

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Akli Mohand Oulhadj –Bouira-



Faculté : Sciences et Sciences Appliquées

Département : Génie Electrique

Domaine : Sciences et Technologie

Filière : *Electronique*

Spécialité : *Electronique des Systèmes Embarqués*

Mémoire de fin d'études
Pour l'obtention du diplôme de MASTER

Thème

Développement d'un système intelligent pour vérification de l'état des produits dans une chaine de conditionnement

Soutenu le 28/12/2020

Réalisé par :

- MOKKADEM Ayoub

- MOUILID Achouak

Devant le Jury composé de :

- Dr. ASRADJ	M.A.A	Président	UAMOB
- Dr. HAROUN Ismail	M.C.B	Encadreur	UAMOB
- Dr. BOUHADA	M.A.A	Jury	UAMOB

2019/2020

Remerciements

La réalisation de ce mémoire hamdoullAallah a été possible grâce aux nos efforts même si les empêchements qui nous avons arrivé a cause de COVID 19 mais on a pu le faire.

En tout premier lieu, je remercie ALLAH, tout puissant, de nous avoir donné la sagesse, ainsi que le courage pour dépasser toutes les difficultés.

Nous tenons avant tout à exprimer notre reconnaissance à Mr. HAROUN pour son encadrement dans cette étude et ses importants conseils. Nous le remercions pour son implication, son soutien et ses encouragements tout au long de ce travail.

Nous remercions tout particulièrement les professeurs de l'université Akli Monhand Oulhadj, ainsi le chef du département génie électrique, aussi sans oublie tous nos professeurs qui nous ont accompagné dans notre parcours universitaire.

Je voudrais personnellement moi Achouak remercier Mr. FARES Omar et l'équipe de maintenance de Diagnopharme.

Nous voudrions exprimer notre reconnaissance envers les amis et collègues qui nous ont apporté leur support moral et intellectuel tout au long de nos démarches.

A toutes nos familles, qui porte le nom MOUILID et MOKKADEM, nous tenons à remercier nos parents très forts pour leurs présences toujours, nos frères et sœurs. A tout ceux qui ont participé à l'élaboration de ce modeste travail et tous ceux qui nous sont chers.

*A cœur vaillant rien d'impossible
A conscience tranquille tout est accessible*

*Quand il y a la soif d'apprendre
Tout vient à point à qui sait attendre*

*Quand il y a le souci de réaliser un dessein
Tout devient facile pour arriver à nos fins*

*Malgré les obstacles qui s'opposent
En dépit des difficultés qui s'interposent*

*Les études sont avant tout
Notre unique et seul atout*

*Ils représentent la lumière de notre existence
L'étoile brillante de notre réjouissance*

*Comme un vol de gerfauts hors du charnier natal
Nous partons ivres d'un rêve héroïque et brutal*

*Espérant des lendemains épiques
Un avenir glorieux et magique*

*Souhaitant que le fruit de nos efforts fournis
Jour et nuit, nous mènera vers le bonheur fleuri*

*Aujourd'hui, ici rassemblés auprès des jurys,
Nous prions dieu que cette soutenance
Fera signe de persévérance
Et que nous serions enchantés
Par notre travail honoré*

Je dédie cette mémoire à...

A Allah, tout puissant, que la gloire et l'honneur vous soient rendus pour les grâces que vous m'accordait.

A ma chère maman, Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études.

A mon cher papa, Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A ma chère grand-mère qu'est immortalisée dans sa tombe, malheureusement elle n'est pas dans ce jour important pour moi.

A mes chères sœurs : Amina, Anissa et Ahlem.

A mon cher unique frère Akram.

A mes adorables chères nièces et neveux Malak, Med Aïmen, Anaïs et Med Adam.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

Achouak

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.

Vous mon père qui n'êtes plus parmi nous que dieu vous accueille dans son vaste paradis.

Et vous chère mère que dieu le très haut vous accorde santé bonheur et longue vie et fasse que je serais toujours votre fierté et que jamais je ne vous déçoive, quoi que je fasse je ne pourrais vous acquitter.

A mes frère zakí saddam et akram qui n'ont cessé de me soutenir, encourager et conseiller que dieu les protèges et leurs offre la santé et le bonheur

A mes amis à qui je souhaite un avenir radieux et plein de réussite.

A tous ceux qui m'ont aidé de près ou de loin depuis mon enfance.

Ayoub

Sommaire

<i>Remerciements</i>	<i>i</i>
<i>Je dédie cette mémoire à</i>	<i>ii</i>
Liste des figures	vii
Liste des tableaux	ix
Les abréviations	x
Introduction générale	1
 Chapitre I: Généralité sur les systèmes d'inspection et la vision par ordinateur	
Introduction	3
I.1. Les systèmes d'inspection industrielle	3
I.1.1. Système industriel	4
I.1.2. Exemples de systèmes d'inspection industrielle	5
I.1.2.1. Vérification des récipients vides (qualité, intégrité et propreté)	5
I.1.2.2. Contrôle des niveaux avec changement de format automatique	6
I.1.2.3. Contrôle des bouchons et des capsules (position et intégrité) par caméra	7
I.1.3. Travaux de recherché dans le domaine d'inspection industrielle.....	8
I.2. Notions sur la vision par ordinateur	14
I.3. Description d'un système de vision par ordinateur	15
I.3.1. Définition	15
I.3.2. Classification des systèmes de vision par ordinateur	15
I.3.2.1. Les systèmes de vision bidimensionnelle.....	16
I.3.2.1. Les systèmes de vision tridimensionnelle	16
I.3.3. Applications de la vision par ordinateur	16
I.4. Traitement d'image dans la vision	17
I.4.1. Traitement image:	18
I.4.1.1. Prétraitement	18
I.4.1.2. Morphologie.....	20
I.4.1.3. Détection de contours.....	21
I.4.2 Traitement d'image couleur.....	21
A. L'espace colorimétrique RVB.....	21

B.	L'espace colorimétrique HSV	22
C.	Conversion RGB en HSV	23
Conclusion		24

Chapitre II: Conception du système et son architecture

Introduction.....	25
--------------------------	-----------

II.1. Conception de système d'inspection.....	25
------------------------------------------------------	-----------

II.1.1. Partie matérielle	25
---------------------------------	----

II.1.2. Partie logicielle	27
---------------------------------	----

II.2. Architecture du système	28
--------------------------------------------	-----------

II.2.1. Architecture matérielle.....	28
--------------------------------------	----

1) La carte Raspberry Pi	28
--------------------------------	----

2) Moteur a courant continu.....	29
----------------------------------	----

3) Le Moteur pas à pas.....	30
-----------------------------	----

4) Circuit de commande de puissance des moteurs.....	31
------------------------------------------------------	----

5) Capteur IR	31
---------------------	----

II.2.2. Architecture logiciel.....	32
------------------------------------	----

1) Système d'exploitation Ubuntu-Mate	32
---------------------------------------------	----

2) Vnc viewer	32
---------------------	----

3) Putty	33
----------------	----

4) Python.....	33
----------------	----

5) Thony python IDE.....	34
--------------------------	----

6) OpenCV.....	34
----------------	----

1) Algorithmes de traitement d'image.....	34
-------------------------------------------	----

Conclusion	35
-------------------------	-----------

Chapitre III: La réalisation matérielle du système

Introduction.....	37
--------------------------	-----------

III.1. Partie mécanique.....	37
-------------------------------------	-----------

III.1.1. Réalisation du Convoyeur	37
-----------------------------------------	----

III.1.2. Réalisation du mécanisme Poussoir	38
--------------------------------------------------	----

III.1.3. Réalisation du système de tri	39
----------------------------------------------	----

III.2. Partie électronique	39
-----------------------------------------	-----------

III.2.1. La carte Raspberry PI 4	39
----------------------------------------	----

III.2.2. Driver (pilote) du moteur pas à pas	40
----------------------------------------------------	----

III.2.3. Driver (pilote) du moteur a courant continu	41
------------------------------------------------------------	----

III.2.4. Le Capteur IR	42
III.2.5. La camera (Webcam).....	43
III.3. Branchement final de la partie électronique	43
III.4. Réalisation finale du système d'inspection	44
III.5. Evaluation de la réalisation matérielle	46
III.5.1. La Caméra avec le capteur IR.....	46
III.5.2. Le mécanisme Pousoir	46
III.5.3. Convoyeur avec le moteur à courant continu	46
III.5.4. Le mécanisme de triage	47
Conclusion	47

Chapitre IV: Implémentation de la partie logicielle du système

Introduction.....	49
IV.1. L'organigramme de fonctionnement général du système.....	49
IV. 2. Implémentation de l'algorithme.....	51
IV.2.1. Inspection du bouchon de bouteille.....	51
IV.2.2. Détection du niveau la bouteille	55
IV.3. Evaluation des algorithmes	58
IV.3.1. L'inspection de bouchon	61
IV.3.2. L'inspection de niveau	62
Conclusion	63
Conclusion générale.....	65
Travaux future.....	67
Référence bibliographique.....	68

Resumé

Liste des figures

Chapitre I : Généralité sur les systèmes d'inspection et la vision par Ordinateur

Figure I.1 : Système industriel [3] Page 4

Figure I.2: Structure d'un système industriel [5] Page 5

Figure I.3 : Loginspect machine contrôle [6] Page 6

Figure I.4 : Logilevel Plus machine contrôle [6] Page 7

Figure I.5 : Logics & control machine (inspection du bouchon) [6] Page 8

Figure I.6 : Ligne de production Festo MPS avec placement de produit précis [9] Page 8

Figure I.7 : Mécanisme d'inspection préposée dans la référence [9] Page 10

Figure I.8 : Solidification dans l'indice de regroupement [1] Page 11

Figure I.9 : Chevauchement avec la couleur d'origine du capuchon [1] Page 12

Figure I.10 : Segmentation d'image [1] Page 12

Figure I.11 : Afficher la sortie avec les propriétés de la région [1] Page 12

Figure I.12 : Schéma général pour un AVIS d'embouteillage typique [14] Page 13

Figure I.13 : Schéma fonctionnel de l'installation de remplissage de bouteilles du système [14] Page 14

Figure. I.14 : Schéma fonctionnel d'un système informatique pour la vision artificielle [15] Page 15

Figure I.15 : L'effet de filtre médiane [17] Page 18

Figure I.16 : Un exemple d'un seuillage adaptatif d'une image ayant 8 niveaux [18] Page 19

Figure I.17 : Cube couleur de bit [23] Page 21

Figure I.18 : Schéma de cube de couleur RVB [23] Page 22

Figure I.19 : Représentation cylindrique de l'espace HSV [23] Page 22

Figure I.20 : Le modèle HSV basé sur un plan circulaire [23] Page 23

Figure I.21 : Teinte et saturation dans le modèle circulaire HSV [23] Page 23

Chapitre II : La conception du système et son architecture

Figure II.1 : Vue 3D du système d'inspection réalisée sur logiciel BLENDER. Page 26

Figure II.2 : schéma synoptique de système réalisé. Page 27

Figure II.3 : Architecture du système. Page 28

Figure II.4 : La carte Raspberry pi 4. Page 29

Figure II.5 : Le moteur a courant continu. Page 30

Figure II.6 : Les types de moteur pas à pas . Page 30

Figure II.7: Different types de Motor driver. Page 31

Figure II.8: Module du capteur IR. Page 32

Figure II.9: Logos de différents outils logiciels utilisés. Page 34

Chapitre III : La réalisation matérielle du système

Figure III.1 : L'emplacement de moteur DC qui roule le convoyeur. Page 38

Figure III.2 : Conception et réalisation du mécanisme de poussoir. Page 38

Figure III.3 : Conception et réalisation du mécanisme de tri. Page 39

Figure III.4 : Branchement de la Raspberry Pi. Page 40

Figure III.5 : Branchement du Circuit DRV8825. Page 41

Figure III.6 : Le module L298N et son branchement. Page 41

Figure III.7 : Branchement et positionnement du Capteur IR. Page 42

Figure III.8 : La webcam utilisée dans le projet. Page 43

Figure III.9 : Câblage global des composants électroniques. Page 44

Figure III.10 : Visualisation de la mise en œuvre globale des matériels du système. Page 45

Figure III.11 : Vue globale du notre système. Page 45

Chapitre IV : La réalisation logicielle du système

Figure IV.1: Software architecture. Page 50

Figure IV.2 : Zone de bouchon. Page 52

Figure IV.3 : image masquée HSV. Page 53

Figure IV.4: Algorithme de détection de bouchon. Page 54

Figure IV.5 : Zone de liquide. Page 55

Figure IV.6: Algorithme de détection de niveau. Page 57

Liste des tableaux

Tableau IV. 1: Résultats du test du flacon avec du liquide orange. Page 59

Tableau IV. 2: Résultats du test du flacon avec du liquide violet. Page 60

Les abréviations

2D : 2 Dimensions

3D : 3 Dimensions

CPU: Unité centrale de traitement

DRV : Driver

DC : Courant continue

GPU : Processeur graphique

GPIO : General Purpose Input/Output (les portes)

GND : Ground (Terre)

HDMI : Interface Multimédia Haute Définition

HDPE : ou bien PEHD PolyEthylène haute densité

IR : Infrarouge

LAN : Local area network

OPC : Open platform communication

PET : Polyéthylène Téréphtalate

PLC : Courants porteurs en ligne

PWM : Pulse Width Modulation, ou MLI en français pour Modulation de largeur d'impulsion

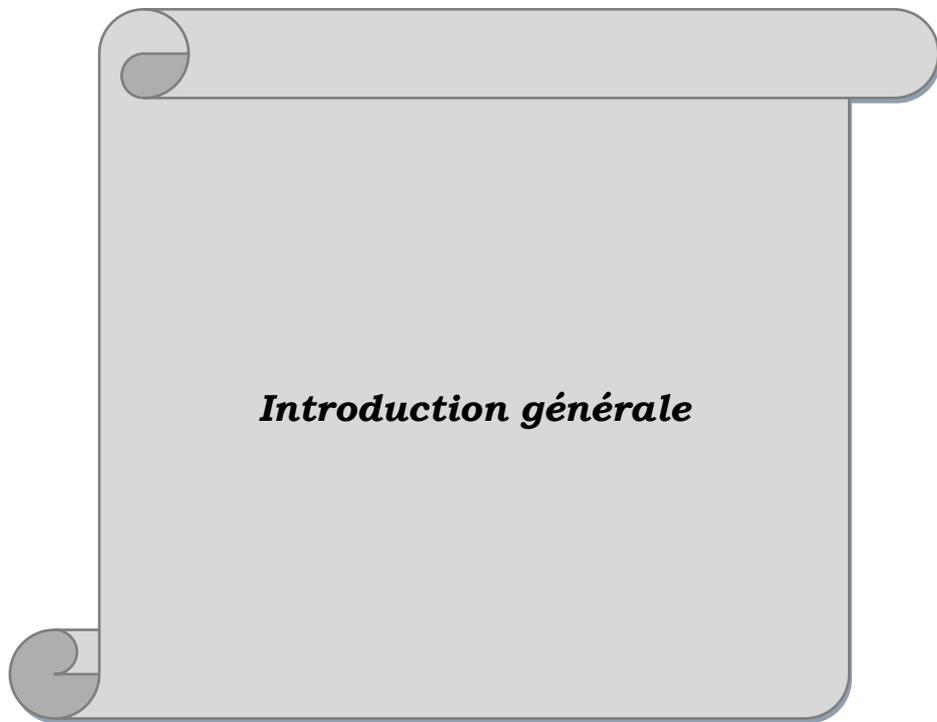
RAM : Random access memory (memoire vivre)

SSH : Secure shell

TCP: Transmission control

USB : Universel Serial Bus ou USB (en français bus universel en série)

USD : Dollar américain



Introduction générale

Introduction générale

Au cours des dernières années, l'industrie mondiale des boissons a explosé de façon exponentielle. Les premières statistiques ont montré que la taille du marché mondial des boissons non alcoolisées était évaluée à 967,3 milliards USD en 2016 [1], cette demande croissante s'accompagne d'un besoin de qualité de haut niveau et d'un suivi très rapide des processus de production, où l'assurance de la satisfaction du client et la préservation de la réputation de la marque est essentielle. Aujourd'hui, les humains effectuent des inspections finales d'objets dans des industries qui s'avèrent coûteuses et peu fiables. En gardant cette vue et les exigences croissantes d'objectivité, d'uniformité et de compétence ont imposé l'insertion de systèmes d'inspection automatisés. De nombreuses méthodes ont été utilisées pour les inspections comme les capteurs de couleurs pour l'inspection des couleurs et la vision par ordinateur, ...etc. L'évaluation et le catalogage automatisés peuvent réduire les charges humaines et les coûts de main-d'œuvre tout en augmentant le rendement.

Un système d'évaluation automatisé précis est utilisé dans de nombreuses industries pour de nombreux objets comme la couleur des biscuits de boulangerie, la couleur des croustilles, les produits textiles, les produits alimentaires et le bois. Cependant, les progrès et la classification automatisés des défauts ont été légèrement moins avancés en raison du comportement délicat du problème [1].

La vision automatique a été utilisée pour calculer avec précision la couleur de divers aliments, car elle offre des avantages supplémentaires et compréhensibles par rapport à une technique régulière telle que l'utilisation d'un colorimètre, à savoir la probabilité d'évaluer un pixel individuel de la surface complète de l'aliment et de calculer les caractéristiques extérieures et défauts.

L'inspection manuelle des produits ne peut souvent pas rattraper les exigences indispensables en vigueur, en termes de vitesse de production et de contrôle qualité. Les systèmes de vision industrielle, en revanche, offrent des solutions intégrales aux fabricants, avec leur faible coût, leur grande précision, leur robustesse et leur fiabilité. Ces caractéristiques ont ouvert la voie à une adoption croissante de la vision industrielle par les industries du monde entier. En fait, de nos jours, ces systèmes jouent un rôle essentiel dans une multitude de processus industriels, en particulier; l'industrie des boissons.

Les applications d'emballage et de distribution dans l'industrie des boissons impliquent l'inspection de l'intégrité des bouteilles, la garantie d'un étiquetage approprié et la vérification des niveaux de remplissage acceptables. Ces tâches sont très cruciales pour assurer une qualité parfaite des produits à livrer.

Dans ce travail, nous proposons de réaliser un système d'inspection et de triage de produit final dans une ligne de production de boissons. Ce système basé sur l'acquisition et traitement d'image, permet de détecter le niveau de remplissage et l'inspection de la fermeture du bouchon. Ce système permet aussi de trier le produit final en différentes catégories selon sa couleur. Le système d'inspection à réaliser est basé sur une caméra (Webcam) connectée à une carte Raspberry PI 4 et un ensemble de mécanismes motorisés par des moteurs pas à pas permettent de rejeter le produit non accepté et de réaliser la fonction du triage. Afin de faire fonctionner ce système convenablement, des algorithmes de traitement d'image seront développés et implémentés en langage Python sur la carte Raspberry PI.

Afin d'atteindre cet objectif, nous avons organisé notre travail en quatre chapitres :

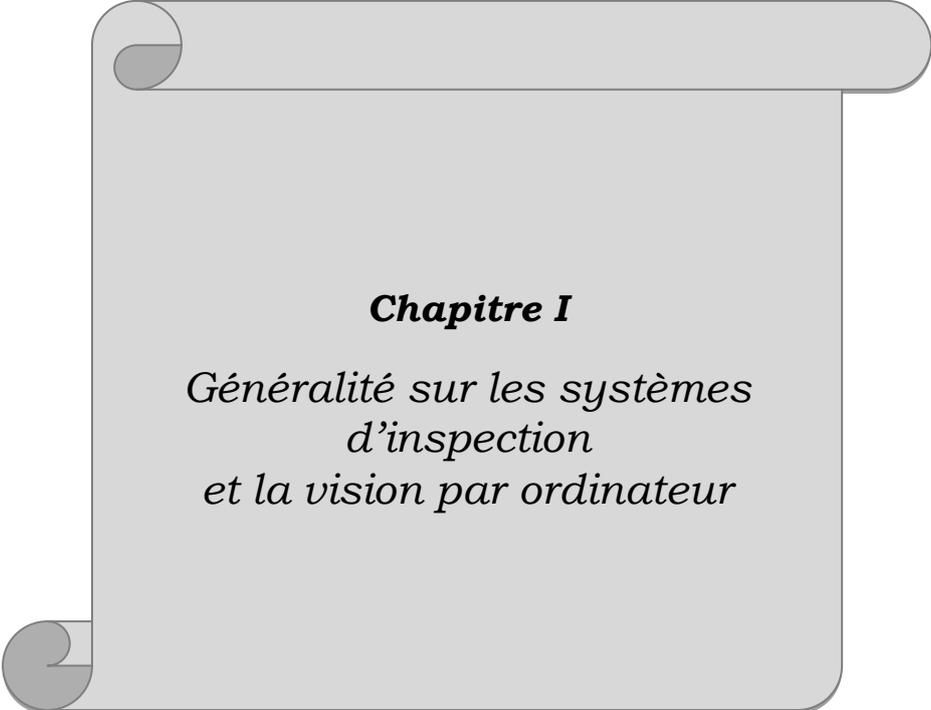
Dans le premier chapitre, nous allons présenter un aperçu général sur les systèmes d'inspection industriel par vision par ordinateur, ainsi que les travaux de recherches réalisés dans ce domaine. Nous présentons aussi dans ce chapitre la théorie derrière les algorithmes de traitement de signal et de vision par ordinateur utilisés dans notre système.

Le deuxième chapitre sera consacré pour la présentation de la conception globale du système à réaliser en commençant par un schéma synoptique puis la description de sa constitution matérielle et logiciel.

Dans le troisième chapitre, nous abordons les étapes de réalisation pratiques du système d'inspection proposé, où nous allons représenter en détail la partie mécanique, et la partie électronique du projet.

Le quatrième et dernier chapitre porte sur la présentation de la partie software du projet, où nous décrivons les algorithmes de traitement d'image, utilisés et l'organigramme du programme à implémenter sur la carte Raspberry. Nous présentons dans ce chapitre aussi quelques résultats pratiques de système d'inspection réalisés.

Enfin, nous terminons ce manuscrit par une conclusion générale et des perspectives permettant d'améliorer ce modeste travail.



Chapitre I

*Généralité sur les systèmes
d'inspection
et la vision par ordinateur*

Chapitre I

Généralité sur les systèmes d'inspection et la vision par ordinateur

Introduction

L'inspection des produits alimentaires implique l'exécution d'un certain nombre de fonctions. Celles-ci pourront être systématiques ou spécifiques aux produits inspectés, des procédures génériques ou spécifiques pouvant être élaborées en conséquence. Par exemple, quel que soit le produit dont il s'agit, l'expédition sera traitée à l'arrivée de la même manière (avis d'arrivée, point de savoir si une inspection s'impose, etc.), jusqu'au prélèvement des échantillons. Jusque-là, c'est un document ou une procédure d'instruction de caractère général précisant qui est responsable et à quel moment il y a lieu de procéder à telle ou telle intervention qui convient le mieux. En résumé, la procédure de fonctionnement précise le responsable et la nature de l'intervention, ainsi que les modalités de temps et de lieu.

Avec la naissance des machines de calcul, des recherches scientifiques sont effectuées, en essayant de concevoir une machine qui peut remplacer le système de vision humaine, pour arriver à des résultats similaires à la vision humaine.

Dans ce chapitre, nous allons présenter quelques définitions concernant les systèmes d'inspection et la vision par ordinateur. On présente aussi quelques techniques de traitement du signal utilisé dans la vision par ordinateur.

I.1. Les systèmes d'inspection industrielle

L'inspection est un ensemble d'activités qui recherche de façon systématique les anomalies pouvant se retrouver dans un milieu de travail. Elle permet ainsi d'identifier les dérogations aux normes, l'usure et la dégradation de l'équipement, l'absence de méthodes de travail sécuritaires ou tout autre risque relié aux machines, aux outils, au matériel, aux tâches, à l'environnement ou à l'aménagement des lieux. L'inspection est une technique d'identification des risques dite « préventive » parce qu'elle tente d'identifier les risques et vise à les corriger avant que les accidents ne surviennent. Son but est d'éliminer les lésions professionnelles [2].

La mise en place d'un système d'inspection entraîne plusieurs avantages pour une organisation. Elle permet [2]:

- L'élimination des lésions professionnelles.
- La détection :
 - De l'usure et de la détérioration des équipements.
 - De l'absence, l'inefficacité ou le non-respect d'une procédure de travail.
 - Des risques communs à plusieurs départements.
 - Du non-respect des mesures de sécurité.
- La diminution des pertes matérielles et financières.
- L'amélioration de la qualité des services.
- L'implication des travailleurs et des gestionnaires en santé et sécurité.
- La démonstration de l'importance que la direction accorde à la santé et à la sécurité de ses travailleurs.

Les systèmes d'inspection par vision étendent le système automatisé d'inspection basé sur l'image au profit de nombreuses applications industrielles et de fabrication. Des systèmes basés sur la vision bidimensionnelle (2D) et (3D) sont mis en œuvre pour l'observation automatisée, l'assurance qualité et la classification, ... etc. [2].

I.1.1. Système industriel

Un système industriel comprend le contexte, les ressources, les activités, les processus, les acteurs et les interdépendances qui soutiennent la création et la fourniture de produits et de services. Une compréhension plus claire des systèmes industriels - une vision holistique - peut identifier les «leviers» qui sont disponibles pour générer et, surtout, capturer de la valeur.



Figure I.1 : Système industriel [3].

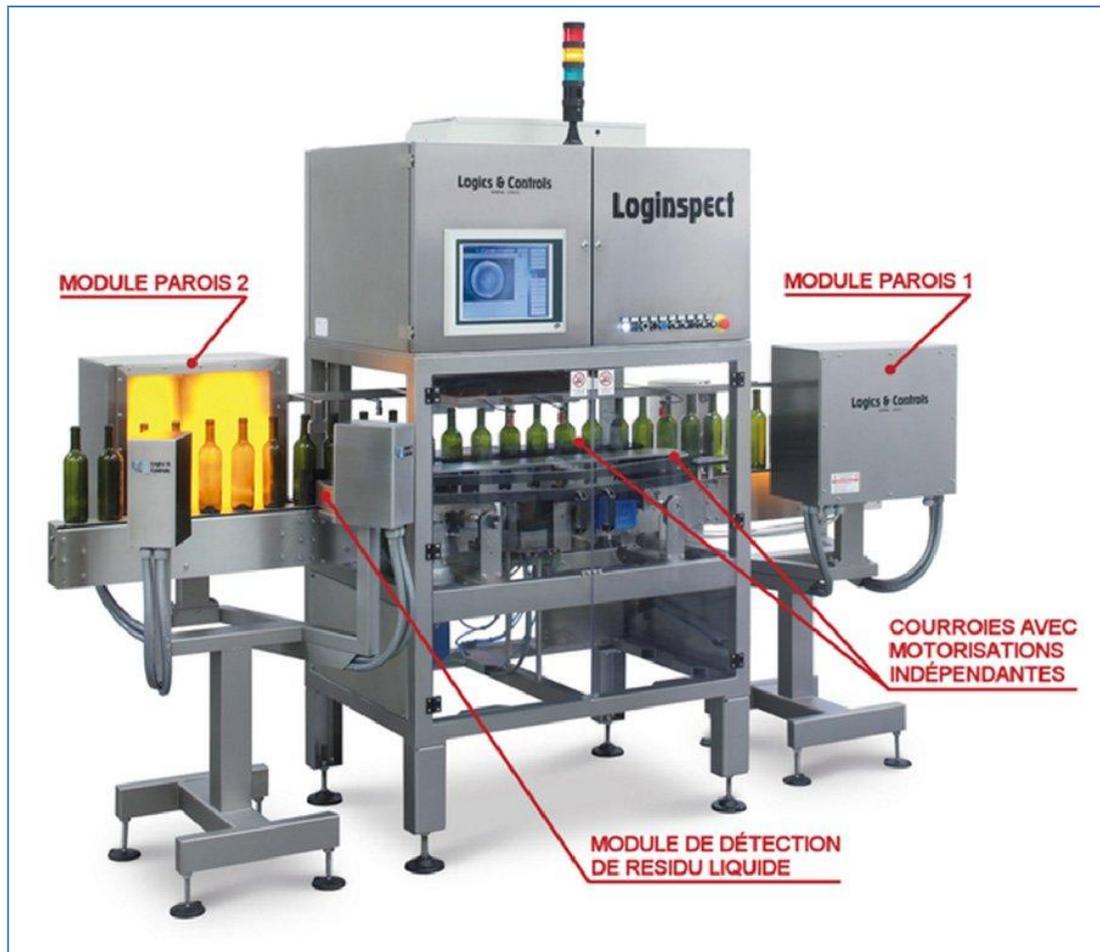


Figure I.3 : Loginspect machine contrôle [6].

I.1.2.2. Contrôle des niveaux avec changement de format automatique

La machine présentée dans la figure I.4 est un système dédié au contrôle du bon dosage du produit dans les récipients en verre, PET et HDPE.

Compact, simple, rapide et facile à installer, il offre la réponse idéale aux entreprises en quête d'un système fiable et technologiquement avancé. Il suffit de fournir la hauteur de la bouteille en mm et du niveau depuis le ras du goulot en mm. Pourvu de son propre système motorisé pour le réglage du format, c'est la dernière évolution de ce type de contrôle aussi bien en matière de niveau que de bouchon.

Ne requérant pas la présence de l'opérateur pour le réglage manuel du contrôle, il dépasse définitivement la limite du contrôle de niveau traditionnel pour donner naissance à un type de contrôle de niveau absolu. L'information relative au produit et au format à contrôler peut venir directement de la tireuse. Produit polyvalent et robuste, il est conçu et prédisposé pour être étendu et contrôler la présence du bouchon, des étiquettes, de la capsule et du muselet [6].



Figure I.4 : Logilevel Plus machine contrôle [6].

I.1.2.3. Contrôle des bouchons et des capsules (position et intégrité) par caméra

La figure I.5 présente un système à caméra dédié au contrôle de la fermeture correcte des récipients en plastique ou en verre. Il vérifie avec une grande précision l'intégralité et le positionnement correct du bouchon et de la bague de garantie. Il peut également effectuer le contrôle du niveau et le contrôle des vignettes, des étiquettes ou d'autres détails tels que le code du lot. Il peut être complété par la fonction LOGIFILLER de diagnostic des robinets, des têtes de bouchage et du contrôle éclatement bouteilles. Compact, simple à installer sur n'importe quelle nouvelle chaîne de production ou chaîne existante, ce système englobe tout ce qui est nécessaire pour vérifier le fonctionnement correct du système de remplissage et de bouchage bouteilles [6].



Figure I.5 : Logics & control machine (inspection du bouchon) [6].

I.1.3. Travaux de recherche dans le domaine d'inspection industrielle

- ✓ En 2010, *Joni Siponen* [7] a présenté un travail sur «la mise en œuvre de la vision industrielle sur une ligne de fabrication ». L'objectif principal de cette thèse était de concevoir un système de vision par ordinateur pour une chaîne de montage afin de réparer l'agencement inexact de la pièce [7].



Figure I.6 : Ligne de production Festo MPS avec placement de produit précis. [7]

Cette thèse a été principalement menée pour corriger une éventuelle situation problématique dans une chaîne de montage. Le capteur de vision industrielle vérifie si une pièce était correctement positionnée. Si le placement de la pièce est inexact, la chaîne de montage s'arrête.

✓ ***Bikarna Pokharel (2013)*** a présenté une thèse sur «la vision industrielle et le tri d'objets, la communication PLC avec LabVIEW utilisant OPC». L'intention centrale de cette thèse était de signifier une implémentation de vision par ordinateur pour l'assurance qualité des produits à petite échelle industries [8].

Cette thèse a été principalement menée pour trier les produits corrects et endommagés. Il a collaboré à la technique de vision industrielle avec les outils de vision NI pour le tri des produits en ligne de production [8]. Il est intégré à PLC LABVIEW en utilisant le protocole de communication OPC.

Le projet a utilisé le logiciel STEP7 pour coder le programme de séquence logique pour l'automatisation du système. Ce programme a été téléchargé dans l'automate programmable. VI (inspection visuelle / programme LABVIEW) a été réalisé pour l'inspection et le tri des objets. L'image de l'objet est obtenue par LABVIEW à l'aide d'une caméra Web. L'image est analysée à l'aide d'un algorithme de correspondance de motifs.

✓ ***Développement d'un système d'inspection automatisé par Mme Shubhada.K. Nagrale, M.S.T. Bagde (2013)***, Mme Shubhada.K. Nagrale et Mr.S.T. Bagde a présenté une étude sur «le développement d'un système d'inspection automatisé à l'aide du traitement d'images». Le but principal de ce projet était le développement d'un système qui accepte ou rejette un produit en fonction de son poids et sa géométrie à l'aide du traitement d'images [9].

Trois types de mécanismes ont été discutés dans ce projet. Mécanisme de type disque rotatif, mécanisme Flip drop et mécanisme de type poussoir. Dans le mécanisme du disque rotatif, l'objet est placé sur le convoyeur et sur le portique, la caméra est fixée sur la pièce. L'appareil photo prend la photo et l'associe à d'autres modèles, puis progresse vers la structure du disque où la pondeuse est placée. Si la géométrie et les autres modèles de poids sont mis en correspondance, passe à AC puis au centre d'expédition. Sinon, va au RC et au centre de reprise [9].

Mécanisme poussoir, la pièce est envoyée au centre d'acceptation et de rejet à l'aide du poussoir. Il y a un bloc triangulaire où la machine de pondération est conservée et la caméra est installée dans cette zone sur la pièce. Si le poids et les autres motifs correspondent, alors pousser la partie inférieure du poussoir dans le centre d'acceptation. Sinon, pousser le

poussoir supérieur dans le centre de rejet. Pendant le basculement, la pièce est retournée avec un moteur et déposée dans le centre de rejet ou d'acceptation de 90 degrés.

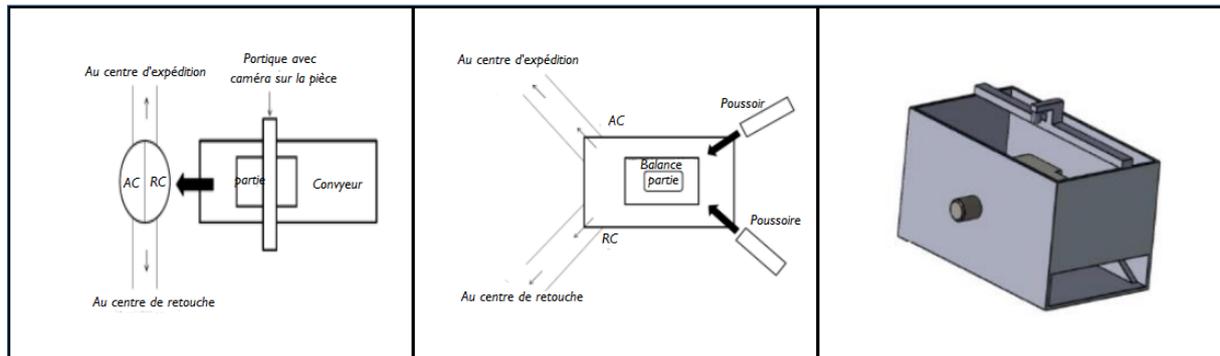


Figure I.7 : Mécanisme d'inspection préposée dans la référence [9].

- ✓ **Traitement d'image numérique à l'aide de LabView** : c'est une thèse dans laquelle ils ont décrit une technique de traitement d'image numérique utilisant LABVIEW [10]. Ces dernières années, le traitement numérique de l'image a été effectué en utilisant le logiciel LabView en combinaison avec la boîte à outils de d'acquisition et vision vision d'image qui propose une variété d'outils très utiles pour l'acquisition et le traitement d'images [10]. Il améliore non seulement l'efficacité mais réduit également l'effort de programmation en donnant des résultats en très peu de temps. Un robot mobile a été conçu qui utilise une webcam pour capturer l'image et reconnaître l'objet, puis se diriger vers lui et le saisir par une pince. Le robot peut être contrôlé manuellement par le joystick et automatiquement par les algorithmes de traitement d'images utilisés pour la prise de décision.
- ✓ **En 2015, Paranesh Dahal** a présenté une étude sur «La conception d'un système d'identification d'objets basés sur la vision industrielle». L'objectif de ce projet était de signifier une application de vision pour l'identification d'objet basée sur la taille, la forme et la couleur à l'aide de LabView.

Ce projet a utilisé LabVIEW pour l'interface utilisateur et pour le traitement d'images. Le développement de Gcode a été fait pour fournir un panneau interactif à l'utilisateur. Le cadre comportait une bande transporteuse sur laquelle les objets étaient passés et leurs images capturées par caméra. L'image de l'objet a été obtenue par LabView à l'aide d'une caméra et analysée à l'aide d'algorithmes d'appariement de motifs.

- ✓ **Inspection du bouchon de bouteille par Pruthvi Kumar et Dr H.V Ramakrishna** : Un autre projet similaire a été réalisé en 2015 sur le même sujet par Pruthvi Kumar S, Dr H.V Ramakrishna MTech in Industrial Automation and Robotics, Department of Mechanical

Engineering, Professor, Department of Mechanical Engineering in Malnar College of Engineering, Hassan 573202, Karnataka.

Ce projet est basé sur un algorithme de combinaison qui comprend quatre cas ; détection de défaut latéral, détection de défaut par le haut, détection de bouteille sans bouchon et bouchon de bouteille de couleurs différentes sur le convoyeur. Il y a deux unités qui sont l'unité d'inspection et celle de rejet. MATLAB a été la base de programmation pour développer l'algorithme. Sur la base de l'image obtenue, le processeur prendrait alors une décision, c'est-à-dire soit acceptable ou non [11].

Cette unité d'inspection se compose essentiellement d'un module IR, d'un microcontrôleur et d'une webcam, d'un convoyeur et d'un pilote de moteur. Chaque fois que la bouteille passe le long du convoyeur et qu'elle tombe sous la gamme du capteur IR, il envoie un signal au microcontrôleur et qui transmet le signal aux pilote de moteur pour contrôler la vitesse du convoyeur puis enfin le convoyeur s'arrête. La webcam capture l'image et l'envoie à l'ordinateur où le traitement est effectué sur la base de quatre cas, y compris la bouteille sans défaut. Lorsqu'un défaut est détecté, l'unité de rejet est activée. Il s'agit d'un bras à commande automatique, qui est lié au moteur à courant continu, qui frappe la bouteille défectueuse, hors du convoyeur.

L'outil d'acquisition d'image est utilisé pour capturer l'image de la webcam qui est interfacée avec le PC [12] [13]. Le contrôleur AT mega32 est utilisé pour traiter le signal du capteur infrarouge, et le contrôle des moteurs à travers son pilote [11].

Le traitement d'images comprend 3 étapes : l'étape 1 : séparer la couleur RVB, trouver la taille de la matrice et trouver la taille du pixel rouge pour la détection. Étape 2 : L'index de regroupement des images est effectué à l'aide de l'algorithme K. Dans le regroupement, le processus de solidification a lieu.



Figure I.8 : Solidification dans l'indice de regroupement [1].

Étape 3 : il montrait les bouchons de bouteille segmentés en couleur par chevauchement de la couleur d'origine à la partie du bouchon de l'image groupée [1]. La couleur d'origine est comparée à la partie défectueuse du capuchon et à la fin la couleur d'origine est visible.



Figure I.9 : Chevauchement avec la couleur d'origine du capuchon [1].

Ensuite, la segmentation de l'image est effectuée où les calculs du capuchon sont effectués en comparant avec l'image d'origine qui est la longueur, le diamètre et la largeur.



Figure I.10 : Segmentation d'image [1].

Par la suite, la sortie finale est affichée en calculant la surface du bouchon de la bouteille. Si cette zone est inférieure à la zone du bouchon d'origine, la bouteille est défectueuse et transmise à l'unité de rejet. Si l'aire est supérieure à r égaux, elle est considérée comme exacte et réussie.



Figure I.11 : Afficher la sortie avec les propriétés de la région [1].

Cela s'est avéré être un système simple, très rentable et flexible car aucun changement n'est nécessaire si différents types de défauts doivent être détectés, ce qui économise du temps de programmation et de traitement.

- ✓ **En 2016, Varshal Shah** a présenté une étude sur «le remplissage de bouteilles basés sur le traitement d'images et la vérification d'étiquettes à l'aide de systèmes embarqués». L'objectif principal de ce projet était l'inspection des bouteilles par traitement d'images. Le matériel du projet consiste en un tapis roulant sur lequel les bouteilles ont été passées et les images des bouteilles sont capturées par une caméra. L'image avait été envoyée au logiciel de traitement d'images, puis des algorithmes de traitement d'images avaient été appliqués, puis un bon fonctionnement avait été effectué [14].

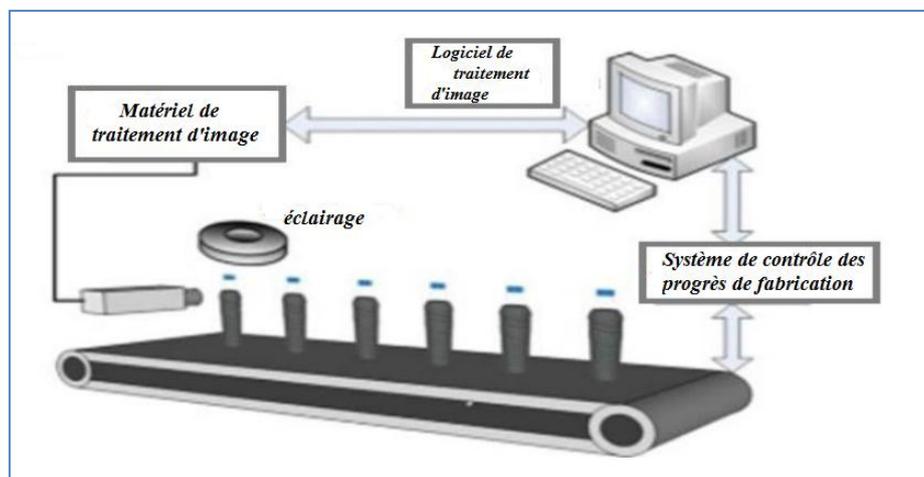


Figure I.12 : Schéma général pour un AVIS d'embouteillage typique [14].

Dans ce projet, lorsque la bouteille a été détectée par un capteur infrarouge, la valve commence le remplissage de la bouteille. Si le niveau d'eau atteint le niveau de référence, le moteur est mis en marche et l'électrovanne s'arrête. Si le niveau déborde l'étiquette manque sur la bouteille, un Buzzer est déclenché.

La réduction du bruit a été effectuée par filtrage médian et la correspondance des étiquettes a été effectuée par correspondance des modèles. La détection du niveau d'eau a été effectuée par un algorithme de seuil et un algorithme de remplissage de trous. La programmation a été faite en python.

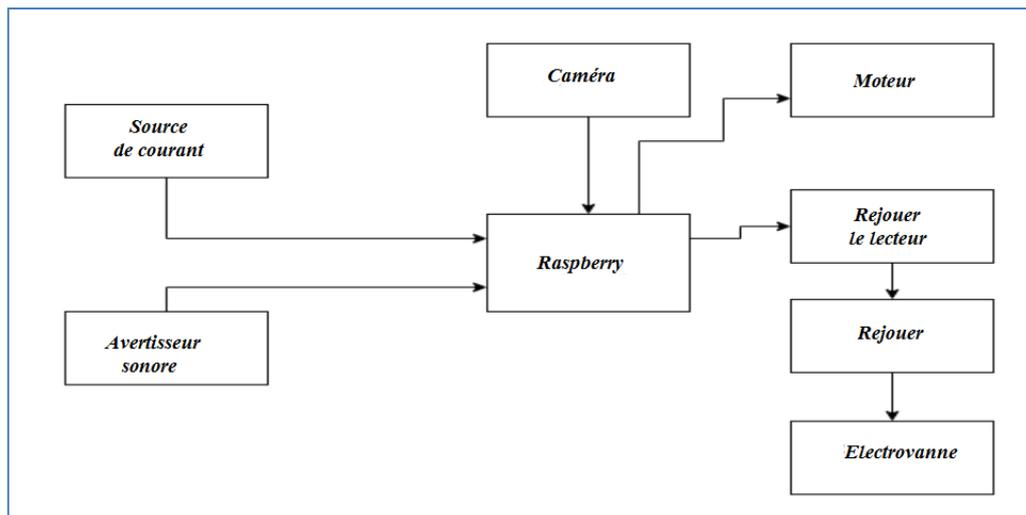


Figure I.13 : Schéma fonctionnel de l'installation de remplissage de bouteilles du système [14].

I.2. Notions sur la vision par ordinateur

La vision est le sens qui nous fournit le plus d'informations sur le monde extérieur, elle nous permet de voir, de décrire, et de reconnaître les objets qui sont présent dans la scène. C.à.d. : La vision est un processus de traitement de l'information. Elle utilise des stratégies bien définies afin d'atteindre ses buts. L'entrée d'un système de vision est constituée par une séquence d'images. Le système lui même apporte un certain nombre de connaissances qui interviennent à tous les niveaux. La sortie est une description de l'entrée en termes d'objets et de relations entre ces objets.

La vision par ordinateur (aussi appelée vision artificielle ou plus récemment vision cognitive) constitue une des branches de l'intelligence artificielle, c'est un domaine d'investigation et d'applications nées à la fin des années 50 dont les premières bases théoriques ont été définies dans les années 60. Depuis, étant donné le spectre très large d'applications, très peu de problèmes ont trouvé des solutions entièrement satisfaisantes. La recherche en vision est divisée en plusieurs approches différentes, chacune étudie des problèmes différents utilisant des techniques différentes.

La vision par ordinateur 2D met l'accent sur les techniques actuelles d'analyse d'image, d'apprentissage et de reconnaissance des formes. La vision par ordinateur 3D approfondit quand à elle les approches permettant d'extraire d'une ou plusieurs images des informations tridimensionnelles relatives à la scène photographiée. Par exemples l'industrielle, militaires, aérospatiales, la télédétection... [15].

La vision par ordinateur est la discipline qui cherche à reproduire la perception visuelle humaine sur un ordinateur. Le processus d'acquisition d'images est effectué à l'aide de caméras. Ces images seront traitées afin de produire une description des éléments qui composent la scène 3D.

La figure I.14 explique un processus qui utilise un ensemble d'images et des connaissances afin d'aboutir à une description d'une scène.

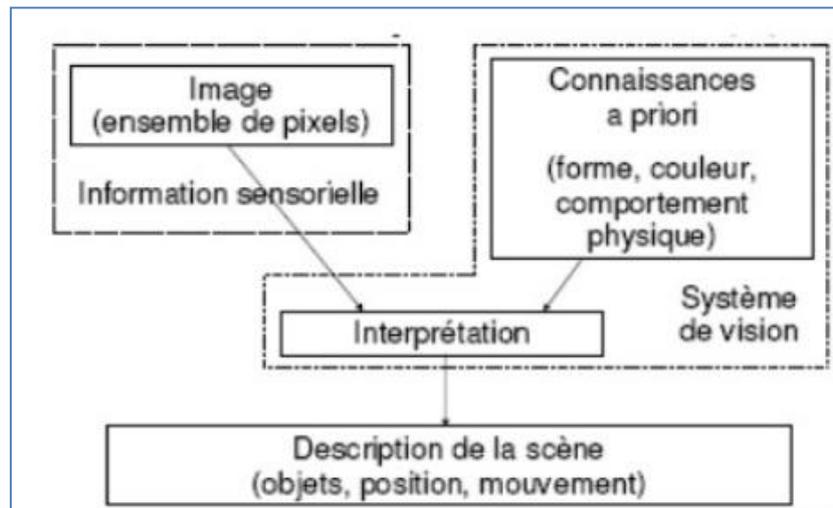


Figure. I.14 : Schéma fonctionnel d'un système informatique pour la vision artificielle [15].

I.3. Description d'un système de vision par ordinateur

I.3.1. Définition

Un système informatique de vision est un ensemble de processus fortement liés et composés de matériels et de logiciels. A l'aide de ces processus et des images prises de la scène, il fournit en sortie une description symbolique puis sémantique de la scène observée. La description est réalisée à l'aide de l'identification des attributs de la scène et des objets qui la composent.

Le système informatique reçoit en entrée une séquence d'images qu'il traite en utilisant la connaissance spécifique aux différents niveaux de traitements, en sortie il fournira une description en termes d'objets et de relations entre ces derniers [15].

I.3.2. Classification des systèmes de vision par ordinateur

Selon la nature de la scène observée on peut déterminer le type de système, on distingue deux types de systèmes qui sont :

I.3.2.1. Les systèmes de vision bidimensionnelle

Ces systèmes permettent une acquisition de l'image de la scène 2D, ils n'exploitent pas la notion de profondeur et donc reconnaissent uniquement les objets plans et isolés. Ces systèmes sont utilisés dans des domaines spatiaux tels que la télédétection et la reconnaissance de caractères, la robotique industrielle... [15]

I.3.2.1. Les systèmes de vision tridimensionnelle

Ces systèmes sont capables d'identifier les objets à partir d'une image d'un monde tridimensionnel, c'est pour cela qu'ils sont utilisés en robotique car ils permettent à un robot de se déplacer dans un environnement inconnu à obstacles [15].

I.3.3. Applications de la vision par ordinateur

Les applications vont de la vision industrielle (par exemple dans l'industrie de fabrication de bouteilles), à la recherche dans le domaine de l'intelligence artificielle et des ordinateurs ou robots capables de « comprendre » le monde qui les entoure. La vision par ordinateur et la vision industrielle sont des domaines qui se croisent ou se confondent assez souvent [16].

- La vision par ordinateur recouvre la technologie centrale de l'analyse d'image automatique, qui est utilisée dans de nombreux contextes.
- La vision industrielle fait plutôt référence à une combinaison de techniques d'analyse d'image automatique, de mise en correspondance, de traitement de l'image acquise par caméra (éventuellement stéréoscopique) et de technologies d'acquisition d'image, par exemple, pour inspecter des objets ou des lieux, de façon automatique, afin d'orienter des robots ou de permettre le déroulement d'un processus industriel.

Dans les deux cas, l'image est parfois acquise dans des gammes de longueur d'onde que l'homme ne perçoit pas (infrarouge, ultraviolet, rayon x, etc.) ou à travers une paroi et parfois par des capteurs spéciaux, par exemple adaptés à des milieux extrêmes tels que l'intérieur de certaines installations nucléaires, fours industriels, accélérateur de particule, etc.).

En tant que discipline scientifique, la vision par ordinateur traite de la théorie qui se trouve derrière les systèmes de traitement de l'information (information contenue dans les images ou pouvant être déduite d'une analyse plus contextuelle). Les données extraites le sont sous différentes formes : séquences vidéo, vues depuis différentes caméras, données multidimensionnelles provenant d'un scanner (scanner médical ou industriel) [16].

En tant que discipline technologique, la vision par ordinateur cherche à appliquer ses théories et ses modèles à différents systèmes. Quelques exemples de systèmes d'application de la vision par ordinateur :

- ✓ Procédés de contrôle, par exemple dans la robotique industrielle.
- ✓ Navigation, par exemple dans un véhicule autonome ou un robot mobile.
- ✓ Détection d'événements, par exemple pour la surveillance ou le comptage automatique de personnes.
- ✓ Organisation d'informations, par exemple pour indexer des bases de données d'images et de suites d'images.
- ✓ Modélisation d'objets ou d'environnements, par exemple dans l'analyse d'image médicale ou la topographie.
- ✓ Interaction, par exemple en tant que dispositif d'entrée pour une interaction homme-machine.
- ✓ Inspection automatique, par exemple dans des domaines de production.

On considère comme sous-domaines de la vision par ordinateur la reconstruction de scène, la détection d'événements, la détection d'intrusion, le match moving, la reconnaissance d'objets, l'estimation de mouvement, ainsi que certaines formes d'apprentissage automatique, d'indexation, de restauration d'image, etc. [16]

Dans la plupart des applications pratiques de la vision par ordinateur, les ordinateurs sont préprogrammés pour effectuer une certaine tâche, mais les méthodes fondées sur l'apprentissage sont de plus en plus populaires. [16]

Les problèmes posés par la modélisation de la vision sont loin d'être résolus. [16]

I.4. Traitement d'image dans la vision

Les images sont des grilles rectangulaires bidimensionnelles composées d'éléments fondamentaux, appelés «pixels» (abréviation d'élément d'image) couvrant la zone de l'image. Chaque pixel est numérisé en un certain nombre de bits, souvent appelé «profondeur de bits». On peut alors classer les images en trois types principaux [16]:

- Images binaires: le pixel peut prendre jusqu'à 21 états différents conduisant à une image en noir et blanc.
- Images à échelle de gris: généralement, le pixel est numérisé en un octet, ce qui donne une plage allant de 0 à 255 représentant différents nuances de gris.

- Images couleur: le pixel comprend trois composantes de couleur (rouge, vert et bleu), où chaque couleur est souvent représentée par un seul octet.

I.4.1. Traitement image:

Il existe une panoplie d'opérateurs en traitement d'images qui permettent d'améliorer l'aspect d'une ou des images. Ces opérateurs sont tantôt locaux, tantôt globaux. En ce qui concerne le projet, les images acquises sont plusieurs fois modifiées afin d'extraire des informations utiles permettant de tirer des conclusions (par exemple, décider si un contour trouvé est une semence ou pas). Dans un premier temps, le prétraitement sera abordé. Puis, des traitements de plus haut niveau seront décrits [16]

I.4.1.1. Prétraitement

Il faut signaler, d'emblée, que les images acquises par la caméra du système sont en niveau de gris. Plus précisément, des images constituées de 1928x1448 pixels ayant chacune une valeur de niveau de gris dans l'intervalle [0,255].

A. Réduction du bruit

Les dispositifs qui captent des images sont imparfaits. Dès lors, les images acquises sont bruitées. Il est important de l'éliminer autant que faire se peut tout en minimisant l'élimination de bons pixels. Un des filtres utilisés est le filtre médian qui élimine le bruit impulsif (poivre et sel). L'idée est d'enlever les pixels ayant des valeurs extrêmes et donc improbables en les rendant plus proches des pixels voisins [17]. L'effet de ce filtre peut être visualisé sur la figure suivante

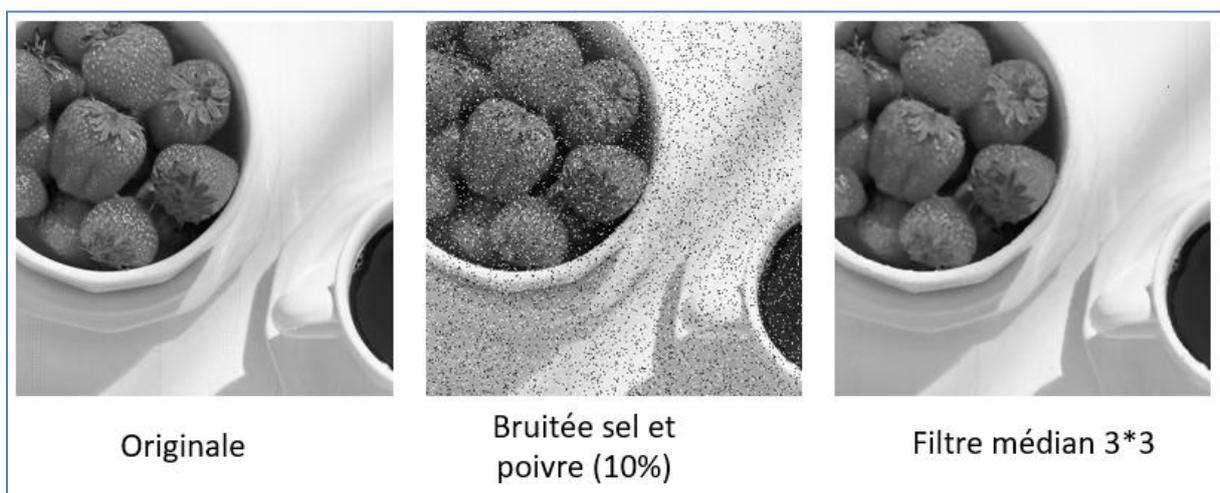


Figure I.15 : L'effet de filtre médiane [17].

B. Seuillage binaire adaptatif

Plusieurs fonctions en traitement d'images (détection de contours, notamment) nécessitent une image binaire. C'est-à-dire, un background noir et l'objet d'intérêt en blanc ou l'inverse. Il existe une multitude de façons de faire, mais une façon intéressante est le seuillage binaire adaptatif. L'image est parcourue par une fenêtre carrée de $b \times b$ et un seuil T E $[0,255]$ est choisi. Le seuillage s'effectue donc localement dans chaque sous-région. Cette méthode est donc utile lorsque les gradients d'illumination ou de réflexion sont élevés [18]. Schématiquement, sur la figure I.16, le seuillage est effectué dans le bloc, schématisé par un cadrage et non pas d'une manière globale.

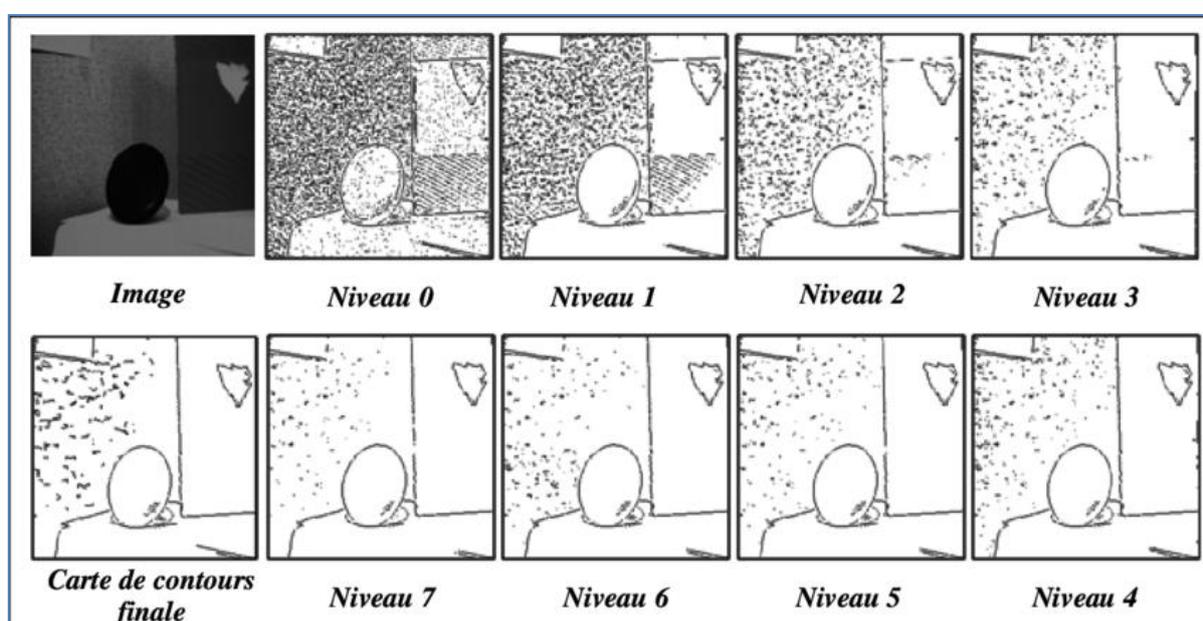


Figure I.16 : Un exemple d'un seuillage adaptatif d'une image ayant 8 niveaux [18]

C. Méthode d'Otsu

En vision par ordinateur et traitement d'images, la méthode d'Otsu [19] est utilisée pour effectuer un seuillage automatique à partir de la forme de l'histogramme de l'image, ou la réduction d'une image à niveaux de gris en une image binaire. L'algorithme suppose alors que l'image à binariser ne contient que deux classes de pixels, (c'est-à-dire le premier plan et l'arrière-plan) puis calcule le seuil optimal qui sépare ces deux classes afin que leur variance intra-classe soit minimale [19]. L'extension de la méthode originale pour faire du seuillage à plusieurs niveaux est appelée Multi Otsu Méthode. Le nom de cette méthode provient du nom de son initiateur, Nobuyuki Otsu. Elle ne doit pas être confondue avec la méthode d'Antzu.

I.4.1.2. Morphologie

La morphologie mathématique [20] est une branche des mathématiques et de l'informatique et trouve beaucoup d'applications en traitement de l'image, notamment. Actuellement, dans le cadre du projet, deux opérations de base tirées de cette discipline sont utilisées dans la phase de prétraitement.

a. Dilatation et érosion

Pour passer en revue le processus de dilatation et d'érosion, nous devons introduire ce qu'on appelle l'élément structurant :

L'élément structurant est un masque de forme d'une taille et d'une origine spécifiées utilisé dans les opérations morphologiques de base.

Dilatation : Avec une image binaire A et un élément structurant B, la dilatation de A par B est obtenue en déplaçant B sur A et en plaçant l'origine à chaque point, un pixel est attribué 1 si au moins un élément sous l'élément structurant en est un.

La dilatation remplit de petits trous dans un objet, permet aux pixels de premier plan d'une image de croître ou de se développer et de combiner des objets suffisamment proches les uns des autres mais non connectés [21].

Érosion : L'érosion est juste l'opposé de la dilatation: en déplaçant A sur B et en plaçant l'origine à chaque point, un pixel l'image d'origine (1 ou 0) ne sera considérée comme 1 que si tous les pixels sous le noyau sont 1, sinon elle est érodé (mis à zéro) [22].

Comme la dilatation est utilisée pour développer des objets dans une image, l'érosion est utilisée pour réduire les entités en supprimant les pixels de la limite de cet objet. L'érosion peut également être utilisée pour supprimer les branches, les ponts et les saillies.

b. Ouverture et fermeture (Opening et Closing)

L'Opening est l'opération composée d'érosion suivie d'une dilatation. Alors que la fermeture est l'opération composée de dilatation suivie d'érosion où pour chaque opération le même élément structurant est utilisé.

L'Opening est généralement utilisée pour lisser le contour d'un objet, brise les isthmes étroits et élimine les saillies minces.

Le Closing tend également à lisser les sections des contours mais, contrairement à l'Opening, elle fusionne généralement les ruptures étroites et les longs golfes minces, élimine les petits trous et comble les lacunes dans le contour [23].

I.4.1.3. Détection de contours

La détection de contours n'est pas simple et plusieurs approches peuvent être employées. Par exemple, le modèle de contour actif, introduit par Kass et Witkin [24], utilise une approche variationnelle. Il existe aussi des approches morphologiques. En ce qui concerne le projet, la fonction 4 qui détecte les contours d'une image binaire utilise un algorithme introduit par Suzuki et Abe dans l'article [25] : {Topological Structural Analysis of Digitized Binary Images by Border Following}. Cet algorithme inspecte chaque pixel et sa connectivité afin de le classer, grâce à un code (Freeman), comme étant un élément de contour ou pas. Cette façon de le faire est beaucoup plus proche de la morphologie mathématique que les contours actifs.

I.4.2 Traitement d'image couleur

Le modèle de couleur (également appelé espace colorimétrique ou système de couleurs) a pour but de faciliter les couleurs d'une manière standard, généralement acceptée [23]. Nous ne sommes intéressés que par deux espaces colorimétriques: le RVB et les modèles HSV.

A. L'espace colorimétrique RVB

Dans le modèle RVB, chaque couleur est le résultat de la combinaison linéaire des couleurs primaires: rouge, vert et bleu comme le montre la figure I.17, qui peut être représentée en utilisant un système de coordonnées cartésiennes comme illustré dans la figure I.18. Le cube montré est normalisé de telle sorte que les valeurs des couleurs primaires tombent dans la plage [0, 1].

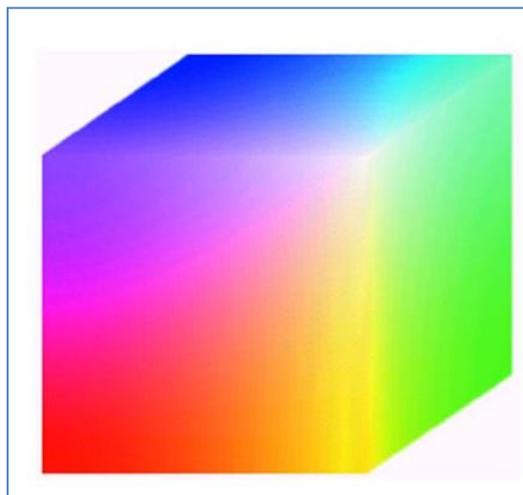


Figure I.17 : Cube couleur de bit. [23]

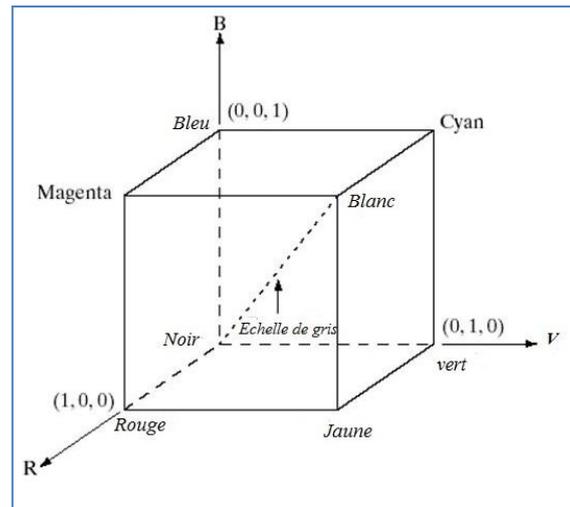


Figure I.18 : Schéma de cube de couleur RVB. [23]

B. L'espace colorimétrique HSV

Le modèle HSV comprend les trois éléments suivants :

- H- Hue (ou teinte) : Il représente la particularité de la couleur.
- S- Saturation : la saturation donne une mesure du degré auquel une couleur pure est diluée par la lumière blanche [23]. Cela peut être vu comme la couleur du pixel.
- V- Valeur (ou intensité) : indique la luminosité de la couleur.

Le HSV peut être considéré comme un cylindre tridimensionnel, comme le montre la figure I.19. Le fait que le HSV sépare la luminance de l'image des informations sur les couleurs le rend très approprié dans de nombreuses applications, car il capture les composantes de couleur telles que les humains perçoivent la couleur.

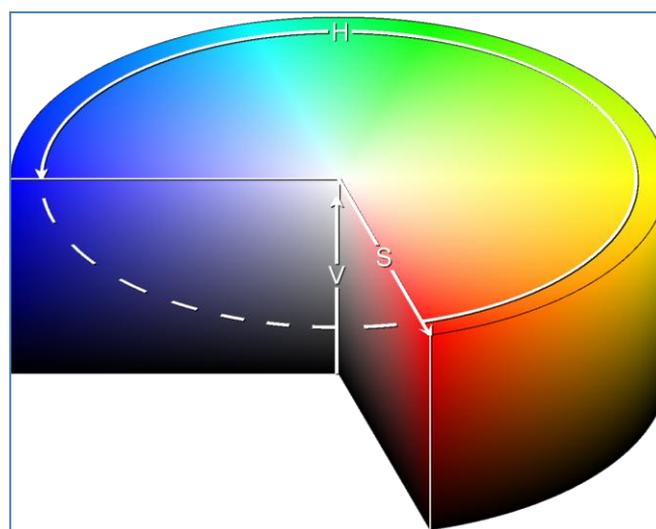


Figure I.19 : Représentation cylindrique de l'espace HSV. [23]

C. Conversion RGB en HSV

La conversion de RGB à HSV est mieux introduite en utilisant la représentation géométrique montrée dans la figure I.20

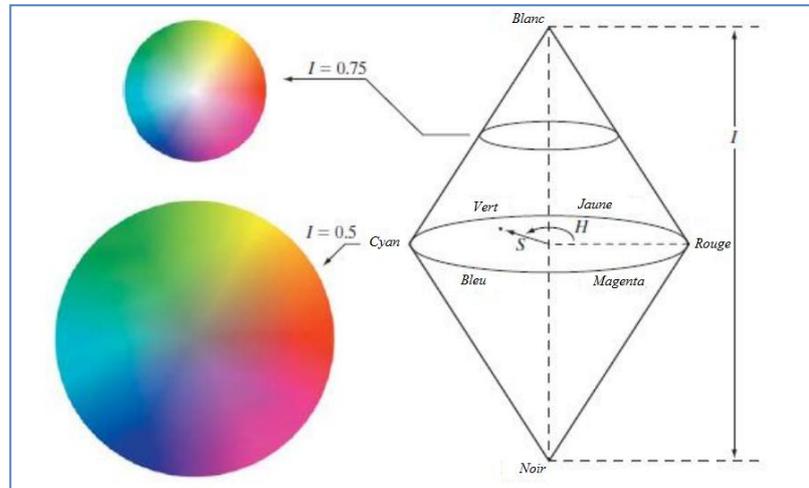


Figure I.20 : Le modèle HSV basé sur un plan circulaire. [23]

La ligne verticale représente l'axe d'intensité, tandis que le plan circulaire perpendiculaire montre les teintes et la saturation d'un certain point qui s'y trouve. Les valeurs d'intensité varient en déplaçant le plan circulaire vers le haut et vers le bas de l'axe vertical.

La figure I.21 illustre la vue de dessus du plan précédent, comme on le voit, la saturation peut être définie comme l'amplitude du vecteur joignant le centre et le point spécifié tandis que la teinte peut être déterminée par l'angle mesuré à partir de la ligne de référence rouge.

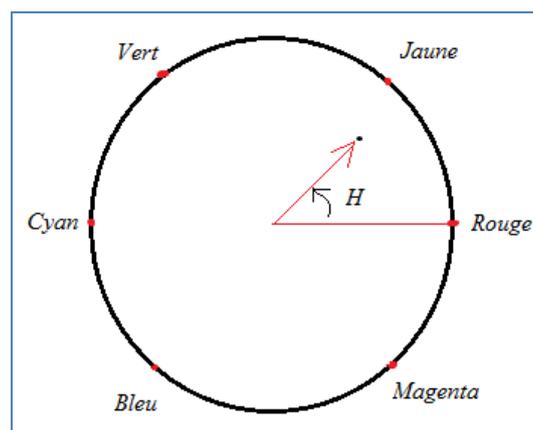
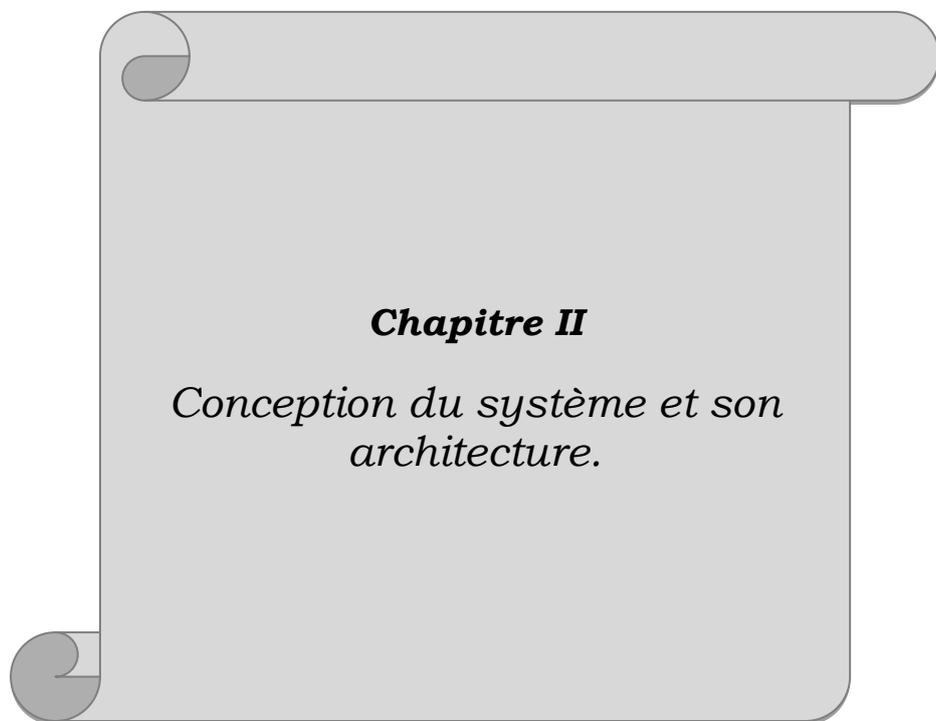


Figure I.21 : Teinte et saturation dans le modèle circulaire HSV. [23]

Conclusion

Dans ce premier chapitre, nous avons présenté un aperçu général sur les systèmes d'inspection industriel par vision par ordinateur, ainsi que les travaux de recherches réalisés dans ce domaine. Nous avons présenté aussi dans ce chapitre quelques techniques de traitement du signal utilisés dans le vision par ordinateur. la théorie derrière les algorithmes de traitement de signal et de vision par ordinateur utilisé dans notre système.

Dans le prochain chapitre nous allons aborder la conception de notre système d'inspection en s'inspirant de la théorie et les travaux présentés dans ce premier chapitre.



Chapitre II

*Conception du système et son
architecture.*

Chapitre II

Conception du système et son architecture

Introduction

Après avoir parlé en général sur les systèmes d'inspection industriel par vision par ordinateur, et présenté les travaux de recherches réalisés dans ce domaine. Nous allons entamer dans ce chapitre la conception globale du système à réaliser en commençant par un schéma synoptique puis la description de sa constitution matérielle et logiciel.

II.1. Conception de système d'inspection

Dans cette section, nous discuterons la conception matérielle et logicielle de notre système d'inspection de bouteilles à réaliser, en présentant tous les équipements et outils logiciels utilisés.

II.1.1. Partie matérielle

La partie matérielle est constituée essentiellement d'un convoyeur électrique doté motoréducteur qui transporte des bouteilles. Pour l'inspection, un capteur IR détecte l'arrivée d'une bouteille, le convoyeur s'arrête alors et une caméra prend une image en temps réel puis l'envoi à la carte Raspberry Pi. La carte Raspberry sera programmée par des algorithmes de traitement d'image qui permettent de d'inspecter le produit et détecter les bouteilles à rejeter. Un mécanisme Poussoir permet de rejeter la bouteille du convoyeur. A la fin du tapis convoyeur, il y aura un autre mécanisme pour le tri de produit en fonction de sa couleur.

Une vue 3D a été réalisée sous le logiciel de conception BLENDER pour donner un premier aperçu notre système à réaliser. Cette modélisation nous permet de visualiser notre système à partir n'importe quel point de vue. De plus, il nous permet d'éditer de manière totalement interactive la forme et la position des éléments constituant notre système global.

La vue tridimensionnelle de notre système réalisé sous le logiciel BLENDER est présentée sur la figure II.1.

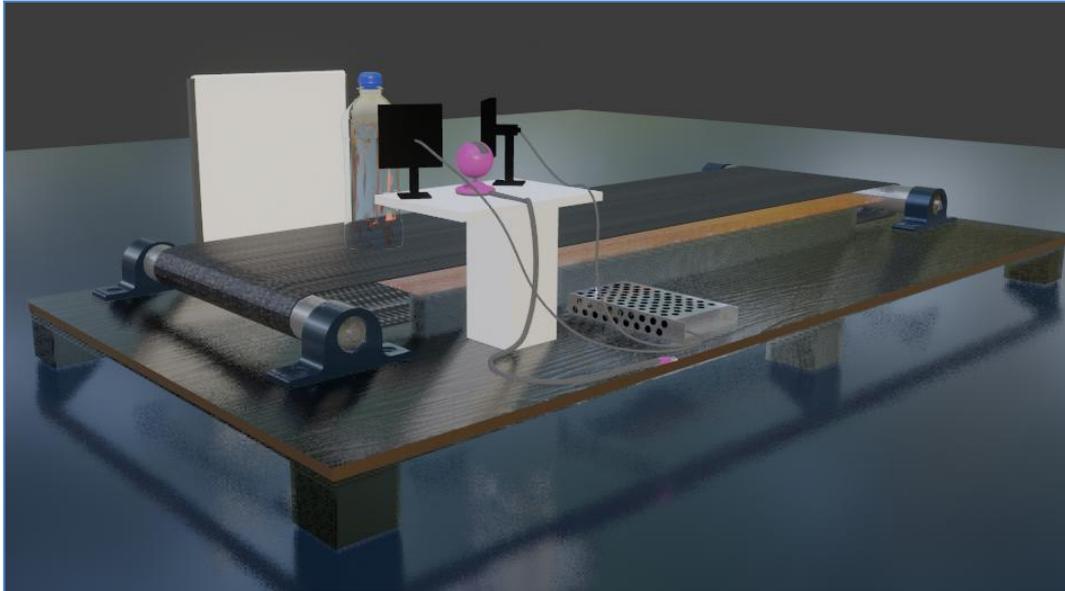


Figure II.1 : Vue 3D du système d'inspection réalisée sur logiciel BLENDER.

Lorsque la bouteille arrive sur le tapis et détectée par le capteur infrarouge, ce dernier envoie un signal à la Raspberry Pi. La Carte Raspberry ordonne donc la camera de prendre une image de la bouteille. Le capteur IR et la camera sont positionnés d'une manière à prendre une image complète et clair de la bouteille sur le tapis.

Après la capture d'image, La carte Raspberry PI applique des algorithmes que nous avons implémentés pour effectuer un filtrage afin d'éliminer le bruit puis applique les algorithmes d'inspection par traitement d'image.

Les paramètres à prendre en considération lors de l'inspection des bouteilles sont les suivants :

- Le bouchon.
- Le niveau de remplissage.

Après avoir appliqué les algorithmes d'inspection, le contrôleur transmet le signal au poussoir, si la bouteille est défectueuse, le poussoir sort et la rejette du tapis convoyeur, sinon, elle continue le chemin pour arriver à la fin du tapis. A la sortie du tapis, Le contrôleur ordonne et actionne aussi le mécanisme de tri afin de séparer les produits en deux catégories selon sa couleur (Orange et raisin).

Le schéma synoptique qui présente la conception et le fonctionnement de système d'inspection à réaliser est décrite dans la figure II.2.

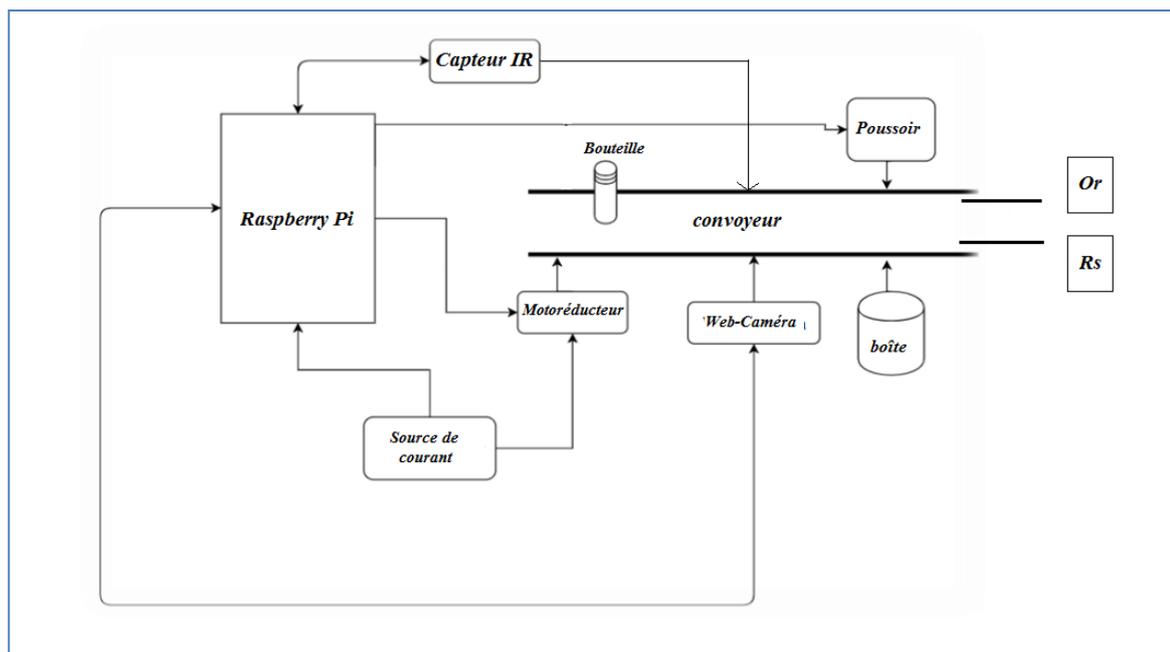


Figure II.2 : schéma synoptique de système réalisé.

II.1.2. Partie logicielle

Cette partie s'intéresse aux outils logiciels utilisés et implantés sur la carte Raspberry PI afin de développer les algorithmes d'inspection et de pouvoir la connectée avec son environnement matériel.

La carte Raspberry pi utilise Ubuntu-mate comme système d'exploitation. Dans ce projet, nous avons opté pour la bibliothèque Open CV de Python qui facilite l'implémentation de différents techniques de traitement de signal, de détection et de reconnaissance d'objets ce qui nous permet à bien mener notre projet.

Une image numérique est prise par la camera en temps réel et ensuite envoyée à la Raspberry Pi pour commencer le traitement et la détection.

Premièrement, le bruit présent dans l'image capturée doit être supprimé. Pour le faire, des algorithmes de filtrage du bruit sont appliqués. Ces algorithmes de filtrage du bruit sont : le filtrage gaussien, le filtrage moyen, le filtrage médian et le filtrage max-min. Parmi ces algorithmes, le mieux adapté à notre application est l'algorithme de filtrage médian.

Après l'élimination du bruit, on passe à la segmentation de l'image. La segmentation est utilisée pour supprimer l'arrière-plan de l'image. Pour la segmentation d'image, l'algorithme de seuil est appliqué pour séparer la région liquide de la bouteille. La détection des contours fait partie de la segmentation de l'image, la détection des bords est utilisée pour détecter les bords

de la bouteille. La détection des bords montre le squelette de l'image ainsi, la partie défectueuse peut être facilement détectée. Cette partie sera détaillée dans le prochain chapitre.

II.2. Architecture du système

II.2.1. Architecture matérielle

Le système global qui comporte le convoyeur, les mécanismes poussoir et de triage, la carte Raspberry Pi 4, la caméra, et les capteurs infrarouges est illustré dans le schéma synoptique de la figure II.3.

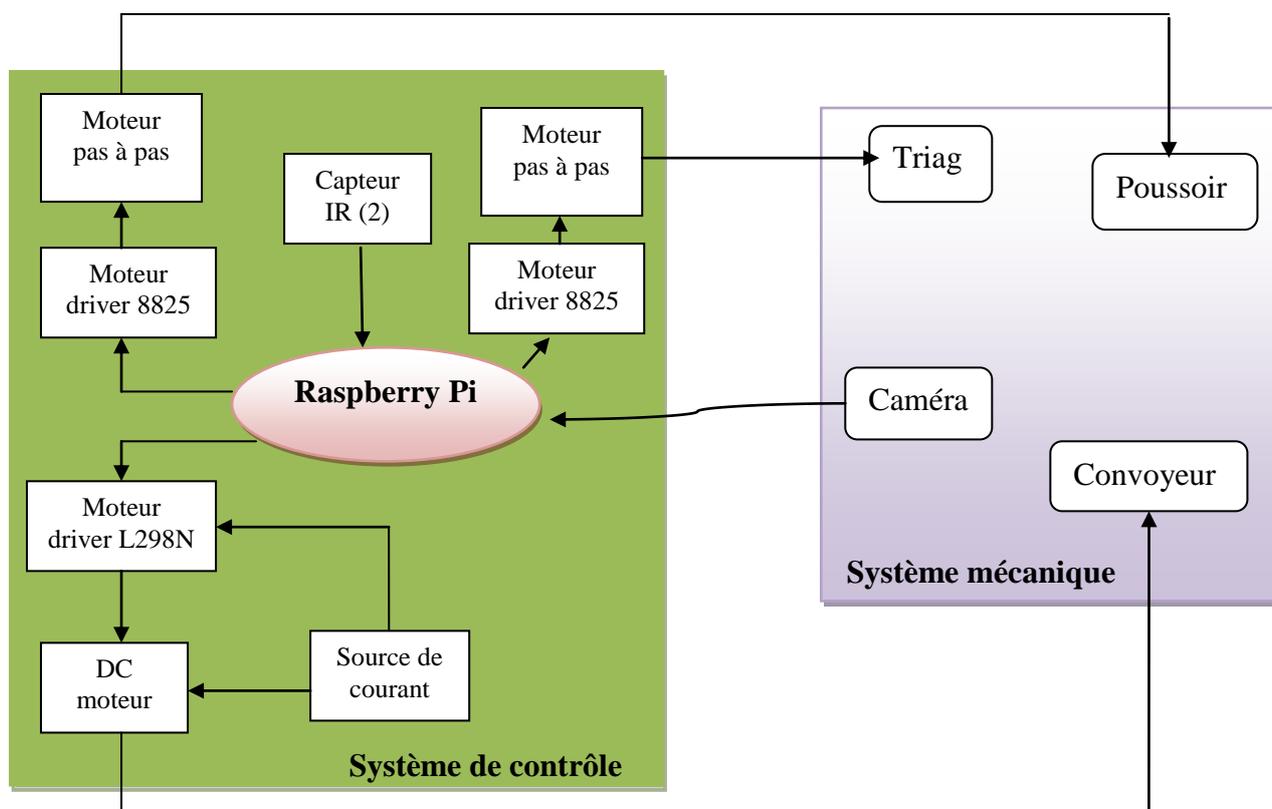


Figure II.3 : Architecture du système.

Dans ce qui suit, nous allons présenter en détail les éléments constituant notre système de d'inspection en tenant compte de leurs caractéristiques et aspect théoriques.

1) La carte Raspberry Pi

Le Raspberry Pi est un nano-ordinateur mono-carte à processeur 'ARM' conçu par des professeurs du département informatique de l'université de Cambridge dans le cadre de la fondation Raspberry Pi [26].

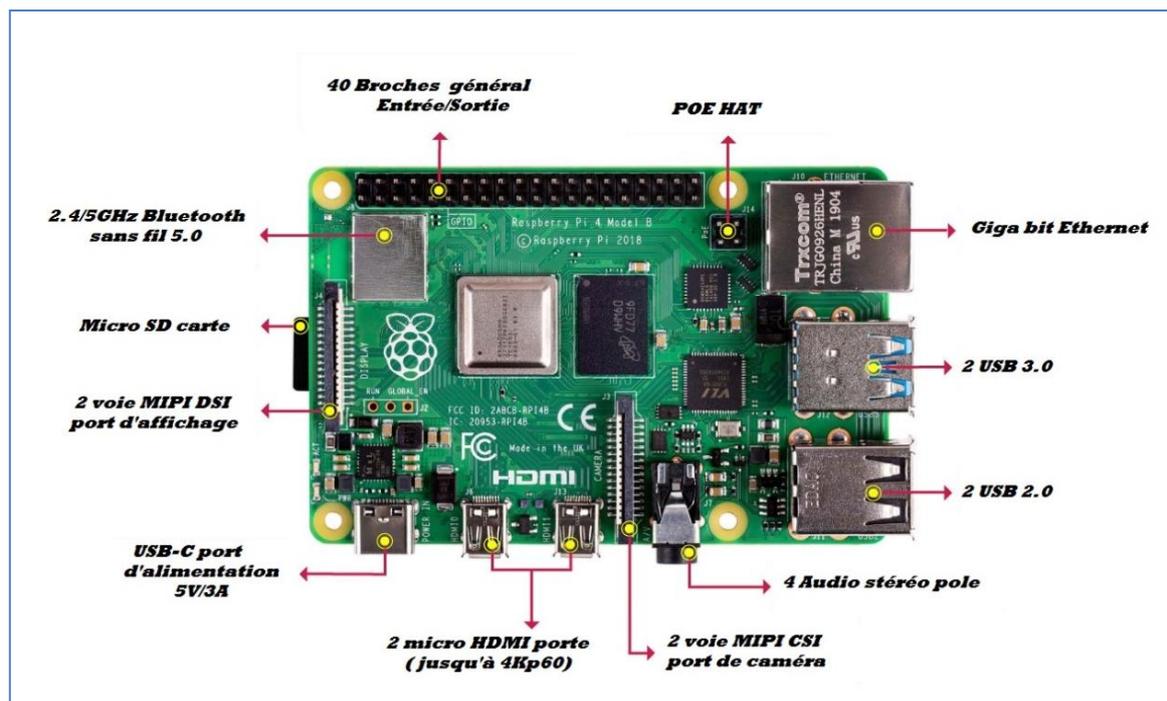


Figure II.4 : La carte Raspberry pi 4

Cet ordinateur, de la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique [26], il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux, notamment Debian, et des logiciels compatibles. Mais il fonctionne également avec le système d'exploitation Microsoft Windows : Windows 10 IoT Core [27], celui de Google Android Pi [28] et même une version de l'OS/MVT d'IBM accompagnée du système APL\360 [29].

2) Moteur a courant continu

Dans ce projet, nous avons utilisé un moteur à courant continu pour l'entraînement du tapis convoyeur.

Le moteur à courant continu, appelé aussi moteur DC (pour Direct Current en anglais), fait partie de la classe des moteurs électriques et sert essentiellement à transformer de l'énergie électrique en énergie mécanique. La plupart des formes de moteurs DC reposent, dans ce contexte, sur les forces magnétiques et disposent de mécanismes internes de types électronique ou électromécanique. Les moteurs à courant continu conventionnels se caractérisent d'abord par le collecteur qui change périodiquement le sens du flux électrique à l'intérieur du moteur. Une des plus puissantes versions du moteur DC classique que l'on retrouve dans de nombreuses applications est le moteur DC sans balais ; celui-ci fonctionne sans frottement et offre donc notamment une durée de vie plus longue.

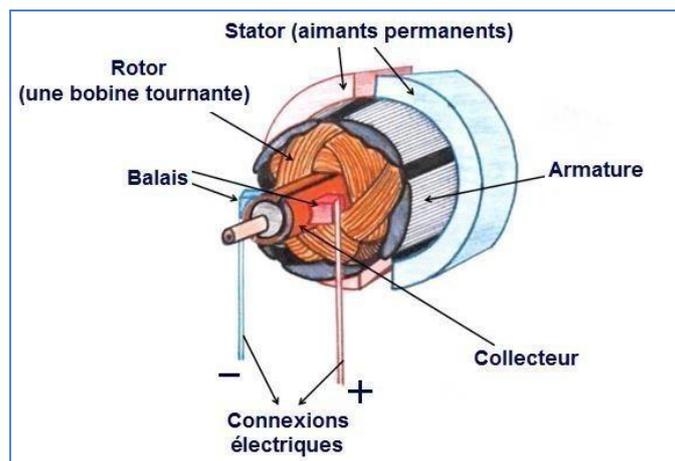


Figure II.5 : Le moteur à courant continu.

3) Le Moteur pas à pas

Deux moteurs pas à pas sont utilisés dans ce projet, ils sont destinés pour actionner les deux mécanismes de poussoir et celui de triage.

Les moteurs pas à pas sont utilisés pour les positionnements angulaires précis (imprimantes, scanners, disques durs ...). Contrairement aux moteurs à courant continu, ils ne nécessitent pas de boucle d'asservissement et sont plus simples à commander.

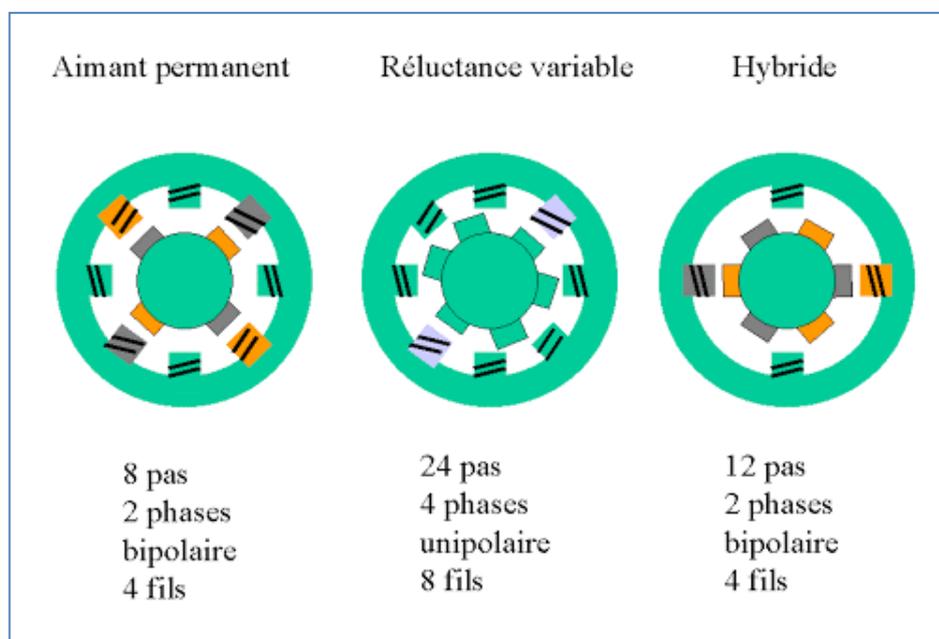


Figure II.6 : Les types de moteur pas à pas .

Il existe trois types de moteur : à aimant permanent, à réductance variable et hybrides. Dans tous les types de moteur, on positionne le rotor en modifiant la direction d'un champ magnétique créé par les bobinages du stator. Ils nécessitent un circuit de commande qui comporte une partie logique et une commande de puissance. La partie logique détermine pour

chaque pas quelles sont les bobines alimentées et le sens de rotation. La fréquence de l'horloge du circuit logique détermine la vitesse de rotation.

4) Circuit de commande de puissance des moteurs

Pour contrôler un moteur à l'aide d'un microcontrôleur ou d'un processeur, nous avons besoin d'un étage de puissance appelé Motor driver ou Motor controller (un pilote). Les pilotes de moteurs sont des petits modules qui permettent d'alimenter en puissance et de contrôler la rotation d'un moteur en fonction des instructions reçues de la carte de pilotage. Ils sont connectés à cette dernière, soit directement, soit par l'intermédiaire d'une interface.

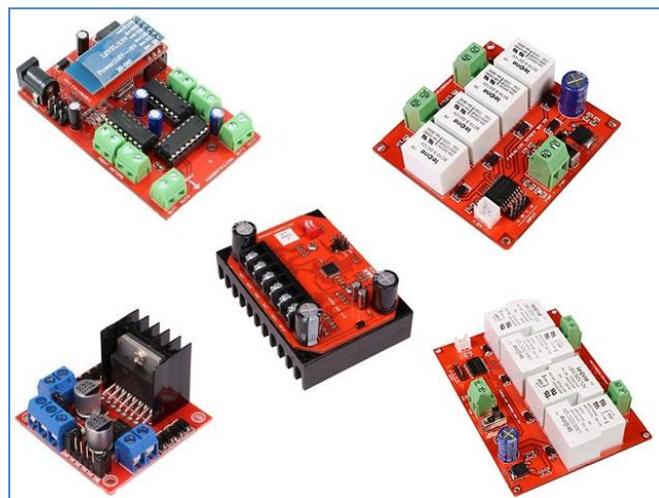


Figure II.7: Different types de Motor driver.

5) Capteur IR

Un capteur infrarouge se présente comme un équipement électronique capable de réagir au rayonnement infrarouge renvoyé par les objets qui lui font face. Ces rayons se situent en dehors du spectre de la lumière visible : ils sont donc invisibles à l'œil nu. Le capteur ne fonctionne pas seul : il est associé à une optique qui se charge de canaliser les rayons. En simplifiant, il ressemble à un capteur d'appareil photo, à ceci près qu'il ne capture pas les images visibles, mais le rayonnement infrarouge. Souvent, le capteur infrarouge est associé à un dispositif électronique chargé de déclencher un signal électrique en cas de variation subite du phénomène mesuré. C'est l'exemple typique du détecteur de mouvements infrarouge. Il peut aussi être associé à une caméra de surveillance, pour apporter à cette dernière une forme de vision nocturne.

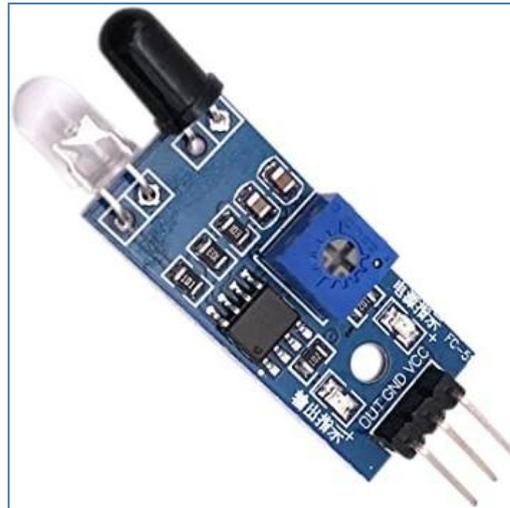


Figure II.8: Module du capteur IR.

II.2.2. Architecture logiciel

Dans la partie logicielle, des algorithmes de traitement d'image sont appliqués pour détecter la bouteille défectueuse se déplaçant sur le tapis. Pour développer ces algorithmes nous avons opter pour le langage Python et la bibliothèque de traitement d'image OpenCV comme cité précédemment.

Le système d'exploitation Ubuntu-mate est installé sur la carte Raspberry Pi 4.

Les logiciels logiciel VNC Viewer, et PuTTY permettent de connecter le Raspberry Pi à l'ordinateur portable et de la programmer sans utiliser un écran, clavier et souris séparés.

1) Système d'exploitation Ubuntu-Mate

Ubuntu-Mate est l'un des systèmes d'exploitation Linux-Distribution. Il est simple à utiliser avec un environnement de bureau traditionnel [30]. Il est open-source et peut exécuter Linux sur presque n'importe quel matériel. Plus précisément, dans ce projet, il est utilisé pour exécuter Linux sur Raspberry pi.

2) Vnc viewer

VNC (*Virtual Network Computing*, littéralement « informatique virtuelle en réseau ») est un système de visualisation et de contrôle de l'environnement de bureau d'un ordinateur distant. Il permet au logiciel client VNC de transmettre les informations de saisie du clavier et de la souris à l'ordinateur distant, possédant un logiciel

serveur VNC à travers un réseau informatique. Il utilise le protocole RFB pour les communications.

VNC est indépendant du système d'exploitation : un client VNC installé sur n'importe quel système d'exploitation peut se connecter à un serveur VNC installé sur un système d'exploitation différent ou identique. Il existe des clients et des serveurs VNC pour la plupart des systèmes d'exploitation. Plusieurs clients peuvent se connecter en même temps à un unique serveur VNC.

Parmi les utilisations de ce protocole, on peut citer le support technique à distance, l'administration et la maintenance de systèmes ou logiciels ne permettant que des contrôles graphiques et demandant l'utilisation de la souris ou bien encore la visualisation distante d'applications diverses et variées, dans un but éducatif par exemple.

3) Putty

PuTTY est un émulateur de terminal doublé d'un client pour les protocoles SSH, Telnet, rlogin, et TCP brut. Il permet également des connexions directes par liaison série RS-232. À l'origine disponible uniquement pour Windows, il est à présent porté sur diverses plates-formes Unix (et non-officiellement sur d'autres plates-formes). PuTTY est écrit et maintenu principalement par Simon Tatham [31]. C'est un logiciel libre distribué selon les termes de la licence MIT.

4) Python

Python est un langage de programmation interprété, de haut niveau et polyvalent. La philosophie de conception de Python met l'accent sur la lisibilité du code avec son utilisation notable d'espaces blancs importants. Ses constructions de langage et son approche orientée objet visent à aider les programmeurs à écrire un code clair et logique pour des projets à petite et grande échelle. [32]

Python est typé dynamiquement et récupéré. Il prend en charge plusieurs paradigmes de programmation, y compris la programmation structurée (en particulier, procédurale), orientée objet et fonctionnelle. Python est souvent décrit comme un langage «avec piles» en raison de sa bibliothèque standard complète. [33]

5) Thony python IDE

Thonny est un environnement de développement intégré pour Python conçu pour les débutants. Il prend en charge différentes façons de parcourir le code, l'évaluation des expressions étape par étape, la visualisation détaillée de la pile d'appels et un mode pour expliquer les concepts de références et de tas.

6) OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque de fonctions de programmation principalement destinées à la vision par ordinateur en temps réel [34]. Développé à l'origine par Intel, il a ensuite été soutenu par Willow Garage puis Itseez (qui a ensuite été acquis par Intel [35]). La bibliothèque est multiplateforme et gratuite pour une utilisation sous la licence open-source Apache 2. À partir de 2011, OpenCV propose une accélération GPU pour les opérations en temps réel [36].

1) Algorithmes de traitement d'image

Des algorithmes de traitement d'image sont appliqués aux images capturées par la caméra pour détecter un défaut dans la bouteille. Tout d'abord, l'extraction de fonctionnalités est appliquée à l'image pour obtenir les fonctionnalités souhaitées. Une fois que cette image est passée à travers les algorithmes de détection de cap, de détection d'étiquette et de détection de niveau. Si un défaut dans la bouteille est trouvé, il est rejeté.

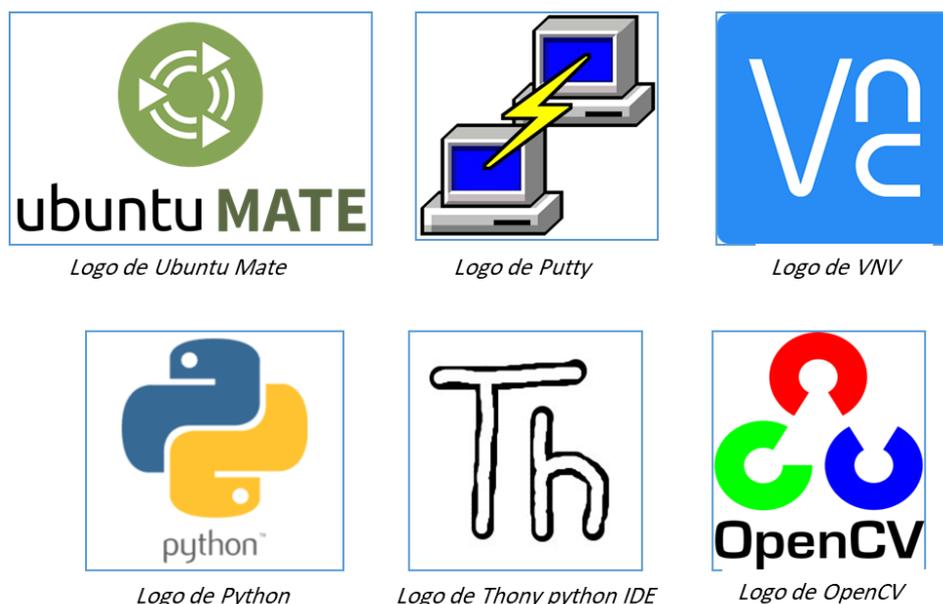
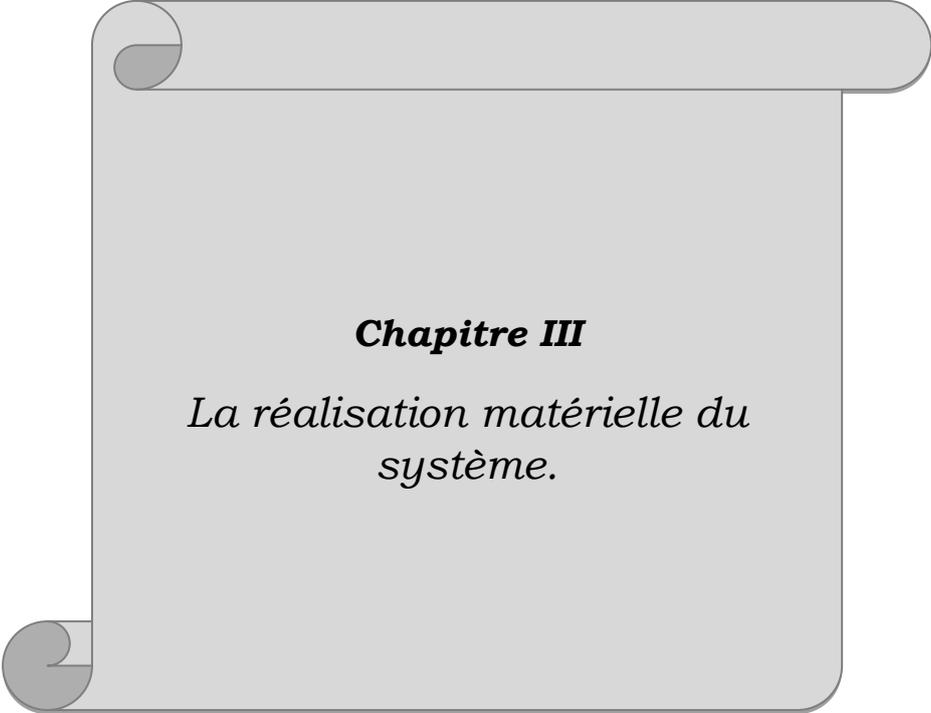


Figure II.9: Logos de différents outils logiciels utilisés.

Conclusion

Dans ce chapitre, on a présenté les différents composants électroniques qui vont être utiles lors de la réalisation de notre projet ainsi leurs définition et caractéristique, on a aussi parlé sur la partie logicielle ou nous avons cité les différents outils utilisés ainsi le langage de programmation et la bibliothèque de traitement du signal adoptés.



Chapitre III

*La réalisation matérielle du
système.*

Chapitre III

La réalisation matérielle du système

Introduction

Grâce au développement de la technologie de nos jours, il existe des milliers de composantes électroniques qui facilitent le développement dans le domaine de la commande électronique et numérique des machines et des systèmes industriels. On se focalise dans ce chapitre sur la réalisation pratique de notre système d'inspection de bouteilles, en se basant sur le schéma synoptique les composants présentés dans le chapitre précédent.

Nous avons mis en œuvre la conception matérielle suggérée à différentes étapes. Ils sont discutés ci-dessous en détail.

III.1. Partie mécanique

III.1.1. Réalisation du Convoyeur

Un moteur à courant continu est utilisé pour convertir l'énergie électrique à courant continu en énergie mécanique. Il peut être contrôlé par une alimentation variable ou directe. Le moteur à courant continu utilisé dans ce projet fonctionne à 24 volts et jusqu'à 4 A de courant. Une alimentation en tension directe de 12 V est fournie pour déplacer le convoyeur à une vitesse modérée.

Nous pouvons donner différentes tensions en fonction de notre vitesse de convoyeur requise. Ici, nous avons utilisé un adaptateur direct de 12 volts, 3 ampères pour actionner le moteur à courant continu du convoyeur. La commande du moteur est dans un seul sens uniquement (convoyeur vers l'avant).

La figure ci-dessous présente l'emplacement du moteur DC au niveau de convoyeur.



Figure III.1 : L'emplacement de moteur DC qui roule le convoyeur.

III.1.2. Réalisation du mécanisme Pousoir

Dans le pousoir on utilise une résolution de pas complet. Sa broche RSP et SLP est connectée à 3,3v des broches Raspberry Pi, STEP et DIR aux broches 21 et 20 respectivement et à la broche de masse avec la broche de masse du contrôleur. Les broches B2, B1, A1 et A2 sont connectées aux 1ère, 2ème, 3ème et 4ème broches du moteur pas à pas.

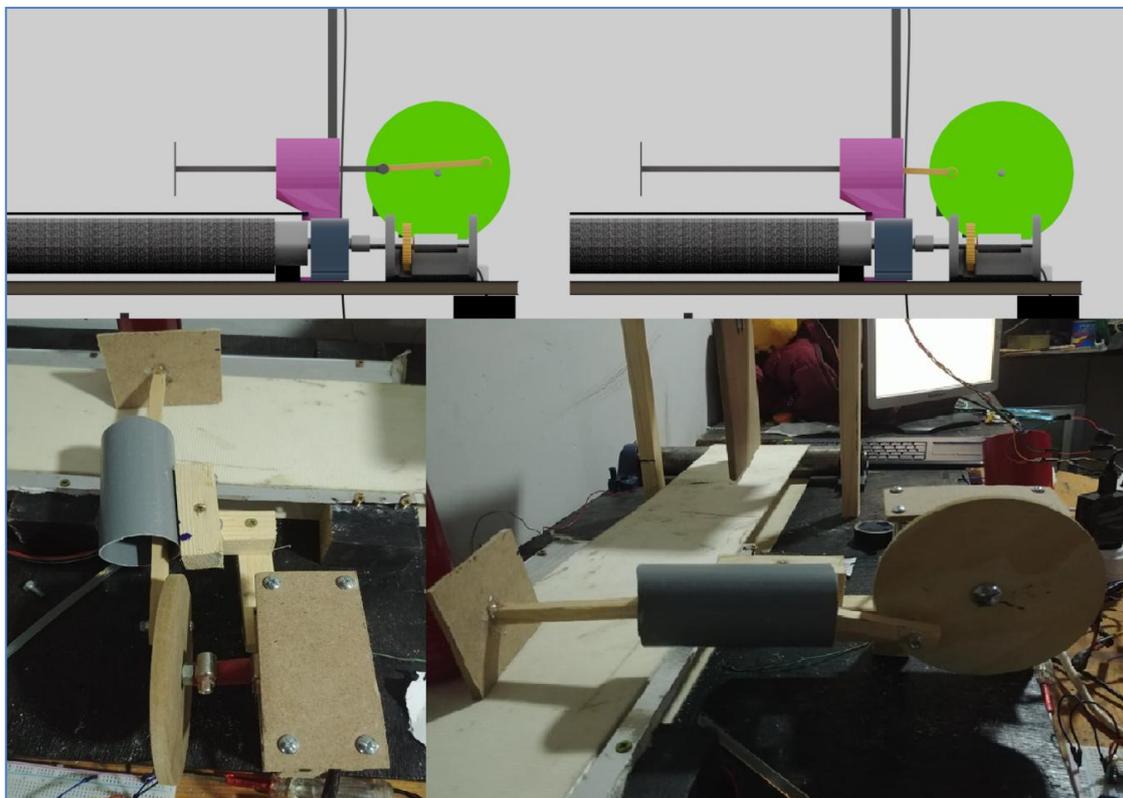


Figure III.2 : Conception et réalisation du mécanisme de pousoir.

III.1.3. Réalisation du système de tri

Drv8825 est utilisé pour trier les bouteilles, sa résolution de pas est de 1/8. Sa broche RSP et SLP est connectée à 3,3v des broches Raspberry Pi, STEP et DIR aux broches 18 et 15 respectivement et à la broche de masse avec la broche de masse du contrôleur. Les broches B2, B1, A1 et A2 sont connectées aux 1ère, 2ème, 3ème et 4ème broches du moteur pas à pas.

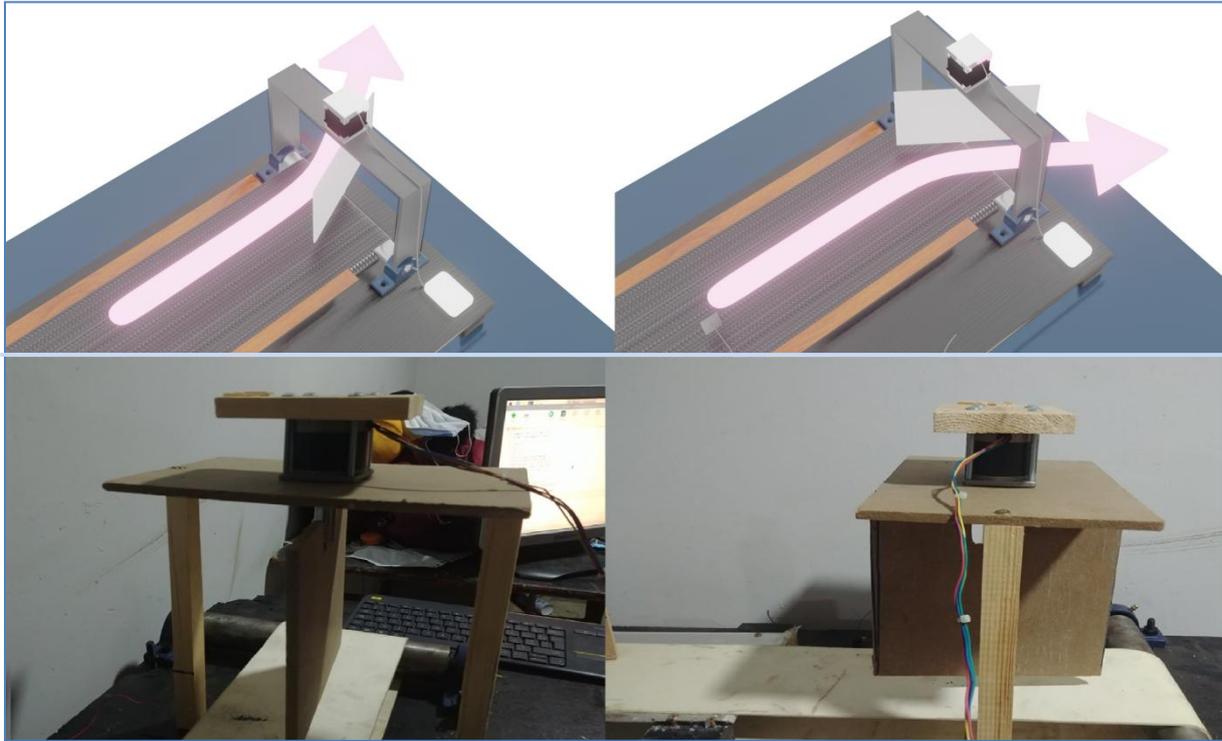


Figure III.3 : Conception et réalisation du mécanisme de tri.

III.2. Partie électronique

L'élément essentiel dans cette partie est la Carte Raspberry PI qui constitue le cerveau de notre système. Tous les autres composants électroniques comme la camera, les drivers des moteurs, et les capteurs IR sont connectés à la Raspberry Pi.

III.2.1. La carte Raspberry PI 4

Raspberry Pi est un mini-ordinateur. Il dispose d'un processeur BCM2711 Quad core Cortex-A72 64 bits, le CPU avec 1.5 GHz et pour Le GPU est également un peu plus rapide et offre 500Mhz. Pour la ram avec 1, 2 ou même 4 Go, la troisième génération n'est disponible qu'avec 1 Go de RAM. Il contient Bluetooth 5.0, il a deux petits connecteurs microHDMI. LAN sans fil et support Ethernet rapide. Il fonctionne sur le système d'exploitation Linux. Il contient 2 ports USB 2.0 auxquels la caméra est connectée.

Comme nous pouvons le voir à la partie supérieure de la figure III.5, les fils sont reliés par des broches. Ces broches sont des broches GPIO. Certaines broches sont des broches GND, d'autres sont des broches TENSION. Il y a deux tensions ici, l'une est 5v qui est la deuxième broche et l'autre est 3,3v qui est la première broche.

Les broches restantes sont des broches de d'Entrée /Sortie générales sur lesquelles les autres composants sont connectés. Le fil blanc doré est un adaptateur secteur 2,2 V. Sur la partie droite, des fils noirs sont connectés aux ports USB.



Figure III.4 : Branchement de la Raspberry Pi.

III.2.2. Driver (pilote) du moteur pas à pas

Drv8825 est un circuit de commande de moteur pas à pas bipolaire. Il contient un indexeur micro-pas à pas et deux pilotes de pont en H. Il fonctionne généralement de 8,5 volts à 45 volts et peut produire environ 1,6 ampères de courant sans aucune chute de chaleur ni flux d'air [23]. Le pilote permet de faire fonctionner le moteur pas à pas jusqu'à une sortie de 2,2 ampères de courant par bobine. Sa principale caractéristique est qu'il peut contenir six résolutions de pas (1/4 pas, pas complets, demi-pas, 1/32 pas, 1/16 pas, 1/8 pas).

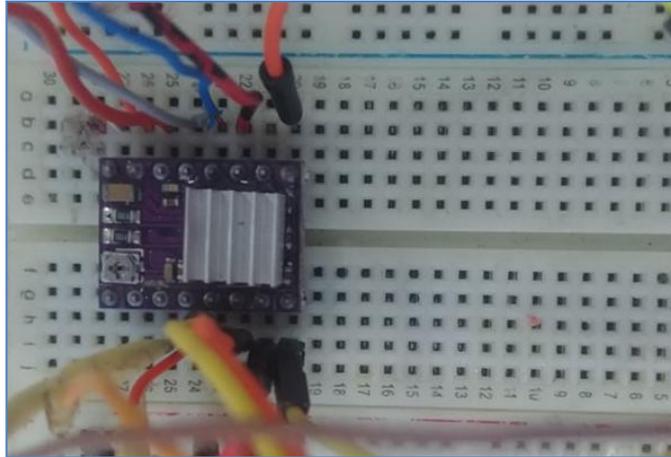


Figure III.5 : Branchement du Circuit DRV8825.

III.2.3. Driver (pilote) du moteur a courant continu

Le L298N permet de contrôler 2 moteurs continus ou 1 moteur pas à pas. Il peut faire tourner les moteurs en vitesse fixe ou variable par la commande PWM. De plus, il inclut des diodes pour protéger le circuit, des résistances de rappel et un dissipateur de chaleur en cas de forte charge.

Les ports ENA et ENB permettent de gérer l'amplitude de la tension délivrée au moteur, grâce à un signal PWM. Les ports In1, In2 pour le moteur A et In3, In4 pour le moteur B, permettant de contrôler le pont en H et par conséquent le sens de rotation des moteurs.

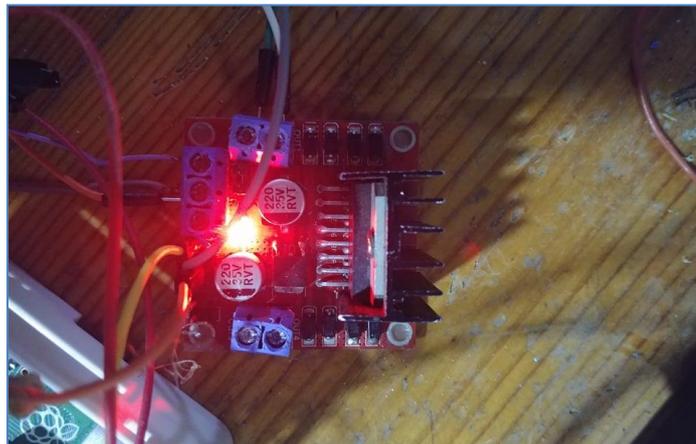


Figure III.6 : Le module L298N et son branchement.

Le L298N est branché comme suite : 12 v avec l'alimentation, le premier GND avec GND de l'alimentation, le deuxièmes fil de GND avec GND de Raspberry Pi. Pour les portes : ENA avec la 24eme broche de GPIO, IN1 avec la 23eme broche de GPIO, IN2 avec la 25eme broche de GPIO, OUT1 avec le sens positive (+) de moteur DC et pour OUT2 avec le sens négative (-) de moteur DC.

III.2.4. Le Capteur IR

Le 1^{er} capteur IR est utilisé ici pour la détection des bouteilles. Il est aligné devant la caméra de telle manière que lorsque la bouteille arrive devant lui, il détecte et envoie un signal à la Raspberry pour capturer l'image réelle à cet instant. Le capteur IR a trois broches, la première pour VCC, la deuxième pour GND et la troisième est la sortie numérique qui indique la présence de l'objet.

La broche VCC est connectée à la 2eme broche de la Raspberry Pi, c'est-à-dire 5V, la broche GND est connectée à la broche GND de la Raspberry Pi qui est la 6eme broche et la broche de sortie est connectée à la 17eme broche GPIO de la Raspberry Pi. Cette broche est configurée comme broche d'entrée à la Raspberry.

La position du capteur IR est très importante ici car en raison de cette image est capturée à un certain point. Si d'une manière ou d'une autre, il y a un retard dans la capture d'image qui provoque un danger dans la sortie du traitement d'image.

Le 2eme capteur IR est branché de même façon que le 1^{er} mais pour la broche de sortie de contrôle est connectée à la 14eme broche de GPIO de la Raspberry Pi. Ce capteur est positionné devant le mécanisme poussoir, ce qui permet de détecter la présence de la bouteille devant lui afin d'arrêter le tapis et rejeter la bouteille dans le cas de la réception de l'ordre par la Raspberry.

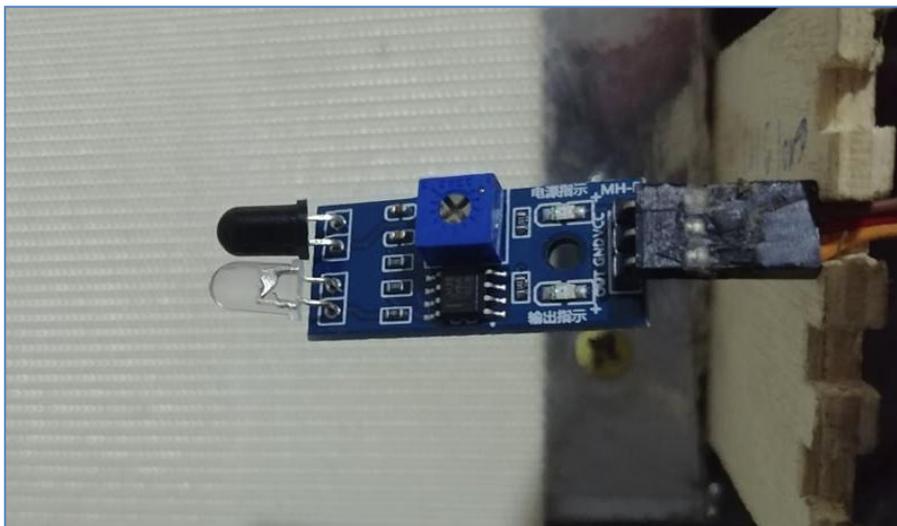


Figure III.7 : Branchement et positionnement du Capteur IR.

III.2.5. La camera (Webcam)

Une webcam est une caméra conçue pour être utilisée comme un périphérique d'ordinateur, et qui produit une vidéo dont la finalité n'est pas d'atteindre une haute qualité, mais de pouvoir être transmise en direct au travers d'un réseau, typiquement Internet.

La webcam utilisée pour l'acquisition d'images est de type USB-VGA avec une résolution de (640×480).



Figure III.8 : La webcam utilisée dans le projet.

III.3. Branchement final de la partie électronique

Après avoir présenté tous les éléments électroniques qui constituent le système d'inspection, nous présentons dans cette partie le schéma de branchement global des composants électroniques avec la carte Raspberry.

Le schéma de câblage présenté sur la figure II.9 est réalisé à l'aide de logiciel Fritzing.

Ce dernier est un logiciel libre, une initiative open source destiné pour le développement et la conception CAO de matériel électronique, afin de soutenir les concepteurs et les artistes prêts à passer de l'expérimentation d'un prototype à la construction d'un circuit plus permanent.

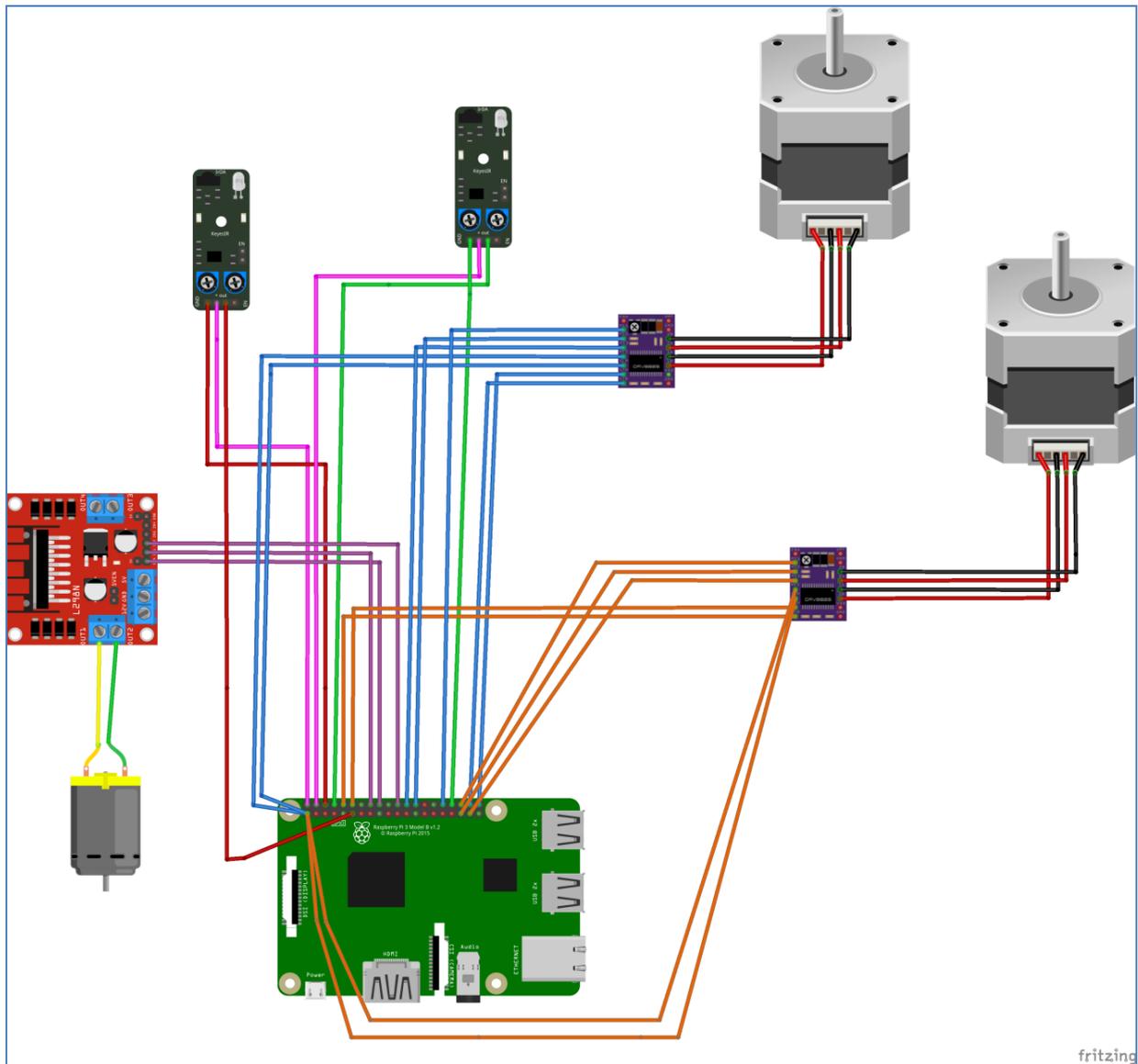


Figure III.9 : Câblage global des composants électroniques.

III.4. Réalisation finale du système d'inspection

Après avoir présenté les parties mécaniques, électronique, on arrive à la présentation de notre système d'inspection réalisé. La figure III.10 qui suit présente une visite de la conception 3D proposée est modélisée sur le logiciel BLENDER, alors que celle qui vient après (figure III.11) présente l'image réelle de la réalisation du système.

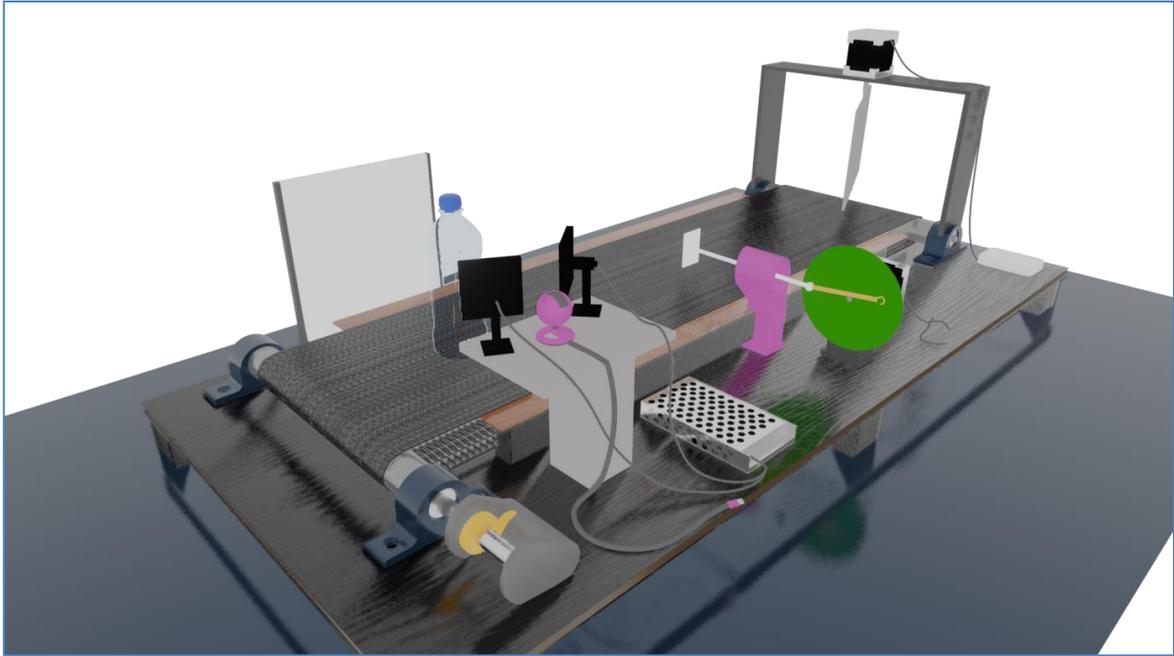


Figure III.10 : Visualisation de la mise en œuvre globale des matériels du système.



Figure III.11 : Vue globale du notre système.

III.5. Evaluation de la réalisation matérielle

III.5.1. La Caméra avec le capteur IR

La caméra et le capteur IR ont été correctement placés sur le convoyeur, de sorte que lorsqu'une bouteille arrive sur le convoyeur, elle est immédiatement détectée par le capteur et la caméra capture l'image. L'image est ensuite envoyée aux algorithmes de traitement d'image pour un traitement ultérieur.

Le fond est considéré comme statique. Nous avons utilisé un fonds statique blanc pour éviter le bruit de fond d'image rugueux. Si nous n'avons pas utilisé cet arrière-plan statique, nous pourrions avoir plusieurs pixels de bruit dans l'image pouvant entraîner des résultats erronés. Les effets de lumière ont été pris en compte pour éviter les variations de teinte des bouteilles en cas de bouchon et de niveau car ceux-ci sont de couleurs différentes et une légère variation de lumière provoquait un changement dans les gammes de couleurs.

III.5.2. Le mécanisme Poussoir

Le mouvement du mécanisme poussoir était contrôlé par un moteur pas à pas. Six filaires de moteurs pas à pas ont été utilisées, dont quatre fils et deux paires communes; contenant chacun 3 fils, ont été fabriqués. Sur les deux paires, les fils de résistance les plus élevés ont été sélectionnés et connectés au pilote Drv8825. Il contient six résolutions de pas (pas complets, demi-pas, 1/4 pas, 1/8 pas, 1/16 pas, 1/32 pas) et nous avons utilisé une résolution de pas complet pour un mouvement rapide du poussoir. Le délai entre chaque étape est de 0,28s. Lorsque les pas en avant et en arrière étaient égaux, le poussoir ne se déplaçait pas complètement en avant et en arrière. Si le nombre de pas vers l'arrière était supérieur aux pas vers l'avant, le poussoir ne s'est pas déplacé complètement vers l'avant et le moteur continu de tourner jusqu'à un certain temps même après l'arrêt du poussoir. Cela a réduit l'efficacité. Ainsi, 930 pas en avant et en arrière ont été sélectionnés pour éviter les problèmes évoqués ci-dessus.

III.5.3. Convoyeur avec le moteur à courant continu

Nous travaillons sur un convoyeur en mouvement continu connecté avec un moteur à courant continu, fournissant 12V. Le convoyeur est actionné en continu à une vitesse modérée pour éviter les retards. Comme la bande transporteuse mesure environ 20 cm de largeur et 100 cm de longueur, nous devons respecter le temps de passage des bouteilles, c'est-à-dire on dépose la bouteille dans le début du convoyeur qu'est lui-même en marche, le capteur IR

capte la bouteille, la camera prend une image et l'envoi au Raspberry Pi, quand la bouteille dépasse la camera on peut mettre une autre bouteille. Cela pour obtenir des meilleurs résultats sinon, à mesure que la distance par rapport à la caméra change, les résultats changent en conséquence. Ainsi, lorsque nous avons un point précis où les bouteilles arrivent, la distance entre la caméra et la bouteille reste presque la même, ce qui rend les résultats plus précis.

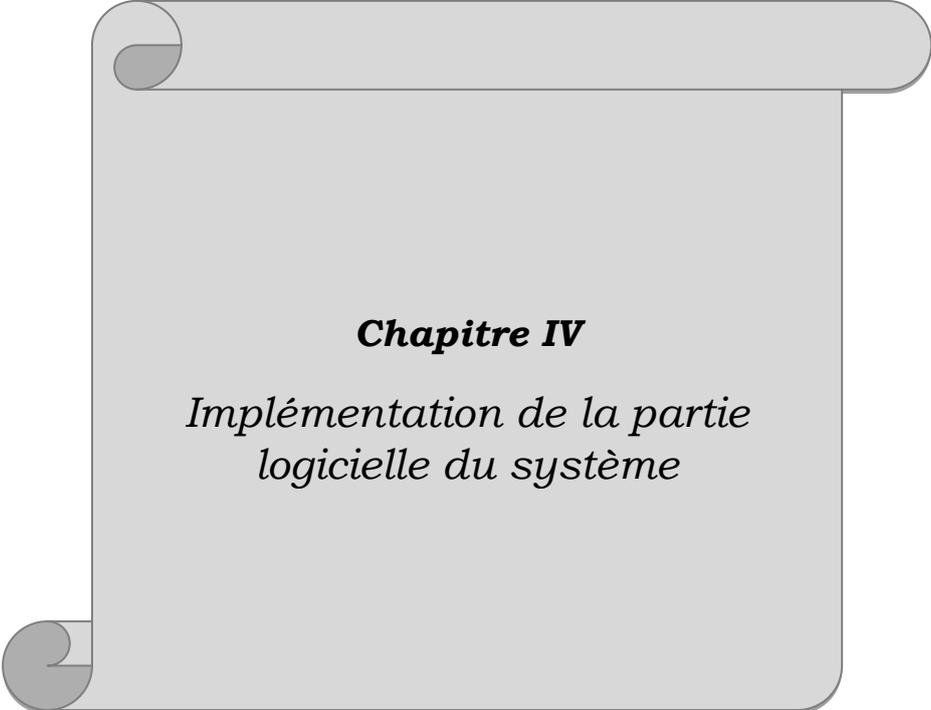
III.5.4. Le mécanisme de triage

La camera quand elle détecte les couleurs, plus précisément le contour de couleur. Si le contour zéro = 1, c'est-à-dire la couleur orange, une décision sera prise telle que la direction égale à zéro (gauche) et le moteur tourne avec une résolution de 1/8. Sinon le contour deux = 1, c'est-à-dire la couleur mauve, une décision sera prise telle que la direction égale à un (droite) et le moteur tourne avec une résolution de 1/8. Dans les deux cas, le moteur revient à sa position initiale après avoir tourner vers la gauche ou la droite.

Conclusion

Dans ce chapitre nous avons exposé la réalisation matérielle et les composants électroniques utilisés dans notre système d'inspection. Nous avons présenté la réalisation de la partie mécanique ainsi que schéma de branchement global du circuit électronique. Une évaluation des parties réalisées a été aussi pressée à la fin de ce chapitre.

Dans le prochain chapitre nous allons entamer à l'implémentation logiciel, c.à.d. la les différents algorithmes et organigramme du code a implémenter sur la Raspberry PI.



Chapitre IV

*Implémentation de la partie
logicielle du système*

Chapitre IV

Implémentation de la partie logicielle du système

Introduction

Tous système a base de commande électronique numérique constitué d'un ensemble de composants et modules électroniques, nécessite une partie programmation à implémenter sur le contrôleur ou le processeur principal.

Jusqu'à présent, nous avons discuté de la partie matérielle et de sa mise en œuvre de notre système d'inspection de bouteilles. Nous nous dirigeons maintenant vers la partie logicielle. Cette partie que nous allons présenter dans ce chapitre constitue la partie principale de notre projet car elle traite de l'algorithme de traitement d'image, de ses techniques et du langage python que nous avons utilisé.

IV.1. L'organigramme de fonctionnement général du système

La bouteille étant sur le convoyeur en mouvement, quand elle arrive devant le 1er capteur IR la Camera prend une photo et l'envoie au Raspberry Pi, la bouteille continue son parcours au même temps le traitement d'image se fait au niveau de la Raspberry, si la bouteille n'a pas du bouchon ou si le niveau de remplissage n'a pas atteint la valeur requise donc la bouteille sera dégagée par le poussoir, mais si elle est valide, elle continue son chemin. Au même temps dans le traitement d'image, la couleur du produit (boisson) sera détectée pour reconnaître le type de produit, à la sortie du convoyeur se fera le tri de produit : si le liquide de la bouteille est orange donc elle va être orientée vers la gauche, par contre si le liquide de la bouteille est mauve, elle sera orientée vers la droite, cette action sera réalisée par le mécanisme de triage qui tourne à droite ou à gauche.

Ce processus de fonctionnement est présenté dans l'organigramme de la figure IV.1.

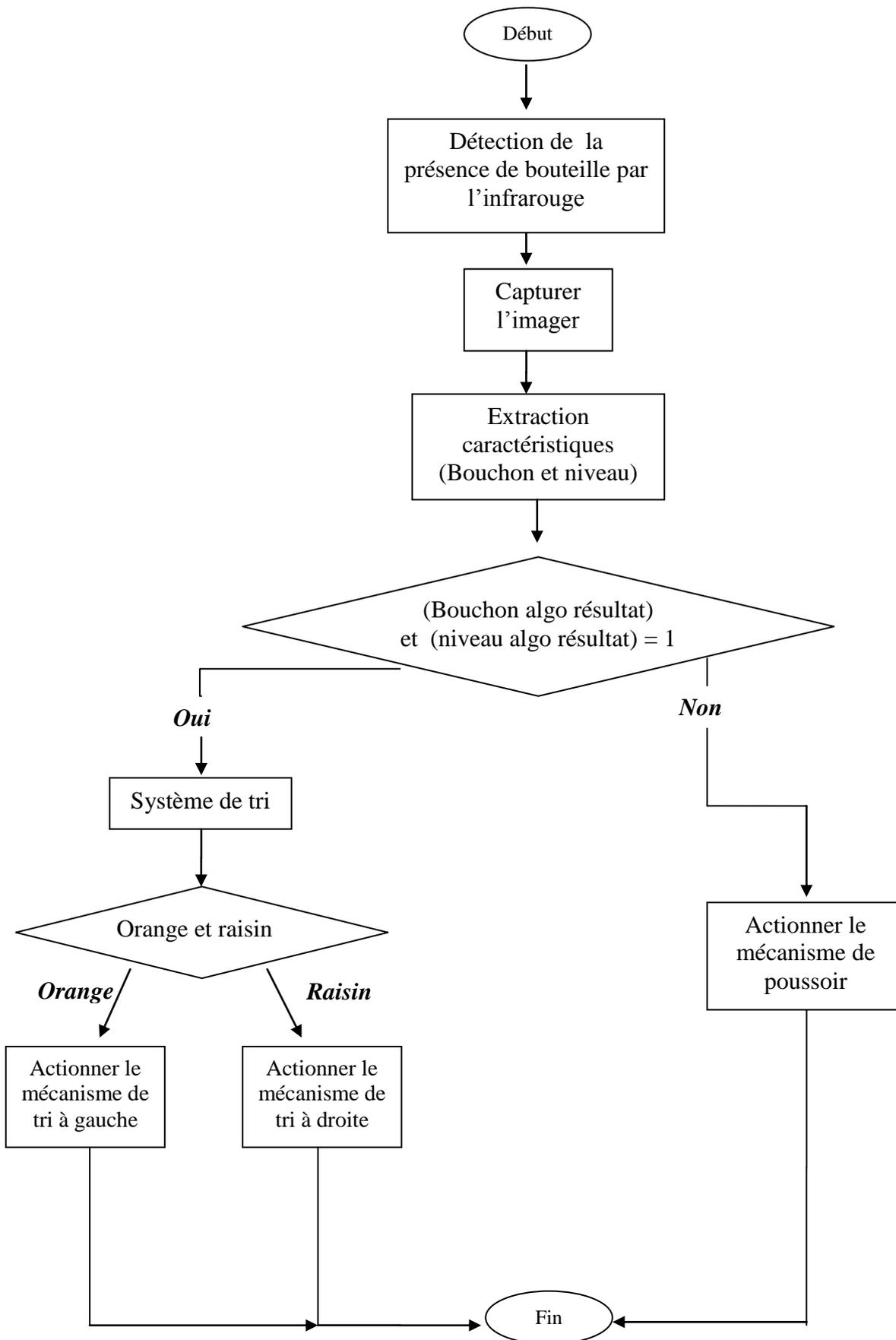


Figure IV.1: Software architecture.

IV. 2. Implémentation de l'algorithme

Comme nous l'avons vu précédemment dans le chapitre 2, nous avons opté pour le langage python dans notre projet. Pour les tests primaires et le développement du programme, nous avons utilisé le logiciel Paycharm sur Windows qui prend en charge python.

Nous avons d'abord installé les bibliothèques nécessaires telles qu'OPEN CV, MATPLOTT LIB, TIME et Numpy. Sans ces bibliothèques, nous ne pouvons pas travailler sur le traitement d'image.

Après avoir développé et testé le programme sur un ordinateur Windows, nous l'avons déplacé sur la Raspberry pi car c'est seulement là que nous pouvions tester pratiquement l'ensemble du système d'inspection. Comme sur Windows, nous avons également dû télécharger les mêmes bibliothèques de traitement d'image et la version python 3.7.3 est utilisée. Sur Raspberry pi, nous avons utilisé le système d'exploitation UBUNTO MATE. Comme il s'agit d'un système d'exploitation basé sur Linux, toutes les bibliothèques ont été téléchargées à l'aide du terminal du système d'exploitation.

Nous utilisons des bouteilles de jus d'orange et de raisin dans les tests pratiques de notre système d'inspection.

Comme notre objectif principal est l'inspection de bouteilles de boissons en deux points de vue : détection de problème du bouchon (absent ou mal fixé), et détection de niveau de liquide dans la bouteille, Nous allons discuter dans ce qui suit ces deux tâches séparément.

IV.2.1. Inspection du bouchon de bouteille

Le premier objectif qui est l'inspection de bouchon défectueux. Tout notre traitement est en temps réel et sur un convoyeur en mouvement sans l'arrêter. Lorsque les bouteilles passent sur le convoyeur une par une, une image est capturée pour chaque bouteille en temps réel est envoyée au processeur Raspberry pour le traitement.

Les deux bouteilles de jus ont des couleurs différentes mais portent le même bouchon, donc ils seront traités de même façon.

Pour l'inspection du bouchon, la première étape consiste à la détection la zone de bouchon, qui est nécessaire pour la détection si elle est défectueuse ou non. Nous avons utilisé la transformation BGR en HSV pour détecter la couleur spécifique de la bouteille, puis en fonction de cette couleur, un traitement supplémentaire est effectué.

Un test pratique qui illustre le résultat de ce traitement est présenté dans la figure IV.2.

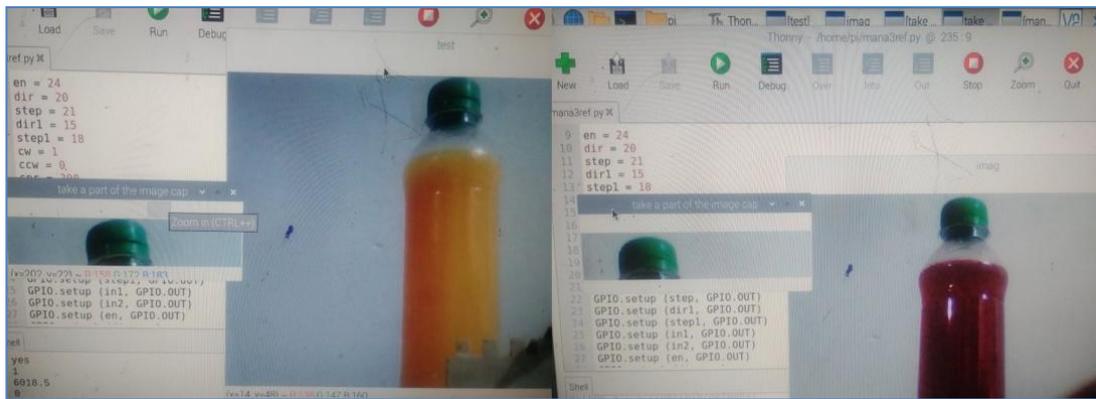


Figure IV.2 : Zone de bouchon.

La deuxième étape est l'identification de l'espace colorimétrique. Pour notre bouteille, nous savons que la couleur du bouchon est le vert, nous avons donc défini une gamme de couleurs verts pour HSV. Fondamentalement, cela est fait pour identifier si le bouchon de la bouteille est présent ou absent, cela peut être facilement identifié à l'aide de cette technique.

Si une couleur spécifique mentionnée dans la plage est présente sur cette zone spécifique en vedette, cela en donne une image binaire. Nous pouvons ensuite superposer cette image binaire à l'image colorée d'origine pour obtenir une région colorée. Ceci est également appelé masquage.

L'image du masque est essentiellement l'image binaire de bouchon et seule la zone en surbrillance est dans la plage de couleur vert. Après le masquage avec l'image d'origine, nous obtenons une couleur verte spécifique du bouchon.

Une fois que nous obtenons cette image, nous pouvons maintenant identifier le statut de bouchon car s'il y a un problème ou si le bouchon n'est pas présent, la couleur spécifique de bouchon n'est pas détectée. Avant d'entrer dans les résultats ou de prendre des décisions, certains facteurs affectent l'image. Cela s'appelle le bruit. Le bruit est présent dans l'image binaire. Même un petit bruit entraîne des problèmes de sortie et les résultats appropriés ne sont pas calculés. Donc, pour contrôler le bruit, nous avons utilisé des opérations morphologiques. Il comprend l'érosion qui minimise le petit bruit dans l'image. Et puis une dilatation qui se vante essentiellement de l'image restante signifie que la zone blanche est élargie. Cela aide essentiellement à obtenir maintenant des limites appropriées de bouchon.

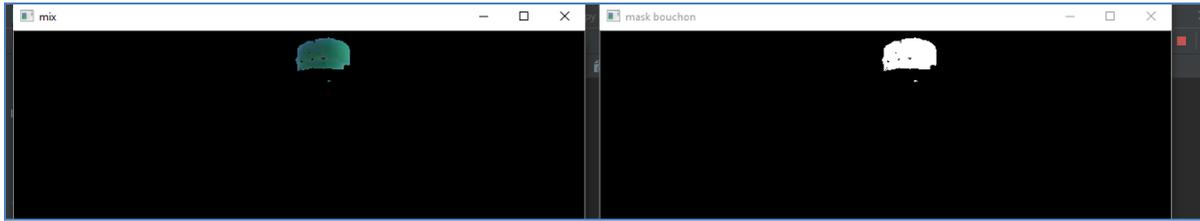


Figure IV.3 : image masquée HSV.

Ainsi, après avoir supprimé le bruit, nous obtenons une image segmentée HSV claire et sans bruit de notre zone de bouchon requise. Maintenant, nous avons des limites claires et nettes de bouchon que nous pouvons utiliser pour comparer avec une valeur de limite de bouchon précise pour identifier si notre bouchon est correct ou s'il y a un problème tel que le bouchon n'est pas présent, etc.

Donc, à cette fin, nous avons utilisé la fonction de contour qui est appliquée à une image masquée sans bruit. Le contour est effectué automatiquement et ne crée qu'une limite autour d'arêtes spécifiques. Une fois l'image est contournée, nous utilisons la fonction de zone de contour pour trouver la zone de contour. Cette valeur est ensuite comparée à une plage donnée de bouteille précise. Si la valeur mesurée est hors de la plage, cela signifie que le bouchon n'est pas présent.

L'organigramme de l'algorithme de détection de bouchon est présenté ci-dessous

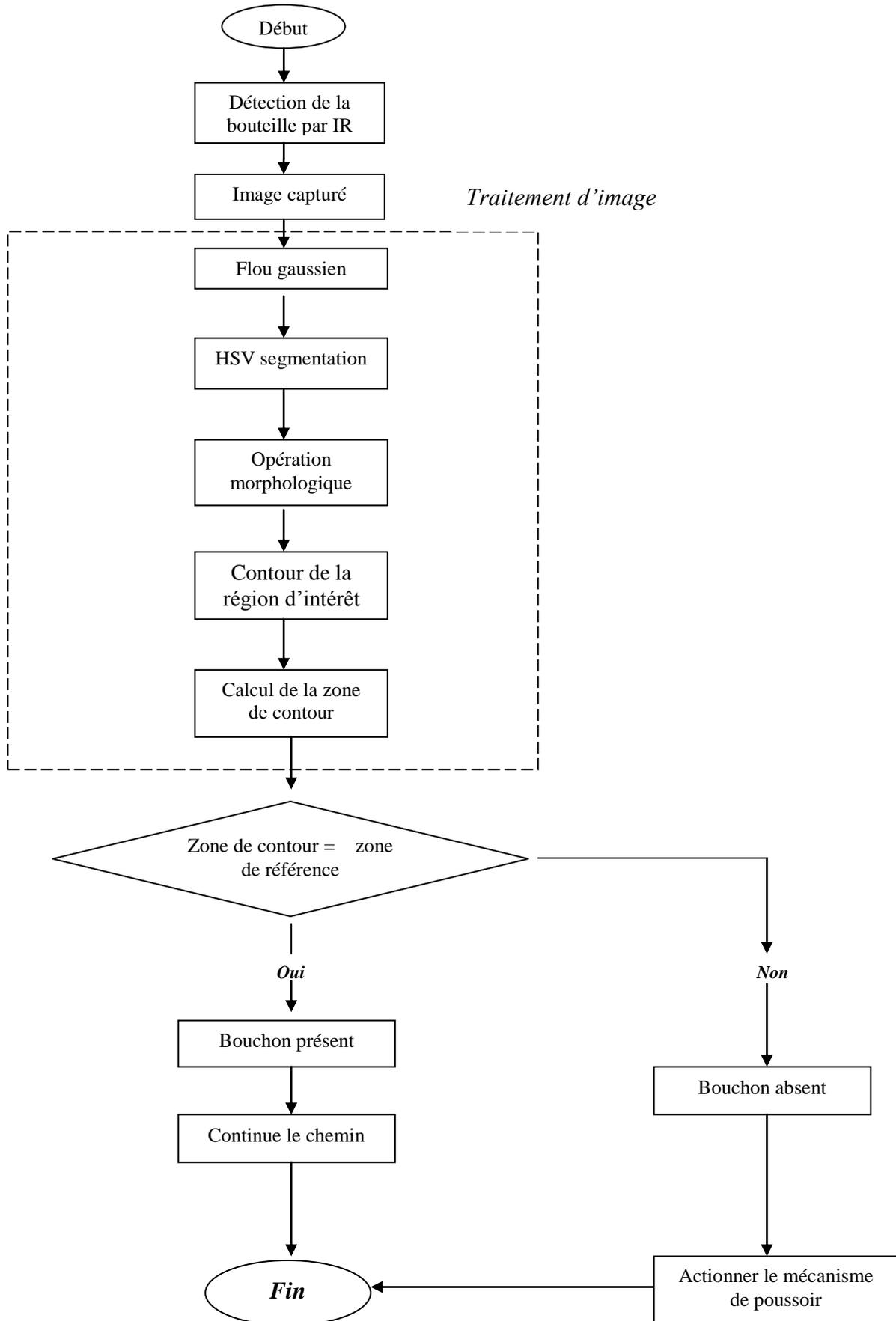


Figure IV.4: Algorithme de détection de bouchon.

IV.2.2. Détection du niveau la bouteille

Le deuxième objectif était d'éliminer les bouteilles qui ont un niveau de remplissage inadmissible. Comme indiqué précédemment, nous utilisons des bouteilles de jus pour l'inspection. Tout le traitement se fait en temps réel sur un convoyeur en mouvement.

Après la capture d'image, le traitement est effectué par des espaces colorimétriques.

Pour détecter le niveau, nous utilisons la segmentation des couleurs. Dans la segmentation de couleur, l'objet peut prendre valeurs comprises entre 0 et 255, où le noir est représenté par (0, 0, 0) et le blanc est représenté par (255, 255, 255). RVB est un espace colorimétrique supplémentaire dans la segmentation des couleurs et d'autres couleurs peuvent être supposé et produit à partir de quantités brillantes de lumière rouge, bleue et verte sur un noir contexte. RVB est utilisé pour définir la teinte, la saturation et la valeur. La teinte d'une couleur réfère à une couleur pure qui lui ressemble, la saturation représente le blanc de la couleur et la valeur représente l'intensité ou la lumière.

Dans notre cas, nous convertissons RVB en HSV pour détecter la bonne couleur du liquide dans les bouteilles. Pour La couleur liquide de jus d'orange est orange et pour jus de raisin, la couleur est mauve. Pour une détection correcte des liquides, HSV est supérieur et inférieur sur des plages avec des valeurs définies.

Au début, nous avons défini la zone de liquide où nous voulions inspecter le niveau.

La figure suivante présente un essai pratique afin d'avoir une bonne compréhension de l'idée de détection du niveau de liquide dans la bouteille.

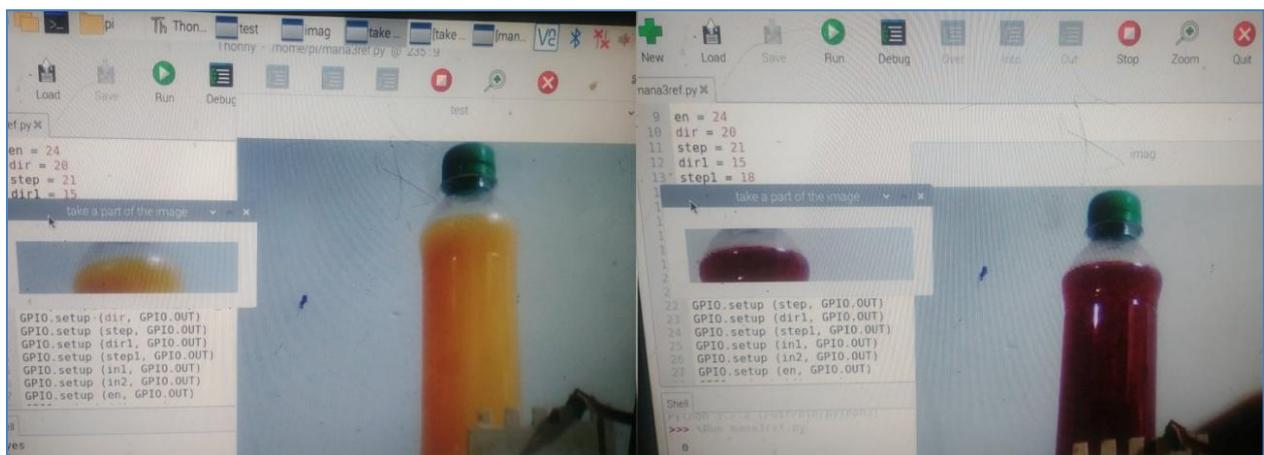


Figure IV.5 : Zone de liquide.

Après avoir défini la zone, nous avons supprimé le bruit en utilisant un filtre de flou gaussien, puis nous l'avons converti en espace colorimétrique HSV. En HSV, nous avons défini la plage supérieure et inférieure pour la couleur orange. En les combinant, nous avons créé une image masquée. L'image masquée est simplement une image dans laquelle certaines des valeurs d'intensité des pixels sont nuls et d'autres non nuls. Si la couleur mentionnée est présente dans cette zone spécifiée, cela donne une image binaire et nous pouvons les superposer.

Après avoir obtenu ces images et valeurs, nous avons effectué le contour sur l'image masquée. Le contour crée une limite autour des bords de l'image masquée. Il a une fonction de longueur dans laquelle nous pouvons calculer la longueur ou la taille du contour. Après le contour, nous avons appliqué la fonction de longueur de contour (Lengh). Length a renvoyé une valeur, puis nous avons comparé la valeur avec la plage spécifiée. Si la valeur se situe dans la plage, le niveau du liquide est correct, sinon il n'est pas correct. Le contour sur un liquide précis a une grande limite de contour, de sorte que la longueur de ce contour doit être plus grande que le contour sur un liquide non précis.

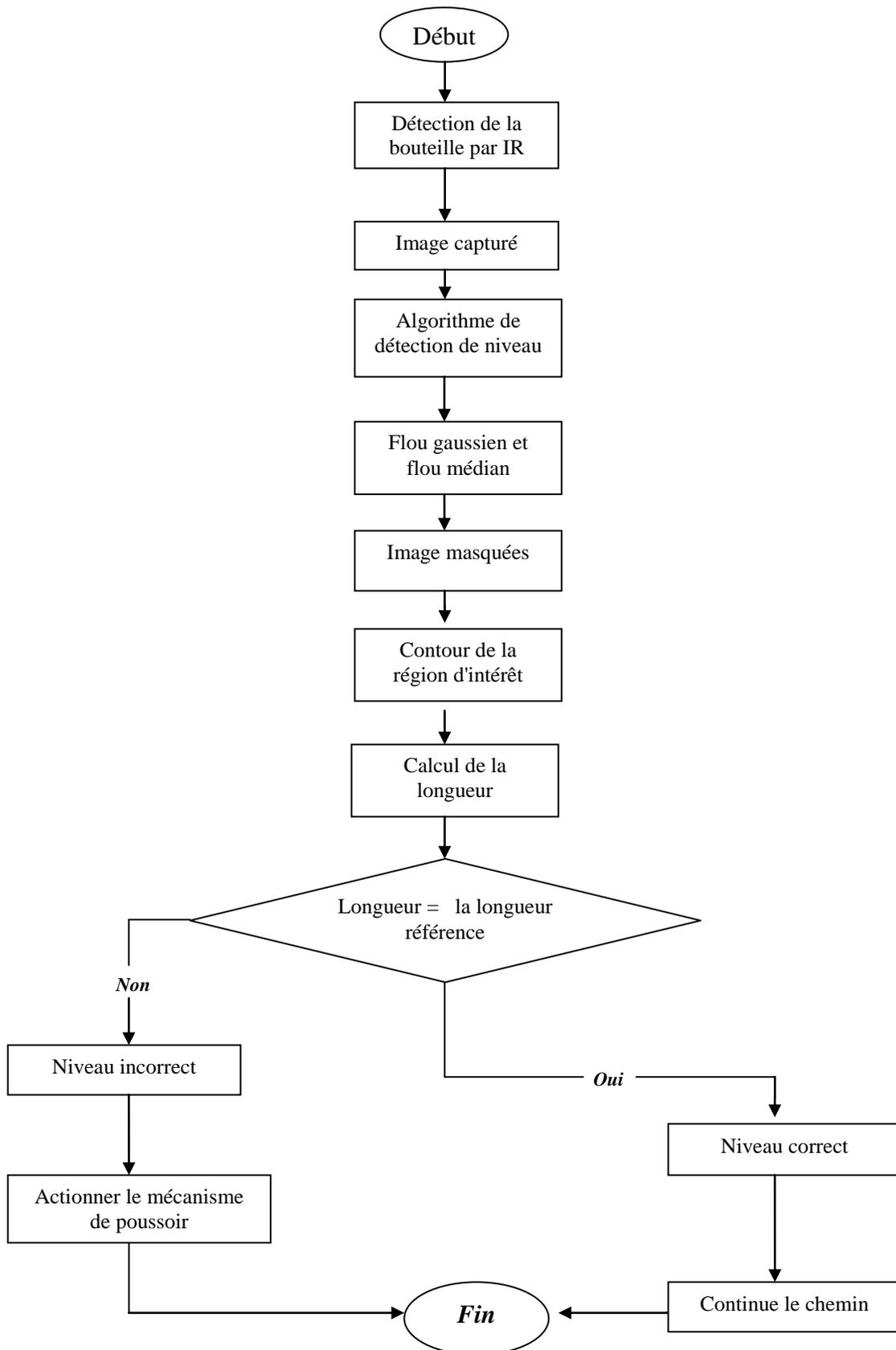


Figure IV.6: Algorithme de détection de niveau.

IV.3. Evaluation des algorithmes

Dans cette section, nous illustrons quelques résultats expérimentaux réalisés dans des conditions d'éclairage et un fond blanc pour réduire les reflets et les bruits indésirables. L'algorithme proposé a été testé en utilisant deux liquides couleurs : orange et violet, avec différents scénarios de la bouteille :

- ✓ Niveau admissible (avec bouchon détecté).
- ✓ Aucun bouchon détecté.
- ✓ Niveau inadmissible (avec bouchon détecté).
- ✓ Niveau inadmissible - Aucun bouchon détecté.

Les résultats des bouteilles avec boisson orange et violet sont présentés dans les tableaux IV.1 et IV.2 respectivement.

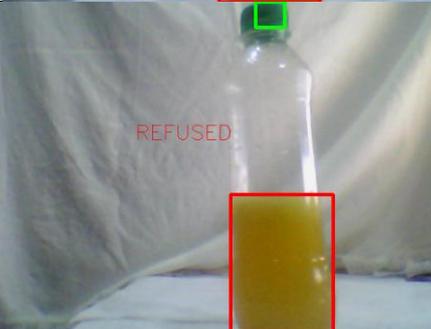
Capture d'image	Le cas
	<p>Niveau admissible (avec bouchon détecté)</p>
	<p>Aucun bouchon détecté</p>
	<p>Niveau inadmissible (avec bouchon détecté) Bouteille insuffisamment remplie</p>
	<p>Niveau inadmissible (avec bouchon détecté) Bouteille insuffisamment remplie</p>
	<p>Niveau inadmissible – Aucun bouchon détecté</p>

Tableau IV. 1: Résultats du test du flacon avec du liquide orange.

Capture d'image	Le cas
	Niveau inadmissible (avec bouchon détecté)
	Aucun bouchon détecté
	Niveau inadmissible (avec bouchon détecté) Bouteille insuffisamment remplie
	Niveau inadmissible – Aucun bouchon détecté

Tableau IV. 2: Résultats du test du flacon avec du liquide violet.

Comme on peut le constater clairement à partir des résultats pratiques illustrés dans les tableaux ci-dessus, le système a donné des résultats très précis et a pu détecter tous les cas possibles. L'algorithme était insensible aux effets de flou causés par le mouvement. Cependant, pour une meilleure qualité des images capturées, certaines techniques d'amélioration sont nécessaires pour réduire le flou et le bruit.

IV.3.1. L'inspection de bouchon

L'inspection des bouchons de bouteilles est notre objectif majeur. Lors des tests unitaires, nous avons défini les exigences qui ont été utilisées pour tester correctement le bouchon de la bouteille, comme mentionné précédemment, car nous travaillons sur un convoyeur en mouvement continu et il s'arrête pour dégager la bouteille inadmissible avec le poussoir.

il a été testé correctement que le convoyeur se déplaçait à la vitesse requise qui a été atteinte en fournissant 12V alimentation en courant continu et également que la caméra était correctement synchronisée avec le capteur infrarouge de façon que à chaque fois que la bouteille passe elle sera détectée par un capteur IR, donc l'image en temps réel est capturée immédiatement et envoyée à l'algorithme pour un traitement ultérieur. Il convient de noter ici que l'arrière-plan de l'image est toujours présent. Ainsi, lorsque les conditions préalables sont remplies après les tests unitaires, nous avons commencé à tester les fonctionnalités de l'algorithme. On a deux bouteilles de différentes couleurs de liquide, avec un bouchon de même couleur et les mêmes caractéristiques (forme et zone). De plus, comme le bouchon est la partie la plus haute de la bouteille, pour obtenir uniquement la détection de la fonction de bouchon, nous avons mis en évidence la zone du bouchon uniquement pour l'inspection du bouchon. Pour cela, nous avons dû choisir uniquement la portion de bouchon.

Après avoir effectué plusieurs tests pour détecter la meilleure zone d'image possible, nous avons arrivé à choisir les dimensions de la zone [0 :60,200 :500]

Nous avons un bouchon de couleur verte. Pour obtenir la détection de la couleur verte, nous avons utilisé la transformation RVB vers HSV. HSV nous a aidés à obtenir un espace colorimétrique ciblé à partir de plusieurs espaces colorimétriques. La sélection de la valeur HSV de la bouteille était un test le plus important car nous devons définir une valeur différente une par une et tester à chaque fois en définissant une valeur à la fois. HSV a trois parties à savoir H pour Hue, S pour Saturation et V pour la valeur qui est l'intensité.

Comme nous l'avons mentionné, la couleur du bouchon est verte. Donc, pour obtenir la couleur requise, nous avons dû vérifier plusieurs fois en changeant les valeurs HSV une par une. Un petit changement dans la gamme de couleurs a rendu le résultat erroné, par conséquent plusieurs essais ont été faits pour sélectionner les valeurs requises. La gamme de

couleur verte est obtenue, nous pouvons choisir en fonction de la teinte verte requise. Le problème était de tester la saturation.

Comme nous pouvons voir les résultats de la détection de la couleur verte. En raison d'une saturation incorrecte de la couleur verte, nous avons obtenu une nuance noire dans notre bouchon au lieu de la couleur verte. Après plusieurs tests utilisant une plage de saturation différente, nous avons obtenu notre espace colorimétrique souhaité de couleur verte dans une plage de saturation.

Après avoir trouvé et testé l'espace colorimétrique correct, nous avons décidé d'appliquer des opérations morphologiques telles que l'érosion et la dilatation pour éliminer le bruit. Nous avons simplement utilisé les fonctions Erode et Dilatation pour vérifier les effets avec l'itération : une pour l'érosion et deux pour la dilatation. L'itération est essentiellement l'intensité de la suppression du bruit. Après avoir sélectionné avec succès les valeurs de ces opérations morphologiques, nous avons procédé à un test final du bouchon de la bouteille qui nous a permis de déterminer si le bouchon était correct ou non. Pour cela, nous avons utilisé le contour et le contournement avec une fonction simple approximative en chaîne. Il s'agit essentiellement de dessiner le contour sur les limites du bouchon qui a été trouvé par la gamme HSV. Ici, nous avons rencontré de nombreux problèmes avec la zone des contours en raison des différents effets sur l'image, la zone changeant chaque fois que nous testons la bouteille. En changeant légèrement la saturation ou l'intensité du HSV, notre zone de contour, qui n'est en fait pas un pixel de l'image, a changé et a faussé notre résultat. L'échantillon de résultat est joint ci-dessous pour la gamme de couleurs défectueuses.

Comme nous pouvons le voir, en raison d'une mauvaise plage de saturation du HSV, nous avons obtenu une limite de contour appropriée et la surface du compteur est inférieure à la surface échantillonnée de la bonne bouteille. Nous avons utilisé une gamme de bouchons de bouteilles modèles corrects qui va de 1500 à 2000.

IV.3.2. L'inspection de niveau

Notre deuxième objectif majeur est l'inspection de niveau du liquide dans la bouteille. Deux algorithmes différents ont été testés pour l'inspection de niveau. Le premier algorithme traite de la transformation de la ligne de Hough. Cet algorithme a été utilisé pour détecter la limite circulaire liquide sur la bouteille. Le seuil de la fonction astucieuse a été fixé à des plages spécifiques. Mais les résultats n'étaient pas précis lorsque l'image en temps réel était

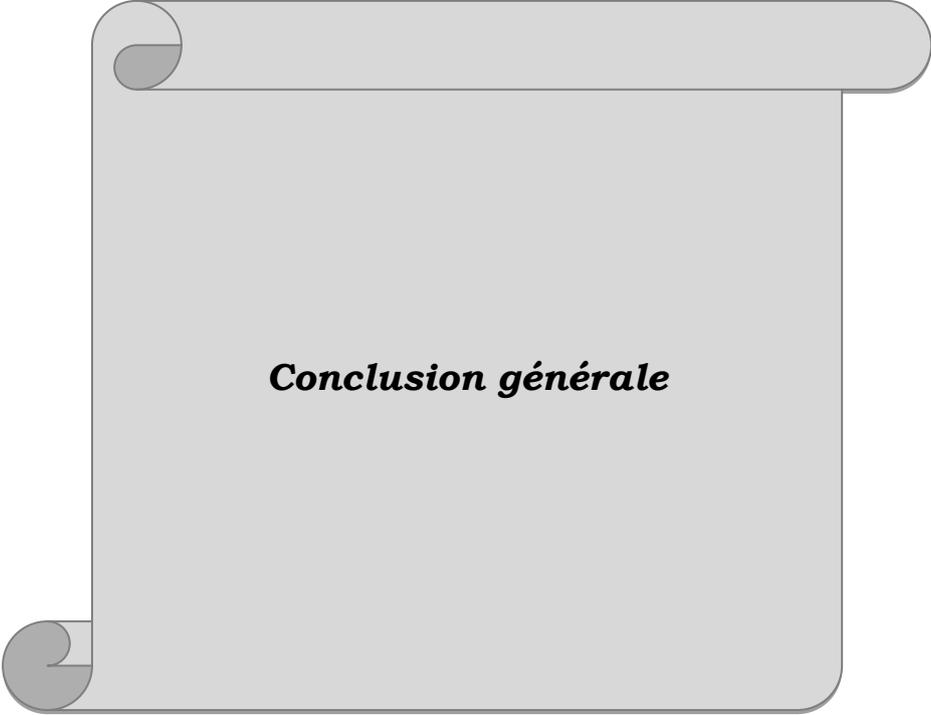
traitée car plusieurs lignes apparaissaient sur la bouteille en raison du bruit que le seuil varie pour chaque test.

Le deuxième algorithme utilisait la segmentation HSV telle qu'utilisée dans tous les algorithmes. Comme mentionné ci-dessus, toutes les bouteilles utilisées ont une couleur de liquide spécifique. Donc, une valeur pour la teinte, et une valeur de la saturation a été définie pour détecter le liquide dans la bouteille.

Après avoir trouvé et testé le bon espace colorimétrique, nous avons appliqué des opérations morphologiques telles que l'érosion et la dilatation pour éliminer le bruit. Après une sélection réussie des valeurs de ces opérations morphologiques, nous sommes allés au test final du niveau de la bouteille qui devait identifier si le niveau est correct ou non. Pour cela, nous avons utilisé le contour et le contour avec une fonction simple d'approximation de chaîne. Il dessine essentiellement le contour sur les limites du liquide qui a été trouvé par la gamme HSV. Cette dernière a été appliquée sur la zone de dimension [80 :150,200 :500].

Conclusion

La couleur est la caractéristique clé qui caractérise les produits de l'industrie alimentaire, médicale et des boissons pour les tâches d'inspection et de tri des produits. Dans ce travail, un système de vision industrielle a été proposé pour l'inspection de bouteilles de boissons où nous détectons le niveau de remplissage des bouteilles et la présence du bouchon par traitement d'image couleur.



Conclusion générale

Conclusion générale

Avec le système d'inspection automatique précis, les industries peuvent améliorer leur rapport de rendement un plus grand nombre de produits peuvent être inspectés en moins de temps contrairement à la précision, l'efficacité et rentable. Comme la vision industrielle est utilisée dans un tel système qui fonctionne sur des données spécifiées donné comme une donnée prédéfinie. Ainsi, la précision peut être améliorée. De plus, ces industries peuvent utiliser un tel système pour une durée maximale car les humains ne peuvent fonctionner que dans une certaine mesure.

Dans ce mémoire, nous avons présenté et réalisé un système d'inspection des bouteilles boissons basée sur la carte Raspberry. Ce système a la capacité de vérifier la qualité des bouteilles sous différents aspects et de détecter les défauts pour le contrôle de la qualité en utilisant différentes techniques de traitement d'image. Il peut détecter les bouteilles inadmissibles par rapport à deux paramètres qui sont : bouchon de la bouteille c'est-à-dire vérifié s'il est bien présent ou non, et le niveau de liquide présent dans la bouteille est admissible ou non. Si l'un des paramètres suivants (bouchon et niveau) n'est pas valide, la bouteille est considérée comme défectueuse et sera rejeter du convoyeur par un mécanisme poussoir.

A la sortie du convoyeur, les produits sont triés par rapport à leur couleur c'est-à-dire les bouteilles de raisin dans un carton et les bouteilles d'orange dans un autre carton.

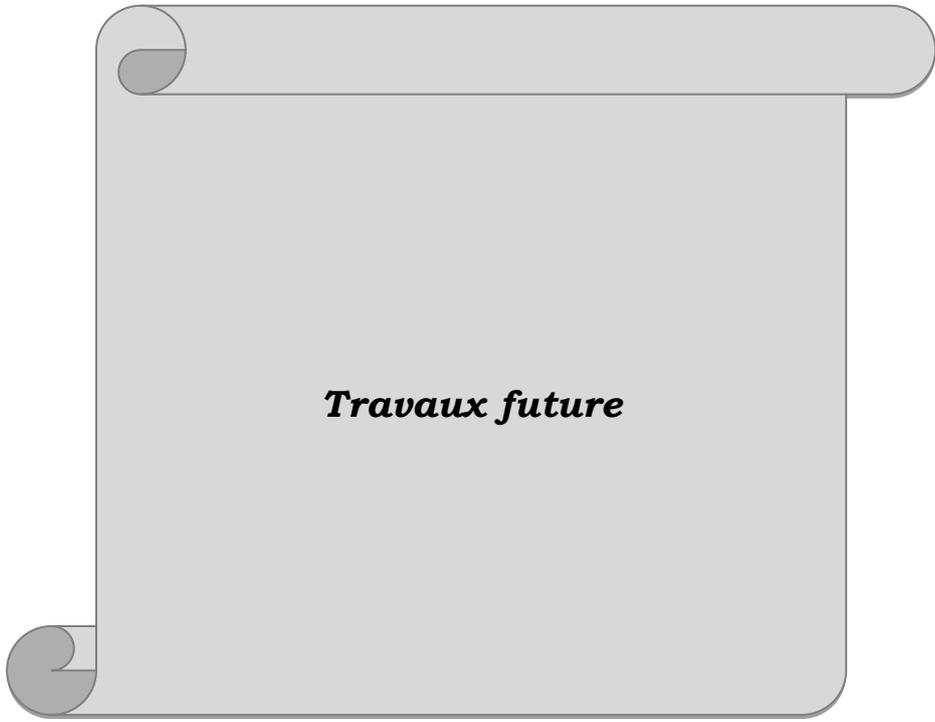
La bouteille placée sur le convoyeur. La caméra et le capteur infrarouge sont synchronisés de telle manière que lorsque la bouteille arrive devant le capteur infrarouge, il détecte la bouteille et la caméra capture l'image. L'image est ensuite envoyée au Raspberry pi où le traitement d'image est effectué.

Le système d'inspection présenté dans ce travail présente une solution industrielle automatisée rapide, efficace et peu coûteuse. Il peut être utilisé dans le système d'inspection dans la production industrielle pour améliorer les normes de qualité et pour surmonter la méthode manuelle d'inspection qui, non seulement améliore la précision, mais permet également de réduire considérablement le temps d'inspection des produits. De plus, ce système peut être étendu à plus de travaux d'inspection comme lorsque nous travaillons sur une tâche majeure.

Il peut être utilisé dans différentes industries comme dans les industries textiles pour inspecter les textures sur leur des produits. De plus, dans notre domaine, il peut être utilisé pour vérifier si les bouteilles se sont fissurées.

Nous avons mis en œuvre avec succès les deux objectifs principaux qui étaient, l'inspection de bouchon et l'inspection du niveau de jus, et d'autre objectif aussi on site :

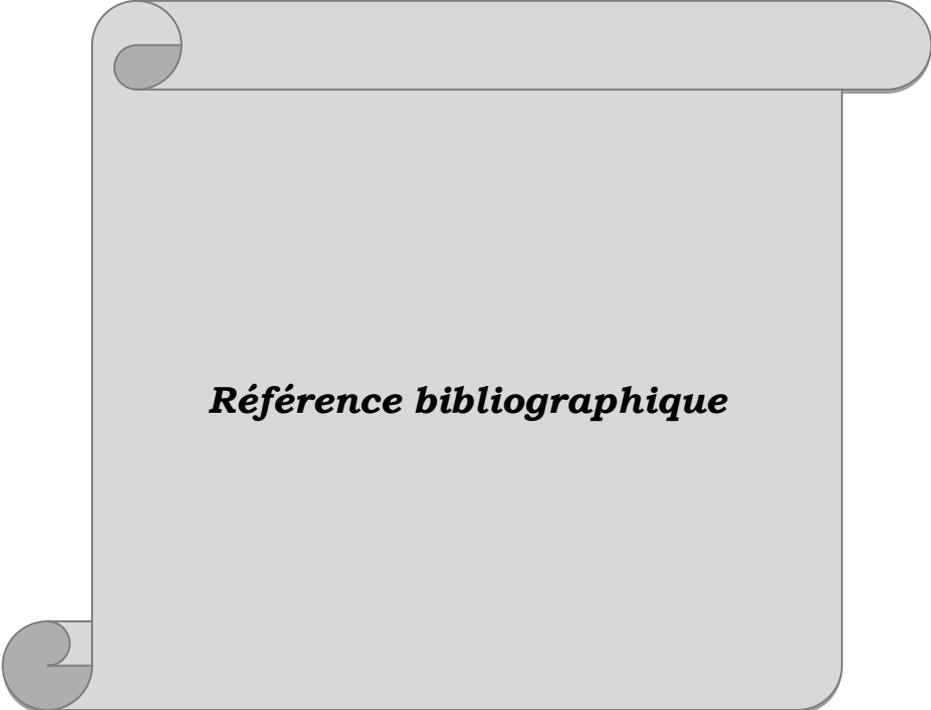
- ✓ Implémentation du matérielle du convoyeur.
- ✓ Implémentation du circuit de commande du moteur et des capteurs IR dans le circuit.
- ✓ Détecter les bouteilles sur un convoyeur à l'aide d'un capteur infrarouge.
- ✓ Contrôle du convoyeur à l'aide d'un microcontrôleur.
- ✓ Capturez des images de bouteilles en temps réel à l'aide d'une webcam pendant que le convoyeur continue de se déplacer.
- ✓ Correspondance de modèle de capture en temps réel d'une image avec des données prédéfinies.
- ✓ Détecter si les bouchons sont desserrés ou manquants ou non en faisant correspondre le modèle.
- ✓ Détecter le niveau de liquide à l'intérieur de la bouteille par filtrage d'image.



Travaux future

Travaux future

Ce travail peut également être amélioré en travaillant en profondeur sur les paramètres tels que vérifier si l'étiquette est présente, déformée, de mauvaise qualité, ou les données et les informations sont correctes ou non, etc. et de même en cas de contrôle de niveau ; si le liquide rempli est de meilleure qualité ou non, etc. Ceci conduit au concept de technologie de vision stéréo dans laquelle deux ou plus de deux caméras sont utilisées, et cela augmente l'efficacité et améliore le contrôle de qualité. Ce travail peut être élargi à un système générique pour des boissons et des produits de différentes tailles, ce qui signifie qu'il n'est pas seulement lié à une taille ou à une forme spécifique de l'objet détecté. Ce système peut également être mis en œuvre dans un environnement où des mains robotisées peuvent être utilisées à la place du poussoir pour prélever et placer les boissons sur les lignes de convoyeur et pour retirer les bouteilles de boissons défectueuses du convoyeur. Cela nécessite une efficacité et un niveau de précision plus élevés. Le codage effectué dans le présent travail est si flexible qu'il ne nécessite pas autant de changements ou de variations pour mettre en œuvre les scénarios discutés ci-dessus et peut également être utilisé pour différentes boissons autres que les bouteilles. C'est un projet industriel et peut également être utilisé dans les industries alimentaires, les industries laitières, les industries médicales, etc.



Référence bibliographique

Référence bibliographique

- [01] B. Pokharel, "MACHINE VISION AND OBJECT SORTING PLC Communication with LabVIEW..using..OPC",.Theseus.fi,2013..Available:https://www.theseus.fi/bitstream/handle/10024/58533/Pokharel_Bikarna.pdf?sequence=1
- [02] *Document Organisation de la prévention. Les techniques d'indentification des risques l'inspection*
- [03] *La jaune rouge : l'industrie du future et numérique*
- [04] *CYBERSÉCURITÉ DES SYSTÈMES INDUSTRIELS Iste Systèmes 24 Janvier 2019 Sciences & Techniques*
- [05] *Les systèmes industriels HB pour Melec V2.0 Septembre 2020*
- [06] www.logicscontorls.it
- [07] J. Sipponen, "EMPLOYING MACHINE VISION TO A MANUFACTURING LINE", Theseus.fi.,2010..Available:https://www.theseus.fi/bitstream/handle/10024/15849/Sipponen_Joni.pdf?sequence=1
- [08] National Instruments 2013. *Connect LabVIEW to any industrial Network and PLC.* Available: <http://www.ni.com/white-paper/5407/en>
- [09] Ms.Shubhada.K.Nagrle & Mr.S.T.Bagde "Developing Automated Inspection System with the Help of Image Processing" Available:.<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.670.6883&rep=rep1&type=pdf>.
- [10] Rubén Posada-Gómez, Oscar Osvaldo Sandoval-González, Albino Martínez Sibaja, Otniel Portillo- Rodríguez and Giner Alor-Hernández (2011). *Digital Image Processing Using LabView, Practical Applications and Solutions Using LabVIEW™ Software, Dr. Silviu Folea (Ed.).*
- [11] P. Kumar and D. Ramakrishna, "AUTOMATED BOTTLE CAP INSPECTION USING MACHINE.VISION.SYSTEM",.Pdfs.semanticscholar.org.,2018.Available: <https://pdfs.semanticscholar.org/a8a0/c11303ff5fe5d415a58b3ddeece3e1c877ae.pdf>.

- [12] *Bolhouse V, "Fundamentals of Machine vision", version 3, May 2009 PDF.*
- [13] *Elias Malamas, Euripides Petrakis, Michalis Zervakis, "A Survey on Industrial Vision Systems, Applications and Tools"PDF, HIPER (BE97-5084) under program Bright Euram of the European Union (EU). 2009.*
- [14] *V. Shah and D. Bhatt, "Image Processing Based Bottle Filling and Label Checking Using Embedded System", Ijircce.com, 2018. [Online]. Available: http://www.ijircce.com/upload/2016/may/239_Image.pdf. [Accessed: 26- Oct- 2018].*
- [15] *Radu Horaud et Oulivier Monga , vision par ordinateur 1993(first edition) [ISI 98] ISIMA 1988,1999. Introduction à C++ builder PDF*
- [16] *Wikipedia.org Vision pas ordinateur*
- [17] *E.R. Davies. Images and imaging operations. In E.R. Davies, editor, Computer and Machine Vision (Fourth Edition) PDF.*
- [18] *Gray Bradsky and Adrian Kaehler. Learning OpenCV. O'Reilly,2008.PDF*
- [19] *Wikipedia. Méthode Otsu*
- [20] *Wikipedia. Morphologie.*
- [21] *Image Processing and Acquisition Using Python by Ravishankar Chityala and Sridevi Pudipeddi PDF.*
- [22] *OpenCV-Python Tutorials Documentation Release 1 Alexander Mordvintsev & Abid K*
- [23] *Digital Image Processing (3rd Edition) by Rafael C. Gonzalez -Richard E. Woods.PDF*
- [24] *Julio César Hernandez, José Mar'a Sierra, and André Seznec. The sac test: A new randomness test, with some applications to prng analysis. In Proceedings of the International Conference Computational Science and Its Applications - ICCSA 2004, LNCS vol. 3043 PDF.*
- [25] *Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing.PDF*
- [26] *Site officiel de Raspberry Pi*

- [27] *Site de officiel de Raspberry Pi France - Windows 10 IoT Core* Consulté
- [28] *Site de Raspberry Pi France - Comment installer Android pi*
- [29] *<http://wotho.ethz.ch/mvt4apl-2.00/>*
- [30] *U. Team, "What is Ubuntu MATE?", Ubuntu MATE, 2019. <https://ubuntu-mate.org/what-is-ubuntu-mate/>.*
- [31] *Wikipedia. PuTTY*
- [32] *Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF)*
- [33] *Python Software Foundation. Retrieved 24 April 2012., second section "Fans of Python use the phrase "batteries included" to describe the standard library, which covers everything from asynchronous processing to zip files."*
- [34] *Pulli, Kari; Baksheev, Anatoly; Korniyakov, Kirill; Eruhimov, Victor (1 April 2012). "Realtime Computer Vision with OpenCV". PDF*
- [35] *Intel acquires Itseez: <https://opencv.org/intel-acquires-itseez.html>*
- [36] *"CUDA". opencv.org.*

