-

tée au nombre de classes spécifiques de notre dataset. Le taux d'apprentissage est augmenté pour favoriser un apprentissage rapide des nouvelles classes. De même, la nouvelle couche de sortie de classification a été adaptée à notre problème de classification. Ces nouvelles couches ont été intégrées dans le réseau en remplacement des anciennes, ce qui permet au modèle de mieux apprendre et de classifier les défauts de surface présents dans nos données spécifiques. Nous avons également gelé les poids des premières couches convolutives du réseau pour conserver les caractéristiques générales apprises sur le vaste ensemble de données ImageNet. Cela permet d'accélérer le processus d'entraînement et assurer que les nouvelles couches apprennent les caractéristiques spécifiques nécessaires pour distinguer les différents défauts de surface. La figure (3.18) affiche sous forme de graphe les couches du réseau après modification. On peut observer la structure du réseau avec les couches ajoutées ou modifiées, et les connexions entre les différentes couches, ce qui permet de confirmer l'intégration des ajustements réalisés.

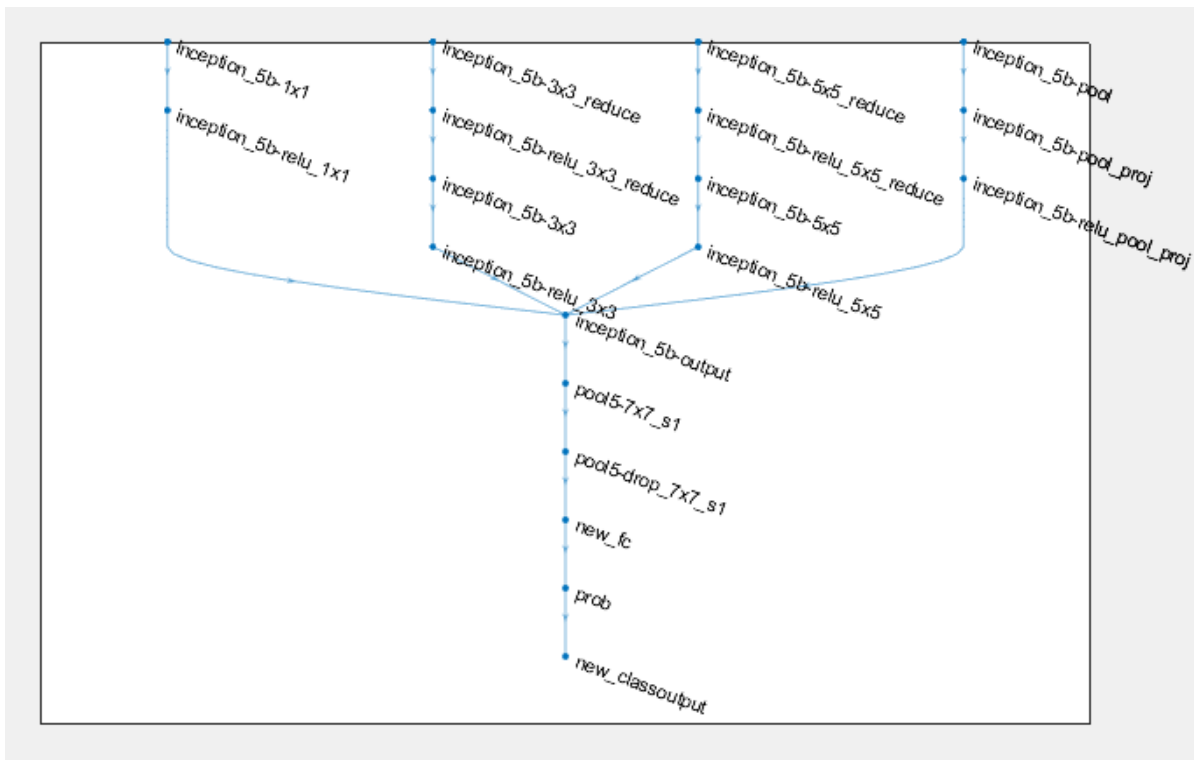


FIGURE 3.18 – Graphe de couches du modèle après modifications

3.2.2 Résultats de classification

Dans cette section, nous présentons les résultats obtenus pour la reconnaissance des défauts de surface en utilisant le modèle pré-entraîné GoogleNet sur les trois bases de données testées (Pill Qc, Hazelnut et Wood). L'évaluation des résultats est basée sur la matrice de confusion et l'accuracy. Pour réaliser les tests, nous avons divisé les données des deux bases Pill

Qc et Hazelnut en ensembles d'entraînement et de validation, en attribuant 70% des images à l'entraînement et 30% à la validation. Cependant, pour la base Wood, en raison du faible nombre d'images par classe, nous avons ajusté la répartition en attribuant 80 % des images à l'entraînement et 20 % à la validation, afin de mieux équilibrer les données (Tableau.3.4). De plus, la classe « Combined » a été supprimée car elle regroupait plusieurs types de défauts très différents, ce qui rendait difficile l'apprentissage du modèle. En raison de son manque de cohérence interne, elle perturbait la classification et diminuait la précision globale. En l'éliminant, nous avons pu mieux équilibrer les autres classes et améliorer la qualité des prédictions.

Base de Données	Résolution d'Image	Classe	Total d'Images	Base d'Entraînement	Base de Validation
Pill QC	225x225	Classe « Normal »	149	104	45
		Classe « Chip »	43	30	13
		Classe « Dirt »	138	97	41
Hazelnut	224x224	Classe « Quality_nuts »	5350	3745	1605
		Classe « Damaged_nuts »	5190	3633	1557
		Classe « Nuts_kernel »	5230	3661	1569
Wood	1024x1024	Classe « Good»	247	198	49
		Classe « Color »	8	6	2
		Classe « Scratch »	21	17	4
		Classe « Hole»	10	8	2
		Classe « Liquid »	10	8	2

TABLE 3.4 – Répartition des données pour l'évaluation du modèle de la classification

Les figures (3.19), (3.20) et (3.21) illustrent respectivement la progression de l'entraînement pour chacun des trois ensembles de données. Dans chaque cas, on constate que l'accuracy atteint presque 1, tandis que la perte (loss) diminue de manière significative, tendant vers zéro. Ces résultats montrent que, pour chaque base, le modèle a convergé rapidement, témoignant de sa capacité à reconnaître efficacement les caractéristiques des classes et à effectuer une classification presque parfaite.

Détection Des Défauts De Surface À L'aide Des CNNs

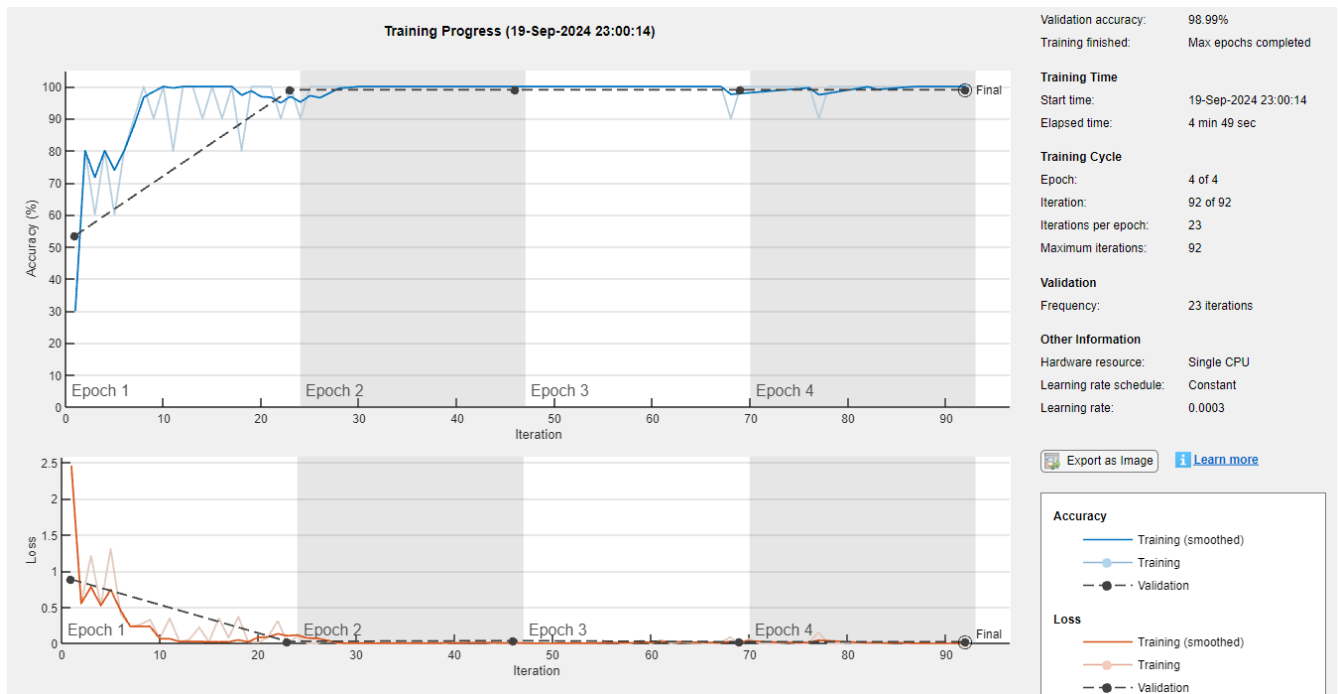


FIGURE 3.19 – Progression de l’accuracy et perte durant l’entrainement du CNN pour la base Pill Qc

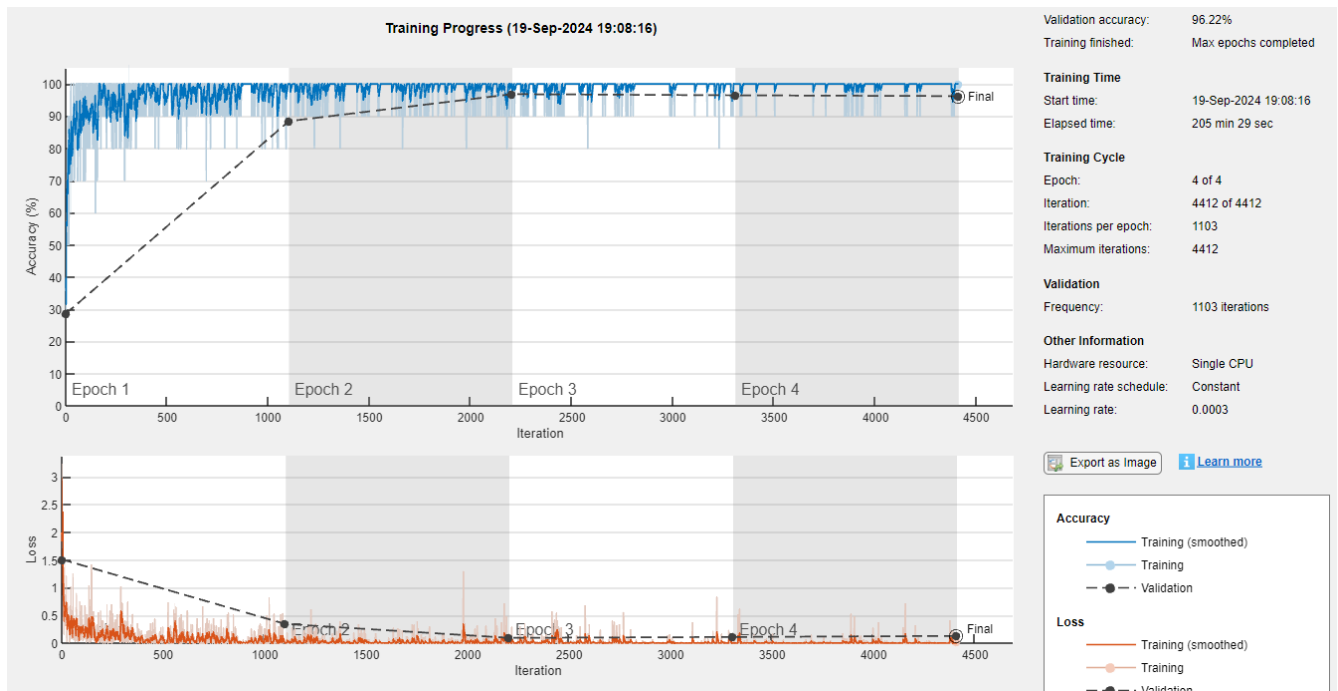


FIGURE 3.20 – Progression de l’accuracy et perte durant l’entrainement du CNN pour la base Hazelnut

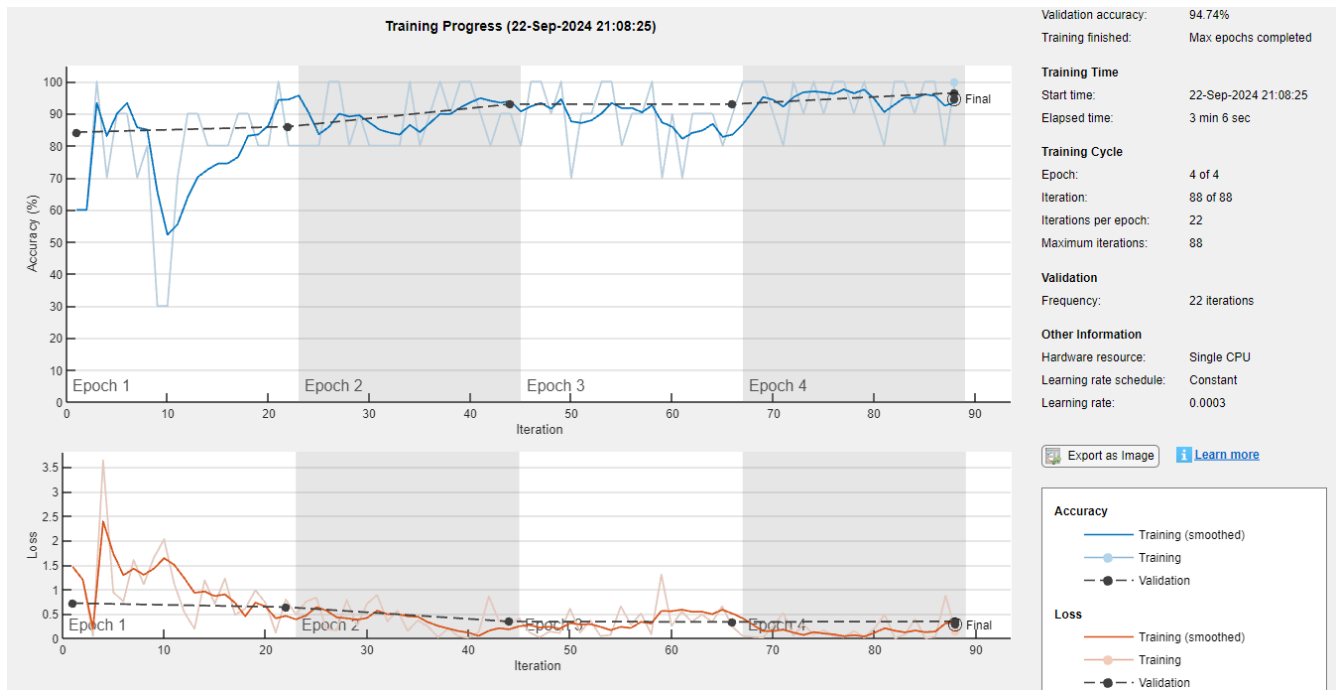


FIGURE 3.21 – Progression de l’accuracy et perte durant l’entraînement du CNN pour la base Wood

Les matrices de confusion obtenues sont regroupées dans la figure (3.22). Elles illustrent les performances du modèle sur les données de validation pour chacune des trois bases de tests. Ces matrices de confusion indiquent que les prédictions sont quasiment correctes sur les trois bases et que les taux de classification (accuracy) restent supérieures à 90% pour les trois bases

Confusion Matrix

	chip	dirt	normal	
chip	13 13.1%	0 0.0%	0 0.0%	100% 0.0%
dirt	0 0.0%	40 40.4%	0 0.0%	100% 0.0%
normal	0 0.0%	1 1.0%	45 45.5%	97.8% 2.2%
	100% 0.0%	97.6% 2.4%	100% 0.0%	99.0% 1.0%
	chip	dirt	normal	
	Target Class			

Matrice de confusion de la base Pill Qc

Output Class	damaged_n uts	nuts_k_ernel	quality_n uts	
damaged_n uts	1510 31.9%	11 0.2%	74 1.6%	94.7% 5.3%
nuts_k_ernel	17 0.4%	1558 32.9%	47 1.0%	96.1% 3.9%
quality_n uts	30 0.6%	0 0.0%	1484 31.4%	98.0% 2.0%
	97.0% 3.0%	99.3% 0.7%	92.5% 7.5%	96.2% 3.8%
	damaged_n uts	nuts_k_ernel	quality_n uts	
	Target Class			

Matrice de confusion de la base Hazelnut

Output Class	color	hole	liquid	normal	scratch	
color	2 3.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
hole	0 0.0%	2 3.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
liquid	0 0.0%	0 0.0%	2 3.5%	0 0.0%	0 0.0%	100% 0.0%
normal	0 0.0%	0 0.0%	0 0.0%	47 82.5%	2 3.5%	95.9% 4.1%
scratch	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 3.5%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	50.0% 50.0%	96.5% 3.5%
	color	hole	liquid	normal	scratch	
	Target Class					

Matrice de confusion de la base Wood

FIGURE 3.22 – Matrices de confusion des trois bases tests

4 Conclusion

Nous avons présenté dans ce chapitre deux méthodes de détection de défaut de surface basées sur les réseaux de neurones convolutifs (CNNs). la première est la détection des dé-

fauts par classification à une classe avec FCDD, La seconde est la reconnaissance des défauts par classification Multiclasses. Les résultats de tests obtenus sur les trois bases de données issues des différents domaines industriels pharmaceutique (Pill Qc), agroalimentaire (Hazelnuts) et bois (Wood) attestent de l'efficacité de la détection d'anomalies par le modèle FCDD et la capacité du modèle GoogleNet à reconnaître les différents types de défauts.

Conclusion générale

Conclusion Générale

Nous avons traité dans ce mémoire l'application des réseaux de neurones convolutifs (CNNs) à la détection des défauts de surface. Cette thématique reste très importante pour l'optimisation des processus de contrôle qualité dans l'industrie. Deux approches ont été abordées pour la détection des défauts de surface à base des CNNs. La première est la détection des défauts par classification à une classe, permettant de détecter la présence ou non d'une anomalie. La seconde est la reconnaissance des défauts par classification Multiclasses, permet d'identifier le type du défaut. L'évaluation de ces deux approches a été appliquée sur différents bases de données Pill Qc, Hazelnut et Wood. Les résultats obtenus par les deux méthodes sont appréciables. Les taux de détection obtenus par FCDD avoisine 90% et le taux de classification de GoogleNet dépasse les 96%.

Enfinement ce travail nous a permis d'une part de découvrir le vaste domaine des défauts de surface à travers les différents types de défaut, l'importance de leurs détection dans différents domaines industriels et la diversité des techniques employées allant de l'inspection manuelle jusqu'aux méthodes automatisées basées sur le traitement d'images et l'intelligence artificielle. D'autre part nous avons enrichi nos connaissances dans les CNNs en apprenant à adapter des modèles pré-entraînés comme VGG16 et GoogleNet à des cas spécifiques de détection de défauts. Comme perspectives, nous préconisons d'exploiter d'autres modèles de CNNs pré-entraînés et réaliser d'autres tests sur d'autres bases (textiles, verres, métaux,..., etc).

Bibliographie

Bibliographie

- [1] *Atteindre l'excellence en matière de fabrication grâce à des modèles de reconnaissance d'images pour la détection des défauts de surface*, <https://www.unite.ai/fr/achieving-manufacturing-excellence-with-image-recognition-models-for-surface-defect-detection/#:~:text=En%20moyenne%2C%20le%20co%3%BBt%20d,de%20la%20plus%20haute%20importance.>
- [2] *Comment améliorer la détection des défauts de fabrication ?*, <https://www.creaform3d.com/blog/fr/comment-ameliorer-la-detection-des-defauts-de-fabrication/>.
- [3] *comment classifier des défauts lors d'une inspection ?*, <https://www.v-trust.com/fr/nos-services/standard-d-inspection/classification-default.>
- [4] *comment le deep learning automatise-t-il les inspections pour l'automobile et les autres secteurs ?*, <https://www.cognex.com/fr-fr/blogs/deep-learning/deep-learning-defect-detection-tool.>
- [5] ALI RAID, *Contrôle non destructif, ressuage, magnétoscopie, radiographie, courants de foucault et ultrasons*, 2017.
- [6] M.DERRADJ Imad et M.AZOUZ Abdelouhab , *Étude de la détection des défauts dans les structures aéronautiques par capteur à courants de Foucault rotatif*, Mémoire de Fin d'études, Université de Bejaia, 2020.
- [7] *Détection des défauts - Utilisation de l'IA pour les inspections visuelles de la qualité*, <https://tulip.co/fr/blog/defect-detection-using-ai-for-visual-quality-inspections/>.
- [8] *Common illumination types*, <https://www.edmundoptics.com/knowledge-center/application-notes/illumination/choose-the-correct-illumination/>.
- [9] Madjaoui Amina et Fares Fadia, *Segmentation des images par contours actifs : application sur les Images Satellitaires à Haute Résolutions*, Mémoire de Fin d'études, Université Abou Bakr Belkaid Tlemcen, 2012.
- [10] *La segmentation des images*, <https://pfl-cepia.hub.inrae.fr/axe-images/tutoriel/la-segmentation-des-images.>

- [11] Redouane Lekhal, *Application des SVMs pour la reconnaissance d'extrasystoles*, Mémoire de Fin d'études, Université Mouloud Mammeri, 2015.
- [12] *Aperçu de la structure et de la fonction du neurone*, <https://fr.khanacademy.org/science/biologie-a-l-ecole/x5047ff3843d876a6:bio-5e-annee-sciences-de-base/x5047ff3843d876a6:bio-5-1h-le-neurone-et-le-systeme-nerveux-description/a/overview-of-neuron-structure-and-function>.
- [13] *McCulloch & Pitts publish the first mathematical model of a neural network*, <https://www.historyofinformation.com/detail.php?entryid=782>.
- [14] *Most popular activation functions for deep learning*, <https://lucidar.me/en/neural-networks/most-popular-activation-functions-for-deep-learning/>.
- [15] *Notions de base sur l'apprentissage profond*, <https://rpubs.com/alexvezeau/SCI1035-CH1>.
- [16] *Identification et commande des systèmes non linéaires*, <https://www.memoireonline.com/04/12/5750/Identification-et-commande-des-systemes-non-lineaires.html>.
- [17] Houacine Noura et Khelifa Nadia, *Classification des textures par les réseaux de neurones convolutifs*, Mémoire de Fin d'études, Université Mouloud Mammeri, 2018.
- [18] *Apprentissage par rétropropagation du gradient de réseaux neuromimétiques multicouches*, <https://pageperso.lis-lab.fr/bernard.espinasse/wp-content/uploads/2021/12/Retropropagation-Gradient-2018-4p.pdf>.
- [19] *Réseau de neurones convolutionnel CNN*, https://staff.univ-batna2.dz/sites/default/files/merzougui_ghalia/files/support_de_cours_-deep_learning-chapitre3-cnn.pdf.
- [20] *Pense-bête de réseaux de neurones convolutionnels*, <https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels>.
- [21] Mokri Mohammed Zakaria, *Classification des images avec les réseaux de neurones convolutionnels*, Mémoire de Fin d'études, Université Abou Bakr Belkaid Tlemcen, 2017.
- [22] *Comment choisir sa fonction de perte en Computer Vision ? - Partie 1*, [https://www.quantmetry.com/blog/choix-fonction-de-perte-en-computer-vision/#:~:text=Objectif%20de%20la%20fonction%20de%20perte&text=Cet%20apprentissage%20r%C3%A9sultat%20d'un,de%20convolution%20dans%20notre%20cas\)](https://www.quantmetry.com/blog/choix-fonction-de-perte-en-computer-vision/#:~:text=Objectif%20de%20la%20fonction%20de%20perte&text=Cet%20apprentissage%20r%C3%A9sultat%20d'un,de%20convolution%20dans%20notre%20cas)).
- [23] *Learning Parameters, Part 5 : AdaGrad, RMSProp, and Adam*, <https://towardsdatascience.com/learning-parameters-part-5-65a2f3583f7d>.
- [24] *Optimization in Deep Learning : AdaGrad, RMSProp, ADAM*, <https://artemoppermann.com/optimization-in-deep-learning-adagrad-rmsprop-adam/>.

- [25] *Qu'est-ce qu'un réseau de neurones convolutifs ?*, <https://www.ibm.com/fr-fr/topics/convolutional-neural-networks#:~:text=Qu'est%20ce%20qu'un%20r%C3%A9seau%20de%20neurones%20convolutifs%20%3F&text=IBM,et%20de%20reconnaissance%20d'objets>.
- [26] *Batch Normalization in Convolutional Neural Networks*, <https://www.baeldung.com/cs/batch-normalization-cnn>.
- [27] *Batch Normalization in Convolutional Neural Networks*, <https://blog.paperspace.com/batch-normalization-in-convolutional-neural-networks/>.
- [28] *The Essential Guide to Data Augmentation in Deep Learning*, <https://www.v7labs.com/blog/data-augmentation-guide>.
- [29] *Classification d'images : les réseaux de neurones convolutifs en Toute Simplicité*, <https://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicité>.
- [30] *L1 and L2 Regularization Explained, When to Use Them & Practical How-To Examples*, <https://spotintelligence.com/2023/05/26/l1-l2-regularization/>.
- [31] Philipp Liznerski, Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus-Robert Müller, *Explainable deep one-class classification*, arXiv preprint arXiv :2007.01760, 2021.
- [32] *Hazelnuts*, <https://data.mendeley.com/datasets/dvvx6kst3f/2>.
- [33] Ouachemi Ahcene et Ouchfoun Aness , *Classification de Panneaux de signalisation routière par les réseaux neuronaux Convolutifs*, Mémoire de Fin d'études, Université SAAD DAHLAB de BLIDA, 2019.