



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique et informatique industriels
Réf. :

Présenté et soutenu par :
GHARSA OUMAIMA
Le : mardi 1er septembre 2020

Fauteuil Roulant à Commande Oculaire pour Handicapés Physique

Jury :

Mr. ABADA Khaled	MCA	Université de Biskra	Président
Mr. BOUMEHRAZ Mohammed	Pr	Université de Biskra	Encadreur
Mr. ZITOUNI Athmane	MCA	Université de Biskra	Examineur

Année universitaire : 2019 - 2020

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Automatique
Option : Automatique et informatique industriels

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER

Thème

Fauteuil Roulant à Commande Oculaire pour
Handicapés Physique

Présenté par :

GHARSA OUMAIMA

Avis favorable de l'encadreur :

Pr. BOUMEHRAZ Mohammed

Signature

Avis favorable du Président du Jury :

Mr. ABADA khaled

Signature

Cachet et signature

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electrotechnique
Option : Réseau électrique

Thème :

Fauteuil Roulant à Commande Oculaire pour Handicapés Physique

Proposé et Dirigé par : **Pr. BOUMEHRAZ Mohammed**

RESUME

Le projet est un nouveau système de contrôle d'un fauteuil roulant commandé par les mouvements oculaires des yeux. Ce dispositif de commande intelligent convient à un grand nombre des handicaps physiques qui ne peuvent pas manipuler un fauteuil roulant à un organe de commande standard. Le système de contrôle du mouvement des yeux utilise une caméra fixée sur le fauteuil roulant devant le conducteur. Les mouvements des yeux du patient sont reconnus à l'aide d'une technologie de reconnaissance des points de repère du visage. Le système proposé est basé sur les mouvements des pupilles, le moteur du fauteuil roulant se déplace à gauche, à droite et en avant et utilise le clignement des yeux pour l'arrêt et pour se marche en arrière.

Mots clés : Fauteuil roulant intelligent, Personnes handicapées motrices, OpenCV, Dlib, Raspberry Pi.

ABSTRACT

the project is a new control system of an eye movement controlled wheelchair. This smart control device is suitable for a large number of physically disabled who cannot manipulate a standard control unit wheelchair. The eye movement control system uses a camera fixed on the wheelchair in front of the conductor. The patient's eyes movements are recognized using face landmarks recognition technology. The proposed system

based on eye pupil movements, the wheelchair motor moves left, right and forward and used the eye blinking for the stop and moves backward of wheelchair motor.

Keywords: Intelligent wheelchair, Motor disabilities, OpenCV, Dlib, Raspberry Pi.

الملخص

المشروع عبارة عن نظام تحكم جديد لكروسي متحرك يتم التحكم فيه بحركة العين. جهاز التحكم الذكي هذا مناسب لعدد كبير من المعاقين جسديًا والذين لا يستطيعون التعامل مع كروسي متحرك باستخدام وحدة التحكم. يستخدم نظام التحكم في حركة العين كاميرا مثبتة على الكروسي المتحرك أمام المستخدم. يتم التعرف على حركات عيون المريض باستخدام تقنية التعرف على معالم الوجوه. النظام المقترح يعتمد على حركات بؤبؤ (التلميذ) العين ، بحيث يتحرك الكروسي المتحرك لليساار واليمين والأمام وكما يستخدم وميض العين لإيقاف الكروسي المتحرك و تحرك للخلف.

الكلمات المفتاحية : كروسي متحرك ذكي. ذوي الإعاقات الحركية, مكتبة رؤية الكمبيوتر, راسبيري باي.

Remerciements

*Avant toute chose, je remercie dieu tout puissant de
M'avoir aidé et éclairé le chemin pour la réalisation*

De cette mémoire

Je tiens à exprimer mes sincères remerciements à :

*Mon encadreur Monsieur 'Boumehraz Mohamed'
pour le temps qu'il m'a consacré, la Patience dont il
a fait preuve et les conseils...Merci !*

*Mes chers parents, pour leur sacrifice, leurs
encouragements et conseils.*

Mon marie pour leur encouragement et aide.

*J'exprime toute ma gratitude et mon respect Aux
membres de jury qui me feront l'honneur*

D'apprécier ce modeste travail

Dédicaces

Je dédie ce Travail :

*A mon père, qui m'a appris et me sauver tout et
pour ses précieux conseils.*

*A ma mère, pour son soutien et encouragement
durant mes études.*

*A ma famille mes grands-mères, mes deux frères,
ma sœur, tantes, oncles, chacune par son nom.*

*Je remercie énormément mes amies et collègues
d'automatique Oumaima Hattab, Nessrine,
Selma, Doha.*

*Enfin Tous les enseignants de la filière
automatique pour les connaissances qu'ils m'ont
transmis pendant La période de ma formation*

Table des matières

Résumé.....	iii
Remerciements	v
Dédicaces.....	vi
Table des matières.....	vii
Liste des Tableaux.....	x
Liste des Figures.....	xi
Liste des Abréviation	xiv
Introduction générale	1
Chapitre I : Généralité sur le fauteuils roulants intelligents	3
I.1. Introduction :	3
I.1. Généralités sur les fauteuils roulants électriques intelligents :	3
I.2.1. Les origines du fauteuil roulant électrique intelligent :	3
I.2.2. Descriptif du fauteuil roulant électrique intelligent :	4
I.2. Interface de commande :	6
I.3. Les méthodes d'entrée :	6
I.4. Les Fauteuils roulants intelligents :	11
I.5.1. Le projet VAHM :	11
I.5.2. Le projet WAD :	12
I.5.3. Le projet NavChair :	13
I.5.4. Le projet TinMan :	14
I.5.5. Le Projet Wellesley College, MIT :	15
I.5.6. Le projet de SENARIO, TIDE Programme :	16
I.5.7. Le projet d'OMNI, Fauteuil roulant d'omnidirectionnel :	17
I.5.8. Le projet FRIEND :	18

I.5.9. Le projet SPAM :	19
I.5. Conclusion :	20
Chapitre II : Détection et Reconnaissance du visage et des yeux	21
II.1. Introduction :	21
II.2. Architecture du système :	21
II.3. OpenCV :	22
II.3.1. Histoire :	22
II.3.2. Fonctionnalités :	24
II.3.3. Structure de la librairie OpenCV :	26
II.3.4. Installation d'OpenCV :	26
II.4. Les méthodes des traitements du visage :	28
II.4.1. La détection du visage :	28
II.4.2. La Système de Reconnaissance de Visage :	31
II.4.2.1. Reconnaissance de visages 2D :	31
II.4.2.2. Reconnaissance de visages 3D :	33
II.5. Les méthodes de traitement des yeux :	36
II.5.1. La détection des yeux :	36
II.5.2. Reconnaissance des yeux :	36
II.6. Conclusion :	40
Chapitre III : Résultats de l'étude Et Système de Commande	42
III.1. Introduction :	42
III.2. Le Système proposé :	42
III.2.1. Description du partie hardware :	42
III.2.1.1. Matériels utilisés :	43
III.2.2. Description du partie Software :	45
III.2.2.1. Système d'exploitation de Raspberry Pi :	45
III.2.2.2. L'organigramme de système :	48

III.3.	La méthode de détection de visage :.....	49
III.4.	La méthode de détection des mouvements des yeux :.....	50
III.5.	Description du programme :.....	56
III.6.	Résultat de Partie de commande :.....	63
III.7.	Conclusion :.....	67
Conclusion générale		68
Références bibliographique.....		69

Liste des Tableaux

Chapitre I : Généralité sur les fauteuils roulants intelligents

Tableaux I-1 : liste des principaux projets de fauteuils roulants intelligents.....05

Chapitre II : Détection et Reconnaissance du visage et des yeux

Tableaux II-1 : Catégorisation des méthodes de détection des visages.....30

Liste des Figures

Chapitre I : Généralité sur les fauteuils roulants intelligents

Figure I-1 : Galerie des fauteuils roulants intelligents.....	04
Figure I-2 : Exemple de joystick de fauteuil roulant.....	07
Figure I-3 : Modèle d'un Fauteuil roulant intelligent à commande vocale.....	08
Figure I-4 : Interface Cerveau Machine (BCI) pour la 3D	09
Figure I-5 : Signal EOG acquis en temps réel.....	09
Figure I-6 : Bandeau-caméra intégré ; une minuscule (tiny IR) caméra infrarouge a été Montée sur un actionneur réglable fixé au serre-tête.....	11
Figure I-7 : Le prototype du VAHM	12
Figure I-8 : le projet de WAD (Wheelchair Attractor Dynamics).....	13
Figure I-9 : le projet de NavChair, Université de Michigan.....	14
Figure I-10 : TinMan II de l'Institute de KISS.....	15
Figure I-11 : Wheelesley Robot.....	16
Figure I-12 : Le SENARIO autonome fauteuil roulant.....	17
Figure I-13 : OMNI, Fauteuil roulant d'omnidirectionnel.....	18
Figure I-14 : FRIEND Robot	19
Figure I-15 : Le module SPAM placé sur un fauteuil manuel.....	19

Chapitre II : Détection et Reconnaissance du visage et des yeux

Figure II-1 : architecture du système proposé	21
Figure II-2 : Logo D'OpenCV	23

Figure II-3 : La figure suivante montre la comparaison entre une image colorée et une image en niveaux de gris.....	25
Figure II-4 : Structure de la bibliothèque d'OpenCV	26
Figure II-5 : terminal du système	27
Figure II-6 : tester cv2 (pour opencv) et numpy	28
Figure II-7 : (a) Image texture (b) Image 2.5D (c) Image 3D	31
Figure II-8 : Une classification des algorithmes principaux utilisés en reconnaissance faciale.....	33
Figure II-9 : Illustration de « snakes », [ou les contours flexibles dans la méthode de daugman].....	38
Figure II-10 : l'ensemble des appareils utilisé dans cette méthode	39
Figure II-11 : Système de reconnaissance d'iris Régions d'intérêt détectées (ROI).....	40

Chapitre III : Résultats de l'étude et Système de Commande

Figure III-1 : Structure du système Hardware et software	42
Figure III-2 : Raspberry pi 4 (modèle B)	43
Figure III-3 : Arduino Uno	44
Figure III-4 : (A) caméra pi, (B) webcam	45
Figure III-5 : Raspbian Logo	45
Figure III-6 : Kit Raspberry Pi 4.....	46
Figure III-7 : Allumez le Raspberry Pi pour la première fois	46
Figure III-8 : Exécution du système pour la première fois (d'écran HD)	47
Figure III-9 : Organigramme de fonctionnement du système (fauteuil roulant)	48
Figure III-10 : 68 FACIAL LANDMARKS de visage	49
Figure III-11 : détecte de visage	50
Figure III-12 : Les deux lignes de scan pour détecter le mouvement de l'œil	51

Figure III-13 : L'œil	52
Figure III-14 : Les directions de la pupille.....	52
Figure III-15 : Les mouvements de la pupille au niveaux de gris	53
Figure III-16 : l'œil ouvert avec les lignes horizontale et verticale	54
Figure III-17 : l'œil fermé avec les lignes horizontale et verticale.....	54
Figure III-18 : les deux parties de l'œil	55
Figure III-19 : Landmarks des yeux	56
Figure III-20 : Résultats de la commande du fauteuil roulant par la direction des yeux	64
Figure III-21 : Résultats de la commande du fauteuil roulant par les clignotements des yeux	65
Figure III-22 : Résultats de programme de commande de fauteuil roulant sur le Raspberry pi	66

Liste des Abréviations

BCI : Brain Computer Interface.

EEG : Électro-Encéphalographie.

EOG : Électro-oculographie.

EV : Environnement Virtuel.

FRE : Fauteuil Roulant Electrique.

HDMI : High Definition Multimedia Interface

HID : Handicaps-Incapacités-Dépendance.

IR : Infra Rouge.

LED : Diode électroluminescente.

MIT : Massachusetts Institute of Technology.

Introduction

générale

Bien que les récents développements de la science et de la technique aient changé radicalement la façon de vivre des personnes normales, il y a certains groupes de personnes n'ont pas pu bénéficier de cette évolution. En particulier, les personnes handicapées à mobilité réduite vivent encore une vie misérable.

Aujourd'hui, un fauteuil roulant intelligent vise à aider ces personnes handicapées et particulièrement les handicapées physiques en leur offrant des organes de commande (une sorte de mobilité) qui les aiderait énormément. Le fauteuil roulant intelligent se compose d'une unité de commande principale qui permet à l'utilisateur de fournir des informations sous la forme d'un joystick ou d'une commande vocale, ou un souffle, ou un écran tactile, ou par les mouvements des yeux... etc.

Les mouvements des yeux jouent un rôle important dans la façon dont notre cerveau interprète et analyse ce que nous voyons. Les recherches sur les mouvements des yeux ont permis aux scientifiques de comprendre la commande du mouvement oculaire et la capacité du cerveau a traité les perceptions sensorielles.

La problématique abordée dans ce travail consiste à proposer un système autonome pour une personne qui souffre d'un handicap moteur, il est très difficile d'utiliser certains types des organes des commandes de fauteuil roulant intelligent. Les yeux sont les seuls organes qui peuvent bouger, le système de contrôle oculaire (par les mouvements des yeux) améliore les capacités physiques des personnes âgées et handicapées physiques et rendra leur vie plus facile et accessible, plus indépendante et plus pratique.

Il y un domaine plus récent aide à la détection des yeux et leur mouvement, qui utilise le traitement d'images et vidéo comme moyen d'information d'entrée, appelé vision par ordinateur.

L'objectif de notre projet, est de trouver un algorithme fiable qui détectait le visage et les mouvements des yeux en temps réel et qui comprennent cinq modes de

contrôle, selon la position de l'œil, les moteurs du fauteuil roulant sont dirigés pour se déplacer vers la gauche, la droite et vers l'avant ... etc. Nous sommes besoin d'utiliser la bibliothèque OpenCV avec le langage python.

Le manuscrit est divisé en trois chapitres : **le premier chapitre**, nous parlons brièvement sur les origines du fauteuil roulant intelligent, détaillent les différents modes de commande (méthodes d'entrée) et nous clôturons le chapitre par une revue des travaux évoqués concernant la conduite de fauteuils électriques intelligents.

Deuxième chapitre, ce chapitre donne un aperçu sur la bibliothèque OpenCV et fait le tour sur les méthodes de détection et reconnaissance de visage et des yeux, en exposant les approches utilisées pour la détection et reconnaissance de visage et yeux.

Le dernier chapitre, porte sur les résultats expérimentaux obtenus et les matériels utilisés dans notre projet et présentant les méthodes utilisées dans le programme pour détecter le visage et les mouvements des yeux et montreront le résultat final d'exécution (simulation) de programme.

Chapitre I :

Généralité sur le fauteuils

roulants intelligents



I.1. Introduction :

Le fauteuil roulant intelligent est un véhicule de mobilité à roues dans lequel s'assoit l'utilisateur. Il est utilisé pour le déplacement des personnes en difficulté physique ou souffrant d'une maladie, d'une blessure, ou d'un handicap. Les fauteuils roulants intelligents ont souvent des variations de conception de base, mais il en existe de nombreux types.

Le fauteuil roulant intelligent se compose d'une unité de commande principale qui permet à l'utilisateur de le mettre en marche manuellement à l'aide d'un joystick, ou d'une commande oculaire par caméra ou d'une commande vocale par microphone.

Dans ce chapitre, je présente tout d'abord les origines des fauteuils roulants intelligents avec, en particulier, la description de ces véhicules.

Enfin je conclus ce chapitre par la présentation des différents types de commande d'un fauteuil roulant intelligent.

I.2. Généralités sur les fauteuils roulants électriques intelligents :

I.2.1. Les origines du fauteuil roulant électrique intelligent :

Les fauteuils roulants intelligents ont fait l'objet de recherches depuis le début des années 1980 et ont été développés sur les quatre continents [1]. Ce mémoire présente un modèle des fauteuils roulants intelligents qui sont devenu actuellement des objets d'art et d'orientation pour les futures recherches.

Cependant, le fauteuil roulant manuel n'a pas été abandonné et son utilisation reste très largement répandue comme en attestent les résultats des enquêtes « H.I.D. » réalisées en France, où la prévalence des utilisateurs de fauteuils roulants (manuel ou électrique) peut être estimée à 62 personnes pour 10 habitants soit un total de 360 000 personnes. Cette prévalence de l'usage du fauteuil roulant est plus faible en France que dans d'autres pays occidentaux (Angleterre, Canada, États-Unis) [2].

I.2.2. Descriptif du fauteuil roulant électrique intelligent :

Les fauteuils roulants intelligents à propulsion par moteur électrique sont indiqués pour les personnes incapables de se déplacer seules sur fauteuil manuel.

Ils permettent une certaine autonomie notamment pour sortir à l'extérieur du domicile. La conduite du fauteuil s'effectue par commande électronique. Il existe différents types de fauteuils [1].

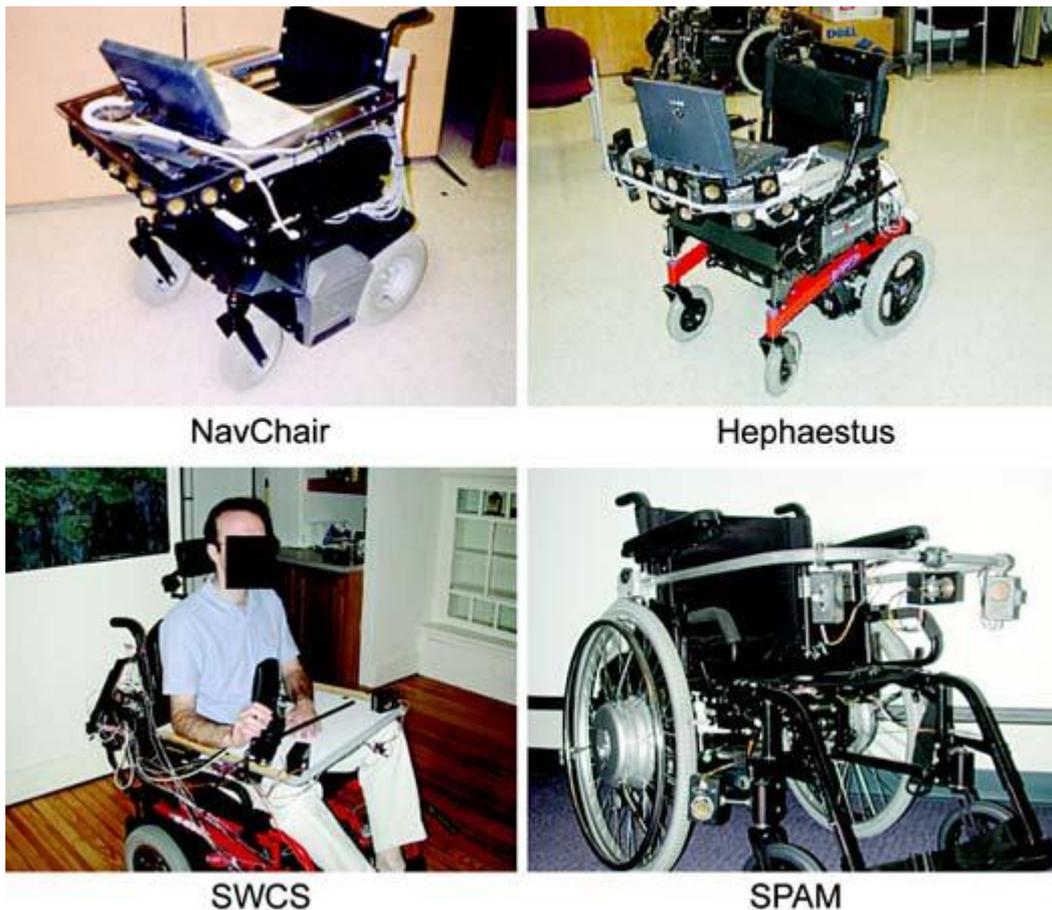


Figure I-1 : Galerie des fauteuils roulants intelligents.

SWCS = Smart Wheelchair Component System

SPAM = Smart Power Assistance Module.

Chapitre I : Généralité sur les fauteuils roulants intelligent

Tableau I-1 : Ci-dessous une liste des principaux projets de fauteuils roulants intelligents décrits dans la littérature. [1][3].

Fauteuils Roulants Intelligents	Date de Publication	Description
Automated Guided wheelchair, NEC cooperation , Japan	1992	Il est guidé automatiquement grâce à des marqueurs magnétiques placés sur le sol. Il utilise des capteurs infrarouges pour la détection des obstacles.
Autonomous Wheelchair, Arizona state university, U.S.A	1986	C'est un fauteuil à navigation autonome. Il est doté de capteurs ultrasons qui lui permettent de naviguer dans des environnements encombrés. Utilise la vision industrielle pour identifier les points de repère et centrer le fauteuil roulant dans le couloir.
CHARHM CDTA, Algérie	1996	C'est un fauteuil à navigation automatique dans un environnement structuré. Il se base sur une carte interne et un système de détection d'obstacles par vision.
COACH, CEA, France	1993	Un fauteuil roulant intelligent conçu pour la navigation dans des environnements dynamiques. Il est équipé de capteurs infrarouges et ultrasonores.
INCH, Yale university, U.S.A	1989	Parmi les premières tentatives utilisant un robot en guise de fauteuil roulant. Il a un système d'évitement d'obstacles basé sur des capteurs à ultrasons.

Tableau I-1 : liste des principaux projets de fauteuils roulants intelligents.

I.3. Interface de commande :

L'interface de commande la plus commune est un joystick. Cependant de nombreux autres types de système de contrôle ont été développés, basés sur des mouvements plus fins (capteur de pression du souffle ou micro-joystick) ou au contraire des mouvements moins maîtrisés (bouton poussoir large pour des personnes présentant des dystonies importantes).

Le système de joystick a complètement été retiré du fauteuil pour être remplacé par un microcontrôleur (arduino uno) et un circuit d'amplification/contrôle permettant de transformer la commande du microcontrôleur (0-5V, quelques milliampères maximum) en un signal ayant la puissance nécessaire pour actionner les moteurs.

I.4. Les méthodes d'entrée :

Les fauteuils roulants intelligents ont été développés pour reconnaître de nouveaux choix aux méthodes de saisie plus « traditionnelles » qui sont associés aux fauteuils roulants électriques (par ex. les joysticks, les interrupteurs pneumatiques).

Les sous-parties suivantes montrent des exemples de méthodes d'entrée utilisées dans les fauteuils roulants électriques et/ou intelligents.

Joystick :

Une étude faite par Fehr et al. a révélé que plus de 95% des utilisateurs des fauteuils roulants électriques optent pour le joystick, Cette recherche a été présentée en 2000. Et Après des années de développement des technologies, les méthodes de saisie les plus sophistiquée sont devenues disponibles [4].

Toutefois, le joystick reste le dispositif de commande le plus courant pour contrôler un fauteuil roulant électrique.

Le joystick permet de mieux reproduire la tâche de conduite en fauteuil, produisant les mêmes actions pour piloter le fauteuil qu'en réel.

Il existe différents types de joystick que l'on peut aisément tester en simulation : les joysticks conventionnels et les joysticks isométriques [5].

Le joystick est l'interface de commande la plus répandue pour piloter des fauteuils roulants dans des EV. Quoique seuls les fauteuils roulants électriques utilisent

un joystick, il reste l'organe de commande le plus intuitif et le moins coûteux. Il n'est cependant pas suffisant pour l'apprentissage de conduite en fauteuil roulant manuel [6].



Figure I-2 : Exemple de joystick de fauteuil roulant

La reconnaissance vocale :

La reconnaissance vocale a été très souvent utilisée pour les fauteuils roulants intelligents (ex., NavChair, SENARIO, TetraNauta [7] et MIT "The MIT Intelligent Wheelchair Project") en raison du coût et de la large disponibilité du matériel de reconnaissance vocale aussi bien en hardware qu'au software [1].

Le contrôle de la reconnaissance vocale applicable aux fauteuils roulants est doté de systèmes de détection et d'évitement des obstacles [8].

Un système de reconnaissance vocale dépendant des systèmes de capteurs ultrasons et infrarouges a été intégré dans un fauteuil roulant. De cette façon, on a pu obtenir un fauteuil roulant qui peut être conduit à l'aide de commandes vocales et avec la possibilité d'éviter les obstacles et, au bas niveau, la détection de trous [9].

Dans son début, le contrôle de la reconnaissance vocale a obligé l'utilisateur à prononcer des mots-clés spécifiques et prédéfinis tel que le vocabulaire du module NavChair [10].

D'autre part, le fauteuil roulant intelligent MIT peut-être contrôler par un vocabulaire naturel.

Cependant, sur le NavChair [10], les capteurs d'obstacles sont intégrés à l'un des logiciels de contrôle protégeant l'utilisateur contre les conséquences des commandes vocales non reconnues (ou mal reconnues) [1].



Figure I-3 : Modèle d'un Fauteuil roulant intelligent à commande vocale

Interface Cerveau-Machine :

Une Interface Cerveau-Machine ou Brain Computer Interface (BCI) est une interface de communication entre un cerveau et un ordinateur (Figure I-4).

À l'aide d'un casque muni d'électrodes placées sur le crâne, il est possible d'acquérir les signaux cérébraux passants à travers la boîte crânienne (appelés signaux EEG) [6].

La première découverte de signaux d'électro-encéphalogramme (EEG) remonte à 1875 [11]. Et Après des années de travail sur l'EEG, ses premières utilisations pratiques faites grâce à un effort de collaboration entre RIKEN, Toyota, L'Institut de recherche de Genesis et d'autres groupes, ont permis de développer un système qui utilise l'EEG pour contrôler un fauteuil chez des personnes dépourvues du contrôle de leurs membres supérieurs tels que les personnes atteintes de myopathie [12], leur épargnant ainsi l'utilisation d'un joystick ou de la main courante. Et plus récemment [13], ils se servent d'Electromyographie (l'activité des muscles faciaux) pour réaliser leur commande tout en se servant [6] de signaux EEG. La complexité des traitements de signaux limite ces interfaces à l'activation des commandes simples de type : tourner à gauche/droite, avancer/reculer et s'arrêter.



Figure I-4 : Interface Cerveau Machine (BCI) pour la 3D

Expressions d'utilisateur :

L'une des méthodes de saisie les plus exotiques qui ont été exécutées, est celle qui inclut la détection du chemin du fauteuil roulant par la vision de l'utilisateur (là où l'utilisateur regarde) grâce à une activité d'électro-oculographie (EOG) [16] (ex., Wheellessly, Siamo) d'où l'utilisation de la vision industrielle (machine vision) pour calculer la position et l'orientation de la tête de l'utilisateur du fauteuil roulant (ex., Université d'Osaka [14], Watson [15]) ; comme mentionné par Kuno et al. [14].

Depuis, les contrôles des entrées dérivées des expressions de l'utilisateur sont généralement bruyants. Les systèmes qui utilisent les méthodes d'expression d'utilisateur nécessitent une intégration à partir des informations obtenues par la détection de l'environnement aussi [8].

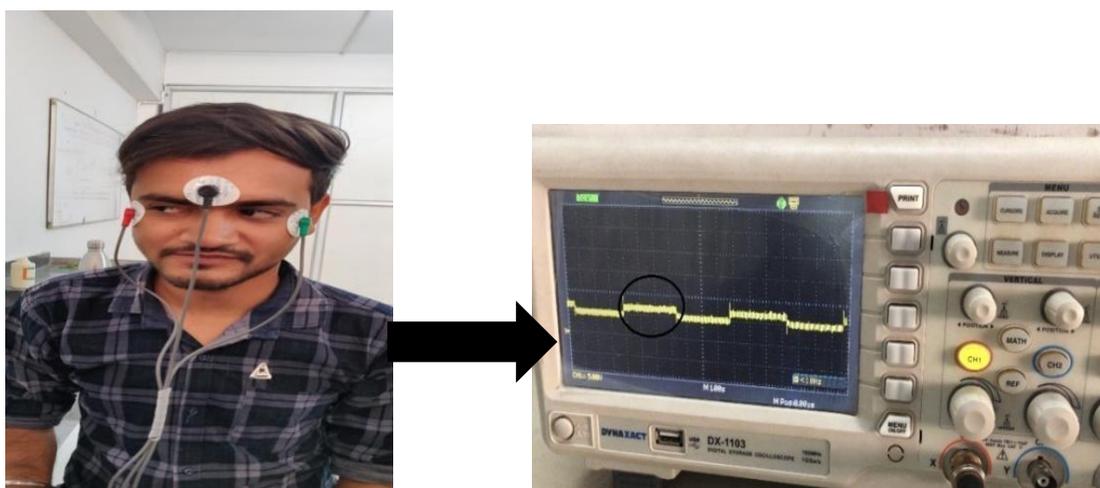


Figure I-5 : Signal EOG acquis en temps réel

Systeme de suivi oculaire :

Certains groupes de recherche ont développé des fauteuils roulants contrôlés par les yeux (Gautam 2014 [17] ; Arai et Ronny 2011 [18] ; Dilok et al. 2014 [19]) qui permettent aux personnes handicapées de contrôler la direction du fauteuil roulant grâce aux mouvements de leur regard.

Certains systèmes utilisent un capteur de proximité pour éviter les collisions. Cependant, le système disponible utilise une énorme caméra attachée à des lunettes [17].

La grande caméra peut gêner la vision de l'utilisateur. De plus, le contrôle du fauteuil roulant par des mouvements oculaires sur de longues distances provoque une fatigue de l'utilisateur.

Des travaux antérieurs [19] ont démontré que la commande du fauteuil roulant basée sur le suivi oculaire pouvait aider l'utilisateur à contrôler la direction du fauteuil et à sélectionner une destination.

Les directions des pupilles pourraient être capturées par une (mini) caméra minuscule (tiny camera) et traitées par une carte intégrée (Raspberry Pi). L'utilisation d'une mini caméra diminue l'obstruction visuelle de l'utilisateur. De plus, l'utilisateur se fatigue moins lorsqu'il au contrôle sur de longues distances en mode de navigation automatisé [20].

Une des conceptions précédentes de fauteuil roulant utilise une caméra CMOS (Caméra PI pour Raspberry PI) dont la précision dépend de l'intensité lumineuse [20]. Mais il y a un problème au niveau du contrôle de l'éclairage, ce qui est impossible car ce fauteuil doit être utilisé dans différents environnements.

Pour éliminer le problème d'intensité lumineuse, ils ont utilisé une minuscule caméra infrarouge (IR). Cette caméra sera attachée au bandeau, comme indiqué sur la figure I-6 [20].

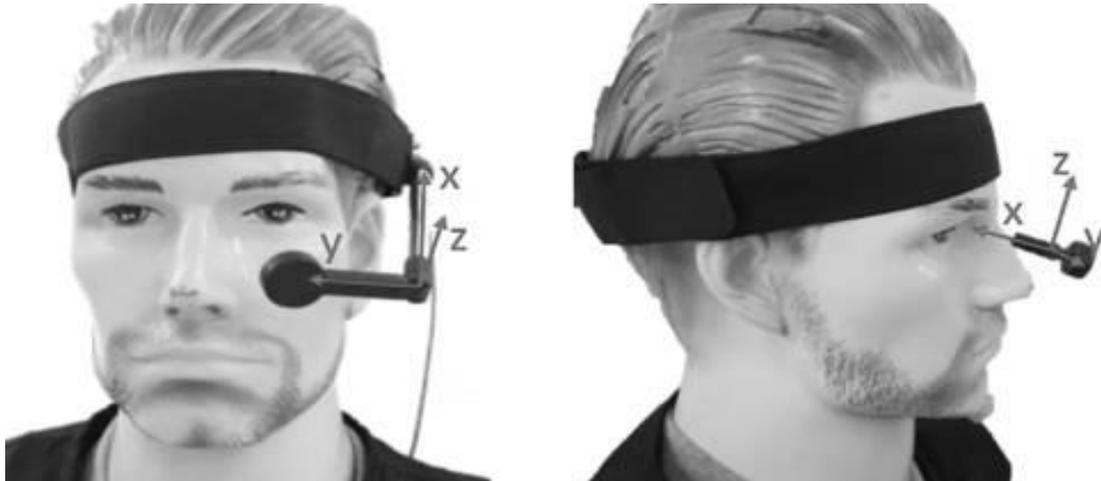


Figure I-6 : Bandeau-caméra intégré ; une minuscule (tiny IR) caméra infrarouge a été montée sur un actionneur réglable fixé au serre-tête.

Les fauteuils roulants intelligents sont d'excellents spécimens pour tester de nouvelles méthodes d'essai car, étant différents des fauteuils roulants standards, les fauteuils intelligents sont dotés d'un ordinateur de bord (onboard computer) avec lequel les capteurs d'entrée peuvent faire interface.

La commande vocale, par exemple, s'est avérée très difficile à appliquer (adaptation) avec succès sur des fauteuils roulants standards [1].

I.5. Les Fauteuils roulants intelligents :

I.5.1. Le projet VAHM :

Le projet VAHM (Véhicule Autonome pour Handicapés Moteurs) [21] a pour but de proposer aux personnes ayant un handicap physique lourd de se déplacer facilement.

Le premier VAHM a été fabriqué à base de robot mobile modifiée. Deuxième VAHM basé sur un fauteuil roulant électrique modifié. VAHM fournissant une navigation autonome et des comportements semi-autonomes. Les modes de décisions sont prises manuellement [1].

Pour cela, il est équipé de capteurs qui collectent les données sur son environnement et d'agents réactifs qui lui permettent des comportements simples (longer les murs, éviter les obstacles, etc.). Ce système présente l'avantage qu'il ne nécessite qu'un seul type d'action de la part de l'utilisateur, mais avec l'inconvénient d'être lent et fastidieux.

Le prototype du robot VAHM qui était initialement implanté sur une base d'un mobile ROBUTER est actuellement transféré sur un fauteuil électrique Power Push (figure I-7) [23].



Figure I-7 : Le prototype du VAHM

I.5.2. Le projet WAD :

Le fauteuil électrique du projet WAD (Wheelchair Attractor Dynamics) dispose d'une aide à la navigation adaptée à un large public.

L'architecture de commande du Projet WAD (Wheel chair Attractor Dynamics) [24] est installée actuellement sur un fauteuil roulant électrique modèle CRUISER de chez INVACARE (figure I-8). Ce fauteuil est équipé d'un système DX de base (joystick et module de puissance) auquel on a rajouté le module DX Key, qui permet un interfaçage aisé via le port parallèle d'un ordinateur portable [24] qui sert à naviguer, soit de manière autonome pour se positionner, soit pour éviter les obstacles pendant que l'utilisateur navigue.

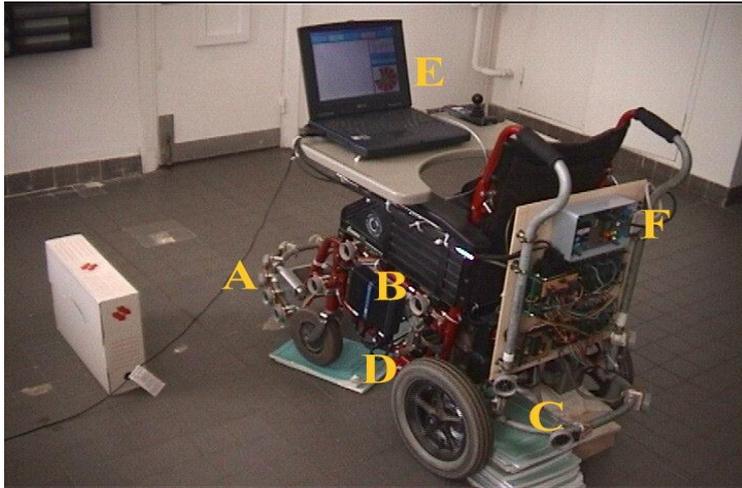


Figure I-8 : le projet de WAD (Wheelchair Attractor Dynamics)

I.5.3. Le projet NavChair :

Le NavChair a été développé à l'Université du Michigan de 1993 à 2002 [8].

Il (figure I-9) a un système de contrôle conçu pour éviter les obstacles, suivre les murs et voyager en toute sécurité dans des environnements encombrés. Il est équipé de douze capteurs ultrasons et d'un ordinateur de bord.

Cet équipement utilise un système de contrôle dans lequel l'utilisateur planifie son itinéraire et indique la destination ainsi que la vitesse de voyage [25].

En utilisant les données (informations) des capteurs ultrasons et des encodeurs, on peut générer un plan d'obstacles où le fauteuil roulant est au centre de ce plan.

Il existe différents modes d'assistance pour les utilisateurs sur le système Navchair [8] ; à savoir :

- Évitement d'obstacles.
- Passage par les portes.
- Suivi des murs.



Figure I-9 : le projet de NavChair, Université de Michigan

I.5.4. Le projet TinMan :

Ils été développé à KIPR, U.S.A de 1994 à 1999. David Miller et Marc Slack de L'Institut de KISS ont développé TinMan I et II. Dans TinMan II (voir figure I-10) un contrôleur supplémentaire de fauteuil roulant est installé entre le joystick et le contrôleur du moteur du fauteuil standard [25].

C'est une série de prototypes de fauteuils roulants intelligents basés sur des fauteuils roulants électriques standards. TinMan peut automatiser une partie de la navigation, notamment les tâches qui paraissent impossibles pour certains [3].



Figure I-10 : TinMan II de l'Institute de KISS

I.5.5. Le Projet Wellesley College, MIT :

Le projet MIT Intelligent (Wheelchair Project) a commencé en 2005 et actuellement en cours et développé [8].

Wheesley est le nom donné à la chaise qu'il a utilisée dans le développement expérimental fait par Holly Yanco, d'abord au Wellesley Collège et maintenant au MIT [21].

Il utilise une caméra, des télémètres ultrasonores et des capteurs infrarouges pour la détection des obstacles. Ceci permet à l'utilisateur de naviguer en sécurité à l'extérieur. Et peut basculer automatiquement entre les modes de navigation l'extérieur et intérieur. Il a été employé pour tester la navigation commandée par les signaux EOG [3].



Figure I-11 : Wheelesley Robot

I.5.6. Le projet de SENARIO, TIDE Programme :

SENARIO [26] était un projet de trois ans qui a démarré en 1994 dans le but de développer un système de navigation autonome en tant que modèle commerciale disponible sur le marché. Le projet a été pris en charge par le programme TIDE de la CE. Le système se compose de cinq sous-systèmes [25] :

- Mitigation des risques,
- Détection,
- Positionnement,
- Panneau de commande,
- Contrôle de puissance.

Le projet a réussi à développer toutes les technologies nécessaires pour répondre à l'ensemble des problèmes (planification de trajectoire globale, planification de trajectoire locale - évitement d'obstacles, localisation de robots, etc.) [26]. Le système a deux modes de fonctionnement : Le mode Teach et le mode Run.

En mode Teach, l'utilisateur sélectionne les chemins souhaités dans un diagramme topologique.

En mode Run (sur un chemin prédéfini) l'utilisateur sélectionne un chemin et le système le suivra en fonction des informations stockées et des informations obtenues

pendant le mode apprentissage. Ce système a été construit sur un fauteuil roulant disponible (Sprint, Meyra) [25].



Figure I-12 : Le SENARIO autonome fauteuil roulant

I.5.7. Le projet d'OMNI, Fauteuil roulant d'omnidirectionnel :

L'OMNI a été développé à l'Université de Hagen en Allemagne de 1995 à 1999 [1]. Le projet OMNI vise à augmenter le mouvement des fauteuils roulants dans certaines limites à proximité immédiate de la maison, et en plus explorer des stratégies de contrôle partagé (shared control) [26].

C'est un fauteuil dont la principale caractéristique est l'omnidirectionnalité. Il est équipé de roues particulières appelées « mecanum » et d'un ensemble de roulettes à roulement libre orientées à 45° et fixées sur le pourtour des roues [26].

Il utilise des capteurs à ultrasons et infrarouges pour l'évitement d'obstacles. Il reçoit les commandes à partir d'un joystick 3D [3]. Son architecture repose sur une structure hiérarchique composée d'un ensemble de modules activés par l'utilisateur à l'aide d'une interface textuelle ou iconique. Selon ses capacités physiques, l'utilisateur va choisir la fonction qu'il juge la plus propice au contexte telle que :

- Évitement d'obstacles.
- Mode de fonctionnement spécifique à la tâche (suivi de mur, passage de porte).
- Navigation autonome.
- Planification de trajectoire

Ainsi, Omni peut effectuer n'importe quelle combinaison de mouvement vers l'avant, sur les côtés et en rotation.



Figure I-13 : OMNI, Fauteuil roulant d'omnidirectionnel

I.5.8. Le projet FRIEND :

FRIEND (Functional Robot Arm with User Friendly Interface for Disabled People) est un prototype de laboratoire qui a été développé par l'Institution d'Automation de l'Université de Brême en Allemagne en 2001 [26].

C'est un fauteuil roulant électrique équipé d'un bras de robot à six degrés de liberté. Le fauteuil et son bras sont commandés par un ordinateur de bord. Un écran plat et un système de reconnaissance vocale (ViaVoice de IBM) sont utilisés comme interface pour l'utilisateur. La détection des obstacles lors de la navigation se fait à l'aide d'un système de stéréo vision. Le bras manipulateur peut être commandé vocalement, ou en désignant un objet capté par la caméra [3]. C'est ainsi qu'il effectue la prise automatique de cet objet.



Figure I -14 : FRIEND Robot

I.5.9. Le projet SPAM :

Le SPAM (Smart Power Assistance Module) a été développé à l'Université du Pittsburgh, U.S.A, de 2003 à 2005 [1].

SPAM (Smart Power Assistance Module) est un module montable sur fauteuil manuel pour les non-voyants. Plusieurs types de capteurs peuvent être utilisés comme les ultrasons et les infrarouges. Il a aussi un capteur de couple pour mesurer la force manuelle appliquée sur la roue. Le système permet de rajouter un couple pour assister la conduite du fauteuil et donc éviter les risques de collision et de fatigue de la personne. L'utilisateur reçoit un retour haptique au niveau des roues, ce qui lui permet d'avoir une représentation de l'environnement proche [3].



Figure I-15 : Le module SPAM placé sur un fauteuil manuel

I.6. Conclusion :

Dans ce chapitre, nous avons présenté les fauteuils roulants électriques intelligents ainsi que des généralités concernant les origines et l'évolution de ce domaine. En particulier, nous avons abordé les différentes méthodes d'entrée, (i.e. Joystick, La reconnaissance vocale, Système de suivi oculaire, etc.).

En outre, Nous avons détaillé les différents Projets réalisés sur les fauteuils roulants intelligents comme, par exemple, le projet VAHM, le projet NavChair et autres.

Les fauteuils roulants intelligents resteront un terrain fertile pour la recherche technologique pour de nombreuses années à venir.

Chapitre II :

Détection et Reconnaissance du visage et des yeux



II.1. Introduction :

La reconnaissance des visages et des yeux est un domaine très actif dans la vision par ordinateur, dans la biométrie et en robotique et joue un rôle important dans de nombreuses applications tel que le contrôle d'accès aux bâtiments et aux magasins, l'identification et la surveillance de suspects ainsi que la commande des fauteuils roulants. Notre système fera intervenir les modalités du visage et de l'iris.

Cet aperçu sera organisé comme suit : Tout d'abord, nous présenterons l'architecture du système et puis expliquant la bibliothèque d'OpenCV et sa fonctionnalité sur les traitements d'images et des vidéos en temps réel. Nous détaillerons en particulier les méthodes des traitements des visages et des yeux. Enfin nous exposerons les méthodes de détection et de reconnaissance du visage et des yeux.

II.2. Architecture du système :

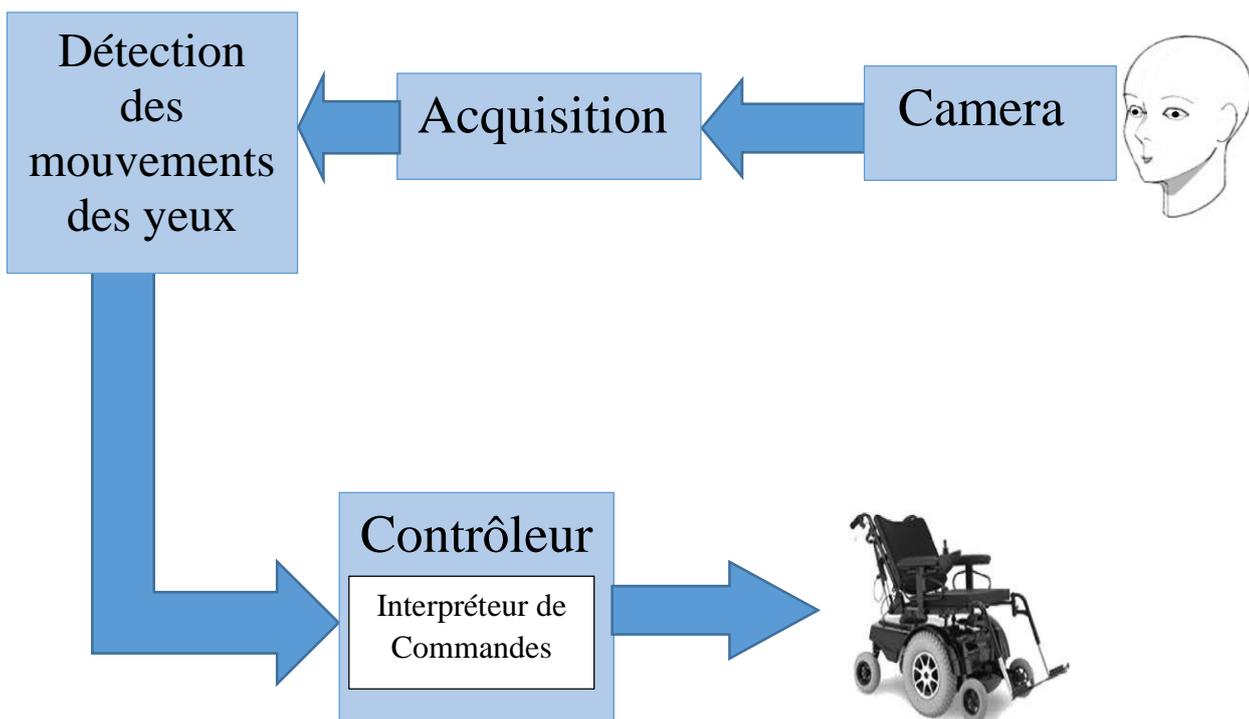


Figure II-1 : architecture du système proposé

Comme le montre dans la structure du système sur la figure II-1, une caméra est montée devant le visage pour capturer l'image des yeux (gauche et droit).

La partie principale de ce système est l'algorithme de détection de visage et des mouvements oculaires des yeux. Une bibliothèque ouverte de vision par ordinateur (OpenCV) contenant de nombreux algorithmes de vision par ordinateur est utilisée pour la reconnaissance du visage et des yeux.

II.3. OpenCV :

II.3.1. Histoire :

OpenCV (Open-source Computer Vision) est une bibliothèque graphique libre, conçue pour le traitement d'image en temps réel. Il est compatible avec Windows, Linux et MacOS.

La bibliothèque OpenCV (Open Computer Vision), officiellement lancée en août 1999 comme une initiative de recherche Intel. Elle est développée par Intel pour optimiser les applications gourmandes en temps processeur. Cette bibliothèque est distribuée sous licence BSD [26].

Cela faisait partie d'une série de projets tel que l'affichage d'un mur en 3 dimensions, reconnaissance de texte (text recognition), correspondance de surface opencv (surface matching) [26]. Aujourd'hui elle est couplée avec d'autres bibliothèques de Deep Learning afin de proposer des applications de plus en plus efficaces.

Maintenant, c'est la bibliothèque la plus populaire pour Computer Vision et Machine Learning (l'apprentissage automatique).

Au début, c'était un ensemble de fonctions de la bibliothèque C pour le traitement d'images et la computer vision. Maintenant elle a des liaisons C ++, Python, Java, MATLAB, etc. [27].

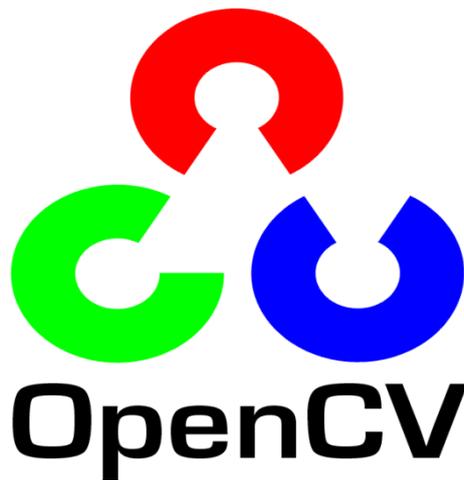


Figure II -2 : Logo D'OpenCV

Les objectifs de base du projet consistent à :

- Faire des recherches sur la vision par ordinateur en vue de fournir un logiciel libre et optimisé.
- Établir une infrastructure commune s'appuyant sur les développeurs pour obtenir un code plus lisible et transférable.
- Continuer à développer en rendant le code portable et permettre des performances optimisées gratuites avec une licence qui est libre de toutes contraintes commerciales.

Voici une liste des versions d'OpenCV téléchargeables pour les systèmes d'exploitation Linux et Windows [28] :

1. OpenCV 1.0;
2. OpenCV 1.1pre1;
3. OpenCV 2.1;
4. OpenCV 2.2;
5. OpenCV 2.3;
6. OpenCV 2.3.1;
7. OpenCV 3.0.0;

8. OpenCV 3.2.0;

9. OpenCV 3.6.0;

10. OpenCV 4.1.0;

II.3.2. Fonctionnalités :

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes à partir de données brutes pour aller jusqu'à la création d'interfaces graphiques basiques [30].

a) Traitement d'images :

Elle propose la plupart des opérations classiques en traitement bas niveau des images [30] :

- Lecture, écriture et affichage d'une image.
- Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs.
- Lissage, filtrage.
- Binarisation, segmentation en composantes connexes.
- Morphologie mathématique.

b) Traitement des vidéos :

Elle met également à disposition de l'utilisateur quelques fonctions d'interfaces graphiques, comme les curseurs à glissière, les contrôles associés aux événements souris, ou bien l'incrustation de texte dans une image.

Le traitement vidéo est une technique très importante utilisée de nos jours par la police pour trouver ou localiser des suspects après.

La prochaine fois que vous visitez une petite boutique de votre quartier, ne soyez pas surpris de voir les caméras de vidéosurveillance installées à l'intérieur. C'est pour aider les petits commerçants à coincer les voleurs à l'étalage.

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'art de vision des ordinateurs tels que [30] :

Chapitre II : Détection et Reconnaissance du visage et des yeux

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra) ;
- Détection de droites, de segments et de cercles par Transformée de Hough ;
- Détection de visages par la méthode de Viola et Jones ;
- Cascade de classifieurs boostés ;
- Détection de mouvement, historique du mouvement ;
- Poursuite d'objets par mean-shift ou Camshift ;
- Détection de points d'intérêts ;
- Estimation de flux optique (tracker de Kanade-Lucas) ;
- Triangulation de Delaunay ;
- Diagramme de Voronoi ;
- Enveloppe convexe ;
- Ajustement d'une ellipse à un ensemble de points par la méthode des moindres carrés.

Et pour avoir une brève idée du traitement vidéo, nous allons vous montrer un exemple sur la conversion du flux vidéo couleur en temps réel de votre webcam en niveaux de gris et l'afficher dans une fenêtre (Figure II-3) [27].

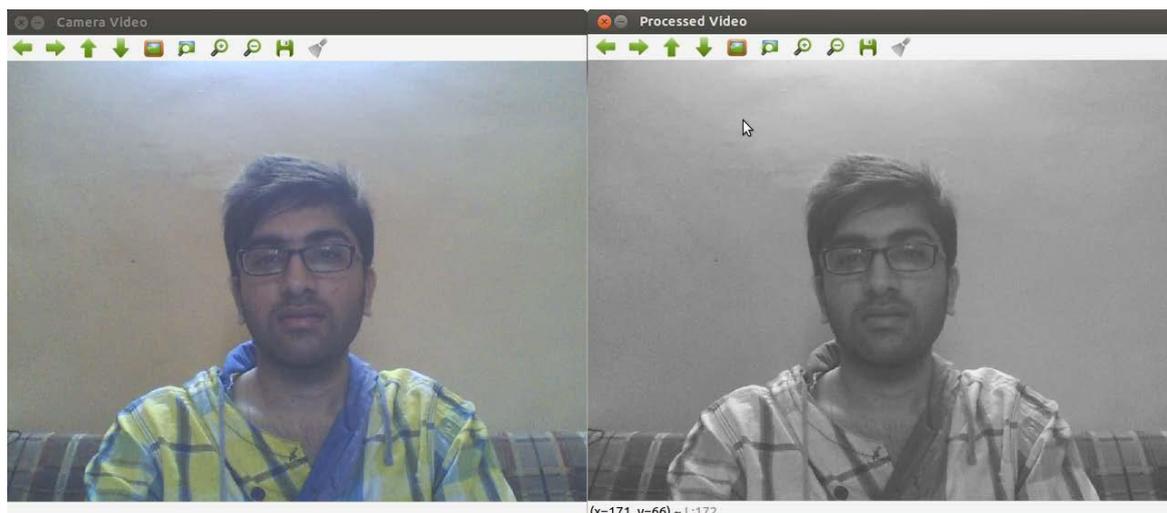


Figure II-3 : La figure suivante montre la comparaison entre une image colorée et une image en niveaux de gris

II.3.3. Structure de la librairie OpenCV :

La librairie OpenCV se présente de la manière suivante [31] :

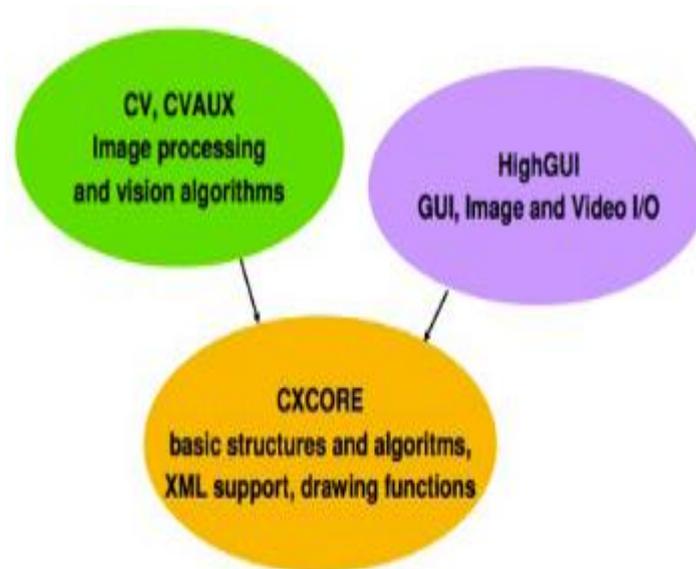


Figure II-4 : Structure de la bibliothèque d'OpenCV

- a) CV & CVAUX : Cette partie de la bibliothèque permet le traitement d'images.
- b) HIGHGUI : En plus du traitement d'images, celle-ci permet le traitement des vidéos en temps réel.
- c) CXCORE : Cette partie regorge de nombreuses fonctions de dessin.

II.3.4. Installation d'OpenCV :

L'installation d'OpenCV dépend de deux facteurs essentiels [29] :

- Le système d'exploitation sur lequel on travaille.
- Le langage de développement qu'on souhaite utiliser.

Dans le cas du système Windows avec Python comme langage, il faut d'abord installer de Numpy puis OpenCV depuis un terminal.

Les étapes à suivre pour l'installation d'OpenCV :

- Etape 1

La première étape consiste à télécharger les fichiers binaires d'Opencv et de Numpy. Choisissez la version d'Opencv et de Numpy qui soit compatible sur le Python.

Chapitre II : Détection et Reconnaissance du visage et des yeux

Pour notre part installé python version **3.6.5**, et téléchargé :

- Pour OpenCV: `opencv_python-4.2.0+contrib-cp36-cp36m-win_amd64.whl`
- Pour Numpy: `numpy-1.14.3+mkl-cp36-cp36m-win_amd64.whl`

- Etape 2

Exécutez terminal « l'invite de commande » de Windows. Vous pouvez taper « CMD » dans la barre de recherche pour le trouver. Puis ouvrir le terminal comme illustré dans la figure II-5.



Figure II -5 : terminal du système.

- Etape 3

Maintenant installez Opencv à l'aide de pip :

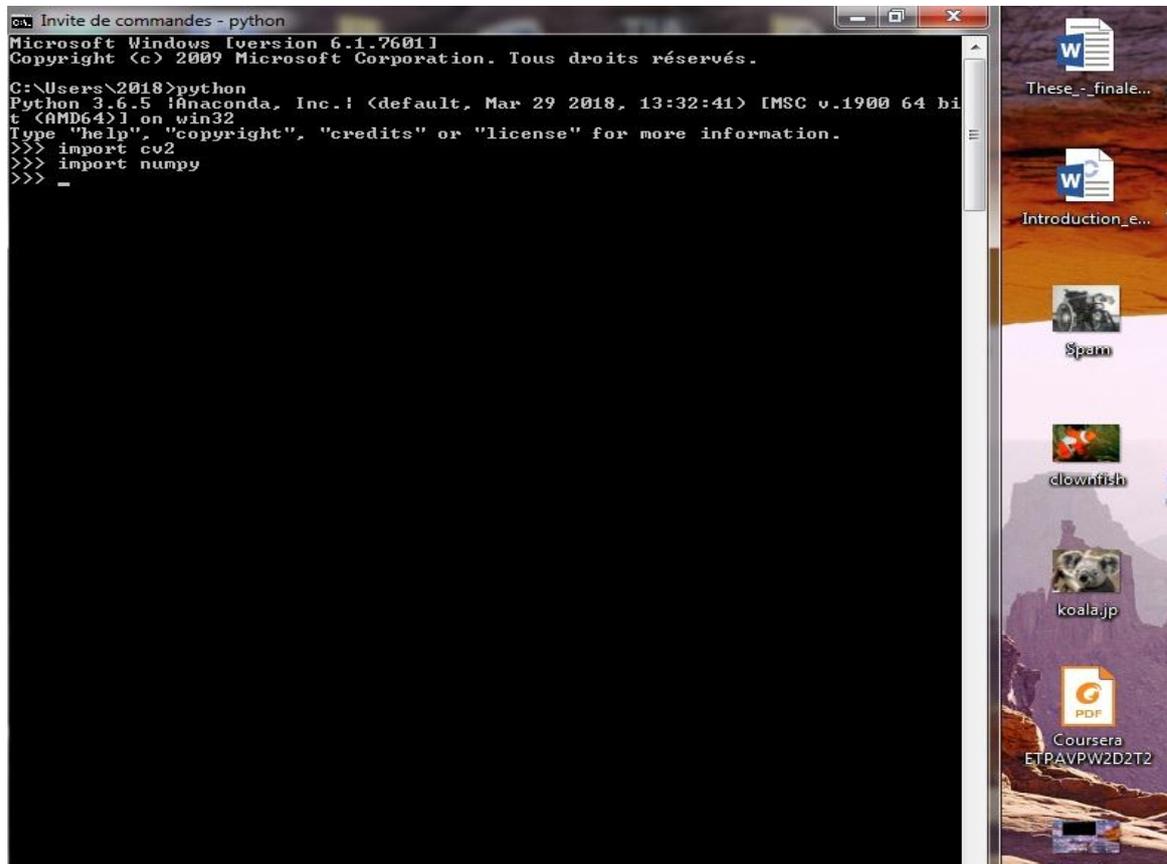
```
python pip -m install opencv_python-4.2.0+contrib-cp36-cp36m-win_amd64.whl
```

Et enfin Numpy :

```
python pip -m install numpy-1.14.3+mkl-cp36-cp36m-win_amd64.whl
```

- Etape 4

L'installation est terminée. Nous pouvons maintenant le tester en exécutant terminal et taper python et en important les bibliothèques cv2 (pour opencv) et numpy. S'il n'y a aucune erreur, cela signifie qu'il est installé avec succès.



```
cmd: Invite de commandes - python
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\2018>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit
 (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> import numpy
>>> _
```

Figure II -6 : tester cv2 (pour opencv) et numpy

II.4. Les méthodes des traitements du visage :

II.4.1. La détection du visage :

La détection du visage est un domaine de la vision par ordinateur consistant à détecter un visage humain dans une image numérique. C'est un cas spécifique de détection d'objet, où l'on cherche à détecter la présence et la localisation précise d'un ou de plusieurs visages dans une image.

Les challenges de la détection des visages peuvent être attribués aux facteurs suivants [32] :

- La pose.

- Présence ou absence de composantes structurelles.
- Expression faciale.
- Occlusion par d'autres visages.
- Orientation de l'image.
- Conditions d'illumination et qualité de l'image.

Dans cette partie, nous exposons les méthodes existantes pour détecter des visages. Les méthodes de détection de visages peuvent être réparties en quatre catégories [32] :

A. Approches basées sur les connaissances acquises :

Ces méthodes basées sur des règles encodent la connaissance humaine de ce que constitue un visage typique.

Généralement, les règles capturent les relations entre les caractéristiques du visage. Ces méthodes sont conçues principalement pour la localisation du visage.

B. Approches basées sur des caractéristiques invariantes :

Les algorithmes développés visent à trouver les caractéristiques structurales existantes même si la pose, le point de vue, ou la condition d'éclairage changent. Puis ils emploient ces caractéristiques invariables pour localiser les visages. Ces approches sont utilisées principalement pour la localisation du visage.

C. Approches basées sur le « Template-matching » :

Plusieurs modèles standard d'un visage sont stockés pour définir le visage soit "manuellement", soit paramétrés à l'aide de fonctions.

L'idée est de calculer la corrélation entre l'image d'entrée et les motifs stockés. Ces méthodes ont été utilisées à la fois pour la localisation et la détection des visages.

D. Approches basées sur l'apparence :

Ces approches appliquent généralement des techniques d'apprentissage automatique.

Chapitre II : Détection et Reconnaissance du visage et des yeux

Ainsi, les modèles (ou Template) sont appris à partir d'un ensemble d'images représentatives de la variabilité et de l'apparence du visage de l'aspect facial. Ces modèles appris sont ensuite utilisés pour la détection. Ces méthodes sont principalement conçues pour la détection des visages.

Le **Tableau II -1** Résume les algorithmes et les études représentatives pour la détection des visages parmi ces quatre catégories [32].

Approche	Les Travaux Représentatifs
Approches basées sur les connaissances acquises	Méthodes multirésolution basées sur des règles.
Approches basées sur des caractéristiques invariantes <ul style="list-style-type: none"> – Les caractéristiques du visage. – Texture. – La couleur de la peau. – Des caractéristiques multiples. 	Regroupement des contours (edges). Matrice dépendance des niveaux d'espace gris de motif du visage (SGLD). Mixture de gaussian. L'intégration de la couleur de la peau, la taille et la forme (shape).
Approches basées sur le « Template-matching » <ul style="list-style-type: none"> – Des modèles de visage prédéfinis. – Le template déformable. 	Shape template. Modèle de la forme (shape) active (ASM).
Approches basées sur l'apparence <ul style="list-style-type: none"> – Interface propre (Eigenface). – Distribution-based – Réseau de neurones – Support des machines à vecteurs (SVM) – Réseau Sparse de Winnows – Naive Bayes Classifier – Modèle de Hidden Markov (HMM) – Approche d'Information-Theoretical – Inductive Learning 	Décomposition et regroupement des vecteurs propres (Eigenvector). Distribution gaussienne et perceptron multilayer. Ensemble des réseaux des neurones et des schémas d'arbitrage. SVM avec noyau (kernel) polynomial. Des statistiques jointes sur l'apparence et la position locales. Des statistiques d'ordre supérieur avec HMM. Kullback information relative.

Tableau II-1 : Catégorisation des méthodes de détection des visages.

II.4.2. La Système de Reconnaissance de Visage :

La reconnaissance de visages est un gros domaine tellement intéressant, qu'il a attiré les chercheurs des différents domaines : psychologie, identification de modèles, réseaux de neurones, vision d'ordinateur et infographie [33].

Pour la partie concernant la reconnaissance faciale, nous allons faire l'étude des méthodes 2D et 3D.

Les méthodes de reconnaissance faciales en 2D peuvent être réparties en trois grandes catégories, les méthodes globales (ou holistiques), les méthodes locales et les méthodes hybrides [34]. Nous exposerons aussi les méthodes de reconnaissance de visage 3D.

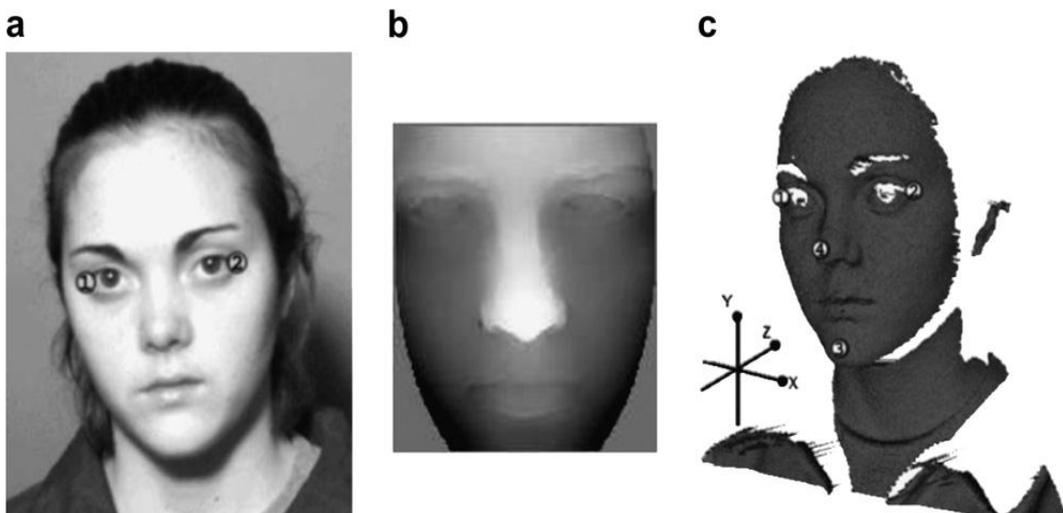


Figure II-7 : (a) Image texture (b) Image 2.5D (c) Image 3D

II.4.2.1. Reconnaissance de visages 2D :

A. Méthodes globales :

Ces méthodes utilisent l'ensemble de la région du visage comme entrée crue (qui peuvent être vues comme des matrices de valeurs pixels) d'un système de reconnaissance.

Il n'est pas nécessaire de repérer certains points caractéristiques du visage (comme les centres des yeux, les narines, le centre de la bouche, etc.) sauf pour normaliser les images.

L'une des représentations les plus utilisées de la région du visage est l'image propre [35] [36], qui sont basées sur des techniques d'analyse statistique bien connues (principal component analysis, PCA) [33].

A. Méthodes locales :

Généralement, dans ces méthodes, en détectant certains points ou traits caractéristiques d'un visage puis en les comparant avec des caractéristiques locales extraites d'autres visages tels que les yeux, le nez et la bouche et en construisant un espace de caractéristiques local et des statistiques locales (géométriques et/ou d'apparence), on utilise des classificateurs structurels [33].

B. Méthodes hybrides :

Les méthodes hybrides consistent à combiner entre des méthodes globales et locales comme le système de perception humaine utilise tous les deux, les caractéristiques locales et toute la région faciale pour reconnaître un visage. Un système de reconnaissance automatique doit utiliser les mêmes caractéristiques [33].

On peut affirmer que ces méthodes peuvent potentiellement offrir le meilleur des deux types de méthodes.

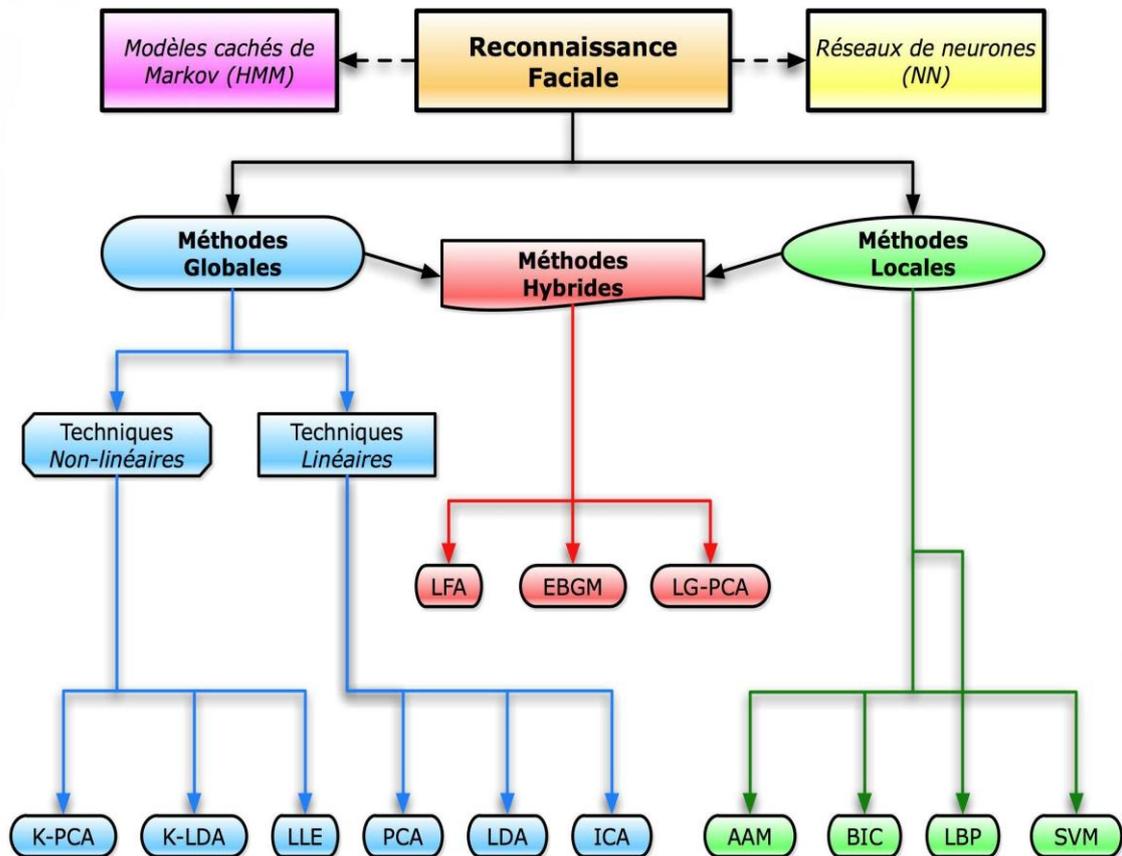


Figure II-8 : Une classification des algorithmes principaux utilisés en reconnaissance faciale.

II.4.2.2. Reconnaissance de visages 3D :

En général, pour la reconnaissance 3D des visages, il est prévu une classe de méthodes qui travaillent sur un ensemble de données tridimensionnelles, représentant à la fois le visage et la forme de la tête comme des données de gamme ou des maillages polygonaux. Le principal avantage des approches basées sur la 3D est que le modèle 3D conserve toutes les informations sur la géométrie du visage [37].

De plus, la reconnaissance des visages en 3D devient une évolution du problème de la reconnaissance en 2D, car une représentation plus précise des caractéristiques du visage entraîne un pouvoir de discrimination potentiellement plus élevé. Dans un modèle de visage 3D, les caractéristiques faciales sont représentées par des courbures locales et globales qui peuvent être considérées comme la véritable signature identifiant un individu.

A. Systèmes d'acquisition 3D :

La technique basée sur la reconnaissance des visages en 3D doit posséder plusieurs propriétés telles que la robustesse par rapport aux variations d'éclairage ainsi que la position, la rotation et la mise à l'échelle du modèle original dans un cadre de référence absolu. Malheureusement, la technique 3DFR n'atteint pas l'ensemble de ces objectifs.

Par exemple, aligner un maillage polygonal 3D dans un cadre de référence absolu pourrait être coûteux en termes de calcul, et les méthodes existantes ne sont pas toujours convergentes. En outre, l'affirmation selon laquelle l'acquisition de données 3D (laser et scanners de lumière structurée) est indépendante de la lumière, n'est pas totalement vraie ; les capteurs 3D pourraient être affectés par une source de lumière puissante ou par des surfaces réfléchissantes, on peut donc affirmer que différentes sources de lumière pourraient générer des ensembles de données 3D très différents [37].

En fait, le visage est acquis par étapes successives [37] :

- 1) Les données 3D sont alignées selon le système de référence de la caméra, de sorte que l'axe (Z) se trouve le long de l'axe optique ;
- 2) Les données 3D provenant de différents points de vue sont soumises à un processus de fusion ;
- 3) Les données 3D sont optimisées selon certains critères. Enfin, le maillage polygonal 3D est construit à partir des nuages de points 3D en utilisant un algorithme de génération de maillage.

B. Approches modèle :

Le système de reconnaissance des visages présenté par Blanz et Vetter combine des modèles 3D déformables avec une simulation infographique de projection et d'illumination.

Ces approches construisent, à partir des points 3D, des modèles de visages qu'elles utilisent par la suite pour la reconnaissance. Blanz et al. [39] [40] ont proposé une méthode basée sur un modèle 3D « morphable » du visage [38].

Peu d'articles sur ce sujet ont été publiés, bien que la recherche sur la 3D sur la reconnaissance des visages a commencé dans les années 80. De nombreux critères peuvent être adoptés pour comparer les algorithmes de reconnaissance des visages en 3D existants en tenant compte du type de problèmes qu'ils traitent ou de leurs propriétés intrinsèques. En effet, certains ne donnent de très bons résultats que sur les visages à l'expression neutre, tandis que d'autres essaient également de traiter les changements d'expression [37].

C. Approches 3D :

Cette sous-section explore plusieurs méthodologies qui travaillent directement sur des ensembles de données en 3D. Le premier problème concernant la reconnaissance des visages en 3D est d'établir un alignement correct entre deux surfaces de visage [37].

Ils ont construit des modèles 3D de 100 sujets en utilisant un système stéréo ; chaque sujet a été acquis dans sept poses différentes à l'intérieur de degrés par rapport à la vue frontale [37].

D. Approches 3D+2D :

Les approches multimodales combinent des informations provenant d'une image 2D ainsi que d'un modèle 3D des visages. Récemment, Chang et ses collègues (2003) ont étudié les améliorations possibles que la biométrie des visages 2D peut recevoir en intégrant également la 3D. La méthode qu'ils ont proposée consiste à effectuer séparément l'ACP sur les images d'intensité et de distance, puis à combiner les résultats obtenus par les deux stratégies pour obtenir une réponse globale du système [37].

Ils rapportent divers résultats sur un ensemble de données recueillies auprès de 62 sujets avec 13 maillages 3D distincts et des cartes de texture 2D en tenant compte de plusieurs expressions faciales et poses. Le pourcentage de correspondances correctes et de rejets corrects sont utilisés comme mesures de performance. Dans le cas de la pose frontale, les résultats montrent que l'utilisation à la fois de la texture et de la forme améliore les performances, tandis qu'un pourcentage de reconnaissance correcte allant de 66,5 % à 100 %, en fonction de plusieurs poses et expressions [37].

II.5. Les méthodes de traitement des yeux :

II.5.1. La détection des yeux :

La détection des yeux est un autre élément important pour la commande de fauteuil roulant, car c'est l'état des yeux que l'on doit analyser pour avoir une idée du niveau de mouvement des yeux de la personne. Évidemment dans notre cas, pour contrôler les mouvements des yeux il faut d'abord avoir les yeux ouverts.

La détection des yeux, ou des centres des yeux, est l'une des étapes les plus importantes dans le processus de reconnaissance [41]. En fait, la taille du visage est définie par la distance intraoculaire ainsi que l'orientation par l'alignement des yeux ; puisque la détection des yeux va permettre de normaliser ensuite les images ce qui est indispensable avant tout algorithme de mise en correspondance. Pour cela, on trouve dans les publications plusieurs méthodes de détection des yeux. De l'efficacité de ces méthodes dépend la performance de l'algorithme de reconnaissance.

Plusieurs approches ont été utilisées pour détecter la position des yeux. Ces approches peuvent être classées en trois catégories [42] [41] :

- Approche basée sur l'apparence : Cette méthode détecte les yeux en se basant sur l'intensité lumineuse de l'œil. On utilise la projection du niveau de gris pour détecter les yeux [43].
- Approche basée sur les modèles : Cette approche construit un modèle (Template) déformable de l'œil qui est mis en comparaison avec l'œil de l'image de test.
- Approche basée sur les traits de l'œil : On s'intéresse dans cette méthode aux traits caractéristiques des yeux tels que les coins des yeux. Dans cette catégorie [44], les auteurs ont mis en place un algorithme de détection à partir de deux filtrages : un filtrage de **Gabor** pour la détection de l'œil et un second non linéaire pour la détection des coins des yeux.

II.5.2. Reconnaissance des yeux :

L'ophtalmologiste américain Frank Burch est la première personne qui a proposé l'idée d'utiliser la texture de l'iris en 1936, qu'offre cette méthode, lors d'une conférence pour l'American Academy of Ophthalmology [34].

En 1987 deux ophtalmologistes américains, Leonard Flom et Aron Safir, propose le concept que deux iris de deux personnes différentes ne peuvent pas être identiques et en voulant développer un système d'identification opérationnel, les deux ophtalmologistes engagent le Professeur John Daugman, professeur à l'université de Harvard, (et maintenant à l'Université de Cambridge). La collaboration entre Daugman, Safir et Flom avait abouti à un prototype fonctionnel breveté en 1994 [45] [46] [47].

En 1998 l'approche "zero-crossings", proposée par Boles et Boashash ; cette étude démontre un algorithme basé sur une transformée unidimensionnelle par ondelettes orthogonales [48].

En 2000 Zhu et al. Ont fait des études comparatives entre les ondelettes réelles de Gabor non orthogonales et l'analyse par ondelettes orthogonales de Daubechies d'ordre4 ("db4") [49].

En 2002 Tisse et al. [50], utilisent la transformée de Hough qu'ils combinent avec l'opérateur de Daugman, ensuite l'opérateur intégral-différentiel est appliqué pour localiser avec précision les deux frontières de l'iris.

Le Tisse et al. Système de reconnaissance d'iris, comprend également un procédé robuste pour la pupille, détection du centre à l'aide d'un Hough décomposé par gradient Transform (Transformée de Hough suivant les Gradients Décomposés), qui profite de son forme circulaire pour améliorer le taux de réussite de la localisation de l'iris [50].

En 2004 Rydgren et al. [51] ont proposé une approche pour la reconnaissance de l'iris. Ils utilisent une transformée en paquets d'ondelettes sur trois niveaux. L'algorithme permet d'identifier correctement tous les iris authentiques et de rejeter tous les imposteurs. Le nouvel algorithme représente une alternative intéressante aux méthodes basées sur Gabor [51].

En 2006 Chen et al, [52] ont étudié les effets de la qualité de l'image de l'iris sur la performance de la reconnaissance de l'iris suivant l'approche de Daugman.

Des mesures de qualité locales basées sur des bandes annulaires concentriques dans la région de l'iris sont développées en utilisant des ondelettes 2D [52].

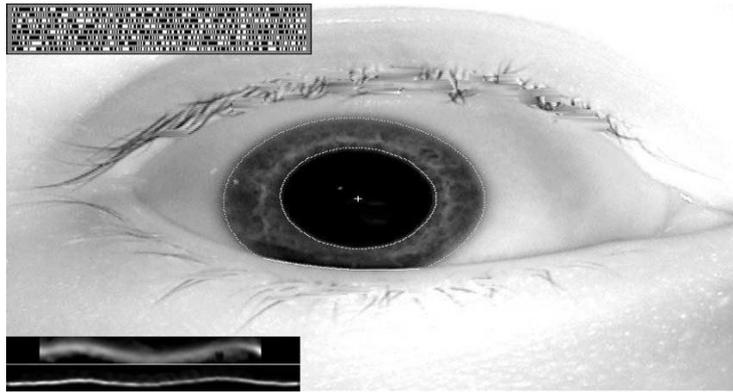


Figure II-9 : Illustration de « snakes », [ou les contours flexibles dans la méthode de daugman]

En 2008 M. Nabti et Bouridane [53-54], ont introduit une nouvelle approche de détection de contours multi-échelles pour la segmentation de l'iris. Dans l'approche qu'ils proposent, il a été démontré que les techniques multi-échelles combinées sont efficaces et robustes pour analyser les textures de l'iris et pour obtenir une haute performance du système.

Ces méthodes permettent de détecter et de modéliser fidèlement ces limites, quel que soit leur forme, et de définir sur cette base un système de coordonnées plus souple et plus généralisé.

En 2011 Tsukada et al. [55] proposent une méthode d'estimation du regard pour surmonter la tâche difficile de l'extraction de l'iris. Les résultats expérimentaux sur des données réelles montrent que la méthode est précise et compétitive par rapport aux produits commerciaux qui utilisent les rayons infrarouges. Cette méthode améliore la robustesse par rapport aux autres méthodes utilisant des rayons visibles.

Ils ont ensuite **en 2012** proposé une méthode pour construire le modèle 3D de l'œil automatiquement [56].

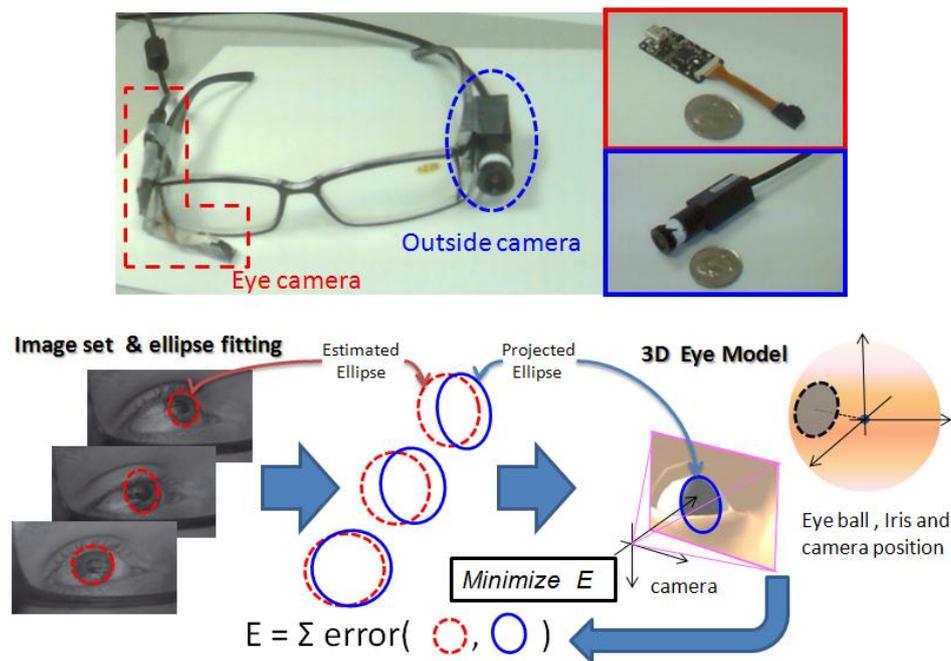


Figure II-10 : l'ensemble des appareils utilisé dans cette méthode

En 2017 S. Viriri et J. Tapamo [57] ; reconnaissance de l'iris basé sur des sommes cumulées et des méthodes de vote majoritaire.

Ils ont présenté deux approches efficaces basées sur un vote majoritaire pour créer un modèle d'iris unique pour un individu ; et un algorithme basé sur l'analyse des changements de gris des sommes cumulées (cumulative sums grey) est utilisé pour extraire les caractéristiques de la texture de l'iris pour la création du gabarit.

Les résultats expérimentaux sur les bases de données d'iris CASIA et UBIRIS montrent que l'approche est efficace et efficiente, avec un taux de reconnaissance de 99,6%.

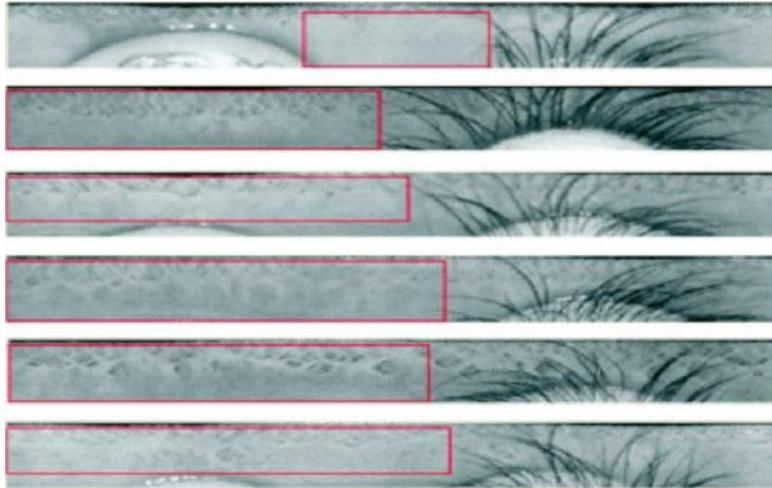


Figure II-11 : Système de reconnaissance d'iris
Régions d'intérêt détectées (ROI)

Enfin, on trouvera une présentation de quelques méthodes actuelles en reconnaissance de l'iris dans [47]. La dernière technologie, cependant, permet de capturer des informations sur l'iris également à distance et sans coopération spécifique de l'individu.

II.6. Conclusion :

Dans ce chapitre, nous avons présenté les technologies utilisées dans les systèmes biométriques pour l'identification de visages et des yeux d'individus ; avec un aperçu des techniques et des méthodes de détection et de reconnaissance de visages et des yeux.

Nous avons commencé par le passage d'utilisation de la bibliothèque OpenCV. En premier lieu, nous avons passé en revue des méthodes de détection de visage basée sur les techniques d'acquisition de données faciales et nous avons illustré les approches de reconnaissance de visages 2D et 3D.

Ensuite [58], nous avons passé en revue des thèmes sur l'iris dans plusieurs domaines de recherche actifs, tels que l'acquisition d'image, l'évaluation de la qualité, la segmentation, la réduction du bruit, la fonctionnalité extraction, la correspondance de code d'iris, ainsi que la recherche dans une grande base de données sur les applications, les évaluations et les performances dans des conditions variables et multibiométrie.

Chapitre II : Détection et Reconnaissance du visage et des yeux

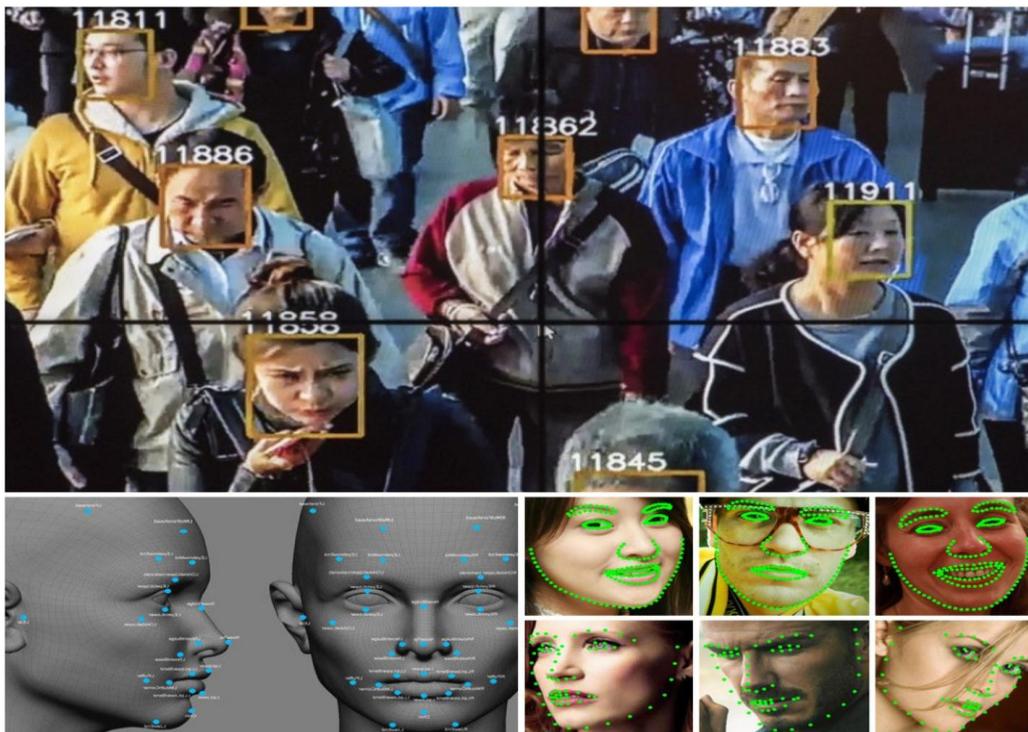
La reconnaissance de visage et de l'iris est une méthode biométrique fiable actuellement utilisée par plusieurs applications en temps réel tels que les guichets automatiques, authentification de prisonniers, opérations bancaires, contrôles aux frontières et aéroports, et aussi dans la vidéosurveillance et robotique.

Actuellement, on utilise les systèmes de la reconnaissance de visage et de l'iris plus que d'autres dans de nombreuses applications suite à la pandémie mondiale Covid-19.

Chapitre III :

Résultats de l'étude

Et Système de Commande



III.1. Introduction :

Dans ce chapitre, nous présentons le projet général de notre expérience, les procédures que nous avons suivies lors de l'expérimentation de méthode de détection du visage et des yeux et montrer les bibliothèques utilisées dans le programme ainsi que les matériels utilisés et enfin, nous présentons l'organigramme que nous avons suivi.

III.2. Le Système proposé :

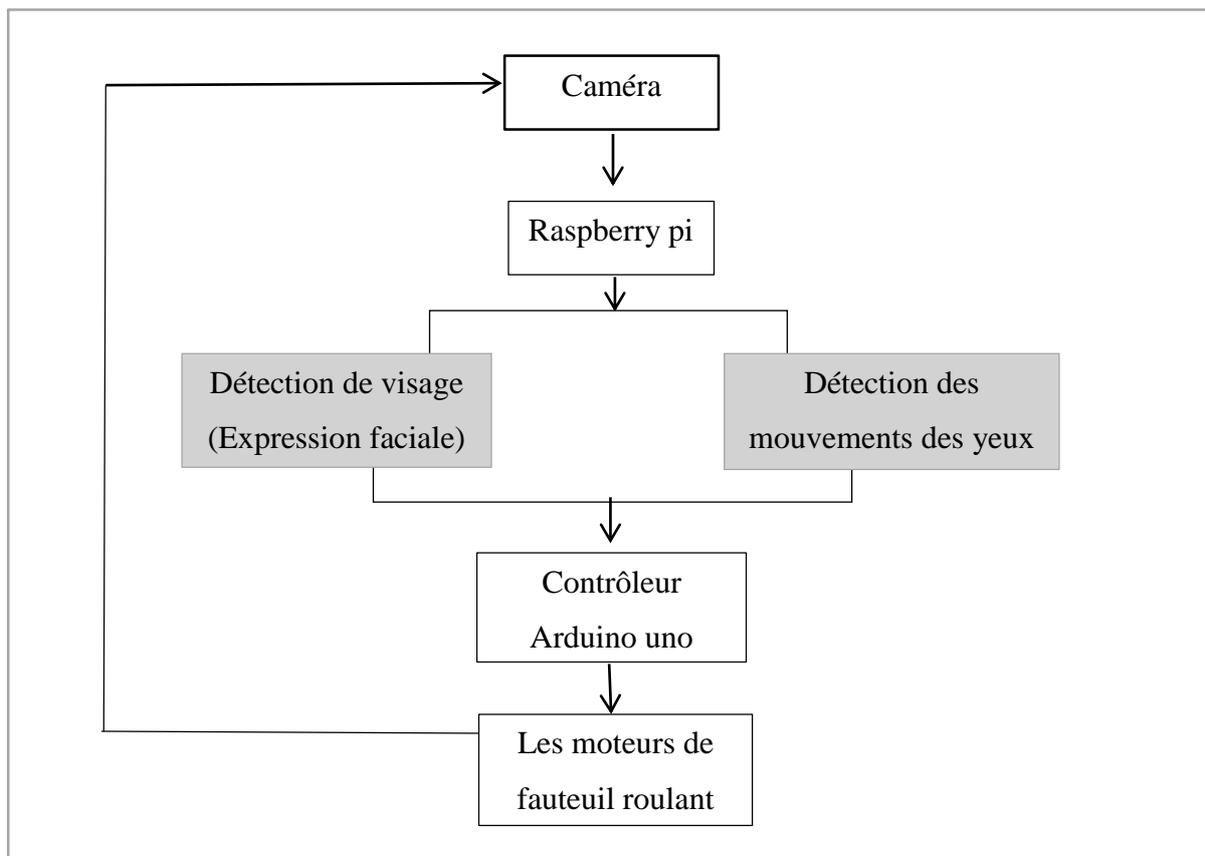


Figure III-1 : Structure du système Hardware et software

III.2.1. Description du partie hardware :

Pour la partie Hardware d'un fauteuil roulant électrique équipé avec caméra nous besoin les matériels suivent

III.2.1.1. Matériels utilisés :

Raspberry pi 4 : Le Raspberry Pi est ce qu'on appelle un micro-ordinateur, c'est-à-dire un ordinateur avec de plus faibles performances mais aussi de très petite taille.

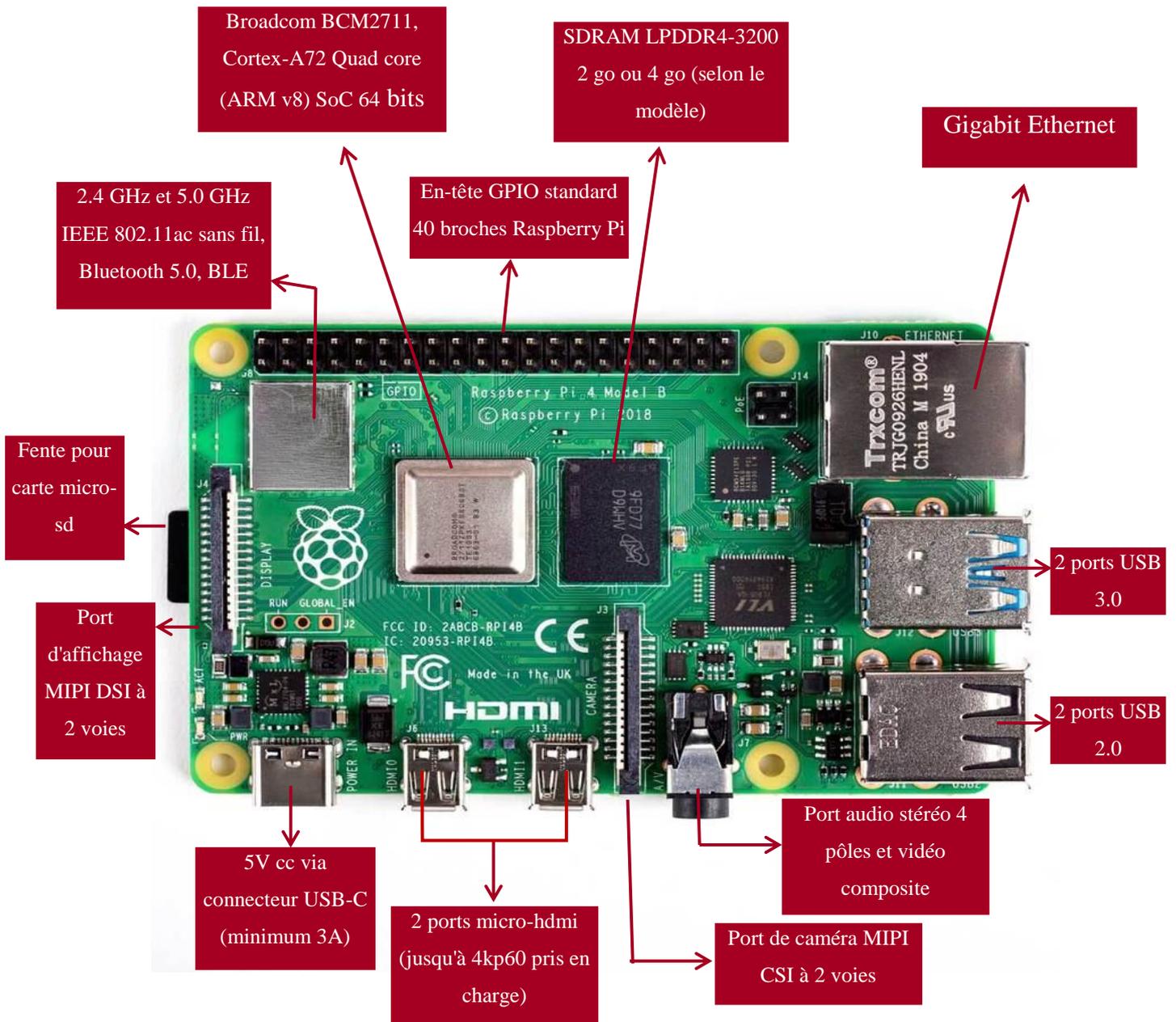


Figure III-2 : Raspberry pi 4 (modèle B)

Arduino uno :

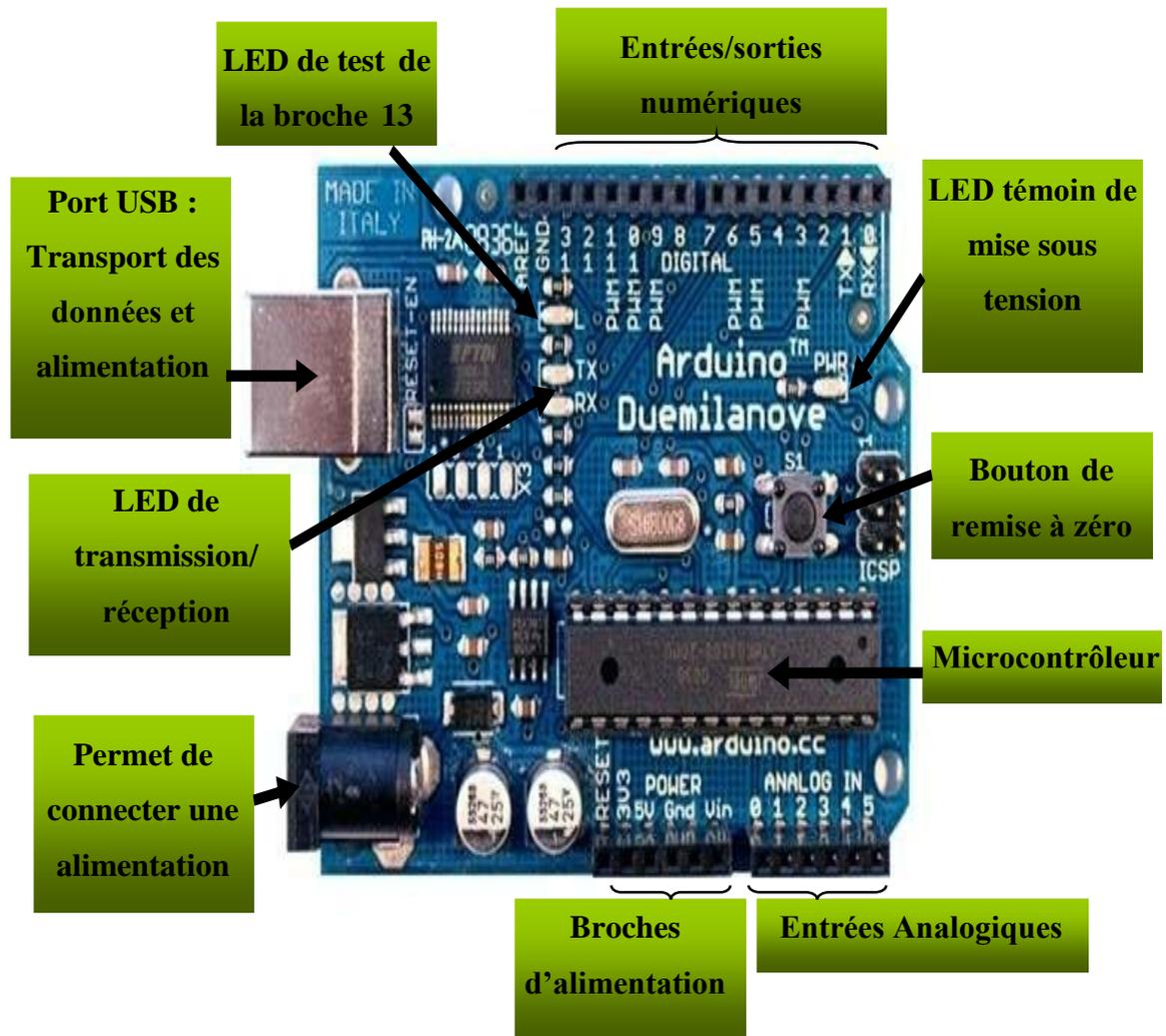
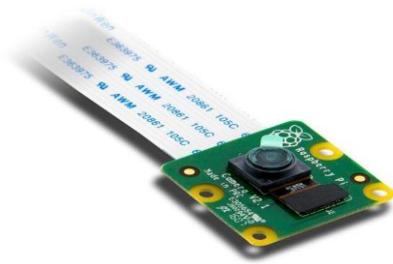


Figure III-3 : Arduino Uno

La caméra Pi et la webcam :

La carte caméra Pi n'utilise pas de port USB et elle est directement interfacée avec le Pi. Elle offre donc de meilleures performances qu'une webcam en termes de fréquence d'images et de résolution. Nous pouvons utiliser directement le module pi camera en Python pour travailler sur les images et les vidéos. Mais la caméra Pi ne peut être utilisée avec aucun autre ordinateur [67].

Une webcam utilise un port USB pour l'interface, et de ce fait, elle peut être utilisée avec n'importe quel ordinateur. Mais, par rapport à la caméra Pi, ses performances sont inférieures en termes de fréquence d'images et de résolution [67].



A



B

Figure III-4 : (A) caméra pi, (B) webcam

III.2.2. Description du partie Software :

III.2.2.1. Système d'exploitation de Raspberry Pi :

Il existe de nombreux systèmes d'exploitation dédiés au Raspberry Pi [68]. Dans ce projet nous utiliserons Raspbian comme système d'exploitation pour Raspberry Pi.

Il s'agit d'un système d'exploitation GNU/Linux, spécialement conçu et optimisé pour les Raspberry Pi. Sa version actuelle est basée sur la dernière version de la distribution Debian, baptisée Stretch (version 9.X).

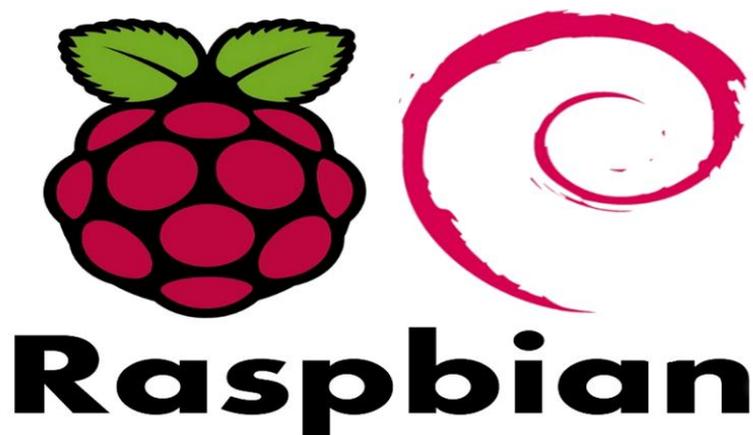


Figure III-5 : Raspbian Logo

Téléchargement du système Raspbian et déplacer sur la carte SD :

Tout d'abord, on va donc commencer par télécharger son système d'exploitation Raspbian Stretch sur le PC. Pour cela, nous aurons besoin d'une Carte SD avec la capacité qu'il faut (16 Go minimum recommandé). Si on ne dispose pas d'ordinateur

Chapitre III : Résultats de l'étude et Système de Commande

sous Linux, il est bien évidemment possible de réaliser la préparation de la Carte SD à partir d'un ordinateur Windows. Pour cela, on aura besoin d'installer tout d'abord un logiciel spécifique, dé

dié à cette opération : le logiciel libre Win32DiskImager. C'est un programme qui nous laisse déplacer l'image de système d'exploitation (forme ISO) dans la carte SD puis nous mettons la carte SD dans Raspberry Pi [68] [69].



Figure III-6 : Kit Raspberry Pi 4

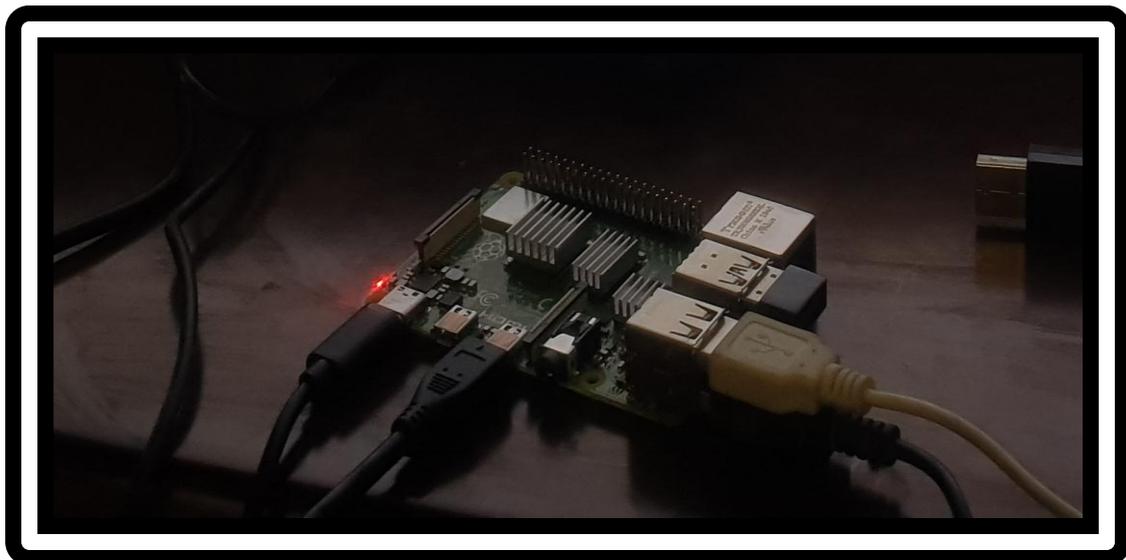


Figure III-7 : Allumez le Raspberry Pi pour la première fois

Chapitre III : Résultats de l'étude et Système de Commande

- Brancher le câble HDMI sur la prise correspondante du Raspberry Pi, derrière votre écran pour établir la connexion.
- Brancher le clavier et la souris sur les ports USB du Raspberry Pi.
- Brancher le cordon d'alimentation du Raspberry dans sa prise dédiée puis le boîtier transformateur sur le secteur, et voir que Raspberry Pi commence à installer le système automatiquement.
- Le Raspberry et l'écran s'allument, au bout de quelques secondes vous êtes « sur le bureau » de votre tout nouvel ordinateur.



Figure III-8 : Exécution du système pour la première fois (d'écran HD)

III.2.2.2. L'organigramme de système :

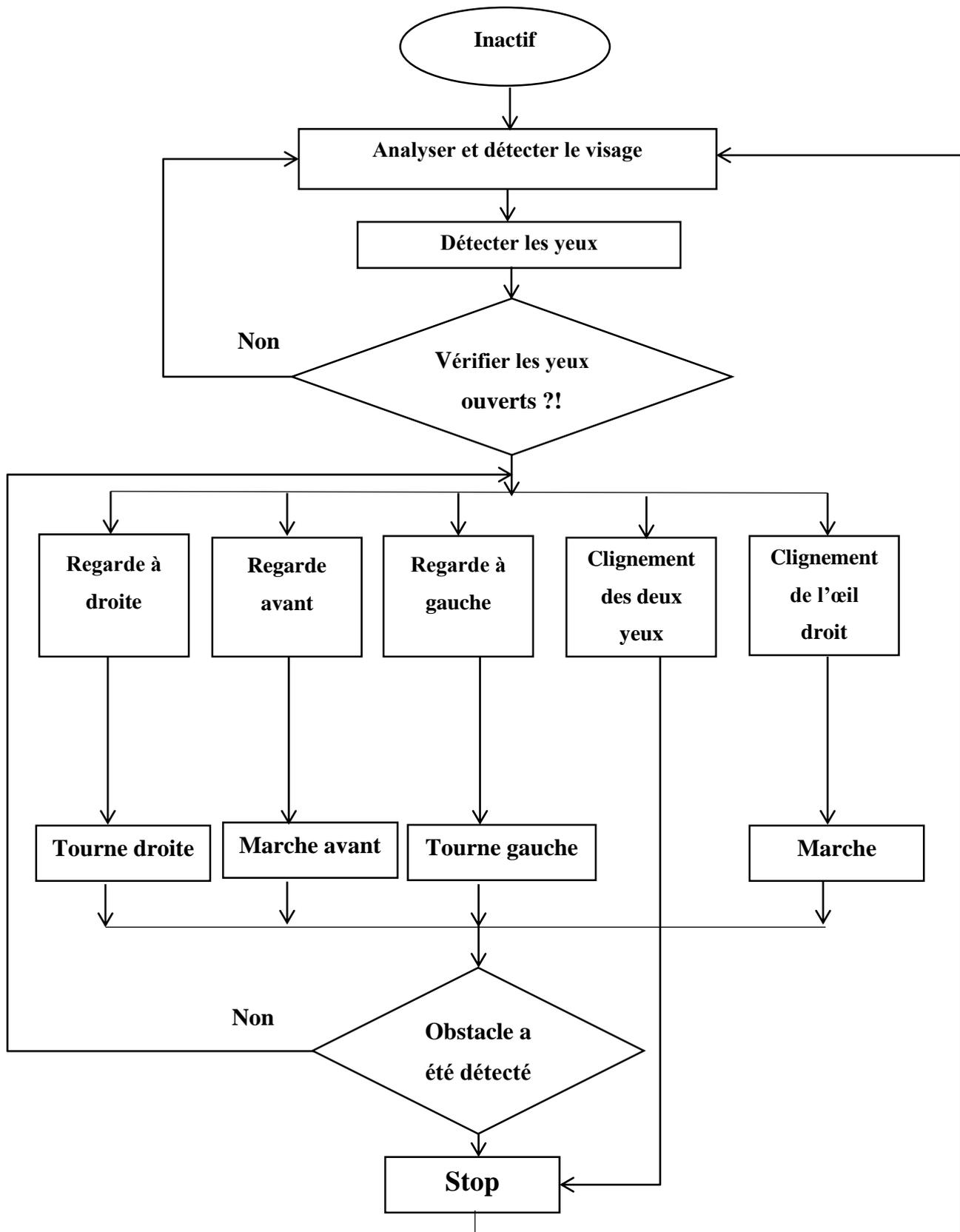


Figure III-9 : Organigramme du fonctionnement du système (fauteuil roulant)

III.3. La méthode de détection de visage :

Pour commander ce fauteuil roulant, le système doit extraire les cadres des images une fois les vidéos capturées par la caméra, détecter l'emplacement du visage et des yeux, et marquer la région du visage et des yeux en utilisant les bibliothèques dlib [59] [60] (pour face detection and facial landmark detection) et OpenCV.

Premièrement, détecter la région du visage du conducteur du fauteuil ; s'il y a plusieurs visages ou aucun cadre dans la fenêtre, il détecte faussement le visage et envoie une commande « visage non détecté » au processeur. Sinon il envoie une commande « visage détecté » au processeur et nous utiliserons la fonction Dlib [59] appelée (get_frontal_face_detector) assez intuitive. Il y a une mise en garde cependant, cette fonction ne marche qu'avec des images en niveaux de gris, nous devons donc le faire d'abord avec OpenCV.

FACIAL LANDMARKS (Détection de visage) :

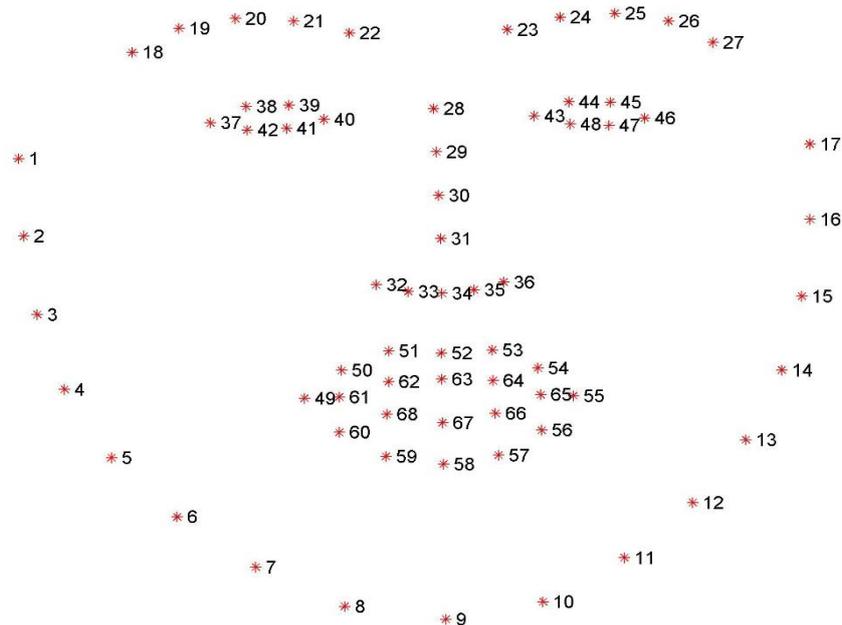


Figure III-10 : 68 FACIAL LANDMARKS de visage

Dans la boîte à outils Dlib, la pose du visage est représentée par 68 points de repère, comme le montre la figure 4. La boîte à outils génère un régresseur pour obtenir les repères faciaux du conducteur avec l'entrée de 68 repères étiquetés.

Dlib [61] est une boîte à outils open source qui inclut la machine algorithmes d'apprentissage et les outils pour créer des algorithmes complexes pour résoudre des problèmes du monde réel. Il est largement utilisé dans l'industrie et le monde universitaire, y compris la robotique, les appareils embarqués, les téléphones mobiles et grands environnements informatiques hautes performances.

Le détecteur de visage frontal (frontal face detector) de dlib fonctionne très bien. Et c'est simple pour le travail.

```
detector = dlib.get_frontal_face_detector ()
```

Rendre un rectangle sur le visage, comme suit :

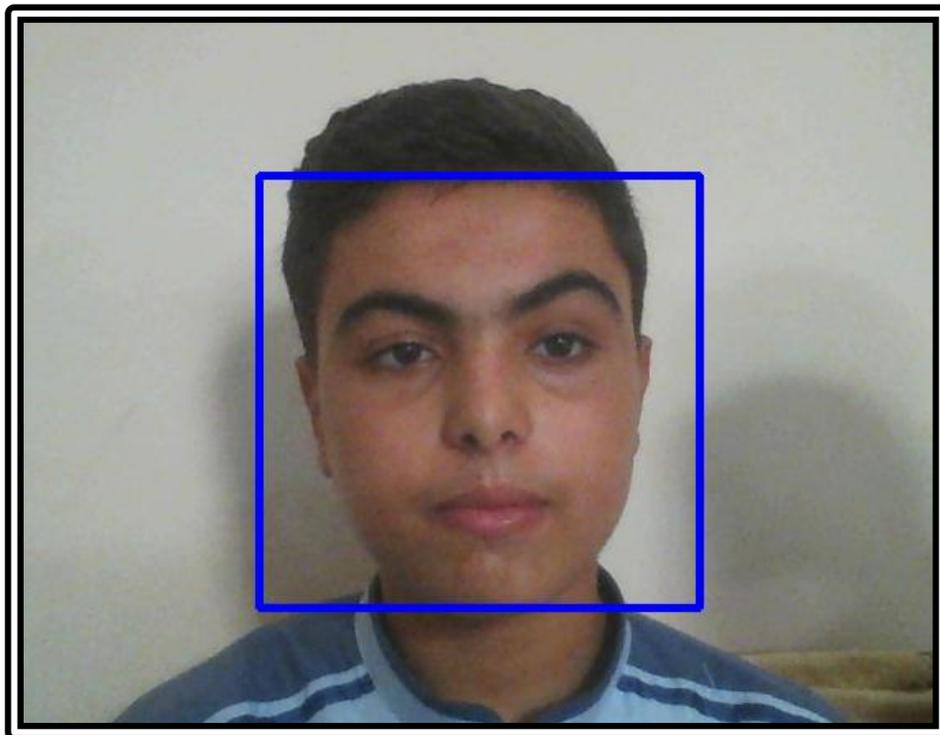


Figure III-11 : détecte de visage

III.4. La méthode de détection des mouvements des yeux :

Pour contrôler les mouvements des yeux on détecte la région des yeux ; nous utiliserons la fonction Dlib appelée `dlib.shape_predictor()`, utilisant les point 68 facial landmarks pour marquer la région des yeux (`eyes_region`) qui est déterminée par 6 points pour chaque œil.

Dans notre system nous avons besoin de la position de la pupille dans les yeux pour trouver la direction exacte de l'œil afin de guider les moteurs du fauteuil.

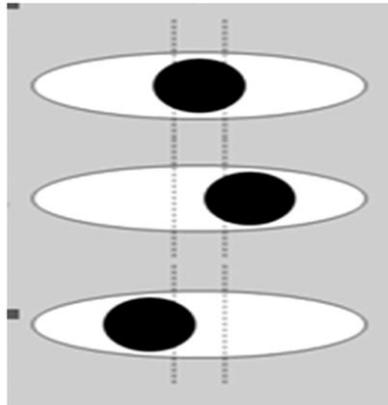


Figure III-12 : Les deux lignes de scan pour détecter le mouvement de l'œil

Comment fonctionner ?

- Si la pupille est au centre (côté gauche blanc == côté droit blanc) le fauteuil marche avant.
- Si la pupille est au côté gauche de l'œil (côté gauche blanc < côté droit blanc) le fauteuil marche gauche.
- Si la pupille est au côté droit de l'œil (côté gauche blanc > côté droit blanc) le fauteuil marche droit.
- Quand les yeux sont fermés le fauteuil s'arrête.
- Si l'œil droit est fermé et l'œil gauche est ouvert le fauteuil fait marche arrière.

Détecter le regard :

Nous allons détecter le regard, et plus précisément à détecter si les yeux regardent vers la gauche, la droite ou le centre. Pour détecter le regard, nous devons d'abord comprendre comment l'œil apparaît lorsqu'il regarde dans différentes directions.

Avant d'entrer dans les détails du traitement d'image, examinons un peu l'œil et réfléchissons aux solutions possibles pour y parvenir.

Dans l'image ci-dessous, nous voyons un œil. L'œil est composé de trois parties principales [62] :

- Pupille - le cercle noir au milieu.

Chapitre III : Résultats de l'étude et Système de Commande

- Iris - le plus grand cercle qui peut avoir une couleur différente pour différentes personnes.
- Sclera (SCLÉROTIQUE) - c'est toujours blanc.

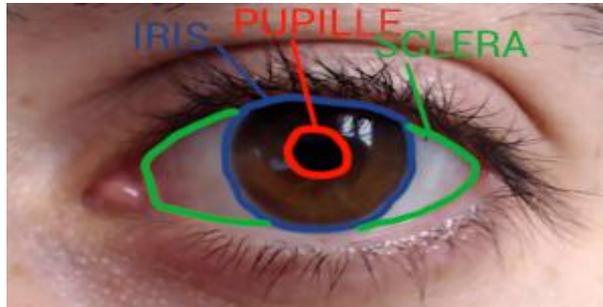


Figure III-13 : L'œil

Normalement, les deux yeux regardent dans la même direction, donc si nous détectons correctement le regard d'un seul œil, nous détectons le regard des deux yeux.

Du seuil, nous trouvons les contours. Et nous supprimons simplement tout le bruit en sélectionnant l'élément avec la plus grande surface (qui est censée être la pupille) et sautons tout le reste.

Voyons toutes les directions possibles (dans l'image ci-dessous) que l'œil peut avoir et trouvons les éléments communs et inhabituels entre elles.



Figure III-14 : Les directions de la pupille

Que pouvons-nous comprendre de cette image ?

En commençant par la gauche, nous voyons que la sclérotique recouvre le côté opposé de l'endroit où la pupille et l'iris pointent. Lorsque l'œil tout droit, la sclérotique est bien équilibrée du côté gauche et du côté droit.

Détecter le mouvement :

Pour la détection, nous pourrions utiliser différentes approches, en se concentrant sur la sclérotique, l'iris ou la pupille.

Nous optons pour l'approche la plus simple possible, et probablement la meilleure solution de toute façon.

Nous nous concentrerons simplement sur la pupille [63]. En convertissant l'image en format de niveau de gris, nous verrons que la pupille est toujours plus sombre que le reste de l'œil. Peu importe où l'œil regarde et quelle que soit la couleur de la sclérotique de la personne.

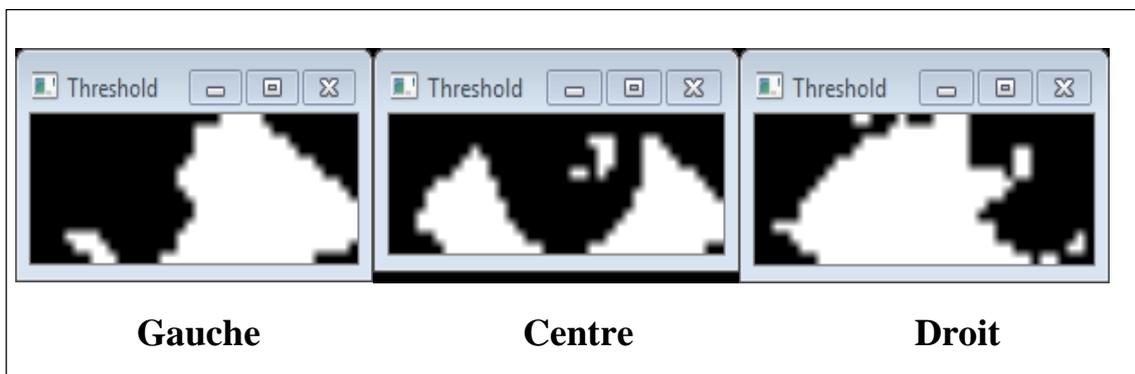


Figure III-15 : Les mouvements de la pupille au niveaux de gris

Qui, à première vue est vraiment clair, en regardant l'image ci-dessus, c'est que la sclérotique (partie blanche de l'œil) remplit la partie droite de l'œil lorsque l'œil regarde vers la gauche, l'inverse se produit quand il regarde la droite et quand on regarde au centre, le blanc est bien équilibré entre la gauche et la droite.

Détecter le clignement (blinking) :

Le clignement des yeux peut être détecté en regardant sur les importants traits faciaux dans le visage. Pour les clignements oculaires, nous devons tenir compte des points 37-46, qui marquent les yeux ; si les yeux clignent en temps réel en utilisant les repères faciaux (facial_landmarks) [64].

La détection du clignement des yeux c'est pour le contrôle du fauteuil roulant. Quand les yeux sont fermés pendant 0.3 à 0.4 seconde le fauteuil s'arrête. Si un œil est ouvert et l'autre cligne, les moteurs font marche arrière (backward).

Un œil clignant lorsque :

- La paupière est fermée ;
- On ne voit plus le globe oculaire ;
- Les cils inférieurs et supérieurs se connectent ensemble.

Et nous devons également prendre en compte le fait que toutes ces actions doivent se produire pendant un temps (environ un clin d'œil prend 0,3 à 0,4 seconde) sinon cela signifie que l'œil est juste fermé.

Maintenant que nous avons trouvé une réponse possible pour détecter le clignement de l'œil, nous devons nous concentrer sur ce qu'il est possible de détecter à l'aide d'Opencv et dlib, en choisissant éventuellement la solution la plus simple et la plus fiable avec ce que nous avons déjà.

Nous avons détecté l'œil par les point de facial landmarks, nous avons aussi détecté deux lignes : une ligne horizontale et une ligne verticale traversant l'œil. Nous allons les examiner de près car ils vont être le noyau de l'approche de la détection du clignement.

Ceci quand l'œil est ouvert :

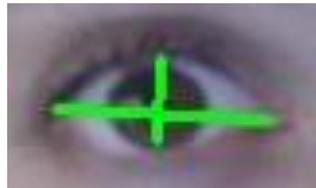


Figure III -16 : l'œil ouvert avec les lignes horizontale et verticale

Ceci quand l'œil est fermé :



Figure III -17 : l'œil fermé avec les lignes horizontale et verticale

Rapport de regard (Eye gaze ratio) :

Le rapport du regard nous indique où un œil spécifique regarde. Il faut maintenant se concentrer sur la détection de la partie blanche.

L'idée est de diviser l'œil en deux parties et de savoir dans laquelle des deux parties il y a le plus de sclérotique visible.



Figure III-18 : les deux parties de l'œil

Si la sclera est plus visible sur la partie droite, donc l'œil regarde vers la gauche (notre gauche) comme dans ce cas.

Techniquement pour détecter la sclérotique on convertit l'œil en niveau de gris, on trouve un seuil et on compte les pixels blancs [65].

Le rapport de regard : pour obtenir le rapport du regard nous divisons les pixels blancs de la partie gauche et ceux de la partie droite.

$$\text{gaze_ratio} = \text{left_side_white} / \text{right_side_white}$$

Rapport de Blinking (Blinking ratio):

Soukupová et Čech adoptent une équation qui représente le rapport hauteur / largeur des yeux. Le rapport de clignement des yeux est une évaluation de l'état d'ouverture des yeux. Comme représenté sur la figure (6), le rapport hauteur / largeur des yeux peut être déterminé à partir de l'équation suivante [66].

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \quad (\text{III.4.1})$$

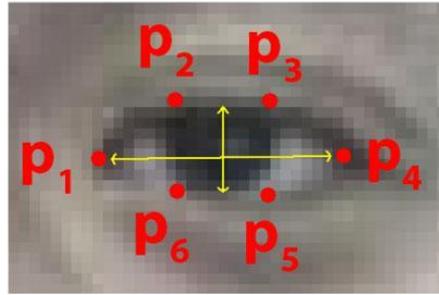


Figure III-19 : Landmarks des yeux

A partir de la détection de la ligne horizontale et la ligne verticale dans l'œil (figure III-16 et figure III-17). Nous pouvons clairement voir que la taille de la ligne horizontale est presque identique à l'œil fermé et à l'œil ouvert alors que la ligne verticale est beaucoup plus longue à l'œil ouvert par rapport à l'œil fermé. Dans l'œil fermé, la ligne verticale disparaît presque.

Sur le programme python, nous créons ensuite une fonction pour détecter le ratio (rapport) de clignement (Blinking ratio) et le rapport de regard où nous insérons les points des yeux et les coordonnées des points de repère du visage (facial landmarks) et nous obtiendrons le rapport entre ces deux lignes (horizontale et verticale) et aussi trouvons le ratio de partie blanc (sclera) de l'œil [64] [65].

La prochaine étape, consiste à brancher notre webcam (caméra) et à effectuer une reconnaissance du visage et des yeux en temps réel à partir de flux vidéo pour commander notre fauteuil roulant.

III.5. Description du programme :

Code python :

Les codes de base qui seront utilisés dans le programme de détection de mouvement et de clignement (blinking) des yeux sont :

- Chargement du module cv2 et les bibliothèques dlib, numpy, math dans le script Python, comme suit :

```
Import cv2
Import dlib
Import numpy as np
From math import hypot
```

- Après avoir importé les bibliothèques cv2 et numpy, nous devons définir l'objet cap. Pour charger les frames depuis la webcam :

```
Cap = cv2.VideoCapture(0)
```

Le chiffre 0 signifie la première webcam. Si vous avez installé plus de webcam, vous pouvez changer le nombre à 1 pour la seconde ou à 2 pour la troisième webcam et ainsi de suite.

- À ce point, nous détectons le visage et les points de repère du visage, de sorte qu'à partir d'eux, nous pouvons trouver les limites externes du visage par dlib.
- La bibliothèque dlib utilise un détecteur de visage pré-entraîné basé sur une modification de la méthode Histogramme des dégradés orientés + SVM linéaire pour la détection d'objets.

```
detector = dlib.get_frontal_face_detector()
predictor =
dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

Avec Shape_Predictor nous obtenons le téléchargement de github [73].

- Nous créons une fonction dont nous aurons besoin plus tard pour détecter le point médian. Sur cette fonction, nous mettons simplement les coordonnées de deux points et renverrons le point moyen (le point au milieu entre les deux points).

```
def midpoint(p1 ,p2):
    return int((p1.x + p2.x)/2), int((p1.y + p2.y)/2)
```

- Nous créons une fonction pour deux lignes, l'une traversant l'œil horizontalement et l'autre traversant l'œil verticalement. Nous en aurons besoin pour la prochaine étape lorsque nous détecteront le clignement des yeux.

```
def get_blinking_ratio(eye_points, facial_landmarks):
    left_point = (facial_landmarks.part(eye_points[0]).x,
                  facial_landmarks.part(eye_points[0]).y)
    right_point = (facial_landmarks.part(eye_points[3]).x,
                   facial_landmarks.part(eye_points[3]).y)
    center_top = midpoint(facial_landmarks.part(eye_points[1]),
                          facial_landmarks.part(eye_points[2]))
    center_bottom = midpoint(facial_landmarks.part(eye_points[5]),
                             facial_landmarks.part(eye_points[4]))

    hor_line_lenght = hypot((left_point[0] - right_point[0]),
                             (left_point[1] - right_point[1]))
    ver_line_lenght = hypot((center_top[0] - center_bottom[0]),
                             (center_top[1] - center_bottom[1]))

    ratio = hor_line_lenght / ver_line_lenght
    return ratio
```

- Nous devons créer une fonction pour détecter le regard des deux yeux, mais nous concentrerons uniquement sur un œil (left eye) et nous appliquerons la même méthode pour le deuxième œil.
- Nous pouvons sélectionner le deuxième œil en prenant simplement les coordonnées du point de repère.

```
def get_gaze_ratio(eye_points, facial_landmarks):
    left_eye_region = np.array([(facial_landmarks.part(eye_points[0]).x,
                                facial_landmarks.part(eye_points[0]).y),

                                (facial_landmarks.part(eye_points[1]).x,
                                facial_landmarks.part(eye_points[1]).y),

                                (facial_landmarks.part(eye_points[2]).x,
                                facial_landmarks.part(eye_points[2]).y),

                                (facial_landmarks.part(eye_points[3]).x,
                                facial_landmarks.part(eye_points[3]).y),

                                (facial_landmarks.part(eye_points[4]).x,
                                facial_landmarks.part(eye_points[4]).y),

                                (facial_landmarks.part(eye_points[5]).x,
                                facial_landmarks.part(eye_points[5]).y),
                                ], np.int32)
```

- Une fois que nous avons les coordonnées d'un œil (left eye), nous pouvons créer le masque pour extraire exactement l'intérieur de l'œil gauche (left eye) et exclure tout l'environnement.

```
height, width, _ = frame.shape
mask = np.zeros((height, width), np.uint8)
cv2.polylines(mask, [left_eye_region], True, 255, 2)
cv2.fillPoly(mask, [left_eye_region], 255)
eye = cv2.bitwise_and(gray, gray, mask=mask)
```

- Techniquement, pour détecter la sclérotique nous convertissons l'œil en niveaux de gris, nous trouvons un seuil et nous comptons les pixels blancs.
- On divise les pixels blancs de la partie gauche et ceux de la partie droite et on obtient le rapport de regard.

```
left_side_threshold = threshold_eye[0: height, 0: int(width / 2)]
left_side_white = cv2.countNonZero(left_side_threshold)

right_side_threshold = threshold_eye[0: height, int(width / 2):
width]
right_side_white = cv2.countNonZero(right_side_threshold)

if left_side_white == 0:
    gaze_ratio = 1
elif right_side_white == 0:
    gaze_ratio = 5
else:
    gaze_ratio = left_side_white / right_side_white
return gaze_ratio
```

- Initialisation des compteurs, nous utilisons des compteurs pour faire fonctionner les moteurs lorsque le conducteur du fauteuil regarde dans une certaine direction dans laquelle le fauteuil tourne

```
blinking_frames = 0
Right_frames = 0
frames_to_direction = 25
frames_to_blink = 10
```

- Après cela, nous pouvons exécuter la boucle While, prendre les images de la webcam et détecter le visage.

```
while True:
    _, frame = cap.read()
    new_frame = np.zeros((500, 500, 3), np.uint8)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)
    for face in faces:
        x, y = face.left(), face.top()
        x1, y1 = face.right(), face.bottom()
        cv2.rectangle(frame, (x, y), (x1, y1), (0, 255, 0), 2)

        landmarks = predictor(gray, face)
```

- Nous avons besoin de détecter séparément les deux yeux : Points de l'œil gauche : (36, 37, 38, 39, 40, 41), points de l'œil droit : (42, 43, 44, 45, 46, 47) ; et nous avons trouvé le rapport de clignement (blinking_ratio)

```
# Detect Blinking
left_eye_ratio = get_blinking_ratio([36, 37, 38, 39, 40, 41],
                                     landmarks)
right_eye_ratio = get_blinking_ratio([42, 43, 44, 45, 46, 47],
                                     landmarks)
blinking_ratio = (left_eye_ratio + right_eye_ratio) / 2
```

- Nous allons ensuite utiliser le nombre de rapport pour détecter l'arrêt (stop) et l'arrière (backward) du fauteuil en définissant quand l'œil cligne (blink) ou non.

```
# Gaze stop
if blinking_ratio > 5:
    blinking_frames += 1
    Right_frames = 0
    new_frame[:] = (0, 255, 0)
    cv2.putText(frame, "Stop", (50, 150), font, 7, (255, 0, 0))
    if blinking_frames == frames_to_blink:
        print("stop")
        blinking_frames = 0
```

Chapitre III : Résultats de l'étude et Système de Commande

- Nous avons trouvé le rapport de regard (gaze_ratio) de deux yeux :

```
# Gaze Detection
gaze_ratio_left_eye = get_gaze_ratio([36, 37, 38, 39, 40, 41],
landmarks)
gaze_ratio_right_eye = get_gaze_ratio([42, 43, 44, 45, 46, 47],
landmarks)
```

- Nous allons en suite utiliser le nombre de rapport du regard. Nous avons découvert que lorsque le rapport de regard est inférieur à 0.9 vous regardez du côté droit et si supérieur à 1,7 les yeux sont tournés vers le côté gauche et entre 0.9 et 1.7 vous regardez au centre.

```
if gaze_ratio <= 0.9:
    Right_frames += 1
    blinking_frames = 0
    new_frame[:] = (0, 0, 255)
    cv2.putText(frame, "RIGHT", (50, 100), font, 2, (0, 0,
255), 3)
    if Right_frames == frames_to_forward:
        print("RIGHT")
        Right_frames = 0
```

- Enfin on l'affiche les deux frames à l'écran :
- Le premier frame (**Frame**) affiche le visage capturé par la webcam.
- Le deuxième frame (**New frame**) affiche des couleurs.

```
cv2.imshow("Frame", frame)
cv2.imshow("New frame", new_frame)
```

Configuration Arduino Uno sur Raspberry Pi :

La première étape de programmation d'une carte Arduino avec un Raspberry Pi consiste à installer l'IDE Arduino (environnement de développement intégré) sur votre

Raspberry Pi. Ce programme vérifie le code et le charge sur l'Arduino. Installer la dernière version d'Arduino IDE en utilisant apt [71] :

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install arduino
```

III.6. Résultat de Partie de commande :

Nous avons choisi de tester notre système sur différentes plateformes et il y'a deux choix pour tester le langage python :

Ordinateur : Micro-ordinateur portable (PC), pour installation des bibliothèques OpenCV et Dlib on a besoin d'Anaconda (une distribution python) qui permet une excitation rapide et plus fiable de notre programme sur le PC.

Raspberry Pi : nous avons lancé le programme sur un processeur. Une architecture du système a également été conçue et "mappée" sur le Raspberry Pi, l'excitation va prendre du temps et le traitement de vidéo en temps réel fais du retard. L'installation des bibliothèques OpenCV et Dlib est plus longue ; ça peut même prendre plus de 3 heures pour les installer tous ensemble.

Résultat sur le micro portable PC :

La direction du regard est interprétée à partir des positions des deux yeux de l'utilisateur de fauteuil roulant. Comme le montre dans la Figure III-20 et Figure III-21 :



Figure III-20 : Résultats de la commande du fauteuil roulant par la direction des yeux

Chapitre III : Résultats de l'étude et Système de Commande

Une interprétation du clignotement des yeux :

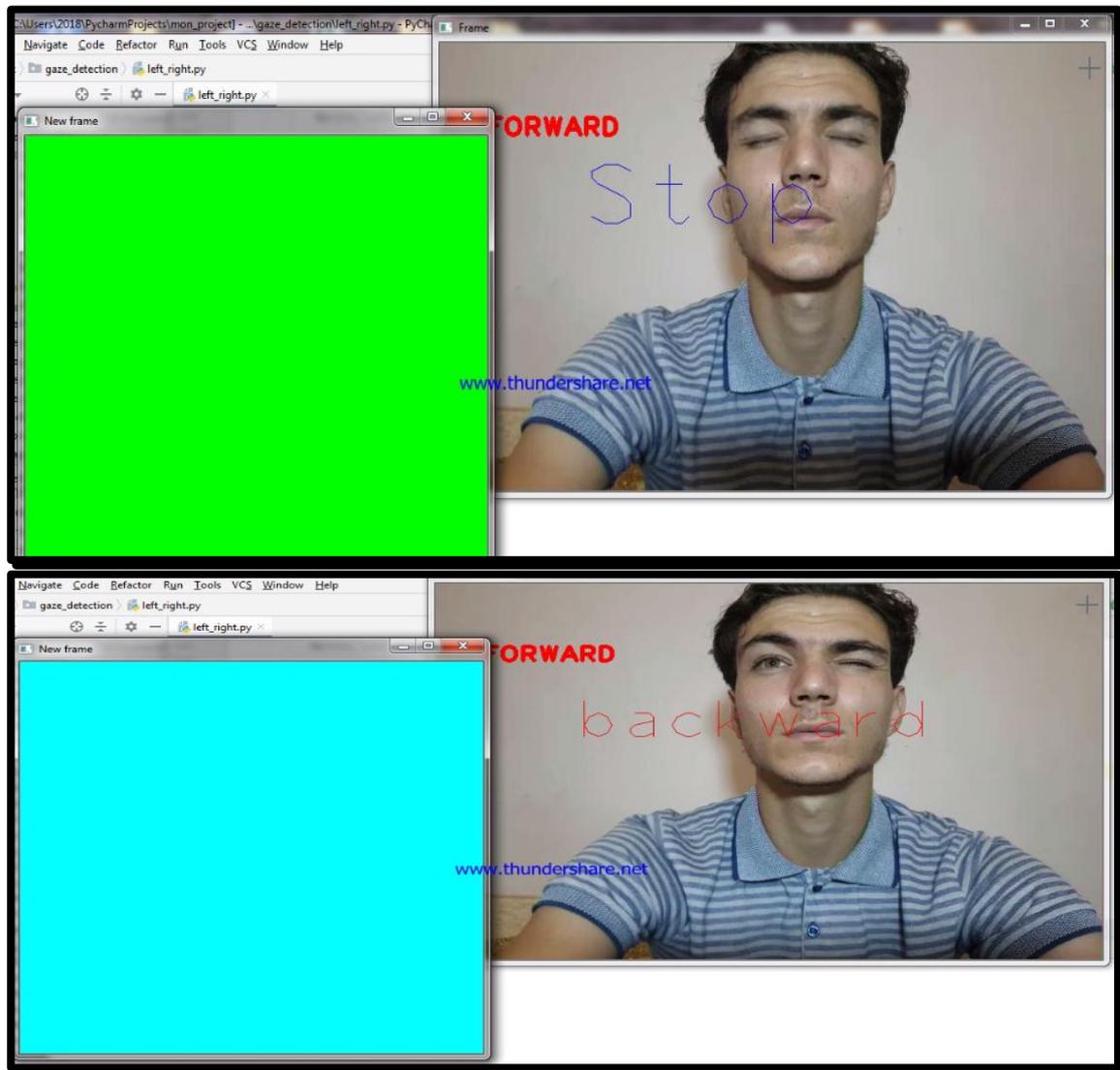


Figure III-21 : Résultats de la commande du fauteuil roulant par les clignotements des yeux

Résultat sur le Raspberry pi :

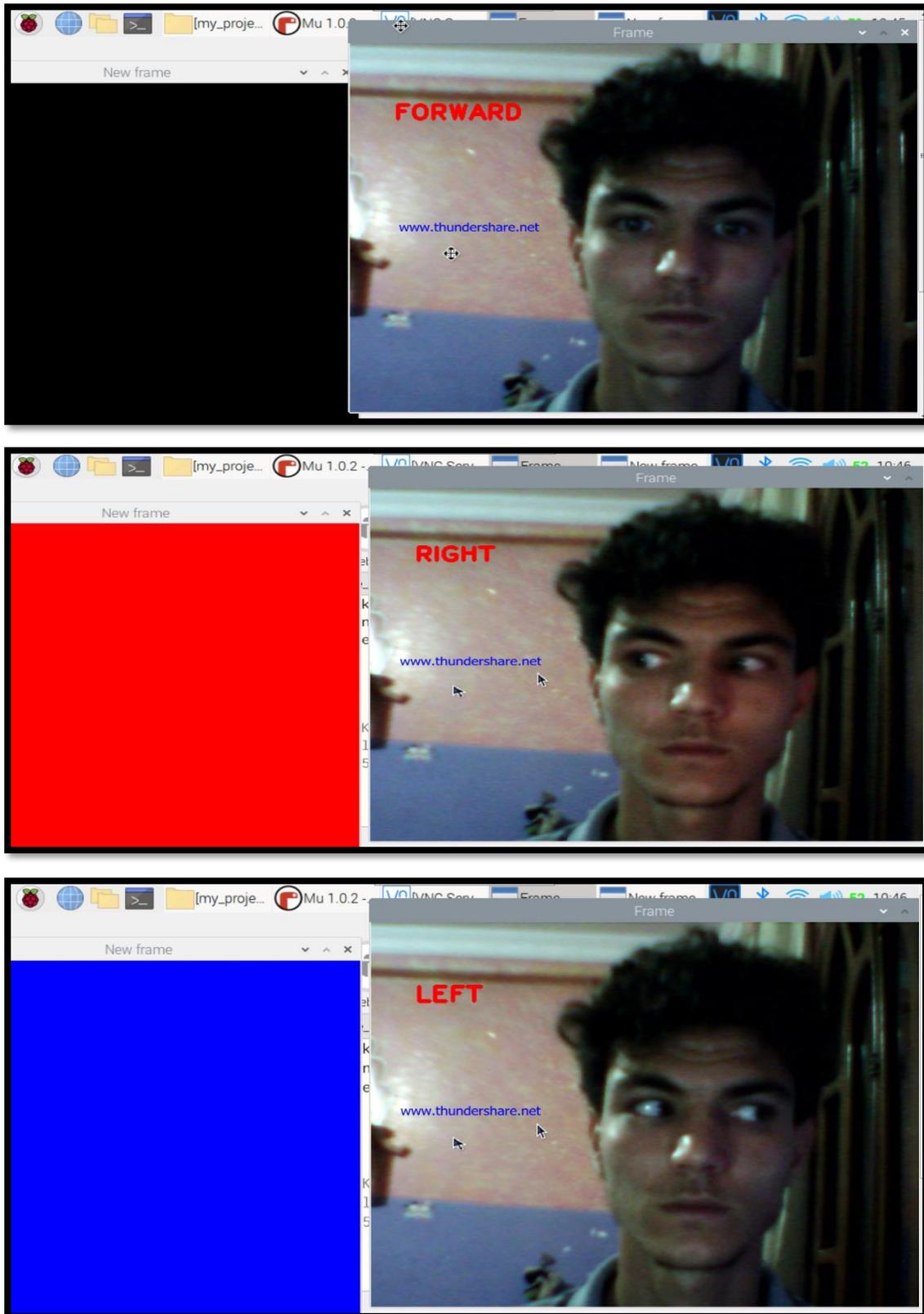


Figure III-22 : Résultats de programme de commande de fauteuil roulant sur le Raspberry pi

III.7. Conclusion :

Les résultats obtenus dans la partie de traitement des yeux sont satisfaisants.

Parfois, le capteur caméra ne peuvent pas détecter le visage et les yeux s'il y'a plus qu'une personne devant la webcam ou la position de la tête de conducteur n'est pas au centre de la webcam.

Nous avons constaté aussi que Le programme de système prend un certain temps pour exécuter le système de traitement de la vidéo en environnement temps réel.

Il est très difficile de suivre la pupille dans les endroits sombres, donc le système fonctionne parfaitement à la lumière ambiante et dans une salle éclairée par des lampes.

Nous avons pu conclure que les méthodes utilisées pour la détection de visage et des mouvements des yeux ont montrées des résultats favorables.

Conclusion générale

Dans ce projet, nous avons présenté une nouvelle implémentation d'un système autonome pour les personnes complètement handicapées, ce qui leur permet de commander le fauteuil roulant intelligent par les yeux. Cela consiste à réaliser un système de commande en utilisant la carte électronique Raspberry Pi ou en utilisant le pc comme unité de contrôle à l'aide d'une caméra.

L'être humain a une tendance naturelle à faire appel à ses yeux pour communiquer avec les autres. C'est pourquoi on choisit la technique de la commande oculaire qui constitue le meilleur mode d'accès à un dispositif de communication pour les personnes handicapées physique.

Le logiciel python avec la bibliothèque OpenCV permet de faire le traitement de l'image et de vidéo. et le programme utiliser basé sur la boîte à outils dlib, qui permet à détecter les mouvements des yeux pour commander FRE « intelligent ». Ce système au total permet de faciliter le déplacement à ce handicapé.

Nous avons obtenu des résultats significatifs qui montrent l'efficacité de la méthode proposée, les commandes effectuées dans ce projet montrent bien la faisabilité de concevoir un programme pour la commande oculaire du FRE.

À la fin de ce mémoire, nous proposons quelques perspectives pour développer ce travail et rendre le fauteuil roulant à commande oculaire disponible sur le marché.

Perspectives :

- Il faut travailler sur le mouvement du fauteuil roulant lorsque la personne regarde simplement dans différentes directions.
- Nous pouvons intégrer certaines séquences pour allumer et éteindre des appareils électriques ou des serrures de porte.
- Nous pouvons ajouter un GPS sur un écran pour illustrer le chemin vers la destination.
- Nous pouvons ajouter au précédent système de GPS qui a été proposé, et un mécanisme permettant de connecter le fauteuil avec l'internet pour déterminer sa localisation en cas d'un bug qui touche le GPS.

Références bibliographique

Chapitre I : Généralité sur les fauteuils roulants intelligents

- [1] Richard C. Simpson, « Smart wheelchairs: A literature review », *Journal of Rehabilitation Research & Development(JRRD)*, vol.42, no.4, July/August 2005, pp.423-436.
- [2] Vignier N, Ravaud J. F, « Les utilisateurs de fauteuil roulant en France : aspects socioépidémiologiques. In : Le Fauteuil roulant ». Actes des 21èmes entretiens de la Fondation Garches, 2008, pp.21-33.
- [3] M'hamed SAHNOUN, « Conception et Simulation d'une Commande à Retour D'Effort pour Fauteuil Roulant Electrique ». Thèse de doctorat, l'école doctorale IAEM Lorraine. Pour l'obtention du grade de Docteur de l'Université de Metz. (Octobre 2007).
- [4] Fehr. L, Langbein. W, Skaar. S, « Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey ». *Journal of Rehabilitation Research & Development*, vol.37, no.3, 2000, pp.353-360.
- [5] Cooper, R. A., Spaeth, D. M., Jones, D. K., Boninger, M. L., Fitzgerald, S. G., & Guo. S, «Comparison of virtual and real electric powered wheelchair driving using a position sensing joystick and an isometric joystick», *Medical Engineering & Physics*, vol.24, no.10, 2002, pp.703-708.
- [6] Frédéric Goncalves, « Conception d'un environnement virtuel avec adaptation de l'immersion pour la simulation de conduite en fauteuil roulant », Thèse de doctorat, Laboratoire d'Ingénierie Des Systèmes de Versailles (LISV). Pour l'obtention du grade de Docteur en Robotique et Réalité Virtuelle, (2014).
- [7] Cagigas D, Abascal J. « Hierarchical path search with partial materialization of costs for a smart wheelchair ». *J Intell Robotic Sys*, vol.39, 2004, pp.409–31.
- [8] Robert Tang, « A Semi-autonomous Wheelchair Navigation System » Une thèse soumise en partial fulfilment des exigences du Master of Engineering, Département de génie mécanique Université de Canterbury, (Février 2012).

- [9] Mazo. M, Rodriguez. F.J, et al. « Wheelchair for physically disabled People with voice, ultrasonic and infrared sensor control ». Autonomous Robots 2, Septembre 1995, pp.203–224. <https://doi.org/10.1007/BF00710857>
- [10] Simpson. R, Levine. S, « Voice control of a powered wheelchair ». IEEE, Transactions on Neural Systems and Rehabilitation Engineering, vol.10, no.2, 2002, pp. 122-125.
- [11] Swartz. B. E, Goldensohn. E. S. « Timeline of the history of EEG and associated fields ». Electroencephalography and Clinical Neurophysiology, vol.106, no.2, février 1998, pp.173-176.
- [12] RIKEN. Real-time control of wheelchairs with brain waves. 2009, <https://www.todaysmedicaldevelopments.com/article/real-time-control-of-wheelchairs-with-brain-waves/>
- [13] Firoozabadi, S. M. P., Oskoei, M. A., & Hu, H. (2008). « A Human-Computer Interface based on Forehead Multi-Channel Bio-signals to control a virtual wheelchair ». Dans les actes de la 14e Conférence iranienne sur le génie biomédical (ICBME), pp.272–277.
- [14] Kuno Y, Shimada N, Shirai Y. « Look where you're going [robotic wheelchair] ». IEEE Robot Autom Mag, vol.10, no.1, 2003, pp.26–34.
- [15] Matsumoto Y, Ino T, Ogasawara T. « Development of intelligent wheelchair system with face- and gaze-based interface ». Actes du 10e atelier international de l'IEEE sur les robots et la communication humaine interactive (ROMAN), Bordeaux-Paris, France. Piscataway (NJ): IEEE; 2001, pp. 262–67.
- [16] Daizy C, Animesh K, Jignal P, Kavil Pu, Dimpal Kh, « Designing of Wheelchair to provide Ability to Move around for Disabled People using Electrooculography », International Journal of Engineering Trends and Technology(IJETT), vol.67, no.3, Mars 2019, pp.75-79.
- [17] Gautam G. « Eye movement based electronic wheel chair for physically challenged persons ». Int J Sci Technol Res, 2014, 206–212. [Google Scholar]
- [18] Arai K, Ronny M. « Eyes based electric wheel chair control system ». Int J Adv Comput Sci Appl, 2011; pp.98–105. [Google Scholar]

- [19] Dilok P, Sarawin K, Pongsakom W, Boonyanuch W, Yodchanan W (2014) « Automated navigation system for eye-based wheelchair controls ». Dans : Biomedical engineering international conference (BMEiCON).
- [20] Dilok P, Sarawin Kh, Pongsakorn We, Boonyanuch Wi, Yodchanan Wo, « Navigation-synchronized multimodal control wheelchair from brain to alternative assistive technologies for persons with severe disabilities », PMC, vol.11, no.2, 2017, pp.117-134.
- [21] Yanco, H. A. (1998). « Wheelchair: A robotic wheelchair system: Indoor navigation and user interface », Lecture Notes in Computer Science, vol.1458.
- [22] Odile H, Alain C, Michel K, « Le projet VAHM (Véhicule Autonome pour Handicapés Moteur) : la localisation », Traitement du Signal 2000, vol.17, no.3, pp.221-232
- [23] J.M. Pergandi, P.Mallet, D.Mestre, « Evaluation d'un aide à la navigation d'un fauteuil intelligent », 2006.
- [24] Gomi T, Griffith A, « Developing intelligent wheelchairs for the handicapped », Lecture Notes in Computer Science, vol.1458.Springer, Berlin, Heidelberg, 1998, pp.150-178.
- [25] Nikos KTENAS (2001), « Mobile robotics and Automation in Healthcare and Rehabilitation » (chap. 4), dans Mobile Robotics in Healthcare, p.81-85, IOS press[google livre].
- [26] Bryan. WC. Chung, « Pro Processing for Images and Computer Vision with OpenCv », New York, Springer Science, 2017, p.301.

Chapitre II : Détection et Reconnaissance du visage et des yeux

- [27] Jayneil. D, Sohil. P, Instant OpenCV Starter, Birmingham.UK, Packt Publishing Ltd, May 2013, p.56.
- [28] <http://sourceforge.net/projects/opencvlibrary/files/>(consulté le 20/07/2020 à 15h18)
- [29] <http://opencv.org/> (consulté le 20/07/2020 à 15h18)

- [30] Méthode de Viola et Jones
http://fr.wikipedia.org/wiki/Méthode_de_Viola_et_Jones
- [31] Gary. B, Adrian. K, « Learning OpenCV », the United States of America, O'Reilly Media, September 2008, 571p.
- [32] Ming-Hsuan. Y et al, « Detecting Faces in Images: A Survey », IEEE, vol. 24, no.1, JANUARY 2002, pp.34-58.
- [33] W. ZHAO et al, « Face Recognition: A Literature Survey », ACM Computing Surveys, vol. 35, no. 4, December 2003, pp. 399–458
- [34] Nicolas MORIZET, « Reconnaissance Biométrique par Fusion Multimodale du Visage et de l'Iris ». Thèse de doctorat, de l'Ecole Nationale Supérieure des Télécommunications. Pour l'obtention du grade de Docteur de l'école de Telecom Paris Tech. (18 Mars 2009).
- [35] KIRBY, M. ET SIROVICH, L. 1990. « Application of the Karhunen-Loeve procedure for the characterization of human faces ». IEEE Trans. Patt. Anal. Mars. Intell. 12.
- [36] SIROVICH, L. ET KIRBY, M. 1987. « Low-dimensional procedure for the characterization of human face ». J. Opt. Soc. Am. Vol.4, pp.519–524.
- [37] Andrea F. Abate ET al, « 2D and 3D face recognition: A survey», ScienceDirect, Département de mathématiques et d'informatique, Université de Salerne, Fisciano (SA), Italie, 26 janvier 2007, pp. 1885-1906
- [38] SOUHILA GUERFI ABABSA, « Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D ». Thèse de doctorat, Pour l'obtention du grade de Docteur de l'Université Evry Val d'Essonne. (3 October 2008).
- [39] V. Blanz, R. Sami, V. Thomas, « Face identification across different poses and illuminations with a 3d morphable model ». IEEE International Conference on Automatic Face and Gesture Recognition, 2002, pp. 202–207.
- [40] V. Blanz, T. Vetter, « Face recognition based on fitting a 3D morphable model », IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 2003, pp. 1063–1074.

- [41] Walid Hizem, « Capteur Intelligent pour la Reconnaissance de Visage ». Thèse de doctorat. L'Institut National des Télécommunications et l'Université Pierre et Marie Curie - Paris 6, Pour l'obtention du grade de Docteur de L'INSTITUT NATIONAL DES TELECOMMUNICATIONS. (2009).
- [42] Z. Zhu and Q. Ji. « Robust real-time eye detection and tracking under variable lighting conditions and various face orientations ». *Comput. Vis. Image Underst.*, vol. 98, no.1, pp.124-154, 2005.
- [43] G. C. Feng and P. C. Yuen. « Variance projection function and its application to eye detection for human face recognition ». *Pattern Recogn. Lett.*, vol.19, no.9, 1998.
- [44] Saad A. Sirohey and Azriel Rosenfeld. « Eye detection in a face image using linear and nonlinear filters ». *Pattern Recognition*, vol.34, 2001, pp.1367-1391.
- [45] J. Daugman. « How iris recognition works ». *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, 2004, pp. 21–30.
- [46] J. Daugman. « Probing the Uniqueness and Randomness of IrisCodes: Results from 200 Billion Iris Pair Comparisons ». Dans: *Proceedings of the IEEE*, November 2006, pp. 1927–1935.
- [47] J. Daugman. « New Methods in Iris Recognition ». Dans: *IEEE Transactions on Systems, Man, and Cybernetics*, October 2007, pp. 1167–1175.
- [48] W. Boles and B. Boashash. « A human identification technique using images of the iris and wavelet transform ». Dans: *IEEE Transactions on Signal Processing*, Avril 1998, pp. 1185–1188.
- [49] Y. Zhu, T. Tan, and Y. Wang. « Biometric personal identification based on iris patterns ». Dans: *Proceedings of the 15th International Conference on Pattern Recognition*, 2000, pp. 801–804.
- [50] C.Tisse L. Martin L. Torres and M. Robert, «Person Identification Technique Using Human Iris Recognition», *Proceedings of Vision Interface*, 2002, pp. 294-299.
- [51] E. Rydgren, T. Ea, F. Amiel, F. Rossant, ET A. Amara. «Iris features extraction using wavelet packets ». *International Conference on Image Processing (ICIP)*, Vol. 2, October 2004, pp. 861–864.

[52] Y. Chen, S. Dass, and A. Jain, «A Localized Iris Image Quality Measure Using 2D Wavelets», in Proc. of International Conf. on Biometrics, Hong Kong, China, Jan 6-8, 2006.

[53] M. Nabti & A. Bouridane, «An improved iris recognition system using feature extraction based on wavelet maxima moment invariants», Advances in Biometrics, Springer Berlin/Heidelberg, vol. 462, 2007, pp. 988–996.

[54] M. Nabti et A. Bouridane. «An effective and fast iris recognition system based on a combined multiscale feature extraction technique», Pattern Recognition, vol.41, 2008, pp. 868–879.

[55] A. Tsukada et al. (2011). « Illumination-Free Gaze Estimation Method for First Person Vision Wearable Device », In ICCV Workshop on Computer Vision in Vehicle Technology: From Earth to Mars, pp. 2084–2091.

[56] Tsukada, A. et Kanade, T. (2012). « Automatic acquisition of a 3d eye model for a wearable first-personvision device ». In ACM Symposium on Eye Tracking Research and Applications (ETRA), pp. 213–216.

[57] S. Viriri et J. Tapamo, « Iris pattern recognition based on cumulative sums and majority vote methods », In International Journal of Advanced Robotic Systems, 2017, pp.1-9.

[58] Arezou Banitalebi D, Syed A. R. Abu-Bakar, « A REVIEW OF IRIS RECOGNITION SYSTEM», Jurnal Teknologi (Sciences & Engineering) vol.77, 2015, pp. 275–282.

Chapitre III : Résultats de l'étude et Système de Commande

[59] D. E. King, « Dlib-ml: A machine learning toolkit » J. Mach. Learn. Res., vol. 10, Jan. 2009, pp. 1755–1758.

[60] Matt Bailey, ET. Al, « Development of Vision Based Navigation for a Robotic Wheelchair », in Proceedings of 2007 IEEE 10th International conference on rehabilitation robotics.

[61] FENG YOU, XIAOLONG LI, YUNBO GONG, « A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration », IEEE. Access., vol.7, 2019, pp.179396-179408.

- [62] Crihalmeanu, Simona & Ross, Arun, « Multispectral scleral patterns for ocular biometric recognition ». Pattern Recognition Letters. vol.33, 2012, pp.1860–1869.
- [63] Shyam Narayan Patel, « Autonomous Camera Based Eye Controlled Wheelchair System Using Raspberry-Pi », IEEE Sponsored 2nd International Conference, 2015.
- [64] Rohit Gupta¹, Rajesh kori¹, Swapnali Hambir¹, Ajit Upadhayay¹, Shridhar Sahu², « EYE CONTROLLED WHEELCHAIR USING RASPBERRY PI », SSRN.
- [65] eye-gaze-detection-2-gaze-controlled-keyboard-with-python-and-opencv
<https://pysource.com/2019/01/17/eye-gaze-detection-2-gaze-controlled-keyboard-with-python-and-opencv-p-4/>.
- [66] T. Soukupova and J. Cech. (2016, Feb. 3) Real-Time Eye Blink Detection using Facial Landmarks. Center for Machine Perception, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University in Prague. Prague, Czech Republic. [Electronic].
- [67]<https://hub.packtpub.com/working-webcam-and-pi-camera/>(consulté le 9/09/2020 14:05).
- [68] Alain MICHEL, "PREMIERS PAS AVEC UN RASPBERRY PI 3", <https://alain-michel.canoprof.fr/eleve/tutoriels/raspberry/premiers-pas-raspberrypi/res/premiers-pas-raspberrypi.pdf>; pp.1-50.
- [69] Abd Allah Ali Abd Allah, "Simply Raspberry pi", Mars 2014
- [70] Patrick Fuchs et Pierre Poulain, "Cours de Python", Université de Paris, France(28 août 2020), <https://python.sdv.univ-paris-diderot.fr/>
- [71] "Program an Arduino UNO with your Raspberry Pi"
<https://magpi.raspberrypi.org/articles/program-arduino-uno-raspberry-pi> (consulté le 20/09/2020 09:12).
- [72] Adrian Rosebrock. Eye blink detection with opencv, python et dlib.URL <https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/>.

[73]shape_predictor_68_face_landmarks

https://github.com/AKSHAYUBHAT/TensorFace/blob/master/openface/models/dlib/shape_predictor_68_face_landmarks.dat.