RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Aboubakr Belkaïd - Tlemcen -
Faculté de TECHNOLOGIE

MÉMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Automatique

Spécialité : Automatique et Informatique Industrielle

Par :

BELLAHCENE Kheir Eddine & LAKHDARI Issam

Sujet

Étude et développement d'un Eye Tracker temps réel

Soutenu à distance, le 27/06/2021, devant le jury composé de :

Mme. Choukchou-Braham Amal	Pr	Univ. Tlemcen	Présidente
Mr. Hadj-Abdelkader Amine	Pr	Univ. Tlemcen	Directeur de mémoire
Mme. Handouzi Wahida	MCB	Univ. Tlemcen	Examinatrice

Année Universitaire : 2020/2021

Étude et développement d'un Eye Tracker temps réel

BELLAHCENE Kheir Eddine LAKHDARI Issam

4 juillet 2021

Remerciments

Nous tenons à remercier en premier lieu Dieu le tout puissant et miséricordieux de nous avoir accordé la puissance et la volonté pour achever ce travail.

Nous tenons à exprimer toute notre reconnaissance à notre directeur de mémoire **Mr Hadj-Abdelkader Amine**, Professeur à l' Université de Tlemcen, on le remercie de nous avoir encadré, orienté, aidé et conseillé, nous présentons notre respect et notre gratitude.
Nous tenons à remercier **Mme Choukchou-Braham Amal**, Professeur à l'université de Tlemcen d'avoir acceptée d'être la présidente du jury, et nous tenons également à remercier **Mme Handouzi Wahida**, Maître de conférences à l'université de Tlemcen d'avoir acceptée d'éxaminer ce travail.
Nous tenons à remercier nos chers parents, nos frères et sœurs qui ont toujours été là pour nous, et toutes nos familles pour leurs encouragements et soutien pendant la réalisation de ce mémoire.
Ensuite, nous tenons à remercier **Mr Temri Chahreddine**, Docteur en médecine physique et réadaptation pour ses informations précieuses concernant les handicapés moteurs.
Enfin nous adressons un grand merci à **Mr Bellahcene Abderrezzak**, Ingénieur en informatique de nous avoir aidé avec ses conseils et ses encouragements pendant la réalisation de ce mémoire.

Kheir Eddine, **Issam**

Table des matières

Introduction générale **10**

I État de l'art **12**

I.1 Introduction 12
I.2 Définition du eye tracker 12
I.3 Historique du eye tracker 13
I.3.1 L'évolution du eye tracker 14
I.4 Description du Eye Tracking 15
I.4.1 Anatomie de l'œil humain 15
I.4.1.1 Anatomie externe : 16
I.4.1.2 Région des yeux 16
I.4.1.3 Le mouvement de l'œil 17
I.4.2 Structure des Eye Trackers 18
I.4.2.1 Eye Tracker basé sur l'écran 18
I.4.2.2 Eye Tracker sous forme des lunettes 18
I.5 Notions dans l'eye tracking 20
I.5.1 Précision et Exactitude dans l'eye tracking 20
I.5.2 La moyenne quadratique RMS 21
I.5.3 Temps de latence total du système eye tracker 22
I.6 Modèles des Eye Trackers 22
I.6.1 Modèles commerciaux 22
I.6.1.1 Tobii Pro Fusion 22
I.6.1.2 EyeLink 1000 Plus 24
I.6.1.3 Eyegaze Edge 25
I.6.1.4 Les lunettes DIKABLIS 3 26
I.6.2 Prototypes de recherche 27
I.6.2.1 openEyes 27
I.6.2.2 EyeSecret 28
I.6.2.3 InvisibleEye 28
I.7 Les domaines d'application de eye tracker 29
I.7.1 La psychologie et le médicale 30

I.7.1.1 La psychologie cognitive et les sciences cognitives 30
I.7.1.2 La psychologie du développement 30
I.7.1.3 La psychologie expérimentale 31
I.7.1.4 La psycholinguistique et la lecture 31
I.7.1.5 L'ophtalmologie 31
I.7.2 Interface homme machine 31
I.7.3 Commercialisation 32
I.7.4 Sites web 33
I.7.5 Jeux vidéo et réalité virtuelle 34
I.7.6 Automobile 35
I.8 Conclusion 36

II Méthodologie du Eye - Tracking **37**
II.1 Introduction 37
II.2 Vision par ordinateur avec OpenCV 37
II.2.1 Traitement d'images 39
II.2.2 Traitement d'image analogique 40
II.2.3 Traitement d'image numérique 40
II.3 Traitement d'images 41
II.3.1 Filtre bilatéral 41
II.3.2 Érosion 41
II.3.3 Image en niveau de gris 44
II.3.4 Le seuillage 44
II.3.5 La détection de visage 46
II.3.6 Isolation des yeux 47
II.3.6.1 Détection des yeux 47
II.3.6.2 Région des yeux 47
II.3.6.3 Application de masque 48
II.3.6.4 Isolation 48
II.3.7 Détection de l'iris 49
II.3.7.1 Détection de contour 49
II.3.7.2 Le seuillage automatique 50
II.4 Estimation de la position de la pupille 51
II.4.1 Les coordonnés des deux pupilles 51
II.4.2 Marquage de la position des deux pupilles 51
II.5 Estimation de la position (x,y) des yeux sur l'écran 52
II.5.1 Calibration 52
II.6 Les coordonnées (x,y) de la pupille sur l'écran 53
II.7 Lissage 54

II.8 Conclusion 54

III Applications et Résultats **55**

III.1 Introduction 55
III.2 Contrôle de curseur avec les yeux 55
III.3 Click par clignement des yeux 56
III.4 Application de surveillance du conducteur 57
III.4.1 Détection de la fatigue du conducteur 57
III.4.2 La surveillance de la direction de regard du conducteur 57
III.5 Application Médicale 58
III.5.1 Présentation de l'application 58
III.5.2 Fonctionnement de l'application 59
III.5.3 Tkinter 60
III.5.4 Thread 61
III.5.5 Le son dans l'application 62
III.5.6 Le clavier visuel 63
III.5.6.1 Fonctionnement du clavier 64
III.5.6.2 Description des symboles fréquents sur le clavier 64
III.5.7 L'exécutable de l'application 65
III.6 Résultat et Discussion 65
III.6.1 Discussion 68
III.7 Conclusion 68

Conclusion générale et perspectives **70**

Annexe **72**

Table des figures

I.1 Les électrodes fixées sur la peau autour des yeux [23]. 13
I.2 Photochronograph fonction avec la réflexion de la cornée [41] 14
I.3 Anatomie externe de l'œil [28]. 16
I.4 L'image des yeux avec le filtre IR [28]. 17
I.5 Les muscles de l'œil [26]. 17
I.6 La structure d'eye tracker basé sur l'ecran Tobii Pro Fusion [11]. 18
I.7 La structre d'eye tracker sous forme des lunette Tobii Pro Glasses 3 [12]. . 19
I.8 L'unité d'enregistrement [12]. 19
I.9 Illustration des différents comportements typiques entre l'exactitude et la précision [1]. 20
I.10 L'angle du regard α [1] 21
I.11 Illustration de la précision mesurée à partir de la moyenne quadratique "RMS" d'un échantillon à l'autre [1]. 21
I.12 Temps de latence du système de suivi oculaire (eye tracker) de la série Tobii TX [15] 22
I.13 Tobii Pro Fusion [16]. 23
I.14 EyeLink 1000 Plus [9]. 24
I.15 Eyegaze Edge® [3]. 25
I.16 Les lunettes DIKABLIS 3[6]. 26
I.17 Le prototype du openEyes [30]. 27
I.18 (c) l'image d'une scène obtenue par l'eye tracker openEyes et (d) l'image de l'œil droit de l'utilisateur illuminé avec une lumière infrarouge [25]. . . 27
I.19 Le protoype du EyeSecret [44]. 28
I.20 Le prototype du invisbleEye [42]. 29
I.21 Vue d'ensemble du réseau de neurone utilisé dans InvisibleEye pour l'estimation du regard basée sur l'apprentissage [42]. 29
I.22 IHM Samsung avec une souris commandée par l'eye tracker pour les personnes handicapés moteurs [18]. 31
I.23 Un système de eye tracker proposé par Tobii [19]. 32
I.24 Modèle de eye tracker Tobii pour la commercialisation [26]. 33
I.25 Application de eye tracker dans un sitweb [21]. 34

I.26 Application de eye tracker dans les jeux [19]. 35
I.27 La position de l'eye tracker dans un véhicule [36]. 36

II.1 Représentation des pixels dans une image [29] 39
II.2 (a) Image en entrée du filtre bilatéral et (b) Image en sortie du filtre bilatéral. 41
II.3 Illustration de la méthode d'érosion de $I \ominus E$ dans les pixels $I_{1,1}$ et $I_{5,5}$. . . 42
II.4 Le résultat de la figure II.3. 43
II.5 On remarque que l'opération de l'érosion a éliminé les petits bruits blancs présent dans l'image (a) sous forme d'un petit rectangle blanc due à la réflexion de la lumière du laptop dans l'œil, après trois itérations le rectangle a complètement disparu (b). 43
II.6 (a) Image originale (b) Image en niveau de gris. 44
II.7 Le graphe qui illustre le seuil en bleu (échelon unité) avec les valeurs d'intensité des pixels de l'image source triangle en rouge [2]. 45
II.8 Le graphe de la figure (II.7) aprés l'application du seuillage binaire [2]. . . 45
II.9 (a) image originale et (b) image originale après seuillage binaire. 46
II.10 Image qui présente la détection de visage avec l'utilisation de la bibliothèque dlib "Shape predictor 68 face Landmark"[34]. 47
II.11 Image qui présente les régions des yeux en bleu. 48
II.12 Image qui présente l'utilisation de masque sur le visage 48
II.13 Image qui présente juste l'oeil isoler dans une image 49
II.14 Image de l'oeil en niveau de gris avant filtrage. 49
II.15 Image de l'oeil (la figure II.14) aprés filtrage. 49
II.16 Image de l'oeil (la figure II.15) aprés l'érosion. 49
II.17 Image de l'oeil (la figure II.15) aprés seuillage. 50
II.18 Le contour de l'iris. 50
II.19 Marquage de la position des deux pupilles en vert. 52
II.20 Image qui présente les étapes(points) de calibration 52
II.21 La relation pour trouver Les coordonnées (x,y) de la pupille sur l'écran . . 53

III.1 Landmarks des yeux 56
III.2 La ligne verticale des yeux en vert. 56
III.3 L'isolation de l'oeil et la séparation de sa figure en deux cotés 58
III.4 L'interface graphique de l'application médicale. 59
III.5 Le chemin du défilement semi-automatique du curseur dans un seul sens suivant les flèches en rouges 59
III.6 Un processus avec deux threads [14] 62
III.7 Lecture des sons de click puis de l'alarme avec l'utilisation des temps de retard pour que le programme attend que la lecture des sons sera terminé. 63

III.8 La lecture du son de bip de sélection du clavier visuel. 63
III.9 Le clavier visuel de communication par clignement des yeux. 64
III.10Les symboles fréquents sur le clavier visuel de communication. 64
III.11Le fichier exécutable de l'application médicale. 65
III.12Résultat d'écriture de la chaine de caractère MAL TETE par le clavier visuel. 66
III.13Résultat d'écriture de la chaine de caractère DOULEUR DOS par le clavier visuel. 66
III.14Résultat d'écriture de la chaine de caractère BESOIN HYGIENE par le clavier visuel. 67
III.15Résultat d'écriture de la chaine de caractère AUTOMATIQUE par le clavier visuel. 67

Liste des tableaux

I.1 Les parties de l'unité de tête [12]. 19
I.2 Les spécifications techniques du Tobii Pro Fusion [11]. 23
I.3 La configuration du Tobii Pro Fusion [11]. 24
I.4 Les spécifications techniques du EyeLink 1000 Plus [10]. 25
I.5 Les spécifications techniques du Eyegaze Edge® [8]. 26
I.6 Les spécifications techniques du DIKABLIS 3 [7]. 27

III.1 Les temps de sélection des besoins présent dans l'application par une personne entrainé sur cette interface graphique. 65
III.2 Temps de réponse du clavier visuel en fonction des chaines de caratère . . . 67

Table des Abréviations

IR :	Infra Rouge
IHM :	Interface Homme machine
RMS :	Root-Mean-Square(Moyenne quadratique)
Open**CV :**	Open Source **Computer Vision** Library
fps :	Frames per second(images par seconde)

Introduction générale

Avec l'invention de l'ordinateur au milieu du dernier siècle, le besoin des interfaces de communication pour les utilisateurs est devenu incessant. Au départ, les experts utilisaient le télétype pour s'interfacer avec l'ordinateur. En raison des énormes progrès de la technologie informatique au cours des dernières décennies, les capacités des ordinateurs ont énormément augmenté et l'utilisation d'un ordinateur est devenue une activité normale pour tout le monde. Avec toutes les possibilités qu'un ordinateur peut offrir, les humains et leur interaction avec les ordinateurs sont désormais un facteur limitant. Cela a donné lieu à de nombreuses recherches dans le domaine des IHM (interface homme-machine) visant à les rendre plus faciles, plus intuitives et plus efficaces. L'interaction avec les ordinateurs ne se limite plus aux claviers et aux imprimantes. Différent types de dispositifs de pointage, de surfaces tactiles, d'écran de haute résolution, de microphones et de haut-parleurs sont des dispositifs normaux pour l'interaction informatique tels que l'interaction vocale, la saisie par les gestes ou par les objets tangibles munis des capteurs. Une autre modalité d'entrée est le regard qui trouve aujourd'hui son application dans divers systèmes qui l'utilisent généralement comme étant une entrée qui pourrait servir les méthodes d'interaction et, au-delà, le domaine de l'accessibilité. Les eye-trackers sont des appareils qui peuvent estimer la direction du regard d'une personne, les premiers eye-tracker ont été développés pour l'exploitation scientifique dans des environnements contrôlés ou des laboratoires. Les données sur le regard oculaire ont été utilisées en ophtalmologie, en neurologie, en psychologie et dans des domaines connexes pour étudier la caractéristique et les anomalies oculomotrices et leurs relations avec la cognition et les états mentaux. Il existe des applications plus récentes de la recherche en Eye-Tracking en marketing, en automobile, en jeux et réalité virtuelle et en publicité, ainsi qu'en ingénierie des facteurs humains pour évaluer les interfaces informatiques et les sites web. Selon Duchowski [27] les applications de suivi oculaire peuvent être classées comme applications de diagnostique ou applications interactives. Les applications de diagnostique utilisent les données du regard comme preuve quantitative des processus visuels et attentionnels de l'utilisateur. Les applications interactives utilisent les données du regard pour répondre ou interagir avec l'utilisateur en fonction des mouvements oculaires observés. De nombreuses techniques traditionnelles de suivi du regard sont intrusives, c'est-à-dire qu'elles nécessitent un certain équipement pour être mis en contact physique avec l'utilisateur. Ces techniques incluent, par exemple,

les lentilles de contact, les électrodes et les dispositifs montés sur la tête. Les techniques non intrusives (ou techniques à distance) sont principalement basées sur la vision, c'est-à-dire qu'elles utilisent des caméras pour capturer des images de l'œil. Nous proposons dans ce mémoire d'utiliser ces méthodes pour réaliser un suivi oculaire efficace pouvant être utilisé dans diverses applications. Notre mémoire est structuré en trois chapitres : dans le premier chapitre, nous présentons ce qu'est l'eye-tracking, son historique, les modèles des eye trackers et les domaines d'applications des systèmes de suivi oculaire. Dans le deuxième chapitre, nous présentons la méthodologie de l'eye-tracking, pour laquelle nous détaillerons les différentes techniques de traitement d'images et de vision par ordinateur utilisées pour programmer cette fonctionnalité. Finalement, dans le dernier chapitre, nous présentons les deux applications que nous avons réalisées, la première est la surveillance du conducteur d'automobile et la deuxième est l'application médicale d'aide à la communication pour personnes handicapés moteurs. Nos deux applications sont basées sur la technologie de l'eye-tracking et sont en relation directe avec les domaines d'applications de l'automobile et du génie biomédical.

Chapitre I

État de l'art

I.1 Introduction

Le domaine de la poursuite du regard (Eye tracking) a beaucoup évolué depuis les premières études jusqu'à ces dernières années. Cette évolution est due aux nombre croissant des applications de cette technique, que ce soit dans le domaine de l'automatique (Exp. en conduite automobile), du biomédical (Exp. utilisation des aides techniques pour personnes handicapés moteurs) et des technologie de l'information général (Exp. utilisation d'ordinateur, navigation sur internet, ...).

Dans ce chapitre nous allons donner une introduction sur le système eye tracker en définissant et décrivant ce système puis nous allons citer quelques modèles commerciaux des eye trackers qui existent dans le marché, proposés par différentes sociétés, et les prototypes de recherche. Ensuite nous présentons les différents domaines d'applications tels que le domaine de l'automobile, du biomédical, ...Ces domaines qui utilisent le suivi oculaire sont devenus de plus en plus populaires dans les domaine des interfaces homme-machine IHM [31] surtout leurs impacts sur la vie des personnes à mobilité réduite et plus exactement les personnes handicapés moteurs qui ne peuvent pas interagir avec les IHM. Ce système a donné une certaine indépendance a ces personnes pour faciliter leurs vie.

I.2 Définition du eye tracker

Le terme suivi oculaire tel qu'il est utilisé ici signifie l'estimation de la direction du regard de l'utilisateur. Dans la plupart des cas, l'estimation de la direction du regard signifie l'identification de l'objet sur lequel le regard tombe, l'interprétation de la direction du regard est plus complexe pour le suivi oculaire dans le monde virtuel 3D et devient difficile lors de l'interaction avec le monde réel. Les suivis oculaires diffèrent par le degré de liberté qu'ils peuvent suivre. Les eye trackers simples ne rapportent que la direction du regard par rapport à la tête (système montés rigidement sur la tête) ou pour une position

fixe du globe oculaire (système nécessitant une fixation de la tête). Des systèmes plus sophistiqués permettent des mouvements libres de la tête devant un système stationnaire. De plus, les eye trackers portables destinés à être utilisés dans les mondes virtuels 3D doivent signaler la direction du regard dans l'espace et pas seulement par rapport à la tête. La plupart des suiveurs oculaires basés sur la vidéo fournissent non seulement la direction du regard mais également la taille de la pupille. L'élargissement et le rétrécissement de la pupille est une réponse assez intéressante pour la recherche. Cependant, comme la fonction principale de la pupille est la régulation de la quantité de lumière pénétrant dans l'œil, une telle recherche nécessite des conditions d'éclairage stable pour le suivi oculaire. Pour l'interaction avec l'ordinateur traditionnel, l'objet regardé est identifié par les coordonnées du regard sur l'écran, l'interprétation de la direction de regard devient plus difficile lors des interactions avec l'environnement physique. Il existe trois méthodes différentes pour l'eye tracker. La première méthode consiste à fixer un capteur (des lentilles) sur l'œil, la deuxième méthode est l'électrooculographie, pour laquelle les électrodes sont fixées sur la peau autour des yeux afin de mesurer le champ électrique comme le montre la figure I.1 La troisième méthode utilise une caméra et l'estimation de la direction du regard est faite par le traitement des images [23].

FIGURE I.1 – Les électrodes fixées sur la peau autour des yeux [23].

I.3 Historique du eye tracker

La technologie "eye tracker" a connu une très grande croissance ces dernières années mais les premières études de cette technique sont apparues à la fin du dix-huitième siècle avec l'utilisation seulement de l'observation directe pour comprendre les schémas de lecture qui apparaissent : on tentait de détecter une série de sauts et de fixations oculaires et non pas la détection de la position de l'œil ou le balayage régulier comme les technologies des dernières années "tobii eye tracker" par exemple. A partir du vingtième siècle la technologie est devenue plus capable à utiliser dans le domaine des "eye tracker" et contient des techniques capables d'enregistrer les mouvements oculaires. Le premier "eye tracker" qui a été monté sur la tête a libéré les participants des mouvements contraints de la tête qui ont présenté un grand problème et qui étaient subies auparavant. Avec le temps

les méthodologies ont été améliorées par des chercheurs et des experts dans ce domaine. Aujourd'hui la technologie de suivi oculaire va des systèmes portables aux systèmes mobiles qui permettent un mouvement libre de tête dans le champ de vision du caméra. la flexibilité a ouvert des portes à des applications de la technique vers l'interaction avec l'environnement à travers les yeux. Les principaux points qui s'intéressent les chercheurs maintenant c'est sa validité, sa précision et sa fiabilité dans les étalonnages et l'analyse des données, et probablement le coût d'acquisition de l'appareil pour une gamme d'utilisation dans divers domaines d'applications.

I.3.1 L'évolution du eye tracker

les premières descriptions qualitatives des mouvements oculaires remontre à la fin du dix-huitième siècle, Wells utilise des images également appelées images fantômes, qui apparaissent dans la perception visuelle après avoir regardé quelque temps au même endroit pour décrire le mouvement des yeux. Au 19e siècle, javal et lamare ont observé les mouvements oculaires pendant la lecture. Ils ont utilisé un couplage mécanique des yeux et des oreilles à l'aide d'un élastique pour rendre les mouvements oculaires audibles [31]. Ahrens, Delabarre et Huey ont été les premiers à essayer d'enregistrer les mouvements oculaires transférant les mouvements sur une surface recourverte de suie par des petits leviers fixés au globe oculaire. Dodge et Cline ont fait les premières mesures discrètes en 1901. Ils ont utilisé une méthode photographique et des reflets lumineux de l'œil en enregistrant les mouvements oculaires dans la direction horizontal seulement. Le premier appareil de Dodge a vu une ligne verticale de lumière rebondir sur une corné, tombant ainsi sur une fente horizontale. La plaque photographique se déplace verticalement derrière la fente est régulée par la fuite d'air d'un cylindre (la figure I.2), La plaque peut montrer à la fois le mouvement horizontal des yeux (sur l'axe x) et le temps (sur l'axe y).

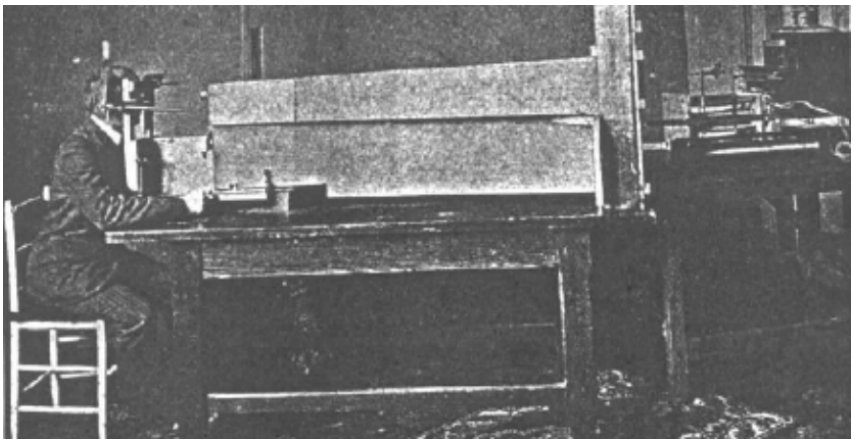


FIGURE I.2 – Photochronograph fonction avec la réflexion de la cornée [41]

Quelques années plus tard judd,McAllister et Steel ont appliqué la photographie cinématographique à l'enregistrement des mouvements oculaires. L'invention de la photogra-

phie cinématographique a donné la possibilité d'analyser image par image le mouvement des yeux et a permis une recherche quantitative sur une base solide. Miles Tkinter a étudié les mouvements oculaires en lecture dans les années 1930 et étudié l'effet de la police, la taille de la police et de la mise en page sur la vitesse de lecture. En 1939, Jung a mesuré les mouvements oculaires verticaux et horizontaux simultanément avec des électrodes appliquées sur la peau près des yeux, cette méthode également appelée electrooculographie (EOG) mesure les champs électriques du globe oculaire qui est un dipôle. La méthode a également donné les premières possibilités théoriques de traitement en temps réel des données du regard en utilisant l'électronique analogique. En 1947, Paul Fitts, qui devient plus tard célèbre pour sa loi Fitts, a utilisé des caméras cinématographiques pour enregistrer les mouvements oculaires des pilotes de l'armée de l'air lors de l'atterrissage de leur avion. Son intérêt était la façon dont les pilotes utilisent leurs commandes de cockpit. Il s'agissait de la première méthode utilisant le suivi oculaire. En 1948, Hartbridge et Thopson ont inventé le premier "eye tracker" monté sur la tête avec la possibilité des mouvement de la tête. Dans les 1970, la technologie de eye tracker est améliorée et a fourni une meilleur précision et on peut dissocier l'œil suivant les mouvements de la tête par les multiples réflexions sur l'œil (Cornsweet and crane, 1973). En 1980 les mini-ordinateurs sont devenus suffisamment puissants pour effectuer un suivi oculaire en temps réel, ce qui a donné la possibilité d'utiliser des suiveurs oculaires basés sur les vidéos pour l'interaction entre l'homme et l'ordinateur. Bolt a présenté une telle vision en 1981. c'était aussi le moment des premiers eye trackers pour aider les utilisateurs handicapés. Depuis les années 1990 la communauté de recherche sur le suivi oculaire s'est développée à mesure que les spécifications des appareils se sont également améliorées, tandis que leur prix a baissé. Aujourd'hui le prix de l'eye tracker est de 95 dollar. La réflexion cornéenne du centre de la pupille(PCCR) est la technique la plus couramment utilisé sur les ordinateurs du bureau et les ordinateurs de portable ou un éclairage infrarouge ou proche infrarouge est utilisé pour réfléchir la lumière des deux parties de l'œil ainsi que le logiciel de "eye work" ou " tobii studio" est aussi utilisé pour interpréter les données [41].

I.4 Description du Eye Tracking

I.4.1 Anatomie de l'œil humain

L'œil c'est un système visuel qui représente l'un des cinq organes de sens du corps (organe de la vue) c'est un formidable organe qui est le plus développé et extraordinaire par des informations qu'il nous fournit soit du coté quantitatif ou qualitatif malgré que le diamètre de l'œil sagittal ou antérieur ne dépasse pas 24,5 mm et le diamètre transversal 24 mm et avec un poids de 7 grammes un coup d'oeil rapide suffit pour connaitre la position, la taille, la forme, la couleur et la texture des objets, qu'ils soient immobiles ou

en déplacement même leur direction et leur vitesse relative. L'œil est constitué de fibre optique, nerfs optique, membranes ... qui travaillent conjointement pour filtrer la lumière et ajuster la mise au point, les récepteurs de cette lumière sont les photorécepteurs de la rétine qui est tissu neuronal qui recouvre le fond de l'œil pour qu'elle puisse voir dans la plus faible lumière des étoiles(vision scotopique), et a la plus éclatante lumière du jour (vision photopique).

I.4.1.1 Anatomie externe :

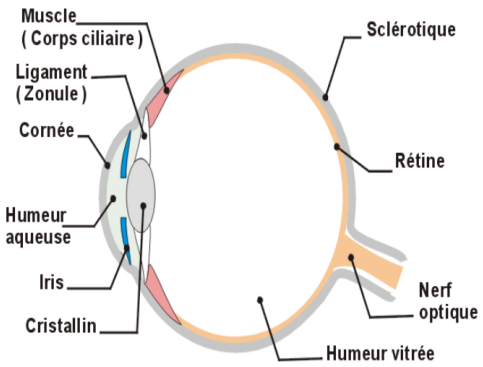


FIGURE I.3 – Anatomie externe de l'œil [28].

La figure I.3 montre l'anatomie externe de l'œil qui contient plusieurs parties qui aident dans la formation de l'image dans la fovéa. La mise au point se fait principalement par la cornée qui est la façade de l'œil. L'iris régule la quantité de la lumière et agit comme un diaphragme ajustant la taille de la pupille pour contrôler la quantité de la lumière. La mise en point se fait par l'objectif situé derrière la pupille grâce à un processus connu sous le nom d'accommodement, cela aide à former une image nette même si l'objet mis au point est proche ou éloigné. La lumière focalisée tombe sur la fovéa et les photorécepteurs de la fovéa convertissent la lumière en un signal électrique qui est ensuite transmis au visuel cortex via le nerf optique, c'est toute une opération compliquée pour la formation d'une image nette et sous une vitesse extraordinaire [22].

I.4.1.2 Région des yeux

La figure I.4 représente l'image de l'œil avec le filtre infrarouge (NIR). Dans l'image visible, la frontière entre l'iris et la sclérotique est plus visible que la frontière entre la pupille et l'iris. La plupart des chercheurs ou programmeurs de "eye tracking " utilisent ces bornes et ces frontières pour faire le suivi du regard. On filtre la frontière entre l'iris et la pupille et plus important dans la méthodologie de "eye tracking " parce qu'il facilite la programmation de cette dernière et on trouve la plupart des suiveurs oculaires commerciaux qui utilisent la limite de la pupille suivant une certaine méthodologie pour estimer la position de regard. Dans la méthodologie de "eye tracking" la précision de la

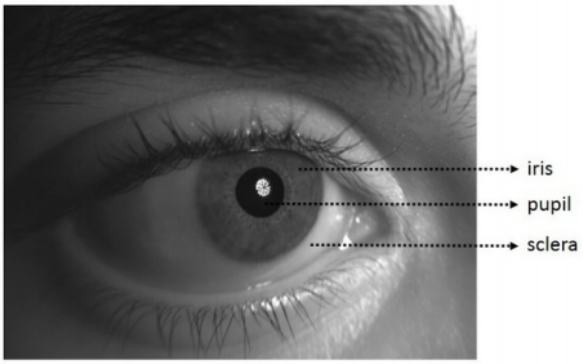


FIGURE I.4 – L'image des yeux avec le filtre IR [28].

limite entre l'iris et la pupille dans les vidéos NIR est bien plus que la frontière entre la sclérotique et l'iris dans les vidéos visibles. Dans une autre part l'utilisation des images et des vidéos visibles ont l'avantage à ne pas utiliser un matériel spécial ou les composants d'éclairage [22].

I.4.1.3 Le mouvement de l'œil

Dans une vue très simple et en termes d'ingénierie, l'œil peu être vu comme une caméra de haute qualité avec différentes techniques qui servent à stabiliser l'image. On trouve six muscles illustrés à la figure I.5 ci-dessus reliant l'œil à la tête. Les muscles sont organisés pour donner à l'œil 3 degrés de liberté. Un muscle est responsable du mouvement horizontal, un autre muscle contrôle le mouvement vertical et le dernier permet le mouvement de la rotation autour de la direction de la vue on compensant tous les mouvements de la tête pour accomplir cette tâche, les nerfs contrôlant les muscles oculaires sont étroitement liés à l'organe d'équilibre et c'est le cerveau qui traite les informations obtenues pour que nous puissions reconnaitre les objets [37].

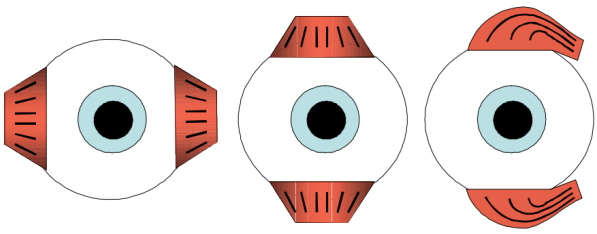


FIGURE I.5 – Les muscles de l'œil [26].

I.4.2 Structure des Eye Trackers

I.4.2.1 Eye Tracker basé sur l'écran

La figure I.6 montre la structure interne d'un eye tracker Tobii Pro Fusion basé sur l'écran.

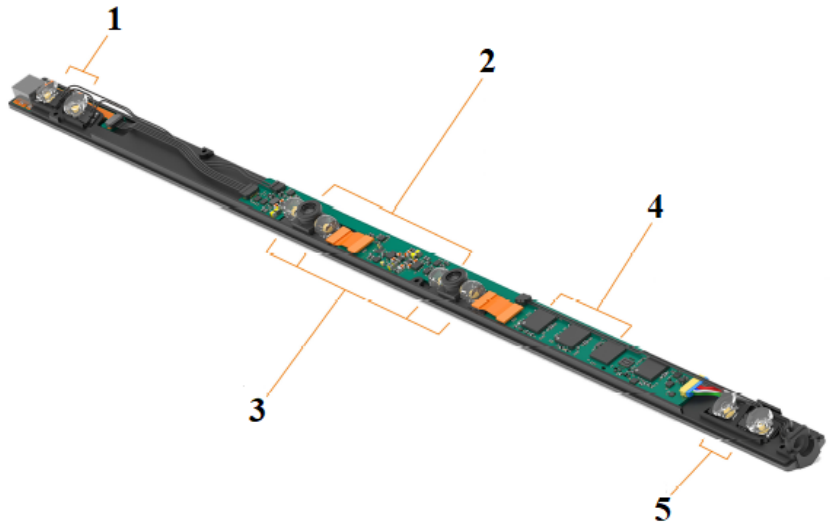


FIGURE I.6 – La structure d'eye tracker basé sur l'ecran Tobii Pro Fusion [11].

- **1 et 5 : Modules d'éclairage pour la pupille sombre**
  Les modules d'éclairage pour la pupille sombre sont à base de la lumière infrarouge.
- **2 : 2×Modules Caméras Tobii EyeSensor™**
  Les deux caméras conçues par tobii à haute sensibilité et résolution pour le suivi des yeux .
- **3 : Module d'éclairage pour la pupille lumineuse**
  Les modules d'éclairage pour la pupille lumineuse sont à base de la lumière infrarouge.
- **4 : 3×Modules de circuit intégré Tobii EyeChip™ASIC**
  L'unité de traitement entièrement embarquée en permettant le système de travailler d'une manière autonome pour minimiser la charge sur la CPU du système.

I.4.2.2 Eye Tracker sous forme des lunettes

La figure I.7 montre la structure interne d'un eye tracker sous forme des lunette de la société Tobii, généralement ce genre de modèle est composé de deux unités :

1. **L'unité de tête**
   L'unité de tête est un appareil de mesure très sophistiqué. Il est composé de plusieurs capteurs très sensibles.

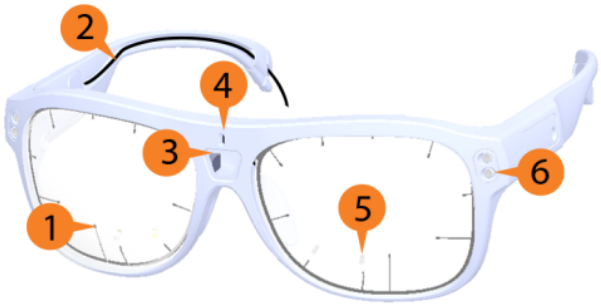


FIGURE I.7 – La structre d'eye tracker sous forme des lunette Tobii Pro Glasses 3 [12].

TABLE I.1 – Les parties de l'unité de tête [12].

1. Illuminateurs infrarouges (x 8 par œil)	Illumine les yeux pour aider les capteurs oculométriques.
2. Câble de l'unité de tête	Se connecte à l'unité d'enregistrement pour sauvegarder les données collectées.
3.Caméra de scène à haute définition	Capture une vidéo Full HD de ce qui se trouve devant le participant.
4. Microphone	Capte le sons du participant et le son d'environnement.
5. Caméras de suivi des yeux (x 2 par œil)	Enregistre l'orientation et les mouvements des yeux.
6. Fixation des accessoires	Utilisé pour fixer certains accessoires en option.

2. **L'unité d'enregistrement**

L'unité d'enregistrement (La figure I.8) est un petit ordinateur qui contrôle l'unité de tête, elle enregistre, stocke et traite les données de suivi oculaire, le son et la vidéo de la caméra de la scène sur une carte SD amovible. L'unité d'enregistrement est équipée d'une batterie Li-ion remplaçable et rechargeable qui alimente l'unité d'enregistrement et l'unité de tête.

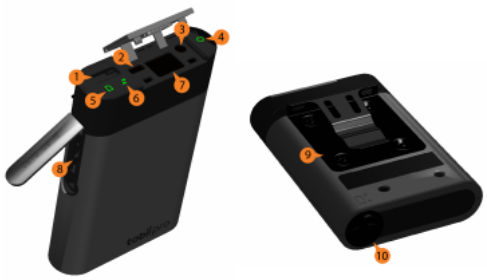


FIGURE I.8 – L'unité d'enregistrement [12].

I.5 Notions dans l'eye tracking

Avant d'entamer la partie des modèles des eye trackers nous avons besoin d'expliquer quelques notions fondamentales dans le domaine de suivi oculaire "eye tracking" selon le constructeur des eye trackers Tobii.

I.5.1 Précision et Exactitude dans l'eye tracking

L'exactitude et la précision sont utilisées comme des indicateurs de la validité des données de l'eye tracker. Un système avec une bonne exactitude et une bonne précision (c) dans la figure I.9 fournira des données plus valides car il est capable de décrire de manière véridique l'emplacement du regard d'une personne sur un écran, ces concepts sont très importants pour comprendre comment fonctionne un eye tracker et comment évaluer la qualité des données d'eye tracking enregistrées [5].

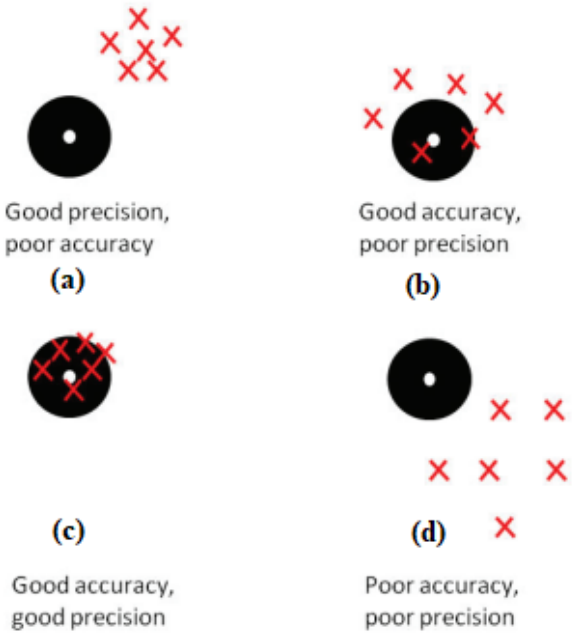


FIGURE I.9 – Illustration des différents comportements typiques entre l'exactitude et la précision [1].

La figure I.9 montre tous les cas possibles qui peuvent exister entre la précision et l'exactitude (accuracy) en terme de l'évaluation de l'eye tracker [1] :

- **L'exactitude (accuracy) :** c'est la différence moyenne entre la position réelle où la personne regarde (les cercles noirs) dans la figure I.9 et les points de regard mesurés par l'eye tracker (les croix rouges).
- **La précision :** Le mésure de précision montre la capacité de l'eye tracker à reproduire de manière fiable la même mesure du point de regard c'est-à-dire qu'elle mesure la variation des données enregistrées via la moyenne quadratique (RMS) des échantillons successifs.

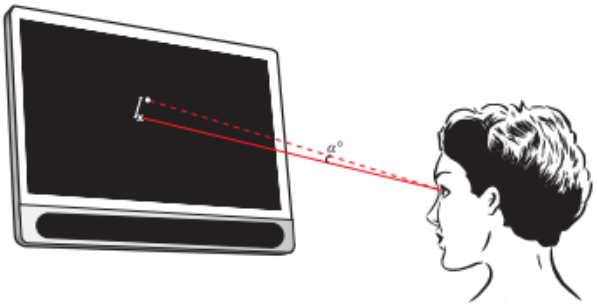


FIGURE I.10 – L'angle du regard α [1]

La figure I.10 montre l'angle du regard qui est exprimé comme la déviation en degrés entre les deux lignes la ligne rouge pointillée représente la direction réelle du regard par l'utilisateur et la ligne rouge pleine qui représente le point de regard mesuré par l'eye tracker où l'exactitude et la précision sont mesurés en fonction de l'angle du regard [1].

I.5.2 La moyenne quadratique RMS

La précision est calculée par le biais de la moyenne quadratique RMS des points de données successifs d'angle visuel θ_i en degrés entre les échantillons successifs (x_1, y_1) à (x_{i+1}, y_{i+1}) i.e la différence angulaire entre l'échantillon 1 et l'échantillon 2 (La figure I.11), à la fois pour chaque œil individuellement et comme moyenne des deux par [1] :

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\theta_i^2} = \sqrt{\frac{\theta_1^2 + \theta_2^2 + \cdots + \theta_n^2}{n}} \tag{I.1}$$

Avec :

- θ : l'angle visuel en degrés.
- n : le nombre d'échantillons dans l'ensemble de données.

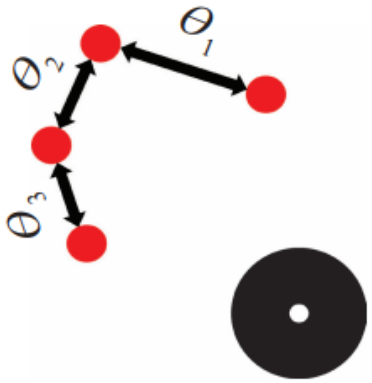


FIGURE I.11 – Illustration de la précision mesurée à partir de la moyenne quadratique "RMS" d'un échantillon à l'autre [1].

I.5.3 Temps de latence total du système eye tracker

La figure I.12 montre le temps de latence total d'un système d'eye tracking qui correspond à l'intervalle mesuré entre le moment où l'image est capturée par le capteur oculaire (la caméra) jusqu'au moment où les données valides sur le regard sont envoyées par le serveur TET au réseau ou à une application de suivi des yeux, ce temps est la somme des temps d'exposition de la caméra, de transfert, de calcul et des retards dans le système.

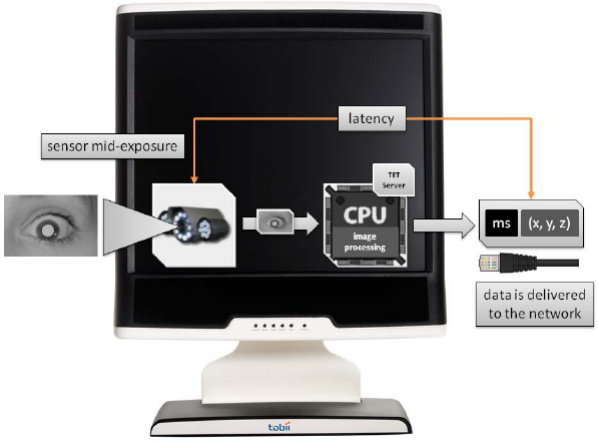


FIGURE I.12 – Temps de latence du système de suivi oculaire (eye tracker) de la série Tobii TX [15]

.

I.6 Modèles des Eye Trackers

Il existe deux types de modèles de l'Eye Tracker le premier est sous forme des lunettes et le deuxième c'est un dispositif basé sur l'écran qu'on veut commander par exemple l'écran de l'ordinateur.

I.6.1 Modèles commerciaux

I.6.1.1 Tobii Pro Fusion

Tobii Pro Fusion montré dans la figure I.13 est un produit de la nouvelle génération d'eye trackers de Tobii Pro AB. Avec une fréquence d'échantillonnage allant jusqu'à 250Hz [16], doté d'une technologie bi caméra développé par la société Tobii Pro AB la société parent qui contient plusieurs filiales de recherche en matière d'eye tracking depuis 2001.

FIGURE I.13 – Tobii Pro Fusion [16].

I.6.1.1.1 Les Spécifications techniques du Tobii Pro Fusion

1. **Les techniques d'eye tracking utilisés dans Tobii Pro Fusion**
   - Suivi vidéo de l'œil par réflexion de la pupille et de la cornée, avec deux modes d'éclairage sombre et lumineux de la pupille [11].
   - Deux caméras qui capturent des images stéréos des deux yeux pour une mesure précise et robuste du regard et de la position des yeux dans l'éspace 3D ainsi que du diamètre de la pupille [11].
2. **Les specifications du Tobii Pro Fusion**

TABLE I.2 – Les spécifications techniques du Tobii Pro Fusion [11].

Fréquence d'échantillonnage	60, 120 et 250 Hz ou 60 et 120 Hz,selon la version du Produit Tobii Pro Fusion
Précision	0.04°RMS dans des conditions optimales 0.2° RMS dans des conditions optimales (signal brut)
Exactitude"Accuracy"	0.3° RMS dans des conditions optimales
Suivi des yeux binoculaire	Oui
Temp de latence totale du système	3 images ($<$ 12 ms dans 250 Hz)
Temps de récupération du clignement des yeux	1 image (immédiatement)
Temps de récupération du regard	250 ms
Données en sortie d'un echantillon	Origine du regard Point de regard Diamètre de la pupille
Synchronisation du temps Entre Eye Tracker - Client	Synchronisation entre le domaine temporel de l'eye tracker et le domaine temporel de l'ordinateur client intégrée
Traitement des données	3 Tobii EyeChip™ASIC, avec traitement des données entièrement intégré
Caméras d'Eye Tracking	2 Modules Tobii EyeSensor™
Mode d'éclairages	Module d'éclairage de la pupille sombre Module d'éclairage de la pupille Lumineuse
Consommation électrique	Puissance nominale 4.3 W Puissance maximale 9 W

3. **La configuration du Tobii Pro Fusion**

TABLE I.3 – La configuration du Tobii Pro Fusion [11].

L'intervalle de mouvement de tête autorisé(à une distance de 65 cm)	Largeur × hauteur : 40 cm × 25 cm Au moins un œil est suivi
L'intervalle de mouvement de tête autorisé(à une distance de 80 cm)	Largeur × hauteur : 45 cm × 30 cm Au moins un œil est suivi
Distance de fonctionnement (monté sur l'écran)	50 − 80 cm de l'eye tracker
Taille optimale de l'écran	24"(format 16 : 9)

I.6.1.2 EyeLink 1000 Plus

La figure I.14 montre EyeLink 1000 Plus c'est un modèle d'eye tracker développé par la société SR Research en 2013, et possède des spécifications techniques avec un faible bruit spatial et des taux d'échantillonnage élevés et dispose de plusieurs options de montage pour l'utilisation (par exemple, montage sur le bureau, sur le bras... etc) [9].

FIGURE I.14 – EyeLink 1000 Plus [9].

1. **Les Spécifications techniques du EyeLink 1000 Plus**
   Pour ce modèle il comporte plusieurs options de montage pour l'utilisation on s'intéresse à l'option montée sur le bureau.

TABLE I.4 – Les spécifications techniques du EyeLink 1000 Plus [10].

Mode de suivi des yeux	Monoculaire /Binoculaire
Fréquences d'échantillonnages	$250, 500, 1000, 2000$ Hz
Exactitude moyenne	Jusqu'à $0,15°$ ($0,25°$ à $0,5°$en général)
Temps de récupération du clignement des yeux	1 msec à 1 kHz 0.5 msec à 2 kHz
Résolution de la pupille	$0,1\%$ du diamètre
L'intervalle de mouvement de tête autorisé	± 25 mm horizontal ou verticale
Distance de fonctionement optimale	$40 - 70$ cm
Modes de detection de la pupille	Par ajustement de la centroïde ou de l'ellipse

I.6.1.3 Eyegaze Edge

La figure I.15 montre La tablette de communication Eyegaze Edge contrôlée par l'eye tracker développé par la société américaine EYE GAZE INC (anciennement LC Technologies) qui a vu le jour en 1987, ce système eye tracker - tablette à plusieurs options parmi ces options on cite quelques exemples [3] :

- Effectuer des appels et envoyer des messages.
- Contrôler n'importe quelle application dans la tablette.
- Contrôlez l'environnement [1]avec des fréquences infrarouges et des ondes radio.

FIGURE I.15 – Eyegaze Edge® [3].

Ce modèle d'eye tracker a pour but de faciliter la vie des personnes à mobilité réduite qui n'ont pas la capacité de bouger leurs mains pour manipuler la souris ou le clavier d'un ordinateur, et ce système fonctionne pour les utilisateurs présentant une grande variété de diagnostics [4].

1. les systèmes de divertissement et de sonorisation, les lumières et les interrupteurs.

1. **Les spécifications techniques du Eyegaze Edge ®**

TABLE I.5 – Les spécifications techniques du Eyegaze Edge® [8].

Mode de suivi des yeux	Monoculaire /Binoculaire
Fréquence d'échantillonnage	50 Hz
Exactitude de la position du regard	$< 0.45°$
Mode de calibration	5 pt, 9 pt et 13 pt
Temps de latence du système	18 ms
Distance de fonctionnement	$40 - 73$ cm
Compatibilité avec des lunettes	Fonctionne avec la majorité des lunettes et des lentilles.
Résolution maximale de l'écran	2160×1440
Taille de l'écran	12.5"(format 3 : 2)

I.6.1.4 Les lunettes DIKABLIS 3

La figure I.16 montre un modèle portable des eye trackers sous forme des lunettes DIKABLIS 3 développées par l'entreprise allemande ERGONEERS GmbH qui même compte deux sites dans le monde en Allemagne et aux États-Unis, ce modèle offre la possibilité de l'étude du comportement humain dans des domaines d'application hautement dynamiques [6] en connectant l'eye tracker à la plateforme de mesure et d'analyse D-Lab pour enregistrer et analyser les données de plusieurs manières différentes.

FIGURE I.16 – Les lunettes DIKABLIS 3[6].

Ce modèle d'eye tracker (La figure I.16) est composé de deux unités : la première unité c'est des lunettes qui comportent les deux caméras pour filmer les yeux et la caméra pour filmer la scène, et la deuxième unité pour le traitement et le stockage des données.

1. **Les spécifications techniques des lunettes DIKABLIS 3**

TABLE I.6 – Les spécifications techniques du DIKABLIS 3 [7].

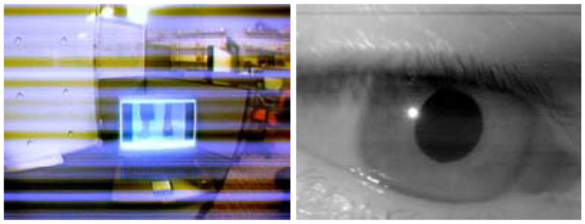
Mode de suivi des yeux	Binoculaire
Fréquence des caméras de suivi des yeux	60 Hz
Résolution des caméras pour le suivi des yeux	648 × 488 pixels
Résolution de la caméra de la scène	1920× 1080, 30 fps [2]
Compatibilité avec des lunettes	Fonctionne même pour les personnes portant des lunettes

I.6.2 Prototypes de recherche

I.6.2.1 openEyes

La figure I.17 montre le prototype du openEyes qui est un système de suivi des yeux mobile "eye tracker" de haute qualité, mais à faible coût, capable d'effectuer un suivi robuste en temps réel [30].

FIGURE I.17 – Le prototype du openEyes [30].



(c) (d)

FIGURE I.18 – (c) l'image d'une scène obtenue par l'eye tracker openEyes et (d) l'image de l'œil droit de l'utilisateur illuminé avec une lumière infrarouge [25].

L'évolution de ce système a passé par plusieurs prototypes de la première génération jusqu'au la quatrième génération (la figure I.17) [30], et ce qui concerne le développement logiciel du openEyes ils ont développé un algorithme qui s'appelle Starburst qui est un

2. Frames per second : Les images par seconde sont le nombre d'images que la caméra capture par seconde dans une vidéo.

algorithme de suivi oculaire hybride (open-source) qui combine entre des approches basées sur des caractéristiques et des modèles d'où l'objectif essentiel de cet algorithme est d'extraire les emplacements du centre de la pupille et de la réflexion de la cornée de manière à relier la différence vectorielle entre ces emplacements à des coordonnées dans l'image de la scène à travers la calibration [25] et l'algorithme "Starburst" commence par appliquer un filtre gaussien de 5× 5 pour réduire le bruit de fond et le bruit de ligne dans l'image de l'œil présenté dans (c) et (d) (La figure I.18) en raison de la construction à faible coût [25].

I.6.2.2 EyeSecret

La figure I.19 montre le prototype du EyeSecret qui est un eye tracker coûteux mais très performant avec l'auto calibration qui est l'avantage de cet eye tracker, et ce système utilise la réflexion cornéenne de la pupille pour obtenir le point de regard, ainsi que l'image de l'œil sous la lumière infrarouge [44].

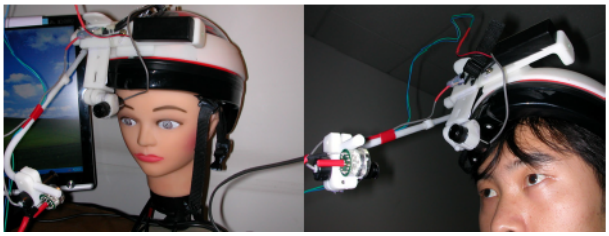


FIGURE I.19 – Le protoype du EyeSecret [44].

Le plan d'auto-calibration a été réalisé en utilisant un pointeur laser avec télécommande et une technologie de traitement d'images, l'utilisateur doit simplement pointer le faisceau laser vers le marqueur de calibration qu'il regarde et cette opération se répète pour les 9 points de calibration, et ce système a pu estimer de manière fiable la position des yeux avec le mouvement normal de la tête à l'intérieur ou à l'extérieur et atteindre une précision d'environ 1° d'angle visuel [44].

I.6.2.3 InvisibleEye

La figure I.20 montre le prototype du InvisibleEye c'est un eye tracker mobile à l'aide de plusieurs caméras millimétriques de taille 1×1 mm à basse résolution et à usage médicale de la marque Awaiba NanEye qui sont des capteurs d'images CMOS miniatures encadrés en rouge dans la figure I.20 [42].

FIGURE I.20 – Le prototype du invisbleEye [42].

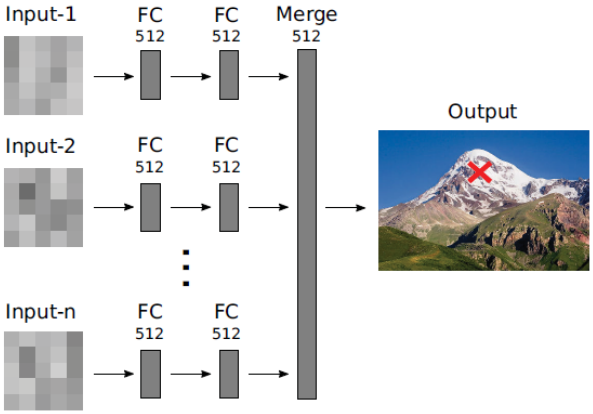


FIGURE I.21 – Vue d'ensemble du réseau de neurone utilisé dans InvisibleEye pour l'estimation du regard basée sur l'apprentissage [42].

Dans ce projet InvisibleEye c'est une nouvelle approche dans le domaine de l'eye tracking qui repose sur l'utilisation des caméras millimétriques qui peuvent être intégrées de manière presque invisible dans des lunettes normales et c'est l'avantage de ce système et pour compenser la faible résolution d'image des caméras, qui n'est que de quelques pixels ils ont utilisé un réseau de neurone qui prend en entrée plusieurs images oculaires à basse résolution, chaque image est codée à l'aide de deux couches entièrement connectées dans la figure I.21, les représentations spécifiques à l'image sont ensuite fusionnées pour prédire la direction du regard dans les coordonnées de la caméra de la scène (croix rouge) dans la figure I.21[42].

I.7 Les domaines d'application de eye tracker

Auparavant, les chercheurs et les experts de eye tracker étaient limités à des études scientifiques dans des conditions contrôlées. Les applications typiques incluent l'étude de la psychologie, de l'ophtalmologie, de la neurologie, des caractéristiques oculomotrices et anomalies. Récemment, il y a eu un développement concernant les applications de eye tracker qui contiennent la recherche médicale, recherche sur le processus de traduction, simulateur de véhicule, recherche embarquée, simulateur d'entrainement, réalité virtuelle, recherche sur les comportements adultes et infantiles, recherche sur les adolescents, recherche gériatrique, recherche sur les primates, entrainement sportif, eye tracking com-

merciale que ce soit dans le domaine d'utilisation web, publicité, marketing, automobile, aussi recherche pour trouver le bon indice, recherche sur la communication avec des systèmes et interaction homme machine pour des études en psychologie, en biométrie, en diagnostic et l'application sur des jeux. Les applications de eye gaze tracking peuvent être globalement classées en deux catégorie, diagnostique et interactives. Dans les applications de diagnostic les données de regard sont utilisées pour estimer les processus visuels et d'attention des utilisateurs. Les applications interactives traitent les données du regard une information d'entrée pour interagir avec les machines. Duchowski, divise les applications interactives en deux sélective et contingentes au regard. Dans le paradigme sélectif, le regard est utilisé comme un dispositif de pointage similaire à une souris d'ordinateur. Le paradigme du regard contingent décrit un système d'affichage qui dépend de la région exacte du regard et cette division a une grand importance par rapport à la relation de eye tracker avec les différents domaines.

I.7.1 La psychologie et le médicale

Le suivi oculaire en combinaison avec des méthodes de recherches conventionnelles ou d'autre capteurs biométriques peut aider à évaluer et potentiellement diagnostiquer des maladies neurologiques telles que le trouble déficitaire de l'attention avec hyperactivité (TDAH), le trouble spectre autistique (TSA), le trouble obsessionnel-compulsif(TOC), la maladie de parkinson et la maladie d'Alzheimer. De plus, la technologie de suivi oculaire peut être utilisée pour détecter les états de somnolence ou prendre en charge plusieurs autres domaines dont les médecins qui utilisent la technologie eye tracker arrivent à mieux comprendre les symptômes des patients avec des blessures aux yeux et les psychologues sont capable de mieux comprendre le processus d'une certaine action, comme par exemple le processus de lecture d'un article[26]. Le suivi oculaire peut être utilisé dans :

I.7.1.1 La psychologie cognitive et les sciences cognitives

Cette psychologie se présente dans les études de suivi oculaire de la perception des stimuli visuels, des relations entre la forme de l'information et la façon dont elle est perçue, le comportement des personnes au volant de véhicule et les interactions homme-machine.

I.7.1.2 La psychologie du développement

Cette psychologie se présente dans les études de suivi oculaire sur le développement des compétences d'attribution de l'attention, les relations entre le système de contrôle du mouvement et la compréhension de texte et les études de l'autisme.

I.7.1.3 La psychologie expérimentale

Cette psychologie se présente dans les études de suivi oculaire de la perception et de la reconnaissance de visage, la perception visuelle des scènes et des images dans l'utilisation de fonction visuelles et spatiales chez les personnes en bonne santé et les personnes atteintes de lésion du système nerveux.

I.7.1.4 La psycholinguistique et la lecture

Cette psychologie se présente dans les études de suivi oculaire sur la reconnaissance des difficultés de lecture, programmes de formation soutenant les compétences en lecture et la corrélation entre la perception visuelle et la maitrise de la lecture.

I.7.1.5 L'ophtalmologie

Cette psychologie se présente dans les études de suivi oculaire des propriétés des mouvements oculaires rapides, diagnostic des troubles des muscles déplaçant le globe oculaire et évaluation de l'effet du traitement chirurgical et conservateur, par exemple du strabisme sur les yeux et les mouvements du patient [19].

FIGURE I.22 – IHM Samsung avec une souris commandée par l'eye tracker pour les personnes handicapés moteurs [18].

I.7.2 Interface homme machine

Le début de l'utilisation de l'eye tracker pour les IHM était à la fin du 20 ème siècle avec un intérêt majeur pour faciliter la vie des personnes à mobilité réduite qui n'ont que les mouvements des yeux. Eye tracker leur permet de réaliser les gestes qui leurs sont impossible par les mains à l'aide d'une technique contrôlé seulement avec les yeux à l'aide d'un clavier sur l'écran, les utilisateurs peuvent taper des mots et des phrases en fixant les yeux sur les touches virtuelles qui représentent les lettres arrangées selon leur fréquence d'utilisation dans une langue spécifiée comme l'application de Kate et al (eye-switch). Un autre système qui s'appelle EagleEyes est un programme à 2 niveaux : les malades

sélectionnent d'abord un groupe de 5 ou 6 lettres (groupe ABCDE) après la sélection les lettres de ce groupe apparaissent sur l'écran de façon plus grandes et facile à choisir. Les applications de eye tracker ne sont pas limitées à la saisie d'une phrase mais aussi les enfants à mobilité réduite peuvent dessiner sur un écran en utilisant l'application Eyedraw , il est possible aussi de passer des commandes diverses en fixant leur regard sur l'écran à l'aide de l'ERICA (Eye-gaze Réponse Interface computer) et on trouve même des produits commerciaux comme par exemple, société Tobii qui propose des écrans augmentés d'un eye tracker, spécialement conçus pour les personnes qui n'ont pas la capacité de bouger leur corps afin de leur permettre d'interagir avec le monde extérieur par les yeux. Une autre application présente l'idée d'utiliser le point du regard comme une indication de l'attention parmi de multiples fenêtres d'applications et de zoomer ce que l'utilisateur est en train de regard, cela peut être 2 fois plus rapide que la souris ou le clavier. D'autre part une application utilise l'eye tracker dans le système GazeGalaxy qui sert à la recherche d'un fichier multimédia désiré dans une collection d'image ou de musique pour grandir localement l'affichage des fichiers afin que le malade puisse obtenir les contextes [26].

FIGURE I.23 – Un système de eye tracker proposé par Tobii [19].

I.7.3 Commercialisation

Le suivi oculaire est devenu un outil populaire et de plus en plus essentiel dans les études de marché. De nombreuses grandes marques utilisent activement le suivi oculaire pour évaluer l'attention des clients sur les messages clés et la publicité, ainsi pour évaluer la performance des produits, la conception des produits et des emballages. Lorsqu'il est appliqué aux tests en magasin, le suivi oculaire fournit des informations sur la facilités et la difficulté de la navigation en magasin, le comportement de recherche et les choix d'achat pour obtenir les réponses aux questions suivantes : «la personne a-t-elle regardé le produit ? combien de temps le regard va-t-il passé sur le logo de l'entreprise ? les gens remarquent quels produits ? et quelle chose fait d'être remarquée ? (la forme, la couleur ou le positionnement du produit)". Il concerne à la fois la navigation de niveau macro

et micro. La navigation de niveau macro fait référence à la disposition des catégories des produits dans toutes les zones d'un magasin ou d'une étagère, il est lié à la communication des catégories et des sections de vente. Les études de eye tracker visent à proposer une disposition appropriée des produits afin que les clients puissent se déplacer librement dans la boutique et en même temps avoir les produits d'une manière très facile. La navigation au micro niveau consiste à construire une disposition optimale de produits particuliers sur une étagère, cela vise à adapter l'espace de vente afin qu'il soit le plus efficace possible, il convient également de prendre en compte les besoins des clients, c'est-à-dire de placer le produit sur les étagères en fonction de leurs besoins. Le suivi oculaire est utilisé lors des études qualitatives et complète le processus d'étude des étagères. Dans cette partie d'études, divers agencement d'étagère, de matériaux publicitaires, d'étiquettes de prix sont testés [43].

FIGURE I.24 – Modèle de eye tracker Tobii pour la commercialisation [26].

I.7.4 Sites web

Les sites web utilisent le suivi oculaire comme méthodologie d'évaluation et comme un test d'expérience d'utilisateur. Le suivi oculaire est une approche souvent utilisée pour les tests des sites web qui donne des informations sur la façon dont les sites web sont consulté et expérimentés et pour répondre aux questions suivantes : comment les gens assistent-ils aux publicités, à la communication et aux appels à l'action ? Quelle est la trajectoire du mouvement du globe oculaire et le temps de mise au point sur des objets particuliers ? dans quelle séquence les objets attirent l'attention du client ?Quels sont les éléments distrayants et les éléments fréquemment consultés ? est ce que que l'affichage est correct ou incorrect des modules sur le site web ? si vous perdez des revenus, les données de suivi oculaire peuvent fournir des informations précieuses sur les schémas de regard des visiteurs de votre site web - combien de temps leur faut-il pour trouver un produit spécifique sur votre site ? quel type d'information visuelle ignorent-ils ? le suivi oculaire offre la possibilité d'analyser l'interaction de l'utilisateur entre les clics, cela fournit des informations précieuses sur les fonctionnalités les plus accrocheuses, celles qui créent de la confusion et celles qui sont complètement ignorées. Plus précisément, le suivi oculaire peut être utilisé pour évaluer l'efficacité de la recherche, la marque, les publicités en ligne, la convivialité de la navigation, la conception globale et de nombreux autres composants du

site web. Ces analyses peuvent cibler un prototype ou un site concurrent et efficace [38].

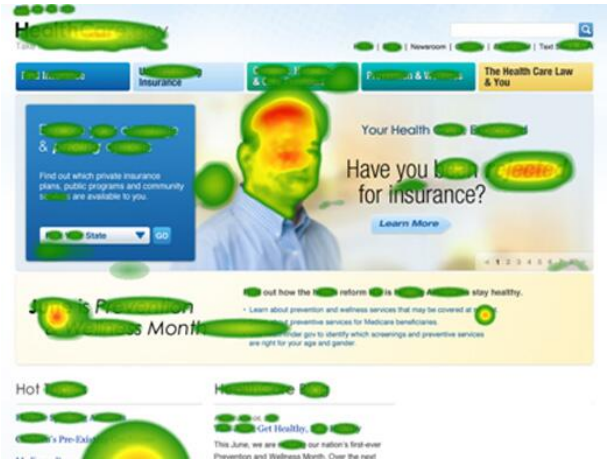


FIGURE I.25 – Application de eye tracker dans un sitweb [21].

I.7.5 Jeux vidéo et réalité virtuelle

Le suivi oculaire a récemment été introduit dans l'industrie du jeu puisqu'il est devenu un outil important car les concepteurs sont désormais en mesure d'évaluer et de quantifier des mesures telles que l'attention visuelle et les réactions aux moments clés pour améliorer l'expérience de jeu globale. Lorsqu'elles sont combinées avec d'autre capteurs biométriques les concepteurs peuvent utiliser le données pour mesurer les réponses émotionnelles et cognitives au jeu. De nouvelles tendances et développements pourraient bientôt permettre de contrôler le jeu en fonction de la dilatation des pupilles et des mouvements oculaires, et l'une des applications de eye tracker dans les jeux c'est le mouvement de l'environnement de jeu avec les mouvements des yeux, lorsque le gamer tourne ces yeux droit l'environnement tourne aussi et suit les mouvement de la tête [19]. On peut aussi avec cette technique utiliser la réalité virtuelle parce que les humains ont une vision fovéale ou une information visuelle maximale est obtenue à partir de la région focalisée sur la fovéa. Les casques de réalité virtuelle peuvent utiliser ces informations pour effectuer un rendu de haute résolution aux endroits ou l'utilisateur regarde .Cela améliore la perception de la scène visuelle mais a une charge de calcul qui réduit le rendu de l'image. Les mouvements oculaires peuvent également être utilisés comme modalité d'interaction soit comme dispositif de pointage, soit pour la saisie basée sur les yeux [28].

FIGURE I.26 – Application de eye tracker dans les jeux [19].

I.7.6 Automobile

La grande majorité des systèmes de suivi oculaire utilisé dans les automobiles ont a voir avec la détection de la fatigue du conducteur. Ces produits sont des unités montables qui peuvent être placées sur n'importe quel véhicule et se composent d'un système comprenant une caméra et des capteurs infrarouges (IR) et une unité de commande, l'unité de contrôle se compose du microcontrôleur (il est responsable de la prise de décision) et du système de son d'alerte. Un système de suivi oculaire pleinement fonctionnel ne pourrait pas fonctionner de la même manière, il a besoin d'un ordinateur analysant les données tant qu'il est opérationnel, il y a une communication entre l'eye-tracker et le véhicule concernant la connexion directe aux capteurs et aux systèmes internes de l'automobile. Cependant , il existe plusieurs similitudes sur la façon dont la caméra doit être installée dans une automobile comme vous voyez sur la figure I.27. Les chercheurs ont fait valoir que de nombreuses considérations doivent être prises en compte avant d'installer des technologies embarquées. De plus, le système ne doit pas surprendre ni confondre le conducteur, ce qui entraine des effets négatifs, Ces facteurs ont un effet direct sur la conception générale, l'installation de système et son utilisation. Nombreux systèmes adoptent uniquement une tonalité ou un signal d'avertissement, qui peut être auditif ou visuel, les signaux auditifs peuvent être relayés via les systèmes de haut-parleurs de la voiture avec des signaux visuels affichés comme un voyant clignotant sur le tableau de bord. Un système plus efficace peut être obtenu avec des technologie qui incluent l'interface des unités de commande électronique (ECU) embarqué qui permettent le freinage automatique et la transmission de message à une unités centralisée, qui peut contacter le conducteur et s'assurer que les mesures correctes ont été prise [20].

FIGURE I.27 – La position de l'eye tracker dans un véhicule [36].

I.8 Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur le domaine des eye trackers en commençant par une définition de l'eye tracker puis l'évolution des eye trackers dans l'histoire ensuite on a décrit l'eye tracking et l'anatomie de l'oeil humain puisque c'est le membre ciblé dans ce projet, ainsi que la structure matérielle des eye trackers de chaque type. Avant d'entamer la partie des modèles nous avons expliqué les notions de base dans le monde de l'eye tracking puis les modèles des eye trackers commerciaux avec leurs spécifications techniques et les prototypes de recherche avec les méthodes utilisées dans ces travaux. Enfin nous avons parlé des différentes applications de l'eye trackers dans plusieurs domaines tels que le domaine biomédical, l'automobile, sit web, jeux vidéo et réalité virtuelle et commercialisation.

Chapitre II

Méthodologie du Eye - Tracking

II.1 Introduction

Dans ce chapitre, nous allons présenter la méthodologie des différents traitements d'images que nous avons effectué dans notre travail, en commençant par les pré-traitements d'images par l'implémentation d'un filtre bilatéral, et l'opération de l'érosion pour éliminer certains bruits et la réflexion de lumière de l'écran à commander dans l'œil. Ensuite, nous allons procéder à une détection de visage de l'utilisateur et la détermination de certains points sur celui-ci à l'aide de la bibliothèque Dlib, dont nous avons besoin pour isoler les images qui contiennent uniquement l'œil de l'utilisateur par l'application d'un masque. Par la suite, nous allons effectuer une détection de l'iris qui contient plusieurs étapes pour aboutir à une estimation de la position des deux pupilles. Ceci est grâce à la détermination du centre de l'iris par le calcul des moments de l'image binaire de celle-ci. Finalement, nous allons procéder à une calibration pour trouver la relation entre le déplacement dans l'image de l'oeil et le déplacement dans l'écran, ce qui permettra de déplacer le curseur sur l'écran.

II.2 Vision par ordinateur avec OpenCV

La vision par ordinateur est un domaine en pleine croissance, il est consacré à l'analyse, à la modification et à la compréhension de haut niveau des images. Son objectif est de déterminer ce qui se passe devant une caméra et d'utiliser cette compréhension pour contrôler un ordinateur ou un système robotique, ou pour fournir aux gens de nouvelles images plus informatives ou plus esthétiques que les images originales. Les domaines d'applications de la technologie de la vision par ordinateur comprennent la surveillance vidéo, la biométrie, l'automobile, la photographie, la production des films, la recherche sur le web, la médecine, les jeux en réalité augmentée, les nouvelles interfaces et bien d'autres applications. Par exemple, les appareils photo modernes peuvent faire la mise au point

automatiquement sur le visage des gens et déclencher l'appareil lorsque les personnes sourient, les systèmes de reconnaissance optique de texte aident à transformer les documents numérisés en texte qui peut être analysé ou lu à haute voix par un synthétiseur vocal, les voitures peuvent inclure des systèmes d'assistance à la conduite automatisée qui aident les utilisateurs à se garer ou les avertissent des situations potentiellement dangereuses et même la détection de la fatigue du conducteur par le déclenchement d'une alarme, il y a aussi la correction de la direction de la vue du conducteur qui intervient lorsqu'il ne voit pas en face pendant un certain moment. Parmi les applications, nous avons aussi la vidéo surveillance intelligente qui joue un rôle de plus en plus important dans la surveillance de la sécurité des espaces publics, étant donné que les appareils mobiles tels que les téléphones portables et les tablettes sont équipés d'appareils photo qui sont devenus suffisamment intelligents pour fusionner plusieurs photos dans un panorama haute résolution. On peut également s'en servir pour lire un code QR, le reconnaitre et récupérer des informations sur un produit sur internet .

La vision par ordinateur à une relation avec des yeux humaines, par un traitement d'image on peut utiliser l'empreinte de l'œil comme un système de sécurité dans les différents appareils et on peut suivre le mouvement oculaire afin de trouver pas mal d'information concernant le point de regard de l'utilisateur sur un écran pour définir ce qui l'attire dans un site ou dans un magasin et même dans le coté médical, le suivi oculaire détermine l'état des yeux, si la vision de l'utilisateur est correcte ou pas. La vision par ordinateur est cependant coûteuse en calcul. Même un algorithme dédié à la résolution d'un problème très spécifique, tel que l'assemblage de panorama ou la détection de visage et de sourire, nécessite beaucoup de puissance. De nombreux scénarios de vision par ordinateur doivent être exécutés en temps réel, ce qui implique que le traitement d'une seule image doit être terminé en 30 à 40 millisecondes. OpenCV est une bibliothèque de vision par ordinateur open source, elle est écrite en C et C++ et fonctionne sous Linux, Windows et Mac OS. OpenCV a été conçue pour une efficacité de calcul et avec un fort accent sur les applications en temps réel, OpenCV utilise automatiquement la IPP (Integrated Performance Primatives) d'Intel, appropriée au moment de l'exécution (Runtime) si cette bibliothèque est installée. L'un des objectifs d'OpenCV est de fournir une infrastructure de vision par ordinateur simple à utiliser qui aide les gens à créer rapidement des applications de vision assez sophistiquées. La bibliothèque OpenCV contient plus de 500 fonctions couvrant de nombreux domaines de la vision, notamment l'inspection des produits en usine, l'imagerie médicale, la sécurité, l'interface utilisateur, l'étalonnage de la caméra, la vision stéréo et la robotique. OpenCV contient une bibliothèque d'apprentissage automatique complète et polyvalente. Cette sous-bibliothèque se concentre sur la reconnaissance statistique des formes et le regroupement, elle est très utile pour les tâches de vision qui sont au cœur de la mission d'OpenCV, mais elle est assez générale pour être utilisée pour tout problème d'apprentissage automatique [17].

II.2.1 Traitement d'images

Une image se compose d'un tableau bidimensionnel de nombres. La couleur ou la nuance de gris affichée pour un élément d'image donné (pixel) dépend du nombre stocké dans le tableau pour ce pixel. Le type de données d'image le plus simple est le noir et blanc. Il s'agit d'une image binaire puisque chaque pixel est égal à 0 ou 1. Le type d'image suivant dans la complexité est l'échelle de gris, ou chaque pixel prend une valeur comprise entre zéro et le nombre d'échelle de gris ou de niveaux de gris que le scanner peut enregistrer. Ces images ressemblent à des photographies en noir et blanc courant, elle sont en noir et blanc et en nuances de gris. La plupart des images en échelle de gris ont aujourd'hui 256 nuance de gris. Les gens peuvent en distinguer environ 40 nuances de gris, donc une image à 256 nuances "ressemble à une photographie". Le type d'image le plus complexe est de type couleur. Les images couleurs sont similaire à l'échelle de gris sauf qu'il existe trois bandes, ou canaux correspondant aux couleur rouge, verte et bleue. Ainsi, chaque pixel a trois valeurs qui lui sont associées. Un scanner couleur utilise des filtres rouge, vert et bleu pour produire ces valeurs [35].

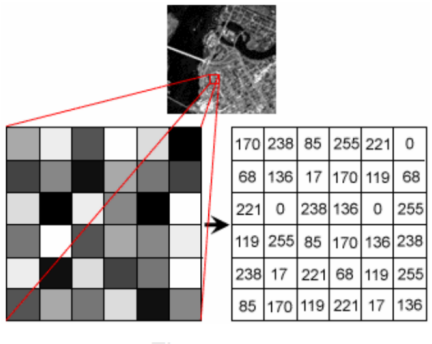


FIGURE II.1 – Représentation des pixels dans une image [29]

Le traitement d'images est une technique pour améliorer les images brutes reçues par des caméras comme capteur placé sur les satellites, des sondes spatiales et des avions ou des images prises dans la vie quotidienne normale pour diverses applications. Diverse techniques ont été développées dans le traitement d'images au cours des quatre à cinq dernières décennies. La plupart des techniques sont développées pour améliorer les images obtenues à partir d'engins spatiaux sans pilote, des sondes spatiales et des vols de reconnaissance militaire. Les systèmes de traitement d'image sont de plus en plus populaires en raison de la disponibilité facile d'ordinateurs personnels puissants, de dispositifs de mémoire de grande taille, de logiciels graphique, etc. Le traitement d'image est utilisé dans diverses applications telles que :

- La télédétection,
- L'image médicale,

- L'évaluation non destructive,
- Les études médico-légales,
- Les textiles,
- La science des matériaux,
- Militaire,
- Industrie cinématographique,
- Traitement de document,
- Arts graphique,
- Industrie de l'imprimerie,
- etc.

Les étapes courantes du traitement d'image sont la numérisation, le stockage, l'amélioration et l'interprétation des images. Il existe deux méthodes disponibles dans le traitement d'image, décrites dans les deux sections suivantes.

II.2.2 Traitement d'image analogique

Le traitement d'image analogique fait référence à l'altération de l'image par des moyens électrique. L'exemple le plus courant est l'image de télévision. Le signal de télévision est un niveau de tension dont l'amplitude varie pour représenter la luminosité à travers l'image. En faisant varier électriquement le signal, l'apparence de l'image afficher est modifiée. Les commandes de luminosité et de contraste sur un téléviseur servent à ajuster l'amplitude et la référence du signal vidéo, ce qui entraine un éclaircissement, un assombrissement et une modification de la plage de luminosité de l'image afficher [24].

II.2.3 Traitement d'image numérique

Dans ce cas, des ordinateurs numériques sont utilisée pour traiter l'image. L'image sera convertie en forme numérique à l'aide d'un scanner numériseur, puis il la traitera. Il se définit comme la soumission de représentation numérique d'objet à une série d'opérations afin d'obtenir un résultat souhaité. Il commence par une image et produit une version modifiée de celle-ci. C'est donc un processus qui prend une image dans une autre. Le terme traitement d'image numérique se réfère généralement au traitement d'une image bidimensionnelle par un ordinateur numérique. Dans un contexte plus large cela implique le traitement de toute donnée bidimensionnelle. Une image numérique est un tableau de nombres réels représentés par un nombre fini de bit. Le principal avantage des méthodes de traitement numérique des images est leur polyvalence, leur répétabilité et la préservation de la précision des données d'origine. Les différentes techniques de traitement d'image sont [24] :

- Représentation d'image,

- Pré-traitement d'image,
- Amélioration d'image,
- Restauration d'image,
- Analyse d'image,
- Reconstruction d'image,
- Compression de données d'image.

II.3 Traitement d'images

II.3.1 Filtre bilatéral

Tout d'abord pour arriver au but principal, qui est l'estimation de la position des deux pupilles des deux yeux par rapport à la caméra, nous commençons notre traitement d'image par l'implémentation d'un **Filtre Bilatéral** qui est un filtre de lissage non linéaire préservant les bords et réduisant le bruit dans les images. Il remplace l'intensité de chaque pixel par une moyenne pondérée des valeurs d'intensité des pixels voisins d'une manière très similaire à la convolution gaussienne. La différence est que le filtre bilatéral prend en compte la différence de valeur avec les pixels voisins pour préserver les bords tout en lissant [33], d'où l'intérêt d'utiliser ce filtre.

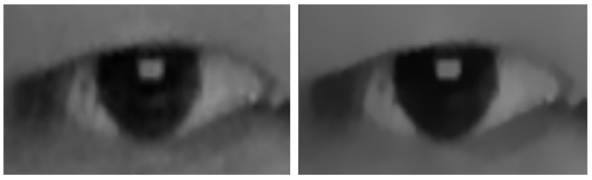


FIGURE II.2 – (a) Image en entrée du filtre bilatéral et (b) Image en sortie du filtre bilatéral.

La figure II.2 nous montre la conversion d'une image avant le filtrage (a) en une version lissée (b) par le filtre bilatéral, on remarque que ce dernier supprime la plupart des textures, du bruit et des détails fins, mais préserve les grands bords nets sans les rendre flous.

II.3.2 Érosion

L'érosion est l'une des deux opérations fondamentales (avec la dilatation) du traitement morphologique des images, sur laquelle toutes les autres opérations morphologiques sont basées, et elle est utilisé pour supprimer les petits bruits blancs et réduire les objets de premier plan dans l'image [39]. On peut définir l'érosion en niveau de gris en termes de minimum local sur la zone du noyau (l'élément structurant) donné par :

Soit $I : A \subset \mathbb{Z}^n \to B$($I$ est donc une image n-dimensionelle de domaine A et dont les valeurs d'arrivée se situent dans B), et $E \subset \mathbb{Z}^n$ un élément structurant.

On définit l'érosion de I par E l'application notée $I \ominus E$ telle que,

$$\forall x \in A,\ (I \ominus E)(x) = min_{y \in E_x} I(y)$$

Nous avons l'exemple suivant et on veut calculez $I \ominus E$ pour bien expliquer cette opération :

On note :

- I : Une image en niveau de gris.
- E : L'élément structurant (le noyau).

Avec :

$$I = \begin{bmatrix} 82 & 76 & 81 & 81 & 81 & 75 \\ 80 & 77 & 80 & 84 & 79 & 78 \\ 82 & 82 & 84 & 80 & 83 & 84 \\ 91 & 92 & 94 & 96 & 93 & 96 \\ 102 & 100 & 103 & 103 & 110 & 112 \\ 100 & 110 & 109 & 112 & 110 & 115 \end{bmatrix} \qquad E = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

En supposant que l'origine E est au centre(encadré en rouge dans la figure II.3) pour chaque pixel de I on superpose l'origine de E et ceci est fait pour tous les pixels de l'image I par exp.$(I \ominus E)(1,1) = min\{82, 76, 80\} = 76$ et $(I \ominus E)(5,5) = min\{96, 93, 110, 112, 110\} = 93$ et le résultat est dans la figure II.3. De manière similaire, le reste du calcul est effectué pour chaque pixel de l'image I.

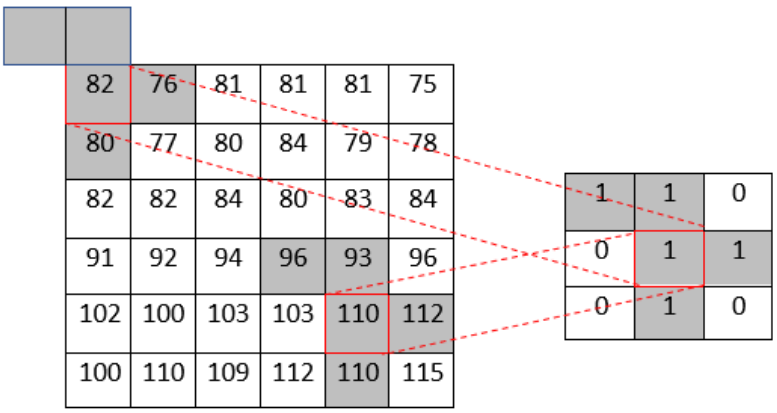


FIGURE II.3 – Illustration de la méthode d'érosion de $I \ominus E$ dans les pixels $I_{1,1}$ et $I_{5,5}$.

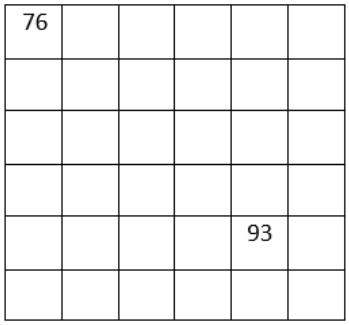


FIGURE II.4 – Le résultat de la figure II.3.

Donc le résultat final après l'érosion avec une seule itération est une matrice de la même dimension de I (6×6) :

$$I \ominus E = \begin{bmatrix} 76 & 76 & 80 & 81 & 75 & 75 \\ 77 & 76 & 76 & 79 & 78 & 75 \\ 80 & 77 & 77 & 80 & 79 & 78 \\ 82 & 82 & 82 & 80 & 80 & 83 \\ 91 & 91 & 92 & 94 & 93 & 93 \\ 100 & 100 & 100 & 103 & 103 & 110 \end{bmatrix}$$

L'implémentation de cette méthode sur une image de l'œil après filtrage (b) dans la figure II.2 avec trois itérations et avec un élément structurant égal à une matrice remplie de 1 et de dimension(3×3), et le résultat est présenté dans la figure II.5.

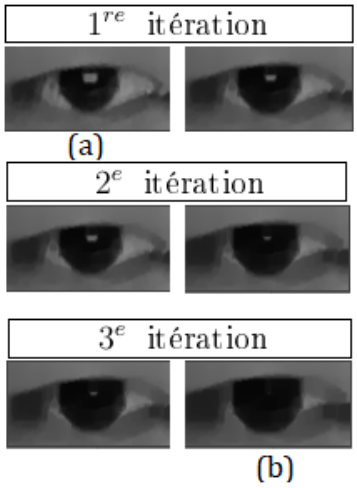


FIGURE II.5 – On remarque que l'opération de l'érosion a éliminé les petits bruits blancs présent dans l'image (a) sous forme d'un petit rectangle blanc due à la réflexion de la lumière du laptop dans l'œil, après trois itérations le rectangle a complètement disparu (b).

II.3.3 Image en niveau de gris

L'image en niveau de gris ou bien a mise à l'échelle de gris «GrayScale» c'est le processus de conversion d'une image à partir d'autres espaces de couleurs, par exemple RGB, CMYK, HSV... , en nuances de gris ((b) dans la figure II.6) qui varie entre le noir complet et le blanc complet. Le but de cette conversion est la réduction de la dimension pour gagner en espace mémoire et en temps de calcul car dans les images RGB il y a trois canaux de couleur, alors que les images à échelle en gris sont à canal unique. La deuxième raison c'est pour que tous les algorithmes fonctionnent car il y a de nombreux algorithmes qui sont personnalisés pour fonctionner que sur des images en niveaux de gris comme image d'entrée [13].

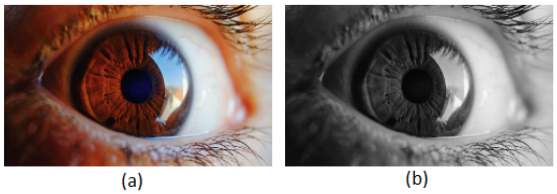


FIGURE II.6 – (a) Image originale (b) Image en niveau de gris.

II.3.4 Le seuillage

Le seuillage ou bien «thresholding» est une technique fondamentale dans le traitement d'image et c'est la méthode la plus simple pour la segmentation d'une image, et cette méthode consiste à transformer une image en niveau de gris en une image binaire [32] dont les valeurs des pixels sont généralement sur un octet 8 bits (256 valeurs) par convention soit 0 qui représente le noir (intensité lumineuse nulle) ou 1 qui représente le blanc (intensité lumineuse maximale 255). Dans le cas du seuillage binaire, ces deux niveaux sont attribués par la comparaison de la valeur d'intensité du pixel $I(x, y)$ dans la figure II.7 avec la valeur du seuil qui peut varie suivant l'application, il existe des types de seuillage simple tels que le seuillage binaire ou binaire inversé, tronqué, etc. où pour chaque pixel la même valeur de seuil est appliquée et il existe d'autres méthodes comme le seuillage adaptatif, Otsu,...etc [2].

Pour bien expliquer le seuillage on illustre cette opération graphiquement et aussi l'implémentation du seuillage binaire par OpenCV sur une image de l'œil (Fig. II.9) :

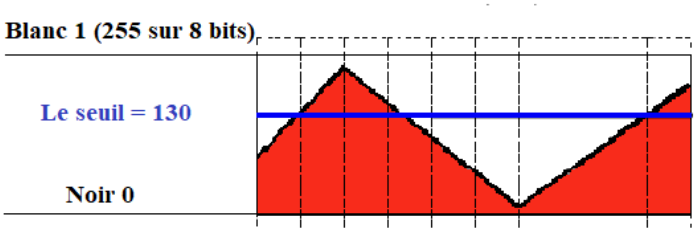


FIGURE II.7 – Le graphe qui illustre le seuil en bleu (échelon unité) avec les valeurs d'intensité des pixels de l'image source triangle en rouge [2].

L'opération de seuillage binaire peut être exprimée comme suit :

$$I'(x,y) = \begin{cases} 255 & Si\ I(x,y) \geq seuil\ (130) \\ 0 & Sinon \end{cases}$$

Avec :

- $I(x,y)$ L'intensité du pixel de l'image avant seuillage.
- $I'(x,y)$ l'intensité du pixel de l'image aprés seuillage.

Si l'intensité du pixel $I(x,y)$ est supérieure à la valeur de seuil qui est égale à 130 dans l'exemple (La figure II.7), l'intensité du nouveau pixel $I'(x,y)$ est fixée à une valeur maximale(255 dans le cas de 8 bits), sinon les pixels sont mis à 0.

Dans la représentation graphique de la figure II.7 après l'application de la fonction du seuillage binaire, le résultat est dans la figure II.8 on remarque que l'image est binaire (noire et blanc) après l'application du seuillage .

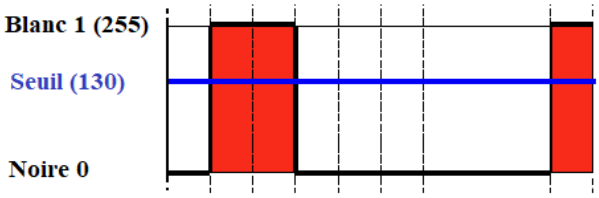


FIGURE II.8 – Le graphe de la figure (II.7) aprés l'application du seuillage binaire [2].

On appliquant la fonction de seuillage binaire sur une image de l'œil comme entrée après la conversion en niveau de gris et avec un $seuil = 15$ et on trouve le résultat finale suivant (Fig.II.9) :

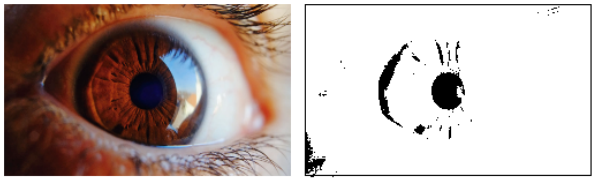


FIGURE II.9 – (a) image originale et (b) image originale après seuillage binaire.

II.3.5 La détection de visage

La détection de visage est un problème fondamentale et important de la vision par ordinateur, qui a été largement étudié au cours des dernières décennies. La détection de visage est l'une des étapes clés importantes vers des nombreuses applications ultérieures liées au visage, telles que la vérification et la reconnaissance des personnes. Du cadre de détection d'objet, de nombreuses méthodes ont été proposées pour la détection de visage au cours de la dernière décennies. Les premières études de recherches dans la littérature étaient principalement axées sur l'extraction des différents types de fonctionnalités artisanales, avec des experts du domaine en vision par ordinateur, et la formation de classificateurs efficaces pour la détection et la reconnaissance de visage avec les algorithmes d'apprentissage automatiques traditionnels. Ces approches sont limitées en ce sens qu'elles nécessitent souvent des experts en vision par ordinateur pour créer des fonctionnalités efficaces et que chaque composant individuel est optimisé séparément. La bibliothèque dlib c++ également très utilisée dans l'industrie est livrée avec une licence logicielle lui donnant la permission pour qu'elle soit utilisable gratuitement dans les applications commerciales. La détection de visage avec la bibliothèque dlib est très efficace et la plupart des détecteurs de repère ont besoin d'un cadre de délimitation autour du visage pour initialiser l'algorithme ou pour cadrer l'image. Ainsi, avant d'appliquer un détecteur de repère dans la pratique, un détecteur de visage est nécessaire pour localiser les visages dans l'image. Lors de la mesure des performances des détecteurs de point de repère, une boite faciale, calculée sur la base des points de repère est utilisée comme initialisation. cette boite faciale qui est sous forme d'un rectangle qui suit le déplacement du visage en temps réel avec une bonne précision, en plus de ça la détection de visage est une préparation et une étape nécessaire pour la détection des yeux et donne la possibilité de détecter les 68 points de repère sur le visage et surtout les points des yeux. Après l'utilisation de «Shape predictor 68 face Landmark » pour détecter les points sur le visage, le nez, les yeux et la bouche comme il est présenté dans la figure II.10, la détection de visage est une première étape qui aide à l'isolation des yeux et donne la possibilité de trouver leur position afin que l'iris soit détecté [40].

FIGURE II.10 – Image qui présente la détection de visage avec l'utilisation de la bibliothèque dlib "Shape predictor 68 face Landmark"[34].

II.3.6 Isolation des yeux

II.3.6.1 Détection des yeux

Après la détection de visage qui facilite et prépare la deuxième étape qui est la détection des yeux en utilisant la bibliothèque dlib c++, qui sert à positionner des points sur chaque œil, 6 points pour chacune, ces points suivent le déplacement et le mouvement des yeux en temps réel et son erreur avec une bonne précision. Les coordonnées (x, y) de ces repères sont connues et stockées pour d'autre utilisations comme l'isolation de l'œil et aussi pour tracer une ligne verticale et horizontale, et si cette ligne verticale est inférieure à la valeur naturelle, on conclut qu'il y a un clignement fait par l'utilisateur. Nous avons utilisé ce geste comme un click dans les différentes applications. Nous avons également la possibilité de changer la couleur de ces repères et modifier ces positions pour les différents besoins. Pour un résultat plus précis il faut que la résolution de la caméra soit acceptable ou si vous utilisez une vidéo, il faut qu'elle soit de bonne qualité. Il faut aussi utiliser ce traitement sous un éclairage net et bon et comme un dernier point il faut que la distance entre le visage de l'utilisateur et la caméra soit petite ou moyenne parce que le traitement de visage et le positionnement des repères sur les yeux est faible et non précis si la distance entre l'utilisateur et la caméra est grande.

II.3.6.2 Région des yeux

Après avoir positionné ces repères sur les yeux et avoir connu leurs coordonnées, nous allons utiliser une commande prédéfinie sur la bibliothèque OpenCV qui aide a dessiner une ligne entre deux points selon leurs coordonnées. Avec l'utilisation de cette commande, nous allons relier ces points avec une ligne entre chacune d'elles pour avoir un contour sur chaque œil, une région est ainsi dessinée autour de l'œil. Cette région est collée sur les yeux et elle les suit dans leur mouvements et déplacements avec une bonne précision. On peut modifier l'épaisseur de ces lignes de région et même changer leurs couleurs. Cette région est une étape essentielle dans l'isolation des yeux, la figure II.11 présente seulement ces région en bleu.

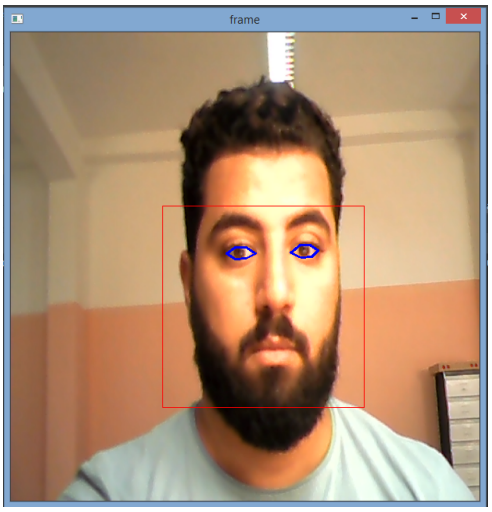


FIGURE II.11 – Image qui présente les régions des yeux en bleu.

II.3.6.3 Application de masque

Après avoir mis une région sur les yeux, nous allons utiliser une commande prédéfinie dans la bibliothèque OpenCV pour appliquer un masque sur la fenêtre de la vidéo, ce masque sert à éliminer ce qui se trouve à l'extérieur de la région et couvre sa couleur par la couleur noire (les autres régions de visage) et laisse ce qui se trouve à l'intérieur des régions des deux yeux tel qu'il est, ce qui donne une image qui présente juste les yeux et le reste de visage en noir comme il est présenté dans la figure II.12.

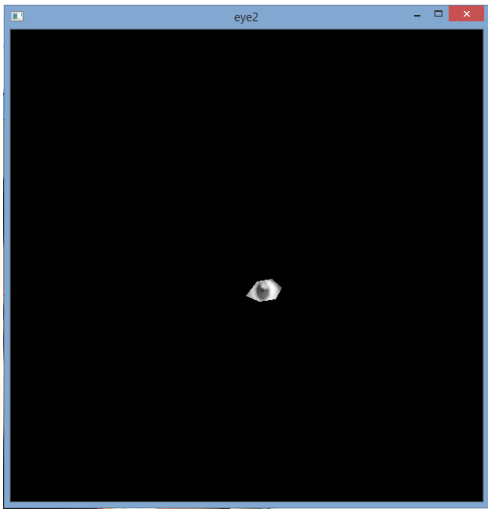


FIGURE II.12 – Image qui présente l'utilisation de masque sur le visage

II.3.6.4 Isolation

Finalement et comme une dernière étape nous revenons en arrière pour utiliser les coordonnées des repères et des points positionnés sur les yeux, et nous prenons le minimum de x et de y comme un origine de la nouvelle frame et le maximum de x et de y comme un

point final afin de construire une petit nouvelle frame qui représente juste l'œil et l'iris. Cette dernière étape permet l'isolation des yeux tel que présenté dans la figure II.13.

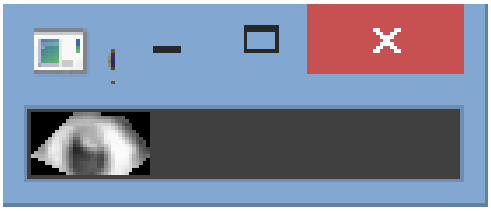


FIGURE II.13 – Image qui présente juste l'oeil isoler dans une image

II.3.7 Détection de l'iris

Pour arriver au but principal de ce traitement d'images qui est l'estimation de la position des deux pupilles par rapport à la caméra (i.e trouver les cordonnés (x, y) de chaque pupille, gauche et droite, dans le cadre de la caméra), il faut tout d'abord passer par la détection de l'iris, ensuite trouver le centre de chaque iris qui représente lui même le centre de la pupille.

II.3.7.1 Détection de contour

En appliquant les précédents traitements à l'image en niveau de gris de l'œil (la figure II.14) i.e. après le filtrage (la figure II.15) et l'érosion (la figure II.16), en faisant ensuite une transformation à une image binaire qui ne contient que l'iris en noir et le reste des pixels en blanc (la figure II.17) et grâce au seuil optimal calculé par la méthode du seuillage automatique, il reste maintenant la détection de contour de l'iris.

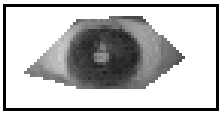


FIGURE II.14 – Image de l'oeil en niveau de gris avant filtrage.

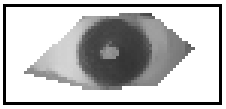


FIGURE II.15 – Image de l'oeil (la figure II.14) aprés filtrage.

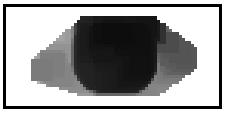


FIGURE II.16 – Image de l'oeil (la figure II.15) aprés l'érosion.

L'image binaire (la figure II.17) est utilisé comme une image d'entrée pour la détection de contour dans la fonction **findContours** de OpenCV et à l'aide de la fonction **drawContours** nous traçons ce contour calculé qui est un tableau de Numpy [1] qui contient les coordonnées (x, y) des points limites de l'objet (l'iris). Sur l'image en niveau de gris (la figure II.17) avant le traitement juste pour visualisation.

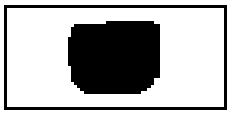


FIGURE II.17 – Image de l'oeil (la figure II.15) aprés seuillage.

FIGURE II.18 – Le contour de l'iris.

II.3.7.2 Le seuillage automatique

Cette méthode consiste à calibrer l'algorithme de détection de la pupille en trouvant la valeur optimale de seuil de binarisation pour le suivi "Tracking" des centres des deux pupilles, et elle contient les étapes suivantes :

II.3.7.2.1 Estimation de la taille de l'iris Cette méthode consiste à estimer la taille de l'iris par le calcul du pourcentage d'espace que l'iris occupe sur la surface de l'image "frame" binaire de l'œil, en commençant par extraire la taille de cette image (x, y) qui est calculée comme suit :

- le nombre des pixels total $= x \times y$
- le nombre des pixels noirs = le nombre des pixels total − les pixels non nulles [2]
- la taille de l'iris=le nombre des pixels noires / le nombre des pixels total

Ensuite cette estimation de la taille de l'iris est utilisée pour trouver le seuil optimal en posant une valeur moyenne de la taille de l'iris égale à 0.48 et en variant le seuillage de 5 à 150 avec un pas de 5 est à chaque valeur de seuil nous faisons une estimation de la taille de l'iris par le calcul du pourcentage expliqué précédemment. Nous stockons tout ça dans un dictionnaire python, le but est de trouver dans le dictionnaire python la valeur de seuil qui correspond à la valeur de la taille de l'iris la plus proche à la valeur moyenne de la taille de l'iris précédemment déclarée.

1. librairie de calcul matriciel de Python
2. Le nombre des pixels non nuls (non noirs) est calculé par la fonction de Open CV **countNonZero**

II.4 Estimation de la position de la pupille

II.4.1 Les coordonnés des deux pupilles

Après avoir calculé le contour (la figure II.18) on trouve la surface de ce contour qui est une caractéristique du contour à l'aide de la fonction **contourArea** qui prend comme argument d'entrée le contour déjà calculé et ce contour est utilisé pour faire une estimation du centre de l'iris par le calcul des moments II.1 de l'image binaire grâce à la fonction **moments** de Open CV et les coordonnés du centre de l'iris sont définis par(x_c *et* y_c) tels que :

$$M_{i,j} = \sum_x \sum_y x^i y^j I(x,y) \tag{II.1}$$

$$x_c = \frac{M_{10}}{M_{00}}$$
$$y_c = \frac{M_{01}}{M_{00}}$$

Les coordonnés des deux pupilles(centre de l'iris) sont calculées par :

$$\{x_c, y_c\} = int\{\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\}$$

Avec :

- M_{ij} : les moments de l'image,
- $I(x,y)$: le pixel avec ces coordonnés(x,y) dans l'image I,
- M_{00} : la surface de l'objet(iris).

II.4.2 Marquage de la position des deux pupilles

Après avoir trouvé les coordonnées des deux pupilles à travers une estimation par le calcul des moments de l'image binaire qui ne contient que l'iris, nous marquons la position des deux pupilles par deux lignes en vert à chaque pupille (en formant deux lignes vertes perpendiculaire entre eux (la figure II.19)) avec la fonction de OpenCV **line** qui prend comme argument d'entrée deux points : le point de départ et le point d'arrivée.

FIGURE II.19 – Marquage de la position des deux pupilles en vert.

II.5 Estimation de la position (x,y) des yeux sur l'écran

II.5.1 Calibration

Après avoir isolé l'œil et détecté l'iris et trouvé leur position sur une petite fenêtre qui représente la lecture de la vidéo, nous voulons maintenant trouver la position (x, y) des yeux sur un écran d'un ordinateur. Il est impossible de trouver ces coordonnées sans passer par la calibration des yeux sur l'écran. La calibration c'est un ensemble des points qui sont positionnés sur l'écran, le premier est positionné sur le pixel le plus haut, le deuxième est positionné sur la droite, le troisième est positionné sur le pixel le plus bas et le dernier est positionné sur la gauche. Ces points servent à localiser la direction de la vision de l'utilisateur sur les différents cotés de l'écran afin de stocker les x et les y de la pupille, d'après ces x et y on peut calculer la distance en pixel parcourue par la pupille après le changement de vision de droite vers la gauche et du haut vers le bas. Cette distance est calculée par rapport à la résolution de la petite fenêtre qui présente la lecture de vidéo qui est une étape essentielle pour trouver une relation qui calcule les nouvelles coordonnées de x, y sur l'écran à partir de x, y de la petite fenêtre.

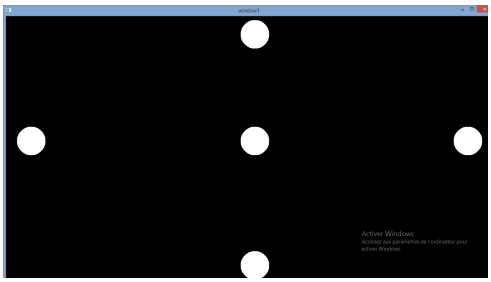


FIGURE II.20 – Image qui présente les étapes(points) de calibration

II.6 Les coordonnées (x,y) de la pupille sur l'écran

Après avoir déterminer les coordonnées (x, y) sur la petite fenêtre, nous allons utiliser une relation mathématique pour trouver le x et le y sur le grand écran, pour cela nous avons considéré le point A la position de la pupille quand l'utilisateur voit en haut et le point B dans la même figure la position de la pupille quand l'utilisateur voit en bas et le C comme un point de vue à droite, et finalement le point D comme un point de vue à gauche. La distance et la différence entre les coordonnées de A et de B en pixel par rapport à la résolution de la petite fenêtre est généralement entre 10 et 15 pixel, nous allons stocker cette distance, et comme la largeur d'un écran de 15" est de 765 pixels, ça veut dire que le point A dans la petite fenêtre représente le pixel numéro zéro dans l'écran et le point B dans la petite fenêtre représente le pixel 765 dans l'écran de façon horizontale, donc si on considère le A comme le pixel numéro zéro, ça veut dire qu'il est un point de départ et le point B comme le pixel numéro 765, c'est un point d'arrivée, la distance calculée de A vers B en pixel est égale à 15 donc le 0 c'est le A et le 15 c'est le B.

$$A = 0 \longrightarrow B = 15$$
$$Pixel = 0 \longrightarrow Pixel = 765$$

Si on considère que la vision de l'utilisateur est en milieu dans un point E dans la petite fenêtre, pour trouver les coordonnées (x, y) sur l'écran de ce point E, nous allons d'abord calculer la distance entre le point A et le point E, et comme ce dernier point est au milieu donc sa valeur est de 7.5, maintenant la position en pixel de ce point E sur l'écran sera calculée par cette relation mathématique $\frac{765}{15} \times 7.5$ pour trouver la coordonnée y et la même chose pour trouver l'autre coordonnée x. Après ceci, nous avons utilisé ces deux coordonnées dans une commande prédéfinie pour placer le curseur sur ces coordonnées dans l'écran pour avoir un marquage qui présente le point de vue de l'utilisateur sur l'écran et qui donne la possibilité de sélectionné les différents icônes du bureau.

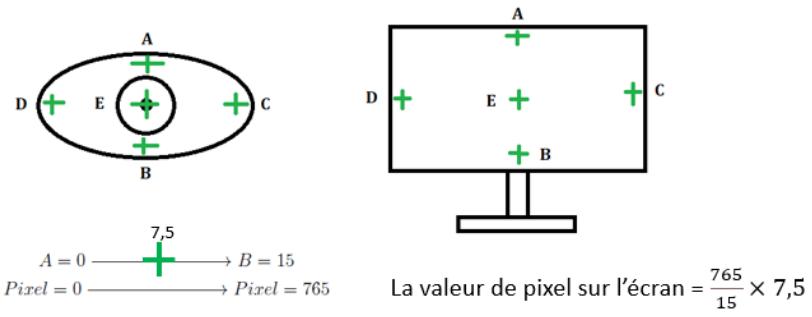


FIGURE II.21 – La relation pour trouver Les coordonnées (x,y) de la pupille sur l'écran

II.7 Lissage

Après la détection de la pupille qui est faite par le traitement d'images pendant chaque itération, cette détection est discrète à cause de la répétition du traitement qui change les coordonnées de la position de la pupille d'une façon négligeable en un premier temps, mais ce petit changement donne un résultat instable et une grande vibration du curseur sur l'écran même si les yeux de l'utilisateur sont fixées sur un point. Pour éliminer ces vibrations, nous avons utilisé une relation mathématique de lissage, cette relation permet de stocker 5 coordonnées de la pupille suivant leurs changements et calculée leur moyenne pour transférer les 5 déplacement à un seul mouvement. Cette relation augmente la précision et la stabilité du curseur sur l'écran avec un petit retard assez négligeable.

II.8 Conclusion

Dans ce chapitre, nous avons présenté la méthodologie utilisée pour réaliser notre but principal qui est le suivi du regard "eye tracking". Cette méthodologie contient deux titre essentiels : le premier titre c'est le traitement d'images qui sert à détecter le visage, détecter et isoler les yeux, détecter l'iris et estimer la position de la pupille par le calcul des moments de l'image binaire qui ne contient que l'iris en noir. Le deuxième titre est l'estimation de la position de la pupille sur l'écran qui sert à déterminer les coordonnées des deux pupilles et faire un marquage sur chacune d'elles afin de satisfaire une calibration et un lissage pour avoir un résultat de mouvement de curseur stable et précis. Ce résultat donne la possibilité à l'utilisateur de contrôler le curseur avec ses yeux et réaliser les différentes opérations sur un IHM. Les répétitions de traitement d'images pendant chaque instruction diminue la stabilité de la position de la pupille même si la tête est fixée et la vision de l'utilisateur est stable sur un point, nous avons régler ce problème par l'utilisation d'une relation mathématique qui fait la moyenne de 5 mouvements pour avoir un seule déplacement plus précis qui traduit vraiment la décision visuelle de l'utilisateur avec un petit retard assez négligeable. Le bon résultat de détection de la pupille effectué par une grande résolution de la caméra, une bonne luminosité de la vidéo et avec une position de visage par rapport à la caméra. La non satisfaction de ces derniers points diminue la stabilité de détection de la pupille et donne un résultat moins précis.

Chapitre III

Applications et Résultats

III.1 Introduction

L'utilisation et les applications des "eye trackers" se trouvent dans plusieurs domaines, parmi lesquels on peut citer : le médical, les interfaces hommes-machines, la commercialisation, les sites web, les jeux, la réalité virtuelle et l'automobile. L'utilisation de l'eye tracker dans ces domaines est basée sur la surveillance des yeux et sur le suivi oculaire afin de traduire les mouvements de l'œil en un déplacement du curseur, en déplacement d'un marquage ou encore en un déplacement d'environnement virtuel. Ces mouvements ne sont pas la seule méthode pour réagir avec les applications basées sur le suivi oculaire, on trouve même le clignement des yeux qui se traduit en une action de clic pour ouvrir, activer et valider les différentes opérations. A partir de tout ça, nous allons travailler dans notre mémoire sur deux applications basé sur l'eye tracking : la première application est la surveillance de la fatigue du conducteur pendant la conduite, cette surveillance concerne la détection de la fatigue du conducteur et détecte même le changement de direction de regard. Ces deux fonctionnalités sont désormais disponibles dans les récents modèles de voitures grâce à leurs rôles dans la protection de la vie humaine par la diminution du nombre d'accidents liés à la perte d'attention. La deuxième application est une application médicale spécialisée pour les personnes handicapés moteurs et les personnes n'ayant pas la possibilité de parler ou communiquer facilement. Cette application aide ces personnes à communiquer avec leurs familles et avec le personnel soignant, en se servant simplement du clignement des yeux. Dans ce chapitre, nous allons détailler les techniques de réalisation et d'utilisation de ces applications.

III.2 Contrôle de curseur avec les yeux

Après avoir détecté la pupille et déterminer ses coordonnées x et y sur l'écran, nous allons utiliser un package python applée *pyautogui* qui donne la possibilité de contrôler la

position de curseur après avoir indiqué les entrées x et y, déjà calculées dans le deuxième chapitre. Par le placement des coordonnés des pupilles dans la commande prédéfinie qui contrôle le déplacement du curseur de façons automatique, ce curseur va prendre la position du regard de l'utilisateur. Avec cette commande, nous allons réaliser un suivi oculaire qui présente un curseur qui suit les mouvements des yeux sur l'écran d'une manière continue.

III.3 Click par clignement des yeux

C'est une continuité de ce que nous avons fait dans le deuxième chapitre, après la détection des yeux avec la bibliothèque *dlib* qui sert à positionner 6 points sur l'œil comme indiqué dans la figure III.1

FIGURE III.1 – Landmarks des yeux

Nous prenons donc les coordonnées des points 37 et 38 et nous les divisons par deux pour avoir un point au milieu, nous prenons ensuite un point qui se trouve au milieu des points 40 et 41 et nous traçons une ligne verticale qui relie les deux milieux comme présenté dans la figure III.2 A. Les coordonnées x et y des deux milieux nous permettent de calculer la distance entre eux, si l'œil est ouvert la ligne prend la plus grande distance, par contre si l'œil est fermé la ligne prend la plus petite distance comme présenté dans la figure III.2 B. Cette petite distance signifie qu'il y a un clignement des yeux. Cette fonctionnalité de clignement est utilisé pour faire une action ou un clic dans notre application médicale pour choisir et ouvrir une certaine fenêtre.

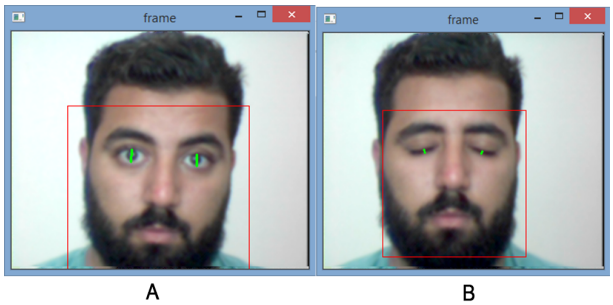


FIGURE III.2 – La ligne verticale des yeux en vert.

III.4 Application de surveillance du conducteur

Le but principal des applications de l'eye tracker dans le domaine de l'automobile est la détection de la fatigue du conducteur pour éviter les accidents et diminuer le nombre des morts dans les routes, cette option est faite à base des techniques de vision par ordinateur qui donne même la possibilité de détecter la direction de regard et la distance du vue sur la rue. Nous avons implémenté deux fonctionnalités dans ce domaine qui présentent une surveillance complète du conducteur au cours de la conduite.

III.4.1 Détection de la fatigue du conducteur

A base de l'option de la détection de clignement, qui a été faite par la mesure de longueur de la ligne verticale, nous avons détecté si le conducteur est fatigué ou non avec une caméra positionné dans la voiture en face du conducteur, cette caméra devrait être connectée avec l'ordinateur de bord qui lance une alarme ou un message vocal au moment où le conducteur est vraiment fatigué. Dans notre projet, nous avons utilisé seulement l'ordinateur portable connecté à une caméra, la fatigue se détecte avec la mesure du temps parce que si le clignement dépasse 4 second, ce n'est plus un clignement normale mais un sommeil. Dans ce cas, les yeux sont fermées et le conducteur est très fatigué. A ce moment-là, l'ordinateur lance une alarme et un message vocale pour alerter le conducteur et lui demander d'arrêter de conduire. Les clignements rapides qui ne dépassent pas les deux secondes ne sont pas pris en compte. Cette option est déjà installé dans les systèmes de bord des voitures modernes, comme elle peut être un module ou un appareil séparé à placer dans toute voiture.

III.4.2 La surveillance de la direction de regard du conducteur

La direction de regard du conducteur est une fonctionnalité très nécessaire, il y a pas mal d'accidents à cause de l'absence de concentration du conducteur sur la route, sans faire attention à la circulation en face, comme par exemple le temps qu'il passe à naviguer ou à communiquer sur son téléphone portable. Nous avons fait cette surveillance après la détection des yeux comme une première étape ensuite par l'isolation des yeux en appliquant une région et un masque sur eux. Ensuite, nous allons prendre juste l'œil dans une figure et couper cette figure en deux, une figure présentant le côté droit et l'autre présentant le côté gauche de l'œil. Nous allons après appliquer un seuillage sur les deux figure pour avoir deux image binaire (noire et blanc) comme c'est présenté dans la figure III.3.

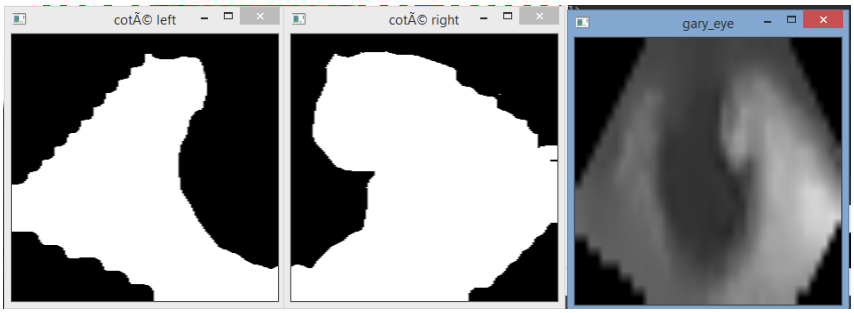


FIGURE III.3 – L'isolation de l'oeil et la séparation de sa figure en deux cotés

Afin d'utiliser une commande prédéfinie dans la bibliothèque OpenCV, qui sert à calculer le nombre des pixels noirs dans chaque figure, si l'utilisateur voir à gauche, la figure qui présente le côté gauche de l'œil est complètement noire, donc on conclut que le conducteur ne voit pas en face et la même chose pour l'autre coté avec une mesure du temps avant prendre une décision et lancer une alarme après 4 secondes dans cet état.

III.5 Application Médicale

III.5.1 Présentation de l'application

La figure III.4 nous présente l'application médicale réalisée dans ce projet, c'est une application essentiellement desktop et qui utilise la vision par ordinateur pour interagir et plus exactement pour faire la sélection. Le but principal de cette application est l'assistance à la communication pour les personnes handicapés moteurs n'ayant pas la capacité physique de communiquer avec leurs entourage de manière naturelle ou intuitive, les icônes qui sont illustrées dans l'interface graphique de l' application (La figure III.4) représentent les besoins les plus fréquents et les plus essentiels que peut rencontrer quotidiennement une personne en situation de handicap moteur, tels que :

- Déclencher une alarme,
- Répondre par Oui ou Non,
- Ouvrir un clavier visuel pour communiquer une autre requête,
- Déclarer qu'il y a des maux dans le corps,
- Solliciter une personne tierce pour des besoins naturels tels que manger, boire, dormir, recevoir les besoins d'hygiène et de confort ou changer la position.

FIGURE III.4 – L'interface graphique de l'application médicale.

III.5.2 Fonctionnement de l'application

La figure III.5 nous montre le chemin du défilement semi-automatique du curseur, ce dernier défile d'une manière souple et dans un seul sens suivant les flèches en rouges en démarrant de la position initiale ($x = 30$, $y = 250$) en passant sur toutes les icônes jusqu'à l'arrivée à la position finale ($x = 1325$, $y = 550$) comme illustré dans la figure III.5. Lorsque le curseur arrive à la position finale, il revient à la position initiale et ce processus de défilement se répète tant que l'application est ouverte. La vitesse de défilement du curseur sur les icônes de l'application a été réglée minutieusement pour permettre une utilisation facile et ergonomique de l'interface et de manière a minimiser le stress de l'utilisateur, en choisissant un pas de déplacement du curseur égale à 7 pixels.

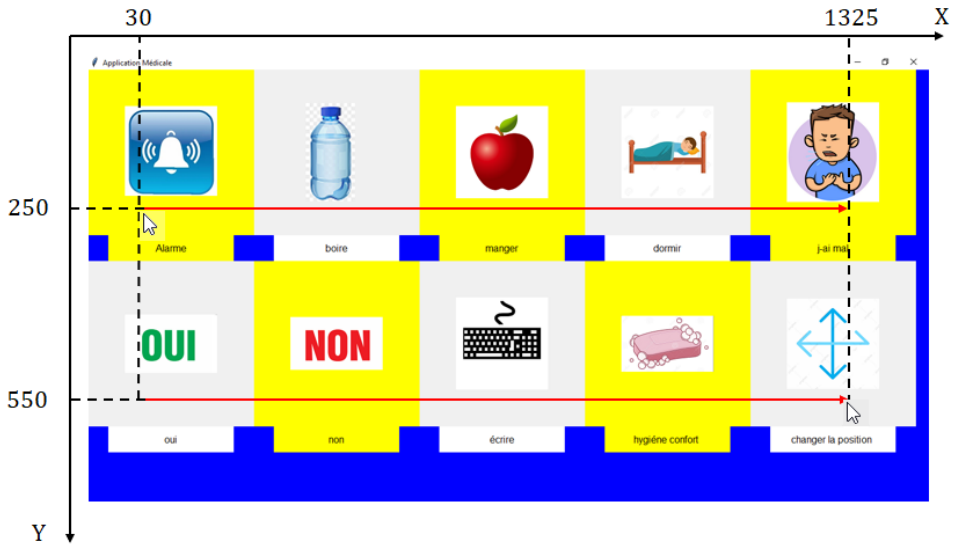


FIGURE III.5 – Le chemin du défilement semi-automatique du curseur dans un seul sens suivant les flèches en rouges

La sélection du besoin désirée est effectué par un simple clignement des yeux un peu

plus retardé que le clignement normal, ce choix est exécuté si le clignement des yeux survient et lorsque la position du curseur se trouve dans l'intervalle de l'icône souhaitée.

III.5.3 Tkinter

Tkinter est une bibliothèque graphique libre pour le langage Python, permettant la création des interfaces graphiques et puisque notre application est développée à travers cet outil nous allons expliquer les étapes de création de l'application en suivant la structure du programme.

Tout d'abord, la ligne de code suivante permet d'importer l'ensemble des éléments (*) du module *Tkinter* dans l'espace du programme principale

```
from tkinter import*
```

Ensuite, la création de la fenêtre principale de l'application par l'instanciation d'un nouveau objet qui est notre fenêtre d'application du *Tkinter*

```
window=Tk()
```

La configuration de la taille de la fenêtre de l'application, la couleur de l'arrière plan, le titre, la taille et l'emplacement de la fenêtre sur l'écran est configuré à l'aide de **geometry** qui accepte comme argument d'entrée une chaine de caractère de la forme générale :

```
"wxh± x± y"
```

Et dans le programme de l'application médicale nous avons :

```
window.geometry('1500x1500+0+0')
```

Avec :

- $w = 1500$ et $h = 1500$ donnent la largeur et la hauteur de la fenêtre en pixels, elles sont séparées par le caractère "x".
- ± x± y (+0+0 dans notre application), elle spécifie que le côté gauche de la fenêtre doit être à 0 pixel du côté gauche du bureau et le haut de la fenêtre doit être à 0 pixel en dessous du haut du bureau.

La configuration de la couleur d'arrière plan de l'application à l'aide du **config** qui accepte comme argument d'entrée une chaine de caractère qui représente la couleur de l'arrière plan (background ou bg) de l'application.

```
window.config(background='blue')
```

Le titre de l'application est affectée par **title** sous forme d'une chaine de caractère.

window.title('Application Médicale')

Ensuite, les étapes les plus importantes dans l'interface c'est la création des images des besoins dans des buttons et le texte en bas de chaque bouton, pour cela nous allons expliquer pour le premier besoin qu'est l'alarme et le principe est le même pour les autres buttons.

Il faut lire les images avec **PhotoImage** pour les pouvoir utiliser dans les boutons :

alarme1 = PhotoImage(file='alarmepng.png')

Ensuite la création des boutons avec **Button** qui accepte comme argument d'entrée plusieurs paramètres dont on cite l'essentiel :

alarme = Button(window, ...width=165,height=165,...image=alarm1).grid(column=0, row=0)

- La fenêtre parent ou on veut créer ce bouton dans notre cas "window"
- La taille du bouton width× height=165×165 en pixels
- L'affectation de l'image en entrée du **Button** déjà lu précédemment image=alarm1.
- **grid** pour faire une grille dans la fenêtre précisant la ligne et la colonne de chaque button et de chaque texte, la fenêtre de l'application est décomposée de 4 lignes et 5 colonnes.

Après, nous créons les textes des besoins avec la classe **Label** qui accepte des arguments un peu similaire à **Button** juste il faut préciser le text à afficher, 'Alarme' dans le cas du premier texte.

txtalarme = Label(window,text='Alarme',...).grid(column=0,row=1)

Enfin, quand tous les boutons et les textes sont créés dans l'application comme il est expliqué à travers les exemples précédents, il reste à exécuter l'interface.

window.mainloop()

III.5.4 Thread

Un *thread* ou fil d'exécution est une entité qui contient une ou plusieurs tâches. Les différents threads d'un programme sont exécutés simultanément de point de vue utilisateur ce qui permet une interaction avec l'application en cours d'exécution d'où l'intérêt de les utiliser, et nous savons qu'un programme est une suite d'instructions exécutées dans l'ordre et puisque nous avons dans notre application un traitement de vidéo avec l'interaction avec l'application et que ces deux tâches doivent être simultanées, alors sans l'utilisation du *thread* nous aurons un blocage de l'application.

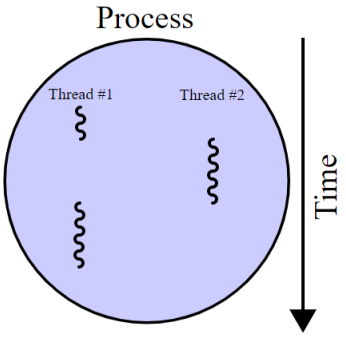


FIGURE III.6 – Un processus avec deux threads [14]

Dans python pour utiliser Thread, il faut importer le module :

```
from threading import Thread
```

Dans cette application nous avons utilisé deux threads : un pour le traitement de vidéo et l'autre pour le clavier visuel. Pour charger ces deux tâches dans des threads il faut qu'elle soient dans des fonctions. Prenons l'exemple du thread du traitment de vidéo :

```
t1 = Thread(target=video, args=(1,))
t1.start()
t1.join()
```

Avec :

- **target** : la fonction cible à exécuter dans le nouveau fil d'exécution,
- la méthode **start()** de l'instance Thread pour démarrez le thread,
- la méthode **join()** de l'instance Thread pour que le thread principal attendra que le second thread soit terminé avant de s'arrêter.

III.5.5 Le son dans l'application

Le son est utilisé dans cette application comme une exécution des différents besoins de l'utilisateur présents dans l'interface principale, par exp. le déclenchement d'une alarme sonore, demande du changement de la position ... etc. Ceci est Comme une confirmation de l'exécution de la tâche désirée car quand l'utilisateur cligne les yeux pour sélectionner ou choisir une lettre dans le clavier visuel pour une meilleure interaction avec lui on a mis un son similaire à celle du click de la souris ou un bip sonore dans le clavier pour il sache qu'il a effectivement sélectionné. Pour cela nous avons utlisé le module **playsound**, tout d'abord on importe ce module.

```
import playsound
```

Par exemple pour lire le son en format MP3 de l'alarme on lance premièrement le click de la souris pour une confirmation d'exécution et ceci est valable pour toutes les autres activités.

```
playsound.playsound("Button_Click.mp3")
time.sleep(1)
playsound.playsound("alarm2.mp3")
time.sleep(7)
```

FIGURE III.7 – Lecture des sons de click puis de l'alarme avec l'utilisation des temps de retard pour que le programme attend que la lecture des sons sera terminé.

Et dans le clavier visuel chaque sélection d'un caractère est suivie d'un bip de confirmation de l'exécution

```
playsound.playsound("click.mp3")
```

FIGURE III.8 – La lecture du son de bip de sélection du clavier visuel.

III.5.6 Le clavier visuel

Le clavier visuel de communication par clignement des yeux (la figure III.9) est sous forme d'une matrice contenant des zéros (arrière plan noire) de dimension(650×1320) créée par **Numpy**,[1] et il est composé de deux parties (1) c'est l'ensemble des 44 caractères (dans le programme ces 44 caractères sont stockés dans un dictionnaire) et (2) qui représente la partie d'affichage du texte, ou l'arrière-plan de cette partie est modifiée en blanc dans la matrice (fenêtre) du clavier par $([422 : 650, :] = 255)$ pour pouvoir affiché le texte dans la même figure qui contient les caractères.

1. NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques.

Dans le clavier visuel de communication (la figure III.9) un texte de 11 caractères **AUTOMATIQUE** est écrit par une personne entrainée sur ce clavier qui a mis 4 fautes et un temps total de $15\,m\ :\ 33\,s, 07$ pour écrire ce texte.

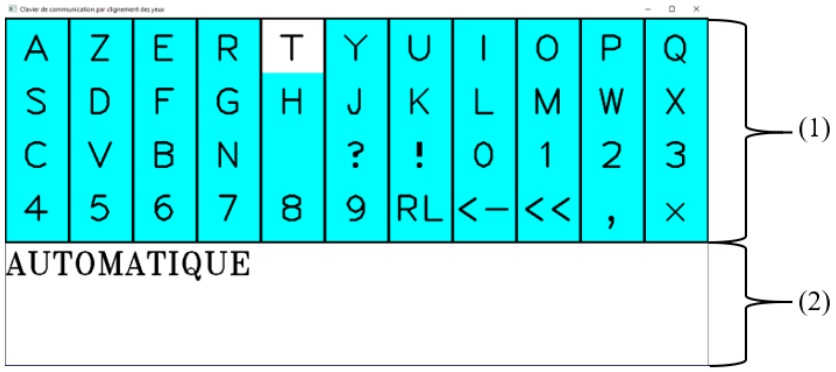


FIGURE III.9 – Le clavier visuel de communication par clignement des yeux.

III.5.6.1 Fonctionnement du clavier

Le clavier visuel fonctionne d'une manière similaire à l'application. La seule différence c'est que le défilement est fait par un carreau de couleur blanche qui se déplace d'une façon discrète et la sélection du caractère est faite par un clignement des yeux. Un bip sonore est émis pour la confirmation d'exécution et le caractère sélectionné est affiché dans la partie (2) du clavier.

III.5.6.2 Description des symboles fréquents sur le clavier

La figure III.10 nous présente les différents symboles qui existe dans le clavier visuel.

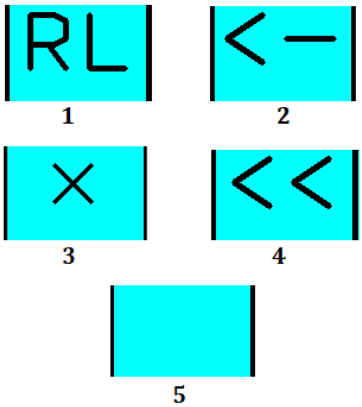


FIGURE III.10 – Les symboles fréquents sur le clavier visuel de communication.

De la figure III.10 nous avons :

1. Retour à la ligne,

2. Retour arrière,
3. Retour à l'interface principale,
4. Supprimer tout,
5. Espace.

III.5.7 L'exécutable de l'application

Cette étape consiste à transformer le programme principal de notre application en un fichier exécutable (la figure III.11). Lors cette conversion, nous avons eu des erreurs et après plusieurs essais nous avons détecté qu'il y a des conflits entre deux modules **pynput** et **pyglet** qui sont utilisés pour commander le curseur de la souris et la lecture des sons MP3. Lorsque nous avons changé ces deux modules par d'autre modules qui ne présente pas des conflits entre eux, la conversion en un fichier exécutable a été réussie.

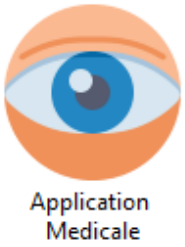


FIGURE III.11 – Le fichier exécutable de l'application médicale.

III.6 Résultat et Discussion

Tout d'abord, nous allons mesurer le temps de sélection de chaque besoin présent dans notre application par une personne entrainé sur cette application.

TABLE III.1 – Les temps de sélection des besoins présent dans l'application par une personne entrainé sur cette interface graphique.

Besoin désiré	**Temps de sélection**
Alarme	1s,89
Boire	5s,29
Manger	8s,41
Dormir	12s,39
Mal	15s,02
Oui	18s,06
Non	21s,68
Écrire	26s,60
Hygiène confort	29s,48
Changer la position	35s,28

Ensuite, nous allons mesurer le temps de réponse du clavier visuel pour communication par clignement des yeux par une personne entrainée sur ce clavier en fonction de quelques chaines de caractère, et les résultats obtenus de ces tests sont présentés dans les figures suivantes :

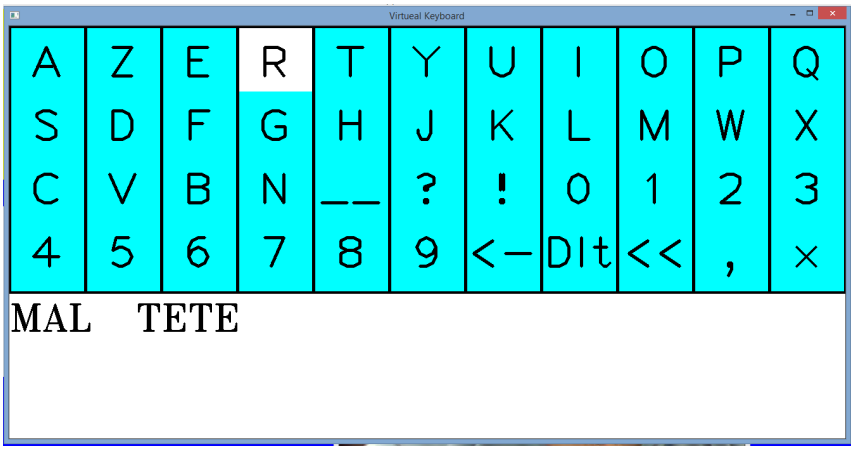


FIGURE III.12 – Résultat d'écriture de la chaine de caractère MAL TETE par le clavier visuel.

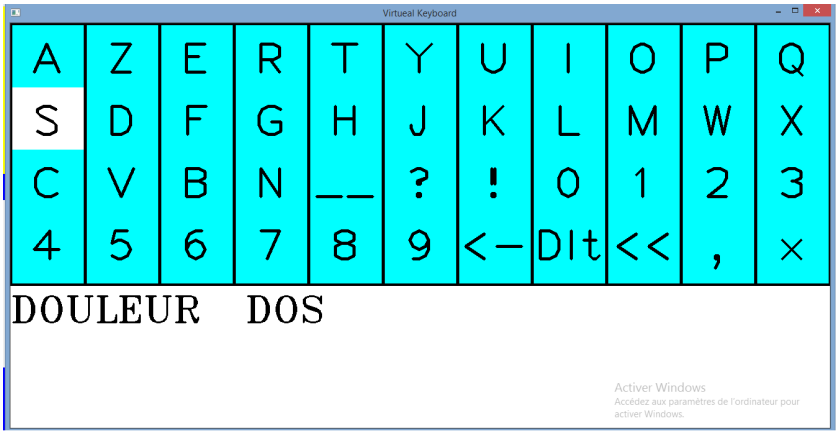


FIGURE III.13 – Résultat d'écriture de la chaine de caractère DOULEUR DOS par le clavier visuel.

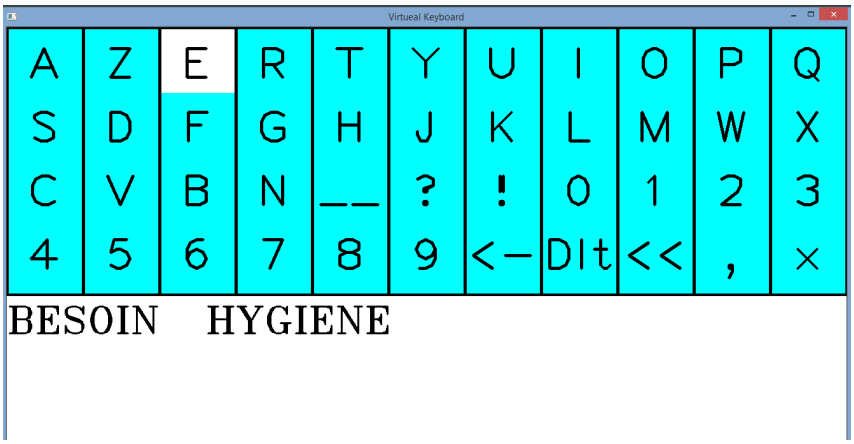


FIGURE III.14 – Résultat d'écriture de la chaine de caractère BESOIN HYGIENE par le clavier visuel.

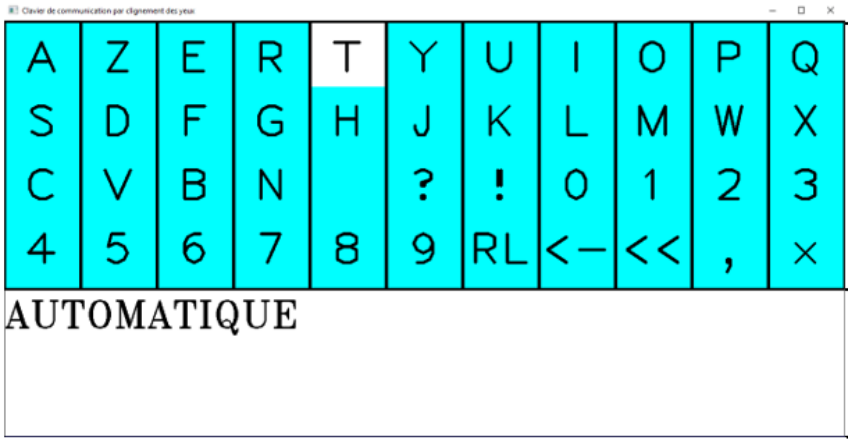


FIGURE III.15 – Résultat d'écriture de la chaine de caractère AUTOMATIQUE par le clavier visuel.

On résume tous les résultats des tests de clavier visuel de communication par clignement des yeux dans le tableau suivant :

TABLE III.2 – Temps de réponse du clavier visuel en fonction des chaines de caratère

Chaine de caractère	**Nombre de caractère**	**Temps**	**Nombre d'erreurs**
MAL TETE	8	8m : 04s	1
DOULEUR DOS	11	12 m : 51s	0
BESOIN HYGIENE	14	20m : 19s	3
AUTOMATIQUE[2]	11	15m : 33s	4
AUTOMATIQUE[3]	11	09m : 20s	0

2. 1 èr essai
3. 2 ème essai

III.6.1 Discussion

Nous remarquons bien que les résultats obtenus dans le test de l'application (Table. III.1) pour effectuer une sélection de l'application du besoin "déclenchement d'une alarme" qui se trouve dans la position initiale dans le chemin du curseur, nous mettons un temps d'environ $1.89s$ tandis que pour sélectionner le besoin du "changement de position" nous mettons un temps d'environ $35.28s$ et le temps entre ces deux sélections augmente parce que la distance parcouru par le déplacement semi-automatique du curseur augmente également.

Pour les résultats des tests du clavier visuel de communication par clignement des yeux (Table. III.2), pour une chaine de caractère de 8 caractères, nous avons mis un temps de $8m : 04s$ avec la présence d'une erreur et pour la plus grande chaine de caractère dans les tests effectués qui est "BESOIN HYGIENE" avec un nombre de caractère égale à 14 nous avons enregistré un temps de $20m : 19s$ avec la présence de 3 erreurs. On conclut à travers ces deux cas que plus le nombre de caractère augmente le temps d'écriture visuelle augmente ainsi que le nombre d'erreurs, mais c'est pas le cas toujours car dans les deux chaines de caractère DOULEUR DOS et AUTOMATIQUE(2 ème essai) qui sont avec des nombres de caractère et des nombres d'erreurs égaux la première a enregistrée un temps de $12m : 51s$ et la deuxième a fait un temps de $9m : 20s$ et ça c'est dû aux lettres qui compose ces deux chaines de caractères et leurs classement dans le clavier visuel .

III.7 Conclusion

Dans ce chapitre, nous avons présenté deux applications, basé sur la détection de clignement des yeux afin de transférer ce clignement en une action ou un signal qui représente une consigne de l'utilisateur. Ces deux applications sont : premièrement la surveillance du conducteur qui détecte l'état de ses yeux pendant la conduite et détecte même la direction de son regard, ces deux fonctionnalités sont très nécessaires dans une voiture pour protéger la vie humaine on prend en considération la fiabilité de ce système pour qu'il ne dérange pas le conducteur, d'où le choix d'une durée d'attente de 4 secondes avant de déclencher l'alarme. Si le conducteur dépasse cette période et ses yeux restent fermées ou sa direction de regard n'est pas sur la route, le système lance une alarme et un message vocal pour éviter les accidents. La deuxième application est une application médicale qui aide les personnes handicapés moteurs à communiquer avec d'autre personnes seulement avec les yeux. L'application affiche un ensemble d'icônes, chaque icône présente un besoin principal de l'utilisateur, le choix des besoins se fait juste par le clignement des yeux qui est utilisé même pour l'écriture visuelle avec le clavier alphanumérique. Cette application est efficace pour connaitre les besoins des malades handicapés moteurs et pour connaitre l'état psychologique des malades. L'utilisation de ces deux applications nécessite

que la position de l'utilisateur soit proche du caméra pour avoir des résultats justes et précis.

Conclusion générale et perspectives

Ce mémoire avait pour ambition de faire du suivi oculaire "Eye Tracking" en temps réel ; qui est une technologie utilisée pour suivre le regard d'une personne sur un écran à travers l'estimation des coordonnées de son point de regard en utilisant un seul dispositif qu'est la webcam de l'ordinateur, c'est à dire essayer de faire une réalisation minimale en termes de coût, et en utilisant les outils de la vision par ordinateur par le biais de la bibliothèque libre OpenCV.

Pour répondre à cette problématique, il a fallu, dans un premier temps, détecter les pupilles des deux yeux de l'utilisateur afin de les suivre. Dans cette tâche, nous avons rencontré plusieurs obstacles d'ordre technique tels que le cas où l'iris est foncé, ce qui a rendu est difficile de différencier entre les deux contours de la pupille et de l'iris par la caméra. Ceci est du principalement à l'utilisation d'une caméra à faible résolution. Nous avons dépassé ce problème par l'estimation de la position de la pupille grâce à l'estimation du centre de l'iris.

Après l'estimation de la position des deux pupilles nous sommes passé à l'étape de calibration qui est une étape clé dans ce domaine de suivi oculaire pour déterminer la relation entre le déplacement des deux centres des deux pupilles dans l'image de la caméra et le déplacement du curseur dans l'écran. Dans cette étape, la combinaison des deux calibrations des déplacements selon l'axe des x et selon l'axe des y, l'erreur a nettement grandit) cause de la mise à l'échelle (calibration), ce qui nous a donné une instabilité dans le déplacement du curseur, d'où l'intérêt d'automatiser cette tâche par un déplacement semi automatique et analogique du curseur et de faire la sélection (arrêt du curseur) par le clignement des yeux pour gagner en stabilité du système malgré la perte au niveau des options. Ceci a permis de ne pas perturber et stresser l'utilisateur par les mouvements indésirables du curseur autour de la position de regard de l'utilisateur et de gagner au niveau du contrôle de l'application.

Une première perspective à court terme de ce travail pourrait être d'augmenter la précision et la stabilité du système de suivi oculaire par l'utilisation d'autres méthodes de détection et de prendre en considération les mouvements de tête de l'utilisateur.

Une deuxième perspective consiste en l'intégration de certaines modalités comme la commande par un Bouton ON/OFF où l'utilisation d'un détecteur EMG (électromyographique) qui permet de mesurer les courants électriques qui accompagnent l'ac-

tivité musculaire pour utiliser cette dernière information comme une commande via un micro-contrôleur. Ces deux modalités peuvent-être utilisées pour le contrôle de l'interface développée dans ce mémoire pour les personnes handicapés moteurs n'ayant pas la possibilité d'utiliser l'application dans son état actuel.

Une dernière perspective serait d'ajouter de nombreuses fonctionnalités à cette application qui permettront de contrôler l'environnement d'une manière automatique comme par exemple les applications de domotique : allumer et éteindre les lumières, la climatisation, les rideaux, etc. pour que les personnes handicapés moteurs acquièrent une certaine indépendance dans leurs vie quotidienne.

Annexe

Listing III.1 – Programme python de l'application de surveillance du conducteur

```
import cv2
import dlib
import numpy as np
from math import hypot
import time
import pyglet

sound_fatigue = pyglet.media.load("la_fatigue.mp3")
sound_alarme = pyglet.media.load("alarm.m4a")
sound_direction = pyglet.media.load("la_direction.mp3")

font = cv2.FONT_HERSHEY_TRIPLEX
cap = cv2.VideoCapture(0)
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")


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


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 = cv2.line(frame, left_point, right_point, (0, 255, 0), 1)
ver_line = cv2.line(frame, center_top, center_bottom, (0, 255, 0), 1)

```

```
31      ver_line_lenght = hypot((center_top[0] - center_bottom[0]), (
    center_top[1] - center_bottom[1]))
32      hor_line_lenght = hypot((left_point[0] - right_point[0]), (
    left_point[1] - right_point[1]))
33      center = ((hor_line_lenght) / (ver_line_lenght))
34      print("center", center)
35      print("verticale ligne", ver_line_lenght)
36      print("horisontale ligne", hor_line_lenght)
37      return center
38
39
40      def get_gaze_ratio(eye_point, facial_ladmarks):
41      left_eye_region = np.array([(facial_ladmarks.part(eye_point[0]).x,
    facial_ladmarks.part(eye_point[0]).y),
42      (facial_ladmarks.part(eye_point[1]).x, facial_ladmarks.part(
    eye_point[1]).y),
43      (facial_ladmarks.part(eye_point[2]).x, facial_ladmarks.part(
    eye_point[2]).y),
44      (facial_ladmarks.part(eye_point[3]).x, facial_ladmarks.part(
    eye_point[3]).y),
45      (facial_ladmarks.part(eye_point[4]).x, facial_ladmarks.part(
    eye_point[4]).y),
46      (facial_ladmarks.part(eye_point[5]).x, facial_ladmarks.part(
    eye_point[5]).y)], np.int32)
47
48      height, width, _ = frame.shape
49      mask = np.zeros((height, width), np.uint8)
50      cv2.polylines(mask, [left_eye_region], True, 255, 2)
51      cv2.fillPoly(mask, [left_eye_region], 255)
52      eye2 = cv2.bitwise_and(gray, gray, mask=mask)
53      print(left_eye_region)
54      cv2.polylines(frame,[left_eye_region],True,(0,0,255),2)
55      min_x = np. min(left_eye_region[:, 0])
56      max_x = np. max(left_eye_region[:, 0])
57      min_y = np. min(left_eye_region[:, 1])
58      max_y = np. max(left_eye_region[:, 1])
59
60      gray_eye = eye2[min_y:max_y, min_x:max_x]
61
62      #gray_eye=cv2.cvtColor(eye,cv2.COLOR_BGR2GRAY)
63
64      _, threshold_eye = cv2.threshold(gray_eye, 120, 255, cv2.
    THRESH_BINARY)
65      eye1 = cv2.resize(gray_eye, None, fx=5, fy=5)
66      thresh = cv2.resize(threshold_eye, None, fx=5, fy=5)
67      height, width = threshold_eye.shape
68      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)
    print("right_side_white", right_side_white)
    print("left_side_white", left_side_white)
    if left_side_white == 0:
    gaze_ratio = 5
    elif right_side_white == 0:
    gaze_ratio = 0.2

    else:
    gaze_ratio = (left_side_white) / (right_side_white)
    cv2.imshow("eye1", eye1)
    cv2.imshow("THRESH", thresh)

    return gaze_ratio


    timme = 0
    timme2 = 0
    while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)
    for face in faces:
    print(face)
    x, y = face.left(), face.top()
    w, h = face.right(), face.bottom()
    cv2.rectangle(frame, (x, y), (w, h), (0, 0, 255), 1)
    landmarks = predictor(gray, face)
    left_eye_center = get_blinking_ratio([36, 37, 38, 39, 40, 41],
landmarks)
    right_eye_center = get_blinking_ratio([42, 43, 44, 45, 46, 47],
landmarks)
    center = (left_eye_center + right_eye_center) / 2
    if center > 4:
    if  timme >10:

    sound_alarme = pyglet.media.load("alarm.m4a")
    sound_fatigue = pyglet.media.load("la_fatigue.mp3")

    sound_alarme.play()
    time.sleep(1)
    sound_fatigue.play()
    # cv2.putText(frame, "blinking", (50, 150), font, 1, (0, 0, 255))
    time.sleep(5)
```

```
113     timme += 1
114     else:
115     timme = 0
116 
117     gaze_ratio_left_eye = get_gaze_ratio([36, 37, 38, 39, 40, 41],
    landmarks)
118     gaze_ratio_right_eye = get_gaze_ratio([42, 43, 44, 45, 46, 47],
    landmarks)
119     gaze_ratio = (gaze_ratio_left_eye + gaze_ratio_right_eye) / 2
120 
121     if gaze_ratio < 1.5 :
122     if timme2>10:
123     cv2.putText(frame, "right", (100, 200), font, 2, (0, 0, 255), 3)
124     # sound_alarme = pyglet.media.load("alarm.m4a")
125     # sound_direction = pyglet.media.load("la_direction.mp3")
126     # sound_alarme.play()
127     # time.sleep(1)
128     # sound_direction.play()
129     # time.sleep(5)
130     timme2 += 1
131 
132     elif gaze_ratio > 3 :
133     if timme2>10:
134 
135     cv2.putText(frame, "left", (100, 200), font, 2, (0, 0, 255), 3)
136     # sound_alarme = pyglet.media.load("alarm.m4a")
137     # sound_direction = pyglet.media.load("la_direction.mp3")
138     # sound_alarme.play()
139     # time.sleep(1)
140     # sound_direction.play()
141     # time.sleep(5)
142     timme2 += 1
143     else:
144     timme2 = 0
145     cv2.putText(frame, "center", (100, 200), font, 2, (0, 0, 255), 3)
146     print("timme=",timme)
147     print("timme2=",timme2)
148     cv2.putText(frame,
    str(gaze_ratio), (50, 100), font, 2, (0, 0, 255), 3)
149     cv2.imshow("frame", frame)
150     # cv2.imshow("img2", eye2)
151     # cv2.imshow("imgG", RECT2)
152     cv2.waitKey(1)
153 
154     cv2.release()
155     cv2.destroyAllWindows()
156 
```

Bibliographie

[1] Accuracy and precision test method for remote eye trackers. https://tinyurl.com/yjg75nnf. Consulté le 17/Avril/2021.

[2] Basic thresholding operations. https://tinyurl.com/yfg7wzso. Consulté le 03/Mai/2021.

[3] Eye gaze inc. https://eyegaze.com/. Consulté le 04/Avril/2021.

[4] Eye gaze users diagnoses. https://eyegaze.com/users/#diagnoses. Consulté le 06/Avril/2021.

[5] Eye tracker accuracy and precision. https://tinyurl.com/yhwz87tt. Consulté le 17/Avril/2021.

[6] Eye tracking dikablis glasses3. https://www.ergoneers.com/en/hardware/dikablis-glasses/. Consulté le 07/Avril/2021.

[7] Eye tracking dikablis glasses3 product flyer. https://tinyurl.com/yftqrnqq. Consulté le 07/Avril/2021.

[8] Eyegaze edge® tech specifications. https://eyegaze.com/products/eyegaze-edge/. Consulté le 06/Avril/2021.

[9] Eyelink 1000 plus. https://www.sr-research.com/eyelink-1000-plus/. Consulté le 03/Avril/2021.

[10] Eyelink 1000 plus brochure. https://tinyurl.com/yfgjgwf3. Consulté le 03/Avril/2021,page 17.

[11] Pro fusion product description. https://tinyurl.com/yjgelyqj. Consulté le 02/Avril/2021,pages 6 et 10-12.

[12] Pro glasses 3 product description. https://tinyurl.com/ydnvpml5. Consulté le 13/Avril/2021.

[13] Python grayscaling of images using opencv. https://tinyurl.com/yeanvj65. Consulté le 05/Mai/2021.

[14] Thread (informatique). https://fr.wikipedia.org/wiki/Thread_(informatique). Consulté le 08/Juin/2021.

[15] Timing guide for tobii eye trackers and eye tracking software. https://tinyurl.com/yjpfbdcd. Consulté le 17/Avril/2021.

[16] Tobii pro fusion. https://www.tobiipro.com/product-listing/fusion/. Consulté le 02/Avril/2021.

[17] *opencv computer vision with python*. packt publishing ltd, 2013.

[18] samsung lauches eye tracker mouse for people with disabilities. *www.deccznchronicle.com*, 2014.

[19] *Eye Tracking The Complete Pocket Guide*. • , 2018.

[20] *Eye Movement Research An Introduction to its Scientific Foundations and Applications*. the registered company Springer Nature Switzerland AG, 2019.

[21] eyetracker. *www.usability.gov*, 2021.

[22] Stijn DE BEUGHER. Computer vision techniques for automatic analysis of mobile eye-tracking data. Master's thesis, University of Bielefeld, 2016.

[23] Stijn DE BEUGHER. Computer vision techniques for automatic analysis of mobile eye-tracking data. Master's thesis, ARENBERG DOCTORAL SCHOOL Faculty of Engineering Technology, November 2016.

[24] castleman. digital image processing.

[25] Dongheng Li, D. Winfield, and D.J. Parkhurst. Starburst : A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, volume 3, pages 79–79, San Diego, CA, USA, 2005. IEEE.

[26] Heiko Drewes. Eye gaze tracking for human computer interaction. Master's thesis, Dissertation an der LFE Medien-Informatik der Ludwig-Maximilians-Universität München, 2010.

[27] A.T. Duchowski. A breadth-first survey of eye tracking applications. *Methods Instrum. Comput*, 2002.

[28] Anjith George. Image based eye gaze tracking and its applications. Master's thesis, ELECTRICAL ENGINEERING DEPARTMENT INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR, AUGUST 2017.

[29] kovasznay. image processing. *proceedings of the IRE*, 1995.

[30] Dongheng Li, Jason Babcock, and Derrick J. Parkhurst. openEyes : a low-cost head-mounted eye-tracking solution. In *Proceedings of the 2006 symposium on Eye tracking research & applications - ETRA '06*, page 95, San Diego, California, 2006. ACM Press.

[31] Abdallahi Ould Mohamed, Matthieu Perreira Da Silva, and Vincent Courboulay. A history of eye gaze tracking. 2007.

[32] Senthilkumaran N and Vaithegi S. Image Segmentation By Using Thresholding Techniques For Medical Images. *Computer Science & Engineering : An International Journal*, 6(1) :1–13, February 2016.

[33] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédo Durand. *Bilateral filtering : Theory and applications*. Now Publishers Inc, 2009.

[34] md zahidur saifur rahman partha chakraborty, dipa roy. eye gaze controlled virtual keyboard. *international journalof recent technologie and engineering*, 2019.

[35] maria mp petrou. image processing.

[36] Selimis and Vasileios. Eye-tracking technology in vehicles : application and design. Master's thesis, City University London), 2015.

[37] Vasileios Selimis. Eye-tracking technology in vehicles : Application and design. Master's thesis, City University London, 2015.

[38] Vasileios Selimis. Eye-tracking technology in vehicles : Application and design. Master's thesis, City University London, August 2015.

[39] K Sreedhar and B Panlal. Enhancement of images using morphological transformation. *arXiv preprint arXiv :1203.2514*, 2012.

[40] thabo beeler. Analysis and improvement of facial landmark detection. *Researchgate*, 2019.

[41] Lerothol Thite and Ronnie Brown. The history of eye tracking. Master's thesis, University of the Free State, South Africa.

[42] Marc Tonsen, Julian Steil, Yusuke Sugano, and Andreas Bulling. InvisibleEye : Mobile Eye Tracking Using Multiple Low-Resolution Cameras and Learning-Based Gaze Estimation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3) :1–21, September 2017.

[43] Barbara Wąsikowska. The application of eye tracking in business. *ResearchGate*, 2017.

[44] Zhang Yun, Zhao Xin-Bo, Zhao Rong-Chun, Zhou Yuan, and Zou Xiao-Chun. EyeSecret : an inexpensive but high performance auto-calibration eye tracker. In *Proceedings of the 2008 symposium on Eye tracking research & applications - ETRA '08*, page 103, Savannah, Georgia, 2008. ACM Press.

Résumé

L'eye tracker est une technologie moderne qui permet de suivre le regard de l'utilisateur pour deux objectifs principaux : le premier est l'analyse de l'état des yeux où l'estimation du point de regard d'une personne sur une IHM, le deuxième est l'interaction avec ces IHMs en traduisant les mouvements oculaires et les clignements des yeux en actions. Cette technologie se décline en deux techniques : la première faisant appel à un certain équipement physique en contact avec l'utilisateur sous forme de lunettes pour analyser le regard des personnes dans le monde réel, et la deuxième technique nécessite une caméra fixe pour récupérer le point de regard de l'œil à distance. Cette dernière technique est plus rapide et plus facile, et permet d'utiliser le système de suivi oculaire pendant des périodes plus longues que la première technique, bien que la précision des systèmes à distance est en général inférieure à celle des systèmes à contact. Dans notre mémoire, nous avons utilisé la deuxième technique pour réaliser deux applications : la première c'est la surveillance du conducteur pendant la conduite afin de détecter sa fatigue et la direction de son regard, et la deuxième est une application médicale pour l'aide à la communication par les yeux pour les personnes handicapés moteurs sur une IHM.

Mot clés : Eye Tracking, suivi oculaire, OpenCV, Dlib, MultiThreading, face landmarks, détection de visage, automobile, IHM, handicap moteur, détection de la pupille, calibration.

ملخص

تتبع النظر هو تقنية حديثة تسمح بمتابعة نظرالمستخدم لهدفين رئيسيين: الأول هو تحليل حالة العينين أو تقدير نقطة نظر الشخص على واجهات إنسان ـ آلة، والثاني هو التفاعل مع هذه الواجهات من خلال ترجمة حركات العين ورمش العينين إلى أفعال. هذه التكنولوجيا تتضمن تقنيتين : الأولى باستخدام معدات مادية معينة على تلامس مع المستخدم في شكل نظارات لتحليل نظرالمستخدمين في العالم الحقيقي، والتقنية الثانية تتطلب كاميرا ثابتة لتحديد نقطة نظر العين عن بعد. وهذه التقنية الأخيرة أسرع وأسهل، وتسمح باستخدام نظام تتبع العين لفترات زمنية أطول من التقنية الأولى، على الرغم من أن دقة الأنظمة البعيدة أقل عموما من دقة الأ نظمة على تلامس مع المستخدم. في مذكرتنا، استخدمنا التقنية الثانية لبرمجة تطبيقين: الأول هو مراقبة السائق أثناء القيادة من أجل الكشف عن تعبه واتجاه بصره، والثاني هو تطبيق طبي لمساعدة الأشخاص ذوي الإعاقات الحركية على التواصل بإستخدام العينين على واجهة المستخدم.

الكلمات المفتاحية : تتبع النظر، اوبنسيفي، دليب، تعدد مؤشرات الترابط، معالم الوجه، تحديدالوجه، مجال السيارات، واجهات إنسان ـ آلة، إعاقة حركية، تحديد موقع البؤبؤ، القياس .