

1.5em



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre /IA/M2/2022

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence Artificielle (IA)**

Lunettes Intelligentes Basées sur l'Apprentissage Profond pour Aider Les aveugles

**Par : Makhlouf Dounia
Daci Radia
Mohammedi Amira**

Soutenu le 02/07/2023 devant le jury composé de :

Saouli Rachida	Professeur	Présidente
ZOUAI Meftah	MCB	Encadrant
AMMARI Asma	MCB	Encadrante
Belaala Abir	MCB	Examinatrice

Année universitaire 2022-2023



Dédicace

Nous dédions ce travail, fruit de nombreuses années d'études, à :

À nos chers parents pour leur soutien, leur travail acharné et leurs sacrifices
pour nous en tout temps et tout au long de notre vie scolaire,

À tous nos professeurs, en particulier Meftah ZOUAI et Asma AMMARI,

À tous ceux qui se sont engagés dans ces humbles travaux,

À nos chers frères et sœurs pour leur soutien et leurs encouragements,

À tous nos amis et notre famille,

À toutes les personnes qui nous ont aidées de près ou de loin à accomplir ce
travail.

Merci.

Amira, Dounia et Radia.

Remerciement

Tous d'abord, Nous tiendrons à remercier le bon Dieu de nous avoir accordé toute la détermination, la volonté et la force pour qu'on puisse réaliser ce modeste travail.

Nous voudrions également être reconnaissante infiniment à nos superviseurs **Dr. Meftah ZOUAI**, **Dr. Asma AMMARI** pour leurs efforts et leurs conseils, leurs patiences, leurs disponibilités et leur soutien tout au long de cette période.

Nous voudrions exprimer notre reconnaissance envers notre parents notre sœur et notre frères, qui nous ont apporté leur support moral et les discussions animées tout au long de notre démarches .

Nous tenons à exprimer notre profonde gratitude et nos sincères remerciements aux membres de jury d'avoir accepté de juger notre travail et de l'avoir enrichi.

Summary

According to the World Health Organization (WHO), around 285 million people have visual impairments worldwide, of which 39 million are blind and 246 million have low visual acuity. Computer vision and deep learning technology have played an important role in helping blind people to perform their daily activities without the need for other people. Smart glasses are one of those solutions that allow the blind or visually impaired to become independent.

In our project, we made a prototype of smart glasses that have a camera that compensates for human vision, because these glasses can recognize the surrounding objects and obstacles, the texts in the boards, as well as the menu, and finally read the description for the blind. Based on the latest version of YOLO for object detection, OCR for text recognition, and TTS for reading, all in real time.

Key words : deep learning, computer vision, object detection, obstacles, visual impairment, text recognition, Text to Speech .

Résumé

Selon l'Organisation mondiale de la santé (OMS), environ 285 millions de personnes ont une déficience visuelle dans le monde, dont 39 millions sont aveugles et 246 millions ont une acuité visuelle réduite. La vision par ordinateur et la technologie d'apprentissage en profondeur ont joué un rôle important en aidant les malvoyants à effectuer leurs activités quotidiennes sans avoir besoin d'autres personnes. Les lunettes intelligentes font partie de ces solutions qui permettent aux personnes aveugles ou ayant des difficultés visuelles d'être autonomes. Dans notre projet, nous avons réalisé un prototype de lunettes intelligentes contenant une caméra qui compense la vision humaine, car ces lunettes peuvent reconnaître les objets et les obstacles environnants et les textes dans les panneaux, ainsi que le menu, et enfin lire la description pour les aveugles. basé sur la dernière version YOLO pour la détection d'objets et OCR pour la reconnaissance de texte et TTS pour la lecture, et le tout en temps réel.

Mots-clés : apprentissage en profondeur, vision par ordinateur, détection d'objets, obstacles, déficience visuelle, reconnaissance de texte, synthèse vocale

Table des matières

Dédicace	I
Remerciement	II
Introduction Générale	1
I Internet des Objets	4
I.1 Introduction	4
I.2 Concepts de base	5
I.2.1 Objet connecté (OC)	5
I.2.1.1 Définitions	5
I.2.1.2 Composants d'un OC	6
I.2.1.3 Propriétés d'usage d'un OC	6
I.2.1.4 Caractéristiques de l'objet connecté	6
I.2.2 Communication machine à machine (M2M)	7
I.2.3 Internet des objets (IoT)	8
I.3 Quatre piliers d'une solution IoT	8
I.3.1 Hardware(Device)	8
I.3.2 Connectivité	9

I.3.3	Data	9
I.3.4	Valeur	9
I.4	Fonctionnement de L'IoT	10
I.5	Architecture IoT	10
I.5.1	Éléments de base d'une architecture IoT	11
I.5.1.1	Couche de perception	12
I.5.1.2	Couche réseau	12
I.5.1.3	Couche de traitement de données	13
I.5.1.4	Couche Applicative	14
I.5.1.5	Couche de sécurité	14
I.6	Importance de l'internet des objets	14
I.7	Avantages Et Inconvénients	15
I.8	Domaines d'applications de l'IoT	15
I.8.1	IoT dans le secteur médical	16
I.8.2	IoT dans l'agriculture (Smart Farming)	16
I.8.3	IoT dans la domotique (Maison Intelligente)	17
I.8.4	IoT dans l'Industrie	18
I.9	Conclusion	19
II	Apprentissage en Profondeur et Vision par Ordinateur	20
II.1	Introduction	20
II.2	Intelligence artificielle (IA)	21
II.3	Apprentissage automatique	22
II.3.1	Réseau de neurones	23
II.3.2	Types d'Apprentissage automatique	24

II.3.2.1	Apprentissage non supervisé	24
II.3.2.2	Apprentissage supervisé	24
II.3.2.3	Apprentissage semi-supervisé	25
II.3.2.4	Apprentissage par renforcement	25
II.4	Apprentissage en profond	25
II.4.1	Types d'apprentissage en profondeur	25
II.4.1.1	Un réseau neuronal convolutif(CNN)	26
II.4.1.2	Réseau neuronal récurrent (RNN)	26
II.4.1.3	Réseau Antagonistes Génératifs (GAN)	27
II.4.2	Domaines d'applications de l'apprentissage en profondeur	27
II.4.2.1	Traitement d'images	27
II.4.2.2	Traitement du langage naturel	27
II.4.2.3	Reconnaissance de la parole	28
II.4.2.4	Traduction automatique	28
II.4.2.5	Diagnostic médical	28
II.4.2.6	Négociation financière	28
II.5	Vision par ordinateur	29
II.5.1	Définition	29
II.5.2	Tâches importantes en vision par ordinateur	30
II.5.2.1	Classification d'images	30
II.5.2.2	Détection d'objets	30
II.5.2.3	Segmentation d'images	31
II.5.3	Domaines d'applications de vision par ordinateur	34
II.5.3.1	Agriculture	34

II.5.3.2	Imagerie médicale	34
II.5.3.3	Éducation	34
II.5.3.4	Transport	35
II.6	Conclusion	35
III	Contexte Générale	36
III.1	Introduction	36
III.2	Non-voyant	36
III.3	Déficience visuelle	37
III.3.1	Définitions	37
III.3.2	Types de déficience visuelle	37
III.3.3	Degrés de déficience visuelle	38
III.3.4	Formes de malvoyance	38
III.4	Etude de l'environnement	40
III.4.1	Types d'obstacles	40
III.4.2	Défis des aveugles	41
III.5	Travaux connexes	41
III.6	Conclusion	46
IV	Conception du Système	47
IV.1	Introduction	47
IV.2	Lunettes intelligentes	47
IV.3	Architecture général	48
IV.4	Fonctionnement du Système	49
IV.5	Conception Détaillée	49

IV.5.1 Lunettes	50
IV.5.1.1 Module de perception	50
IV.5.1.2 Module de communication	50
IV.5.2 Station de calcule	50
IV.5.2.1 Détection des objets	50
IV.5.2.2 Reconnaissance de texte	56
IV.5.3 Application mobile	58
IV.6 Conclusion	60
V Implémentation du Système	61
V.1 Introduction	61
V.2 Environnement de développement	61
V.2.1 Langages de programmation et Les bibliothèques	62
V.2.1.1 Java	62
V.2.1.2 Python	62
V.2.1.3 OpenCV	63
V.2.1.4 Pytesseract	63
V.2.1.5 PyTorch	64
V.2.1.6 NumPy	64
V.2.1.7 Keras	65
V.2.1.8 Tensorflow	66
V.2.2 Les outils de développement	66
V.2.2.1 Pycharm	66
V.2.2.2 Jupyter Notebook	67
V.2.2.3 Android Studio	67

V.2.2.4	Firebase	68
V.3	Schéma Electronique	68
V.3.1	Esp32-Cam	69
V.3.2	FTDI Driver	70
V.4	Installation du matériel	71
V.5	Station de traitement	72
V.5.1	Détection d'objets et texte	72
V.5.2	Détection d'objets	76
V.5.3	Reconnaissance de texte	76
V.5.4	Application mobile	77
V.6	Résultats	82
V.7	Conclusion	86
	Conclusion Générale	87
	Bibliographie	88

Table des figures

I.1	Les 4 piliers d'une solution IoT [1]	10
I.2	Une architecture IoT a 4 couches [2]	12
I.3	Domaines d'applications de l'IoT [3]	15
I.4	Porteuse Digital Health [3]	16
I.5	IoT dans l'agriculture [4]	17
I.6	IoT dans la domotique [5]	18
I.7	IoT dans l'Industrie [6]	19
II.1	L'intelligence artificielle [7]	21
II.2	L'apprentissage automatique[8]	22
II.3	Architectures des réseaux de neurones [9]	23
II.4	Classification et Régression [10]	24
II.5	L'apprentissage en profondeur[11]	26
II.6	Human Vision vs. Computer Vision [12]	29
II.7	Classification [13]	30
II.8	Detection des Object [14]	31
II.9	Segmentation d'images [15]	32
IV.1	Architecture général	48

IV.2 Architecture Détaillée	49
IV.3 Classes d'algorithmes de détection d'objets [16]	51
IV.4 Architecture de YOLOv8	54
IV.5 Algorithme de détection d'objets	55
IV.6 Étapes procédurales pendant le processus d'OCR	56
IV.7 Composants du moteur TTS	58
V.1 Java[17]	62
V.2 Python[18]	63
V.3 OpenCV[19]	63
V.4 Pytesseract OCR[20]	64
V.5 PyTorch[21]	64
V.6 Numpy[22]	65
V.7 Keras[23]	65
V.8 Tensorflow[24]	66
V.9 Pycharm[25]	67
V.10 Jupyter Notebook[26]	67
V.11 Android Studio[27]	68
V.12 Firabase[28]	68
V.13 Schéma électronique[29]	69
V.14 ESP32-CAM[30]	69
V.15 FTDI Driver[31]	70
V.16 Prototype de projet	72
V.17 Résultats de détection des objects exemple 1	82
V.18 Résultats de détection des objects exemple 2	82

V.19 Résultats de reconnaissance de texte	83
V.20 Résultats d'application mobile interface 1	84
V.21 Résultats d'application mobile interface 2	84
V.22 Résultats d'application mobile interface 3	85
V.23 Temps de reponse	85

Liste des tableaux

- III.1 Comparaison des travaux connexes 43
- V.1 Connexion de l'esp32-cam avec le FTDI [29] 71

Listings

V.1	Les bibliothèques nécessaires pour se connecter firebase	72
V.2	Détection d'objets	73
V.3	Les bibliothèques nécessaires pour la détection des objets	76
V.4	Les bibliothèques nécessaires pour la reconnaissance de texte	76
V.5	Les bibliothèques nécessaires pour l'application mobile	77
V.6	Application mobile	78

Introduction Générale

Contexte

Selon l'Organisation mondiale de la santé (OMS), environ 285 millions de personnes souffrent de déficiences visuelles dans le monde, dont 39 millions sont aveugles et 246 millions ont une basse vision[32]. En Algérie, ils représentent 1,5% de la population totale estimée à 44 millions en 2023. Les personnes aveugles [33] souffrent de discrimination notamment dans le monde du travail, et c'est un problème social majeur, donc la vie sociale est compliquée par le handicap. Le simple fait de se sentir différent des gens ordinaires provoque une anxiété psychologique chez l'individu, car le handicap de l'aveugle lui impose un monde limité.

Problématique et motivation

Les personnes aveugles éprouvent des difficultés dans l'exercice de leurs fonctions quotidiennes (se déplacer , notamment dans de nouveaux lieux, lire des journaux, des magazines et des livres).

Avec les avencenemt de l'intelligence artificiel un systeme intelligente base sur l'iot sera switable et beneficiaire pour aider les aveugles

Dans ce contexte : Comment l'aveugle lire un document ? Comment reconnaît-il les choses et les objets qui l'entourent et leur position ?

Objectif

Les aveugles peuvent être accompagnés et assistés par : Assistance humaine ou utilisation d'un chien-guide, aide à l'utilisation de la technologie (lunettes connectives, ordinateurs compatibles, cannes aveugles).

Dans notre projet, nous fabriquerons des lunettes intelligentes pour aider les personnes aveugles à entendre le monde qui les entoure. Le but de ce projet est de redonner l'espoir à l'aveugle dans son quotidien, en connaissant ce qui l'entoure, en se déplaçant librement, et en gérant ses affaires pour lui-même en tant que personne physique sans avoir besoin des autres .

Après l'introduction générale, la mémoire sera divisée en cinq chapitres comme suit :

Chapitre I : Internet des objets. nous aborderons le concept de l'internet des objets, son fonctionnement, de s'architecturer et de ses domaines d'application, etc.

Chapitre II : Apprentissage en profondeur Et Vision par ordinateur . nous aborderons l'apprentissage en profondeur et ces types et certains concepts connexes, tels que l'intelligence artificielle et les réseaux de neurones et ses domaines d'application , la vision par ordinateur et ses tâches importantes et ses domaines d'application.

Chapitre III : Contexte general. nous aborderons la déficience visuelle, ses types et ses degrés, ainsi que les difficultés des personnes aveugles dans leur vie quotidienne, dont nous verrons les solutions par les chercheurs.

Chapitre IV : Conception du Système. Dans ce chapitre, nous allons décrire le système que nous allons développer et expliquer le fonctionnement de chacune de ses parties.

Chapitre V : Implémentation du système. Ce chapitre présentera les outils de mise en œuvre et les détails du code ainsi que les résultats obtenus.

Conclusion générale et perspective.

Chapitre I

Internet des Objets

I.1 Introduction

L'internet des objets (Anglais : IoT, Français : IdO) fait maintenant partie de nos vies, et représente la prochaine étape de l'évolution d'Internet tel que l'intégration d'internet avec la communication sans fil et les réseaux de capteurs sans fil intégrés apportera un changement de paradigme en transformant nos appareils quotidiens en appareils intelligents et sensibles au contexte [34]. En raison de ses structures technologiques, de ses parts de marché, de ses valeurs et de ses profits, il a trouvé sa place dans presque tous les aspects de la vie humaine. Il est également envisagé que dans un avenir proche, Internet sera intégré à de nombreux objets tels que les vêtements, les brosses à dents et les emballages alimentaires[35]

Dans ce chapitre, nous allons étudier en détail l'Internet des Objets, ses piliers, son fonctionnement, ses domaines d'application, et tout ce qui rapporte

I.2 Concepts de base

Dans cette section, nous définissons l'internet des objets et ses piliers comme suivant :

I.2.1 Objet connecté (OC)

I.2.1.1 Définitions

Avant de définir les concepts d'IdO, il est important de définir l'objet connecté [36] qui est un dispositif dont la finalité première n'est pas d'être un système informatique ni une interface d'accès au web, exemple, un objet tel qu'une machine à café ou une serrure était conçue sans intégration de systèmes informatiques ni connexion à Internet. L'intégration d'une connexion Internet a un OC permet de l'enrichir en terme de fonctionnalité, d'interaction avec son environnement, il devient un OC Enrichi (OCE), par exemple, l'intégration d'une connexion internet à la machine à café la rendant accessible à distance.

Un OC peut interagir avec le monde physique de manière indépendante sans intervention humaine.

Il possède plusieurs contraintes telles que la mémoire, la bande passante ou la consommation d'énergie, etc. Il doit être adopté à un usage, il a une certaine forme d'intelligence, une capacité de recevoir, de transmettre des données avec des logiciels grâce aux capteurs embarqués . Un objet connecté a une valeur lorsqu'il est connecté à d'autres objets et briques logicielles, par exemple :une montre connectée n'a d'intérêt qu'au sein d'un écosystème orienté santé/bien-être, qui va bien au-delà de connaître l'heure.

I.2.1.2 Composants d'un OC

- Les données produites ou reçues, stockées ou transmises.
- Les algorithmes pour traiter ces données.
- L'écosystème dans lequel il va réagir et s'intégrer [37].

I.2.1.3 Propriétés d'usage d'un OC

- Ergonomie (utilisabilité, maniabilité, ...).
- Esthétisme (formes/couleurs/sons/sensations, ...).
- Usage (histoire culturelle, profil, matrice sociale, ...).
- Méta-Morphisme (adaptabilité, personnalisation, modulation, ...) [38].

Certains chercheurs parlent des « hyper objets » [MAV 2003] capables de mutualiser leurs ressources afin d'effectuer des tâches du quotidien, ils sont reliés par des « liens invisibles » au sein d'un même écosystème.

I.2.1.4 Caractéristiques de l'objet connecté

Dans cette section, nous allons voir les Caractéristiques de l'objet connecté [39] :

- **Adressage** : Chaque objet sur l'Internet des objets doit avoir une identité unique. Grâce à l'adresse IPv6 (Internet Protocol Version 6), l'adresse IP (Internet Protocol) de nouvelle génération 128 bits peut fournir 2^{128} adresses différentes.
- **Communication** Chaque objet doit pouvoir communiquer. Il existe un certain nombre de technologies sans fil modernes qui rendent les communications possibles, telles que le WiFi, le Bluetooth Low Energy, ZigBee

et 6LoWPAN.

- **Capteur** Chaque objet doit avoir des capteurs afin que nous puissions obtenir des informations à son sujet (capteurs de température, d'humidité, de lumière..) . Les nouveaux capteurs sont de plus en plus petits, moins chers et plus durables.
- **Microcontrôleur** Chaque objet doit avoir un microcontrôleur (ou microprocesseur) pour gérer les capteurs et les communications, et pour effectuer les tâches. Il existe de nombreux microcontrôleurs qui pourraient être utilisés pour l'IoT.

I.2.2 Communication machine à machine (M2M)

La communication machine à machine (M2M) est une technologie prometteuse pour les systèmes de communication de nouvelle génération. Ce paradigme de communication facilite les communications omniprésentes avec une automatisation mécanique complète, où un grand nombre de dispositifs intelligents connectés par des liaisons filaires/sans fil, interagissent les uns avec les autres sans intervention humaine directe. En conséquence, la communication M2M trouve des applications dans de vastes domaines tels que les réseaux intelligents, les soins de santé en ligne, les réseaux domestiques, les systèmes de transport intelligents, la surveillance de l'environnement, les villes intelligentes et l'automatisation industrielle [40].

I.2.3 Internet des objets (IoT)

Une technologie construite sur la base du concept que tous les appareils souhaités peuvent être connectés à l'intérieur d'un réseau spécifique pour partager des données et des informations sans aucune intervention manuelle. Les appareils utilisés à cette fin peuvent être séparés en fonction de leur capacité à envoyer, recevoir et collecter des données dans le réseau donné [40].

L'Internet des objets (IoT) [39] fait référence au réseau d'objets physiques, Il connaît une croissance rapide et compte déjà des milliards d'appareils connectés . Ceci est différent de l'Internet actuel, qui est en grande partie un réseau d'ordinateurs, y compris des téléphones et des tablettes, capables d'échanger des données entre eux sans intervention humaine. Ils sont accessibles et contrôlables à distance.

Le concept de connexion d'appareils entre eux n'est pas nouveau. En 1982, une machine à Coca à l'Université Carnegie Mellon est devenue le premier appareil connecté à Internet. Il pourrait garder une trace de l'inventaire et savoir si les boissons étaient froides.

I.3 Quatre piliers d'une solution IoT

La solution iot présenté par un pyramide telle que [1] :

I.3.1 Hardware(Device)

Device est l'ensemble des objets du terme Internet des objets

— **Capteur** :Le capteur mesure une variable du monde réel(température).

- **Gateway :** Une gateway est une passerelle qui va envoyer les données captées sur le réseau
- **Actionneur :** Le donnée “captée” peut être actionnée localement les actionneurs déclenchent une action sans intervention humaine pour qu’elles soient analysées.

I.3.2 Connectivité

Le hardware est le T(thing) du terme IoT, la connectivité représente le I(internet). Elle est le cœur de votre solution IoT, le pont entre le monde physique et le monde digital. L’objectif final de la connectivité est de remonter les données collectées par votre device dans le cloud.

I.3.3 Data

La création de valeur d’une solution IoT ne réside ni dans son capteur ou ses capacités de stockage mais bien dans sa capacité à transformer, analyser et interpréter les données récupérées.

I.3.4 Valeur

Il s’agit du but ultime d’une solution IoT : créer de la valeur pour l’utilisateur final.

La Figure [I.1](#) montrant Les 4 piliers d’une solution IoT.

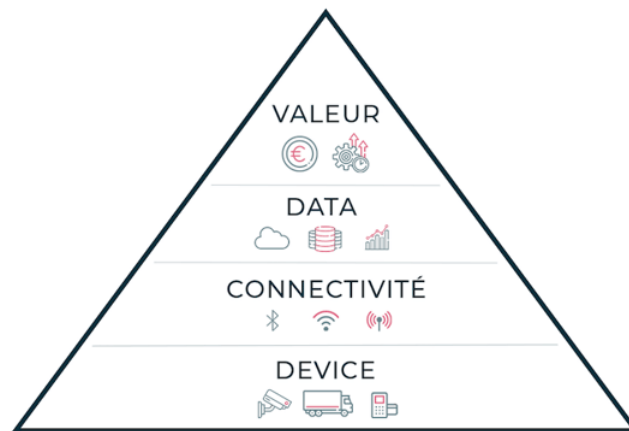


FIGURE I.1 – Les 4 piliers d’une solution IoT [1]

I.4 Fonctionnement de L’IoT

Nous allons voir les étapes pour faire fonctionner l’Internet des objets (IoT) :

Un objet connecté à l’IoT est capable de collecter des données sur son environnement via des capteurs. Il exploite ensuite ces dernières via des micro-processeurs avant de les envoyer à un ou plusieurs destinataires à l’aide de son matériel de communication intégré. Il partage les données collectées par le biais d’une passerelle IoT. Il s’agit d’une solution permettant la communication entre appareils ou entre ceux-ci et le cloud. Les informations sont donc transférées directement vers le cloud à des fins d’analyse et d’exploitation ou à un autre appareil IoT pour une analyse locale[41].

I.5 Architecture IoT

L’architecture IoT comprend plusieurs briques de systèmes IoT connectés pour garantir que les données des objets générées par les capteurs sont collec-

tées, stockées et traitées dans les big data warehouse et que les actionneurs des objets exécutent les commandes envoyées via une application utilisateur.

C'est un cadre qui définit les composants physiques, l'organisation fonctionnelle et la configuration du réseau, les procédures opérationnelles et les formats de données à utiliser[42].

Cependant, il n'existe pas d'architecture de référence standard unique pour l'IoT car elle englobe une variété de technologies.

I.5.1 Éléments de base d'une architecture IoT

L'architecture IoT [2] peut en fait varier considérablement en fonction de la mise en œuvre. Elle doit être suffisamment ouverte avec des protocoles ouverts pour pouvoir prendre en charge plusieurs applications réseau.

Dans la majeure partie des cas elle se compose de 4 blocs constitutifs :

- * La scalabilité
- * La fonctionnalité
- * La disponibilité
- * La maintenabilité

Même s'il n'existe pas d'architecture IoT unique universellement acceptée, le format le plus basique et le plus largement accepté est une architecture IoT à quatre couches. La Figure I.2 montrant Une architecture IoT a 4 couches.

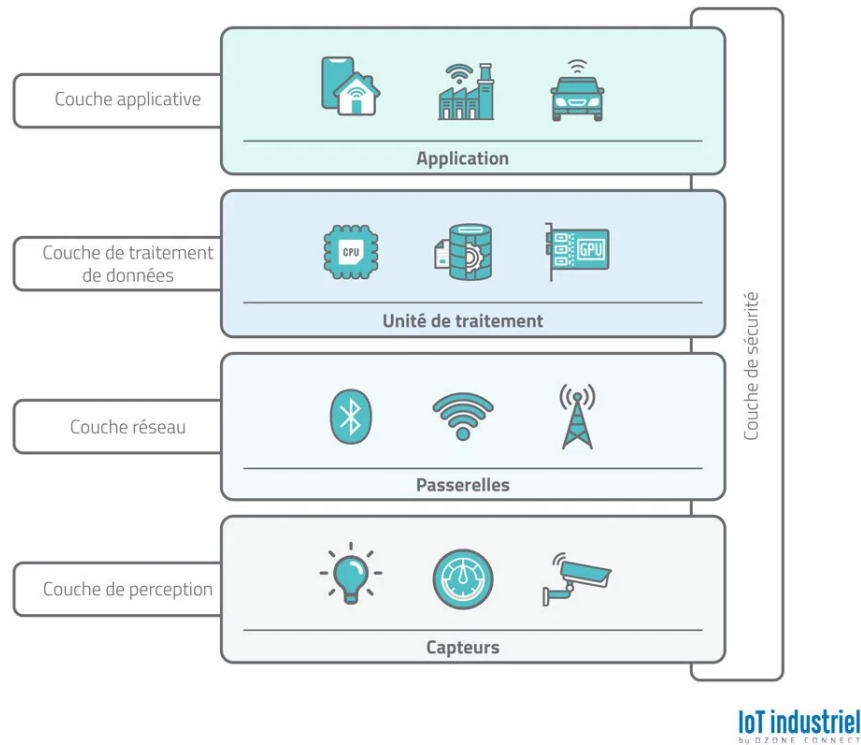


FIGURE I.2 – Une architecture IoT a 4 couches [2]

I.5.1.1 Couche de perception

Cette couche est responsable de convertir des signaux analogiques en données numériques et vice versa. L'étape initiale de tout système IoT englobe un large éventail d'objets qui agissent comme un pont entre les mondes réel et numérique. Leur forme et leur taille varient, des minuscules puces de silicium aux gros véhicules. Par leurs fonctions.

I.5.1.2 Couche réseau

Les données collectées par tous ces appareils doivent être transmises et traitées. C'est le travail de la couche réseau, elle connecte donc ces appareils à d'autres objets intelligents, serveurs et appareils réseau. Elle gère également la

transmission de toutes les données. Les communications entre les appareils et les services cloud ou les passerelles impliquent différentes technologies :

- **Ethernet** : connecte des appareils IoT fixes tels que des caméras de sécurité et vidéo, des équipements industriels installés en permanence.
- **Réseaux cellulaires** : Les plus connus actuellement étant la 5G et la 4G, elles offrent un transfert de données fiable et une couverture presque mondiale. Il existe aussi deux normes cellulaires développées spécifiquement pour les objets IoT. LTE-M (Long Term Evolution for Machines) permet aux appareils de communiquer directement avec le cloud et d'échanger de gros volumes de données. NB-IoT ou Narrowband IoT utilise des canaux basse fréquence pour envoyer de petits paquets de données.
- **LPWAN (Low-power Wide-area Network)** : a été créée spécifiquement pour les appareils IoT. Cette technologie offre une connectivité sans fil longue portée avec une faible consommation d'énergie avec une autonomie de plus de 10 ans. Nous vous conseillons fortement de consulter notre comparatif sur les différentes technologies LPWAN
- **WiFi** : la technologie de réseau sans fil la plus populaire, convient parfaitement aux solutions IoT gourmandes en données, faciles à recharger et à utiliser dans une petite zone. Un bon exemple d'utilisation est celui des appareils domestiques intelligents connectés au réseau électrique.

1.5.1.3 Couche de traitement de données

La couche de traitement accumule, stocke et traite les données provenant de la couche précédente. Toutes ces tâches sont généralement traitées via des

plateformes IoT et comprennent deux étapes principales.

I.5.1.4 Couche Applicative

La couche application est ce avec quoi l'utilisateur interagit. C'est ce qui est chargé de fournir des services spécifiques à l'application à l'utilisateur. Actuellement, les applications peuvent être construites directement sur les plateformes IoT qui offrent une infrastructure de développement logiciel avec des outils prêts à l'emploi pour l'exploration de données, l'analyse avancée et la visualisation de données. Sinon, les applications IoT utilisent des API pour s'intégrer à la couche précédente.

I.5.1.5 Couche de sécurité

Cette couche est transverse à toutes les couches précédentes. La sécurité de l'IoT est primordiale, nous avons observé certaines vulnérabilités comme celle du Log4j.

I.6 Importance de l'internet des objets

L'IoT est rapidement devenu l'une des technologies les plus importantes dans notre monde. Nous pouvons interconnecter des milliards d'appareils à travers le monde, ce qui conduit à une communication transparente, presque en temps réel, le tout avec une intervention humaine minimale et des coûts relativement faibles[43].

I.7 Avantages Et Inconvénients

- Rationalisation de la consommation des ressources énergétiques.
- Accès ubiquitaire à l'information.
- Amélioration de la productivité, de la qualité d'expérience des clients et du gain de temps.
- Optimiser la prise de décision, stratégique ou commerciale.
- Des problèmes de sécurité des données et de confidentialité .
- Pas de norme internationale de compatibilité dans ce domaine[44] .

I.8 Domaines d'applications de l'IoT

Les domaines d'application de l'Internet des Objets sont nombreux, La Figure I.3 montrant les domaines d'application de l'IoT :

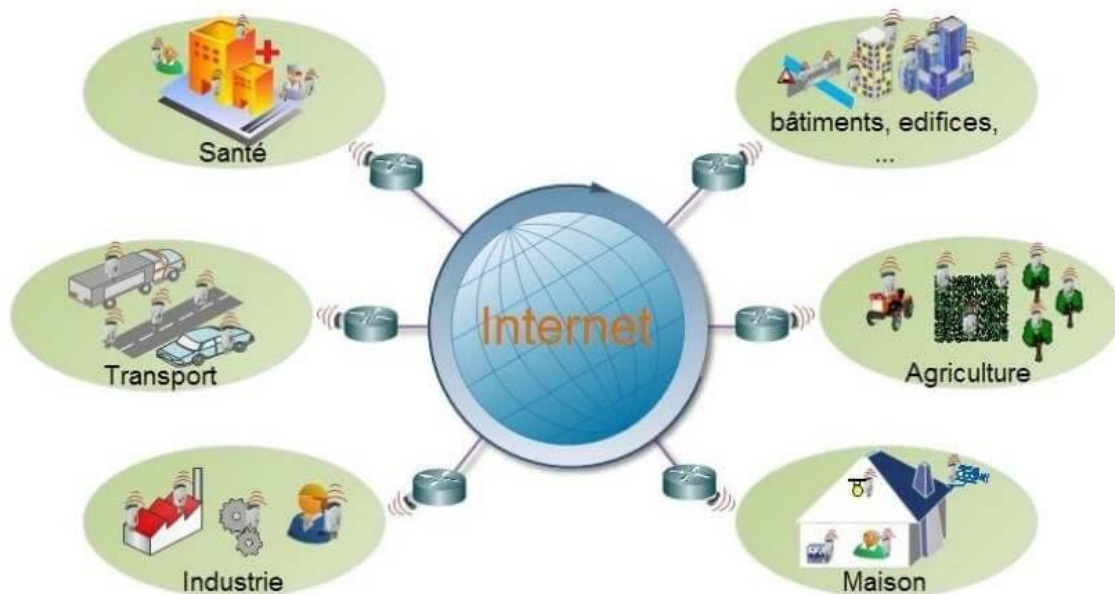


FIGURE I.3 – Domaines d'applications de l'IoT [3]

I.8.1 IoT dans le secteur médical

Dans le domaine médical, l'internet des objets sert à améliorer les opérations médicales. Il aide aussi à surveiller les établissements médicaux et à assurer leur maintenance. L'IoT, dans le secteur de la santé, sert à maximiser la productivité des personnes soignantes sans demander trop de leurs temps et de leurs énergies[3] .

Par exemple , (Porteuse Digital Health) qui est le premier médicament connecté sur le marché grâce à un capteur directement intégré dans l'être humain qui permet après ça le suivi des patients à distance Comme le montre la Figure I.4 Porteuse Digital Health suivante.



FIGURE I.4 – Porteuse Digital Health [3]

I.8.2 IoT dans l'agriculture (Smart Farming)

L'agriculture moderne se doit de proposer une nourriture saine et abondante. Pour augmenter la productivité de l'industrie alimentaire, il a fallu relever un défi technologique. C'est à ce moment qu'intervient l'IoT [4]. Grâce à des drones connectés, il est possible d'améliorer le rendement agricole. Cet appareil peut

faire un état des lieux d'une exploitation. Il peut faire un rapport sur l'état des cultures. Il peut aussi vérifier l'humidité de la terre. Si celle-ci est sèche, il va transmettre les informations de façon autonome à un tracteur qui lui aussi est connecté. Ces informations vont permettre de faire un bon dosage d'engrais pour que la récolte puisse être bonne Comme le montre la Figure I.5 IoT dans l'agriculture (Smart Farming).



FIGURE I.5 – IoT dans l'agriculture [4]

I.8.3 IoT dans la domotique (Maison Intelligente)

La domotique est l'un des secteurs dans lesquels l'IoT est le plus utilisé. Aujourd'hui, à travers votre mobile ou vos appareils connectés, vous pouvez allumer/éteindre votre lampe ou votre chauffage, surveiller en temps réel et à distance votre maison, évaluer votre consommation d'énergie, connaître la température et l'humidité de chaque pièce de votre habitation [5]. Comme le montre la Figure I.6 IoT dans la domotique (Maison Intelligente) suivant :



FIGURE I.6 – IoT dans la domotique [5]

I.8.4 IoT dans l'Industrie

Les industriels peuvent bénéficier d'un avantage concurrentiel grâce à la surveillance des chaînes de production, afin de mettre en place une maintenance proactive des équipements sur lesquels les capteurs détectent une défaillance imminente. Les capteurs peuvent déterminer le moment où la production sera compromise. À l'aide des alertes émises par les capteurs, les industriels peuvent rapidement s'assurer de la précision du matériel ou le retirer de la production jusqu'à sa réparation.

Les entreprises peuvent ainsi réduire leurs coûts d'exploitation, améliorer les temps de bon fonctionnement et mieux gérer les performances des actifs [6] La Figure I.7 montrant IoT dans l'Industrie.



FIGURE I.7 – IoT dans l'Industrie [6]

I.9 Conclusion

Dans ce chapitre, nous avons abordé le concept de l'Internet des Objets, ses caractéristiques, et quelques domaines d'application, Dans le chapitre suivant, nous verrons deux nouveaux concepts d'apprentissage en profondeur et de vision par ordinateur, leurs principes et certains domaines de leur applications.

Chapitre II

Apprentissage en Profondeur et Vision par Ordinateur

II.1 Introduction

L'apprentissage en profondeur et la vision par ordinateur sont des domaines interdépendants qui visent à faire en sorte que les machines et les ordinateurs «voient» et comprennent le monde visuel. L'apprentissage en profondeur et la vision par ordinateur sont utilisés pour créer des systèmes de vision par ordinateur capables d'effectuer des tâches d'analyse d'images qui étaient auparavant trop difficiles pour les logiciels informatiques conventionnels. Dans ce chapitre, nous verrons le concept de deep learning incluant l'intelligence artificielle et le machine learning, puis la vision par ordinateur, sa définition, les tâches, et enfin ses applications.

II.2 Intelligence artificielle (IA)

L'intelligence Artificielle (IA) est la science dont le but est de faire par une machine des tâches que l'homme accomplit en utilisant son intelligence.

consiste à mettre en œuvre un certain nombre de techniques visant à permettre aux machines d'imiter une forme d'intelligence réelle [7].

désigne des systèmes ou des machines qui imitent l'intelligence humaine pour effectuer des tâches et qui peuvent s'améliorer de manière itérative en fonction des informations qu'ils recueillent [45]. Comme le montre la Figure suivante [II.1](#)

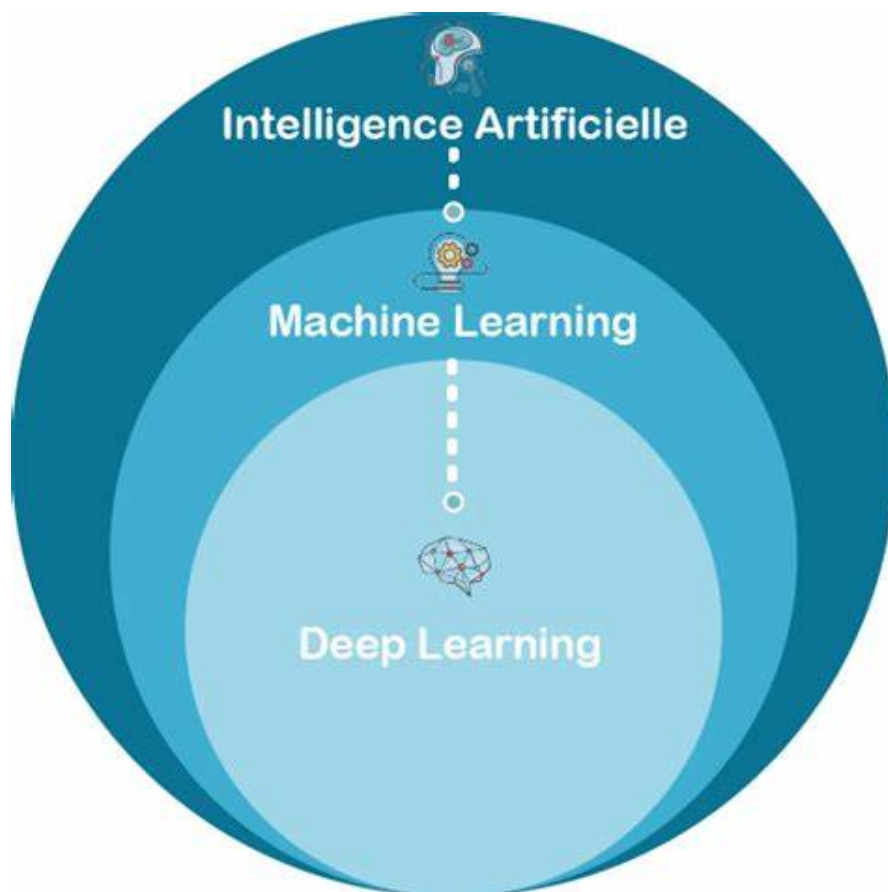


FIGURE II.1 – L'intelligence artificielle [7]

II.3 Apprentissage automatique

L'apprentissage automatique vise à faire exécuter à un programme (ou une machine) des tâches pour lesquelles on ne connaît pas de méthode algorithmique efficace, en lui faisant apprendre ce que l'on attend de lui sur plusieurs exemples, puis en le laissant exécuter la généralisation nécessaire à la résolution de problèmes similaires [46].

L'apprentissage automatique est une branche de l'IA. Il résout des problèmes tels que la prédiction et la classification, grâce à l'analyse des données et à la production d'algorithmes [47].

L'apprentissage automatique s'appuie sur des méthodes statistiques pour donner aux ordinateurs la capacité "d'apprendre" à partir de données, c'est-à-dire d'améliorer leurs performances dans la résolution de tâches sans être explicitement programmés pour chacune [8].

Fait référence à un réseau neuronal composé de trois couches, incluant les entrées et la sortie [48]. Comme le montre la Figure suivante II.2

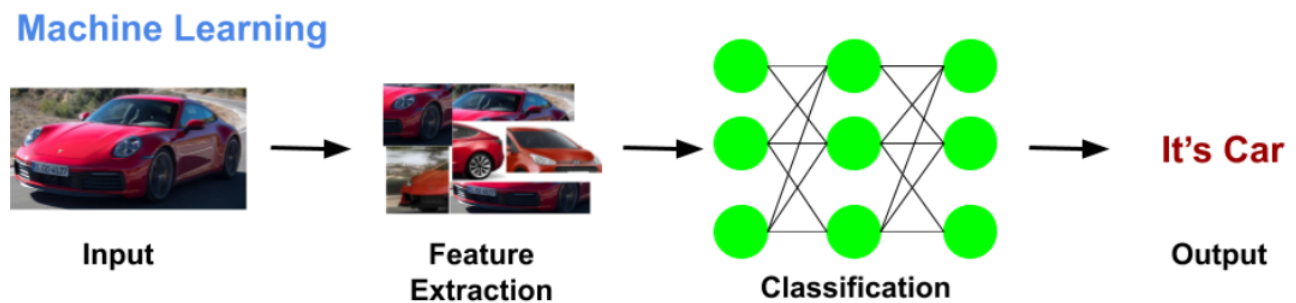


FIGURE II.2 – L'apprentissage automatique[8]

II.3.1 Réseau de neurones

Un réseau de neurones est un modèle informatique dont la structure en couches est similaire à la structure en réseau des neurones du cerveau, avec des couches de nœuds connectés. Un réseau de neurones peut apprendre à partir de données ; il peut ainsi être entraîné à reconnaître des tendances, classer des données et prévoir des événements à venir [9].

De manière générale, les réseaux de neurones sont un ensemble de neurones, organisés en plusieurs couches et reliés entre eux. La première couche est appelée input layer, la dernière output layer et les couches qui se situent entre les deux sont appelées hidden layers. Il peut y en avoir un très grand nombre, selon les domaines d'applications [49]. Comme le montre la Figure suivante II.3

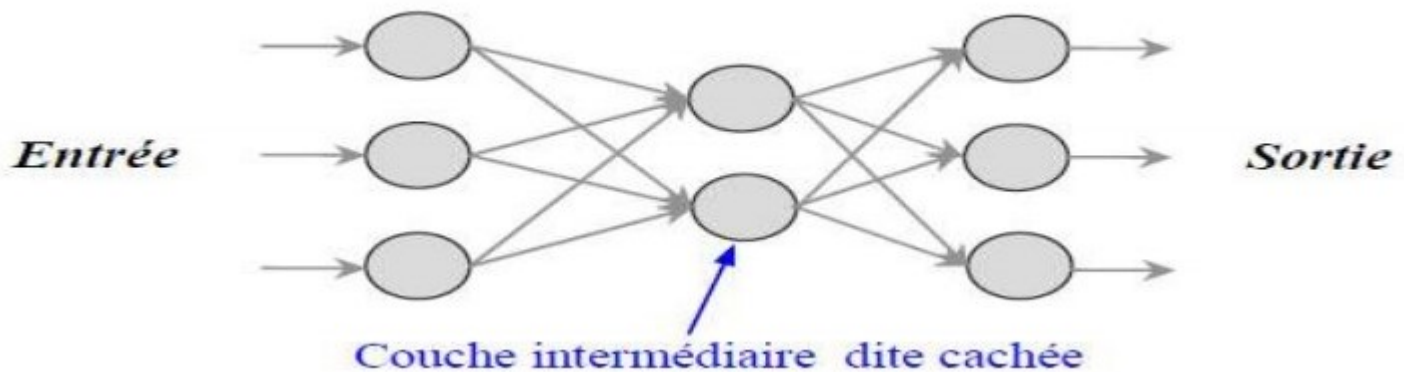


FIGURE II.3 – Architectures des réseaux de neurones [9]

II.3.2 Types d'Apprentissage automatique

II.3.2.1 Apprentissage non supervisé

Le but de l'algorithme ici est de regrouper ou d'étiqueter les données en extrayant des relations, des modèles de données, l'apprentissage non supervisé fonctionne uniquement sur les données d'entrée sans sorties ni cible [50].

II.3.2.2 Apprentissage supervisé

il se caractérise par l'utilisation de jeux de données étiquetés qui entraînent des algorithmes permettant de classer des données ou de prédire des résultats avec précision [10]. il existe deux principaux types de l'apprentissage supervisé :

La régression vise à prédire une valeur (étiquetée) en sortie. La sortie peut être de n'importe quel type : chaîne, numérique.

La classification inclut dans l'apprentissage supervisé qui consiste à prédire une étiquette de classe. Comme le montre la Figure suivante II.4 [50]

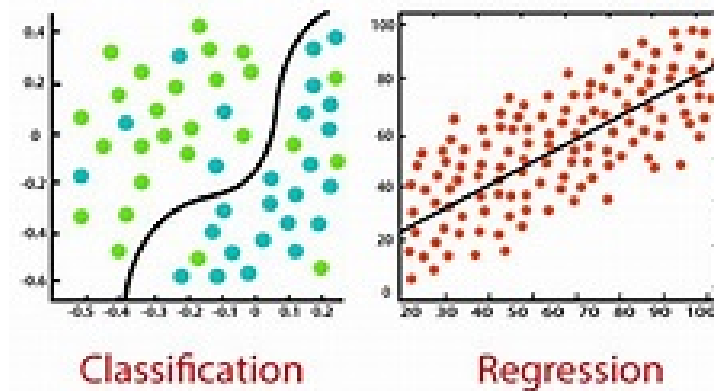


FIGURE II.4 – Classification et Régression [10]

II.3.2.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé est capable d'apprendre à l'aide de quelques exemples de données étiquetées et d'apprendre le reste de l'ensemble de données contrôle externe [10].

II.3.2.4 Apprentissage par renforcement

L'apprentissage par renforcement est une technique d'apprentissage avancée. Ici, nous avons un agent examinant un environnement inconnu, effectuant des actions aléatoires pour déterminer quelle action est bonne et laquelle est mauvaise en obtenant des récompenses ou des pénalités. L'agent vise à optimiser les récompenses à long terme. montre un exemple d'apprentissage par renforcement dans l'apprentissage automatique[10].

II.4 Apprentissage en profond

L'apprentissage en profondeur [11] est un sous-ensemble de l'apprentissage automatique, telle que le terme "en profondeur" fait référence à un réseau neuronal composé de plus de trois couches, incluant les entrées et la sortie.

Ces réseaux de neurones tentent de simuler le comportement du cerveau humain, bien que loin de correspondre à ses capacités, lui permettant«d'apprendre» à partir de grandes quantités de données. Comme le montre la Figure suivantII.5

II.4.1 Types d'apprentissage en profondeur

L'apprentissage en profondeur peuvent être classés en différents types :

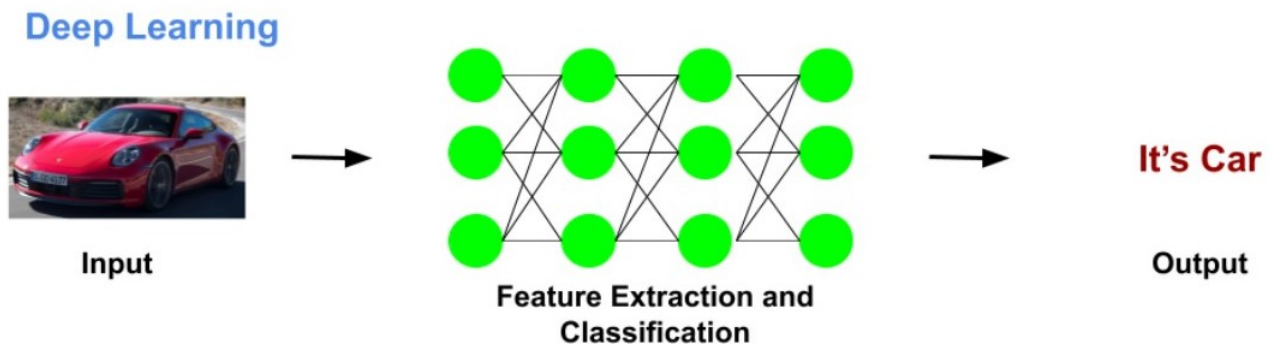


FIGURE II.5 – L'apprentissage en profondeur[11]

II.4.1.1 Un réseau neuronal convolutif(CNN)

Réseau neuronal convolutif(CNN) est un réseau neuronal d'apprentissage en profondeur[51] conçu pour traiter des tableaux structurés de données telles que des images. Les réseaux de neurones convolutifs sont largement utilisés en vision par ordinateur et sont devenus l'état de l'art pour de nombreuses applications visuelles telles que la classification d'images, et ont également rencontré le succès dans le traitement du langage naturel pour la classification de texte.

II.4.1.2 Réseau neuronal récurrent (RNN)

Les réseaux de neurones récurrents RNN [52], sont un type de réseau de neurones largement utilisé dans le domaine de l'apprentissage en profondeur (Deep Learning) . Les RNN utilisent les sorties précédentes comme entrées supplémentaires et sont parfaitement adaptés au traitement de données séquentielles.

II.4.1.3 Réseau Antagonistes Génératifs (GAN)

GAN est une technique de deep Learning. [53] Elle repose sur la mise en compétition de deux réseaux au sein d'un framework. Ces deux réseaux sont appelés " générateur " et " discriminateur". Le générateur est un type de réseau neuronal convolutif dont le rôle est de créer de nouvelles instances d'un objet. De son côté, le discriminateur est un réseau neuronal » déconvolutif » qui détermine l'authenticité de l'objet ou s'il fait ou non partie d'un ensemble de données.

II.4.2 Domaines d'applications de l'apprentissage en profondeur

L'apprentissage en profondeur peut être utilisé pour une variété d'applications, [54] notamment :

II.4.2.1 Traitement d'images

L'apprentissage en profondeur peut être utilisé pour identifier des objets dans des images, tels que des visages, des voitures et des animaux. Cette technologie est utilisée dans une variété d'applications, telles que les logiciels de reconnaissance faciale, les voitures autonomes et la recherche d'images.

II.4.2.2 Traitement du langage naturel

L'apprentissage en profondeur peut être utilisé pour comprendre le langage humain. Cette technologie est utilisée dans une variété d'applications, telles que la reconnaissance vocale, la traduction automatique et l'analyse de texte.

II.4.2.3 Reconnaissance de la parole

L'apprentissage en profondeur peut être utilisé pour transcrire la langue parlée en texte. Cette technologie est utilisée dans une variété d'applications, telles que les assistants à commande vocale, les logiciels de dictée et le sous-titrage codé.

II.4.2.4 Traduction automatique

L'apprentissage en profondeur peut être utilisé pour traduire un texte d'une langue à une autre. Cette technologie est utilisée dans une variété d'applications, telles que les outils de traduction en ligne, les sites Web multilingues et les affaires internationales.

II.4.2.5 Diagnostic médical

L'apprentissage en profondeur peut être utilisé pour diagnostiquer des maladies à partir d'images médicales, telles que des radiographies, des IRM et des tomodensitogrammes. Cette technologie en est encore à ses premiers stades de développement, mais elle a le potentiel de révolutionner la façon dont les maladies sont diagnostiquées et traitées.

II.4.2.6 Négociation financière

L'apprentissage en profondeur peut être utilisé pour prédire les cours des actions et d'autres marchés financiers. Cette technologie est utilisée par les fonds spéculatifs, les banques d'investissement et d'autres institutions financières pour prendre des décisions commerciales.

Ce ne sont là que quelques-unes des nombreuses applications de l'apprentissage en profondeur.

II.5 Vision par ordinateur

II.5.1 Définition

La vision par ordinateur exploite l'intelligence artificielle (IA) pour permettre aux ordinateurs d'obtenir des données significatives à partir d'entrées visuelles telles que des photos et des vidéos. Les connaissances acquises grâce à la vision par ordinateur sont ensuite utilisées pour prendre des mesures automatisées. Tout comme l'IA donne aux ordinateurs la capacité de "penser", la vision par ordinateur leur permet de "voir" [12]. Comme le montre la Figure suivante II.6

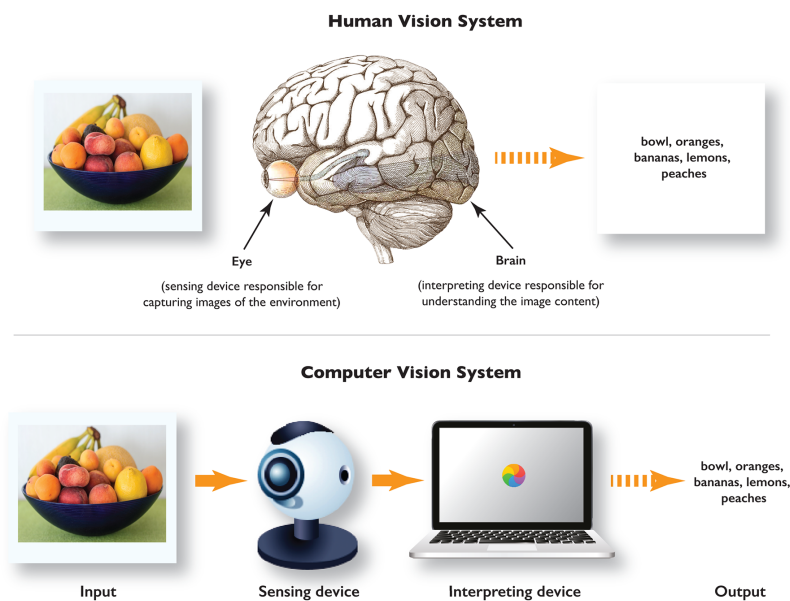


FIGURE II.6 – Human Vision vs. Computer Vision [12]

II.5.2 Tâches importantes en vision par ordinateur

Voici trois tâches importantes en vision par ordinateur :

II.5.2.1 Classification d'images

La classification est une tâche d'apprentissage automatique permettant de déterminer quels objets se trouvent dans une image ou une vidéo ,dans le but de découvrir quelles classes (objets) sont présentes . [13]Comme le montre la Figure suivantII.7

Classification



CAT

FIGURE II.7 – Classification [13]

II.5.2.2 Détection d'objets

La détection d'objets est une tâche qui consiste à identifier la présence d'un objet spécifique dans une image et à localiser sa position. Elle peut utiliser la classification d'images pour identifier la classe d'objet recherchée, puis détec-

ter et localiser les instances de cet objet dans l'image.[14] Comme le montre la Figure suivant II.8

Object Detection

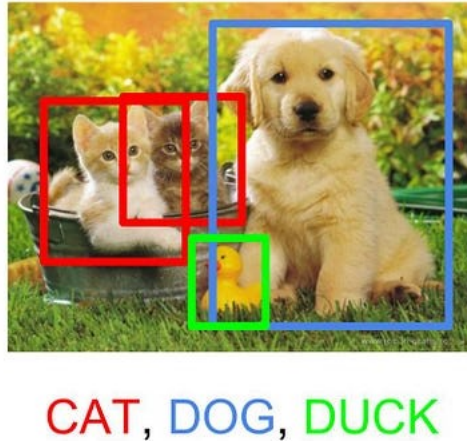


FIGURE II.8 – Detection des Object [14]

II.5.2.3 Segmentation d'images

Segmentation d'images est la segmentation d'image, une technique qui consiste à diviser une image en plusieurs sections en attribuant différentes couleurs ou tons à différentes zones. Cela permet à chaque zone d'être identifiée indépendamment, ce qui facilite la reconnaissance et l'analyse de chaque segment par les ordinateurs. Par exemple, une scène de rue peut être segmentée en différentes sections telles que la route, le trottoir et les bâtiments, permettant une reconnaissance distincte de chaque zone. De plus, la segmentation d'image peut être utilisée pour identifier des objets dans une image en attribuant à chaque objet une couleur ou un ton unique, permettant aux ordinateurs de faire la distinction entre des objets individuels dans une scène et facilitant une reconnais-

sance et une analyse précises.[15] Comme le montre la Figure suivante II.9

Instance Segmentation



CAT, DOG, DUCK

FIGURE II.9 – Segmentation d’images [15]

En résumé, la classification d'images, la détection d'objets et la segmentation d'images sont trois tâches essentielles en vision par ordinateur. La classification permet de classer les images en catégories prédéfinies, la détection d'objets permet d'identifier et de localiser des objets spécifiques dans une image, et la segmentation d'images permet de diviser une image en régions distinctes pour une analyse plus précise.

II.5.3 Domaines d'applications de vision par ordinateur

II.5.3.1 Agriculture

La vision par ordinateur [55] a de nombreuses applications existantes et à venir dans l'agriculture, notamment la surveillance des cultures par drone, la pulvérisation automatique de pesticides, le suivi du rendement et le tri et la classification intelligents des cultures. Grâce à la technologie de vision par ordinateur dans les enregistrements météorologiques, les données forestières.

II.5.3.2 Imagerie médicale

Il y a une augmentation notable de l'application des techniques de vision par ordinateur pour le traitement de l'imagerie médicale. Ceci est particulièrement répandu en pathologie, radiologie et ophtalmologie. La reconnaissance visuelle des formes, alimentée par la vision par ordinateur, permet à des produits avancés, tels que Microsoft InnerEye, de fournir des diagnostics rapides et précis dans un nombre croissant de spécialités médicales

II.5.3.3 Éducation

Un exemple courant de ceci est les webcams compatibles avec la vision par ordinateur, qui sont utilisées pour surveiller les étudiants pendant les examens. Cela facilite la détection des pratiques déloyales grâce à l'analyse des mouvements oculaires et du comportement corporel.

II.5.3.4 Transport

Enfin, les systèmes de vision par ordinateur sont de plus en plus appliqués pour accroître l'efficacité des transports. Par exemple, la vision par ordinateur est utilisée pour détecter les contrevenants aux feux de circulation, permettant ainsi aux forces de l'ordre de minimiser les comportements dangereux sur la route.

II.6 Conclusion

Dans ce chapitre, nous avons abordé les concepts de base du deep learning et certaines de ses applications, ainsi que la vision par ordinateur et vu certaines de ses tâches et applications, notamment dans le domaine de la santé. Dans le chapitre suivant, nous nous pencherons sur les personnes handicapées, en particulier les aveugles et les malvoyants, pour les aider et trouver des solutions aux défis auxquels elles sont confrontées, et nous verrons également les solutions techniques apportées par les chercheurs à travers l'Internet des objets et par vision par l'ordinateur et Apprentissage en profondeur.

Chapitre III

Contexte Générale

III.1 Introduction

Les personnes handicapées font partie des groupes sociaux qui nécessitent des soins et une attention particulière car elles souffrent de problèmes de santé qui peuvent être physiques ou mentaux, entraînant leur incapacité totale ou partielle à mener à bien leurs activités quotidiennes. Dans ce chapitre , nous allons parler de la déficience visuelle, ainsi que des difficultés que rencontrent les personnes aveugles dans leur vie quotidienne, on va voir aussi des aides techniques disponibles sur le marché et tout ce qui s’y rapporte.

III.2 Non-voyant

Le non-voyant est une personne qui souffre d’un trouble de la vue, qui ne voit plus ou qui voit particulièrement mal. C’est un handicap qui explique une complication dans le champ visuel qui influe son contact avec autrui. Dans ces conditions précises, les non-voyants cherchent à développer leurs autres capacités sensorielles comme : le toucher, l’ouïe, et l’odorat [56].

III.3 Déficience visuelle

III.3.1 Définitions

La déficience visuelle désigne [57] les troubles liés à la fonction visuelle, qui persistent après traitements (thérapeutiques, médicaux, chirurgicaux,..). Elle est définie à l'aide de deux critères que sont l'état du champ visuel (étendue de l'espace qu'un œil peut saisir) et la mesure de l'acuité visuelle (aptitude d'un œil à apprécier les détails).

A partir de là, on distingue deux catégories de déficience visuelle :

Les aveugles, atteints de cécité, dont l'acuité visuelle corrigée est inférieure ou égale à 1/20.

Les malvoyants, atteints d'amblyopie, dont l'acuité visuelle après correction du meilleur œil est comprise entre 4/10 et 1/10.

La déficience est évaluée selon cinq degrés : cécité absolue, cécité sévère, déficience visuelle profonde, déficience sévère, déficience moyenne.

III.3.2 Types de déficience visuelle

Il existe deux type de déficience visuelle[58] :

- **Malvoyance** : est définie par une acuité visuelle située entre 1/20e et 3/10e ou un champ visuel inférieur ou égal à 20°. L'acuité visuelle prise en compte s'entend après correction (par des lunettes ou lentilles de contact) et au meilleur œil.
- **Cécité** : est l'absence de vision d'un ou des deux yeux. Les personnes souffrant de cécité sont donc privées du sens de la vue, partiellement (cécité

partielle) ou totalement (cécité totale), de façon invalidante.

III.3.3 Degrés de déficience visuelle

La déficience est évaluée selon cinq degrés[59] : La cécité absolue, cécité sévère, déficience visuelle profonde, déficience sévère, déficience moyenne.

- déficience visuelle moyenne : qui se caractérise par un champ visuel supérieur à 20 degrés, et une acuité visuelle comprise entre 1/10 et 3/10.
- La déficience visuelle sévère : qui se caractérise par une acuité visuelle entre 1/20 et 1/10. La personne est alors capable de compter les doigts à 3 mètres.
- La déficience visuelle profonde, qui se caractérise par une acuité visuelle entre 1/50 et 1/20. La personne peut compter les doigts à 1 mètre.
- La cécité sévère, qui se caractérise par une acuité inférieure à 1/50, un champ visuel réduit, et la perception de la lumière uniquement .
- Une cécité totale, qui se caractérise par la non perception y compris de la lumière .

III.3.4 Formes de malvoyance

Il existe également quatre principaux types de déficience visuelle qui sont les suivants[59] :

- **Atteinte de la vision centrale** :La partie centrale de la rétine concentre les cellules responsables de l'acuité visuelle.Elle permet la perception des formes et des couleurs, mais surtout des détails.

Les personnes ayant une atteinte de la vision centrale éprouvent des dif-

ficultés pour lire, écrire et réaliser des travaux de précision. La reconnaissance des visages devient également délicate. Elles conservent en revanche la perception des espaces et du mouvement, ce qui leur permet généralement de se déplacer sans aide.

- **Atteinte de la vision périphérique** : Chez les personnes ayant une atteinte de la vision périphérique, le champ visuel se rétrécit. On parle alors d'une vision tubulaire, en tunnel ou en canon de fusil. L'acuité visuelle centrale est préservée mais la vision se limite à ce qui se trouve autour du point de fixation.

Ce type de malvoyance est souvent déstabilisant pour l'entourage. En effet, les personnes dont la vision périphérique est détériorée peuvent être capables de lire les petits caractères d'un journal mais se heurtent l'instant d'après contre un poteau, aussi coloré soit-il. Elles n'ont pas de perception globale de leur environnement et sont incapables de suivre un objet en mouvement. Ce type de malvoyance est très handicapant pour les déplacements.

- **Vision floue** : Ce type de malvoyance s'apparente au fait de regarder au travers d'une vitre dépolie. La luminosité se diffuse et rend les contours des objets imprécis. Les personnes ayant une vision floue ne perçoivent que des formes vagues et des masses, ce qui rend les contrastes, les reliefs et les distances difficiles à apprécier. La lumière, surtout quand elle est vive, peut devenir insupportable.

Ces personnes sont particulièrement gênées pour la lecture, l'écriture et les travaux de précision. Mais elles éprouvent également de grandes dif-

ficultés lors de leurs déplacements, car il leur est impossible d'évaluer le danger et de se repérer dans l'espace.

- **Atteintes visuelles d'origine cérébrale** : Les troubles visuels causés par un traumatisme ou une lésion du cerveau sont variés et souvent associés à d'autres perturbations comme les troubles de l'attention, de la mémoire ou du comportement.

Dans la plupart des troubles visuels d'origine cérébrale, ce n'est pas la fonction visuelle proprement dite qui est altérée, mais la capacité du cerveau à analyser l'information.

III.4 Etude de l'environnement

Pour obtenir la solution optimale[57]et efficace pour aider le handicapé à faire leur activités quotidiennes,pour cela nous analyserons en détail l'entourage des aveugles, ce dernier nous permettant de distinguer.

III.4.1 Types d'obstacles

Il existe trois types les obstacles [57] :

- **Obstacles en hauteur** : Les obstacles en hauteur sont très dangereux car ils sont beaucoup plus difficiles à détecter que les obstacles au sol.

La canne longue, ou canne d'action utilisée par les aveugles ou très malvoyants, permet de détecter les obstructions en dessous de la hauteur ombilicale[60].

- **Obstacles en bas** : Ce type des obstacles et le plus trouvé dans l'environnement des personnes non-voyant, en peut distinguer deux types des obstacles en bas :

1. Obstacle sous-terrain comme les faussé et les fin des territoires, les escaliers de descendre, etc.
 2. Obstacles sur sol comme les coffres de la poste, les escaliers de monte, les assemble d'eau, etc.
- **Obstacles dynamiques :** Lorsqu'un aveugle se déplace d'une rue à l'autre, le mouvement des personnes peut causer perturbation, pour cela, il est nécessaire de signaler les mouvements à proximité.

III.4.2 Défis des aveugles

Les aveugle et malvoyants[57]ont des difficultés à se déplacer en ville pour les transports en commun et repérage préalable du parcours. En plus, ils ont des difficultés au quotidien :la lecture des textes,reconnaissance d'objets etc. La chose la plus importante est les interactions sociales car ils ne peuvent pas reconnaître les personnes ce qui leurs faisant se sentir inférieurs et marginalisés.

III.5 Travaux connexes

Dans cette section, nous passons en revue les études menées dans le domaine des aides techniques intelligentes destinées aux personnes déficientes visuelles :

Dans cette papier [61], les auteurs présentes la conception et la mise en œuvre d'une canne basée sur un capteur à ultrasons pour les personnes malvoyantes. Un module de capteur à ultrasons, HC-SR04 est utilisé pour la détection d'obstacles sur le chemin de la personne aveugle et un buzzer est utilisé pour rendre la personne alerte. Le système proposé est mis en œuvre à l'aide du microcon-

trôleur PIC 16F877A.

Les personnes aveugles peuvent utiliser cette canne pour naviguer en toute sécurité. Il peut détecter un obstacle dans une plage de distance de 5 à 35 cm.

les auteurs dans cette travaille [62], ils présentent des lunettes intelligentes à partir d'un nouveau verre intelligent qui a la capacité d'extraire et de reconnaître le texte capturé à partir d'une image et de le convertir en parole. Cette solution a été largement testée dans un cadre universitaire pour aider les étudiants malvoyants. La nouveauté de la solution mise en œuvre est qu'elle fournit les fonctions de vision par ordinateur nécessaires à la reconnaissance d'image/-texte, qui sont économiques, compactes, précises et utilisent des outils logiciels open source.

Dans [63], les auteurs proposé des lunettes intelligentes en utilisant une caméra stéréo, tel que :

La détection de type d'obstacle en utilisant le modèle YOLO v3 L'informe du niveau dangereux grâce au moteur de vibrations et en utilisant le camera. De plus, l'emplacement de l'utilisateur est informé à l'aide de lumières LED pour éviter les accidents de nuit.

les auteurs dans [64], ils ont faire et réalisation d'une canne blanche intelligente qui utilise une carte Arduino La reconnaissance des objets par et le capteur ultrason, se fait par des bips sonneurs et des vibrations . Le système est constitué aussi d'un indicateur de sécurité, fonction avec l'absence de lumière détectée par le capteur LDR.

Dans [65], les chercheurs ont proposé un système de verre intelligent pour les aveugles en utilisant les deux technologie 5G/Wifi pour aider de la manière

suivante : Améliorer le contraste de l'image dans des conditions de faible luminosité à l'aide d'un réseau d'exposition à deux fentes . Guidez les utilisateurs par la voix à l'aide d'un modèle de détection d'objet Accédez aux informations visuelles à l'aide de l'extraction d'objets saillants la reconnaissance de texte et d'un écran tactile actualisable.

Le tableau suivant montre la comparaison des travaux connexes :

TABLE III.1: Comparaison des travaux connexes

Type de produit	Matériel	Technique(Méthode)	Connectivité
Bâton aveugle intelligent[61]	Capteur ultrasonore Microcontrôleur PIC 16F877A Module GPS Caméras stéréo Moteur à vibrations	Organigramme d'algorithme Programmation du microcontrôleur en langage C	Bluetooth

Continué sur la page suivante

Table III.1 – Suite de la page précédente

Type de produit	Matériel	Technique (Méthode)	Connectivité
Lunettes intelligentes [62]	Raspberry Pi 3 - Modèle B+ Caméra numérique Capteur ultrasonore Capteur RFID Écouteurs	Reconnaissance de texte : Outils OCR - Tesseract et EAST API de synthèse vocale Google Text to Speech (gTTS) Détection d'objets : OpenCV (BSD)	5G/WiFi

Continué sur la page suivante

Table III.1 – Suite de la page précédente

Type de produit	Matériel	Technique (Méthode)	Connectivité
Lunettes intelligentes [65]	Lunettes avec caméra Capteur GPS Capteur ultrasonore	Détection d'objets : Encodeur-décodeur basé sur un transformateur COCO 2017 Reconnaissance de texte : CNN et Tesseract OCR ExDark Génération de graphiques tactiles LOL	5G/WiFi
Lunettes intelligentes [63]	Lunettes avec caméra stéréo Moteur de vibrations sonores LED	Détection d'objets : YOLO v3 COCO 2017 Alertes pour les niveaux dangereux (Mis en œuvre dans la caméra stéréo)	4G/WiFi

Continué sur la page suivante

Table III.1 – Suite de la page précédente

Type de produit	Matériel	Technique (Méthode)	Connectivité
Canne blanche intelligente [64]	Canne blanche avec buzzer Carte Arduino Capteur ultrasonore Capteur LDR de lumière	Reconnaissance d'objets (détectée par le capteur ultrasonore en fonction de la distance mesurée) Détection des obstacles par absence de lumière (détectée par le capteur LDR et le capteur ultrasonore)	/

III.6 Conclusion

Dans ce chapitre, nous avons présenté quelques notions de déficience visuelle, en commençant par sa définition, ses types, son degré. Nous avons étudié aussi l'environnement des aveugles tels que les types d'obstacles et les défis des aveugles. Enfin, ce chapitre traite de quelques études et solutions de chercheurs dans ce domaine.

Chapitre IV

Conception du Système

IV.1 Introduction

Dans ce chapitre, nous parlerons de ce projet, qui est des lunettes intelligentes, et donnerons une définition complète des lunettes et de leurs fonctions. Nous verrons également les architectures générales et détaillées. Et aussi la méthode qui a été utilisée dans notre projet.

IV.2 Lunettes intelligentes

Les lunettes intelligentes aident les personnes aveugles à entendre le monde qui les entoure. Dans notre solution, il s'agit d'un appareil mobile facile à utiliser composé d'une caméra placée sur la monture des lunettes, qui remplace les yeux des aveugles pour surveiller et détecter les obstacles autour d'eux. pour aider les aveugles dans leurs déplacements et les informer sur l'environnement. Connecté à des écouteurs qui lui permettent également de lire des documents, des menus de restaurants et des panneaux de signalisation.

IV.3 Architecture général

Dans cette partie nous allons présenter l'architecture générale : Comme le montre la Figure suivante [IV.1](#)

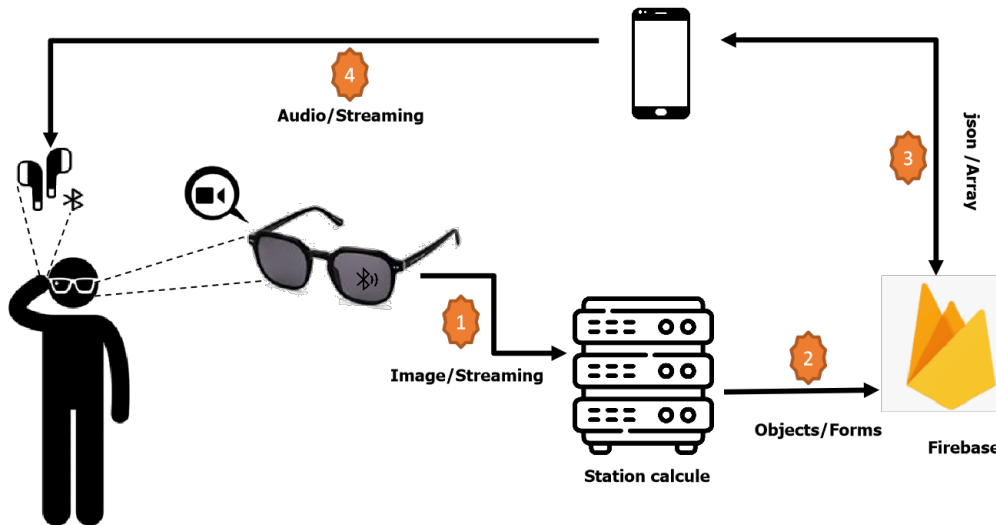


FIGURE IV.1 – Architecture général

- Etape 1 : La personne aveugle porte des lunettes intelligentes qui prennent des photos en temps réel et les transmettent au station de calcule via le gateway .
- Etape 2 et 3 : Le station de calcule traiter et analyser les images reçues et fournira une description sur l'entourage de l'aveugle sous forme objets et sauvegarder dans la base de donnees .
- Etape4 : Dans cette dernière étape, l'Application Mobile connecté avec des écouteurs via bluetooth va convertir les données récupérées de la base de données en audio, lisent par des écouteurs.

IV.4 Fonctionnement du Système

La fonction principale du système proposé peut être résumée comme suit :

- Prendre l'image.
- Détection des objets en temps réel.
- Extraire le texte de l'image.
- Convertir du texte en voix .

IV.5 Conception Détaillée

Dans cette partie ,nous allons presenter les 3 partie de notre architecteure(Lunettes,Station de calcule , Application mobile) :Comme le montre la Figure suivante [IV.2](#)

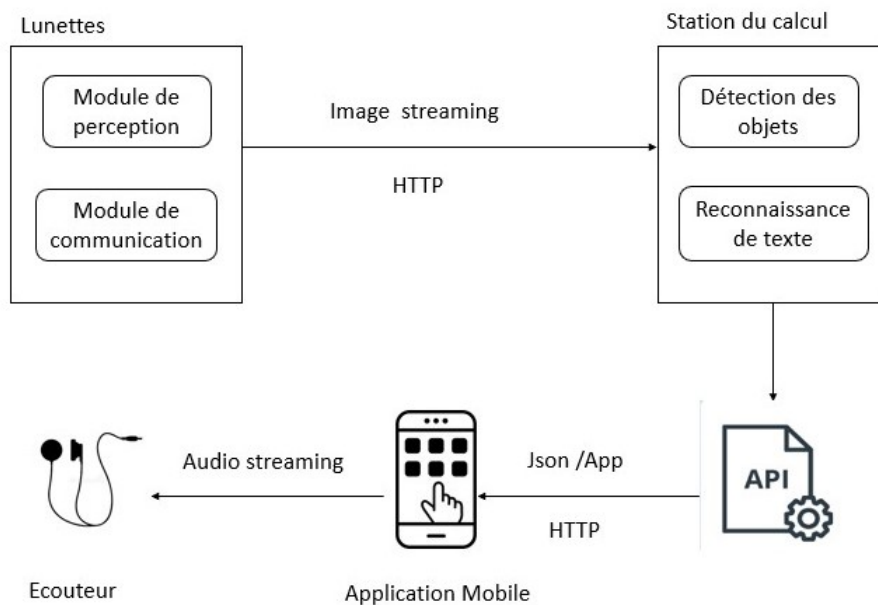


FIGURE IV.2 – Architecture Détaillée

IV.5.1 Lunettes

Les lunettes sont une composante essentielle de notre projet, composé de deux modules principaux :

IV.5.1.1 Module de perception

Les lunettes ont remplacé les yeux des personnes aveugles pour percevoir leur environnement, au moyen d'un capteur d'image.

- **Capteur d'image** : La caméra est un composant qui offre le champ de vision d'une personne aveugle à la place d'un œil qui envoie en permanence des images à un serveur pour obtenir une description de son environnement.

IV.5.1.2 Module de communication

Les lunettes communiquent avec la station de calcul via un réseau sans fil pour transmettre les images captées à cette dernière .

IV.5.2 Station de calcul

La station de calcul traite les images envoyées et captées par la caméra et elles sont traitées et analysées par les algorithmes ci-dessous pour le restituer à son porteur sous forme de description.

IV.5.2.1 Détection des objets

La personne aveugle se déplace dans l'environnement pour accomplir ses tâches quotidiennes et rencontre des obstacles qui doivent être détectés à l'aide d'un algorithme de détection d'objets par le deep learning

1. Classements des algorithmes de détection d'objets

Les algorithmes de détection d'objets sont généralement classés en deux catégories en fonction du nombre de fois que la même image d'entrée est passée à travers un réseau[16]. Comme le montre la Figure suivante IV.3

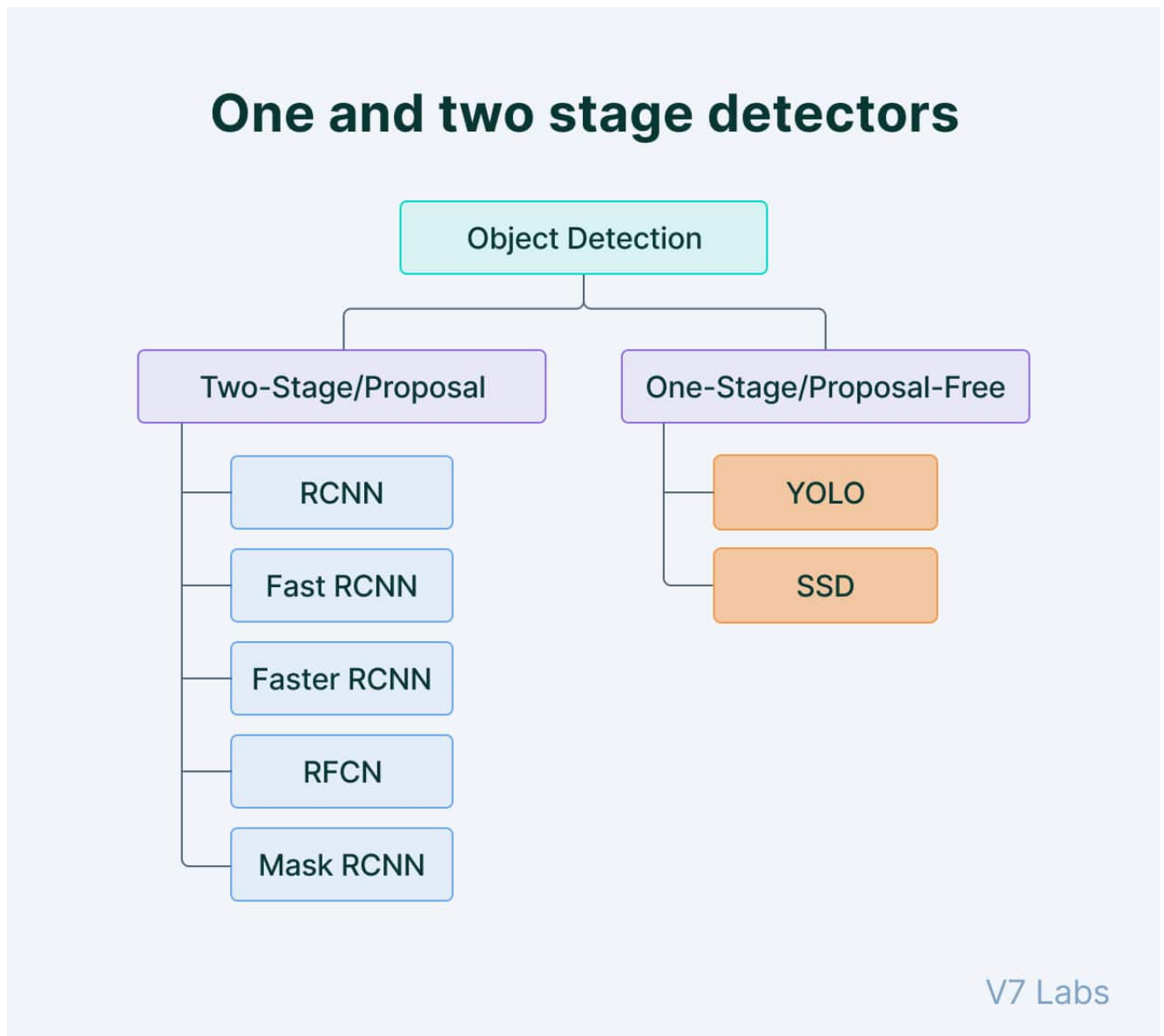


FIGURE IV.3 – Classes d'algorithmes de détection d'objets [16]

(a) Détection d'objets en une seule fois

utilise un seul passage de l'image d'entrée pour faire des prédictions sur la présence et l'emplacement des objets dans l'image. Il traite une image entière en un seul passage, ce qui les rend efficaces en termes de calcul.

Cependant, la détection d'objets à un seul coup est généralement moins précise que les autres méthodes, et elle est moins efficace pour détecter de petits objets. De tels algorithmes peuvent être utilisés pour détecter des objets en temps réel dans des environnements à ressources limitées.

(b) **Détection d'objet à deux coups**

La détection d'objets en deux temps utilise deux passages de l'image d'entrée pour faire des prédictions sur la présence et l'emplacement des objets. La première passe est utilisée pour générer un ensemble de propositions ou d'emplacements d'objets potentiels, et la seconde passe est utilisée pour affiner ces propositions et faire des prédictions finales. Cette approche est plus précise que la détection d'objets à un seul coup, mais elle est également plus coûteuse en calculs.

le choix entre la détection d'objets à un coup et à deux coups dépend des exigences et des contraintes spécifiques de l'application.

En règle générale, la détection d'objets à un seul coup est mieux adaptée aux applications en temps réel, tandis que la détection d'objets à deux coups est meilleure pour les applications où la précision est plus importante .

2. YOLO

Dans notre projet, nous avons choisi Yolo pour sa fonctionnalité en temps réel.

YOLO (You Only Look Once), un modèle populaire de détection d'objets et de segmentation d'images, a été développé par Joseph Redmon et Ali Farhadi de l'Université de Washington. Lancé en 2015, YOLO a rapidement gagné en popularité pour sa grande vitesse et sa précision.

3. Versions de YOLO

Il y a eu 8 versions du modèle jusqu'à présent [66] , chaque nouvelle version améliorant la précédente en termes de vitesse et de précision.

YOLOv2, publié en 2016, a amélioré le modèle d'origine .

YOLOv3, introduit par Redmon et al , en 2018.

YOLOv4 , publié par Bochkovskiy et al, en 2020.

YOLOv5 et YOLOv6, publié par Ultralytics, en 2022.

YOLOv7 , publié en 2022.

Dans notre projet, nous avons utilisé la dernière version Yolo v8.

YOLOv8

YOLOv8 est le dernier modèle YOLO à la pointe de la technologie qui peut être utilisé pour la détection d'objets, la classification d'images et les tâches de segmentation d'instance. YOLOv8 a été développé par Ultralytics, qui a également créé le modèle influent et définissant l'industrie YOLOv5. YOLOv8 inclut de nombreuses modifications et améliorations de l'architecture et de l'expérience des développeurs par rapport à YOLOv5[67].

YOLOv8 ,la dernière version du célèbre modèle de détection d'objets et de segmentation d'images en temps réel. YOLOv8 s'appuie sur des avan-

cées de pointe en matière d'apprentissage en profondeur et de vision par ordinateur, offrant des performances inégalées en termes de vitesse et de précision. Sa conception simplifiée le rend adapté à diverses applications et facilement adaptable à différentes plates-formes matérielles, des appareils de périphérie aux API cloud[68] .

Architecture de YOLOv8

Architecture de YOLOv8 s'appuie sur les versions précédentes des algorithmes YOLO YOLOv8 utilise un réseau neuronal convolutif qui peut être divisé en deux parties principales : la colonne vertébrale et la tête.

Une version modifiée de l'architecture CSPDarknet53 constitue l'épine dorsale de YOLOv8. Cette architecture se compose de 53 couches convolutionnelles et utilise des connexions partielles entre les étages pour améliorer le flux d'informations entre les différentes couches.

La tête de YOLOv8 se compose de plusieurs couches convolutionnelles suivies d'une série de couches entièrement connectées.

Ces couches sont chargées de prédire les boîtes englobantes, les scores d'objectivité et les probabilités de classe pour les objets détectés dans une image ,Comme le montre l'Architecture suivante [IV.4](#).

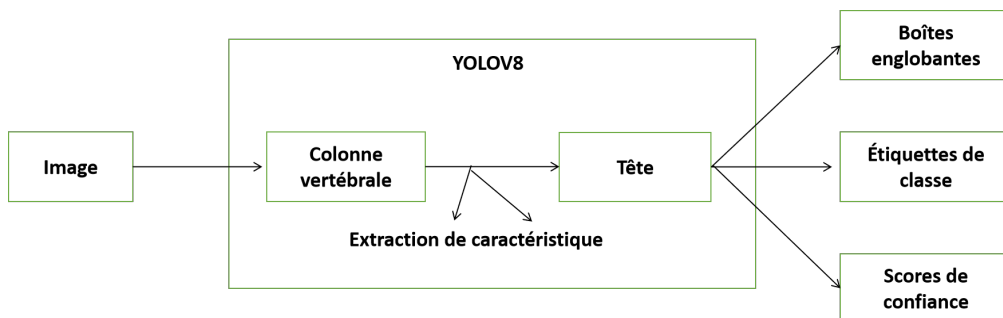


FIGURE IV.4 – Architecture de YOLOv8

Nous allons maintenant voir l'algorithme utilisé pour la détection d'objets à l'aide du célèbre modèle YOLOV8, donc dans cette partie, nous allons montrer l'algorithme de détection d'objets en temps réel. Comme le montre l'Algorithme suivant [IV.5](#)

Algorithm 1: Algorithme de détection des objets

```

Chemin ← source;
// charger le modèle
YoloModel ← chargerModelYolov8();
while True do
    // lire l'image
    image ← VideoCapture(Chemin);
    // détection des objets
    objects ← Model.detect(image);
end

```

FIGURE IV.5 – Algorithme de détection d'objets

Fonctionnement d'un algorithme 1 : On utilisant Yolov8 pour détecter des objets est une méthode aux performances inégalées en termes de vitesse et de précision en temps réel basée sur l'apprentissage en profondeur et la vision par ordinateur, nous aurons donc besoin d'accéder à la caméra des lunettes pour détecter l'objet dans chaque image. Dans la première étape, nous devons lire la vidéo à travers la caméra des lunettes en temps réel de la manière suivante : VideoCapture (Path) et la convertir en images(trames) et la prendre en entrée du réseau neuronal de détection, une fois les objets et les obstacles détectés. nous dessinons le cadre si la valeur de confiance ≥ 70 et ajoutons également le nom de l'objet.

IV.5.2.2 Reconnaissance de texte

La lecture et le traitement de textes sont très difficiles pour les personnes malvoyance, en particulier les aveugles, nous avons donc utilisé une technologie qui leur permet d'extraire des textes à partir d'images et de les reconnaître.

Dans notre projet, nous avons utilisé la technique ocr pour extraire des textes à partir d'images.

- **OCR (Optical Character Recognition)** est utilisé pour la conversion Text [62]ou manuscrit dans du texte codé par machine.Certains moteurs logiciels OCR tentent de reconnaître Tout texte dans des images telles que Tesseract . Dans ce projet, Tesseract est utilisé car c'est le meilleur open source Moteurs OCR. Le processus d'OCR se compose de plusieurs étapes, comme indiqué dans la Figure :

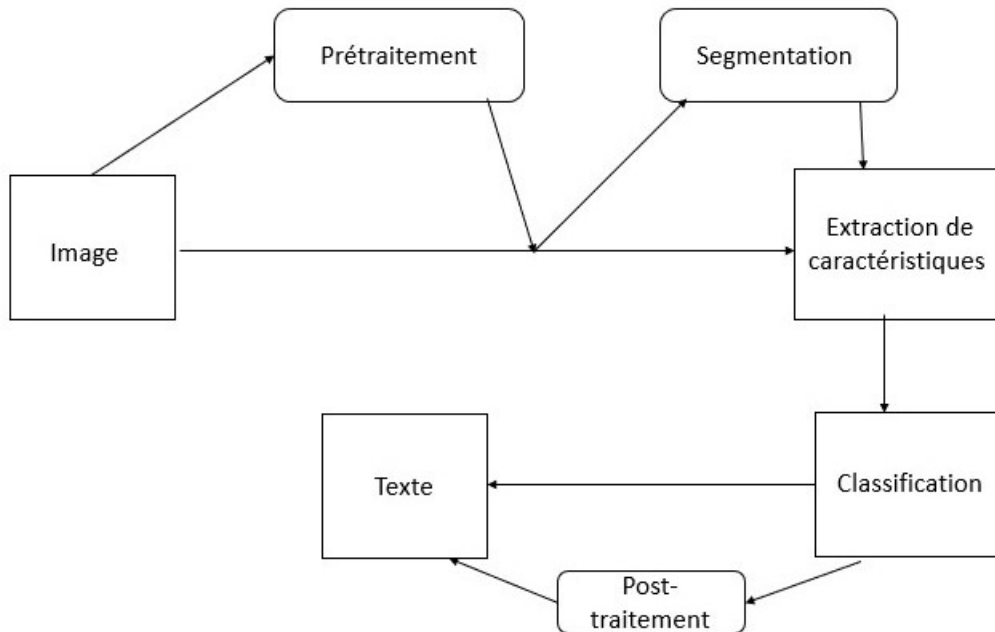


FIGURE IV.6 – Étapes procédurales pendant le processus d'OCR

— **Le processus d'OCR peut être décomposé en les étapes suivantes :**

1. **Acquisition d'image :** La première étape consiste à acquérir l'image du document à traiter. Cela peut être fait en numérisant le document ou en le prenant en photo avec un appareil photo numérique.
2. **Prétraitement :** L'objectif principal de cette étape est de réduire les caractères peuvent être brisés ou tachés et entraîner de faibles taux d'accusé de réception. Le prétraitement est effectué en lissant les caractères . il est généralement nécessaire de traiter l'image pour supprimer le bruit, améliorer le contraste et la luminosité, et normaliser l'image pour faciliter la reconnaissance des caractères..
3. **Segmentation :** dans ce processus Les mots sont segmentés en caractères isolés qui sont comptabilisés séparément. La plupart des algorithmes OCR segmenter les mots en caractères isolés reconnus individuellement.
4. **Extraction de caractéristiques :** ce processus capturera les caractéristiques significatives des symboles et il a deux types des algorithmes qui sont la reconnaissance de formes et comportent extraction/détection.
5. **Classification :** Les systèmes OCR utilisent les techniques de reconnaissance de formes qui attribuent un échantillon inconnu à une classe prédéfinie. L'un des moyens d'aider à la classification des caractères consiste à utiliser le répertoire anglais.
6. **Post-traitement :** Cette étape concerne les techniques et les algorithmes qui améliorent la précision de l'extraction des données pour un résultat optimal. Tout d'abord, les données sont détectées, puis corrigées si

nécessaire. Les données extraites sont comparées à un vocabulaire ou à une bibliothèque de caractères pour des vérifications grammaticales et des considérations contextuelles afin de compléter la phase de post-traitement.

IV.5.3 Application mobile

l'Application Mobile connecté avec des écouteurs bluetooth va convertir les données récupérées de la base de données en audio, lissent par des écouteurs.

Dans notre projet nous avons utilisé le (TTS technologie) pour la conversion des données récupérées à partir de la base en audio .

- **Text-to-Speech (TTS)** est une technologie qui convertit le texte écrit en mots parlés. Les systèmes TTS peuvent analyser le texte et générer un son naturel qui peut être lu sur un appareil, tel qu'un ordinateur ou un smartphone. . Les moteurs TTS se composent des quatre composants principaux , comme indiqué dans la Figure IV.7 :

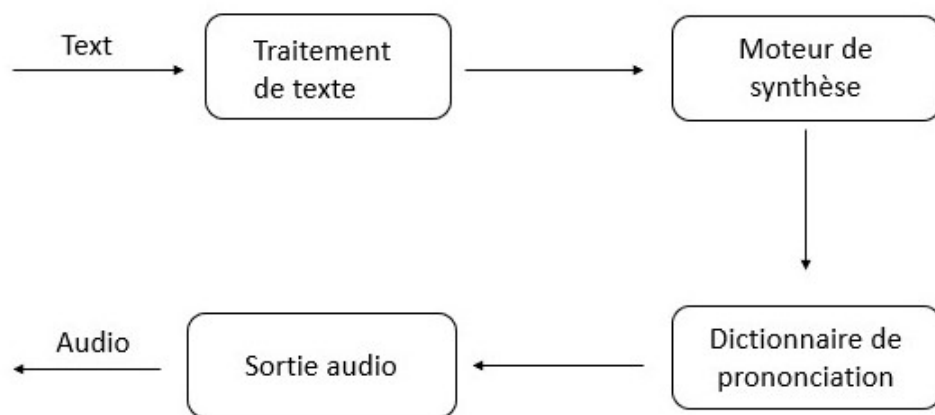


FIGURE IV.7 – Composants du moteur TTS

— **Les moteurs TTS se composent des quatre composants principaux suivants :**

1. **Traitement de texte :** ce composant prend le texte d'entrée et le traite pour identifier les limites des phonèmes, des syllabes et des mots, ainsi que d'autres éléments linguistiques tels que la ponctuation, la grammaire et la prosodie. Ces informations sont ensuite transmises au moteur de synthèse.
2. **Moteur de synthèse :** ce composant est responsable de la génération du signal vocal à partir du texte traité. Il utilise les informations générées par le composant de traitement de texte pour déterminer l'ordre, la durée et la hauteur de la parole synthétisée. Le moteur de synthèse crée une forme d'onde vocale qui peut être lue par un ordinateur ou un autre appareil audio.
3. **Dictionnaire de prononciation :** ce composant stocke les règles de prononciation de chaque mot du texte d'entrée. Il peut également inclure des règles de gestion des homophones et autres ambiguïtés phonétiques.
4. **Sortie audio :** ce composant final produit la parole synthétisée sous forme de signal audio, qui peut être lu par un ordinateur ou un autre appareil audio. Ce composant peut également être responsable du réglage du volume, de la hauteur et d'autres propriétés de la sortie audio pour obtenir l'effet souhaité [69].

IV.6 Conclusion

Dans ce chapitre, nous avons montré les architectures système générales et détaillées de notre projet et expliqué chacune d'elles séparément. nous avons expliqué en détail les algorithmes utilisés. Dans le chapitre suivant, nous parlerons des technologies que nous avons utilisées pour mettre en œuvre notre projet.

Chapitre V

Implémentation du Système

V.1 Introduction

Dans le chapitre précédent, nous avons décrit la conception du système et expliqué chaque étape séparément. Dans ce chapitre, nous définirons les programmes, les outils, les langages de programmation et les bibliothèques utilisés dans la mise en œuvre du projet. Représentez ensuite le matériel utilisé après on va présneté quelques codes. Enfin, nous présentons quelques résultats.

V.2 Environnement de développement

Dans cette section, nous verrons deux aspects, le premier concerne les langages de programmation et les bibliothèques, et le deuxième aspect concerne les outils de développement utilisés dans notre projet.

V.2.1 Langages de programmation et Les bibliothèques

Dans cette section nous parlerons sur les softwares utilisant dans notre projet comme les langages de programmation, les bibliothèques.

V.2.1.1 Java

Java est un langage de programmation orienté objet [17] et une plateforme logicielle largement utilisée qui fonctionne sur des milliards d'appareils, notamment des ordinateurs portables, des appareils mobiles, des consoles de jeux, des appareils médicaux et bien d'autres encore. Les règles et la syntaxe de Java sont basées sur les langages C et C++.. La Figure montre un logo javaV.1



FIGURE V.1 – Java[17]

V.2.1.2 Python

Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et Langages de programmation orientée objet(Poo)[18].on l'a choisi parce que il a obtenu un grand succès dans le domaine de Deep learning grâce aux développeurs qui ont développé de nombreuses bibliothèques, ce qui rend l'utilisation de ce langage facile. La Figure montre un logo de pythonV.2 .



FIGURE V.2 – Python[18]

V.2.1.3 OpenCV

Open Source Computer Vision Library (OpenCv) : est une bibliothèque [19] pour les programmes de vision par ordinateur et d'apprentissage automatique, elle fournit une infrastructure importante pour les applications de vision par ordinateur.

La Figure montre un logo de openCV[19]



FIGURE V.3 – OpenCV[19]

V.2.1.4 Pytesseract

Python-tesseract[20] est un outil de reconnaissance optique de caractères (OCR) pour Python, c'est à dire un logiciel capable de reconnaître du texte à partir d'une image. La Figure montre un logo Pytesseract [20]



FIGURE V.4 – Pytesseract OCR[20]

V.2.1.5 PyTorch

PyTorch[21] est une bibliothèque d'IA, développée par Meta (ex-Facebook), écrite en Python pour se lancer dans le deep learning (ou apprentissage profond) et le développement de réseaux de neurones artificiels. À partir de plusieurs variables, elle peut servir à réaliser des calculs de gradients ou à utiliser des tableaux multidimensionnels obtenus grâce à des tenseurs. La Figure montre un logo PyTorch V.5



FIGURE V.5 – PyTorch[21]

V.2.1.6 NumPy

NumPy est une bibliothèque Python très populaire qui est principalement utilisée pour effectuer des calculs mathématiques et scientifiques. Elle offre de nombreuses fonctionnalités et outils qui peuvent être utiles pour les projets de Data Science[22]. La Figure montre un logo NumPy V.6



FIGURE V.6 – Numpy[22]

V.2.1.7 Keras

Keras [23] est une bibliothèque open source écrite en Python (sous licence MIT) basée principalement sur les travaux du développeur de Google François Chollet dans le cadre du projet ONEIRO (Open-ended Neuro-Electronic Intelligent Robot Operating System). Le but de cette bibliothèque est de permettre la constitution rapide de réseaux neuronaux. La Figure montre un logo V.7



FIGURE V.7 – Keras[23]

V.2.1.8 Tensorflow

TensorFlow [24] est une bibliothèque de Machine Learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissage expérimentales et de les transformer en logiciels. La Figure montre un logo TensorFlow V.8



FIGURE V.8 – Tensorflow[24]

V.2.2 Les outils de développement

Il existe de nombreux outils différents qui sont utilisés pour lire les langages de programmation, dans ce projet nous avons utilisé google colab et pycharm et jupyter.

V.2.2.1 Pycharm

PyCharm est [25] un environnement de développement intégré utilisé pour programmer en Python, il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web. La Figure montre un logo Pycharm V.9



FIGURE V.9 – Pycharm[25]

V.2.2.2 Jupyter Notebook

Jupyter Notebook[26] est un outil de travail collaboratif à l'usage des programmeurs et des développeurs logiciels. Il offre à ces derniers la possibilité de partager du code, des graphiques ou des fichiers multimédias dans des blocs-notes interactifs. la Figure montre un logo Jupyter Notebook V.10



FIGURE V.10 – Jupyter Notebook[26]

V.2.2.3 Android Studio

Android Studio est un environnement de développement intégré (IDE) pour créer des applications Android.[27] Il comprend un éditeur de code, des outils de construction et un gestionnaire de packages, parmi de nombreuses autres ressources. la Figure montre un logo Android Studio V.11



FIGURE V.11 – Android Studio[27]

V.2.2.4 Firebase

Firestore est considéré comme une plate-forme d'applications Web. Ça aide les développeurs créent des applications de haute qualité. Il stocke les données dans le format JavaScript Object Notation (JSON) qui n'utilise pas de requête pour y insérer, mettre à jour, supprimer ou ajouter des données [28]. C'est le backend d'un système utilisé comme base de données pour le stockage des données. La Figure montre un logo V.12



FIGURE V.12 – Firebase[28]

V.3 Schéma Électronique

Ici, nous montrons comment les appareils électroniques sont connectés les uns aux autres lors de la préparation des verres, comme le montre la figure suivante :

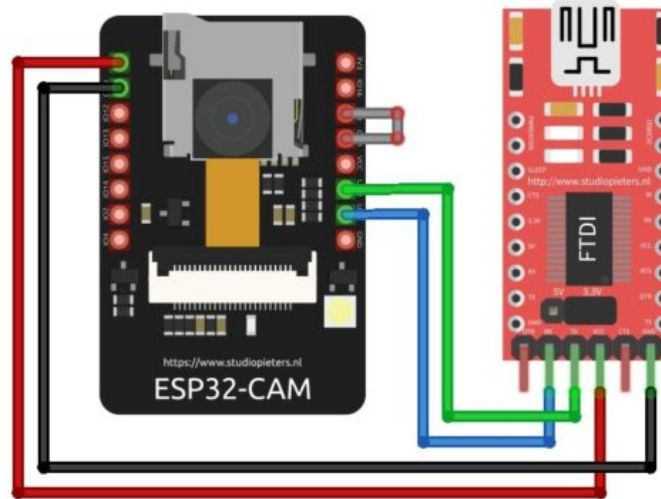


FIGURE V.13 – Schéma électronique[29]

V.3.1 Esp32-Cam

ESP32-CAM [30] est une carte de développement à faible coût basée sur ESP32 avec caméra embarquée, de petite taille. C'est une solution idéale pour les applications IoT. Le module de caméra basé sur ESP32 développé par AI-Thinker. Le contrôleur est basé sur un processeur 32 bits et dispose d'une puce combinée Wi-Fi + Bluetooth/BLE. Il dispose d'une SRAM intégrée de 520 Ko avec une PSRAM externe de 4 Mo. Ses broches GPIO prennent en charge comme UART, SPI, I2C, PWM, ADC et DAC. Comme le montre la figure suivante V.14



FIGURE V.14 – ESP32-CAM[30]

V.3.2 FTDI Driver

USB FTDI est un type de périphérique USB qui utilise la puce FTDI pour communiquer avec un ordinateur. [31] Les puces FTDI sont couramment utilisées dans les systèmes embarqués et les cartes de développement. Ils fournissent un moyen de connecter ces appareils à un ordinateur via USB, ce qui leur permet d'être programmés, débogués et utilisés dans une variété d'applications. Comme le montre la figure suivante V.15



FIGURE V.15 – FTDI Driver[31]

V.4 Installation du matériel

Dans cette section, nous expliquerons comment installer les équipements utilisés, et nous détaillerons les pièces essentielles pour atteindre l'objectif de notre projet, qui est « des lunettes intelligentes pour aider les aveugles ». La caméra esp32 place sur l'une des montures des lunettes, elle est connectée à une intermédiaire FTDI pour alimenter et uploader le code et ce dernier avec la batterie. Esp32-cam dans les lunettes, est un capteur d'image qui permet d'envoyer un flux d'images au serveur ce dernier va le traiter et analyser pour fournir une description sur l'environnement de l'aveugle[29].

Le Tableau suivant V.1 montre comment l'esp32-cam et le FTDI sont connectés.

Esp32-cam	FTDI
GND	GND
5V	VCC
UOR	TX
UOT	RX
GPIO0	GND

TABLE V.1 – Connexion de l'esp32-cam avec le FTDI [29]

Connectez la broche 5V et GND de l'ESP32 au 5V et GND du module FTDI. De même, connectez le Rx à l'UOT et le Tx à la broche UOR. Et la chose la plus importante, vous devez court-circuiter les broches IO0 et GND ensemble. Il s'agit de mettre l'appareil en mode programmation. Une fois la programmation terminée, vous pouvez l'enlever.

la figure V.16 suivante montre la forme finale de notre projet



FIGURE V.16 – Prototype de projet

V.5 Station de traitement

Le Matériel utilisé est un ordinateur avec un CPU : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz et un RAM :8.00 GB.

On verra les fonctions la plus importante dans les codes des algorithmes d'écrits précédemment

V.5.1 Détection d'objets et texte

Code suivant pour détection d'objets et texte.

- Instructions 2,3,6 pour se connecter et accéder à Firebase et instructions à la ligne 80 pour stocker nos données dans Firebase.

— **Bibliothèques nécessaires** :pour se connecter et accéder à Firebase.

```
1 # IMPORTER LES BIBLIOTHQUES NCESSAIRES
2 import firebase_admin
3 from firebase_admin import credentials
4 from firebase_admin import db
5 import math
```

```

6 import cv2
7 from ultralytics import YOLO
8 import pytesseract
9 import time

```

Listing V.1 – Les bibliothèques nécessaires pour se connecter firebase

Comme montré dans le code suivante :

```

1 def Detect():
2     cred = credentials.Certificate("amadora-714a2-firebase-adminsdk-2pbt4-0f7717ae1d.
3     json")
4     firebase_admin.initialize_app(cred, {
5         'databaseURL': 'https://amadora-714a2-default-rtdb.firebaseio.com/'
6     })
7     ref = db.reference('objet')
8     pytesseract.tesseract_cmd = 'C:\Program Files\Tesseract-OCR\tesseract.exe'
9     path_x =0
10    video_capture = path_x
11    # Create a cam Object
12    cap = cv2.VideoCapture(video_capture)
13    object = {}
14    model = YOLO("../YOLO-Weights/yolov8n.pt")
15    classNames = ["person", "bicycle", "car", "motorbike", "aeroplane", "bus", "train
16    ", "truck", "boat",
17        "traffic light", "fire hydrant", "stop sign", "parking meter", "
18    bench", "bird", "cat",
19        "dog", "horse", "sheep", "cow", "elephant", "bear", "zebra", "
20    giraffe", "backpack", "umbrella",
21        "handbag", "tie", "suitcase", "frisbee", "skis", "snowboard", "
22    sports ball", "kite",
23        "baseball bat",
24        "baseball glove", "skateboard", "surfboard", "tennis racket", "
25    bottle", "wine glass", "cup",
26        "fork", "knife", "spoon", "bowl", "banana", "apple", "sandwich", "
27    orange", "broccoli",
28        "carrot", "hot dog", "pizza", "donut", "cake", "chair", "sofa", "

```

```

pottedplant", "bed",
22         "diningtable", "toilet", "tvmonitor", "laptop", "mouse", "remote",
"keyboard", "cell phone",
23         "microwave", "oven", "toaster", "sink", "refrigerator", "book", "
clock", "vase", "scissors",
24         "teddy bear", "hair drier", "toothbrush"
25     ]
26     while True:
27         success, img = cap.read()
28         img = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
29         results = model(img, stream=True, show=True)
30         for r in results:
31             start_time = time.time()
32             boxes = r.boxes
33             for box in boxes:
34                 x1, y1, x2, y2 = box.xyxy[0]
35                 x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
36                 print(x1, y1, x2, y2)
37                 cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 255), 3)
38                 conf = math.ceil((box.conf[0] * 100)) / 100
39                 cls = int(box.cls[0])
40                 class_name = classNames[cls]
41                 if class_name in object.keys():
42                     object[class_name] += 1
43                 else:
44                     object[class_name] = 1
45                     # Get the text that was detected
46                     # by the OCR library
47                 text = pytesseract.image_to_string(img)
48                 if (text!=' \n\x0c'):
49                     object["text"] = text
50                 label = f'{class_name}{conf}'
51                 t_size = cv2.getTextSize(label, 0, fontScale=1, thickness=2)[0]
52                 print(t_size)
53                 c2 = x1 + t_size[0], y1 - t_size[1] - 3
54                 cv2.rectangle(img, (x1, y1), c2, [255, 0, 255], -1, cv2.LINE_AA) #

```

```

filled
55         cv2.putText(img, label, (x1, y1 - 2), 0, 1, [255, 255, 255],
thickness=1, lineType=cv2.LINE_AA)
56
57         new_dict = {}
58         end_time = time.time()
59         total_time = end_time - start_time
60         #print("end_time",end_time)
61         #print("total_time",total_time)
62         time_struct = time.localtime(end_time)
63
64         hour = "{:02d}".format(time_struct.tm_hour)
65         minute = "{:02d}".format(time_struct.tm_min)
66         second = "{:02d}".format(time_struct.tm_sec)
67
68         # Print the hour, minute, and second as a formatted string
69         # print("Time: {:}:{:}:".format(hour, minute, second))
70         for key, value in object.items():
71             if isinstance(value, int):
72                 new_dict[key] = str(value)
73             else:
74                 new_dict[key] = value
75         new_dict["time"] = "{:}:{:}:".format(hour, minute, second)
76         print(new_dict)
77         print("object:", new_dict)
78         if(new_dict!={}) :
79             ref.delete()
80             ref.update(new_dict)
81         object = {}
82 if __name__ == '__main__':
83     Detect()

```

Listing V.2 – Détection d'objets

V.5.2 Détection d'objets

- **Bibliothèques nécessaires** : pour détecter d'objets, nous avons besoin d'une bibliothèque, qui est OpenCV et Ultralytics

```
1 # IMPORTER LES BIBLIOTHQUES NCESSAIRES
2 from ultralytics import YOLO
3 import math
4 import cv2
```

Listing V.3 – Les bibliothèques nécessaires pour la détection des objets

- **Détection des objets** : pour détecter des objets, nous avons utilisé le modèle Yolov8n.
 - Instruction à la ligne 13 pour charger le modèle.
 - Classe VideoCapture permet de capturer de la vidéo à partir d'une caméra qu'elle existe à la ligne 11.
 - Variable cap contiendra une référence à l'objet VideoCapture. Cet objet peut être utilisé pour lire les images de la vidéo, obtenir des propriétés vidéo comme la ligne 27.
 - Instruction à la ligne 29 pour exécuter la détection d'objet sur l'image img à l'aide du modèle de modèle YOLO.

V.5.3 Reconnaissance de texte

- **Bibliothèques nécessaires** : pour reconnaissance de texte , nous avons besoin d'une bibliothèque, qui est pytesseract.

```
1 # IMPORTER LES BIBLIOTHQUES NCESSAIRES
2 import pytesseract
3 import math
```

```
4 import cv2
```

Listing V.4 – Les bibliothèques nécessaires pour la reconnaissance de texte

- **Détection des text** :pour reconnaissance de texte , nous utilisons une fonction définie dans la bibliothèque tesseract appelée :pytesseract.image to string() prend une image en entrée et renvoie une chaîne du texte qui a été extrait de l'image à la ligne 47.

V.5.4 Application mobile

- **Bibliothèques nécessaires.**

```
1 # IMPORTER LES BIBLIOTHQUES NCESSAIRES
2 import androidx.appcompat.app.AppCompatActivity;
3 import android.annotation.TargetApi;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.content.SharedPreferences;
7 import android.os.Build;
8 import android.os.Bundle;
9 import android.preference.PreferenceManager;
10 import android.speech.tts.TextToSpeech;
11 import android.util.Log;
12 import android.view.Menu;
13 import android.view.MenuItem;
14 import android.view.Window;
15 import android.view.WindowManager;
16 import android.widget.TextView;
17 import com.google.firebase.database.DataSnapshot;
18 import com.google.firebase.database.DatabaseError;
19 import com.google.firebase.database.DatabaseReference;
20 import com.google.firebase.database.FirebaseDatabase;
21 import com.google.firebase.database.ValueEventListener;
22 import org.json.JSONException;
```

```

23 import org.json.JSONObject;
24 import java.util.Locale;

```

Listing V.5 – Les bibliothèques nécessaires pour l'application mobile

— **Application mobile** : pour convertir les données récupérées à partir de la base de données en audio, lisons par des écouteurs, nous nous référons à la table puis récupérons les données à la ligne 25 du code et puis nous convertissons les données récupérées en son à l'aide de TextToSpeech (TTS) à la ligne 25 du code.

```

1 public class MainActivity extends AppCompatActivity {
2     TextView tv;
3     static TextToSpeech mTTS;
4
5
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         getWindow().requestFeature(Window.FEATURE_NO_TITLE);
9         getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
10             WindowManager.LayoutParams.FLAG_FULLSCREEN);
11
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
15         getSupportActionBar().setIcon(R.drawable.ic_amadora);
16         tv = (TextView) findViewById(R.id.tv_output);
17         if (isFirstTimeLaunch(getApplicationContext())) {
18
19             speak("Welcome to our AmaDoRa app, which is designed to help you understand
20 and interact with the world around you.");
21             setFirstTimeLaunch(getApplicationContext(), false);
22         }
23

```



```
24 DatabaseReference myRef = FirebaseDatabase.getInstance().getReference("objet");
25 myRef.addValueEventListener(new ValueEventListener() {
26     @TargetApi(Build.VERSION_CODES.O)
27     @Override
28     public void onDataChange(DataSnapshot dataSnapshot) {
29
30         JSONObject json = new JSONObject();
31         for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
32             String key = snapshot.getKey().toString();
33             String value = snapshot.getValue().toString();
34             try {
35                 json.put(value, key);
36             } catch (JSONException e) {
37                 e.printStackTrace();
38             }
39         }
40         String Description=json.toString();
41         Description = Description.replace(',','\n');
42         Description = Description.replace('{','\n');
43         Description = Description.replace('}','\n');
44         Description = Description.replace(':', '\n');
45
46         speak(Description);
47     }
48
49     @Override
50     public void onCancelled(DatabaseError databaseError) {
51
52         Log.w("h", "Failed to read value.", databaseError.toException());
53     }
54 });
55
56
57 }
58 public void speak(String word) {
59
```

```
60     mTTS = new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener
61     () {
62         @Override
63         public void onInit(int status) {
64             if (status == TextToSpeech.SUCCESS) {
65                 int result = mTTS.setLanguage(Locale.US);
66                 if (result == TextToSpeech.LANG_MISSING_DATA || result ==
67                 TextToSpeech.LANG_NOT_SUPPORTED) {
68                     Log.d("TTS", "Language not supported");
69                 } else {
70                     mTTS.speak(word, TextToSpeech.QUEUE_FLUSH, null);
71                     Log.d("done", "done");
72                 }
73             }
74         }
75     });
76
77 }
78
79 public void setFirstTimeLaunch(Context context, boolean isFirstTime) {
80     SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(context);
81     SharedPreferences.Editor editor = prefs.edit();
82     editor.putBoolean("FIRST_TIME_LAUNCH", isFirstTime);
83     editor.apply();
84 }
85
86 public boolean isFirstTimeLaunch(Context context) {
87     SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(context);
88     return prefs.getBoolean("FIRST_TIME_LAUNCH", true);
89 }
90
91 @Override
92 public void onBackPressed() {
93     if (mTTS != null) {
```

```
94     mTTS.stop();
95     mTTS.shutdown();
96 }
97     super.onBackPressed();
98 }
99 @Override
100 public boolean onOptionsItemSelected( MenuItem item) {
101     switch (item.getItemId()) {
102         case R.id.share:
103             ShareApp(MainActivity.this);
104             return true;
105
106         case R.id.help:
107             openActivity();
108             return true;
109     }
110     return super.onOptionsItemSelected(item);
111 }
112 @Override
113 public boolean onCreateOptionsMenu(Menu menu) {
114     getMenuInflater().inflate(R.menu.menu, menu);
115     return true;
116 }
117
118 private void ShareApp(Context context) {
119
120     final String appPakageName = context.getPackageName();
121     Intent sendIntent = new Intent();
122     sendIntent.setAction(Intent.ACTION_SEND);
123     sendIntent.putExtra(Intent.EXTRA_TEXT, "Download Now : https://play.google.com/
124     sendIntent.setType("text/plain");
125     context.startActivity(sendIntent);
126 }
127
128 private void openActivity() {
```

```
129     Intent intent = new Intent(getApplicationContext(), helpActivity.class);
130     startActivity(intent);
131
132 }
133 }
```

Listing V.6 – Application mobile

V.6 Résultats

Nous présenterons quelques-uns des résultats que les lunettes atteindront

— **Résultats pour la détection des objects :** comme montrer dans les figures suivantes



FIGURE V.17 – Résultats de détection des objects exemple 1

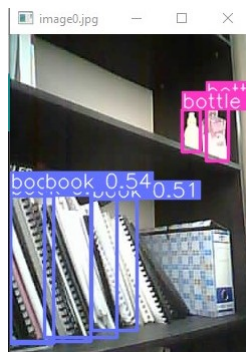


FIGURE V.18 – Résultats de détection des objects exemple 2

— **Résultats pour la reconnaissance de texte :** comme montre la figure suivante

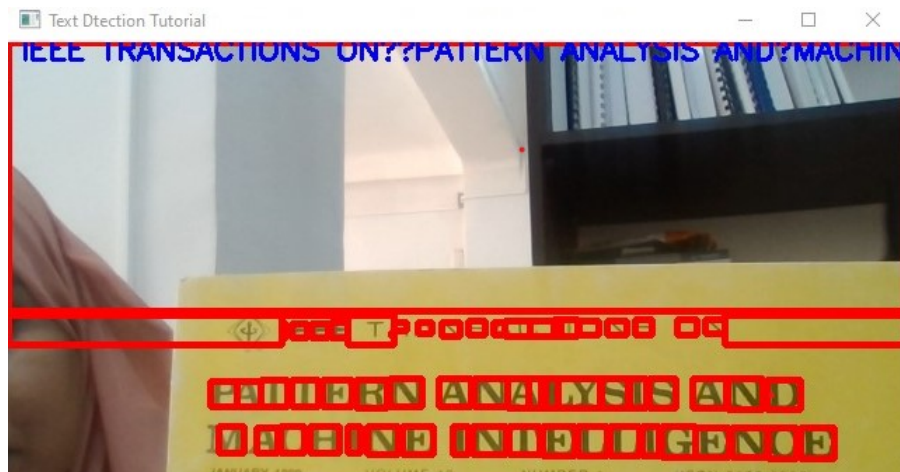


FIGURE V.19 – Résultats de reconnaissance de texte

— **Résultats pour l'application mobile :** comme montrer dans les figures suivantes

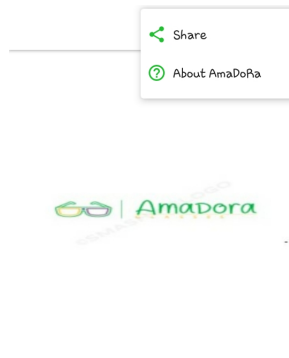


FIGURE V.20 – Résultats d'application mobile interface 1

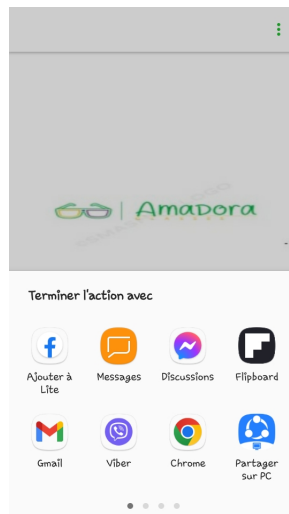


FIGURE V.21 – Résultats d'application mobile interface 2

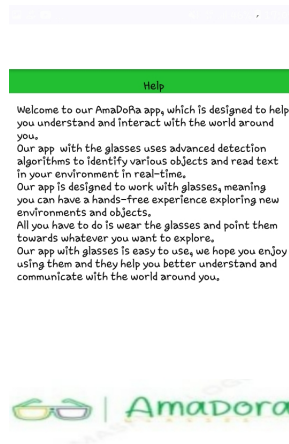


FIGURE V.22 – Résultats d'application mobile interface 3

- **Résultats pour le temps de reponse :** Tanque que les objets et le texte sont détectés l'aveugle après [2s - 7s] secondes écoutera la description ,comme montrer la figure suivante :

```
I/TextToSpeech: Successfully bound to com.samsung.SMT
I/System.out: Current time of detection in image:15:51:12
I/System.out: Current time after reading the description for the blind:15:51:16
while the objects and the text are detected the blind after 4 secondes will listen the description
```

FIGURE V.23 – Temps de reponse

V.7 Conclusion

Dans ce chapitre, nous avons défini les outils, logiciels et matériels utilisés pour développer notre système, et nous avons également montré les morceaux de code utilisés à chaque étape de l'architecture du système, ainsi que la façon dont les appareils électroniques sont connectés les uns aux autres pour créer un modèle des lunettes intelligentes identifiant chaque appareil séparément. Enfin, nous avons également présenté les résultats obtenus à l'issue du développement de notre système.

Conclusion Générale

Dans cette mémoire, nous avons présenté cinq chapitres, le premier chapitre portait sur l'Internet des objets et certains concepts qui s'y rapportent, son fonctionnement, son architecture, ses structures et ses domaines, tandis que le deuxième chapitre portait sur l'apprentissage en profondeur et ces types et certains concepts connexes, tels que l'intelligence artificielle et les réseaux de neurones, portait aussi sur la vision par ordinateur et ses tâches importantes et ses domaines d'application.

Nous l'avons présenté dans le troisième chapitre la déficience visuelle, ses types et ses degrés, ainsi que les difficultés rencontrées par les personnes aveugles dans leur vie quotidienne, comme nous le verrons à propos des aides techniques disponibles sur le marché. et dans notre quatrième chapitre nous avons montré la structure générale et détaillée de notre système et expliqué son fonctionnement. Enfin, nous avons présenté et discuté le schéma électronique du système et ses composants, selon les différents résultats que nous avons obtenus. À l'avenir, nous voulons ajouter la position des objets et leur emplacement et l'identification des personnes.

Bibliographie

- [1] Partha Pratim Ray. A survey of iot cloud platforms. *Future Computing and Informatics Journal*, 1(1-2) :35–46, 2016.
- [2] Zhihong Yang, Yingzhao Yue, Yu Yang, Yufeng Peng, Xiaobo Wang, and Wenji Liu. Study and application on the architecture and key technologies for iot. In *2011 International Conference on Multimedia Technology*, pages 747–751. IEEE, 2011.
- [3] Dekhmouche Racha, Gabsi Sana, and A Merazga. Conception et réalisation d'un surveillance médicale basé iot. 2021.
- [4] Vasileios Moysiadis, Panagiotis Sarigiannidis, Vasileios Vitsas, and Adel Khelifi. Smart farming in europe. *Computer science review*, 39 :100345, 2021.
- [5] Messaouda Bourezak, Safa Chetibi, and Ammar Encadreur Soukkou. *Conception et réalisation d'un système de pilotage d'une installation domotique à distance (IoT) à base d'Arduino*. PhD thesis, Université de Jijel, 2019.
- [6] Marcelo T Okano. Iot and industry 4.0 : the industrial new revolution. In *International Conference on Management and Information Systems*, volume 25, page 26, 2017.

-
- [7] A-T Dinh-Xuan. Intelligence artificielle, «machine learning» et «deep learning» : de nouvelles notions bientôt incontournables en pneumologie? *Revue des Maladies Respiratoires Actualités*, 11(2) :59–62, 2019.
- [8] Mariette Awad and Rahul Khanna. *Efficient learning machines : theories, concepts, and applications for engineers and system designers*. Springer nature, 2015.
- [9] Charu C Aggarwal et al. Neural networks and deep learning. *Springer*, 10(978) :3, 2018.
- [10] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3 :19–48, 2010.
- [11] Aymen MHAMMEDI, Imane YAKOUB, Abdelwhab OUAHAB, et al. *La détection de Covid-19 par l'apprentissage profonde (Deep Learning)*. PhD thesis, UNIVERSITE AHMED DRAIA-ADRAR, 2021.
- [12] Jagadish K Mahendran, Daniel T Barry, Anita K Nivedha, and Suchendra M Bhandarkar. Computer vision-based assistance system for the visually impaired using mobile edge artificial intelligence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2418–2427, 2021.
- [13] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep learning for visual understanding : A review. *Neurocomputing*, 187 :27–48, 2016.

- [14] Guide to Computer Vision : Why It Matters and How It Helps Solve Problems | NVIDIA Technical Blog, 3 2023.
- [15] I Maione. Computer vision : Definition, applications, tools, history future [vidéo], 2.
- [16] Lixuan Du, Rongyu Zhang, and Xiaotian Wang. Overview of two-stage object detection algorithms. *Journal of Physics : Conference Series*, 1544(1) :012033, may 2020.
- [17] Michael Juntao Yuan. *Enterprise J2ME : developing mobile Java applications*. Prentice Hall Professional, 2004.
- [18] Giang Nguyen, Stefan Dlugolinsky, Martin Bobák, Viet Tran, Álvaro López García, Ignacio Heredia, Peter Malík, and Ladislav Hluchý. Machine learning and deep learning frameworks and libraries for large-scale data mining : a survey. *Artificial Intelligence Review*, 52 :77–124, 2019.
- [19] Chad Kaplowitz. *Characterization of a Stereo Vision System for Object Classification for USV Navigation*. PhD thesis, Florida Atlantic University, 2022.
- [20] Jaskirat Singh and Bharat Bhushan. Real time indian license plate detection using deep neural networks and optical character recognition using lstm tesseract. In *2019 international conference on computing, communication, and intelligent systems (ICCCIS)*, pages 347–352. IEEE, 2019.
- [21] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow : Solution for Edge Computing Applications*, pages 87–104, 2021.

- [22] Travis E Oliphant et al. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- [23] Nikhil Ketkar and Nikhil Ketkar. Introduction to keras. *Deep learning with python : a hands-on introduction*, pages 97–111, 2017.
- [24] Fatih Ertam and Galip Aydın. Data classification with deep learning using tensorflow. In *2017 international conference on computer science and engineering (UBMK)*, pages 755–758. IEEE, 2017.
- [25] Haouas Mounir. Traitement d’image médicale par python sur la carte fpga.
- [26] Sacha Hodencq. *Méthodes et outils pour un processus de modélisation collaboratif et ouvert des systèmes énergétiques*. PhD thesis, Université Grenoble Alpes, 2022.
- [27] Apache Cordova. Apache cordova. *Recuperado el*, 17, 2015.
- [28] Laurence Moroney and Laurence Moroney. The firebase realtime database. *The Definitive Guide to Firebase : Build Android Apps on Google’s Mobile Platform*, pages 51–71, 2017.
- [29] Dedi Setiawan, Hendra Jaya, Saiful Nurarif, Trinanda Syahputra, and Muhammad Syahril. Implementasi esp32-cam dan blynk pada wifi door lock system menggunakanteknik duplex. *JOURNAL OF SCIENCE AND SOCIAL RESEARCH*, 5(1) :159–164, 2022.
- [30] Henry Dietz, Dillon Abney, Paul Eberhart, Nick Santini, William Davis, Elisabeth Wilson, and Michael McKenzie. Esp32-cam as a programmable camera research platform. *Imaging*, 232(2), 2022.

-
- [31] Michal Sedlák. Data acquisition from educational plants over usb in gnu/linux. In *International Conference February*, volume 10, page 13, 2010.
- [32] Ruth N Wambua and Robert Oboko. Elearning for persons with visual disabilities : Case of low vision. In *Handbook of Research on Educational Technology Integration and Active Learning*, pages 247–267. IGI Global, 2015.
- [33] Didier Fassin. L'invention française de la discrimination. *Revue française de science politique*, (4) :403–423, 2002.
- [34] Mohsen Hallaj Asghar, Atul Negi, and Nasibeh Mohammadzadeh. Principle application and vision in internet of things (iot). In *International Conference on Computing, Communication & Automation*, pages 427–431. IEEE, 2015.
- [35] Mrutyunjaya Panda and BK Tripathy. Internet of things and artificial intelligence : a new road to the future digital world. In *Internet of Things (IoT)*, pages 41–58. CRC Press, 2017.
- [36] Azeddine Khiat, Ayoub Bahnasse, Jamila Bakkoury, Mohamed El Khaili, and Fatima Ezzahraa Louhab. New approach based internet of things for a clean atmosphere. *International Journal of Information Technology*, 11 :89–95, 2019.
- [37] Chirihane Gherbi. Internet of things and heterogeneous networks technologies : Concepts, challenges and perspectives. *Ingénierie des Systèmes d'Information*, 26(4), 2021.
- [38] Clive WJ Granger. Some properties of time series data and their use in

- econometric model specification. *Journal of econometrics*, 16(1) :121–130, 1981.
- [39] Perry Xiao. *Designing Embedded Systems and the Internet of Things (IoT) with the ARM mbed*. John Wiley & Sons, 2018.
- [40] Pawan Kumar Verma, Rajesh Verma, Arun Prakash, Ashish Agrawal, Kshirasagar Naik, Rajeev Tripathi, Maazen Alsabaan, Tarek Khalifa, Tamer Abdelkader, and Abdulhakim Abogharaf. Machine-to-machine (m2m) communications : A survey. *Journal of Network and Computer Applications*, 66 :83–105, 2016.
- [41] Amir M Rahmani, Tuan Nguyen Gia, Behailu Negash, Arman Anzanpour, Iman Azimi, Mingzhe Jiang, and Pasi Liljeberg. Exploiting smart e-health gateways at the edge of healthcare internet-of-things : A fog computing approach. *Future Generation Computer Systems*, 78 :641–658, 2018.
- [42] Mohammad Ali Jabraeil Jamali, Bahareh Bahrami, Arash Heidari, Parisa Allahverdizadeh, Farhad Norouzi, Mohammad Ali Jabraeil Jamali, Bahareh Bahrami, Arash Heidari, Parisa Allahverdizadeh, and Farhad Norouzi. Iot architecture. *Towards the Internet of Things : Architectures, Security, and Applications*, pages 9–31, 2020.
- [43] Nenad Gligorić, Ana Uzelac, and Srdjan Krco. Smart classroom : real-time feedback on lecture quality. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 391–394. IEEE, 2012.

-
- [44] Ercan Oztemel and Samet Gursev. Literature review of industry 4.0 and related technologies. *Journal of intelligent manufacturing*, 31 :127–182, 2020.
- [45] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai) : Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58 :82–115, 2020.
- [46] Tom Michael Mitchell. *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning ..., 2006.
- [47] H Wang, ZeZXEZBePJ Lei, X Zhang, B Zhou, and J Peng. Machine learning basics. *Deep learning*, pages 98–164, 2016.
- [48] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv :1511.08458*, 2015.
- [49] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12) :310–316, 2017.
- [50] Jason Brownlee. 14 different types of learning in machine learning.
- [51] Convolutional neural network. deepai. <https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network>, 2020.
- [52] Hoyos Nathalie. Deep learning : les réseaux de neurones récurrents (rnn). datavalue consulting. <https://datavalue-consulting.com/deep-learning-reseaux-neurones-recurrents-rnn/>, 2022.

- [53] ABDENNOUR YAHIAOUI. Système de discrimination visages/faux visages par réseaux de neurones convolutifs (cnn). 2021.
- [54] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [55] Solemane Coulibaly. *Analyse intelligente des images pour la surveillance dans une agriculture de précision*. PhD thesis, 2021.
- [56] James J Corbett, Peter J Savino, H Stanley Thompson, Tulay Kansu, Norman J Schatz, Linda S Orr, and Daniel Hopson. Visual loss in pseudotumor cerebri : follow-up of 57 patients from five to 41 years and a profile of 14 patients with permanent severe visual loss. *Archives of neurology*, 39(8) :461–474, 1982.
- [57] Carol M Mangione, Paul P Lee, Peter R Gutierrez, Karen Spritzer, Sandra Berry, Ron D Hays, National Eye Institute Visual Function Questionnaire Field Test Investigators, et al. Development of the 25-list-item national eye institute visual function questionnaire. *Archives of ophthalmology*, 119(7) :1050–1058, 2001.
- [58] Ludivine Becerra 1 3, Drina Candilis-Huisman 4, Edith Thoueille 5, and Jacqueline Wendland 2 3 6. Maternité et déficience visuelle : accession à la parentalité, soutien social, compétence parentale, dépression et représentations d’attachement. *Devenir*, 28(2) :91–108, 2016.
- [59] C Kestens. Les déficiences visuelles. *Rev Med Brux*, 40 :278–84, 2019.
- [60] Lise. Signaler les obstacles aux malvoyants, 7 2022.

- [61] Naiwrita Dey, Ankita Paul, Pritha Ghosh, Chandrama Mukherjee, Rahul De, and Sohini Dey. Ultrasonic sensor based smart blind stick. In *2018 international conference on current trends towards converging technologies (ICCTCT)*, pages 1–4. IEEE, 2018.
- [62] Hawra AlSaid, Lina AlKhatib, Aqeela AlOraidh, Shoaab AlHaidar, and Abul Bashar. Deep learning assisted smart glasses as educational aid for visually challenged students. In *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pages 1–6. IEEE, 2019.
- [63] Jung-Hwna Kim, Sun-Kyu Kim, Tea-Min Lee, Yong-Jin Lim, and Joonhong Lim. Smart glasses using deep learning and stereo camera. In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pages 294–295, 2019.
- [64] Abdelfettah Zeghoudi and Benoudina Hadjer. Etude et réalisation d’un bâton d’aveugle intelligent à base des capteurs ultrason. 11 2019.
- [65] Jyun-You Lin, Chi-Lin Chiang, Meng-Jin Wu, Chih-Chiung Yao, and Ming-Chiao Chen. Smart glasses application system for visually impaired people based on deep learning. In *2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*, pages 202–206, 2020.
- [66] Trad Houssein Eddine. La détection d’objet avec opencv et deep learning. 2020.
- [67] Abderaouf Termoul. *Géolocalisation des objets détectés par une caméra mo-*

noculaire à bord d'un véhicule. PhD thesis, École de technologie supérieure, 2023.

- [68] Armstrong Aboah, Bin Wang, Ulas Bagci, and Yaw Adu-Gyamfi. Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5349–5357, June 2023.
- [69] John F Pitrelli, Raimo Bakis, Ellen M Eide, Raul Fernandez, Wael Hamza, and Michael A Picheny. The ibm expressive text-to-speech synthesis system for american english. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4) :1099–1108, 2006.