

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
المدرسة الوطنية العليا للتكنولوجيا  
Ecole national supérieur de technologie  
Département : Génie électrique Et Informatique Industrielle

Mémoire de fin d'étude en vue de l'obtention du diplôme

## D'Ingénieur d'Etat

Filière

**Automatique**

Spécialité

**Automatique et Informatique Industrielle**

Thème

# Développement d'un système d'inspection visuelle par apprentissage profond

Réalisé par

**TAIBECHE Ahmed**

**FEZZAI Oussama Chams Eddine**

Les membres de Jury :

<b>M. REZOUG Amar</b>	Président
<b>Mme. REBAI Karima</b>	Promoteur
<b>M.CHOUTRI Kheireddine</b>	Examineur
<b>Mme. CHOUAF Seloua</b>	Examineur

Alger, le 04/ 07/ 2023

Année universitaire 2022 –2023

## Dédicaces

“

Tout d'abord, je souhaite remercier tout puissant Allah, notre Dieu. Sans le soutien infailible d'Allah, cette réalisation n'aurait pas été possible

À la femme la plus merveilleuse et à la mère du monde, ma mère, je te remercie. Tu as toujours été ma force. Tes sacrifices, ton amour et ton dévouement pour nous élever et nous maintenir sur le droit chemin ont forgé la personne que je suis aujourd'hui. Ce travail t'est dédié, ma chère maman.

À mon père, qui a travaillé sans relâche chaque jour pour veiller à ce que je ne manque de rien, je suis infiniment reconnaissant. Ton appréciation inébranlable pour la connaissance et la science m'ont inspiré au-delà de toute mesure. Ce travail est le témoignage de ton soutien indéfectible et de ton amour, Papa.

À mes grands frères, Mohammed et Rida. Vous avez été bien plus que des frères pour moi ; vous avez été comme des seconds pères, toujours à mes côtés, et je suis reconnaissant de vos conseils tout au long de ce voyage.

À mes sœurs, qui ont été là pour moi et m'ont soutenu émotionnellement, je vous suis vraiment reconnaissant.

À mon petit frère, Motez, tu es non seulement un frère mais aussi un cher ami. Ton amitié et ton soutien indéfectible ont été d'une valeur inestimable pour moi.

Je remercie Allah de m'avoir donné une famille merveilleuse.

Et enfin, à toutes les personnes précieuses dans ma vie que je n'ai pas pu mentionner, je vous dédie également cette réalisation.

”

-Ahmed

“

Je souhaite dédier humblement cette œuvre à ma mère, qui est toujours présente pour me soutenir dans tous mes projets, et qui m'apporte sourire et réconfort dans les moments difficiles. Ma chère mère, j'espère que ce projet soit un témoignage de reconnaissance envers son dévouement, ses efforts inlassables, tes sacrifices et tous les efforts que tu as déployés pour moi. Je prie Allah le Tout-Puissant de lui accorder une longue vie remplie de santé et de bonheur, inchallah.

À mon père, Si Abd ElKader, je te remercie d'avoir toujours été mon allié dans les combats de ma vie. Je t'aime profondément. Rien n'aurait été possible dans ma vie sans ta lutte acharnée. Aucun mot ne peut véritablement exprimer ce que je ressens. Aucun geste ne peut refléter toute la gratitude que j'ai envers toi. Papa, c'est grâce à toi que je n'ai jamais manqué de rien dans la vie.

À mes frères, Taha, Fouzzi, Mohamed et Aissa, qui, malgré les distances, continuent de m'encourager à réaliser mes rêves les plus audacieux.

Enfin, je tiens à exprimer ma profonde gratitude envers ma superviseuse, Karima Rebai, pour son soutien et son accompagnement précieux tout au long de ce travail.

”

-Oussama

## Remerciements

Tout d'abord, nous exprimons notre gratitude à Dieu tout-puissant pour nous avoir accordé la force, le courage et la volonté de mener à bien ce travail.

Nous aimerions exprimer notre profonde gratitude envers Madame REBAI Karima pour avoir accepté de superviser notre Projet. Nous la remercions chaleureusement pour son écoute attentive et constante, sa patience et surtout sa confiance, ses commentaires bienveillants et ses conseils pertinents tout au long de la période de travail avec elle.

Nous souhaitons également remercier Monsieur DERMOUCHE Reda pour son aide précieuse dans la concrétisation de notre projet, ainsi que pour ses conseils et remarques.

Nos remerciements vont également au corps professoral et administratif du département de Génie Électrique et d'Informatique Industrielle de l'École Nationale Supérieure de Technologie. Leur enseignement de qualité et leurs efforts pour offrir une formation à la pointe de la technologie sont hautement appréciés.

Nous tenons à remercier tous les membres de notre jury d'avoir accepté de juger et d'évaluer ce travail de fin d'études, ainsi que d'avoir consacré du temps à la lecture de ce mémoire.

Enfin, nous exprimons nos sincères remerciements à nos familles et à nos amis pour leur soutien inconditionnel et leur présence constante.

En conclusion, nos remerciements vont également à toutes les personnes qui ont contribué, de près ou de loin, à la rédaction de ce mémoire.

**OUSSAMA & AHMED**

# Table des Matières

<b>Table des figures.....</b>	<b>I</b>
<b>Liste des tableaux .....</b>	<b>III</b>
<b>Liste des abréviations.....</b>	<b>IV</b>
<b>Introduction générale.....</b>	<b>1</b>
<b>I. CHAPITRE I : Description Des Systèmes de Détection des Défauts.....</b>	<b>3</b>
I.1 Introduction .....	4
I.2 Systèmes de détection de défauts et leur importance dans diverses industries .....	4
I.2.1 Principe général des systèmes de détection de défauts .....	4
I.2.2 Aperçu des défauts courants trouvés dans diverses industries .....	5
I.3 Architectures et emplacements des systèmes d'inspection visuelle .....	8
I.4 Techniques de détection des défauts .....	13
I.4.1 Techniques traditionnelles de détection des défauts par traitement d'image ..	13
I.4.2 Techniques de détection de défauts par Intelligence artificielle .....	14
I.5 Évolution des systèmes d'inspection des défauts .....	16
I.6 Détection de défauts dans des bouteilles : étude de cas .....	17
I.7 Conclusion.....	19
<b>II.CHAPITRE II : Deep Learning.....</b>	<b>20</b>
II.1 Introduction .....	21
II.2 Introduction au Deep Learning.....	21
II.2.1 Historique.....	21
II.2.2 Position du Deep Learning dans l'IA .....	22
II.2.3 Caractéristiques attractives du DL .....	23
II.2.4 Deep Learning dans la détection de défauts de bouteilles d'eau .....	23
II.2.5 Méthodes d'apprentissage en DL.....	24
II.3 Réseaux de neurones artificiels : principe et notions fondamentales .....	26
II.3.1 Réseaux de neurones : Composition et Architecture .....	26
II.3.2 Processus d'apprentissage.....	27
II.3.3 Qualité d'apprentissage et méthodes de régularisation.....	31

## Table des Matières

---

II.4	Réseaux de neurones à convolution (Convolutional Neural Network CNN).....	35
II.4.1	Architecture des CNN.....	35
II.5	Présentation des modèles réussit de classification d'image .....	36
II.6	Évaluation des performances.....	43
II.7	Conclusion.....	45
<b>III.</b>	<b>Chapitre III : Inspection Visuelle Des Bouteilles D'eau Par Apprentissage Profond...</b>	<b>46</b>
III.1	Introduction.....	47
III.2	Préparation du dataset.....	48
III.3	Types de défauts considérés.....	49
III.4	Augmentation du dataset.....	49
III.5	Prétraitement des images .....	51
III.5.1	Redimensionnement d'image.....	52
III.5.2	Normalisation des images.....	52
III.6	Détection de défauts : solutions proposées .....	53
III.6.1	Modèles proposés .....	53
III.6.2	Apprentissage des modèles.....	54
III.6.3	Choix des hyperparamètres.....	56
III.7	Implémentation et évaluations .....	56
III.7.1	Environnement de développement.....	56
III.7.2	Pseudo-codes d'entraînement et d'évaluation .....	59
III.8	Evaluation des différents modèles .....	61
III.8.1	Discussion des résultats d'entraînement des modèles .....	61
III.8.2	Temps d'entraînement .....	65
III.9	Performances des modèles face aux données de test.....	65
III.10	Inspection de la partie haute d'une bouteille : un réseau spécialisé (InsPart) ..	67
III.11	Architecture finale du système de détection des défauts 'InsComb' pour les deux modèles en parallèle .....	68
III.12	Conclusion .....	70
<b>IV.</b>	<b>CHAPITRE IV : Prototype du Système d'inspection visuelle.....</b>	<b>71</b>
IV.1	Introduction.....	72
IV.2	Architecture globale du système complet.....	72
IV.2.1	Cahier des charges .....	72
IV.2.2	Description du système complet d'inspection.....	72

## Table des Matières

---

IV.3	Environnement de développement (système de détection de défauts) .....	75
IV.4	Automate Programmable Industriel API (PLC) .....	79
IV.4.1	Communication entre PLC et Système de détection de défauts .....	80
IV.4.2	Programme de PLC .....	81
IV.4.3	Les Alarmes .....	83
IV.5	Interface Homme Machine (HMI) .....	83
IV.6	Page Web et Base de Données MySQL.....	87
IV.7	Conclusion .....	89
	<b>Conclusion Générale .....</b>	<b>90</b>
	<b>Bibliographie.....</b>	<b>91</b>
	<b>ملخص.....</b>	<b>95</b>
	<b>Résumé.....</b>	<b>95</b>
	<b>Abstract .....</b>	<b>95</b>

# Table des figures

Figure I.1 : Système Cognex pour le contrôle de qualité de bouteille d'eau [6].....	6
Figure I.2 : Production des récipients en verre à usage pharmaceutique contrôlée par système de vision [7] .....	6
Figure I.3 : Contrôle de qualité automatisé du support moteur à l'aide de MetraSCAN3D-R[8] ..	7
Figure I.4 : Machine de test de la résistance au lavage des couleurs [10] .....	8
Figure I.5 : Contrôle de qualité par vision dans la fabrication des CPU [12] .....	9
Figure I.6 : caméra balayage surface et Line [14].....	10
Figure I.7 : Emplacement et types de sources lumineuses [15] .....	11
Figure I.8 : Système d'inspection visuelle [18].....	13
Figure I.9 : Système d'inspection visuelle CAD2SCAN [28] .....	16
Figure I.10 : Système d'inspection visuelle InspektoS70 [29].....	17
Figure I.11 : Système d'inspection visuelle CAD2SCAN [30] .....	17
Figure II.1 : Position de DL dans le domaine d'IA [40] .....	22
Figure II.2 : Les techniques d'apprentissage et exemples d'application.....	26
Figure II.3 : Neurone artificiel (Perceptrone) [46].....	27
Figure II.4 : Perceptrone multicouche [47] .....	27
Figure II.5 : Perceptrone multicouche avec une couche cachée .....	28
Figure II.6 : Early stopping .....	33
Figure II.7 : Principe de Dropout [56].....	34
Figure II.8 : Quelques techniques d'augmentation de données [57] .....	34
Figure II.9 : Architecture de GAN [59].....	35
Figure II.10 : Architecture de CNN .....	37
Figure II.11 : Max-Pooling avec 2x2 filtre et un pas de 2 .....	37
Figure II.12 : Opération de Flatten.....	37
Figure II.13 : Connexion résiduelle [60].....	38
Figure II.14 : Architecture de ResNet50 .....	39
Figure II.15 : Block de convolution et résiduelle dans ResNet50.....	40
Figure II.16 : architecture d'Inception_V3 [64].....	42
Figure II.17 : architecture de SqueezeNet [65] .....	43
Figure III.1 : Etapes de développement d'une approche de détection des défauts par transfert d'apprentissage.....	48
Figure III.2 : Exemples des défauts dans les bouteilles d'eau considérés dans notre dataset .....	50
Figure III.3 : Exemples des images générées par des fonctions utilisées pour l'augmentation des données (images).....	51
Figure III.4 : Etapes de préparation du Dataset.....	53
Figure III.5 : Architecture de l'InceptionV3 modifié.....	54



## Table des Figures

---

Figure III.6 : Architecture du modèle développé .....	54
Figure III.7 : Évolution de l'Accuracy et Loss pendant l'apprentissage des différents modèles en utilisant ADAM pour Lr initiale 0.0001.....	63
Figure III.8 : Evolution de l'Accuracy et de l'erreur des modèles ResNet.....	64
Figure III.9 : Evolution de l'Accuracy et de l'erreur des modèles Inception V3.....	64
Figure III.10 : Evolution de l'Accuracy et l'erreur du modèle développé (Scratch) .....	65
Figure III.11 : l'évolution de l'Accuracy et l'erreur pour le modèle spécialisée ( <b>InsPart</b> ) dans la phase d'entraînement et validation.....	68
Figure III.12 : schéma du système totale " <b>InsComb</b> " les deux modèles en parallèle .....	69
Figure IV.1 : Architecture globale du système d'inspection.....	74
Figure IV.2 : La chaine d'inspection.....	74
Figure IV.3 : La chambre d'inspection .....	74
Figure IV.4 : Test de modèle <b>Insbot</b> en temps réels.....	74
Figure IV.5 : Test de modèle <b>InsPart</b> en temps réels .....	75
Figure IV.6 : Test des <b>InsComb</b> en parallèle en temps réels.....	78
Figure IV.7 : Principe de communication entre l'API et Système de détection de défauts .....	78
Figure IV.8 : Bloc de données (DB1) .....	81
Figure IV.9 : Grafctet de la chaine d'inspection d'un point de vue commande.....	81
Figure IV.10 : Bloc OB1.....	84
Figure IV.11 : Page d'accueil de l'IHM.....	86
Figure IV.12 : Page de contrôle de l'IHM .....	86
Figure IV.13 : Page des Alarmes de l'IHM .....	87
Figure IV.14 : Dashboard pour la visualisation des données .....	88
Figure IV.15 : La base de données MySQL.....	89

# Liste des tableaux

Tableau I.1 : Avantages et inconvénients des inspection en fin de ligne et en ligne de production .....	12
Tableau II.1 : Matrice de confusion.....	44
Tableau III.1 : Modèles entraînés et leurs caractéristiques .....	55
Tableau III.2 : Hyperparamètres utilisés pour l’entraînement des différents modèles .....	56
Tableau III.3 : Comparaison entre les performances des cinq modèles en fonction de différents paramètres. ....	62
Tableau III.4 : Matrice de confusion du test pour chaque modèle entraîné .....	66
Tableau III.5 : Performances des modèles dans la phase de test.....	66
Tableau III.6 : matrice de confusion du modèle entraîné InceptionV3 pour chaque défaut .....	67
Tableau III.7 : Évaluation de l’apprentissage par transfert d’un InceptionV3 ( <b>InsPart</b> ) pour la détection des défauts dans la partie haute d’une bouteille .....	67
Tableau III.8 : Matrice de confusion et performance du modèle spécialisé selon plusieurs critères .....	68
Tableau III.9 : Performances du modèle spécialisé dans la phase de test.....	68
Tableau III.10 : Matrice de confusion et performances du système d’inspection final .....	69
Tableau III.11 : Performances du système d’inspection final.....	69
Tableau III.12 : Matrice de confusion du modèle final pour chaque défaut .....	69
Tableau IV.1 : Caractéristiques des GPUs utilisés.....	76
Tableau IV.2 : Mesure de performance en temps réel des deux modèles .....	79
Tableau IV.3 : Adresses des variables pour Grafct .....	82
Tableau IV.4 : Les Alarmes utilisées dans L’API et IHM .....	83

## Liste des abréviations

<b>AE</b>	Auto-encodeurs
<b>API</b>	Automate programmable industrielle
<b>CCD</b>	Charge-Coupled Device
<b>CMOS</b>	Complementary Metal-Oxide-Semiconductor
<b>CUDA</b>	Compute Unified Device Architecture
<b>DB</b>	bloc de données
<b>END</b>	Essais Non Destructifs
<b>FB</b>	bloc fonctionnel
<b>FC</b>	Function Chart
<b>FN</b>	False Negative
<b>FP</b>	False Positive
<b>GAN</b>	Generative Adversarial Network
<b>GPU</b>	Graphics Processing Unit
<b>GUI</b>	Graphique user interface
<b>HMI</b>	Human-Machine Interface
<b>InsBot</b>	Inspection Bottle
<b>InsComb</b>	Inspection Combination
<b>InsPart</b>	Inspection upper Part
<b>LQV</b>	Low-Quality Visual Inspection
<b>Lr</b>	Learning rate (Taux d'apprentissage)
<b>ONNX</b>	Open Neural Network Exchange
<b>PLC</b>	Programmable Logic Controller
<b>POC</b>	Phase-Only Correlation
<b>ResNet</b>	ResNet Residual Network
<b>RNN</b>	Réseaux de neurones récurrents
<b>SGD</b>	Stochastic Gradient Descent
<b>SQL</b>	Structured Query Language
<b>TP</b>	True Negative
<b>TN</b>	True Positive

# Introduction générale

L'inspection visuelle des défauts est une méthode couramment utilisée dans divers secteurs industriels pour repérer les anomalies présentes dans les produits. Elle implique des traitements spéciaux pour la détection des défauts tels que des rayures, des fissures ou des erreurs de fabrication. L'objectif est d'améliorer la qualité en identifiant et en corrigeant les défauts dès leur apparition, ce qui permet de réduire les coûts et de satisfaire les clients. Cependant, il est important de souligner que l'inspection visuelle peut présenter certaines limites en raison des erreurs potentielles commises par les opérateurs humains et de son coût élevé. Dans ce contexte, l'automatisation de ce processus a été envisagée depuis plusieurs années. Des systèmes automatiques basés sur la vision par ordinateur sont intégrés actuellement dans des chaînes de production en améliorant la réalisation de cette étape.

L'apprentissage profond a atteint ces dernières années des niveaux de précision dans des applications de reconnaissance jamais atteints auparavant dans différents domaines. Plusieurs solutions ont été développées en utilisant en particulier des données visuelles (images ou vidéos). Ce qui encourage des industriels pour s'orienter vers l'intégration des systèmes de contrôle de qualité dans différents procédés industriels afin de répondre aux attentes des clients. Le degré d'importance de ce type de solution est très lié aux types de produits et de clients visés. Lorsqu'il s'agit des produits liés à la santé des êtres humains, de fortes exigences sont imposées.

Le développement des solutions en utilisant de nouvelles méthodes de l'apprentissage profond nécessite généralement l'utilisation de grandes datasets. La collecte et la préparation de ce types de données est très difficile. Par conséquent, dans ces situations, il est recommandé souvent d'opter pour une autre méthode d'apprentissage. Il s'agit d'effectuer un transfert d'apprentissage et de l'affiner selon les objectifs ciblés.

Le transfert d'apprentissage est une méthode utilisée en apprentissage automatique qui consiste à tirer parti des connaissances acquises dans une tâche pour améliorer les performances dans une autre tâche. Cette approche permet de réduire la quantité de données d'apprentissage nécessaires, d'accélérer le processus d'apprentissage et de faciliter l'adaptation à de nouveaux domaines ou de nouvelles tâches. En transférant les connaissances d'un modèle pré-entraîné, il est possible d'améliorer ses performances pour une nouvelle tâche en utilisant les caractéristiques apprises précédemment. Cela permet d'optimiser l'utilisation des ressources informatiques en exploitant des parties de modèles déjà entraînés. Le transfert d'apprentissage offre donc une méthode puissante pour transférer des connaissances existantes et de les appliquer à de nouveaux problèmes en apprentissage automatique.

Dans le domaine de l'industrie et le remplissage des bouteilles d'eau, le développement d'un système d'inspection visuelle (IV) vise à garantir la qualité des produits en détectant les défauts et les anomalies. Étant donné que cette industrie connaît une demande croissante en eau embouteillée, avec des taux de production élevés, il est essentiel de maintenir des normes élevées en termes d'apparence, d'intégrité structurelle et de sécurité des bouteilles d'eau. L'application de l'inspection

visuelle dans cette industrie a une relation directe avec les clients. Les consommateurs attendent des produits de haute qualité et sans défauts. La détection et l'élimination des défauts avant que les bouteilles n'atteignent les rayons des magasins renforcent la confiance des consommateurs dans la marque et réduisent les retours et les réclamations. Un système d'inspection visuelle efficace garantit également la sécurité des produits, en évitant que des bouteilles défectueuses ne parviennent aux clients.

L'objectif du travail présenté dans ce mémoire est de développer un système capable de détecter automatiquement des défauts et classer chaque bouteille soit Propre ou Défectueuse en utilisant un algorithme d'apprentissage profond. Notre approche consiste principalement à exploiter des modèles pré-entraînés et connus par leurs performances et d'effectuer un transfert d'apprentissage.

Le présent mémoire est structuré de la manière suivante : il débute par une présentation des systèmes de contrôle qualité des produits, en mettant en avant les techniques traditionnelles ainsi que les méthodes récentes basées sur l'apprentissage profond. Pour chaque technique, nous mentionnons quelques travaux de référence et décrivons les architectures des systèmes et leur position dans la chaîne de production. De plus, nous mettons en évidence les équipements nécessaires à l'inspection visuelle et présentons quelques produits disponibles sur le marché pour la détection des défauts.

Par la suite, nous abordons les théories de l'intelligence artificielle dans le chapitre 2. Nous mettons l'accent sur l'apprentissage profond, ainsi que les techniques utilisées à chaque étape de l'apprentissage, allant de la préparation de données à l'évaluation d'un modèle. Nous décrivons également l'approche de transfert d'apprentissage, sa pertinence et les modèles pré-entraînés utilisés dans notre travail. Nous décrivons les caractéristiques de chaque modèle, ainsi que leurs avantages.

Les deux derniers chapitres sont consacrés à la présentation de notre solution pour résoudre la problématique de la détection des défauts des bouteilles d'eau. Dans le premier, détaillons les étapes à suivre dans le développement logiciel de notre approche allant de la préparation des données, traitements effectués sur les images, architectures et entraînement et l'analyse des résultats obtenus. Nous présentons donc les modèles pré-entraînés considérés, les métriques d'apprentissage et l'évaluation des modèles entraînés afin de sélectionner le meilleur modèle en fonction des résultats souhaités. Le dernier chapitre décrit le prototype complet où le modèle développé précédemment est combiné avec d'autres modules pour être exploitable dans une chaîne de production. L'implémentation matérielle de la solution proposée se fait en utilisant une carte NVIDIA Jetson Nano et Automate Programmable Industriel (API). Cette mise en œuvre permet de gérer la chaîne de production où se trouvent la chambre d'inspection et les caméras d'acquisition d'images. Pour tenir compte des aspects de supervision et contrôle local et distant, nous avons construit une interface homme-machine (IHM), une page web et base de données MySQL pour sauvegarder les données de manière permanente.

# **CHAPITRE I**

## **Description Des Systèmes de Détection des Défauts**

## **I.1 Introduction**

La détection des défauts est un aspect essentiel dans plusieurs industries, allant de la fabrication à la production en passant par la distribution et la logistique, les défauts peuvent avoir des conséquences importantes sur la qualité du produit, la santé des consommateurs, les coûts de production et la réputation de l'entreprise. C'est pourquoi les systèmes d'inspection visuelle sont de plus en plus utilisés pour détecter les anomalies à différentes phases du processus de production.

L'architecture d'un système d'inspection visuelle dépend de plusieurs facteurs de l'application et des exigences de l'utilisateur, la plupart des systèmes d'inspection visuelle utilisent les ressources matérielles et logicielles pour l'acquisition des images des produits et traiter ces images et identifier les anomalies existe dans ces produits.

Les solutions d'inspection visuelle traditionnelles sont composées d'une caméra ou d'autres capteurs, d'un éclairage approprié et des logiciels pour le traitement des images capturées. Cependant, avec l'évolution rapide des techniques basées sur l'intelligence artificielle, et spécialement l'apprentissage profond, de nouveaux systèmes d'inspection visuelle sont apparus, offrant des performances plus élevées et une meilleure précision dans la détection des défauts.

## **I.2 Systèmes de détection de défauts et leur importance dans diverses industries**

### **I.2.1 Principe général des systèmes de détection de défauts**

Les systèmes de détection de défauts sont utilisés pour identifier et reconnaître les éventuels défauts dans les produits manufacturés. Ces systèmes peuvent être utilisés pour améliorer la qualité des produits en identifiant les défauts tôt dans le processus de fabrication. Les systèmes de détection de défauts peuvent également être utilisés pour améliorer l'efficacité des processus de fabrication en identifiant et en retirant les produits défectueux de la ligne de production. Il existe une variété de systèmes de détection de défauts et qui sont souvent développés pour répondre aux exigences particulières de différentes catégories d'industries et de produits.

Afin de tester la qualité d'un produit sans l'endommager, l'approche à utiliser consiste à effectuer des tests ou essais non destructifs (END). Les techniques d'END comprennent les essais en utilisant différents types de capteurs : radiographie, ultrason, vision. [1]

La méthode conventionnelle de détection visuelle de défaut, qui est toujours utilisée, est le type le plus basique de système de détection de défauts. Elle consiste à désigner des personnes pour inspecter visuellement les produits et détecter les éventuels défauts. L'inspection visuelle peut être efficace pour détecter les grands défauts, mais elle n'est pas aussi efficace pour détecter les petits défauts [2]. Elle dépend considérablement de l'état des inspecteurs. D'autres méthodes automatiques de détection de défauts ont développé en utilisant différents types de capteurs à savoir :

Vision par ordinateur : Il s'agit d'un type plus sophistiqué de système de détection de défauts qui utilise des caméras et des logiciels de traitement d'images, y compris des méthodes classiques et intelligentes, pour identifier les défauts. La vision par ordinateur peut être plus efficace que

l'inspection visuelle pour détecter les petits défauts, mais elle peut être plus coûteuse à mettre en œuvre [1]. L'utilisation de l'information visuelle permet l'inspection de la qualité de beaucoup de produit dont l'apparence est très importante comme pour certains produits agroalimentaires, industriels et beaucoup de produits destinés au grand public.

Les systèmes d'inspection par ultrasons sont des systèmes qui utilisent des ondes ultrasonores pour pénétrer les matériaux et identifier les défauts [1]. On peut les utiliser pour détecter les défauts de surface et de sous-surface, tester l'épaisseur des matériaux et la profondeur des défauts à travers l'analyse des échos.

Les systèmes d'inspection par particules magnétiques et courant de Foucault sont des systèmes qui utilisent des champs magnétiques pour identifier les défauts dans les matériaux ferromagnétiques et le courant Foucault pour identifier les défauts dans les matériaux conducteurs [3] En plaçant des matériaux ferromagnétiques dans un champ magnétique, la discontinuité dans les matériaux peut provoquer des discontinuités magnétiques. On peut aussi identifier des défauts dans des matériaux à travers le test de sa conductivité. Le type de système de détection de défauts le mieux adapté à une application particulière dépendra de plusieurs facteurs, notamment le type de produit ou de matériau inspecté, la taille et la complexité du produit ou du matériau et le niveau de précision souhaité [4].

Dans ce projet, nous nous sommes intéressés en particulier à l'inspection visuelle des défauts. De ce fait, le reste de chapitre présentera des aspects liés à ce type de système.

### **I.2.2 Aperçu des défauts courants trouvés dans diverses industries**

Le choix d'une technologie particulière pour l'inspection des défauts dans différents secteurs dépend principalement de la nature des défauts et comment peut-on les détecter. Nous présentons ci-dessous quelques défauts couramment rencontrés dans certains types d'industrie.

#### **a. Industrie alimentaire [5]**

- Détection des contaminants : Dans l'industrie alimentaire, des techniques telles que la spectrométrie de masse, chromatographie et les tests microbiologiques sont utilisées pour détecter les contaminants potentiels tels que les bactéries, les virus, les moisissures, les résidus de pesticides, les métaux lourds, etc.
- Contrôle de la qualité des ingrédients : Des méthodes d'analyse chimique et biologique sont utilisées pour évaluer la qualité des ingrédients utilisés dans la production alimentaire. Cela peut inclure des tests de composition nutritionnelle, de fraîcheur, d'authenticité des ingrédients, de teneur en allergènes, etc.
- Suivi des procédés de fabrication : Des technologies de surveillance et de contrôle sont utilisées pour garantir le respect des normes de fabrication alimentaire. Cela peut inclure des systèmes de Contrôle de température, de pression, de débit, de pH, etc., afin d'assurer la conformité et la sécurité des produits alimentaires. La figure I.1 présente système de vision pour le contrôle les bouteilles d'eau dans la chaîne de production.





Figure I.1 : Système Cognex pour le contrôle de qualité de bouteille d'eau [6]

**b. Industrie pharmaceutique [7]**

- Tests de stabilité des médicaments : Des études de stabilité sont menées pour évaluer la durée de conservation et l'intégrité des médicaments. Cela implique des tests de dégradation, de dissolution, de pureté, de toxicité, etc., pour s'assurer que les médicaments restent efficaces et sûrs pendant toute leur durée de vie.
- Contrôle des impuretés : Des méthodes d'analyse sophistiquées, telles que la chromatographie en phase liquide haute performance (HPLC) et la spectroscopie infrarouge (IR), sont utilisées pour détecter et quantifier les impuretés présentes dans les médicaments. Cela garantit la qualité et la sécurité des produits pharmaceutiques.
- Validation des procédés de fabrication : Des études de validation des procédés sont réalisées pour s'assurer que les méthodes de production pharmaceutique sont fiables et reproductibles et même les outils utilisés sont fiables et pur de bactéries et d'organismes nocifs comme indique dans la figure I.2. Cela comprend la validation des équipements, des paramètres de fabrication, des méthodes d'analyse, etc., afin de garantir la qualité et l'uniformité des produits pharmaceutiques.

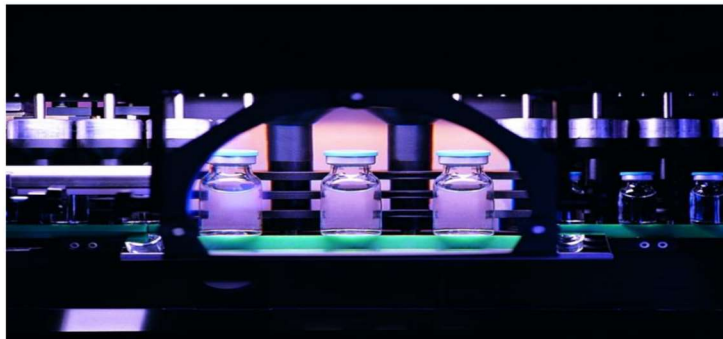


Figure I.2 : Production des récipients en verre à usage pharmaceutique contrôlée par système de vision [7]

**c. Industrie automobile [8]**

- Contrôle qualité des composants : Des techniques d'inspection et de mesure, telles que la vision par ordinateur, la tomographie, la métrologie dimensionnelle, sont utilisées pour vérifier la conformité des composants automobiles aux spécifications comme montrée dans la figure I.3. Cela comprend la détection de défauts, la mesure de tolérance, la vérification des propriétés matérielles, etc.
- Tests de performance et de sécurité : Des essais et des simulations sont réalisés pour évaluer la performance et la sécurité des véhicules. Cela inclut des tests de collision, de résistance structurelle, de freinage, de maniabilité, d'émissions, etc., afin de garantir la conformité aux normes réglementaires et d'assurer la sécurité des conducteurs et des passagers.
- Surveillance et diagnostic des systèmes embarqués : Des systèmes de diagnostic embarqués et des capteurs sont utilisés pour surveiller en temps réel le fonctionnement des composants et des systèmes automobiles. Cela permet de détecter et de signaler les défauts potentiels, tels que les pannes de moteur, les dysfonctionnements électriques, les problèmes de freins, etc., facilitant ainsi la maintenance préventive et la réparation rapide.

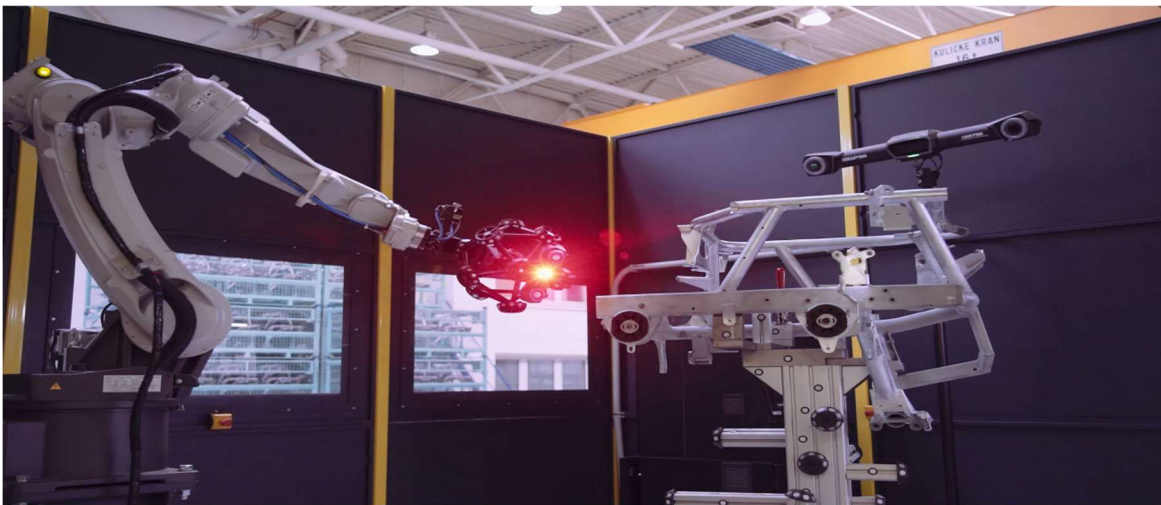


Figure I.3 : Contrôle de qualité automatisé du support moteur à l'aide de MetraSCAN3D-R[8]

**d. Industrie du textile [9]**

- Contrôle qualité des matériaux textiles : Des tests de solidité des coutures, de rétrécissement, de composition des fibres, etc., sont réalisés pour évaluer la qualité des matériaux utilisés dans la fabrication textile.
- Détection des défauts de tissage ou de tricotage : Des techniques d'inspection visuelle, d'imagerie ou de vision par ordinateur sont utilisées pour détecter les défauts de tissage ou de tricotage tels que les fils cassés, les trous, les irrégularités de motif, les points manqués, etc.

- Tests de performance des textiles : Des tests de performance sont effectués pour évaluer les propriétés des textiles à l'aide d'équipements spéciaux, comme illustré dans la figure I.4, afin de tester la résistance au lavage des couleurs, l'abrasion, la déchirure, l'imperméabilité, et bien d'autres aspects. Ces tests garantissent la conformité aux normes de qualité et permettent aux fabricants de produire des textiles durables et de haute qualité.



Figure I.4 : Machine de test de la résistance au lavage des couleurs [10]

### e. Industrie électronique [11]

- Tests de composants électroniques : Des tests rigoureux sont effectués sur les composants électroniques tels que les circuits intégrés, les transistors, les diodes, etc., pour s'assurer de leur bon fonctionnement et de leur conformité aux spécifications techniques.
- Contrôle de la qualité de la fabrication : Des procédures de contrôle qualité sont mises en place pour surveiller et évaluer les différentes étapes de fabrication des produits électroniques. Cela inclut le contrôle de la qualité des soudures par système vision comme présenté dans la figure I.5, l'inspection visuelle des cartes de circuits imprimés, la vérification des connexions et des câblages, etc.
- Conformité aux normes de sécurité et aux réglementations : L'industrie électronique est soumise à des normes et des réglementations strictes en matière de sécurité des produits. Des tests de conformité aux normes de sécurité électrique, de compatibilité électromagnétique (CEM), de certification environnementale (RoHS), etc., sont effectués pour garantir la sécurité des utilisateurs et la conformité aux réglementations en vigueur.

## I.3 Architectures et emplacements des systèmes d'inspection visuelle

Le système d'inspection visuelle est principalement composé d'une caméra, de sources d'éclairage unité de traitement. Il peut être installé à différentes positions dans la chaîne de production pour l'inspection de la qualité des produits.



Figure I.5 : Contrôle de qualité par vision dans la fabrication des CPU [12]

### I.3.1 Caméras

Selon les caractéristiques clés à considérer pour l'inspection de la qualité des produits, il existe différents types de caméras vidéo industrielles à savoir :

#### a. Les types des caméras industrielles

- **Les caméras à balayage linéaire**

Ils traitent des bandes très étroites, une à la fois. Elles capturent des rangées uniques de pixels en succession rapide (Figure I.6.a), tandis que la caméra peut se déplacer (balayage) pour percevoir tout l'objet ou ce sont les objets à inspecter qui passent sous la caméra. Les caméras à ligne sont utilisées lorsqu'il est nécessaire d'analyser de nombreux éléments dans une rangée sans s'arrêter, ou lorsqu'on essaie de capturer une image très longue.

Comme les systèmes de balayage linéaire nécessitent des pièces en mouvement pour construire l'image, ils sont souvent bien adaptés aux produits en mouvement continu ou aux applications à haute vitesse [13].

- **Les caméras à balayage de surface**

Ils utilisent une matrice de pixels pour capturer une scène (Figure I.6.b), similaire à la façon dont fonctionne l'appareil photo d'un smartphone. Ils ont une résolution fixe et sont plus faciles à installer et à configurer. Les caméras à balayage de surface sont plus polyvalentes et sont souvent utilisées pour capturer rapidement une zone définie, où une image peut être segmentée en régions pour rechercher des objets spécifiques.

Ce type de caméra convient le mieux aux applications où l'objet est stationnaire, même si ce n'est que momentanément, ou se déplaçant à un rythme lent. C'est également la technique la plus appropriée lors de l'inspection d'objets individuels, où une seule image suffit pour capturer toute la zone d'intérêt [13].

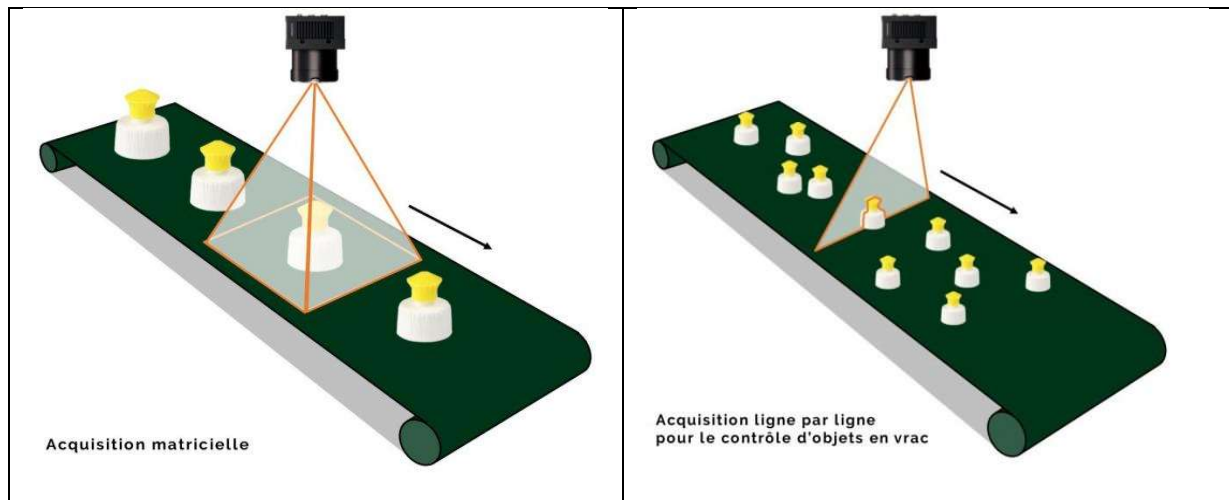


Figure I.6 : caméra balayage surface et Line [14]

### b. Le choix des objectifs d'une caméra (Lens)

A divers effets sur la qualité de l'image, les performances et le coût du traitement de l'image. Des objectifs plus coûteux qui produisent des images de haute résolution et de haute qualité peuvent réduire les temps de traitement, tandis que des objectifs qui produisent des résolutions plus faibles sont plus rentables mais l'exploitation des images et l'extraction des informations utiles nécessitent plus de traitement. Les facteurs à considérer lors du choix des objectifs :

- Tailles du capteur et du cercle d'image : Il est important de choisir un objectif suffisamment grand pour ne pas créer de vignette sur le bord de la zone du capteur, mais pas si grand qu'il y ait une zone "supplémentaire" dans le cercle d'image qui n'est pas utilisée.
- Résolution : pour une image haute résolution, un objectif avec un nombre élevé de mégapixels capable de résoudre la taille des pixels est suffisant.
- Longueur focale : elle détermine l'angle de vue, ou la taille de la zone qu'un objectif peut capturer. La longueur focale doit correspondre à la taille du capteur.
- Ouverture et conditions d'éclairage : Il est important de choisir un objectif qui a suffisamment d'ouvertures (réglages qui permettent à l'ouverture d'être plus ouverte ou fermée) en fonction des conditions d'éclairage de l'environnement. Plus le nombre d'ouvertures est bas, plus l'ouverture sera large, mais cela réduira également la profondeur de champ.
- Filtres : certains filtres peuvent être fixés à l'objectif, comme un filtre polarisant pour aider à réduire l'éblouissement sur les objets réfléchissants. Certains filtres colorés peuvent également aider à mettre en évidence des zones importantes sur les objets inspectés [13].

### c. Type de capteur de caméra

Un autre aspect important à considérer pour sélectionner la caméra utilisée est le type de capteur. Une caméra peut être construite autour d'une technologie de capteur d'image à dispositif à couplage de charge (CCD) ou à semi-conducteur complémentaire à oxyde métallique (CMOS). Les deux ont la même tâche de transformer la lumière (photons) en signaux électriques (électrons).

Dans les capteurs CCD, les charges des pixels sensibles à la lumière sont déplacées et converties en signaux. Les charges des pixels, créées par exposition à un semi-conducteur, sont transportées vers un convertisseur A/N central. Dans les capteurs CMOS, un condensateur en tant que stockage de charge est mis en parallèle avec chaque pixel individuel. Ce condensateur est chargé avec l'exposition de chaque pixel par son courant photoélectrique. La tension créée dans le condensateur est proportionnelle à la luminosité et au temps d'exposition. Actuellement, le marché des capteurs s'oriente de plus en plus vers la technologie CMOS.

### I.3.2 Sources d'éclairage

Plusieurs types de sources lumineuses sont couramment utilisés en vision industrielle [13]. Les lampes quartz-halogène, les lampes fluorescentes et les diodes électroluminescentes (LED) sont de loin les sources lumineuses les plus utilisées dans les systèmes d'inspection visuelle [15]. À partir du placement et des types de sources lumineuses, différentes configurations sont possibles comme le montre la Figure I.7. Par exemple,

- **L'éclairage arrière (Figure I.7.a)** : génère un contraste instantané car il crée des silhouettes sombres sur un fond lumineux. Les utilisations les plus courantes sont pour détecter la présence/absence de trous et d'espaces, placer ou orienter des pièces ou mesurer des objets. Souvent, il est utile d'utiliser une lumière monochrome, telle que rouge, verte ou bleue, avec une polarisation du contrôle lumineux si une détection précise (sous-pixel) la détection de bord devient nécessaire
- **L'éclairage diffus (champ lumineux complet)** : est le plus couramment utilisé sur des échantillons brillants spéculaires ou à réflectivité mixte où une lumière uniforme mais multidirectionnelle est nécessaire. Plusieurs implémentations de l'éclairage diffus sont généralement disponibles, mais il existe différents types (Figure 7.b et Figure 7.c), avec un dôme/cylindre hémisphérique ou sur axe étant les plus courants.
- **L'éclairage en champ sombre (Figure I.7.d)** : est peut-être la technique la moins bien comprise de toutes, bien que ces techniques puissent être utilisées dans la vie quotidienne. Par exemple, l'utilisation des phares d'automobile repose sur la lumière incidente à faibles angles sur la surface de la route, réfléchissant en retour des petites imperfections de surface et également des objets proches.

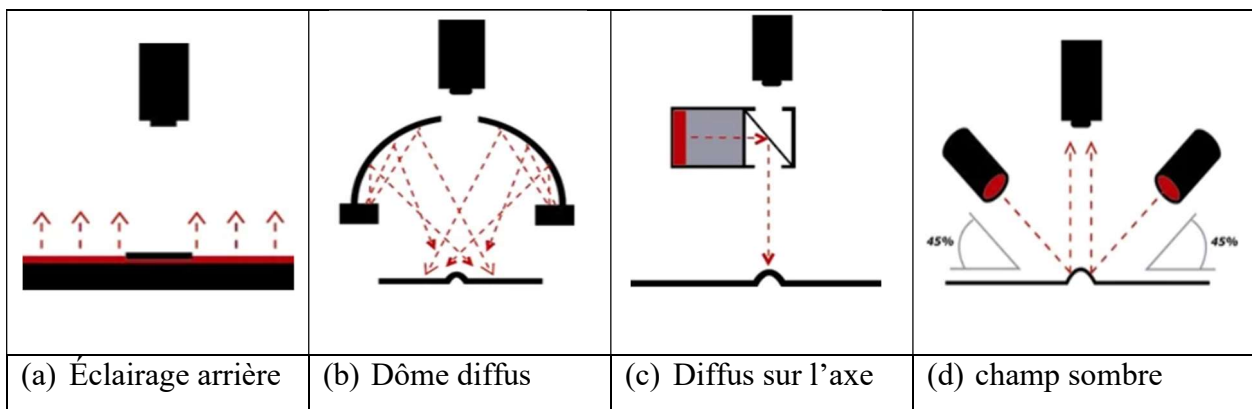


Figure I.7 : Emplacement et types de sources lumineuses [15]

### I.3.3 Emplacement du système d'inspection visuelle

Les systèmes d'inspection basés sur la vision sont placés différemment sur les lignes de fabrication en fonction [16].

Des facteurs tels que le type d'inspection nécessaire, le produit, les défauts spécifiques qu'ils inspectent fabriqué et l'agencement de la ligne de production doivent être pris en compte lors de la détermination de l'emplacement du système. Il est important En outre, s'assurer que le système d'inspection visuel soit est placé dans une zone avec un éclairage suffisant et des vibrations minimales pour aboutir peut aider à garantir des inspections précises [17]. Selon l'emplacement de ces systèmes dans une chaîne de production, on peut distinguer les deux configurations suivantes.

- **Inspection visuelle en fin de ligne**

Les systèmes d'inspection visuelle sont généralement placés à la fin de la ligne de production, où ils peuvent inspecter le produit final après que tous les processus de fabrication aient été achevés.

- **Inspection en ligne**

Les systèmes d'inspection visuelle peuvent être placés entre les processus de fabrication pour inspecter le produit à chaque étape de la production. L'emplacement du système d'inspection visuelle doit tenir compte de la vitesse de la ligne de production, du type d'inspection nécessaire et du produit fabriqué. Le tableau I.1 résume les avantages et les inconvénients de chaque système [13].

Tableau I.1 : Avantages et inconvénients des inspection en fin de ligne et en ligne de production

	Inspection en fin de ligne	Inspection en ligne
Avantages	<p>Permet l'inspection du produit final</p> <p>Vitesses d'inspection plus rapides</p> <p>Intégration facile dans les lignes de production existantes</p>	<p>Détection des défauts en temps réel</p> <p>Capacité à corriger les défauts en temps réel</p> <p>Réduction des coûts de rebut et de retouche, peut identifier quel processus de fabrication a causé le défaut</p> <p>L'objectif est une fabrication sans défaut, avec un accent sur l'inspection en cours de processus et le contrôle qualité à la source plutôt qu'à la fin du processus.</p>
Inconvénient	<p>Incapacité à corriger les défauts en temps réel</p> <p>Peut entraîner une perte de matériaux et de temps de production si des défauts sont trouvés après la production</p>	<p>Vitesses d'inspection plus lentes</p> <p>Peut nécessiter une ligne de production dédiée pour l'inspection</p> <p>Peut-être plus compliqué à mettre en place par rapport à l'inspection en fin de ligne</p>

### 1.3.4 Nombre de caméra à utiliser dans un système d'inspection

L'inspection mono-image est une approche simple et directe de l'inspection visuelle qui peut être facilement intégrée dans les processus de production existants. Cependant, elle peut avoir des limites dans la détection des défauts dans des géométries ou des textures complexes ou pour détecter les changements au fil du temps. L'inspection multi-images est une approche plus complexe qui peut capturer plus d'informations à partir de différentes perspectives, la rendant plus précise et plus robuste en inspectant des objets complexes. Cependant, elle peut être plus difficile à intégrer dans les processus de production existants et peut nécessiter un développement logiciel plus sophistiqué. Le choix entre l'inspection mono-image et l'inspection multi-images dépendra des exigences spécifiques du processus de fabrication et des types de défauts à détecter.

Le schéma présenté dans la figure I.8 illustre un système d'inspection visuelle complet utilisé pour évaluer la qualité des produits. Ce système se compose d'une caméra haute résolution qui capture des images détaillées des produits en cours d'inspection. Les images sont ensuite traitées par un logiciel d'analyse d'image avancé qui identifie les défauts et les anomalies. Les résultats de l'inspection sont ensuite envoyés à un contrôleur de chaîne de production pour prendre décision. Ce système d'inspection visuelle offre une solution rapide et précise pour garantir la conformité des produits aux normes de qualité établies.

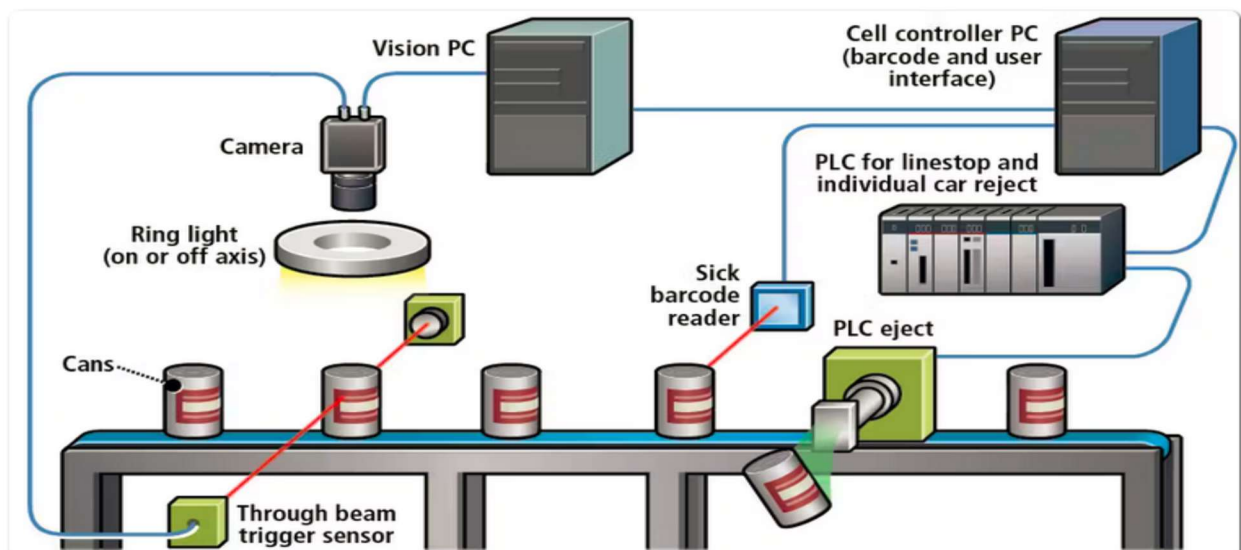


Figure I.8 : Système d'inspection visuelle [18]

## I.4 Techniques de détection des défauts

### I.4.1 Techniques traditionnelles de détection des défauts par traitement d'image

#### a. Inspection Optique Automatisée (Automatic Optical Inspection, AOI)

Un système d'inspection optique entièrement automatisé comprend des configurations matérielles et logicielles. La configuration matérielle comprend un capteur d'image et des réglages d'éclairage responsables de l'acquisition de l'image numérique, tandis que la partie logicielle met



en œuvre un algorithme d'inspection pour extraire les caractéristiques des images acquises et les classer en défectueuses ou non défectueuses en fonction des exigences de l'utilisateur. Un mécanisme de tri peut être utilisé pour séparer les produits défectueux des bons.

Une inspection automatique des circuits imprimés et des composants électroniques. Cela permet de détecter les défauts de manière rapide, précise et fiable, tout en réduisant les coûts et en améliorant la qualité des produits [19]

### **b. Détection des contours par Canny (Canny Edge Detection)**

La détection des contours est une technique de traitement d'images qui permet la détection des bords d'objets dans une image. La méthode développée par John F. Canny en 1986 est une des méthodes de détection de contours les plus utilisés en traitement d'images [20].

### **c. Corrélation de phase seule (Phase-Only Correlation ou POC)**

Une technique de traitement d'images utilisée pour trouver des correspondances entre deux images. Elle est souvent utilisée pour le suivi d'objets en mouvement dans des séquences vidéo, la détection de motifs répétitifs dans des images et d'autres applications nécessitant une correspondance précise entre deux images.

La corrélation de phase seule peut être utilisée pour la détection des défauts en analysant les variations de phase de la lumière réfléchiée par la surface des objets. Cette technique permet de détecter des défauts tels que les irrégularités de surface, les fissures, les bulles d'air, les rayures, etc., en comparant la phase de référence avec la phase mesurée. Les différences de phase indiquent la présence de défauts, ce qui permet de les détecter et de les localiser avec précision [21].

### **d. Reconnaissance de défauts par LQV (pour Low-Quality Visual Inspection)**

Une méthode de reconnaissance de défauts basée sur l'analyse d'images de faible qualité. Cette méthode utilise des techniques de traitement d'images pour détecter et classifier les défauts sur des images de faible qualité.

La reconnaissance de défauts par LQV utilise des algorithmes de traitement d'images pour améliorer la qualité des images et extraire des caractéristiques importantes pour la reconnaissance des défauts [22].

## **I.4.2 Techniques de détection de défauts par Intelligence artificielle**

L'apprentissage automatique et profond sont de plus en plus utilisés pour la détection de défauts dans les industries manufacturières, la surveillance de l'état des équipements et d'autres domaines où la détection précoce de défauts est cruciale pour minimiser les coûts de maintenance et maximiser la qualité et la fiabilité des produits.

L'apprentissage automatique est une technique d'analyse de données qui utilise des algorithmes pour apprendre à partir de données et faire des prédictions ou prendre des décisions sans être explicitement programmé. L'apprentissage en profondeur est une branche de l'apprentissage automatique qui utilise des réseaux de neurones profonds pour traiter des données complexes.

Dans la détection de défauts, l'apprentissage automatique et l'apprentissage profond peuvent être utilisés pour détecter automatiquement les anomalies et les défauts à partir de données de différentes natures. Plusieurs modèles ont été proposés ces dernières années et qui peuvent être entraînés à partir de données de défauts connus pour identifier les anomalies et les défauts dans de nouvelles données. Nous citons ci-dessous quelques exemples.

### **a. Auto-encodeurs [23]**

Les auto-encodeurs sont un type de réseau de neurones qui apprend à reconstruire leurs données d'entrée aussi fidèlement que possible. La détection d'anomalies peut être réalisée en entraînant un auto-encodeur sur des données normales, puis en l'utilisant pour classifier les nouvelles données comme normales ou anormales.

### **b. Transfert de connaissances [24]**

Le transfert de connaissances est une technique qui consiste à utiliser un modèle pré-entraîné sur une tâche similaire pour améliorer ses performances en réalisant une nouvelle tâche. Dans le contexte de la détection d'anomalies, le transfert de connaissances peut être utilisé pour extraire des caractéristiques pertinentes à partir d'un modèle pré-entraîné, qui peut ensuite être utilisé pour détecter les anomalies dans un nouvel ensemble de données.

### **c. Réseaux Génératifs Antagonistes (GAN) [25]**

Les réseaux génératifs antagonistes sont des techniques d'apprentissage profond qui implique deux réseaux neuronaux : un générateur et un discriminant. Le générateur est entraîné à générer des données similaires à celles de l'ensemble de données d'entraînement, tandis que le discriminant est entraîné à distinguer les données générées du reste des données. Dans le contexte de la détection d'anomalies, les GAN peuvent être utilisés pour générer des données synthétiques qui peuvent être utilisées pour identifier les anomalies dans un nouvel ensemble de données.

### **d. Réseaux de neurones récurrents (RNN) [26]**

Les réseaux de neurones récurrents sont une famille de réseaux neuronaux capables de traiter des données séquentielles, telles que des séquences temporelles. Les RNN peuvent être utilisés pour détecter les anomalies dans les séquences temporelles en apprenant à reconstruire la séquence d'entrée à partir de ses caractéristiques. Les anomalies peuvent alors être identifiées en comparant la séquence reconstruite à la séquence d'entrée originale

### **e. Auto-encodeurs variationnels (VAE) [27]**

Les auto-encodeurs variationnels sont une extension des autoencodeurs standards qui apprennent à représenter les données en termes de distributions plutôt que de valeurs exactes. Dans le contexte de la détection d'anomalies, les VAE peuvent être utilisés pour générer des données synthétiques qui peuvent être utilisées pour identifier les anomalies dans un nouvel ensemble de données.

## I.5 Évolution des systèmes d'inspection des défauts

Récemment, plusieurs entreprises et start-ups utilisent l'apprentissage automatique dans le développement des systèmes de détection de défauts pour améliorer la qualité des produits et réduire les coûts de maintenance. Par exemple, dans l'industrie manufacturière, les algorithmes d'apprentissage automatique peuvent être utilisés pour détecter les défauts de surface sur les pièces métalliques, les défauts dans les soudures, les fissures dans les structures en béton, etc.

De même, dans la surveillance de l'état des équipements, les algorithmes d'apprentissage automatique peuvent être utilisés pour détecter les défauts dans les roulements, les courroies, les engrenages, les pompes, les turbines, etc. En utilisant des capteurs pour surveiller les vibrations, les températures, les pressions, les débits, etc.

Plusieurs start-ups ont développé et commercialisé plusieurs produits. Nous présentons ci-dessous quelques solutions d'inspection visuelle.

### I.5.1 CAD2SCAN [28]

Le logiciel du système Kitov CAD2SCAN simplifie considérablement la configuration des systèmes d'inspection visuelle en utilisant une planification d'inspection automatisée basée sur CAD. En combinant CAD2SCAN (Figure I.9) avec les solutions Kitov Smart Planner et l'apprentissage profond, les clients peuvent automatiser des solutions d'inspection qui sont trop difficiles et longues à programmer et à déployer manuellement. Particulièrement utile pour les pièces avec des géométries complexes, CAD2SCAN permet aux responsables qualité de définir leurs exigences d'inspection directement sur le modèle CAD de la pièce.



Figure I.9 : Système d'inspection visuelle CAD2SCAN [28]

### I.5.2 Inspektos70[29]

Le système INSPEKTO S70 (Figure I.10) est une solution industrielle de contrôle de qualité visuelle pilotée par l'IA. Il dispose de trois moteurs d'IA qui permettent une grande polyvalence et simplicité d'utilisation. Le premier moteur s'occupe des paramètres optiques et s'adapte dynamiquement aux changements en production. Le deuxième moteur se charge de l'identification

des pièces, tandis que le troisième moteur est responsable de la détection des défauts. Le système est capable d'apprendre en continu via l'apprentissage profond.



Figure I.10 : Système d'inspection visuelle InspektoS70 [29]

### I.5.3 Cognex Tech [30] In-Sight D900 Series

Le système de vision In-Sight D900, illustré par la figure I.11, utilise des outils avancés basés sur l'IA pour résoudre des tâches difficiles de reconnaissance optique de caractères (OCR), de vérification d'assemblage et de détection de défauts. Le traitement se fait sur l'appareil, ce qui élimine le besoin d'un PC, simplifiant le déploiement d'applications et offrant la puissance de l'apprentissage profond aux non-programmeurs.



Figure I.11 : Système d'inspection visuelle CAD2SCAN [30]

## I.6 Détection de défauts dans des bouteilles : étude de cas

La détection de défauts dans les bouteilles est une pratique essentielle pour garantir la qualité et la sécurité des produits alimentaires et des boissons. Les fabricants doivent s'assurer que chaque bouteille est exempte de tout défaut qui pourrait affecter la qualité du produit ou constituer un risque pour la santé des consommateurs. Les défauts courants des bouteilles comprennent

l'absence de bouchons, les déformations, les niveaux inappropriés et les objets étrangers à l'intérieur de la bouteille.

L'absence de bouchons est l'un des défauts les plus courants des bouteilles. Les bouchons jouent un rôle crucial dans la préservation de la qualité des boissons en empêchant l'air et d'autres éléments indésirables de pénétrer dans la bouteille. Si une bouteille est produite sans bouchon ou avec un bouchon défectueux, cela peut entraîner une altération de la qualité du produit et des risques pour la santé des consommateurs [31].

Les déformations sont un autre défaut courant des bouteilles. Les bouteilles peuvent subir des déformations pendant le processus de production ou d'emballage, ce qui peut affecter leur forme et leur étanchéité. Les bouteilles déformées peuvent également causer des problèmes lors du remplissage ou de l'étiquetage, ce qui peut entraîner des pertes de produits et des coûts supplémentaires pour les fabricants [31].

Le niveau de remplissage représente également un défaut important des bouteilles. Les fabricants doivent s'assurer que chaque bouteille est remplie à la quantité appropriée pour garantir une qualité constante du produit et éviter des pertes financières. Des niveaux de remplissage inappropriés peuvent également avoir des conséquences sur la santé et la sécurité des consommateurs [31].

Les objets étrangers à l'intérieur de la bouteille sont également un défaut potentiellement dangereux. Des corps étrangers tels que des fragments de verre ou des débris peuvent pénétrer dans la bouteille pendant le processus de production ou de remplissage, ce qui peut causer des blessures aux consommateurs ou endommager l'équipement de production.

Afin de détecter efficacement les défauts des bouteilles, les fabricants peuvent utiliser des systèmes de détection basés sur l'IA et la vision par ordinateur pour analyser les images des bouteilles en temps réel. Ces systèmes peuvent être utilisés pour détecter rapidement les défauts et alerter les opérateurs de production afin qu'ils puissent prendre des mesures correctives immédiates. Cela permet aux fabricants de garantir la qualité de leurs produits, de réduire les pertes et d'améliorer la satisfaction des clients.

Le couvercle du contenant est une pièce importante du processus de conditionnement et peut comporter des imperfections telles que des égratignures de surface, des déformations, des taches, des écarts d'impression et d'autres faiblesses qui surviennent pendant la production (par exemple, marquage et injection de plastique). Par conséquent, l'inspection des couvercles défectueux est d'une importance critique pour le contrôle de la qualité dans le processus d'emballage. Pour détecter ce type de défaut, il est possible d'utiliser des procédures d'inspection basées sur la vision par ordinateur, qui permettent de remplacer l'inspection humaine. Une solution proposée dans une étude récente [31] offre une détection rapide des imperfections qui peuvent survenir sur la surface des couvercles de contenants. Ce système extrait la zone d'intérêt (ROI) sous forme de région circulaire à partir de la surface d'un couvercle, puis utilise l'histogramme de projection de région circulaire (CRPH) comme caractéristiques pour la correspondance et l'identification des défauts.

Des systèmes d'inspection visuelle automatisés qui inspectent la fermeture du bouchon et le niveau de remplissage dans l'industrie des bouteilles ont été proposés dans [32]. Ces systèmes ne peuvent inspecter que le niveau du contenu et la fermeture du bouchon.

Le travail présenté dans [33] c'est un système qui fait la détection des bouteilles il classe les types de défauts de bouteille en tant que défauts de base, défauts de col, défauts de surface de scellage, défauts de paroi latérale et défauts de finition. Le défaut de base comprend la contamination, les portes fissurées, les plis bouchons, les trous dans la porte ou les pieds, les portes excentrées et les pelures, tandis que les défauts de col sont classés comme sous-soufflage, plis, perlé, contamination, conteneur de couleur incorrecte, trous, contenants mal formés, cols étouffés et cols tordus. Le défaut de surface de scellage comprend les surfaces de scellage endommagées, les coupes, les creux, les finitions ovales, les tirs courts, les finitions soufflées, la contamination, la mauvaise couleur, le flash de surface de scellage et la surface roulée. De plus, les défauts de sous-soufflage, plis, perlé, contamination, conteneur de couleur incorrecte, trous et contenants mal formés sont les sous-catégories des défauts de paroi latérale. Enfin, les dommages importants, les contrôles dimensionnels et la contamination sont classés sous les défauts de finition.

### **I.7 Conclusion**

En conclusion, la détection des défauts dans les industries est une tâche cruciale pour garantir la qualité et la sécurité des produits fabriqués. Les systèmes d'inspection visuelle ont connu une évolution rapide ces dernières années, passant des méthodes traditionnelles à des techniques plus avancées en profitant de nouvelles méthodes de l'apprentissage profond. L'utilisation de l'IA et de l'apprentissage profond dans les systèmes d'inspection visuelle a permis de résoudre des problèmes complexes de détection de défauts dans les produits et de passer à un autre niveau d'inspection.

L'architecture des systèmes d'inspection visuelle modernes se compose de plusieurs éléments, tels que des caméras, des sources d'éclairages, des logiciels de traitement d'images et des algorithmes de détection de défauts. Ces systèmes peuvent être configurés pour s'adapter à différentes applications industrielles telles que l'assemblage, la reconnaissance optique de caractères (OCR) et la détection de défauts de surface.

Les systèmes d'inspection visuelle modernes ont permis aux industries de détecter les défauts avec une grande précision et une efficacité accrue. L'utilisation de l'IA et d'apprentissage profond offre de nombreuses opportunités pour améliorer encore ces systèmes et pour résoudre des problèmes de détection de défauts de plus en plus complexes.

Plusieurs entreprises et start-up contribuent dans le développement de ce secteur en proposant de nouvelles solutions et systèmes qui doivent répondre aux exigences d'un point de vue qualité et modernisation des procédés industrielles à l'ère de l'industrie 4.

## **CHAPITRE II**

### **Deep Learning**

## II.1 Introduction

L'apprentissage profond (Deep Learning, DL) est une sous-discipline de l'intelligence artificielle qui utilise des réseaux de neurones artificiels pour apprendre à partir de données et accomplir des tâches complexes. Il a montré des résultats remarquables dans divers domaines, notamment dans le traitement d'images. Dans ce chapitre, nous donnerons un aperçu du Deep Learning et de ses applications pour le traitement d'images. Nous commencerons par présenter brièvement l'histoire et la définition du Deep Learning, ainsi que sa position et son rôle dans le domaine de l'IA. Ensuite, nous présenterons quelques exemples de l'application du Deep Learning pour résoudre des problèmes de traitement d'images dans différents domaines. Ensuite, nous introduirons les différents types de méthodes de Deep Learning qui peuvent être utilisées en fonction de la disponibilité et de la qualité des données, comme les différents types d'apprentissage. Après cela, nous expliquerons les concepts de base et les composants des réseaux de neurones artificiels, qui sont le cœur des modèles de Deep Learning. Nous nous concentrerons également sur les réseaux de neurones convolutionnels (CNN), qui sont un type spécial de réseaux de neurones conçus pour le traitement d'images. De plus, nous présenterons certains des modèles de classification les plus populaires et avancés basés sur les CNN, tels que SqueezeNet, ResNet, Inception\_v3, etc. Enfin, nous discuterons de l'évaluation de l'efficacité et des performances l'exactitude des modèles de Deep Learning en utilisant différentes mesures et techniques.

## II.2 Introduction au Deep Learning

### II.2.1 Historique

Alan Turing a proposé l'idée de l'apprentissage machine dans les années 1950 [42-34], et les premiers réseaux de neurones simples ont été créés par Alexey Ivakhnenko au milieu des années 1960 [43-35]. En 1989, Yann LeCun [44-36] a montré comment utiliser la rétropropagation pour la reconnaissance des codes manuscrits.

L'idée des Réseaux de neurones convolutifs a été proposée par Kunihiko Fukushima en 1979 lorsqu'il a conçu le Neocognitron [37]. C'est un réseau de neurones artificiels avec une structure hiérarchique et en couches [37].

La recherche sur les réseaux de neurones a connu un revers pendant l'hiver de l'IA" de 1987 à 1993, lorsque le financement et l'intérêt pour la recherche en IA ont diminué.

En 1999, le deep learning a fait un bond significatif lorsque les ordinateurs sont devenus plus rapides dans le traitement des données et que les unités de traitement graphique (GPU) ont permis un traitement d'image plus rapide.

En 2009, Fei-Fei Li a lancé ImageNet [38], une immense base de données de plus de 14 millions d'images avec des étiquettes, ce qui a inspiré de nombreux chercheurs à explorer les possibilités du deep learning et des big data dans divers domaines de l'IA.

Depuis le début des années 2010, deep learning a réalisé des percées et de nouvelles architectures, méthodes et algorithmes ont été proposés pour contribuer dans la résolution de différentes problématiques dans tous les domaines.



L'évolution des méthodes de deep learning a un impact significatif sur l'avenir des entreprises. Ses applications se sont déjà étendues à la détection de fausses informations, au développement de modèles prédictifs pour la résolution des problèmes dans différents secteurs (santé, énergie, économie, ...etc.), et à la génération automatique d'images et des textes.

### II.2.2 Position du Deep Learning dans l'IA

L'intelligence artificielle (IA) consiste à doter une machine (un ordinateur) par des capacités qui imitent le cerveau humain et ses capacités de réflexion. L'IA se concentre sur trois aspects majeurs : l'apprentissage, le raisonnement et l'autocorrection, afin d'obtenir une efficacité maximale.

Le Machine Learning (ML) est une application ou une sous-discipline de l'IA qui permet aux systèmes informatiques d'apprendre et de s'améliorer automatiquement, en se basant sur leurs propres expériences. Cela se fait sans la nécessité d'une programmation explicite ou d'une intervention humaine. L'objectif principal du ML est de développer des systèmes capables d'apprendre et de s'améliorer de manière autonome, en utilisant l'expérience pour construire des connaissances et s'adapter à de nouvelles situations.

Le Deep Learning est une sous-discipline du Machine Learning. Les algorithmes de DL utilisent des réseaux de neurones artificiels pour identifier et classifier les motifs de données. Un modèle de DL est composé de plusieurs couches de neurones interconnectés, le terme "profond" faisant référence à la profondeur de ces couches [39]. Chaque couche reçoit et traite les informations de la couche précédente avant de les transmettre à la suivante. Contrairement au Machine Learning traditionnel, qui nécessite l'extraction manuelle des caractéristiques des données, les modèles de DL peuvent apprendre et extraire automatiquement les caractéristiques pertinentes à partir de vastes ensembles de données. Cela permet aux modèles de Deep Learning de faire des prédictions hautement précises sans intervention humaine.

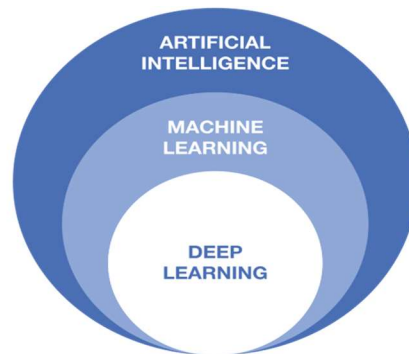


Figure II.1 : Position de DL dans le domaine d'IA [40]

### II.2.3 Caractéristiques attractives du DL

Les approches de Deep Learning ont démontré des avancées remarquables en termes de performance dans diverses applications à l'ère actuelle de l'Industrie 4.0, notamment l'automatisation pilotée par la technologie, les systèmes intelligents et autonomes, l'intelligence économique, etc. [39]. Ces approches constituent une solution efficace pour décoder des architectures complexes dans des données de grande dimension. L'utilisation massive de cette approche peut s'expliquer par les raisons suivantes :

- 1) Généralisation : La même approche de Deep Learning peut être appliquée à différentes applications et types de données, ce qui la rend utile pour le transfert d'apprentissage. Cela est particulièrement utile dans les cas où il y a peu de données disponibles pour entraîner un modèle.
- 2) Robustesse et extraction de caractéristiques : Les algorithmes de Deep Learning apprennent de manière progressive des caractéristiques de haut niveau à partir des données. Les approches de Deep Learning robustes ne nécessitent pas la conception préalable de caractéristiques, éliminant ainsi le besoin d'expertise dans l'extraction de caractéristiques. Cela diffère des techniques traditionnelles d'apprentissage automatique, qui reposent souvent sur un expert du domaine pour identifier les caractéristiques pertinentes et réduire la complexité des données afin que les algorithmes d'apprentissage puissent fonctionner [39].
- 3) Grands ensembles de données : Le Deep Learning surpasse les autres techniques lorsque l'ensemble de données est vaste [39].
- 4) Capacité à exploiter le transfert d'apprentissage et les modèles pré-entraînés : Les modèles pré-entraînés sont des modèles qui ont déjà été entraînés sur un grand ensemble de données pour une tâche spécifique, telle que la détection d'objets ou la reconnaissance vocale. Le transfert d'apprentissage consiste à prendre un modèle pré-entraîné et à l'ajuster finement pour une tâche spécifique [39].

### II.2.4 Deep Learning dans la détection de défauts de bouteilles d'eau

Dans le cadre de ce projet, nous nous intéressons à l'inspection visuelle de la qualité des bouteilles d'eau. Pour cela, nous limitons notre discussion de l'intérêt que peut présenter le DL dans le développement de ce type de système.

La détection des défauts des bouteilles d'eau est un processus important de contrôle qualité qui peut être amélioré grâce aux techniques de Deep Learning. Le Deep Learning offre une approche autonome qui réduit les erreurs humaines et la fatigue. La robustesse des modèles de Deep Learning, combinée à leurs capacités d'extraction de caractéristiques, permet la classification des défauts dans les bouteilles d'eau avec une grande précision [39]. Des techniques telles que la détection d'anomalies et le transfert d'apprentissage peuvent être utilisées pour entraîner des modèles de Deep Learning avec moins d'images d'entraînement et en moins de temps [41].

## II.2.5 Méthodes d'apprentissage en DL

Il existe plusieurs méthodes qu'on peut utiliser dans l'apprentissage des modèles. On peut citer notamment l'apprentissage supervisé, non supervisé, semi-supervisé, par renforcement et le transfert d'apprentissage.

### a. Apprentissage supervisé

L'apprentissage supervisé consiste à entraîner un réseau à apprendre une tâche ou une fonction spécifique en utilisant à la fois des données d'entrées et leurs sorties. Les problèmes les plus courants couverts par l'apprentissage supervisé sont les tâches de régression et de classification. Les tâches de régression prédisent des valeurs continues, telles que la taille, tandis que les tâches de classification catégorisent les données en classes discrètes, telles que la classification des produits en bons ou défectueux. Le défi de l'apprentissage supervisé réside dans l'obtention de données étiquetées, ce qui peut être long et coûteux.

### b. Apprentissage non supervisé

Contrairement à l'apprentissage supervisé, les données ne sont pas associées à des sorties désirées. Ici, le modèle essaie de découvrir des motifs et des structures dans les données d'entrées. Par conséquent, l'apprentissage non supervisé peut être utilisé pour des tâches de regroupement telles que l'algorithme K-means, la détection d'anomalies comme la détection d'anomalies avec les réseaux antagonistes génératifs (AnoGan) [38] et les algorithmes d'auto-encodeur [39], ainsi que pour la réduction de la dimensionalité telle que l'analyse en composantes principales (PCA) [39]. Cette approche peut être utile lorsque l'accès aux données étiquetées (sorties) est impossible ou pour l'analyse exploratoire des données.

### c. Apprentissage semi-supervisé

Cette approche est un mélange d'apprentissage supervisé et non supervisé, certaines données possèdent leurs étiquettes correspondantes, mais pas toutes. Parfois, l'étiquetage des échantillons de données peut être chronophage, il peut donc être plus pratique de ne marquer que certaines données [42], puis d'utiliser un algorithme d'apprentissage semi-supervisé tel que self-training.

### d. Apprentissage par renforcement

L'apprentissage par renforcement, ou apprentissage par renforcement profond (DRL), est une interaction dynamique entre un agent et son environnement. L'agent effectue des actions et reçoit des récompenses ou des pénalités de l'environnement. Les données sont séquentielles, donc le processus d'apprentissage se répète jusqu'à ce que l'agent trouve la politique optimale qui maximise la récompense cumulative. Cette approche est généralement utile pour les problèmes impliquant la prise de décisions séquentielles et l'interaction avec l'environnement. Un exemple d'algorithme DRL est l'apprentissage par imitation, qui a été utilisé pour construire le modèle AlphaGo Zero par David Silver et ses collègues chez Google DeepMind en 2016 [43]. Il s'agit d'un programme informatique qui a battu le champion du monde de Go, un jeu de société complexe.

### e. Transfert d'apprentissage

Le transfert d'apprentissage est une technique d'apprentissage automatique où un modèle formé pour une tâche est réutilisé pour une tâche connexe [44]. Par exemple, un modèle capable de reconnaître les chats peut être utilisé pour accélérer l'apprentissage de la reconnaissance des chiens. Le transfert d'apprentissage est utile dans le domaine du Deep Learning où l'entraînement des modèles nécessite beaucoup de données et de puissance de calcul. En utilisant des modèles pré-entraînés, on peut tirer parti des connaissances acquises pour résoudre un problème complexe et volumineux et les appliquer à un nouveau problème qui dispose de moins de données. Le transfert d'apprentissage fonctionne mieux lorsque les tâches sont similaires et que les caractéristiques sont générales, c'est-à-dire adaptées aux deux tâches. Il existe différentes approches du transfert d'apprentissage, telles que :

- Fine tuning : le modèle pré-entraîné est utilisé avec ses poids comme des poids initiaux pour le nouveau modèle et continuons l'entraînement sur les nouvelles données [44].

- L'adaptation de domaine : le modèle sera ajusté à un nouvel ensemble de données d'exemples connexes, tels que des images de piétons dans différentes conditions d'éclairage. Cela est nécessaire lorsque l'on est confronté à un problème de changement de domaine, où la distribution des données change entre les ensembles d'entraînement et de test [44].

- L'extraction de caractéristiques : un modèle pré-entraîné peut être exploité comme un extracteur de caractéristiques. Dans ce cas, on refait l'entraînement uniquement de nouvelles couches de classification. Cela peut accélérer l'entraînement lorsque les tâches sont différentes ou si on ne dispose que de données limitées pour éviter le sur ajustement [44].

Le transfert d'apprentissage peut être appliqué à la fois à des modèles d'apprentissage supervisé et non supervisé [44]. Dans des problèmes de reconnaissance tels que l'inspection visuelle des produits, il est courant d'utiliser un modèle pré-entraîné pour une tâche de classification sur un grand ensemble de données tel que ImageNet, qui compte plus de 14 millions d'images. Certains modèles de pointe ont pu atteindre des taux de précision élevés sur cet ensemble de données comme le modèle VGG d'Oxford, le modèle Inception de Google et le modèle ResNet de Microsoft. Un autre choix courant est d'utiliser des modèles ayant atteint des performances de pointe dans diverses tâches de détection, telles que l'ensemble de données COCO. Trois exemples de tels modèles pré-entraînés sont Faster R-CNN développé par une équipe de chercheurs de Microsoft Research, SSD (Single Shot MultiBox Detector) et YOLO (You Only Look Once) [45].

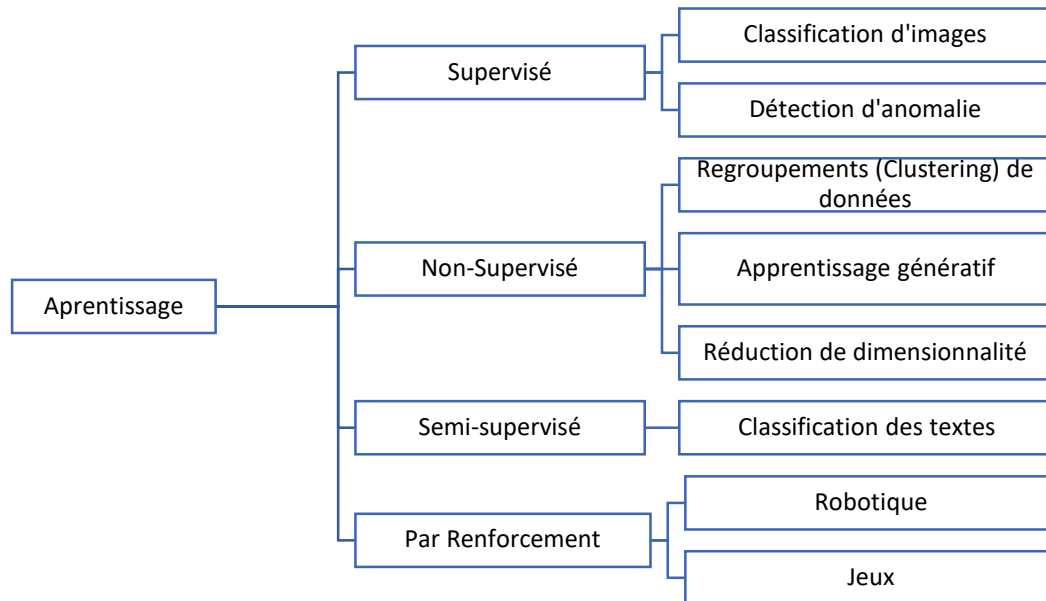


Figure II.2 : Les techniques d'apprentissage et exemples d'application

## II.3 Réseaux de neurones artificiels : principe et notions fondamentales

### II.3.1 Réseaux de neurones : Composition et Architecture

Un réseau de neurones artificiels (RNA) [39] est un modèle computationnel qui imite le fonctionnement du traitement de l'information par le cerveau. Il est composé de nombreuses unités (neurones) qui sont connectées par des poids et des biais (Figure II.3). Selon la fonction d'activation, le modèle peut être utilisé soit pour des tâches de classification, soit pour des tâches de régression ou aussi d'autres fonctions. En classification, le réseau Perceptron multicouches représente le modèle de base (Figure II.4), Il est constitué de :

- Une couche d'entrée a une unité pour chaque caractéristique d'entrée. Ces unités transmettent leurs valeurs à la couche suivante sans effectuer de calcul.
- Une ou plusieurs couches cachées. Chaque couche comporte une ou plusieurs unités (neurones) qui effectuent un calcul. Le calcul consiste à additionner les entrées pondérées de la couche précédente et le terme de biais, puis à appliquer une fonction d'activation. La fonction d'activation (f) peut être linéaire, sigmoïde, tangente hyperbolique (tanh), relu, ...etc.
- Le calcul de la sortie d'un neurone se fait selon l'équation (1) :

$$y = f(\sum_i(w_i * inputs) + b_i) \quad (1)$$

- La couche de sortie a un neurone pour chaque classe. Ces neurones produisent la sortie finale du réseau en effectuant le même calcul que les unités cachées.
- Les poids et les biais du réseau sont appris en minimisant une fonction de perte (Loss) qui mesure la différence entre la sortie du réseau et la sortie désirée. La fonction de perte peut être l'erreur quadratique moyenne, l'entropie croisée, la perte charnière, etc. Le processus

d'apprentissage peut être réalisé à travers la rétropropagation en utilisant la descente de gradient, la descente de gradient stochastique, etc.

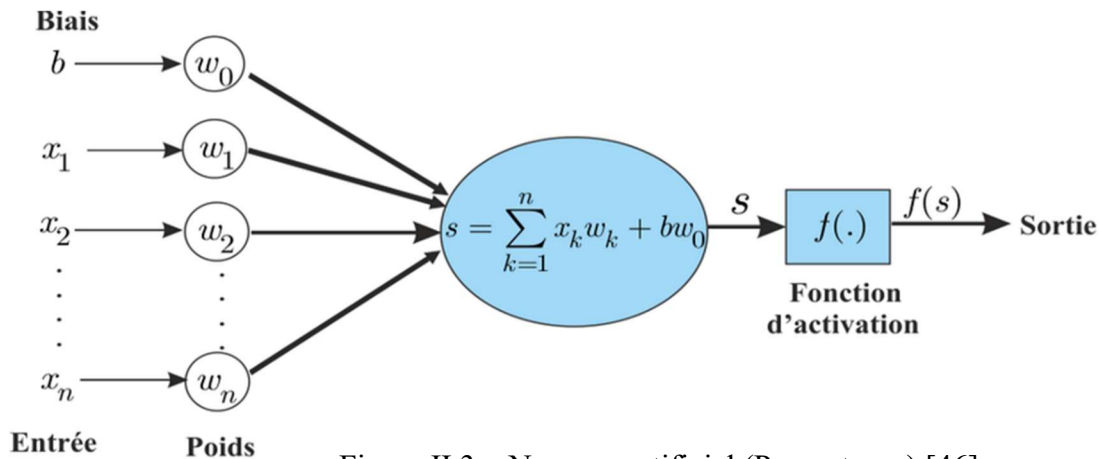


Figure II.3 : Neurone artificiel (Perceptrone) [46]

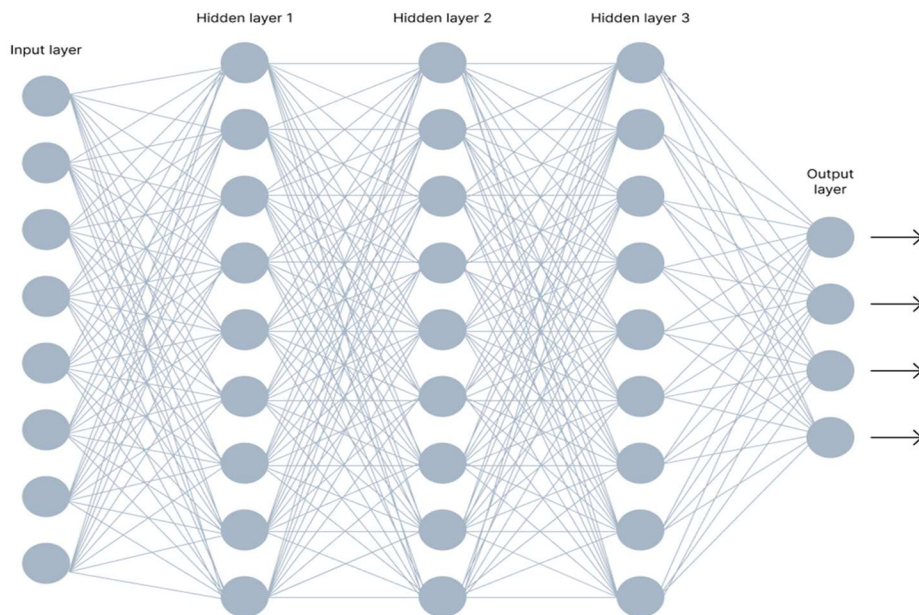


Figure II.4 : Perceptrone multicouche [47]

### II.3.2 Processus d'apprentissage

L'apprentissage d'un réseau de neurones artificiels est un processus où les données d'apprentissage sont fournies au réseau afin de trouver les meilleurs paramètres (les poids et les biais) qui minimisent la fonction de perte, qui mesure l'erreur pour une prédiction donnée. Ce processus est réalisé en utilisant la rétropropagation du gradient, qui modifie les paramètres en fonction du gradient de la fonction de perte [48].

**a. Fonction perte (Loss)**

La fonction de perte, également appelée fonction de coût ou fonction d'erreur [49], mesure l'erreur entre la sortie réelle (désirée) et la sortie prédite, et sera utilisée ultérieurement pour mettre à jour les paramètres. Les fonctions de perte les plus courantes et utilisées sont la fonction d'entropie croisée et la fonction d'erreur quadratique moyenne.

**Fonction cross-entropie** : L'une des fonctions de perte les plus courantes pour les tâches de classification correspond à l'entropie croisée entre la sortie réelle et les prédictions. Elle pénalise davantage le modèle lorsqu'il attribue une probabilité faible à la classe correcte ou une probabilité élevée à la mauvaise classe. Elle encourage le modèle à faire des prédictions confiantes et précises [49].

$$J(w) = - \sum_i \log (P(y = y^{(i)} | x = x^{(i)}; W)) \tag{2}$$

Avec  $(P(y = y^{(i)} | x = x^{(i)}; W))$  est probabilité qu'un échantillon soit correctement classé.

L'erreur quadratique moyenne : est utilisée pour les problèmes de régression, où le modèle tente de prédire une valeur continue. Elle mesure l'écart entre la prédiction du modèle et la vraie valeur [49].

$$J(w) = \sum_i \log ||y^{(i)} - f(x^{(i)}; W)||^2 \tag{3}$$

**b. Rétropropagation**

La fonction de perte est optimisée dans le processus de rétropropagation pour mettre à jour les paramètres. La rétropropagation applique la descente de gradient [54] pour trouver un minimum local. Elle calcule le gradient de la fonction de perte  $J(W)$  par rapport à chaque poids en utilisant la règle de la chaîne afin de mettre à jour les paramètres [52]. Elle commence par la dernière couche et remonte jusqu'à la première couche. De cette manière, elle n'effectue pas les mêmes calculs pour les termes de la règle de la chaîne. Ce processus est expliqué dans la figure II.5.



Figure II.5 : Perceptrone multicouche avec une couche cachée

La rétropropagation de ce perceptron multicouche est calculée comme suit :

$$\begin{aligned} \frac{\partial J(W)}{\partial w_2} &= \frac{\partial J(W)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_2} \\ \frac{\partial J(W)}{\partial w_1} &= \frac{\partial J(W)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_1} = \frac{\partial J(W)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_1} \frac{\partial z_1}{\partial w_1} \end{aligned} \tag{4}$$

**c. Optimisation**

L'optimisation est le processus de recherche des paramètres optimaux  $W$  qui minimisent la fonction de perte  $J(W)$ , qui mesure l'erreur des prédictions du réseau sur les données [50].

L'optimisation dans les réseaux neuronaux est difficile car la fonction de perte est souvent non convexe, ce qui signifie qu'elle possède de nombreux minima locaux [50]. Pour trouver le minimum global, différentes méthodes d'optimisation mettent à jour les paramètres en se basant sur le gradient de la fonction de perte, qui indique la direction de la plus forte descente. Certaines de ces méthodes sont la descente de gradient, la descente de gradient stochastique, Momentum, les taux d'apprentissage adaptatifs (ADAM). Ces méthodes visent à améliorer la vitesse et la précision de l'optimisation dans les réseaux neuronaux.

- **Descente de gradient**

La descente de gradient est l'une des approches les plus courantes et basiques de l'optimisation des réseaux neuronaux. C'est un algorithme d'optimisation du premier ordre utilisé pour trouver les minima locaux d'une fonction objective [39]. La descente de gradient est insuffisante pour les problèmes d'apprentissage automatique avec un grand nombre d'échantillons d'entraînement. Mettre à jour les paramètres  $W$  en se basant sur le gradient de tous les échantillons augmenterait considérablement la complexité computationnelle de l'optimisation [50].

La loi de mise à jour des paramètres  $W$  est la suivante :

$$W^{t+1} \leftarrow W^t - \eta \frac{\partial J(W^t)}{\partial W^t} \quad (5)$$

- **Descente de gradient stochastique**

Étant donné que le temps d'entraînement prolongé est le principal inconvénient de l'approche traditionnelle de descente de gradient, la descente de gradient stochastique (DGS) [49] est une simplification radicale. Au lieu de calculer exactement le terme  $\frac{\partial J(W^t)}{\partial W^t}$ , elle calcule le gradient de la fonction de perte à partir d'un sous-ensemble de données (un mini-lot) [49]. La formule de mise à jour est la suivante :

$$W^{t+1} = W^t - \eta \frac{\partial J(W^t:(x,y))}{\partial W^t} \quad (6)$$

Où  $W$  est le vecteur de paramètres,  $\eta$  est le taux d'apprentissage,  $J$  est la fonction de perte et  $(x,y)$  est le point de données.

- **Momentum**

Le Momentum est une technique utilisée, avec des techniques d'optimisation telles que SGD et ADAM, pour accélérer le processus d'apprentissage et empêcher le modèle de rester coincé dans des minima locaux. L'idée conceptuelle du Momentum est d'utiliser la moyenne mobile du gradient plutôt que d'utiliser uniquement la valeur actuelle du gradient. Cela se fait en introduisant un nouveau paramètre, le taux de Momentum, qui détermine dans quelle mesure la mise à jour précédente des poids est conservée dans le terme de la vitesse ( $v^{t+1}$ ) et ajoutée à la mise à jour actuelle [50]. Cela conduit à une convergence plus rapide et à une meilleure généralisation. La mise à jour de SGD avec Momentum s'exprime comme suit :

$$v(t+1) = \gamma * v(t) + (1 - \gamma) * \frac{\partial J(W^{t+1}:(x,y))}{\partial W^t} \quad (7)$$



$$W^{t+1} = W^t - \eta * v^{t+1} \quad (8)$$

Où :  $\gamma$  est le taux de Momentum et  $v$  est le terme de vélocité.

En général,  $\gamma$  doit être dans la plage  $[0,1]$ . Des valeurs de momentum plus élevées dépasseront leur minimum, voire provoqueront une divergence. D'autre part, des valeurs de momentum faibles peuvent faire en sorte que le réseau reste coincé dans des minima locaux ou ralentissent la convergence. Une bonne valeur de départ est généralement  $\gamma = 0,9$  [50].

- **Taux d'apprentissage adaptatifs**

Le taux d'apprentissage est un hyperparamètre important pour l'entraînement. Il s'agit de la taille de pas considérée pendant l'entraînement. Autrement dit, il détermine dans quelle mesure les paramètres sont mis à jour pendant le processus d'optimisation, ce qui affecte la vitesse et la précision du processus d'entraînement. Cependant, le choix de la valeur du taux d'apprentissage est délicat [51]. Par exemple, Lorsqu'on choisit une valeur élevée pour  $\eta$ , le réseau risque de diverger. D'autre part, si on utilise une valeur plus petite pour  $\eta$ , il faudra plus de temps au réseau pour converger, et il risque de rester coincé dans des minima locaux. Une solution courante consiste à utiliser des techniques de taux d'apprentissage adaptatif [51].

L'apprentissage adaptatif est une technique utilisée pour mettre à jour le taux d'apprentissage pendant le processus d'entraînement, en se basant sur l'historique du gradient. Ainsi, il peut rendre le processus d'entraînement plus rapide, plus flexible et plus robuste. Il existe différents types de taux d'apprentissage adaptatif tels que AdaGrad, RMSProp et ADAM.

- **ADAM**

Adam [52] est un algorithme d'optimisation qui combine les avantages d'AdaGrad et de RMSProp. Il maintient des estimations des premiers et seconds moments des gradients. Le premier moment est une mesure de la direction globale des gradients, et le second moment est la moyenne des gradients au carré.

Le taux d'apprentissage pour chaque paramètre est calculé comme une moyenne pondérée du premier moment et du second moment. Cela signifie que le taux d'apprentissage d'un paramètre sera plus élevé si le premier moment est grand et le second moment est petit. Cela est dû au fait qu'un premier moment élevé indique que les gradients sont orientés dans une direction cohérente, et qu'un second moment faible indique que les gradients ne sont pas très bruités [52].

La règle de mise à jour est la suivante [48] :

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) J(w)$$

$$m_t = \beta_2 m_{t-1} + (1 - \beta_2) J(w)^2$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_1^t}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_2^t}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{m}_t + \epsilon}} (\gamma v_t + (1 - \gamma) \hat{v}_t) \quad (9)$$

Où :

$v_t$ : estimation du premier moment (ou moyenne mobile) des gradients à l'étape t

$m_t$ : estimation du second moment (ou moyenne mobile des gradients au carré) à l'étape t

$\beta_1$ : taux de décroissance exponentielle pour l'estimation du premier moment

$\beta_2$ : taux de décroissance exponentielle pour l'estimation du second moment

$\epsilon$ : petite constante ajoutée au dénominateur pour la stabilité numérique

ADAM présente de nombreux avantages à savoir :

- Utilisation efficace de la mémoire : Contrairement à AdaGrad et RMSProp, il n'est pas nécessaire de stocker tous les gradients précédents. Adam n'a besoin de stocker que quelques statistiques pour chaque paramètre, ce qui le rend plus efficace en termes de mémoire [52].
- Robuste aux gradients bruités : Adam fonctionne bien même en présence de gradients bruités, ce qui peut être courant dans les applications d'apprentissage profond. Il est plus robuste aux gradients bruités que SGD avec Momentum.
- Facilité d'utilisation : Adam est facile à utiliser et ses hyperparamètres sont généralement faciles à régler sans nécessiter beaucoup de réglages, ce qui en fait un choix populaire pour de nombreuses applications d'apprentissage profond.

### II.3.3 Qualité d'apprentissage et méthodes de régularisation

#### a. Surapprentissage : Overfitting

Overfitting se produit lorsqu'un modèle ou un réseau parvient à bien performer sur l'ensemble d'entraînement et commence à le mémoriser plutôt qu'à apprendre les schémas sous-jacents. Cela affecte la capacité du réseau à généraliser et à bien performer sur des données inconnues (ensemble de test). Il existe également le problème de sous-apprentissage, dans lequel le réseau ne parvient à bien performer ni sur l'ensemble d'entraînement ni sur l'ensemble de test. Cela peut se produire en raison de plusieurs raisons telles que la similarité élevée, l'insuffisance de données de l'ensemble de données, la complexité du modèle et un entraînement trop long.

Pour résoudre le problème d'Overfitting, de nombreuses approches de régularisation ont été introduites, telles que la décroissance des poids en utilisant différentes méthodes de régularisation (L1, L2), Normalisation par lots (Batch normalisation), l'arrêt précoce et le dropout. De plus, d'autres techniques dont l'objectif principal n'est pas de prévenir le surapprentissage, mais qui peuvent aider à prévenir ou réduire le risque de surapprentissage, telles que l'augmentation des données. Notez qu'une combinaison de ces techniques peut être utilisée conjointement pour obtenir de meilleurs résultats.

## b. Méthodes de Régularisation

- **Décroissance des poids**

La décroissance des poids est une technique de régularisation utilisée pour prévenir le surapprentissage en ajoutant un terme de pénalité à la fonction de perte qui dépend de la taille des poids [53]. Plus les poids attribués à chaque entrée ou connexion sont grands, plus les connexions deviennent complexes et fortes. Par conséquent, pour résoudre ce problème, la décroissance des poids augmente le terme de pénalité pour réduire la taille des poids [48]. Deux techniques de régularisation des décroissances des poids couramment utilisées sont la régularisation L1 et L2.

### 1. Régularisation L1

Aussi connue sous le nom de régression de Lasso. Le terme de pénalité dans la régularisation L1 est la somme de la valeur absolue des poids [53], ce qui conduit à une solution plus clairsemée. Cela signifie que certains des poids deviennent nuls, ce qui aide le modèle à se concentrer uniquement sur les poids ou caractéristiques importantes.

$$\text{L1 regularization term} = \lambda \sum_{i=1}^n |w_i|$$

$$\min_w J(W) = \frac{1}{n} \sum_{i=1}^n \frac{\partial J(Wt:(x,y))}{\partial w_t} + \lambda \sum_{i=1}^n |w_i| \quad (10)$$

Où  $\lambda$  est le paramètre de régularisation L1.

### 2. Régularisation L2

Cette méthode est également connue sous le nom de régression Ridge et est l'une des méthodes de décroissance des poids les plus courantes et largement utilisées. Comme pour la régularisation L1, elle ajoute un terme de pénalité à la fonction objective, mais dans ce cas, la pénalité est la racine carrée de la somme des carrés de chaque poids. Cette pénalité décourage les valeurs importantes ou extrêmes pour les paramètres individuels, mais contrairement à la régularisation L1, elle ne fixe aucun poids à zéro. Par conséquent, la régularisation L2 favorise une distribution plus équilibrée des valeurs des poids [53], réduisant ainsi l'amplitude des paramètres et le risque de surapprentissage.

$$\text{L2 regularization term} = \lambda_2 \sum_{i=1}^n |w_i|^2$$

$$\min_w J(W) = \frac{1}{n} \sum_{i=1}^n \frac{\partial J(Wt:(x,y))}{\partial w_t} + \lambda_2 \sum_{i=1}^n |w_i|^2 \quad (11)$$

Où  $\lambda_2$  représente le paramètre de régularisation L2.

- **Batch normalisation**

La normalisation par lots peut être considérée comme une forme de régularisation [54] qui peut aider le réseau à apprendre de manière plus efficace et réduire le risque de surapprentissage ; En particulier, elle permet d'éviter le problème de décalage covariant interne qui modifie la distribution des entrées de chaque couche et qui affecte le taux d'apprentissage du réseau. Cela est réalisé en veillant à ce que chaque couche ait la même échelle et distribution en transformant

linéairement les entrées pour obtenir une moyenne nulle et une variance unitaire [53,54]. Cette technique est particulièrement importante pour les réseaux de neurones récurrents profonds [54].

- **Early stopping**

Lors de l'entraînement de modèles volumineux et profonds, les erreurs d'entraînement et de validation diminuent généralement ensemble, mais après un certain nombre d'époques, il arrive parfois que la perte de validation commence à augmenter ou cesse de s'améliorer, même si la perte d'entraînement continue de diminuer. C'est un signe de surapprentissage. Pour éviter cela, on peut arrêter le processus d'apprentissage lorsque les performances de validation atteignent leur meilleure valeur ou commencent à se détériorer. Cela nécessite de surveiller les performances du modèle et de se souvenir de l'erreur de validation. L'arrêt précoce est facile à mettre en œuvre et peut être basé sur différents critères. Une meilleure illustration de ce concept est présentée dans la figure II.6.

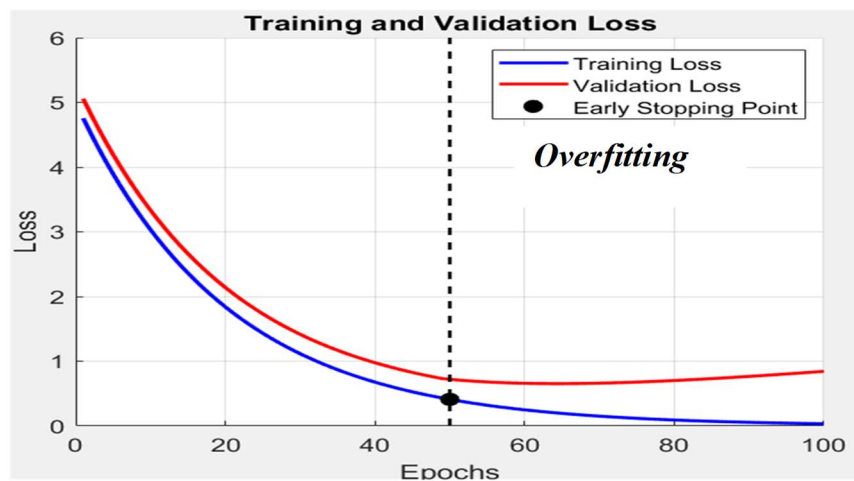


Figure II.6 : Early stopping

- **Dropout**

Le dropout [55] est l'une des méthodes les plus efficaces pour résoudre le problème de surapprentissage. Cela consiste à mettre un pourcentage  $p$  aléatoire de poids à zéro pendant le processus d'apprentissage. Cela réduit la complexité du modèle et l'empêche de se baser sur des caractéristiques ou des motifs spécifiques dans les données, l'obligeant ainsi à être plus robuste et généralisable [55]. Lors de l'inférence, tous les poids sont utilisés mais ils sont multipliés par le pourcentage  $p$  du dropout. La figure II.7 illustre ce concept.

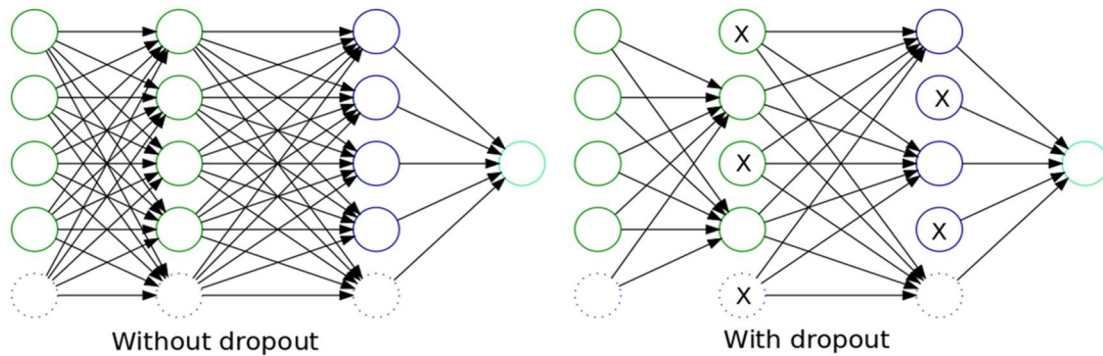


Figure II.7 : Principe de Dropout [56]

- **Augmentation des données**

L'augmentation des données est une autre technique qui peut prévenir et réduire le risque de surapprentissage, bien que sa fonction principale ne soit pas la régularisation. Il s'agit d'une technique visant à augmenter la taille et la diversité des données d'entraînement en appliquant des transformations ou en générant de nouveaux échantillons. Cette méthode est particulièrement utile lorsque les données sont insuffisantes ou présentent une forte similitude. De nombreuses techniques d'augmentation des données existent pour la génération de nouvelles images par exemple, telles que le miroir, le flou, l'injection de bruit, la rotation et bien d'autres. Certaines de ces techniques sont illustrées dans la figure II.8.

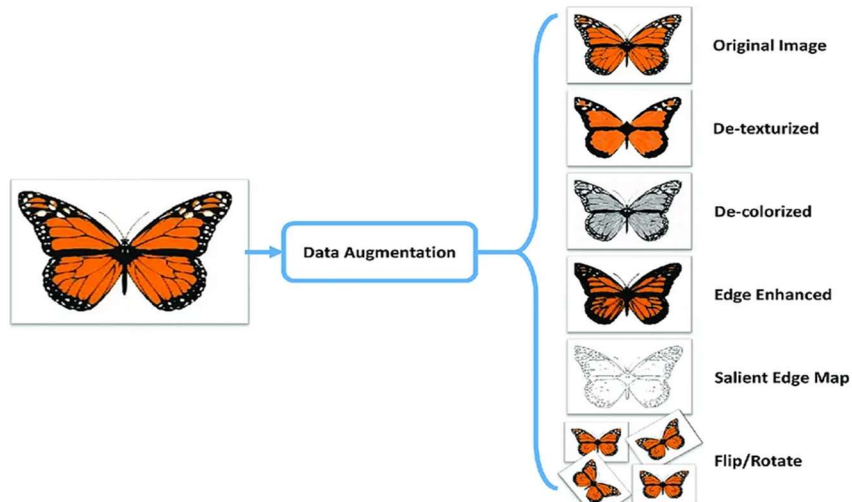


Figure II.8 : Quelques techniques d'augmentation de données [57]

L'évolution du Deep Learning a fait apparaître aussi un type de réseaux de neurones qui peut servir à l'augmentation des données, il s'agit du Réseau Adversarial Génératif (GAN) [58]. Il peut générer des données réalistes et diverses à partir de bruit aléatoire. Un GAN est composé de deux éléments : un générateur et un discriminateur. Le générateur produit des données qui ressemblent aux données réelles, tandis que le discriminateur tente de distinguer les données réelles des données fausses [58]. La structure du GAN est illustrée dans la figure II.9. Grâce à ce processus, le générateur apprend à produire des données qui correspondent à la distribution des données réelles, mais les GAN nécessitent généralement des ressources informatiques importantes.



### a. Couches de convolution

Les couches convolutives extraient des caractéristiques locales à partir des données d'entrée à l'aide de filtres, les couches de Pooling réduisent la dimensionnalité des cartes de caractéristiques, les données sont aplaties et passées aux couches entièrement connectées qui classifient les caractéristiques apprises. La combinaison de ces couches forme une structure hiérarchique qui permet une analyse et une compréhension efficaces des images.

### b. Pooling

Les couches de Pooling sont utilisées pour le sous-échantillonnage. Il est courant d'appliquer une couche de Pooling après une couche convolutive pour réduire la dimension spatiale de l'entrée, comme illustré dans la figure II.11. Cette opération est généralement réalisée en prenant la valeur maximale (max Pooling) ou la valeur moyenne (Average Pooling) dans chaque région. Les couches de Pooling servent à deux objectifs principaux dans les CNN. Premièrement, elles contribuent à rendre le réseau plus robuste aux petites translations ou déplacements dans les données d'entrée, car le sous-échantillonnage réduit la sensibilité aux positions précises des pixels. Deuxièmement, les couches de Pooling réduisent la complexité de calcul du réseau en réduisant le nombre de paramètres et la dimensionnalité spatiale des cartes de caractéristiques, ce qui peut aider à prévenir le surapprentissage et améliorer l'efficacité de calcul.

### c. Flatten (Aplatissement)

L'opération d'aplatissement est appliquée après les couches de convolution et de Pooling, avant les couches entièrement connectées. Elle transforme les cartes de caractéristiques multidimensionnelles en une représentation vectorielle qui peut être traitée par les couches entièrement connectées (figure II.12). En aplatissant les cartes de caractéristiques, le réseau peut considérer les caractéristiques apprises comme une représentation globale de l'entrée, permettant une meilleure intégration et compréhension des caractéristiques extraites. Les caractéristiques aplaties sont ensuite transmises aux couches entièrement connectées, qui effectuent la classification ou d'autres tâches de haut niveau basées sur les représentations apprises.

## II.5 Présentation des modèles réussit de classification d'image

Des modèles de reconnaissance d'images basés sur les CNN ont révolutionné le domaine de la vision par ordinateur, atteignant une précision et des performances sans précédent. On peut citer par exemple ResNet [60], Inception\_V3 [61], SqueezeNet [62] et Alexnet [39], etc. La plus haute précision atteinte jusqu'à présent sur l'ensemble de données ImageNet a été réalisée par Swin Transformer-V2 [63]. Il a atteint une précision top-1 de 91,3 % et une précision top-5 de 95,7 % sur l'ensemble de validation ImageNet. Il a établi de nouveaux records de performance sur trois autres tâches de vision représentatives, notamment la détection d'objets COCO, la segmentation sémantique ADE20K et la classification d'actions vidéo Kinetics-400 [63].

Ces réseaux entraînés sur de très grandes datasets, ce qui a rendu leur exploitation à travers l'apprentissage par transfert très fréquent. Nous décrivons ci-dessous quelques architectures dont l'étude représente une étape importante dans la réalisation de ce projet.

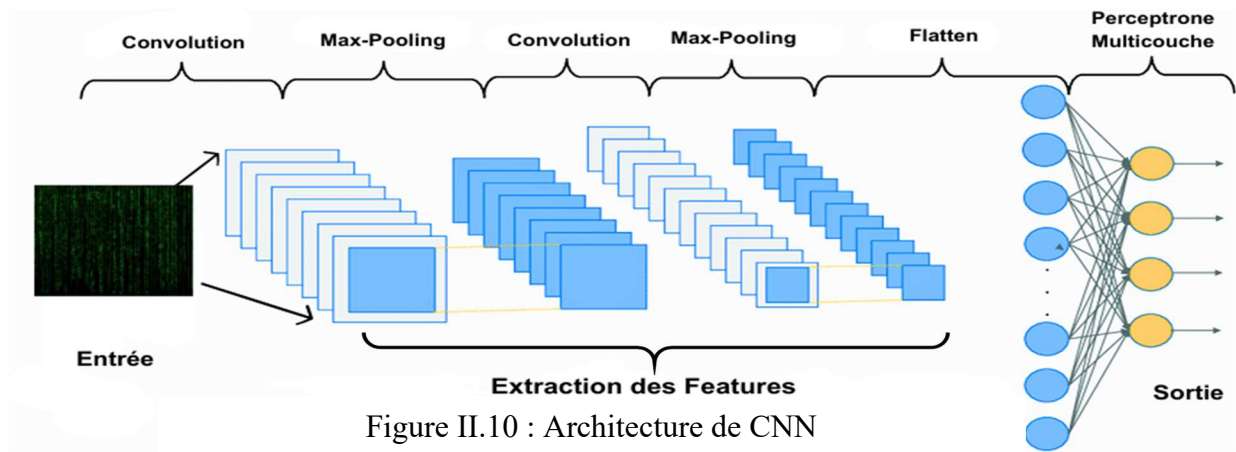


Figure II.10 : Architecture de CNN

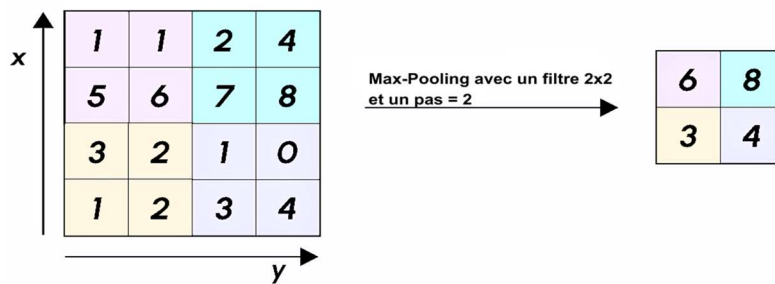


Figure II.11 : Max-Pooling avec 2x2 filtre et un pas de 2

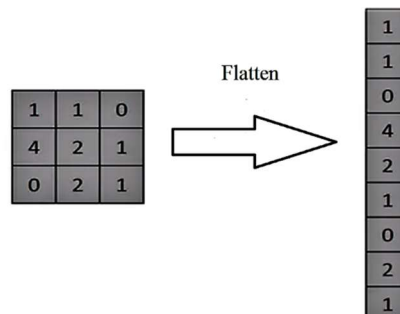


Figure II.12 : Opération de Flatten

### II.5.1 ResNet

L'augmentation du nombre de couches dans un réseau entraîne un problème courant dans l'apprentissage profond appelé le gradient qui s'estompe ou explose. Un gradient qui s'estompe devient proche de zéro, tandis qu'un gradient qui explose devient très grand. Ainsi, l'augmentation du nombre de couches augmentera le taux d'erreur d'entraînement et de test. Pour résoudre ce problème, Residual Network (ResNet) a introduit le concept de blocs résiduels [60]. ResNet est une architecture de modèle d'apprentissage profond principalement utilisée pour les tâches de classification d'images. Il a été introduit par des chercheurs de Microsoft en 2015 [60] et il a remporté l'ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2015, 2016 et 2017. ResNet-50 a démontré une précision top-1 d'environ 76-78 % et une précision top-5 d'environ 93-94 %.



La caractéristique importante de ResNet réside dans l'utilisation de connexions résiduelles ou de skip connections (Figure II.13). Ces connexions permettent au modèle d'apprendre des fonctions résiduelles, qui sont les différences entre l'entrée et la sortie souhaitée. L'avantage d'ajouter ce type de connexion de saut est que si une couche nuit aux performances de l'architecture, elle sera contournée par la régularisation. En propageant ces résidus à travers le réseau, ResNet surmonte le défi de l'apprentissage de réseaux extrêmement profonds, atteignant plus de 100 couches (ResNet152), où la précision du réseau atteint un plateau voire se dégrade avec l'augmentation de la profondeur. Cela permet de maintenir ou même d'améliorer les performances tout en réduisant le nombre de paramètres par rapport à d'autres architectures profondes, ce qui le rend plus efficace en termes de mémoire et plus facile à entraîner.

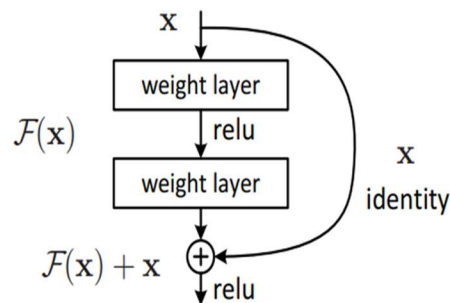


Figure II.13 : Connexion résiduelle [60]

L'architecture ResNet est généralement composée de blocs résiduels. Chaque bloc est constitué de plusieurs couches de convolution suivies d'une normalisation par lots et d'activations linéaires rectifiées (ReLU). La connexion directe relie l'entrée du bloc directement à la sortie, contournant les couches de convolution. Plusieurs versions de ResNet ont été développées telle que ResNet 18 et ResNet50.

- **ResNet18 [60]**

ResNet-18 est une variante de l'architecture ResNet avec 18 couches, elle est composée de :

- Quatre étapes avec des blocs résiduels.
- Des couches de convolution 3x3 et des connexions directes.
- Global Average Pooling suivi d'une couche entièrement connectée.

- **ResNet50 [60]**

ResNet-50 est une architecture plus profonde avec 50 couches (Figure II.14), comprenant des blocs de bottleneck et une réduction de la dimensionnalité. ResNet-50 offre un nombre de paramètres plus élevé et une meilleure capacité de représentation pour des tâches de classification plus complexes. Elle se caractérise par :

- Similaire à ResNet-18 mais avec plus de blocs résiduels.
- Le bloc de bottleneck (Figure II.15) est un block convolutionnel qui consiste de 3 filtres (1x1, 3x3, 1x1) pour réduire la complexité computationnelle tout en maintenant la capacité de représentation. La première convolution 1x1 est utilisée pour réduire la dimensionnalité des

cartes de caractéristiques d'entrée. Puis, la deuxième convolution 3x3 effectue l'extraction principale des caractéristiques. Enfin, la dernière convolution 1x1 est utilisée pour restaurer la dimensionnalité avant de passer la sortie au bloc suivant. Cette idée est basée sur le fait que certaines couches dans un réseau neuronal profond peuvent être coûteuses en termes de calcul.

- Réduction de la dimensionnalité avec des convolutions 1x1, réduisant les dimensions spatiales (largeur et hauteur) des cartes de caractéristiques tout en augmentant leur profondeur (nombre de canaux). Cette réduction aide à rendre le réseau plus efficace en réduisant la charge de calcul et les exigences de mémoire.
- Nombre de paramètres plus élevé et meilleure capacité de représentation.

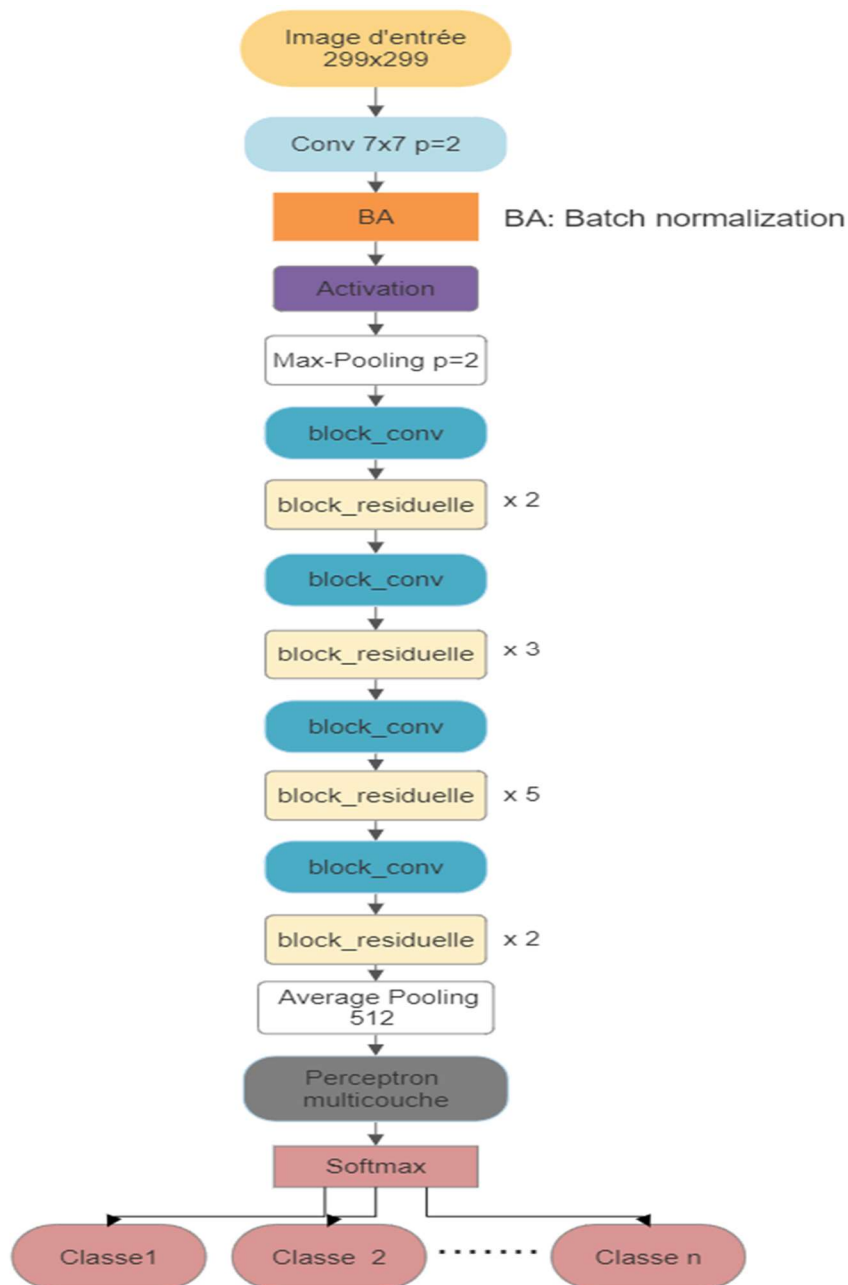


Figure II.14 : Architecture de ResNet50

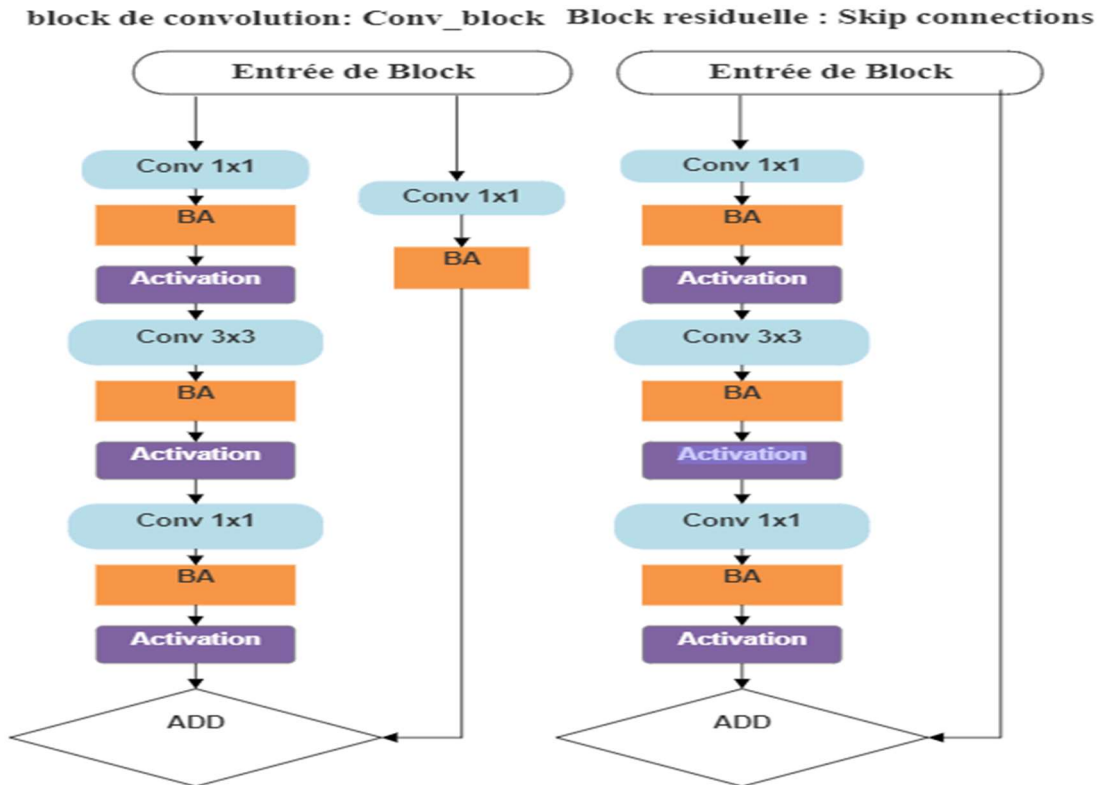


Figure II.15 : Block de convolution et résiduelle dans ResNet50

## II.5.2 Inception\_v3 [61]

Inception-v3, introduit par des chercheurs de Google en 2015 [61], s'appuie sur le succès de son prédécesseur, Inception-v1, et intègre plusieurs améliorations. L'idée principale de cette architecture est de capturer efficacement des caractéristiques multi-échelles en utilisant différentes tailles de filtres au sein de la même couche. Les précisions rapportées pour Inception-v3 sur ImageNet sont d'environ 77 à 79% en termes de précision Top-1 et d'environ 93 à 94% en termes de précision Top-5. Il a obtenu un taux d'erreur Top-5 de 3,46% lors du défi de reconnaissance visuelle à grande échelle ImageNet (ILSVRC) 2015, se classant deuxième parmi toutes les soumissions [61].

Caractéristiques d'Inception-v3 :

- **Modules Inception** : Les blocs de construction principaux d'Inception-v3 sont les modules Inception. Ces modules sont constitués de couches de convolution parallèles avec différentes tailles de filtres, ce qui permet au modèle de capturer simultanément des caractéristiques à plusieurs échelles [61]. En incorporant des convolutions 1x1, 3x3 et 5x5, ainsi que des opérations de Max-Pooling, le modèle acquiert la capacité de capturer efficacement à la fois des informations locales et globales.
- **Factorisation** : Pour réduire la complexité computationnelle, Inception-v3 intègre des techniques de factorisation. Au lieu d'appliquer directement une grande taille de filtre, telle qu'une convolution 5x5, il décompose l'opération en deux convolutions plus petites (par exemple, 1x5 et 5x1), qui sont moins coûteuses en termes de calcul. Cette stratégie de

factorisation permet de maintenir la capacité de représentation tout en réduisant le nombre de paramètres.

- Inception-v3 utilise des couches de convolution parallèles avec différentes tailles de filtres au sein de chaque module Inception. Les sorties de ces couches de convolution parallèles sont concaténées pour former un seul tenseur de sortie. En concaténant les cartes de caractéristiques provenant de différentes branches parallèles, Inception-v3 capture efficacement des caractéristiques multi-échelles et permet au modèle d'apprendre à la fois des informations locales et globales.
- Classificateurs auxiliaires : Inception-v3 utilise des classificateurs auxiliaires à des couches intermédiaires lors de l'entraînement. Ces classificateurs auxiliaires, composés de convolutions 1x1 et de couches entièrement connectées, sont ajoutés au réseau pour atténuer le problème du gradient qui s'éteint et fournir une régularisation supplémentaire [61]. Ils favorisent la propagation des gradients des couches supérieures vers les couches inférieures, facilitant un meilleur flux de gradients et améliorant l'efficacité de l'entraînement.
- Blocs de réduction : Inception-v3 inclut des blocs de réduction qui sont responsables de la réduction des dimensions spatiales des cartes de caractéristiques tout en augmentant leur profondeur. Ces blocs de réduction utilisent généralement une combinaison de max pooling 3x3 et de convolutions 1x1 avec des pas appropriés pour réduire la dimension des cartes de caractéristiques.

**L'architecture Inception-v3** est composée d'environ 48 couches comme le montre la figure II.16 à savoir :

- Étape initiale (Stem) : L'image d'entrée est traitée à travers des couches de convolution et de Max-Pooling initiales pour extraire des caractéristiques de base.
- Modules Inception : Ces modules sont constitués de couches de convolution parallèles avec différentes tailles de filtres pour capturer des caractéristiques multi-échelles.
- Blocs de réduction : Ils utilisent une combinaison de convolutions 1x1 et de Max-Pooling pour réduire efficacement les cartes de caractéristiques.
- Classificateurs auxiliaires : Ils consistent en des convolutions 1x1 et des couches entièrement connectées.
- Global Average Pooling : Les cartes de caractéristiques de sortie sont moyennées globalement pour obtenir un vecteur de caractéristiques de longueur fixe qui représente l'image.
- Couche entièrement connectée : Le vecteur de caractéristiques est alimenté dans une couche entièrement connectée suivie d'une activation SoftMax pour produire les probabilités de classe.

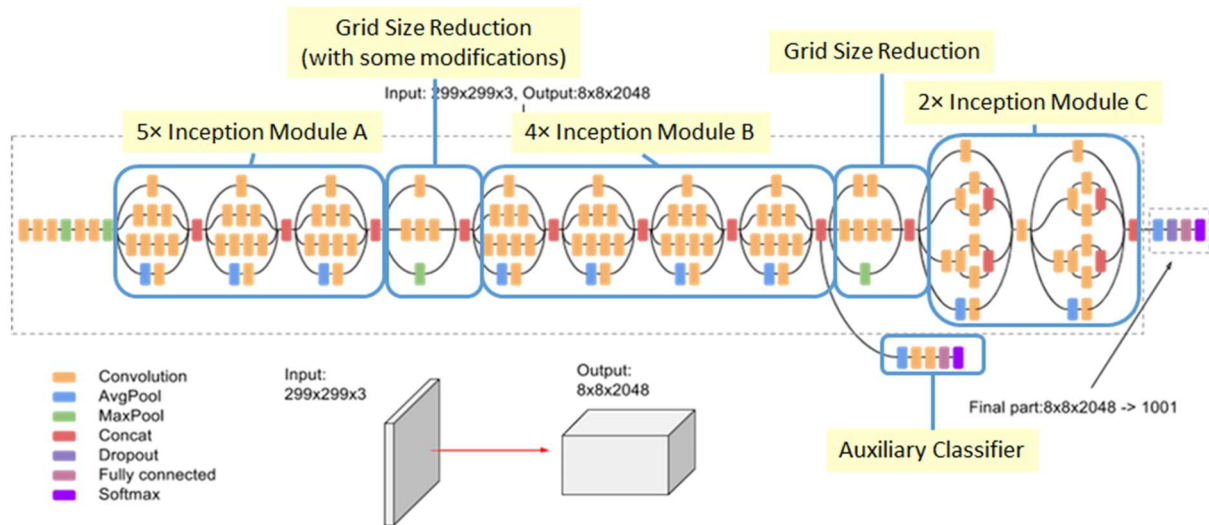


Figure II.16 : architecture d'Inception\_V3 [64]

### II.5.3 SqueezeNet [62]

SqueezeNet est un modèle de classification connu pour sa taille compacte et son architecture efficace [62]. Développé pour pallier les limitations liées au déploiement de modèles d'apprentissage profond sur des appareils aux ressources limitées, SqueezeNet offre un équilibre entre la taille du modèle et sa précision. Il a démontré des performances compétitives sur le jeu de données ImageNet, atteignant une précision TOP-1 de 57% et une précision TOP-5 de 81%.

SqueezeNet se compose d'un total de 18 couches. Les composants clés et les techniques utilisées dans SqueezeNet comprennent :

- Fire Modules : SqueezeNet utilise des modules Fire, qui se composent d'une couche de compression (squeeze layer) et d'une couche d'expansion (expand layer). La couche de compression est une convolution Pointwise 1x1 qui réduit le nombre de canaux d'entrée tout en préservant des informations spatiales importantes, ce qui réduit le nombre de paramètres du réseau, tandis que la couche d'expansion combine des convolutions 1x1 et 3x3 pour capturer et étendre la représentation des caractéristiques.
- Réduction agressive : SqueezeNet applique des couches de Pooling tôt dans le réseau pour réduire de manière agressive les dimensions spatiales de l'entrée, ce qui diminue efficacement les exigences computationnelles [62].

L'architecture et les choix de conception de SqueezeNet donnent lieu à un modèle très efficace et léger sans sacrifices significatifs en termes de précision. Il s'agit d'une option pratique pour les scénarios où les ressources computationnelles et la taille du modèle sont des facteurs critiques.

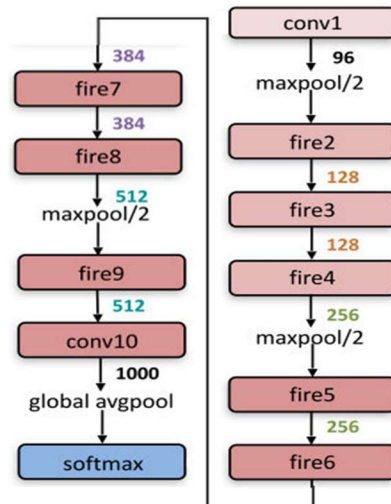


Figure II.17 : architecture de SqueezeNet [65]

## II.6 Évaluation des performances

L'évaluation des performances est un aspect crucial pour évaluer l'efficacité des modèles d'apprentissage profond en réalisant différentes tâches. Pour garantir une évaluation précise, plusieurs méthodes et métriques peuvent être utilisées.

Tout d'abord, une pratique courante dans l'évaluation des performances est l'utilisation de jeux de validation ou de techniques de validation croisée. En divisant les données disponibles en ensembles séparés, le jeu de validation sert de référence pour évaluer les performances du modèle pendant l'entraînement, ce qui aide à sélectionner les paramètres optimaux. De plus, des jeux de test ou des ensembles de rétention sont utilisés pour évaluer la capacité de généralisation du modèle final. En évaluant les performances du modèle sur cet ensemble de test non observé, on peut obtenir des informations sur sa capacité à gérer des scénarios réels.

### II.6.1 Cross-validation

La validation croisée est une technique utilisée pour évaluer les performances d'un modèle en divisant les données disponibles en plusieurs sous-ensembles ou plis. Le modèle est entraîné et évalué de manière itérative, en utilisant différents sous-ensembles comme ensembles d'entraînement et de validation. Cela permet d'estimer les performances du modèle sur des données non observées et peut aider à l'ajustement des hyperparamètres.

### II.6.2 Matrice de confusion

La matrice de confusion est une matrice qui fournit un résumé complet des performances d'un modèle de classification. Elle indique le nombre de vrais positifs (TP), de vrais négatifs (TN), de faux positifs (FP) et de faux négatifs (FN).

Tableau II.1 : Matrice de confusion

	Prévu Positif	Prévu Négatif
Positif	Vrais positifs (TP)	Faux positifs (FN)
Négatif	Faux négatifs (FP)	Vrais négatifs (TN)

### II.6.3 Accuracy

L'Accuracy est une métrique largement utilisée pour l'évaluation d'un modèle neuronal. Elle mesure la proportion d'échantillons correctement classés parmi l'ensemble des échantillons dans une tâche de classification. Elle fournit une mesure générale de la justesse du modèle.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

### II.6.4 Précision

La précision est une mesure largement utilisée comme mesure de performance. Elle évalue la proportion d'instances correctement classées parmi toutes les instances d'une tâche de classification. Elle offre une mesure générale de la justesse du modèle.

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

### II.6.5 Recall

Également connu sous le nom de sensibilité ou taux de vrais positifs. Recall mesure la capacité d'un modèle à identifier correctement les instances positives. Il est calculé en tant que ratio des vrais positifs sur la somme des vrais positifs et des faux négatifs. La formule pour le rappel est :

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

### II.6.6 Score F1

Le score F1 est une métrique couramment utilisée dans les tâches de classification pour évaluer l'équilibre entre la précision et le rappel. Il s'agit de la moyenne harmonique de la précision et du rappel, fournissant un score unique qui représente les performances du modèle. La formule du score F1 est la suivante :

$$F1_{score} = 2 \times \frac{precision \times recall}{precision + recall} \quad (15)$$

### II.6.7 TOP-1 et TOP-5

**Précision TOP-1 :** La précision TOP-1 est une métrique de classification qui mesure le pourcentage d'étiquettes ou de classes correctement prédites en ne considérant que la prédiction la plus probable. Pour évaluer les performances du modèle en utilisant la précision TOP-1, chaque échantillon d'entrée se voit attribuer une classe prédite unique. Si cette prédiction correspond à l'étiquette réelle, elle est comptée comme une prédiction correcte. La précision TOP-1 est

déterminée en divisant le nombre de prédictions correctes par le nombre total d'échantillons. Cette métrique donne une indication de l'aptitude du modèle à attribuer la probabilité la plus élevée à la classe correcte.

**Précision TOP-5 :** La précision TOP-5 est une autre métrique de classification qui mesure le pourcentage d'étiquettes correctement prédites parmi les cinq classes prédites les plus probables. Elle permet une certaine flexibilité en considérant non seulement la prédiction la plus probable, mais aussi les quatre prédictions les plus probables suivantes. Si l'étiquette réelle (Ground Truth) se trouve parmi ces cinq meilleures prédictions, elle est considérée comme une prédiction correcte. La précision TOP-5 est calculée en divisant le nombre de prédictions correctes parmi les cinq meilleures prédictions par le nombre total d'échantillons. Cette métrique est utile dans les tâches de classification complexes et volumineuses où le modèle peut avoir des difficultés à déterminer la prédiction absolue la plus élevée.

## II.7 Conclusion

Dans ce chapitre, nous avons discuté de l'utilisation de l'apprentissage profond pour la reconnaissance d'images. Nous avons examiné l'historique de l'apprentissage profond, sa place dans l'IA et pourquoi il connaît un tel succès dans la recherche et les applications d'aujourd'hui. Nous avons également discuté des différents types d'approches d'apprentissage profond, ainsi que des bases des réseaux neuronaux artificiels et des réseaux neuronaux à convolution. Enfin, nous avons passé en revue certains modèles de classification de pointe et des métriques d'évaluation pour les modèles d'apprentissage profond.

L'apprentissage profond a le potentiel de révolutionner l'inspection visuelle, en particulier l'inspection des bouteilles d'eau. En automatisant le processus d'inspection, l'apprentissage profond peut contribuer à améliorer l'efficacité, réduire les coûts et garantir la qualité des bouteilles d'eau. Cependant, il reste encore des défis à relever avant que l'apprentissage profond puisse être largement adopté pour l'inspection visuelle. Ces défis comprennent la nécessité de disposer de grands ensembles de données d'images étiquetées, la nécessité de ressources informatiques puissantes et la nécessité de développer des modèles robustes capables de généraliser sur de nouvelles données.

Le chapitre suivant présentera notre approche d'inspection visuelle des bouteilles en utilisant Deep Learning.



## **Chapitre III**

### **Inspection Visuelle Des Bouteilles D'eau Par**

### **Apprentissage Profond**

### III.1 Introduction

Dans cette partie du projet, nous présentons notre approche pour résoudre le problème d'inspection visuelle de la qualité des bouteilles d'eau. Nous décrivons en détail chaque étape de notre méthodologie, depuis la préparation des données jusqu'à la validation et l'évaluation des modèles.

Notre approche consiste en particulier à exploiter des modèles pré-entraînés et d'effectuer un apprentissage par transfert (Transfert Learning). Nous concentrons en particulier sur des modèles pré-entraînés connus par leurs performances tels que ResNet18, ResNet50, InceptionV3. Ces modèles ont été entraînés sur de grandes datasets et ce transfert d'apprentissage permet de tirer parti des connaissances préalablement acquises par ces modèles. Le but est de re-entraîner ces réseaux pour qu'ils seront capables de reconnaître des images de bouteilles sans défauts (Propre) ou Défectueuses.

La figure III.1 montre les étapes à suivre dans le développement de notre solution pour la détection des défauts dans les bouteilles d'eau. La première étape consiste à construire un dataset composé des images adéquates à notre travail. Afin d'enrichir le dataset, nous procédons à l'augmentation des données pour que l'entraînement soit plus efficace et suffisant. Ensuite, nous divisons le dataset en parties pour l'entraînement et la validation et le test pour évaluer le modèle entraîné, des modèles pré-entraînés vont être re-entraîner sur ces données et tester pour des résultats désirés, et après on fait le choix de modèle si nous voulons utilise le transfert d'apprentissage où nous développons un modèle dès le début et lançons l'apprentissage et évaluer leurs performances.

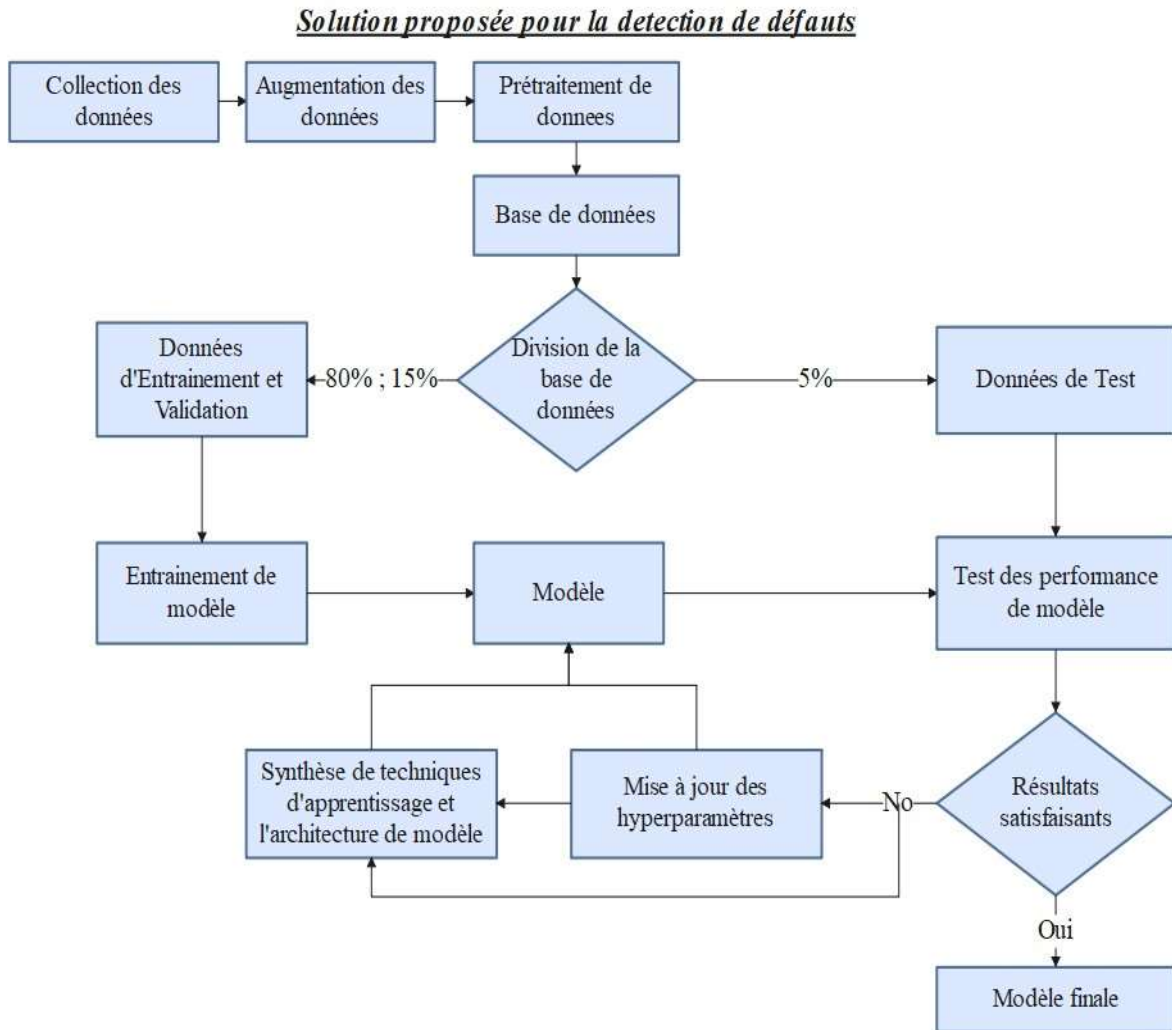


Figure III.1 : Etapes de développement d’une approche de détection des défauts par transfert d’apprentissage

### III.2 Préparation du dataset

Dans le cadre de ce projet, nous avons accordé une attention particulière à la collecte de données pertinentes et représentatives de la réalité du problème traité. Nous avons effectué cette collecte dans divers endroits, y compris la chambre d’inspection visuelle et en dehors de celle-ci, afin de généraliser les données et éviter les problèmes de sur-apprentissage. Cette approche nous permet de mieux aligner notre projet avec des scénarios réels et d’obtenir des résultats fiables et généralisables.

Parmi les différentes catégories de défauts considérées dans ce travail, une attention particulière a été portée aux quatre types spécifiques de défauts : les défauts de bouchon, les défauts de niveau, les défauts d’objet à l’intérieur, les défauts de déformation et finalement la combinaison de ces défauts. Notre jeu de données initial est constitué de 2400 images pour chaque classe. Pour chacun de ces défauts, nous avons 480 images avec différentes variantes et conditions de défauts. Le reste d’image représente des situations où il y a plusieurs défauts dans la même bouteille.

Cette approche nous a permis de disposer d'un ensemble de données riche et diversifié, reflétant de manière réaliste les défis auxquels notre système d'inspection sera confronté dans des situations réelles. Ces données représentent une base solide pour entraîner des modèles en utilisant des méthodes de l'apprentissage profond. De plus, l'utilisation de données provenant de différentes sources et de différents contextes améliore la robustesse de notre système, en lui permettant de généraliser et de s'adapter à des situations variées.

Il convient de noter que la collecte de données est une étape essentielle dans tout projet d'apprentissage automatique, car la qualité et la représentativité des données jouent un rôle important dans les performances du modèle. Ce dataset a été construit sans utiliser de jeu de données open source. En effet, notre objectif était de développer un système d'inspection spécifiquement adapté à l'industrie de fabrication d'eau, et il était donc essentiel d'avoir un jeu de données qui reflète les caractéristiques et les défis spécifiques de ce domaine.

La création de notre propre dataset a également permis une meilleure personnalisation et adaptation aux besoins spécifiques de notre projet. Nous avons pu cibler les défauts les plus courants et les plus critiques rencontrés dans l'industrie de remplissage des bouteilles d'eau, tels que les défauts de bouchon, les défauts de niveau, les défauts d'objet à l'intérieur et les défauts de déformation.

Cependant, afin d'améliorer la diversité et la qualité de ce dataset, nous avons procédé à une augmentation de données. Cette technique consiste à appliquer des transformations et des modifications aux images existantes pour générer de nouvelles instances d'apprentissage. Grâce à cette augmentation de données, nous avons pu augmenter notre dataset jusqu'à 30000 images, ce qui signifie 5 images générées à partir de chaque image originale, offrant ainsi une plus grande variété et une meilleure représentativité des différents cas que notre système d'inspection doit traiter.

### III.3 Types de défauts considérés

La figure III.2 montre des exemples des défauts considérés dans la construction du dataset. Il s'agit des défauts les plus rencontrés dans cette industrie.

### III.4 Augmentation du dataset

L'augmentation d'un dataset joue un rôle crucial dans le domaine de l'apprentissage automatique. Elle consiste à générer de nouvelles données en introduisant certaines variations et transformations aux données initiales. Elle permet d'augmenter la quantité et la diversité des exemples disponibles, ce qui peut améliorer les performances des modèles après apprentissage. Plusieurs opérations ont été appliquées telle que :

La Rotation "rotate" : effectuer une rotation de l'image autour de son centre en utilisant un angle aléatoire (Figure III.3.a). Cela permet de simuler différentes orientations et perspectives des objets présents dans l'image [66]. En effectuant des rotations aléatoires, on peut rendre le modèle plus robuste et capable de reconnaître des objets dans différentes positions.



Figure III.2 : Exemples des défauts dans les bouteilles d'eau considérés dans notre dataset

L'étirement "stretch" : étire l'image verticalement en appliquant un facteur d'échelle aléatoire (Figure III.3.b). Cela peut simuler des variations de taille des objets ou des déformations dues à la perspective [66]. L'étirement permet également de générer des images avec des proportions différentes, ce qui peut rendre le modèle plus résistant aux variations dans les formes et les tailles des objets.

Le contraste "contrast" : ajuster le contraste de l'image en modifiant les valeurs des pixels (Figure III.3.c). Cela peut aider à mettre en évidence les détails importants dans l'image ou à améliorer la visibilité des objets dans des conditions d'éclairage difficiles [66]. En modifiant le contraste, on peut simuler des variations dans l'éclairage de l'environnement et rendre le modèle plus adaptable à des conditions différentes.

Générer des images floues "blur" : appliquer un flou gaussien à l'image, ce qui donne une apparence plus lisse et moins détaillée (Figure III.3.d). Cette transformation peut être utile pour réduire le bruit indésirable dans l'image ou pour simuler des objets flous dans des scènes réelles

[66]. En ajoutant du flou, on peut rendre le modèle plus robuste aux variations dans la netteté des images et aux imperfections de la capture.

Cacher une partie rectangulaire "cutout" : réaliser un découpage aléatoire sur l'image en masquant une partie rectangulaire (Figure III.3.e). Cela permet de simuler des objets partiellement visibles ou occultés par d'autres objets [66]. En utilisant cette transformation, on peut entraîner le modèle à reconnaître les objets même lorsqu'ils sont partiellement visibles ou cachés.

L'introduction d'un bruit "noise" : ajouter du bruit aléatoire à l'image (Figure III.3.f). Cela peut simuler des variations dans les conditions d'éclairage ou des imperfections de l'appareil photo [66]. En introduisant du bruit, on peut rendre le modèle plus résistant aux perturbations et aux variations dans les images réelles.

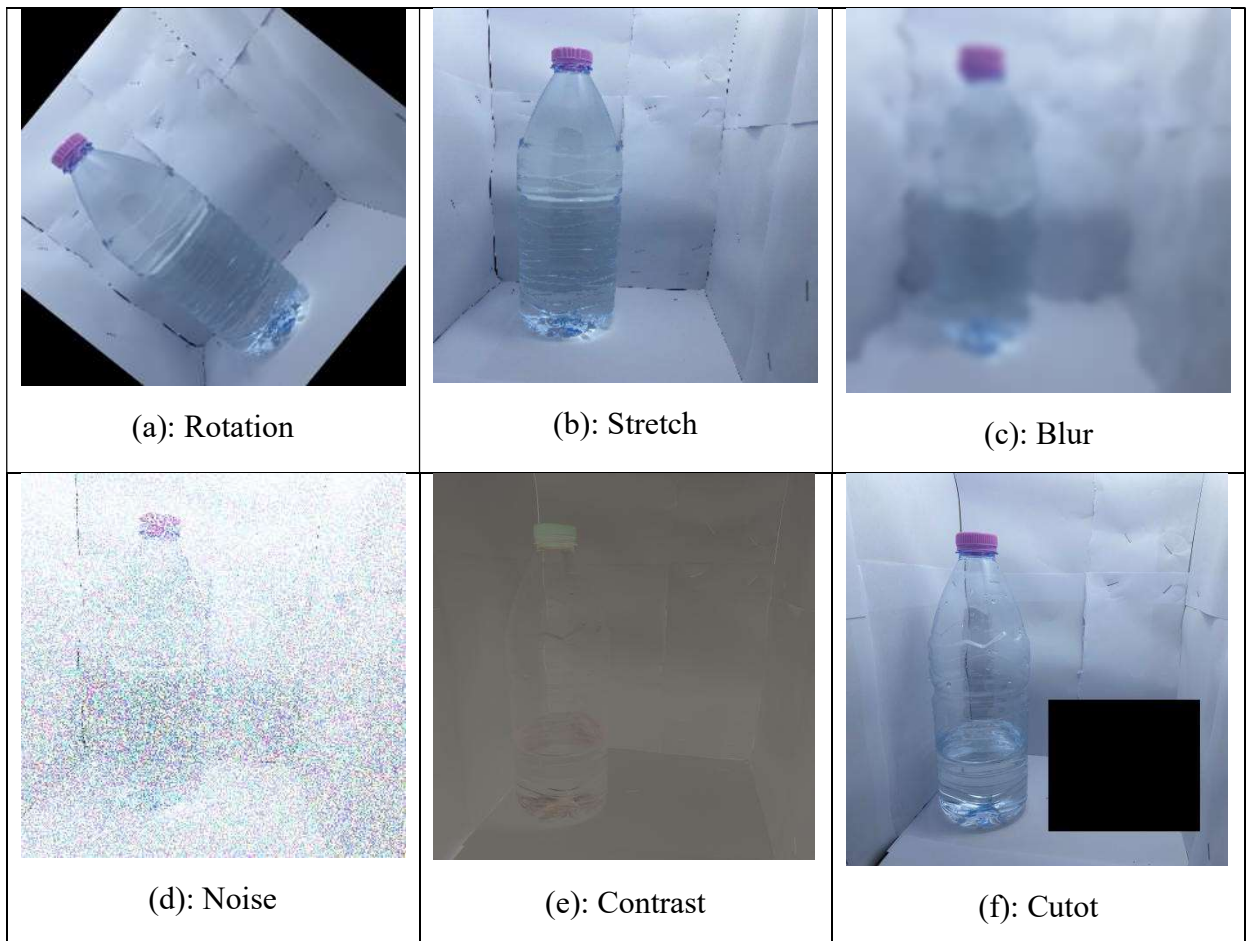


Figure III.3 : Exemples des images générées par des fonctions utilisées pour l'augmentation des données (images)

### **III.5 Prétraitement des images**

L'objectif du prétraitement est d'améliorer les images en supprimant les distorsions indésirables ou en améliorant certaines caractéristiques importantes de l'image pour un traitement ultérieur. Le prétraitement d'images est une étape souhaitable dans tout système de reconnaissance de motifs pour améliorer ses performances et elle est utilisée pour réduire les variations et produire

un ensemble de données plus cohérent. Certaines opérations sont nécessaires selon l'application. Les techniques de prétraitement utilisées dans ce projet sont décrites brièvement ci-dessous.

### III.5.1 Redimensionnement d'image

Le redimensionnement d'une image consiste à modifier ses dimensions, que ce soit la largeur seule, la hauteur seule ou les deux sans la couper. De plus, le rapport d'aspect de l'image d'origine peut être préservé dans l'image redimensionnée. Le redimensionnement d'image peut être interprété comme un type de rééchantillonnage d'image ou de création d'image du point de vue de l'hypothèse de test de Nyquist. Selon cette hypothèse, le sous-échantillonnage vers une image plus petite à partir d'une image originale de résolution plus élevée doit être effectué après l'application d'un filtre de lissage ante-crénelage 2D approprié pour éviter les artefacts de crénelage [67].

Le redimensionnement des images pour des applications liées au Deep Learning permet de réduire le temps d'apprentissage des modèles ainsi que le coût d'un point de vue ressources matérielles (GPU et RAM). En plus, le développement des solutions par transfert d'apprentissage nécessite le redimensionnement des entrées pour les adapter aux exigences des modèles utilisés. Dans notre cas, les images ont été redimensionnées à 224x224 et 299x299 pixels pour des raisons liées à la taille des entrées des modèles considérés.

### III.5.2 Normalisation des images

La normalisation des images est une technique couramment utilisée dans le domaine du traitement d'images pour améliorer les performances des modèles d'apprentissage automatique [66].

L'objectif de la normalisation des images est de rendre les valeurs des pixels de l'image plus adaptées à l'apprentissage automatique en réduisant la variance et en recentrant les données. Cela permet d'égaliser l'échelle des valeurs des pixels et d'éliminer les biais potentiellement présents dans l'ensemble de données.

La normalisation des images se fait généralement en deux étapes :

- Réduction de l'échelle : Les valeurs des pixels sont divisées par une valeur constante, généralement 255 (la valeur maximale possible pour un pixel dans une image en niveaux de gris ou en couleur). Cela permet de ramener les valeurs des pixels dans une plage de 0 à 1, ce qui est souvent nécessaire pour certaines fonctions d'activation.
- Recentrage des données : Les valeurs des pixels sont centrées sur zéro en soustrayant la moyenne de l'ensemble de données. Cela permet d'éliminer les biais et de recentrer les données autour de zéro, ce qui facilite l'apprentissage en évitant des écarts importants dans les valeurs des pixels.

La normalisation des images peut également inclure d'autres techniques, telles que le calcul de l'écart type des pixels et la division par cet écart type pour rendre les données plus stables [66].

Il convient de noter que la normalisation des images est généralement effectuée séparément pour chaque canal de couleur dans les images couleur, afin de maintenir les relations entre les

canaux et de préserver les informations de couleur. La figure III.4 montre les étapes effectuées dans la préparation des données.

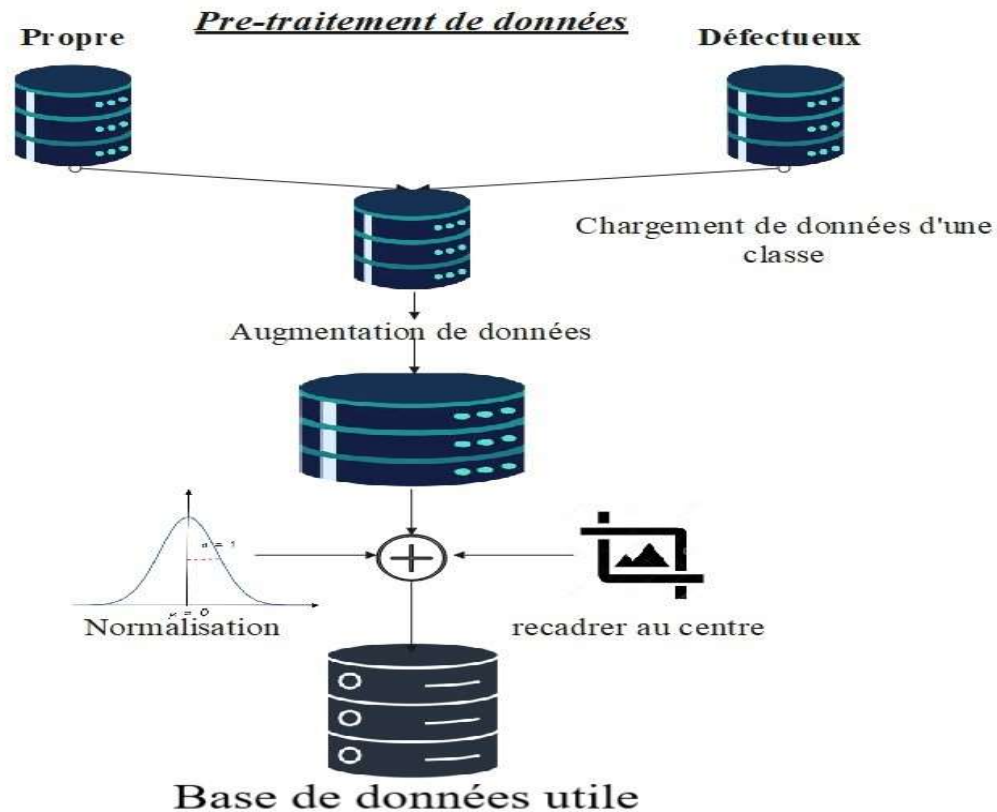


Figure III.4 : Etapes de préparation du Dataset

### III.6 Détection de défauts : solutions proposées

#### III.6.1 Modèles proposés

Au départ, nous avons opté pour effectuer un transfert d'apprentissage en utilisant des modèles pré-entraînés connus par leurs performances. Nous avons commencé par ajuster les dernières couches de classification pour correspondre à notre problème spécifique de deux classes : "Bouteille propre" et "Bouteille défectueuse". Les parties conservées de ces modèles ont le potentiel d'extraire des caractéristiques à partir des images. Les modèles utilisés dans ce projet sont : **ResNet18**, **ResNet50** et **Inception V3**.

Nous avons également proposé une version modifiée du modèle Inception V3 en ajoutant 4 couches complètement connectées comme le montre la figure III.5.

Parallèlement à cela, nous avons également développé un modèle à partir de zéro en utilisant notre dataset composé de 30 000 images de bouteilles propres et défectueuses pour son entraînement. Ce modèle a été conçu spécifiquement pour résoudre notre problème de classification. L'architecture proposée est illustrée par la figure III.6.



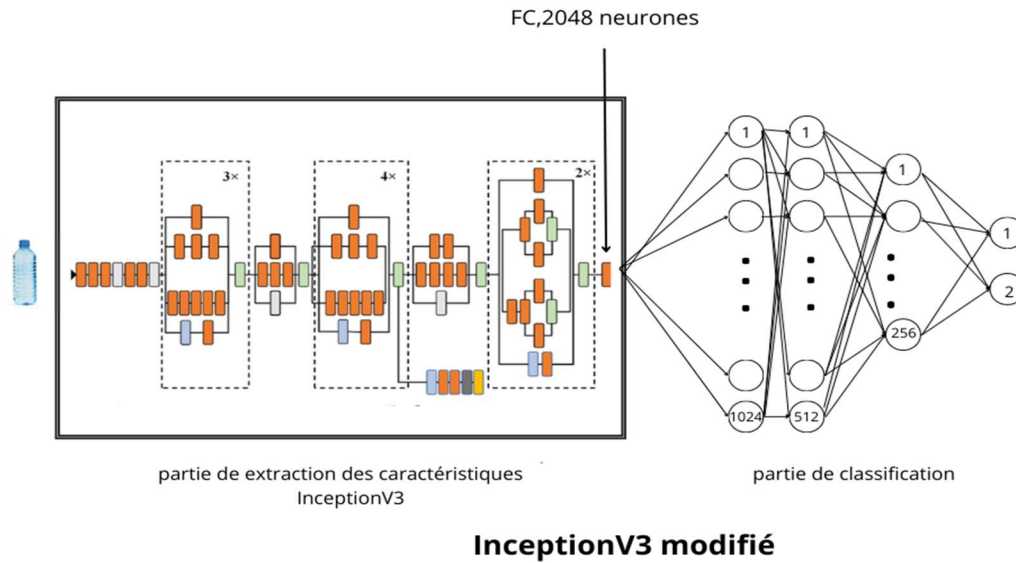


Figure III.5 : Architecture de l’InceptionV3 modifié

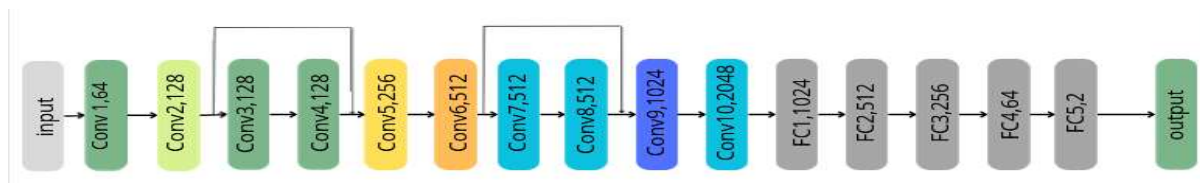


Figure III.6 : Architecture du modèle développé

La conception de ce modèle a été inspirée par la structure du réseau ResNet, car elle offrirait un cadre simple et efficace. Il est composé de 10 couches : 5 couches de classification, chaque couche est suivie d'un max pooling et de 2 blocs résiduels pour atténuer le problème des gradients qui disparaissent et faciliter l'entraînement de réseaux plus profonds. Après la dernière couche CNN, nous avons appliqué un pooling adaptatif moyen, puis une mise en forme. Le classifieur (Perceptron) est composé de 4 couches cachées (Denses) et une couche de sortie avec 2 neurones.

### III.6.2 Apprentissage des modèles

Pour tous les modèles, nous avons adapté les dimensions des images et nous avons fixé le nombre de classes à deux pour représenter les deux cas bouteilles "Propre" et "Défectueuse". Pour améliorer les performances des modèles et éviter le sur-apprentissage, nous avons appliqué aussi plusieurs techniques de régularisation. Tout d'abord, nous avons appliqué la régularisation L2, également connue sous le nom de régularisation de poids au carré, qui ajoute une pénalité proportionnelle au carré des poids du modèle. Cette technique favorise des poids plus petits, ce qui peut réduire l'impact des valeurs aberrantes et améliorer la stabilité du modèle.

La méthode de régularisation « Dropout » a été aussi utilisée. Elle permet d'éviter la dépendance excessive entre les neurones et de rendre le modèle plus robuste en termes de généralisation. Pour accélérer l'entraînement et stabiliser le processus d'apprentissage, la normalisation par lots (Batch normalization) a été aussi exploitée. Cette technique normalise les

activations des couches en calculant la moyenne et l'écart type des activations sur un mini-lot d'échantillons. Cela permet d'atténuer les effets de la dérive de distribution des données lors de l'entraînement, ce qui facilite la convergence du modèle.

La fonction Loss utilisée est la fonction "cross-entropy" (entropie croisée), qui est couramment utilisée dans les tâches de classification. Cette fonction d'erreur mesure la différence entre les probabilités prédites par le modèle et les véritables étiquettes de classe des images.

En ce qui concerne l'optimisation, nous avons choisi d'utiliser l'algorithme Adam, qui est une méthode d'optimisation populaire dans l'apprentissage profond. L'algorithme Adam combine les avantages des méthodes de descente de gradient stochastique (SGD) avec moment et de l'estimation adaptative du taux d'apprentissage. Il consiste à utiliser une estimation adaptative du premier moment (moyenne mobile des gradients) et du second moment (moyenne mobile des carrés des gradients) pour mettre à jour les poids de manière plus précise.

Cela permet d'ajuster automatiquement le taux d'apprentissage en fonction des caractéristiques du problème et d'accélérer la convergence du modèle.

Le tableau III.1 résume les caractéristiques des modèles et des méthodes d'apprentissages utilisés.

Tableau III.1 : Modèles entraînés et leurs caractéristiques

	ResNet18	ResNet50	Inception V3	InceptionV3 modifiée	Notre Modèle
Taille des images	224*224	299*299	299*299	299*299	299*299
Architectures (couches ajoutées)	-----	-----	-----	4 couches Dense pour la classification (Linéaire)	10 pour Conv et 5 pour couches linéaire
Optimiseur	Adam, SGD				
Méthodes de régularisation	Normalisation par lots L2	Normalisation par lots L2 Dropout	Normalisation par lots L2 Dropout	Normalisation par lots L2 Dropout	Normalisation par lots L2 Dropout

Pour déterminer le meilleur taux d'apprentissage, nous avons effectué des expérimentations en utilisant différentes valeurs. Nous avons testé plusieurs plages de taux d'apprentissage et nous avons sélectionné celui qui offrait les meilleures performances.

A base des résultats obtenus, les performances des modèles pré-entraînés InceptionV3, ResNet50, ResNet18 ainsi que notre modèle sont comparés et évalués pour sélectionner le meilleur modèle dans le reste de ce chapitre.

### III.6.3 Choix des hyperparamètres

Une attention particulière a été portée à l'optimisation des hyperparamètres et à l'application de techniques de régularisation pour améliorer les performances de tous les modèles. Les deux optimiseurs (SGD et Adam) ont été testés en ajustant le taux d'apprentissage (learning rate) et le poids de régularisation (weight decay). Rappelant que c'est le taux d'apprentissage qui détermine l'ampleur des modifications apportées aux poids du modèle à chaque itération, tandis que le poids de régularisation contrôle l'importance de la pénalité appliquée aux poids afin de prévenir le surapprentissage.

La combinaison entre ces différents paramètres, un large éventail de combinaisons d'hyperparamètres et de techniques de régularisation a été exploré afin d'identifier les configurations qui produisaient les meilleures performances d'inspection visuelle de la qualité des bouteilles d'eau.

Tableau III.2 : Hyperparamètres utilisés pour l'entraînement des différents modèles

	Taux d'apprentissage	Nombre des Epoques	Poids de régularisation	Pourcentage de désactivation aléatoire (Dropout)
SGD	0.01	50	0.0001	50%
	0.001	60	0.0001	20%
	0.0001	90	0.001	40%
Adam	0.001	50	0.0001	50%
	0.001	60	0.0001	20%
	0.0001	90	0.001	40%

## III.7 Implémentation et évaluations

### III.7.1 Environnement de développement

Pour le développement logiciel des modèles, nous avons opté pour l'utilisation du langage Python en raison de sa flexibilité et de sa richesse en bibliothèques et modules adaptés à l'apprentissage automatique et à la vision par ordinateur. Pour la mise en œuvre des modèles et leurs apprentissages, nous avons utilisé le framework PyTorch, reconnu pour sa facilité d'utilisation, sa puissance et sa capacité pour les implémentations sur des processeurs graphiques (GPU). D'autres librairies ont été aussi utilisées à savoir OpenCV et ONNX.

#### a. Python

Python est un langage de programmation orienté objet polyvalent utilisé dans de nombreux domaines tels que le développement web, la science des données, l'apprentissage automatique et

l'intelligence artificielle. Il possède plusieurs caractéristiques clés qui le rend actuellement le langage le plus utilisé pour le développement des applications de Deep Learning.

### **b. OpenCV**

OpenCV (Open Source Computer Vision Library) est une bibliothèque open source largement utilisée dans le domaine du traitement d'images et de la vision par ordinateur. Elle fournit une vaste gamme de fonctions et d'outils pour manipuler, analyser et comprendre les images et les vidéos.

OpenCV offre des fonctionnalités avancées pour le traitement d'images, telles que la détection de contours, le filtrage, la transformation d'images, la segmentation, la reconnaissance d'objets, la stéréovision, et bien plus encore. Elle permet également de travailler avec des flux vidéo en temps réel, des caméras, des fichiers multimédias et des périphériques d'acquisition d'images.

La bibliothèque OpenCV est écrite en C++ et dispose d'interfaces pour de nombreux autres langages de programmation populaires, tels que Python, Java et MATLAB. Cela rend OpenCV accessible et utilisable dans divers environnements de développement.

Nous avons exploité la bibliothèque OpenCV surtout pour l'augmentation des données et toutes les opérations liées aux images.

### **c. Pytorch**

PyTorch est un framework open source, développé par Facebook, pour l'apprentissage automatique. Il offre une interface flexible et dynamique qui permet aux développeurs de créer et de former des modèles d'apprentissage en profondeur de manière efficace. PyTorch est largement utilisé pour la recherche en intelligence artificielle et il est devenu également populaire dans l'industrie [68].

Une des principales caractéristiques de PyTorch est sa capacité à exécuter des calculs sur des tenseurs de manière efficace, en utilisant des méthodes avancées d'optimisation du calcul sur les GPU. Il fournit des outils pour la gestion automatique des gradients, ce qui facilite la mise en œuvre des algorithmes d'apprentissage en profondeur tels que la rétropropagation.

PyTorch offre une approche orientée vers le code, ce qui signifie que les utilisateurs peuvent écrire du code Python standard pour définir et manipuler leurs modèles. Cela permet une plus grande flexibilité et un contrôle accru par rapport à d'autres frameworks [68]. De plus, PyTorch offre une intégration transparente avec les bibliothèques de calcul scientifique et les outils de visualisation, ce qui facilite l'analyse et la compréhension des données.

Grâce à sa communauté active, PyTorch bénéficie d'un large écosystème de modules complémentaires, de modèles pré-entraînés et de ressources d'apprentissage. Cela permet aux utilisateurs de tirer parti des avancées récentes dans le domaine de l'apprentissage automatique et d'accélérer leur développement [68].

### d. ONNX

ONNX (Open Neural Network Exchange) est une bibliothèque et un format de modèle open source utilisés dans le domaine de l'apprentissage automatique. Elle permet de représenter, partager et exécuter des modèles de réseaux de neurones sur différentes plates-formes et frameworks.

ONNX offre une approche standardisée pour la représentation des modèles de réseaux de neurones, indépendamment du framework utilisé pour les entraîner. Cela signifie qu'un modèle entraîné dans un framework spécifique, comme PyTorch ou TensorFlow, peut être converti en un format ONNX et ensuite utilisé dans d'autres frameworks compatibles avec ONNX.

La bibliothèque ONNX fournit des outils et des API pour faciliter la conversion entre les formats de modèle, ainsi que pour l'importation, l'exportation et l'exécution des modèles ONNX. Elle est disponible dans plusieurs langages de programmation populaires, notamment Python, C++, C#, Java, et permet une intégration aisée avec les frameworks d'apprentissage automatique.

Les principales fonctionnalités de la bibliothèque ONNX comprennent :

- Conversion de modèle : ONNX permet de convertir des modèles entraînés dans différents frameworks en un format ONNX commun. Cela facilite l'échange et la collaboration entre les utilisateurs de différents frameworks.
- Importation et exportation : ONNX permet d'importer des modèles ONNX préalablement entraînés dans les frameworks compatibles, ce qui permet de les utiliser pour l'inférence et la prédiction. De même, les modèles entraînés dans un framework peuvent être exportés au format ONNX pour une utilisation ultérieure dans d'autres frameworks.
- Exécution de modèle : La bibliothèque ONNX offre des fonctionnalités pour charger et exécuter des modèles ONNX sur différentes plates-formes et architectures matérielles. Cela permet de déployer les modèles sur des dispositifs spécifiques, tels que des CPU, des GPU ou des dispositifs mobiles.
- Optimisation et accélération : ONNX propose des techniques d'optimisation pour améliorer les performances des modèles, notamment la fusion d'opérations, la quantification et l'accélération matérielle. Cela permet d'obtenir des prédictions plus rapides et plus efficaces par une version d'exécution ONNX de taille réduite.

En résumé, la bibliothèque ONNX est une plateforme puissante pour représenter, partager et exécuter des modèles de réseaux de neurones. Elle facilite l'interopérabilité entre les frameworks d'apprentissage automatique, permettant ainsi une plus grande flexibilité dans l'échange, le déploiement et l'exécution des modèles [69].

### e. TensorRT

Afin d'accélérer davantage le processus d'inférence, nous avons intégré la bibliothèque TensorRT. C'est une bibliothèque d'optimisation d'inférence de réseau neuronal, développée par NVIDIA, qui exploite les fonctionnalités des GPU pour accélérer les calculs. Ce qui permet d'obtenir des résultats d'inspection en temps réel, réduisant ainsi les temps de traitement et améliorant l'efficacité globale du système [70].

Pour l'implémentation hardware, deux types de GPU ont été utilisés : la GeForce 3050 et la Nvidia Jetson Nano. Chacun de ces GPU présente des caractéristiques et des avantages spécifiques qui les rendent adaptés à différentes étapes du processus.

### III.7.2 Pseudo-codes d'entraînement et d'évaluation

Les pseudo-codes qui font l'entraînement du modèle sur les données d'apprentissage et l'autre qui fait l'évaluation du modèle :

```
Training Algorithm
# Step 1: Load dataset with classes (labels)
dataset = load_dataset()
# Step 2: Resize the images into the network resolution
resized_dataset = resize_images(dataset)
# Step 3: Normalize in range [0,1]
normalized_dataset = normalize_images(resized_dataset)
# Step 4: Create (from scratch) or load model with parameters (Transfer learning)
if load_model:
    model = load_pretrained_model()
    # Freeze convolution layers
    model.freeze_conv_layers()
    # Reshape the classification layers into desired architecture and output layer to
the number of classes
    model.reshape_classification_layers(num_classes)
    # Define loss function
    loss_function = define_loss_function()
    # Choose optimizer (Adam or SGD with momentum)
    optimizer = choose_optimizer()
    # Apply regularization (Wd, Dropout percentage, Batch normalization)
    model.apply_regularization()
    # Initialize hyper-parameters:  $\eta$ , momentum, epochs, batch size
    learning_rate =  $\eta$ 
    momentum = momentum_0
    epochs = nb_epochs
    batch_size = batch_number
    best_accuracy = 0.0
    # Training loop
    for epoch in range(epochs):
        # Perform training steps
        train_loss = train(model, normalized_dataset, loss_function, optimizer,
batch_size)
        # Perform validation steps
```

```

validation_accuracy = validate(model, normalized_dataset)
if validation_accuracy > best_accuracy:
    best_accuracy = validation_accuracy
    save_model(model, "checkpoint.pth")
else:
    save_model(model, "best_model.pth")
# End loop
else: # Train from scratch
    model = create_model()
    # Define loss function
    loss_function = define_loss_function()
    # Choose optimizer (Adam or SGD with momentum)
    optimizer = choose_optimizer()
    # Apply regularization (Wd, Dropout percentage, Batch normalization)
    model.apply_regularization()
    # Initialize hyper-parameters:  $\eta$ , momentum, epochs, batch size
    learning_rate =  $\eta$ 
    momentum = momentum_0
    epochs = nb_epochs
    batch_size = batch_number
    best_accuracy = 0.0
    # Begin training
    for epoch in range(epochs):
        # Perform training steps
        train_loss = train(model, normalized_dataset, loss_function, optimizer,
batch_size)
        # Perform validation steps
        validation_accuracy = validate(model, normalized_dataset)
        if validation_accuracy > best_accuracy:
            best_accuracy = validation_accuracy
            save_model(model, "checkpoint.pth")
        else:
            save_model(model, "best_model.pth")
    # End loop
# Step 7: Export the model into ONNX format
export_model_to_onnx(model, "model.onnx")

```

### Inference Phase

```
# Step 1: Load the ONNX model
model = load_onnx_model("model.onnx")
# Step 2: Set the input stream to the camera
camera.set_input_stream()
while True:
    # Step 3: Resize the image into the model input
    resized_image = camera.resize_image()
    # Step 4: Convert the frames to RGB format
    rgb_image = convert_to_rgb(resized_image)
    # Step 5: Pre-process the frames (e.g., normalization, center-crop)
    preprocessed_image = preprocess(rgb_image)
    # Step 6: Convert the frames to tensors
    tensor_image = convert_to_tensor(preprocessed_image)
    # Step 7: Perform inference on the model
    prediction = model.inference(tensor_image)
    # Step 8: Display the frames in GUI with the predicted class and precision
    GUI.display_frame(rgb_image, prediction.class, prediction.precision)
```

## III.8 Evaluation des différents modèles

Les modèles sélectionnés ont été entraînés sur un ensemble de données d'entraînement comprenant 24000 exemples, validés sur 4500 exemples de validation, et finalement testés sur 1500 exemples de test.

Différentes expérimentations ont été effectuées pour chercher les meilleurs modèles et les configurations qui offrent de bonnes performances pour l'inspection visuelle des bouteilles d'eau. Nous avons évalué les modèles en utilisant des métriques telles que la précision, l'erreur, Recall et Score F1 pour mesurer leur capacité à détecter les bouteilles propres et défectueuses avec précision.

Tableau III.3 suivants résume les résultats trouvées par chaque modèle avec différents optimiseurs et différents hyperparamètres.

### III.8.1 Discussion des résultats d'entraînement des modèles

L'entraînement du modèle ResNet18 composé de 18 couches en utilisant notre dataset a donné une précision meilleure 90.5%. En augmentant le nombre de couches à 50 par le choix de ResNet50, le modèle re-entraîné a pu atteindre une précision d'entraînement de 96.55% et une précision de validation de 97.5%. Les autres modèles pré-entraînés InceptionV3 et InceptionV3 modifié ont permis l'obtention des précisions de validation de 97.87% et de 96.04% respectivement. Le modèle modifié avec plus de couches connectées possède moins de performances que le modèle initial.



Tableau III.3 : Comparaison entre les performances des cinq modèles en fonction de différents paramètres.

Modèle	Optim	Lr	Nombre d'Ép	Précision		Loss	
				Train	Val	Train	Val
ResNet18	SGD	0.001	60	0.84	0.86	0.12	0.14
		0.0001	60	0.85	0.87	0.1	0.11
	Adam	0.001	50	0.88	0.89	0.12	0.11
		0.0001	50	0.9	0.905	0.1	0.11
ResNet50	SGD	0.001	60	0.92	0.93	0.097	0.093
		0.0001	60	0.93	0.935	0.09	0.092
	Adam	0.001	50	0.95	0.96	0.09	0.07
		0.0001	50	0.9655	0.975	0.085	0.06
Inception V3	SGD	0.001	60	0.94	0.92	0.08	0.07
		0.0001	60	0.95	0.93	0.073	0.07
	Adam	0.001	50	0.96	0.95	0.072	0.07
		0.0001	50	0.97	0.9787	0.065	0.068
Inception V3 modifié	SGD	0.001	60	0.95	0.94	0.1	0.12
		0.0001	60	0.96	0.95	0.099	0.11
	Adam	0.001	50	0.96	0.96	0.098	0.105
		0.0001	50	0.9614	0.9604	0.097	0.1
Notre modèle	SGD	0.001	60	0.82	0.82	0.2	0.19
		0.0001	60	0.83	0.83	0.19	0.2
	Adam	0.001	50	0.84	0.83	0.18	0.19
		0.0001	50	0.84	0.85	0.19	0.16

Pendant l'entraînement des modèles, nous avons considéré différents algorithmes d'optimisation avec différents taux d'apprentissage, en plus des mesures de précision, pour savoir dans quelle mesure le modèle peut apprendre à classifier les défauts des bouteilles d'eau. Les taux d'apprentissage utilisés pour tester chaque optimiseur sont 0,001 et 0,0001. Les algorithmes d'optimisation utilisés étaient la descente de gradient stochastique (SGD) et l'estimation adaptative du moment (Adam). Les performances obtenues pour les différentes situations sont présentées dans le tableau III.3. Les meilleurs résultats correspondent à l'utilisation de l'optimiseur Adam avec un taux d'apprentissage Lr initiale de 0.0001.

La figure III.7 montre l'évolution de la précision (Accuracy) et de l'erreur (Loss) des différents modèles en utilisant Adam comme optimiseur avec un taux d'apprentissage Lr de 0.0001. Les performances des deux réseaux ResNet50 et Inception V3 sont très proches.

D'un point de vue convergence, les deux modèles Inception V3 (original et modifié) convergent très rapidement par rapport aux autres modèles. Cette convergence rapide est due à l'architecture d'Inception, qui a été conçue pour avoir une grande capacité à capturer diverses caractéristiques comme indique dans la figure III.7.

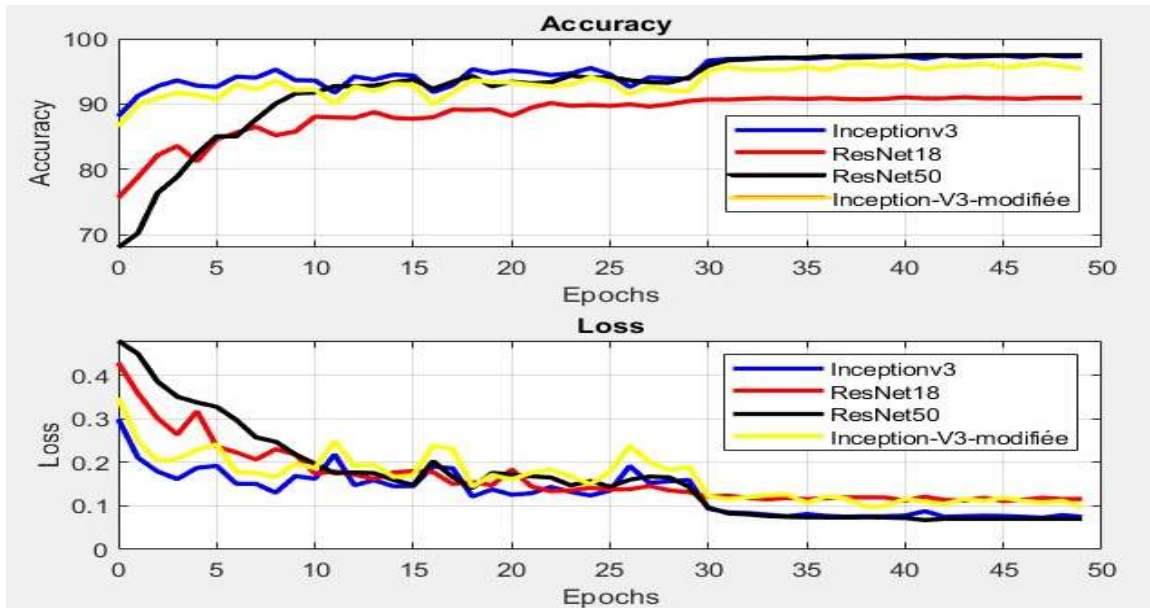


Figure III.7 : Évolution de l'Accuracy et Loss pendant l'apprentissage des différents modèles en utilisant ADAM pour Lr initiale 0.0001.

Comme nous l'avons discuté dans le chapitre deux, ResNet18 et ResNet50 sont deux modèles CNN pré-entraînés populaires. Rappelant que ResNet18 est composé de 18 couches, dont 17 couches de convolution et une couche entièrement connectée. Les filtres utilisés dans les couches de convolution varient en nombre, allant de 64 à 512, en respectant les puissances de deux. D'autre part, ResNet50 est composé de 50 couches, dont 49 couches de convolution et une couche entièrement connectée. Les deux modèles utilisent des filtres de taille 3x3 mais un nombre variable de filtres appliqués. Les couches entièrement connectées utilisent 2048 neurones. La figure III.8 illustre l'avancement de la précision d'apprentissage et validation pour les 2 modèles en utilisant optimisateur ADAM avec Lr initiale 0.0001.

D'une autre part, Dans notre projet, InceptionV3 est un modèle composé de plusieurs blocs d'Inception, qui sont des modules de convolution conçus pour capturer des informations à différentes échelles spatiales. Ces blocs contiennent des filtres de différentes tailles (1x1, 3x3, 5x5) pour extraire des caractéristiques à différentes résolutions.

D'autre part, la version d'InceptionV3 modifié contient 4 couches de classifications. Ces couches entièrement connectées sont utilisées pour réduire progressivement les dimensions et générer à la fin uniquement deux sorties correspondant aux classes considérées.

Le modèle InceptionV3 a donné de meilleurs résultats par rapport aux modèles évalués précédentes ResNet18 et 50 comme le montre le tableau III.3 et Figures III.7. Les valeurs de précision obtenues sont de 97% pour l'entraînement et 97.8% pour la validation. Cependant, les erreurs sont de 6.5% pour l'entraînement et de 6.8% la validation.

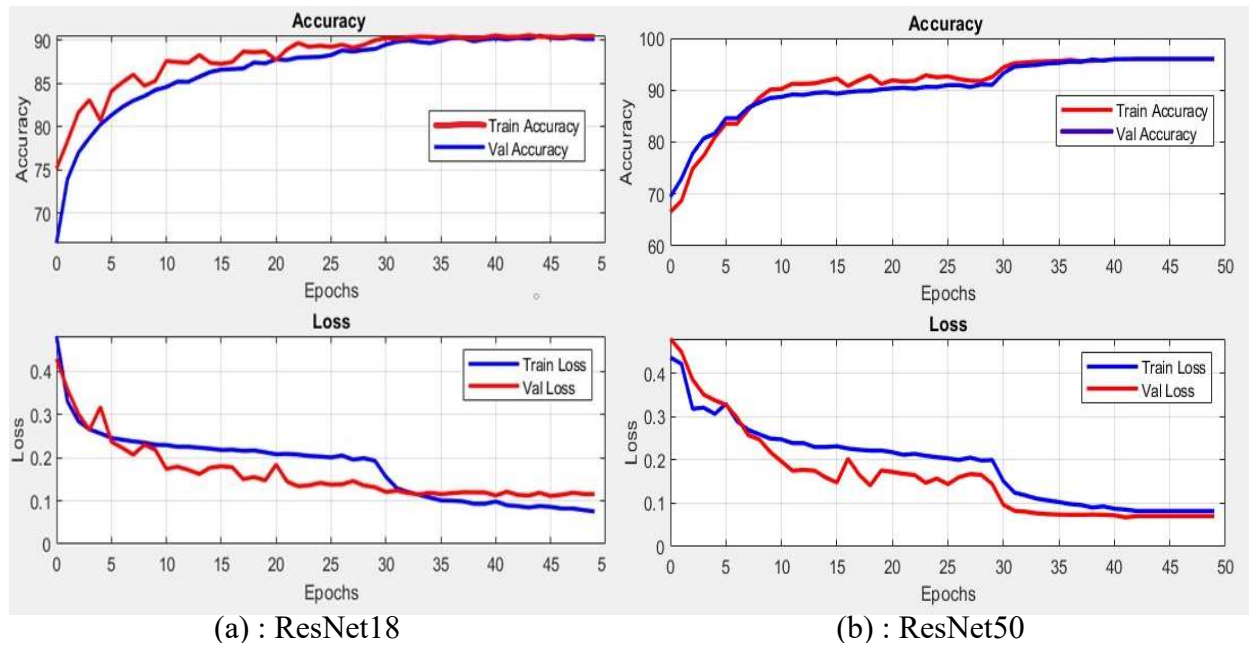


Figure III.8 : Evolution de l'Accuracy et l'erreur de ResNet 18 et ResNet 50

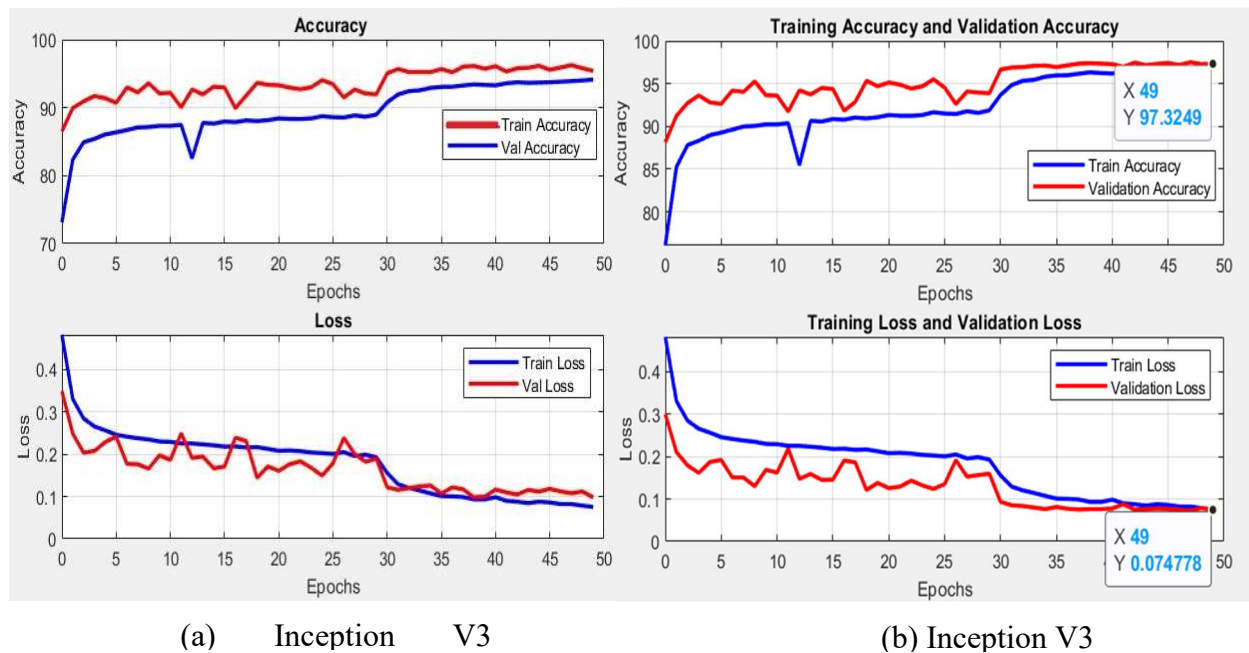


Figure III.9 : Evolution de l'Accuracy et de l'erreur des modèles Inception V3.

La figure III.9 montre les métriques d'évaluation de la précision des modèles sélectionnés InceptionV3 original et modifié (optimiseur Adam avec un taux d'apprentissage de 0,0001). Les deux modèles ont convergé au voisinage de 50ème époque.

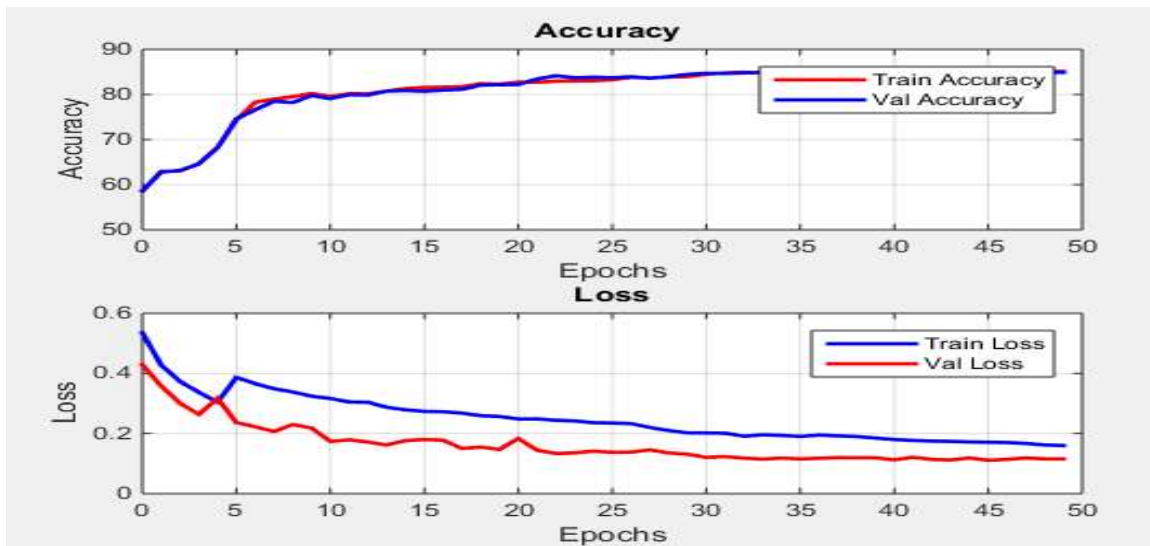


Figure III.10 : Evolution de l'Accuracy et l'erreur du modèle développé (Scratch)

En évaluant le modèle construit dans ce projet (Figure III.6), les performances les modèles pré-entraînés ont largement dépassé celles obtenues avec ce modèle (from scratch Figure III.9). Les meilleures performances avec ce modèle sont une précision de 85% et une erreur de 19% dans la validation par l'optimiseur Adam avec un taux d'apprentissage de 0.0001. Cela montre l'intérêt principal du l'apprentissage par transfert. Notre dataset, les ressources utilisées ainsi que le temps d'entraînement sont insuffisantes pour rendre le réseau capable d'extraire les caractéristiques importantes des images.

### III.8.2 Temps d'entraînement

Les modèles que nous avons entraînés ils pris une durée d'entraînement :

Inception\_V3 : 10 heures                      ResNet18 :6 heures                      Notre Modèle : 26 heures  
 Inception\_V3 modifié :11 heures      ResNet50 : 10 heures

### III.9 Performances des modèles face aux données de test

Les modèles ont été testés en tenant compte à la fois des phases de développement et de déploiement. Lors de la phase de développement, les modèles ont été testés sur l'ensemble de données d'images de test qui est composé de 480 images (le nombre d'images par classe est 240 images).

L'étape de test commence par l'application de toutes les opérations de prétraitement qui ont été appliquées aux données d'entraînement. Ensuite, les images de test sont traitées par les différents modèles et les prédictions obtenues sont utilisées pour leurs évaluations. Le tableau III.4 représente les taux de TP, FP, TN et FN. Les mesures de performances (accuracy, recall, précision et score F1) sont résumées dans le tableau III.5.

Tableau III.4 : Matrice de confusion du test pour chaque modèle entraîné

	True Positive	False Positive	True Negative	False Negative
Inception_v3	94%	6%	92%	8%
Modified inception_v3	86%	14%	88%	12%
Notre modèle (SCRATCH)	80%	19.9%	86%	16%
ResNet18	82,1%	18%	87%	13%
ResNet50	97%	3%	87,7%	12,3%

Tableau III.5 : Performances des modèles dans la phase de test

	Accuracy	Recall	Precision	Score F1
Inception_V3	91%	92.1%	94%	93%
Inception_V3 Modifié	87%	87.7%	86%	86.8%
Notre modèle (SCRATCH)	84.5%	86.3%	82%	84%
ResNet18	83.05%	83.3%	80,1%	81.6%
ResNet50	92.35%	88.7%	97%	92.66%

A partir des tableaux III.4 et III.5, les modèles InceptionV3 et ResNet50 ont permis d'obtenir des résultats proches et qui dépassent celles des autres modèles.

- En se basant sur le taux de vrais négatifs plus élevé d'Inception\_v3 (92%) par rapport à ResNet50 (87,7%), cela suggère qu'Inception\_v3 a une meilleure capacité à identifier avec précision les bouteilles défectueuses. Cela est avantageux lorsque l'objectif est de minimiser les faux négatifs et de garantir que les bouteilles défectueuses sont correctement classées.
- De plus, en tenant compte du taux de rappel plus élevé d'Inception\_v3 (92,1%) par rapport au taux de rappel de ResNet50 (88,7%), cela renforce l'idée selon laquelle Inception\_v3 a une meilleure capacité à détecter les bouteilles défectueuses, réduisant ainsi les chances de faux négatifs.

Dans le reste de ce travail, les performances du meilleur modèle, InceptionV3, vont être analysées en profondeur en considérant différentes situations.

Le Tableau III.6 présente des tests individuels considérant des types particuliers de défauts. Pour effectuer cette évaluation, nous avons préparé de nouvelles données de test plus général. Cet ensemble de données est composé de 60 images pour chaque défaut. Les résultats obtenus sont très encourageants. Cependant, nous avons constaté que le modèle est moins sensible aux défauts qui concernent dans la partie haute d'une bouteille donc dans ce cas nous n'avons pas pris les résultats de défaut fermeture du bouchon. Cette limite et la solution proposée pour la contourner sont discutées dans le paragraphe suivant.

Tableau III.6 : matrice de confusion du modèle entraîné InceptionV3 pour chaque défaut

	Impuretés	Plusieurs défauts	Déformation	Défaut niveau	Fermeture de Bouchons
True Negative	91%	95%	93%	89%	57%
False Negative	9%	5%	7%	11%	43%

### III.10 Inspection de la partie haute d'une bouteille : un réseau spécialisé (InsPart)

L'analyse des performances du modèle obtenu précédemment (Inception V3 après transfert d'apprentissage) a montré certaines limites. Des taux de true Négative considérable ont été notés surtout lorsqu'il s'agit des défauts dans la partie haute des bouteilles. Pour dépasser cette limite, nous avons proposé de construire un nouveau modèle qui se chargera de concentrer sa recherche de défauts dans la partie haute des bouteilles.

En utilisant les résultats obtenus précédemment, nous avons décidé de développer un nouveau modèle Inception V3 que nous avons nommé "**InsPart**" et de le re-entraîner en utilisant de nouvelles données. Pour différencier entre ce modèle spécialisé et le premier, nous avons nommé le modèle développé précédemment "**InsBot**".

Ce choix a été effectué en se basant sur les observations indiquées dans les tableaux III.3, III.4 et III.5 (Nombre d'époque, précision, erreur, ...). Pour entraîner ce modèle, nous avons construit un dataset composé de de 3200 images, dont 2600 pour l'entraînement et 400 pour la validation. Les résultats obtenus sont présentés dans le tableau III.7 et montrent les paramètres (Adam, Taux d'apprentissage de 0.0001) qui ont permet d'obtenir les meilleures performances (une précision de validation de 98% et une erreur de 8% seulement). L'évolution des performances, de ce modèle, durant l'apprentissage est illustrée dans la figure III.10.

Tableau III.7 : Évaluation de l'apprentissage par transfert d'un InceptionV3 (**InsPart**) pour la détection des défauts dans la partie haute d'une bouteille

Modèle	Optim	Lr	Nombre d'Ep	Précision		Loss	
				Train	Val	Train	Val
InsPart	SGD	0.001	60	0.90	0.92	0.17	0.19
		0.0001	60	0.94	0.94	0.12	0.1
	Adam	0.001	50	0.96	0.95	0.1	0.09
		0.0001	50	0.97	0.98	0.09	0.08

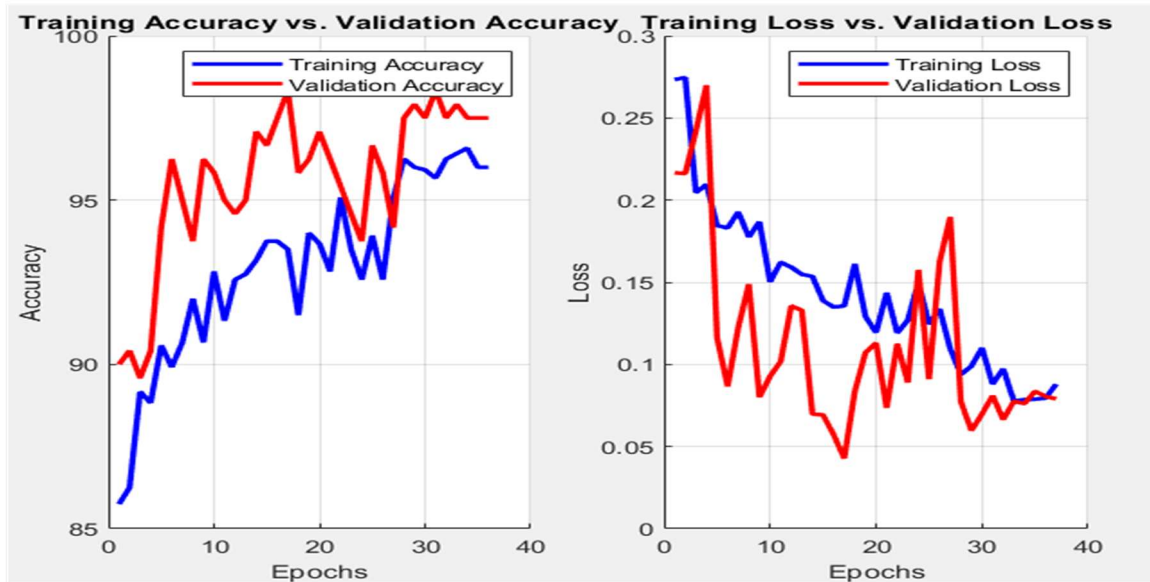


Figure III.11 : l'évolution de l'Accuracy et l'erreur pour le modèle spécialisé (**InsPart**) dans la phase d'entraînement et validation

Après avoir entraîné le modèle, nous avons préparé un ensemble de 200 images pour les deux classes et nous avons évalué ses capacités de généralisation. Le tableau III.8 montre les qualités de ses prédictions. Il a pu détecter et reconnaître correctement 98% des bouteilles propres et 97% pour les bouteilles défectueuses. Ces performances sont très satisfaisantes.

Tableau III.8 : Matrice de confusion et performance du modèle spécialisé selon plusieurs critères

	TRUE POSITIVE	FALSE POSITIVE	TRUE NEGATIVE	FALSE NEGATIVE
InsPart	98%	2%	97%	3%

Tableau III.9 : Performances du modèle spécialisé dans la phase de test

	Accuracy	Recall	Precision	Score F1
InsPart	97.5%	97.02%	98%	97.5%

### III.11 Architecture finale du système de détection des défauts 'InsComb' pour les deux modèles en parallèle

Après l'entraînement des modèles et avoir les testé et évaluer, nous avons choisi les modèles qui ont donnée de meilleurs résultats. Nous allons combiner en parallèles les deux modèles : le modèle " **InsBot**" qui fait l'inspection de toute la bouteille et le modèle " **InsPart**" qui fait l'inspection juste la partie haute de bouteille comme le montre la figure III.11. Pour faire l'inspection visuelle d'une manière efficace et fiable de tous les défauts dans une bouteille, les deux modèles sont combinés. Une bouteille est considérée sans défauts (Propre) si les deux modèles (**InsPart** et **InsBot**) indiquent que la bouteille est propre. Si un des modèles détecte des défauts, on la considère défectueuse.

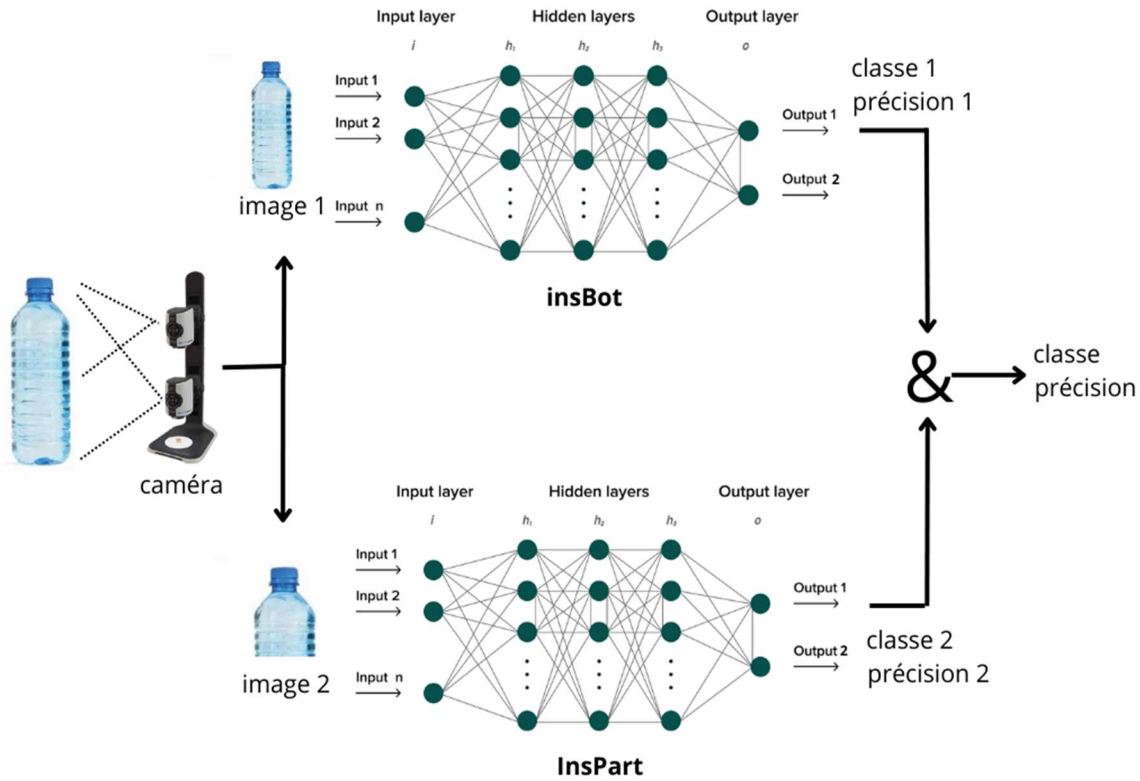


Figure III.12 : schéma du système totale ‘InsComb’ les deux modèles en parallèle

Les tableaux III.10 et III.11 présentent les résultats de test du système final composé des deux modèles et qui fonctionnent en parallèle. Ce test a été effectué en utilisant 800 images par classe.

Tableau III.10 : Matrice de confusion et performances du système d’inspection final

	TRUE POSITIVE	FALSE POSITIVE	TRUE NEGATIVE	FALSE NEGATIVE
InsComb	95%	5%	94%	6%

Tableau III.11 : Performances du système d’inspection final

	Accuracy	Recall	Précision	Score F1
InsComb	94.5%	94%	95%	94.5%

Tableau III.12 : Matrice de confusion du modèle final pour chaque défaut

	Impuretés	Plusieurs défauts	Déformation	Défaut niveau	Fermeture de Bouchons
True negative	94%	93%	94%	95%	95%
False negative	6%	7%	6%	5%	5%



Les tableaux III.10, III.11 et III.12 montrent que le système proposé ‘**InsComb**’ donne de bons résultats par rapport à l’utilisation d’un seul réseau. Ce modèle sera exploité dans le développement du système complet d’inspection visuelle des défauts dans les bouteilles d’eau.

### III.12 Conclusion

À la fin de ce chapitre, nous avons validé notre une solution de détection de défauts par transfert d’apprentissage. Le modèle final composé de deux modèles élémentaires complémentaires a été évalué en considérant différentes métriques et a montré des performances très encourageantes. Dans le chapitre suivant, nous présentons la conception matérielle du système global et l’implémentation du système de détection ‘InsComb’ dans l’unité de traitement dédiée à cette application. Nous procéderons également à la réalisation de plusieurs tests pour évaluer ses performances.

## **CHAPITRE IV**

### **Prototype du Système d'inspection visuelle**

### IV.1 Introduction

Dans le chapitre précédent, nous avons présenté la solution logicielle pour un système complet d'inspection de bouteilles d'eau, en mettant l'accent sur le développement et l'évaluation des modèles d'apprentissage profond adaptés à des parties spécifiques de la bouteille. Ce chapitre sera consacré à la conception et la réalisation de l'aspect matériel de notre système et l'intégration de toutes les parties logicielles. Le système global sera évalué et examiner selon différents critères de performance.

Le système complet intègre les cinq composants clés suivant : système de détection spécialisé, Automate programmable (PLC) pour le contrôle, Interface homme-machine (HMI) pour l'interface utilisateur et la supervision, Page web pour la surveillance à distance et la Base de données SQL pour le stockage des données statistiques. La combinaison entre ces modules permet de construire un système complet et efficace. Ensemble, ils garantissent la qualité, la fiabilité et le bon fonctionnement du processus d'inspection des bouteilles d'eau.

Pendant la phase de développement, plusieurs métriques d'évaluation ont été utilisées pour tester l'efficacité et l'efficacité de nos modèles d'apprentissage profond dans le chapitre 3. Cependant, dans le chapitre 4, nous portons notre attention sur les implications pratiques de notre solution. L'évaluation sera focalisée en particulier sur des aspects liés au fonctionnement temps réel du processus d'inspection. L'objectif est d'optimiser l'efficacité du système et de garantir un processus d'inspection rapide et sans accroc.

### IV.2 Architecture globale du système complet

#### IV.2.1 Cahier des charges

La conception et la réalisation d'un système complet d'inspection visuelle des défauts dans les bouteilles d'eau prendront en compte les objectifs suivants :

- Détection de défauts, à base de Deep Learning, avec de bonnes performances ;
- Bonne organisation des éléments matériels (chambre d'inspection) et intégration dans une chaîne de production et ;
- Possibilité d'intervention des opérateurs pour un contrôle sur site à travers une simple IHM
- Possibilité de supervision à distance à travers une page web et conservation des données d'inspection.

#### IV.2.2 Description du système complet d'inspection

Pour répondre à ces exigences nous avons proposé l'architecture globale du système représentée par la figure IV.1. Le système d'inspection des bouteilles d'eau est une solution complète composée de cinq principaux éléments. Au cœur du système se retrouve le système de détection en utilisant deux modèles **InsComb** neuronal basés sur Inception\_v3 (développés dans le chapitre 3). Ces réseaux fonctionnent sur la plateforme Jetson Nano Developer Kit et qui est équipé de caméras vidéo dédiées. Le premier réseau inspecte la bouteille dans son intégralité, tandis que le deuxième se concentre sur la détection des défauts de la partie supérieure de la bouteille.

L'inspection des bouteilles est réalisée dans une chambre d'inspection où les deux caméras sont positionnées. Des conditions spécifiques d'éclairage ont été mises en place pour garantir que l'inspection ne soit pas perturbée par des facteurs externes. Une fois le système prédit l'état de la bouteille, l'information sera transmise à la deuxième partie du système, le PLC S7-1516-3PN. Cette unité de traitement communique avec une API (S7 1500) via un câble Profinet afin de transmettre les données. Le PLC contrôle le système d'inspection en émettant des commandes de démarrage et d'arrêt pour les différents composants de la chaîne d'inspection (Tapis roulante, Système de détection, la lumière ...etc.). Il gère également la ligne d'inspection et, en cas de bouteille défectueuse, active l'actionneur pour la rediriger vers un processus de vérification supplémentaire.

Le troisième élément du système est l'interface homme-machine (HMI) KTP700 Basic, qui permet aux opérateurs de commander la ligne d'inspection. Ils peuvent démarrer ou arrêter le processus selon les besoins. L'HMI facilite également la supervision de la ligne d'inspection en temps réel et affiche les éventuelles alarmes.

Le quatrième composant est une page web développée avec NODE-RED, servant d'outil de supervision à distance. Les opérateurs peuvent surveiller le système d'inspection à partir de cette page web et être informés des éventuelles alarmes ou notifications.

Enfin, le cinquième élément est une base de données SQL qui stocke les statistiques du processus d'inspection. Cela inclut le nombre de bouteilles inspectées, le nombre de bouteilles défectueuses et d'autres métriques pertinentes. La base de données SQL permet l'analyse et le suivi de la performance du système d'inspection.

- **Chambre d'inspection**

Afin de garantir une inspection sans perturbations externes et d'optimiser la précision du système de détection de défauts, une chambre d'inspection a été installée sur la chaîne d'inspection. Les deux caméras sont positionnées à l'intérieur de cette chambre. Elle est aussi équipée d'une source d'éclairage LED placée en haut et elle est recouverte pour éliminer les reflets lumineux sur les caméras. La Figure IV.3 illustre cette configuration.

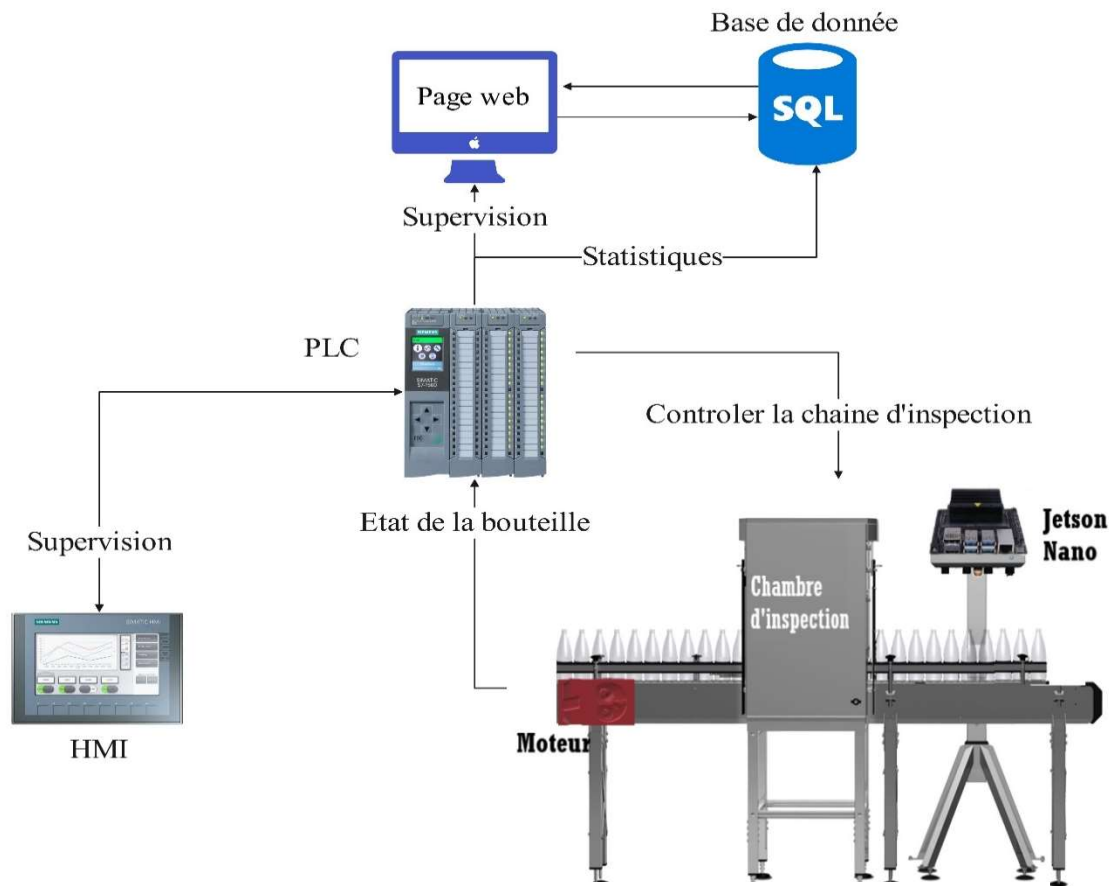


Figure IV.1 : Architecture globale du système d'inspection



Figure IV.2 : La chaîne d'inspection

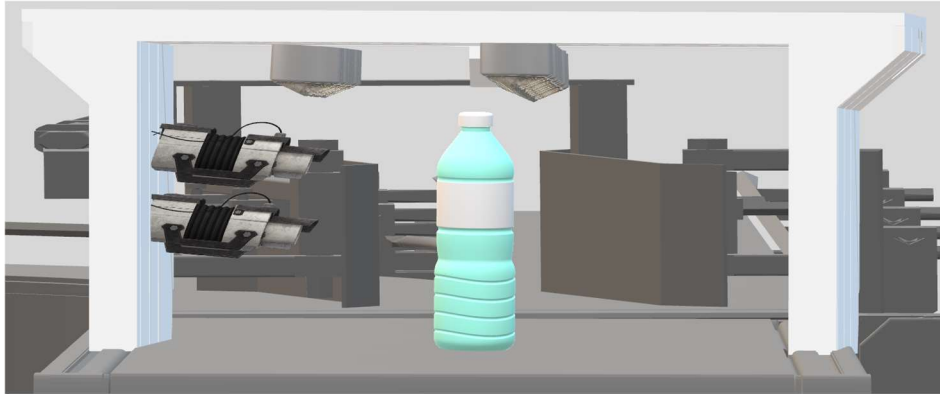


Figure IV.3 : Chambre d'inspection

### IV.3 Environnement de développement (système de détection de défauts)

Les GPU (unités de traitement graphique) jouent un rôle essentiel dans le projet d'inspection visuelle des bouteilles d'eau, tant pour l'entraînement du modèle que pour le déploiement. Dans ce projet, deux types de GPU ont été utilisés : Nvidia GeForce RTX 3050 et Nvidia Jetson Nano Developer Kit. Chacune de ces GPU présente des caractéristiques et des avantages spécifiques qui les rendent adaptés à différentes étapes du processus.

La Nvidia RTX 3050 [71] est une carte graphique puissante spécialement conçue pour les jeux vidéo et les applications d'intelligence artificielle. Elle dispose des cœurs de traitement CUDA, qui sont optimisés pour exécuter des opérations parallèles sur les tenseurs utilisés dans les modèles de réseaux de neurones. La GeForce 3050 offre des performances élevées en termes de vitesse de calcul, ce qui accélère l'entraînement du modèle sur de grands ensembles de données. Grâce à sa capacité à gérer des charges de travail intensives, la GeForce 3050 permet de réduire considérablement le temps nécessaire pour entraîner des modèles complexes et des résultats de modèle plus précis.

La NVIDIA Jetson Nano Developer Kit [72], quant à elle, est un ordinateur embarqué spécialement conçu pour les applications d'intelligence artificielle et de vision par ordinateur. Elle est équipée d'un GPU intégré et d'un processeur ARM, ce qui lui permet d'exécuter des tâches de vision par ordinateur directement sur le dispositif, sans avoir besoin d'une connexion à un serveur distant. La Jetson Nano est idéale pour le déploiement du modèle d'inspection visuelle sur des systèmes embarqués ou des appareils à faible consommation d'énergie. Elle offre une combinaison de puissance de calcul et d'efficacité énergétique, ce qui la rend appropriée pour des applications sur le terrain ou dans des environnements où les ressources sont limitées.

Afin d'optimiser le processus d'entraînement, nous avons exploité la puissance de calcul exceptionnelle de la NVIDIA RTX 3050. En tirant parti de ses capacités, nous avons pu entraîner et évaluer différentes architectures (chapitre 3) et sélectionner les meilleurs modèles. Pour un déploiement efficace dans des applications industrielles embarquées ayant des ressources limitées, nous avons utilisé le kit de développement Jetson Nano, connu par sa capacité à exécuter efficacement des modèles d'IA lors de la phase d'inférence. Cette combinaison stratégique de

matériel de pointe garantit des performances optimales et une intégration fluide tout au long du processus d'entraînement et de déploiement de l'IA. Tableau IV.1 montre les caractéristiques des deux GPUs.

Tableau IV.1 : Caractéristiques des GPUs utilisés

GPU	GEFORCE RTX 3050	JETSON NANO
Architecture	Ampere Cœurs CUDA : 2048	Processeur : ARM Cortex-A57 64 bits  GPU : Nvidia Maxwell 128 cœurs CUDA
Mémoire	8GB GDDR6	4 GB de RAM LPDDR4
Système d'exploitation	NON, intégré dans un ordinateur	Oui, Linux for Tegra (L4T)
Connectivité	DisplayPort, HDMI, DVI	Gigabit Ethernet, Wi-Fi 802.11ac, Bluetooth
Interface mémoire	128 bits	64 bits

- **Test en temps réel de système de détection**

Afin d'évaluer les performances du système d'un point de vue exécution en temps réel du modèle, une série de tests d'inférence ont été réalisés en utilisant les modèles sur un flux en direct provenant des caméras. L'inférence a été appliquée sur le modèle Inception\_v3 afin de détecter les défauts sur l'ensemble de la bouteille, tandis qu'un autre test a été réalisé spécifiquement pour la partie supérieure de la bouteille (réseau spécialisé). Enfin, les deux modèles ont été testés en parallèle.

Une interface graphique (GUI) affiche en temps réel l'inspection des bouteilles par le modèle, générant un résultat sous forme de texte indiquant si la bouteille détectée par les caméras est considérée comme "Propre" ou "Défectueuse", ainsi que la précision de la prédiction. Les figures IV.4, IV.5 et IV.6 illustrent respectivement des exemples des résultats retournés en utilisant les trois configurations (Modèle **InsBot**, **InsPart**, les 2 modèles en parallèle **InsComb**).

Dans le cadre de cette évaluation, le nombre des images traitées par seconde (Frames Per Second, FPS) a été calculé, fournissant une mesure quantitative du taux auquel les images individuelles sont affichées et traitées par seconde dans une vidéo. Une valeur de FPS plus élevée indique généralement que module d'inspection est capable de traiter un nombre important d'images et qu'on peut l'intégrer dans un processus où la vitesse du flux de bouteilles est importante. C'est une mesure importante pour déterminer à la fois la vitesse de réponse du modèle et sa réponse aux exigences liées aux critères liés à une application d'inspection temps réel.

L'évaluation a été réalisée sur trois plateformes matérielles afin d'évaluer les exigences et les performances du modèle sous différents frameworks. Le matériel utilisé dans le test comprenait

## CHAPITRE IV : Prototypé du Système d'Inspection Visuelle

le GPU Nvidia RTX 3050, le CPU Intel Core i5-6300U et le kit de développement Jetson Nano. Les résultats obtenus sont portés dans le Tableau IV.2.

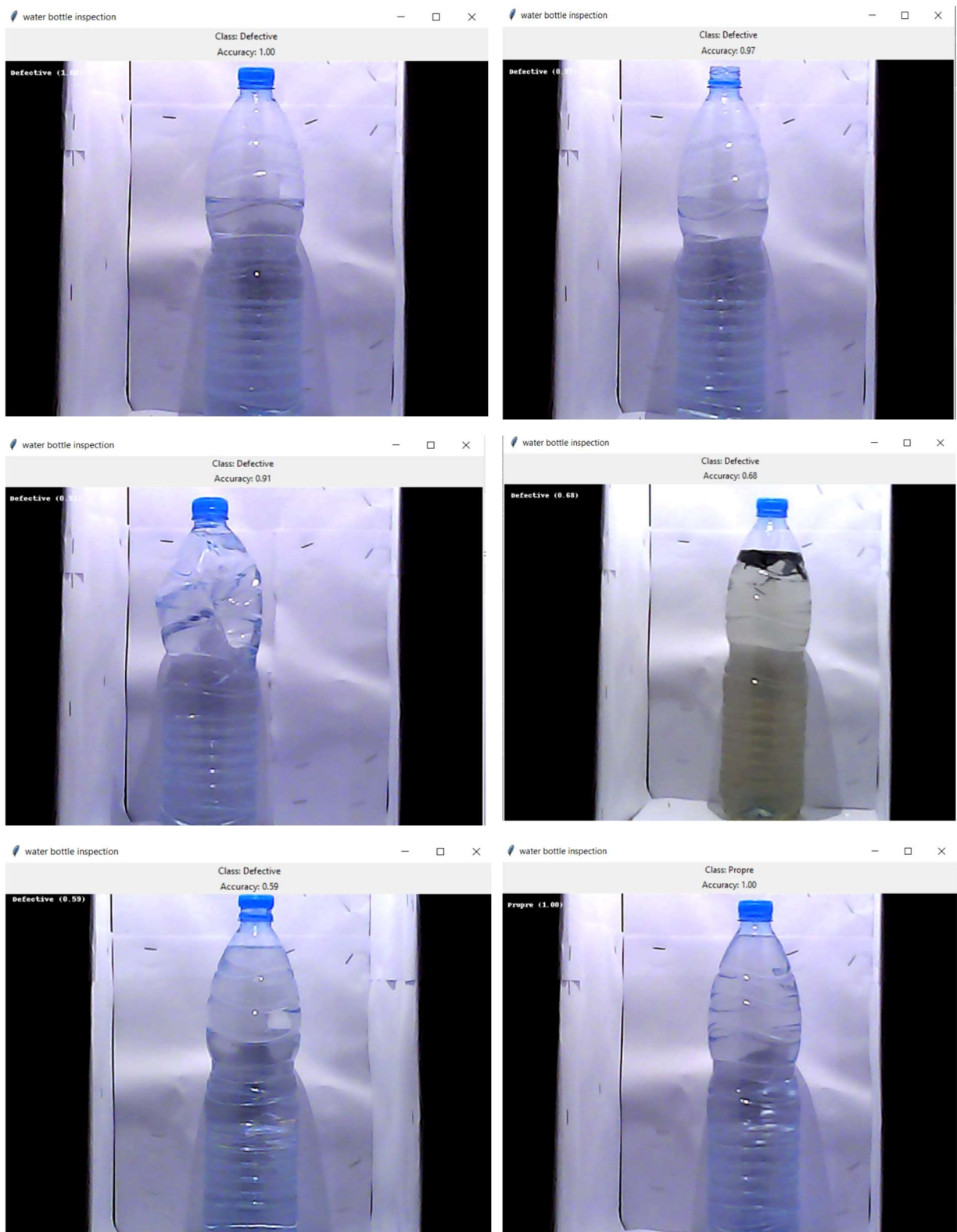


Figure IV.4 : Test de modèle **InsBot** en temps réels



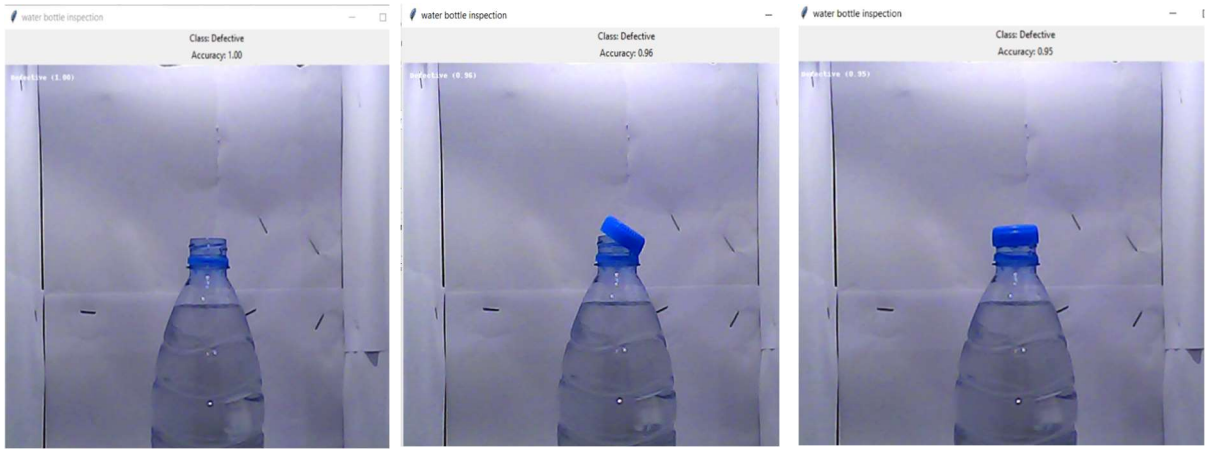


Figure IV.5 : Test de modèle **InsPart** en temps réels

- Étant donné que les deux réseaux sont basés sur le modèle Inception\_V3 et que les résultats de l'inférence en temps réel sont identiques, la seule différence réside dans la spécification d'inspection de chaque modèle (le modèle 1 inspecte la partie supérieure de la bouteille, tandis que l'autre inspecte l'ensemble de la bouteille de manière générale). Les deux modèles présentent de bonnes performances en termes de performances en temps réel, avec un suivi et un affichage fluide de l'inspection.

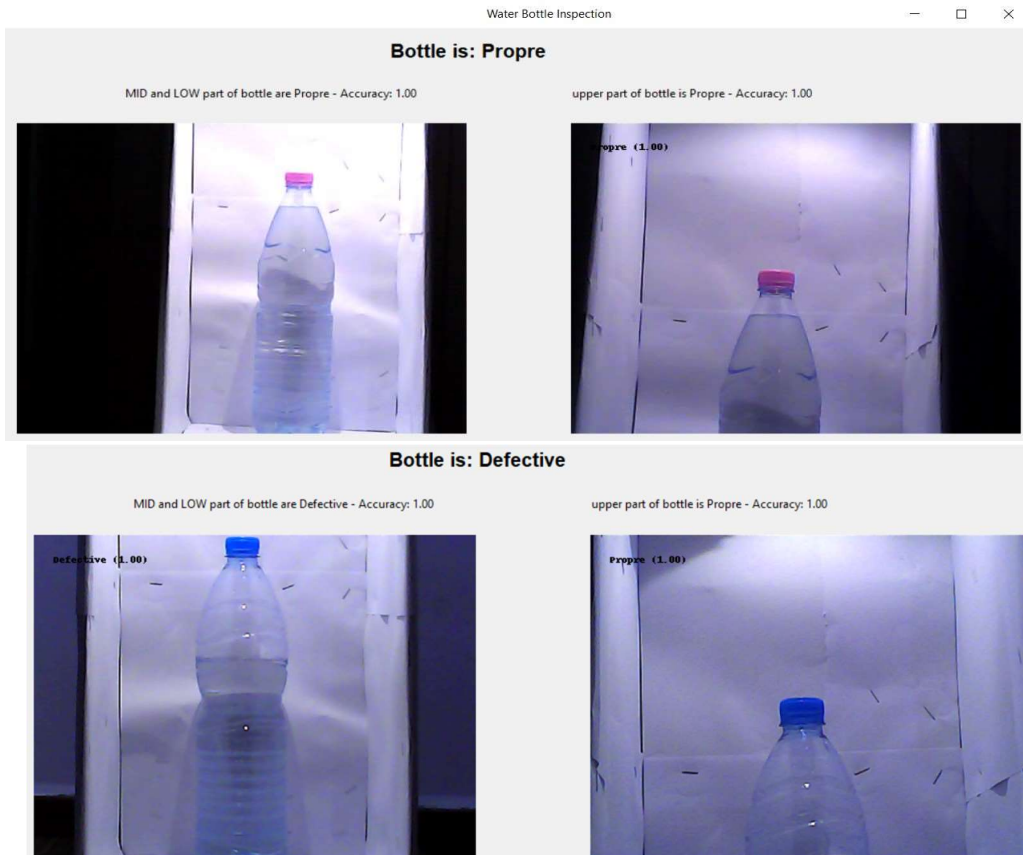


Figure IV.6 : Test des **InsComb** en parallèle en temps réels

- L'utilisation simultanée de ces deux modèles permet au système global de combiner les avantages spécifiques de chacun d'entre eux et d'améliorer la précision globale de la détection. Pendant l'exécution, les images en entrée sont soumises simultanément aux deux modèles, qui génèrent des prédictions en se basant sur leurs architectures respectives. En combinant les résultats des deux modèles, nous obtenons une détection plus robuste et précise, offrant ainsi une solution fiable pour les applications nécessitant une réponse en temps réel.

Tableau IV.2 : Mesure de performance en temps réel des deux modèles

Hardware	CPU i5	Nvidia RTX 3050	Jetson Nano
FPS/ un seul modèle	13	61	10
FPS / les deux modèles en parallèle	10	61	9

- L'analyse des résultats des deux modèles, telle qu'indiquée dans le Tableau 2, met en évidence l'impact du matériel utilisé sur les performances en temps réel du système. Le nombre de FPS (images par seconde) montrent des différences significatives en fonction du matériel. La NVIDIA RTX est supérieure en performance aux autres calculateurs grâce à son architecture GPU avancée.
- Les deux GPU (NVIDIA RTX 3050 et Jetson Nano) donnent des résultats similaires en exécutant un ou deux modèles en parallèle, tandis que le CPU (i5 6300U) présente une différence de performance. Cela est causé principalement par leurs capacités de traitement parallèle dédiées aux tâches d'inférence.
- Le kit Jetson Nano offre un équilibre entre performances et portabilité et cout, en faisant un bon choix pour les déploiements d'IA sur le terrain.

### IV.4 Automate Programmable Industriel API (PLC)

L'API utilisé dans ce projet est S7 1516-3 PN/DP. Le S7-1516-3PN/DP est une unité centrale de traitement (CPU) faisant partie du système d'automatisation SIMATIC S7-1500. Il s'agit d'un CPU standard intégré aux contrôleurs avancés de ce système. Il est équipé de deux interfaces Profinet et d'une interface Profibus et il dispose également d'une mémoire de travail de 1 Mo et d'une mémoire de chargement de 5 Mo.

L'API est utilisée pour le contrôle de l'ensemble du processus d'inspection. Il émet des commandes de démarrage et d'arrêt pour les différents composants de la chaîne d'inspection tels que le tapis roulant, le système de détection et la source de lumière. En fonction des prédictions fournies par le modèle **InsComb**, l'API gère la ligne d'inspection et active l'actionneur pour rediriger les bouteilles défectueuses vers un processus de vérification supplémentaire.

En outre, l'API S7-1516-3 PN/DP facilite également la communication et la coordination entre les autres éléments du système. Il assure la transmission de données entre le système de détection, l'interface homme-machine (HMI) et la base de données SQL. Cela permet aux

opérateurs de commander la ligne d'inspection à travers l'HMI KTP700 Basic, de superviser en temps réel grâce à l'HMI, et d'enregistrer les statistiques du processus d'inspection dans la base de données SQL.

### IV.4.1 Communication entre PLC et Système de détection de défauts

La communication entre le kit Jetson Nano et l'API Siemens S7-1516-3 PN/DP se fait grâce à la bibliothèque Snap7, qui utilise le protocole de communication S7. Ce protocole fonctionne sur la pile de protocoles TCP/IP, où TCP/IP gère la communication réseau de bas niveau, tandis que le protocole S7 gère les échanges de données spécifiques et les commandes liées à l'API Siemens S7.

La bibliothèque Snap7 est principalement conçue pour le langage de programmation Python. Elle offre une gamme de fonctionnalités permettant d'établir des connexions pour lire et écrire des données depuis et vers l'API, effectuer des conversions de données ...etc.

Snap7 offre la possibilité d'accéder à différentes zones de l'API : les entrées, les sorties, les zones de mémoire et les blocs de données. Elle gère différents types de données, tels que les booléens, les nombres à virgule flottante, les chaînes de caractères, et bien plus encore. La figure IV.7 illustre le principe de communication entre l'API et le Système de détection de défauts. La figure IV.8 représente le bloc de données (DB1) où les opérations d'écriture et de lecture nécessaires à la communication sont effectuées.

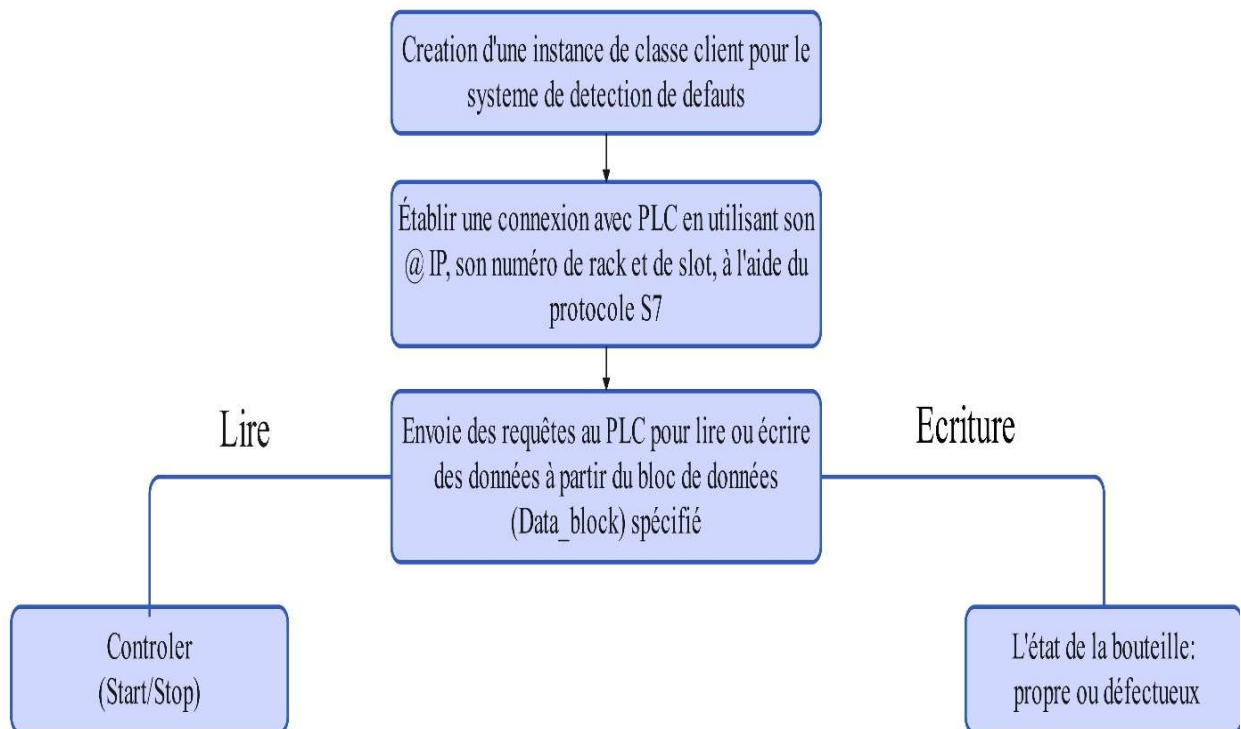


Figure IV.7 : Principe de communication entre l'API et Système de détection de défauts

Data_block_1												
	Name	Data type	Offset	Start value	...	Monitor value	...	...	...	Visi...	Sup...	Comment
1	▼ Static											
2	AI_P	Bool	0.0	false	...	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	bottle state is propre
3	AI_D	Bool	0.1	false	---	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	bottle state is defective
4	AI_state	Bool	0.2	false	...	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AI system state
5	AI_start	Bool	0.3	false	---	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Start command
6	AI_stop	Bool	0.4	false	---	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Stop command

Figure IV.8 : Bloc de données (DB1)

### IV.4.2 Programme de PLC

Le programme de l'API (PLC) est divisé en deux parties. La première partie concerne la gestion de la chaîne d'inspection et la détection des défauts, réalisée à travers une fonction (FC1). La seconde partie concerne la gestion du stock, où les valeurs sont stockées dans un bloc de données dédié (DB) à l'intérieur du bloc de fonction (FB). Cette approche permet d'assurer une certaine redondance et une reprise des données en cas de problème avec la base de données. Le Grafcet illustré dans la figure IV.9 montre les étapes du processus de la chaîne d'inspection. La désignation de différentes variables et leurs adresses ont dans le tableau IV.3. La Figure IV.10 illustre le bloc OB1 et la structure générale de programme.

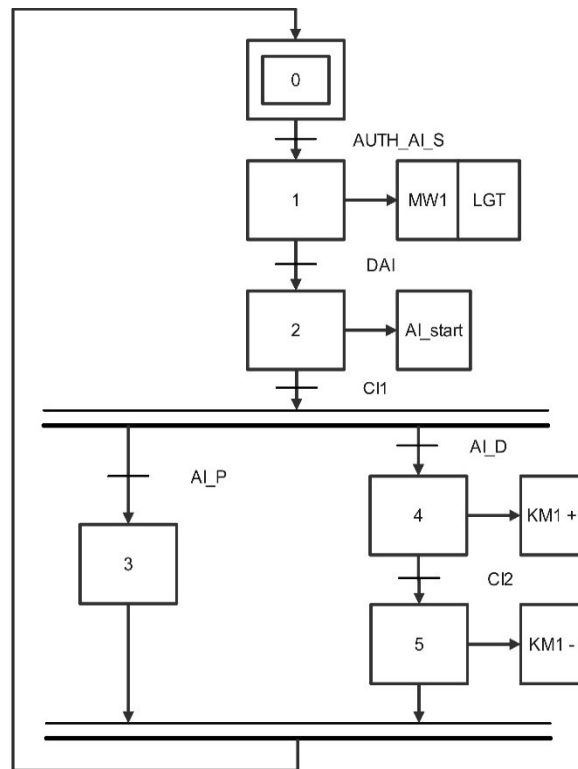


Figure IV.9 : Grafcet de la chaîne d'inspection d'un point de vue commande

Tableau IV.3 : Adresses des variables pour Grafcet

Variable	Adresse	Description
AUTH_AI_S	M1.7%	Autorisation de démarrage pour la chaîne d'inspection
DAI	M0.0%	Commande Start à partir de l'HMI
MW1	Q0.0%	Moteur de convoyeur de la chaîne d'inspection
AI_start	DB0.3%	Démarrage système de détection
LGT	Q0.2%	La lumière dans la chambre d'inspection
CI1	I0.2%	Capteur TOR dans la chambre d'inspection
AI_P	DB0.0%	Etat de bouteille Propre
AI_D	DB0.1%	Etat de bouteille Défectueuse
KM1+	Q0.2%	Contacte KM1+ de l'Actionneur de redirection excité
CI2	I0.4%	Capteur de présence à l'actionneur
KM1-	Q0.3%	Contact KM1- de l'Actionneur de redirection (état repos)

- **Processus de la ligne d'inspection**

Le processus de la ligne d'inspection débute par la nécessité d'une autorisation provenant de "AUTH\_AI\_S". Une fois que l'autorisation est obtenue, le moteur du tapis roulant, "MW1", est démarré et les lumières dans la salle d'inspection, appelées "LGT", sont allumées.

Pour lancer l'inspection, une commande de démarrage, "DAI", est émise. Cette commande active le drapeau d'inspection, "AI\_start", en le mettant à Vrai. Lorsque "AI\_start" est activé, cela déclenche le lancement du processus d'inspection et active la diffusion en direct des deux caméras dans l'interface GUI.

Pendant l'inspection, si le capteur "CI1" détecte la présence d'une bouteille, deux cas différents sont pris en compte en fonction des prédictions du système de détection.

Dans le premier cas, si le système de détection prédit que la bouteille est défectueuse "AI\_D", le séparateur du tapis roulant est déclenché (contacte "KM1+" de l'actionneur est activé). Cela modifie la direction de la bouteille, la détournant vers un autre tapis roulant pour un traitement ultérieur.

Après la redirection de la bouteille défectueuse, Si le signal du deuxième capteur "CI2" TOR indique un front descendant, cela signifie que la bouteille a franchi le séparateur. En réponse, le séparateur du tapis roulant est réinitialisé à sa position initiale (le contacte "KM1-" de l'actionneur est activé).

Dans le deuxième cas, si le système de détection, "AI\_P", prédit que la bouteille est en bon état, la bouteille poursuit son mouvement normal le long du tapis roulant sans aucune déviation.

**IV.4.3 Les Alarmes**

Les différentes alarmes utilisées dans ce projet sont définies dans tableau IV.4.

Tableau IV.4 : Les Alarmes utilisées dans L'API et IHM

Alarme	ID	Type d'alarme	Description
ASNW	1	Erreur	L'ordre de démarrage pour le système d'inspection a été envoyé mais il ne fonctionne pas
LA	2	Erreur	La lumière de la salle d'inspection est éteinte. Veuillez la vérifier.
NBI	3	Avertissement	Le système d'inspection est fonctionnel mais aucune bouteille n'a été détectée jusqu'à présent (5 minutes)
HDR	4	Avertissement	Taux de défaut élevé dans un lot (20 %)
HDRL	5	Avertissement	Taux de production défectueuse élevé

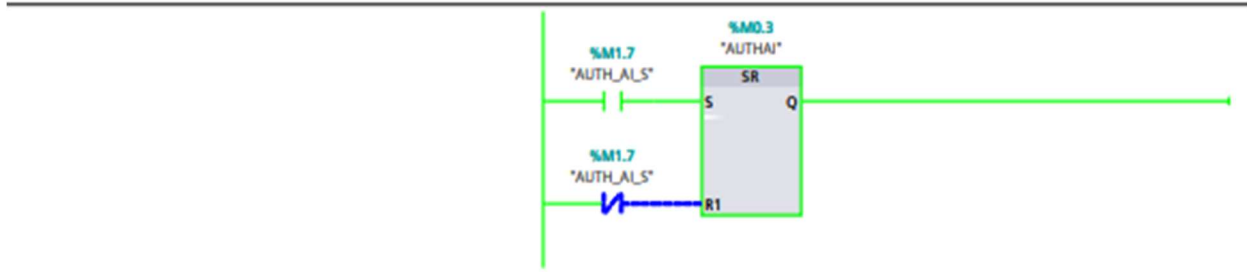
**IV.5 Interface Homme Machine (HMI)**

Pour l'interface utilisateur et la supervision, l'HMI KTP700 Basic PN a été utilisée. Elle est constituée de trois fenêtres en plus de la page d'accueil : Page d'accueil, Contrôle screen et Alarms screen. L'interface homme-machine (IHM) est sécurisée et n'est accessible qu'aux administrateurs autorisés. Grâce à une interface visuellement intuitive, elle représente efficacement l'état des bouteilles dans la page d'accueil.

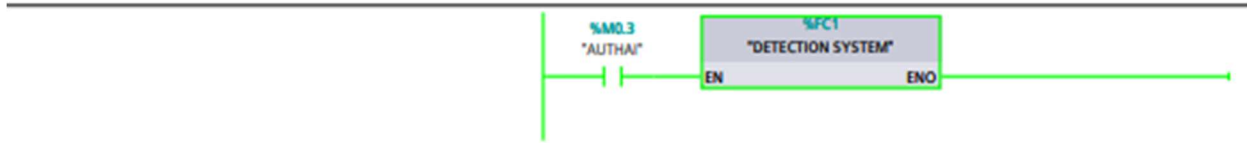
L'écran initial lors du démarrage de l'IHM est le Menu Principal, qui est sécurisé par un mot de passe. Seuls les administrateurs autorisés disposant du mot de passe correct peuvent accéder à cet écran. De plus, pour garantir la sécurité, si l'IHM reste inactive pendant 5 minutes, elle demandera automatiquement de saisir à nouveau le mot de passe avant toute utilisation ultérieure.

La Page d'accueil offre une représentation visuelle de l'état de l'inspection des bouteilles, indiquant s'il n'y a pas de bouteilles, des bouteilles correctes ou des bouteilles défectueuses. Elle sert d'écran de synthèse où les utilisateurs peuvent évaluer l'état actuel du processus d'inspection. L'accès à toutes les autres pages et fonctionnalités est disponible directement depuis ce Menu Principal, permettant aux utilisateurs de naviguer et de contrôler divers aspects du système.

Network 1: Authorization



Network 2: Detection system



Network 3: gestion de stock

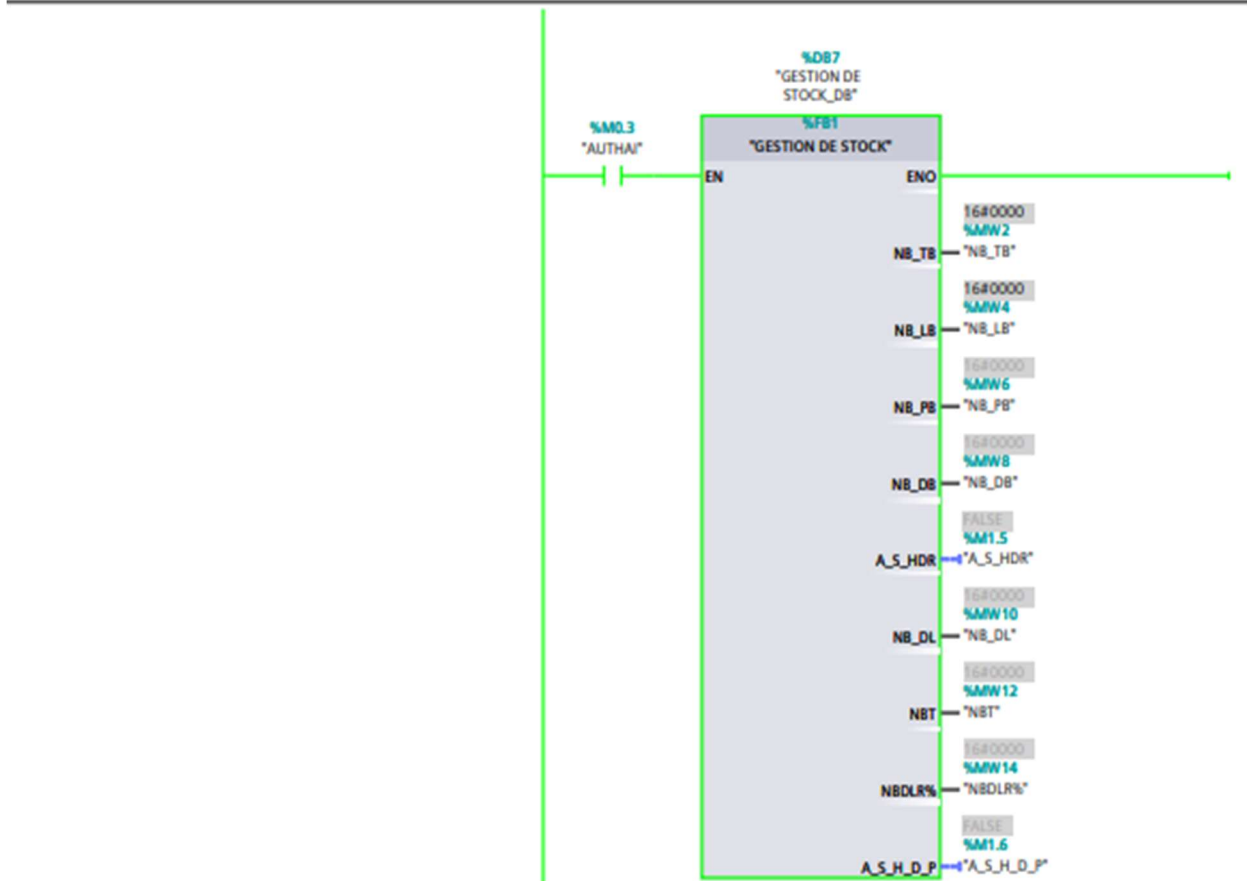
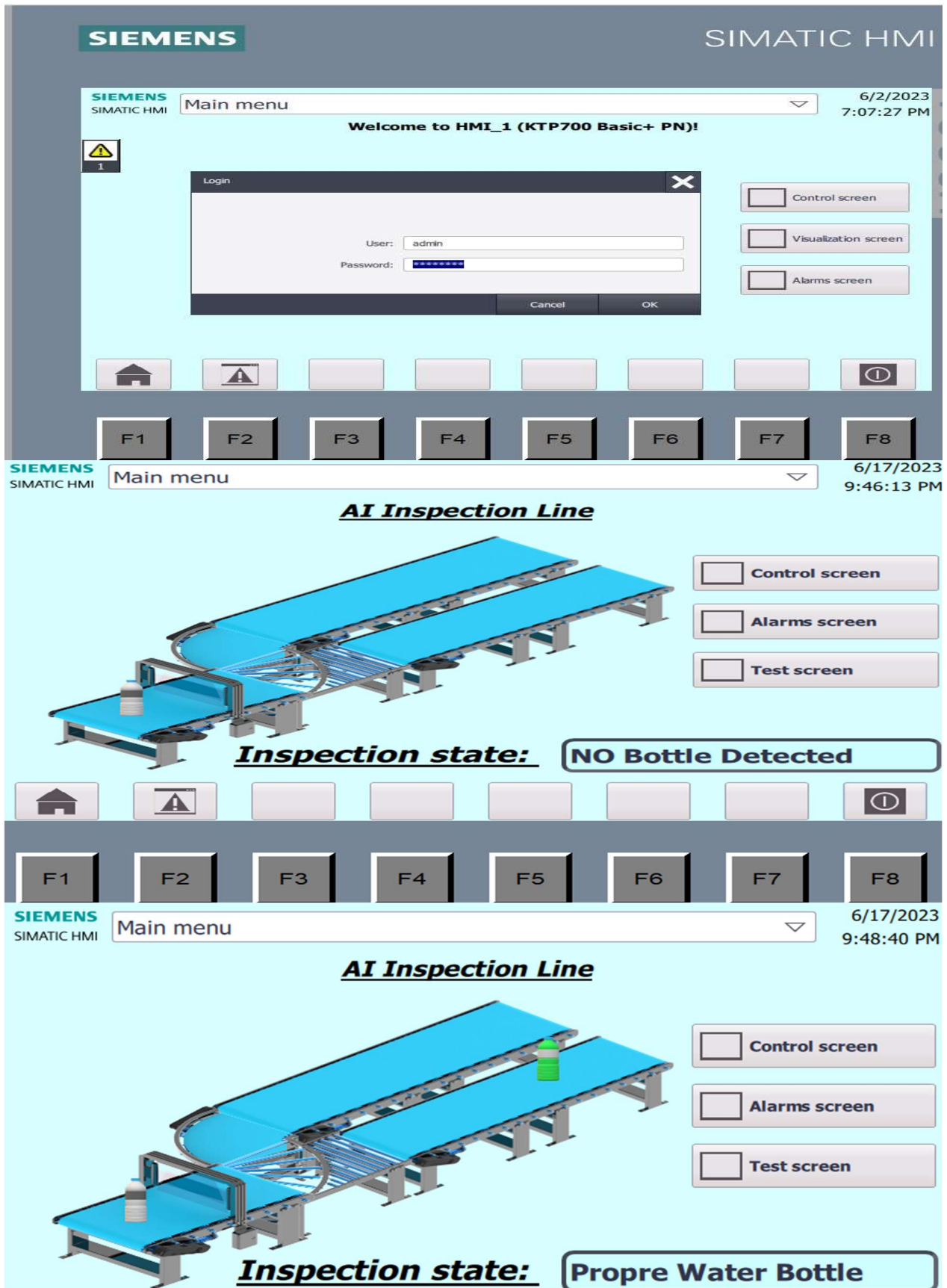


Figure IV.10 : Bloc OB1.





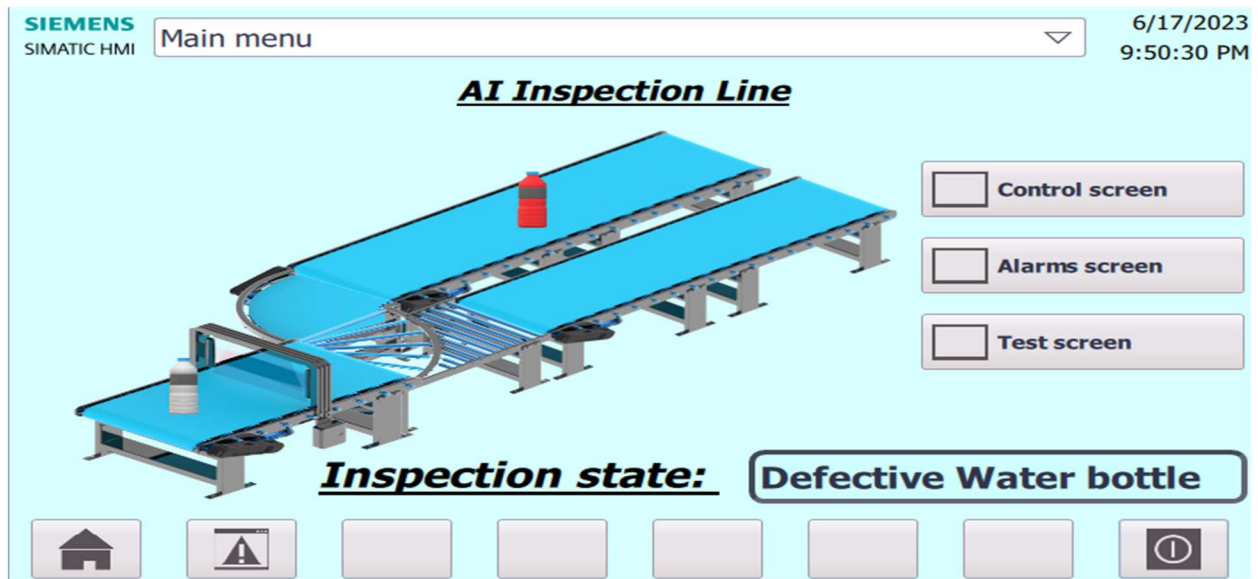


Figure IV.11 : Page d'accueil de l'IHM

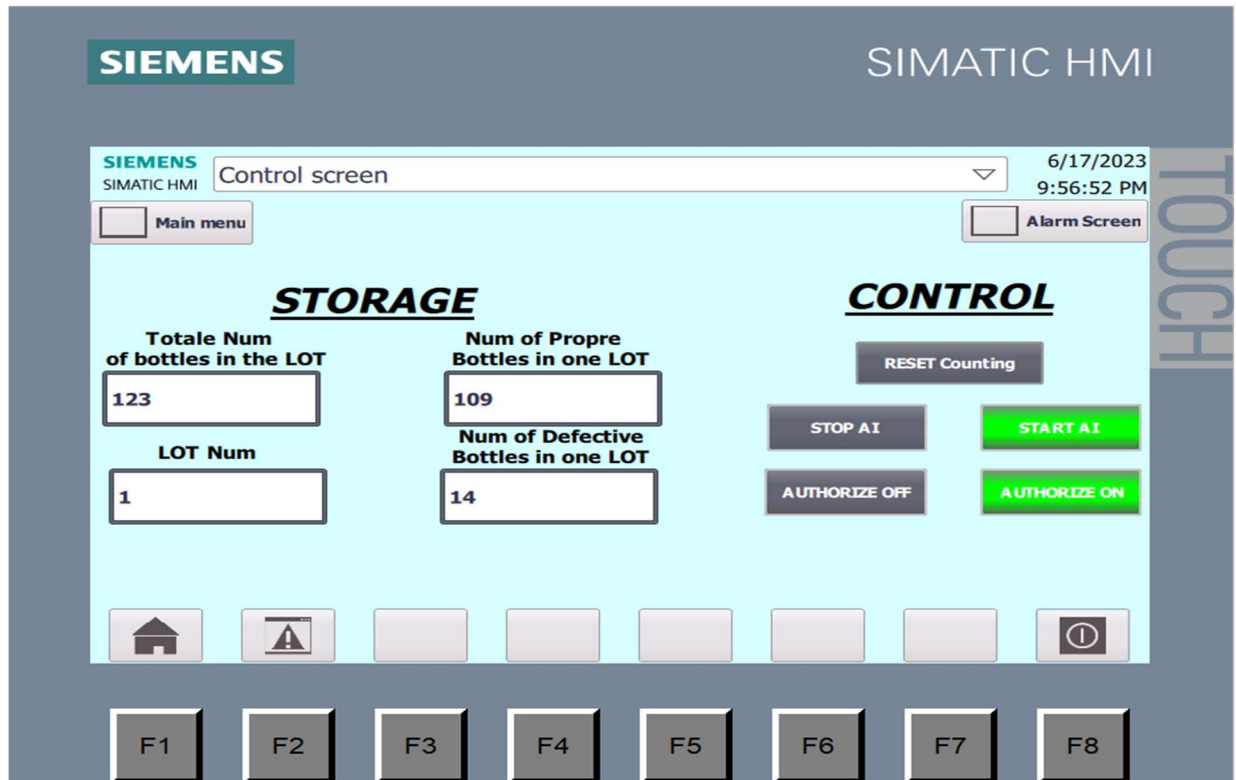


Figure IV.12 : Page de contrôle de l'IHM

L'écran de Contrôle est une page essentielle qui comprend des boutons de contrôle, fournissant une autorisation et initiant le processus d'inspection. De plus, cet écran affiche des informations liées à la gestion du stock, telles que le nombre de bouteilles correctes, et intègre des boutons de réinitialisation des compteurs et des paramètres pertinents. L'écran de Contrôle permet aux utilisateurs de gérer et de surveiller efficacement les opérations d'inspection.

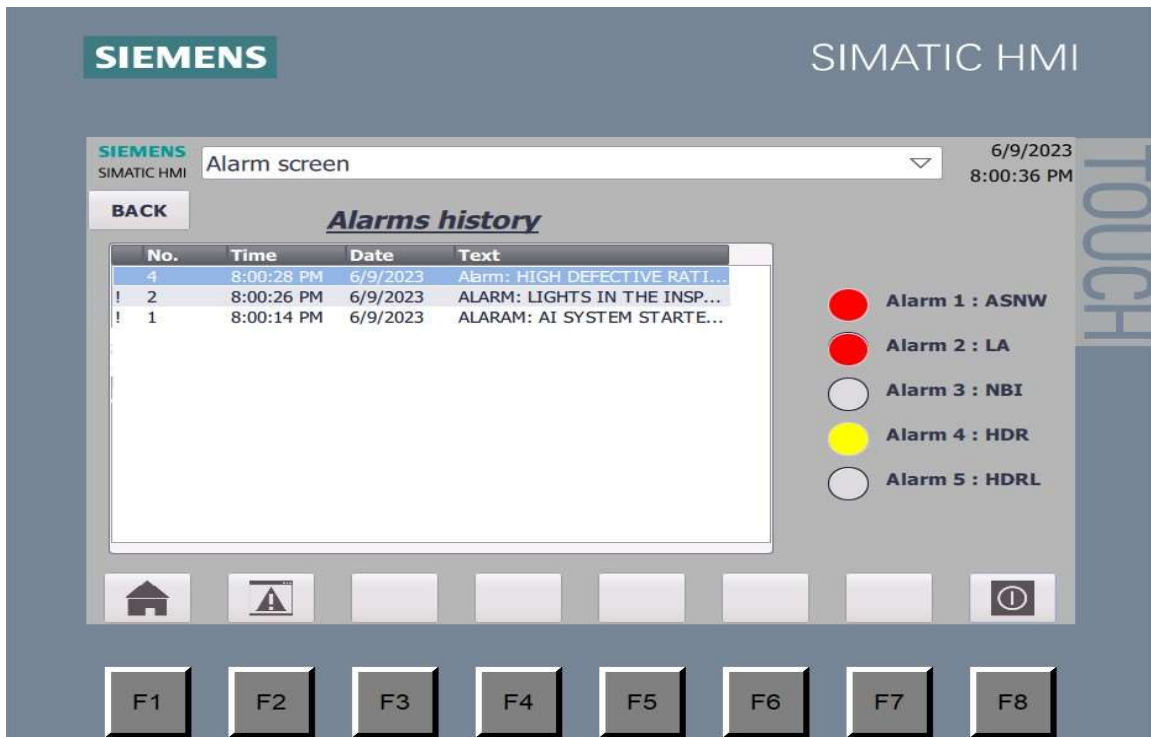


Figure IV.13 : Page des Alarmes de l'IHM

L'écran d'Alarme joue un rôle crucial dans le suivi et la gestion des alarmes. Il conserve un journal historique des alarmes passées et utilise des indicateurs visuels, tels que des voyants, pour afficher les alarmes actives en cours.

- Lorsqu'une nouvelle alarme se produit, indépendamment de la page en cours d'accès, un écran contextuel affiche immédiatement les détails de l'alarme ou des alarmes, garantissant une prise de conscience et une réponse rapides aux événements critiques.

## IV.6 Page Web et Base de Données MySQL

Pour la surveillance à distance, nous avons créé une simple interface WEB (figure IV.14) qui présente l'historique de l'inspection. La base de données MySQL (figure IV.15) a été créée pour sauvegarder les données et les statistiques. La mise à jour des données de la base de données se fait de manière continue à partir des données de l'API et du web (Dashboard).

La base de données MySQL pour l'inspection des bouteilles d'eau est organisée dans une seule table qui stocke des informations complètes. Elle inclut les identifiants des bouteilles pour identifier de manière unique chaque bouteille, les horodatages des inspections pour enregistrer quand chaque inspection a eu lieu, ainsi que les états indiquant si une bouteille est propre ou défectueuse. La table intègre également des données statistiques telles que le nombre total de bouteilles, le nombre de bouteilles propres, le nombre de bouteilles défectueuses, le ratio de bouteilles propres et le nombre total de lots produits. En regroupant toutes les informations dans une seule table, cela simplifie la récupération des données et fournit une vue d'ensemble complète du processus d'inspection.

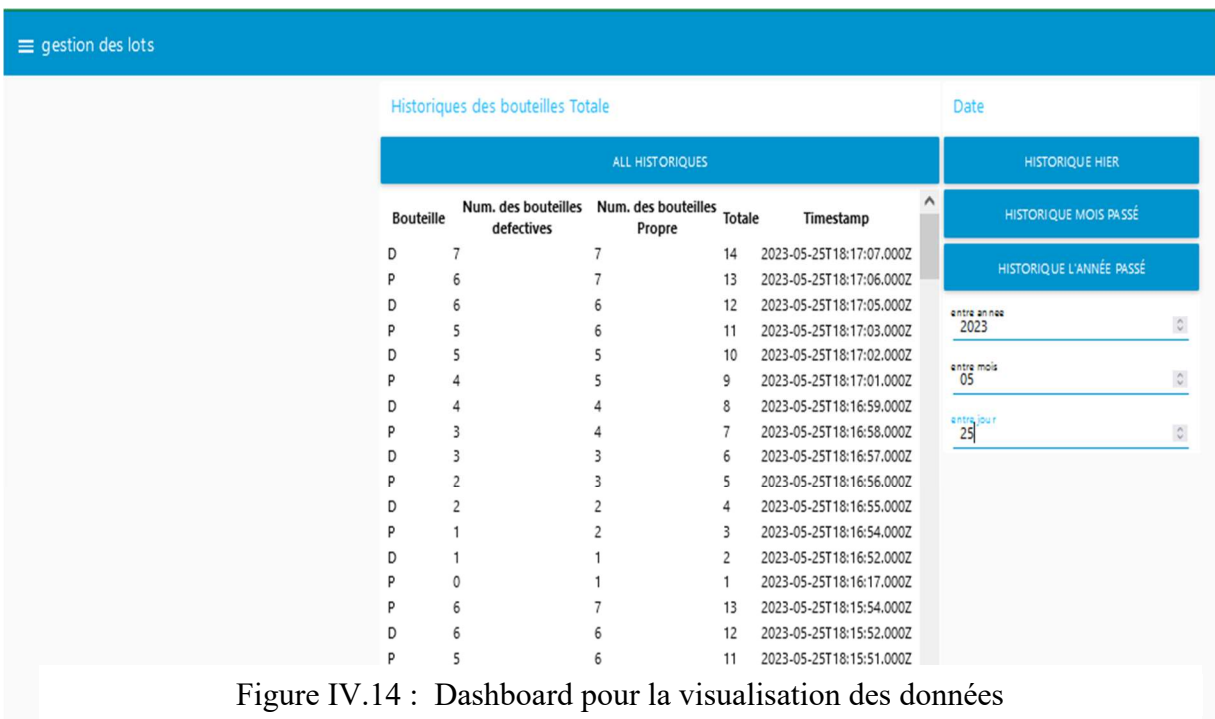
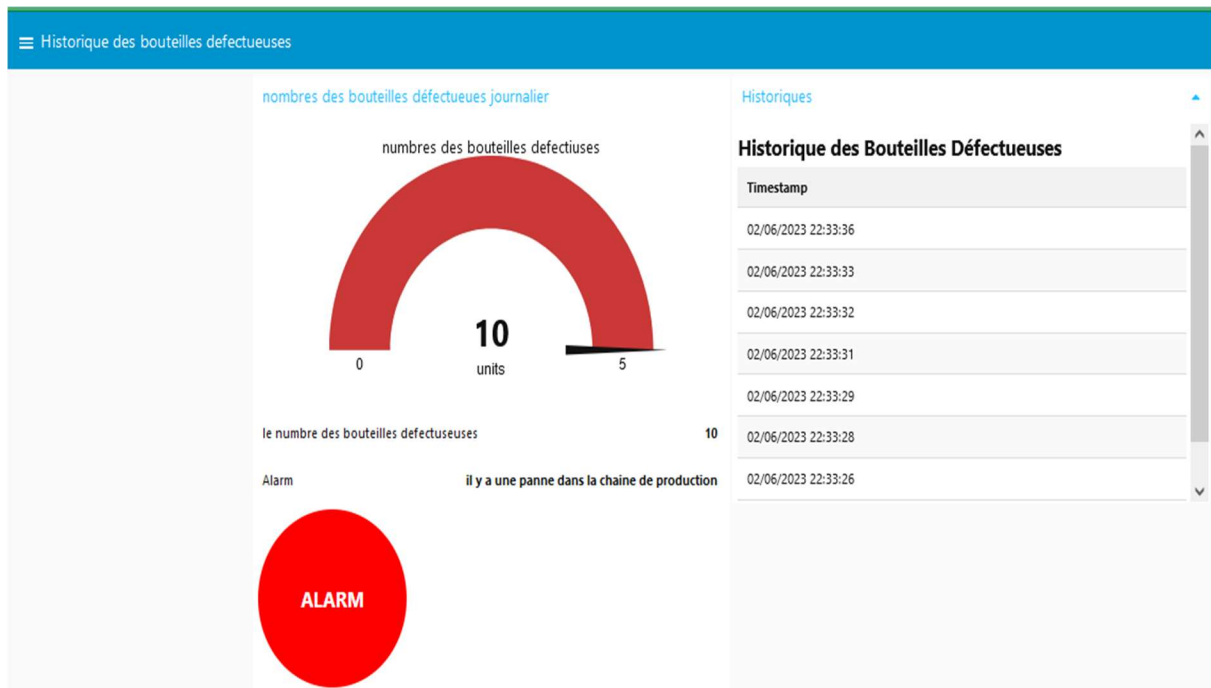


Figure IV.14 : Dashboard pour la visualisation des données

bouteille	num_des_bouteilles_defectives	num_des_bouteilles_Propre	Totale	Timestamp	id
D	5	3	8	2023-06-02 22:33:29	522
P	5	4	9	2023-06-02 22:33:30	523
D	6	4	10	2023-06-02 22:33:30	524
P	6	5	11	2023-06-02 22:33:31	525
D	7	5	12	2023-06-02 22:33:32	526
P	7	6	13	2023-06-02 22:33:32	527
D	8	6	14	2023-06-02 22:33:33	528
D	9	6	15	2023-06-02 22:33:35	529
P	9	7	16	2023-06-02 22:33:36	530
D	10	7	17	2023-06-02 22:33:36	531
P	10	8	18	2023-06-02 22:33:37	532
NULL	NULL	NULL	NULL	NULL	NULL

Figure IV.15 : La base de données MySQL

## IV.7 Conclusion

La mise en œuvre matérielle de notre système globale d'inspection de la qualité des bouteilles d'eau a été présentée en détail dans ce chapitre. Elle constitue un fondement essentiel pour la solution globale. L'évaluation des aspects en temps réel, tels que la rapidité et le temps de réponse, ainsi que l'interconnexion des différents composants, créent un système complet qui offre de bonnes performances d'un point de vue précision, efficacité et redondance, garantissant ainsi la qualité constante des bouteilles d'eau inspectées.

Tout au long de notre analyse, nous avons constaté l'impact significatif du matériel sur les performances du système. Notamment, l'analyse des résultats des deux modèles Inception\_v3 met en évidence l'influence du matériel sur l'efficacité en temps réel. Les mesures de FPS révèlent des variations de performance en fonction du matériel choisi. La NVIDIA RTX, avec son architecture GPU avancée, surpasse les autres options, ce qui en fait un choix privilégié pour maximiser les capacités de traitement en temps réel. Cependant, il est essentiel de reconnaître que l'aspect matériel n'est qu'un élément de la solution complète. L'intégration des cinq composants clés - le système de détection spécialisé, le PLC pour le contrôle, l'interface homme-machine (HMI) pour l'interface utilisateur et la supervision, la page web pour la surveillance à distance et la base de données SQL pour le stockage des données statistiques - est essentielle pour garantir la robustesse, la fiabilité et le bon fonctionnement du système.

En tenant compte de tous ces facteurs, la Jetson Nano émerge comme un choix attrayant en raison de ses performances équilibrées, de sa portabilité et de son coût, ce qui en fait un choix adapté aux déploiements sur le terrain avec des ressources limitées pour les systèmes d'IA.

# Conclusion Générale

Dans ce projet, nous avons développé un système d'inspection visuelle de la qualité des bouteilles d'eau. Ce système est prêt pour être intégré dans une chaîne de production et à être exploiter pour la détection des bouteilles défectueuses, la supervision de l'opération sur site et à partir d'un poste de supervision distant à travers une page web avec enregistrement de tout l'historique dans une base de données. Il permet également le contrôler des composants de la chaîne de production surtout en cas de détection d'une bouteille défectueuse et qui doit être transférer vers une zone particulière.

La détection de défaut se fait par un modèle constitué de deux modèles neuronales élémentaires. Ces derniers ont été développés à partir d'Inception V3 par transfert d'apprentissage. Au départ, nous avons entraîné et évalué plusieurs modèles. L'affinement des hyperparamètres de ces réseaux a été effectué en utilisant notre propre dataset pour rendre les modèles plus sensibles aux défauts rencontrés dans l'industrie de remplissage des bouteilles d'eau. Les performances de ces modèles a été effectué en considérant plusieurs métriques et celui qui a permis d'obtenir de meilleurs résultats a été choisi. Cependant, nous avons remarqué que l'utilisation d'un seul modèle avec une seule caméra présente des limites surtout pour détecter les défauts de la partie haute des bouteilles comme les défauts dans les bouchons. Cela est dû en particulier à la petite région d'intérêt dans la partie supérieure d'une bouteille et la petite taille des images utilisées pendant l'inférence et l'apprentissage. Pour remédier à cela, nous avons décidé d'inspecter séparément la partie supérieure et de combiner les résultats d'inspection des deux réseaux. En incorporant les prédictions des deux modèles, nous avons obtenu de meilleures performances. La précision dans ce cas a été de 94,5 % sur 800 images par classe, soit un total de 1600 images.

En regardant vers l'avenir, il existe plusieurs pistes de recherche dans le domaine de l'inspection des bouteilles d'eau utilisant l'apprentissage profond. Par exemple, l'incorporation de défauts plus complexes comme la considération des défauts dans le texte de l'emballage, des codes-barres ou même la détection des plus petites impuretés dans l'eau pourrait améliorer davantage les capacités d'inspection. De plus, l'accent devrait être mis sur le maintien de vitesses de production élevées sans compromettre l'efficacité de l'inspection, ce qui permettrait une production à grande échelle avec un minimum de défauts.

En conclusion, notre projet a démontré l'efficacité des techniques d'apprentissage profond dans le développement d'un système fiable d'inspection des bouteilles d'eau. La combinaison de l'apprentissage par transfert, de l'inspection par caméras distinctes et de la prise en compte minutieuse de l'architecture matérielle nous a permis d'aboutir à une solution puissante. Notre travail améliore non seulement les processus de contrôle de la qualité, mais ouvre également la voie à de nouvelles avancées dans le domaine de l'apprentissage profond pour les applications industrielles.

## Bibliographie

- [1] "Non-destructive Testing Techniques for Quality Evaluation of Industrial Products" by A. Bala, N. Sharma, and R. Mishra (2019)
- [2] "Defect Detection Techniques for Industrial Quality Control" by A. Gharbi and L. Romdhane (2020)
- [3] Mook, Gerhard, Jouri Simonin, Johann H. Hinken, Herbert Wrobel, Wolf Dieter Feist, and Geraldine Perrin. "Detection and Characterization of Magnetic Anomalies in Gas Turbine Disks." *NDT & E International* 49.1 (2016): 13-21. doi:10.1016/j.ndteint.2015.10.002
- [4] "A Review of Defect Detection Systems in Manufacturing" by R. Singh and R. Roy (2018)
- [5] Al-Ghamdi, M. A., Al-Dabbagh, M. A., & Al-Othman, A. A. (2008). X-ray detection of defects and contaminants in the food industry. *Food Control*, 20(1), 12-18. <https://doi.org/10.1016/j.foodcont.2007.09.007>
- [6] Cognex. (2023, March 8). Foreign object inspection (FOI) solutions for the food and beverage industry. Retrieved from <https://www.cognex.com/industries/food-and-beverage/product-quality-inspection/foreign-object-inspection>
- [7] "Production de récipients en verre à usage pharmaceutique contrôlée par système de vision." *IndustrieWeb*, [www.industrieweb.fr/news/1728-production-de-r%C3%A9cipients-en-verre-%C3%A0-usage-pharmaceutique-contr%C3%B4l%C3%A9-par-syst%C3%A8me-de-vision](http://www.industrieweb.fr/news/1728-production-de-r%C3%A9cipients-en-verre-%C3%A0-usage-pharmaceutique-contr%C3%B4l%C3%A9-par-syst%C3%A8me-de-vision). Accessed 28 May 2023.
- [8] Creaform, [www.creaform3d.com/fr/solutions/controle-de-la-qualite-automatise-une-strategie-cle-pour-ameliorer-le-rendement-des](http://www.creaform3d.com/fr/solutions/controle-de-la-qualite-automatise-une-strategie-cle-pour-ameliorer-le-rendement-des). Accessed 28 May 2023.
- [9] Zhang, X., Zhang, Y., & Zhang, J. (2019). Visual analytics for quality control in the automotive industry. *Visualization and Computer Graphics, IEEE Transactions on*, 25(1), 103-114., doi:10.1109/TVCG.2018.2873709.
- [10] Alibaba, [french.alibaba.com/product-detail/Textile-washing-resistance-color-fastness-testing-60781707773.html](http://french.alibaba.com/product-detail/Textile-washing-resistance-color-fastness-testing-60781707773.html). Accessed 28 May 2023.
- [11] Adibhatla, V. A., Chih, H.-C., Hsu, C.-C., Cheng, J., Abbod, M. F., & Shieh, J.-S. (2022). Defect detection in printed circuit boards using YOLOv5 deep learning model. 2022 11th International Conference on Information Technology: New Generations (ITNG), 1068-1073. doi:10.1109/ITNG54364.2022.9752281
- [12] Alamy. "Quality control in microchip factory." Alamy, [www.alamy.com/quality-control-in-microchip-factory-image267280691.html](http://www.alamy.com/quality-control-in-microchip-factory-image267280691.html). Accessed 28 May 2023.
- [13] Intel. "Your Industrial Physical Environment Guide." Intel, <https://www.intel.com/content/dam/www/central-libraries/us/en/documents/your-industrial-physical-environment-guide.pdf> (accessed May 14, 2023).
- [14] Teledyne DALSA. "Web Inspection | Teledyne DALSA." Teledyne DALSA, <https://www.teledynedalsa.com/en/learn/markets-and-applications/mv/web-inspection/> (accessed May 14, 2023).
- [15] N. Instruments, A practical guide to machine vision lighting, 2017, [Online; accessed 9-October-2017]. [Online]. Available: <http://www.ni.com/white-paper/6901/en/>
- [16] M. Bigas, E. Cabruja, J. Forest, and J. Salvi, Review of cmos image sensors, *Microelectronics Journal*, vol. 37, no. 5, pp. 433-451, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0026269205002764>
- [17] K. G. Harding, Machine vision lighting. in *Encyclopedia of Optical Engineering*. CRC Press, 2003, pp.12271336
- [18] "High-speed vision checks food cans concisely." *Vision Systems Design*, [www.vision-systems.com/cameras-accessories/article/16737568/highspeed-vision-checks-food-cans-concisely](http://www.vision-systems.com/cameras-accessories/article/16737568/highspeed-vision-checks-food-cans-concisely). Accessed 28 May 2023.
- [19] C. Tikhe, J.S. Chitode, Metal surface inspection for defect detection and classification using Gabor Filter, *Int. J. Innov. Res.Sci. Eng. Technol.* 3 (6) (2014) pp

## Bibliographie

---

- [20] Devi, T. Gayathri, P. Neelamegam, and S. Sudha. (2017) "Machine vision based quality analysis of rice grains." 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), IEEE
- [21] M. Miura, S. Sakai, S. Aoyama, J. Ishii, K. Ito, T. Aoki, High-accuracy image matching using phaseonly correlation and its application, in: SICE Annual Conference, 2012, pp. 307–312
- [22] Wu, G., Zhang, H., Sun, X., Xu, J., & Xu, K. (2007). A Bran-new Feature Extraction Method and its application to Surface Defect Recognition of Hot Rolled Strips. 2007 IEEE International Conference on Automation
- [23] Li, Z., Li, J., Wang, Y., & Wang, K. (2019). A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment. *Mechanical Systems and Signal Processing*, 131, 106402. doi:10.1016/j.ymsp.2019.106402
- [24] Imoto, K., Nakai, T., Ike, T., Haruki, K., & Sato, Y. (2018). A CNN-based transfer learning method for defect classification in semiconductor manufacturing. In 2018 international symposium on semiconductor manufacturing (ISSM) (pp. 1–3). IEEE
- [25] Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. (2017, March). Retrieved May 14, 2023, from <https://arxiv.org/abs/1703.05921>
- [26] Nanduri, A., & Sherry, L. (2016). Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). 2016 13th International Conference on Information Systems for Aviation (ISA), 1-6. doi:10.1109/ISA.2016.7486356
- [27] Bengio, Y., Courville, A., & Vincent, P. (2013). Deep learning. MIT press. pp. 175-176.
- [28] Kitov, AI. "CAD2SCAN." Kitov.AI, [www.kitov.ai/products/cad2scan/](http://www.kitov.ai/products/cad2scan/).
- [29] Inspekto. "Inspekto S70 Industrial." Inspekto, <https://inspekto.com/inspekto-s70-industrial/>.
- [30] Cognex Corporation. "In-Sight 2800 Vision System." Cognex Corporation, <https://www.cognex.com/products/machine-vision/2d-machine-vision-systems/in-sight-2800>.
- [31] Zhou, W. e. (2013, August 29). A sparse representation based fast detection method for surface defect detection of bottle caps. *Neurocomputing*
- [32] Yazdi, L. e. (2011). Feature Extraction Algorithm for Fill Level and Cap Inspection in Bottling Machine. International Conference on Pattern Analysis and Intelligent Robotics. Malasia
- [33] TECHNOLOGY, P. (n.d.). Empty bottle inspection (PET). (PRESSCO TECHNOLOGY) Retrieved May 2, 2020, from <https://www.pressco.com/applications/plastic-2/petbottles/empty-bottle-inspection-pet/>
- [34] A. M. TURING, I.—COMPUTING MACHINERY AND INTELLIGENCE, *Mind*, Volume LIX, Issue 236, October 1950, Pages 433–460, <https://doi.org/10.1093/mind/LIX.236.433>
- [35] Ivakhnenko, A. G. (Aleksēi Grigor'evich). *Cybernetics and Forecasting Techniques*. New York :American Elsevier Pub. Co., 1967.
- [36] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel; Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput* 1989; 1 (4): 541–551. doi: <https://doi.org/10.1162/neco.1989.1.4.541>
- [37] Fukushima, Kunihiko. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." *Biological Cybernetics* 36 (1980): 193-202.
- [38] Hempel, Jessi. "*Fei-Fei Li's Quest to Make Machines Better for Humanity*". *Wired*. ISSN 1059-1028. Retrieved January 10, 2023.
- [39] Alom, Md. Zahangir et al. "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches." *ArXiv abs/1803.01164* (2018)

## Bibliographie

---

- [40] “Technologies.” ACIC - Advanced Cytometry Instrumentation & Consulting GmbH, <https://www.acic.eu/technologies>. Accessed 13 May 2023.
- [41] Gayakwad, Ekansh & Jayagopal, Prabhu & Anand, R. & Kumar, M.Sandeep. (2021). Training Time Reduction in Transfer Learning for a Similar Dataset Using Deep Learning. 10.1007/978-981-15-5679-1\_33.
- [42] **Kveim, Kjetil**. Plastic Bottle Cap Detection - Utilizing Artificial Intelligence. University of Oslo, 2019. <http://urn.nb.no/URN:NBN:no-72945>.
- [43] DeepMind. “AlphaGo.” DeepMind, <https://www.deepmind.com/research/highlighted-research/alphago>. Accessed 17 May 2023.
- [44] Zhuang, Fuzhen, et al. “A Comprehensive Survey on Transfer Learning.” arXiv, arXiv.org, 23 Jun 2020, <https://arxiv.org/abs/1911.02685v3>.
- [45] Gizaw, Asaminew. “AN AUTOMATIC WATER BOTTLE DEFECT DETECTION SYSTEM USING DEEP LEARNING APPROACH.” Addis Ababa University, 2020. <http://hdl.handle.net/123456789/11702>.
- [46] Truong, Phuoc. (2016). Optimisation des performances de la machine synchrone à réluctance variable : approches par la conception et par la commande.
- [47] Sharma, Pulkit. “Fundamentals of Deep Learning – Activation Functions and When to Use Them?” Analytics Vidhya, 6 Jan. 2020, <https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/>. Accessed 13 May 2023.
- [48] Bengio, Yoshua, Ian Goodfellow, and Aaron Courville. Deep Learning. MIT Press, 2016. <https://www.deeplearningbook.org/>.
- [49] Maria Vakalopoulou, Stergios Christodoulidis, Ninon Burgos, Olivier Colliot, Vincent Lepetit. Deep learning: basics and convolutional neural networks (CNN). Olivier Colliot. Machine Learning for Brain Disorders, Springer, inPress. fihal-03957224f
- [50] Sutskever, Ilya, et al. "On the importance of initialization and momentum in deep learning." International conference on machine learning. 2013.
- [51] Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. Neural networks, 1(4), 295–307. 265
- [52] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
- [53] Moradi, Reza, et al. “A survey of regularization strategies for deep models.” Neural Computing and Applications, vol. 31, no. 10, 2019, pp. 6469-6494.
- [54] Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, PMLR, pp 448–456
- [55] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." Journal of Machine Learning Research, vol. 15, 2014, pp. 1929-1958.
- [56] “Dropout Layer in CNN - Dot Net Tutorials.” Dot Net Tutorials, 2020, <https://dotnettutorials.net/lesson/dropout-layer-in-cnn/>. Accessed 15 May 2023.
- [57] Kumar, Sourav. "Data Augmentation Increases Accuracy of Your Model, But How?" Medium, 15 July 2020, <https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722>. Accessed 1 June 2023.
- [58] GOODFELLOW, Ian, POUGET-ABADIE, Jean, MIRZA, Mehdi, *et al.* Generative adversarial networks. *Communications of the ACM*, 2020, vol. 63, no 11, p. 139-144.
- [59] GAN Deep Learning: A Comprehensive Guide.” DataGen, 2021, <https://datagen.tech/guides/computer-vision/gan-deep-learning/>. Accessed 15 May 2023.



## Bibliographie

---

- [60] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
- [61] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.
- [62] Iandola, Forrest N., et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size." *arXiv preprint arXiv:1602.07360*, 2016, <https://arxiv.org/abs/1602.07360>.
- [63] Liu, Ze, et al. "Swin Transformer V2: Scaling Up Capacity and Resolution." arXiv preprint arXiv:2111.09883, 2021, <https://arxiv.org/abs/2111.09883>.
- [64] Tsang, Sik-Ho. "Review: Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015." Medium, 9 Feb. 2019, <https://medium.com/@smallfishbigsea/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c>. Accessed 21 May 2023.
- [65] Nguyen, Thi Hai Binh, et al. "fPADnet: Small and Efficient Convolutional Neural Network for Presentation Attack Detection." *Sensors*, vol. 18, no. 8, 2018, p. 2532, doi:10.3390/s18082532.
- [66] Rana, Ashutosh, et al. "Image Augmentation for Deep Learning: A Survey." *Journal of Big Data*, vol. 6, no. 1, 2019, doi:10.1186/s40537-019-0197-0.
- [67] Zhang, Jie, et al. "Image Augmentation for Medical Image Analysis: A Survey." *Journal of Zhejiang University-Science*, vol. 15, no. 1, 2014, pp. 1-17. doi:10.1631/jzus.C1400102.
- [68] PyTorch Team. (n.d.). PyTorch. Retrieved May 15, 2023, from <https://pytorch.org/>
- [69] "ONNX | Home." ONNX, [onnx.ai/](https://onnx.ai/). Accessed 14 May 2023.
- [70] NVIDIA. "TensorRT." NVIDIA Developer website, <https://developer.nvidia.com/tensorrt>, accessed March 8, 2023
- [71] "NVIDIA GeForce RTX 3050 Graphics Card." NVIDIA, <https://www.nvidia.com/en-me/geforce/graphics-cards/30-series/rtx-3050/>. Accessed 17 June 2023.
- [72] "Jetson Nano Developer Kit." NVIDIA Developer, <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>. Accessed 17 June 2023.

لضمان سلامة المنتج والحفاظ على سمعة منتجي مياه الشرب المعبأة، يعتبر التفقيش البصري لِقَوَارِيزِ المياه أمرًا حاسمًا. يركز مشروعنا على تطوير نظام قوي للتفقيش البصري في إنتاج زجاجات المياه باستخدام مختلف تقنيات التعلم العميق. لقد استخدمنا تقنية نقل المعرفة على 'inception\_v3'، مستخدمين كاميرتين لتفقيش مناطق مختلفة، حيث يتم توصيل كل كاميرا بشبكته المخصصة. من خلال تقييم واختبارات مكثفة، توصلنا إلى نتيجة جيدة ودقيقة تبلغ 94.5%. تسلط هذه النتائج الضوء على فعالية نظامنا في تعزيز عمليات التفقيش البصري في صناعة تصنيغ لِقَوَارِيزِ المياه، مما يسهم في تحسين جودة المنتج ورضا المستهلك.

كلمات مفتاحية: التفقيش البصري، مراقبة الجودة، التعلم العميق، تقنية نقل المعرفة، شبكة الأعصاب التكرارية المغطاة (CNN)، 'inception\_v3'، وحدة تحكم منطقية قابلة للبرمجة «PLC».

## Résumé

Afin de garantir la sécurité des produits et préserver la réputation des producteurs d'eau en bouteille, l'inspection visuelle des bouteilles d'eau est cruciale. Notre projet se concentre sur le développement d'un système robuste d'inspection visuelle dans la production de bouteilles d'eau en utilisant différentes techniques d'apprentissage profond. Nous avons utilisé l'apprentissage par transfert avec des modèles Inception, en utilisant deux caméras pour inspecter des zones distinctes, chacune étant connectée à son réseau dédié. Grâce à des tests et à une évaluation approfondie, notre approche atteint un taux de précision impressionnant de 94,5 %. Ces résultats mettent en évidence l'efficacité de notre système dans l'amélioration des processus d'inspection visuelle au sein de l'industrie de la fabrication de bouteilles d'eau, contribuant ainsi à une meilleure qualité de produit et à la satisfaction des consommateurs.

Mots-clés : Inspection visuelle, Contrôle qualité, Apprentissage profond, CNN, Transfert d'apprentissage, Inception\_v3, API.

## Abstract

To ensure product safety and uphold the reputation of bottled water producers, visual inspection of water bottles is crucial. Our project focuses on developing a robust system for visual inspection in water bottle production using different deep learning techniques. We employed transfer learning with Inception models, utilizing two cameras to inspect distinct areas, each connected to its dedicated network. Through extensive testing and evaluation, our approach achieves an impressive accuracy rate of 94.5%. These results highlight the effectiveness of our system in enhancing visual inspection processes within the water bottle manufacturing industry, ultimately contributing to improved product quality and consumer satisfaction.

Keywords: Visual inspection, Quality control, Deep Learning, CNN, Transfer Learning, Inception\_v3, PLC.