

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Mention Électronique
Spécialité Systèmes De Vision Et Robotique

présenté par

KORICHI ILYES

&

GUESSOUM SIDALI

Commande D'un Robot Mobile Par La Vision Artificielle , Implémentation Sur Raspberry Pi 3

Proposé par : M.Abderrahmane Namane

Année Universitaire 2017-2018

Remerciements

Je teins profondément à remercier tout d'abord ALLAH seigneur de l'univers, qui a prescrit que ce travail voit la lumière. Et qui m'adonné le savoir, la volonté, la force et le courage pendant toute lapériode de ma vie et de mes études...

Je remercie cordialement, Mon encadreur Mr. Namane Abderahmane , de m'avoir proposé ce thème, de m'avoir guidé et suivi durant l'accomplissement de ce travail.

Je remercie l'ensemble des Enseignant du département d'électronique pour les efforts qu'ils fournissent pour notre réussite dans nos études.

Et Sans Oublier les membres de (CSCCLUB ,ALKINDI ,ITC) ma deuxième famille sans exception.

Enfin, je remercie tout particulièrement mes parents, pour leur soutien inconditionnel tout au long de ces longues années d'études.

نظام التعرف على اشارة اليد هو تطبيق جهاز كمبيوتر قادر على تحديد اليد وقرائة الاشارة من صورة او إطار فيديو من مصدر الفيديو . العمل المقدم في هذا المشروع يهدف الى تحقيق نظام التعرف على اشارة اليد البشرية من خلال اليد عبر استعمال شبكة ذكاء اصطناعي متعددة الطبقات ، ثم يقوم بتحويل الاشارة الى امر تنفيذي للروبوت . هذا النظام تم تطبيقه على لوح راسبيري بي 3 .

كلمات المفاتيح: التعرف على اليد. اشارة اليد . التعلم .روبوت جوال .شبكة ذكاء اصطناعي.

Résumé :

La reconnaissance gestuelle est le processus capable de détecter et reconnaître le geste de la main à partir d'une image ou d'une séquence vidéo. Le travail présenté dans ce projet consiste à réaliser un système de commande de robot mobile basé sur la reconnaissance gestuelle de la main. Ce système de reconnaissance utilise un réseau de neurones afin de transformer le geste de la main en ordre exécutable pour le robot mobile. Ce système est implémenté sur Raspberry PI3.

Mots clés : Reconnaissance de mains, Geste de mains, Apprentissage, Robot Mobile, Réseau de neurones.

Abstract :

Gesture recognition is the process that can detect and recognize the gesture of the hand from an image or video clip. The work presented in this project consists in realizing a mobile robot control system based on gestural recognition of the hand. This recognition system uses a neural network to transform the hand gesture into an executable order for the mobile robot. This system is implemented on Raspberry PI3.

Keywords : Gesture recognition, hand gesture, learning, Robot Mobile, Neural Network.

Table des matières

Introduction générale	1
CHAPITRE 1 : généralités sur le traitement d'image et Réseau de neurones.....	3
La Première Partie :Généralités sur le traitement d'image	3
1.1 Introduction	3
1.2 Définition d'une image.....	3
1.3 Image numérique.....	4
1.4 La chaîne d'acquisition d'une image	6
1-5 Prétraitement et amélioration d'images	7
1.5.1 Egalisation de l'histogramme.....	7
1.5.2 Filtrage	8
1.6 Segmentation	8
La deuxième partie 2 :Réseau de neurones(Apprentissage).....	10
1.7 Historique.....	10
1.8 Le neurone biologique.....	10
1.9 FONCTIONNEMENT DES NEURONES	12
1.10 Le perceptron	11
1.11 Apprentissage.....	14
1.11.1 Position d'un probleme de classification par reseau de neurones	14
a) Acquisition des données.....	14
b) Prétraitement.....	15
c) Le choix des attributs.....	15
d)La décision	15
e)Apprentissage.....	16
1.12 Les différents types d'apprentissage.....	17
1.13.1 Le perceptron multicouche (PMC)	19
1.13.2 LA RETRO-PROPAGATION PAR DESCENTE DE GRADIENT.....	20
1.14 Conclusion	21
Chapitre 2 le robot mobile	22
La Première Partie :Généralités sur la robotique mobile	22
2.1.1 Introduction.....	22

2.1.2 Définition.....	23
2.1.3 les Types de robots.....	23
a) les robots manipulateurs.....	23
b) Les robots mobiles	24
2.1.4 Comparaison des différents types des robots.....	29
La deuxième partie :Réalisation du robot (mécanique du robot mobile)	30
2.2.1 Introduction	30
2.2.2 Structure	30
2.2.3 Le châssis	31
2.2.4 Les roue	31
2.2.5 Les actionneurs.....	33
a) Moteurs CC (XD-37GB555)	33
2.2.6 Les Capteurs	35
a)Codeur rotatif	35
2.2.7 Fabrication	35
La troisième partie : Du côté électronique.....	36
2.3.1 Introduction	36
2.3.2 Les cartes électroniques programmables	37
a) La carte Arduino	37
b) Le Raspberry Pi.....	39
2.3.3 Le module camera Raspberry Pi.....	40
2.3.4 La carte puissance	41
2.3.5 Les capteurs à ultrasons	42
2.3.6 Le Périphériques sans fil NRF24	43
La Quatrième Partie :Asservir Des Moteurs A Courant Continu Avec Un PID.....	44
2.4.1 Introduction	44
2.4.2 Asservissement PID	44
2.4.3 Régler les coefficients d'un PID.....	45
2.5 Conclusion	46
Chapitre 3 Réalisation et implémentation.....	47
3.1 Introduction.....	47
3.2 Système proposé	49

3.3 Problèmes rencontrés.....	50
3.4 TECHNOLOGIES UTILISÉES.....	50
3.4.1 Software (Programmation)	50
3.4.2 Hardware.....	51
3.5 La détection de la main.....	53
A)La méthode de Haar : détection d'objet.....	53
b)Principe de la méthode.....	53
c)Apprentissage du classifieur	54
d)Les caractéristiques.....	55
e) Implémentation de la méthode de détection de la main “ Viola & Jones ”	57
3.6 Système de reconnaissance de gestes	58
3.6.1 Reconnaissance de geste de la main : Approche Réseau Perceptron Multicouche	59
A)Base utilisée	59
b)Structure du réseau neuronal:	63
c)Les méthodes principales pour l'exécution de réseau de neurone PM.....	64
3.6.2 Reconnaissance pour un ensemble test.....	66
3.6.3 Résultats expérimentaux.....	66
a)Test 1	67
b)Test2 :.....	68
c) Les paramètres utilisés dans notre projet sont	69
3.6.4 Étapes de la phase d'exécution.....	69
3.6.5 La reconnaissance de geste est basée essentiellement sur les étapes suivant... ..	71
3.7 Discussion du résultat	72
3.8 Conclusion.....	72
Conclusion générale.....	74

Liste des figures

Figure 1.1 pixel de l'image.....	4
Figure 1.2 image binaire	5
Figure 1.3 image en teintes de gris.....	5
Figure 1.4 image couleur.....	6
Figure 1.5 la chaîne d'acquisition d'une image	6
Figure 1.6 la chaine de traitement d'images	7
Figure 1.7 Egalisation d'histogramme.....	8
Figure 1.8 segmentation d'une image.	9
Figure 1.9 : Neurone Biologique	11
FIGURE 1.10 – Représentation d'un neurone formel	13
Figure 1.11 Fonction activation à seuil	13
Figure 1.13 Les fonctions d'activation	14
Figure 1.14 – Partitionnement	16
Figure 1.15 –rétro propagation de l'erreur	17
Figure 1.16 Apprentissage non-supervisé.....	19
FIGURE 1.17 – Exemple de perceptron multicouche élémentaire avec une couche cachée et une couche de sortie	20
Figure 1.18 Principe de la rétropropagation	21
Figure 2.1: Robot de construction.....	23
figure 2.2 : Robot da vinci	23
Figure 2.3 : Robot pompiste.....	24
figure 2.4 : Bras manipulateurs dans l'industrie	24
Figure 2.5 robot mobile différentiel	25
figure 2.6 ATRV2, CDTA	25

Figure 2.7 Modélisation du robot.....	26
Figure 2.8 Robot de type tricycle	27
Figure 2.9 robot de type voiture	28
Figure 2.10 Robot de type omnidirectionnel	28
Figure 2.11 Le robot Wall-E	30
Figure 2.12 schéma du châssis	31
Figure 2.13 Les Roues	32
Figure 2.14 Image de la base du robot	32
Figure 2.15 les deux types de roues de robot.	33
Figure 2.16 Moteur à CC (XD-37GB555)	34
Figure 2.17 moteurs placés dans le châssis	34
Figure 2.18 Codeur rotatif	35
Figure 2.19 Schéma électronique de notre robot mobile.....	36
Figure 2.20 La carte Arduino Méga.....	37
Figure 2.21 La carte Arduino Uno	38
Figure 2.22 La carte Raspberry Pi 3 model B.....	39
Figure 2.23 Les caractéristiques du Raspberry Pi 3 model B.....	40
Figure 2.24 Le module camera Raspberry Pi V1.3.	40
figure 2.25 La carte puissance.....	42
Figure 2.26 Le capteur HC-SR04	42
figure 2.28 Le chipset NRF24L01	43
Figure 2.29 parcours optimale pour le robot mobile.....	44
Figure 2.30 l'évolution du nombre de tours par seconde du moteur au cours du temps.....	46
Figure 3.1 La Langue des Signes Française (LSF)	47
Figure 3.2 Geste dynamique	48

Figure 3.3 Geste Statique	48
Figure 3.4 schéma synoptique du système proposé	50
Figure 3.5 schéma détaillé du système proposé	52
Figure 3.6 Passage d'une des caractéristiques sur la main pendant l'apprentissage ..	55
Figure 3.7 Caractéristiques pseudo-haar avec différentes orientations.	55
Figure 3.8 images positif	56
Figure 3.9 images négative	57
Figure 3.10 Test de la méthode de détection de la main [Viola & Jones].....	57
Figure 3.11 Système de reconnaissance de gestes	58
Figure 3.12 Geste 1 (Quelques échantillons)	59
Figure 3.13 Geste 2 (Quelques échantillons)	60
Figure 3.14 Geste 3 (Quelques échantillons)	60
Figure 3.15 Geste 4 (Quelques échantillons)	61
Figure 3.16 Geste 1 (Quelques échantillons*binaire*)	61
Figure 3.17 Geste 2 (Quelques échantillons*binaire*)	62
Figure 3.18 Geste 3(Quelques échantillons*binaire*)	62
Figure 3.19 Geste 4 (Quelques échantillons*binaire*).....	63
Figure 3.20 Structure du réseau neuronal	64
Figure 3.21 schéma de l'algorithme.....	65
Figure 3.22 Qulques valeurs de poids synaptique existant dans ce fichier (81080 Poids synaptique)	65
Figure 3.23 Résultats de premier test	67
Figure 3.24 Test pour des images de 40*40 pixels	68
Figure 3.25 Résultats de 2 éme test.....	68
Figure 3.26 Test pour des images de 20*20 pixels.....	69
Figure 3.27 les étapes essentielles pour la reconnaissance de geste	70

Figure 3.28 schéma électronique de système71

Figure 3.29 Les commandes de contrôle72

Liste des tableaux

Tableau 1 Un tableau des entrées et des attributs.....25

Tableau 2 Les avantages et des inconvénients des différents types de robots à
roues.....39

Tableau 3 Les caractéristiques techniques des carte Arduino Uno et Méga.....48

Tableau 4 Les caractéristiques du module caméra Raspberry Pi v1.3.....51

Introduction générale

Dès la création de l'informatique, l'homme rêve de créer une parfaite image de lui-même, il est parvenu à implémenter et à se développer dans le domaine de l'intelligence artificielle.

L'intelligence artificielle (IA) désigne toutes les techniques et méthodes qui permettent de doter des systèmes informatiques de capacités proches de celles de l'être humain, comme à la science-fiction, ou on imagine une intelligence proche à celle de l'être humain, mais comme toute créature, elle a sûrement besoin d'une vision.

A l'heure actuelle, les systèmes de visions sont de plus en plus répandus (webcam, caméra de surveillance) et les caméras se sont installées partout dans notre quotidien. Elles sont utilisées pour réaliser de la vidéosurveillance (dans les magasins, rues ou aéroports).

Mais la capacité de l'œil et du cerveau humain dépasse de loin la technologie appropriée par l'homme.

Dans notre travail, nous présentons un robot mobile doté de cette intelligence. Ce robot est inspiré du Film Wall-E sortie en 2008, il sera capable de fonctionner et d'agir à partir d'un geste de la main en temps réel. Pourquoi la main ? La main est un outil de communication naturel pour l'homme dans son quotidien, spécialement pour les sourds-muets.

Mais, une interaction humain-machine a plusieurs scenarios, comme le contrôle d'un robot mobile.

Ce mémoire est établi sur trois chapitres. Dans le premier chapitre, nous présentons les généralités sur le traitement d'image et Réseau de neurones. Le deuxième chapitre est consacré au robot mobile. Dans le troisième chapitre, nous présentons la réalisation et son implémentation en temps réel.

Généralités sur le traitement d'image et Réseau de neurones

La Première Partie :

Généralités sur le traitement d'image

1.1 Introduction

Ce chapitre a pour objectif de présenter les techniques de traitement d'images et de vision par ordinateur. Par traitement d'images, on désigne l'ensemble des opérations sur les images numériques, qui transforment une image en une autre image, ou en une autre primitive formelle. La vision par ordinateur désigne la compréhension d'une scène ou d'un phénomène à partir d'informations « image », liant intimement perception, comportement et contrôle. Les domaines traités vont du traitement du signal à l'intelligence artificielle, on ne saurait donc prétendre à l'exhaustivité, mais on vise plutôt l'exploration d'un certain nombre de techniques importantes et actuelles. Traitement d'image et vision par ordinateur sont des disciplines relativement jeunes (~années 60) et qui évoluent rapidement. Elles sont en plein expansion et donnent lieu chaque année à une profusion de travaux, académiques, technologiques et industriels. Cette profusion s'explique par le caractère ardu du sujet : complexité algorithmique dû aux énormes volumes de données, caractère mal posé des problèmes et difficultés à formaliser une faculté biologique « évidente ». D'autre part l'engouement pour ces disciplines s'explique par la multiplication permanente d'applications et d'enjeux industriels dans des domaines aussi variés que : médecine, télécommunications, automobile, météorologie, défense, jeux vidéo, art, écologie...[1]

1.2 Définition d'une image

C'est la représentation d'une personne ou d'un objet par la peinture, la sculpture, la photographie, etc. C'est aussi un ensemble structuré d'informations qui après affichage sur écran, ont une signification pour l'œil humain. Elle peut être écrite sous forme de fonction $I(x, y)$ ou I est une fonction d'intensité lumineuse ou de couleur aux coordonnées spatiales

(x, y). De cette façon l'image est exploitable par la machine, d'où la nécessité de sa numérisation.

1.3 Image numérique

L'image numérique est une matrice codée sous un certain nombre de bit, Les images numériques peuvent être acquises par plusieurs dispositifs qui s'adaptent aux domaines d'application.

L'image numérique est une matrice dont les éléments sont des pixels, chaque pixel est repère par sa position spatiale et sa luminance. Cette luminance indique le niveau de gris (voire la figure 1.1) pour les images à niveau de gris et la couleur pour les images en couleurs [2].

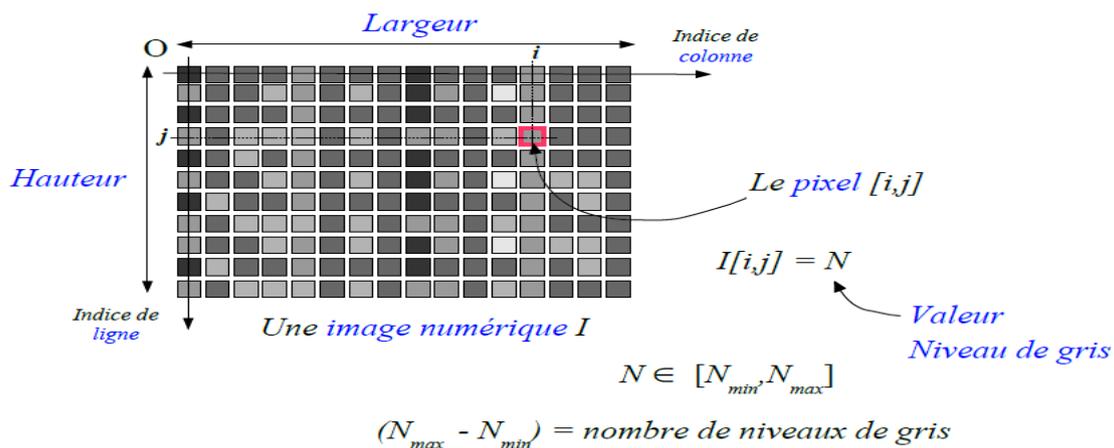


Figure 1.1 pixel de l'image.

On peut classifier les types d'image en 3 catégories :

1. Les images binaires (noir ou blanc) ce sont les images les plus simples, un pixel peut prendre uniquement les valeurs noir ou blanc. C'est typiquement le type d'image que l'on utilise pour scanner du texte quand celui-ci est composé d'une seule couleur (voire la figure 1.2).



Figure 1.2 image binaire

2. les images à niveau de gris: elles se composent de 256 niveaux de gris, la valeur 0 représente le noir (aucune intensité lumineuse) et la valeur 255 représente le blanc (intensité lumineuse maximale) (voire la figure 1.3).



Figure 1.3 image en teintes de gris

3. Les images couleurs: l'espace couleur ou genre vert bleu sont les plus utilisé pour le maniement des images, l'espace couleur est basé sur la synthèse additive des couleurs donc le mélange des composantes initiales (R V B) (voire la figure 1.4) [3].

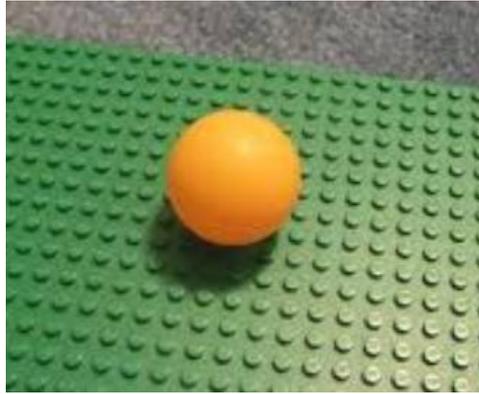


Figure 1.4 image couleur

1.4 La chaîne d'acquisition d'une image

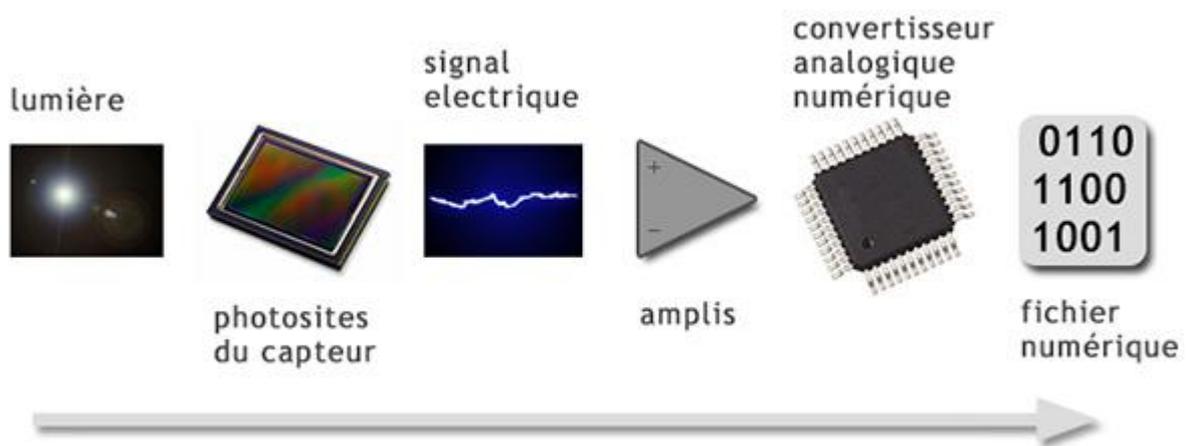


Figure 1.5 la chaîne d'acquisition d'une image

Partir d'une scène réelle pour aboutir à une photo numérique implique de nombreuses étapes: les photosites du capteur récoltent les photons qui composent la lumière pour les transformer en signal électrique. Ce signal est amplifié avant d'être converti en données numériques comme l'indique la figure 1.5 [4].

1-5 Prétraitement et amélioration d'images

Cette phase se produit juste après l'obtention des images et à pour objectif d'améliorer la qualité de l'image dans le but de la segmentation et l'extraction de l'information.

Le temps de traitement est très important si on veut un travail en temps réel. C'est un facteur décisif, il doit être le plus petit possible (donc le plus optimisé). Ceci concerne les opérateurs locaux, ce qui veut dire qu'ils doivent agir sur un nombre limité de pixels et essentiellement sur les pixels dans le voisinage proche du pixel courant.



Figure 1.6 la chaîne de traitement d'images

Globalement, les techniques de prétraitements envisagent l'utilisation des étapes suivantes suivantes :

- Egalisation de l'histogramme.
- Filtrage.

1.5.1 Egalisation de l'histogramme

Dans une image naturelle qui a été quantifiée de manière linéaire, une majorité de pixels ont une valeur inférieure à la luminance moyenne. C'est pourquoi les détails dans les régions sombres sont difficilement perceptibles. Pour résoudre ce problème, on utilise alors une méthode appelée égalisation de l'histogramme. L'égalisation procède de la manière suivante: On part du niveau de gris zéro et on calcule la valeur cumulée des niveaux suivants dans l'histogramme original jusqu'au moment où la somme est la plus près possible d'une certaine valeur; en général, cette valeur est égale à $\frac{P \times P}{N_F}$ qui est le nombre de pixels moyen,

avec P est la dimension de l'image en pixels, et N_F est le nombre de niveaux de gris dans l'image résultat (voire la figure 1.7).

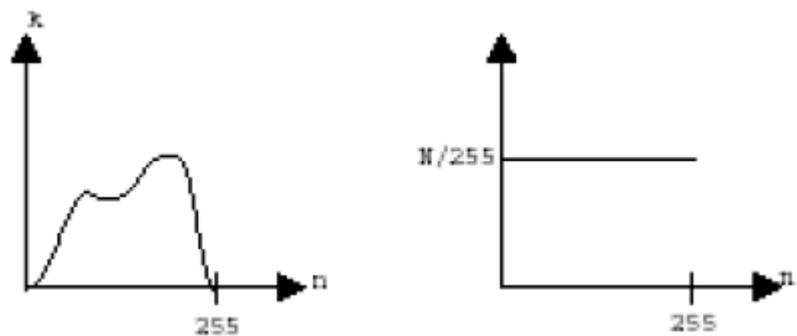


Figure 1.7 Egalisation d'historgramme

1.5.2 Filtrage

Tout ce qui est indésirable dans une image est considéré comme un phénomène de brusque variation d'un pixel isolé par rapport à ses voisins. Il peut provenir de l'éclairage, des dispositifs optiques et électroniques du capteur ou de l'image elle-même. Pour lutter contre les effets du bruit, il est nécessaire de faire des opérations de filtrage sur chaque pixel en tenant compte de son voisinage. [6]

Les opérations de filtrage, qui ont pour objet d'atténuer l'effet du bruit et faire disparaître les défauts et les perturbations continues dans une image, sont indispensables pour une bonne extraction de l'information utile. On distingue deux types de filtrage :

Filtrage linéaire : où la transformation d'un pixel est le fruit d'une combinaison linéaire des pixels voisins.

Filtrage non linéaire : où les pixels voisins interviennent suivant une loi non linéaire.

1.6 Segmentation

La segmentation d'image est une opération de traitement d'image qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond. Si le nombre de classes est égal à deux, elle est appelée aussi binarisation.

Elle consiste à découper l'image considérée en régions ayant une homogénéité selon un critère prédéfini (niveau de gris, moments, coefficients d'ondelettes ...). L'union des régions obtenues doit redonner l'image initiale. La segmentation d'image n'est pas considérée comme un but en soi, mais dépend du type de traitement fixé a priori (voire la figure 1.8).

On regroupe généralement les algorithmes de segmentation en trois grandes catégories :

- La segmentation basée sur les pixels
- La segmentation basée sur les régions
- La segmentation basée sur les contours

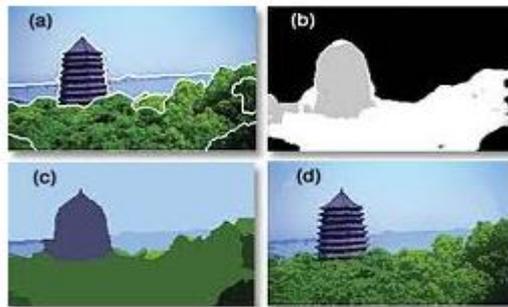


Figure 1.8 segmentation d'une image.

La deuxième partie 2 :

Réseau de neurones

(Apprentissage)

1.7 Historique

L'Intelligence Artificielle, branche de l'Informatique fondamentale s'est développée avec pour objectif la simulation des comportements du cerveau humain. Les premières tentatives de modélisation du cerveau sont anciennes et précèdent même l'ère informatique. C'est en 1943 que Mc Culloch (neurophysiologiste) et Pitts (logicien) ont proposé les premières notions de neurone formel.

Ce concept fut ensuite mis en réseau avec une couche d'entrée et une sortie par Rosenblatt en 1959 pour simuler le fonctionnement rétinien et tacher de reconnaître des formes. C'est l'origine du perceptron. Cette approche dite connexionniste a atteint ses limites technologiques, compte tenu de la puissance de calcul de l'époque, mais aussi théoriques au début des années 70 [14].

Remis en veilleuse depuis le milieu des années 90 au profit d'autres algorithmes d'apprentissage machine ou plutôt statistique : boosting, support vector machine..., les réseaux de neurones connaissent un regain d'intérêt et même un énorme battage médiatique sous l'appellation d'apprentissage profond (Deep Learning). La taille des bases de données, notamment celles d'images issues d'internet, associée à la puissance de calcul disponible, permettent d'estimer les millions de paramètres de perceptrons accumulant des dizaines voire centaines de couches de neurones aux propriétés très spécifiques. Ce succès médiatique est la conséquence des résultats spectaculaires obtenus par ces réseaux en reconnaissance d'image, jeux de go, traitement du langage naturel... [14].

1.8 Le neurone biologique

Un cerveau humain comporte plus de 1000 milliards de neurones avec des milliers de connexions par neurone. Les neurones sont reliés entre eux par des axones en entrée et des dendrites en sortie. Un neurone biologique contient un corps cellulaire, des dendrites qui

sont de fines extensions qui forment une arborescence autour du noyau. Les données envoyées à un neurone sont captées par les dendrites. C'est par l'axone que les informations partent du neurone. L'axone se connecte ensuite aux dendrites d'autres neurones. Une synapse est une connexion entre deux neurones (principalement entre une axone et une dendrite). Les connexions sont établies par impulsions électriques. Les signaux n'opèrent pas de manière linéaire, l'information passe toujours dans le même sens, de l'axone vers les dendrites. Le neurone reçoit donc l'information par les dendrites, une sommation est faite au niveau du noyau. En fonction d'un potentiel d'action, de l'information est véhiculée au travers de l'axone jusqu'à la terminaison synaptique [15].

Pour simplifier, un neurone biologique est composé de:

- Les synapses (point de connexion entre les neurones), fibres nerveuses ou musculaires.
- Les dendrites qui sont les entrées du neurone.
- L'axone qui est la sortie du neurone vers les autres neurones ou fibres musculaires.
- Le noyau qui va activer la sortie en fonction des entrées

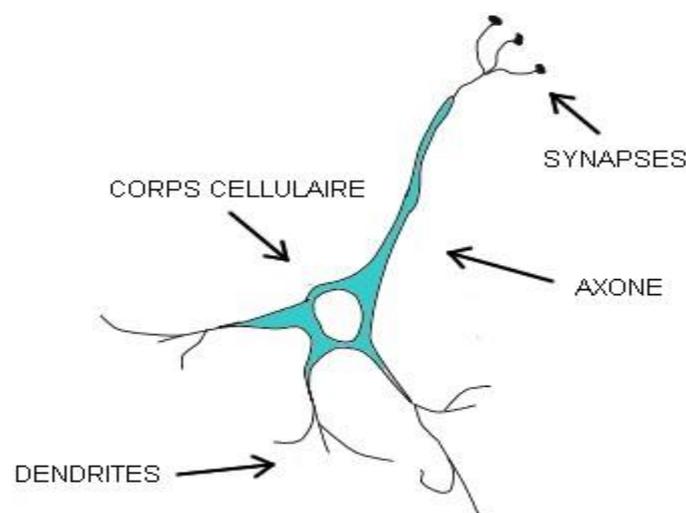


Figure 1.9 : Neurone Biologique

1.9 FONCTIONNEMENT DES NEURONES

Les neurones reçoivent les signaux (impulsions électriques) par des extensions très ramifiées de leur corps cellulaire (les dendrites) et envoient l'information par de longs prolongements (les axones). Les impulsions électriques sont régénérées pendant le parcours le long de l'axone. La durée de chaque impulsion est de l'ordre d'1 ms et son amplitude d'environ 100 millivolts. Les contacts entre deux neurones, de l'axone à une dendrite, se font par l'intermédiaire des synapses. Lorsqu'un potentiel d'action atteint la terminaison d'un axone, des neuromédiateurs sont libérés et se lient à des récepteurs post-synaptiques présents sur les dendrites. L'effet peut être excitateur ou inhibiteur. Chaque neurone intègre en permanence jusqu'à un millier de signaux synaptiques. Ces signaux n'opèrent pas de manière linéaire (effet de seuil) [08].

1.10 Le perceptron

Le perceptron est un modèle inspiré du neurone biologique. Il permet de classifier des objets qui appartiennent à deux classes linéairement séparables. C'est la forme la plus simple pour un réseau de neurones. Le réseau dispose d'un seul neurone. Un neurone artificiel est une unité de traitement élémentaire qui reçoit les entrées et fournit un résultat en sortie. Le neurone fait une somme pondérée de toutes ses entrées. Le résultat est comparé à un seuil appelé biais. Si ce seuil est atteint alors le neurone est excité, sinon le neurone est inhibé.

Le biais est le potentiel d'activation du neurone. Celui-ci est soustrait de la somme des entrées des neurones pondérée par les poids. Plus il est important, moins le neurone a une chance d'être actif.

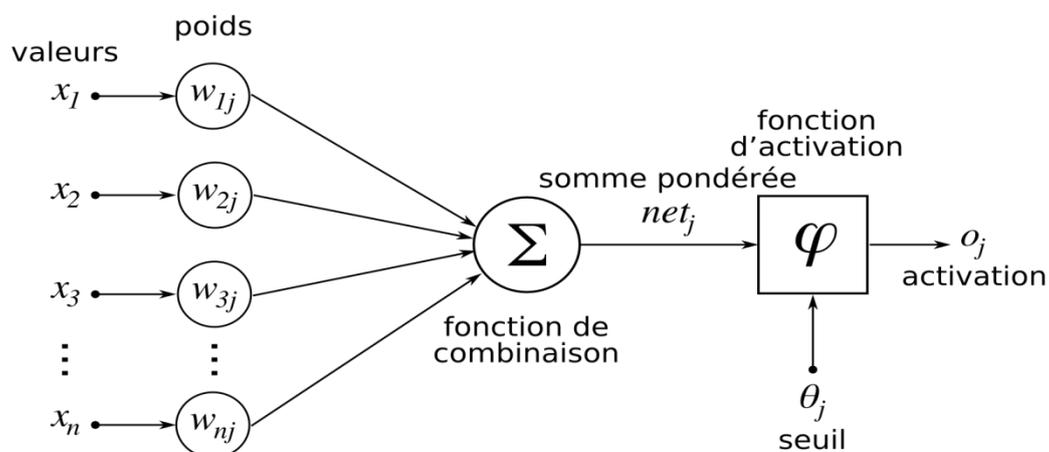


FIGURE 1.10 – Représentation d'un neurone formel

$$\varphi = \left(\sum_{j=0}^n w_j * x_j \right) - b$$

n est le nombre de données en entrée , $O = f(\theta)$

La somme pondérée étant calculée, une décision est ensuite prise pour calculer la sortie O en fonction du seuil θ : Si $\text{net} > \theta$ alors $O = +1$ sinon $O = -1$ [15] .

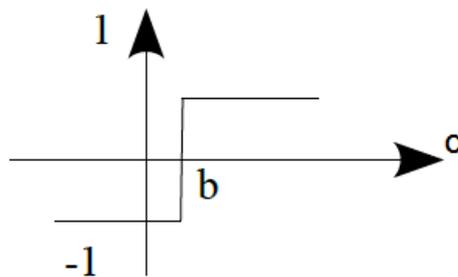


Figure 1.11 Fonction activation à seuil

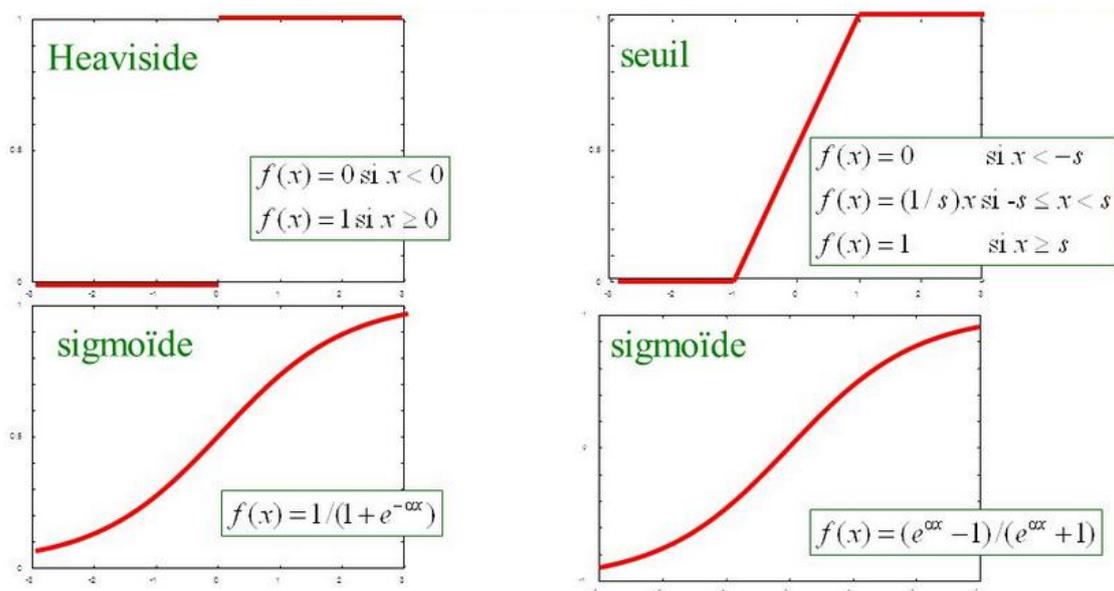


Figure 1.13 Les fonctions d'activation

1.11 Apprentissage :

1.11.1 POSITION D'UN PROBLEME DE CLASSIFICATION PAR RESEAU DE NEURONES

La conception d'un système de classification, qu'il soit un système de reconnaissance de formes conventionnel ou un réseau de neurones artificiels, nécessite les étapes suivantes :

a- ACQUISITION DES DONNÉES

Le choix des données initiales est très délicat car il conditionne en grande partie l'efficacité des performances du réseau de neurones. L'acquisition des données permet de convertir les données de leurs formes physiques (son, image, etc.) en une forme acceptable par l'ordinateur [7].

b- PRÉTRAITEMENT

Il consiste à localiser, segmenter et normaliser les représentations. Par exemple pour une image (en occurrence l'image d'une main), on cherchera à réduire le bruit et à augmenter le contraste [7].

c- LE CHOIX DES ATTRIBUTS

Le choix d'un vecteur d'attributs pour caractériser un objet peut se révéler délicat. En effet, il faut faire un compromis entre la dimension du vecteur et le contenu des informations. Un vecteur de taille trop petite conduirait à de mauvaises performances du réseau de neurones.

Avant le traitement des données nous avons donc à effectuer le choix des objets, la définition des attributs caractérisant les objets et la construction de la base d'apprentissage. A la fin de cette phase on obtient un tableau de nombres à deux entrées : les données et les attributs les caractérisant [17].

		<i>Les données</i>					
		X_1	X_2	...	X_d	...	X_D
<i>Les attributs</i>	1	X_{11}	X_{12}	...	X_{1d}	...	X_{1D}
	2	X_{21}	X_{22}	...	X_{2d}	...	X_{2D}

	n	X_{n1}	X_{n2}	...	X_{nd}	...	X_{nD}

	N	X_{N1}	X_{N2}	...	X_{Nd}	...	X_{ND}

tableau 1 Un tableau des entrées et des attributs

A la fin de ces étapes on obtient une base de connaissance qui sert comme une base d'apprentissage.

d- LA DECISION

Ayant une nouvelle observation, il s'agit de décider à quelle classe l'affecter. Pour que le système soit performant, il faut qu'il se comporte bien sur une base de données autre que sa base d'apprentissage, cette base s'appelle la base de généralisation [17].

e- APPRENTISSAGE

L'apprentissage est le cœur du réseau de neurones. Pendant cette phase il va falloir déterminer les variables du réseau que sont les différents poids. Une base d'apprentissage va permettre d'entraîner le réseau. L'apprentissage supervisé qui consiste à réduire progressivement la différence entre les valeurs de sortie que l'on souhaite obtenir et les valeurs de sortie du réseau. Cela se fait par modification progressive des poids synaptiques. Pour faire cela, il faut propager l'erreur constatée sur la couche de sortie et apporter des corrections à tous les poids du réseau. C'est l'algorithme de la rétro propagation du gradient de l'erreur qui est utilisé dans ce travail. Il s'agit de minimiser une fonction coût basée sur le signal de l'erreur obtenue par les différences entre les réponses désirées des neurones et les réponses actuelles de ces mêmes neurones.

L'algorithme de la rétro propagation consiste à calculer l'erreur entre la sortie attendue et la sortie calculée et à rétro propager l'erreur dans les couches précédentes.

L'apprentissage se termine:

- Quand le réseau détermine correctement les différentes classes.

- En fonction d'un nombre d'itérations.
- Quand l'erreur en sortie se stabilise [15]

Dans le cas de deux entrées x_1 et x_2 , le perceptron réalise une partition en 2 classes (a et b) selon la valeur de sortie (+1 ou -1). La séparation de ces 2 zones est effectuée par un hyperplan. L'équation de la droite séparatrice est $w_1 x_1 + w_2 x_2 - b = 0$

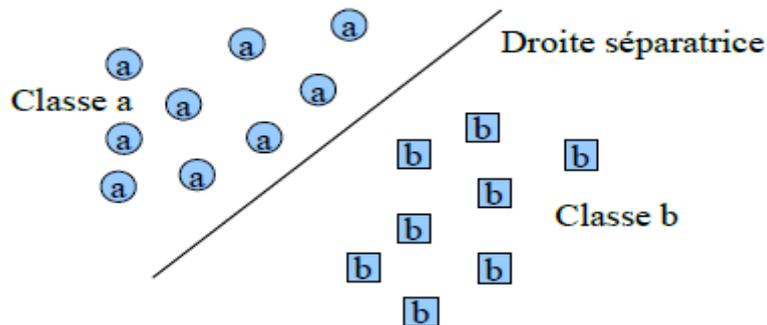


Figure 1.14 – Partitionnement

L'utilisation de cette méthode peut occasionner des blocages: une solution trouvée peut être un minimum local mais pas global. Le problème de l'efficacité de la méthode dépend de nombreux facteurs. Le nombre de couches et le nombre de neurones par couches dépend du problème étudié et de l'expérimentation. Il n'y a pas de règles précises mais des bonnes pratiques, par exemple avoir un nombre important d'exemples dans la base d'apprentissage pour faire de l'apprentissage avec des initialisations de variables différentes pour éviter les problèmes de minima locaux [15].

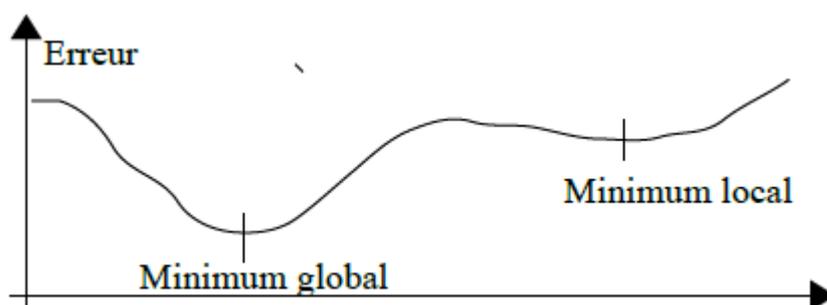


Figure 1.15 –rétro propagation de l'erreur

Il faut un minimum de neurones dans la ou les couches cachées pour avoir une discrimination suffisante entre les différentes classes. Mais par contre un nombre trop important de ces neurones peut engendrer un risque de sur apprentissage. Le réseau aura appris à classer parfaitement les données de la base d'apprentissage mais sera incapable de classer correctement de nouvelles données[15].

1.12 Les différents types d'apprentissage

Les méthodes d'apprentissage peuvent être classées en plusieurs catégories : apprentissage supervisé, apprentissage semi-supervisé, apprentissage non-supervisé et apprentissage par renforcement.

Nous allons nous positionner dans le cas d'une tâche de classification. C'est ce dont nous avons besoin dans le cadre de ma thèse pour réaliser la reconnaissance de geste. Le but est d'associer à un objet donné une étiquette correspondant sa classe [18].

A-Apprentissage supervisé L'apprentissage supervisé a pour but d'apprendre par l'exemple. Il faut fournir au préalable une liste d'objets avec leurs étiquettes de classe appelée ensemble d'apprentissage afin que le système soit capable d'expliquer et ensuite de prédire l'appartenance d'un nouvel objet à une classe connue à priori [18].

B-Apprentissage semi-supervisé

Ce terme regroupe des méthodes qui se situent entre l'apprentissage non-supervisé et l'apprentissage supervisé. Ce type de méthodes est utilisé quand un grand nombre de données est disponible mais sans qu'elles soient toutes étiquetées. L'initialisation de la méthode est faite à partir d'un petit jeu de données correctement étiquetées. Puis l'algorithme doit lui-même étiqueter les exemples suivants et construire son propre modèle.

Les algorithmes d'apprentissage non-supervisé et semi-supervisé sont beaucoup utilisés pour la recherche d'informations sur internet notamment. Ils permettent de traiter ainsi une grande quantité de données [18].

C-Apprentissage non-supervisé Pour l'apprentissage non-supervisé, l'ensemble d'apprentissage est seulement composé d'exemples, sans aucune étiquette de classe. C'est à l'algorithme de trouver des dépendances, des structures entre les différents exemples. Le «clustering» ou partitionnement de données regroupe un ensemble de méthodes d'apprentissage non-supervisé, comme l'algorithme des K-means. Les classes (ou « clusters » en anglais) sont créées par l'algorithme qui regroupe dans une même classe des objets ayant des caractéristiques communes entre elles et différentes avec les objets n'appartenant pas aux mêmes classes.

Prenons un exemple de classification par les K-means. L'algorithme est initialisé aléatoirement avec un certain nombre de clusters pour lesquels un point moyen, appelé centroïde, est évalué. A chaque itération, la distance entre chaque exemple aux différents centroïdes est calculée chaque exemple est alors associé au cluster dont la distance au centroïde est la plus proche. Puis les centroïdes sont réévalués. L'algorithme se termine lorsqu'il n'y a plus aucun changement. Un exemple simple de fonctionnement est montré sur la figure 1.16 [18].

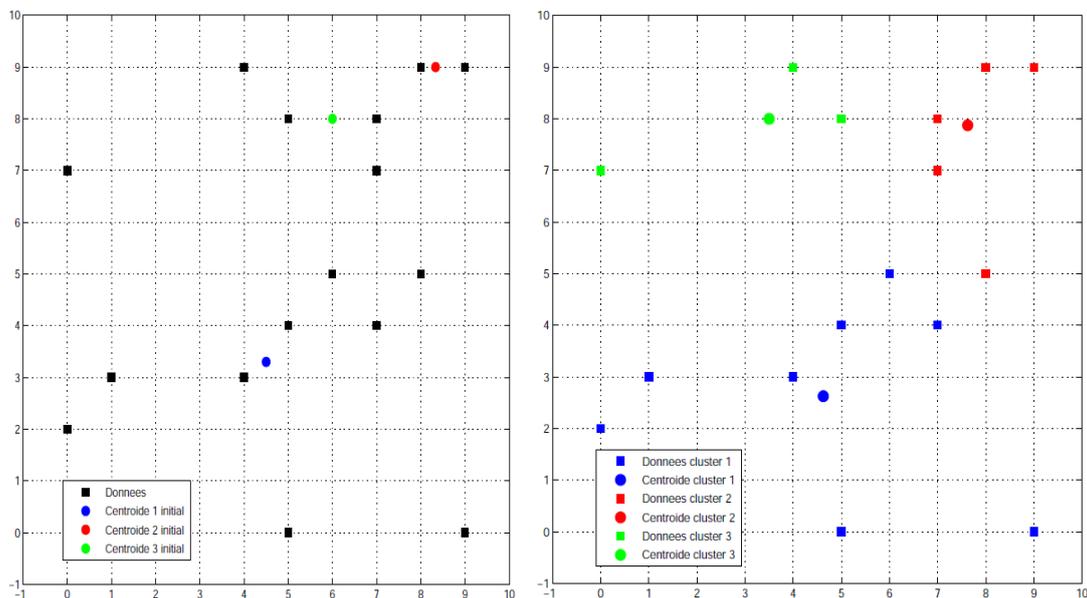


Figure 1.16 Apprentissage non-supervisé

D-Apprentissage par renforcement Ce type de méthode est un apprentissage interactif. A chaque décision que l'algorithme prend, il reçoit en retour des réponses de l'environnement

appelées signaux de renforcement. C'est un processus adaptatif qui améliore la solution en fonction des réponses qu'il reçoit. Ces algorithmes par renforcement permettent d'établir des processus plus complexes, comme le guidage de robots. Ce dernier a un objectif final à atteindre et en fonction des réponses de ses différents capteurs, il va pouvoir affiner ses actions [18].

1.13 Architecture d'un réseau de neurones

1.13.1 Le perceptron multicouche (PMC) :

Le PMC est un réseau composé de couches successives. Une couche est un ensemble de neurones n'ayant pas de connexion entre eux. Une couche d'entrée lit les signaux entrant, un neurone par entrée x_j , une couche en sortie fournit la réponse du système. Selon les auteurs, la couche d'entrée qui n'introduit aucune modification n'est pas comptabilisée. Une ou plusieurs couches cachées participent au transfert [08].

Dans un PMC, un neurone d'une couche cachée est connecté en entrée à chacun des neurones de la couche précédente et en sortie à chaque neurone de la couche suivante.

Un algorithme d'apprentissage tel que la rétro-propagation de gradient est appliqué pour ajuster les poids en réduisant au minimum une fonction d'erreur qui est, en général, la somme des erreurs quadratiques entre la sortie du réseau de neurones et la sortie attendue[9].(voir la figure 1.17).

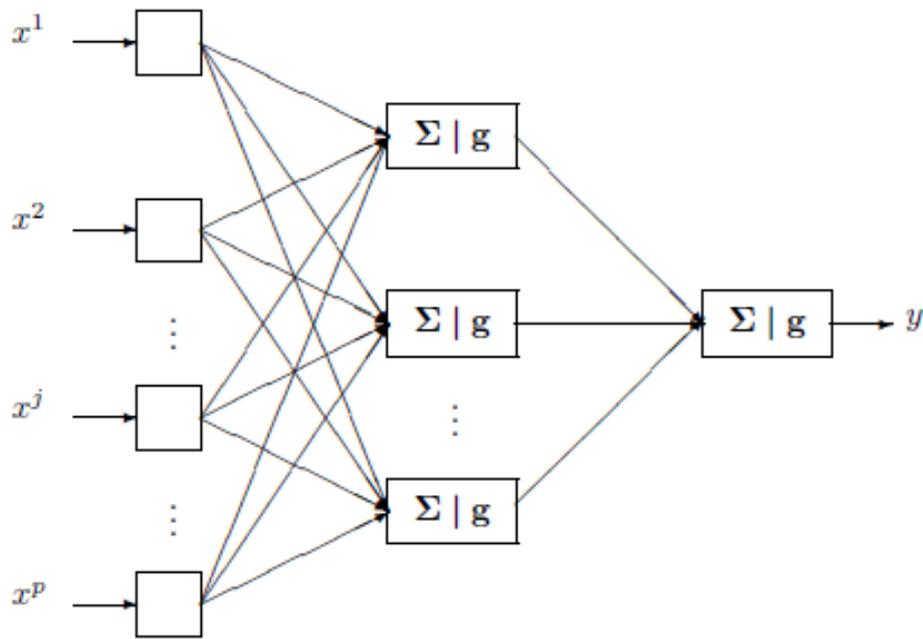


FIGURE 1.17 – Exemple de perceptron multicouche élémentaire avec une couche cachée et une couche de sortie.

- **Equation de propagation avant**

$$g = x_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_d$$

$$f(g) = \frac{1}{1 + e^{-g}}$$

$y = f(g)$ f est la fonction sigmoïde définie par :

1.13.2 LA RETRO-PROPAGATION PAR DESCENTE DE GRADIENT :

Cet algorithme permet de réaliser un apprentissage du réseau de neurones. On cherche à obtenir du réseau une réponse préétablie comme étant correcte. On dispose d'une base de connaissance de type entrée (p) / sortie attendue (d). On compare ensuite la sortie (a) à la sortie attendue. On introduit une fonction $e = d - a$ qui est la fonction d'erreur que l'on va chercher à minimiser en modifiant les poids du réseau. Une fois les poids définis par cet algorithme à partir d'exemples connus de l'utilisateur, on va chercher à extrapoler le réseau, en lui fournissant des entrées inconnues. Un exemple simple de fonctionnement est montré sur la figure 18 [20].

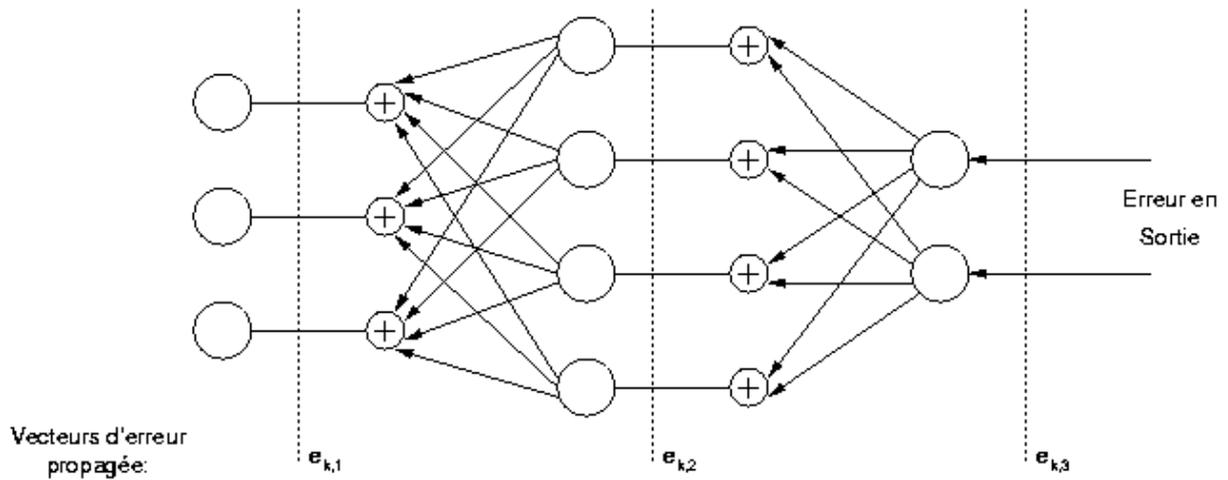


Figure 1.18 Principe de la rétropropagation

1.14 Conclusion

Dans ce chapitre, nous avons présentée sommairement les outils fondamentaux de prétraitement et visualisation des images. Les prétraitements d'images permettent d'améliorer la qualité de l'image.

les réseaux de neurones artificiels ressemblent donc au cerveau car non seulement, la connaissance est acquise au travers d'un apprentissage mais de plus, cette connaissance est stockée dans les connexions entre les entités, soit dans les poids synaptiques. Leur difficulté de mise en œuvre réside dans le choix du modèle (architecture) et le temps de traitement.

Ce chapitre est composé de 4 parties :

- Dans la première partie on fait une Généralités sur la robotique mobile
- La deuxième partie on fait une description mécanique du robot mobile.
- La troisième partie électronique du robot (carte de commande. carte de puissance....ex.)
- Dans la quatrième partie asservir des moteurs à courant continu avec un PID.

La Première Partie :

Généralités sur la robotique mobile

2.1.1 Introduction

La robotique est un ensemble de disciplines techniques (mécanique, électronique, automatique, informatique) articulées autour d'un objectif et d'un objet communs. Cet objectif est l'automatisation flexible de nombreux secteurs de l'activité humaine réputés jusqu'à très récemment comme ne pouvant se passer de la présence de l'homme, et l'objet est le robot, sorte de machine universelle dont l'homme rêve depuis toujours (mais qui n'a pas encore la forme de ce rêve !).[23]

Historiquement, Le terme "robot" vient du mot tchèque "Robota" qui signifie corvée ou travail forcé, utilisé pour la première en 1921 par Karel Capek dans sa pièce théâtrale R.U.R. : Rossum's Universal Robots.[24]

Les robots aujourd'hui ont un impact considérable sur de nombreux aspects de la vie moderne, de la fabrication industrielle aux soins de santé, le transport et l'exploration de l'espace et le profond de la mer.

En robotique, on distingue deux grands pôles d'intérêt, la robotique de manipulation et la robotique mobile. Les robots manipulateurs ont une base fixe on les retrouve dans les chaînes d'assemblage industrielles (peintures, soudage, emboutissage, etc.) contrairement aux robots mobiles qui peuvent se déplacer.

2.1.2 Définition

un robot est une machine capable de réaliser automatiquement des tâches diverses, de s'auto adapter à une situation nouvelle et d'agir sur son environnement.

Pour ce faire, il est doté principalement de capteurs lui permettant de percevoir l'environnement dans lequel il évolue, et d'effecteurs (roue, bras, pince, jambe, etc.) lui permettant d'agir sur cet environnement. [24]

2.1.3 les Types de robots

Il existe deux grandes familles de robots sont :

- Les robots manipulateurs.
- Les robots mobiles.

a)les robots manipulateurs :

Inspiré du bras humain, le bras manipulateur est un bras programmable munie de plusieurs articulations afin de permettre un mouvement en rotation ou en translation, Exemples de bras manipulateurs(voir les figures 2.1 ,2.2,2.3,2.4).



Figure 2.1: Robot de construction

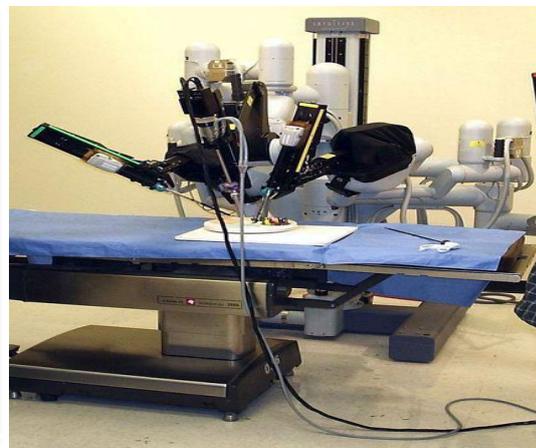


figure 2.2 : Robot da vinci



Figure 2.3 : Robot pompiste



figure 2.4 : Bras manipulateurs dans l'industrie

b) Les robots mobiles :

Un robot mobile est celui qui peut se déplacer dans son environnement de façon indépendante. Pour ce faire, le robot doit pouvoir naviguer, et la portée et la précision de navigation requise varie en fonction de la taille du robot et du type de sa tâche.[19]

Robots à roues :

C'est les plus répandus grâce à leur conception qui est la plus simple possible. Ces systèmes sont dotés de roues motrices pour permettre le déplacement. Ils peuvent être utilisés en environnements d'intérieur ou d'extérieur.

Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées.

Nous citerons ici les quatre classes principales de robots à roues.

Le robot mobile différentiel (differential drive) :

il comporte deux roues fixes non orientables, commandées indépendamment, une ou plusieurs roues folles sont ajoutées à l'avant ou à l'arrière du robot pour assurer sa stabilité.[24]

C'est un robot non-holonome, en effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion.

Ce type de robot est très répandu en raison de sa simplicité de construction et de ses propriétés cinématiques intéressantes, comme sa capacité à tourner sur lui même. N'empêche qu'il existe certains robots différentiels avec quatre roues commandées indépendamment mais ils sont toujours modélisés en deux roues. Les roues folles n'interviennent pas dans la cinématique.[24]

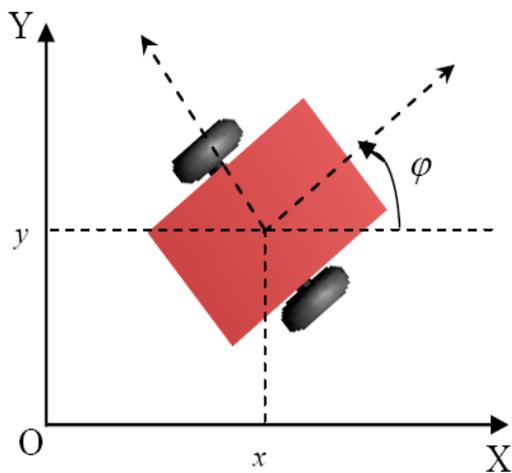


Figure 2.5 robot mobile différentiel



figure 2.6 ATRV2, CDTA

Modélisation :

Centre instantané de rotation: les roues motrices ayant même axe de rotation, le *CIR* du robot est un point de cet axe. Soit,

R : le rayon de courbure de la trajectoire du robot, i.e. la distance entre le *CIR* et le point de gravité O' .

$2L$: la distance qui sépare les deux roues.

Ω : la vitesse angulaire du robot par rapport au *CIR*.

v : vitesse linéaire du robot. v_r, v_g : les vitesses des roues droite et gauche.

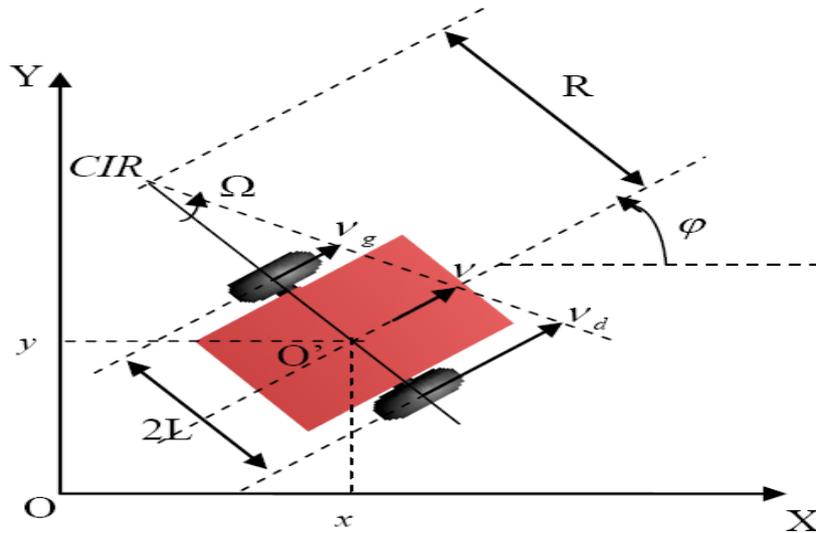


Figure 2.7 Modélisation du robot

Comportement du robot:

- si $vd = vg$ alors $\Omega = 0$ et $R = \infty$, le robot se déplace en ligne droite.
- si $vd = -vg$ alors $\Omega \neq 0$ et $R = 0$, le robot effectue une rotation sur lui-même.
- si $vd \neq vg$ alors le déplacement du robot est un virage à gauche ou à droite selon que vd soit supérieur ou inférieur à vg (dans une direction qui orrespond à la vitesse inférieur).

$$\Omega = \frac{v_d - v_g}{2L}$$

$$R = L \frac{v_d + v_g}{v_d - v_g}$$

Le robot mobile tricycle:

- Ce robot est constitué de deux roues fixes (passives) de même axe et d'une roue centrée orientable placée sur l'axe longitudinal du robot. Le mouvement est conféré au robot par deux actions: la vitesse longitudinale et l'orientation de la roue orientable.
- Ce type de robot possède les mêmes propriétés cinématiques que le robot de type bicycle sauf que celui-ci est constitué d'une seule roue fixe passive et une roue centrée orientable.

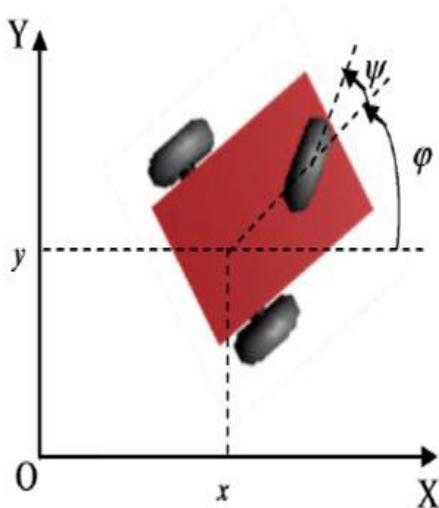


Figure 2.8 Robot de type tricycle

C'est un robot non-holonyme. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable. [24]

Le robot mobile de type voiture: Le cas du robot de type voiture est très similaire à celui du tricycle. La différence se situe au niveau du train avant, qui comporte deux roues au lieu d'une. (voir la figure 9).

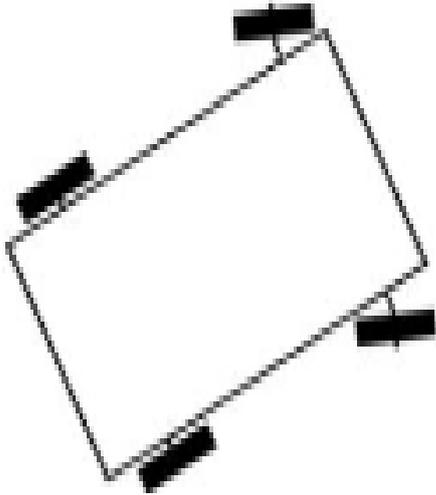


Figure 2.9 robot de type voiture

Le robot mobile omnidirectionnel :

- Un robot mobile est dit omnidirectionnel si l'on peut agir indépendamment sur les vitesses de translation selon les axes x et y , et la vitesse de rotation autour de z .
- Pour ce faire, ce robot est muni dans la plupart des cas d'un ensemble de roues décentrées orientables ou suédoises, disposées aux sommets d'un triangle équilatéral ou un carré.

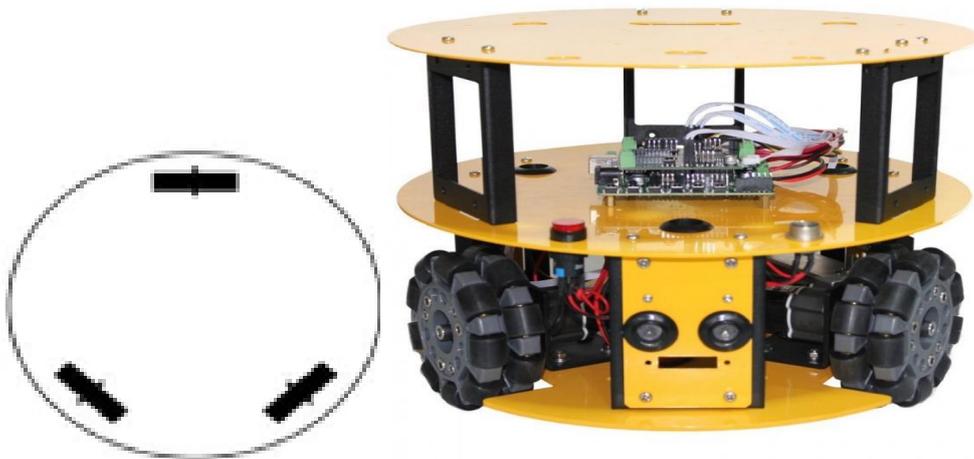


Figure 2.10 Robot de type omnidirectionnel

2.1.4 Comparaison des différents types des robots : Nous pouvons observer dans le tableau ci-dessous un récapitulatif des avantages et des inconvénients des différents types de robots à roues :

Type du robot	Avantage	Inconvénient
unicycle	<ul style="list-style-type: none"> * Stable * Rotation sur soi-même * Complexité mécanique faible 	<ul style="list-style-type: none"> * Non-holonome
Tricycle	<ul style="list-style-type: none"> * Complexité mécanique modérée 	<ul style="list-style-type: none"> * Non-holonome * Peu stable * Pas de rotation sur soi-même
Voiture	<ul style="list-style-type: none"> * Stable * Complexité mécanique modérée 	<ul style="list-style-type: none"> * Non-holonome * Pas de rotation sur soi-même
Omnidirectionnel	<ul style="list-style-type: none"> * Holonome * Stable * Rotation sur soi-même 	<ul style="list-style-type: none"> * Complexité mécanique importante

Tableau 2 Les avantages et des inconvénients des différents types de robots à roues

La deuxième partie :

Réalisation du robot (mécanique du robot mobile)

2.2.1 Introduction

Après avoir vue les composants et les différents types des robots, on entame maintenant la réalisation mécanique du robot. Dans ce partie nous allons détailler chaque partie mécanique du robot. Notre choix des composants repose essentiellement sur la disponibilité, et le cout, tout en respectant les objectifs principaux du projet.

2.2.2 Structure

La structure du robot est entièrement composée d'une plateforme en bois Elle support tous les composants du robot .(voir la figure 2.11).

Le robot possède un corps de hauteur 40 cm et de largeur 38 cm et longueur 27 cm avec trois roues, 2 roues en arrière et 1 roue en avant.



Figure 2.11 Le robot Wall-E

Le système est composé :

un capteur ultrason pour l'évitement des obstacles, et 2 encodeurs optiques intégrés dans le châssis pour la mesure de vitesse.

Le système d'exécution : 2 moteurs à courant continu.

2.2.3 Le châssis

Il reçoit tous les éléments constituant les systèmes mécaniques et électriques du robot à savoir les moteurs, les roues, la carte puissance

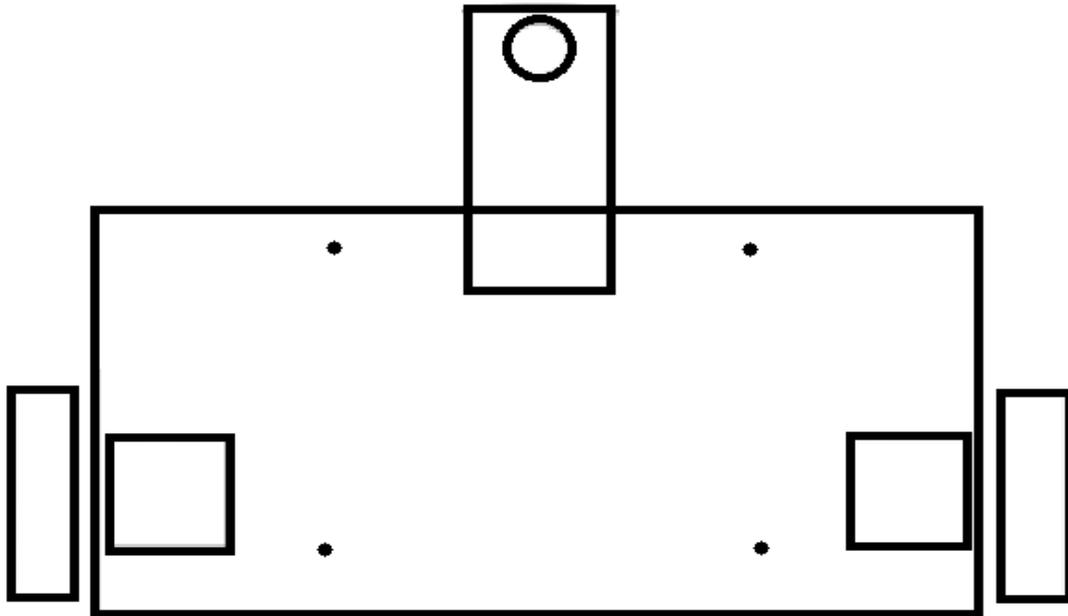


Figure 2.12 schéma du châssis

2.2.4 Les roue

Le robot possède trois roues, deux en arrière sont identiques, ont été fabriqués en Aluminium couvertes d'une couche Caoutchouc procurant une certaine adhérence et facilitant le déplacement et pour éviter le glissement. (voir la figure 2.13) , Et l'autre en avant pour réaliser l'équilibre du robot mobile(voir la figure 2.14).



Figure 2.13 Les Roues

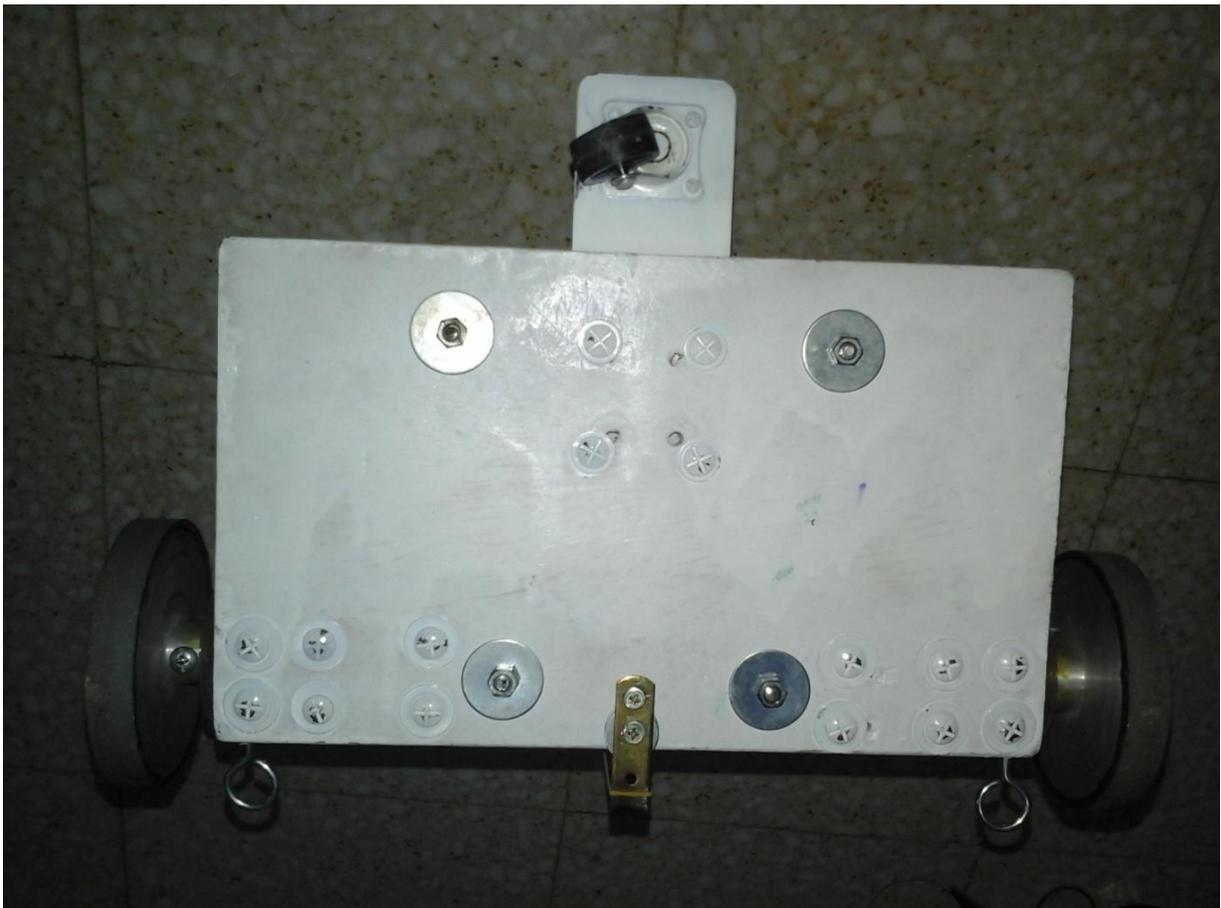


Figure2.14 Image de la base du robot

Les roues de l'arrière de diamètre 100mm et 16 mm de largeur, sont reliées au moteur par une bague d'accouplement (voir la figure 2.15).

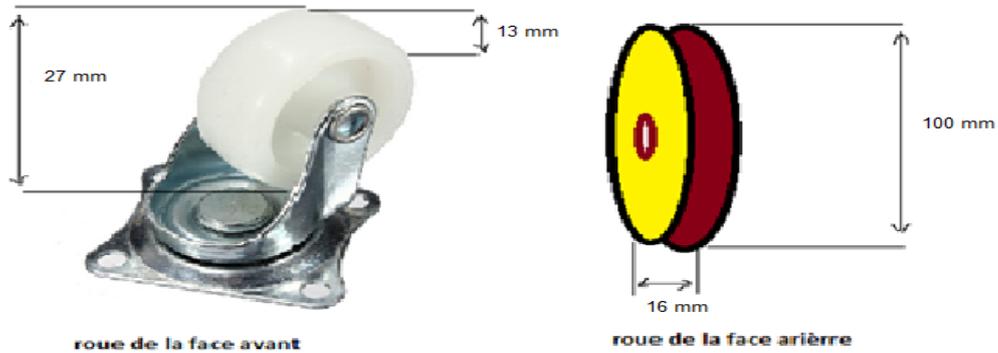


Figure 2.15 les deux types de roues de robot.

2.2.5 Les actionneurs

Dans notre projet on a utilisé:

- Moteurs à courant continu.

a) Moteurs CC (XD-37GB555) :

L' XD-37GB555 est un motoréducteur 12 Vcc. Il convient parfaitement aux applications de robotique.

Alimentation: 12 Vcc

Consommation: 1.25 A

Courant de blocage: 2,5 A

Vitesse de rotation: 100 t/min

Couple: 2.5 kg.cm

Dimensions: $\varnothing 37 \times 85$ mm (axe de 16 mm inclus)



Figure 2.16 Moteur à CC (XD-37GB555)

Les moteurs doivent être placés de manière précise dans le châssis pour assurer la mobilité (évités les handicaps du robot) .voir la figure 2.17

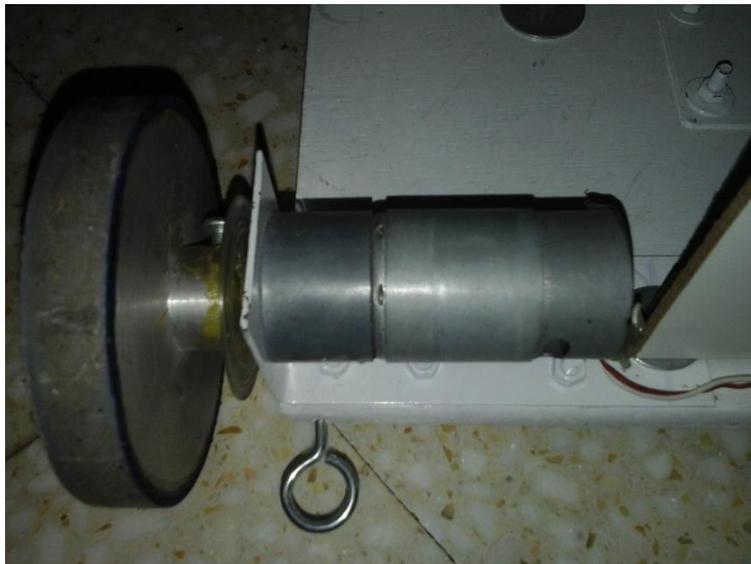


Figure 2.17 moteurs placés dans le châssis

2.2.6 Les Capteurs

a)Codeur rotatif :

Les codeurs rotatifs sont un type de capteurs permettant de délivrer une information d'angle, en mesurant la rotation effectuée autour d'un axe.L'information de vitesse peut alors être déduite de la variation de la position par rapport au temps. Plus le codeur rotatif tourne lentement, plus la déduction de vitesse perd en précision.(voir la figure 2.18).

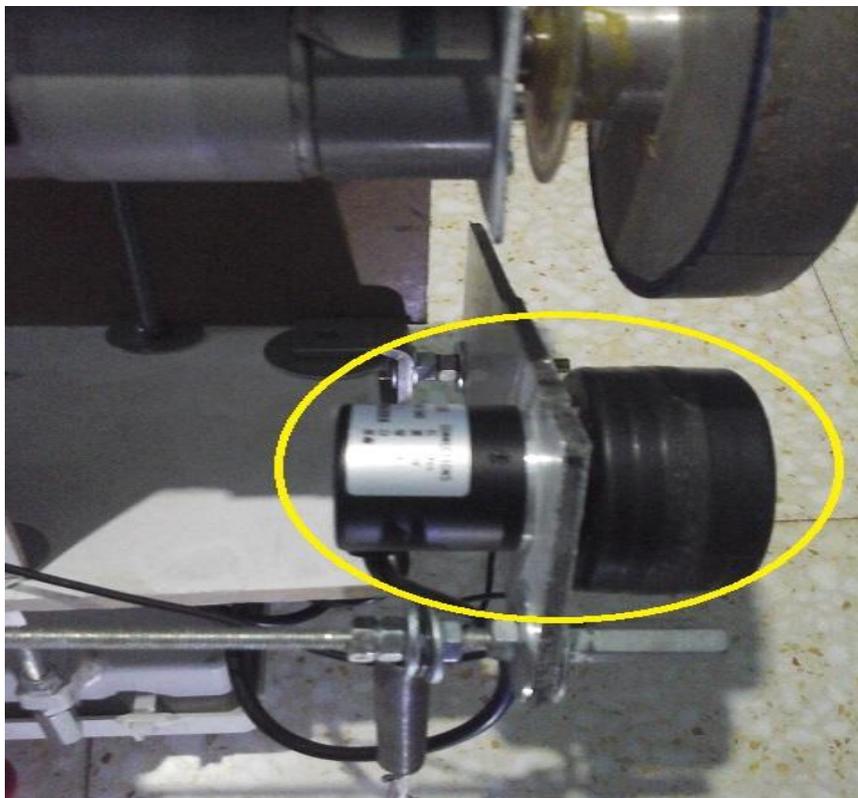


Figure 2.18 Codeur rotatif

2.2.7 Fabrication

Étape 1 : nous avons commencé par l'assemblage sur le châssis, les moteurs et les Les codeurs rotatifs que nous avons dû préalablement câbler. Dès lors, nous avons différencié les moteurs en nomment M1 pour le moteur gauche et M2 pour le moteur droite, c1 pour le codeur gauche et c2 pour le codeur droite. Les moteurs y sont boulonnés.

Étape 2 : nous avons vissé sur le châssis la roue libre et aux axes des moteurs les roues motrices. Cela permet d'apporter la notion de mobilité au robot.

La troisième partie :

Du côté électronique

2.3.1 Introduction

Dans la conception d'un robot, la partie électronique joue un rôle très important dans le fonctionnement du système.

Pour la mise en marche de tous les systèmes mécaniques, notre robot doit disposer de plusieurs cartes électroniques qui assurent son fonctionnement telles que la carte de puissance (L298), la carte de commande (Arduino méga 2560), ainsi que la camera et les autres capteurs .La structure électronique de notre robot mobile peut être représentée par la figure 2.19 .

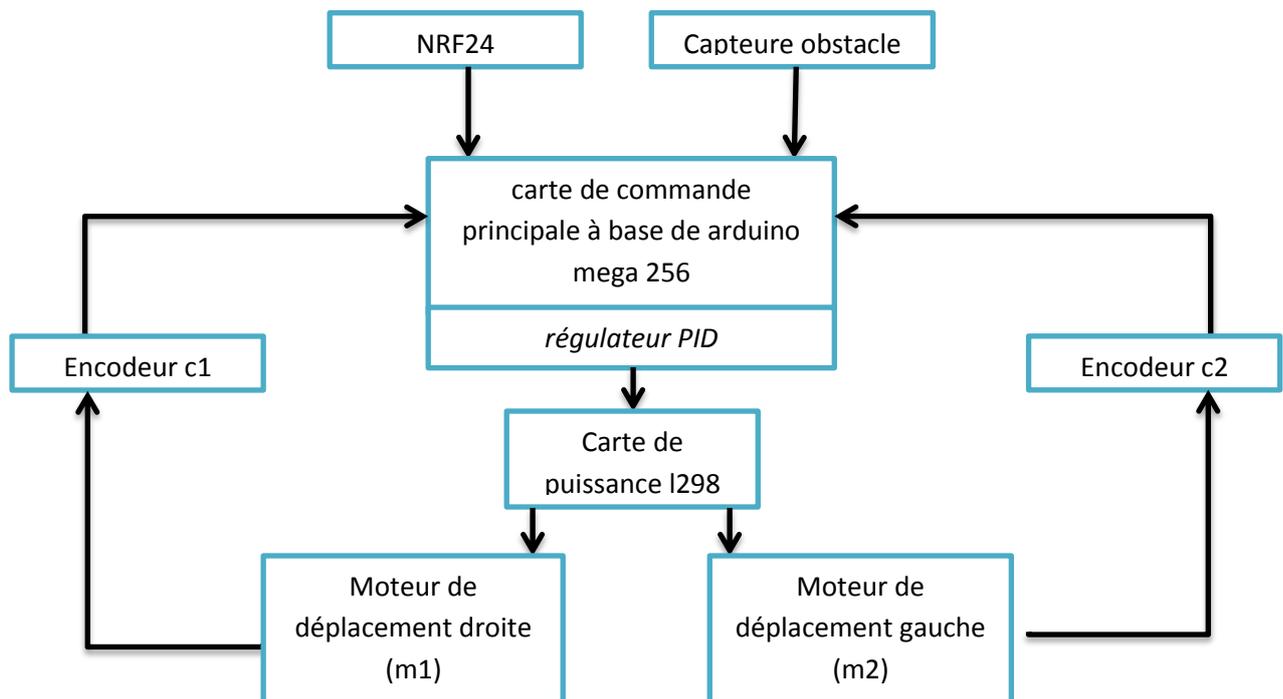


Figure 2.19 Schéma électronique de notre robot mobile

2.3.2 Les cartes électroniques programmables

Pour réaliser les opérations principales demandées, la mise en relation électrique des composants électroniques cités est importante. Pour cela l'implémentation des cartes électroniques est obligatoire.

a) La carte Arduino

L'Arduino est une plateforme de prototypage électronique open-source, basée d'une part sur du matériel et d'autre part sur un ensemble de logiciels faciles à utiliser. Destiné aux artistes, amateurs, designers, On peut l'associer à des capteurs (de lumière, de température, de position, etc.), des actionneurs (moteurs, pompe, etc.), des organes de sortie (lampe, chauffage, etc.), des circuits de puissance, une alimentation (piles, panneaux solaires, etc.), etc. Il existe toute une gamme de carte Arduino chacune ses caractéristiques, utilisées selon le besoin de l'utilisateur. Dans ce projet on s'intéresse à la carte Arduino Méga et Uno.



Figure 2.20 La carte Arduino Méga



Figure 2.21 La carte Arduino Uno

Les caractéristiques techniques des deux cartes UNO et Leonardo et MEGA :

Cartes Arduino	UNO R3 (classique & CMS)	UNO R3 Ethernet (classique & POE)	Leonardo	Mega 2560
Microcontrôleur	ATmega328P	ATmega328P	ATmega32u4	ATmega2560
Cadencement Horloge	16 MHz	16 MHz	16 MHz	16 MHz
Tension d'entrée	7 - 12V	7 - 12V	7 - 12V	7 - 12V
Tension de fonctionnement	5V	5V	5V	5V
Entrée/Sortie Numérique	14/6	14/4	20/7	54/15
Entrée-Sortie (PWM) Analogique	6/0	6/0	12/0	16/0
Mémoire vive (Flash)	32 Ko	32 Ko	32 Ko	256 Ko
Mémoire vive (SRAM)	2 Ko	2 Ko	2,5 Ko	8 Ko
Mémoire morte (EEPROM)	1 Ko	1 Ko	1 Ko	4 Ko
Interface USB	USB-B mâle	USB-B mâle	Micro-USB	USB-B mâle
Port UART	1	1	1	4
Carte SD	✗	✓	✗	✗
Ethernet	✗	✓	✗	✗
Wi-Fi	✗	✗	✗	✗
Dimensions	68x53mm	68x53mm	68x53mm	101x53mm

Tableau 3 Les caractéristiques techniques des carte Arduino Uno et Méga.

b) Le Raspberry Pi

Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM conçu par le créateur de jeux vidéo David Braben, dans le cadre de sa fondation Raspberry Pi.

Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique ; il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles. Il est fourni nu (carte mère seule, sans boîtier, alimentation, clavier, souris ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération [16].

Dans ce projet, on s'intéresse à la version Raspberry Pi 3 model B.



Figure 2.22 La carte Raspberry Pi 3 model B

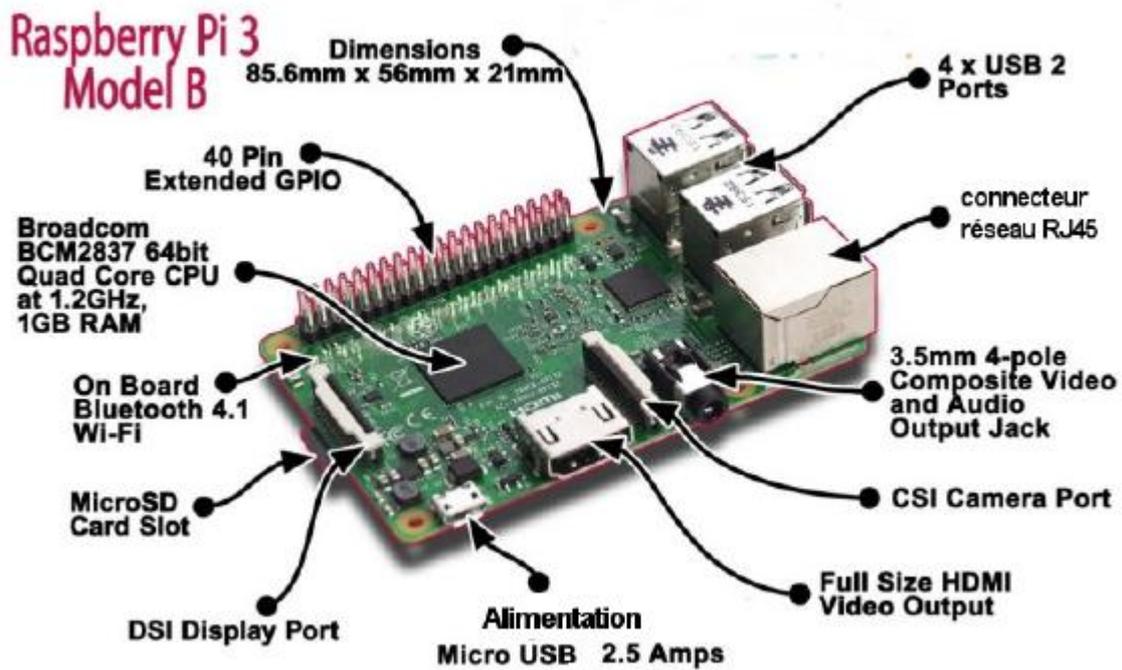


Figure 2.23 Les caractéristiques du Raspberry Pi 3 model B

2.3.3 Le module camera Raspberry Pi

Le module caméra Raspberry Pi est un produit officiel de la fondation Raspberry Pi. Le modèle a été lancé en 2013. Il se branche directement sur le connecteur CSI du Raspberry Pi.

Dans ce projet on utilise la version 1.3 qui est capable de livrer une image de résolution de 5MP ou un enregistrement vidéo HD 1080p à 30fps.

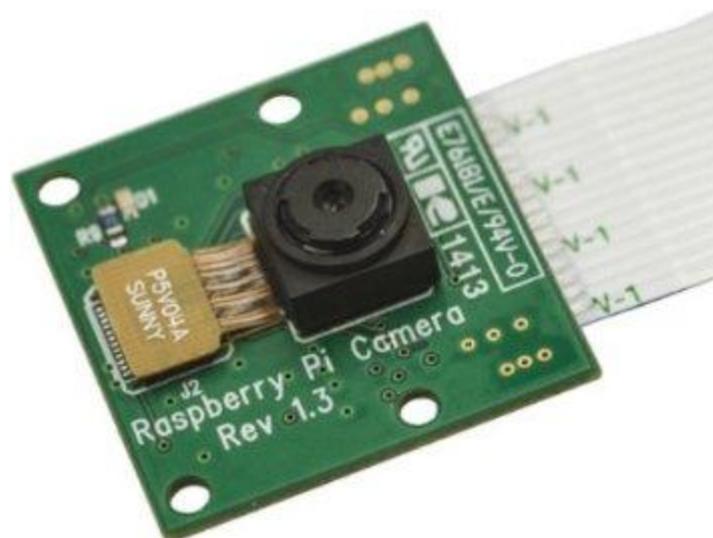


Figure 2.24 Le module camera Raspberry Pi V1.3.

Paramètres	Gamme
La taille et poids	<i>Environ 25 x 24 x 9mm, 3g</i>
Touche resolution	<i>5 Méga pixels</i>
Modes video	<i>1080p30, 720p60 et 640 x 480p60 / 90</i>
Capteur	<i>Omni Vision OV5647</i>
Résolution du capteur	<i>2592 x 1944</i>
Distance focale	<i>3,60 mm ± 0,01</i>
Champs de vision horizontal	<i>53,50° ± 0,13 °</i>
Champs de vision vertical	<i>41,41° ± 0,11 °</i>
Ratio focal (F-Stop)	<i>2,9</i>

Tableau 4 Les caractéristiques du module caméra Raspberry Pi v1.3

2.3.4 La carte puissance

Les cartes Arduino ont des limites à respecter : les sorties, tout comme les entrées, ne peuvent délivrer que 5V et 40mA chacune. Si on branche directement un appareil consommant plus sur une de ces sorties, on risque de griller la carte. Pour brancher des moteurs, il faut y ajouter une carte de puissance [figure 2.25]. Elle se branche sans soudure sur les pins, et nous permet de conduire deux moteurs à courant continu, en contrôlant la vitesse et la direction de chacun d'eux indépendamment.

Le composant central de la carte est un L298N, c'est un double pont en H, à haute tension et à courant élevé, il permet de faire tourner les moteurs dans un sens ou dans l'autre sans avoir à modifier les branchements.



figure 2.25 *La carte puissance*

2.3.5 Les capteurs à ultrasons

utilisent des ondes sonores dont les fréquences ne sont pas perceptibles par l'oreille humaine, généralement de 20 khz à 200 khz.

Ils mesurent le temps de vol d'une onde ultrasonore entre son émission et sa réception après réflexion. A partir de ce temps et connaissant la vitesse de propagation de l'onde, la distance séparant le capteur de l'obstacle perçu est déduite(voir la figure 2.26).

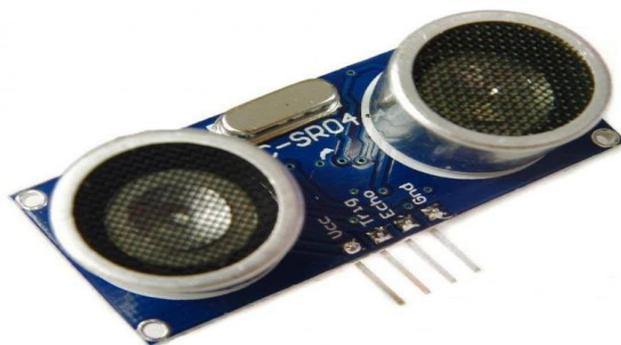


Figure 2.26 Le capteur HC-SR04

2.3.6 Le Périphériques sans fil NRF24

Le chipset NRF24L01 permet d'offrir à un montage électronique une couche de communication radio accessible via une interface SPI standard. La même puce permet de recevoir ou d'émettre, peut fonctionner sur plus d'une centaine de canaux.

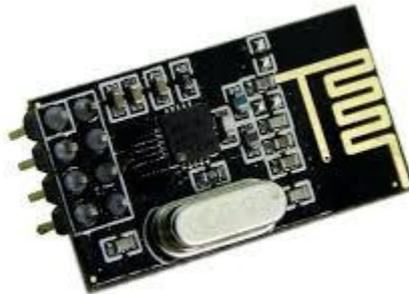


figure 2.28 Le chipset NRF24L01

a) Utilisations du chipset :

De par ses caractéristiques, ce chipset se retrouve dans de nombreuses applications embarquées. Il équipe des capteurs d'équipements de sports des jouets radio-guidés, des télécommandes (cardiofréquencesmètres...), des multimédias (lecteurs blu-ray...), des claviers et souris sans fils, etc.

b) Format des paquets

Les puces NRF24L01 modulent en GFSK sur des canaux démarrant à 2.4GHz et s'étalant sur 1MHz ou 2MHz. Sur ces canaux, deux protocoles sont utilisables : * ShockBurstTM * et * Enhanced ShockBurstTM* [13].

La Quatrième Partie :

Asservir Des Moteurs A Courant Continu Avec Un PID

2.4.1 Introduction

Pour contrôler un robot , il faut souvent passer par une étape d'asservissement et de régulation. La régulation PID (Proportionnelle Intégrale Dérivée) est une méthode efficace pour que votre robot de rouler droit(voir la figure 2.29).

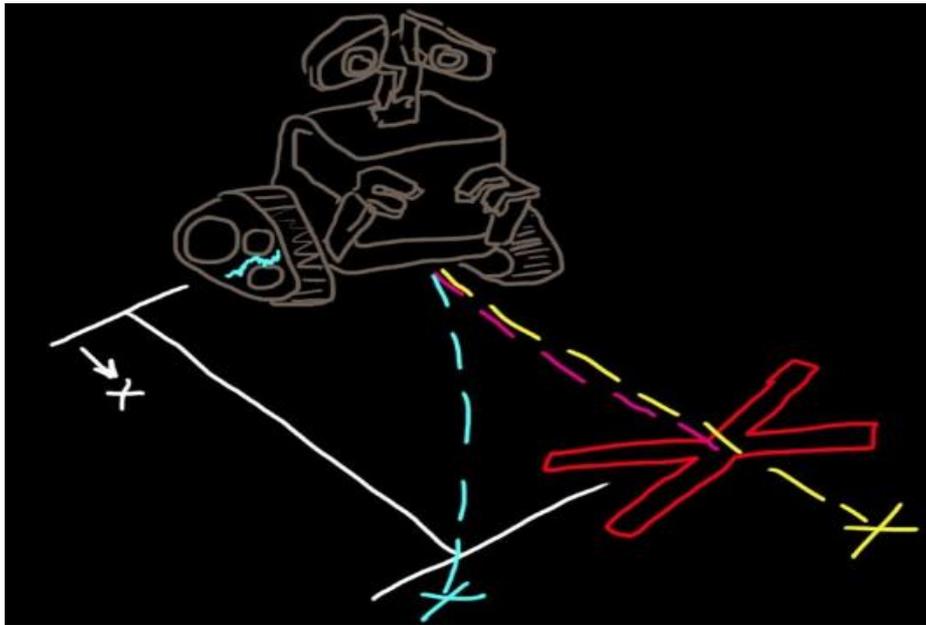


Figure 2.29 parcours optimale pour le robot mobile

2.4.2 Asservissement PID

Le rôle d'un algorithme de correction PID est d'améliorer 3 principales caractéristiques d'un système : la rapidité, la précision et la stabilité.

Pour un moteur par exemple, ces éléments se traduisent par une montée en régime rapide, une vitesse réelle très proche de celle demandée et un fonctionnement à la vitesse de consigne sans oscillations (accélérations, décélération...).

le PID sert à réduire l'écart qu'il y a entre une valeur de consigne et la valeur mesurée. Dans notre exemple : l'erreur entre la vitesse effective d'un moteur et la consigne qu'il lui a été donnée.

Pour fonctionner le PID à donc besoin de plusieurs choses : un actionneur à asservir, un capteur pour mesurer la grandeur voulue et un organe de contrôle (microcontrôleur par exemple) capable de faire les calculs et d'interpréter les informations de consigne et du capteur.[10]

2.4.3 Régler les coefficients d'un PID

Le réglage des coefficients K_p , K_i et K_d d'un PID peut se faire expérimentalement par essais/erreurs. Tout d'abord, sachez qu'il ne sert à rien de vouloir régler les trois coefficients en même temps ! Il y a trop de combinaisons possibles et trouver un triplet performant relèverait de l'exploit. Il vaut mieux y aller par étape.

- Tout d'abord, il faut mettre en place un simple régulateur proportionnel (les coefficients K_i et K_d sont donc nuls). Par essais/erreurs, il faut régler le coefficient K_p afin d'améliorer le temps de réponse du système. C'est-à-dire qu'il faut trouver un K_p qui permette au système de se rapprocher très vite de la consigne tout en faisant attention de garder la stabilité du système : il ne faut pas que le système réponde très vite tout en oscillant beaucoup !
- Une fois ce coefficient réglé, on peut passer au coefficient K_i . Celui-là va permettre d'annuler l'erreur finale du système afin que celui-ci respecte exactement la consigne. Il faut donc régler K_i pour avoir une réponse exacte en peu de temps tout en essayant de minimiser les oscillations apportées par l'intégrateur !
- Enfin, on peut passer au dernier coefficient K_d qui permet de rendre le système plus stable. Son réglage permet donc de diminuer les oscillations.[25]

En général, pour régler ces coefficients, on donne au système une consigne fixe (exemple : pour un moteur : tourne à 3 tours par seconde) et on observe la réponse du système (exemple : l'évolution du nombre de tours par seconde du moteur au cours du temps). Le graphe résultant (voir la figure 2.30).

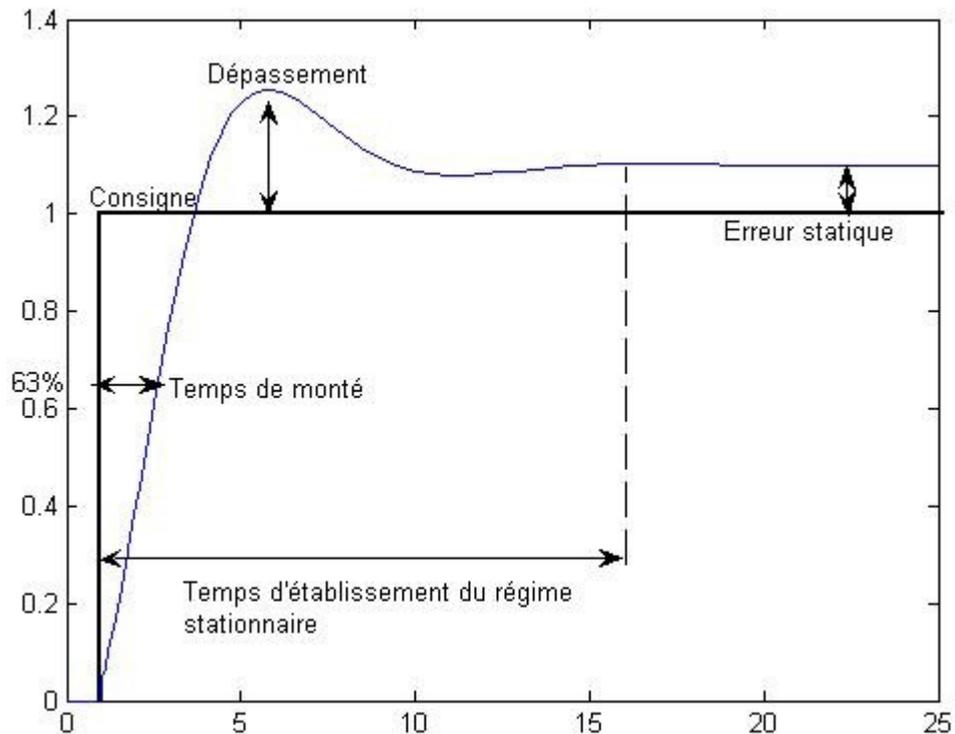


Figure 2.30 l'évolution du nombre de tours par seconde du moteur au cours du temps

2.5 Conclusion : ce chapitre nous permet d'évoquer et aborder quelques notions de base sur la robotique et quelques généralités tels que la mécanique et l'électronique, moyens d'exécutions et de mouvement, classes des robots mobile ainsi que les avantages et les inconvénients de chaque structure. Mais Le robot n'est qu'un tas de fer, une simple boite avec un moteur qui marche n'importe ou n'importe comment si on ne lui offre pas la possibilité de comprendre et d'agir avec le monde extérieur mais comment faire cela sachant que la machine ne comprend que le langage logique (langage binaire 1 & 0) ?

Sur le plan humain le projet tuteuré, concernant le robot mobile, nous a permis de travailler en équipe et d'apprendre comment concevoir, fabriquer et développer un robot à partir de zéro. C'est une expérience enrichissante et valorisante pour la gestion de projet et un travail intense dans plusieurs domaines.

Le prochain chapitre en va porter sur l'étude du coté reconnaissance et analyse du geste de la main et traduction de ce dernier à partir d'un réseau de neurones tout en utilisant un traitement d'image.

Réalisation et implémentation

3.1 Introduction

Pour que la machine puisse comprendre les ordres donnés par le geste humain, nous allons utiliser un traducteur, compilateur ou interpréteur qui va analyser, traduire et transcrire les gestes en langage binaire qui sera compréhensible par la machine.

Les gestes utilisés seront les gestes de mains, un geste universel utilisé par n'importe quel humain même les sourds-muets, comme exemple le langage LSF (Langue des signes française) (voir la figure 3.1).



Figure 3.1 La Langue des Signes Française (LSF)

LSF : est une langue visuelle et gestuelle qui s'exprime uniquement avec des mouvements du corps et des expressions du visage. Il existe deux types de gestes, le geste statique et le geste dynamique :

- **Geste statique** : l'information que porte le geste statique, est exprimée par une seule posture, cette méthode a été utilisée dans notre projet.
- **Geste dynamique** : contrairement au geste statique, l'information est exprimée par deux caractéristiques, qui sont la configuration de la main et sa position, ce qui correspond à un mouvement de la main dans le temps. Ce type est le plus difficile à traiter (voir la figures 3.2 et 3.3).



Figure 3.2 Geste dynamique



Figure 3.3 Geste Statique

Nous avons mis en place un système par lequel l'utilisateur peut donner des commandes à un robot sans fil en utilisant des gestes. Grâce à ce dernier, l'utilisateur peut contrôler ou naviguer dans le robot en utilisant des gestes de sa main, interagissant avec le système robotique. Les signaux de commande sont générés à partir de ces gestes en utilisant le traitement d'image. Ces signaux sont ensuite transmis au robot pour le faire déplacer dans les directions spécifiées.

3.2 Système proposé

Nous proposons un système, à l'aide duquel l'utilisateur peut naviguer dans le robot sans fil dans l'environnement en utilisant diverses commandes de gestes. L'objectif principal est de fournir une technique fiable et plus naturelle.

Dans ce système, l'utilisateur fait fonctionner le robot à partir d'une station de contrôle rpi3. la caméra est utilisée pour capturer le flux vidéo en temps réel des gestes de la main pour générer des commandes pour le robot.

En utilisant ses gestes capturés, le robot est déplacé dans toutes les directions possibles dans l'environnement en utilisant quatre types de commandes possibles: avant, arrière, droite et gauche.

A partir de cette commande gestuelle générée, un signal est généré pour transmettre la commande donnée au robot.

Le signal généré transmet de la station de contrôle au robot. le RF reçoit la commande de la station de contrôle, il est transmis au microcontrôleur Arduino. L'Arduino prend ce signal en entrée de la RF et génère certains signaux de sortie qui sont transmis au pilote du moteur. Cette génération de signal de sortie dépend de l'entrée gestuelle, pour chaque entrée de geste possible on trouve une commande, un signal de sortie différent est généré. Le pilote du moteur est utilisé pour commander les moteurs CC du robot. Il prend les signaux numériques comme l'entrée de l'Arduino et donne ces signaux comme une sortie aux moteurs CC. Une fois qu'un signal de commande est donné au robot, il continue à se déplacer dans cette direction jusqu'à ce que la commande suivante soit donnée ou qu'un obstacle arrive sur le chemin (voir la figure 3.4).

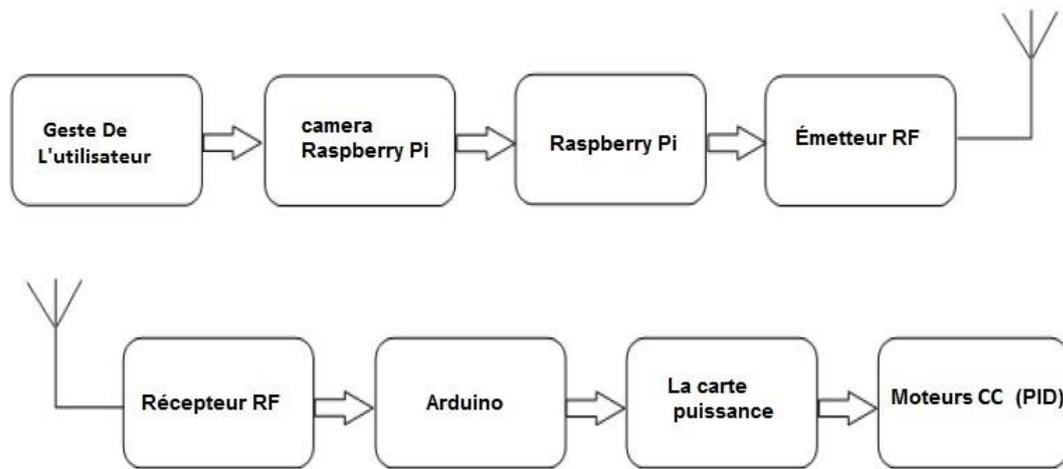


Figure 3.4 schéma synoptique du système proposé

3.3 Problèmes rencontrés

Tout au long de la réalisation de ce projet, nous allons être confrontés à différents problèmes Tels que :

- L'acquisition du geste
- L'éclairage
- La nature du geste
- La complexité de la scène
- Temps réel

3.4 TECHNOLOGIES UTILISÉES

Durant le processus de réalisation, nous avons utilisé plusieurs outils de développement technologique dans le coté software (programmation) et coté hardware tel que le langage Python, et la bibliothèque open CV.

3.4.1 Software (Programmation)

a)Python :

Dans ce projet, nous avons utilisé le langage python avec sa bibliothèque Open CV. Python est un langage de programmation très puissant qui est de plus en plus utilisé pour

développer des applications rapide et en même temps efficaces. Il est très utilisé dans le monde scientifique pour sa syntaxe aérée et une vitesse d'exécution rapide.

Ce langage est aussi excellent pour la création de prototypes car sa simplicité permet d'implémenter un projet aussi vite que l'on y réfléchit.

Caractéristique :

- Portable : (fonctionne dans n'importe quel Système d'exploitation).
- Gratuit : Un open source très puissant qui permet d'offrir ses bibliothèques gratuitement.
- Une syntaxe simple.
- Python a des modules qui peuvent être étendu en C.

Ces points nous ont permis de bien choisir notre langage de programmation.

b)OpenCV :

Open CV (Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel.

Caractéristique :

La bibliothèque Open CV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

La bibliothèque OPENCV sera utilisée dans notre projet pour détecter, suivre la main en mouvement, et la reconnaissance geste en utilisant le module camera du Raspberry pi.

Elle propose la plupart des opérations classiques en traitement bas niveau des images:

- lecture, écriture et affichage d'une image
- calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs
- lissage, filtrage
- seuillage d'image (méthode d'Otsu, seuillage adaptatif)
- segmentation (composantes connexes, GrabCut)
- morphologie mathématique.

3.4.2 Hardware

- Raspberry Pi
- Arduino_

Répartition des tâches :

Sachant les caractéristiques (puissance et robustesse) de chacune des cartes Arduino et Raspberry Pi, il est impératif de bien répartir les tâches entre ces deux dernières :

- **La tâche de l'Arduino :**

L'Arduino est adaptée pour un travail répété, alors ce dernier va être le cerveau du robot ou il va contrôler son mouvement.

- **La tâche de la Raspberry Pi :**

Contrairement à l'Arduino, la Raspberry Pi est adaptée à un travail plus complexe et plus varié.

Son travail consiste à capturer le geste de main à partir d'une caméra, l'analyser avec un traitement d'image, le traduire en binaire, accomplir une classification à partir d'un traitement par réseau de neurones, et l'envoyer par signal à la carte Arduino pour accomplir la tâche (Voir la figure 3.4).

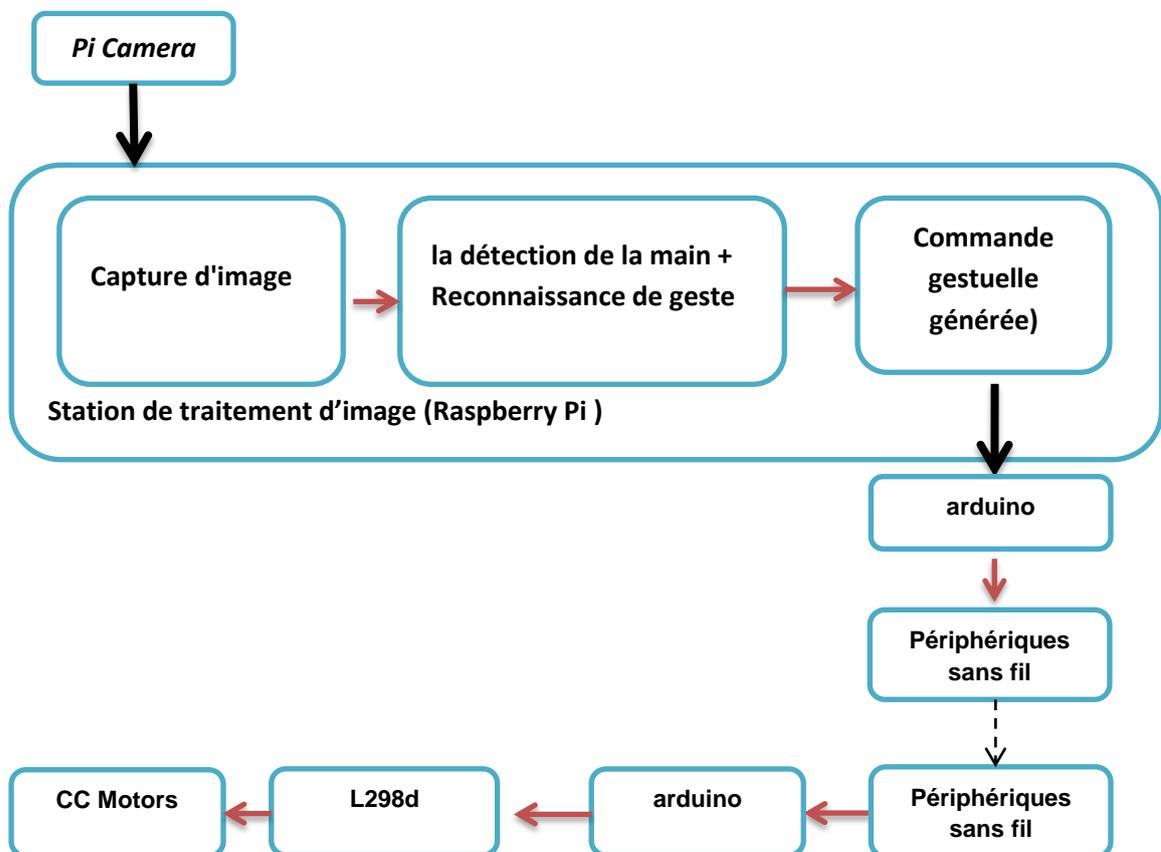


Figure 3.5 schéma détaillé du système proposé

3.5 La détection de la main

La première étape dans la reconnaissance des gestes est la détection de la main. Ce projet de recherche a été basé sur la technologie du détecteur [Viola-Jones] pour la reconnaissance des gestes de la main. Cependant, ce dernier a présenté des problèmes de performance pendant la détection où les échantillons ont été influencés par le bruit de fond d'entraînement.

A)La méthode de Haar : détection d'objet

La méthode de Viola et Jones est une méthode de détection d'objet dans une image numérique, elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures ou des avions (dans notre cas le type d'objet est la main). La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes.

En tant que procédé d'apprentissage supervisé, la méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, pour entraîner un classifieur. Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles.

Considérée comme étant l'une des plus importantes méthodes de détection d'objet, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises ensuite par de nombreux chercheurs en vision par ordinateur, à l'exemple de la notion d'image intégrale ou de la méthode de classification construite comme une cascade de classifieurs boostés. Cette méthode bénéficie d'une implémentation sous licence BSD dans OpenCV, la bibliothèque utilisée dans notre application[11].

b)Principe de la méthode :

La méthode de Viola et Jones consiste à balayer une image à l'aide d'une fenêtre de détection de taille initiale 24x24pixels (dans l'algorithme original) et de déterminer si un visage est présent. Lorsque l'image a été parcourue entièrement, la taille de la fenêtre est augmentée et le balayage recommence, jusqu'à ce que la fenêtre fasse la taille de l'image. L'augmentation de la taille de la fenêtre se fait par un facteur multiplicatif de 1.25. Le balayage, quant à lui, consiste simplement à décaler la fenêtre d'un pixel. Ce décalage peut être changé afin d'accélérer le processus, mais un décalage d'un pixel assure une précision maximale.

Ce qui différencie cette méthode des autres est notamment :

- 1- l'utilisation d'images intégrales qui permettent de calculer plus rapidement les caractéristiques.
- 2- La sélection par boosting des caractéristiques.
- 3-La combinaison en cascade de classifieurs boostés, apportant un net gain de temps d'exécution[12].

c)Apprentissage du classifieur :

L'apprentissage du classifieur est une étape préliminaire et très importante. Il s'agit d'entraîner le classifieur afin de le sensibiliser à ce que l'on veut détecter, dans ce cas c'est les mains.

Pour cela, nous avons mis à sa disposition un ensemble d'image pour les traiter sur deux situations :

La première où une énorme quantité de cas positifs lui sont présentés (Voir la figure 3.8),et la deuxième une autre énorme quantité de cas négatifs lui sont présentés(Voir la figure 3.9), et tout cela est présenter sous la forme d'un fichier XML.

d) Les caractéristiques :

Une caractéristique est une représentation synthétique et informative, calculée à partir des valeurs des pixels. Les caractéristiques utilisées ici sont les caractéristiques pseudo-haar (Voir la figure 3.7). Elle sont calculées par la différence des sommes de pixels de deux ou plusieurs zones rectangulaires adjacentes.

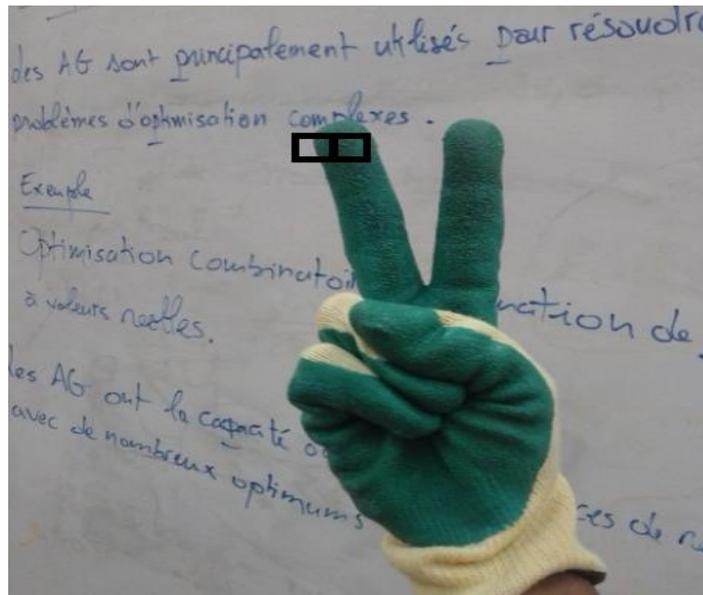


Figure 3.6 Passage d'une des caractéristiques sur la main pendant l'apprentissage

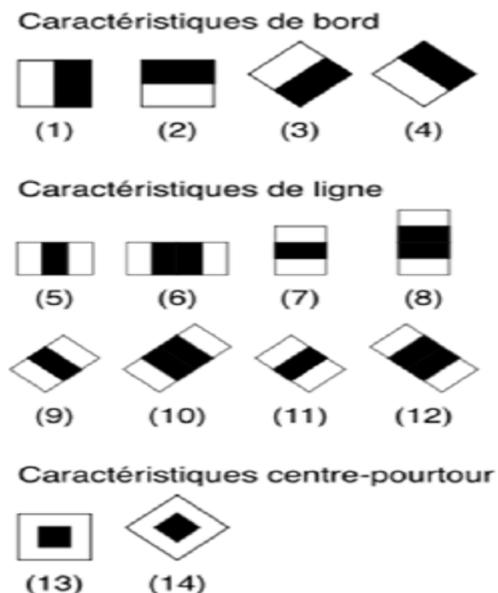


Figure 3.7 Caractéristiques pseudo-haar avec différentes orientations.

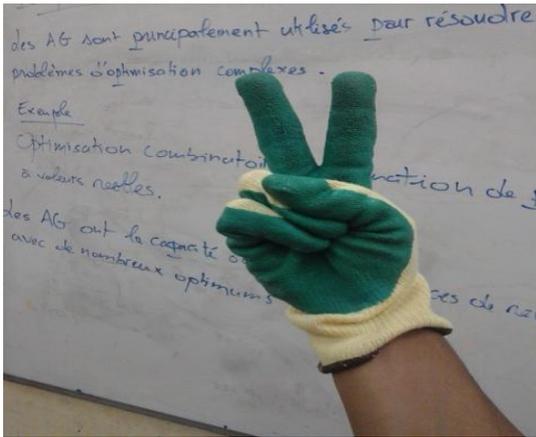


Figure 3.8 images positif





Figure 3.9 images négative

e) Implémentation de la méthode de détection de la main “ Viola & Jones ”

Tout d’abord il faut créer le classificateur en utilisant le Haartraining dans OpenCV. Pour utilise la commande Haartraining, nous devons mettre en place beaucoup d’images négatives, plus d’image de fond, et le meilleur résultat. Une fois le processus terminé, le Haartraining crée un fichier .xml (Voir la figure 3.10).



Figure 3.10 Test de la méthode de détection de la main [Viola & Jones]

3.6 Système de reconnaissance de gestes

Le processus de reconnaissance de formes est proche du raisonnement de l'être humain, grâce à l'intelligence artificielle, qui va servir à automatiser les tâches humaines.

Notre système de reconnaissance de geste est schématisé dans la figure 3.11, résumant les étapes qui seront détaillées par la suite pour l'identification du geste.

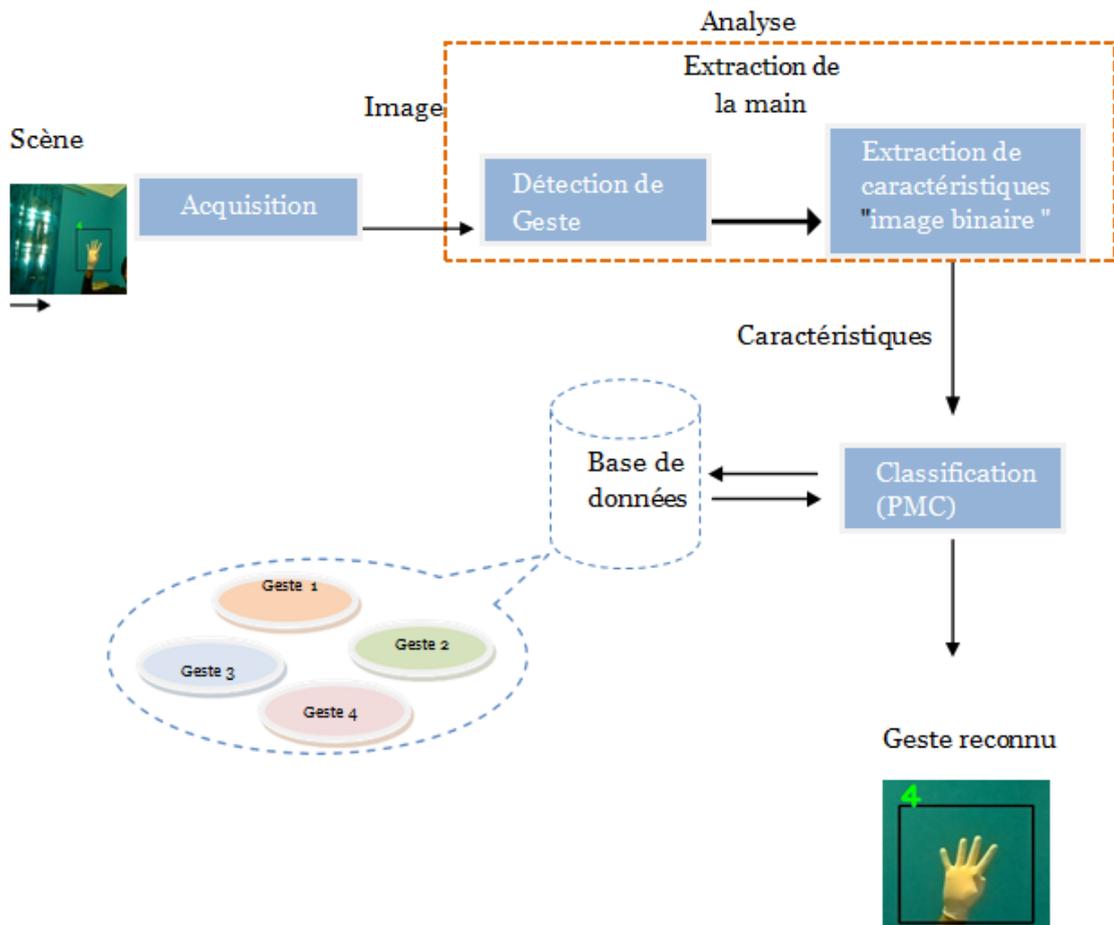


Figure 3.11 Système de reconnaissance de gestes

3.6.1 Reconnaissance de geste de la main : Approche Réseau Perceptron Multicouche

A) Base utilisée

Cette méthode repose sur l'apprentissage d'un réseau de neurones discriminant. On utilise un seul réseau ,et on présente à l'entrée tous les pixels de l'image test et on aura en sortie le geste correspondant.

les valeurs des pixels doivent être normalisées avant de les regrouper par base de data geste dans des fichiers différents grâce au programme de construction, et qui comprend par la suite les 4400 gestes fournis, et avec pour chaque geste 1000 images différentes. Choisissent ensuite pour chacun des gestes une base d'apprentissage de 1000 images et une base de test de 100 images, qui seront prises aléatoirement parmi toutes les images disponibles. Quelques échantillons d'images de la base de données sont représenté par les figures (3.12 , 3.13 ,3.14 ,3.15)

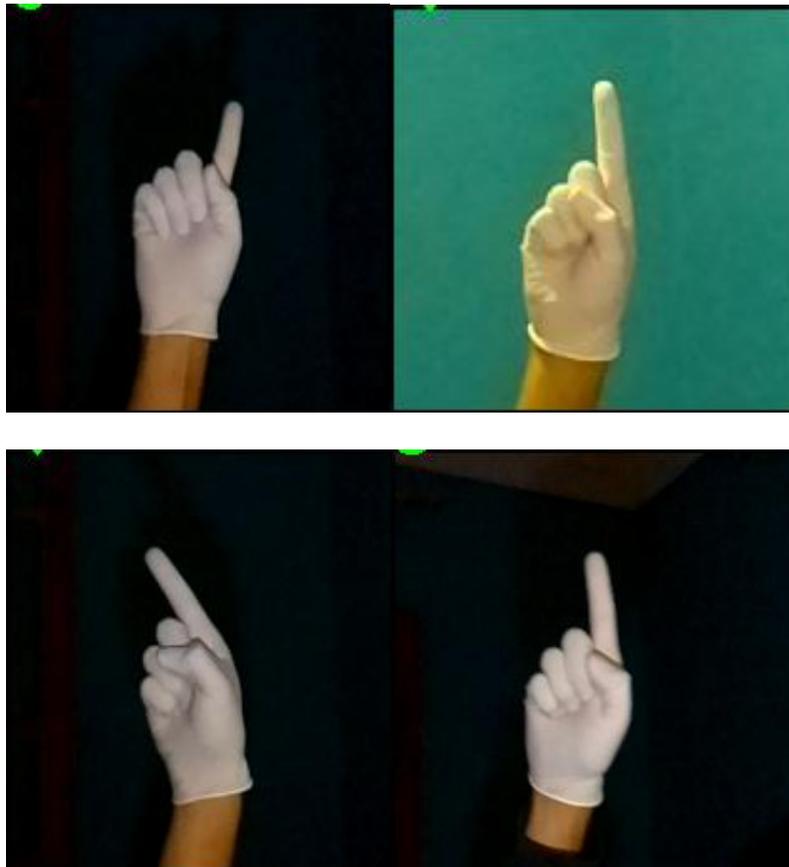


Figure 3.12 Geste 1 (Quelques échantillons)

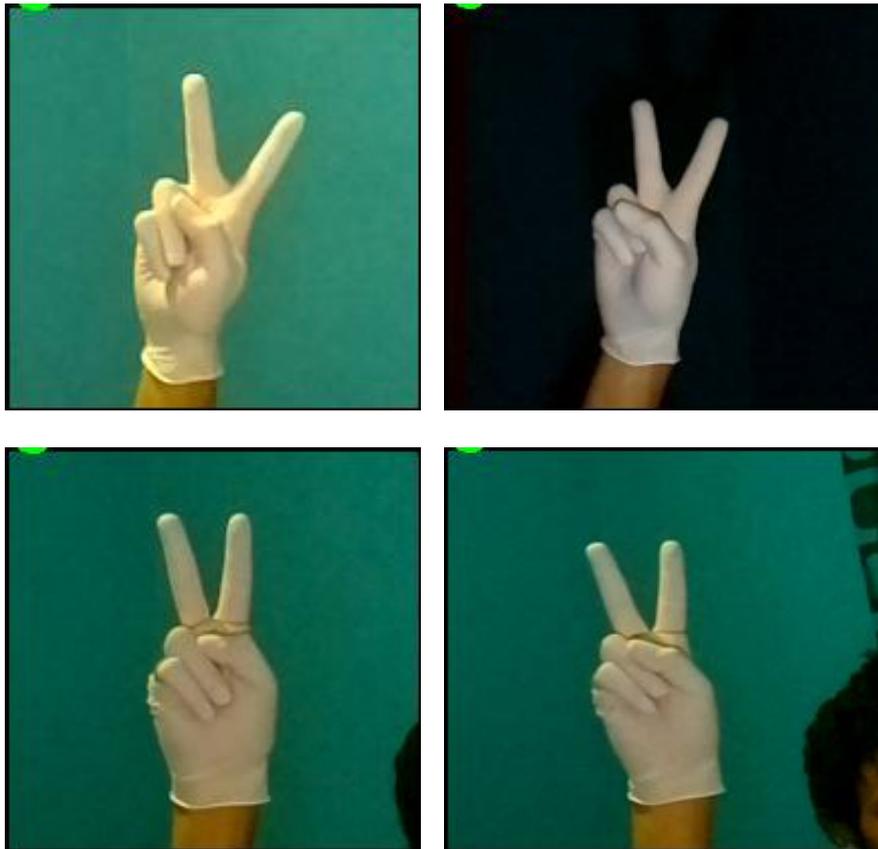


Figure 3.13 Geste 2 (Quelques échantillons)

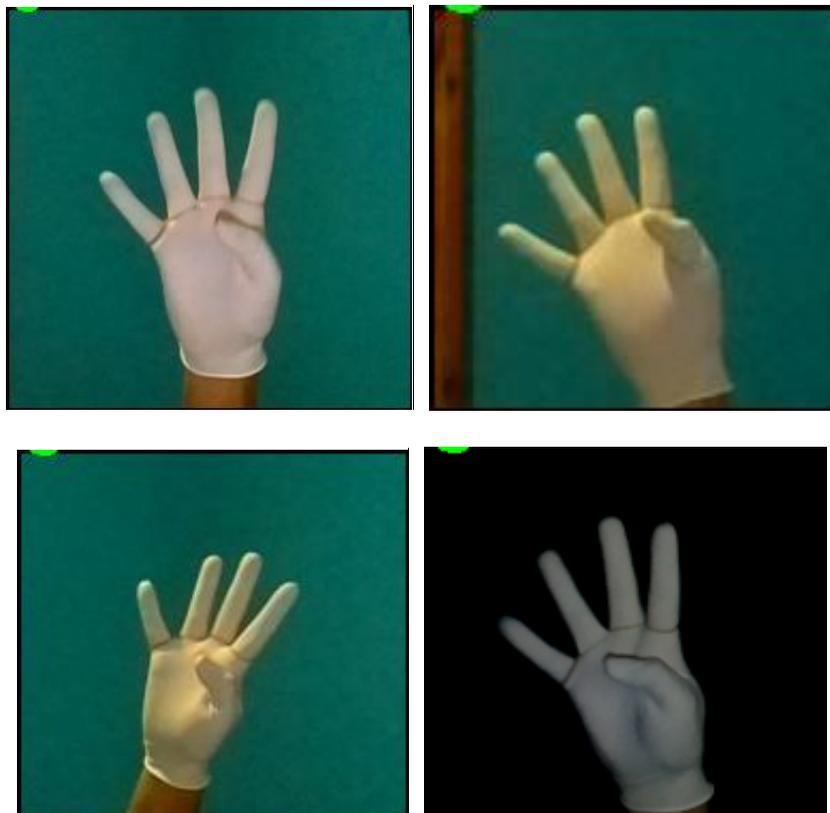


Figure 3.14 Geste 3 (Quelques échantillons)

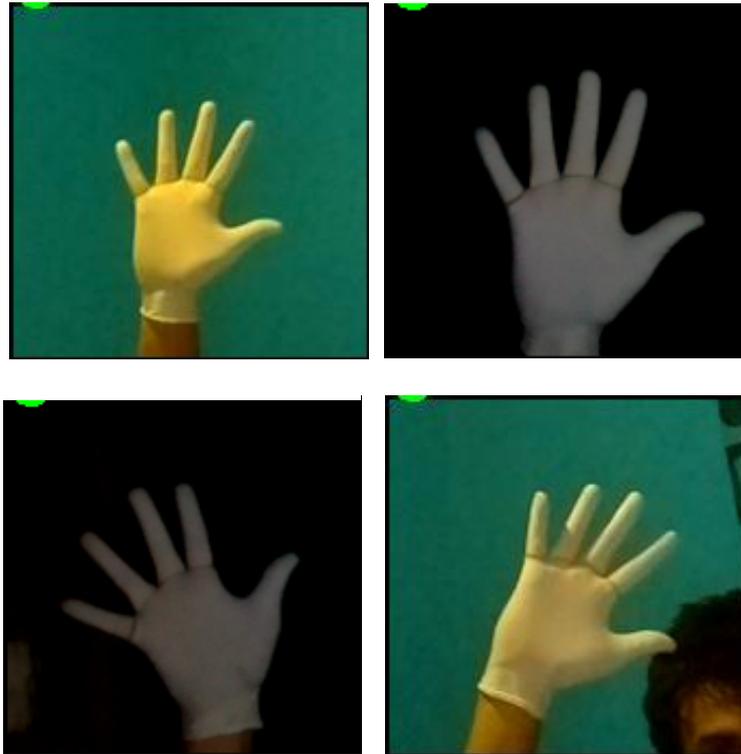


Figure 3.15 Geste 4 (Quelques échantillons)

Les images qui entrent dans le réseau d'apprentissage :

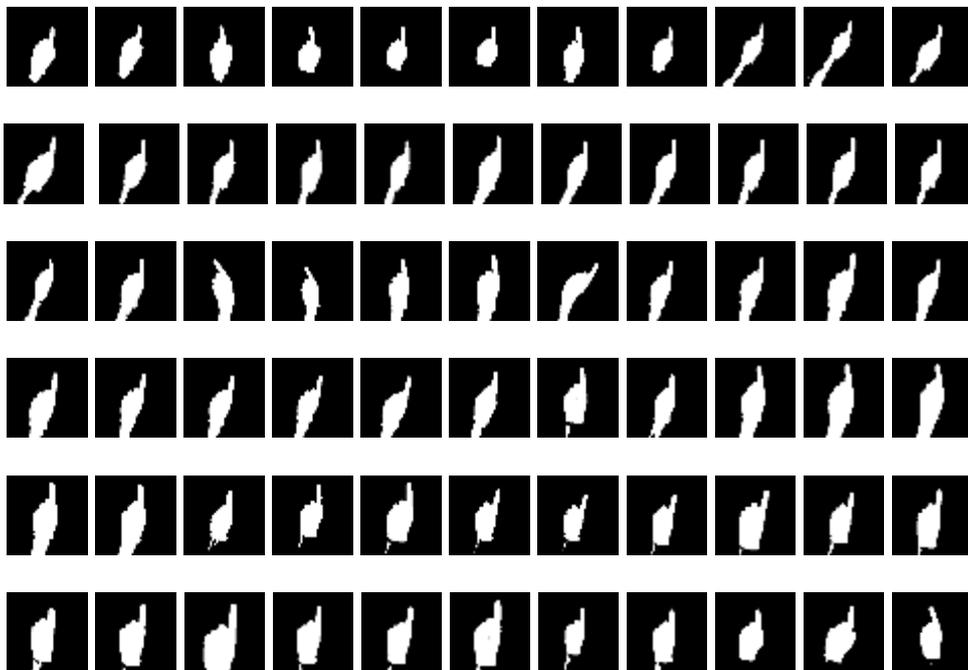


Figure 3.16 Geste 1 (Quelques échantillons*binaire*)

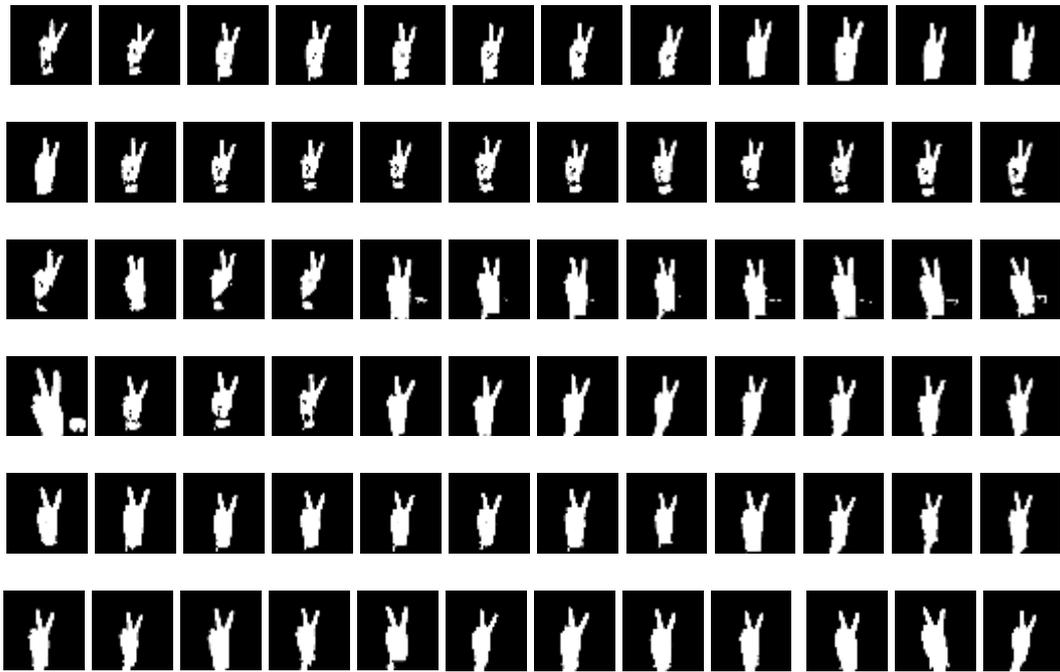


Figure 3.17 Geste 2 (Quelques échantillons*binaire*)

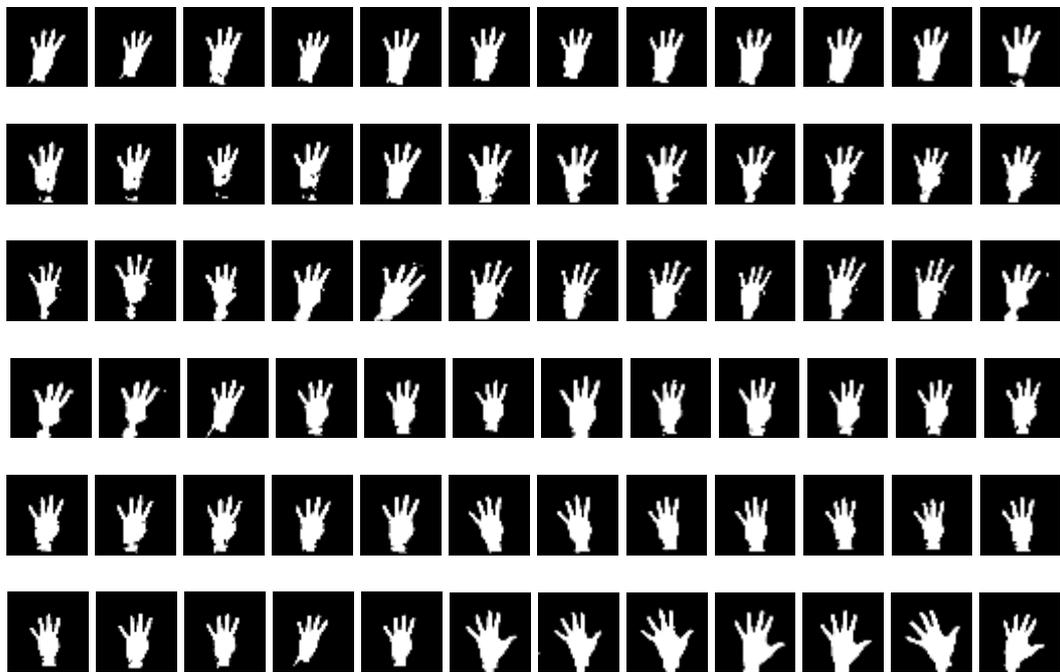


Figure 3.18 Geste 3(Quelques échantillons*binaire*)

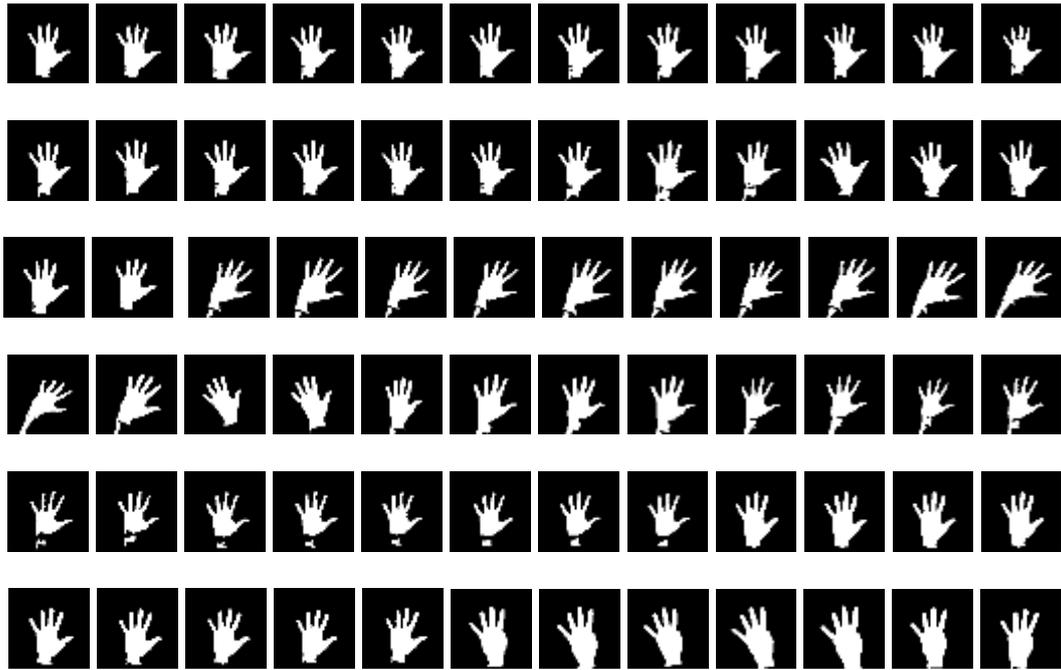


Figure 3.19 Geste 4 (Quelques échantillons*binaire*)

b)Structure du réseau neuronal:

Le réseau de neurones utilisé dans ce projet est un réseau PMC, à deux couches cachées. L'entrée, qui est composée de tous les pixels de l'image (Voir la figure 3.20) , et la sortie du réseau où on a autant de neurones que de gestes à différencier (soit 4 sorties) .

L'apprentissage des poids des neurones du réseau est réalisé avec l'algorithme de rétro propagation du gradient, où chaque neurone est utilisé comme fonction de transfert sigmoïde. Ainsi, lorsqu'un geste est reconnu, le neurone correspondant sur la dernière couche aura une valeur en sortie proche de 1, et les autres neurones auront une valeur proche de 0.

Les paramètres qui pourront être modifiés pour un meilleur apprentissage sont le nombre de neurones de la couche cachée, et le nombre d'itération d'apprentissage maximum.

Les valeurs des entrées sont simplement l'intensité du pixel en niveaux de gris (de 0 à 255), normalisés à des valeurs réels entre 0 et 1.

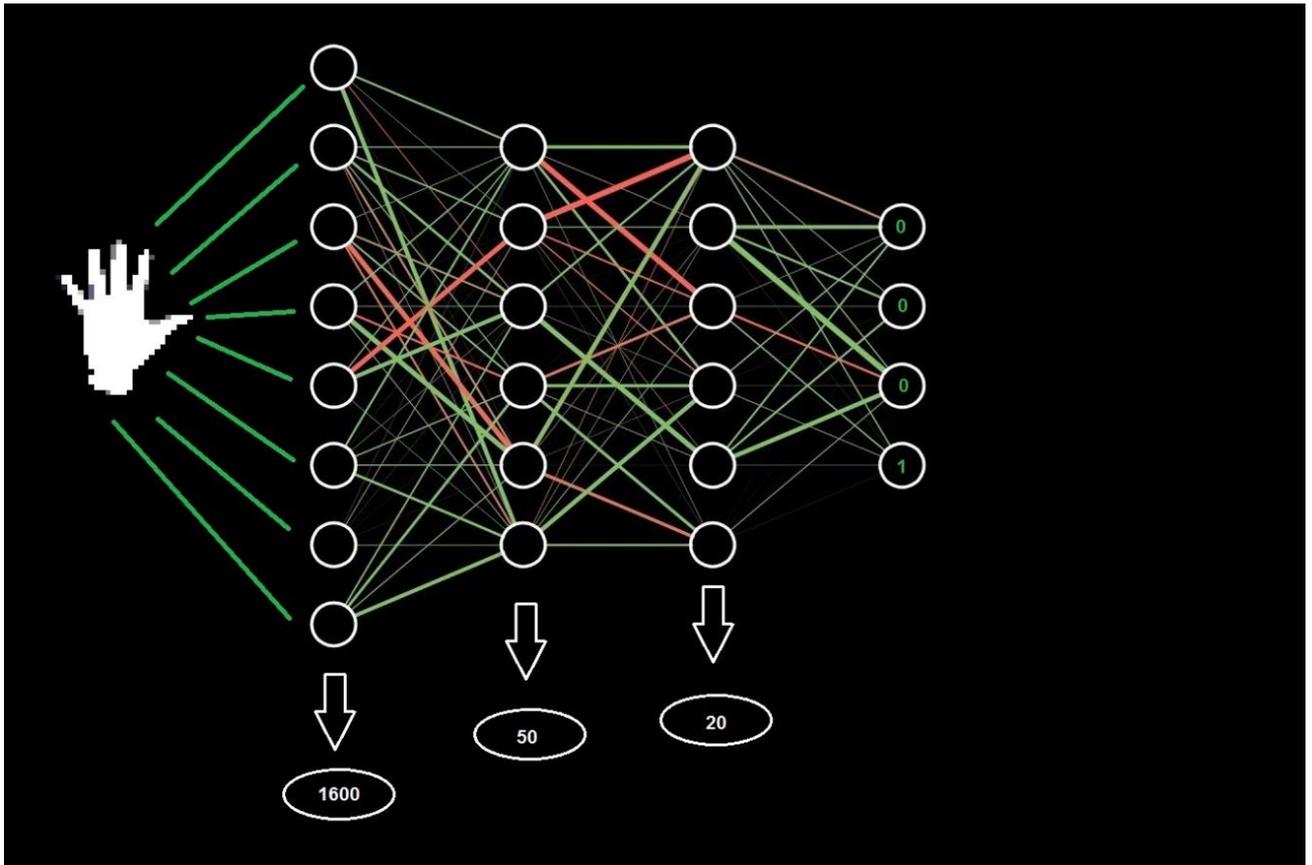


Figure 3.20 Structure du réseau neuronal

c) Les méthodes principales pour l'exécution de réseau de neurone PMC :

- **Lancement du PMC:**

Nous avons développé une méthode qui fait l'appel des images stockées et les place dans une matrice à quatre dimensions. Dans les première et deuxième dimensions, nous trouvons les coordonnées de l'image. La troisième dimension symbolise les images de la même catégorie et dans la quatrième dimension, nous trouvons d'autres types d'images. Nous l'avons fait pour atteindre la vitesse la plus élevée possible et contrôler l'apprentissage et changer les techniques d'apprentissage. Apprentissage non répétitif pour éviter les problèmes de conservation.

- **Construction d'une matrice d'apprentissage pour PMC**

Dans la procédure suivante, nous avons créé un classe qui inclut les caractéristiques des neurones. Une matrice interne contenant des poids et une variable interne qui inclut l'entrée et une autre variable qui symbolise la sortie. Ensuite, une autre fonction construit le réseau désiré.

- **Fonction de propagation**

La propagation vers l'avant se calcule à l'aide de la fonction d'activation sigmoïde. L'actualisation des poids des neurones du réseau est fait avec l'algorithme de rétro propagation du gradient. Ensuite, une autre image est transmise et ainsi de suite jusqu'à ce que le taux d'erreur souhaité soit atteint, Le schéma de l'algorithme pour cet apprentissage est donné par la figure 3.21 .

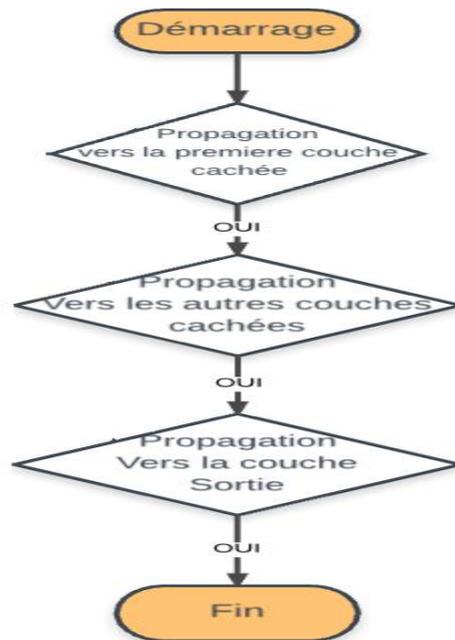


Figure 3.21 schéma de l'algorithme

```

45 sp1[43] fp1, sp2[1] fp2sval=[1600] fvalstar=0[-1,98680711]end=0star=1[2,01319289]end=1star=2[0,01319277]end=2star=3[1,01319277]end=3star=4[-4E-8]
46 sp1[44] fp1, sp2[1] fp2sval=[1600] fvalstar=0[-2]end=0star=1[-1]end=1star=2[-1]end=2star=3[2]end=3star=4[-4E-8]
47 sp1[45] fp1, sp2[1] fp2sval=[1600] fvalstar=0[1,99929297]end=0star=1[-2,00070763]end=1star=2[-0,00070726]end=2star=3[1,00070726]end=3star=4[-4E-8]
48 sp1[46] fp1, sp2[1] fp2sval=[1600] fvalstar=0[3,00005651]end=0star=1[1,00005686]end=1star=2[-0,97204804]end=2star=3[1,00005686]end=3star=4[-4E-8]
49 sp1[47] fp1, sp2[1] fp2sval=[1600] fvalstar=0[1]end=0star=1[2]end=1star=2[2,00106907]end=2star=3[1,00106931]end=3star=4[-4E-8]
50 sp1[48] fp1, sp2[1] fp2sval=[1600] fvalstar=0[1]end=0star=1[-1]end=1star=2[2,99628258]end=2star=3[-2,00371742]end=3star=4[-4E-8]
51 sp1[49] fp1, sp2[1] fp2sval=[1600] fvalstar=0[-2,99999976]end=0star=1[-0,99999988]end=1star=2[-0,99999988]end=2star=3[1,00000012]end=3star=4[-4E-8]
52 sp1[0] fp1, sp2[2] fp2sval=[50] fvalstar=0[-1,38895714]end=0star=1[-0,10644297]end=1star=2[1,75894737]end=2star=3[1,75894737]end=3star=4[-4E-8]
53 sp1[1] fp1, sp2[2] fp2sval=[50] fvalstar=0[1,68368733]end=0star=1[0,79790998]end=1star=2[0,77909571]end=2star=3[1,77909571]end=3star=4[-4E-8]
54 sp1[2] fp1, sp2[2] fp2sval=[50] fvalstar=0[1,64981747]end=0star=1[0,32457092]end=1star=2[-2,57233644]end=2star=3[1,57233644]end=3star=4[-4E-8]
55 sp1[3] fp1, sp2[2] fp2sval=[50] fvalstar=0[0,04453078]end=0star=1[0,36768445]end=1star=2[-0,00195885]end=2star=3[1,00195885]end=3star=4[-4E-8]
56 sp1[4] fp1, sp2[2] fp2sval=[50] fvalstar=0[1,99856532]end=0star=1[1,99461734]end=1star=2[2,0000422]end=2star=3[-2,0000422]end=3star=4[-4E-8]
57 sp1[5] fp1, sp2[2] fp2sval=[50] fvalstar=0[0,756226]end=0star=1[1,50727654]end=1star=2[-0,42068812]end=2star=3[-2,42068812]end=3star=4[-4E-8]
58 sp1[6] fp1, sp2[2] fp2sval=[50] fvalstar=0[-2,95892644]end=0star=1[0,02569542]end=1star=2[-2,99624777]end=2star=3[1,99624777]end=3star=4[-4E-8]
59 sp1[7] fp1, sp2[2] fp2sval=[50] fvalstar=0[-0,49610585]end=0star=1[2,75857496]end=1star=2[-2,43567252]end=2star=3[1,43567252]end=3star=4[-4E-8]
60 sp1[8] fp1, sp2[2] fp2sval=[50] fvalstar=0[0,08460242]end=0star=1[-0,98782498]end=1star=2[-0,88198793]end=2star=3[1,88198793]end=3star=4[-4E-8]
61 sp1[9] fp1, sp2[2] fp2sval=[50] fvalstar=0[0,26501977]end=0star=1[0,02547895]end=1star=2[-1,73293662]end=2star=3[1,73293662]end=3star=4[-4E-8]
  
```

Figure 3.22 Quelques valeurs de poids synaptique existant dans ce fichier (81080 Poids synaptique)

3.6.2 Reconnaissance pour un ensemble test

L'évaluation de tout système de reconnaissance revient à déterminer, par un test le taux de reconnaissance. Ce taux de reconnaissance est la probabilité avec laquelle un geste est identifié, et s'il existe bien sûr dans la base des gestes.

Une fois le réseau est calculé, il faut procéder à des tests pour vérifier que le réseau réagit bien comme on le souhaite.

Dans cette expérience, nous avons exploré plusieurs aspects du système de reconnaissance, et nous avons testé l'influence des différents paramètres qui entrent en jeu sur le résultat final.

Les paramètres investis sont notamment :

- L'influence du nombre de neurones dans la couche cachée,
- L'influence de la dimension de l'image.
- Les paramètres d'apprentissages utilisés dans cette expérience, qui sont:
 - Pas d'apprentissage : $\alpha = 0.1$
 - L'erreur total : $\Delta E = 0.001$
- Les poids sont donnés comme des valeurs aléatoires entre une certaine plage, généralement $[-1, 1]$.
- Nombre d'itération maximale : plusieurs itérations (10000×4)
- Le temps d'apprentissage est (1/2heure)
- On répète la procédure d'apprentissage avec différents nombres de neurones dans la couche cachée, et différentes dimensions d'images, jusqu'au résultat final.

3.6.3 Résultats expérimentaux

Cet apprentissage est fait sous Delphi (plateforme Windows), mais la Raspberry pi est sous python, ce problème nous a causé une incompatibilité.

Pour le résoudre, nous avons construit notre propre algorithme qui consiste à nous donner un fichier XML à partir de cet apprentissage, ce fichier XML est considéré comme une base

de donné robuste qui peut être utilisé sur n'importe quelle plat forme (soit Delphi soit Python), avec cet algorithme, nous avons la possibilité d'acquérir une base de donné portable (Utilisé même sur les smartphone).

Xml Dataset = Poids synaptique.

Ces images de test seront prises aléatoirement parmi toutes les images disponibles.

a)Test 1 :

La dimension d'image : 40*40, Pas d'apprentissage = 0.1, $\Delta E = 0.001$,

Nombre d'itération = 10000 ,bias=1

Structure du réseau neuronal : 1600-50-20-4

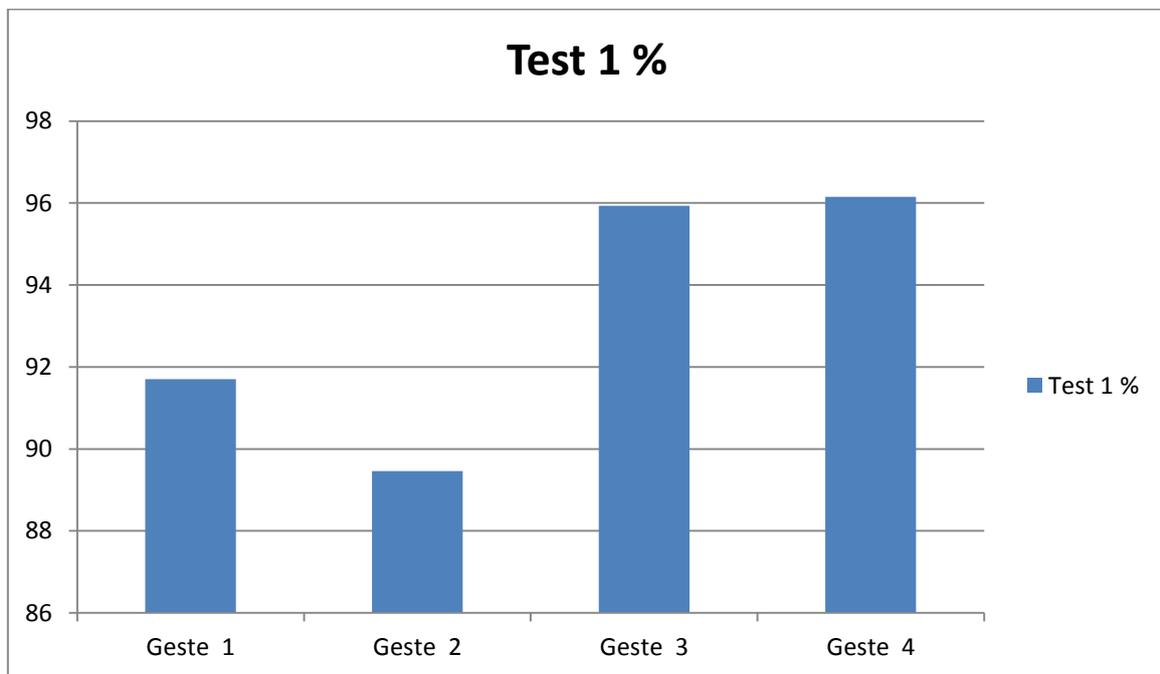


Figure 3.23 Résultats de premier test

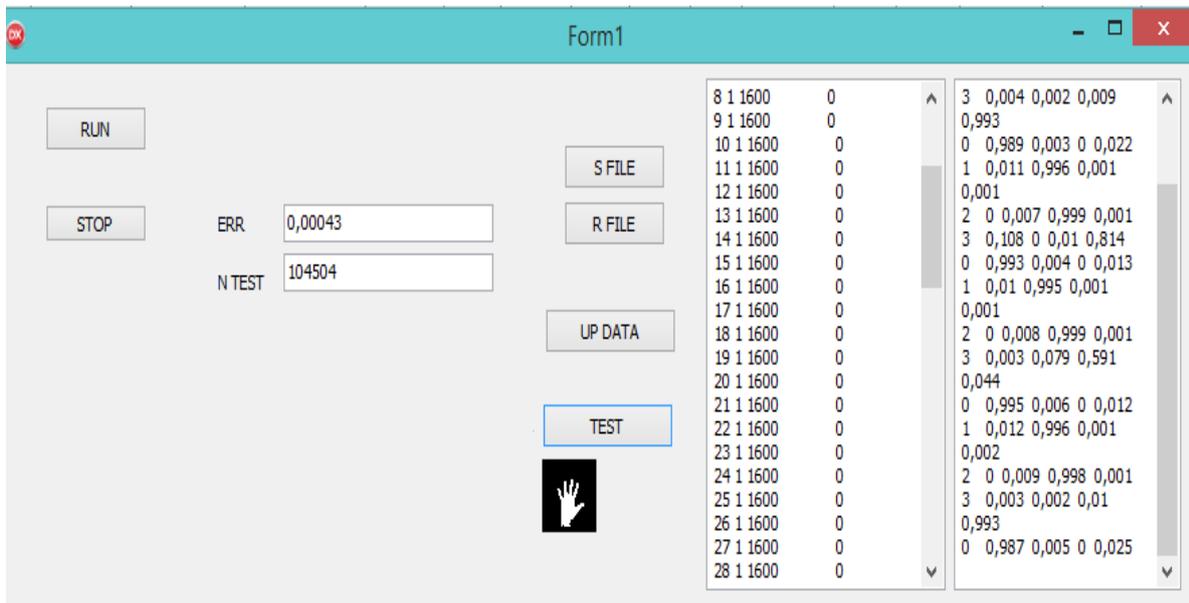


Figure 3.24 Test pour des images de 40*40 pixels

b)Test2 :

La dimension d'image : 20*20 , Pas d'apprentissage = 0.1, $\Delta E = 0.001$,

Nombre d'itération : + 100000 ,bias =1

Structure du réseau neuronal : 400-200-50-4

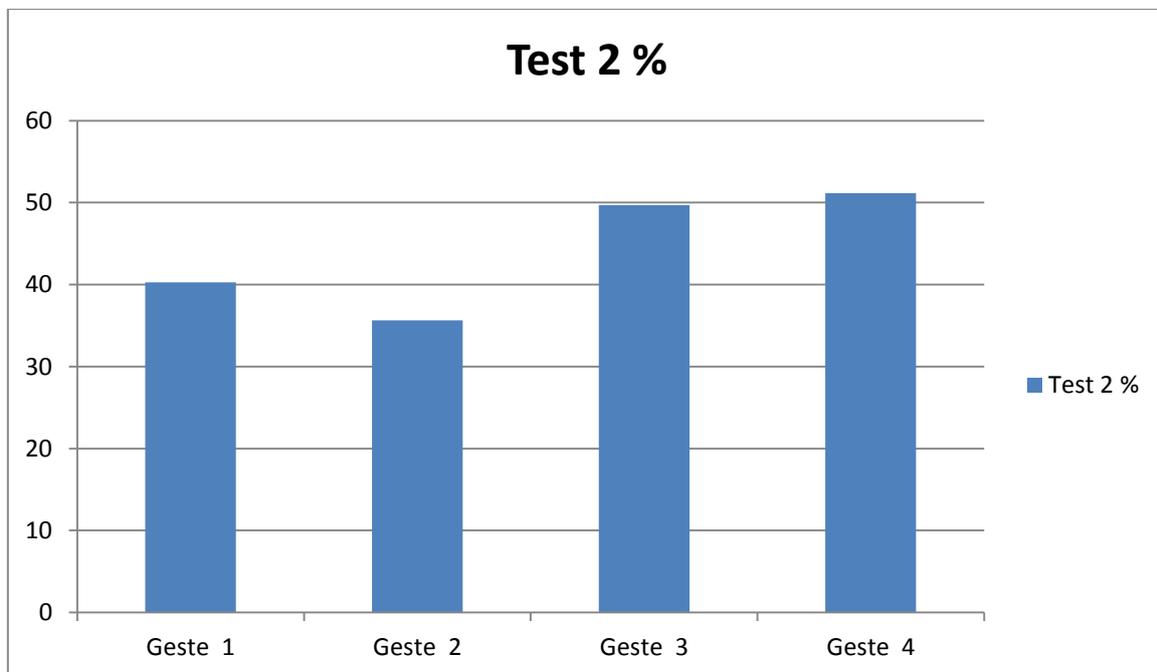


Figure 3.25 Résultats de 2 éme test

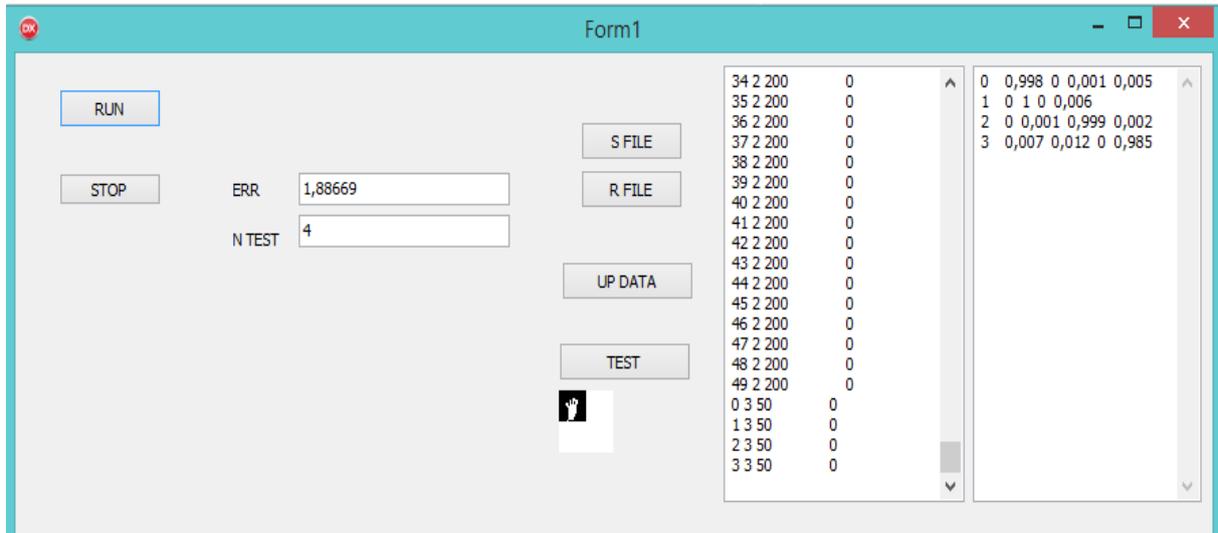


Figure 3.26 Test pour des images de 20*20 pixels

c) Les paramètres utilisés dans notre projet sont

1. Type de Réseau de neurone: PMC
2. Nombre de neurones dans la couche d'entrée: 1600+bias
3. Nombre de neurones dans la couche 1 cachée: 50 +bias
4. Nombre de neurones dans la couche 2 cachée: 20 +bias
5. Nombre de neurones dans la couche de sortie: 4
6. Pas d'apprentissage:0.1
7. Nombre d'itération : +10000

Le temps d'apprentissage est resté toujours long.

3.6.4 Étapes de la phase d'exécution

La chaîne d'exécution de notre robot sera comme suit :

Après avoir créé le réseau de neurones, place à l'exécution :

Nous avons utilisé la Raspberry PI Camera pour l'identification des gestes. Elle diffuse des images de flux vidéo en direct dans un intervalle de temps. Dans notre cas, le taux de capture d'images pour la détection de gestes est de 1.5 image par seconde. L'image obtenue est en format RVB, puis convertie en niveaux de gris pour une taille de l'image obtenue est 640X480. L'image de main capturée est détectée par l'algorithme Haar et traitée par la bibliothèque *opencv* du PYTHON et convertie en image binaire. Ensuite,

l'image est redimensionnée a $40 \times 40 = 1600$ (cette image est considérée comme le vecteur d'entrée). présenté à l'entrée du réseau neurones. Pour chaque vecteur d'entrée, nous assignerons un vecteur cible. Le vecteur cible est supposé être une seule image d'un seul geste, ce qui est commun à tous les vecteurs d'entrée. Enfin, ce vecteurs est envoyé à la carte Arduino et sera après déterminé par cette carte pour qu'elle soit exécuté en fonction de sa nature par le robot. On peut résumé cette exécution en plusieurs sous-étapes :

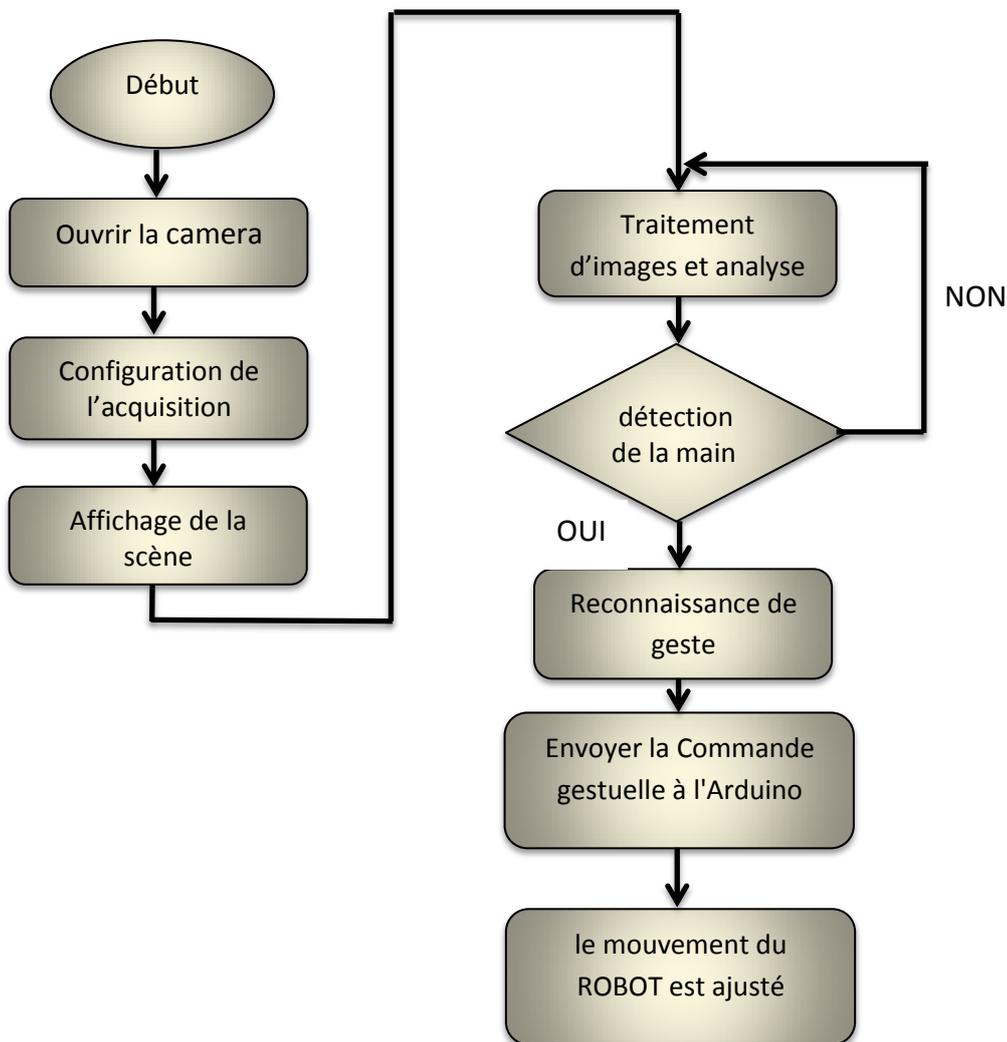


Figure 3.27 les étapes essentielles pour la reconnaissance de geste

3.6.5 La reconnaissance de geste est basée essentiellement sur les étapes suivantes :

- Le système de suivi des mains en temps réel peut être utilisé pour capturer le geste si le geste est disponible dans la plage de vision qu'il est capturé par la caméra
- Convertir l'image en une image binaire
- Redimensionner la zone de geste de la main de l'image capturée.
- Présenter le vecteur caractéristique du geste au réseau de neurone pour effectuer la classification sur la base de données stockée dans le fichier XML et qui retourne le résultat en tant que vecteur.
- La sortie de classification de réseau neuronal est utilisée pour piloter le matériel.
- L'Arduino est utilisé pour la partie hard.
- Les commandes de contrôle sont directement envoyées à Arduino depuis PYTHON (voir la figure 3.28).

Sur la base de cette commande , le mouvement du ROBOT est ajusté (voir la figure 3.29).

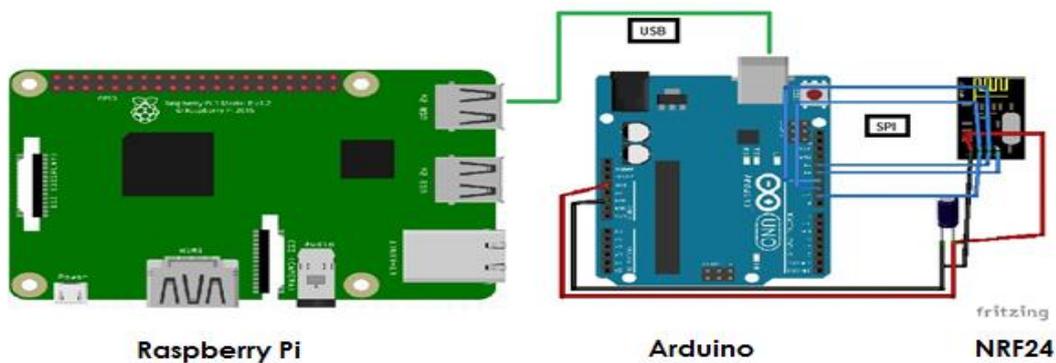
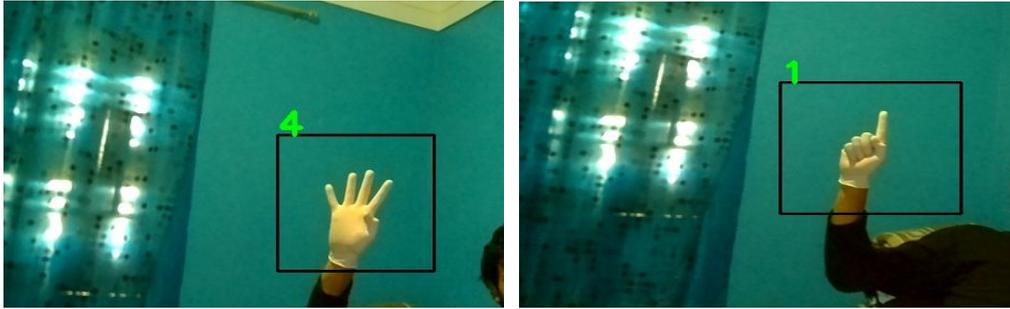


Figure 3.28 schéma électronique de système



Le robot tourne à droite

Le robot avance



Le robot recule

Le robot tourne à gauche

Figure 3.29 Les commandes de contrôle

a) Remarque

l'apprentissage a besoin + 10000 itération pour pouvoir reconnaître les 4 gestes, la durée totale d'apprentissage Environ une demi-heure avec quelques modifications au réseau et la reconnaissance Environ 1.5 secondes (sur la Raspberry Pi).

3.7 Discussion du résultat

Le meilleur taux de reconnaissance est de 93,31 % avec 1600 neurones (la couche d'entrée en dimension 40*40 (Structure du réseau neuronal : 1600-50-20-4), le résultat tombe à 44.19% avec 400 neurones (la couche d'entrée en dimension 20*20 (Structure du réseau neuronal : 400-200-50-4).

3.8 Conclusion

En dépit des problèmes rencontrés lors de cette phase de réalisation, nous avons quand même réussi à réaliser la plus grande partie du projet en implémentant notre propre bibliothèque et qui peut manipuler ce système de réseau de neurones.

L'apprentissage de réseau de neurones a été accompli dans un PC avec des caractéristiques assez élevées, Après l'apprentissage, le réseau de neurones crée un fichier XML de data Set, ensuite, ce fichier XML placé dans la Raspberry Pi. Ce fichier nous aidera par la suite à reconnaître les gestes de mains ; noté que ce fichier XML est robuste et peut être utilisé sur n'importe quelle plateforme.

Nous avons aussi créé deux applications, une pour la Raspberry Pi : Cette application va permettre de détecter le geste de la main (par l'algorithme Haar) en RVB et aussi la

Convertir en niveaux de gris et ensuite en image binaire . c'est fait l'objet d'un vecteur d'entrée d'un réseau de neurons spécifique qui nous donne en sortie un vecteur.

Une deuxième application qui sera attribué pour la carte Arduino : cette application va recevoir le geste traduit (vecteur de sortie) à partir de la connexion établie et va l'utiliser pour après l'exécuter en tant que commande du robot.

Conclusion générale

En cours de ce projet et ce fil de temps, nous avons pu en créer un robot à partir de zéro, c'est-à-dire couvrir toute branche de la robotique, en partant de la mécanique (création du robot avec des pièces matériel, moteur, roues...etc.), ensuite le coté électronique du robot c'est-à-dire en attribué à ce robot le mouvement nécessaire qui lui permettra de manœuvrer dans le monde extérieur (en utilisant une carte Arduino et son IDE pour pouvoir attribuer ces différents mouvement et l'alimentation requise pour cela).

Aussi le point le plus important de ce projet est l'apprentissage et qui a été d'une grande importance. Pour faire cela, nous avons aussi créé des bibliothèques et des algorithmes en partant de zéro, tels que le fichier qui nous permet de conserver nos informations créé à partir du réseau de neurones implémenté en DELPHI à être utilisé dans un environnement sous python. Sans oublié la connexion entre la Raspberry pi et l'Arduino et aussi la répartition des tâches en fonction des caractéristiques de ces dernières ou la Raspberry pi se concentre sur un travail plus complexe tel que la détection du geste de la main, son analyse et aussi sa traduction et son envoi à partir d'un signal vers la carte Arduino.

Et la carte Arduino se contente d'analyser et exécuter la tache (le geste de main traduit) en forme de mouvement spécifié du robot.

Ce projet nous a bien été utile, nous avons appris beaucoup de chose au cours de ce dernier tel que le travail d'équipe et son importance, la valorisation du temps et plus important, les bienfaits du cerveau, car cela semble facile de reconnaitre un visage ou un geste de main et après accomplir une tâche précise à ce geste, un instinct humain facile à interpréter, mais cela demande un grand travail et une immense recherche et de sacrifice pour simplement faire une reconnaissance de geste de main après agir.

Nous espérons que ce robot aura un grand impact dans la société, ou il sera introduit comme support des gens aveugles à la place des chiens pour ceux qui sont allergiques.

Aussi, nous allons donner la possibilité aux sourds-muets de communiquer avec la technologie de l'intelligence artificielle, ils sera d'une grande aide pour les handicapés car l'importance de technologie est de faciliter la vie des gens.

Bibliographie

- [1] Cours ENSTA D9-2 Antoine Manzanera - Unité d'Électronique et d'Informatique
- [2] Cours du module segmentation d'image.
- [3] Martine Bergounioux. Quelques méthodes mathématiques pour le traitement d'image. DEA. Cours M2 - Université d'Orléans 2008-2009, 2008, pp.110. <cel-00125868v4> [4]<http://www.la-photo-en-faits.com/2012/11/le-bruit-en-photo-numerique.html>
- [5] Chapitre Généralités sur le traitement d'images Introduction univ-biskra de D Boukhlof - 2005
- [6] LEMMOUCHI Mansoura Ingénieur, département d'Electronique - Université de Batna Thème :Identification des Visages Humains par Réseaux de Neurones
- [7] Jian Liang, David Doermann, Huiping Li, Camera-based analysis of text and documents: a survey, (Language and Media Processing Laboratory, Institute for Advanced Computer Studies, University of Maryland, College Park, USA, 2005)
- [08] Adriana Climescu-Haulica ENSIMAG <http://ljk.imag.fr/membres/Adriana.Climescu/rncours1.htm>
- [9] Salima Nebti : 'Reconnaissance de Caractères Manuscrits par Intelligence Collective', thèse Doctorat en science, Université Ferhat Abass-Sétif, 07/mars/2013.
- [10]http://www.robot-maker.com/shop/blog/33_Asservir-moteurs-courant-continu-PID.html
- [11] Paul Viola and Michael Jones. Robust real-time object detection. In Second international work shop on statistical and computation theories of vision, Vancouver, Canada, July 13 2001.
- [12] Face tracking : implémentation de la méthode de Viola & Jones en C++
<https://www.firediy.fr/article/face-tracking-implementation-de-la-methode-de-viola-jones-en-c>
- [13] Alain Schneider. NRF24L01+ sniffing for Arduino. <https://github.com/cogiceo>, 2014.
- [14] Réseaux de neurones <https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-rn.pdf>

- [15] Gilles LE BRIS : Optimisation de la reconnaissance automatique de documents pour la dematerialisation des pieces de caisses du Credit Agricole Normandie Seine 2017.
- [16] http://wiki.labaixbidouille.com/index.php/C%27est_quoi_un_Raspberry_Pi_%3F
- [17] Al Falou Wassim : 'Reconnaissance de caractères manuscrits par réseau de neurones', thèse DEA Modélisation et ingénierie du logiciel scientifique, UNIVERSITE SAINT – JOSEPH, UNIVERSITE LIBANAISE, Décembre 1998.
- [18] Laetitia Leyrit : Reconnaissance d'objets en vision artificielle : application à la reconnaissance de piétons
- [19] Conception et réalisation d'un robot mobile à base d'arduino par Hocine TAKHI
Université Amar Telidji - Instrumentation 2014
- [20] Algorithme de rétropropagation
perso.ens-lyon.fr/etienne.moutot/static/etudes/MP/TIPE/preuve_grad.pdf
- [21] Leila CHERGUI : 'Combinaison de classifieurs pour la reconnaissance de mots arabes Manuscrits', Thèse doctorat en Informatique, Université Mentouri Constantine, 2012-2013.
- [22] <http://www.grappa.univ-lille3.fr/polys/apprentissage/sortie005.html>
- [23] MECA 2732 : Introduction à la robotique. Titulaires: P. Fisette, H. Buyse, J.C. Samin
- [24] cour robotique M2 SVR
- [25] http://projet.eu.org/pedago/sin/term/6-asservissement_arduino.pdf