

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DE TECHNOLOGIE
DEPARTEMENT D'ELECTRONIQUE
N° :.....



DOMAINE : SCIENCE ET TECHNOLOGIE
FILIERE : ELECTRONIQUE
OPTION : INSTRUMENTATION

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique
Dans le cadre du décret ministériel 1275**

Par:

Bachiri Bochra

Alizouaoui Sabrina

Benattig Allaa

Meftah Zahra

Intitulé

**Améliorer l'accessibilité pour les malvoyants :
Une solution basée sur l'IA**

Soutenu devant le jury composé de :

| | | |
|-----------------|-----------------------------------|------------|
| Dr. BRIK Mourad | Université Mohamed Boudiaf M'sila | Président |
| | Université Mohamed Boudiaf M'sila | Rapporteur |
| | Université Mohamed Boudiaf M'sila | Examineur |
| | Université Mohamed Boudiaf M'sila | Examineur |
| | Université Mohamed Boudiaf M'sila | Incubateur |

Année universitaire : 2024 / 2025

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DE TECHNOLOGIE
DEPARTEMENT D'ELECTRONIQUE
N° :.....



DOMAINE : SCIENCE ET TECHNOLOGIE
FILIERE : ELECTRONIQUE
OPTION : INSTRUMENTATION

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique
Dans le cadre du décret ministériel 1275**

Par:

Bachiri Bochra

Alizouaoui Sabrina

Benattig Allaa

Meftah Zahra

Intitulé

**Enhancing accessibility for the visually impaired:
an AI-based solution**

Soutenu devant le jury composé de :

| | | |
|-----------------|-----------------------------------|------------|
| Dr. BRIK Mourad | Université Mohamed Boudiaf M'sila | Président |
| | Université Mohamed Boudiaf M'sila | Rapporteur |
| | Université Mohamed Boudiaf M'sila | Examineur |
| | Université Mohamed Boudiaf M'sila | Examineur |
| | Université Mohamed Boudiaf M'sila | Incubateur |

Année universitaire : 2024 / 2025

Remerciements

Au terme de ce travail de fin d'études, nous souhaitons exprimer notre profonde gratitude à toutes les personnes qui nous ont accompagnés, soutenus et conseillés tout au long de la réalisation de ce projet.

Nous adressons nos remerciements les plus sincères à Monsieur Brik Mourad, notre encadrant, pour la qualité de son accompagnement, sa disponibilité constante, et ses conseils éclairés. Son expertise scientifique et ses orientations précieuses ont grandement contribué à la progression de notre travail.

Nous exprimons également notre reconnaissance à Monsieur Tebakh Mostefa, notre chef de département, pour son appui institutionnel et ses encouragements, qui ont facilité le bon déroulement de ce projet.

Nous remercions l'ensemble du corps enseignant et administratif de la Université Mohamed Boudiaf, pour la qualité de l'enseignement dispensé tout au long de notre parcours universitaire, ainsi que pour l'environnement académique stimulant dont nous avons bénéficié.

Enfin, nous tenons à remercier toutes les personnes qui, de près ou de loin, ont contribué au bon déroulement de ce travail par leur aide, leur soutien moral ou leurs conseils.

DEDICACE

Nous dédions ce travail à nos chères familles, pour leur amour inconditionnel, leur patience, et leur soutien indéfectible tout au long de notre parcours.

À nos amis, qui ont su nous encourager, nous motiver, et nous rappeler l'importance de croire en nous-mêmes même dans les moments les plus exigeants.

À toutes les personnes qui nous ont soutenus, ne serait-ce que par un mot bienveillant, un sourire ou une écoute attentive, et qui ont contribué, par leur présence, à rendre cette aventure plus riche et plus humaine.

À vous tous, nous adressons cette modeste dédicace, empreinte de reconnaissance et d'affection.

Résumé

L'apprentissage automatique suscite beaucoup d'attention ces derniers temps dans le domaine de la vision par ordinateur, car il évolue constamment. Les avancées de l'apprentissage automatique ont été utilisées dans de nombreux domaines, y compris dans le développement de solutions d'assistance pour les personnes malvoyantes. Parmi les défis auxquels ces personnes sont confrontées, on trouve la difficulté d'accéder aux informations visuelles dans leur environnement, ce que nous avons abordé dans ce mémoire. Dans ce travail, nous présentons un système intelligent permettant aux utilisateurs malvoyants de détecter et de lire les panneaux médicaux de manière vocale, renforçant ainsi leur autonomie lors de leurs déplacements. L'application utilise un modèle YOLO pour détecter et classifier en temps réel les panneaux à partir des images capturées par la caméra. En plus de la détection des panneaux, le système extrait le texte des images à l'aide de la technologie de reconnaissance optique de caractères (OCR), puis le convertit en parole via un module de synthèse vocale (TTS), afin que l'utilisateur puisse écouter instantanément l'information détectée. Ce travail constitue une avancée significative vers le développement de solutions intelligentes basées sur l'intelligence artificielle, visant à améliorer la qualité de vie des personnes malvoyantes, tout en ouvrant des perspectives prometteuses pour de futures évolutions dans ce domaine.

Mots clés : accessibilité, apprentissage automatique, handicap visuel, lunettes intelligentes, assistance visuelle, analyse d'images en temps réel

Abstract

Machine learning has been attracting a lot of attention recently in the field of computer vision, as it is constantly evolving. Advances in machine learning have been used in many fields, including the development of assistive solutions for visually impaired people. Among the challenges these people face is the difficulty of accessing visual information in their environment, which we have addressed in this dissertation. In this work, we present an intelligent system enabling visually impaired users to detect and read medical signs vocally, thus enhancing their autonomy while on the move. The application uses a YOLO model to detect and classify signs in real time from images captured by the camera. In addition to sign detection, the system extracts text from images using optical character recognition (OCR) technology, then converts it into speech via a text-to-speech (TTS) module, so that the user can instantly listen to the detected information. This work represents a significant advance towards the development of intelligent solutions based on artificial intelligence, aimed at improving the quality of life of visually impaired people, while opening up promising perspectives for future developments in this field.

Key words: accessibility, machine learning, visual disability, smart glasses, visual assistance, real-time image analysis

ملخص

يجذب التعلم الآلي الكثير من الاهتمام في الآونة الأخيرة في مجال الرؤية الحاسوبية، حيث يتطور باستمرار. وقد استُخدمت التطورات في مجال التعلم الآلي في العديد من المجالات، بما في ذلك تطوير الحلول المساعدة للأشخاص ضعاف البصر. ومن بين التحديات التي يواجهها هؤلاء الأشخاص صعوبة الوصول إلى المعلومات البصرية في بيئتهم، وهو ما تناولناه في هذه الأطروحة. في هذا العمل، نقدم نظامًا ذكيًا يمكّن المستخدمين المعاقين بصريًا من اكتشاف وقرأءة الإشارات الطبية صوتيًا، وبالتالي تعزيز استقلاليتهم أثناء التنقل. لاكتشاف العلامات وتصنيفها في الوقت الفعلي من الصور الملتقطة بواسطة الكاميرا. وبالإضافة إلى YOLO يستخدم التطبيق نموذج ، ثم يحولها إلى كلام (OCR) للكشف عن العلامات، يستخرج النظام النص من الصور باستخدام تقنية التعرف الضوئي على الحروف ، بحيث يمكن للمستخدم الاستماع إلى المعلومات المكتشفة على الفور. ويمثل هذا العمل (TTS) عبر وحدة تحويل النص إلى كلام تقدماً كبيراً نحو تطوير حلول ذكية تعتمد على الذكاء الاصطناعي، وتهدف إلى تحسين نوعية حياة الأشخاص ضعاف البصر، مع فتح آفاق واعدة للتطورات المستقبلية في هذا المجال.

الكلمات المفتاحية: امكانية الوصول، والتعلم الآلي، والإعاقة البصرية، والنظارات الذكية، والمساعدة البصرية، وتحليل الصور في الوقت الحقيقي

Liste Des Figures

Chapitre I : Vision artificielle

| | |
|--|----|
| Figure.I.1. Vision globale de système à base de TI..... | 4 |
| Figure.I.2. Numérisation..... | 6 |
| Figure.I.3. Image numériques..... | 7 |
| Figure.I.4. Pixels et niveaux de gris..... | 8 |
| Figure.I.5. Synthèses additives des couleurs..... | 9 |
| Figure.I.6. Image couleurs..... | 10 |
| Figure.I.7. Histogramme..... | 11 |
| Figure.I.8. Exemples d'apprentissage supervisé..... | 15 |
| Figure I.9. L'apprentissage non-supervisé..... | 16 |
| Figure. 10. Représentation de l'observation..... | 7 |
| Figure.I.11. Apprentissage automatique et apprentissage profond..... | 18 |
| Figure.I.12. La relation entre l'intelligence artificielle, la ML et l'apprentissage profond..... | 18 |
| Figure.I.13. La différence de performance entre l'apprentissage profond et la plupart des algorithmes de ML dépend de la quantité de données..... | 21 |
| Figure.I.14. Architecture des réseaux neuronaux profonds..... | 23 |
| Figure.I.15. Le perceptron..... | 24 |
| Figure.I.16. Structure du perceptron multicouche (MLP)..... | 25 |
| Figure.I.17. Graphique de la fonction sigmoïde..... | 25 |
| Figure.I.18. Graphique de la fonction Tanh..... | 26 |
| Figure.I.19. ReLu Graph..... | 26 |
| Figure.I.20. Les réseaux neuronaux récurrents comportent des boucles..... | 29 |
| Figure.I.21. Un réseau neuronal récurrent s'est développé..... | 30 |
| Figure.I.22. Le problème des dépendances à long terme-1..... | 30 |

| | |
|--|----|
| Figure.I.23. Le problème des dépendances à long terme-2..... | 31 |
| Figure.I.24. Le diagramme représente une porte d'entrée LSTM..... | 33 |
| Figure.I.25. Le diagramme représente une porte d'oubli LSTM..... | 33 |
| Figure.I.26. Le diagramme représente une porte d'oubli LSTM..... | 34 |
| Figure.I.27. Le diagramme représente une porte de sortie LSTM..... | 34 |
| Figure.I.28. Le diagramme représente une chaîne de trois cellules LSTM..... | 35 |
| Figure.I.29. L'architecture d'un réseau de neurone convolutionnels..... | 37 |
| Figure.I.30. L'opération de convolution..... | 37 |
| Figure.I.31. L'opération de sous échantillonnage..... | 38 |
| Figure.I.32. AlexNet Architecture..... | 39 |
| Figure.I.33. Vue comprimée de l'architecture de Google Net (version 3)..... | 40 |

Chapitre II : Méthodologie et Outils Utilisés

| | |
|--|----|
| Figure.II.1. Importation des images..... | 45 |
| Figure.II.2. Annotation Manuelle..... | 46 |
| Figure.II.3. Attribution des étiquettes (Labels)..... | 46 |
| Figure.II.4. Exportation au format YOLO..... | 47 |
| Figure.II.5. Comparaison de Nombre de paramètres dans différentes versions de L'architecture YOLO b) Vitesse d'inférence pour chaque version YOLO..... | 49 |
| Figure.II.6. La disposition de la carte Raspberry Pi 5 | 52 |

Chapitre III : Résultats et discussion

| | |
|---|----|
| Figure.III.1. Matrice de corrélation pour identifier les caractéristiques essentielles de l'ensemble..... | 59 |
| Figure.III.2. Le logo Python..... | 61 |
| Figure.III.3. Le logo de Jupyter Notebook..... | 62 |
| Figure.III.4. Interface logicielle Visual Studio Code..... | 63 |

| | |
|---|----|
| Figure.III.5. keras logo..... | 64 |
| Figure.III.6. Création de fichier data.yaml..... | 65 |
| Figure.III.7. Évaluation de model..... | 66 |
| Figure.III.8. Créer le dossier « my_model » pour stocker les poids du modèle et les résultats de l'entraînement..... | 66 |
| Figure.III.9. Résultats des quatre premières époques du mode..... | 67 |
| Figure.III.10. Performances du model Yolo sur notre dataset..... | 68 |
| Figure.III.11. Ecran d'accueil de Raspbian..... | 70 |

Liste Des Tableaux

Tableau.I.1 : Données quantitatives des images.....8

Tableau.I.2 : Histoire de l'apprentissage profond..... 19

Tableau II.1 : Benchmark de performance pour yolov8.....50

Liste Des Abréviations

ANN : Réseau de Neurones Artificiel

CSI : Camera Serial Interface

CNN : Réseau de Neurones Convolutif

DL: L'apprentissage profond

DNN : Réseau de Neurones Profond

FFT : Transformée de Fourier Rapide

FNN : Réseau de Neurones à Propagation Avant

GPU: Graphics Processing Unit

LSTM : Mémoire à Long et Court Terme

ML: L'apprentissage machine

MLP : Perceptron Multi-Couches

NN: Réseau de Neurones

OCR : Reconnaissance Optique de Caractères

RVB : Rouge vert bleu

RNN : Réseau de Neurones Récurrent

SBC : Ordinateur monocarte

TNN : Réseaux de Neurones Traditionnels

TTS : Conversion du Texte en Parole

YOLO: You Only Look Once

Sommaire

| | |
|------------------------------|------|
| Remerciement | i |
| Dédicace | ii |
| Résumé (en fra, Eng) | iii |
| Résumé (en arabe)..... | iv |
| Liste des figures | v |
| Liste des tableaux..... | viii |
| Liste des abréviations | ix |
| Introduction générale..... | 1 |

Chapitre I :vision artificielle

| | |
|--|----|
| I. Introduction..... | 3 |
| I.1 Traitement d'image et vision..... | 4 |
| I.2 Acquisition et représentation des images..... | 4 |
| I.2.1 Dispositifs d'acquisition..... | 5 |
| I.2.2 Processus d'acquisition..... | 5 |
| I.3 Représentation des images..... | 7 |
| I.3.1 Types d'images numérique..... | 8 |
| I.4. Traitement et filtrage | 11 |
| I.4.1. Filtre Sobel (Détection de Contours)..... | 11 |
| I.4.2. Traitement spatial | 11 |
| I.5. Types de Filtres..... | 12 |
| I.6. Traitement Matriciel..... | 12 |
| I.7. Introduction..... | 13 |
| I.8 .L'apprentissage automatique..... | 14 |
| I.9. Principe de base..... | 14 |
| I.9.1. Types d'apprentissage dans l'apprentissage automatique..... | 14 |

| | |
|--|----|
| I.9.1.1. Apprentissage supervisé..... | 14 |
| I.9.1.2 Apprentissage non supervisé..... | 15 |
| I.9.1.3 Apprentissage par renforcement..... | 16 |
| I.10. Régression et classification..... | 16 |
| I.10.1. Classification..... | 16 |
| I.10.2. Régression..... | 17 |
| I.11. Apprentissage automatique et apprentissage profond..... | 17 |
| I.12. L'apprentissage profond..... | 18 |
| I.12.1. Histoire de l'apprentissage profond..... | 19 |
| I.12.2. Pourquoi l'apprentissage profond ?..... | 20 |
| I.12.3. Domaines d'application de l'apprentissage profond..... | 21 |
| I.13. Algorithmes D'apprentissage Profond..... | 21 |
| I.13.1. Réseaux neuronaux profonds..... | 22 |
| I.13.1.1. Perceptron..... | 23 |
| I.13.1.2. Perceptron multicouche..... | 24 |
| I.13.1.3 fonctions d'activation..... | 25 |
| I.13.1.4. Fonctions d'optimisation..... | 27 |
| I.13.1.5. optimisation des erreurs | 28 |
| I.14. Réseaux neuronaux récurrents..... | 29 |
| I.14.1 Le problème des dépendances à long terme..... | 30 |
| I.15 Mémoire à long terme (LSTM)..... | 31 |
| I.15.1 Introduction..... | 31 |
| I.15.2 Principe de la LSTM..... | 32 |

| | |
|---|----|
| I.16 Les réseaux de neurones convolutionnels..... | 35 |
| I.16.1 Pourquoi CNN ?..... | 35 |
| I.17 Les réseaux de neurones convolutionnels (CNN)..... | 35 |
| I.17.1 Couche de convolution..... | 37 |
| I.17.2. Couche de sous-échantillonnage (Pooling)..... | 38 |
| I.17.3. Couche entièrement connectée..... | 48 |
| I.17.4. CNN Architectures..... | 39 |
| I.17.4.1. AlexNet (2012)..... | 39 |
| I.17.4.2. GoogLeNet/Inception(2014)..... | 40 |
| I.18. conclusion..... | 42 |
| Chapitre II : Méthodologie et Outils Utilisés | |
| II. Introduction..... | 44 |
| II.1. Collecte des données..... | 44 |
| II.1.1. Constitution du jeu de données..... | 44 |
| II.2. Annotation avec Label Studio..... | 45 |
| II.2.1. Importation des images..... | 45 |
| II.2.2. Annotation Manuelle..... | 45 |
| II.2.3. Attribution des étiquettes (Labels)..... | 46 |
| II.2.4. Exportation au format YOLO..... | 47 |
| II.3. Modèles de détection YOLOv8..... | 47 |
| II.3.1 Détection d'objets avec YOLOv8..... | 47 |
| II.3.2. Présentation de YOLO..... | 48 |

| | |
|--|----|
| II.3.3. Evolution jusqu'à YOLOv8..... | 48 |
| II.3.4. Pourquoi YOLOv8 dans notre projet ?..... | 49 |
| II.4. Optimisation pour Raspberry Pi..... | 50 |
| II.5. OCR et Text-to-Speech..... | 53 |
| II.5.1. Module OCR..... | 53 |
| II.6. Module Text-to-Speech (TTS)..... | 53 |
| II.7. Processus global..... | 54 |
| II.8. Architecture du Pipeline..... | 54 |
| II.8.1. Architecture matérielle..... | 54 |
| II.8.2. Architecture logicielle..... | 55 |
| II.9. Workflow détaillé du système..... | 55 |
| II.9.1. Étapes du flux de travail..... | 55 |
| II.9.2. Synchronisation des modules..... | 55 |
| II.10. Conclusion..... | 56 |
| Chapitre III : Résultats et discussion | |
| III. Introduction..... | 58 |
| III.1. Système proposé..... | 58 |
| III.2. Description de l'ensemble de données..... | 59 |
| III .3. Caractéristiques..... | 60 |
| III.4 Évaluation prévue..... | 61 |
| III.4.1. Environnement de développement..... | 61 |
| III.4.1.1. Google Colab..... | 61 |
| III.4.2. Python..... | 61 |
| III.4.3. Jupyter Notebook..... | 62 |
| III.4.4. Visual Studio Code..... | 62 |

| | |
|--|----|
| III.4.5. Présentation des outils..... | 63 |
| III.4.5.1. Le logiciel..... | 63 |
| III.5. Créer un modèle..... | 64 |
| III.6. Prédiction et évaluation..... | 65 |
| III.7. Résultat du modèle..... | 67 |
| III.8. Déploiement de l'application..... | 69 |
| III.8.1 Raspberry PI..... | 69 |
| III.8. 2. Installation de la Raspberry PI..... | 70 |
| III.9. conclusion..... | 72 |
| Conclusion générale..... | 73 |
| Références Bibliographiques..... | 75 |

Introduction générale :

L'accessibilité à l'information visuelle reste un défi majeur pour les personnes malvoyantes ou aveugles, en particulier dans les environnements urbains où les indications écrites jouent un rôle fondamental (enseignes, panneaux, noms de rues, horaires, etc.). Malgré les progrès technologiques, l'autonomie de ces individus est encore fortement limitée par l'absence de solutions efficaces, discrètes et en temps réel pour interpréter le contenu visuel de leur environnement.

Dans ce contexte, l'intelligence artificielle (IA) ouvre de nouvelles perspectives grâce à l'intégration de techniques avancées en vision par ordinateur, en reconnaissance de texte, et en synthèse vocale. Le présent travail s'inscrit dans cette dynamique et vise à concevoir un prototype de lunettes intelligentes, équipées d'une caméra, capables de détecter des enseignes spécifiques (pharmacies, cabinets médicaux, laboratoires d'analyses), d'en extraire automatiquement le texte, puis de restituer cette information sous forme audio à l'utilisateur.

Pour ce faire, nous avons utilisé des algorithmes performants comme YOLOv8 pour la détection des objets visuels (enseignes), et un modèle de réseau neuronal convolutif (CNN) pour l'extraction et la classification du texte. L'ensemble du système est déployé sur une plateforme embarquée telle que le Raspberry Pi, ce qui permet un fonctionnement autonome et portable. La transformation du texte détecté en audio est assurée à l'aide de techniques de synthèse vocale adaptées.

Ce projet constitue ainsi une solution innovante à fort impact social, en alliant intelligence Artificielle et dispositifs embarqués au service de l'accessibilité visuelle. Il présente également un intérêt technique en intégrant plusieurs domaines clés de l'IA dans une application concrète et utile.

Chapitre I
Vision artificielle

I. Introduction

La vision artificielle, également appelée vision par ordinateur, est une branche de l'intelligence artificielle qui vise à permettre aux ordinateurs de percevoir, analyser et comprendre le contenu des images et des vidéos numériques, de manière similaire à la vision humaine. Elle repose sur l'utilisation de capteurs comme les caméras pour capturer des données visuelles, puis sur des algorithmes sophistiqués, souvent basés sur l'apprentissage profond, pour traiter ces données et extraire des informations pertinentes [1][2][5].

L'objectif principal de la vision artificielle est de reproduire les capacités de la vision humaine dans des machines, ce qui implique non seulement la reconnaissance d'objets, mais aussi la compréhension contextuelle et la prise de décision basée sur l'image perçue. Cette discipline scientifique et technologique englobe des tâches telles que l'acquisition, le traitement, l'analyse et la compréhension des images, permettant ainsi d'automatiser des processus complexes dans divers domaines [4].

Les applications de la vision artificielle sont vastes et touchent de nombreux secteurs. Par exemple, dans le domaine médical, elle aide au dépistage et au diagnostic en analysant des images radiologiques pour détecter des anomalies comme des tumeurs ou des hémorragies[3][5]. Dans l'industrie, elle est utilisée pour le contrôle qualité, la détection de défauts et la robotique automatisée [1][5]. De plus, la vision artificielle est essentielle dans le développement des véhicules autonomes, où elle permet la navigation, la reconnaissance des panneaux de signalisation et la détection d'obstacles [3]. Elle est aussi employée dans la cybersurveillance via la reconnaissance faciale, soulevant cependant des questions éthiques [2].

Les progrès récents, notamment dans le domaine de l'apprentissage profond et des réseaux neuronaux, ont considérablement amélioré les performances des systèmes de vision artificielle, leur permettant d'égaliser voire de surpasser les capacités humaines dans certaines tâches spécifiques [1][6]. Ces avancées sont soutenues par la disponibilité massive de données d'images, la puissance accrue des processeurs graphiques (GPU) et les infrastructures de cloud informatique.

I.1 Traitement d'image et vision

La vision, ou vision artificielle, va plus loin en utilisant ces techniques pour permettre à une machine de comprendre et d'interpréter automatiquement le contenu des images, ouvrant la voie à des applications comme la reconnaissance d'objets, l'analyse de scènes ou la prise de décision automatisée [3]. Le traitement d'image constitue donc une étape clé dans la chaîne de la vision artificielle, en préparant et en extrayant les données nécessaires à une interprétation intelligente des images [3][7].

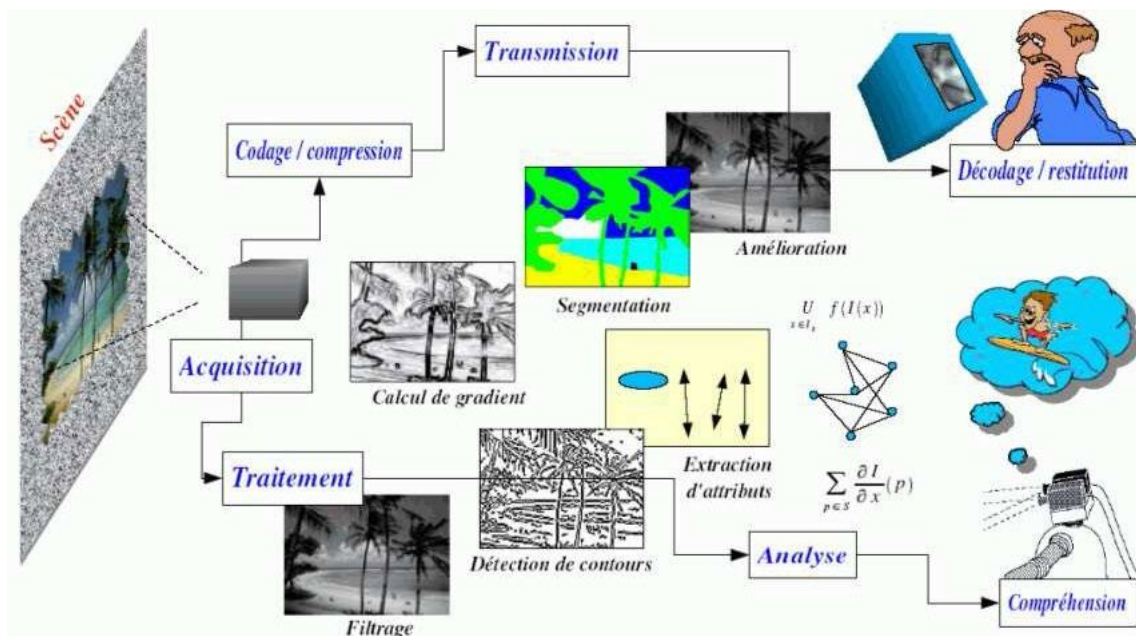


Fig.I.1. Vision globale de système à base de TI

I.2 Acquisition et représentation des images

L'acquisition est la première étape qui consiste à capturer une image du monde réel à l'aide de dispositifs optiques et électroniques. Cette étape transforme une scène physique en données numériques exploitables par un ordinateur.[8]

I.2.1 Dispositifs d'acquisition

- **Caméras numériques**

Le composant central du dispositif d'acquisition est une caméra numérique fixée sur la monture des lunettes. Cette caméra est chargée de capter en continu l'environnement devant l'utilisateur. Elle doit offrir une résolution suffisante pour permettre la détection d'enseignes et l'extraction de texte avec un niveau de détail acceptable.

Parmi les critères de choix, on note :

Une résolution minimale de 720p (HD) pour assurer la lisibilité du texte.

Une bonne gestion de l'exposition et de la luminosité, essentielle pour des conditions d'éclairage. Une compatibilité matérielle avec le Raspberry Pi, qui est le cœur du système embarqué. Dans notre prototype, une caméra compatible CSI (Camera Serial Interface), telle que la Pi Camera V2, a été utilisée. Elle permet une intégration simple avec le Raspberry Pi, un faible encombrement, et une consommation réduite.

I.2.2 Processus d'acquisition

- **Capture optique (Préparation de la scène)**

La scène visuelle est captée par la caméra intégrée aux lunettes. Cette étape produit une image brute, souvent affectée par divers facteurs tels que l'éclairage, le flou, ou le bruit. L'image capturée contient toutes les informations nécessaires, mais sous forme non structurée.

- **Conversion photoélectrique (Numérisation physique)**

Le capteur de la caméra transforme la lumière reçue en signaux électriques analogiques. Du point de vue du traitement d'image, cette phase marque le passage du monde réel vers le monde des données numériques, où chaque point lumineux (pixel) devient une information exploitable.

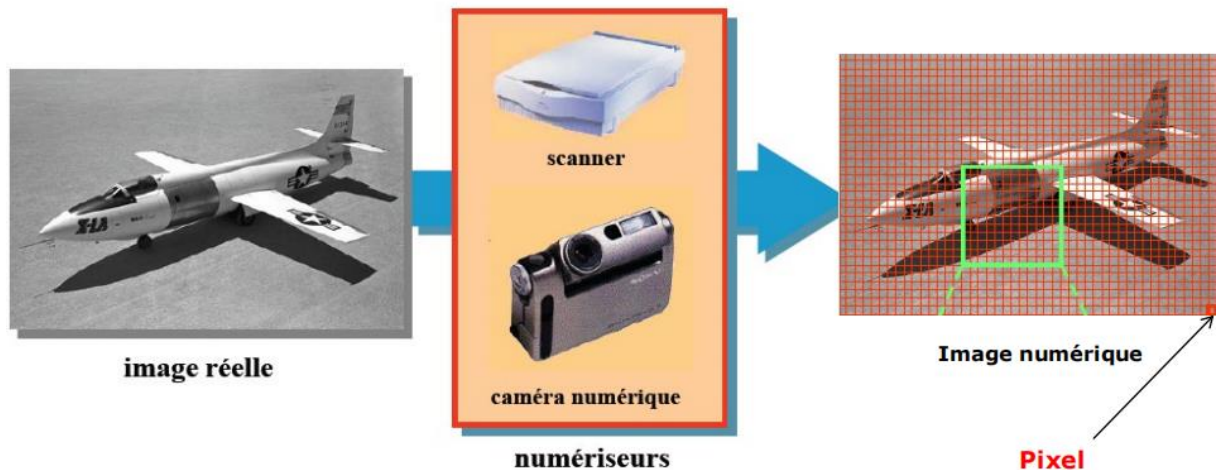
- **Conversion analogique-numérique (Formation de l'image numérique)**

Les signaux analogiques sont ensuite convertis en valeurs numériques discrètes, formant une matrice d'intensités (pixels) qui représente l'image dans l'espace informatique. Cette image numérique est l'entrée du pipeline de traitement, où :

Chaque pixel est identifié par ses coordonnées (x, y) et ses niveaux de couleur (RVB ou niveaux de gris),

L'image est prête à subir des traitements comme le filtrage, la segmentation, la détection d'objets (YOLO), ou encore la reconnaissance de texte.

- **Numérisation**



Numérisation = Échantillonnage + Quantification

Fig.I.2. Numérisation

La conversion d'une image optique en une représentation numérique exploitable par des systèmes informatiques repose sur deux étapes fondamentales : l'échantillonnage spatial et la quantification. Ces deux opérations constituent le cœur du processus de numérisation d'une image analogique, préalable indispensable à toute forme de traitement par vision par ordinateur.

- **Échantillonnage spatial**

L'échantillonnage spatial est l'opération par laquelle une image continue est discrétisée dans l'espace. Concrètement, cela consiste à diviser l'image en une matrice régulière de petits éléments appelés pixels (picture elements). Chaque pixel occupe une position spatiale unique (x, y) et représente la valeur moyenne de l'intensité lumineuse dans sa zone.

La résolution spatiale d'une image est définie par le nombre de pixels en largeur et en hauteur. Une résolution plus élevée permet de capter un plus grand nombre de détails visuels, ce qui améliore la précision dans les tâches de détection, de reconnaissance ou d'interprétation d'images.

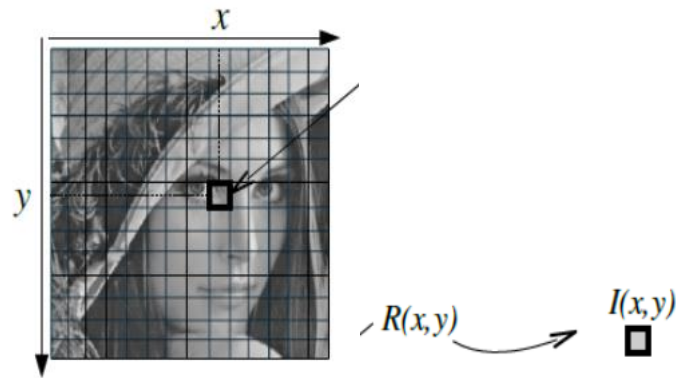


Fig.I.3. Image numériques

- **Quantification**

La quantification est l'étape qui suit l'échantillonnage. Elle consiste à attribuer à chaque pixel une valeur numérique discrète représentant son intensité lumineuse (dans une image en niveaux de gris) ou sa couleur (dans une image RGB).

Cette opération implique l'utilisation d'un convertisseur analogique-numérique (CAN) qui transforme les signaux analogiques reçus du capteur en valeurs numériques codées sur un nombre fixe de bits.

Par exemple :

Une quantification sur 8 bits permet 256 niveaux de gris (valeurs de 0 à 255).

En couleur, chaque pixel possède trois composantes (rouge, vert, bleu), chacune codée sur 8 bits, soit un total de 24 bits par pixel.

Une quantification adéquate est essentielle pour assurer la fidélité visuelle de l'image numérique et garantir la performance des traitements ultérieurs

I.3 Représentation des images

Une fois acquise, l'image est représentée sous forme numérique, généralement comme une matrice de pixels.

La représentation la plus élémentaire correspond à l'image binaire pour laquelle chaque Pixel ne peut prendre qu'une valeur parmi deux autres. Pour les images monochromes, chaque pixel peut prendre une valeur parmi N. N correspond généralement à une puissance de 2, ce qui facilite la représentation de l'image en machine. Par exemple, pour une image en niveau de gris chacun des pixels peut prendre une valeur parmi 256. Sa valeur est alors codée par un octet

De donnée. Une image est constituée par une matrice X lignes et Y colonnes de pixels chacun codé par x bits [9]. Les données quantitatives liées à la représentation des images sont représentées dans le tableau suivant :

| | |
|---------|-------------------------------------|
| 1 bit | 2 couleurs (noir & blanc) |
| 4 bits | 16 couleurs |
| 8 bits | 256 couleurs ou niveaux de gris |
| 16 bits | 65536 couleurs |
| 24 bits | 16777216 couleurs (vraies couleurs) |
| 32 bits | 4 294 967 296 couleurs |

Tableau.I.1. Données quantitatives des images

I.3.1 Types d'images numérique

- Image en niveaux de gris (ou image en tons de gris)

Une image en niveaux de gris est composée de pixels ayant chacun une seule valeur représentant une intensité lumineuse.

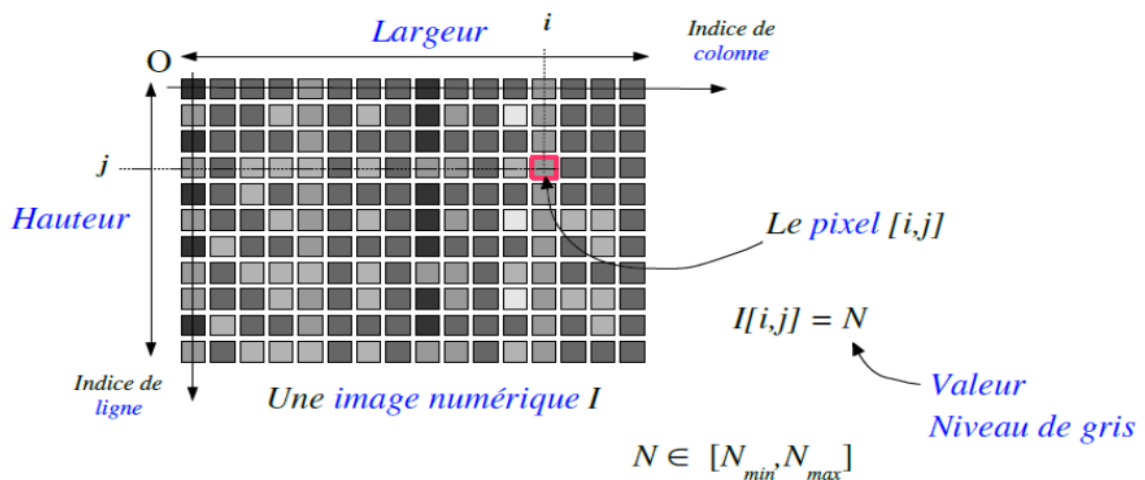


Fig.I.4. Pixels et niveaux de gris

On passe en niveaux de gris en considérant comme seule composante la luminosité. Il existe plusieurs manières de convertir une image RGB en niveaux de gris. La plus simple est de faire:

$$gris = \frac{rouge+vert+bleu}{3} \quad (I.1)$$

Cela équivaut aussi à affecter la couleur en niveau de gris à chacune des trois composantes RGB. L'idéal est de faire ressortir la luminosité d'un pixel. Celle-ci vient principalement de la présence de la couleur verte. On emploie généralement les coefficients suivants :

$$Gris = 0,299 \cdot rouge + 0,587 \cdot vert + 0,114 \cdot bleu$$

Dans ce cas on dispose d'une échelle de teintes de gris, et la plupart du temps on dispose de 256 niveaux de gris avec [10] :

0 — noir,127 — gris moyen,, 255 — Blanc

Certaines images peuvent être codées sur deux octets ou plus (certaines images médicales, des images astronomiques,) ce qui peut poser des problèmes dans la mesure où les systèmes de traitement d'images courants supposent l'utilisation des pixels d'un octet. Notons au passage que l'humain standard ne reconnaît au plus que 64 niveaux de gris

- **Espace RVB / RGB**

L'espace colorimétrique **RGB** (Rouge, Vert, Bleu) est un modèle **additif** basé sur le mélange des trois couleurs primaires pour reproduire une large gamme de couleurs. Il est principalement utilisé dans les dispositifs numériques (écrans, caméras, scanners) en raison de sa compatibilité avec la perception humaine des couleurs.

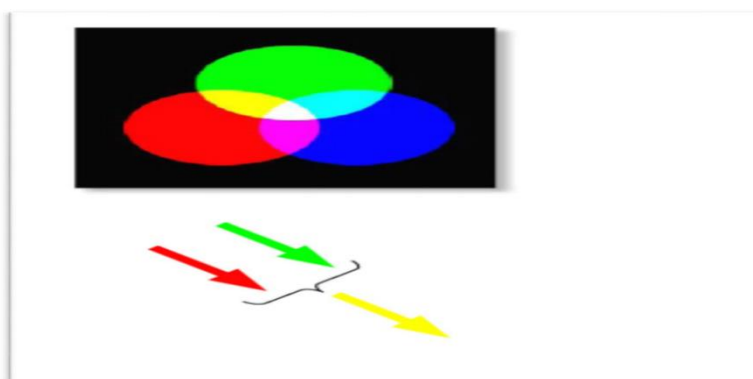


Fig.I.5. Synthèses additives des couleurs

Exemple :

Basé sur la synthèse additive des couleurs → mélange de 3 composantes Souvent, chaque couleur sur 8bits donc $I = [0,255]^3$

Utilisé par les écrans d'ordinateurs

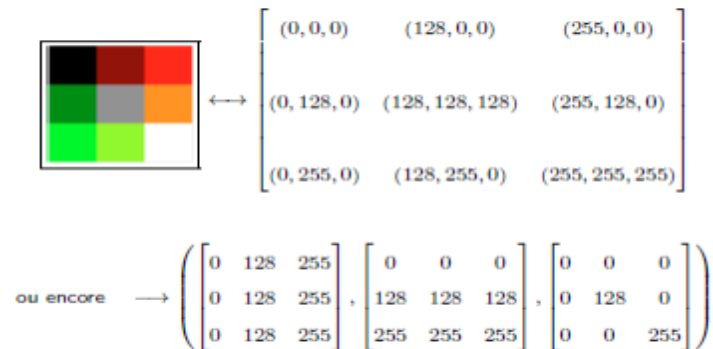


Fig.I.6. Image couleurs

- **Contraste**

Le contraste mesure la variation d'intensité entre les pixels clairs et sombres dans une image :

$$c = \frac{\max_{i,j}[f(i,j)] - \min_{i,j}[f(i,j)]}{\max_{i,j}[f(i,j)] + \min_{i,j}[f(i,j)]} \quad (I.2)$$

$\max_{i,j}[f(i,j)]$: Intensité maximale dans l'image.

$\min_{i,j}[f(i,j)]$: Intensité minimale.

- Un contraste élevé indique une grande différence entre les zones claires et sombres (ex : texte noir sur fond blanc).

- **Histogramme**

L'histogramme d'une image est une fonction discrète $H(k)$ qui décrit la **répartition des pixels** en fonction de leur niveau de gris k (généralement codé sur 8 bits : $k \in [0,255]$)

$H(k)$ =Nombre de pixels de niveau de gris [11].

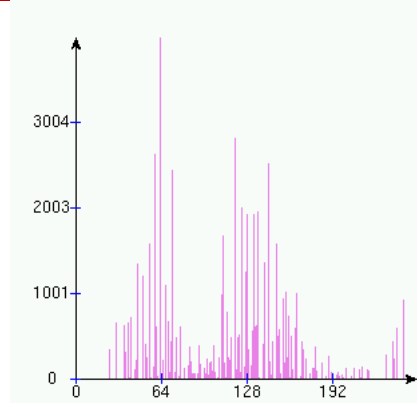


Fig.I.7. Histogramme

I.4. Traitement et filtrage

I.4.1. Filtre Sobel (Détection de Contours)

Le filtre **Sobel** est utilisé pour détecter les **bords** dans une image en calculant le gradient des intensités.

○ Équations des Noyaux (Kernels)

- ✓ Gradient horizontal (détection des bords verticaux) :

$$G_x = \begin{matrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -2 & 0 & +1 \end{matrix} \quad (I.3)$$

- ✓ Gradient vertical (détection des bords horizontaux) :

$$G_y = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & 1 \end{matrix} \quad (I.4)$$

- ✓ Calcul du Gradient (Magnitude et Direction)

$$G = \sqrt{G_x^2 + G_y^2} \quad \theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (I.5)$$

I.4.2. Traitement spatial

Le **traitement spatial** désigne l'ensemble des opérations appliquées directement sur les pixels d'une image, en fonction de leurs valeurs et de leurs positions. Contrairement au traitement

Fréquentiel (qui utilise des transformations comme la FFT), le traitement spatial travaille sur le domaine **direct** de l'image.[12]

- **Filtrage Spatial**

Opération qui modifie les pixels en fonction de leurs voisins, à l'aide d'un **noyau (kernel)**.

I.5. Types de Filtres

- **Filtres Passe-Bas (Lissage) :**

Flou Gaussien : Réduit le bruit en atténuant les hautes fréquences.

Moyenneur : Remplace chaque pixel par la moyenne de son voisinage.

- **Filtres Passe-Haut (Détection de Contours) :**

Sobel : Détecte les contours horizontaux/verticaux.

Laplacien : Détecte les contours dans toutes les directions.

I.6. Traitement Matriciel

Le traitement matriciel consiste à utiliser des opérations matricielles pour manipuler les images numériques, qui sont naturellement représentées sous forme de matrices.[15]

- **Opérations Matricielles Fondamentales**

- **Convolution (Filtrage)**: Opération clé pour appliquer des filtres (flou, détection de contours)

Partie 2 : L'apprentissage automatique

I.7 INTRODUCTION

Au cours de la dernière décennie, l'intelligence artificielle (IA) est devenue un sujet populaire tant au sein de qu'en dehors de la communauté scientifique ; une multitude d'articles dans les revues technologiques et non technologiques traitent de l'apprentissage machine (ML), de l'apprentissage profond (DL) et de l'IA. La confusion règne toujours sur à propos de l'IA, de l'apprentissage automatique et de l'apprentissage profond. Ces termes sont fortement interconnectés, mais ne sont pas interchangeables. Dans cette revue, nous n'utilisons pas (ou essayons d'utiliser) de jargon technique pour mieux expliquer ces concepts à un public clinique.

En 1956, un groupe d'informaticiens a suggéré que les ordinateurs pouvaient être programmés pour penser et argumenter, "que chaque aspect de l'apprentissage ou toute autre propriété de l'intelligence [pouvait] être décrit d'une manière si précise qu'une machine [pouvait] être amenée à le simuler". Ils ont qualifié ce principe d'"intelligence artificielle". En termes simples, l'IA est un domaine axé sur l'automatisation des tâches intellectuelles normalement accomplies par les humains, La ML et la DL sont des méthodes spécifiques permettant d'atteindre cet objectif. Cela signifie qu'elles font partie du domaine de l'IA, mais l'IA comprend des approches qui n'incluent aucune forme d'"apprentissage". Ainsi, le sous-domaine connu sous le nom de symbolic AI se concentre sur le codage en dur de règles (c'est-à-dire l'écriture explicite) pour chaque scénario possible dans un domaine d'intérêt particulier. Ces règles, écrites par des personnes, proviennent d'une connaissance a priori du sujet particulier et de la tâche à accomplir. Par exemple, si vous programmez un algorithme pour moduler la température ambiante d'un bureau, il ou elle sait probablement déjà quelles températures conviennent aux gens pour travailler et il ou elle refroidira la pièce, lorsque les températures dépassent un certain seuil et la réchauffera lorsqu'elles descendront en dessous d'un seuil inférieur à programmer. Bien que l'IA symbolique puisse résoudre des problèmes logiques clairement définis, elle échoue souvent dans les tâches qui nécessitent une reconnaissance des formes plus poussée, comme la reconnaissance vocale ou la classification d'images. C'est dans ces tâches plus compliquées que les méthodes d'apprentissage automatique et d'apprentissage profond donnent de bons résultats. Ce rapport résume les méthodes d'apprentissage automatique et d'apprentissage profond à l'intention du public qui n'a pas de connaissances approfondies en programmation informatique [16].

I.8 L'apprentissage automatique

L'apprentissage automatique est l'un des domaines de l'intelligence artificielle. Un algorithme d'apprentissage automatique a pour objectif d'analyser les informations disponibles à partir d'un grand nombre de données statistiques pour apprendre à réaliser une étude sans avoir été explicitement programmé pour cela en amont.

L'apprentissage automatique a connu un véritable succès ces dernières années. Avec la croissance exponentielle du nombre de données numériques disponibles, nous avons besoin d'utiliser de nouvelles méthodes d'analyse, et les méthodes dites d'apprentissage automatique correspondent à ce besoin. Dans cet article, nous allons étudier plusieurs types d'algorithmes d'apprentissage automatique, et dans cette partie, nous expliquerons différentes généralités communes à tous. Nous commencerons par une présentation rapide des autres familles d'apprentissage, puis du fonctionnement général d'un algorithme d'apprentissage automatique. Nous énumérerons ensuite les principaux avantages et inconvénients, en nous intéressant de plus près au sur-apprentissage. Enfin, nous expliquerons le modèle d'apprentissage automatique le plus simple, ainsi que son utilisation dans ce mémoire [17].

I.9 Principe de base

I.9.1. Types d'apprentissage dans l'apprentissage automatique

I.9.1.1. Apprentissage supervisé

L'apprentissage supervisé vise à créer une fonction prédictive pour l'une des variables de notre base de données en fonction d'autres variables. La variable que l'on veut prédire est appelée "variable à expliquer", les autres variables utilisées pour guider la prédiction sont les "variables explicatives". L'algorithme tente d'apprendre, en parcourant la base de données disponible, les différents liens de causalité entre les variables explicatives et la variable à expliquer. Une fois notre modèle créé, il fait correspondre à chaque combinaison possible de variables explicatives, une prédiction de la variable correspondante à expliquer. Pour cela, il doit regrouper les individus de la base en sous-groupes en maximisant l'homogénéité de la variable à définir. Encore faut-il que les groupes ne soient séparés qu'en fonction de leurs variables explicatives. Il existe deux catégories d'apprentissage supervisé :

- Algorithmes de régression, lorsque la variable à expliquer est quantitative. Dans ce cas,

La prédiction est une valeur.

- Algorithmes de classification, lorsque la variable à expliquer est qualitative. Dans ce cas,

La prédiction est souvent une probabilité d'appartenance aux différentes modalités de la variable en question [25].

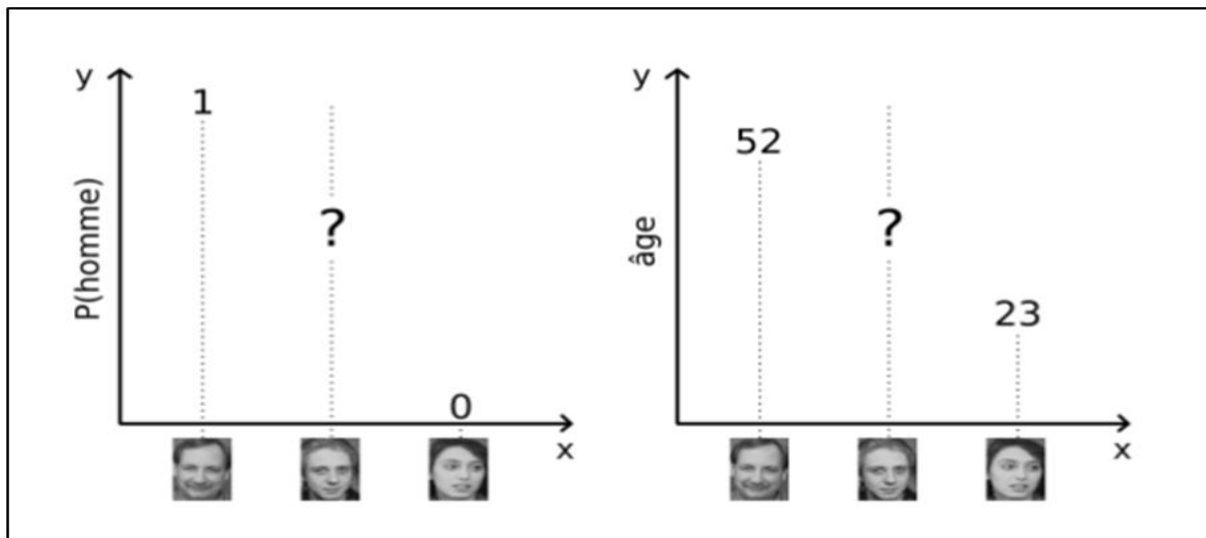


Figure.I.8. Exemples d'apprentissage supervisé

Classification (à gauche) pour prédire le sexe, et régression (à droite) pour prédire l'âge, Les photos aux deux extrémités de l'axe des x sont des exemples d'entraînement pour lesquels l'étiquette est connue, tandis que la photo du milieu est celle pour laquelle on veut obtenir une prédiction

I.9.1.2 Apprentissage non-supervisé

L'apprentissage non supervisé (Figure I.9) désigne la situation d'apprentissage automatique où les données ne sont pas étiquetées (par exemple, étiquetées "balle" ou "poisson"). Il s'agit donc de découvrir la structure derrière ces données non étiquetées. Étant donné que les données ne sont pas étiquetées, l'algorithme ne peut pas calculer de manière fiable la note de passage. Les méthodes non supervisées présentent ainsi une auto-organisation, capturant des modèles sous forme de densités de probabilité ou, dans le cas des réseaux de neurones, sous forme de combinaisons de poids de machine et de préférences de caractéristiques neuronales codées dans les activations.

L'apprentissage non supervisé c'est une branche de l'apprentissage automatique s'intéressant aux problèmes qui peuvent être formalisés par : Étant donné n est décrit dans un espace X , qui est un problème d'apprentissage d'une fonction sur X Vérifiez observations $\{x_i\}_{i=1, \dots, n}$ certaines propriétés [18]

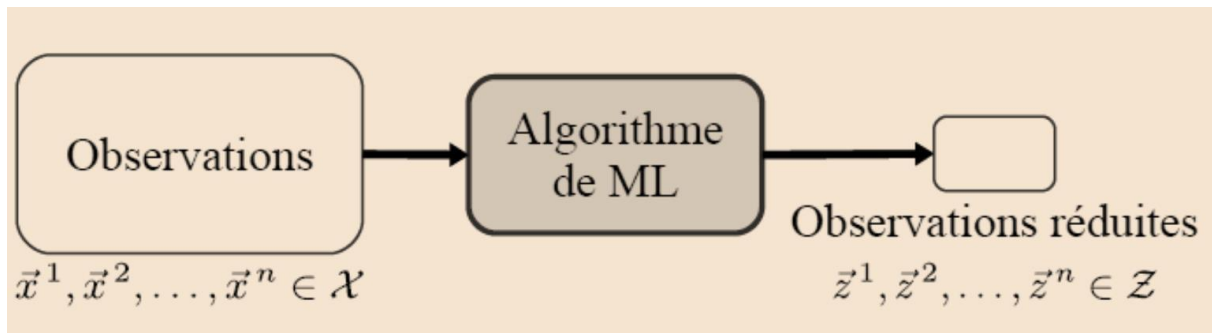


Fig.I.9. L'apprentissage non-supervisé

I.9.1.3 Apprentissage par renforcement

Dans ce type d'étude, chaque statistique marquée et non marquée peut être utilisée pour façonner les connaissances indispensables. Le cadre reçoit une récompense pour chaque prévision bonne ou incorrecte. En fonction de la récompense, la prévision suivante doit être générée. Au moment où de nouvelles informations lui sont communiquées, le système passe du temps à tomber sur l'excellente méthode d'exécution ou à faire partie de plusieurs méthodes d'exécution pour les prévisions et à attendre la récompense. Lorsque la récompense reçue s'avère être la plus élevée avec reconnaissance des récompenses précédentes pour une entrée égale, à ce moment-là, cette voie s'avère être agréable L'acquisition de connaissances par renforcement est utilisée dans les jeux basés sur le réseau, par exemple, les échecs [19].

I.10. Régression et classification

I.10.1. Classification

La classification consiste à déterminer les instructions d'appartenance de nouveaux objets à partir d'exemples antérieurs reconnus, la variable à considérer peut donc prendre des valeurs discrètes appelées classes. Nous verrons que notre problème est multi-classes, c'est-à-dire que L'échange d'opinions peut être représenté à l'aide de plusieurs cours, en opposition à la classification binaire représentée à l'aide d'une variable de sortie à deux classes [20].

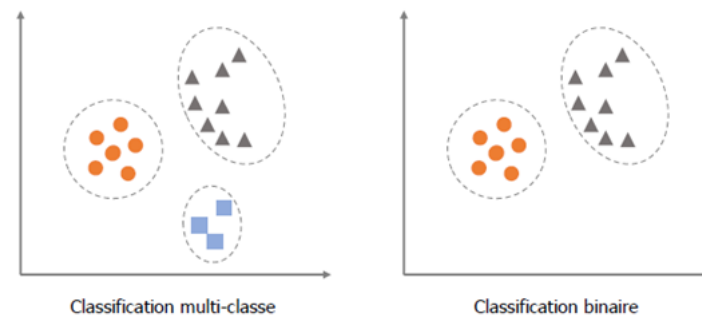


Fig. 10. Représentation de l'observation

I.10.2. Régression

La régression est utilisée pour prédire une variable continue, qui peut donc prendre n'importe quelle valeur. Les classes représentant le changement de note ont donc été considérées comme constantes et l'algorithme de régression a permis d'obtenir des valeurs décimales que nous avons finalement arrondies pour obtenir un vecteur de niveaux prédits [20].

I.11. Apprentissage automatique et apprentissage profond

L'apprentissage automatique est largement utilisé pour identifier plusieurs types d'attaques. Une méthodologie d'apprentissage automatique pourrait aider l'administrateur du réseau à effectuer les actions nécessaires pour éviter les intrusions.

Néanmoins, la plupart des méthodes traditionnelles de ML résident dans l'apprentissage superficiel et mettent généralement l'accent sur l'ingénierie et la sélection des caractéristiques. Avec l'énorme quantité de données d'intrusion qui augmente dans l'environnement réseau en temps réel, l'apprentissage externe n'est pas en mesure de résoudre efficacement le problème de classification de [30]. Une approche d'apprentissage profond a le potentiel d'extraire des représentations améliorées des ensembles de données dynamiques et est capable de créer de nombreux prototypes efficaces. G.Hinton et al [22] a introduit l'hypothèse de la DL en 2006, et au cours des années, la théorie a connu un essor spectaculaire dans le domaine de la ML. La plupart des méthodologies récentes de ML fonctionnent bien grâce à des représentations et des Caractéristiques d'entrée conçues par l'homme. Dans ML, « l'apprentissage des représentations » ou « l'apprentissage des caractéristiques » représente un ensemble de méthodes qui rendent un système capable d'apprendre automatiquement les représentations nécessaires à la détection des caractéristiques.

À partir de l'ensemble de données d'apprentissage. DL, au contraire, pourrait être considéré comme établissant à la fois l'apprentissage de la représentation et l'apprentissage automatique. Le DL vise à apprendre ensemble les caractéristiques essentielles d'avec de multiples niveaux de complexité et d'abstraction cumulés et la prédiction finale de La figure (I.11) illustre la différence fondamentale entre ML et DL, où ML traditionnel implique une sélection manuelle des caractéristiques, alors que DL emploie une sélection automatisée des caractéristiques

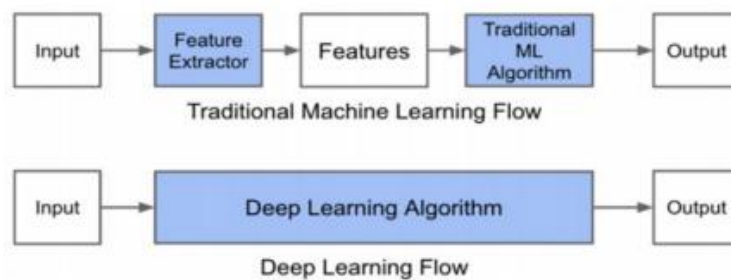


Fig.I.11. Apprentissage automatique et apprentissage profond

I.12. l'apprentissage profond

L'apprentissage profond est un nouveau domaine de recherche en ML, qui a été introduit pour rapprocher le ML de son objectif principal : l'intelligence artificielle. Il s'agit d'algorithmes inspirés de la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation pour modéliser des relations complexes entre les données.

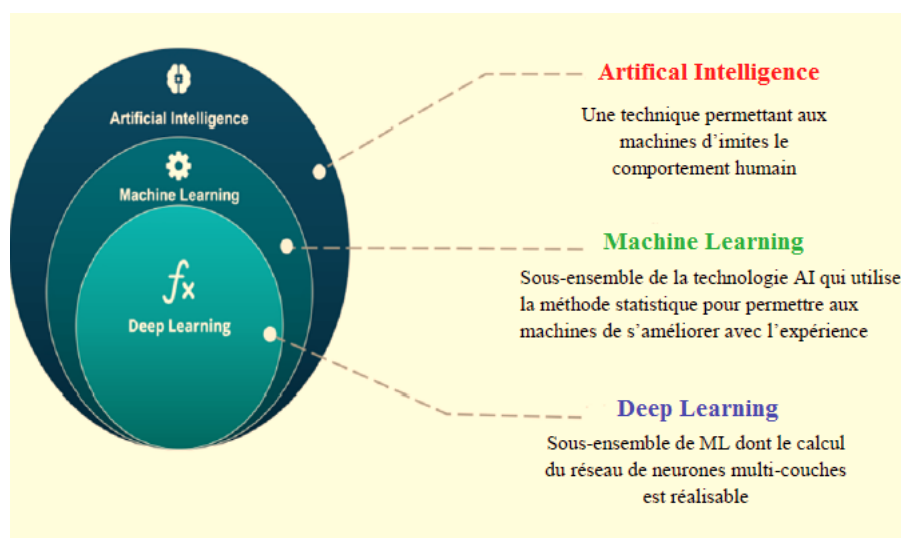


Fig.I.12. La relation entre l'intelligence artificielle, la ML et l'apprentissage profond.

L'apprentissage profond est basé sur l'idée de réseaux neuronaux artificiels et est conçu pour gérer de grandes quantités de données en ajoutant des couches au réseau. Un modèle d'apprentissage profond peut extraire des caractéristiques de données brutes par le biais de plusieurs couches de traitement consistant en de multiples transformations linéaires et non linéaires et apprendre ces caractéristiques étape par étape. L'apprentissage en profondeur se fait par étapes à travers chaque couche, avec une intervention humaine minimale. Au cours des cinq dernières années, l'apprentissage profond est passé d'un marché de niche qui n'intéressait qu'une poignée de chercheurs au domaine le plus recherché par les chercheurs. Les recherches liées à L'apprentissage profond sont désormais publiées dans les meilleures revues telles que Science, Nature et Nature Methods, pour n'en citer que quelques-unes. L'apprentissage profond a permis à de conquérir le GO, d'apprendre à conduire une voiture, de diagnostiquer le cancer et l'autisme, et même de devenir un artiste. Le terme "Deep Learning" Le terme "Deep Learning" a été ajouté pour la première fois à ML par Dechter (1986) et aux réseaux neuronaux artificiels par Aizenberg et al. (2000) [23].

I.12.1 histoire de l'apprentissage profond

Ce document présente l'histoire de l'apprentissage profond depuis Aristote jusqu'à aujourd'hui. Les différentes étapes du site sont résumées dans le tableau suivant

| Année | Contributeur | Contribution |
|--------|------------------------|---|
| 300 AC | Aristotle | introduction de l'associationnisme, début de l'histoire de les humains essayant de comprendre le brai |
| 1873 | Alexander Bain | introduction des groupements neuronaux comme premiers modèles de réseaux neuronaux |
| 1943 | McCulloch and Pitts | introduction du modèle McCulloch-Pitts (MCP) considéré comme l'ancêtre des réseaux neuronaux artificiels |
| 1949 | Donald Hebb | Considéré comme le père des réseaux neuronaux, il introduit la règle d'apprentissage de Hebb qui servira de base aux réseaux neuronaux modernes |
| 1958 | Frank Rosenblatt | Introduction du premier perceptron |
| 1974 | Paul Werbo | introduction de la retro propagation |

| | | |
|------|----------------------------|---|
| 1980 | Teuvo Kohonen | introduction des cartes auto-organisatrices |
| 1982 | John Hopfield | introduction des réseaux Hopfield |
| 1985 | Hilton and Sejnowski | introduction of Boltzmann machines |
| 1986 | Paul Smolensky | Introduction de l'Harmonium, qui sera plus tard connu sous le nom de machines de Boltzmann restreintes |
| 1986 | Michael Jordan | définition et introduction des réseaux neuronaux récurrents |
| 1990 | Yann LeCun | Introduction de LeNet et démonstration des capacités des réseaux neuronaux profonds |
| 1997 | Schuster and Paliwa | introduction des réseaux neuronaux récurrents bidirectionnels |
| 1997 | Hochreiter and Schmidhuber | l'introduction de la LSTM, qui a permis de résoudre le problème de gradient de disparition dans les réseaux neuronaux récurrents. |
| 2006 | Geoffrey Hinton | introduction du Deep Belief Network (réseau de croyances profondes) |
| 2009 | Salakhutdinov and Hinton | introduction de la machine de Boltzmann profonde |
| 2012 | Alex Krizhevsky | Présentation d'AlexNet, qui a remporté le défi ImageNet |

Tableau.I.2. Histoire de l'apprentissage profond

I.12.2 pourquoi l'apprentissage profond ?

Les algorithmes de ML fonctionnent bien pour une grande variété de problèmes. Cependant, ils n'ont pas réussi à résoudre certains problèmes importants de l'IA, tels que la reconnaissance vocale et la reconnaissance d'objets. Le développement de l'apprentissage profond a été motivé en partie par l'échec des algorithmes traditionnels à une telle tâche d'IA. Mais ce n'est que lorsque davantage de données ont été rendues disponibles grâce au Big Data

et que les objets connectés et les machines informatiques sont devenus plus puissants que le potentiel réel de Deep Learning a été compris.

L'une des différences significatives entre l'apprentissage profond et les algorithmes ML traditionnels est qu'il s'adapte bien. Plus la quantité de données fournies est importante, meilleure est la performance d'un algorithme de Deep Learning. Contrairement à plusieurs algorithmes classiques de ML qui ont une borne supérieure à la quantité de données qu'ils peuvent recevoir, parfois appelée "performance plateau", les modèles de Deep Learning n'ont pas de telles limites (théoriquement), et ils sont même allés jusqu'à surpasser les performances humaines dans des domaines tels que le traitement d'images [24].

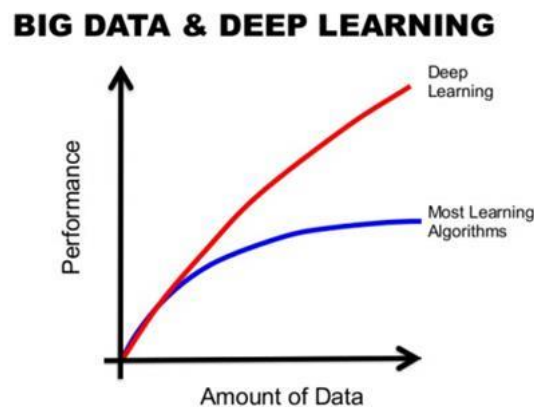


Fig.I.13. La différence de performance entre l'apprentissage profond et la plupart des algorithmes de ML dépend de la quantité de données.

I.12.3 Domaines d'application de l'apprentissage profond

L'apprentissage en profondeur investit progressivement notre vie quotidienne :

- Reconnaissance vocale
- Marquage automatique des morceaux de musique
- Synthèse vocale avancée
- La conception de nouvelles molécules pharmaceutiques

Toutes ces applications utilisent désormais des techniques de Deep Learning [24]

I.13. Algorithmes d'apprentissage profond

Il existe un grand nombre de variantes d'architectures profondes. La plupart d'entre elles sont dérivées de certaines architectures parentales originales. Il n'est pas toujours possible de comparer les performances de toutes les architectures, car elles ne sont pas toutes évaluées sur

les mêmes ensembles de données. L'apprentissage profond est un domaine en pleine expansion, et de nouvelles architectures, variantes ou algorithmes apparaissent chaque semaine.

I.13.1. Réseaux neuronaux profonds

Les réseaux neuronaux profonds (DNN), également connus sous le nom de réseaux neuronaux (NN), sont des modes de maîtrise des dispositifs informatiques qui sont stimulés par le cerveau humain. Les premières communautés neuronales ont été développées par le Neurophysiologiste Warren McCulloch et le mathématicien Walter Pitts en 1943. Ils ont expliqué comment les neurones pourraient fonctionner. Les réseaux neuronaux se composent d'une couche de neurones d'entrée ou d'alertes, qui peuvent être différentes différentes valeurs caractéristiques, d'une couche de sortie où le résultat de la communauté est obtenu, et d'une quantité de couches cachées entre la couche de sortie et la couche de sortie. Quantité de couches cachées entre les couches d'entrée et de sortie. En outre, chaque couche comporte quelques neurones ou plusieurs.

Les alertes d'entrée sont transmises au réseau, couche par couche, par l'intermédiaire des connexions pondérées, pour atteindre finalement la couche de sortie. Des connexions pondérées pour, au final, atteindre la couche de sortie. Au niveau de certains neurones, une non linéaire peut être déclenchée. L'objectif de l'acquisition de connaissances sur la méthode est de découvrir les poids qui permettraient à la couche neuronale de s'adapter à l'environnement. les poids qui permettraient à la communauté neuronale d'afficher un comportement favorable. C'est le cas Perceptron multicouche (MLP), également appelé Feedforward Neural Networks (FNN). Une structure acceptée des réseaux neuronaux artificiels est présentée dans la figure (I.14) Malgré le fait que les réseaux neuronaux de type feedforward ont été efficacement et ont obtenu de meilleures performances globales dans de nombreuses tâches, ils ne tiennent pas Compte de la perspective transitoire qui caractérise les données séquentielles. La perspective transitoire qui caractérise les données séquentielles. En d'autres termes, ils ne sont plus modérément Les réseaux neuronaux ont été appliqués de manière efficace et ont permis d'améliorer les performances globales dans de nombreuses tâches. Pour remédier à ces divers Pour remédier à ces divers problèmes, les réseaux neuronaux ont évolué vers des réseaux neuronaux dits récurrents. Dans les sous-sections suivantes [37].

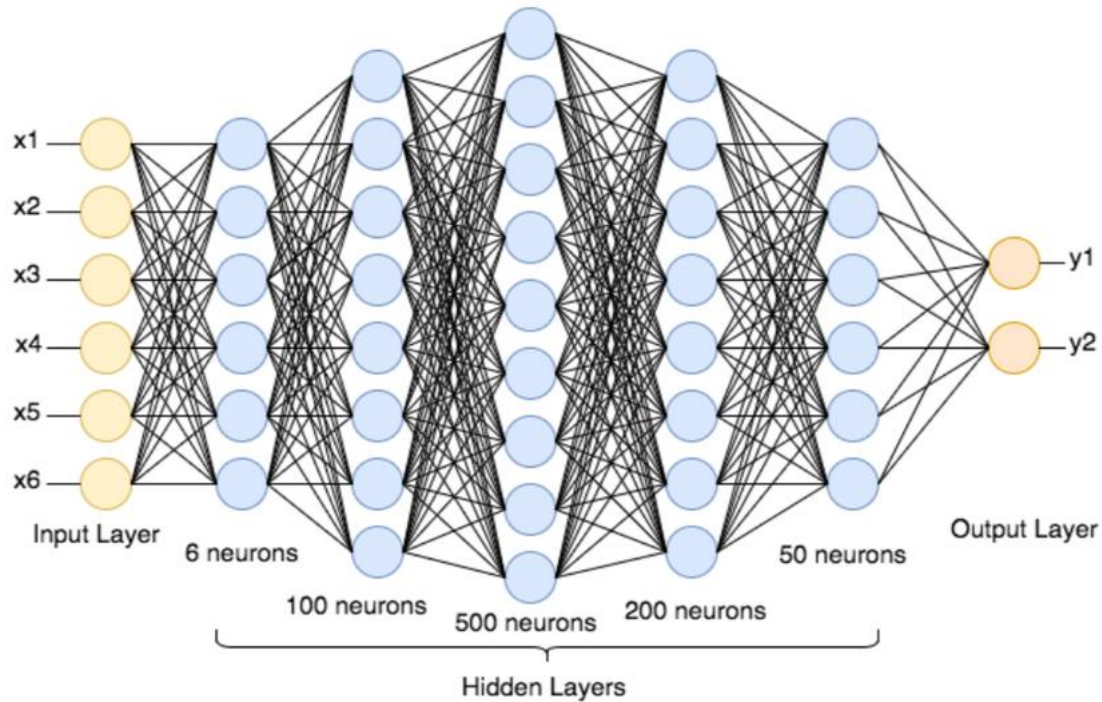


Fig.I.14. Architecture des réseaux neuronaux profonds

I.13.1.1. Perceptron

L'unité fondamentale des réseaux neuronaux est l'un des types de neurones artificiels, appelé perceptron. Un perceptron prend un nombre fixe d'entrées et produit une seule sortie. Le mode de calcul du résultat comporte essentiellement plusieurs étapes, telles que la prise d'une entrée, le calcul de la somme pondérée en introduisant des poids, un biais et une valeur de sortie. La somme pondérée en introduisant des poids, un terme de biais et en appliquant une fonction d'activation. Le processus décrit ci-dessus peut être formalisé comme suit :

Le perceptron a n entrées représentées par un vecteur d'entrée $x = (x_1, x_2, \dots, x_n)$. Chaque entrée a un poids défini par un vecteur de poids $w = (w_1, w_2, \dots, w_n)$. Par conséquent, les valeurs d'entrée pondérées sont combinées, ce qui donne la somme pondérée :

$$\varepsilon = w \cdot x = \sum w_i \cdot x_i \quad (I.6)$$

À cette étape, la fonction d'activation est appliquée à la somme pondérée pour calculer la sortie, et la somme pondérée est comparée à un seuil θ pour produire une sortie y qui est soit 0, soit 1, selon qu'elle dépasse ou non le seuil. Ainsi,

$$y = \sigma(\varepsilon) = \begin{cases} 1, & \varepsilon \geq 0 \\ 0, & \varepsilon < 0 \end{cases} \quad (I.7)$$

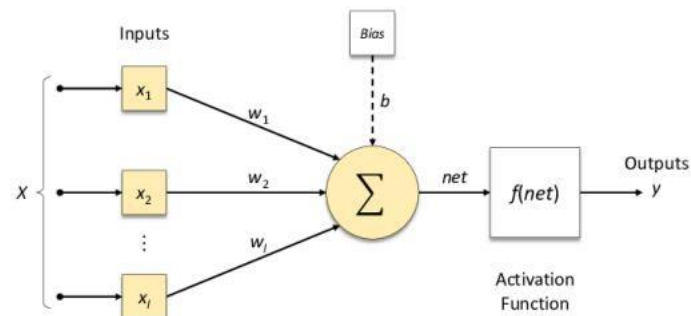


Figure.I.15. Le perceptron

I.13.1.2 Perceptron multicouche

Un perceptron multicouche (ou réseau neuronal) est une forme composée d'innombrables couches cachées de neurones où la sortie d'un neurone d'une couche deviendra l'entrée d'un neurone de la couche suivante. De plus, le résultat final d'un neurone peut également être l'entrée d'un neurone de la même couche ou du neurone des couches précédentes (c'est le cas pour les réseaux neuronaux récurrents).

Les nœuds qui ne font l'objet d'aucune connexion sont appelés neurones d'entrée. Un MLP qui doit être utilisé pour saisir des modèles de dimension n doit avoir n neurones d'entrée, un pour chaque dimension. Les neurones d'entrée sont généralement énumérés comme suit : neurone 1, neurone 2, neurone 3

- Les nœuds qui ne fournissent aucune connexion sont appelés neurones de sortie. Un MLP peut avoir plus d'un neurone de sortie. Le nombre de neurones de sortie dépend de la façon dont sont décrites les valeurs cibles (valeurs souhaitées) des modèles éducatifs.
- all nodes that neither enter neurons nor output neurons are referred to as hidden neurons.

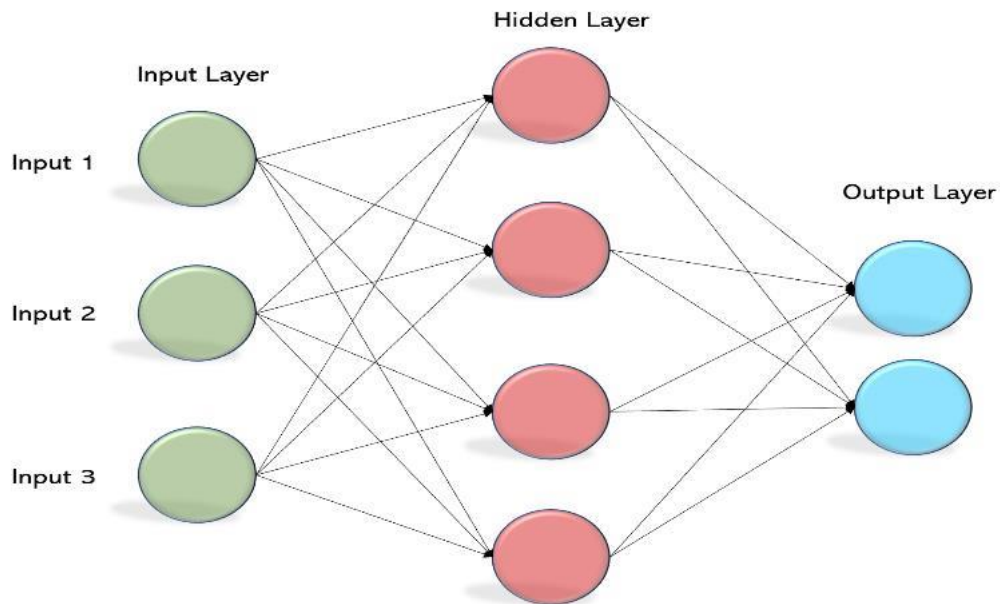


Figure.I.16. Structure du perceptron multicouche (MLP)

I.13.1.3. Fonctions d'activation

Une fonction d'activation est une fonction mathématique appliquée à une combinaison de x_i entrées. Elle introduit ainsi le principe de non-linéarité, qui permet d'adapter et de différencier les résultats. Le choix de la fonction d'activation dépend de l'application, s'il est nécessaire d'avoir des sorties binaires ou non. Dans ce travail, les fonctions Sigmoidé, Softmax, Tanh ainsi que ReLu sont utilisées [38].

- **Sigmoidé**

La fonction sigmoïde est définie par :

$$\sigma(x) = \frac{1}{1 + e^{-x}} \text{ Or } x \in \mathbb{R} \quad (\text{I.8})$$

L'objectif est de convertir la valeur d'entrée en une probabilité de 1 s'il s'agit d'un très grand nombre positif et, inversement, de 0 s'il s'agit d'un très grand nombre négatif.

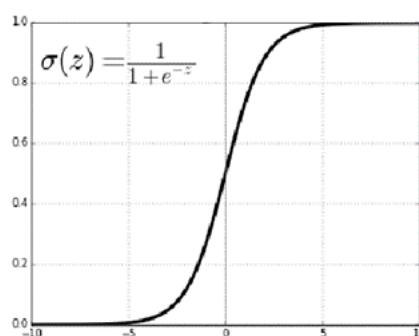


Figure.I.17. Graphique de la fonction sigmoïde.

- **Softmax**

La fonction Softmax, ou fonction exponentielle normalisée, est définie comme suit :

$$f(x)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}} \text{ for all } j \in \{1, \dots, K\}, x = \{x_1, \dots, x_k\} \text{ ou } k \in \mathbb{R} \quad (\text{I.9})$$

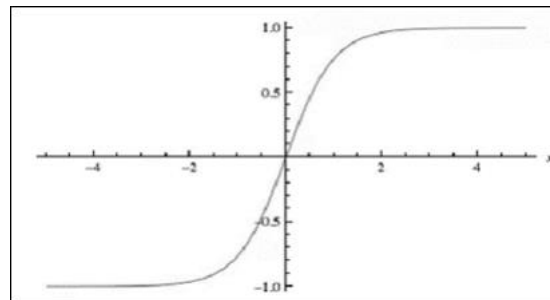
La fonction donne un vecteur de sortie composé de K nombres réels strictement positifs et d'une somme de 1.

- **Tanh**

La fonction tangente hyperbolique, notée Tanh, est définie par :

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (\text{I.10})$$

La sortie en Tanh sera stockée dans le champ $[-1,1]$.


Fig.I.18. Graphique de la fonction Tanh.

- **ReLU**

La fonction de l'unité de broyage linéaire ReLu est la fonction la plus simple, définie par :

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (\text{I.11})$$

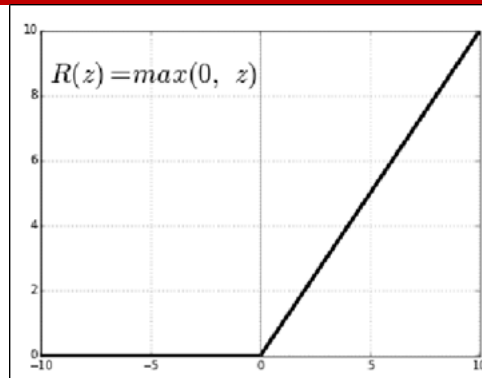


Fig.I.19. ReLu Graph.

La sortie est donc 0 si l'entrée est inférieure à 0, sinon elle est égale à l'entrée.

I.13.1.4. Fonctions d'optimisation

Généralement, les fonctions d'optimisation dans les réseaux neuronaux ont pour objectif de calculer le gradient pour trouver les valeurs optimisées (poids), et ce en modifiant les poids dans le sens opposé, ce cycle est répété jusqu'à ce que la fonction objective soit minimisée. Dans ce travail, différentes fonctions d'optimisation sont testées : Adam, Adamax, Adadelta et SGD [39].

• SGD

Le mot « stochastique » fait référence à un système ou à un processus de probabilité aléatoire. Par conséquent, dans descente stochastique du gradient, certains échantillons sont sélectionnés au hasard au lieu de l'ensemble de données de chaque itération SGD (descente stochastique du gradient), qui met à jour les paramètres pour chaque exemple de l'ensemble de données x_i et de l'étiquette y_i (label).

- Initialize with x_0 (at random)
- Répéter : $x_{t+1} = x_t - \eta \times \nabla(x_t)$ Jusqu'à convergence
- η : module la correction (η trop bas, convergence lente; η trop haut, oscillation)
- $\nabla(x_t)$: La pente au point x_t indique la direction et l'importance de la pente au Voisinage de x_t .

Cette méthode est généralement plus rapide. Cependant, en raison des mises à jour fréquentes, la convergence devient plus difficile (trouver la fonction objective minimale).

• Adam

Adam Adaptive Moment optimisation" est l'un des algorithmes les plus récents et les plus efficaces pour l'optimisation par descente de gradient. Adam calcule la moyenne exponentielle du gradient ainsi que les carrés du gradient pour chaque paramètre. Le taux d'apprentissage est ensuite multiplié par la moyenne du gradient et divisé par la racine carrée de la moyenne exponentielle des gradients. La mise à jour est ensuite ajoutée

$$v_t = \beta_1 \times v_{t-1} + (1 - \beta_1) \times g_t \quad (\text{I.12})$$

$$s_t = \beta_2 \times s_{t-1} + (1 - \beta_2) \times g_t^2 \quad (\text{I.13})$$

$$\Delta w_t = -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} \times g_t \quad (\text{I.14})$$

$$w_{t+1} = w_t + \Delta w_t \quad (\text{I.15})$$

- η : taux d'apprentissage initial.
- g_t : le gradient à l'instant t.
- v_t : la moyenne exponentielle des gradients.
- s_t : la moyenne exponentielle des carrés des gradients.
- $\beta_1, \beta_2, \epsilon$: hyperparamètres généralement initialisés à 0,90, 0,99 et 1e-10.

• Adamax

Adamax est une généralisation d'Adam pour obtenir une convergence plus stable, ou au lieu de prendre :

$$s_t = \beta_2 \times s_{t-1} + (1 - \beta_2) \times g_t^2$$

s_t become :

$$(I.16)$$

$$s_t = \beta_2^p \times s_{t-1} + (1 - \beta_2^p) \times g_t^p$$

Ou $p = \infty$.

$$(I.17)$$

• Adadelta

Adadelta propose une adaptation du gradient en fonction de son moment d'ordre 2 et de l'état de la couche corrigée du réseau et utilise une taille fixe w pour prendre en compte la somme des carrés des gradients accumulés dans le passé.

I.13.1.5. Optimisations des erreurs

Les modes de réseaux neuronaux profonds (RNP) ont plusieurs paramètres et peuvent modéliser des fonctions relativement complexes. Cette capacité est à la fois un avantage et un inconvénient. De tels prototypes s'adaptent souvent de manière excessive à l'ensemble d'apprentissage et perdent en précision et en généralisation sur l'ensemble de test. Dans la terminologie ANN, la régularisation désigne la technique de régulation des couches du réseau neuronal afin de mettre fin à la sur adaptation. L'abandon (également connu sous le nom de chance d'abandon ou de taux d'abandon) est l'approche de régularisation la plus utilisée dans les réseaux neuronaux. Au cours de l'apprentissage, les neurones de la ou des couches cachées sont choisis au hasard et sont éliminés en fonction du taux d'abandon. Plus précisément, les neurones choisis au hasard sont éliminés, c'est-à-dire que les neurones éliminés ne peuvent plus remplacer les poids, ce qui permet à la méthode d'apprentissage d'éviter le problème de surajustement [40].

I.14. Réseaux neuronaux récurrents

Les humains ne commencent pas à penser à partir de zéro à chaque seconde. En lisant cet essai, vous comprenez chaque mot en fonction de votre compréhension des mots précédents. Vous ne pas tout jeter et recommencer à penser à partir de zéro. Vos pensées ont persévérance.

Les réseaux neuronaux traditionnels (TNN) ne peuvent pas faire cela, et cela semble être une lacune importante sur le site. Imaginez par exemple que vous souhaitez classer le type D'événement qui se produit à chaque étape d'un film. On ne sait pas comment un réseau neuronal traditionnel pourrait utiliser son raisonnement sur les événements précédents du film pour les informer ultérieurement.

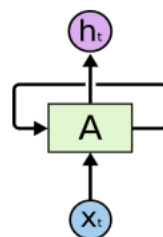
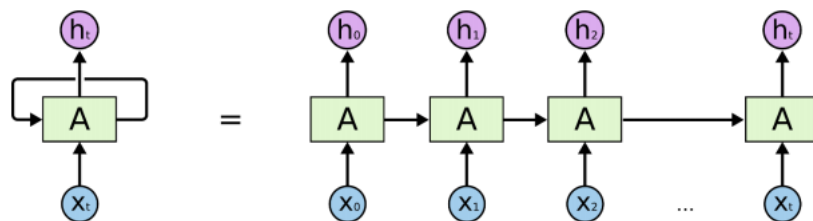


Fig.I.20. Les réseaux neuronaux récurrents comportent des boucles.

Les réseaux neuronaux récurrents (RNN) résolvent ce problème. Il s'agit de réseaux dotés de boucles qui permettent à l'information de persister. Dans la figure I.21 ci-dessus, un segment de réseau neuronal, "A", regarde une entrée X_t et fournit une valeur h_t . Une boucle permet à l'information de passer d'un stade du réseau à un autre.

Ces boucles rendent les réseaux neuronaux récurrents un peu mystérieux. Cependant, si l'on réfléchit un peu plus (), il s'avère qu'ils ne sont pas tous différents d'un réseau neuronal standard. Un réseau de neurones récurrents peut être considéré comme de multiples copies du même réseau, chaque transmettant un message à un successeur, comme le montre la figure I.22. Considérons ce qui se passe si nous démêlons la boucle


Fig.I.21. Un réseau neuronal récurrent s'est développé.

Cette nature en chaîne révèle que les réseaux neuronaux récurrents sont intimement liés aux séquences et aux listes . Ils constituent l'architecture naturelle du réseau neuronal à utiliser pour ce type de données

Ces dernières années, l'application des RRN à diverses questions a connu un succès incroyable : reconnaissance vocale, modélisation linguistique, traduction, sous-titrage d'images, etc [41].

I.14.1. Le problème des dépendances à long terme

L'un des attraits des RNN est qu'ils peuvent être capables de relier les enregistrements précédents à la tâche actuelle, comme l'utilisation des images vidéo restantes peut informer de l'appréciation de la structure contemporaine. Si les RNN devaient faire cela, ils seraient instrumentaux. Mais le peuvent-ils ? Cela dépend [41].

Parfois, il suffit d'examiner les statistiques actuelles pour mener à bien la tâche en cours. Pour exemple, réfléchir à un mannequin de langage consiste à essayer de prédire la phrase suivante en se basant totalement sur les phrases précédentes. Si nous essayons de prévoir la dernière

phrase de « les nuages sont dans le ciel », nous n'avons pas besoin d'un contexte supplémentaire - il est assez évident que l'expression suivante sera le ciel. Dans de tels cas, lorsque le fossé entre les enregistrements applicables et la région recherchée est faible, RNN peut examiner l'utilisation des données précédentes.

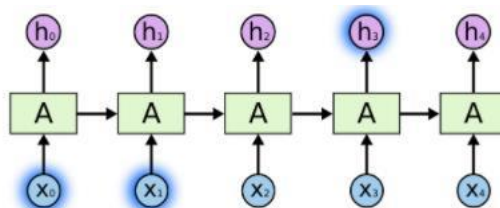


Figure.I.22. Le problème des dépendances à long terme-1

Mais il y a aussi des cas où nous avons besoin d'un contexte supplémentaire. Prenons l'exemple de qui tente de prédire la dernière phrase du texte « J'ai grandi en France... Je communique couramment en français. » Des données récentes suggèrent que la phase suivante est probablement l'identité d'une langue. Cependant, si nous voulons déterminer quelle langue, nous devons connaître le contexte de la France, depuis le début jusqu'à la fin. Il est tout à fait possible que le trou entre les statistiques applicables sur et le facteur où il est requis émerge comme très grand. Malheureusement, lorsque l'écart se creuse, les RNN deviennent incapables d'apprendre à relier les informations.

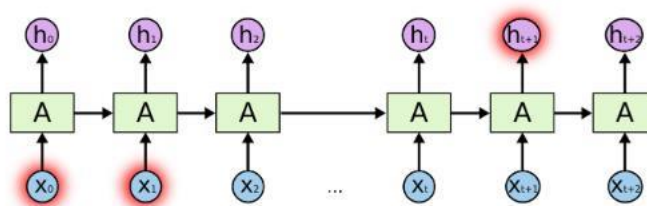


Figure.I.23. Le problème des dépendances à long terme-2.

En théorie, les RNN parviennent à gérer ces « dépendances à long terme ». Un humain devrait choisir prudemment les paramètres pour résoudre les problèmes de jouets de cette forme. Malheureusement, dans la pratique, les RNN ne semblent pas être en mesure de les découvrir. Le problème a été exploré en profondeur par Hochreiter (1991) [allemand] et Bengio et al.

(1994), qui ont trouvé des raisons essentielles pour lesquelles il peut également être problématique. Heureusement, les LSTM n'ont pas ce problème !

I.15.Mémoire à long terme (LSTM)

I.15.1.Introduction

Les réseaux de mémoire à court terme et à long terme (LSTM) sont une structure synthétique de communauté neuronale récurrente (RNN) utilisée dans la discipline de l'apprentissage profond. ... Les réseaux LSTM sont bien adaptés pour classer, traiter et faire des prédictions basées entièrement sur des données de collecte de temps, car il peut y avoir des délais d'une durée inconnue entre les opportunités vitales d'une série temporelle. LSTM est un type de communauté neuronale récurrente qui traite les problèmes de disparition et d'explosion du gradient qui peuvent se produire avec les RNN, les empêchant d'avoir de longues séquences temporelles. Par conséquent, la LSTM peut avoir une « mémoire à court terme », comme son nom l'indique. La LSTM incorpore des dispositifs extraordinaires appelés blocs de réminiscence dans la couche cachée récurrente. Les blocs de réminiscence comprennent des cellules de réminiscence avec des auto-connexions stockant la nation temporelle de la communauté en plus d'outils multiplicatifs spécifiques appelés portes pour gérer le mouvement avec le flux d'informations. Chaque bloc de réminiscence dans la structure unique contenait une porte d'entrée et une porte de sortie. La porte d'entrée contrôle le flottement des activations d'entrée dans la cellule de réminiscence. La porte de sortie contrôle le flottement de sortie des activations téléphoniques dans la relaxation du réseau. Plus tard, la porte en surplomb a été introduite sur le site dans le bloc de mémoire. La porte de négligence met à l'échelle la nation intérieure du téléphone mobile avant de l'inclure en tant qu'entrée dans le téléphone via la connexion auto-récurrente de la cellule, donc en adaptant ou en réinitialisant la réminiscence de la cellule [41]

I.15.2. Principe de la LSTM

Les LSTM possèdent des mécanismes internes appelés portes, qui sont utilisés pour réguler le flux d'informations en apprenant quelles sont les informations qu'il est utile de conserver ou d'oublier au fil du temps. Ce faisant, le réseau sera en mesure de transmettre des données pertinentes, grâce à l'état de la cellule, agissant comme une voie de transport qui transfère des informations connexes tout au long de la chaîne de séquences. Au fur et à mesure que l'état se poursuit, l'information est ajoutée ou supprimée par trois portes principales. Au départ, l'entrée

est la sortie prédite précédemment ($ht-1$) ainsi que l'entrée actuelle Xt). Ces données seront décomposées en trois flux. L'objectif est de mettre à jour l'état de la cellule de ($Ct-1$ à Ct). [45]

A- Porte d'entrée

La porte d'entrée met à jour l'état de la cellule. La fonction d'activation est la Sigmoide. Pour chaque entrée, la porte fournit une valeur de sortie comprise entre 0 et 1 et décide de la valeur à mettre à jour sur (0 signifie inutile et 1 signifie important). Ce résultat est ensuite multiplié par l'état actuel

$$\begin{aligned} f_t &= (W_t [h_{t-1}, X_t] + b_t) \\ C_t &= f_t \times C_{t-1} \end{aligned} \quad (\text{I.18})$$

avec:

- $ht-1$: La sortie à l'instant t-1
- Xt : Entrée de courant à l'instant t
- bt : Biais
- Wt : Poids
- σ : Fonction sigmoïde

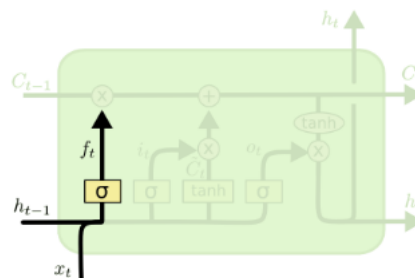


Figure.I.24. Le diagramme représente une porte d'entrée LSTM.

B- Porte de l'oubli

La porte d'oubli oublie et décide quelles informations doivent être éliminées ou conservées. Les informations de l'état précédent sont transmises à la fonction Tanh qui donne des valeurs comprises entre -1 et 1. Ensuite, pour sélectionner les caractéristiques importantes, une couche sigmoïde décidera de la valeur qui sera mise à jour.

$$f_c = \tanh(W_c [h_{t-1}] + b_c) \quad (\text{I.19})$$

$$I_t = (W_i[h_{t-1}, X_t] + b_i) \tag{I.20}$$

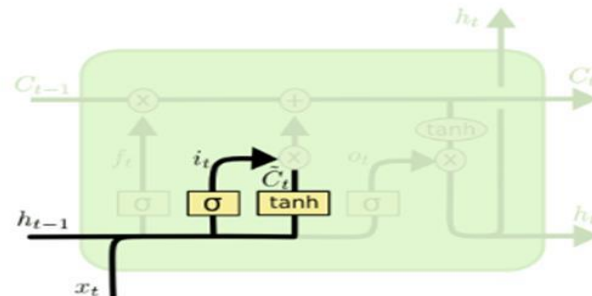


Figure.I.25. Le diagramme représente une porte d'oubli LSTM.

avec:

- tanh : Fonction d'activation de la tangente hyperbolique
- C: Une Valeur candidate

Ces deux résultats seront multipliés et ajoutés à l'état. À ce stade, l'état de la cellule est le suivant :

$$C_t = f_t \times C_{t-1} + f_c \times I_t \tag{I.21}$$

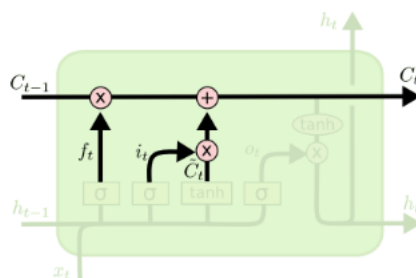


Figure.I.26. Le diagramme représente une porte d'oubli LSTM.

A- Porte de sortie

La porte de sortie transmettra l'état caché précédent et l'entrée actuelle dans une fonction sigmoïde

$$o_t = (W_o[h_{t-1}, X_t] + b_o) \tag{I.22}$$

Le nouvel état caché est donc

$$h_t = o_t \times \tanh(C_t) \tag{I.}$$

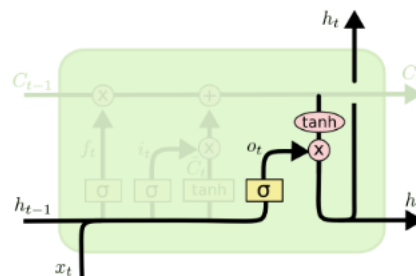


Figure.I.27. Le diagramme représente une porte de sortie LSTM.

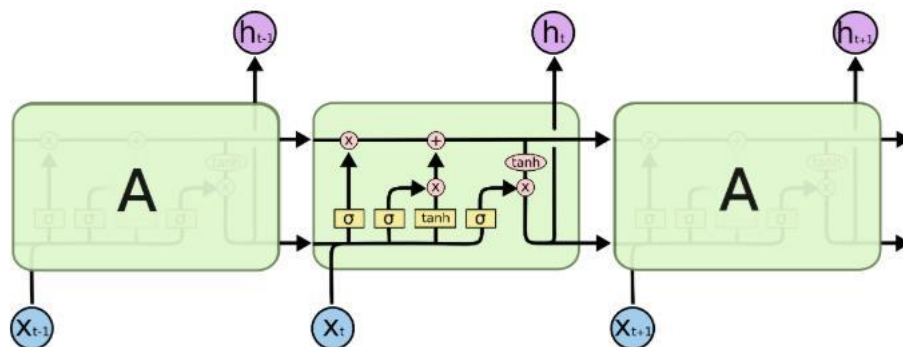


Figure.I.28. Le diagramme représente une chaîne de trois cellules LSTM

I.16. Les réseaux de neurones convolutionnels

I.16.1 Pourquoi CNN ?

CNN, comme les réseaux de neurones, sont constitués de neurones dont le poids peut être appris et les préjugés. Chaque neurone reçoit plusieurs entrées, prend une somme pondérée sur eux, passe-le à travers une fonction d'activation et répond avec une sortie. L'ensemble du réseau a une fonction de perte et tous les trucs et astuces que nous avons développés pour les réseaux de neurones s'appliquent toujours sur les CNN.[39]

La seule différence notable entre les CNN et les RN traditionnels est que Les CNN sont principalement utilisés dans le domaine de la reconnaissance de formes avec Cela nous permet de coder des caractéristiques spécifiques à l'image dans l'architecture, rendre

le réseau plus adapté aux tâches axées sur l'image. Ce choix a été motivé principalement par l'incorporation implicite d'une fonctionnalité phase d'extraction et a été utilisé avec succès dans de nombreuses applications. L'une des principales limites des formes traditionnelles d'IA est qu'elles ont tendance à avoir des difficultés avec la complexité informatique nécessaire pour calculer données d'image.

C'est la raison de la mise en œuvre de CNN, les propriétés que les CNN ont, par exemple, l'extraction de caractéristiques les rendent plus efficient quand manipulation des images.[39]

I.17. Les réseaux de neurones convolutionnels (CNN)

Les réseaux de neurones convolutionnels (CNN pour Convolutional Neural Networks) proposés initialement par Le Cun. Ce choix a été motivé principalement par ce qu'il intègre implicitement une phase d'extraction de caractéristiques et il a été utilisé avec succès dans de nombreuses applications[ref8]. Ils sont réputés pour leur robustesse aux faibles variations d'entrée, le faible taux de prétraitement nécessaires à leur fonctionnement.

Le CNN est un réseau de neurone multicouche qui est spécialisé dans des tâches de reconnaissance de forme. Ces réseaux ont été inspirés par les travaux de Hubel et Wiesel sur le cortex visuel chez les mammifères qui combine trois idées principales :

- les champs récepteurs locaux.
- les poids partagés.
- le sous-échantillonnage.

L'architecture de CNN repose sur plusieurs réseaux de neurones profonds consistant en une succession de couches de convolution et d'agrégation (pooling) est dédié à l'extraction automatique de caractéristiques, tandis que la seconde partie, composée de couches de neurones complètement connectés, est dédiée à la classification. [39] [42]

Chaque cellule des couches de convolution est connectée à un ensemble de cellules regroupées dans un voisinage rectangulaire sur la couche précédente. Les champs récepteurs locaux permettent d'extraire des caractéristiques basiques. Les couches sont dites à convolution car les poids sont partagés et chaque cellule de la couche réalise la même combinaison linéaire (avant d'appliquer la fonction sigmoïde) qui peut être vue comme une simple convolution.

Ces caractéristiques sont alors combinées à la couche suivante afin de détecter des caractéristiques de plus haut niveau. Entre deux phases d'extraction de caractéristiques, le réseau réduit la résolution de la carte des caractéristiques par un moyen de sous-échantillonnage. Cette réduction se justifie à deux titres : diminuer la taille de la couche et apporter de la robustesse par rapport aux faibles distorsions.

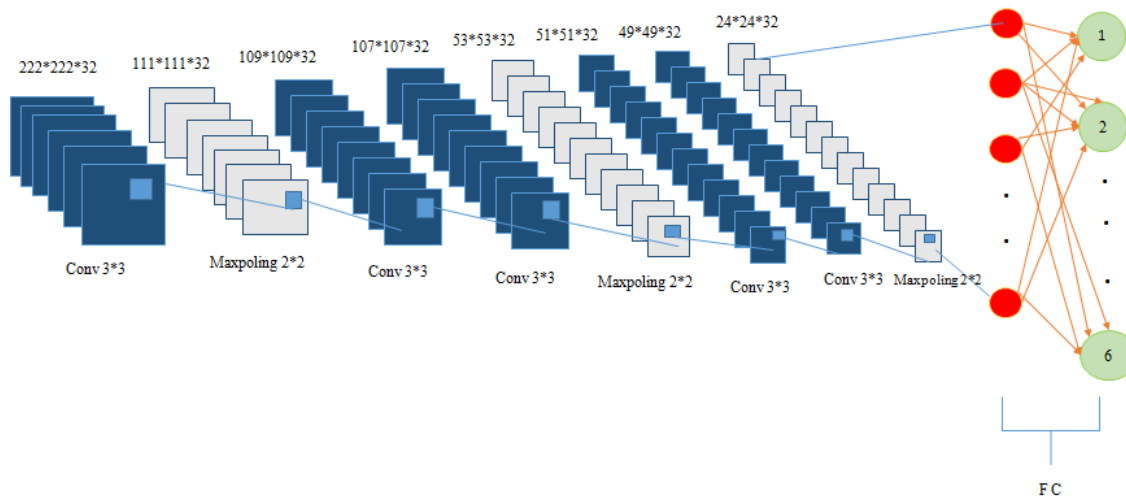


Figure .I.29. L'architecture d'un réseau de neurone convolutifs

I.17.1. Couche de convolution

La convolution est une opération mathématique comme l'addition et la multiplication, il est très utiles de simplifier des équations plus complexe, cette opération est largement utilisée dans le traitement du signal numérique. Lorsque on applique la convolution aux le traitement d'image, on convoler (combiner) l'image d'entrée avec une sous-région de cette image (filtre). Le filtre est aussi connu sous le nom du noyau de convolution, il consiste en des poids de cette sous-région. La sortie de cette couche est l'image entrée avec des modifications qui est souvent appelée une carte de caractéristique (feature Map).

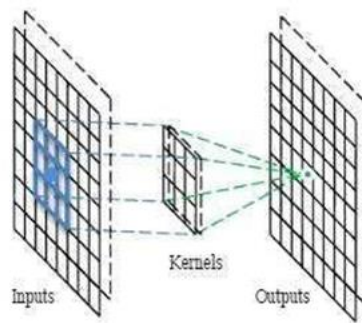


Figure.I.30. L'opération de convolution

En terme mathématique, Une couche de convolution C_i (couche i du réseau) est paramétrée par son nombre N de cartes de convolution $M_{ij}(j=1, \dots, N)$, la taille des noyaux de convolution K_x, K_y (souvent carrée), et le schéma de connexion à la couche précédente L^i . Chaque carte de convolution M_{j^i} est le résultat d'une somme de convolution des cartes de la couche précédente $M_{j^{i-1}}$ par son noyau de convolution respectif. Un biais b_{j^i} est ensuite ajouté et le résultat est passé à une fonction de transfert non-linéaire. Dans le cas d'une carte complètement connectée aux cartes de la couche précédente, le résultat est alors calculé par : [43]

$$M_{j^i} = (b_{j^i} + \sum_{n=1}^n M_{j^{i-1}} \cdot K_j^i)$$

(I.23)

I.17.2. Couche de sous-échantillonnage (Pooling)

Dans les architectures classiques de réseaux de neurones convolutionnels, les couches de convolution sont suivies par des couches de sous-échantillonnage (couche d'agrégation). Cette dernière réduit la taille des cartes de caractéristique pour but de diminuer la taille de paramètre, et renvoie les valeurs maximales des régions rectangulaires de son entrée. [40] [42]

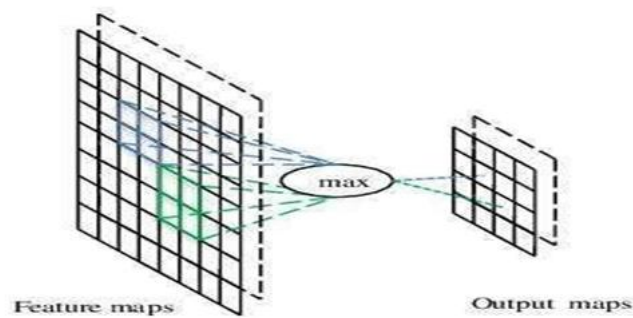


Figure .I.31. L'opération de sous échantillonnage

I.17.3. Couche entièrement connectée

Les paramètres des couches de convolution et de max agrégation sont choisis de sorte que les cartes d'activation de la dernière couche soient de taille 1, ce qui résulte en un vecteur 1D d'attributs. Des couches classiques complètement connectées composées de neurones sont alors ajoutées au réseau pour réaliser la classification.[44] La dernière couche, dans le cas d'un apprentissage supervisé, contient autant de neurones que de classes d'esirées. Cette dernière couche contient N neurones (nombre des classes dans la base), et une fonction d'activation de type sigmoïde est utilisée afin d'obtenir des probabilités d'appartenance à chaque classe. [39] [44]

I.17.4.CNN Architectures

De nombreuses architectures CNN ont été utilisées dans la classification des images via les années, et chacun d'eux a maximisé la performance de l'image la classification à sa manière. certaines des architectures célèbres de CNN sont les Suivant [46] :

I.17.4.1. AlexNet (2012)

AlexNet utilise la fonction d'activation ReLu au lieu de tanh pour ajouter la non-linéarité, qui a accéléré la vitesse de formation (de 6 fois) et augmenté la précision. Il utilise également la régularisation des abandons (une technique évite les problèmes complexes). co-adaptations sur les données d'entraînement pour réduire le surmenage). Une autre caractéristique de AlexNet fait en sorte que le pooling se chevauche pour réduire la taille du réseau. Il réduit les taux d'erreur parmi les premiers et les cinq premiers de 0,4% et 0,3%, respectivement

Le filet contient huit couches avec des poids ; les premiers sont convolutifs et les trois autres sont entièrement connectés. La sortie de la dernière couche entièrement connectée est transmise à un softmax à 1000 voies qui produit une distribution sur les 1000 étiquettes de classe. Notre réseau maximise la logistique multinomiale régression, ce qui équivaut à maximiser la moyenne sur l'ensemble des cas de formation de la log-probabilité de l'étiquette correcte sous la prévision Distribution. [47]

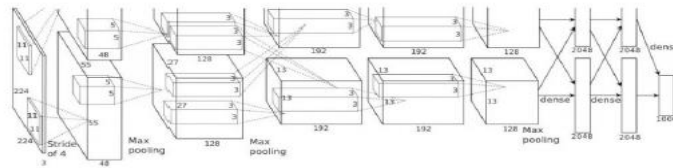


Figure.I.32. AlexNet Architecture

I.17.4.2 GoogLeNet/Inception(2014)

GoogLeNet est développé sur la base de l'idée que plusieurs connexions entre les couches sont inefficaces et contiennent des informations redondantes en raison de la corrélation qui existe entre elles. En conséquence, il utilise un "module de démarrage", une CNN, avec 22 couches dans un traitement parallèle et avantages de plusieurs classes auxiliaires au sein des couches intermédiaires pour améliorer la capacité de discrimination dans les couches inférieures. Contrairement aux CNN conventionnels tels que AlexNet et VGG, dans lesquels une opération de convolution ou de mise en commun peut être utilisée à chaque niveau, le module Inception pourrait bénéficier de les deux à chaque couche. De plus, des filtres (convolutions) de tailles différentes sont utilisés sur la même couche, fournissant des informations plus détaillées et ex-trayant modèles avec différentes tailles [48].

Il est important de noter qu'une couche convolutionnelle 1 x 1, appelée couche de goulot d'étranglement, a été utilisée pour réduire à la fois la complexité informatique et le nombre de paramètres. Pour être plus précis, 1 x 1 couches convolutives ont été utilisées seulement avant un filtre convolutif de noyau plus grand (par exemple 3 x 3 et 5 x 5 convolutionnel) couches) pour diminuer le nombre de paramètres à déterminer à chaque niveau (c'est-à-dire le

processus de la fonctionnalité de mise en commun) De plus, des couches convolutionnelles 1 x 1 rendent le réseau plus profond et ajoutent plus de non-linéarité en utilisant ReLU après chaque couche convolutionnelle 1 x 1. Dans ce réseau, les couches entièrement connectées sont remplacées par un pooling moyen couche. Cela diminue considérablement le nombre de paramètres depuis le début les couches connectées Incluent un grand nombre de paramètres. Ainsi, ce réseau est capable d'apprendre des représentations plus profondes des entités avec moins de paramètres relatifs à Alex Net alors que c'est beaucoup plus rapide que VGG. [48] [49]

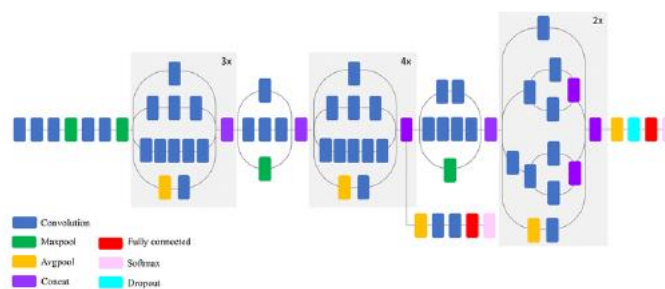


Figure.I.33. Vue comprimée de l'architecture de Google Net (version 3)

I.15. Conclusion

À travers ce premier chapitre, nous avons posé les bases théoriques nécessaires à la compréhension de notre projet. Nous avons d'abord abordé les notions fondamentales du traitement d'image et les différentes étapes de l'acquisition et de la représentation des données visuelles. Ces éléments sont essentiels pour appréhender le rôle de la vision artificielle dans les systèmes intelligents.

Nous avons ensuite exploré le domaine de l'apprentissage automatique, en détaillant ses principes, ses types, ainsi que les approches de classification et de régression. Cette étude a permis de mettre en lumière les capacités des algorithmes à apprendre à partir de données et à prendre des décisions de manière autonome.

La seconde partie du chapitre a été consacrée à l'apprentissage profond, une branche avancée de l'intelligence artificielle qui offre des performances remarquables dans de nombreux domaines. Nous y avons présenté les principales architectures et concepts qui permettent à ces systèmes d'apprentissage de traiter des données complexes de manière efficace.

En somme, ce chapitre constitue un socle théorique indispensable pour comprendre les choix méthodologiques et technologiques que nous présenterons dans les chapitres suivants, consacrés à la conception, à la mise en œuvre et à l'évaluation de notre système intelligent.

Chapitre II
Méthodologie et Outils Utilisés

II. Introduction

Dans toute démarche scientifique sérieuse, la méthodologie constitue la pierre angulaire sur laquelle repose l'ensemble du projet. Elle permet de structurer le travail, de définir clairement les étapes à suivre, d'anticiper les difficultés et d'assurer la reproductibilité des résultats. Dans le cadre de notre projet de développement de lunettes intelligentes destinées aux personnes malvoyantes, nous avons accordé une attention particulière à l'élaboration et à l'application rigoureuse d'une méthodologie adaptée [12].

Notre objectif était de concevoir un système capable de détecter et de lire automatiquement les panneaux médicaux pour en transmettre le contenu à l'utilisateur sous forme vocale. Pour atteindre ce but, plusieurs étapes essentielles ont été nécessaires : la constitution d'une base de données originale, l'annotation précise de ces données, l'entraînement d'un modèle de détection performant, l'intégration d'un système de reconnaissance optique de caractères (OCR) et de synthèse vocale (TTS), ainsi que l'optimisation de l'ensemble du système pour un fonctionnement en temps réel sur un matériel limité comme le Raspberry Pi[13].

Ce chapitre expose de manière détaillée chacune de ces étapes, en expliquant les choix méthodologiques, les outils utilisés, les défis rencontrés, ainsi que les solutions mises en œuvre pour garantir la réussite de notre projet.

II.1. Collecte des données

La qualité d'un modèle de vision artificielle dépend en grande partie de la qualité des données d'entraînement.

Dans notre projet, nous avons accordé une grande importance à la constitution d'un jeu de données riche et représentatif.

II.1.1. Constitution du jeu de données

Pour permettre à notre modèle de détecter efficacement des panneaux médicaux (enseignes de médecins, de pharmacies, etc.), nous avons rassemblé un ensemble d'images issues de deux sources principales :

Internet : Recherche manuelle d'images pertinentes en haute résolution. Captures réelles : Utilisation d'une caméra pour photographier l'environnement réel.

Nous avons veillé à inclure une diversité d'angles de vue, de luminosités et de types D'enseignes pour renforcer la robustesse du modèle.

II.2. Annotation avec Label Studio

Afin de préparer les images pour l'entraînement de notre modèle YOLOv8, une étape essentielle a été l'annotation manuelle via l'outil Label Studio.

Les principales étapes du processus ont été :

II.2.1. Importation des images

L'ensemble des images collectées a été importé dans Label Studio.

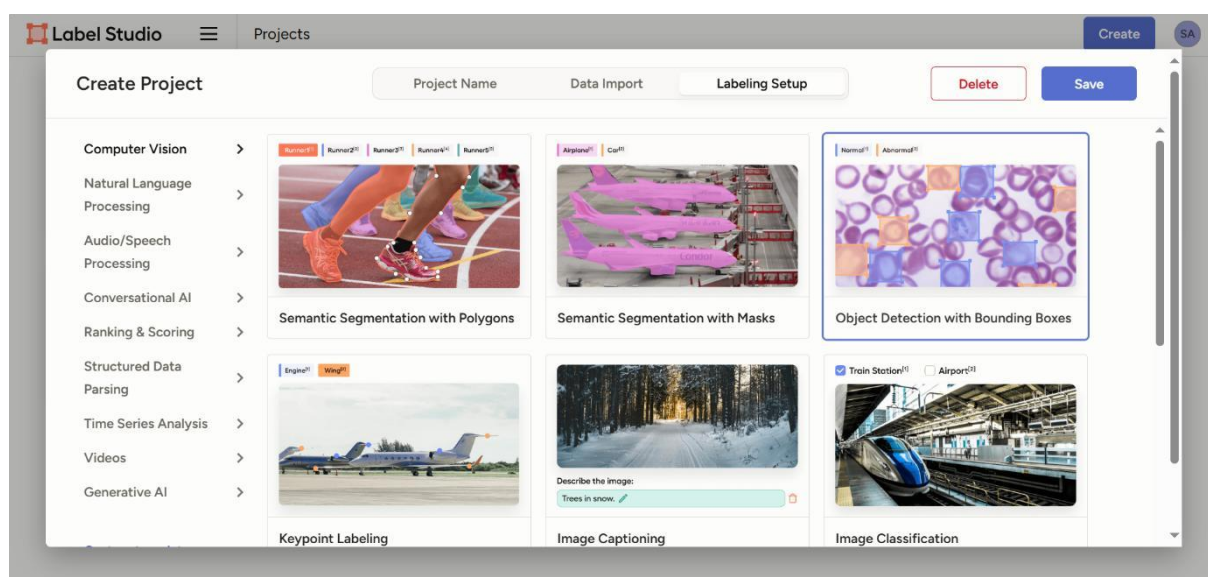


Fig.II.1. Importation des images

II.2.2. Annotation Manuelle

Pour chaque image, nous avons dessiné des boîtes englobantes (bounding boxes) autour des objets d'intérêt (enseignes de médecins, pharmacies).

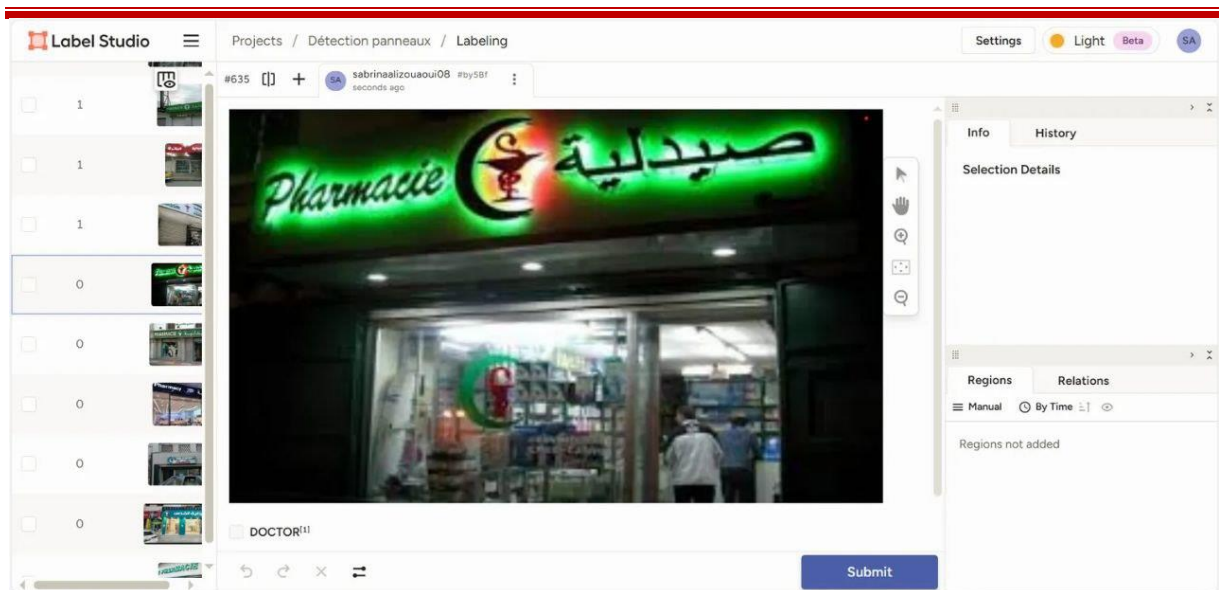


Fig.II.2. Annotation Manuelle

II.2.3. Attribution des étiquettes (Labels)

Chaque boîte a été associée à une catégorie spécifique, en suivant la convention YOLO (ex : 0 = médecin, 1 = pharmacie).

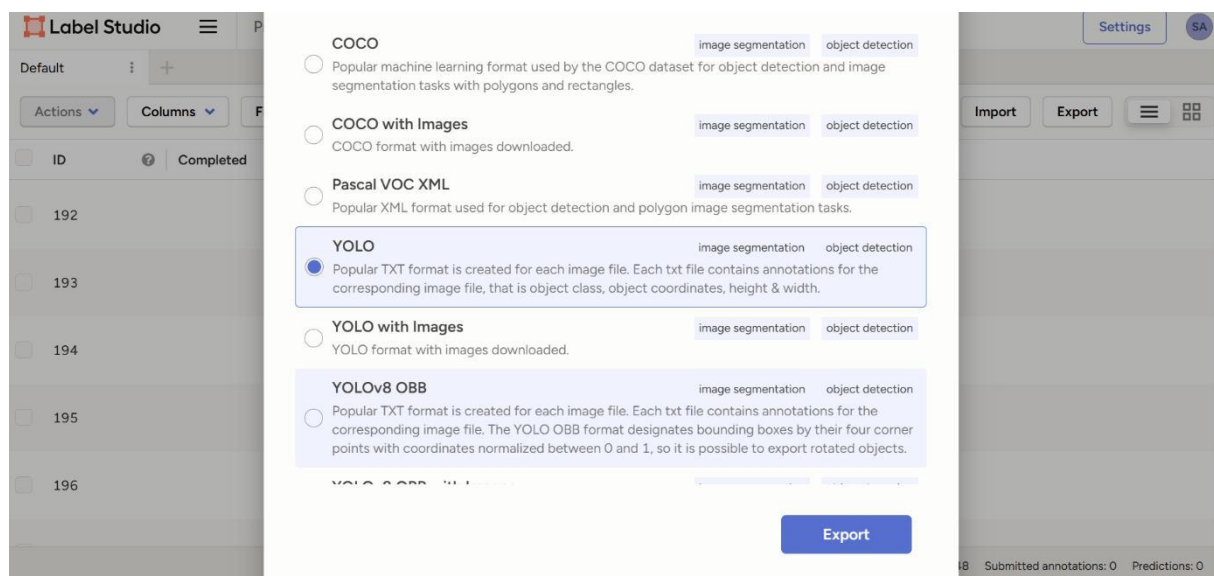


Fig.II.3. Attribution des étiquettes (Labels)

II.2.4. Exportation au format YOLO

Une fois les annotations terminées, nous avons exporté les données sous format YOLO, générant ainsi deux dossiers :

/images/train et /images/val

/labels/train et /labels/val

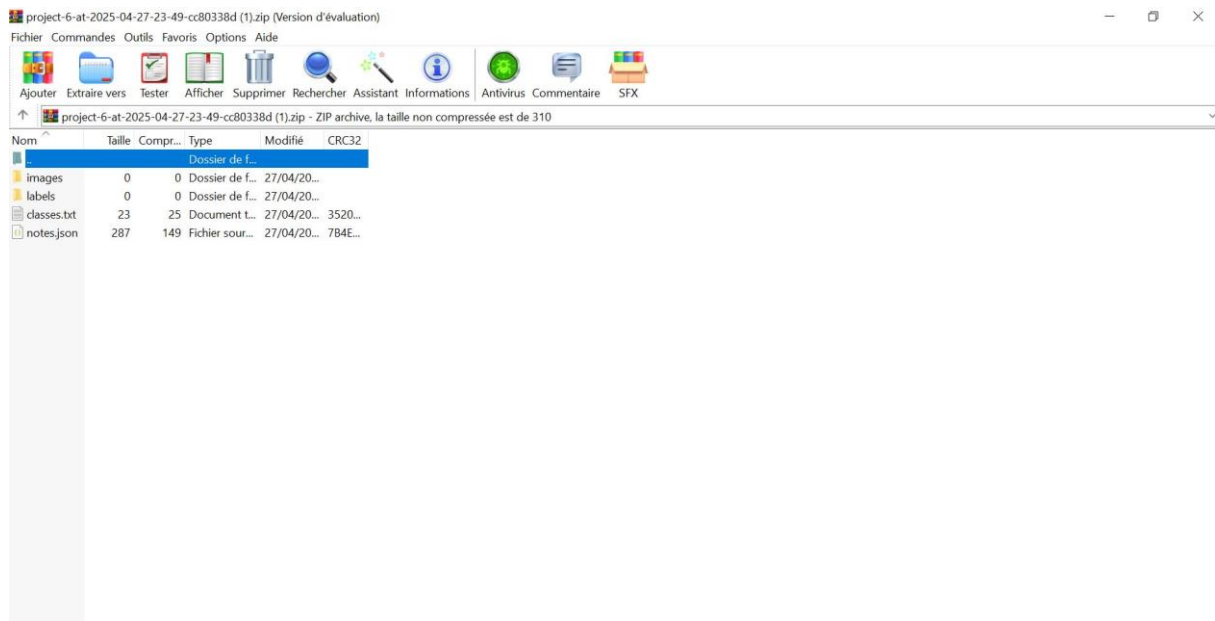


Fig.II.4. Exportation au format YOLO

II.3. Modèles de détection YOLOv8

II.3.1 Détection d'objets avec YOLOv8

La détection d'objets est une fonction essentielle de la vision par ordinateur qui permet d'identifier et de localiser des objets dans une image ou une vidéo. Contrairement à la classification d'images, la détection d'objets ne se contente pas de classer les objets d'une image, mais identifie également leur emplacement dans l'image en dessinant une boîte englobante autour de chaque objet. Les modèles de détection d'objets, tels que R-CNN, Fast R-CNN, Faster R-CNN et YOLO, utilisent des réseaux neuronaux convolutionnels (CNN) pour classer les objets et des réseaux de régression pour prédire avec précision les coordonnées de la boîte englobante pour chaque objet détecté [50][51][52].

Dans notre projet, nous avons choisi d'utiliser YOLOv8 pour ses performances exceptionnelles en termes de précision et de rapidité [53].

II.3.2. Présentation de YOLO

Les modèles YOLO (You Only Look Once) sont des systèmes de détection d'objets en temps réel qui identifient et classent les objets en un seul passage de l'image [50]. En d'autres termes, le modèle ne regarde l'image qu'une seule fois et, à partir de ce « passage unique », il est capable d'identifier les objets présents sur l'image. Cela diffère des modèles précédents qui nécessitaient plusieurs passages pour traiter une image. Ce processus se déroulant en temps réel, il fonctionne à une vitesse incroyable.

La détection d'objets signifie que YOLO peut non seulement localiser un objet sur une image, mais aussi le définir. Pour ce faire, l'image passe par un réseau neuronal qui permet au modèle de détecter tous les objets simultanément. En utilisant la prédiction basée sur la grille et la prédiction de la boîte de délimitation, le modèle est capable de comprendre si les objets se trouvent à l'intérieur d'une certaine cellule ou d'une boîte de délimitation. En outre, YOLO utilise la probabilité de classe pour prédire comment classer les objets, ce qui détermine ce qu'est réellement l'objet.

YOLO présente un certain nombre d'avantages par rapport aux méthodes précédentes. Comme indiqué, elle est extrêmement rapide et peut donc être appliquée à des applications telles que les voitures autonomes et la vidéosurveillance. En outre, elle offre une grande précision, en particulier avec les images naturelles, et moins de faux positifs puisqu'elle analyse une image entière en une seule fois, ce qui offre une plus grande précision contextuelle [54].

II.3.3. Evolution jusqu'à YOLOv8

Historique

YOLO (You Only Look Once) est une famille de modèles de détection d'objets en temps réel qui a profondément transformé le domaine de la vision par ordinateur. Depuis la première version introduite par Joseph Redmon en 2015, chaque évolution a visé à améliorer la précision, la vitesse et l'efficacité du modèle [50].

YOLOv1 a marqué un tournant avec son approche novatrice de détection en une seule passe sur l'image, mais souffrait de limites en matière de précision. YOLOv2 (YOLO9000) a ensuite introduit des encres et permis la détection de plus de 9000 classes, tandis que YOLOv3 a intégré des techniques de détection multi-échelle et des connexions résiduelles, améliorant la performance sur les petits objets. YOLOv4, quant à lui, a optimisé davantage la vitesse et la

robustesse grâce à des méthodes modernes comme la régularisation avancée et l'augmentation de données.

Avec YOLOv5, le projet a pris une nouvelle tournure. Bien qu'il ne soit pas développé par les auteurs originaux, il a été repris et maintenu par l'entreprise **Ultralytics**, qui a porté l'implémentation vers PyTorch[55]. Ce passage a permis de rendre le modèle plus accessible, modulaire et facile à adapter à divers cas d'utilisation. En effet, **Ultralytics YOLO** s'est rapidement imposé comme un outil puissant et complet, combinant une interface conviviale avec un support natif pour l'entraînement, l'évaluation et le déploiement. Il simplifie le processus de développement de l'apprentissage automatique (ML), tout en facilitant la collaboration entre les membres d'une équipe technique, notamment grâce à une plateforme intuitive et parfois sans code.

YOLOv8, dernière version en date, constitue l'aboutissement de cette évolution. Il adopte une architecture plus flexible (anchor-free), propose une précision accrue sur les benchmarks standards (comme COCO), et permet de traiter plusieurs tâches (détection, segmentation, classification, tracking) dans un cadre unifié[56][57]. Grâce à **Ultralytics**, YOLOv8 devient ainsi une solution de référence pour la détection d'objets, combinant robustesse, simplicité d'utilisation et performance de haut niveau.

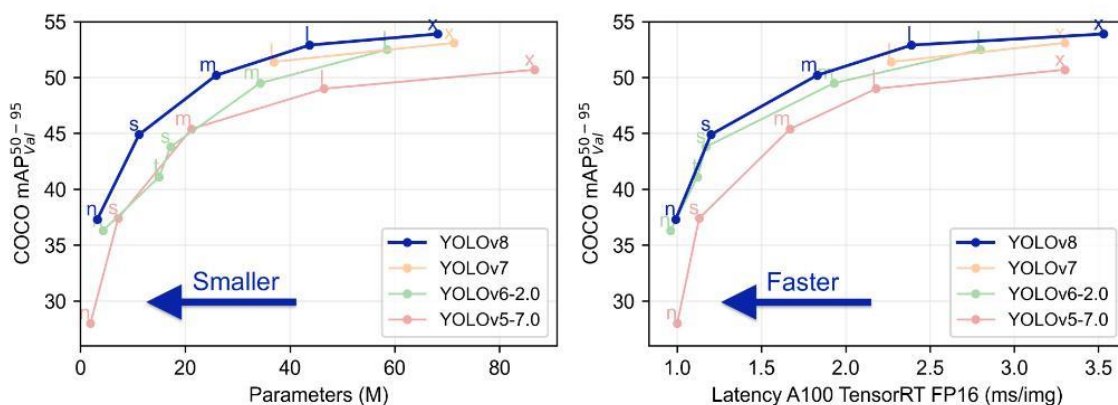


Fig.II.5. Comparaison de Nombre de paramètres dans différentes versions de L'architecture YOLO b) Vitesse d'inférence pour chaque version YOLO

II.3.4. Pourquoi YOLOv8 dans notre projet ?

YOLOv8 offre une précision et une vitesse améliorées par rapport aux versions précédentes de YOLO. Il intègre un réseau fédérateur mis à jour basé sur EfficientNet, ce qui améliore considérablement la capacité du modèle à capturer des fonctionnalités de haut niveau. De plus,

un nouveau module de fusion de fonctionnalités permet d'intégrer efficacement des informations issues de plusieurs échelles, renforçant ainsi la robustesse de la détection. Enfin, YOLOv8 bénéficie de techniques d'augmentation de données améliorées, notamment MixUp et CutMix, qui contribuent à une meilleure généralisation du modèle [54].

Le modèle **Ultralytics YOLOv8**, créé par **Ultralytics**, est un modèle de pointe (SOTA). Il s'appuie sur le succès des versions YOLO précédentes et y ajoute de nouvelles fonctionnalités et améliorations pour accroître encore ses performances et sa polyvalence [56].

YOLOv8 est une excellente option pour divers travaux de reconnaissance d'objets, de segmentation d'images et de classification d'images, car il est rapide, précis et simple.

Ci-dessous, j'ai joint les performances de référence de YOLOv8 en termes de vitesse, de précision, d'efficacité, de coût de calcul, etc.

| Model | size (pixels) | mAP ^{val} ₅₀₋₉₅ | Speed CPU ONNX (ms) | Speed A100 TensorRT (ms) | params (M) | FLOPs (B) |
|---------|---------------|-------------------------------------|---------------------|--------------------------|------------|-----------|
| YOLOv8n | 640 | 37.3 | 80.4 | 0.99 | 3.2 | 8.7 |
| YOLOv8s | 640 | 44.9 | 128.4 | 1.20 | 11.2 | 28.6 |
| YOLOv8m | 640 | 50.2 | 234.7 | 1.83 | 25.9 | 78.9 |
| YOLOv8l | 640 | 52.9 | 375.2 | 2.39 | 43.7 | 165.2 |
| YOLOv8x | 640 | 53.9 | 479.1 | 3.53 | 68.2 | 257.8 |

Tableau II.1. Benchmark de performance pour yolov8

Ainsi, YOLOv8 nous a permis d'assurer une détection en temps réel des panneaux d'affichage pour aider efficacement les utilisateurs malvoyants [57][53].

II.4. Optimisation pour Raspberry Pi

Le **Raspberry Pi** est un ordinateur monocarte (Single-Board Computer - SBC), de la taille d'une carte de crédit, conçu à l'origine pour l'apprentissage de l'informatique et de l'électronique. Grâce à sa faible consommation énergétique, son coût très abordable et sa flexibilité d'usage, il s'est progressivement imposé comme une solution privilégiée dans les domaines de la robotique, de l'IoT (Internet des objets), et plus récemment, dans les applications embarquées d'intelligence artificielle [58][59][62].

La cinquième génération, le Raspberry Pi 5, lancée officiellement en 2023, marque une évolution importante dans les capacités matérielles de cette plateforme. Il est équipé d'un processeur ARM Cortex-A76 quadricœur cadencé à 2,4 GHz, ce qui représente un bond de performance significatif comparé au Raspberry Pi 4[60]. De plus, il prend en charge jusqu'à 8 Go de RAM LPDDR4X, ce qui le rend capable d'exécuter des tâches de calcul plus complexes, notamment celles impliquant des modèles d'apprentissage profond. L'ajout d'un port PCIe est une nouveauté notable, permettant d'y connecter des dispositifs haute vitesse tels que des disques SSD ou des accélérateurs matériels (comme Google Coral ou Intel Neural Compute Stick), ouvrant ainsi la voie à des déploiements IA plus ambitieux sur une architecture légère[61][63].

Sur le plan logiciel, le **Raspberry Pi 5** fonctionne sous Raspberry Pi OS, une distribution Linux dérivée de Debian, spécialement optimisée pour le matériel Raspberry. Ce système est léger, stable et hautement personnalisable, ce qui le rend particulièrement adapté aux projets d'intelligence artificielle embarquée [58][62]. Il est compatible avec les bibliothèques et frameworks populaires comme TensorFlow, PyTorch, OpenCV, ainsi que les outils de développement Python utilisés dans notre projet. De plus, l'écosystème Linux permet l'accès à une large communauté de support et à une documentation riche, ce qui facilite le développement et le déploiement de solutions complexes sur une plateforme à ressources limitées [63][64][65]. Dans notre projet, le Raspberry Pi 5 a été choisi comme cœur du système en raison de sa capacité à exécuter localement des modèles optimisés tels que YOLOv8, tout en restant compact, portable et économe en énergie. Il s'est révélé être un excellent compromis entre performance et mobilité, condition essentielle pour un dispositif intelligent destiné à l'assistance aux personnes malvoyantes.

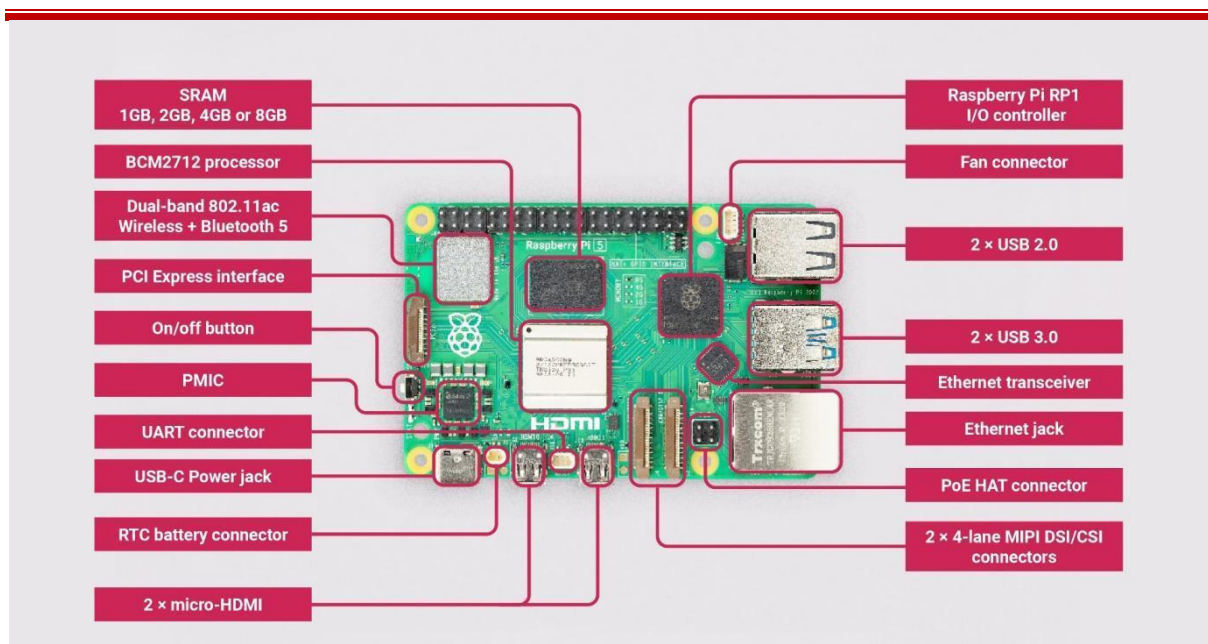


Fig.II.6. La disposition de la carte Raspberry Pi 5

- **Optimisation et Déploiement de YOLOv8 sur Raspberry Pi 5**

Nous avons adapté le modèle **YOLOv8** pour qu'il fonctionne efficacement sur le **Raspberry Pi 5**, grâce à ses performances optimisées permettant une détection rapide des objets en temps Réel, même sur du matériel embarqué. Afin de réduire la charge computationnelle, deux Techniques ont été envisagées : **la quantization**, qui consiste à réduire la précision des poids Du modèle pour améliorer la vitesse, ainsi que **la conversion au format TensorFlow Lite**, Afin de faciliter un déploiement plus léger et adapté à l'environnement contraint du Raspberry Pi [59][61]. Pour maximiser la vitesse d'inférence, l'utilisation d'**accélérateurs matériels** Comme le **Google Coral USB Accelerator (TPU)** ou un **NPU intégré** (lorsqu'il est Disponible via des modules HATs ou PCIe pour le Raspberry Pi) peut être envisagée [63]. Plusieurs **bibliothèques d'intelligence artificielle** ont été exploitées dans ce projet, Notamment **PyTorch** pour l'entraînement initial du modèle YOLOv8, **TensorFlow Lite** pour L'inférence optimisée, ainsi que **OpenCV** pour la manipulation des images et des flux video [64][65][66].

II.5. OCR et Text-to-Speech

II.5.1 Module OCR

La reconnaissance optique des caractères, ou **Optical Character Recognition – OCR** en anglais, est une conversion électronique d'images textuelles dactylographiées, manuscrites ou imprimées. Ce texte est encodé par une machine dans un fichier de format texte [67][68].

Avec l'OCR, un grand nombre de documents papier peuvent être numérisés en texte lisible à la machine, peu importe la langue et le format dans lesquels ils sont rédigés [69][70]. Cette technique facilite non seulement le stockage, mais rend disponibles des données qui auparavant étaient difficilement accessibles [71]

Il suffit simplement de penser à la quantité de données qui dorment dans des boîtes d'archives papier d'une ville ou d'un gouvernement par exemple.

Ces images et documents peuvent être numérisés sous forme de document texte, de photo de document ou de photo de scène (par exemple pour décoder le texte sur un panneau d'affichage).

- **Bibliothèque utilisée: Tesseract OCR**

Tesseract OCR (Optical Caractère Recognition) est un logiciel libre et gratuit qui détecte le texte dans les images. C'est l'un des outils OCR les plus utilisés et il est utilisé pour diverses Applications. Il est réputé pour identifier des textes écrits dans plusieurs langues avec une Excellente précision [67][68]. Tesseract prend en charge plus de 100 langues, ce qui le rend Extrêmement adaptable à une utilisation internationale [69].

Il est régulièrement mis à jour afin d'améliorer ses capacités de reconnaissance [70]. OCR Tesseract est un outil puissant pour transformer des images de texte en texte lisible par une machine. Il convient à un large éventail d'applications allant du simple scan de documents à l'analyse approfondie de documents et à l'extraction de données.

II.6. Module Text-to-Speech (TTS)

Définition

La synthèse vocale, ou Text-to-Speech (TTS), également appelée **conversion texte- parole**, est un système informatique capable de lire à voix haute n'importe quel texte, qu'il soit saisi manuellement par un utilisateur ou obtenu par numérisation à l'aide d'un système de reconnaissance optique de caractères (OCR) [72]. Elle transforme un texte numérique en une voix artificielle compréhensible par l'être humain [73].

- **Bibliothèque utilisée: pyttsx3**

Pour le Raspberry Pi, nous avons opté pour **pyttsx3**, une bibliothèque de TTS légère qui fonctionne **hors ligne**, ce qui est crucial pour une solution embarquée. Elle permet de convertir **un texte numérique en voix de synthèse**, sans nécessiter de connexion Internet [73].

Le module prend en charge plusieurs langues telles que l'anglais, le français, l'hindi, le tamoul, l'allemand, etc.

Cela rend le système plus interactif avec l'utilisateur, en lui permettant d'écouter les informations extraites du texte ou détectées par le système, de manière vocale et fluide.

II.7. Processus global

Le flux de traitement est le suivant :

- Capturer une image via la caméra.
- Appliquer le modèle YOLOv8 pour détecter des objets pertinents.
- Si du texte est détecté, utiliser Tesseract pour extraire le texte.
- Envoyer le texte extrait à pyttsx3 pour synthèse vocale.
- Restituer le message à l'utilisateur via un haut-parleur

II.8. Architecture du Pipeline

- **Architectures matérielles et logicielles du système**

Dans le cadre de notre projet de lunettes intelligentes pour malvoyants, nous avons soigneusement conçu à la fois l'architecture matérielle et logicielle pour assurer une solution efficace, fiable et temps réel.

II.8.1. Architecture matérielle

L'architecture matérielle est composée des éléments suivants :

- **Raspberry Pi 5** : unité de traitement principale.
- **Caméra (Module CSI ou USB)** : pour capturer des images en direct
- **Carte MicroSD** : pour stocker le système d'exploitation et les données.
- **Batterie portable / Power Bank** : pour alimenter l'appareil.
- **Haut-parleur ou écouteurs** : pour restituer la voix synthétisée.
- **Structure physique** : intégration discrète sur la monture des lunettes.

II.8.2. Architecture logicielle

L'architecture logicielle est organisée autour des modules suivants :

- **Système d'exploitation** : Raspberry Pi OS (basé sur Linux Debian).
- **Environnement Python** : pour exécuter les scripts de traitement.
- **Bibliothèques principales** : (- **YOLOv8** pour la détection d'objets (via Ultralytics). - **Tesseract OCR** pour l'extraction du texte. - **pyttsx3** pour la synthèse vocale. - **TensorFlow Lite** (optionnel) pour une éventuelle optimisation des modèles si nécessaire. - **Gestion des entrées/sorties** : via OpenCV (capture vidéo et affichage).

II.9. Workflow détaillé du système

Afin de garantir une expérience utilisateur fluide et en temps réel, notre système suit un enchaînement précis d'étapes, de la capture d'une image jusqu'à la restitution vocale du texte détecté. Le processus se déroule comme suit :

II.9.1. Étapes du flux de travail

- **Capture de l'image** : La caméra du Raspberry Pi capture en temps réel une image de l'environnement.
- **Détection des objets (YOLOv8)** : L'image capturée est traitée par le modèle YOLOv8, qui détecte les panneaux, enseignes, ou tout autre objet pertinent.
- **Extraction du texte (OCR)** : Pour chaque objet pertinent détecté, un traitement OCR est appliqué pour extraire le texte contenu dans l'image.
- **Conversion du texte en audio (TTS)** : Le texte extrait est converti en fichier audio à l'aide du module Text-to-Speech (pyttsx3).
- **Sortie audio vers l'utilisateur** : Le fichier audio est immédiatement lu à travers un haut-parleur ou des écouteurs pour transmettre l'information à l'utilisateur malvoyant.

II.9.2. Synchronisation des modules

Chaque module est exécuté de manière séquentielle et optimisée afin d'assurer :

- Une faible latence entre la capture de l'image et la sortie audio.
- Une gestion efficace de la mémoire sur Raspberry Pi 5.
- Une autonomie énergétique maximale grâce à l'optimisation des ressources.

II.10. Conclusion

À l'issue de ce chapitre, nous avons présenté en détail la méthodologie adoptée pour le développement de lunettes intelligentes destinées aux personnes malvoyantes, en mettant particulièrement l'accent sur la lecture et la vocalisation des panneaux médicaux. Nous avons débuté par la création d'une base de données personnalisée à travers la collecte et la classification d'images de panneaux, assurant ainsi une adéquation parfaite avec notre objectif spécifique.

Par la suite, nous avons utilisé des outils performants tels que Label Studio pour annoter les données avec précision, avant de recourir à Google Colab pour tirer parti des capacités de l'intelligence artificielle, en entraînant un modèle YOLO adapté à notre problématique. Ce choix nous a permis d'obtenir des performances satisfaisantes tout en anticipant les contraintes matérielles liées à l'exécution sur Raspberry Pi.

Nous avons également accordé une attention particulière à l'extraction des textes à l'aide de Tesseract OCR, suivie de leur conversion en audio via une solution TTS, en veillant toujours à choisir des outils conciliant qualité et légèreté pour assurer un fonctionnement fluide.

Chaque étape a présenté ses propres défis, nécessitant des ajustements et des optimisations continus, tant au niveau des données que des modèles et des traitements logiciels.

Cette méthodologie intégrée démontre l'importance d'une planification rigoureuse, du choix judicieux des outils et d'une capacité d'adaptation constante face aux obstacles rencontrés pour atteindre les objectifs d'un projet technologique sensible.

Les résultats obtenus constituent ainsi une base solide pour les développements futurs, qu'il s'agisse d'améliorer les performances, d'élargir la base des panneaux reconnus ou d'enrichir le système par de nouvelles fonctionnalités.

Chapitre III

Résultats et discussion

III.INTRODUCTION

Après avoir défini le cadre théorique et méthodologique de notre projet, ce troisième chapitre est consacré à la présentation de la mise en œuvre concrète du système que nous avons développé, ainsi qu'à l'analyse des résultats obtenus.

La réalisation d'un système intelligent visant à aider les personnes malvoyantes nécessite non seulement une approche conceptuelle rigoureuse, mais également une phase d'implémentation soignée, tenant compte des contraintes techniques et des besoins spécifiques des utilisateurs finaux. C'est dans cette optique que nous avons conçu une solution intégrée, combinant détection d'objets, reconnaissance de texte et synthèse vocale, au sein d'une architecture embarquée.

Dans ce chapitre, nous détaillerons les différentes étapes de l'implémentation, en commençant par la description du système proposé et des outils technologiques utilisés. Nous décrirons ensuite le dataset constitué pour l'entraînement du modèle de détection, avant de présenter les résultats expérimentaux obtenus à travers diverses simulations.

Ce chapitre constitue ainsi une passerelle entre les concepts théoriques explorés précédemment et la réalité de l'application sur le terrain, en mettant en lumière les forces, les limitations et les perspectives d'évolution de notre solution.

III.1 système proposé

Tous les algorithmes basés sur l'apprentissage automatique suivent un modèle prédictif qui estime un certain type de données avec une grande précision. Un vaste ensemble de données est essentiel pour que l'algorithme d'apprentissage puisse comprendre le comportement du système. La première étape de l'apprentissage automatique est l'acquisition des données. Les données collectées ont été partagées par diverses parties intéressées et résumées sur en informations utiles. Les étapes de ce processus sont la purification des données et la délimitation des données. Les données ont été séparées en deux ensembles disjoints : formation et test. L'ensemble de données de formation a été utilisé pour la formation et le test du modèle. L'ensemble de données a été utilisé pour l'optimisation du modèle et l'évaluation.

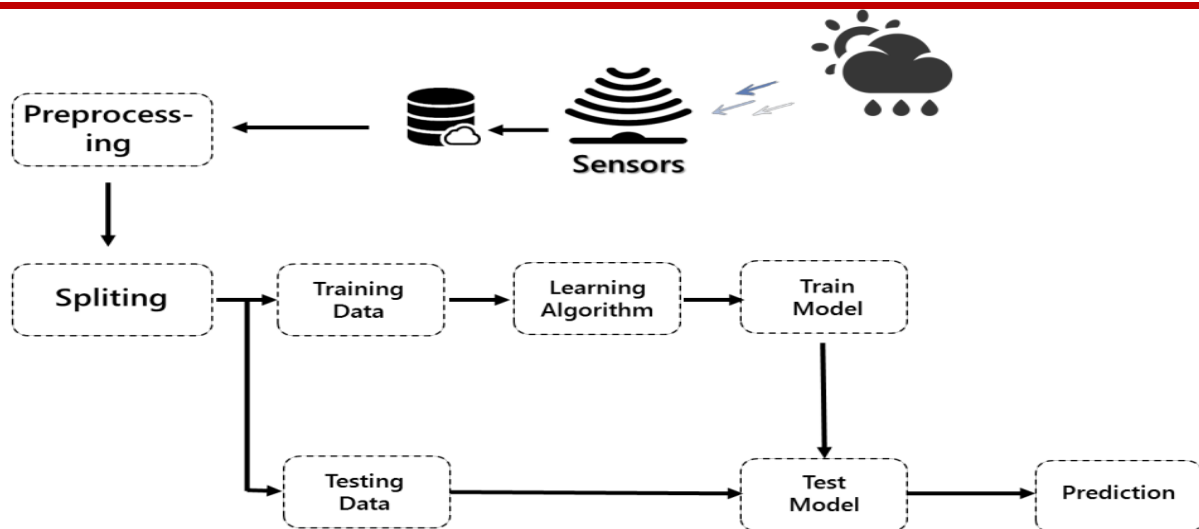


Fig.III.1. Matrice de corrélation pour identifier les caractéristiques essentielles de l'ensemble.

Training Dataset: L'échantillon de données utilisé pour ajuster le modèle.

Testing Dataset: L'échantillon de données utilisé pour fournir une évaluation impartiale de l'ajustement d'un modèle final sur l'ensemble de données d'apprentissage.

Dataset: Un ensemble de données se compose d'environ deux éléments, les deux éléments étant les lignes et les colonnes. En outre, une caractéristique principale d'un enregistrement est qu'il est organisé de manière à ce que chaque ligne contienne une observation.

Machine learning Algorithm : Un « algorithme » dans l'apprentissage automatique est une procédure qui est exécutée sur des données pour créer un « modèle » d'apprentissage automatique.

Machine Learning Model : Dans le domaine de l'apprentissage automatique, un « modèle » est le résultat d'un algorithme d'apprentissage automatique exécuté sur des données.

III.2. Description de l'ensemble de données

Dans le cadre de notre projet de lunettes intelligentes destinées aux personnes malvoyantes, nous avons constitué un dataset personnalisé, orienté principalement vers les panneaux et signalétiques médicaux. Ce choix stratégique vise à répondre à un besoin réel : permettre aux utilisateurs d'identifier rapidement et efficacement les lieux et services médicaux (comme les pharmacies, hôpitaux, cabinets médicaux, etc.) grâce à une assistance vocale.

Le dataset a été collecté manuellement à l'aide d'un smartphone, en prenant des photos de panneaux réels dans divers environnements (urbains, semi-urbains et ruraux). Une attention particulière a été portée à la diversité des images afin d'améliorer la robustesse du modèle : variations de luminosité, angles de prise de vue, différents types de polices d'écriture, couleurs de fond et présence éventuelle d'obstacles partiels.

Chaque image a ensuite été annotée à l'aide de l'outil Label Studio, qui nous a permis de dessiner des boîtes englobantes précises autour des textes et des objets pertinents (ex. : le mot "Pharmacie" sur une enseigne verte). L'annotation a suivi le format YOLO, avec une structure simple (classe, coordonnées normalisées de la boîte), facilitant l'intégration dans notre pipeline d'entraînement.

Le dataset final comprend environ 700 images annotées, réparties en trois classes principales : pharmacie, clinique, et urgence, auxquelles nous avons ajouté une classe générique texte médical pour regrouper les éléments textuels non classifiables directement. Pour garantir une performance optimale du modèle, le dataset a été divisé en trois ensembles : 70% pour l'entraînement, 20% pour la validation, et 10% pour le test.

Cette base de données joue un rôle central dans la réussite de notre projet, car la qualité et la pertinence des données annotées influencent directement les capacités de détection et de reconnaissance du modèle YOLOv8, ainsi que la qualité finale de l'expérience utilisateur.

III.3. Caractéristiques

Le dataset que nous avons utilisé dans le cadre de notre projet a été spécialement conçu pour répondre aux besoins d'un système de reconnaissance de panneaux destinés aux personnes malvoyantes. Il est composé d'un ensemble d'images capturées dans des environnements urbains et semi-urbains, principalement à l'aide d'une caméra fixée sur des lunettes intelligentes simulées, en conditions réelles d'éclairage naturel.

Ce dataset se distingue par la diversité de ses données visuelles. Les images varient en termes de résolution, d'angles de prise de vue, de luminosité, ainsi que de la présence de bruit visuel ou d'objets parasites (voitures, passants, reflets, etc.), ce qui permet au modèle d'apprentissage de mieux généraliser face aux situations imprévues. Les principales classes représentées sont des panneaux de pharmacies, des indications médicales, des enseignes de cliniques et des centres de santé.

Toutes les images ont été annotées manuellement à l'aide de l'outil LabelImg, en suivant le format YOLO (avec des fichiers .txt contenant les coordonnées des boîtes englobantes et les classes correspondantes). Le dataset comprend environ 600 images, réparties entre un ensemble d'entraînement (80%) et un ensemble de validation (20%).

Cette structuration permet non seulement une meilleure performance durant l'entraînement, mais elle garantit également la reproductibilité et l'évaluation rigoureuse des résultats obtenus lors de la phase de test sur Raspberry Pi.

III.4. Évaluation prévue

III.4.1 Environnement de développement

III.4.1.1. Google Colab

Google Colab est un service cloud gratuit qui prend désormais en charge les GPU gratuits, améliorant les compétences de codage dans le langage de programmation Python. Développez des applications d'apprentissage en profondeur à l'aide de bibliothèques populaires telles que Keras, TensorFlow, PyTorch. La caractéristique la plus importante qui distingue Colab des autres services gratuits de cloud computing est la suivante : Colab fournit un GPU et est entièrement gratuit [74].

III.4.2. Python

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Créé par Guido van Rossum et publié en 1991, Python a une philosophie de conception qui met l'accent sur la lisibilité du code , y compris l'utilisation d'espaces virtuels. Il fournit des constructions qui permettent une programmation explicite à petite et grande échelle. Python dispose d'un gadget dynamique et d'une gestion automatisée de la mémoire . Il aide plusieurs paradigmes de programmation, dont orienté objet, impératif, fonctionnel et procédural, et dispose d'une bibliothèque générale complète. Les interprètes Python sont accessibles pour de nombreuses structures d'exécution [75].



Fig.III.2. Le logo Python

III.4.3. Jupyter Notebook

Le Jupyter Notebook est un utilitaire Internet open-source qui vous permet de créer et de partager des fichiers comprenant du code, des équations, des visualisations et du contenu textuel narratif. Les utilisations comprennent le nettoyage et la transformation des données, la simulation numérique, la modélisation statistique, la visualisation des données, l'apprentissage automatique, et bien plus encore [76].



Fig.III.3. Le logo de Jupyter Notebook.

III.4.4. Visual Studio Code

Est un éditeur de code source développé par Microsoft pour Windows, Linux et macOS. Il intègre Git et prend en charge le débogage, la coloration syntaxique, la complétion intelligente du code, snippets, et le remaniement du code. Il est exceptionnellement personnalisable, permettant aux clients de changer le thème, les raccourcis clavier, les préférences, et de configurer des extensions qui ajoutent des fonctionnalités supplémentaires à . Le code source est gratuit et libre, lancé sous la licence permissive MIT. Les binaires compilés sont des logiciels gratuits pour tous les usages [77].

```

nvidia-smi

Sat May 3 22:55:21 2025

+-----+
| NVIDIA-SMI 550.54.15      | Driver Version: 550.54.15      | CUDA Version: 12.4      |
+-----+-----+
| GPU  Name           Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |  Memory-Usage | GPU-Util  Compute M. |
|=====+-----+=====+
| 0   Tesla T4              Off          | 00000000:00:04:0  Off   |    0         0      |
| N/A   43C    PB            11W / 70W   |  0MiB / 15360MiB |      0%    Default  |
+-----+-----+-----+
| Processes:                                                       GPU Memory
|  GPU   GI    CI          PID    Type   Process name          Usage
|=====+-----+=====+
| No running processes found
  
```

Fig.III.4. Interface logicielle Visual Studio Code.

III.4.5. Présentation des outils

III.4.5.1. Le logiciel

✓ **Tensorflow**

TensorFlow a été créé par l'équipe Google Brain pour la recherche en ML et Deep Learning. Il est considéré comme une version moderne de Theano.

- **Avantages**

- ✓ soutenu par Google
- ✓ Une très grande communauté
- ✓ Prise en charge multi-GPU

- **Inconvénients**

- ✓ Plus lent que d'autres frameworks dans de nombreux benchmarks, bien que TensorFlow soit en train de rattraper son retard.
- ✓ Theano surpasse toujours le support RNN
- ✓ **Keras**

Le niveau le plus élevé, le framework le plus convivial de la liste. Il permet aux utilisateurs de choisir si les modèles qu'ils construisent tournent sur Theano ou TensorFlow.



Fig.III.5. keras logo

III.5. Créer un modèle

Dans cette étape, nous avons créé et entraîné un modèle YOLOv8 dans le but de détecter automatiquement trois types de panneaux signalétiques présents dans notre jeu de données, à savoir : **PHARMACY**, **DOCTOR** et **LAB**. Ce modèle constitue le cœur de notre système d'assistance aux personnes malvoyantes.

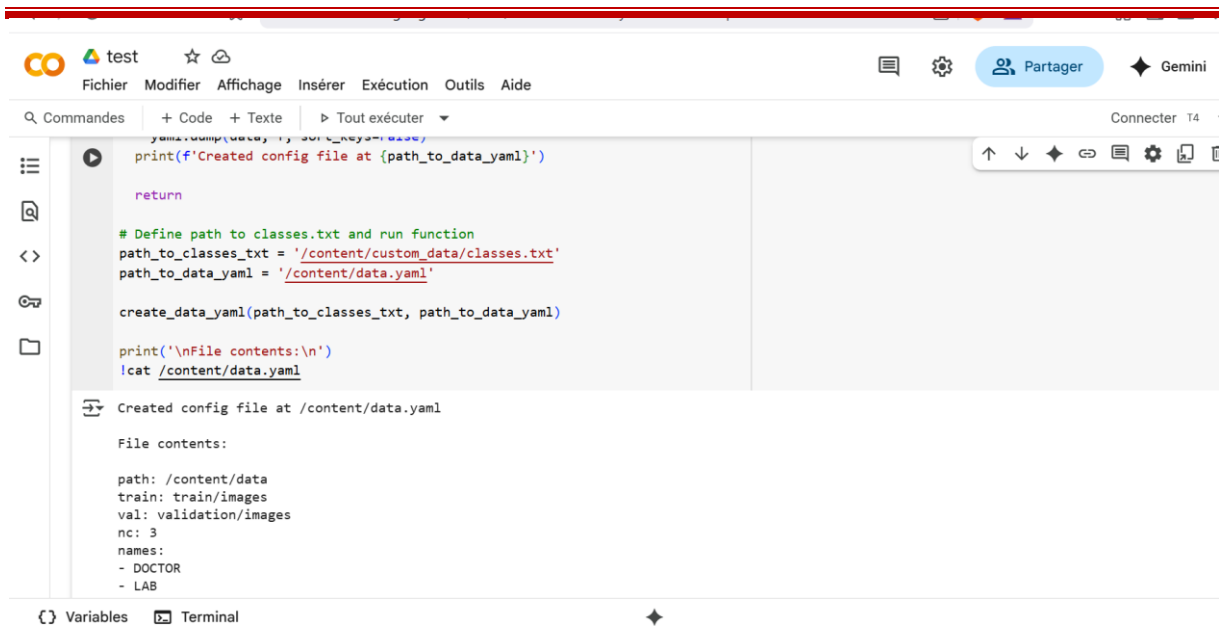
Le processus de création du modèle s'est déroulé comme suit :

- **Préparation des données**

Nous avons commencé par décompresser les images annotées, puis nous avons utilisé un script automatique pour répartir ces données en deux ensembles : **90 % pour l'entraînement** et **10 % pour la validation**, afin de garantir une bonne généralisation du modèle.

- **Création du fichier de configuration data.yaml**

Un fichier YAML a été généré automatiquement à partir d'un fichier classes.txt. Il contient les informations essentielles telles que le nombre de classes (3), les noms de chaque classe, ainsi que les chemins d'accès aux images d'entraînement et de validation.



```
print(f'Created config file at {path_to_data_yaml}')

return

# Define path to classes.txt and run function
path_to_classes_txt = '/content/custom_data/classes.txt'
path_to_data_yaml = '/content/data.yaml'

create_data_yaml(path_to_classes_txt, path_to_data_yaml)

print('\nFile contents:\n')
!cat /content/data.yaml

Created config file at /content/data.yaml

File contents:

path: /content/data
train: train/images
val: validation/images
nc: 3
names:
- DOCTOR
- LAB
```

Fig.III.6. Création de fichier data.yaml

- **Entraînement du modèle YOLOv8**

Le modèle **YOLOv8n (nano)** a été sélectionné pour sa légèreté et son efficacité, particulièrement adaptée à l'exécution sur des dispositifs à ressources limitées comme le Raspberry Pi. L'entraînement a été effectué pendant **60 époques**, avec une taille d'image fixée à 640 pixels.

III.6. Prédiction et évaluation

Une fois l'entraînement terminé, nous avons évalué le modèle en effectuant des prédictions sur les images de l'ensemble de validation. Les résultats ont permis de vérifier que le modèle était capable de localiser et d'identifier correctement les types de panneaux visés.



Fig.III.7. Évaluation de model

- Sauvegarde du modèle

Le fichier de poids best.pt, correspondant à la meilleure version du modèle entraîné, a été extrait et enregistré dans un dossier spécifique. Ce fichier a ensuite été compressé au format .zip pour faciliter son intégration ultérieure dans l'application finale.

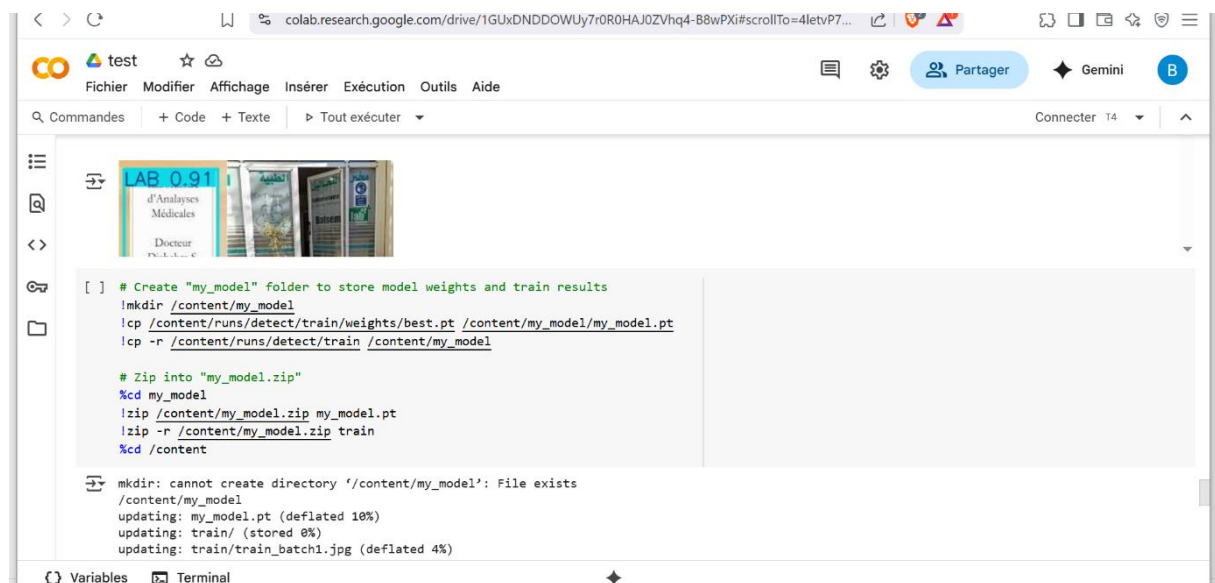
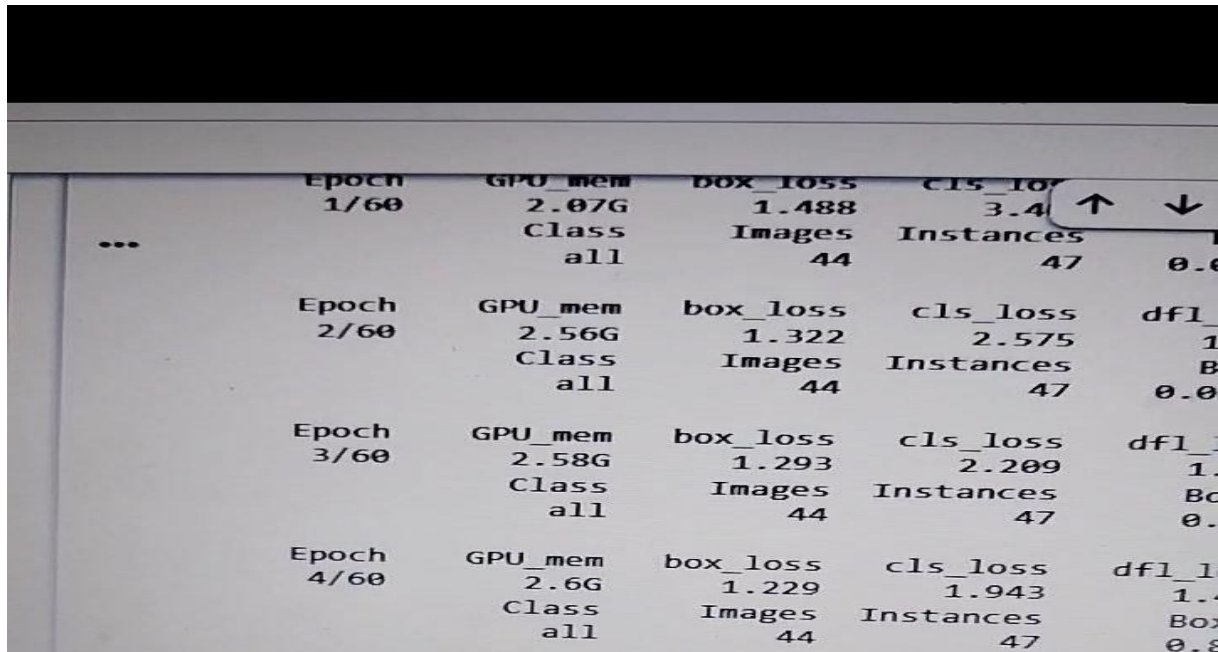


Fig.II.8. Créer le dossier « my_model » pour stocker les poids du modèle et les résultats de l'entraînement.

Ce modèle constitue la base du système intelligent que nous avons développé pour extraire des informations visuelles (sous forme de texte) à partir de l'environnement réel.

III.7. Résultat du modèle



| Epoch | GPU_mem | box_loss | cls_loss | df1 |
|-------|---------|----------|-----------|-----|
| 1/60 | 2.07G | 1.488 | 3.4 | |
| ... | Class | Images | Instances | |
| | all | 44 | 47 | 0.0 |
| Epoch | GPU_mem | box_loss | cls_loss | df1 |
| 2/60 | 2.56G | 1.322 | 2.575 | 1 |
| | Class | Images | Instances | B |
| | all | 44 | 47 | 0.0 |
| Epoch | GPU_mem | box_loss | cls_loss | df1 |
| 3/60 | 2.58G | 1.293 | 2.209 | 1. |
| | Class | Images | Instances | Bc |
| | all | 44 | 47 | 0. |
| Epoch | GPU_mem | box_loss | cls_loss | df1 |
| 4/60 | 2.6G | 1.229 | 1.943 | 1.. |
| | Class | Images | Instances | Bo |
| | all | 44 | 47 | 0.8 |

Fig.III.9. Résultats des quatre premières époques du mode.

L'image ci-dessus(fig.II.10) représente l'évolution des différentes métriques de performance du modèle YOLOv8 pendant 60 epochs. L'analyse de ces courbes permet d'évaluer la qualité de

l'apprentissage du modèle ainsi que sa capacité de généralisation sur les données de validation.

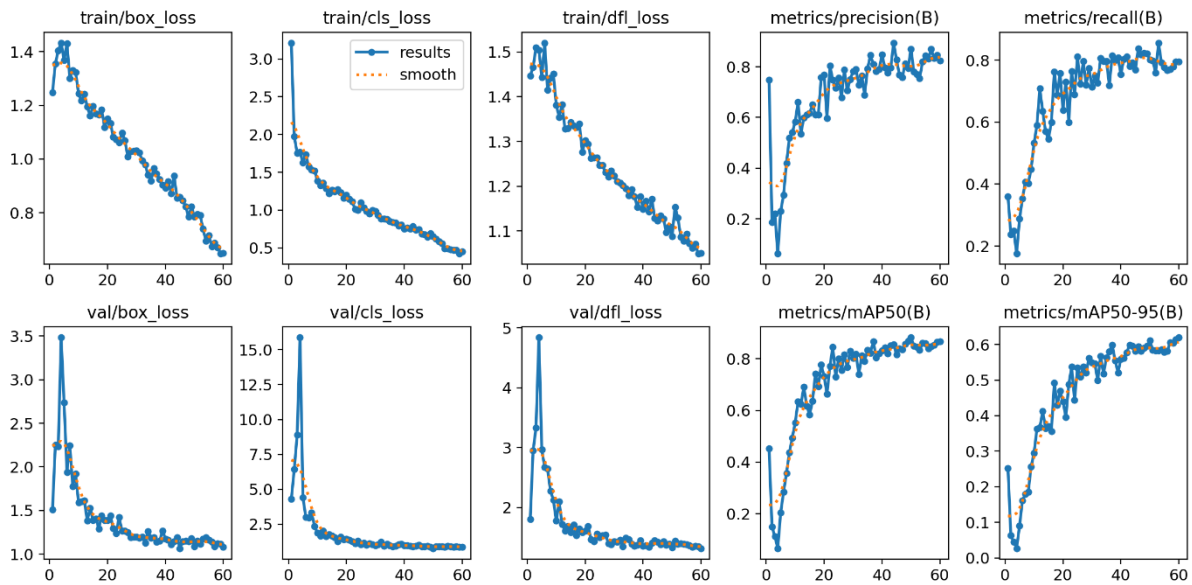


Fig.II.10. Performances du model Yolo sur notre dataset

☒ 1. Courbes de perte d'entraînement (train/...)

- **Train/box_loss** : Cette courbe mesure l'erreur dans la localisation des boîtes englobantes durant l'apprentissage. On observe une décroissance régulière, indiquant que le modèle apprend à mieux encadrer les objets.
- **Train/cls_loss** : Elle indique l'erreur de classification des objets. Sa diminution progressive confirme l'amélioration de la capacité du modèle à distinguer entre les classes PHARMACY, DOCTOR et LAB.
- **Train/dfl_loss** : La Distribution Focal Loss permet de raffiner la localisation des boîtes. Sa baisse constante montre une stabilisation dans la précision des prédictions.

☒ 2. Courbes de perte de validation (val/...)

- **val/box_loss, val/cls_loss, val/dfl_loss** : Similaires aux courbes d'entraînement mais évaluées sur les données de validation, ces pertes diminuent aussi, ce qui indique l'absence de sur-apprentissage (overfitting) et une bonne capacité de généralisation.

☒ 3. Courbes des métriques de performance (metrics/...)

- **metrics/precision(B)** : La précision indique le taux de vraies détections parmi les prédictions positives. Sa progression jusqu'à ~ 0.85 est un bon signe de fiabilité du modèle.

-
- **metrics/recall(B)** : Le rappel mesure le taux de détection correcte parmi tous les objets réellement présents. Il augmente rapidement vers ~ 0.8 , montrant que le modèle apprend à détecter la majorité des objets.
 - **metrics/mAP50(B)** : Moyenne des précisions pour un seuil IoU de 50%. Elle atteint une valeur supérieure à 0.85, ce qui indique une bonne capacité de détection globale.
 - **metrics/mAP50-95(B)** : Moyenne pondérée pour plusieurs seuils IoU de 0.50 à 0.95. Plus exigeante, elle atteint ~ 0.6 , ce qui reste un excellent résultat pour un modèle léger sur des ressources limitées (comme un Raspberry Pi 5).

Ces résultats témoignent de la **convergence progressive du modèle** vers un état optimal. Les pertes diminuent tandis que les performances augmentent, ce qui démontre un entraînement réussi et un bon équilibre entre biais et variance. Le modèle est donc prêt à être intégré dans un système embarqué temps réel comme le Raspberry Pi 5.

III.8. Déploiement de l'application

III.8.1 Raspberry PI

La Raspberry PI (encore appelée Raspi) est un nano ordinateur (environ 40 euros) sous licence libre (open hardware), composé d'une carte à processeur ARM de la taille d'une carte de crédit. Elle peut être utilisée comme serveur, passerelle multimédia, Pc d'appoint carte de prototypage etc...

Elle est équipée de la présence de **GPIO** (general-purpose input/output) port d'entrées/sorties permettant de piloter de nombreux capteurs et moteurs etc.

Elle possède également 1 port Ethernet Gigabit, un module (Bluetooth 5.0 BLE), 1 puce wifi (IEEE 802.11b/g/n/ac) (Figure II.6).

Enfin elle possède de nombreux shields (écran tactile, carte relais, reconnaissance vocal...) mais peut aussi accepter une caméra dédiée (8 Mégapixels résolution max de 3280x2464).

- **Son intérêt:**

Encourager l'apprentissage de la programmation informatique et de ses différents langages en rendant l'informatique abordable et accessible à tous.

III.8. 2. Installation de la Raspberry PI

La Raspberry Pi charge son système d'exploitation à partir du lecteur de carte micro **SD** formater au préalable en **FAT32**. Depuis la Raspberry Pi version 3 et supérieur, nous avons la possibilité de charger le système à partir d'une clé **USB**, d'un disque **SSD** en **USB3** ou du réseau sans avoir recours à la carte **SD**.

Il est donc nécessaire de préparer le système de manière appropriée pour que le nano ordinateur puisse démarrer. Mais quel **OS** choisir car il existe beaucoup, il n'y a en effet pas d'OS idéal à proprement parler : tout dépend des applications que vous souhaitez mettre en œuvre, mais aussi de vos habitudes et de vos préférences. La plupart des distributions pour Raspberry sont des distributions Linux. Nous ne décrivons que ces 3 OS généralement les plus répandu.

Une fois le système d'exploitation installé, mettre la carte SD dans la Raspberry et l'allumer.

Vous devriez avoir écran d'accueil suivant (Figure III. 7) :

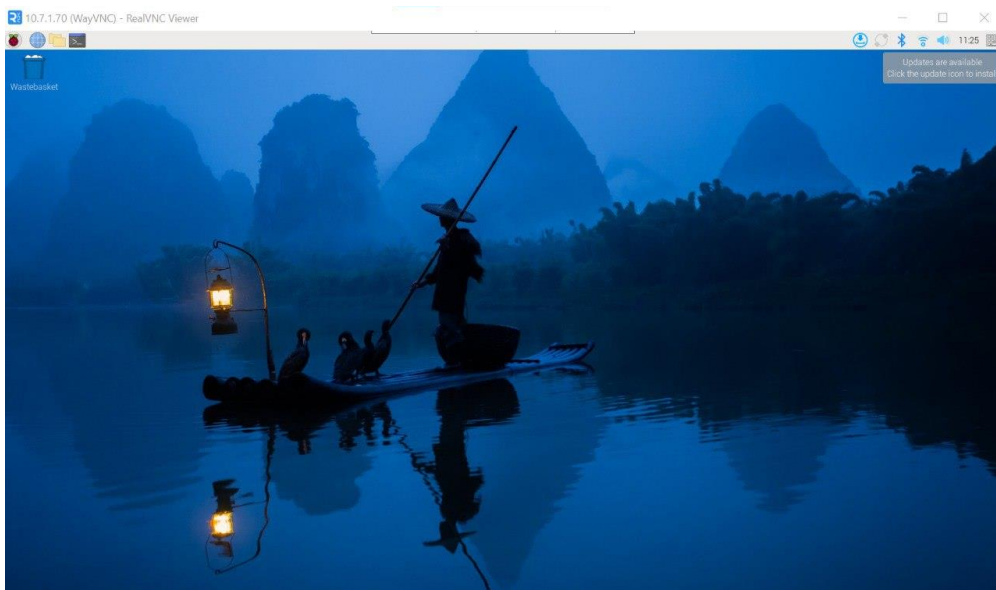


Fig.III.11. Ecran d'accueil de Raspbian


- **Conseil pratique:**

Pour exploiter les performances du Raspberry Pi 5, il est recommandé d'utiliser une carte **microSD d'au moins 16 Go**, idéalement **de classe 10** ou supérieure. Une capacité de **32 Go**

ou plus est fortement conseillée pour les projets nécessitant des bibliothèques lourdes ou des modèles d'intelligence artificielle comme YOLO.

- **Commandes de bases**

La plupart des distributions pour Raspberry sont des distributions Linux, quelques commandes de bases sont donc utiles à connaître. Elles nécessitent une connexion internet. Depuis l'écran

D'accueil dans la barre de tâche cliquez sur : ⇒ 

Dans la nouvelle fenêtre (la console) vous pouvez taper directement les certaines commandes:

Sudo apt-get update : Rechercher les mises à jour des logiciels installés sur la SD dans Raspbian.

Sudo apt-get upgrade : Télécharger et installer les mises à jour des logiciels installés sur la SD.

Sudo apt-get dist-upgrade : Mise à jour de la distribution Raspbian.

Sudo rpi-update : Mettre à jour du firmware du Raspberry.

Donc, « *le Raspberry Pi* » est une plateforme incroyablement versatile qui continue d'inspirer des millions de personnes à explorer le monde de la technologie et de l'innovation.

III.9. Conclusion

Ce troisième chapitre a permis de passer de la théorie à la pratique en présentant l'implémentation détaillée du système intelligent conçu dans le cadre de notre projet. Nous avons proposé une architecture complète combinant plusieurs technologies complémentaires, allant de la détection d'objets à la reconnaissance de texte, jusqu'à la restitution vocale de l'information.

Nous avons ensuite décrit l'ensemble de données utilisé, en soulignant ses caractéristiques, sa structure et son importance dans la qualité de l'apprentissage du modèle. La mise en place d'un environnement de développement adapté a également été un facteur déterminant dans la réussite de l'intégration des différents modules.

Par ailleurs, une évaluation prédictive a été effectuée afin d'anticiper les performances du système, tandis que les résultats de simulation ont permis de valider l'efficacité de notre approche. Ces expérimentations ont mis en évidence les points forts du système, tout en révélant certains axes d'amélioration possibles.

Ce chapitre constitue ainsi une étape charnière dans notre travail, car il illustre de manière concrète comment les concepts étudiés peuvent être mobilisés pour créer une solution technologique utile, fonctionnelle et accessible aux utilisateurs finaux.

Conclusion générale

Ce mémoire a permis de concrétiser un projet ambitieux alliant intelligence artificielle, vision par ordinateur et technologies d'assistance, au service des personnes en situation de handicap visuel. À travers la conception et le développement d'un système intelligent capable de détecter, de lire et de transmettre vocalement les informations visuelles issues de l'environnement, nous avons démontré qu'il est possible de mettre l'innovation technologique au service de l'inclusion sociale.

Ce travail nous a amenés à explorer des thématiques complexes telles que la détection d'objets en temps réel, la reconnaissance optique de caractères, et la synthèse vocale, en intégrant l'ensemble dans une architecture embarquée adaptée à un usage quotidien. Nous avons ainsi relevé le défi de créer une solution autonome, portable et interactive, conçue pour répondre à des besoins concrets sur le terrain.

Au-delà des résultats obtenus, ce projet nous a permis de mesurer l'impact potentiel que peuvent avoir les technologies modernes sur la qualité de vie des personnes vulnérables. Nous avons pris conscience de l'importance de développer des systèmes non seulement performants, mais aussi accessibles, intuitifs et adaptés aux usages réels.

Ce mémoire ne constitue pas une fin en soi, mais bien une étape vers des perspectives d'amélioration et d'élargissement. De nombreuses voies restent à explorer : enrichissement du modèle de détection, optimisation des performances en conditions variées, et extension de l'application à d'autres types d'informations visuelles.

En somme, cette expérience nous a confortés dans la conviction que l'innovation ne prend tout son sens que lorsqu'elle est mise au service de l'humain. Nous espérons que ce travail contribuera, à son échelle, à ouvrir la voie à de futures avancées en matière de technologies d'assistance et d'accessibilité numérique.

Références Bibliographiques

- [1] **Szeliski, R.** (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer.
- [2] DataFranca, *Vision artificielle*, 2025. [En ligne]. Disponible sur: <https://datafranca.org>
- [3] CrossData, "Vision par ordinateur et image : comprendre, appliquer et anticiper l'avenir" (2024)
- [4] Context Understanding in Computer Vision: A Survey
Xuan Wang & Zhigang Zhu (2023)
- [5] Wikipédia, Vision par ordinateur, 2025. [En ligne]. Disponible sur:
https://fr.wikipedia.org/wiki/Vision_par_ordinateur
- [6] Deng, L., & Yu, D. (2014). *Deep Learning: Methods and Applications*. Foundations and Trends® in Signal Processing, 7(3–4), 197–387.
- [7] <https://www.fun-mooc.fr/fr/cours/introduction-au-traitement-des-images/>
- [8] Wikipédia, Traitement d'images, 2025. [En ligne]. Disponible sur:
https://fr.wikipedia.org/wiki/Traitement_d'images
- [9] "Mise au Point d'une Application de Reconnaissance de Formes", Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique, DJABEUR DJEZZAR Mohammed Rafik, BENKADA Feth-a
- [10] J.P. Cocquerez et S. Philipp, *Analyse d'images : filtrage et segmentation* Ed. Masson, Année 1995.
- [11] Gonzalez, R. C., & Woods, R. E. (2018). *Traitement d'images numériques*. Pearson.
- [12] S. Patel, A. Mehta, and T. Shah, "Wearable Technology for Blind Navigation: A Review," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2020.
- [13] A. K. Jain and R. Singh, "Smart Wearable Devices for the Visually Impaired: A Review," *International Journal of Computer Applications*, 2021.
- [14] Forsyth & Ponce, *Computer Vision: A Modern Approach*, Pearson.
- [15] Strang, G. (2016). *Introduction to Linear Algebra*
- [16] Rene Y. Choi; Aaron S. Coyner; Jayashree Kalpathy-Cramer; Michael F. Chiang; J. Peter Campbell, *Introduction to Machine Learning, Neural Networks, and Deep Learning*, *Translational Vision Science & Technology*, February 2020
- [18] "Apprentissage machine efficace : théorie et pratique", par Olivier Delalleau, D'épartement d'informatique et de recherche opérationnelle Faculté des arts et des sciences

- [19] Shalev-Shwartz, S. & Ben-David, S., "Understanding Machine Learning: From Theory to Algorithms", Cambridge University Press, 2014
- [20] Morgane LAUR « Anticipation des changements de notes des obligations du portefeuille d'un assureur par méthode de machine learning » Mémoire présenté devant l'Université Paris Dauphine pour l'obtention du diplôme du Master Actuariat et l'admission à l'Institut des Actuaire
- [21] Sepp Hochreiter Fakultat "LONG SHORT-TERM MEMORY" Technische Universität München 1997
- [22] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," Signal Processing, IEEE Transactions, vol. 45, no. 11, pp. 2673–2681, 1997
- [23] Moualek Djaloul Youcef « Deep Learning pour la classification des images » Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique 2016
- [24] HAWKINS Jeff, BLAKESLEE Sandra. On intelligence : how a new understanding of the brain will lead to the creation of truly intelligent machines. 2007
- [25] Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.
- [26] Wendong, Y., Zhengzheng, L., & Bo, J. (2017, June). A multi-factor analysis model of quantitative investment based on GA and SVM. In 2017 2nd International Conference on Image, Vision and Computing (ICIVC) (pp. 1152-1155). IEEE.
- [27] Kareem, F. Q., & Abdulazeez, A. M. Ultrasound Medical Images Classification Based on Deep Learning Algorithms: A Review.
- [28] Zeebaree, Diyar Qader, Habibollah Haron, and Adnan Mohsin Abdulazeez. "Gene selection and classification of microarray data using convolutional neural network." In 2018 International Conference on Advanced Science and Engineering (ICOASE), pp. 145-150. IEEE, 2018
- [29] Abdulqader, Dildar Masood, Adnan Mohsin Abdulazeez, and Diyar Qader Zeebaree. "Machine Learning Supervised Algorithms of Gene Selection: A Review." Machine Learning 62, no. 03 (2020).
- [30] Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. Journal of Applied Science and Technology Trends, 1(2), 56-70.
- [31] Dai, H. (2018, March). Research on SVM improved algorithm for large data classification. In 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA) (pp. 181-185). IEEE.

- [32] Haji, S. H., & Abdulazeez, A. M. (2021). COMPARISON OF OPTIMIZATION TECHNIQUES BASED ON GRADIENT DESCENT ALGORITHM: A REVIEW. *PalArch's Journal of Archaeology of Egypt/Egyptology*, 18(4), 2715-2743.
- [33] Tao, P., Sun, Z., & Sun, Z. (2018). An improved intrusion detection algorithm based on GA and SVM. *Ieee Access*, 6, 13624-13631
- [34] Chauhan, V. K., Dahiya, K., & Sharma, A. (2019). Problem formulations and solvers in linear SVM: a review. *Artificial Intelligence Review*, 52(2), 803-855.
- [35] Ademujimi, T. T., Brundage, M. P., & Prabhu, V. V. (2017, September). A review of current machine learning techniques used in manufacturing diagnosis. In *IFIP International Conference on Advances in Production Management Systems* (pp. 407-415). Springer, Cham.
- [36] D. Q. Zeebaree, A. M. Abdulazeez, D. A. Zebari, H. Haron and H. Nuzly, "Multi-level fusion in ultrasound for cancer detection based on uniform lbp features," *Computers, Materials & Continua*, vol. 66, no.3, pp. 3363–3382, 2021
- [37] Daniel Durstewitz , Georgia Koppe, Andreas Meyer, Lindenberg Deep neural networks in psychiatry,2019
- [38]Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall , Activation Functions: Comparison of Trends in Practice and Research for DeepLearning ,2018
- [39]Mokri, Mohammed Zakaria. Classification des images avec les réseaux de neurones convolutionnels. Diss. 09-01-2018, 2017.
- [40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [41] "Understanding LSTM Networks -- colah's blog." [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed: 08-Jun-2018].
- [42] Du Terrail, Jean Ogier. Réseaux de neurones convolutionnels profonds pour la détection de petits véhicules en imagerie aérienne. Diss. Normandie Université, 2018.
- [43]Buysens, Pierre, and Abderrahim Elmoataz. "Réseaux de neurones convolutionnels multi-échelle pour la classification cellulaire." 2016
- [44] Toufik, LAKHDARI Ahmed, and A. B. B. A. C. I. Salih. "La reconnaissance des caractères arabes manuscrits par les réseaux des neurones convolutionnels." (2017)
- [45] Understanding LSTM Networks. Oinkina . Posted on August 27, 2015 . URL :

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[46] Robert, Thomas, et al. "Réseaux convolutionnels pour l'image."

[47] Douillard, Arthur, Yifu Chen, and Matthieu Cord. "Visualisation des réseaux de neurones." (2019).

[48] Douillard, Arthur, Yifu Chen, and Matthieu Cord. "Réseaux convolutionnels pour l'image."

[49] Marty, Jean-Marc, et al. "Analyse d'opinions de tweets par réseaux de neurones convolutionnels." Actes de DEFT, Caen, France: TALN (2015).

[50] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA, 2016.

[51] Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000

[52] Li, X.; Lv, C.; Wang, W.; Li, G.; Yang, L.; Yang, J. Generalized Focal Loss: Towards Efficient Representation Learning for Dense Object Detection. IEEE Trans. Pattern Anal. Mach. Intell. 2022, 45, 3139–3153.

[53] Vats, A.; Anastasiu, D.C. Enhancing Retail Checkout Through Video Inpainting, YOLOv8 Detection, and DeepSort Tracking. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 17–24 June 2023; IEEE: New York, NY, USA, 2023.

[54] Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extr. 2023, 5, 1680–1716.

[55] Ultralytics. 2013. Available online: <https://github.com/ultralytics/ultralytics> (consulté le 25 mai 2025)

[56] Ultralytics. Ultralytics YOLOv8 Docs. Available online: <https://docs.ultralytics.com/> (consulté le 25 mai 2025)

[57] Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection. Electronics 2023, 12, 3664.

- [58] E. Upton and H. Gareth, *Raspberry Pi*, PEARSON. 2013
- [59] M. ZiyaUlHaq, "Embedded Web server for Industrial Applications using Raspberry-Pi," *International Journal of Engineering Science and Computing*, vol. 7, no. 4, pp. 6371–6374, 2017.
- [60] E. P. T. Kerja, *Raspberry Pi 4 User Guide*. 2019.
- [61] M. B. Kalpana, M. T. Student, and R. R. Dist, "Online Monitoring Of Water Quality Using Raspberry Pi3 Model B," *INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY ANDRESEARCH*, vol. 4, no. 6, pp. 4790–4795, 2016.
- [62] S. E. Princy and K. G. J. Nigel, "Implementation of Cloud Server for Real Time Data Storage using Raspberry Pi," in *Online International Confernece on Green Engineering and Technologies*, 2015, pp. 0–3
- [63] G. Jadhav, K. Jadhav, and K. Nadlamani, "Environment Monitoring System using Raspberry-Pi," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 4, pp. 1168–1172, 2016.
- [64] A. Fernández-Pacheco, S. Martin, and M. Castro, "Implementation of an arduino remotelaboratory with raspberry pi," *IEEE Global Engineering Education Conference, EDUCON*, vol. April- 2019, pp. 1415–1418, 2019, doi: 10.1109/EDUCON.2019.8725030.
- [65] R. Pi, W. Notifications, D. E. Kurniawan, M. Iqbal, J. Friadi, and R. I. Borman, "Smart Monitoring Temperature and Humidity of the Room Server Using Raspberry Pi and Whatsapp Notifications," in *Journal of Physics: Conference Series*, 2019, pp. 1–8, doi: 10.1088/1742-6596/1351/1/012006.
- [66] P. V Murkute and V. M. Deshmukh, "Implementing the VOIP Communication Principles using Raspberry Pi as Server," *International Journal of Computer Applications*, vol. 124, no. 4, pp. 34–38, 2015.
- [67] SV Rice, FR Jenkins et TA Nartker, *Quatrième rapport technique annuel sur le test de précision de l'OCR 95-03*, juillet 1995.
- [68] RW Smith, *L'extraction et la reconnaissance de texte à partir d'images de documents multimédias*, novembre 1987.
- [69] JSV Rice, G. Nagy et TA Nartker, *Reconnaissance optique de caractères : un guide illustré de la frontière, États-Unis* : Kluwer Academic Publishers, pp. 57-60, 1999.
- [70] G. Nagy, « Aux frontières de l'OCR », *Proc. IEEE*, vol. 80, no. 7, pp. 1093-1100, juil. 1992

- [71] I. Marosi, « Approches OCR industrielles : algorithmes d'architecture et techniques d'adaptation», Reconnaissance et recherche de documents XIV SPIE , pp. 6500-01, janv. 2007.
- [72] Ainsworth, W., "A system for converting English text into speech," Audio and Electroacoustics, IEEE Transactions on , vol.21, no.3, pp. 288-290, Jun 1973
- [73] Breen, A.P., "The future role of text to speech synthesis in automated services," Advances in Interactive Voice Technologies for Telecommunication Services (Digest No: 1997/147), IEE Colloquium on , vol., no., pp.6/1-6/5, 12 Jun 1997
- [74] Google Colab Free GPU Tutorial .Fuat . Jan 26, 2018 .URL : <https://medium.com/deeplearning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d>
- [75] Python.URL: <https://www.python.org/>
- [76] Jupyter .URL : <https://jupyter.org/>
- [77] Visual Studio Code . URL : https://en.wikiversity.org/wiki/Visual_Studio_Code