

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE SAAD DAHLEB BLIDA 1



FACULTE DE TECHNOLOGIE
DEPARTEMENT ELECTRONIQUE
SPECIALITE ELECTRONIQUE DES SYETEMES EMBARQUES

THEME

*Tri automatique d'objets en temps réel
avec la Raspberry Pi*

Présenté par : BOUMEDIENE WALID

BOUZOUIDJA MAHFOUD

Encadré par : Pr. NAMANE ABDERRAHMANE

Dr. FAS MOHAMED LAMINE

Devant le jury composé de : *M^{me}* BOUGHERIRA NADIA

Mr. KABIR YACINE

Soutenu le : 14 / 07 / 2021

Promotion 2020 - 2021

Remerciements

C'est avec grand plaisir qu'on réserve cette page, en signe de gratitude et de reconnaissance à tous ceux qui nous ont aidé à la réalisation de ce travail.

Nous remercions tout d'abord Mr. NAMANE ABDERRAHMANE et Mr. FAS MOHAMED LAMINE pour leurs patiences, leurs conseils judicieux et pour nous avoir suivi et orienté et toujours poussé à viser plus loin.

Nous remercions également, notre chef de spécialité M^{me} NACEUR DJAMILA, nos professeures M^{me} BOUGHERIRA HAMIDA et BOUGHERIRA NADIA et Mr. KABIR YACINE pour son encouragement et ses efforts afin de mener à bien notre formation de master en Électronique des Systèmes Embarqués.

Nous remercions, enfin, toutes personnes ayant contribué de près ou de loin à la réalisation de ce travail.

Table des matières

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

Chapitre I : Généralité sur le matériel

I.1. Introduction :	3
I.2. Généralités sur les convoyeurs :	3
I.2.1. Définition d'un convoyeur :	3
I.2.2. Composants et leur dénomination :	4
I.2.3. Courroie transporteuse :	4
I.2.4. Tambour moteur :	4
I.2.5. Tambour de renvoi :	4
I.2.6. Tambours de contrainte :	5
I.2.7. Rouleaux ou tambours :	5
I.2.8. Stations porteuses en auge en stations-supports inférieures :	5
I.2.9. Trémie :	5
I.2.10. Capots pour convoyeurs :	5
I.2.11. Moto-réducteur :	5
I.3. Vision industrielle :	6
I.3.1. Composants d'une caméra :	6
I.3.1.1. Objectif :	6
I.3.1.2. Distance focale	6
I.3.1.3. Champ d'un objectif :	9
I.3.1.4. Ouverture relative :	10
I.3.1.5. Profondeur de champ et mise au point :	11
I.3.1.6. Obturateur et le temps d'exposition :	13
I.3.1.7. Capteur :	14
a. CCD :	15
b. CMOS :	16
I.3.2. Principe de fonctionnement d'une appareil photo :	17
I.3.3. L'analogie entre l'appareil photo et l'œil :	18
I.3.4. Camera utiliser : pi camera v1.3 :	19
✓Caractéristiques de la Picamera :	19

I.4. Raspberry Pi :	20
I.4.1. Définition :	20
I.4.2. Caractéristiques :	21
I.5. Servomoteur :	23
I.5.1. Principe général :	23
I.5.2. Fonctionnement :	24
I.6. Moteur driver :	26
I.6.1. Description :	26
I.6.2. Spécifications :	27
I.7. Capteur infrarouge :	28
I.7.1. Définition :	28
I.7.2. Principes de la détection infrarouge :	29
I.7.3. Spécifications de capteur utilisé :	30
I.8. Conclusion	31

Chapitre II: Les techniques de traitement d'image

II.1. Introduction :	32
II.2. Notions d'une image numérique :	33
II.2.1. Pixels :	33
II.2.2. Espaces de couleur et canaux :	34
II.2.2.1. Niveaux de gris :	35
II.2.2.2. Espace RVB :	36
II.2.2.3. Espace TSL :	37
II.2.3. Notion d'histogramme :	37
II.2.4. Profondeur d'image :	39
II.2.5. Formation d'une image à partir de canaux :	39
II.2.6. Le système de coordonnées d'image :	40
II.3. Compréhension des filtres d'images :	41
II.3.1. Flou gaussien :	41
II.3.1.1. Description :	41
II. 3.1.2. Formulation :	42
II.3.2. Seuillage des images :	44
II.3.2.1. Description :	44
II.3.2.2. Fonctionnement :	45

II.3.2.3. Formulation :	45
II.3.2.4. Directives d'utilisation :	45
II.3.4. Méthode de seuillage d'Otsu :	49
II.3.5. Représentation polygonale :	51
II.3.5.1. Description :	51
II.3.5.2. Mesures d'erreurs :	52
II.4. Conclusion :	53

Chapitre III: Partie pratique et implémentation du système

III.1. Introduction :	54
III.2. Environnement de travail :	54
III.2.1 Hardware:	54
III.2.2. Software:	54
III.3. Bibliothèques de python utilisé :	55
III.3.1. OpenCV :	55
III.3.2. NumPy :	55
III.3.3. Adafruit	56
III.3.4. Time	56
III.3.5. Future :	56
III.3.6. Picamera :	56
III.4. Principe du système développé :	57
III.4.1. Acquisition d'image :	58
III.4.2. Unité de traitement d'image :	62
III.4.3. Segmentation et identification :	68
III.4.3.1. Prétraitement :	68
III.4.3.2. Segmentation :	69
III.4.3.3 Extraction des contours :	71
III.4.3.4 Identification de classe :	72
III.4.4. Séparation des classes :	73
III.5. Résultats des tests :	78
III.6. Problèmes rencontrés :	83
Conclusion générale	84
Annexe	85
Références	86

Liste des Figures

Chapitre I : Généralité sur le matériel

Figure I.1: Principaux éléments d'un convoyeur à courroie.	4
Figure I.2: Schéma du système optique.	7
Figure I.3: Théorème de Thales.	7
Figure I.4: Application de Thales sur un système optique.	8
Figure I.5: Apparence de l'image sur le capteur	8
Figure I.6: Exemple des différentes distances focale.	9
Figure I.7: Types d'ouverture des objectifs.	10
Figure I.8: Représentation de l'ouverture dans un système optique.	10
Figure I.9: Impact d'ouverture sur la qualité de lumière passé.	10
Figure I.10: Impact de taille de capteur sur la profondeur de champ.	11
Figure I.11: Schéma de mise au point d'un objet et son profondeur de champ.	12
Figure I.12: Pose longue (1/50eme s) image flou, notion de mouvement.	13
Figure I.13: Pose courte (1/1600eme s) image nette, temps figé.	13
Figure I.14: Types des capteurs.	14
Figure I.15: Jonctions d'un CCD.	15
Figure I.16: Camera à capteur CCD.	15
Figure I.17: Caméra à capteur CMOS.	16
Figure I.18: Principe du fonctionnement d'une caméra.	17
Figure I.19: Comparaison entre la caméra et l'œil humaine.	18
Figure I.20: Illustration de caméra utilisé.	19
Figure I.21: Illustration de Raspberry Pi 4.	20
Figure I.22: Configuration des pins de Raspberry Pi.	22
Figure I.23: Signal PWM.	24
Figure I.24: Dimension de moteur utilisé MG996R.	25
Figure I.25: Configuration des câbles.	25
Figure I.26: Illustration de module PCA9685 utilisé.	26
Figure I.27: Schéma fonctionnel de PCA9685.	27
Figure I.28: Principe de capteur infrarouge.	28
Figure I.29: Spectre des rayonnements électromagnétiques.	29
Figure I.30: Configuration des câbles du capteur utilisé.	30

Chapitre II: Les techniques de traitement d'image

Figure II.1: Stockage d'une image sur un ordinateur.	32
Figure II.2: Cette image contient 1000 pixels de largeur et 750 pixels de hauteur, pour un total de $1000 \times 750 = 750000$ pixels totaux.	33
Figure II.3: Espaces de couleurs.	34

Figure II.4: Exemple d'image en niveaux de gris.	35
Figure II.5: Gradient d'image montrant les valeurs de pixels allant du noir (0) au blanc (255).....	35
Figure II.6: Représentation de l'espace RGB.....	36
Figure II.7: Représentation de l'espace HSV.....	37
Figure II.8: Valeur de niveau de gris / nombre de pixels.....	38
Figure II.9: Formation d'une image a partir de 3 canaux.....	39
Figure II.10: Système de coordonnées d'image.....	40
Figure II.11: Effet de flou gaussien.....	42
Figure II.12: Représentation de fonction gaussienne.....	43
Figure II.13: Réduction de bruit avec le flou gaussien.....	43
Figure II.14: Effet de seuillage au l'histogramme de l'image.....	44
Figure II.15: Quelques histogrammes typiques ainsi que des choix appropriés de seuil. ..	46
Figure II.16: Histogramme d'image à niveau de gris.....	47
Figure II.17: Image binaire.....	47
Figure II.18: Histogramme d'image à niveau de gris.....	48
Figure II.19: Résultat de binarisation.....	48
Figure II.20: Forme approximée par 5 segments approximée de droites.....	51
Figure II.21: Cercle, carré et triangle approximés par 13,4 et 3 droites respectivement...	51
Figure II.22: Distance d_k du point p_k au segment de droite (p_i, p_j)	52

Chapitre III: Partie pratique et implémentation du système

Figure III.1: Organigramme de système proposé.....	57
Figure III.2: Schéma de block d'acquisition.....	58
Figure III.3: Placement de caméra dans le système.....	58
Figure III.4: Sortie de caméra en temps réel.....	58
Figure III.5: Commande de test sur le terminal de linux.....	59
Figure III.6: Image de sortie.....	59
Figure III.7: Image recardé.....	60
Figure III.8: Dimensions d'une image numérique.....	61
Figure III.9: Schéma bloc du traitement d'image.....	61
Figure III.10: Configuration du GPU dans les paramètres.....	62
Figure III.11: Activation des interfaces nécessaires dans les paramètres.....	63
Figure III.12: Diagramme d'architecture Raspberry.....	63
Figure III.13: Configuration d'une session SSH dans Putty.....	65
Figure III.14: Saisir de mot de passe Raspberry.....	66
Figure III.15: Création d'une session VNC.....	66
Figure III.16: Saisir d'adresse IP / mot de passe Raspberry dans VNC viewer.....	67
Figure III.17: Schéma bloc d'accès au bureau de Raspberry.....	67
Figure III.18: Schéma bloc des étapes du traitement d'image.....	68
Figure III.19: Schéma bloc des étapes de segmentation.....	69
Figure III.20: Histogramme de l'image originale.....	69
Figure III.21: Histogramme d'image au niveau de gris.....	70
Figure III.22: Histogramme de l'image après l'étape de binarisation.....	70

Figure III.23: Exemple d'un changement brusque d'intensité des pixels.....	71
Figure III.24: Les contours d'objet.....	71
Figure III.25: Identification de la première classe Triangle.....	72
Figure III.26: Identification de la deuxième classe Rectangle.....	73
Figure III.27: Schéma bloc des étapes de l'objet.....	73
Figure III.28: Capteurs infrarouges utilisés.....	74
Figure III.29: Capteurs de présence infrarouge et le servomoteur avec la barrière pour la classe triangle.....	75
Figure III.30: Capteurs de présence infrarouge et le servomoteur avec la barrière pour la classe rectangle.....	75
Figure III.31: Différentes étapes de fonctionnement de la barrière pour la classe triangle.....	76
Figure III.32: les différentes étapes de fonctionnement de la barrière pour la classe rectangle.....	76
Figure III.33: Capture d'écran sur la conception de SolidWorks.....	77
Figure III.34: Précision pour les différentes vitesses du convoyeur.....	79
Figure III.35: Taux d'erreur pour différentes vitesses du convoyeur.....	80

Liste des tableaux

Chapitre I : Généralité sur le matériel

Tableau II. 1: matrice 5× 5.	38
Tableau II. 2: Valeur de niveau de gris / nombre de pixels..	38

Chapitre III: Partie pratique et implémentation du système

Tableau III.1 : Résultats des tests avec la vitesse 7.16cm/s.....	78
Tableau III.2 : Résultats des tests avec la vitesse 8cm/s.....	78
Tableau III.3 : Résultats des tests avec la vitesse 10.25cm/s.....	79

Listes des acronymes et abréviations :

AIA	Automated Imaging Association
CCD	Charged Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CSI	Camera Serial Interface
DMA	Direct Memory Access
FPS	Frame per Second
GPU	Graphics Processing Unit
HDMI	High-Definition Multimedia Interface
HSV	Hue Saturation Value
I2C	Inter-Integrated Circuit
IOT	Internet Of Things
LAN	Local Area Network
MMAL	Multi-Media Abstraction Layer
MPP	Minimum Perimeter Polygon
NO	Normalement Ouvert
PWM	Pulse Width Modulation
RAM	Random Access Memory
RGB	Red Green Blue
RTOS	Real Time Operating System
SCL	Serial Clock Line
SD	Secure Digital
SDA	Serial Data Line
SOC	System On Chip
SSH	Secure SHell
USB	Universal Serial Bus
VNC	Virtual Network Computing

Résumé :

L'objectif de ce projet est de concevoir et réaliser un système de tri automatique d'objets placés sur un convoyeur, le système proposé utilise une chaîne de vision artificielle et un convoyeur, la vision artificielle est réalisée autour d'une carte Raspberry Pi muni d'une caméra. Cet ensemble matériel et un jeu de capteurs contrôlent le tri d'objets moyennant un jeu de servomoteur. Le système donne des résultats satisfaisants et positifs.

Mots clés :

Tri automatique, détection d'objets, chaîne de vision artificielle, Raspberry Pi, servomoteurs, capteur, convoyage industriel.

Abstract:

The objective of this project is to design and realize an automatic object sorting system placed on a conveyor, the proposed system uses an artificial vision chain and a conveyor, the artificial vision is realized around a Raspberry Pi board equipped with a camera, this hardware assembly and a set of sensors control the sorting of objects by means of a servomotor set. The system produces satisfactory and positive results.

Key Words:

Automatic sorting, objects, artificial vision chain, Raspberry Pi, servos, sensors, industrial conveying.

ملخص :

الهدف من هذا المشروع هو تصميم وتنفيذ نظام تلقائي لفرز الأجسام يوضع على جهاز نقل، ويستخدم النظام المقترح سلسلة رؤية اصطناعية ووسيلة نقل، يتم تحقيق الرؤية الاصطناعية عن طريق لوحة راسبري باي مزودة بكاميرا، وهذه المجموعة من الأجهزة ومجموعة من أجهزة الاستشعار تتحكم في فرز الأشياء بواسطة مجموعة من أجهزة السرفموتور. النظام ينتج نتائج مرضية وإيجابية.

الكلمات الرئيسية:

فرز تلقائي ، أشياء ، سلسلة رؤية اصطناعية ، راسبري بي ، سيرفوس ، أجهزة استشعار ، نقل صناعي.

Introduction

générale

La vision industrielle, appelée Machine Vision, est une discipline appliquée de la vision artificielle qui emploie le traitement numérique d'images pour automatiser des tâches. Elle utilise des moyens électroniques et informatiques pour doter des machines et robots de capacités sensorielles analogues à la vision humaine. Autrement dit, elle permet d'utiliser des caméras comme capteurs pour détecter des défauts sur des objets manufacturés, inspecter des pièces en cours de fabrication, les compter, les trier, les classer, les mesurer, ... à partir de leur apparence visuelle.

Par exemple, des systèmes de vision sont intégrés sur des robots de soudure pour vérifier la continuité des cordons, sur des machines à coudre le cuir pour suivre les contours des pièces à assembler, sur des presses de moulage de CD-Rom pour vérifier l'absence de griffes, à la sortie d'imprimantes à gravure laser pour vérifier la lisibilité des numéros de série sur des composants électroniques, ... La gamme d'applications possibles est très étendue [1].

Il faut néanmoins se garder d'imaginer une analogie trop profonde entre les performances de la vision humaine, capable d'explorer la structure spatiale de son environnement et d'y reconnaître des objets complexes et variés, connus ou inconnus, en deux dimensions comme en trois, et les capacités bien plus modestes des systèmes de vision industriels: ces derniers ont essentiellement pour mission de mesurer des propriétés simples sur des objets et scènes connus à l'avance, pour en déduire la conformité par rapport à des modèles de référence. Bien que dans certains cas, des calculs élémentaires suffisent à résoudre les problèmes posés, tout l'arsenal des méthodes du traitement d'image vient bien à point pour parvenir à extraire l'information pertinente.

L'objectif de ce projet est de réaliser un système capable de faire la détection et le suivi d'objets mobiles dans un environnement dynamique, en utilisant une caméra et un convoyeur qui déplace des objets. Il a pour objectif la détection des objets mobiles et compter ces objets après un prétraitement, moyennant une programmation par Python. Ce sujet représente un défi important imposé par le moyen d'acquisition et du traitement pour faire fonctionner l'application en temps réel.

Ce mémoire est composé de trois chapitres le premier chapitre consiste de généralité et des descriptions sur le matériel utilisé dans la réalisation de ce projet, le deuxième chapitre est essentiellement consacré aux techniques de traitement d'image, afin de récupérer une image facilement exploitable et extraire les informations nécessaires pour assurer le bon fonctionnement du comptage et de la séparation des objets avec précision, le troisième chapitre présente la partie pratique et implémentation du système proposé, l'environnement de travail et les problèmes rencontrés.

Chapitre I

Généralité sur le matériel

I.1. Introduction :

Les systèmes de manutention de produits en vrac sont liés à un processus de transport continu ou discontinu. Le système discontinu a été jusqu'au début du vingtième siècle le seul moyen de transport des matériaux sur les grandes distances. Avec l'apparition des systèmes de transport continu, la manutention discontinue est entrée dans sa phase handicapé par aucun problème de chargement, d'accélération, de freinage ou de déchargement. Pourtant, ce moyen modère de transport des produits en vrac, malgré sa fiabilité et son économie a eu des difficultés à s'imposer sur le marché des transports fortement dominé, à l'époque, par le marketing des constructeurs de camions.

Le domaine de prédilection du convoyeur a bande est le transport de tous les matériaux en général notamment, les minerais et ce dans tous les secteurs industriels. Le développement technologique considérable des bandes transporteuses à armature textile et des nouveaux moyens mécanique d'entraînement, positionne très fréquemment aujourd'hui le convoyeur comme un rival direct du camion, du chemin de fer.

I.2. Généralités sur les convoyeurs :

I.2.1. Définition d'un convoyeur :

Le convoyeur est un système de manutention automatique qui permet de déplacer des produits finis ou bruts d'un poste à un autre par le mécanisme de transmission de puissance. Cette dernière est transmise d'un arbre moteur vers un ou plusieurs arbres récepteurs par l'intermédiaire de courroies ou de chaînes [2].

Le produit ou la marchandise étant placé sur une bande ou sur une tôle, se déplace d'une manière uniforme dans un circuit fermé. La vitesse de déplacement est relative à la vitesse de rotation du moteur et peut être réduite ou augmentée selon la volonté de l'opérateur en tenant compte de quelques paramètres tels que la productivité et la cadence de production.

I.2.2. Composants et leur dénomination :

Le schéma ci-dessous (*Figure I.1*) indique les principaux éléments d'un convoyeur à bande.

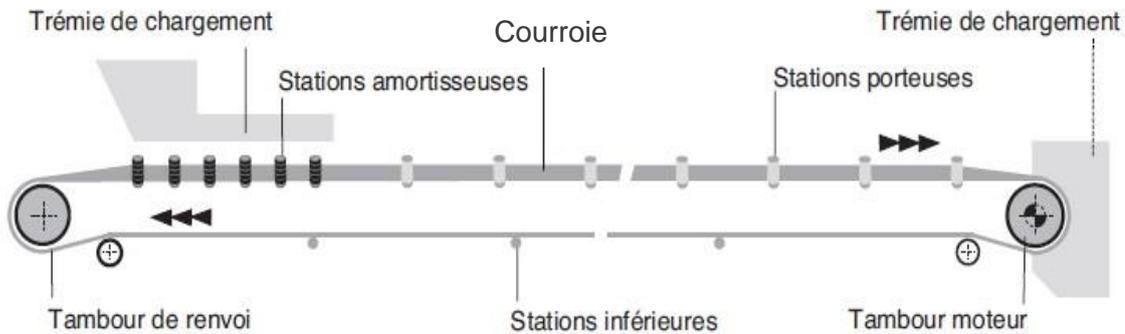


Figure I.1: Principaux éléments d'un convoyeur à courroie.

I.2.3. Courroie transporteuse :

La courroie ou la bande transporte la matière première de la queue jusqu'à la tête du convoyeur. Elle se présente sous deux formes principales, plâtre et en auge, tout bande comporte deux faces :

- La face externe, qui est en contact avec les matériaux transportés.
- La face interne, qui est en contact avec les rouleaux ou les tambours.

I.2.4. Tambour moteur :

La surface du tambour moteur peut être laissée en finition normale ou avoir un revêtement de caoutchouc dont l'épaisseur est calculée en fonction de la puissance à transmettre de manière à augmenter le coefficient de frottement et à faciliter l'évacuation de l'eau à la surface du tambour. Le diamètre du tambour est dimensionné en fonction de la catégorie et du type de courroie (bande), ainsi que des pressions calculées sur sa surface [2].

I.2.5. Tambour de renvoi :

La surface du carter n'a pas nécessairement besoin d'être munie d'un revêtement, sauf dans certains cas. Le diamètre est normalement inférieur à celui qui est prévu pour le tambour moteur.

I.2.6. Tambours de contrainte :

Ils servent à augmenter l'arc d'enroulement de la bande et, d'une manière générale, ils sont utilisés dans tous les cas où il est nécessaire de dévier la bande au niveau des dispositifs de tension à contrepoids, des appareils de déchargement mobiles, etc.

I.2.7. Rouleaux ou tambours :

Ils soutiennent la bande et tournent librement et facilement sous la charge. Ce sont les composants les plus importants du convoyeur et ils représentent une part considérable de l'investissement total.

I.2.8. Stations porteuses en auge en stations-supports inférieures :

Les rouleaux porteurs sont généralement fixés sur des pattes de fixation soudées à une traverse ou un support. L'angle d'inclinaison des rouleaux latéraux varie entre 20 ° et 45 °.

Les stations-supports inférieures peuvent comporter un seul rouleau sur toute la largeur ou bien deux rouleaux formant un « V » en inclines à 10°.

I.2.9. Trémie :

La trémie est conçue pour faciliter le chargement et le glissement du produit en absorbant les chocs de la charge et en évitant les colmatages et l'endommagement de la bande.

I.2.10. Capots pour convoyeurs :

Les capots pour convoyeurs ont une importance fondamentale lorsqu'il est nécessaire de protéger le produit transporté de l'air ambiant de d'assurer le bon fonctionnement de l'installation.

I.2.11. Moto-réducteur :

Dans cette configuration, le moteur, la boîte d'engrenages et les roulements constituent un ensemble complet, enfermé et protégé à l'intérieur d'un carter, qui entraîne directement la bande. Cette solution élimine toutes les complications liées aux transmissions extérieures, couples, etc.

I.3. Vision industrielle :

Selon l'Automated Imaging Association (AIA), la vision industrielle comprend toutes les applications industrielles et non industrielles dans lesquelles une combinaison matérielle et logicielle fournit aux systèmes des directives opérationnelles relatives à l'exécution de leurs fonctions en se basant sur l'acquisition et le traitement d'images. Bien que la vision industrielle utilise les mêmes algorithmes et approches que les applications scolaires/éducatives et publiques/militaires, les contraintes sont différentes.

Les systèmes de vision industrielle exigent une robustesse, une fiabilité et une stabilité supérieures à celles des systèmes de vision scolaires/éducatifs, et sont bien moins coûteux que ceux utilisés dans les applications publiques/militaires. Par conséquent, la vision industrielle est caractérisée par un faible coût, une précision acceptable, une grande robustesse, une grande fiabilité, et une grande stabilité mécanique et thermique.

Les systèmes de vision industrielle reposent sur des capteurs numériques incorporés dans des caméras industrielles dotées d'optiques spéciales permettant l'acquisition d'images, de façon à ce que le matériel et le logiciel puissent traiter, analyser et mesurer différentes caractéristiques d'une pièce pour la prise de décision [3].

I.3.1. Composants d'une caméra :

I.3.1.1. Objectif :

De nos jours, la plupart des caméras et tous les caméscopes sont munis d'un objectif à focale variable communément appelé zoom. Il regroupe un certain nombre de lentille grâce auxquelles il sera possible de dévier des rayons lumineux vers le capteur.

Le principal avantage de ce type d'optique est de permettre de changer de focale sans avoir à changer d'objectif [4].

I.3.1.2. Distance focale

Objectif est caractérisé par sa distance focale, distance à laquelle se forme l'image d'un point à l'infini, comptée à partir du plan principal d'image. Le plan ainsi défini se place à l'intérieur de l'objectif et se déduit de la constitution de ce dernier (*Figure I.2*).

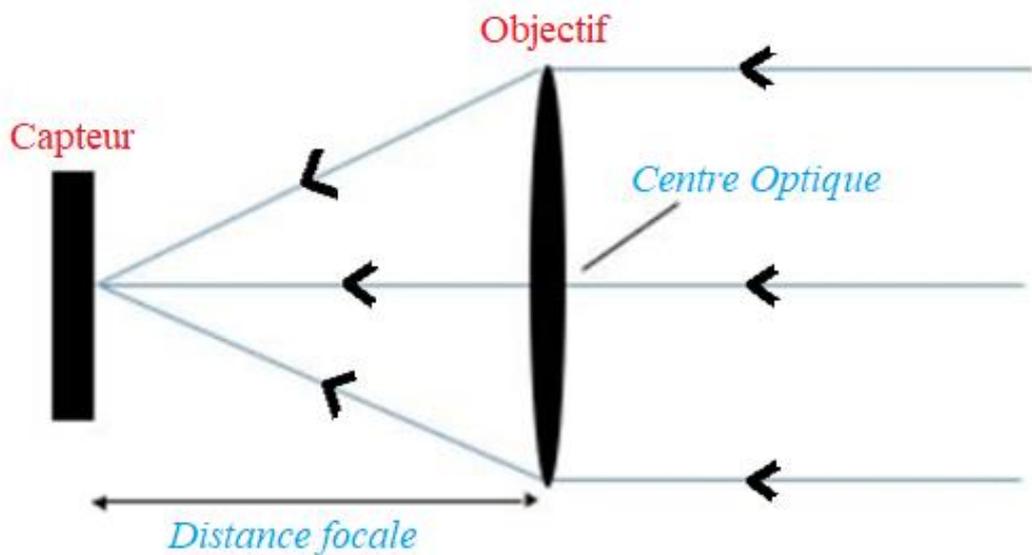


Figure I.2: Schéma du système optique.

Les objectifs considérés comme « normaux » par une vieille règle de pratique photographique doivent présenter une distance focale égale à la diagonale de l'image enregistrée.

Les objectifs de focale plus courte sont dits grands angulaires (ultra Wild) et ceux de focale plus longue sont souvent désignés par le terme téléobjectifs (telephoto).

Le théorème de Thalès permet d'établir la correspondance entre la taille de l'objet et de son image formée sur le capteur avec la distance focale et la distance à laquelle se trouve l'objet observé.

Soit un triangle ABC, et deux points D et E des droites (AB) et (AC) de sorte que la droite (DE) soit parallèle à la droite (BC) (Figure I.3). On obtient alors :

$$\frac{AD}{AB} = \frac{AE}{AC} = \frac{DE}{BC}$$

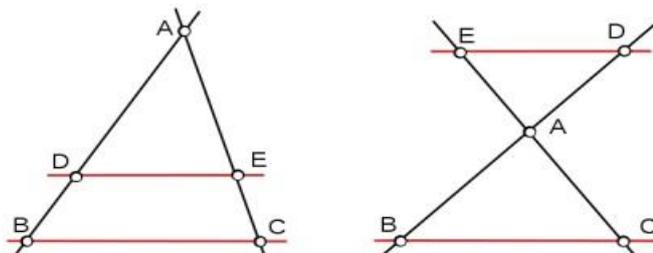


Figure I.3: Théorème de Thalès.

En appliquant ceci a un objectif, on trouve (*Figure I.4*) :

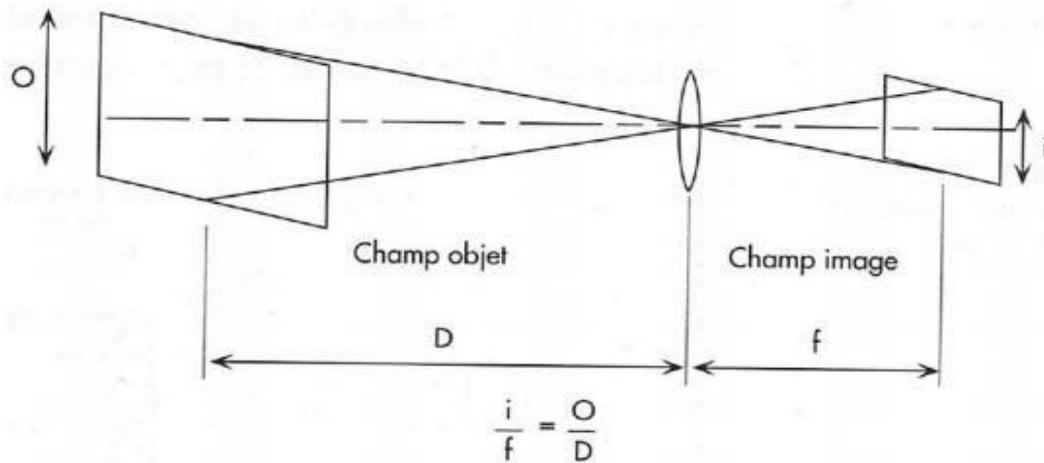


Figure I.4: Application de Thales sur un système optique.

L'image formée sur le capteur sera donc circulaire (comme le système optique) (*Figure I.5*) :

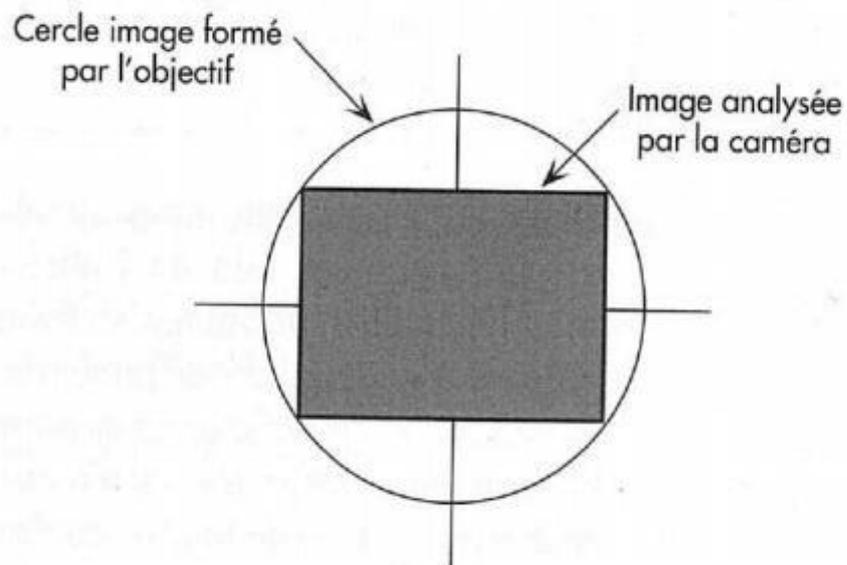


Figure I.5: Apparence de l'image sur le capteur.

I.3.1.3. Champ d'un objectif :

Le champ d'un objectif est l'angle qui correspond à l'ouverture des rayons extrêmes admis par le système. De façon pratique, on considère simplement le champ utile qui résulte directement du format d'image et de la focale de l'objectif. La valeur de ce champ se calcule aisément en fonction de la diagonale de l'image et de la focale de l'objectif [4].

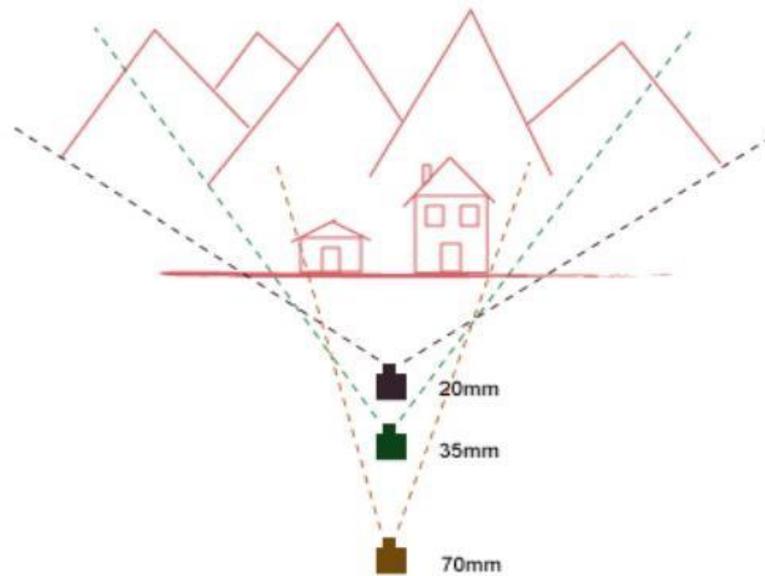


Figure I.6: Exemple des différentes distances focales.

Les valeurs de focales sont les valeurs équivalentes en 35mm :

- Focale de 28mm : grand angle puisqu'il couvre 75° avec une déformation (distorsion).
- Focale de 35mm : grand angle, il couvre 63° avec une diminution des risques de déformation.
- Focale de 50mm : c'est la focale classique qui correspond à peu près à la vision humaine avec un angle de champ de 47° .
- Focale de 135mm : cette focale couvre 18° , utilisé pour des objets relativement éloignés.
- Focale de 200mm : ce téléobjectif donne un champ serré de 12° , la profondeur de champ est alors très faible.

I.3.1.4. Ouverture relative :

Les objectifs comportent tous un diaphragme réglant l'éclairement de l'image par modification de l'ouverture relative (*Figure I.7*). On peut considérer que les rayons lumineux qui pénétreront jusqu'au film sont ceux qui s'appuieront sur le bord du diaphragme : or ces rayons auront déjà subi une réfraction à travers la lentille antérieure, si bien qu'ils correspondront à des rayons initiaux s'appuyant sur le bord de l'image du diaphragme dans la première lentille : cette image s'appelle la pupille entre l'entrée de l'objectif et on désigne son diamètre par la notation f/n .



Figure I.7: Les types d'ouverture des objectifs.

Un calcul photométrique montre l'éclairement de l'image formé par un objectif est inversement proportionnel au carré du coefficient n ; on graduera donc le diaphragme en chiffres croissants selon une progression géométrique de raison $\sqrt{2}=1.414\dots$ soit 1.4 – 2 – 2.8 – 4 – 5.6 – 8 – 11 – 16.

L'éclairement étant divisé par deux à chaque fois que l'on passe d'une ouverture à celle immédiatement supérieure.

Une lentille de diamètre 100mm et de focale 200mm ouvre à $f/2$. (Grande ouverture) Si on passe le diamètre a 50mm, on ouvre à $f/4$. (4fois moins de lumière). Le diaphragme d'une lentille réduit le diamètre de la lentille (donc de laisser passer moins de lumière) [4].

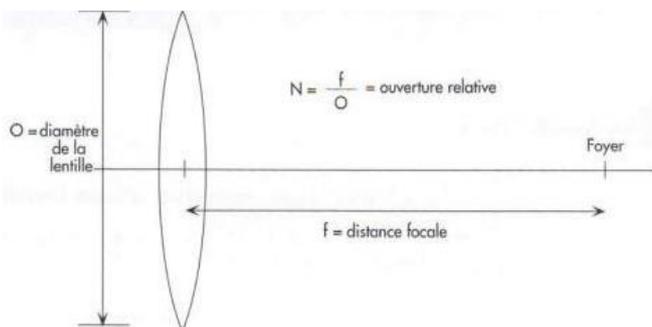


Figure I.8: Représentation de l'ouverture dans un système optique.

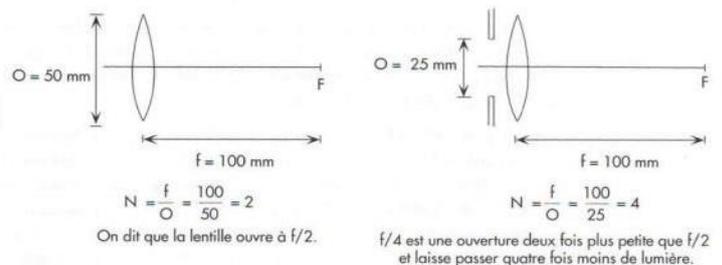


Figure I.9: L'impact d'ouverture sur la qualité de lumière passé.

I.3.1.5. Profondeur de champ et mise au point :

Étant donné un sujet ponctuel placé à l'infini, l'objectif en donne dans le plan focal une image qui peut être assimilée à un point, les rayons lumineux formant cette image définissent un cône ayant pour sommet le point et s'appuyant sur le cercle d'ouverture utile de l'objectif.

Or l'œil perçoit quand même un point comme un cercle de petit diamètre qui se forme si l'image ponctuelle réelle un peu en arrière du plan focal image. En dessous d'un certain seuil de tolérance, l'image sera perçue comme net.

Hyperfocale est la distance la plus courte à laquelle un sujet sera net lorsque la mise au point est réalisée sur l'infini.

On peut remarquer que la profondeur de champ augmente lorsque l'ouverture du diaphragme diminue (*Figure I.10*) :

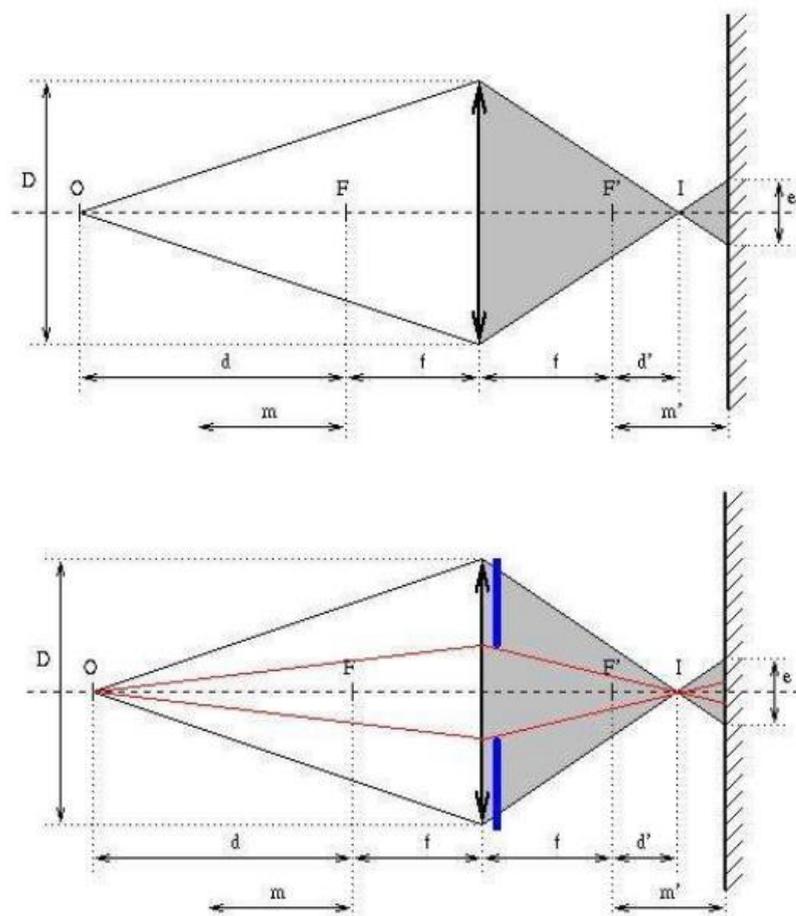


Figure I.10: Impact de taille de capteur sur la profondeur de champ.

Ces schémas montrent quand une ouverture est réduite, le diamètre du cercle de diffusion diminue. C'est pour cette raison que l'on recommande d'utiliser une plus grande ouverture à ceux qui veulent diminuer la profondeur de champ afin de décoller le sujet du décor qui se trouve derrière.

Pour avoir une faible profondeur de champ, il faut donc une longue focale et une grande ouverture de diaphragme (*Figure I.11*) [4].

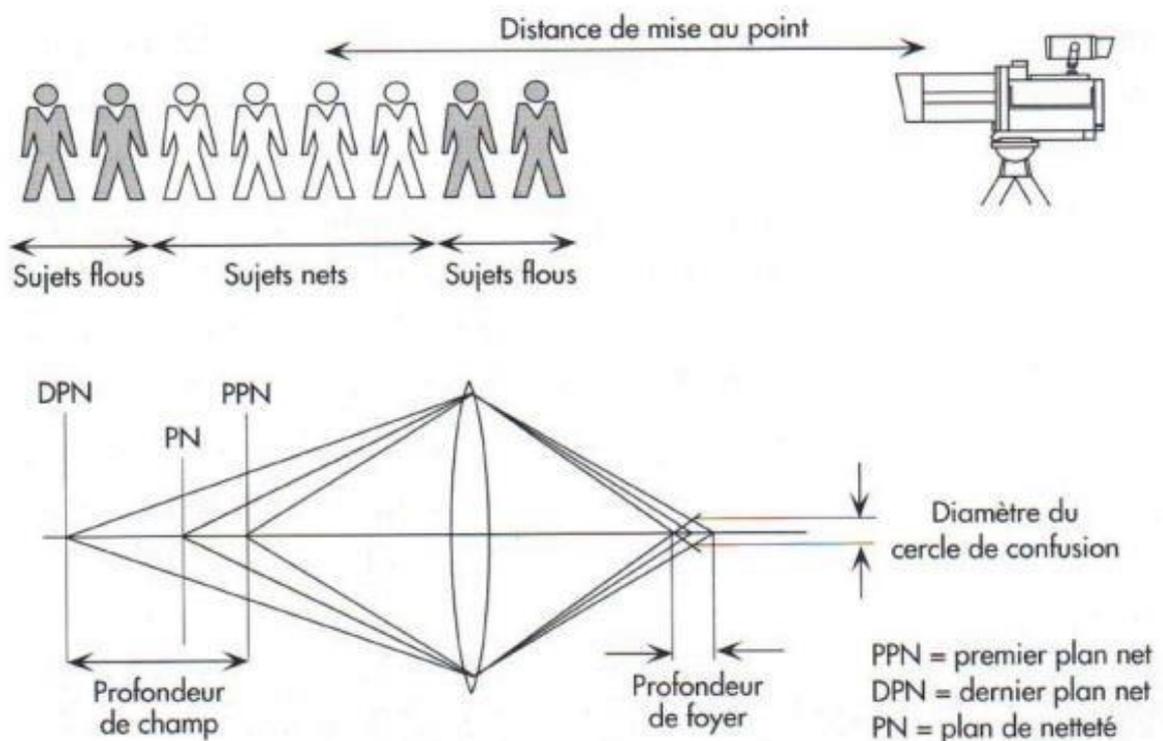


Figure I.11: Schéma de mise au point d'un objet et son profondeur de champ.

I.3.1.6. Obturateur et le temps d'exposition :

Dans un capteur photosensible, l'analyse et la lecture de l'image sont indépendantes. On peut ainsi choisir de réduire la durée d'exposition des cellules lorsqu'il s'agit de capter des mouvements rapides.

L'obturateur permet alors de déterminer une période utile et une période d'obturation durant laquelle les charges créées sont inutilisées. Les images obtenues sont plus nettes (pas de flou de mouvement) mais il faut augmenter l'ouverture pour conserver une quantité de lumière suffisante.

En mode normal, l'obturateur est réglé sur 1/50s, mais il est possible de le régler sur 1/100s, 1/250s, 1/2000s...

On peut aussi utiliser un réglage particulier de l'obturateur pour éliminer les traces disgracieuses qui apparaissent lorsque l'on filme un écran d'ordinateur, ou s'adapter à des conditions de lumières spécifiques (néon, etc....)[4].



Figure I.12: Pose longue (1/50ème s) image floue, notion de mouvement.



Figure I.13: Pose courte (1/1600ème s) image nette, temps figé.

I.3.1.7. Capteur :

Qu'ils soient de type CCD ou CMOS, ce sont tous deux des capteurs photographiques convertissant une image de départ issue d'un rayonnement électromagnétique en un signal analogique (ou numérique) à la fin.

Le principe de base d'un capteur repose sur l'effet photoélectrique, qui permet à un photon incident d'arracher des électrons à chaque élément sensible (appelé photosite) d'une matrice de petits capteurs élémentaires [5].

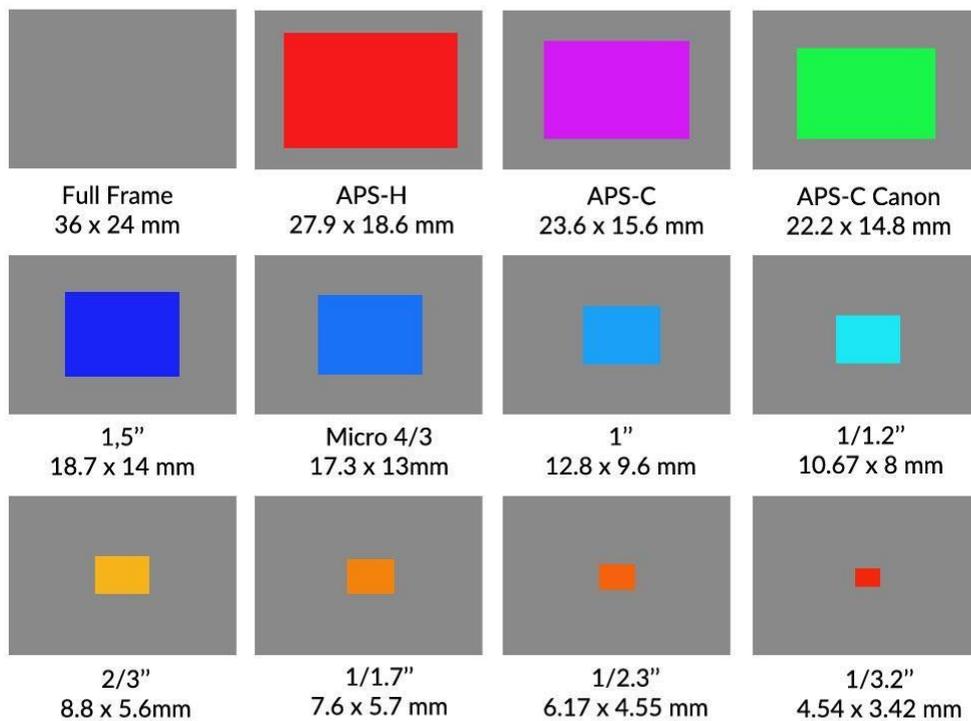


Figure I.14: Types des capteurs.

a. CCD :

Les capteurs appelés CCD (Charged Coupled Device ou en français Dispositif de transfert de Charges) sont des systèmes à transfert de charges (*Figure I.15*).

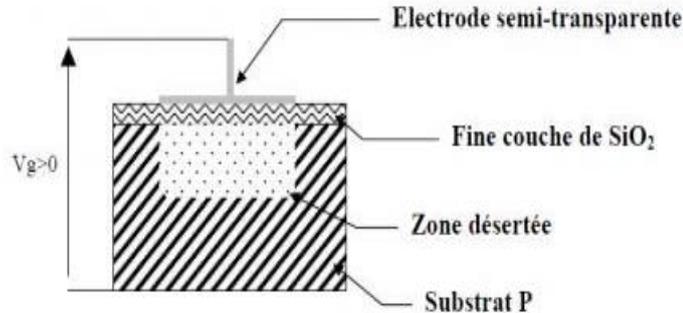


Figure I.15: Jonctions d'un CCD.

Un CCD transforme les photons lumineux qu'il reçoit sur l'électrode semi-transparente en électron par effet photoélectrique, puis collecte les électrons dans la zone désertée. Le nombre d'électrons collectés est proportionnel à la quantité de lumière reçue.

À la fin de l'exposition, les charges sont transférées de photosite en photosite par le jeu de variations de potentiel cycliques provenant d'une horloge externe. Puis elles sont envoyées dans un registre de sortie.

Enfin les charges sont transformées en tension proportionnelle au nombre d'électrons. Ce signal sera, à l'extérieur du CCD, amplifié et numérisé (*Figure I.16*) [5].

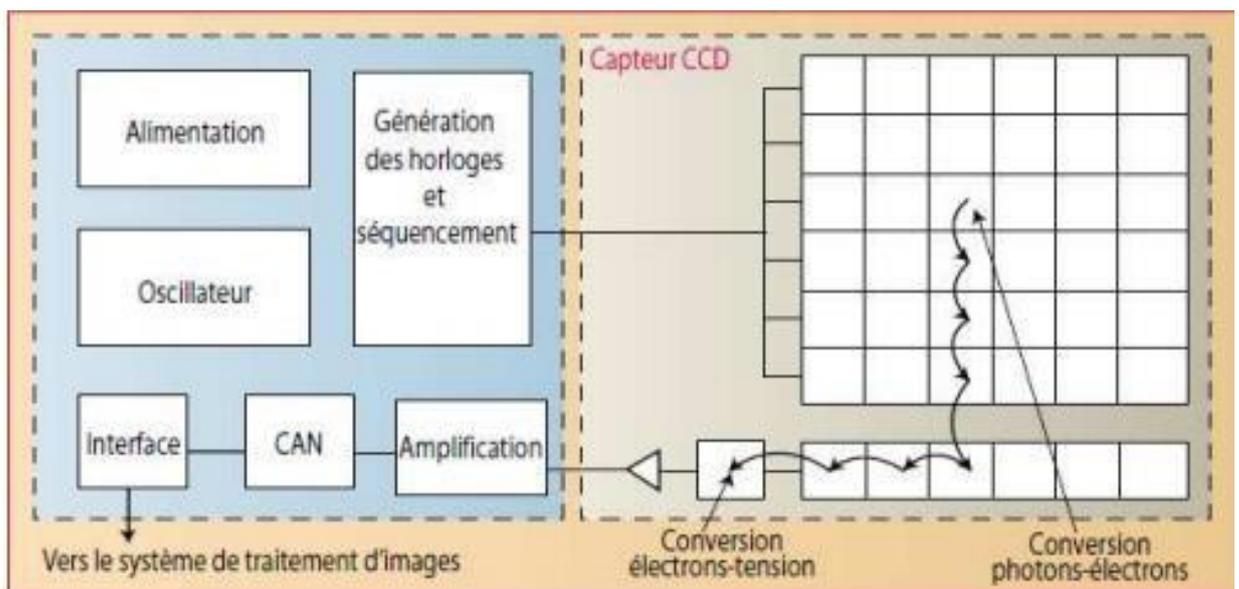


Figure I.16: Camera à capteur CCD.

b. CMOS :

Les capteurs CMOS (Complementary Metal Oxide Semiconductor ou en français Semiconducteur à Oxyde de Métal Complémentaire) fonctionnent différemment, même si le principe de base reste le même (Figure I.17). Les photosites, comme sur un CCD sont sensibles à la lumière et acquièrent une certaine charge électrique en fonction de la quantité de lumière reçue. Jusqu'ici pas de différence.

Mais ensuite, à l'intérieur de chaque photosite, la charge générée est convertie directement en tension utilisable. Cela induit une architecture proche de celle d'un capteur CCD interligne. La surface d'un photosite est en partie occupée par l'électronique, pour convertir la charge en tension [5].

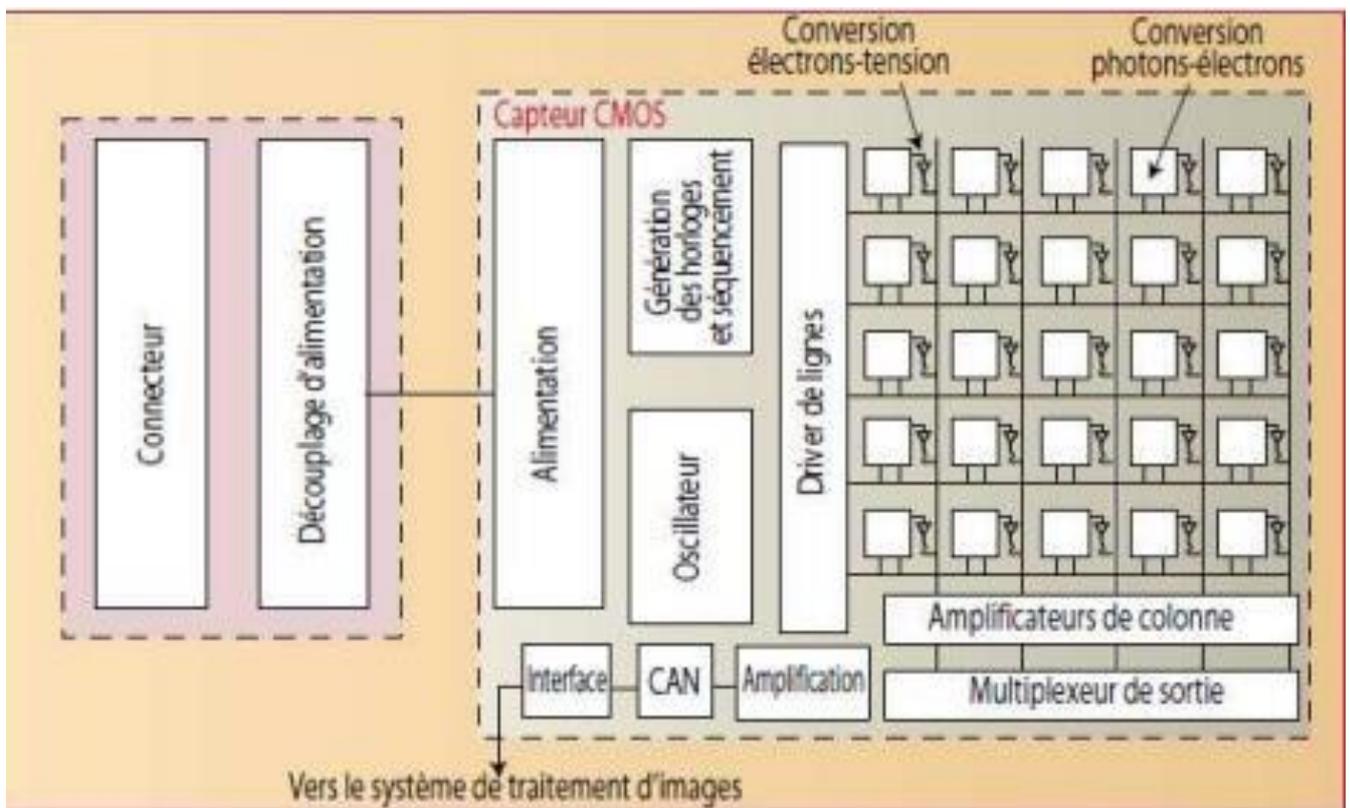


Figure I.17: Caméra à capteur CMOS.

I.3.2. Principe de fonctionnement d'un appareil photo :

Lors de l'appui sur le déclencheur, l'obturateur s'ouvre afin de laisser passer la lumière par l'objectif (lentille) qui va fournir une image nette du sujet à photographier au film ou au capteur numérique. La quantité de lumière entrante est déterminée par le temps durant lequel l'obturateur va rester ouvert et par la taille de l'ouverture du diaphragme. À la fin de l'exposition, l'obturateur reprend sa position fermée initiale (*Figure I.18*).

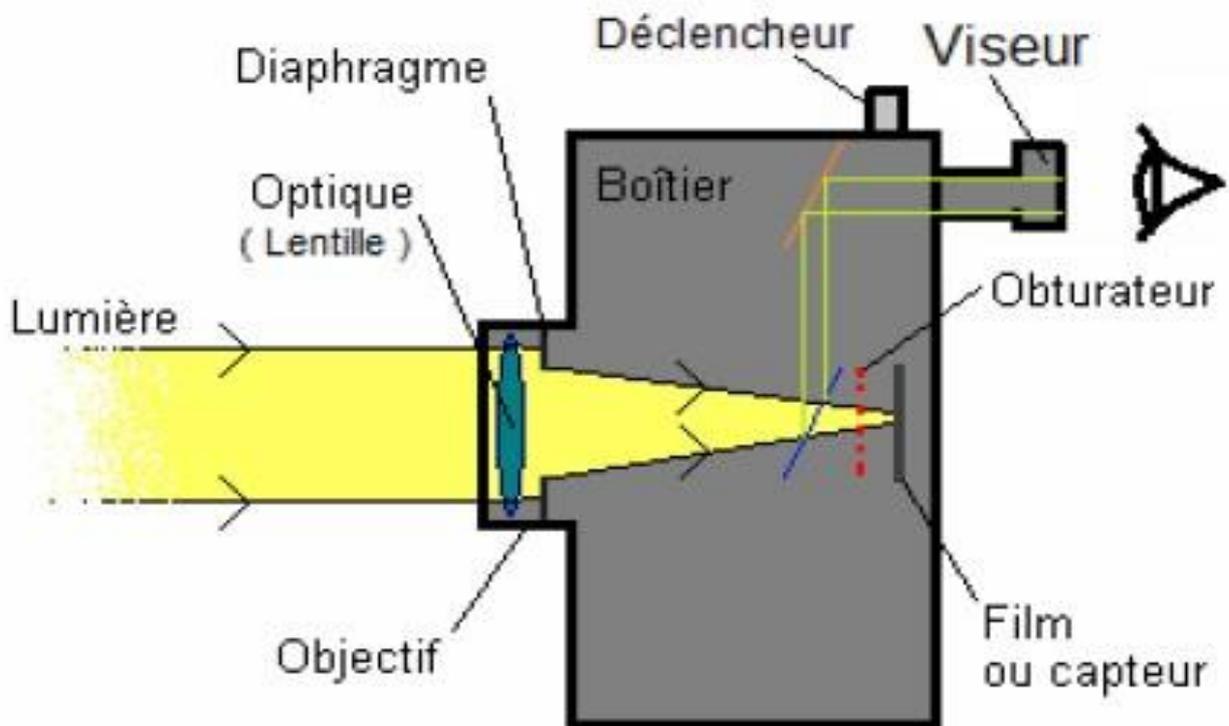


Figure I.18: Principe du fonctionnement d'une caméra.

I.3.3. L'analogie entre l'appareil photo et l'œil :

L'appareil photographique est en quelque sorte un œil artificiel bâti dans le but de « capturer » des images. Il présente diverses similitudes avec le vrai œil (*Figure I.19*). Ces deux récepteurs optiques sont constitués d'une « chambre noire » par laquelle va rentrer la lumière, qui peut être alors diaphragmée, c'est-à-dire que l'on peut moduler la quantité de lumière entrante. Aussi, pour les deux, il y a la possibilité de faire une mise au point de l'image, par le cristallin/objectif, afin que les rayons lumineux convergent et forment une image réelle nette sur les récepteurs optiques. Pour l'œil, ce sera la rétine pour l'appareil photo, on aura un film photographique ou un capteur électronique photosensible. Pour l'accommodation de l'œil, le cristallin se déforme, afin de modifier sa vergence, alors que pour l'appareil photographique, la mise au point est faite en déplaçant l'objectif le long de l'axe optique [6].

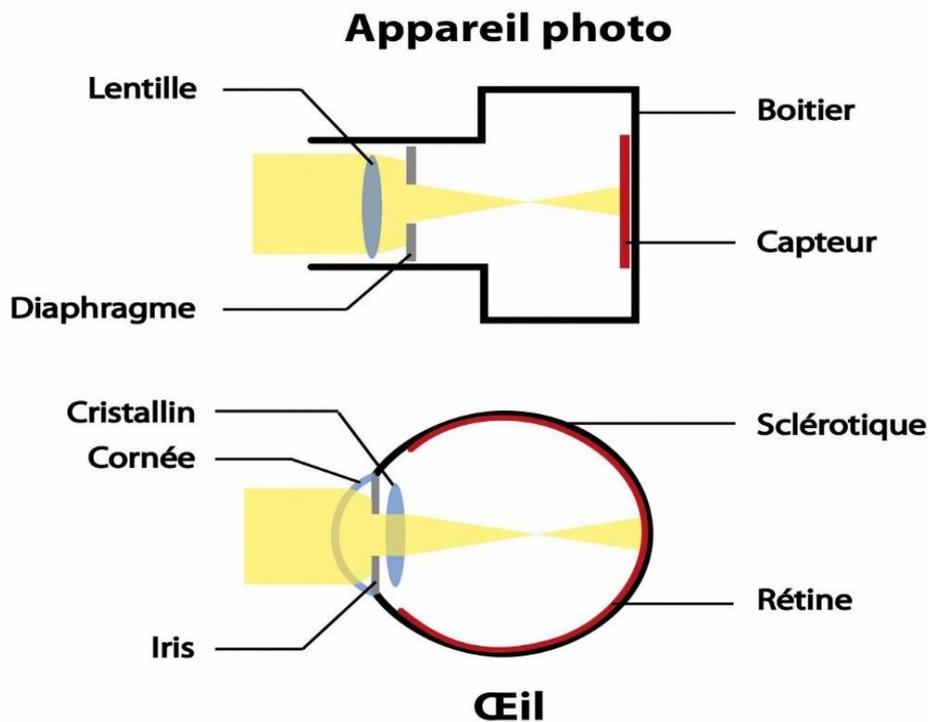


Figure I.19: Comparaison entre la caméra et l'œil humaine.

I.3.4. Camera utiliser : Pi camera v1.3 :

La caméra Raspberry Pi a une résolution de 5 mégapixels est entièrement compatible avec les Raspberry Pi 2B, 3, 3B, 3B+, et 4 (*Figure I.20*). Le Pi camera dispose d'une bibliothèque Python dédiée qui permet aux utilisateurs de contrôler l'appareil photo et de prendre des photos et d'enregistrer des vidéos. La bibliothèque permet également aux utilisateurs de contrôler la balance des blancs, l'obturateur et de lire chaque pixel RVB directement depuis l'appareil photo [7].

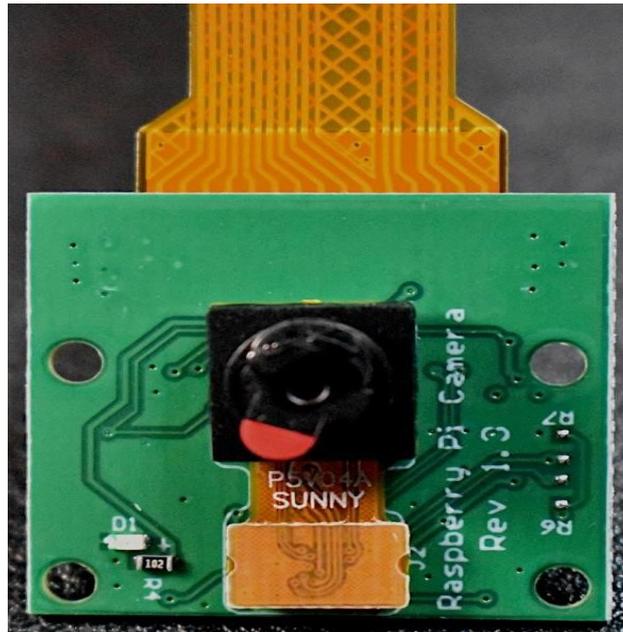


Figure I.20: Illustration de caméra utilisé.

✓ Caractéristiques de la Pi camera :

- Résolution maximale de la photographie 5MP (2592 x 1944 = 5 038 848 pixels).
- Câble ruban qui se fixe directement au Raspberry Pi.
- Taille des pixels : 1,4 x 1,4 μm .
- Objectif : $f=3,6$ mm, $f/2,9$.
- Full-frame équivalent : 35mm.
- Angle de vision : 54° x 41° .
- Résolution vidéo maximale : 1080p @ 30fps.
- Cadence d'image maximale : 480p à 90fps.
- Résolutions vidéo sélectionnables : 1080p @ 30fps, 720p @ 60fps, 480p @ 90fps.
- Taille du capteur : 3,67mm x 2,74mm (format 1/4").
- Dimensions de la carte de circuit imprimé du module de caméra : 25mm x 24mm (épaisseur 9mm).

I.4. Raspberry Pi :

I.4.1. Définition :

Raspberry Pi est le nom d'une série d'ordinateurs mono-carte fabriqué par la fondation Raspberry Pi (*Figure I.21*), une organisation caritative britannique qui vise à éduquer les gens en informatique et à faciliter l'accès à l'enseignement de l'informatique.

Le Raspberry Pi a été lancé en 2012, et il y a eu plusieurs itérations et variations publiées depuis. Le Pi original avait un processeur simple-core de 700MHz et seulement 256MB de RAM, et le dernier modèle a un processeur quad-core pointant dedans à plus de 1.5GHz, et 4GB de RAM.

Partout dans le monde, les gens utilisent le Raspberry Pi pour acquérir des compétences en programmation, construire des projets de matériel, faire de la domotique, mettre en œuvre des clusters Kubernetes et de l'informatique de pointe, et même les utiliser dans des applications industrielles.

Le Raspberry Pi est un ordinateur qui fonctionne sous Linux, il fournit également un ensemble de broches GPIO (entrée/sortie générale), qui permettant de contrôler les composants électroniques pour l'informatique physique et d'explorer l'Internet des objets (IoT) [8].



Figure I.21: Illustration de Raspberry Pi 4.

I.4.2. Caractéristiques :

a-Hardware :

- Quad core 64 bits ARM-Cortex A72 fonctionnant à 1,5 GHz.
- 2 Giga-octets LPDDR4 RAM .
- Décodage matériel H.265 (HEVC) (jusqu'à 4Kp60) .
- Décodage matériel H.264 (jusqu'à 1080p60) .
- VideoCore VI Graphiques 3D .
- Prend en charge une double sortie d'affichage HDMI jusqu'à 4Kp60.

b-Interface :

- LAN sans fil 802,11 b/g/n/ac .
- Bluetooth 5.0 avec BLE .
- 1x carte SD .
- 2x ports micro-HDMI supportant des écrans doubles jusqu'à une résolution de 4Kp60
- 2x ports USB2 .
- 2x ports USB3 .
- 1x port Gigabit Ethernet (RJ45) .
- 1x port caméra Raspberry Pi (MIPI CSI à 2 voies) .
- 1x port d'affichage Raspberry Pi (DSI MIPI à 2 voies) .
- 28x GPIO utilisateur prenant en charge diverses options d'interface :
 - Jusqu'à 6x UART.
 - Jusqu'à 6x I2C.
 - Jusqu'à 5x SPI.
 - 1x interface SDIO.
 - 1x DPI (affichage RVB parallèle).
 - 1x PCM.
 - Jusqu'à 2 canaux PWM.
 - Jusqu'à 3 sorties GPCLK.

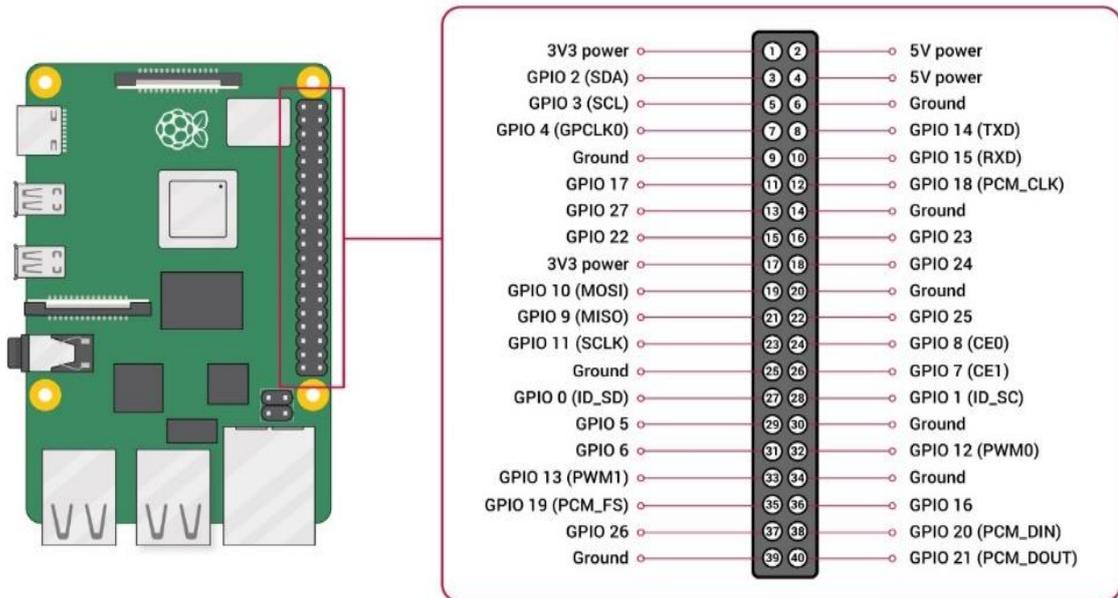


Figure I.22: Configuration des pins de Raspberry Pi.

c. Software :

- Ensemble d'instructions ARMv8.
- Pile logicielle Linux mature.
- Actively developed and maintained.
- Disponibilité des fonctions GPU à l'aide des API standard.

I.5. Servomoteur :

I.5.1. Principe général :

Un servomoteur est avant tout un dispositif asservi en position.

L'industrie est le précurseur de l'utilisation d'équipements asservies en position, le terme de servomoteur a donc une origine industrielle. À l'origine le servomoteur est un pré actionneur, a plupart du temps pneumatique (système membrane/ressort), plus rarement hydraulique ou électrique d'une vanne de régulation. Dans ce cas l'asservissement de position est toujours effectué par un organe distinct nommé positionneur.

C'est un système motorisé capable d'atteindre des positions prédéterminées, puis de les maintenir. La position est : dans le cas d'un moteur rotatif, une valeur d'angle et, dans le cas d'un moteur linéaire une distance. Le démarrage et la conservation de la position prédéterminée sont commandés par un système de réglage.

Le système de réglage souvent électronique compare le signal à une valeur prescrite de la position de consigne. S'il y a une déviation, le moteur est commandé dans la direction qui garantit le plus petit chemin à effectuer pour arriver à la valeur de consigne. Cela a pour conséquence de faire diminuer l'écart.

La procédure se répète aussi longtemps et jusqu'à ce que la valeur courante se trouve incrémentalement ou par l'intermédiaire d'une approximation dans les seuils de tolérance de la valeur consigne. Alternativement, la position du moteur peut être saisie aussi numériquement et comparée via un ordinateur approprié à une valeur prescrite [9].

I.5.2. Fonctionnement :

Mécaniquement, il s'agit d'un moteur avec réducteur. Le mouvement de sortie est une rotation. Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre. Le rôle de l'électronique est alors de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue.

La commande se fait via une petite impulsion, une consigne de position *codé en largeur d'impulsion* plus communément appelé **PWM** (**P**ulse **W**ide **M**odulation en français **MLI** **M**odulation à **L**argeur d'**I**mpulsion) (*Figure I.23*). Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

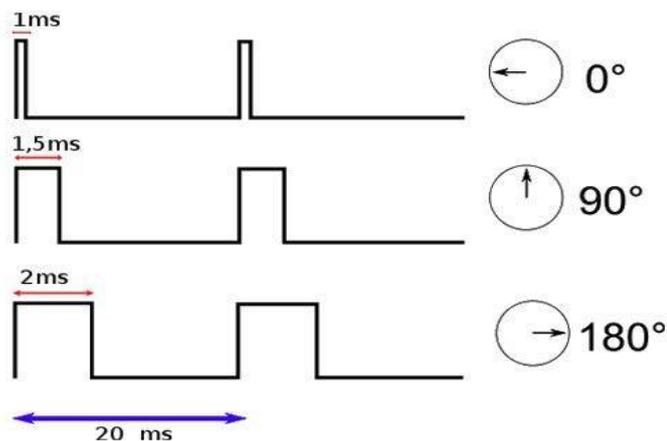


Figure I.23: Signal PWM.

La durée de cette impulsion de consigne est comprise entre 1ms et 2ms. La valeur intermédiaire de 1,5ms donne au servomoteur la position centrale, à 90° en général. *Commande en largeur d'impulsion* : la largeur des impulsions commande la position du servomoteur.

L'augmentation de l'amplitude des impulsions agit sur le contrôle de la vitesse jusqu'au moment où le bras du servomoteur se trouve dans la position souhaitée. Durant la rotation, le potentiomètre indique au circuit électronique le moment où la position désirée est obtenue. Les impulsions diminuent alors d'amplitude jusqu'à ce que plus aucune tension soit appliquée au moteur du servomoteur tout en conservant le bras dans sa nouvelle position [9].

I.5.3. Spécifications de moteur utilisé MG996R :

- Poids : 55 g .
- Dimension : 40,7 x 19,7 x 42,9 mm environ.
- Couple de décrochage : 9,4 kgf cm (4,8 V), 11 kgf cm (6 V).
- Vitesse de fonctionnement : 0,17 s/60° (4,8 V), 0,14 s/60° (6 V) .
- Tension de fonctionnement : 4,8 V à 7,2 V .
- Courant de fonctionnement 500 mA –900 mA (6 V) .
- Courant de décrochage 2,5 A (6 V) .
- Largeur de bande morte : 5 µs .
- Conception à double roulement à billes stable et résistante aux chocs .
- Plage de température : 0 °C à 55 °C [10] .

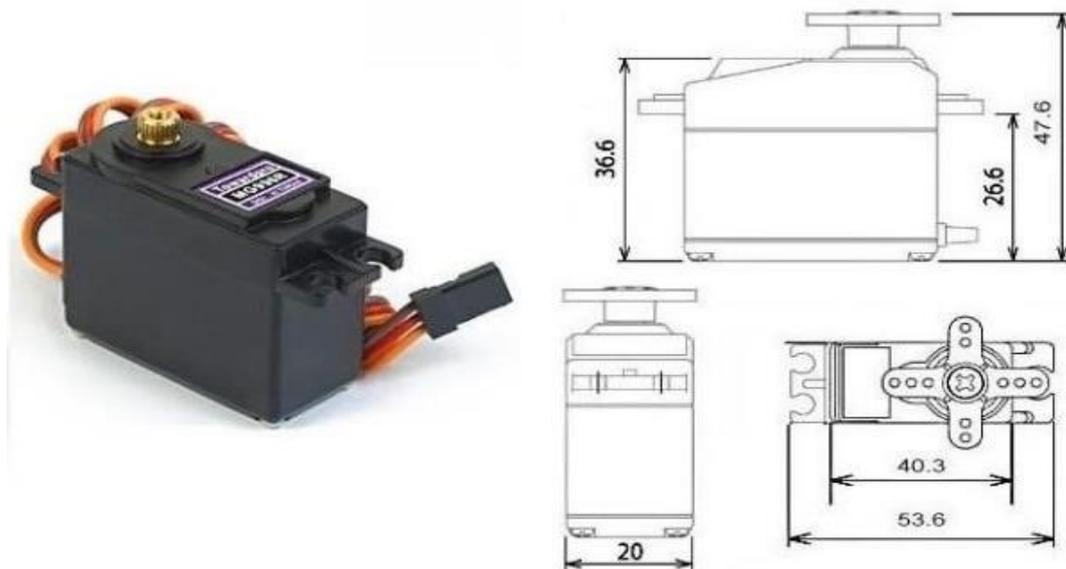


Figure I.24: Dimension de moteur utilisé MG996R.

PWM=Orange (⏏)
 Vcc = Red (+)
 Ground=Brown (-)

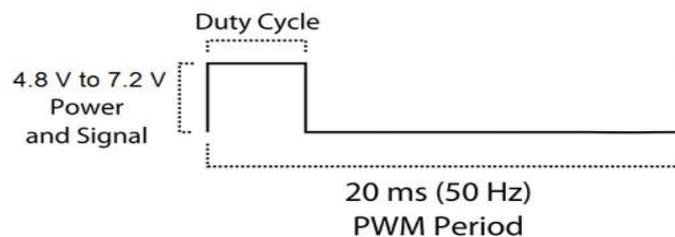


Figure I.25: Configuration des câbles.

I.6. Moteur driver :

I.6.1. Description :

Ce module est une carte de découpage pour le contrôleur PWM 16 canaux NXP PCA9685 (Figure I.26).

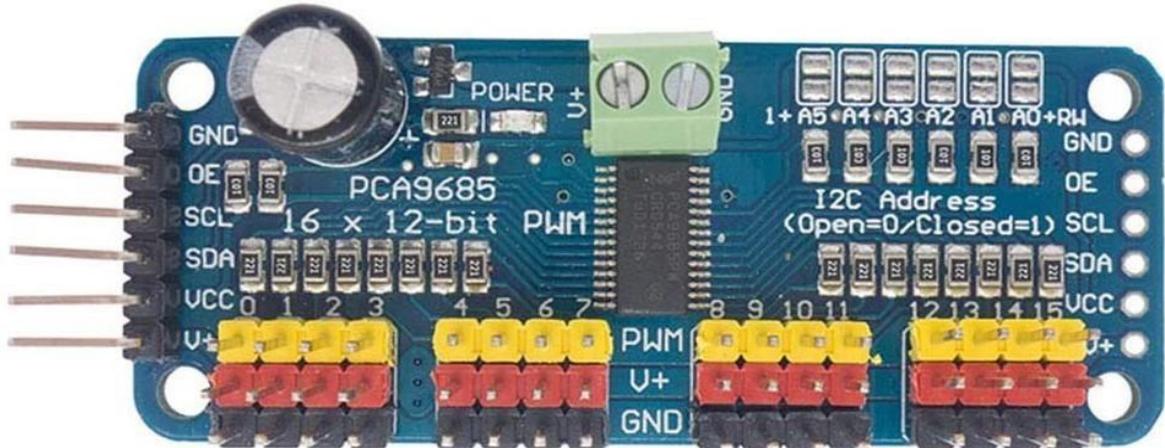


Figure I.26: Illustration de module PCA9685 utilisé.

Il dispose de 16 sorties PWM entièrement programmables avec une résolution 12bit donnant un total de 4096 pas programmables avec un cycle de service réglable de 0% à 100%. En outre, la fréquence de sortie des 16 canaux peut être programmée de 24Hz à 1526Hz.

La programmabilité de ses sorties PWM signifie qu'il est également configuré pour produire des signaux PWM compatibles avec les servomoteurs standard.

En fait, ce module a été conçu dans cette optique avec 16 jeux de headers qui permettent à tout servo avec un en-tête standard d'être directement branché dans le module.

Le module comprend également un en-tête I2C avec des résistances de pullup 10K et nécessite donc seulement deux broches de données (SDA et SCL) pour contrôler le module. Les pads à souder sur le module permettent de changer l'adresse I2C par défaut (0x40) en une des 62 options, ce qui signifie que plus d'un module peut être connecté au même bus I2C [11].

I.6.2. Spécifications :

- C'est un pilote PWM i2c avec horloge intégrée.
- Compatible 5V .
- 6 broches de sélection d'adresse .
- PWM de fréquence réglable jusqu'à environ 1,6 KHz .
- Résolution de 12 bits pour chaque sortie - pour les servos, cela signifie une résolution d'environ 4 US à une fréquence de mise à jour de 60 Hz .
- Sortie push-pull ou open-drain configurable .
- Protection de polarité inversée sur l'entrée du bornier .
- Un spot pour placer un grand condensateur sur la ligne V+ .

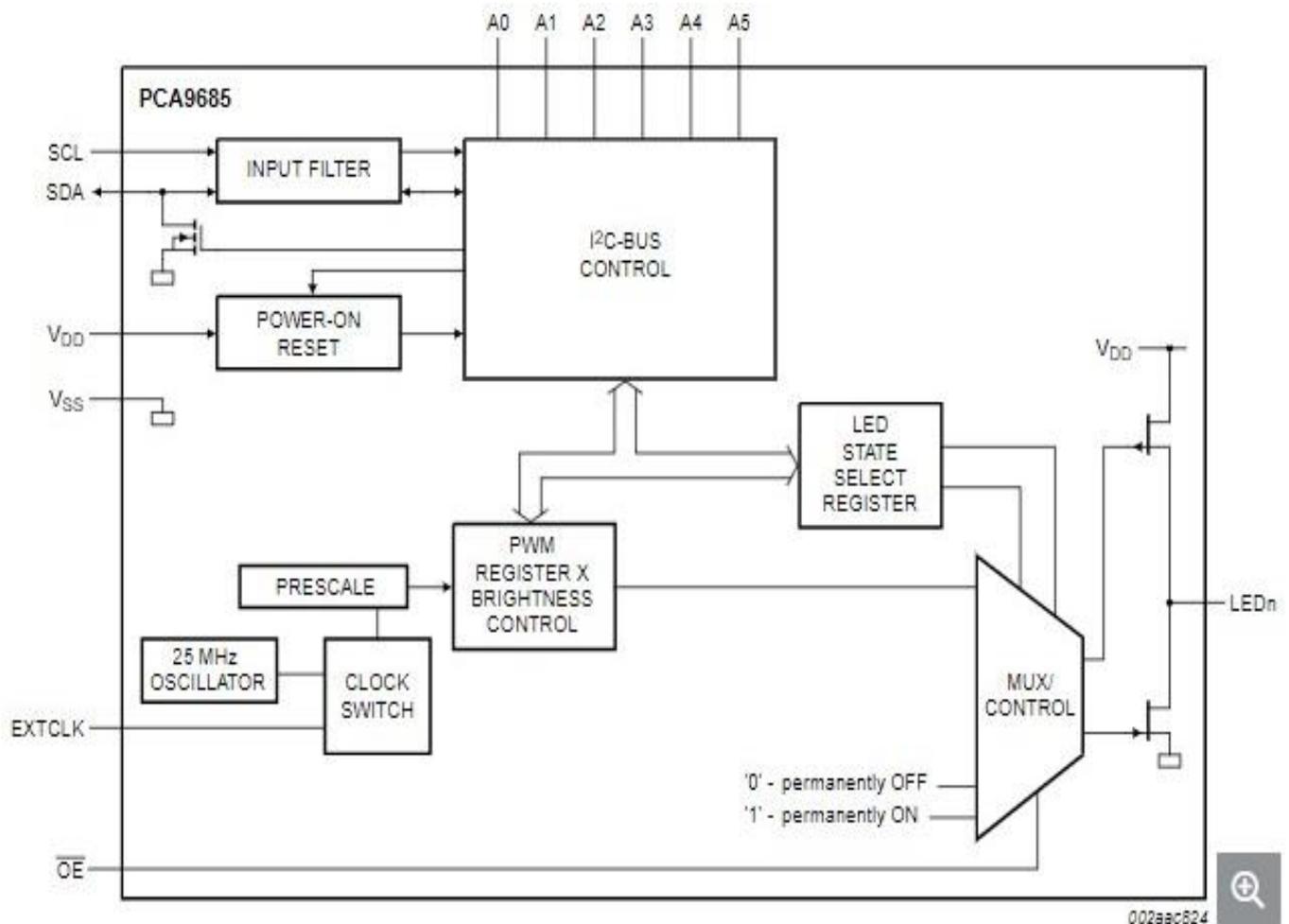


Figure I.27: Schéma fonctionnel de PCA9685.

I.7. Capteur infrarouge :

I.7.1. Définition :

Le détecteur de mouvement infrarouge se compose d'un capteur thermique, d'une centrale électronique et d'un puissant système d'éclairage. Lorsque le capteur détecte un mouvement suspect, la partie électrique du système analyse les données et commande l'activation de la lumière. C'est un dispositif énergétiquement économique puisqu'il ne nécessite pas l'utilisation d'interrupteurs. Vous paramétrez la durée d'éclairage au moment de l'installation pour gérer efficacement votre consommation.

Dans certains cas, le détecteur à infrarouge est relié à une sirène et éventuellement à un centre de télésurveillance.

De nombreux modèles de nouvelle génération disposent d'une caméra ou un appareil photo intégré. Les individus détectés sont alors immédiatement filmés ou pris en photo, permettant la confirmation du danger et l'appréhension des intrus [12].

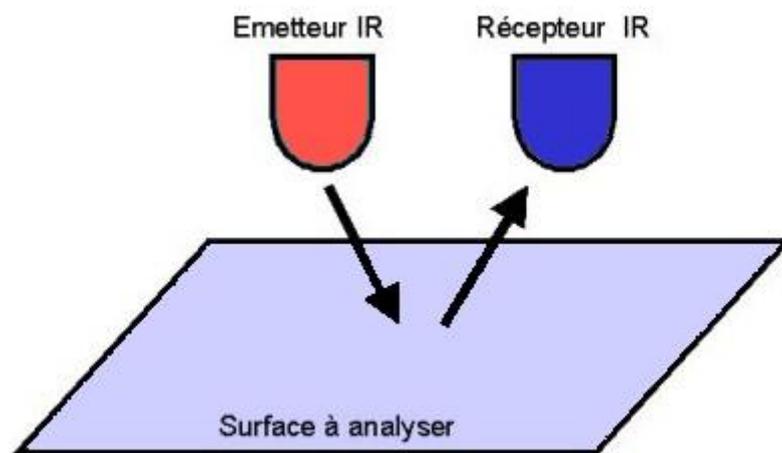


Figure I.28: Principe de capteur infrarouge.

I.7.2. Principes de la détection infrarouge :

Les dispositifs de détection infrarouge possèdent des capteurs qui détectent le rayonnement dans la partie infrarouge du spectre électromagnétique. Soit une longueur d'onde comprise entre 700 nm et 1 millimètre. Ou bien quelques centaines de Gigahertz à plus de 400 THz en termes de fréquence. Juste entre la lumière visible et le rayonnement micro-ondes. Ensuite la chaîne d'acquisition exploite cette mesure analogique. Il y a deux manières principales d'en tirer parti. Soit le signal est converti par décalage de spectre dans le visible et restitué à l'opérateur comme c'est le cas en astronomie (on voit rouge mais il s'agit en fait d'infrarouge), soit le système d'asservissement exploite directement le signal, comme cela est le cas pour le guidage d'un missile.

La lumière infrarouge se situant juste au-delà de la limite visuelle humaine (longueur d'onde située entre 700 nm et 2.5 μm) est appelée conventionnellement « infrarouge proche ». Tandis que la lumière plus éloignée du spectre visible se divise en infrarouges moyen et lointain (entre 2.5 μm et 1 mm). Tous les objets au-dessus du zéro absolu brillent dans l'infrarouge lointain, de sorte qu'aucune source d'éclairage n'est nécessaire pour visualiser un tel rayonnement. En revanche, pour une observation dans l'infrarouge proche, il faut également employer un éclairage spécifique avec une diode électroluminescente ou une ampoule à filtre [13].

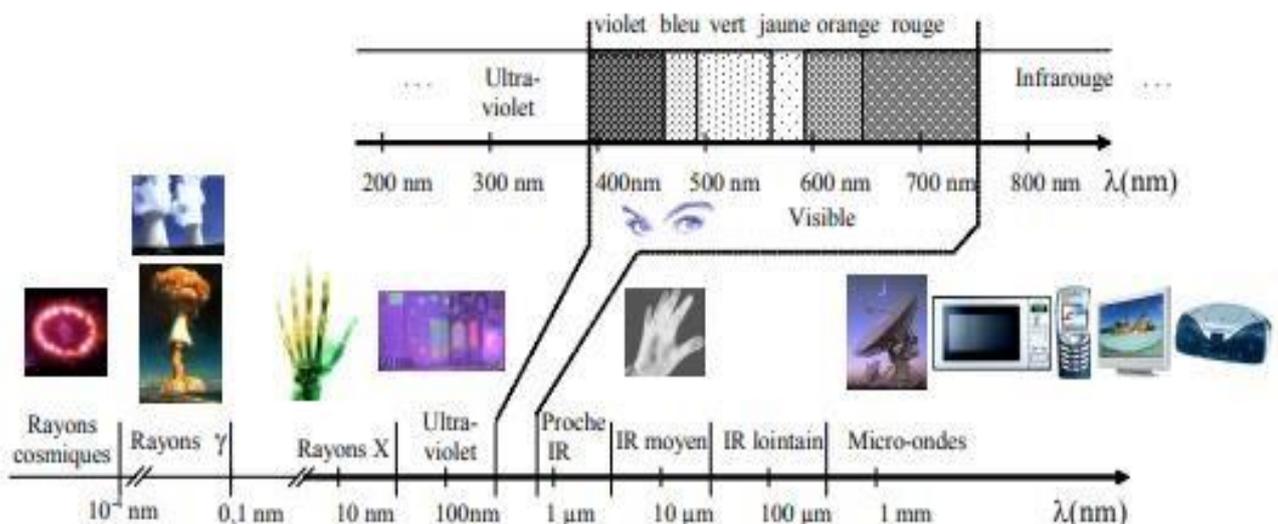


Figure I.29: Spectre des rayonnements électromagnétiques.

I.7.3. Spécifications de capteur utilisé :

C'est un capteur photoélectrique qui intègre l'émission et la réception. La distance de détection peut être ajustée en fonction des besoins. Le capteur a une longue distance de détection et peut être accepté avec des interférences lumineuses faibles [14].

- Portée de détection : 3-80cm
- Fonctionnement de sortie : normalement ouvert (NO)
- Sortie : système a trois fils DC (NPN)
- Sortie de courant DC/SCR/relais : alimentation 100mA/5V
- Consommation actuelle DC < 25mA
- Temps de réponse de réponse : 2ms
- Matériel de logement : plastique
- Température de travail : -25 c ~ + 55 c

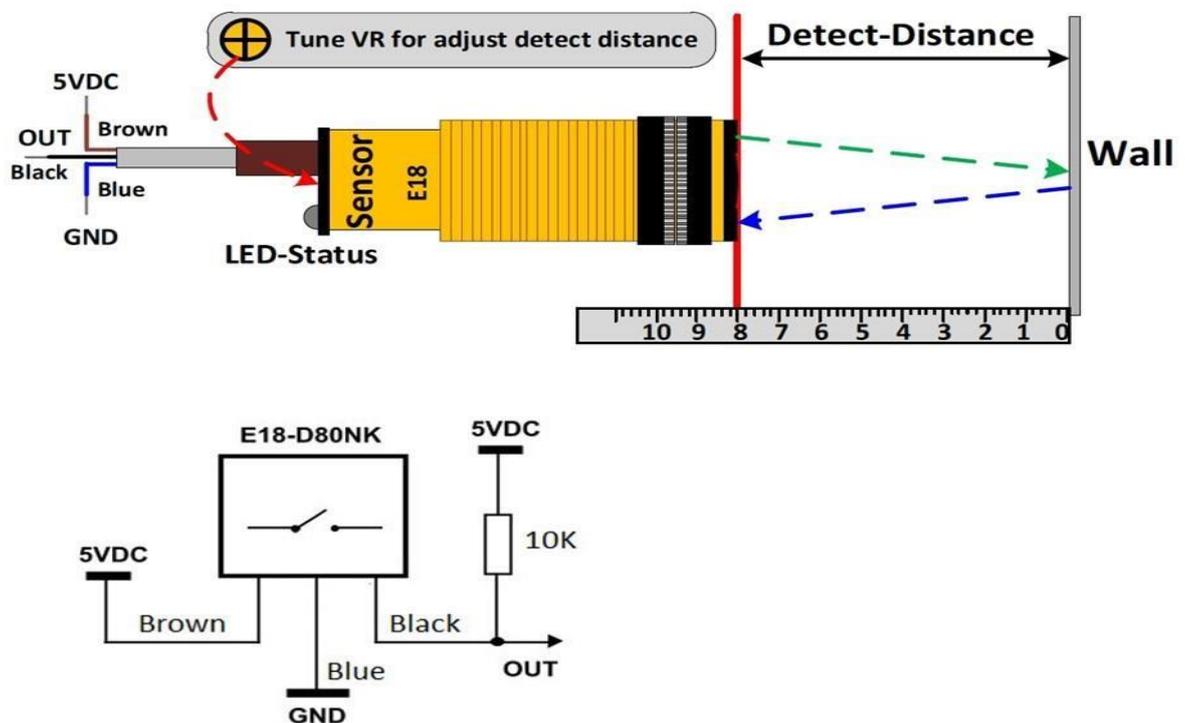


Figure I.30: Configuration des câbles du capteur utilisé.

I.8. Conclusion

La vision humaine est extrêmement complexe. La vision par ordinateur cherche à comprendre ou à reproduire la vision humaine, mais à construire un modèle algorithmique qui vu de l'extérieur possède des propriétés semblables, d'où la vision artificielle est d'extraire des caractéristiques à partir des images. C'est une mise en correspondance image/reconnaissance. La reconnaissance d'objets dans les images est un des problèmes les plus difficiles en vision par ordinateur puisqu'il est indispensable pour toutes sortes d'application industrielles, connaître la forme et le nombre sur un convoyeur nécessite une bonne acquisition. Le choix du type de caméra et de convoyeur est très important, pour arriver à des bons résultats de notre processus de reconnaissance d'objets mais ils ne sont pas les seuls paramètres, Il y en a d'autres pour le traitement numérique de l'image, ou un appareillage plus complexe est indispensable. Après avoir terminé le premier chapitre, nous passons au chapitre suivant, qui est lié à l'étude des méthodes et des techniques de traitement d'image.

Chapitre II

**Les techniques de traitement
d'image**

II.1. Introduction :

Avant de plonger directement dans le traitement de l'image, comprenons d'abord les images. Une image, comme les humains le voient, est une grille bidimensionnelle avec chaque cellule dans la grille remplie d'une couleur, autrement appelée une valeur de pixel. Chaque cellule de la grille est formellement appelée un élément d'image (couramment abrégé en pixels). Un ordinateur voit également l'image de la même façon. Une image sur un ordinateur est une matrice bidimensionnelle de nombres avec chaque cellule de la matrice stockant la ou les valeurs de pixels correspondantes dans l'image. La figure suivante est un exemple de matrice d'images [15]. La matrice de la partie d'image dans l'encadré rouge s'affiche à droite :

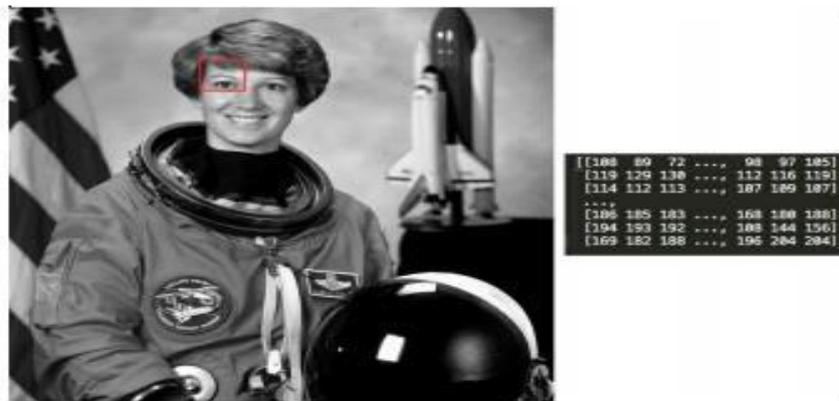


Figure II.1: Stockage d'une image sur un ordinateur.

Le traitement d'images est le domaine de l'étude et de l'analyse d'images. Il y a beaucoup des informations cachées dans une image que nous traitons inconsciemment. Par exemple, ce qui sont les différents objets dans l'image ? Y a-t-il une voiture dans l'image ? Quelles sont les similitudes entre deux images ? Les réponses à ces questions peuvent sembler simples pour nous les humains, mais pour un ordinateur, répondre à de telles questions est extrêmement difficile. Nous visons à mettre en œuvre certains des algorithmes qui peut nous aider à répondre à certaines de ces questions.

L'essence du traitement d'image est d'utiliser les différentes propriétés d'une telle image en couleur, Co-relations entre différents pixels, emplacements d'objets et autres fins détails pour extraire des informations significatives telles que les bords, les objets et les contours, qui sont formellement appelées des caractéristiques d'image. Ces caractéristiques peuvent ensuite être utilisées dans les applications telles que la médecine, la sécurité, les services de médias sociaux et les voitures autonomes [15].

II.2. Notions d'une image numérique :

II.2.1. Pixels :

Les pixels sont les blocs de construction bruts d'une image. Chaque image se compose d'un ensemble de pixels. Il n'y a pas de granularité plus fine que le pixel. Normalement, un pixel est considéré comme la « couleur » ou « intensité » de la lumière qui apparaît à un endroit donné dans notre image. Si nous pensons à une image comme une grille, chaque carré contient un seul pixel [16]. Pour exemple, (figure II.2) :



Figure II.2: Cette image contient 1000 pixels de largeur et 750 pixels de hauteur, pour un total de $1000 \times 750 = 750000$ pixels totaux.

L'image de la figure ci-dessus a une résolution de 1000 x 750, ce qui signifie qu'il est 1000 pixels de large et 750 pixels de haut. Nous pouvons conceptualiser une image comme une matrice (multidimensionnelle). Dans ce cas, notre matrice a 1000 colonnes (la largeur) avec 750 lignes (la hauteur). Globalement, il y a $1000 \times 750 = 750000$ pixels au total dans notre image.

La plupart des pixels sont représentés de deux façons :

1. Niveaux de gris/canal unique.
2. Couleur.

II.2.2. Espaces de couleur et canaux :

Comme dans le monde des mathématiques, nous avons différents systèmes de coordonnées : par exemple, un plan cartésien 2-D et des coordonnées polaires 2-D. Un point pourrait être stocké comme (x, y) ou (r, θ) . Chaque système de coordonnées à un cas d'utilisation spécifique, qui fait des calculs plus faciles. De même, dans le monde du traitement d'image, nous avons différents espaces de couleur. Une image peut stocker ses valeurs de couleur sous forme de Rouge, Bleu, Vert (RVB) ou Cyan, Magenta, Yellow, Key (noir) (CMJN) [15].

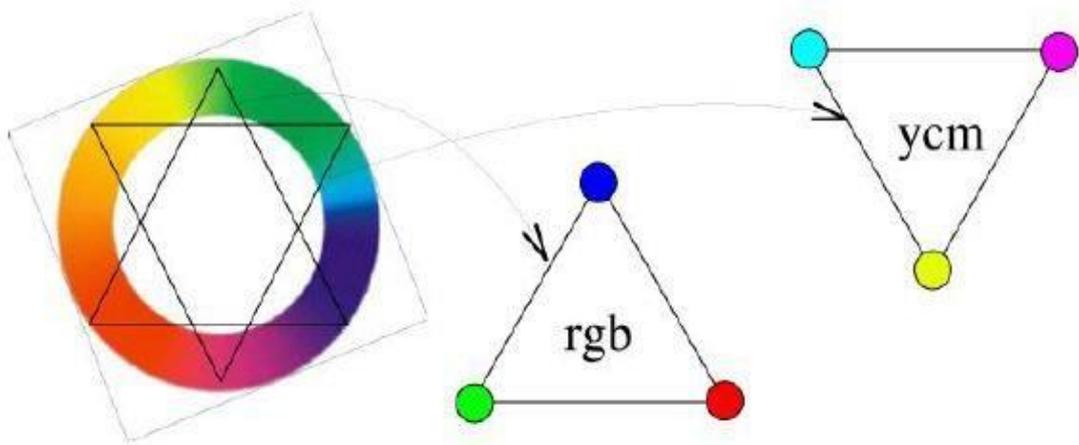


Figure II.3: Espaces de couleurs.

Quelques exemples d'espaces de couleur sont HSV, HSL, CMY. Chaque composante dans l'espace de couleur est appelée un canal de couleur. Par exemple, dans l'espace de couleur RVB, nous disons que Rouge, Bleu, et vert chacun est des canaux de l'image. Une image peut être représentée dans de nombreux différents modes (espaces de couleur) tels que RVB, CMJN, niveau de gris et YUV. Les couleurs dans l'image que nous voyons sont dérivées par le mélange des couleurs dans chaque couleur canal de l'espace de couleur. Regardons certains des espaces de couleur communs en détail :

II.2.2.1. Niveaux de gris :

C'est l'un des espaces de couleur les plus simples en termes de compréhension et stockage sur un ordinateur. Chaque valeur de pixel dans une image en niveaux de gris est une valeur unique comprise entre 0 et 255, 0 représentant le noir et 255 représentant le blanc. Gardez à l'esprit que la valeur 255 n'est pas une valeur fixe, mais dépend de la profondeur de l'image. Les images en niveaux de gris sont aussi parfois appelées images en noir et blanc, mais elles ne sont pas entièrement exactes. Une image en noir et blanc signifie que les valeurs de pixels ne peuvent être que 0 ou 255 et rien entre les deux [15].



Figure II.4: Exemple d'image en niveaux de gris.



Figure II.5: Gradient d'image montrant les valeurs de pixels allant du noir (0) au blanc (255).

II.2.2.2. Espace RVB :

C'est l'un des espaces de couleur les plus communs qui est utilisé dans le monde du traitement d'images et ailleurs. La plupart des images sur Internet ou dans les livres sont dans l'espace RGB. Dans un RGB typique image, chaque pixel est une combinaison de trois valeurs, chacune représentant une couleur dans canaux rouge, vert et bleu. Blanc dans l'espace RVB est écrit comme (255, 255,255) et le noir (0, 0, 0). Le rouge, le vert et le bleu sont représentés par (255, 0, 0), (0, 255, 0) et (0, 0, 255) respectivement. Toute autre couleur n'est qu'une combinaison de certaines valeurs de rouge, vert et bleu [16].

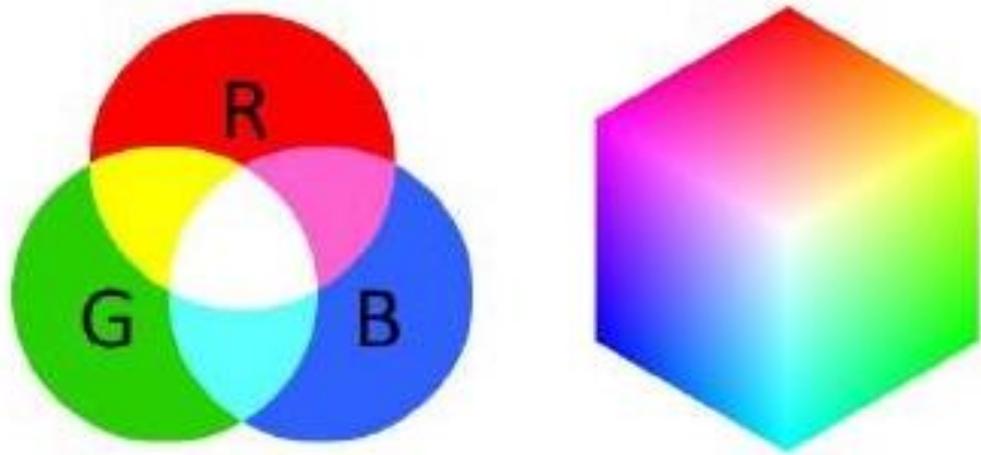


Figure II.6: Représentation de l'espace RGB.

L'espace de couleur RVB est également communément visualisé comme un cube (Figure 36). Dans cette espace une couleur est défini comme un tuple à 3 valeurs, chaque valeur dans la plage [0 ;255] nous pouvons donc penser à un cube contenant $256 \times 256 \times 256 = 16,777,216$ couleurs possibles, selon combien de Rouge, Vert, et bleu sont placés dans chaque seau [16].

II.2.2.3. Espace TSL :

L'espace TSL (Tente, Saturation, lumière) est un mode de coloration des images qui se révèle souvent plus efficace que le système classique RGB (Red, Green, Blue), notamment pour les images fractales. Dans cet article nous verrons comment passer d'un système à l'autre, grâce à des procédures écrites en PANORAMIC [17].

Le système TSL est représenté en coordonnées cylindriques sur cette image :

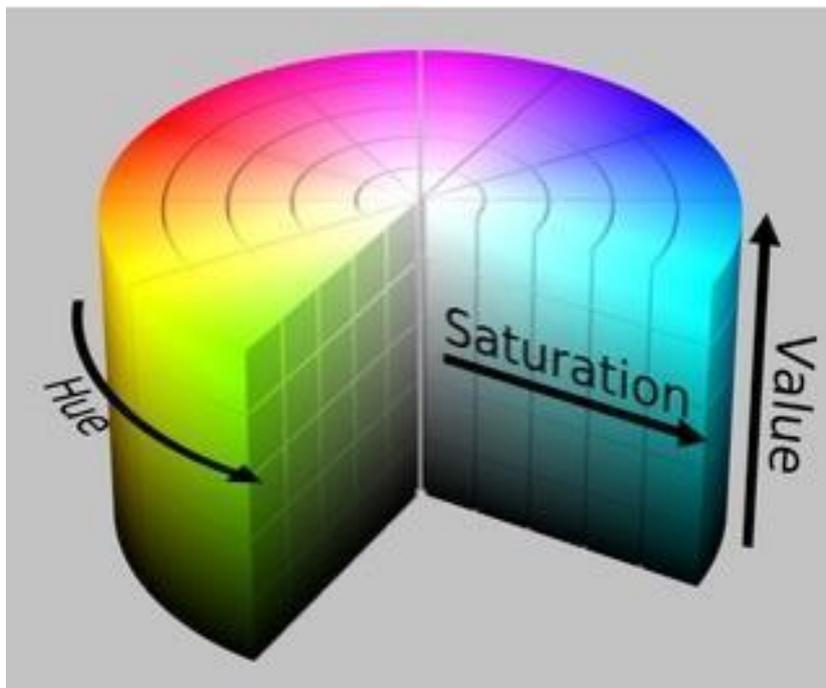


Figure II.7: Représentation de l'espace HSV.

II.2.3. Notion d'histogramme :

L'histogramme d'une image mesure la distribution des niveaux de gris dans l'image. Pour un niveau de gris x , l'histogramme permet de connaître la probabilité de tomber sur un pixel de valeur x en tirant un pixel au hasard dans l'image.

Concrètement, l'histogramme d'une image à valeurs entières est construit de la manière suivante : pour chaque niveau de gris x , on compte le nombre de pixels ayant la valeur x [18].

Par exemple, soit l'image de 5 pixels par 5 pixels de côté avec des valeurs comprises entre 0 et 4 :

0	1	2	2	3
0	1	2	2	3
0	1	2	2	4
0	1	2	2	4
0	1	2	2	4

Tableau II.1: matrice 5× 5.

Son histogramme est une fonction qui, à chaque valeur de niveau de gris compris entre 0 et 4, associe le nombre de pixels ayant cette valeur :

Valeur de niveau de gris	0	1	2	3	4
Nombre de pixels	5	5	10	2	3

Tableau II.2: Valeur de niveau de gris / nombre de pixels.

Ou bien, sous forme de diagramme bâton :

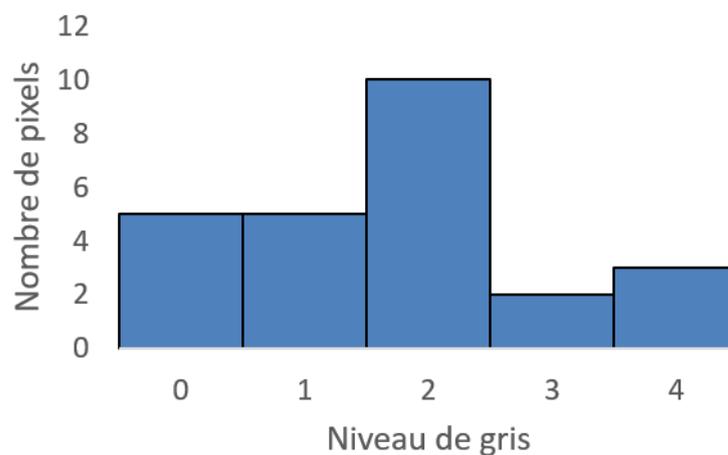


Figure II.8: Valeur de niveau de gris / nombre de pixels.

II.2.4. Profondeur d'image :

La profondeur d'image ou la profondeur de couleur est le nombre de bits utilisés pour représenter une couleur d'un pixel. La profondeur d'image détermine la plage de couleurs qu'une image peut avoir. Par exemple, si nous avons une image avec une profondeur de 4 bits, alors la valeur de pixel va de 0 à 15 (qui est le plus grand nombre que nous pouvons stocker en utilisant 4 bits - $2^4 - 1 = 15$). Alors que si nous utilisons 8 bits, alors la valeur variera de 0 à 255, fournissant un plus fin spectre de couleurs. Une autre façon de penser à la profondeur de l'image est que le nombre de bits détermine également le nombre de couleurs, qui peut être utilisé dans une image. Pour exemple, 1 bit implique deux couleurs, 2 bits - quatre couleurs, et 8 bits - 256 couleurs [15].

II.2.5. Formation d'une image à partir de canaux :

Comme nous le savons maintenant, une image RVB est représentée par trois plans, un pour chacune des composantes rouge, verte et bleue, respectivement. Nous pouvons conceptualiser une image composée de trois matrices indépendantes de largeur W et de hauteur H , une pour chacun des composants du RVB (*Figure II.9*) [16].

Nous pouvons combiner ces trois matrices pour obtenir un tableau multidimensionnel avec forme $W \times H \times D$ où D est la profondeur ou le nombre de canaux (pour la couleur RVB espace, $D=3$) :

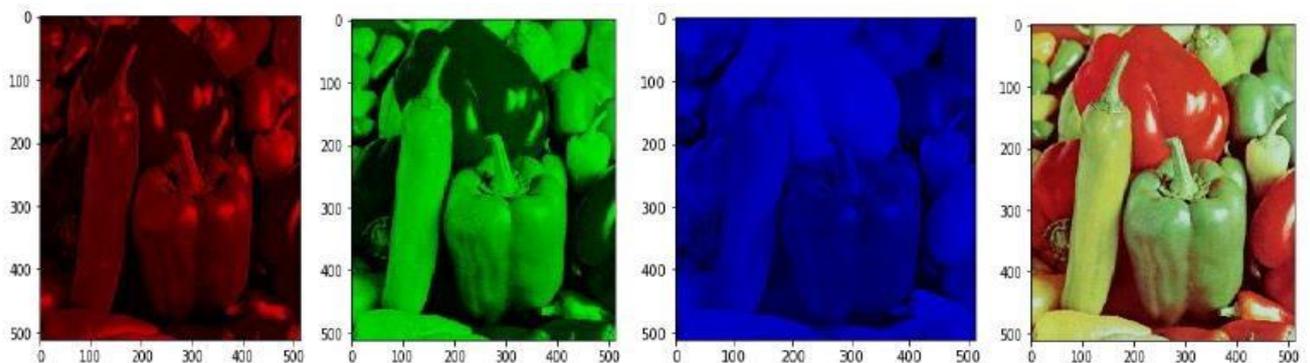


Figure II.9: Formation d'une image à partir de 3 canaux.

II.2.6. Le système de coordonnées d'image :

Comme indiqué à la figure plus haut dans ce chapitre, une image est représentée sous forme de grille de pixels. Pour rendre ce point plus clair, imaginer notre grille comme un morceau de papier graphique. En utilisant ce papier graphique, le point d'origine (0,0) correspond au coin supérieur gauche de l'image. Comme nous nous déplaçons vers le bas et à droite, les valeurs x et y augmentent. La figure fournit une représentation visuelle de cette représentation « graphique ».

Ici, nous avons la lettre « I » sur un morceau de notre papier graphique. Nous voyons qu'il s'agit d'une grille de 8 x 8 avec un total de 64 pixels [16].

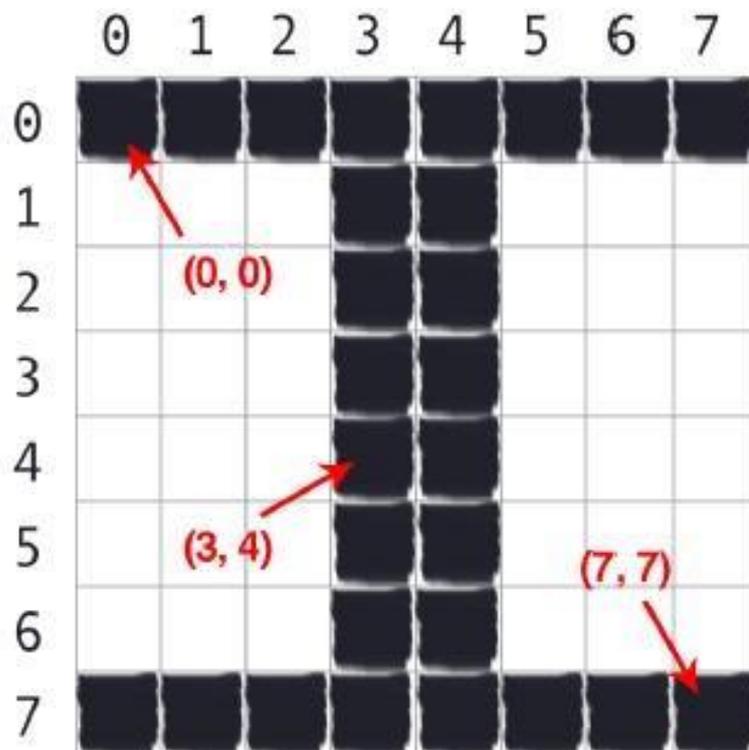


Figure II.10: Le système de coordonnées d'image.

II.3. Compréhension des filtres d'images :

L'amélioration d'une image en appliquant une fonction sur les valeurs de pixels est appelée filtrage. Le processus de filtrage se concentre sur les valeurs du voisinage d'un pixel et utilise certains pour modifier la valeur du pixel central. Ceci est fait en convoquant l'image matrice avec un noyau. Par conséquent, pour différents filtres, vous pouvez créer différents types de noyaux. En convoquant la matrice d'images avec le noyau, vous prenez essentiellement une moyenne pondérée des valeurs voisines. Cette méthode peut être utilisée pour réduire le bruit dans une image, créer des effets, etc. Le filtrage peut également être utilisé pour réduire le bruit dans une image car il prend une moyenne pondérée et en faisant la moyenne du bruit dans une pixel particulier réduit également le bruit. Les types de filtrage utilisés sont les suivants :

- Le flou gaussien (Gaussien blur).
- Image thresholding (Seuillage des images).

II.3.1. Flou gaussien :

II.3.1.1. Description :

En traitement d'image , un flou gaussien (également connu sous le nom de lissage gaussien) est le résultat du floutage d'une image par une fonction gaussienne (du nom du mathématicien et scientifique Carl Friedrich Gauss). C'est un effet largement utilisé dans les logiciels graphiques, généralement pour réduire le bruit de l'image et réduire les détails. L'effet visuel de cette technique de flou est un flou lisse ressemblant à celui de la visualisation de l' image à travers un écran translucide, distinctement différent de l'effet bokeh (flou) produit par un objectif flou ou l'ombre d'un objet sous un éclairage habituel.

Le lissage gaussien est également utilisé comme étape de prétraitement dans les algorithmes de vision par ordinateur afin d'améliorer les structures d'image à différentes échelles [19] .



Figure 1.11: L'effet de flou gaussien.

II. 3.1.2. Formulation :

Mathématiquement, appliquer un flou gaussien à une image revient à convoluer l'image avec une fonction gaussienne . Ceci est également connu sous le nom de transformée de Weierstrass bidimensionnelle. En revanche, la convolution par un cercle (c'est-à-dire un flou de boîte circulaire) reproduirait plus précisément l'effet bokeh .

La transformée de Fourier d'une gaussienne étant une autre gaussienne, l'application d'un flou gaussien a pour effet de réduire les composantes haute fréquence de l'image ; un flou gaussien est donc un filtre passe-bas .

Le flou gaussien est un type de filtre de flou d'image qui utilise une fonction gaussienne (qui exprime également la distribution normale dans les statistiques) pour calculer la transformation à appliquer à chaque pixel de l'image. La formule d'une fonction gaussienne à une dimension est :

(1)

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

En deux dimensions, c'est le produit de deux telles fonctions gaussiennes, une dans chaque dimension :

(2)

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

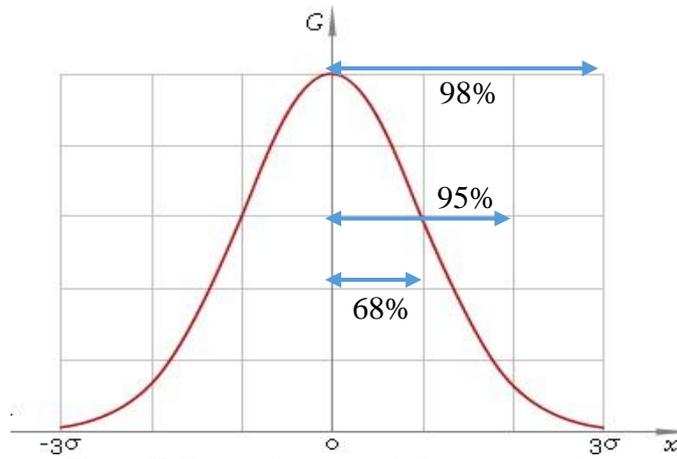


Figure II.12: Représentation de fonction gaussienne.

Où x est la distance de l'origine sur l'axe horizontal, y est la distance à l'origine dans l'axe vertical, et σ est l'écart - type de la distribution gaussienne. Lorsqu'elle est appliquée en deux dimensions, cette formule produit une surface dont les contours sont des cercles concentriques avec une distribution gaussienne à partir du point central.

Les valeurs de cette distribution sont utilisées pour construire une matrice de convolution qui est appliquée à l'image originale. Ce processus de convolution est illustré visuellement sur la figure La nouvelle valeur de chaque pixel est définie sur une moyenne pondérée du voisinage de ce pixel. La valeur du pixel d'origine reçoit le poids le plus lourd (ayant la valeur gaussienne la plus élevée) et les pixels voisins reçoivent des poids plus petits à mesure que leur distance au pixel d'origine augmente. Il en résulte un flou qui préserve mieux les limites et les contours que d'autres filtres de flou plus uniformes [20].



Figure II.13: Réduction de bruit avec le flou gaussien.

II.3.2. Seuillage des images :

II.3.2.1. Description :

Dans de nombreuses applications de vision, il est utile de pouvoir séparer les régions de l'image correspondant aux objets qui nous intéressent, des régions de l'image qui correspondent au fond. Le seuillage fournit souvent un moyen simple et pratique d'effectuer cette segmentation sur la base des différentes intensités ou couleurs dans les régions de premier plan et d'arrière-plan d'une image [21].

De plus, il est souvent utile de pouvoir voir quelles zones d'une image sont constituées de pixels dont les valeurs se situent dans une plage spécifiée, ou une *bande* d'intensités (ou de couleurs). Le seuillage peut également être utilisé pour cela.

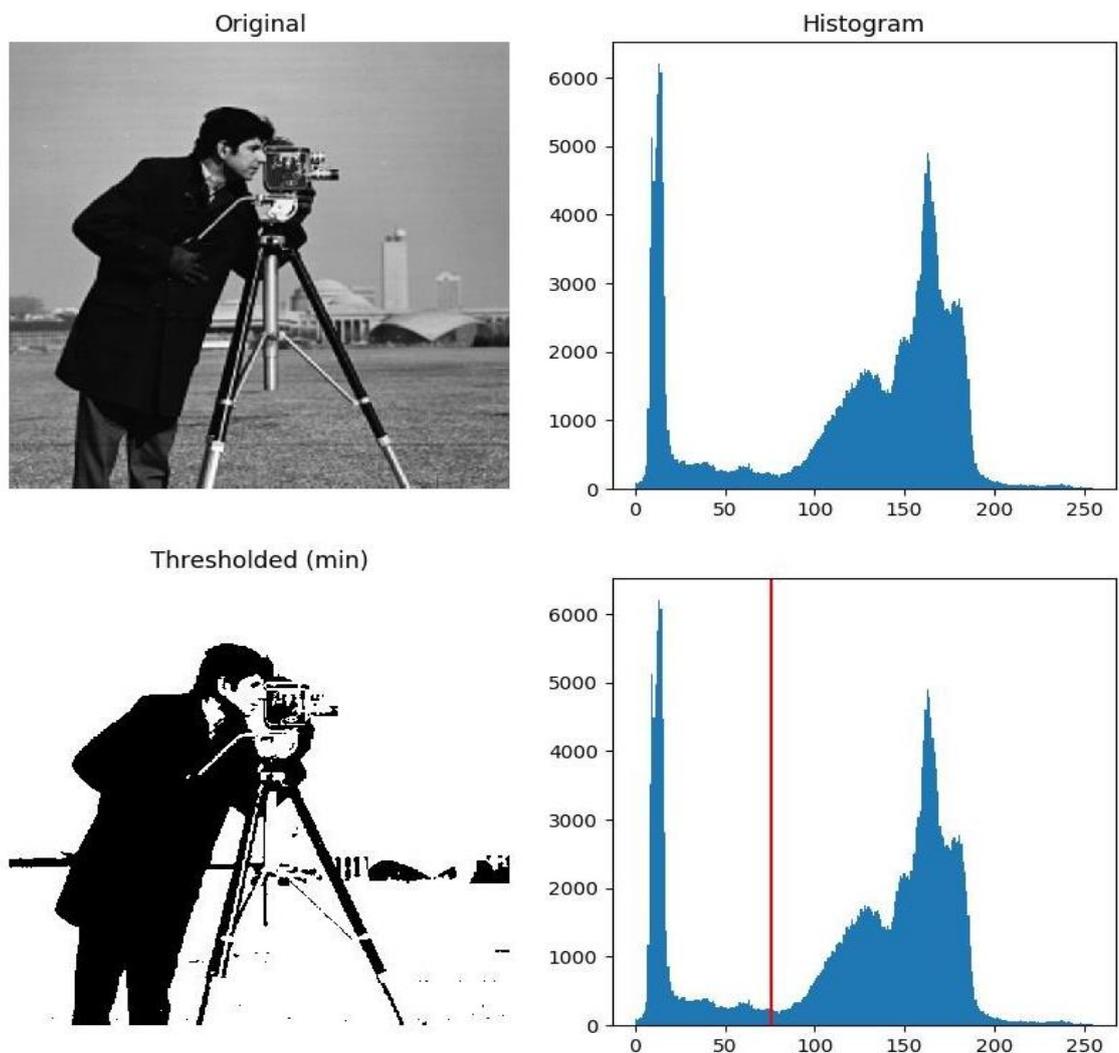


Figure II.14: Effet de seuillage au l'histogramme de l'image.

II.3.2.2. Fonctionnement :

L'entrée d'une opération de seuillage est généralement une image en niveaux de gris ou en couleur . Dans la mise en œuvre la plus simple, la sortie est une image binaire représentant la segmentation. Les pixels noirs correspondent à l'arrière-plan et les pixels blancs correspondent au premier plan. Dans des implémentations simples, la segmentation est déterminée par un seul paramètre appelé *seuil d'intensité*. En un seul passage, chaque pixel de l'image est comparé à ce seuil. Si l'intensité du pixel est supérieure au seuil, le pixel est défini sur, disons, blanc dans la sortie. S'il est inférieur au seuil, il est réglé sur noir.

Dans des implémentations plus sophistiquées, plusieurs seuils peuvent être spécifiés, de sorte qu'une *bande* de valeurs d'intensité puisse être définie sur blanc tandis que tout le reste est défini sur noir. Pour les images couleur ou multi spectrales , il peut être possible de définir des seuils différents pour chaque canal de couleur, et ainsi de sélectionner uniquement ces pixels dans un cuboïde spécifié dans l'espace RVB . Une autre variante courante consiste à mettre en noir tous les pixels correspondant à l'arrière-plan, mais à laisser les pixels de premier plan à leur couleur/intensité d'origine (au lieu de les forcer au blanc), afin que cette information ne soit pas perdue [21].

II.3.2.3. Formulation :

Soit une image f de dimension $N \times M$, $f(i, j)$ étant la valeur d'intensité d'un pixel de la ligne i et la colonne j , le **seuillage global** consiste à fixer un seuil s , à partir duquel tout pixel ayant une intensité supérieure ou égale au seuil s se voit attribuer la valeur 255 et le reste des pixels sera à 0 [22]. L'image binaire est obtenue par la formule suivante :

(3)

$$g(i, j) = \begin{cases} 255 & \text{si } f(i, j) \geq s \\ 0 & \text{sinon} \end{cases}$$

II.3.2.4. Directives d'utilisation :

Toutes les images ne peuvent pas être soigneusement segmentées en premier plan et en arrière-plan à l'aide d'un simple seuillage. Il est possible de déterminer si une image peut être correctement segmentée de cette manière en regardant un histogramme d'intensité de l'image. Nous ne considérerons ici qu'un histogramme en niveaux de gris, mais l'extension à la couleur est triviale.

S'il est possible de séparer le premier plan d'une image sur la base de l'intensité des pixels, alors l'intensité des pixels dans les objets de premier plan doit être nettement différente de l'intensité des pixels dans l'arrière-plan. Dans ce cas, nous nous attendons à voir un pic distinct dans l'histogramme correspondant aux objets de premier plan, de sorte que des seuils peuvent être choisis pour isoler ce pic en conséquence. Si un tel pic n'existe pas, il est peu probable qu'un simple seuillage produise une bonne segmentation. Dans ce cas, le seuillage adaptatif peut-être une meilleure réponse [21].

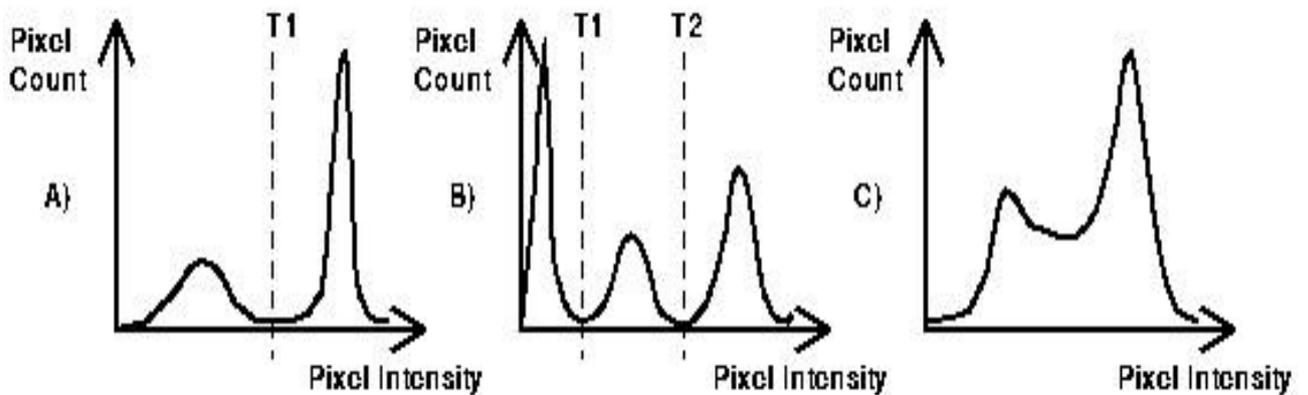


Figure II.15: Quelques histogrammes typiques ainsi que des choix appropriés de seuil.

L'histogramme **A** montre une distribution d'intensité bimodale classique. Cette image peut être segmentée avec succès à l'aide d'un seul seuil $T1$.

L'histogramme **B** est un peu plus compliqué. Ici, nous supposons que le pic central représente les objets qui nous intéressent et donc la segmentation de seuil nécessite deux seuils : $T1$ et $T2$.

En **C**, les deux pics d'une distribution bimodale ont couru ensemble et il n'est donc presque certainement pas possible de segmenter avec succès cette image en utilisant un seul seuil global.

L'histogramme pour l'image :

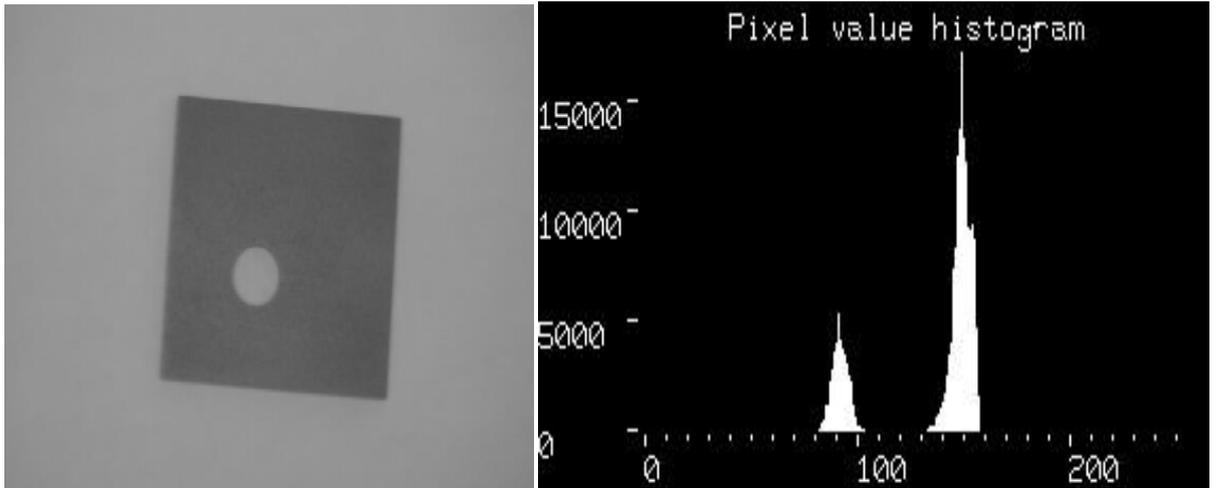


Figure II.16: Histogramme d'image au niveau de gris.

Cela montre une belle distribution bimodale --- le pic le plus bas représente l'objet et le plus haut représente l'arrière-plan. L'image peut être segmentée à l'aide d'un seuil unique à une valeur d'intensité de pixel de 120. Le résultat est démontré dans figure II.17 :

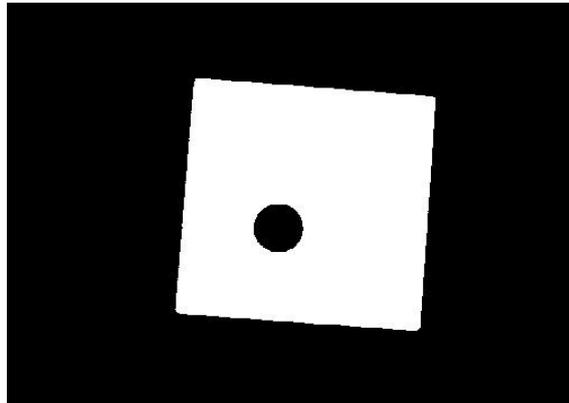


Figure II.17: Image binaire.

L'histogramme pour l'image :

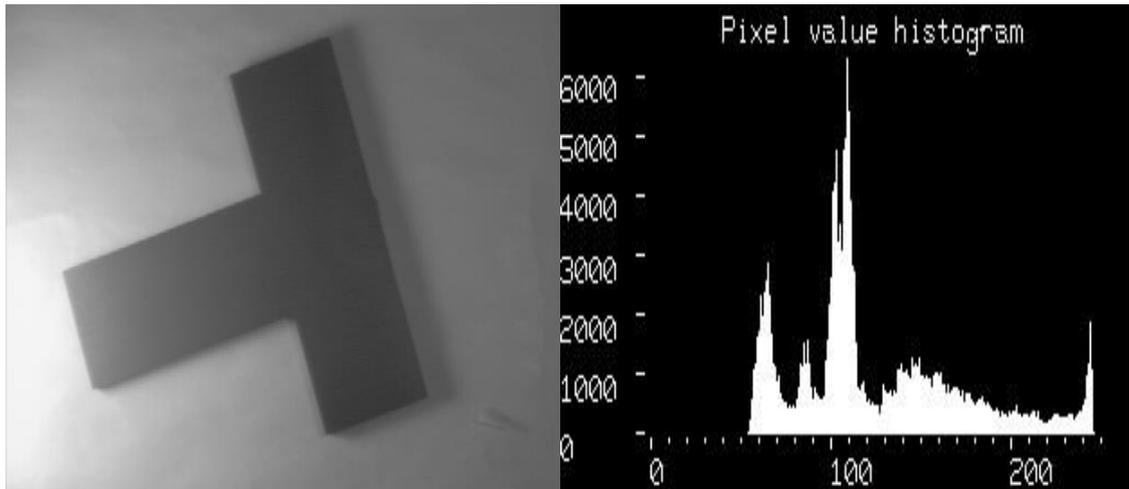


Figure II.18: Histogramme d'image à niveau de gris.

En raison du gradient d'éclairage sévère à travers la scène, les pics correspondant au premier plan et à l'arrière-plan se sont rapprochés et un seuillage simple ne donne donc pas de bons résultats.

Les images suivantes montrent les mauvaises segmentations résultantes pour des valeurs de seuil uniques de 80 et 120 respectivement.



Figure II.19: Résultat de binarisation.

Il existe plusieurs méthodes de seuillage d'images plus complexes et plus performantes que le seuillage simple, et parmi les méthodes les plus utilisées est la méthode de OTSU.

II.3.4. Méthode de seuillage d'Otsu :

Le principe est le même que précédemment, mais ce qui change ici c'est que cette méthode est capable de déterminer elle-même le seuil à attribuer à l'image traitée. Cette méthode peut être utilisée sur une image ayant un histogramme multimodal, c'est-à-dire supérieur à deux classes [23]. Elle nécessite donc le calcul préalable de l'histogramme de l'image indiqué dans la formule (4) suivante :

$$Hist(k) = \sum_{i=1}^n \sum_{j=1}^m (image(i,j) == k) \quad (4)$$

La probabilité de chaque niveau de gris est indiquée dans la formule 5 :

$$P(k) = (Hist(k)) / (\text{Nombre total de pixels dans l'image}) \quad (5)$$

L'image à binariser dans cette méthode ne contient que deux classes, (Les objets et l'arrière-plan). Donc trouve un seuil optimal T qui sépare ces deux classes afin que la variance intra-classe soit minimale et que la variance inter-classe soit maximale de l'histogramme de l'image.

Dans la méthode d'Otsu, le seuil qui minimise la variance intra-classe est recherché à partir de tous les seuillages possibles : l'équation 6 représente comment calculer la variance intra classe.

$$\sigma_w^2 = w_1 \times (t) \times \sigma_1^2(t) + w_2(t) \times \sigma_2^2(t) \quad (6)$$

Les poids w_i représentent la probabilité d'être dans la i^{eme} classe, chacune étant séparée par un seuil t finalement, les σ_i^2 sont les variances de ces classes.

Otsu montre que minimiser la variance intra-classe revient à maximiser la variance inter classe comment l'indique la formule 7 :

$$\sigma_y^2 = \sigma^2 - \sigma_w^2 \quad (7)$$

Qui est exprimée en termes des probabilités de classe w_i et des moyennes de classes $moy(t)$ qui à leur tour peuvent être mises à jour itérativement. Cette idée conduit à un algorithme efficace.

Les formules 8 et 9 représentent comment calculer la probabilité de classe 1 et 2.

Les formules 10 et 11 représentent comment calculer variance de chaque classe.

$$w_1(t) = \sum_{k=1}^t p(k) \quad (8)$$

$$w_2(t) = \sum_{k=t+1}^{256} p(k) \quad (9)$$

$$\sigma_1^2 = \frac{\sum_{i=1}^t ((N1(i) - moy1(t))^2)}{w_1} \times p(i) \quad (10)$$

$$\sigma_2^2 = \frac{\sum_{i=t+1}^{256} ((N2(i) - moy2(t))^2)}{w_2} \times p(i) \quad (11)$$

- t est un seuil choisi.
- $N1$ est un vecteur de 0 à $t-1$.
- $N2$ est un vecteur de t à 255.
- $Moy1$ représente la moyenne de la classe 1.
- $Moy2$ représente la moyenne de la classe 2.

Les formules 12 et 13 calculent la moyenne de chaque classe :

$$Moy1(t) = \frac{\sum_{i=1}^t N1(i) \times P(i)}{w_1} \quad (12)$$

$$Moy2(t) = \frac{\sum_{i=t+1}^{256} N2(i) \times P(i)}{w_2} \quad (13)$$

II.3.5. Représentation polygonale :

II.3.5.1. Description :

Une représentation polygonale d'une courbe numérique peut réduire considérablement la quantité de données à traiter tout en préservant des informations importantes sur la courbe. Il peut être appliqué dans l'analyse de forme, la classification de motifs, la compréhension d'image, la reconstruction 3D et les applications de conception assistée par ordinateur (CAO).

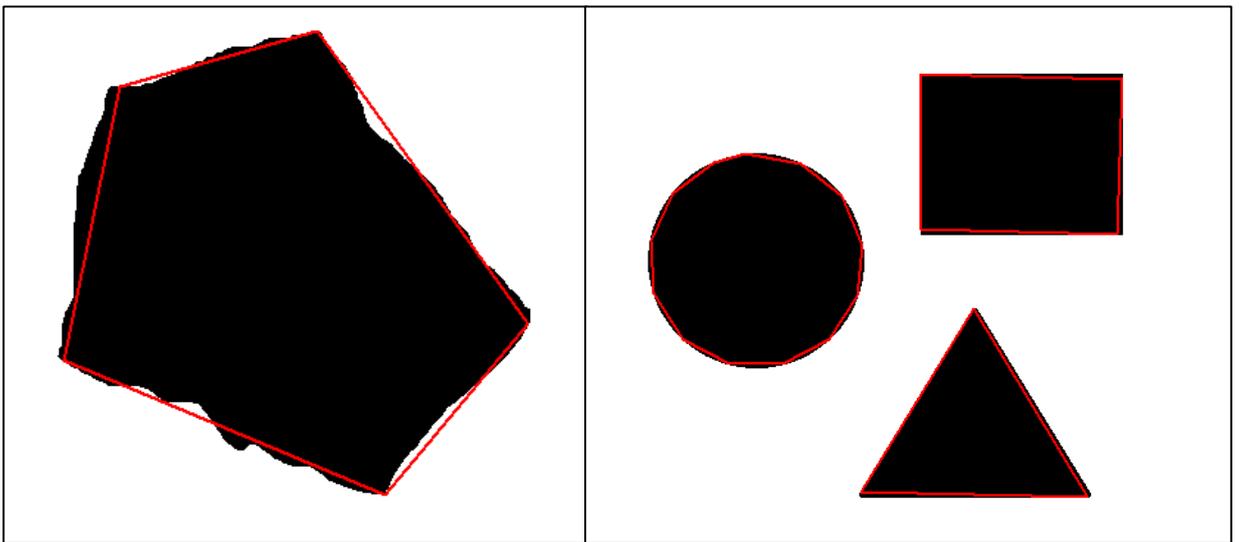


Figure II.20: *Forme approximée par 5 segments approximatés de droites.*

Figure II.21: *Cercle, carré et triangle approximatés par 13, 4 et 3 droites respectivement.*

Le but d'une approximation polygonale est de capturer l'essence de la forme dans une limite donnée en utilisant le moins possible nombre de segments. Cependant, les techniques d'approximation de complexes sont bien adaptées pour les tâches de traitement d'image. Parmi ceux-ci, l'un des plus puissants représente une limite par un polygone de périmètre minimal (MPP), comme défini dans la discussion suivante [24].

II.3.5.2. Mesures d'erreurs :

Une courbe d'approximation doit satisfaire certains critères d'erreurs, qui sont spécifiés de manière appropriée pour chaque application. En pratique, la plupart des mesures d'erreurs pratiques utilisées sont basées sur la distance entre les sommets de la courbe d'entrée et les segments linéaires d'approximation ou segment de droites.

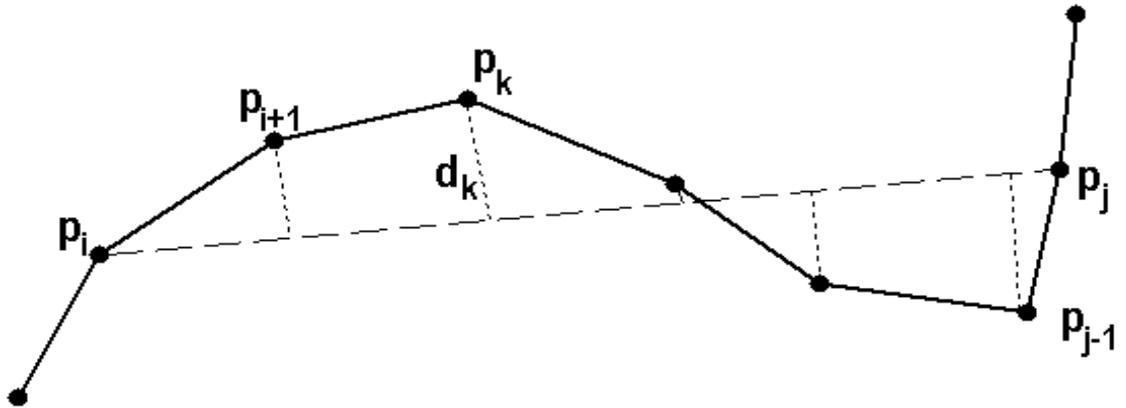


Figure II.22: Distance d_k du point p_k au segment de droite (p_i, p_j) .

La distance $d_k(i, j)$ du point $p_k=(x_k, y_k)$ au segment de droite (p_i, p_j) est défini par:

$$d(k, i, j) = \frac{|y_k - a_{i,j}x_k - b_{i,j}|}{\sqrt{1 + a_{i,j}^2}}$$

Où les coefficients $a_{i,j}$ and $b_{i,j}$ sont les paramètres du segment de droite (p_i, p_j) :

$$a_{i,j} = (y_j - y_i) / (x_j - x_i)$$

$$b_{i,j} = y_i - a_{i,j}x_i.$$

La somme cumulative des erreurs L_p pour le segment $\{p_i, p_j\}$ est définie par la somme des distances $d(k, i, j)$ pour tous les sommets dans le segment $\{p_i, p_j\}$ comme suit:

$$e_p(i, j) = \sum_{k=i+1}^{k=j-1} d^p(k, i, j)$$

II.4. Conclusion :

Le domaine du traitement de l'image est un domaine encore en expansion et qui pénètre tous les secteurs d'activités, tant les environnements industriels et professionnels que le monde des applications dédiées au grand public. Ce chapitre résume quelques techniques de traitement d'image utilisé dans ce projet, des notions et des brèves explications sur les algorithmes. Dans le prochain chapitre, nous expliquerons la partie pratique et l'implémentation de système, l'environnement de travail et les problèmes rencontrés.

Chapitre III

Partie pratique et implémentation du système

III.1. Introduction :

L'intérêt d'un système de vision industrielle est d'améliorer la productivité et la qualité de la production. Le contrôle par vision évite le travail fastidieux d'un contrôle manuel, et permet d'être plus rapide pour des tâches répétitives. Il s'agit de développer un système de vision rapide, fiable et ayant pour objectif le "zéro défaut".

À partir d'un cahier des charges, une étude de faisabilité va permettre de savoir si le système de vision souhaité est réalisable. Un système peut contrôler le produit fini, mais aussi les différentes étapes de fabrication. Il faut choisir l'endroit le plus judicieux pour placer son système de vision en tenant compte de l'environnement. L'étape majeure du système est l'acquisition de l'image.

En effet, il sera plus difficile d'extraire une information d'une image de mauvaise qualité. Voilà pourquoi le choix de la caméra, de l'objectif et de l'éclairage est primordial. Le logiciel de traitement et d'analyse d'images doit être simple et convivial pour l'opérateur. Les algorithmes de traitements doivent être rapides et robustes.

Ce chapitre explique les différents choix des outils hardware et les techniques appliqués pour réussir le système étudié.

III.2. Environnement de travail :

III.2.1 Hardware:

- Laptop i3 5eme génération + 8go ram.
- Imprimante 3D.
- Raspberry pi v4 + 2go ram.
- Arduino méga pour des tests.

III.2.2. Software:

- Windows 10 pro 64bits.
- Linux embarqué : Raspberry OS 64bits.
- Serveur SSH (puTTY + VNC viewer).
- Python 3.7.
- SolidWorks 2016.
- PyCharm IDE 2020.1.

III.3. Bibliothèques de python utilisé :

Python est un langage de programmation qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses.



Parmi ces bibliothèques utilisées dans notre réalisation sont comme suit :

III.3.1. OpenCV :

OpenCV est une immense bibliothèque open source pour la vision par ordinateur, l'apprentissage automatique et le traitement d'images. OpenCV prend en charge une grande variété de langages de programmation comme Python, C++, Java, etc. Il peut traiter des images et des vidéos pour identifier des objets, des visages, ou même l'écriture d'un humain.



III.3.2. NumPy :

NumPy est le paquet fondamental pour l'informatique scientifique en Python. C'est une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux masqués et des matrices), et un assortiment de routines pour des opérations rapides sur des tableaux, y compris mathématique, logique, manipulation de forme, tri, et plus encore.



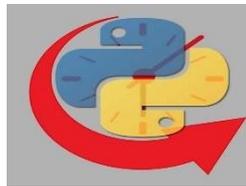
III.3.3. Adafruit

Est une bibliothèque dédiée pour le management des informations dans un environnement compliqué ciblé pour les applications de control composé de capteurs, moteurs, et plus encore.



III.3.4. Time

Le module Time python offre la possibilité de lire, représenter et réinitialiser les informations de temps de nombreuses façons. Le temps en python est représenté par les seconds.



III.3.5. Future :

Le module futur est un vrai module, a un objectif d'éviter de dérouter les outils existants qui analysent les instructions d'importation et s'attendent à trouver les modules qu'ils importent.



III.3.6. Picamera :

C'est un package qui fournit une interface python pure au module de camera Raspberry pi, qui donne une flexibilité de control aux paramètres de camera.



III.4. Principe du système développé :

Nous avons mis un système basé sur la vision artificielle. Le système développé repose plus présent sur l'organigramme donné par la figure III.1 :

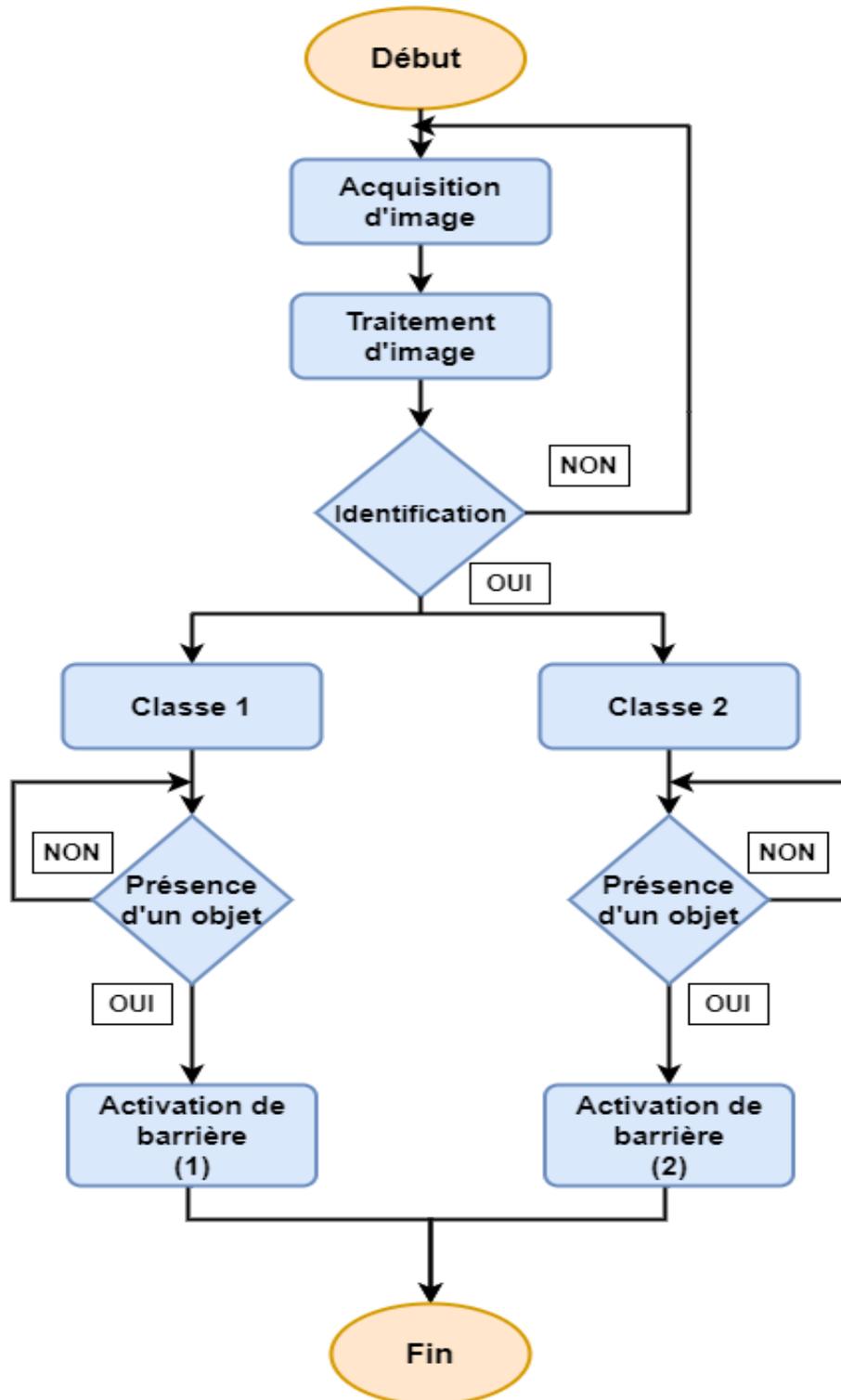


Figure III.1: Organigramme de système proposé.

III.4.1. Acquisition d'image :

La caméra vidéo est l'outil qui permet de capter les images en transformant la lumière qui est composé des ondes électromagnétiques perçues par la vision humaine avec une longueur d'onde comprise entre 380nm et 780nm, en message électrique.

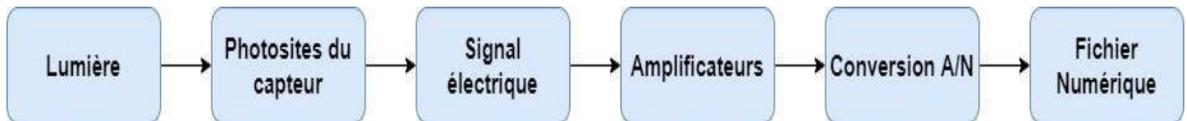


Figure III.2: Schéma de bloc d'acquisition.

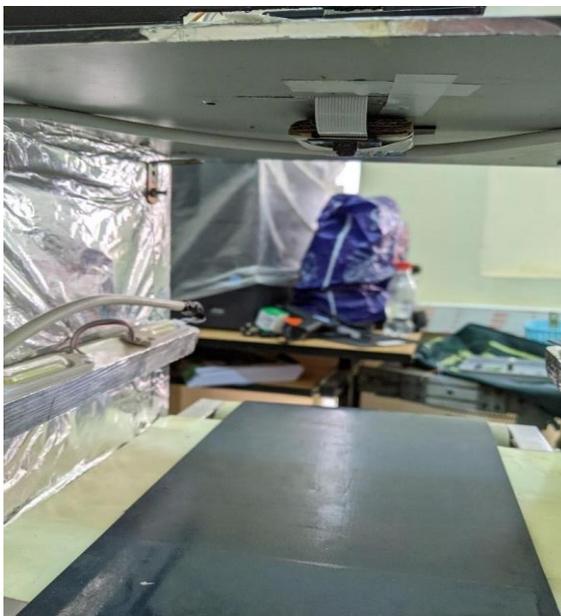


Figure III.3: Placement de camera dans le système.

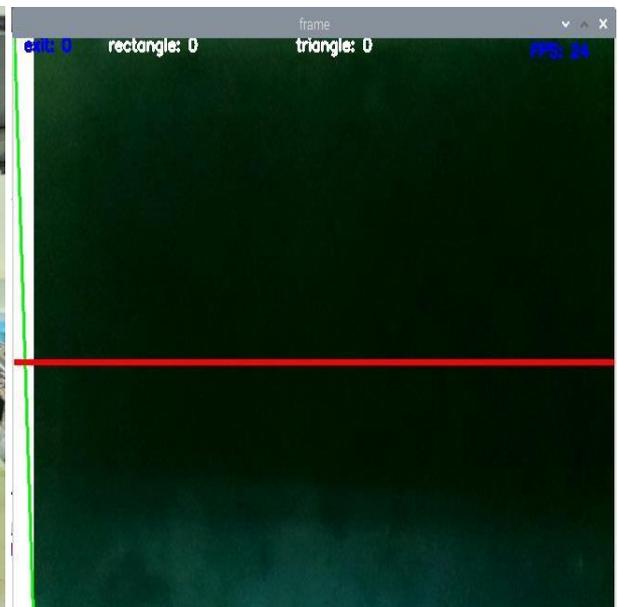
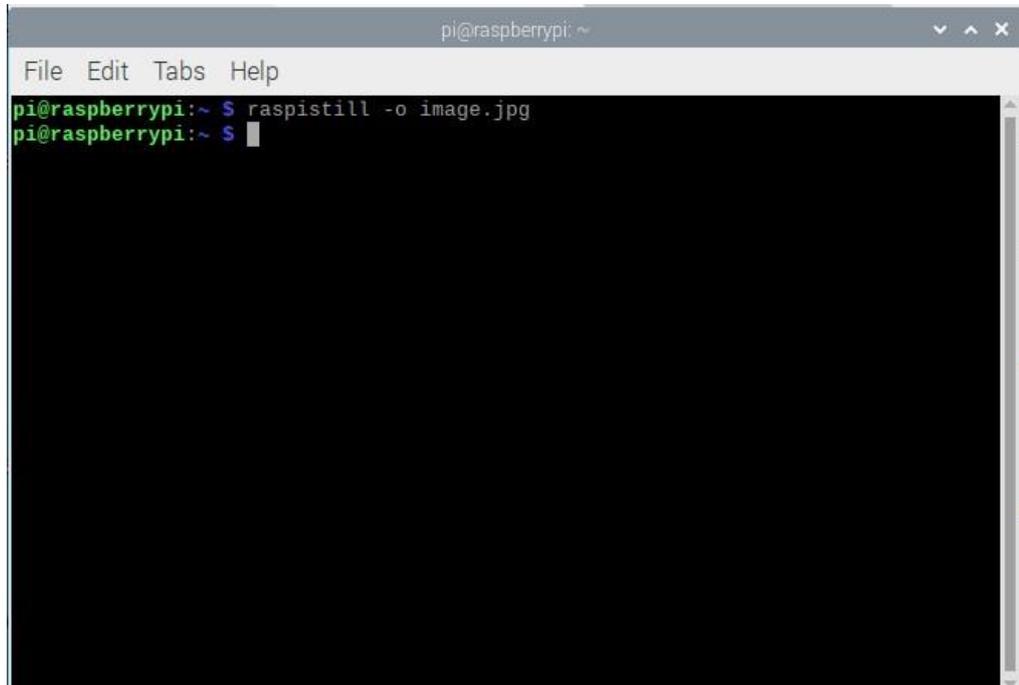


Figure III.4: Sortie de camera en temps réel.

Nous avons utilisé une caméra pi camera v1.3, placé 20cm au-dessus de convoyeur pour l'acquisition d'image.

Avec une ouverture de f/2.9, une distance focale de 3.6mm et une taille de pixel 1.4 x 1.4 μm elle donne une photo de 5MP avec une résolution maximale de (2592 x 1944) en espace RGB.

Un test sur le terminal (commande **raspistill -o image.jpg**) pour assurer le fonctionnement de capteur est nécessaire avant l'utilisation.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ raspistill -o image.jpg  
pi@raspberrypi:~ $
```

Figure III.5: commande de test sur le terminal de linux.



Figure III.6: L'image de sortie.

Dans notre cas, nous l'avons utilisé pour capturer une vidéo en temps réel avec une résolution de (640 x 480) et de cadence 60 FPS, ça veut dire, prendre un cadre toutes les 16ms.

La cadence est le nombre d'images que la caméra peut capturer par seconde. Selon le temps nécessaire pour capturer une image, le temps d'exposition, nous ne pouvons capturer qu'un nombre limité d'images dans un laps de temps spécifique.

Cela peut être exprimé dans l'équation suivante :

$$\text{Taux max d'encadrement en FPS} = \frac{1s}{\text{min temps d'exposition en s}}$$

Lors du lancement du programme, l'obturateur s'ouvre afin de laisser passer la lumière par l'objectif (lentille) qui va fournir une image nette du sujet à photographier au capteur numérique. La quantité de lumière entrante est déterminée par le temps durant lequel l'obturateur va rester ouvert (chaque 1ms) et par la taille de l'ouverture du diaphragme (f/2.9). A la fin de l'exposition, l'obturateur reprend sa position fermée initiale. Ce processus boucle jusqu'à l'arrêt de l'application.

Pour se concentrer sur l'objet, nous avons proposé une région d'intérêt en introduisant ses coordonnées prenant en avantage le système de coordonnées d'image présentée au chapitre II.

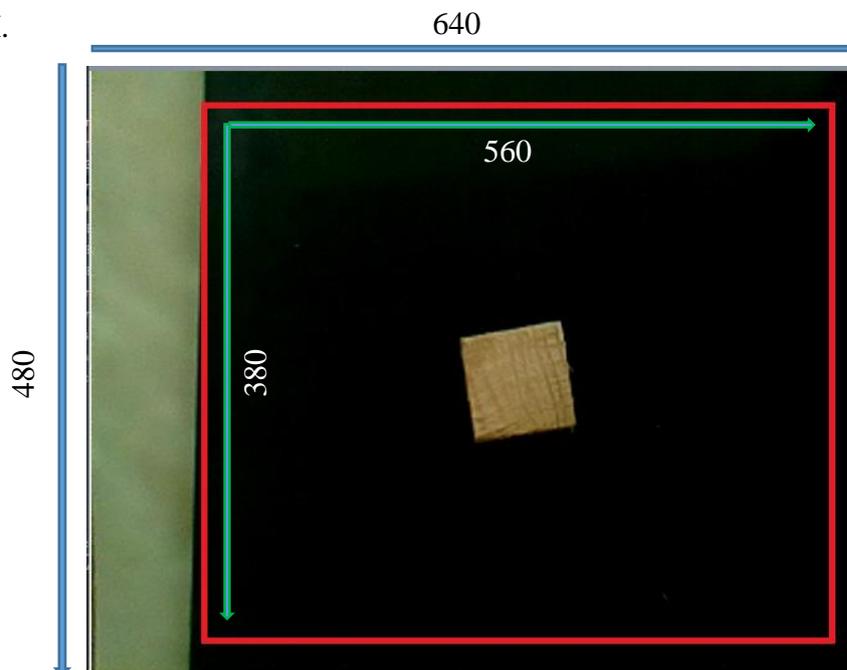


Figure III.7: L'image recardé.

Nouvel frame = original image [30 :410] sur axe des j et [40 :600] sur axe des i.

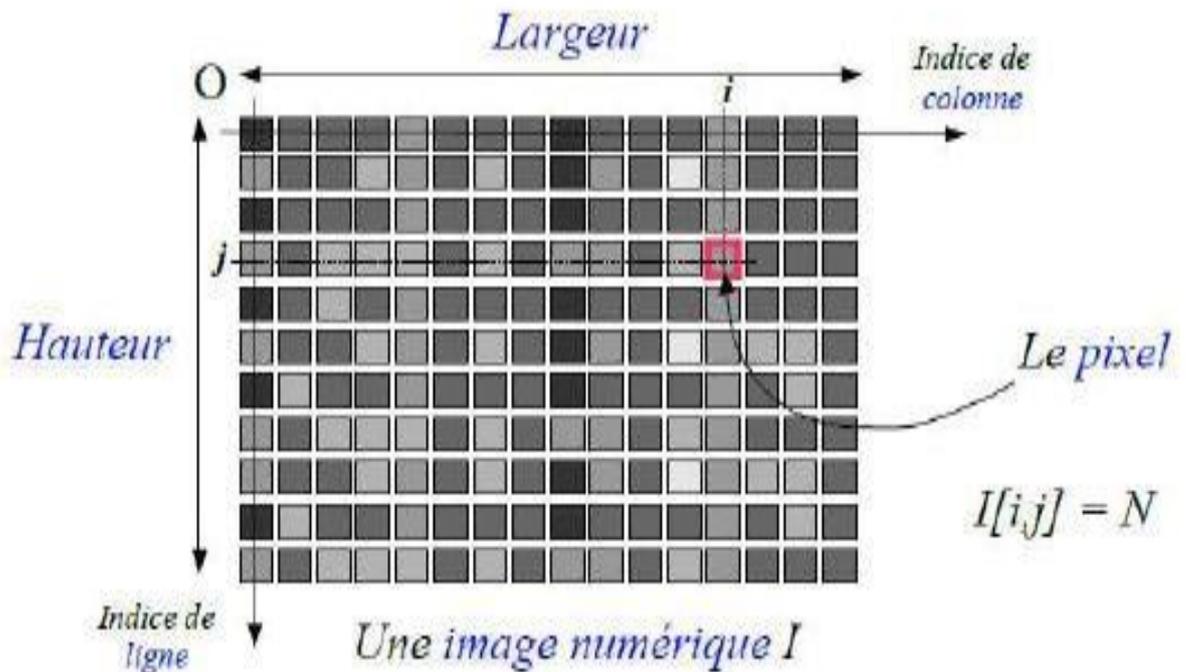


Figure III.8: Les dimension d'une image numérique.

L'image acquise par le capteur de camera et appelée image brute (image RAW), des prétraitements et diverses techniques doivent être appliqués pour obtenir une image utilisable (discuté dans le chapitre II). Pour notre application nous avons choisi un Raspberry pi version4 pour le traitement d'images (voir chapitre I).

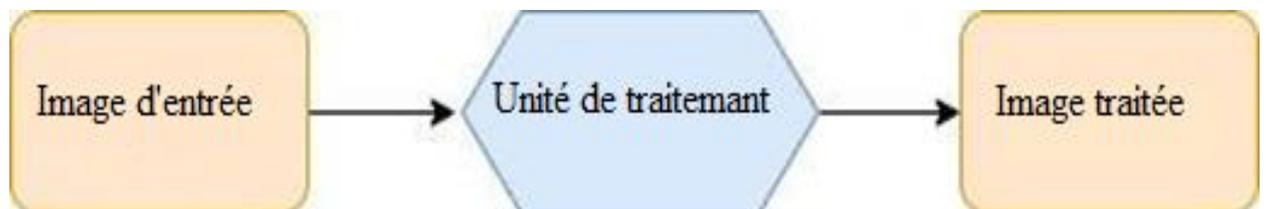


Figure III.9: Schéma bloc du traitement d'image.

III.4.2. Unité de traitement d'image :

Le Raspberry pi est un nano ordinateur a processeur, qui utilise un système linux dédié au applications embarqué surtout. Mais les applications embarquées fonction en RTOS (real time operating system – système temps réel) et linux est un non-RTOS.

Pour ça, on peut dire que aucun traitement de la caméra n’au lien avec le processeur (sous linux). Au lieu de cela, cela se fait sur le GPU du pi (VideoCore IV) qui exécute son propre système d’exploitation en temps réel (VCOS).

Le diagramme suivant montre que le système BCM2835 sur puce (SoC) est composé d’un processeur ARM cortex exécutant linux et d’un GPU videoCore IV exécutant VCOS. L’interface hôte VideoCore (VCHI) est un système de transmission de messages fourni pour permettre la communication entre ces deux composants. La Ram disponible est répartie entre les deux composants (128Mo est une répartition typique de la mémoire GPU lors de l’utilisation de la caméra) [7]. Cette mémoire typique n’était pas suffisante, par résultats changé à 512Mo.

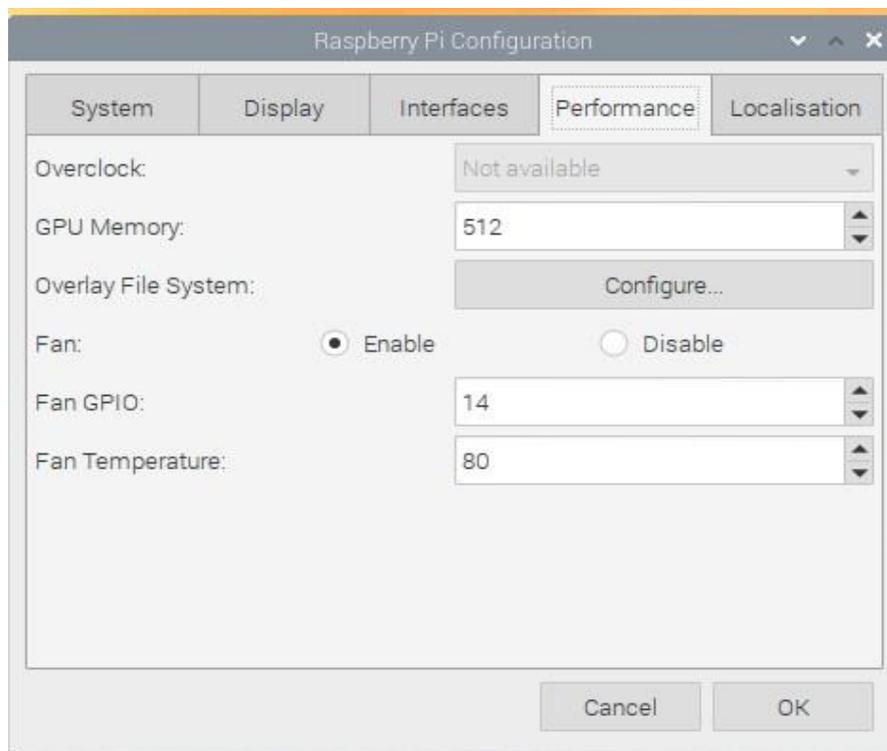


Figure III.10: Configuration de GPU dans les paramètres.

Enfin, le module caméra est présenté au-dessus du SoC. Il est connecté au SoC via une interface CSI-2(fournissant 2 Gbit/s de bande passante). Une activation de module camera dans les paramètres de système est nécessaire pour activer le port d'interface.

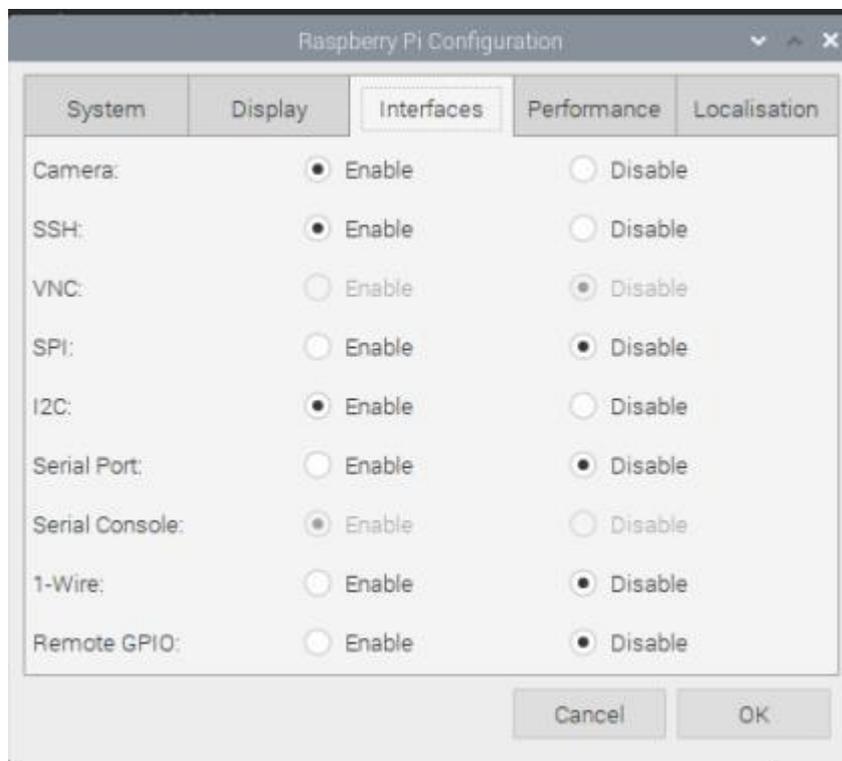


Figure III.11: Activation des interfaces nécessaires dans les paramètres.

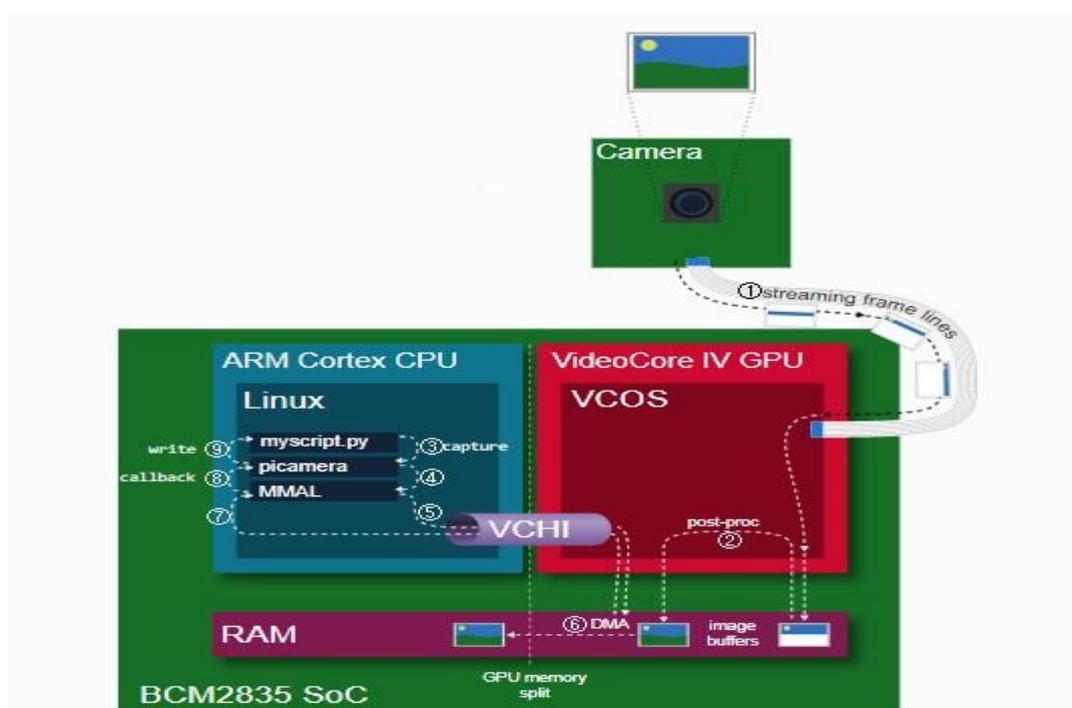


Figure III.12: Digramme d'architecture Raspberry.

Le scénario décrit est le suivant :

1. Le capteur de la caméra a été configuré et diffusé en continu des lignes de trame via l'interface vers le GPU.
2. Le GPU assemble des tampons de trame complets à partir de ces lignes effectue un post-traitement sur ces tampons.
3. Pendant ce temps, sur le processeur, **myscript.py** passe une **capture** appel à l'aide de picamera.
4. La bibliothèque picamera utilise à son tour l'API MMAL pour exécuter cette requête (en fait, il y a beaucoup d'appels MMAL ici, mais par souci de simplicité, nous représentons tout cela avec une seule flèche).
5. L'API MMAL envoie un message via VCHI demandant une capture de trame (encore une fois, en réalité, il y a beaucoup plus d'activité qu'un seul message).
6. En réponse, le GPU initie un transfert DMA de la prochaine trame complète de sa partie de RAM vers la partie de GPU.
7. Enfin, le GPU envoie un message via VCHI indiquant que le capture est terminée.
8. Cela provoque un thread MMAL pour déclencher un rappel dans la bibliothèque picamera, qui a son tour récupère la trame (en réalité, cela nécessite plus d'activité MMAL et VCHI).
9. Enfin, picamera appelle **write** l'objet de sortie fourni par **myscript.py** [7].

Il existe deux méthodes d'accès au Raspberry pi, soit la méthode habituelle avec un câble HDMI branché avec un écran (avec INPUT HDMI ENABLE) ou une méthode sans fils utilisent un serveur **SSH** entre l'ordinateur et notre unité de traitement (Raspberry).

SSH est un protocole permettant d'établir une communication chiffrée, donc sécurisée sur un réseau informatique entre une machine locale (notre ordinateur/client) et une machine distante (notre Raspberry/serveur). La sécurité du chiffrement peut être assurée par différentes méthodes, entre autres par un mot de passe ou par un système de clés publique [25].

En suivant les étapes suivantes, nous avons réussi à établir une connexion réussie entre le serveur et le client :

- Activation de service SSH dans les paramètres de Raspberry (voir figure X).
- Création d'un mot de passe pour la machine de serveur.
- Connecté les deux machines sur un même réseau Local LAN.
- Extraire l'adresse IP v4 de machine serveur via l'instruction **ifconfig** dans le terminal.
- Demander l'accès au serveur par l'ordinateur passant par 2 logiciels : PuTTY et VNC VIEWER.
- Entrer l'adresse IP de machine serveur dans PuTTY.

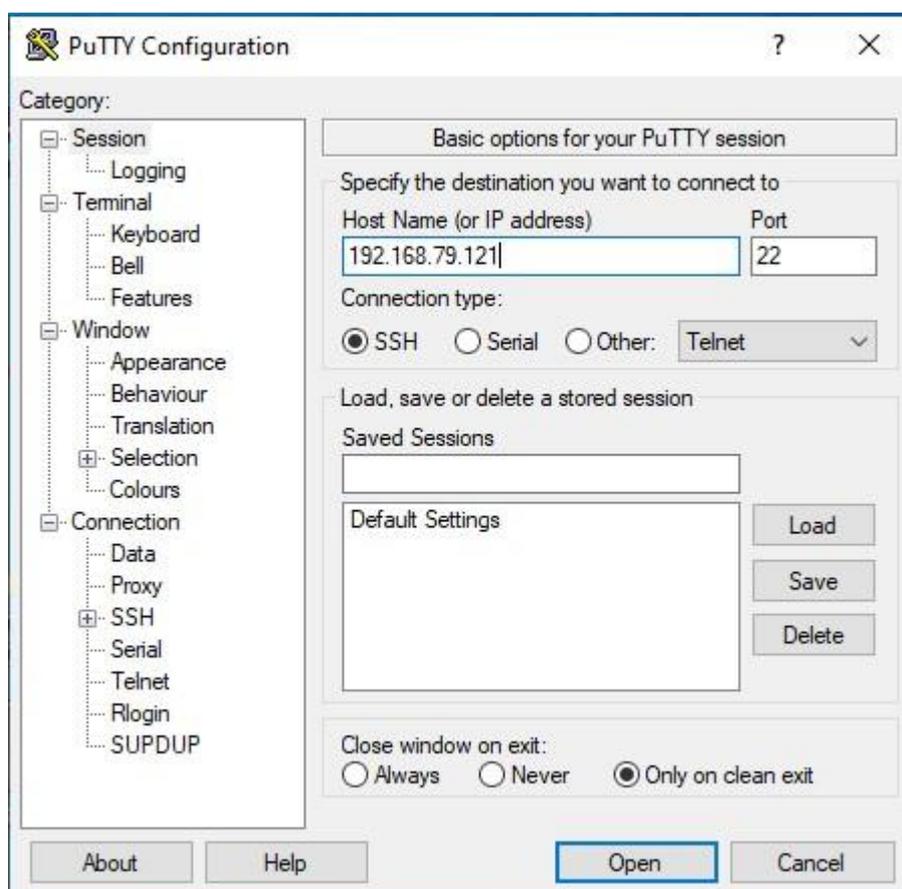


Figure III.13: Configuration d'une session SSH dans Putty.

- Entrer le mot passe du serveur crée précédemment :



Figure III.14: Saisir de mot de passe Raspberry.

- Après avoir établi un lien entre le serveur et le client, on désire voir ce qu'il y a à l'intérieur de serveur.
- Création d'une session tightVNC qui permet de voir le bureau d'une machine à distance et le contrôler avec la souris et clavier de notre ordinateur.

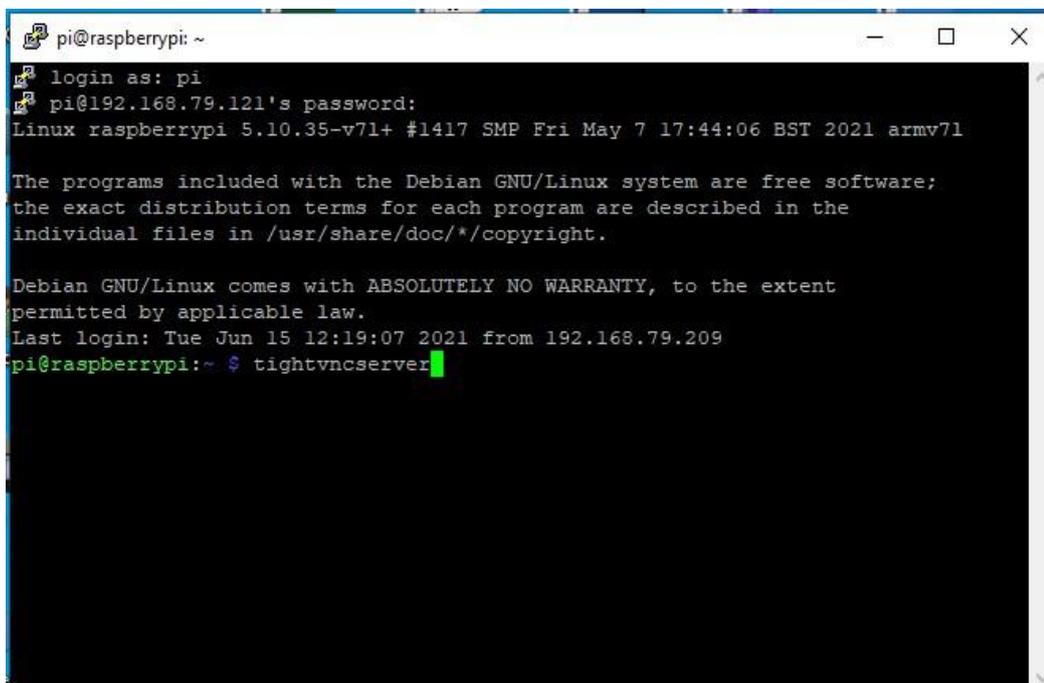


Figure III.15: Création d'une session VNC.

- Entre l'adresse IP v4 de Raspberry dans le logiciel VNC Viewer et le mot passe pour afficher le contenu de ce dernier.

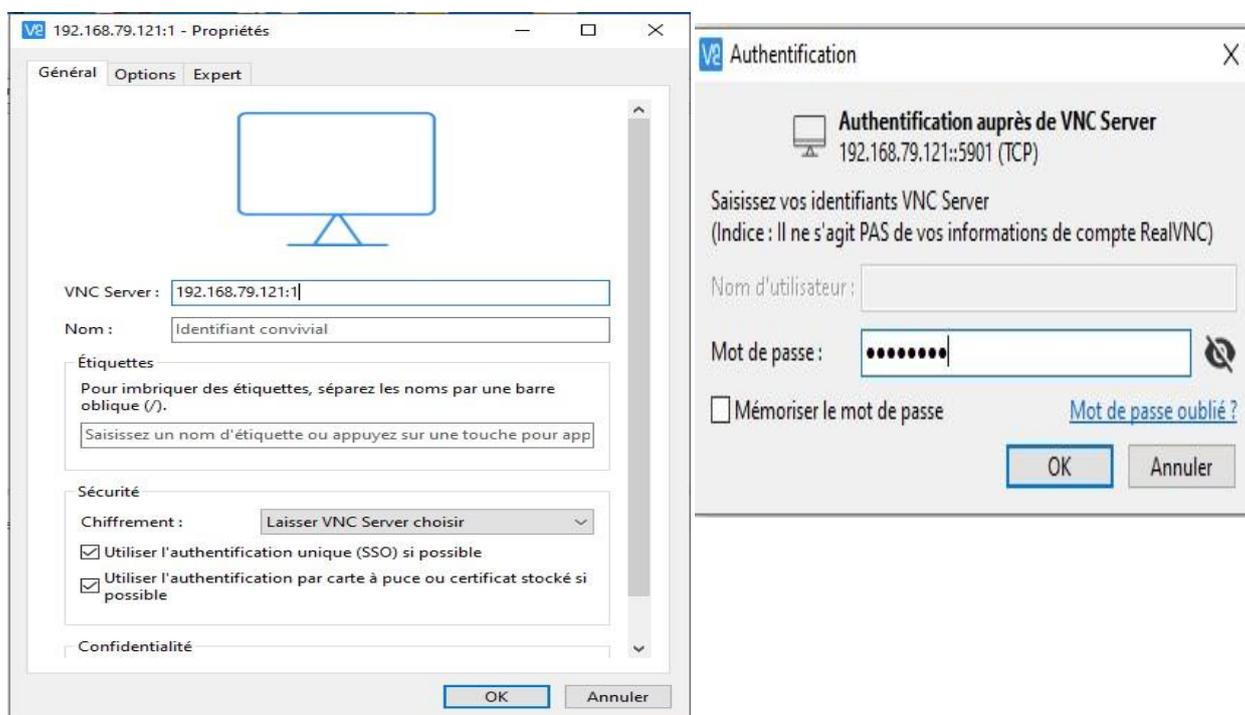


Figure III.16: Saisir d'adresse IP / mot de passe Raspberry dans VNCviewer.

- Obtenir une fenêtre Raspberry de système linux sur l'ordinateur Windows sans utiliser un câble HDMI.

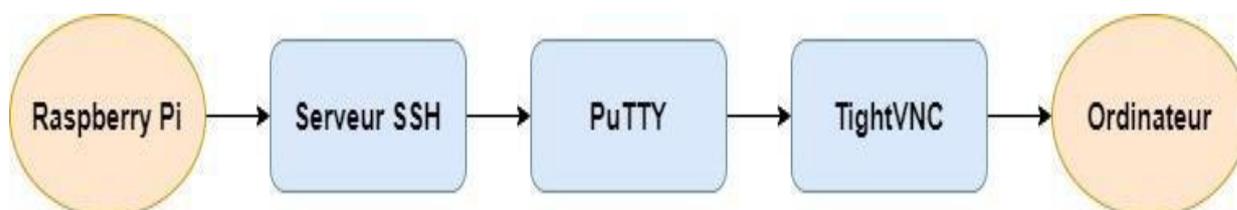


Figure III.17: Schéma block d'accès au bureau de Raspberry.

III.4.3. Segmentation et identification :



Figure III.18: Schéma bloc des étapes du traitement d'une image.

III.4.3.1. Prétraitement :

Les régions formant l'image sont caractérisées par leur intensité moyenne. Les fluctuations autour de cette intensité moyenne peuvent provenir soit du dispositif d'acquisition (caméra, amplificateurs, quantification...) soit de la scène elle-même (poussières, rayures...). Elles constituent des perturbations qui affectent la qualité de la segmentation et sont généralement désignées sous le terme de *bruit d'image*. L'échelle spatiale des fluctuations est relativement faible par rapport aux dimensions des régions. Le bruit est donc plutôt de type haute fréquence.

Pour le but d'une claire apparence d'objet, nous avons utilisé le filtre de flou gaussien. Ce dernier est un opérateur de lissage isotrope qui est utilisé pour réduire le bruit et détail à partir d'images.

Après avoir construit une matrice de convolution, elle est appliquée sur l'image d'entrée avec un noyau de 7×7 pixels, une modification sur les valeurs des pixels est appliquée avec la valeur calculée par la fonction gaussienne. Si on parle des fréquences, c'est un filtre passe bas au but de réduire les hautes fréquences. Et si on parle des amplitudes, c'est la réduction d'amplitude des variations d'intensité dans chaque région.

III.4.3.2. Segmentation :

Le seuillage est une méthode utilisée pour séparer un objet ou un objet cible de son image de fond en attribuant une valeur d'intensité T (seuil) pour chaque pixel tel que chaque pixel soit catégorisé comme un point de fond ou un point objet. C'est une opération complexe qui passe par des différentes étapes (voir chapitre II) pour séparer les objets.

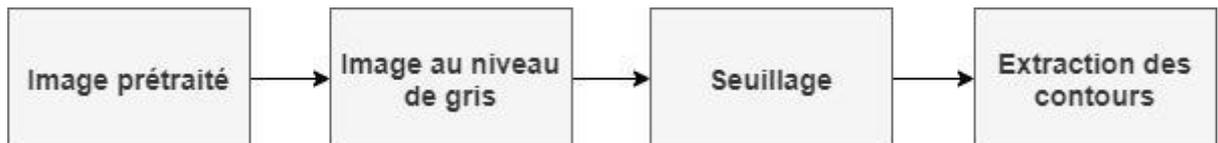


Figure III.19: Schéma bloc des étapes de segmentation.

Elle passe par deux étapes qui sont les suivantes :

- Transformé l'image prétraitée de l'espace RGB vers une image à niveau de gris en utilisant la fonction `cv.cvtColor` de opencv python.

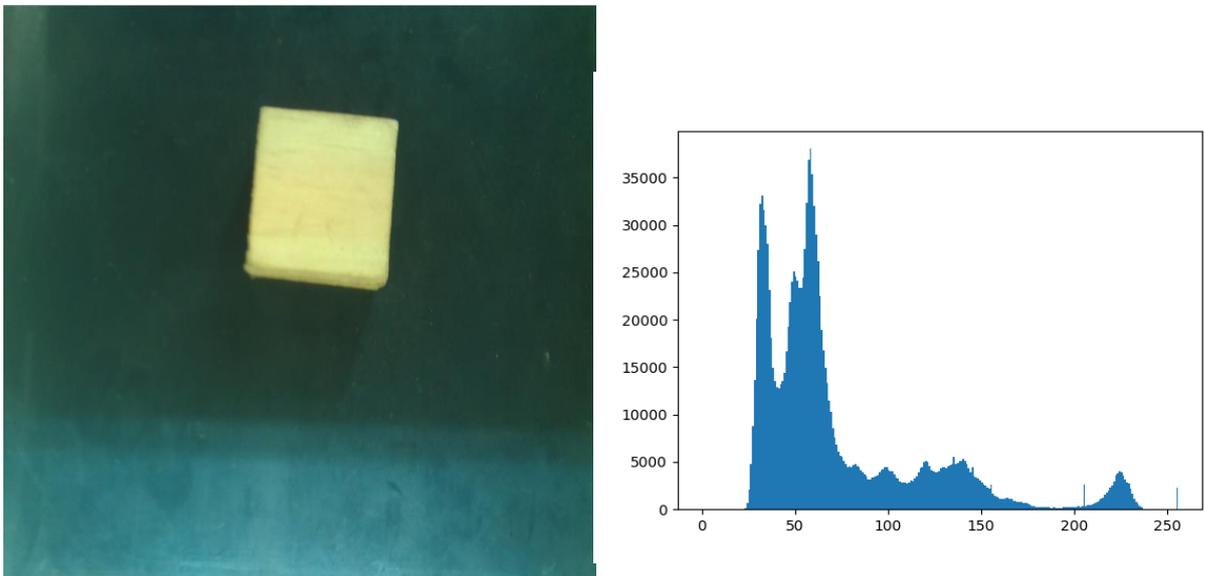


Figure III.20: Histogramme de l'image originale.

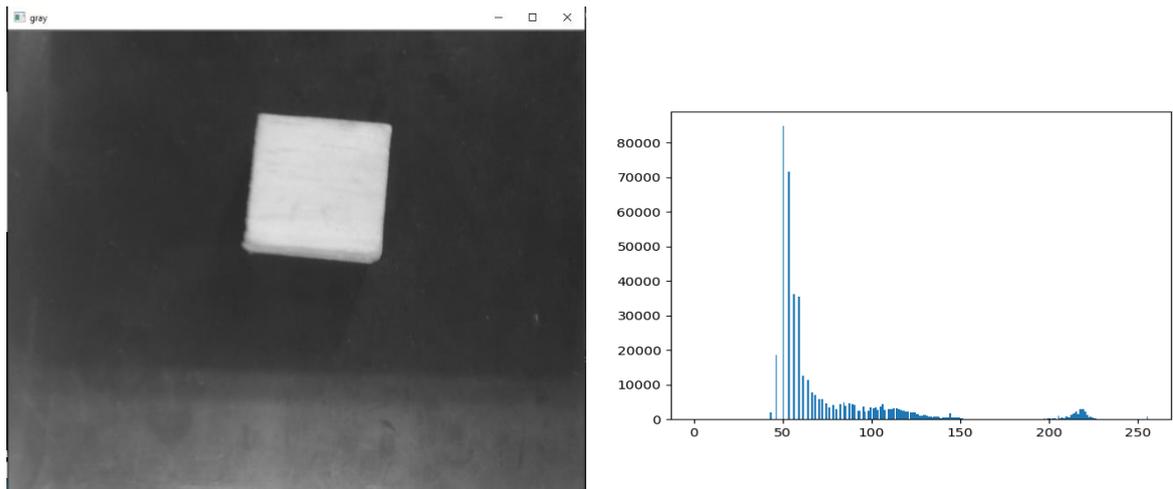


Figure III.21: Histogramme d'image au niveau de gris.

- Binarisation d'image avec un seuillage global (méthode d'Otsu) ; après avoir calculé un seuil optimal T (voir chapitre II) qui sépare deux classes (l'objet et l'arrière-plan) :

$$g(x, y) = \begin{cases} 255, & \text{si } f(x, y) > T \\ 0, & \text{sinon} \end{cases}$$

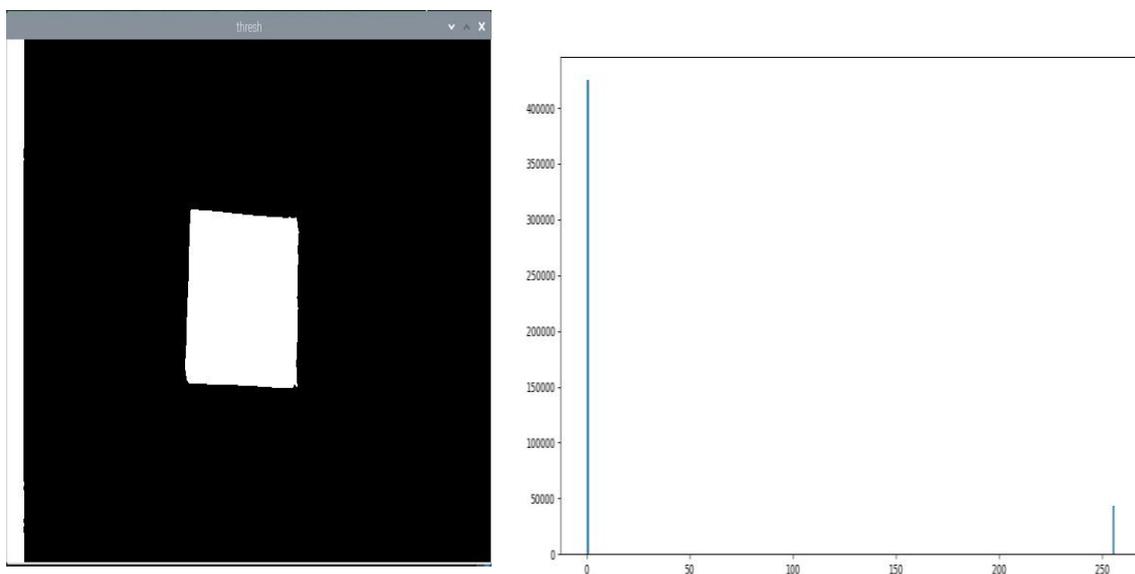


Figure III.22: Histogramme de l'image après l'étape de binarisation.

Comme nous l'avons expliqué plus tôt, la première classe qui a une valeur des pixels égale à zéro (0) représente l'arrière-plan, et la 2eme classe qui a une valeur des pixels 255 représente l'objet.

III.4.3.3 Extraction des contours :

Les pixels de bord sont des pixels auxquels l'intensité d'une image change brusquement, et les bords (ou segments de bord) sont des ensembles de pixels de bord connectés (*Figure III.23*). Les détecteurs de bord sont des outils du traitement d'images locaux conçus pour détecter les pixels de bord. Une ligne peut être vue comme un segment de bord mince dans lequel l'intensité de l'arrière-plan soit côté de la ligne est soit beaucoup plus élevé ou beaucoup plus bas que l'intensité de la ligne pixels [24].



Figure III.23: Exemple d'un changement brusque d'intensité des pixels.

Dans notre cas, Opencv utilise la fonction **findcontours()** pour détecter cette brusque différence d'intensité dans les images binaires issues de la phase de seuillage, donc trouver un contour revient à trouver un objet blanc sur un fond noir.



Figure III.24: Les contours d'objet utilisés.

III.4.3.4 Identification de classe :

L'approximation polygonale de contour est un algorithme pour réduire le nombre de points dans une courbe avec un ensemble de segments de droite — d'où le terme approximation. Cet algorithme repose sur l'hypothèse qu'une courbe peut être approchée par une série de segments de lignes courtes. Il en résulte une courbe approximative qui consiste en un sous-ensemble de points définis par la courbe originale.

Il est important de comprendre qu'un contour consiste en une liste de sommets. Nous pouvons vérifier le nombre d'entrées dans cette liste pour déterminer la forme d'un objet. L'approximation des contours est déjà implémentée dans OpenCV via la méthode `cv2.approxPolyDP`.

- La première classe : si le contour approximatif a trois sommets, alors il doit s'agir d'un triangle.

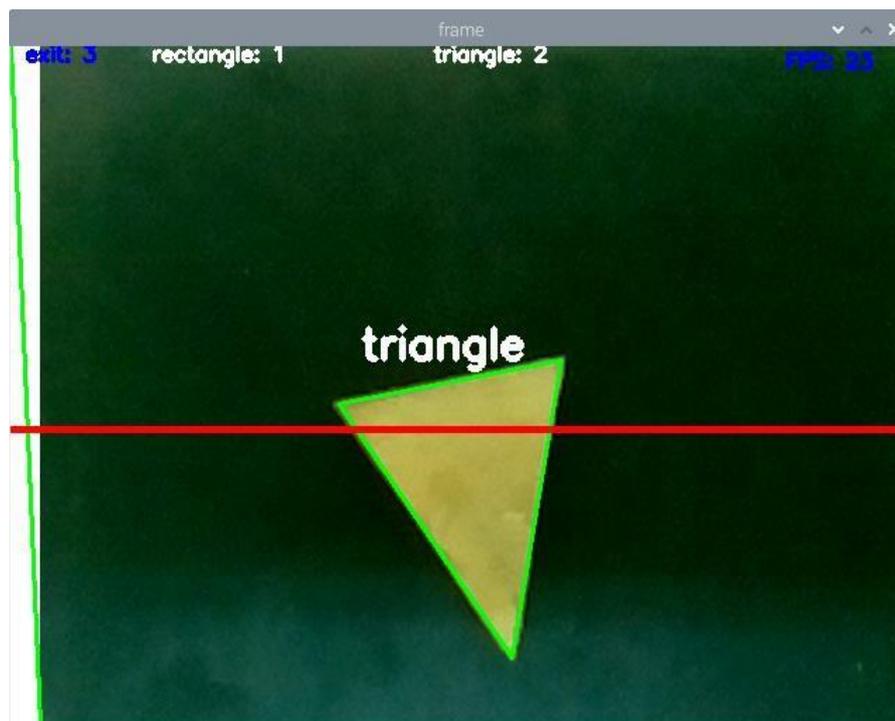


Figure III.25: Identification de la première classe Triangle.

- La deuxième classe : si le contour approximatif a quatre sommets, alors il doit s'agir d'un rectangle.

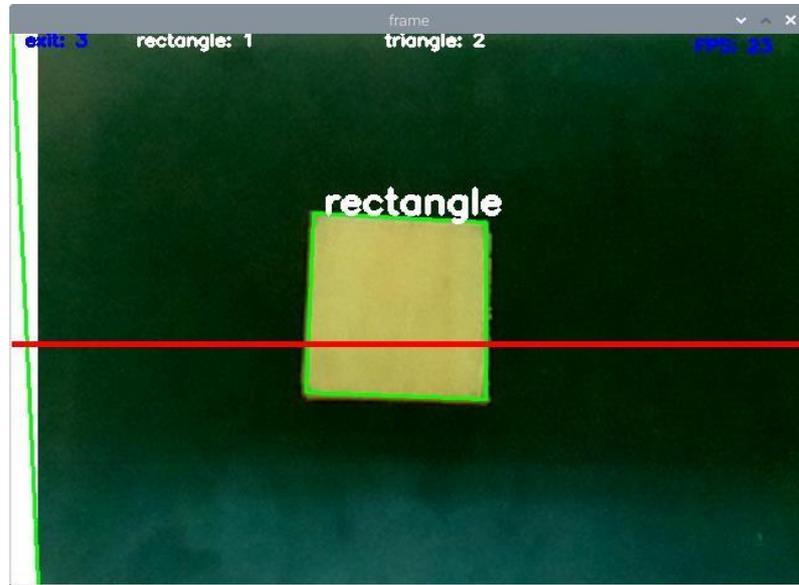


Figure III.26: Identification de la deuxième classe Rectangle.

La ligne rouge représente une ligne virtuelle qui a pour but de faciliter le comptage des objets une seule fois dans une boucle des cadres prise par la caméra. Si l'objet de classe rectangle passe par la ligne, le compteur rectangle incrémente et si un objet de classe triangle le traverse, le compteur de triangle incrémente.

III.4.4. Séparation des classes :

Après l'identification de classe, on désire faire un tri automatique selon la forme d'objet. Pour cela des capteurs à infrarouges placés à 50cm après le point de départ du convoyeur, nous pouvons détecter la présence d'un mouvement dans la portée variable du capteur.

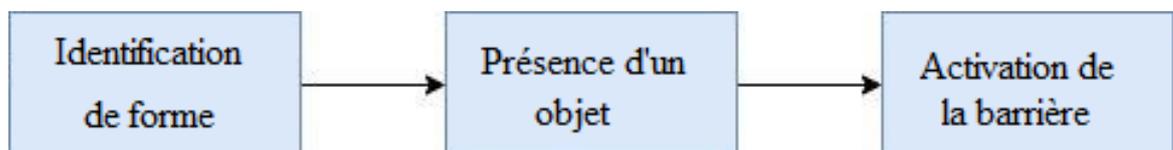


Figure III.27: Schéma bloc des étapes de l'objet.

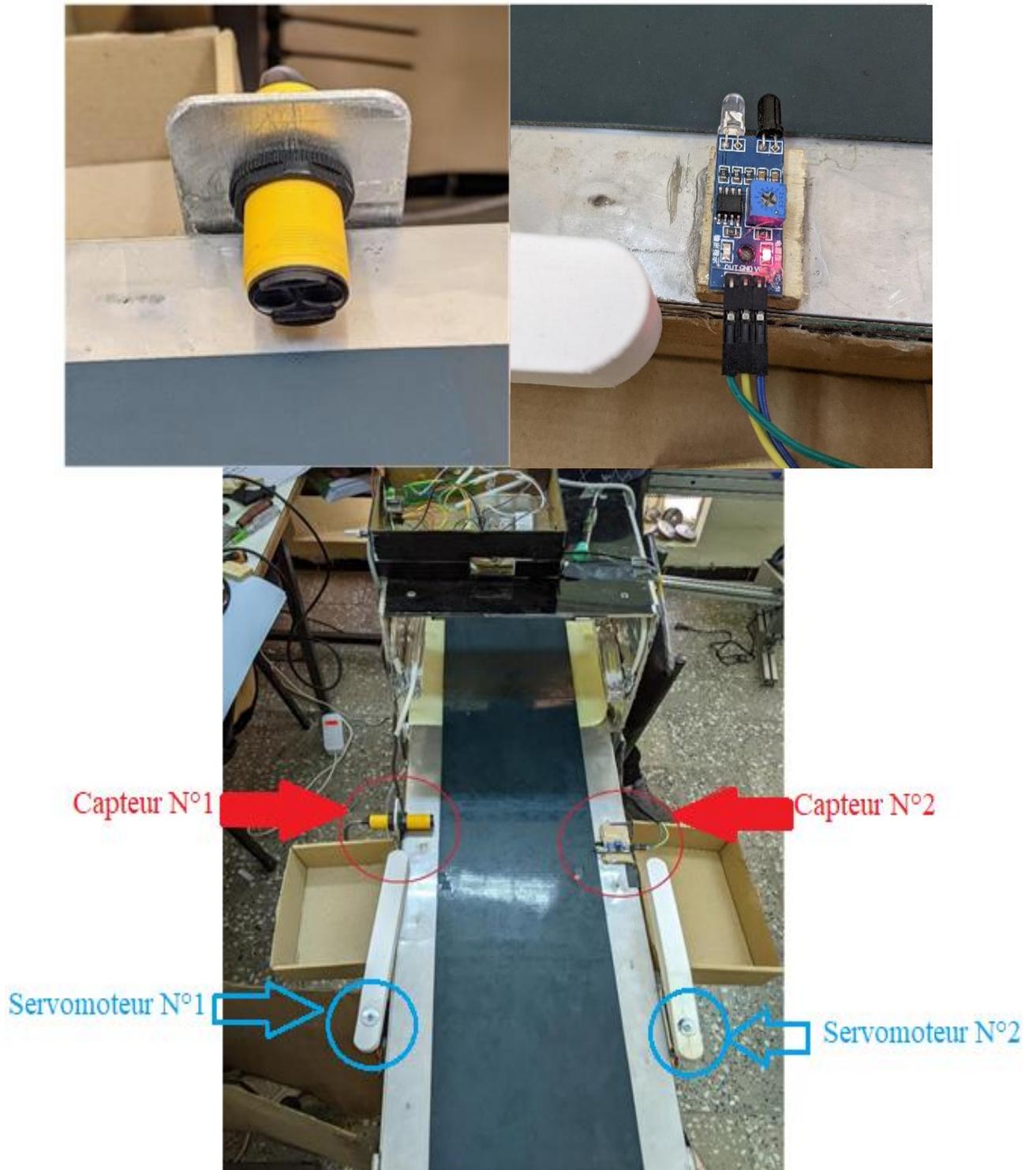


Figure III.28: Capteurs infrarouges utilisés.

Un capteur infrarouge permet de détecter un mouvement dans son champ de vision en se basant sur le rayonnement infrarouge (voir chapitre II). Si un mouvement est détecté le signal en sortie du capteur est mis au niveau BAS (0), si aucun mouvement n'est détecté le signal en sortie du capteur est mis au niveau HAUT (1). L'inversion des niveaux bas et haut est due au type de capteur soit NPN ou PNP.

Nous avons vérifié deux conditions pour l'activation de la barrière :

- Si la classe d'objet est un triangle et le capteur numéro (1) est mis au niveau Bas (0) (présence d'un objet) cela implique l'activation du servo-moteur numéro (1).

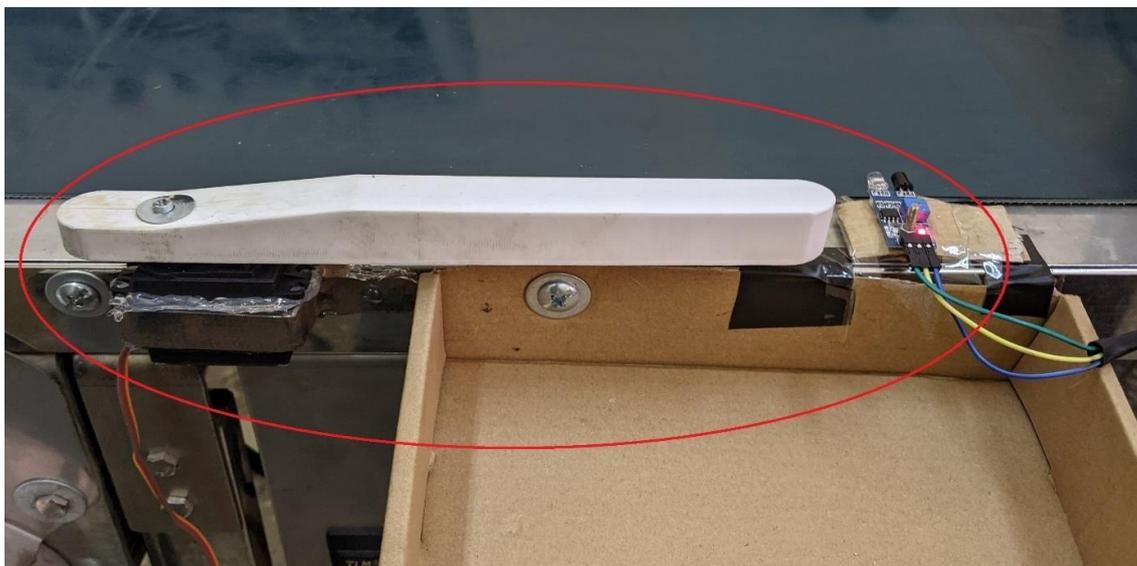


Figure III.29: Capteurs de présence infrarouge et le servomoteur avec la barrière pour la classe triangle.

- Si la classe d'objet est un rectangle et le capteur numéro (2) est mis au niveau Bas (1) (présence d'un objet) cela implique l'activation du Servo-moteur numéro (2).

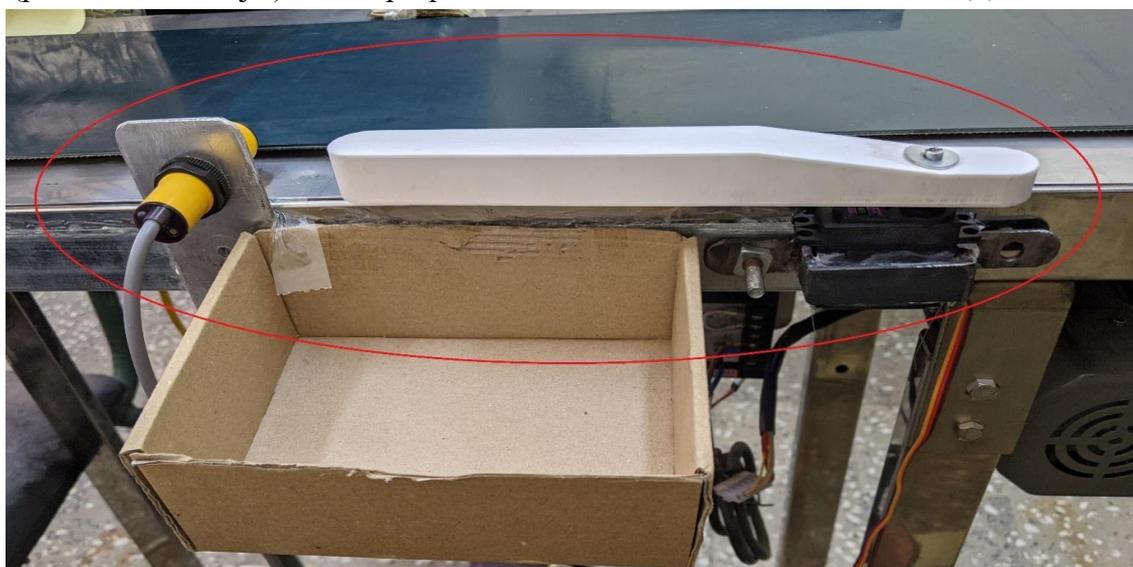


Figure III.30: Capteurs de présence infrarouge et le servomoteur avec la barrière pour la classe rectangle

L'angle du servo moteur bascule entre 0 et 90 ° si les conditions sont vérifiées à travers le signal PWM avec une temporisation de 0.8s entre l'angle initiale et final.

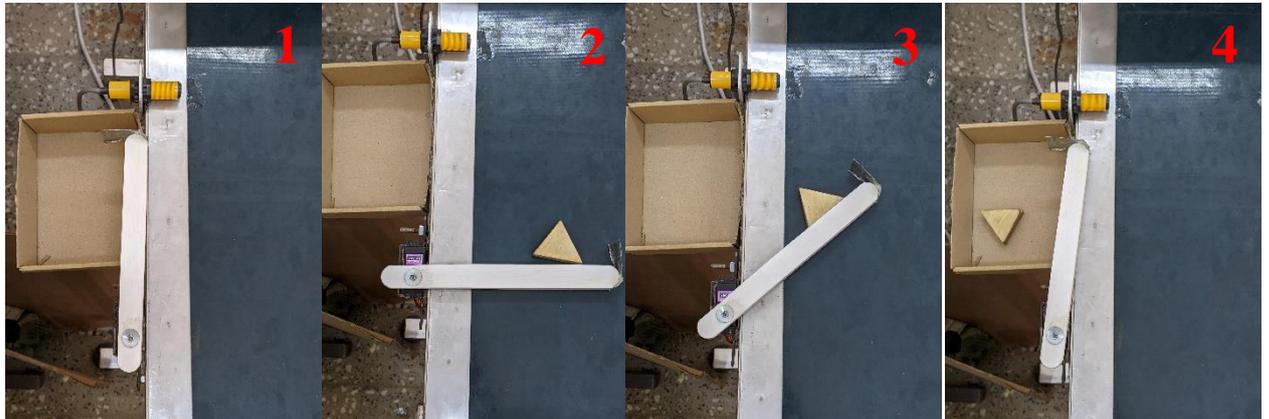


Figure III.31 : Différentes étapes de fonctionnement de la barrière pour la classe triangle.

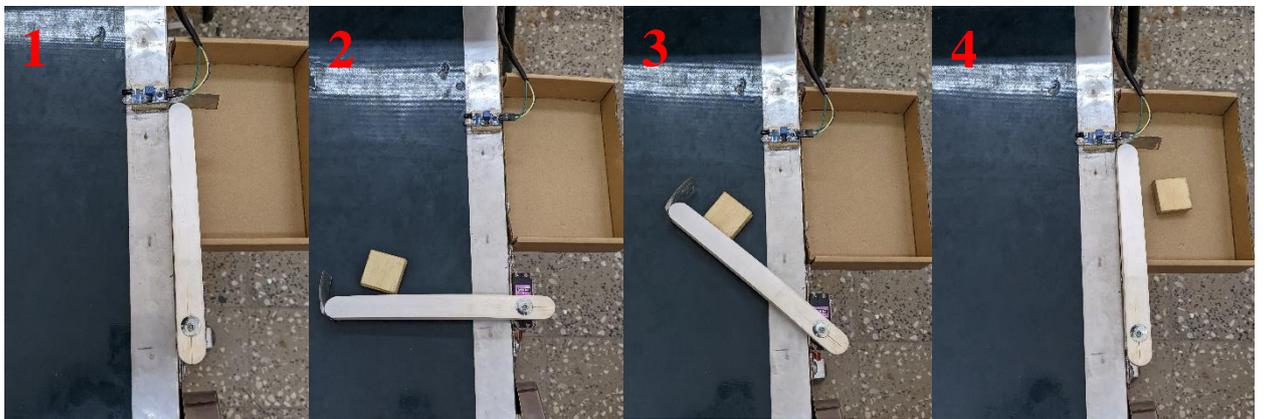


Figure III.32 : Différentes étapes de fonctionnement de la barrière pour la classe rectangle.

L'engrenage du servo moteur est attaché avec une barrière imprimée à l'aide d'une imprimante 3D.

La conception a été faite avec SolidWorks comme suivant :

- Longueur : 22cm.
- Hauteur : 2.5cm.
- Epaisseur : 2.2cm.

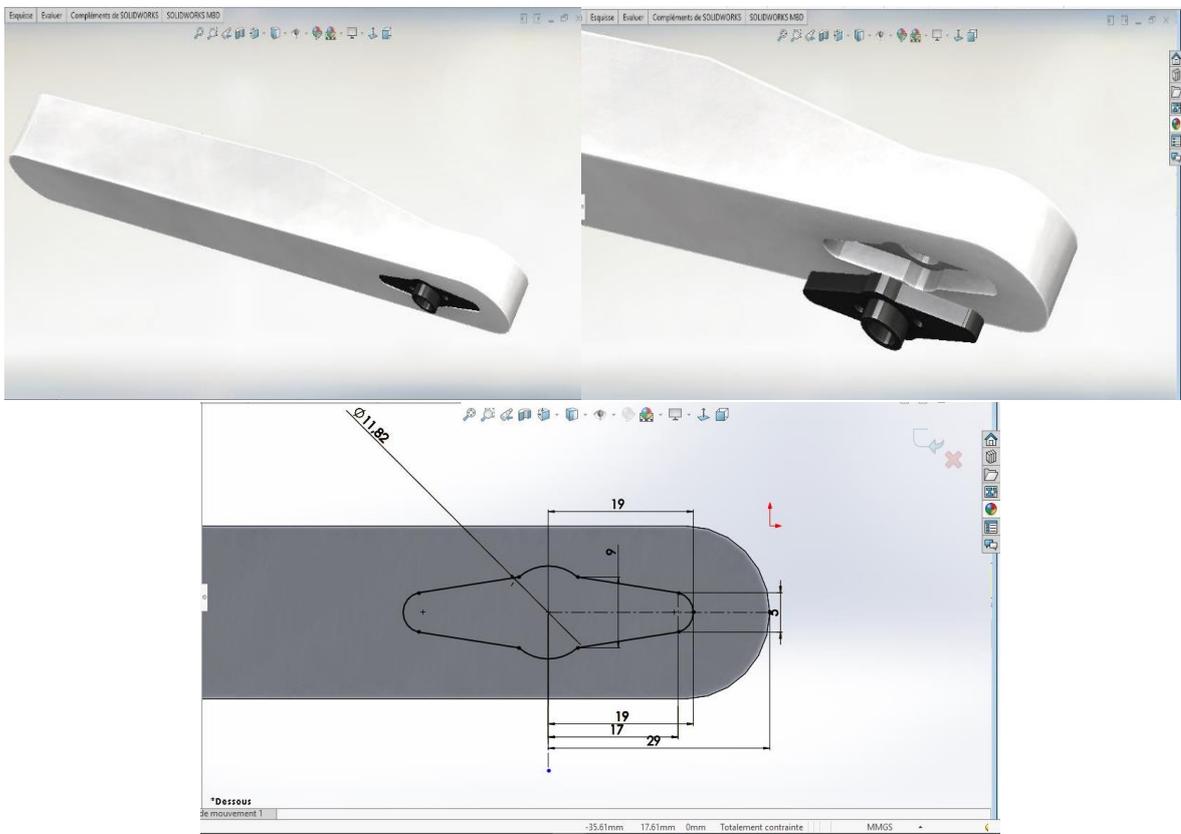


Figure III.33: Capture d'écran sur la conception de SolidWorks.

III.5. Résultats des tests :

Dans cette partie nous avons réalisé une série de tests pour vérifier le bon fonctionnement du système proposé. Nous avons utilisé 3 vitesses du convoyeur 7.16 cm/s (*Tableau III.1*), 8 cm/s (*Tableau III.2*) et 10.25 cm/s (*Tableau III.3*), nous avons utilisé 3 différentes catégories de test d'objets :

- Première catégorie d'objets de 100 rectangles.
- Deuxième catégorie d'objets de 100 triangles.
- Troisième catégorie d'objets de 50 rectangles et de 50 triangles alternativement.

Catégories	Triangle	Rectangle	Nb° triangle détecté	Nb° Rectangle détecté	Nb° d'objet raté	Pourcentage %
1	/	100	/	100	0	100 %
2	100	/	100	/	0	100 %
3	50	50	50	50	0	100 %

Tableau III.1: Résultats des tests avec la vitesse 7.16 cm/s.

- Nous pouvons conclure à partir des résultats obtenus que la vitesse de 7.16 cm/s est une vitesse optimale car l'identification, le comptage et le tri fonctionnent sans aucune erreur et un taux de succès de 100%.

Catégories	Triangle	Rectangle	Nb° triangle détecté	Nb° Rectangle détecté	Nb° d'objet raté	Pourcentage %
1	/	100	/	94	6	94 %
2	100	/	90	/	10	90 %
3	50	50	45	46	9	91 %

Tableau III.2 : Résultats des tests avec la vitesse 8 cm/s.

- Pour les résultats avec la vitesse du convoyeur de 8 cm/s, nous pouvons dire que dans la première catégorie de 100 rectangles nous avons 6 objets non détectés avec un taux de succès de 94%, et pour la deuxième série de 100 triangles le nombre d'objets non détectés a augmenté de 10 objets avec un taux de succès de 90%.

- Pour la catégorie de 50 triangles et 50 rectangles alternativement, nous avons 5 triangles et 4 rectangles non détectés et un taux de succès 91%.

Catégories	Triangle	Rectangle	Nb° triangle détecté	Nb° Rectangle détecté	Nb° d'objet raté	Pourcentage %
1	/	100	/	83	17	83 %
2	100	/	74	/	26	74 %
3	50	50	38	39	23	77 %

Tableau III.3: Résultats des tests avec la vitesse 10.25 cm/s.

- Pour les résultats avec la vitesse du convoyeur de 10.25 cm/s, nous pouvons dire que dans la première catégorie de 100 rectangles nous avons 17 objets non détectés avec un taux de succès de 83%, et pour la deuxième série de 100 triangles le nombre d'objets non détectés a augmenté à 26 objets avec un taux de succès de 74%.
- Pour la catégorie de 50 triangles et 50 rectangles alternativement, nous avons 12 triangles et 11 rectangles non détectés et un taux de succès 77%.

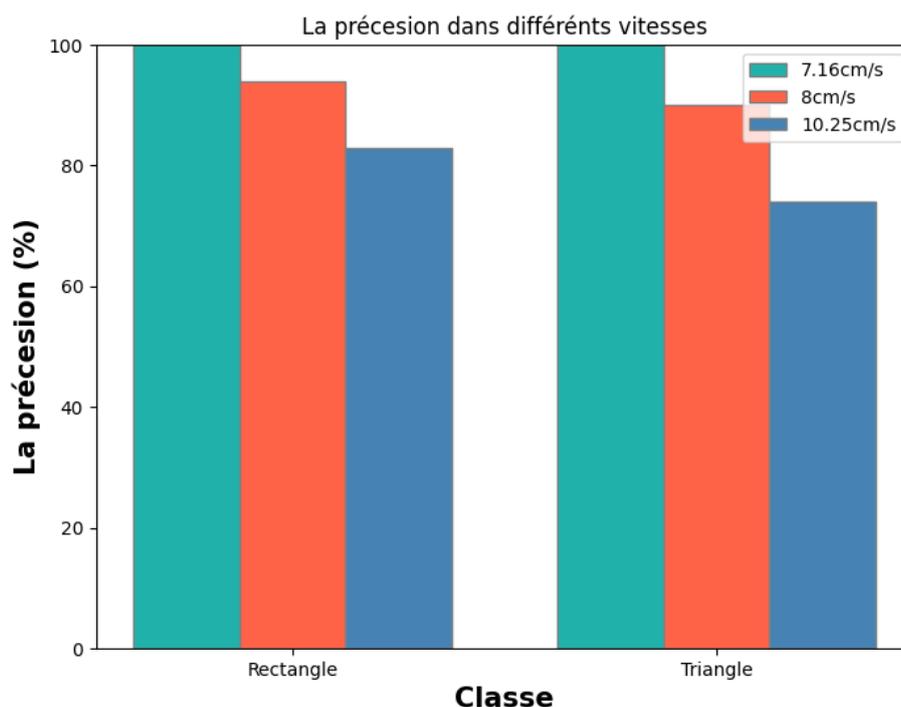


Figure III.34 : Précision pour les différentes vitesses du convoyeur.

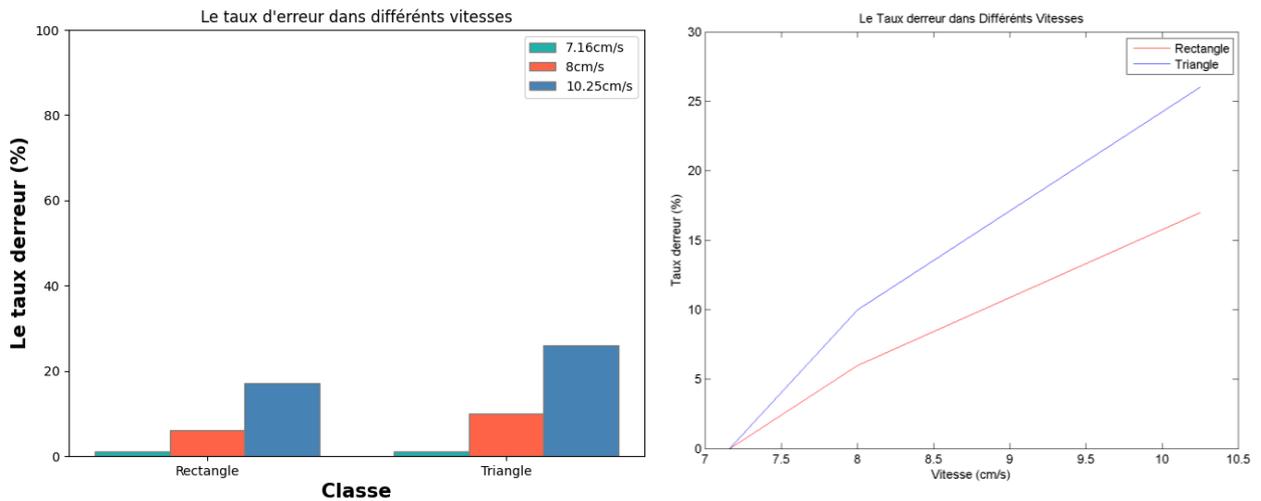


Figure III.35: Taux d'erreur pour différentes vitesses du convoyeur.

- Les paramètres sur lesquels nos tests ont été effectués sont calculés comme suit :
La distance entre le début et la fin de la scène (champ de vision) est de 26 cm, la caméra fait l'acquisition en 60 FPS, donc 1 image toute les 16 ms.
Pour 1 cycle de programme (1 capture) le FPS moyen réalisée par notre programme était de 24 FPS donc :

$$\frac{1 \text{ frame}}{24 \text{ FPS}} = 0.041 \text{ seconde pour chaque cycle de programme}$$

- Tout d'abord, nous avons utilisé une vitesse de 7.16 cm/s :

La durée de passage de l'objet dans la scène :

$$\frac{26 \text{ cm} * 1 \text{ sec}}{7.16 \text{ cm}} = 3.6 \text{ seconde}$$

Dans cette période, la caméra prend 89 instantanés (capture) :

$$\frac{3.6 \text{ sec}}{0.041 \text{ sec}} = 89 \text{ frame}$$

- Ensuite, nous avons utilisé une vitesse de convoyeur jusqu'à 8 cm/s.

La durée de passage de l'objet dans la scène :

$$\frac{26 \text{ cm} * 1 \text{ sec}}{8 \text{ cm}} = 3.25 \text{ seconde}$$

Dans cette période, la caméra prend 80 instantanés (capture)

$$\frac{3.25 \text{ sec}}{0.041 \text{ sec}} = 80 \text{ frame}$$

- Enfin, nous avons terminé par utiliser une vitesse de convoyeur de 10.25 cm / sec.

La durée de passage de l'objet dans la scène :

$$\frac{26 \text{ cm} * 1 \text{ sec}}{10.25 \text{ cm}} = 2.5 \text{ seconde}$$

Dans cette période, la caméra prend 60 instantanés (capture).

$$\frac{2.5 \text{ sec}}{0.041 \text{ sec}} = 60 \text{ frame}$$

La distance de chevauchement d'un objet dans 1 cycle de programme pour chaque vitesse (objet de longueur 3cm) :

1) Vitesse 7.16 cm/s :

La durée de passage de l'objet dans la scène : 3.6s

Pour chaque cycle de programme 0.041 seconde, le nombre de capture est :

$$\frac{3.6 \text{ s}}{0.041 \text{ s}} = 88$$

Dans une période de 3.6s on obtient 88 exécutions ou cycles de programme :

$$\frac{26 \text{ cm}}{88} = 0.295 \text{ cm}$$

Pour une vitesse de 8 cm/s le chevauchement d'un objet est 2.95 mm par 1 cycle de programme.

2) Vitesse 8 cm/s :

La durée de passage de l'objet dans la scène : 3.25s

Pour chaque cycle de programme 0.041 seconde, le nombre de capture est :

$$\frac{3.25 \text{ s}}{0.041 \text{ s}} = 80$$

Dans une période de 3.25s on obtient 80 exécutions ou cycles de programme.

$$\frac{26 \text{ cm}}{80} = 0.325 \text{ cm}$$

Pour une vitesse de 8 cm/s le chevauchement d'un objet est 3.25 mm par 1 cycle de programme.

1) Vitesse 10.25 cm/s :

La durée de passage de l'objet dans la scène : 2.5s

Pour chaque cycle de programme 0.041 seconde, le nombre de capture est :

$$\frac{2.5 \text{ s}}{0.041 \text{ s}} = 61$$

Dans une période de 2.5s on obtient 61 exécutions ou cycles de programme.

$$\frac{26 \text{ cm}}{61} = 0.426 \text{ cm}$$

Pour une vitesse de 10.25 cm/s le chevauchement d'un objet est **4.26 mm** par 1 cycle de programme.

- La vitesse du convoyeur et la complexité de la forme des objets jouent un rôle majeur dans le résultat de ce que nous obtenons dans ce projet, la bonne vitesse nécessite le bon matériau pour exécuter le programme plus rapidement et obtenir des résultats meilleurs et plus précis.

III.6. Problèmes rencontrés :

Tout au long de la réalisation de ce projet, nous avons été confrontés à différents problèmes tels que :

- La lumière réfléchi par la bande transporteuse qui est un matériau glacé qui interféré avec le programme causant un dysfonctionnement de reconnaissances de certains objets et nous avons des chutes FPS majeures lorsque la lumière réfléchi a été reçue par la caméra, l'inverse quand il n'y a pas assez de lumière atteignant l'objet, il devient obscur et donc le traitement n'identifiera pas l'objet comme une forme.
- La puissance du GPU (Raspberry) n'était pas suffisante, donc le FPS est entre 20 FPS et 27 FPS maximum ce qui nous a rendu incapables d'aller à des vitesses du convoyeur plus élevées et d'avoir des temps plus rapides pour trier un grand nombre d'objets.
- Pendant que nous utilisons le Raspberry pi comme centre d'acquisition et de traitement des images nous l'utilisons aussi pour la partie commande et qui créer un problème avec les servos qui ont besoin d'une fonction de veille temporelle (time sleep) pour fonctionner pendant que la barrière se déplace pour séparer un objet, en utilisant cette fonction de veille temporelle qui fonctionne comme une pause pour l'ensemble du programme les conséquences sont que la caméra cesse également de fonctionner et les données dans cette période sont perdues.

Conclusion générale

Notre travail a été scindé en plusieurs chapitres, le premier permettait de présenter les convoyeurs et la vision industrielle sur laquelle notre travail s'est fondé. Par la suite, nous avons donné quelques généralités sur les principaux outils utilisés dans notre travail notamment la caméra, Raspberry Pi, les servos et les capteurs.

En deuxième lieu, nous avons présenté une vue globale sur les notions, les méthodes et les étapes d'amélioration d'une image tels que le flou gaussien, seuillage Otsu et à la fin la présentation polygonale.

En dernier lieu, nous nous sommes focalisés sur la présentation du système que nous avons conçu et qui consiste à réaliser un système de tri automatique en temps réel.

Pour la partie de la vision industrielle nous avons défini deux classes géométriques sur lesquelles le système fonctionne, la première étant une forme géométrique triangle et la seconde une forme géométrique rectangle. Quand le convoyeur commence à transporter les objets à trier, les vitesses utilisées par le convoyeur nécessitent une amélioration en extrayant que des parties importantes de l'objet et en appliquant des méthodes telles que le filtrage pour réduire le bruit suivi d'une binarisation, puis une identification moyennant une détection de contours pour classer les objets qui passent en temps réel et faire le comptage global d'objets et par classe. Les barrières sont activées après une vérification d'une condition reliant le type d'objet et la valeur de retour d'un capteur désigné pour chaque classe.

Au cours de ce projet, nous avons pu acquérir beaucoup de connaissances. Nous avons utilisé les outils appris au cours de notre formation, et enrichi nos connaissances dans le domaine du traitement d'images et de convoyage en apprenant de nouveaux outils. Nous avons aussi appris à utiliser le langage de programmation Python.

Nous estimons que notre travail pourrait être exploité dans l'industrie. Nous proposons comme perspective la réalisation d'un système embarqué basé sur la vision artificielle capable de faire un triage avec précision de différents objets (produit alimentaires, manutention des bagages, pièces mécaniques ...etc.) dans un milieu industriel.

Annexe

Spécification électrique :

Symbol	Parameter	Minimum	Maximum	Unit
V _{IN}	5V Input Voltage	-0.5	6.0	V

Tableau : Cotes maximales absolues

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V _{IL}	Input low voltage ^a	VDD_IO = 3.3V	-	-	TBD	V
V _{IH}	Input high voltage ^a	VDD_IO = 3.3V	TBD	-	-	V
I _{IL}	Input leakage current	TA = +85°C	-	-	TBD	μA
C _{IN}	Input capacitance	-	-	TBD	-	pF
V _{OL}	Output low voltage ^b	VDD_IO = 3.3V, IOL = -2mA	-	-	TBD	V
V _{OH}	Output high voltage ^b	VDD_IO = 3.3V, IOH = 2mA	TBD	-	-	V
I _{OL}	Output low current ^c	VDD_IO = 3.3V, VO = 0.4V	TBD	-	-	mA
I _{OH}	Output high current ^c	VDD_IO = 3.3V, VO = 2.3V	TBD	-	-	mA
R _{PU}	Pullup resistor	-	TBD	-	TBD	kΩ
R _{PD}	Pulldown resistor	-	TBD	-	TBD	kΩ

^a Hysteresis enabled

^b Default drive strength (8mA)

^c Maximum drive strength (16mA)

Tableau : Caractéristiques de DC

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	t _{rise}	10-90% rise time ^a	-	TBD	-	ns
Digital outputs	t _{fall}	90-10% fall time ^a	-	TBD	-	ns

^a Default drive strength, CL = 5pF, VDD_IO = 3.3V

Tableau : Caractéristiques de la broche d'E/S numérique



Figure : Caractéristiques des entrées-sorties numériques

Références

bibliographiques

Références :

- [1] Introduction à la vision industrielle. www.etudier.com/dissertations/Introduction-a-La-Vision-Industrielle/375342.html.
- [2] Etude Mécanique et électrique d'un Convoyeur a bande afin d'augmenter sa charge (mémoire- option électromécanique - Université Abderrahmane Mira-Bejaia – 2012-2013).
- [3] QU'EST-CE QUE LA VISION INDUSTRIELLE ?
[HTTPS://WWW.COGNEX.COM/FR-BE/WHAT-IS/MACHINE-VISION/WHAT-IS-MACHINE-VISION](https://www.cognex.com/fr-be/what-is/machine-vision/what-is-machine-vision).
- [4] Principes de l'audiovisuel - Vincent MAZO - EFREI 2012/2013.
- [5] Licence Professionnelle Gestion de la Production Industrielle, spécialité Vision Industrielle - Léo MEIGNAN/ Jérôme SIX - http://www-iut.univ-lille1.fr/lp_vi/projets/rapport_cmos_ccd_2010.pdf.
- [6] Fonctionnement comparé de l'œil et de l'appareil photographique-
<https://www.maxicours.com/se/cours/fonctionnement-compare-de-l-oeil-et-de-l-appareil-photographique/>.
- [7] Raspberry Pi Camera v1.3 - <https://makersportal.com/shop/raspberry-pi-camera>.
- [8] what is a Raspberry Pi? - <https://opensource.com/resources/raspberry-pi>.
- [9] Micro system - <https://mjcsystem.tumblr.com/post/138675680110/le-servomoteur-comment-%C3%A7a-marche>.
- [10] Datasheet Servo MG996R -
https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf
- [11] 16-Channel 12-bit PWM/Servo Driver - PCA9685 - <https://grobotronics.com/16-channel-12-bit-pwm-servo-driver-pca9685.html>.
- [12] Le détecteur de mouvement infrarouge - <https://www.hagerservices.fr/securite-intrusion/detecteur-mouvement-infrarouge>.

- [13] Comment fonctionne la détection infrarouge, et quelles sont ses applications ? - <https://trustmyscience.com/fonctionnement-detection-infrarouge-et-ses-applications/>.
- [14] Capteur de proximité E18-D80NK spécification - <https://www.smart-cube.biz/produit/capteur-de-proximite-e18-d80nk-3-80cm/>.
- [15] Livre – Compture Vision with Python3 – Sraurabh Kapur -2017 – UK.
- [16] livre – Deep learning for Computer Vision with Python – Adrian Rosebrock - PyImageSearch – 2017.
- [17] Les couleurs dans le système HSV - https://panoramic-language.pagesperso-orange.fr/French/Magazine/ART027_hsv_par_jean_debord.pdf.
- [18] Traitement d’histogramme - <https://perso.esiee.fr/~perretb/I5FM/TAI/histogramme/index.html>.
- [19] Wikipedia – Gaussian blur - 21 - https://en.wikipedia.org/wiki/Gaussian_blur.
- [20] A class of Fast Gaussian Binomial Filters for Speech and Image Processing- Richard A. Haddad and Ali N. Akansu - <https://web.njit.edu/~akansu/PAPERS/Haddad-AkansuFastGaussianBinomialFiltersIEEE-TSP-March1991.pdf>.
- [21] Thresholding - <https://homepages.inf.ed.ac.uk/rbf/HIPR2/threshld.htm>.
- [22] Wikipedia – seuillage d’image - https://fr.wikipedia.org/wiki/Seuillage_d%27image.
- [23] Etudes comparative Entre Les Méthodes de Binarisation des images – option ; vision artificielle - Université Larbi Ben M’hidi Oum EL Bouaghi -2018/2019.
- [24] Digital Image Processing – Rafael C. Gonzalez – Richard E. Woods – 2018.
- [25] Wiki ubuntu-fr – SSH - <https://doc.ubuntu-fr.org/ssh>.