

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Constantine 2- Abdelhamid Mehri



Faculté des Nouvelles Technologies de l'Information et la Communication
Département des Technologies des Logiciels et Systèmes d'Information

Projet de fin d'études pour l'obtention du diplôme de
Master en Informatique

Option : Systèmes d'Information et Technologie Web
Thème

**Une application à base de Deep Learning pour la description
des images du monde réel pour les malvoyants**

Réalisé par :

Baaziz Mouad

Sarroub Mohamed Reda

Encadré par :

Dr. HAMAZ Kamal

Remerciements

C'est avec plaisir que nous réservons ces quelques lignes en signe de gratitude et de profonde reconnaissance à l'égard de tous ceux qui nous ont aidés à la réalisation de notre projet de fin d'études.

Nous tenons tout d'abord à remercier ALLAH de nous avoir donné le courage, la volonté et la patience à mener à terme ce présent travail.

Nos sincères remerciements à notre encadrant Mr. HAMAZ Kamal qui nous a guidés dans ce travail, et nous tenons, à lui témoigner toute notre reconnaissance pour ces conseils et directives dont il n'a omis à tout moment de nous en faire part.

Que les membres de jury trouvent nos profondes gratitude pour l'honneur qu'ils nous font en examinant le présent mémoire et en assistant à notre soutenance.

Nous tenons aussi à remercier chaleureusement nos familles et nos amis pour tout le soutien moral qui ont pu nous apporter.

Nos vifs remerciements à l'ensemble des enseignants ayant contribué à notre formation.

Nous remercions aussi, tous ceux, et celles qui ont fait part, de près ou de loin, de l'accomplissement de ce modeste travail.

Dédicaces

Nous dédions ce modeste travail et notre profonde gratitude :

À Nos très chers parents qui nous ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, nous ne saurions exprimer notre gratitude seulement par des mots.

Que dieu vous protège et vous garde pour nous.

À nos chers frères et nos précieuses sœurs, les mots ne peuvent résumer notre reconnaissance et notre amour à votre égard.

À tous les membres de nos familles

À nos adorables amies, pour votre fidélité et votre soutien

À tous nos amis avec lesquels nous avons partagé nos moments de joie et de bonheur.

À tous nos enseignants pour votre soutien, votre enseignement et vos conseils tout au long de notre parcours éducatif et professionnel.

Que toute personne nous ayant aidé de près ou de loin, trouve ici l'expression de notre reconnaissance.

Résumé

La vision occupe une place essentielle dans la vie de chaque individu, car elle nous offre une perspective unique sur le monde qui nous entoure. Elle nous permet d'explorer notre environnement, de reconnaître les visages de nos proches et d'admirer la beauté des paysages. Malheureusement, il y a des personnes qui souffrent de déficiences visuelles, ce qui limite leur capacité à profiter pleinement de ces expériences visuelles. Ces troubles de vision peuvent être causés par diverses raisons, telles que des maladies oculaires, des accidents ou des conditions génétiques. Ces difficultés visuelles peuvent avoir un impact significatif sur leur indépendance, leur qualité de vie et leur participation sociale. C'est dans ce sens que vient notre travail, où nous avons proposé une application de détection des objets et description des scènes du monde réel pour les malvoyants. Notre système est développé pour être utilisé dans une application mobile, en utilisant les techniques d'Intelligence Artificielle, notamment la Vision par Ordinateur et le Traitement du Langage Naturel. Grâce à ces technologies, les personnes malvoyantes disposent de nouvelles possibilités et d'une plus grande autonomie dans leur vie quotidienne. De cette façon, nous pouvons aider les personnes malvoyantes à connaître la description des objets qui les entourent et la compréhension des scènes présentes devant eux.

Mots clés : Personnes malvoyantes, Vision par Ordinateur, Détection d'objets, Traitement du Langage Naturel.

Abstract

Vision occupies an essential place in the life of every individual, as it offers us a unique perspective on the world around us. It allows us to explore our environment, recognize the faces of our loved ones and admire the beauty of the landscapes. Unfortunately, there are people who suffer from visual impairments, which limits their ability to fully enjoy these visual experiences. These vision disorders can be caused by various reasons, such as eye diseases, accidents or genetic conditions. These visual difficulties can have a significant impact on their independence, quality of life and social participation. It is in this direction that comes our work, where we proposed an application of detection of objects and description of real world scenes for the visually impaired. Our system is developed for use in a mobile application, using Artificial Intelligence techniques, including Computer Vision and Natural Language Processing. Thanks to these technologies, visually impaired people have new possibilities and greater autonomy in their daily lives. In this way, we can help visually impaired people to know the description of the objects around them and the understanding of the scenes in front of them.

Keywords: Visually impaired people, Computer vision, Object detection, Natural language processing.

ملخص

تحتل الرؤية مكانة أساسية في حياة كل فرد ، لأنها تقدم لنا منظورًا فريدًا للعالم من حولنا. يتيح لنا استكشاف بيئتنا والتعرف على وجوه أحبائنا والاستمتاع بجمال المناظر الطبيعية. لسوء الحظ ، هناك أشخاص يعانون من إعاقات بصرية ، مما يحد من قدرتهم على الاستمتاع الكامل بهذه التجارب البصرية. يمكن أن تحدث اضطرابات الرؤية هذه لأسباب مختلفة ، مثل أمراض العيون أو الحوادث أو الحالات الوراثية. يمكن أن يكون لهذه الصعوبات البصرية تأثير كبير على استقلاليتهم ونوعية حياتهم ومشاركتهم الاجتماعية. في هذا الاتجاه يأتي عملنا ، حيث اقترحنا تطبيق الكشف عن الأشياء ووصف مشاهد العالم الحقيقي للمكفوفين. تم تطوير نظامنا للاستخدام في تطبيقات الهاتف المحمول ، باستخدام تقنيات الذكاء الاصطناعي ، بما في ذلك الرؤية الحاسوبية ومعالجة اللغة الطبيعية. بفضل هذه التقنيات ، يتمتع الأشخاص ضعاف البصر بإمكانيات جديدة واستقلالية أكبر في حياتهم اليومية. وبهذه الطريقة يمكننا مساعدة ضعاف البصر في معرفة وصف الأشياء من حولهم وفهم المشاهد أمامهم.

الكلمات الرئيسية: ضعف البصر ، رؤية الكمبيوتر ، كشف الأشياء ، معالجة اللغة الطبيعية.

Sommaire

Introduction générale.....	1
<i>Chapitre 01.....</i>	<i>3</i>
<i>Déficience Visuelle et Aides Technologiques pour Les Malvoyants</i>	<i>3</i>
1. Introduction.....	4
2. Vision humaine.....	4
2. 1. Importance de la vision	4
3. Déficience visuelle	5
3.2. Types de déficiences visuelles.....	6
3.3. Causes de déficiences visuelles	7
3.3.1. Cataracte.....	8
3.3.2. Dégénérescence maculaire liée à l'âge (DMLA)	8
3.3.3. Glaucome.....	9
3.3.4. Rétinopathie diabétique.....	10
3.3.5. Trachome	10
4. Impact de la déficience visuelle	11
4.1. Impact individuel.....	11
4.2. Impacts psychologiques.....	11
4.3. Impact économique	11
5. Aides technologiques pour les malvoyants	11
5.1. Technologies d'assistantes	12
5.1.1. Canne blanche électronique.....	12
5.1.2. Plage braille	12
5.1.3. Loupes électroniques	13
5.1.4. Téléagrandisseurs et Vidéoagrandisseurs	13
5.1.5. Lunettes à reconnaissance visuelle	14
5.1.6. Vocale Presse.....	14
5.2. Applications mobile existantes pour les malvoyantes.....	15
5.2.1. Be MyEyes	15
5.2.2. Seeing AI.....	16
5.2.3. Envision AI.....	16
5.2.4. Nav by ViaOpta.....	17
5.2.5. BigLauncher	17

5.2.6. Voice Over /TalkBack	17
6. Conclusion	18
<i>Chapitre 02.....</i>	<i>19</i>
<i>Intelligence Artificielle et Vision par Ordinateur</i>	<i>19</i>
1. Introduction	20
2. Intelligence artificiel.....	20
3. Apprentissage Automatique	21
3.1. Entrainement d'un modèle	22
3.2. Types d'Apprentissage Automatique.....	23
3.2.1. Apprentissage Supervisé (Supervised Learning).....	23
3.2.2. Apprentissage non Supervisé (Unsupervised Learning)	23
3.2.3. Apprentissage par Renforcement (Reinforcement Learning).....	23
3.2.4. Apprentissage semi-supervisé (Semi-Supervised Learning)	24
3.2.5. Apprentissage par transfert (Transfer Learning)	24
4. Apprentissage Profond	24
4.1. Réseaux de neurones artificiels profonds.....	25
4.1.1. Réseaux de neurones multicouches MLP	26
4.1.2. Réseaux de neurones convolutés CNN.....	27
4.1.3. Réseaux de neurones récurrents RNN	28
4.1.4. Réseaux de neurones récurrents à mémoire court et long terme LSTM	29
5. Vision par Ordinateur	29
5.1. Exemple d'application de Vision par Ordinateur.....	30
6. Détection des objets.....	30
6.1. Algorithmes de détection d'objets.....	31
6.1.1. R-CNN.....	31
6.1.2. SSD (Single ShotMultiBox Detector)	31
6.1.3. YOLO (You Only Look Once).....	32
7. Description automatique des images (Image Captioning).....	32
7.1. Traitement du Langage Naturel.....	33
7.1.1. Transformateur	34
8. Conclusion	34
<i>Chapitre 03.....</i>	<i>35</i>
<i>Méthodologie de Détection et de Description des Objets</i>	<i>35</i>

1. Introduction :	36
2. Architecture du système	36
2.1. Choix du modèle :	37
3. Méthodologie de détection d'objets	38
3.1. Modèle de détection d'objets	38
3.2. Préparation de l'ensemble de données	40
3.2.1. Collecte des images	40
3.2.2. Prétraitement des images	40
3.2.3. Annotation des images	40
3.2.4. Division de l'ensemble de données	41
3.3. Entraînement du modèle	41
4. Méthodologie d'Image Captioning	42
4.1. Modèles d'Image Captioning	42
4.1.1. Extraction des caractéristiques d'images avec CNN	42
4.1.2. Transformateur	42
4.2. Ensembles de données d'Image Captioning	43
4.3. Préparation de l'ensemble de données	44
4.3.1. Prétraitement des images	44
4.3.2. Prétraitement du texte (les descriptions)	45
4.3.3. Division de l'ensemble de données	47
4.4. Entraînement du modèle	47
5. Conversion des résultats en vocale	47
6. Conclusion	48
<i>Chapitre 04</i>	49
<i>Implémentation</i>	49
1. Introduction	50
2. Environnement de travail	50
2.1. Environnement matériel	50
2.2. Environnement logiciel	51
2.2.1. Tensorflow	51
2.2.2. Android Studio	51
2.2.3. Google Colab	51
2.2.4. Google Drive	52

2.2.5. LabelImg.....	52
3. Implémentation de modèle de détection d'objets	52
3.1. Collecte des images	52
3.2. Annotation des images	54
3.3. Division de l'ensemble des images	55
3.4. Entraînement de modèle.....	55
3.4.1. Fichier de configuration des données.....	56
3.4.2. Téléchargement du modèle pré-entraîné	56
3.4.3. Lancement d'entraînement	57
3.5. Évaluation du modèle.....	58
3.5.1. Test et résultat.....	59
4. Implémentation de modèle d'Image Captioning.....	60
4.1. Choix de l'ensemble de données.....	60
4.2. Préparation de l'ensemble de données.....	62
4.2.1. Prétraitement des images	62
4.2.2. Prétraitement du texte (les descriptions)	63
4.2.3. Division de l'ensemble de données.....	64
4.3. Entraînement de modèle.....	65
4.3.1. Construction du modèle.....	65
4.3.2. Prédiction.....	66
4.3.3. Test et résultat.....	67
5. Conversion en TFLITE.....	68
5.1. Conversion de modèle de détection d'objets	68
5.2. Conversion de modèle d'Image Captioning.....	68
6. Choix du modèle.....	69
7. Conversion aux résultats vocaux	69
8. Interfaces et scénario d'exécution	70
9. Conclusion	75
Conclusion générale	76
Références	77

Table des figures

Figure 1: Personne non-voyante.	6
Figure 2: Vue avec une cataracte.....	8
Figure 3: Vue avec une DMLA.	9
Figure 4: Vue avec un glaucome.	9
Figure 5: Paysage vu avec une rétinopathie diabétique.....	10
Figure 6: Canne intelligente Rango de la start-up GoSense.....	12
Figure 7: BrailleNoteTouch Plus, tablette braille tactile adaptée aux personnes aveugles.	13
Figure 8: Exemples des Loupes électroniques.	13
Figure 9: Exemples des téléagrandisseurs.	14
Figure 10: Lunettes MyEye d'OrCam.	14
Figure 11: Utilisation de l'application BeMyEyes.	16
Figure 12: Application mobile Seeing AI pour les personnes malvoyantes.	16
Figure 13: BigLauncher simplifie et améliore la visualisation de l'écran d'accueil.	17
Figure 14: Intelligence Artificielle et Machine Learning.	22
Figure 15: Etapes d'entraînement d'un modèle.....	23
Figure 16: Apprentissage Profond et Apprentissage Automatique.	24
Figure 17: Réseaux de neurone.	25
Figure 18: Composent d'un neurone artificiel.	26
Figure 19: Réseaux de neurones multicouches.	27
Figure 20: Architecture du CNN.....	27
Figure 21: Exemple détaillé sur l'architecture du CNN.....	28
Figure 22: Exemple d'un réseau de neurones récurrents RNN.....	29
Figure 23: Application d'une détection d'objets.	30
Figure 24: Schéma du système R-CNN, issu de Girshick et al. (2014).	31
Figure 25: Schéma explicatif du mécanisme de détection de l'architecture SSD.....	32
Figure 26: Mécanisme de détection de l'architecture YOLO.	32
Figure 27: Exemple d'image captioning.	33
Figure 28: Architecture du système.....	37
Figure 29: Algorithme de choix de modèle.....	38
Figure 30: Technique d'apprentissage par transfert appliquée pour YOLO.....	39
Figure 31: Annotation d'une image pour la détection d'objets.	41
Figure 32: Exemple de l'ensemble de donnée Flickr8k.	44
Figure 33: Méthode de Prétraitement d'images.	45
Figure 34: Méthode de Prétraitement du texte.	46
Figure 35: La méthode de conversion en vocale (TextToSpeech).	48
Figure 36: Nombre d'images collectées par objet.....	53
Figure 37: Annotation d'une image avec LabelImg.	54
Figure 38: Exemple d'un fichier d'annotation au format YOLO.....	55
Figure 39: Structure de répertoire de l'ensemble de données.	55
Figure 40: Fichier de configuration de données.	56
Figure 41: Connexion au compte Google Drive.	57
Figure 42: Commande d'entraînement de modèle YOLO.....	57

Figure 43: Métrique de perte et de précision du modèle YOLO.....	58
Figure 44: Résultats de l'apprentissage obtenu pour chaque objet.	59
Figure 45: Résultat obtenu de détection par yolov5 entraîné.	59
Figure 46: Sélection d'un sous-ensemble de dataset.	61
Figure 47: Image avec sa description de l'ensemble de données COCO.	61
Figure 48: Prétraitement d'images pour la tâche d'Image Captioning.....	62
Figure 49: Augmentation de l'ensemble des images.	63
Figure 50: Nettoyage du texte.	63
Figure 51: Tokenisation et Création du vocabulaire del'ensemble de textes.	64
Figure 52: Répartition de chaque ensemble de données.	64
Figure 53: Modèle InceptionV3 pour l'extraction des caractéristiques.	65
Figure 54: Commande d'entraînement de modèle d'Image Captioning.	66
Figure 55: Méthode de prédiction d'une description d'image.....	67
Figure 56: Résultat obtenu de prédiction parle modèle d'Image Captioning.	67
Figure 57: Conversion du modèle YOLOv5 au TFLITE.....	68
Figure 58: Conversion de modèle d'Image Captioning au TFLITE.	69
Figure 59: Code de choix de modèle.....	69
Figure 60: Méthode de conversion aux résultats vocaux pour la détection d'objets.	70
Figure 61: Méthode de conversion aux résultats vocaux pour l'Image Captioning.	70
Figure 62: Page d'accueil.	71
Figure 63: Page d'accueil avec commande orale.	72
Figure 64: Génération d'une description vocale de l'image.....	73
Figure 65: Interface pour la détection d'objets.....	74
Figure 66: Notification quand un objet est trouvé.....	75

Table des tableaux

Tableau 1: Classification du handicap visuel suivant l'OMS.	7
Tableau 2: Environnement matériel.	50

Introduction générale

La vision joue un rôle crucial dans la vie des êtres humains. Elle est souvent considérée comme le sens le plus important, car elle nous permet de percevoir et d'interagir avec le monde qui nous entoure de manière significative. La vision nous offre un accès direct à une grande quantité d'informations sur notre environnement comme le fait de voir les couleurs, les formes, les mouvements et les détails des objets et des personnes qui nous entourent. Sans oublier la perception visuelle qu'elle nous permet de naviguer dans l'espace, de reconnaître les visages, de lire, d'apprécier l'art visuel et de participer à de nombreuses activités quotidiennes. C'est pourquoi dans une société généralement basée sur la capacité de voir, la déficience visuelle a des conséquences majeures pour ses personnes.

De nos jours, le nombre de personnes malvoyantes augmente de manière inquiétante. L'Organisation Mondiale de la Santé estime que le nombre de personnes atteintes de déficience visuelle (vision avec la correction portée) s'élève à 285 millions, parmi elles, 246 millions souffrent de basse vision, et selon les estimations 39 millions sont aveugles. Les raisons pour qu'une personne souffre d'une déficience visuelle sont plusieurs. Certaines d'elles peuvent être héritées génétiquement, voir même des maladies oculaires telles que la cataracte, des blessures oculaires causées par des accidents, certaines maladies systémiques telles que le diabète, et l'exposition à des produits chimiques toxiques ou des radiations ou des rayonnements UV excessifs.

L'accès à l'information visuelle peut être limité pour les personnes malvoyantes. La lecture de textes imprimés, la perception des petits objets, ainsi que reconnaître et comprendre ce qui se trouve et les scènes présentées devant eux. C'est ici que survient le rôle des technologies modernes pour trouver des solutions à ces problèmes. Parmi ces technologies on trouve l'Intelligence Artificielle et plus précisément la Vision par Ordinateur qui essaie d'imiter le fonctionnement de l'œil humain.

C'est dans ce sens que vient notre travail de mémoire, où nous proposons une application de détection des objets tout en offrant la possibilité d'obtenir la description des scènes, et cela à base de la Vision par Ordinateur. De cette façon, nous pouvons aider les personnes malvoyantes à connaître l'emplacement des objets physiques, et d'avoir une description du monde qui les entoure. Bien entendu, cela aide cette catégorie de personnes d'atteindre un taux d'indépendance plus élevé.

Après cette introduction générale de notre travail, nous présentons la structuration de ce mémoire qui est élaboré en quatre chapitres suivis par une conclusion générale comme suite :

Chapitre 1 : Déficience Visuelle et Aides Technologiques pour les Malvoyants

Ce chapitre présente une étude générale de la déficience visuelle et la contribution des technologies d'assistance et leur rôle dans l'amélioration de la qualité de vie des personnes malvoyantes.

Chapitre 2 : Intelligence Artificielle et Vision par Ordinateur

Dans ce chapitre nous présentons un état d'art qui décrit le domaine de l'Intelligence Artificielle, l'apprentissage automatique, l'apprentissage profond et surtout la Vision par Ordinateur ainsi que ses applications.

Chapitre 3 : Méthodologie de Détection et de Description des Objets

Le troisième chapitre présente l'architecture du système proposé, ainsi que les deux méthodologies principales au niveau de notre système qui sont la détection d'objets et la description de l'environnement.

Chapitre 4 : Implémentation

Le dernier chapitre est consacré à la mise en œuvre du système présenté. Les outils utilisés, suivies d'une présentation de l'implémentation des différents composants du système et leurs interfaces tout en suivant un scénario d'exécution.

Chapitre 01

Déficiência Visuelle et Aides Technologiques pour Les Malvoyants

1. Introduction

La déficience visuelle est un handicap qui peut gravement affecter la qualité de vie des personnes concernées. Les personnes malvoyantes sont particulièrement confrontées à un vrai défi social et moral et des difficultés pour effectuer les activités quotidiennes telles que lire, communiquer ou se déplacer dans leur environnement. Cependant, les progrès technologiques récents ont donné aux personnes malvoyantes de nouvelles façons de communiquer, d'apprendre et de travailler. Les technologies dédiées aux personnes malvoyantes sont de plus en plus nombreuses et complexes, offrant des solutions pour compenser les problèmes de vision.

Ce premier chapitre vise à introduire les problématiques de la déficience visuelle chez les personnes malvoyantes et les enjeux liés à l'utilisation de la technologie pour les aider à surmonter ces difficultés. Nous allons également examiner les différentes technologies disponibles et leur influence sur la vie des personnes malvoyantes.

2. Vision humaine

Dans une société qui repose principalement sur la capacité à voir, la vision occupe une place importante dans toutes les étapes de la vie.

La vision est le sens le plus important et le plus développé chez l'être humain, il constitue à lui seul 80 % des perceptions de notre environnement. Mais c'est aussi le sens le plus immature à la naissance. La plupart des paramètres visuels se développent durant cette période et notamment lors de la première année. Le développement de cette fonction visuelle dépend du développement anatomique des différentes structures qui composent le système optique mais aussi de l'expérience visuelle réalisée dans les premiers mois de vie. [2]

L'œil est l'organe principal du système visuel, qui capte les images et les transforme en signal électrique vers le nerf optique. Ce signal est ensuite « traduit » par le cerveau, au niveau du cortex visuel, qui nous renvoie l'image traitée et permet ainsi l'interprétation de notre environnement. [3]

2. 1. Importance de la vision

Premier de nos cinq sens, la vue joue un rôle important dans tous les aspects de notre vie. Elle est essentielle pour les interactions sociales et interpersonnelles et dans les communications face à face, où l'information est également transmise par le langage non verbal tel que les gestes et les expressions faciales.

Dès la naissance, elle est essentielle au développement de l'enfant. Pour le nourrisson, reconnaître ses parents, les membres de sa famille et les personnes qui s'occupent de lui et échanger visuellement avec eux facilitent son développement social et cognitif et l'acquisition des compétences motrices, de la coordination et de l'équilibre. De la petite enfance à l'adolescence, la vision permet d'avoir facilement accès aux supports éducatif et est un élément charnière de la réussite scolaire. La vision favorise le développement de compétences sociales

: elle contribue à nouer plus facilement des amitiés, à renforcer l'estime de soi et à préserver le bien-être. Elle joue également un rôle important dans la participation aux activités sociales et sportives essentielles au développement physique, à la santé physique et mentale, à l'identité personnelle et à la socialisation. A l'âge adulte, la vision facilite l'intégration dans la vie active, procurant des avantages économiques et un sentiment d'identité. Elle contribue aussi à l'épanouissement dans de nombreux autres domaines de la vie souvent conçus autour de la capacité à voir, tels que le sport et les activités culturelles. Plus tard au cours de la vie, la vision permet de garder une vie sociale et son indépendance et elle facilite la prise en charge d'autres problèmes de santé. Elle participe aussi à préserver la santé mentale et un certain bien-être, qui sont tous deux meilleurs chez les personnes ayant une bonne vision. [1]

3. Déficience visuelle

La déficience correspond à un problème dans la fonction organique ou la structure anatomique, tel un écart ou une perte importante elle est définie par l'OMS comme « toute perte de substance ou altération d'une structure ou fonction psychologique, physiologique ou anatomique ». [4]

Les déficiences visuelles surviennent lorsqu'une maladie oculaire affecte le système visuel et une ou plusieurs fonctions de la vision [1]. C'est une insuffisance ou une absence d'image perçue par l'œil. Elle correspond à une atteinte de l'œil ou des voies visuelles jusqu'au système cérébral. Ces atteintes peuvent être congénitales ou acquises : accidents ou maladies, telles que le diabète, la DMLA ou le glaucome. La déficience peut porter sur l'acuité visuelle (pourcentage restant par rapport à la vision normale) et/ou sur le champ visuel, d'un œil ou des 2 yeux. [5]

Autrement dit, la déficience visuelle désigne les troubles liés à la fonction visuelle, qui persistent après traitements (thérapeutiques, médicaux, chirurgicaux...). Elle est définie à l'aide de deux critères que sont l'état du champ visuel (étendue de l'espace qu'un œil peut saisir) et la mesure de l'acuité visuelle (aptitude d'un œil à apprécier les détails). La déficience visuelle n'est pas seulement une altération anatomique et/ou physiologique, mais également psychologique. Elle est donc un problème de santé public, sur plusieurs plans. [11]

Les répercussions de la déficience visuelle dépendent de nombreux facteurs, par exemple : la disponibilité des interventions de prévention et de traitement, l'accès à la réadaptation visuelle (y compris les produits d'assistance tels que les lunettes et les cannes blanches), les problèmes éventuellement rencontrés en raison de l'inaccessibilité des bâtiments, des transports et des informations. [6]



Figure 1: Personne non-voyante.

3.2. Types de déficiences visuelles

Il existe un certain nombre de classifications qui se concentrent sur la déficience visuelle et visent à fournir un langage commun pour la situation globale du patient, parmi ces classifications il y a :

L'Onzième Classification internationale des maladies (2018) qui distingue deux types de déficience visuelle, selon que la vision de loin ou la vision de près est affectée.

Déficiences affectant la vision de loin :

- Légère – acuité visuelle comprise entre 6/12 et 6/18
- Modérée – acuité visuelle comprise entre 6/18 et 6/60
- Sévère – acuité visuelle comprise entre 6/60 et 3/60
- Cécité – acuité visuelle inférieure à 3/60.

Déficiences affectant la vision de près :

- Acuité visuelle inférieure à N6 ou à M.08 à 40 cm. [6]

L'Organisation mondiale de la santé (OMS) a classifié les déficiences visuelles en cinq catégories en fonction de l'acuité et du champ visuel. Cette classification permet de définir le degré de malvoyance. [7]

Tableau 1: Classification du handicap visuel suivant l'OMS. [7]

Catégorie OMS	Caractéristiques de l'acuité visuelle	Type d'atteinte visuelle (CIM-10*)	Type de déficience visuelle (CIF**)
Catégorie I	Acuité visuelle corrigée comprise entre 3/10 et 1/10 avec un champ visuel d'au moins 20°	Basse vision ou malvoyance	Déficience moyenne
Catégorie II	Acuité visuelle corrigée comprise entre 1/20 et 1/10		Déficience sévère
Catégorie III	Acuité visuelle corrigée comprise entre 1/50 et 1/20 ou champ visuel compris entre 5° et 10°	Cécité	Déficience profonde
Catégorie IV	Acuité visuelle inférieure à 1/50 mais perception lumineuse préservée ou champ visuel inférieur à 5°		Déficience presque Totale
Catégorie V	Cécité absolue, absence de perception lumineuse.		Déficience totale

3.3. Causes de déficiences visuelles

Il existe des variations substantielles dans les causes entre les pays et à l'intérieur des pays, en fonction de la disponibilité des services de soins oculaires, de leur accessibilité économique et des connaissances de la population en matière de soins oculaires. Par exemple, la part des déficiences visuelles imputables à la cataracte est plus élevée dans les pays à revenu faible ou intermédiaire que dans les pays à revenu élevé. Dans ces derniers, des maladies telles que le glaucome et la dégénérescence maculaire liée à l'âge sont plus fréquentes. [6]

Chez l'enfant, les causes de la déficience visuelle varient considérablement d'un pays à l'autre. Par exemple, dans les pays à revenu faible, la cataracte congénitale est l'une des principales causes, tandis que dans les pays à revenu intermédiaire, c'est plus souvent la rétinopathie du prématuré. Comme dans les populations adultes, le défaut de réfraction non corrigé demeure l'une des principales causes de déficience visuelle chez l'enfant dans tous les pays. [6]

Au niveau mondial, les principales causes de déficience visuelle sont :

3.3.1. Cataracte

Première cause de cécité dans le monde, est maintenant la chirurgie la plus pratiquée, avec des taux de succès jamais égalés jusqu'à aujourd'hui [8]. Le groupe d'experts sur la perte de vision (VLEG) estime que plus de 17 millions de personnes sont bilatéralement aveugles à cause de la cataracte dans le monde en 2020, ce qui représente 40 % de tous les cas de cécité dans le monde. Bien que la plupart des cas de cataracte soient liés au processus de vieillissement, il arrive que des enfants naissent avec cette maladie, ou qu'une cataracte se développe après des blessures oculaires, une inflammation et quelques autres maladies oculaires. [9]



Figure 2: Vue avec une cataracte.[17]

3.3.2. Dégénérescence maculaire liée à l'âge (DMLA)

Première cause de déficience visuelle dans les pays développés, est devenue en 10 ans l'objet des plus grandes dépenses de santé en ophtalmologie. [8]

C'est une maladie qui touche la zone centrale de la rétine (macula) à l'arrière de l'œil. Les principaux facteurs de risque de la DMLA sont l'âge, les facteurs génétiques et le tabagisme. La DMLA touche généralement les personnes de plus de 60 ans, mais peut survenir plus tôt. De nombreuses recherches se sont concentrées sur le rôle du régime alimentaire, de l'exposition à la lumière et de l'association avec les maladies cardiovasculaires et leurs facteurs de risque, mais les effets de ces facteurs de risque sont moins certains. Il n'existe actuellement aucun traitement efficace contre la DMLA sèche. Il existe des preuves que les suppléments de vitamines antioxydants peuvent ralentir la progression de la DMLA vers le stade avancé de la maladie et la perte de la vue. [9]



Figure 3: Vue avec une DMLA. [20]

3.3.3. Glaucome

Le glaucome est une maladie oculaire grave, fréquente (1,2 million de personnes atteintes en France) qui entraîne une détérioration lente du nerf optique, aboutissant à une perte progressive du champ visuel puis parfois à la cécité, si elle n'est pas dépistée ou traitée. [10]

Le glaucome est la troisième cause de cécité et la quatrième cause de perte de vision dans le monde. On estime qu'à l'heure actuelle, au moins 3 millions de personnes sont aveugles et 4 millions souffrent d'une déficience visuelle modérée à sévère due au glaucome (Adelson et al, 2020). Cependant, la plupart des formes de glaucome ne présentent pas de symptômes dans les premiers stades et les patients ne se présentent donc souvent pour le traitement qu'après avoir perdu la vue. [9]



Figure 4: Vue avec un glaucome. [17]

3.3.4. Rétinopathie diabétique

Le diabète augmente le risque de toute une série de maladies oculaires, mais la principale cause de cécité associée au diabète est la rétinopathie diabétique (RD). La RD endommage les vaisseaux sanguins à l'intérieur de la rétine, à l'arrière de l'œil. Elle affecte généralement les deux yeux et peut entraîner une perte de vision si elle n'est pas traitée. Comme la rétinopathie diabétique est initialement asymptomatique, de nombreuses personnes atteintes de diabète ne savent pas que leur état, s'il n'est pas pris en charge, peut affecter leur vision et conduire à la cécité. La plupart des patients qui développent une rétinopathie diabétique ne présentent aucun symptôme jusqu'aux stades très avancés (il peut alors être trop tard pour un traitement efficace). [9]



Figure 5: Paysage vu avec une rétinopathie diabétique. [17]

3.3.5. Trachome

Le trachome est la première cause infectieuse de cécité dans le monde et l'une des 20 maladies tropicales négligées (MTN) qui touchent collectivement plus d'un milliard de personnes parmi les plus pauvres de la planète.

En 2021, on sait que le trachome est un problème de santé publique dans 44 pays, touchant des communautés ayant un accès limité aux soins de santé et à d'autres infrastructures essentielles, notamment l'eau, l'assainissement et l'hygiène.

L'Organisation mondiale de la santé estime que 1,9 million de personnes sont aveugles ou souffrent d'une déficience visuelle due au trachome et que deux millions de personnes doivent être opérées d'urgence pour traiter le trichiasis trachomateux. [9]

4. Impact de la déficience visuelle

4.1. Impact individuel

Les jeunes enfants atteints d'une déficience visuelle grave à un stade précoce peuvent éprouver un retard de développement moteur, psychologique, social, cognitif et du langage qui aura des conséquences tout au long de leur vie. Les enfants d'âge scolaire ayant une déficience visuelle peuvent également avoir des niveaux de réussite scolaire inférieurs. La déficience visuelle a de graves répercussions sur la qualité de vie des populations adultes. Bien souvent, chez les adultes ayant une déficience visuelle, les taux de participation et de productivité sur le marché du travail sont plus faibles tandis que les taux de dépression et d'anxiété sont plus élevés. Dans le cas des personnes âgées, une déficience visuelle peut contribuer à l'isolement social, à la difficulté à marcher, à un risque plus élevé de chutes et de fractures, et à une plus grande probabilité d'entrée précoce dans un établissement pour personnes âgées. [6]

4.2. Impacts psychologiques

L'atteinte de la vision ramène l'individu à une notion de perte. Aussi, selon sa solidité ou sa fragilité, en lien avec son histoire, la déficience visuelle représente un terrain à fort risque d'engendrer un vécu dépressif. En effet, l'émotionnel de la dépression peut fragiliser la vision et la perte va engendrer l'état dépressif. Ce lien entre déficience visuelle et dépression fait l'objet de nombreuses publications de psychiatrie, gériatrie et d'ophtalmologie. [1]

La première étude d'envergure effectuée en 1998 par William et al. Montre la profonde réduction de la qualité de vie des patients atteints de dégénérescence maculaire liée à l'âge (DMLA). En 1999, Brody et al. Montrent à partir d'une étude portant sur 151 personnes atteintes de DMLA sévère (avec acuité visuelle [AV] < 3/10) que 33 % de ces personnes présentent une dépression, soit plus du double de ce qui est observé dans une population de cet âge. En parallèle, les études qui demandent aux patients combien d'années de vie ils sont prêts à perdre en échange du retour à une vision parfaite montrent l'impact certain d'une atteinte visuelle sur la qualité de vie. En effet, la plupart des personnes malvoyantes sont prêtes à échanger plus d'un tiers de leur vie restante. [12]

4.3. Impact économique

La déficience visuelle représente un immense fardeau financier à l'échelle planétaire : chaque année dans le monde, les pertes de productivité associées aux déficiences visuelles sont estimées à 411 milliards USD. Ce chiffre est bien supérieur aux dépenses qu'il faudrait engager pour répondre aux besoins non satisfaits en matière de déficience visuelle (estimées à 25 milliards USD environ). [6]

5. Aides technologiques pour les malvoyants

Les progrès rapides de la technologie ouvrent de nouvelles opportunités aux malvoyants, en leur fournissant des outils et des aides technologiques pour améliorer leur accès à l'information et faciliter leur communication. Ces progrès ont considérablement amélioré la

qualité de vie des malvoyants, leur offrant de nouvelles possibilités d'apprendre, de travailler et d'interagir socialement.

5.1. Technologies d'assistantes

On propose ici de présenter certaines des dernières nouveautés de ces technologies :

5.1.1. Canne blanche électronique

La canne blanche électronique est un outil de détection des obstacles adapté aux personnes aveugles comme aux personnes malvoyantes. Elle combine des principes mécaniques et optroniques qui confèrent aux utilisateurs une protection sur toute la hauteur du corps et favorise une meilleure perception de l'environnement. Cette aide technique améliore la mobilité des personnes en situation de déficience visuelle, en fluidifiant et sécurisant leurs trajets, à condition que son utilisation fasse partie intégrante d'un projet de locomotion. Elle n'est en aucun cas un outil de navigation : elle n'indique ni les directions à emprunter ni la distance restante entre le point de départ et d'arrivée. C'est l'utilisateur qui contrôle son déplacement (choix des directions, lieu et moment de traversée...). [13]



Figure 6: Canne intelligente Rango de la start-up GoSense.

5.1.2. Plage braille

C'est un système d'écriture et de lecture tactile à points saillants, à l'usage des personnes aveugles ou fortement malvoyantes. Le braille existait avant l'apparition des techniques vocales et informatiques. Le braille est fondamental, pour suivre une scolarité, faire des études littéraires ou scientifiques et s'intégrer socialement et professionnellement. Pour les déficiences visuelles liées à l'âge telle la dégénérescence maculaire liée à l'âge (DMLA), son apprentissage ne se montre pas toujours pertinent. Il nécessite une sensibilité tactile fine, qui n'est plus chez bon nombre de personnes âgées. Une adhésion au projet et une grande pugnacité sont indispensables pour l'adulte. [14]

Une plage braille est un périphérique d'ordinateur permettant à l'utilisateur d'avoir un affichage braille en temps réel des informations présentes à l'écran. L'utilisateur peut également envoyer des commandes à l'ordinateur grâce à différentes touches. [7]



Figure 7: BrailleNoteTouch Plus, tablette braille tactile adaptée aux personnes aveugles.

5.1.3. Loupes électroniques

Elles intègrent un petit écran, leur taille varie de 3 à 7 pouces, 5 pouces paraît le minimum à recommander. Elles favorisent la lecture de tout document, sont de manipulation simple et facilement transportable. Elles se posent sur le texte.

Une loupe électronique « VOXIONE » a obtenu le « Silmo d'or » lors du salon des opticiens 2019. À son écran de 6,3 pouces, elle intègre une reconnaissance vocale et d'autres fonctions telles un lecteur de codes-barres, des logiciels, un smartphone intégré. [14]



Loupe électronique Léa



Loupe électronique 5 pouces HD Alizée

Figure 8: Exemples des Loupes électroniques. [17]

5.1.4. Téléagrandisseurs et Vidéoagrandisseurs

Systèmes plus imposants, ils sont recommandés pour des travaux prolongés et offrent des possibilités multiples de vision rapprochée. Ils sont adaptables à de nombreuses situations professionnelles. Ils conduisent à dissocier la coordination œil-main ce qui modifie les stratégies de lecture. Ils se présentent sous plusieurs formes : les portables de 16 à 22 pouces, les fixes de 32 à 37 pouces, les « parlants », tel le Narratello 22/37 pouces qui offre, en plus de

la visualisation, une lecture automatique et en continu de pages entières de textes par une voix intégrée à l'appareil. Pour une rentabilité optimale, il ne faut pas négliger une installation adéquate limitant les troubles posturaux, le port de la lunette adaptée à la distance d'utilisation est indispensable. [14]



Figure 9: Exemples des téléagrandisseurs. [17]

5.1.5. Lunettes à reconnaissance visuelle

C'est un appareil portable intuitif avec une caméra intelligente à placer sur la monture des lunettes de la personne. L'appareil utilise la puissance de la vision artificielle pour aider les personnes qui vivent avec une perte de vision. Ce sont des lunettes interactives. La personne indique ce qu'elle veut voir, le système traite l'image et lit le texte. [14]



Figure 10: Lunettes MyEye d'OrCam. [14]

5.1.6. Vocale Presse

Vocal Presse est un logiciel utilisant la synthèse vocale pour offrir un accès audio à un ensemble de périodiques le jour même de leur sortie. C'est bien ici la nécessité de mettre très

vite à disposition des textes audio tout en maîtrisant les coûts de production qui fait de la synthèse vocale l'outil idéal. L'abonné à Vocal Presse utilise un ordinateur connecté à Internet pour avoir accès au texte numérisé. Une interface vocalisée particulièrement simple d'utilisation permet d'accéder aux articles des périodiques lus par une voix de synthèse. Cet outil bouleverse l'accès à l'information de certaines personnes déficientes visuelles en leur offrant pour la première fois la possibilité d'accéder aux quotidiens. [15]

5.2. Applications mobile existantes pour les malvoyantes

Puisque dans ce thème, nous avons réalisé une application mobile comme solution pour les malvoyants, nous allons parler dans cette section sur quelques applications mobiles existantes qui sont destinées aux personnes aveugles ou malvoyantes.

Ces applications sont conçues pour rendre les fonctionnalités et les avantages des smartphones accessibles à tous. En utilisant des fonctionnalités telles que la synthèse vocale, les vibrations et les commandes gestuelles pour compenser le manque de perception visuelle.

Explorons certaines de ces applications qui améliorent le quotidien et la mobilité des personnes aveugles ou malvoyantes :

5.2.1. Be MyEyes

L'application Be MyEyes, créée en 2015 par le Danois Hans Jørgen Wiberg, lui-même malvoyant, met en relation une communauté mondiale de malvoyants avec 4,5 millions bénévoles voyants. Grâce à un appel vidéo en direct, les bénévoles fournissent aux utilisateurs aveugles et malvoyants une assistance visuelle pour, par exemple, distinguer les couleurs des vêtements, vérifier si les lumières sont allumées, lire des étiquettes, faire les courses ou préparer le dîner. Cette application est disponible dans plus de 185 langues. Les services de Be MyEyes sont gratuits, le patient peut appeler à tout moment de la journée et les appels ne sont limités ni dans la durée, ni dans le nombre. Compte tenu de la taille impressionnante de la communauté de bénévoles, la majorité des appels est prise dans les 30 secondes. Cette application altruiste a déjà remporté plusieurs prix et distinctions depuis sa création. [16]

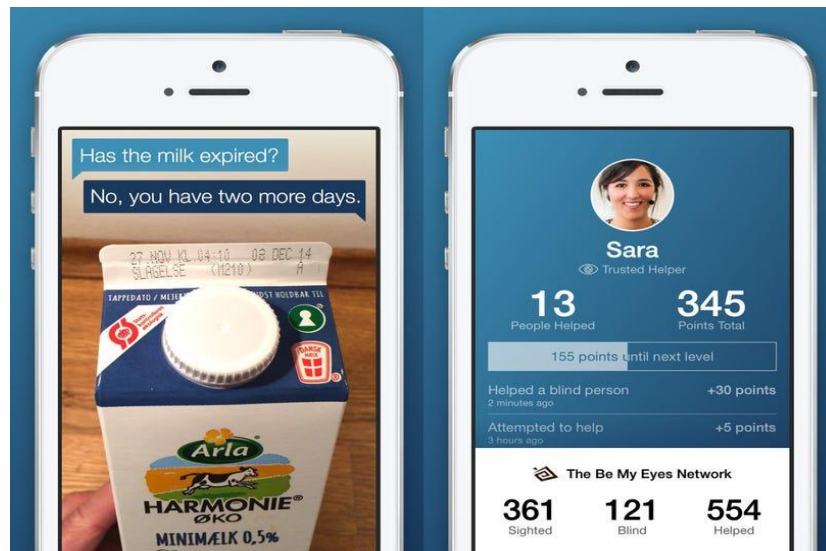


Figure 11: Utilisation de l'application BeMyEyes.

5.2.2. Seeing AI

SeeingAI est avant tout un projet de recherche. L'application du même nom permet de lire un texte court, un document ou un texte manuscrit. Elle peut aussi reconnaître une scène, des produits (grâce à leurs codes-barres), des couleurs, de la monnaie, la luminosité ambiante et même des personnes. Elle est totalement gratuite mais ne fonctionne que sur IOS. [16]



Figure 12: Application mobile Seeing AI pour les personnes malvoyantes.

5.2.3. Envision AI

Permet également la lecture d'un texte reconnu par la caméra ou après importation d'un fichier. Cet outil peut également décrire une scène, des couleurs ou un produit via le code-barres. Enfin, il possède une fonctionnalité de recherche des personnes ou des objets dans la pièce. Disponible sur IOS et Android, Envision AI propose une version d'essai gratuite pendant 14 jours, puis des abonnements. [16]

5.2.4. Nav by ViaOpta

Conçue par Novartis, est une application de guidage GPS adaptée à la malvoyance. Le graphisme de l'application est simplifié et adapté par rapport à une application GPS classique. Un système de guidage audio permet au patient de se déplacer à pied, mais également de connaître sa position en temps réel. Le patient reçoit une notification vocale et par vibration à chaque changement de direction, ou à chaque intersection. Un raccourci par reconnaissance vocale est activable et des sites géographiques favoris sont enregistrables. [16]

5.2.5. BigLauncher

Disponible uniquement sur Android, transforme l'écran d'accueil en une interface épurée avec des boutons et des textes agrandis. L'application permet ainsi une lisibilité maximale simplifiant l'utilisation des smartphones pour les personnes âgées mais aussi malvoyantes. Il est également possible de masquer certaines options, telles que les paramètres ou les notifications, afin qu'il ne reste plus que le strict nécessaire : téléphone, sms, photos, appareil photo ou appel d'urgence. [16]



Figure 13: BigLauncher simplifie et améliore la visualisation de l'écran d'accueil. [16]

5.2.6. Voice Over /TalkBack

Voice Over, pour IOS, et TalkBack, sur Android, sont des utilitaires directement disponibles et activables à partir des paramètres de ces systèmes d'exploitation. Ce sont des lecteurs d'écran qui décrivent à voix haute ce qui est affiché, comme l'appel d'un contact ou la notification en cours. Ils énoncent également les éléments sur lesquels le doigt de l'utilisateur est posé. Ces 2 outils peuvent être activés et utilisés même sans voir l'écran, grâce à des gestes reconnus par le smartphone. [16]

6. Conclusion

En conclusion, dans ce premier chapitre, nous avons introduit la problématique de la déficience visuelle en commençant par un rappel sur la vision humaine et son importance. Et passant par une définition de la déficience visuelle, et ses différents types, causes et conséquences sur les individus et sur l'économie.

Nous avons aussi exploré les différentes aides technologiques existantes pour les malvoyants, notamment les applications mobiles, qui ont permis d'améliorer leur qualité de vie.

Dans le prochain chapitre, nous allons présenter l'état de l'art des technologies d'apprentissage automatique et profond que nous avons utilisé pour développer et implémenter notre système.

Chapitre 02

Intelligence Artificielle et Vision par Ordinateur

1. Introduction

L'Intelligence Artificielle (IA) est un domaine en perpétuelle évolution dont l'objectif est de développer des systèmes informatiques capables de reproduire des processus intelligents similaires à ceux réalisés par les êtres humains. Depuis ses débuts, l'Intelligence Artificielle a connu une avancée significative, transformant notre mode de vie, notre travail et notre interaction avec la technologie. Dans la dernière décennie, l'évolution rapide du matériel informatique a ouvert de nouvelles possibilités à l'Intelligence Artificielle, lui permettant d'exécuter rapidement des algorithmes très complexes. L'Apprentissage Automatique, en particulier, a révolutionné l'Intelligence Artificielle en permettant aux machines d'apprendre à partir des données et d'améliorer les performances de manière autonome. Des innovations telles que les réseaux de neurones artificiels, les algorithmes d'Apprentissage Profond et les techniques de Traitement du Langage Naturel ont propulsé l'Intelligence Artificielle vers de nouveaux sommets. Parmi les nombreux domaines de l'Intelligence Artificielle, la Vision par Ordinateur qui joue un rôle crucial en donnant aux machines la capacité de comprendre et d'interpréter les informations visuelles provenant d'images et de vidéos.

Au cours de ce chapitre, nous présentons une vue d'ensemble des technologies abordées précédemment. Nous commençons tout d'abord l'Intelligence Artificielle, suivi par la Vision par Ordinateur et le Traitement du Langage Naturel par la suite. Bien entendu, nous présentons la nature de leur interconnexion et nous soulignons les multiples avantages offerts par ces technologies modernes et populaires, qui ont pour objectif d'améliorer et de simplifier la vie quotidienne des individus.

2. Intelligence artificiel

L'Intelligence Artificielle (IA) est la simulation de l'intelligence humaine dans des machines programmées pour penser et apprendre comme les humains. Il s'agit d'une branche de l'informatique axée sur le développement de systèmes intelligents capables d'effectuer des tâches qui nécessitent normalement l'intelligence humaine, telles que la perception, le raisonnement, l'apprentissage, la résolution de problèmes et la prise de décision.

Le terme "Intelligence Artificielle" n'a été inventé qu'en 1955. En 1956, John McCarthy et ses collaborateurs ont tenu une conférence intitulée « Dartmouth Summer Research Project on Artificial Intelligence ». Cela a donné naissance à l'Apprentissage Automatique, à l'apprentissage en profondeur, à l'analyse prédictive et plus récemment à l'analyse prescriptive. Un nouveau domaine de recherche appelé science des données a également émergé [18].

L'IA comprend le développement d'algorithmes, de modèles et de techniques qui permettent aux machines de comprendre, d'interpréter et de manipuler des données pour effectuer des tâches spécifiques ou atteindre des objectifs spécifiques. Ces tâches vont de tâches simples telles que la reconnaissance vocale et la classification d'images à des activités complexes telles que le Traitement du Langage Naturel, la conduite autonome et le diagnostic médical

Les systèmes d'Intelligence Artificielle peuvent être classés en deux types : l'Intelligence Artificielle étroite (également appelée IA faible) et l'Intelligence Artificielle générale (également appelée IA forte). Au sens étroit, l'Intelligence Artificielle fait référence à des systèmes conçus pour effectuer des tâches spécifiques et exceller dans des domaines limités. L'Intelligence Artificielle en général, d'autre part, vise à avoir une intelligence et une adaptabilité de type humain, quel que soit le domaine ou le contexte. [19] [20]

Le domaine de l'Intelligence Artificielle englobe plusieurs sous-domaines, notamment l'Apprentissage Automatique, le Traitement du Langage Naturel, la vision par ordinateur, la robotique, les systèmes experts et les réseaux de neurones.

L'Intelligence Artificielle est appliquée dans un large éventail d'industries, notamment la santé, la finance, les transports, la fabrication, le divertissement et bien d'autres, révolutionnant notre façon de vivre et de travailler.

3. Apprentissage Automatique

L'Apprentissage Automatique, aussi appelé Apprentissage Machine ou Machine Learning en anglais, est une discipline de l'Intelligence Artificielle comme le montre la Figure 14, qui repose sur des approches mathématiques et statistiques. L'objectif de l'Apprentissage Automatique est de permettre aux ordinateurs d'apprendre à partir de données, sans nécessiter une programmation explicite pour chaque tâche.

L'Intelligence Artificielle (IA) et l'Apprentissage Automatique (Machine Learning) sont étroitement liés, mais ils ne sont pas exactement la même chose. Si l'Intelligence Artificielle est un concept visant à simuler un ou des comportements humains, l'Apprentissage Automatique n'est qu'une méthode pour atteindre la création d'une Intelligence Artificielle. Ainsi, l'Intelligence Artificielle n'est possible qu'avec l'usage de plusieurs méthodes, dont l'Apprentissage Automatique. [21]

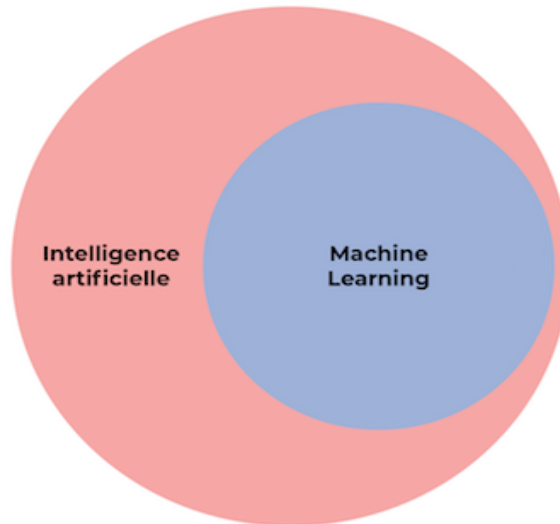


Figure 14: Intelligence Artificielle et Machine Learning.

3.1. Entraînement d'un modèle

Comme la figure 15 le montre, le processus d'apprentissage automatique implique généralement les étapes suivantes :

1. Collecte de données : Les données pour cette tâche particulière sont rassemblées de diverses sources.
2. Prétraitement des données : Les données sont nettoyées, transformées et préparées pour l'Apprentissage Automatique. Cela peut inclure des étapes comme la normalisation des valeurs, le nettoyage des données manquantes et la sélection des caractéristiques pertinentes.
3. Sélection d'un modèle d'apprentissage : Un modèle d'apprentissage automatique approprié est choisi en fonction de la nature de la tâche et des données disponibles. Les réseaux de neurones, les arbres de décision, les machines à vecteurs de support, les algorithmes de forêt aléatoire et d'autres modèles sont fréquemment utilisés.
4. Entraînement du modèle : Le modèle sélectionné est entraîné à partir des données d'entraînement, où il apprend à détecter des schémas et à établir des relations entre les variables.
5. Évaluation du modèle : Le modèle entraîné est évalué à l'aide de données de test distinctes pour estimer sa performance et sa capacité à généraliser sur de nouvelles données.
6. Réglage du modèle : Si nécessaire, les hyper paramètres du modèle sont ajustés pour améliorer ses performances. [29]

Une fois le modèle entraîné et évalué, il peut être utilisé pour faire des prédictions ou classification de nouvelles données. De nombreux domaines, y compris la reconnaissance

vocale, la recommandation de produits, la détection de fraudes, l'analyse prédictive, la Vision par Ordinateur, etc., utilisent l'Apprentissage Automatique.

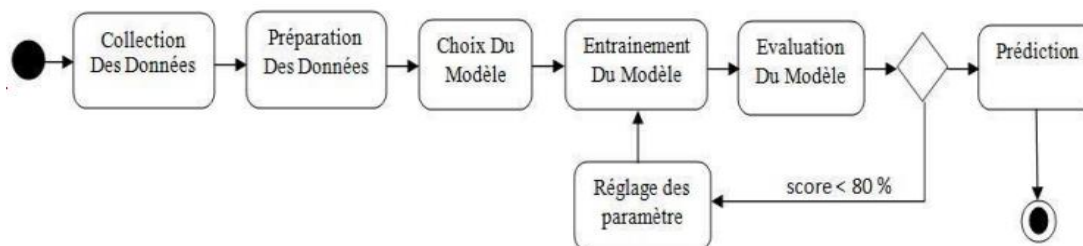


Figure 15: Etapes d'entraînement d'un modèle.

3.2. Types d'Apprentissage Automatique

Il existe plusieurs types d'apprentissage automatique, qui diffèrent par la manière dont les données sont utilisées et les types de tâches qu'ils permettent de résoudre. Voici les principaux types d'apprentissage automatique :

3.2.1. Apprentissage Supervisé (Supervised Learning)

Le modèle est entraîné sur un ensemble de données étiquetées dans l'apprentissage supervisé, où chaque exemple de données est lié à une étiquette ou à une valeur de sortie connue. L'objectif est d'utiliser les exemples étiquetés pour prédire la sortie de nouvelles données. La classification (par exemple, classer les e-mails en spam ou non spam) et la régression (par exemple, prédire le prix d'une maison en fonction de ses caractéristiques) sont des exemples courants d'apprentissage supervisé. [22] [26]

3.2.2. Apprentissage non Supervisé (Unsupervised Learning)

L'apprentissage non supervisé se concentre sur l'analyse de données non étiquetées, c'est-à-dire sans informations de sortie associées. L'objectif est de découvrir des structures, des modèles ou des regroupements intégrés dans les données. La classification non supervisée (par exemple, la division des clients en groupes en fonction de leurs comportements) et la réduction de dimensionnalité (par exemple, la représentation des données dans un espace de dimensions réduites) sont des exemples d'algorithmes d'apprentissage non supervisé. [23] [26]

3.2.3. Apprentissage par Renforcement (Reinforcement Learning)

L'Apprentissage par renforcement consiste à former un agent à prendre des décisions dans un environnement en constante évolution en interagissant avec cet environnement. En fonction de ses actions, l'agent reçoit des récompenses ou des sanctions, ce qui lui permet d'apprendre à maximiser une mesure de performance spécifique, connue sous le nom de récompense cumulée. Les jeux, les robots autonomes et la gestion des ressources sont des exemples d'applications de l'apprentissage par renforcement. Dans ces exemples, l'agent acquiert progressivement des

compétences en explorant diverses actions et en ajustant sa stratégie pour atteindre des résultats optimaux dans son environnement donné. [22] [25]

3.2.4. Apprentissage semi-supervisé (Semi-Supervised Learning)

L'apprentissage semi-supervisé combine des aspects de l'apprentissage supervisé et non supervisé. Pour améliorer la performance du modèle, il utilise à la fois des données étiquetées et non étiquetées. L'idée est que des informations non étiquetées peuvent guider l'apprentissage et améliorer la généralisation du modèle. [24]

3.2.5. Apprentissage par transfert (Transfer Learning)

L'apprentissage par transfert consiste à utiliser les connaissances acquises sur une tâche spécifique pour améliorer la performance sur une tâche similaire ou connexe. Au lieu d'entraîner un modèle à partir de zéro, on utilise des modèles préalablement entraînés sur de grandes quantités de données (par exemple, avec des réseaux de neurones profonds) et on les adapte à la tâche cible en utilisant des données plus spécifiques.

4. Apprentissage Profond

L'Apprentissage Profond, également appelé Deep Learning en anglais, est une branche de l'Intelligence Artificielle (IA) qui se concentre sur la construction de modèles informatiques capables d'apprendre et de s'améliorer à partir de données. C'est une technique d'apprentissage automatique basée sur des réseaux de neurones artificiels complexes.

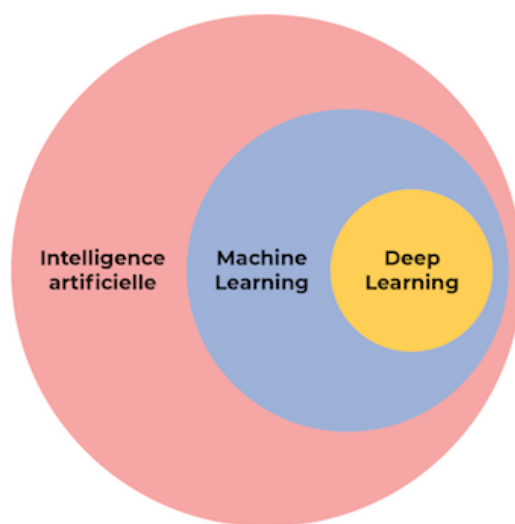


Figure 16: Apprentissage Profond et Apprentissage Automatique.

L'Apprentissage Profond est couramment employé pour des tâches impliquant la détection de motifs complexes, comme la reconnaissance vocale, la reconnaissance d'images, la traduction automatique et la prédiction de séquences. Les réseaux de neurones profonds ont la

capacité d'apprendre à extraire automatiquement des caractéristiques pertinentes des données brutes, ce qui élimine souvent le besoin de créer manuellement des caractéristiques spécifiques pour chaque tâche. La représentation plus riche et plus expressive des données est rendue possible par cette méthode, ce qui facilite la résolution de problèmes complexes et la réalisation de performances élevées dans divers domaines d'application. [27]

Afin de former un modèle d'Apprentissage Profond, on utilise un vaste ensemble de données d'entraînement. Ces données sont présentées au réseau de neurones, qui ajuste les poids des connexions entre les neurones pour réduire l'erreur de prédiction. Afin d'obtenir des modèles performants, ce processus est itératif et nécessite fréquemment une grande quantité de données et de puissance de calcul. [28]

4.1. Réseaux de neurones artificiels profonds

Les réseaux de neurones artificiels profonds, également connus sous le nom de réseaux de neurones profonds ou Deep Neural network en anglais, sont des modèles computationnels qui tirent leur inspiration du fonctionnement du cerveau humain. Ils sont utilisés dans le domaine de l'Apprentissage Profond, pour résoudre des problèmes complexes de reconnaissance, de motifs et de prise de décision.

En plus d'une couche d'entrée et d'une couche de sortie, un réseau de neurones artificiels profonds est composé de plusieurs couches de neurones interconnectés appelées couches cachées (Figure 17). Chaque neurone d'une couche est connecté à tous les neurones de la couche suivante, formant ainsi un réseau dense de connexions. [33]

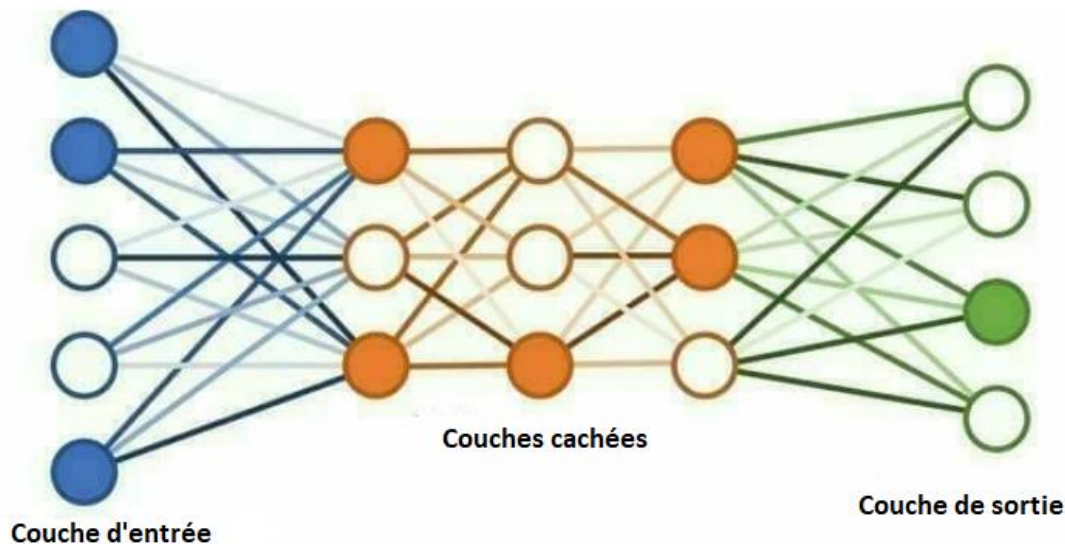


Figure 17: Réseaux de neurone.

Chaque neurone dans une couche reçoit des entrées venant des neurones de la couche précédente ou des données d'entrée. Chaque connexion entre les neurones est associée à un poids, qui détermine l'importance relative de l'entrée pour le neurone suivant. Les valeurs vont être multipliées par les poids et combinées avec la fonction de combinaison pour avoir une somme pondérée. Avant de transmettre cette valeur aux autres neurones, il va pouvoir la modifier avec la fonction mathématique appelée fonction d'activation (Figure 18). La fonction d'activation est là pour déterminer si la valeur doit passer ou non aux prochains neurones. Les poids des connexions entre les neurones sont ajustés lors de l'apprentissage pour optimiser la performance du modèle. [28]

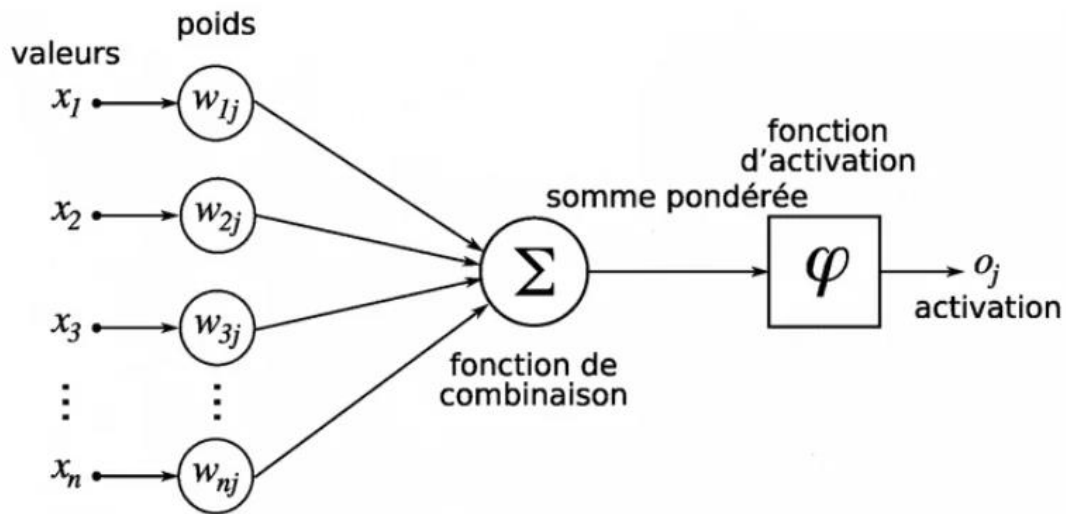


Figure 18: Composant d'un neurone artificiel.

Pour créer des modèles performants, l'apprentissage dans les réseaux de neurones profonds peut être intensif en termes de calcul et nécessite une quantité importante de données d'entraînement. Cependant, ils peuvent apprendre automatiquement les caractéristiques pertinentes des données brutes, ce qui évite souvent d'avoir à concevoir manuellement des caractéristiques spécifiques pour une tâche donnée.

L'Apprentissage Profond utilise une variété de modèles qui sont conçus pour résoudre différents types de problèmes et traiter différentes formes de données. Voici quelques-uns des modèles les plus couramment utilisés dans le domaine de l'Apprentissage Profond :

4.1.1. Réseaux de neurones multicouches MLP

Il s'agit du modèle de base de l'Apprentissage Profond, composé de plusieurs couches de neurones interconnectés comme le montre la Figure 19. Chaque neurone est connecté à tous les neurones de la couche suivante. Les tâches les plus utilisées avec MLP sont les tâches de classification et de régression. [31]

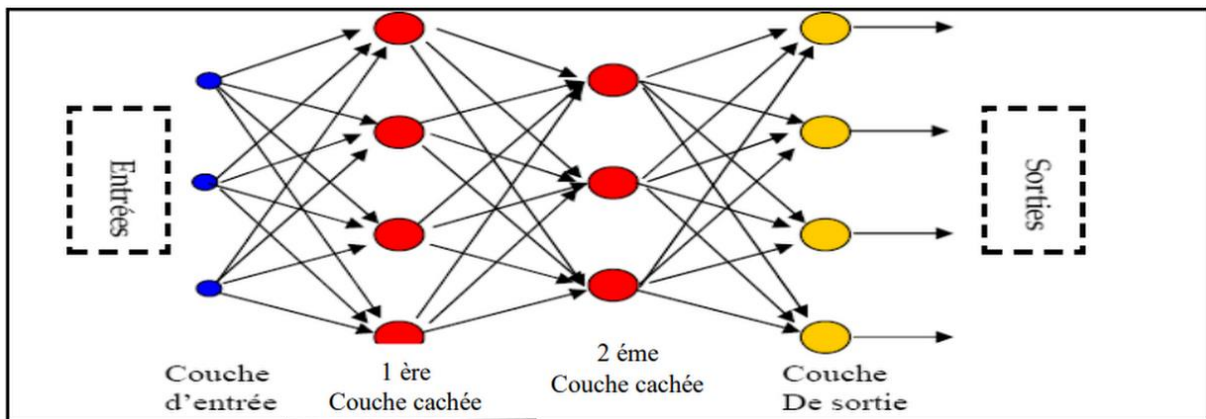


Figure 19: Réseaux de neurones multicouches.

4.1.2. Réseaux de neurones convolutés CNN

Il existe de nombreuses variantes d'architectures de réseaux de neurones Convolutifs (CNN) dans la littérature. Cependant, leurs composants de base sont très similaires. Prenons l'exemple du célèbre LeNet-5, qui se compose de trois types de couches : les couches Convolutionnelles, les couches de Pooling et les couches entièrement connectées (Figure 20, Figure 21). La couche Convolutionnelle vise à apprendre les représentations des caractéristiques des entrées. La couche de Pooling vise à obtenir une Shift-Invariance (invariance par décalage) en réduisant la résolution des cartes des caractéristiques, elle est généralement placée entre deux couches Convolutionnelles. Après plusieurs couches de convolution et de Pooling, il peut y avoir une ou plusieurs couches entièrement connectées qui visent à effectuer un raisonnement de haut niveau.

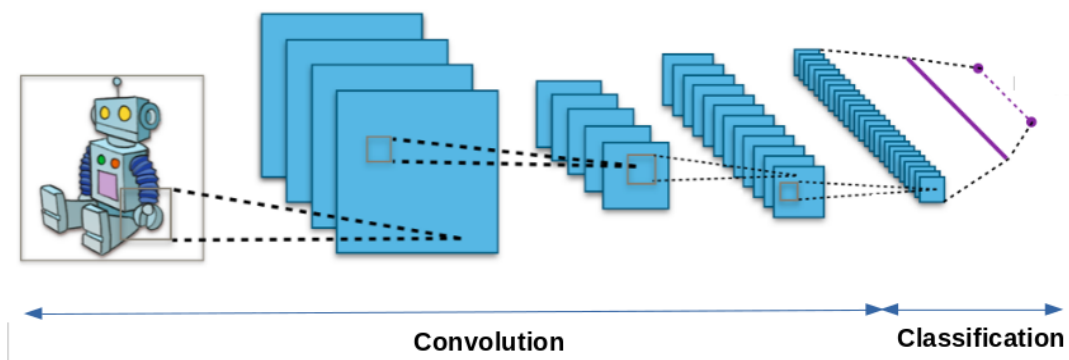


Figure 20: Architecture du CNN.

Ces couches prennent tous les neurones de la couche précédente et les connectent à chaque neurone individuel de la couche actuelle pour générer des informations sémantiques globales.

La dernière couche des CNN est une couche de sortie. Pour les tâches de classification, l'opérateur Softmax est couramment utilisé. [30]

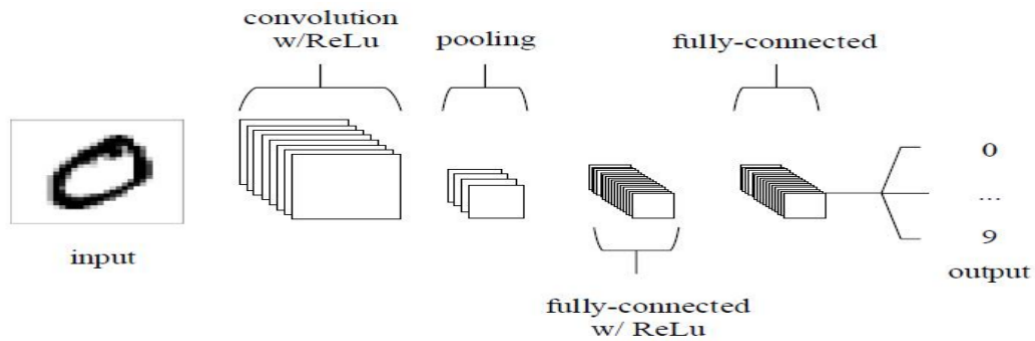


Figure 21: Exemple détaillé sur l'architecture du CNN.

4.1.3. Réseaux de neurones récurrents RNN

Un RNN, abréviation de Réseau de Neurones Récurrents, est un type de modèle de réseau de neurones utilisé pour traiter des données séquentielles, comme des séquences de texte, de musique ou des séquences temporelles.

Son fonctionnement repose sur une boucle de rétroaction qui permet aux informations de circuler entre les différentes étapes de la séquence. À chaque étape, le RNN prend en entrée un élément (par exemple, un mot dans une phrase) et génère une sortie et une représentation cachée, appelée état caché (Figure 22). L'état caché conserve en mémoire des informations contextuelles des étapes précédentes.

La représentation cachée est ensuite réinjectée à l'étape suivante, ce qui permet au modèle de prendre en compte l'historique de la séquence lors du traitement de l'élément actuel. Cette récurrence permet au RNN de capturer les dépendances à long terme présentes dans les données séquentielles. [32]

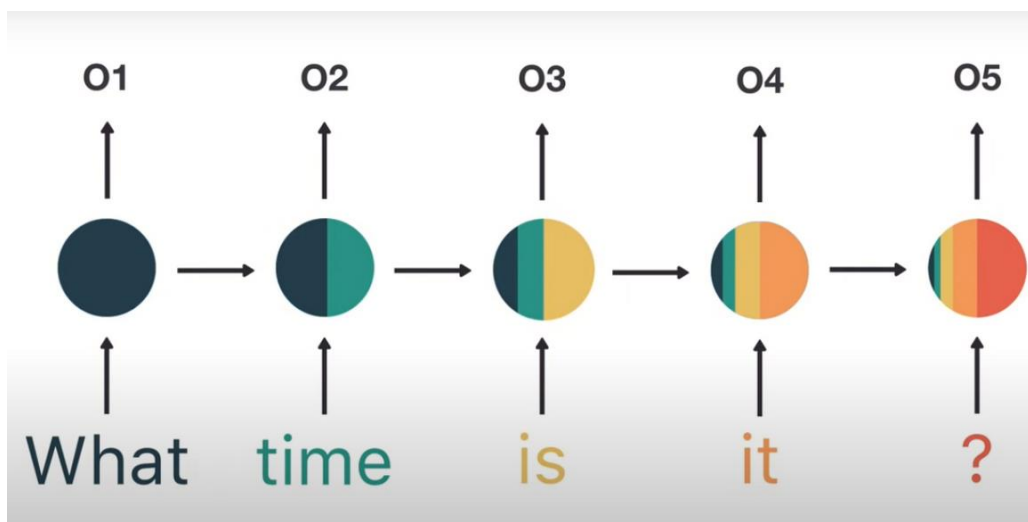


Figure 22: Exemple d'un réseau de neurones récurrents RNN.

4.1.4. Réseaux de neurones récurrents à mémoire court et long terme LSTM

Ils sont une variante RNN qui vise à résoudre le problème du vanishing gradient. Les LSTM sont particulièrement adaptés au traitement de séquences avec des dépendances à long terme en raison de leur capacité à mémoriser des informations à long terme. [33]

5. Vision par Ordinateur

La Vision par Ordinateur est un domaine de l'Intelligence Artificielle qui se concentre sur la compréhension et l'interprétation des informations visuelles provenant d'images numériques ou de vidéos. Son objectif est de permettre aux ordinateurs de percevoir et de donner un sens aux données visuelles en utilisant diverses techniques et algorithmes.

En permettant aux machines de comprendre le monde visuel qui les entoure, la Vision par Ordinateur vise à reproduire les capacités visuelles humaines. Elle comprend une variété de tâches, notamment la reconnaissance d'images et de vidéos, la détection et le suivi d'objets, la segmentation d'images, la compréhension de scènes, la reconstruction en 3D, et bien d'autres encore.

Les algorithmes de vision artificielle fonctionnent habituellement en examinant les valeurs et les motifs des pixels présents dans les images ou les vidéos. Leur capacité leur permet de détecter et de reconnaître des éléments tels que des objets, des personnes, du texte, des visages, des gestes, ainsi que d'autres caractéristiques visuelles. Ces données peuvent être traitées et utilisées dans une variété d'applications, notamment les soins de santé, les véhicules autonomes, la robotique, la surveillance, la réalité augmentée, et bien d'autres. Les réseaux neuronaux, l'Apprentissage Automatique, l'Apprentissage Profond et le traitement d'images sont quelques-unes des techniques et approches utilisées dans la Vision par Ordinateur. [34]

Ces dernières années, la Vision par Ordinateur a connu des progrès importants dans de nombreuses tâches de reconnaissance visuelle grâce aux progrès des capacités matérielles et à la disponibilité de vastes ensembles de données annotées.

5.1. Exemple d'application de Vision par Ordinateur

La Vision par Ordinateur offre un large éventail d'applications pratiques. Par exemple, elle peut être exploitée dans les véhicules autonomes afin de détecter et de suivre les objets sur la route. Dans le domaine de l'imagerie médicale, elle est utilisée pour le diagnostic et l'analyse. Les systèmes de surveillance tirent parti de la Vision par Ordinateur pour identifier les activités suspectes. La reconnaissance faciale fait également partie de ses applications, permettant notamment l'authentification. Nous allons donc nommer quelques applications de Vision par Ordinateur :

- Classification d'objets.
- Identification.
- Détection d'objets.
- Vérification d'objets.
- Segmentation d'objets.

6. Détection des objets

La détection d'objets consiste à trouver et à identifier des objets spécifiques dans une image ou une vidéo comme la Figure 23 le montre. Il est possible de détecter plusieurs éléments dans une scène grâce à cette méthode, tels que des véhicules, des individus, des structures et d'autres. Divers domaines tels que la surveillance, les véhicules autonomes, la réalité augmentée et bien d'autres encore utilisent fréquemment la détection d'objets.



Figure 23: Application d'une détection d'objets.

6.1. Algorithmes de détection d'objets

Il existe de nombreux algorithmes de détection d'objets, mais nous en aborderons trois principaux :

6.1.1. R-CNN

Girshick et al. (2014) ont proposé un système R-CNN qui permet de réaliser la détection d'objets sur des images en utilisant des propositions de régions. Un algorithme de recherche sélective est tout d'abord utilisé pour générer un ensemble de régions (environ 2000). Un CNN extrait ensuite les caractéristiques clés de chacune de ces régions. Enfin, la classe de chaque région est prédite par un SVM linéaire.

Ce système a été utilisé avec succès pour détecter des objets dans les images de scènes naturelles, mais il reste très lent car le traitement d'une image prend en moyenne 47 secondes. [35]

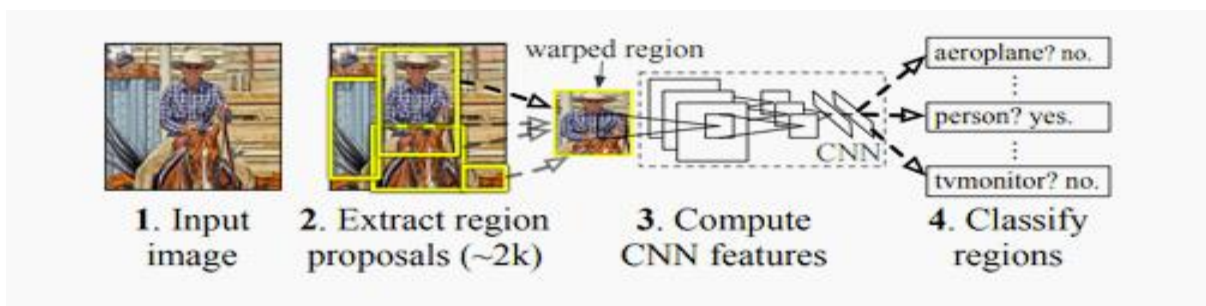


Figure 24: Schéma du système R-CNN, issu de Girshick et al. (2014).

6.1.2. SSD (Single Shot MultiBox Detector)

L'architecture SSD utilise un ensemble prédéfini de boîtes d'ancrage de différents rapports hauteur/largeur et une succession de couches de convolution de taille décroissante plutôt qu'une division de l'image en grille fixe pour détecter des objets de différentes échelles.

La carte des caractéristiques est divisée en cellules à chaque niveau d'échelle. Au centre de chacune d'elles, un ensemble de boîtes d'ancrage prédéfinies est appliqué. Finalement, le modèle prédit leurs boîtes englobantes, leur écart avec les valeurs cibles et leur indice de confiance. [36]

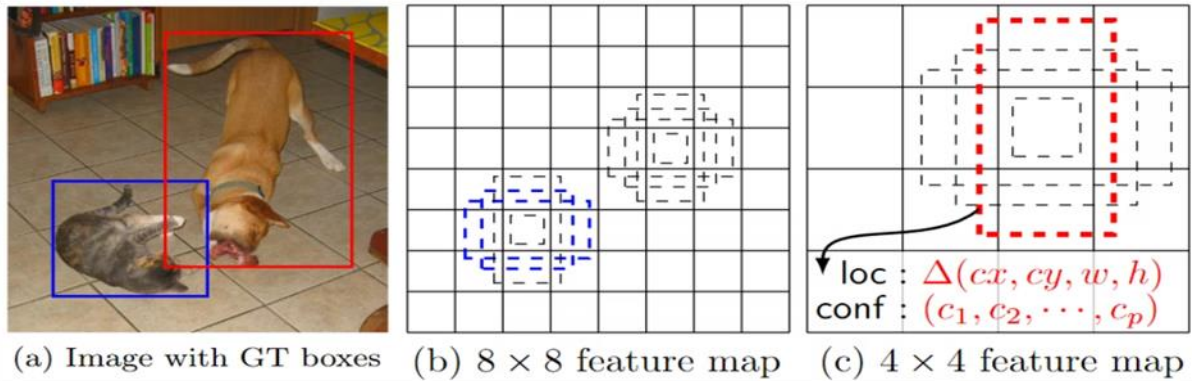


Figure 25: Schéma explicatif du mécanisme de détection de l'architecture SSD.

6.1.3. YOLO (You Only Look Once)

Redmon et al. (2016) ont proposé le système YOLO (You Only Look Once), qui permet de réaliser de la régression de boîtes englobantes d'objets sur des images. YOLO divise l'image d'entrée en grille. Un nombre prédéterminé de boîtes de délimitation et des scores de confiance sont prédits par chaque cellule de la grille. Enfin, les boîtes avec les scores de confiance et la probabilité de classe les plus élevés sont considérées comme les détections finales. [35]

YOLO avance plus rapidement que R-CNN. Cependant, il montre plus de difficultés à détecter les petits objets et les objets proches.

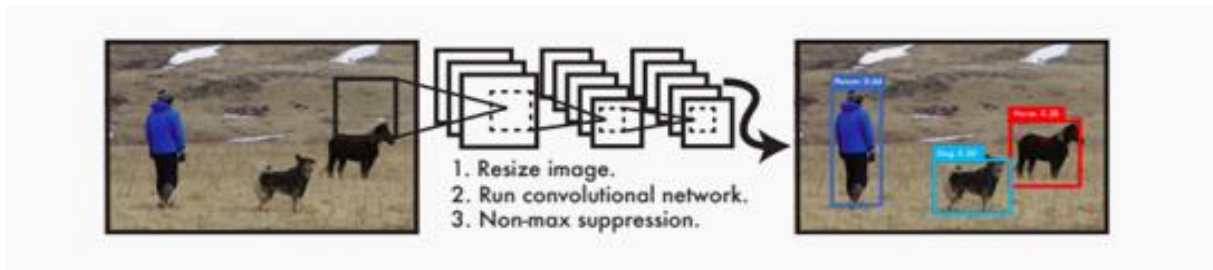


Figure 26: Mécanisme de détection de l'architecture YOLO.

7. Description automatique des images (Image Captioning)

Description automatique des images ou Image Captioning en anglais (et qui est le terme qu'on va utiliser dans la suite), est une tâche qui relève à la fois de la Vision par Ordinateur et du Traitement du Langage Naturel (NLP). Elle a pour but de générer une description textuelle ou une description décrivant le contenu d'une image. L'objectif est de permettre à une machine de comprendre visuellement le contenu d'une image, puis de le traduire en texte de manière interprétable. [37]



Figure 27: Exemple d'image captioning.

La compréhension de l'image et la création de la description sont généralement les deux étapes principales du processus de L'Image Captioning. Lors de la première étape un modèle d'Apprentissage Profond analyse et extrait les caractéristiques visuelles de l'image. Les tâches telles que la détection d'objets, la reconnaissance faciale, la segmentation sémantique et bien d'autres peuvent y être incluses. Dans la seconde étape, un modèle de langage génératif utilise les caractéristiques visuelles extraites pour créer une phrase ou un paragraphe décrivant l'image de manière cohérente et précise. L'objectif est de produire une description qui détaille les objets de l'image, leurs interactions, les scènes représentées et éventuellement d'autres informations pertinentes. L'objectif est d'obtenir une représentation textuelle précise qui reflète le contenu visuel de l'image. [38]

Cette technologie est utilisée dans différents domaines, notamment pour améliorer l'accessibilité pour les personnes malvoyantes, faciliter la recherche d'images basée sur le texte et permettre la génération automatique de descriptions pour les médias en ligne.

7.1. Traitement du Langage Naturel

Le Traitement du Langage Naturel (NLP) est une discipline de l'Intelligence Artificielle (IA) qui se focalise sur la compréhension et la manipulation du langage humain par les machines. Son objectif est de permettre aux ordinateurs de comprendre, d'interpréter, de générer et de répondre au langage naturel utilisé par les humains, que ce soit à travers des textes écrits ou des discours.

L'objectif principal du NLP est de permettre aux machines d'interagir de manière naturelle avec les humains en traitant et en analysant le langage humain de manière similaire à celle des individus. Cela ouvre la voie à une multitude d'applications, telles que les assistants vocaux, les

chatbots, la recherche d'informations, l'analyse des sentiments, la traduction automatique, l'analyse des données textuelles, la reconnaissance de la parole, et bien d'autres encore. [39]

7.1.1. Transformateur

Un transformateur est un modèle d'Apprentissage en Profondeur conçu pour le traitement automatique du langage (NLP). Les transformateurs sont conçus pour traiter des données séquentielles telles que le langage naturel et pour des tâches telles que la traduction et la synthèse de texte. Contrairement aux RNN, les transformateurs ne nécessitent pas de traitement séquentiel des données. Par exemple, si les données d'entrée sont une phrase en langage naturel, le transformateur ne doit pas traiter le début avant la fin. Le transformateur permet une parallélisations beaucoup plus importante que les RNN grâce à cette caractéristique, ce qui entraîne des temps d'apprentissage plus courts.

Les traducteurs modernes basés sur des transformateurs ont la capacité d'interconnecter les mots. Cela leur permet notamment d'obtenir des tournures de phrases bien plus proches du langage écrit ou parlé, et de donner le sens à un mot qui peut en avoir plusieurs. Dans d'autres domaines, comme le traitement d'images, les transformateurs peuvent également être utilisés.

8. Conclusion

Au cours de ce chapitre, nous avons exploré diverses composantes de l'Intelligence Artificielle. Nous avons commencé par une description générale de l'Intelligence Artificielle, puis nous avons abordé l'Apprentissage Automatique et enfin l'Apprentissage Profond, en mettant l'accent sur la détection d'objets et l'Image Captioning. Ces deux applications offrent de nombreuses solutions aux personnes handicapées ou ayant des problèmes visuels, facilitant ainsi leur réintégration dans la société. Ces avancées technologiques contribuent à améliorer leur qualité de vie et à favoriser leur inclusion sociale. Notre travail s'inscrit dans cette perspective de l'amélioration de la qualité de vie des personnes malvoyantes, comme on va voir dans les chapitres 3 et 4.

Chapitre 03

Méthodologie de Détection et de Description des Objets

1. Introduction :

La Vision par Ordinateur et les technologies de Traitement du Langage Naturel jouent un rôle essentiel pour aider les personnes malvoyantes. Pour notre problématique, à savoir la détection d'objets et la description de l'environnement, quelques applications ont été développées mais elles ne sont pas toutes efficaces. L'un des problèmes actuels dans le domaine de la description de l'environnement (Image Captioning) pour les malvoyants est le nombre limité d'applications disponibles. De plus, la majorité de ces applications ne fournit pas une description verbale des images, qui est un inconvénient majeur de ces applications, car les utilisateurs malvoyants dépendent principalement des solutions audibles.

Dans ce chapitre, nous présentons le détail de notre solution à travers une architecture. Nous expliquons chaque partie de notre solution par la suite tout en mettant l'accent sur le mode de fonctionnement qui se base à la fois sur la détection des objets et la description de l'environnement.

2. Architecture du système

L'objectif de notre travail est de proposer une solution basée sur deux modèles principaux basés sur l'Apprentissage Profond : un modèle de détection des objets en temps réel et un modèle d'Image Captioning.

L'objectif du modèle de détection d'objets en temps réel est de localiser et détecter l'emplacement des objets présents dans une image. Cela permet aux malvoyants d'obtenir des informations spécifiques sur les objets qui les entourent.

D'autre part, le modèle d'Image Captioning consiste à générer automatiquement des descriptions textuelles des images, qui permettent aux malvoyants de comprendre le contenu de l'image à travers une représentation verbale. Cela peut être utile pour décrire des scènes, des objets ou des personnes présentes dans l'image.

Dans ce travail, l'utilisateur a la possibilité de choisir une de ces deux approches proposées : la détection d'objets en temps réel ou l'Image Captioning.

Après avoir obtenu les résultats de la détection d'objets ou de l'Image Captioning, le système convertit ces résultats en format audio à l'aide d'un module de synthèse vocale (TextToSpeech). Cela permet aux utilisateurs malvoyants de restituer les informations aux de manière audible, et d'accéder facilement à ces informations de manière pratique et adaptée à leurs besoins.

La figure ci-dessous présente l'architecture globale de notre système, avec ses différents processus :

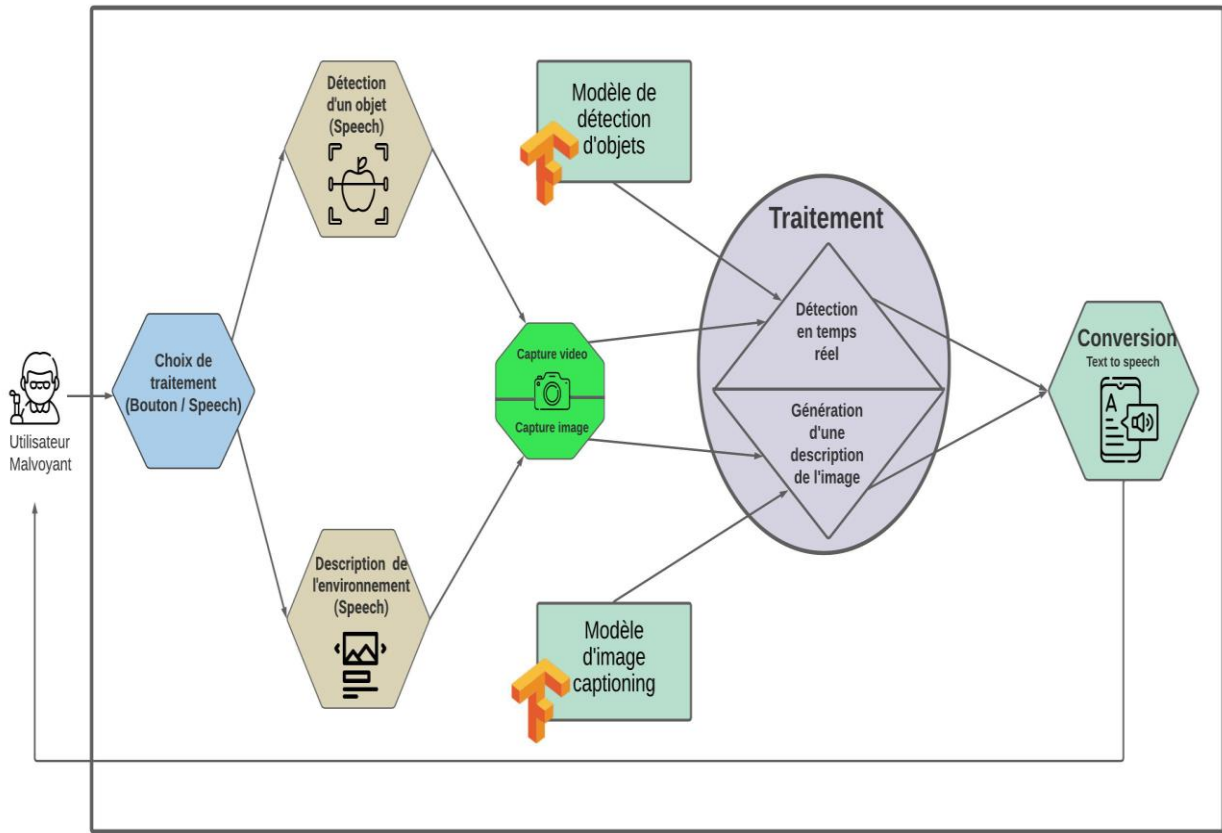


Figure 28: Architecture du système.

2.1. Choix du modèle :

Puisque notre travail s'adresse à la catégorie simple de personnes (les personnes malvoyantes), nous visons qu'il soit le plus simple et le plus facile à utiliser. Notre objectif principal est de s'assurer qu'ils n'ont aucune difficulté à comprendre les étapes d'utilisation de l'application.

Donc, pour atteindre cet objectif une méthode spécifique a été suivie dont la première étape consiste à permettre aux utilisateurs de choisir le type de modèle qui leur convient le mieux.

Dans ce but, nous avons conçu deux modèles pour eux, afin que chacun contienne une catégorie spécifique de données. Le premier modèle est un modèle de détection des objets, qui identifie et décrit les objets présents dans une image. Le deuxième modèle est un modèle d'Image Captioning, qui permet de générer une description textuelle d'une image donnée. Cette première méthode permet aux utilisateurs malvoyants de choisir le modèle qui répond le mieux à leurs besoins et à leurs préférences.

- Algorithme de choix de modèle :

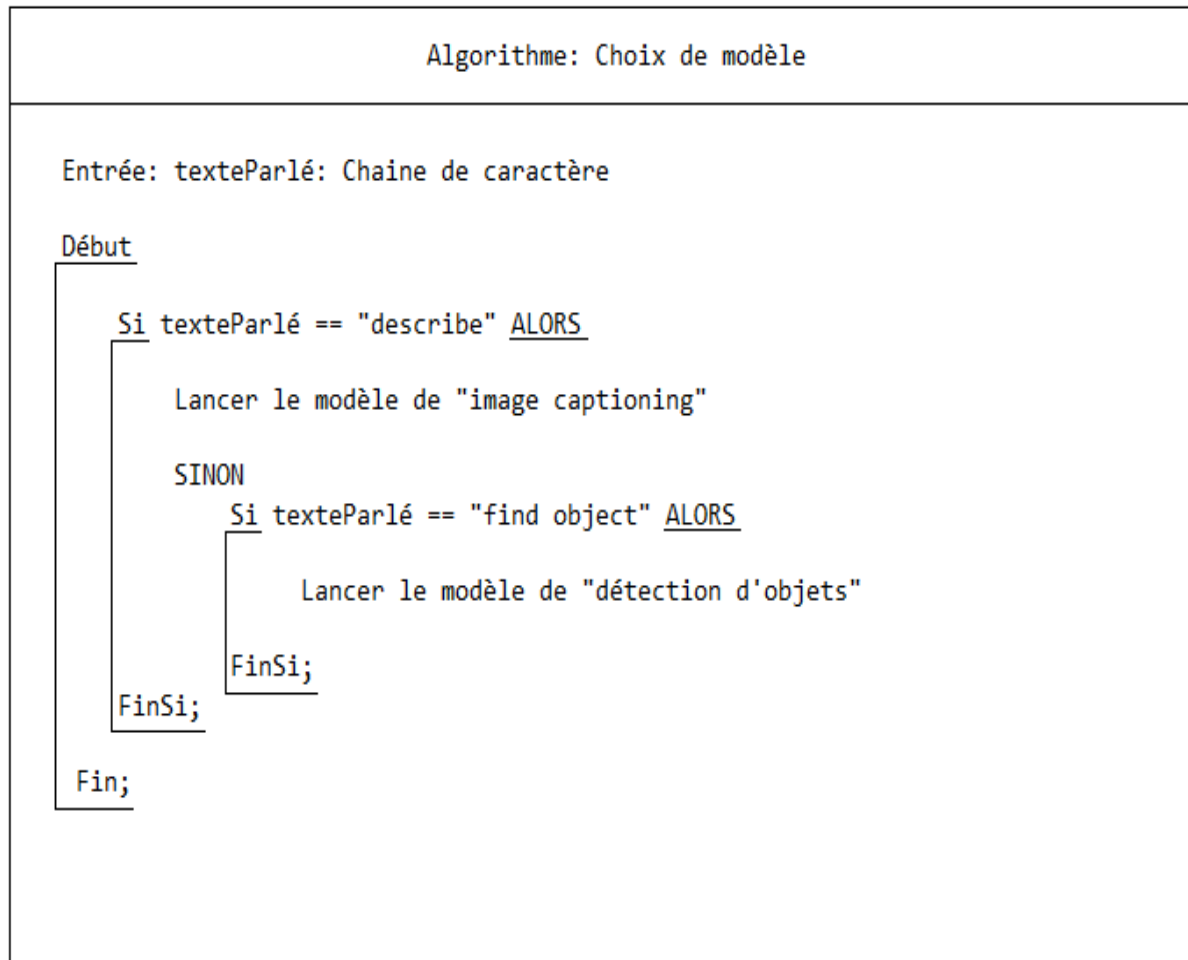


Figure 29: Algorithme de choix de modèle.

3. Méthodologie de détection d'objets

La première approche de notre système consiste à utiliser la détection d'objets. Dans cette partie, nous allons décrire l'architecture de modèle utilisé pour cette approche. Ensuite, nous décrirons le processus d'entraînement de modèle de détection. Cela implique généralement la collecte d'un grand ensemble d'images annotées et préparées, ainsi que les techniques de prétraitement utilisées pour améliorer la performance de modèle.

3.1. Modèle de détection d'objets

Les modèles de détection d'objets sont basés sur des réseaux de neurones, en particulier les réseaux neuronaux convolutifs (CNN). Ces CNN peuvent apprendre à extraire les caractéristiques visuelles des images à partir des données d'entrée en utilisant plusieurs couches de neurones convolutifs et de pooling. Ces caractéristiques permettent la détection et la localisation des objets dans les images.

Il existe plusieurs architectures populaires des modèles de détection d'objets, telles que SSD, R-CNN et YOLO, qui ont toutes été présentées dans le chapitre précédent. Dans notre système, nous avons utilisé l'architecture du modèle YOLO.

Pour entraîner ce modèle, nous avons adopté le mécanisme de l'apprentissage par transfert que nous avons présenté dans le chapitre 2.

L'apprentissage par transfert consiste à utiliser un modèle entraîné sur une base de données pour résoudre un problème donné, on enlève les dernières couches entièrement connectées ainsi que la couche Softmax, puis on insère une nouvelle couche entièrement connectée et une nouvelle couche Softmax, on gèle les poids des couches supérieures, ensuite on refait l'apprentissage en utilisant la nouvelle base de données pour résoudre le nouveau problème. [41]

Dans notre cas, nous avons utilisé un modèle YOLO pré-entraîné sur l'ensemble de données COCO qui détecte 80 objets.

Le principal avantage de l'apprentissage par transfert est dans l'utilisation d'architectures éprouvées, qui ont démontré leur efficacité, ainsi que dans la conservation des cartes de descripteurs complexes entraînées sur des millions d'images. Cela permet de les ajuster à notre problème spécifique, ce qui permet de réduire considérablement le temps d'apprentissage. [41]

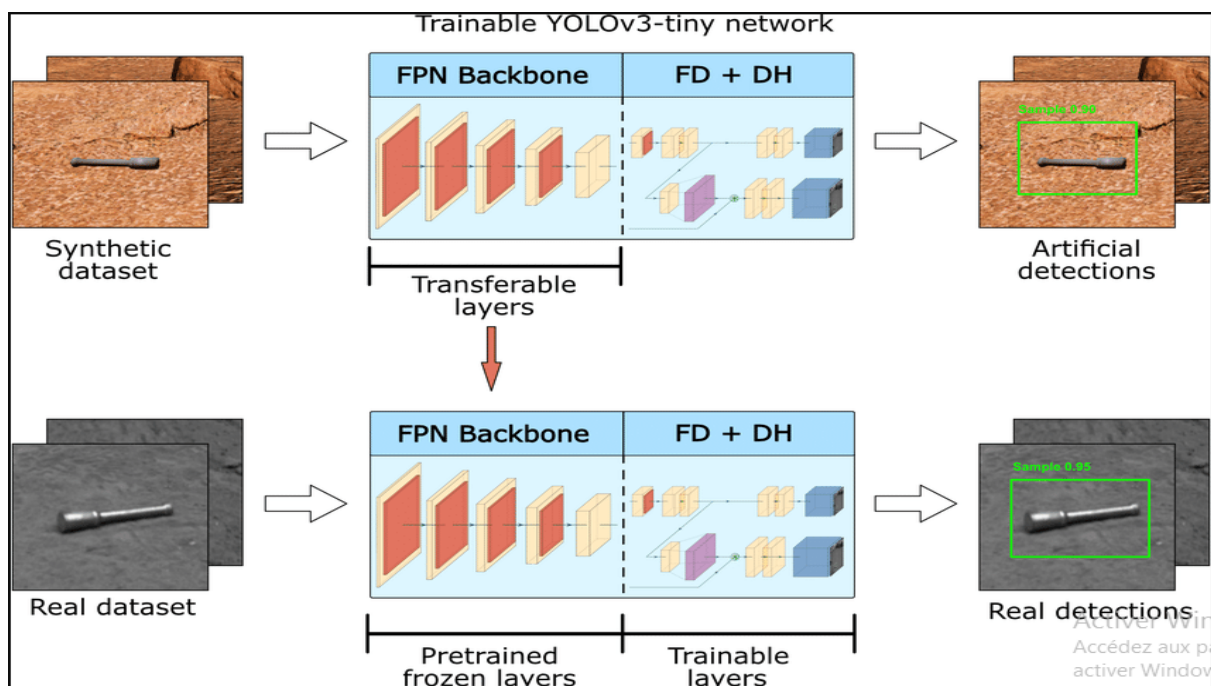


Figure 30: Technique d'apprentissage par transfert appliquée pour YOLO. [40]

Dans la section suivante, nous décrivons en détail le processus de préparation des données, y compris l'annotation des images, ainsi que les techniques de prétraitement utilisées pour améliorer la performance de notre modèle de détection d'objets.

3.2. Préparation de l'ensemble de données

La partie la plus longue de cette méthodologie de détection d'objets est la préparation de l'ensemble de donnée. C'est une étape fondamentale et complexe qui demande beaucoup du temps.

3.2.1. Collecte des images

Tout d'abord, nous devons collecter un ensemble de différentes images représentatives du domaine dans lequel le modèle sera utilisé. Cela peut être fait en utilisant des sources publiques, des bases de données spécialisées ou en collectant des données personnalisées. Pour assurer la généralisation du modèle Il est important de s'assurer que l'ensemble de donnée couvre beaucoup de scénarios et d'angles de vue.

Les conditions d'éclairage et les différents angles d'un objet sont deux facteurs qui affectent directement sa coloration et éventuellement sa forme, il faut chercher à maintenir une variété dans les images. Cette stratégie garantit une probabilité plus élevée d'une prédiction correcte et précise même dans des conditions différentes.

Pour les besoins de ce projet, nous avons rassemblé des images des objets les plus nécessaires pour les malvoyants, à savoir : **(Person, Remote, Stairs, Door, Bed, Table, Chair, Cup, Laptop, Phone)**.

3.2.2. Prétraitement des images

Le prétraitement est une étape majeure et importante dans la détection d'objets, car il permet d'uniformiser les images à traiter afin de ne pas faire des traitements spécifiques pour certaines images. En effet, les images étaient sur différentes formes, différentes résolutions, des images en RGB et des images en niveau de gris. Les images sont toutes redimensionnées en une taille fixe pour garantir que la collecte de données maintienne une qualité standard. [42]

3.2.3. Annotation des images

L'annotation des données est une tâche essentielle dans la préparation des ensembles de données d'entraînement en apprentissage supervisé. Chaque exemple d'un ensemble de données destiné à l'apprentissage supervisé doit contenir au moins une étiquette nommée étiquette cible. Il s'agit de la réponse ou du résultat prédit par d'autres attributs. Par exemple, l'annotation en vision artificielle est le processus manuel d'identification d'objets dans une image avec des étiquettes ou à encadrer des objets avec des rectangles ou des polygones irréguliers, et parfois de description de zones spécifiques d'une image avec du texte. [43]

Dans notre contexte problématique : la détection d'objet, les annotations d'images jouent un rôle important. Cette tâche implique l'étiquetage et l'annotation d'objets spécifiques dans une image pour créer un ensemble de donnée annotés.

Avec des annotations d'image précises, les algorithmes de détection d'objets peuvent être entraînés à partir de ces informations pour identifier et trouver automatiquement différents objets dans de nouvelles images.

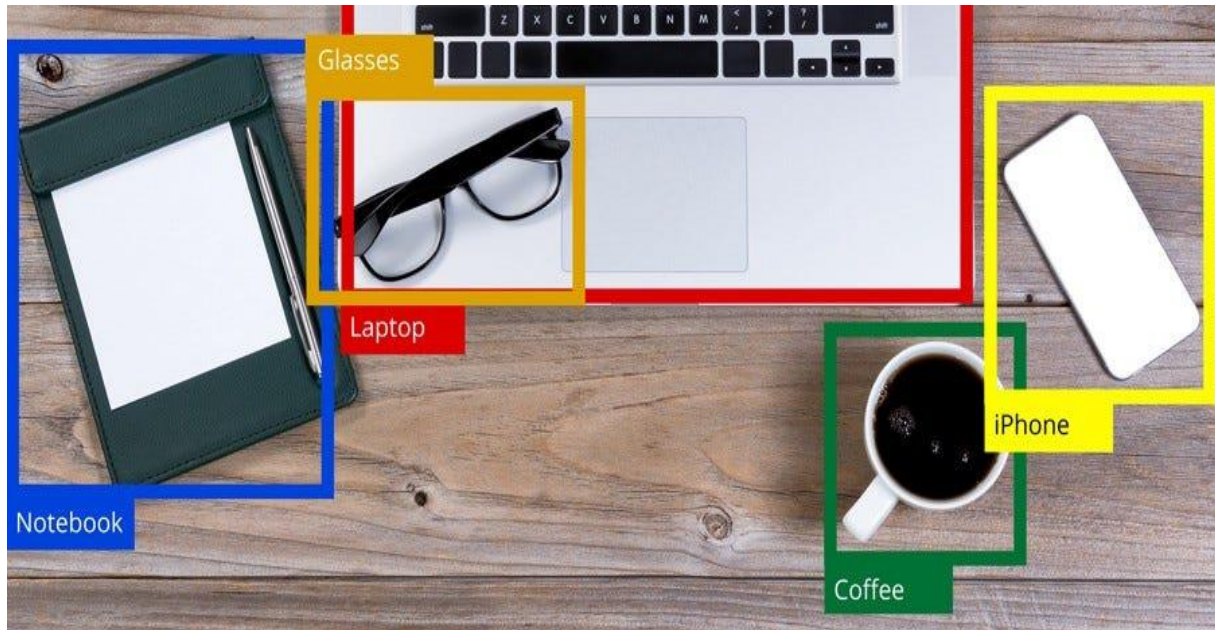


Figure 31: Annotation d'une image pour la détection d'objets.

3.2.4. Division de l'ensemble de données

Enfin, une fois que l'ensemble de donnée est bien préparé et que les images sont annotées, on peut le diviser en ensembles d'entraînement, de validation et de test. L'ensemble d'entraînement est utilisé pour entraîner le modèle de détection, l'ensemble de validation est utilisé pour l'optimisation des hyperparamètres et l'évaluation des performances pendant l'entraînement. L'ensemble de test est utilisé pour tester les performances du modèle avec de nouvelles données non vues auparavant.

3.3. Entraînement du modèle

L'entraînement des modèles de détection d'objets est un processus critique. Le but est d'apprendre au modèle à reconnaître et à localiser divers objets dans des images. Ce processus peut être décomposé en plusieurs étapes.

Pendant l'entraînement, le modèle ajuste les poids et les paramètres internes pour minimiser la fonction de perte, qui calcule la différence entre les prédictions du modèle et les annotations fournies.

Le processus peut être intensif en termes de calcul et nécessite souvent des ressources informatiques puissantes telles que des unités de traitement graphique (GPU) et des systèmes distribués. L'entraînement implique le réglage et l'évaluation itératifs du modèle pour améliorer les performances, et peut nécessiter plusieurs itérations.

Une fois que le modèle est entraîné avec succès, il peut être utilisé pour identifier des objets dans de nouvelles images.

4. Méthodologie d'Image Captioning

La deuxième approche pour notre système est l'Image Captioning. Cette approche utilise des modèles de Deep Learning pour générer des descriptions textuelles pour les images. Nous allons explorer les différentes architectures de modèles utilisées dans cette approche. Nous allons aussi décrire le processus d'entraînement des modèles, en présentant les ensembles de données les plus utilisés pour former ces modèles. Enfin, nous allons discuter les techniques de prétraitement des données (prétraitement d'images et des textes) utilisées pour obtenir des descriptions précises et cohérentes.

4.1. Modèles d'Image Captioning

Il existe plusieurs modèles et architectures qui ont été proposés pour résoudre le problème d'Image Captioning, allant de l'utilisation de réseaux de neurones convolutifs (CNN) pour extraire les caractéristiques des images à des architectures plus avancées telles que des réseaux de neurones récurrents (RNN), des LSTM et des modèles de transformateur.

4.1.1. Extraction des caractéristiques d'images avec CNN

Le processus d'Image Captioning commence généralement par l'utilisation d'un CNN pré-entraîné sur un grand ensemble de données, tel que VGGNet, ResNet ou InceptionV3.

Dans notre système, nous avons utilisé le modèle InceptionV3 comme un CNN pré-entraîné pour extraire les caractéristiques visuelles de l'image. Ce CNN peut capturer les informations sémantiques et structurelles d'une image à travers plusieurs couches de convolutives et de pooling. Les caractéristiques extraites sont ensuite fournies au modèle de transformateur pour créer une séquence de mots qui décrit l'image.

4.1.2. Transformateur

Dans notre système, nous avons adapté le modèle de transformateur pour créer les descriptions d'images, nous avons déjà introduit ce modèle dans le chapitre précédent. Il s'agit d'un modèle largement utilisé dans les domaines du Traitement du Langage Naturel (NLP) et de la Vision par Ordinateur qui est bien adapté au traitement des tâches d'Image Captioning.

Les transformateurs permettent de capturer les liens et les relations entre les caractéristiques visuelles extraites des images et les mots ou phrases correspondants dans les descriptions. Contrairement aux architectures traditionnelles basées sur CNN-RNN, où un réseau convolutif (CNN) extrait des caractéristiques visuelles et un réseau récurrent (RNN) génère les descriptions, les transformateurs utilisent une structure homogène pour effectuer un apprentissage de bout en bout.

L'architecture de transformateur se compose de plusieurs couches d'encodeurs et de décodeurs qui fonctionnent en parallèle. Un encodeur capture des informations visuelles à partir

d'une image, un décodeur génère des descriptions en se basant sur ces informations et utilise un mécanisme d'attention pour se concentrer sur les parties pertinentes de l'image.

Les transformateurs dans l'Image Captioning ont été utilisés avec succès pour modéliser le contexte global de l'image à chaque couche d'encodeur, sans architecture CNN-RNN combinée. Ils ont également démontré leur capacité à capturer les relations spatiales entre les régions de l'image créée des descriptions de haute qualité. [44]

4.2. Ensembles de données d'Image Captioning

Pour former et évaluer des modèles d'Image Captioning, il est essentiel d'avoir accès à des ensembles de données appropriés. Il existe plusieurs ensembles qui ont été utilisés pour cette tâche. Ils sont tous sous la forme [image \rightarrow caption] et se composent d'images d'entrée et de leurs descriptions correspondantes, permettant aux modèles d'apprendre à associer correctement les informations visuelles avec des descriptions textuelles.

Parmi les ensembles de données disponibles, nous pouvons citer et explorer quelques-uns des ensembles populaires utilisés dans ce domaine :

Flickr 8k : Cet ensemble de donnée contient un total de 8000 images chacune avec 5 descriptions différentes.

Ces images sont distribuées comme suit :

- Ensemble d'entraînement : 6000 images
- Ensemble de validation : 1000 images
- Ensemble de test : 1000 images

Flickr 30k : Cet ensemble de donnée est composé de 30 000 images chacune avec 5 descriptions différentes.

Ces images sont distribuées comme suit :

- Ensemble d'entraînement : 28000 images
- Ensemble de validation : 1000 images
- Ensemble de test : 1000 images

MS-COCO (Microsoft Common Objects in Context) : L'un des ensembles de données les plus largement utilisés pour l'Image Captioning, l'ensemble de données contient plus de 82000 images, chacune ayant au moins 5 descriptions différentes. Il couvre une grande variété de scènes et d'objets.

1	101654506_8eb26cfb60.jpg#0	A brown and white dog is running through the snow .
2	101654506_8eb26cfb60.jpg#1	A dog is running in the snow
3	101654506_8eb26cfb60.jpg#2	A dog running through snow .
4	101654506_8eb26cfb60.jpg#3	a white and brown dog is running through a snow covered field
5	101654506_8eb26cfb60.jpg#4	The white and brown dog is running over the surface of the snow
6		
7	1000268201_693b08cb0e.jpg#0	A child in a pink dress is climbing up a set of stairs in an enclosure
8	1000268201_693b08cb0e.jpg#1	A girl going into a wooden building .
9	1000268201_693b08cb0e.jpg#2	A little girl climbing into a wooden playhouse .
10	1000268201_693b08cb0e.jpg#3	A little girl climbing the stairs to her playhouse .
11	1000268201_693b08cb0e.jpg#4	A little girl in a pink dress going into a wooden cabin .

Figure 32: Exemple de l'ensemble de données Flickr8k.

4.3. Préparation de l'ensemble de données

Pour entraîner un bon modèle d'Image Captioning, il est important d'avoir un ensemble de données bien préparé qui comprend plusieurs aspects clés, tels que le prétraitement des images et du texte des descriptions (les captions) et la division de l'ensemble de données en un ensemble d'apprentissage et de test.

4.3.1. Prétraitement des images

Le prétraitement des images est une étape importante dans le domaine d'Image Captioning pour garantir de bons résultats et de meilleures performances du modèle. Le prétraitement des images utilise l'application des différentes techniques pour normaliser, améliorer et préparer les images de les utiliser pour créer des descriptions.

L'une des étapes du prétraitement d'images est le redimensionnement des images. Les images peuvent être de tailles différentes et il est important de les mettre toutes à la même taille avant de les introduire au modèle.

Il y a aussi la normalisation des images cela implique souvent de convertir les valeurs des pixels en une plage standardisée, comme $[0, 1]$ ou $[-1, 1]$. La normalisation permet de stabiliser les caractéristiques visuelles des images et de faciliter la convergence des modèles lors de l'entraînement.

- Méthode de Prétraitement d'images :

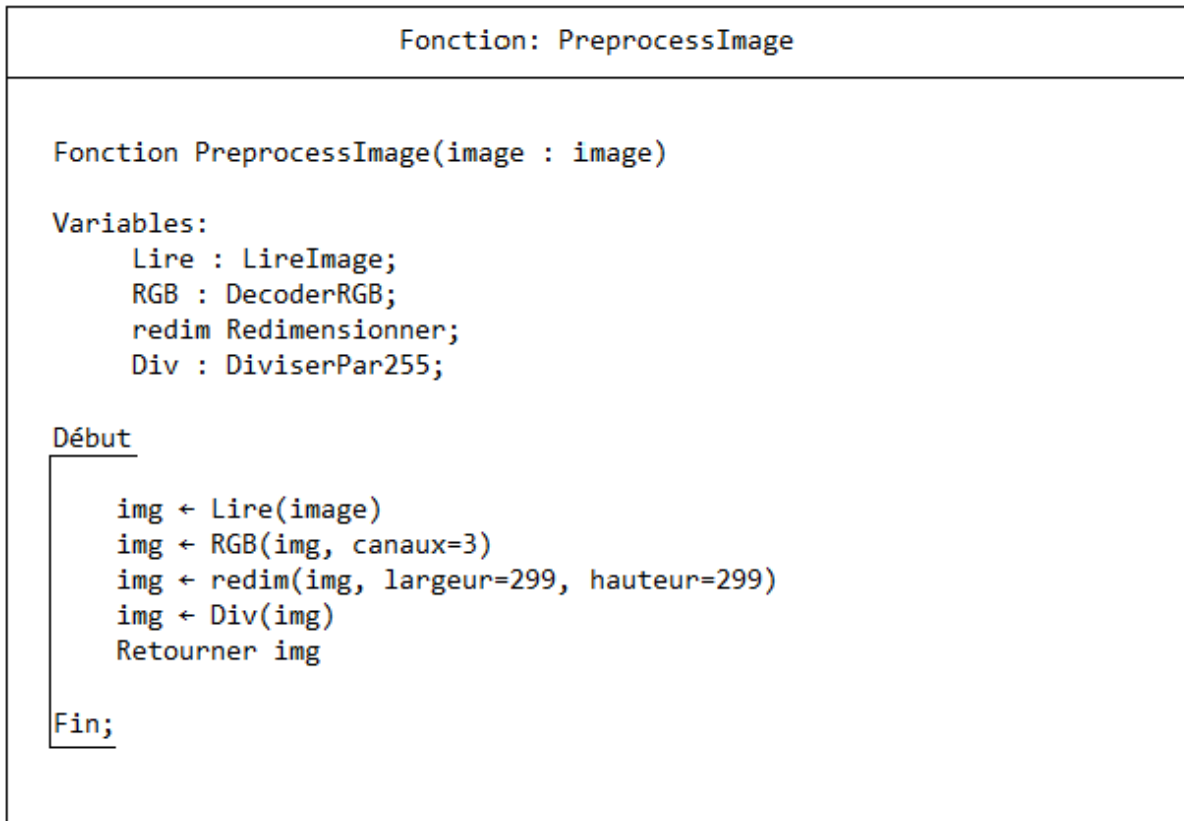


Figure 33: Méthode de Prétraitement d'images.

4.3.2. Prétraitement du texte (les descriptions)

Le prétraitement du texte joue aussi un rôle fondamental dans l'entraînement d'un bon modèle d'Image Captioning, qui vise à produire des descriptions sémantiquement des images. Le traitement des données textuelles dans ce contexte est important pour comprendre, analyser et extraire les informations pertinentes contenues dans les descriptions associées aux images. Le prétraitement du texte implique une série d'étapes visant à nettoyer, normaliser et structurer les données textuelles avant de les utiliser dans le modèle.

- Nettoyage :

Le prétraitement du texte des descriptions d'un ensemble de donnée d'Image Captioning commence généralement par l'étape de nettoyage, qui consiste à supprimer les caractères indésirables tels que les chiffres, les caractères spéciaux et non alphanumériques. Le but de cette étape est de simplifier les données textuelles pour un traitement plus facile. De plus, les mots en majuscules peuvent être convertis en minuscules pour éviter la redondance et normaliser le texte.

- Tokenisation :

Une autre étape importante du traitement de texte est la tokenisation, La tokenisation divise le texte en unités plus petites appelées tokens. Les tokens peuvent être des mots, des phrases ou même des caractères individuels, selon la granularité souhaitée. Cette étape permet une représentation structurée du texte, ce qui facilite son traitement et son analyse ultérieurs.

- Construction d'un vocabulaire :

Après le nettoyage et la tokenisation, une autre étape importante du prétraitement est la construction d'un vocabulaire, c'est à dire créer une liste de tous les mots présents dans les descriptions des images de l'ensemble de données. La construction du vocabulaire est essentielle pour la modélisation ultérieure, car les mots peuvent être représentés comme des vecteurs numériques compréhensibles par les modèles d'Image Captioning. Des techniques telles que l'indexation de chaque mot et la création de vecteurs de mots peuvent être utilisées pour représenter efficacement le vocabulaire.

En conclusion, le prétraitement du texte dans le contexte d'ensemble de donnée d'Image Captioning est une étape importante pour la préparation des descriptions des images pour les utiliser dans les modèles d'Image Captioning.

- Méthode de Prétraitement du texte :

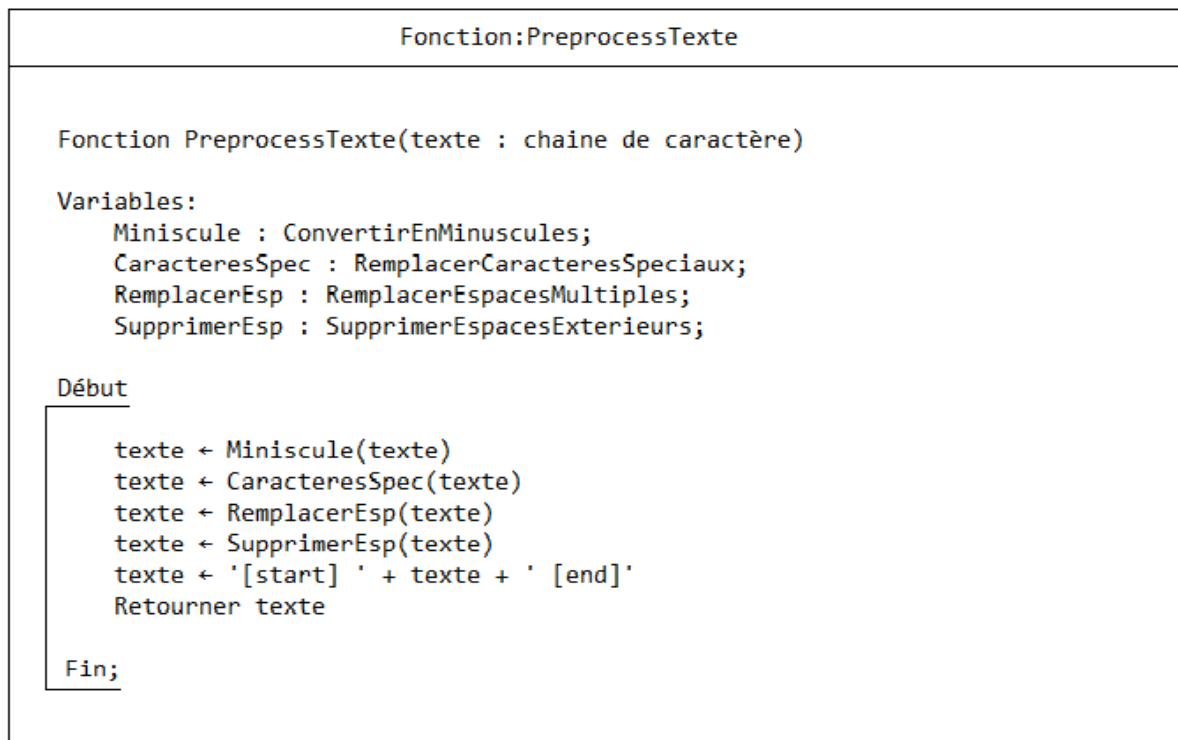


Figure 34: Méthode de Prétraitement du texte.

4.3.3. Division de l'ensemble de données

Enfin, il est important de diviser l'ensemble de données en ensemble d'apprentissage et un ensemble de test. De cette manière, les performances des modèles d'Image Captioning peuvent être évaluées et comparées à d'autres approches. La taille de chaque ensemble peut varier en fonction des besoins spécifiques du projet, mais une distribution équilibrée est généralement recommandée.

4.4. Entraînement du modèle

Les images sont introduites dans un CNN pré-entraîné, (InceptionV3 dans notre cas), pour extraire les caractéristiques visuelles. Le CNN transforme chaque image en un vecteur de caractéristiques de grande dimension qui représente les informations visuelles contenues dans l'image.

Le transformateur est ensuite entraîné pour apprendre à utiliser les caractéristiques visuelles extraites pour générer les descriptions. Le but est de maximiser la probabilité de générer la bonne description pour une image donnée.

Le processus d'entraînement est itératif, et le modèle est ajusté à l'aide de techniques d'optimisation telles que l'algorithme Adam. En règle générale, les modèles sont entraînés à l'aide de GPU ou de TPU pour un calcul de gradient plus rapide et de meilleures performances.

Une fois l'entraînement est terminé, le modèle peut être utilisé pour générer automatiquement de nouvelles descriptions d'images. Il est important de noter que les performances du modèle dépendent fortement de la qualité et de la diversité des données d'apprentissage, et des optimisations et architectures utilisées.

5. Conversion des résultats en vocale

Après que les résultats de traitement seront prêts, nous ajoutons une méthode pour les convertir en discours pour décrire l'image générée ou l'objet détecté en temps réel. Cette méthode facilite la compréhension des personnes malvoyantes, car elles ne peuvent pas lire ce qui est écrit.

- La méthode de conversion :

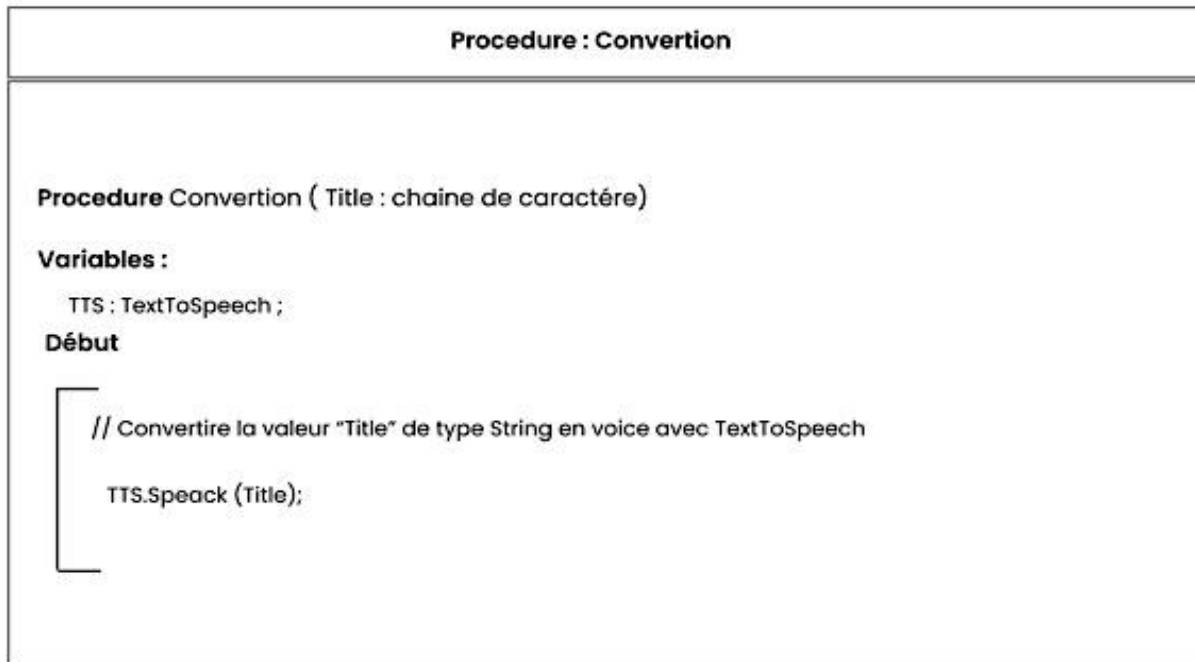


Figure 35: La méthode de conversion en vocale (TextToSpeech).

6. Conclusion

Dans ce chapitre, nous avons présenté notre méthode pour aider les personnes malvoyantes en combinant la détection d'objets et l'Image Captioning. Nous avons présenté l'architecture de notre système, qui inclut deux modèles principaux : un modèle de détection d'objets en temps réel et un modèle d'Image Captioning.

Nous avons détaillé le modèle de détection d'objets, en présentant l'architecture YOLO et le mécanisme d'apprentissage par transfert pour l'entraînement du modèle. Nous avons aussi expliqué le processus de préparation des données, incluant la collecte d'images, le prétraitement et l'annotation des images. Ce qui concerne le modèle d'Image Captioning, nous avons discuté l'utilisation du modèle transformateur pour générer automatiquement des descriptions textuelles des images. De plus, nous avons décrit le processus de préparation des données, le prétraitement des images et des descriptions.

Grâce à notre approche, nous espérons aider les personnes malvoyantes de mieux comprendre leur environnement et à interagir de manière autonome avec le monde extérieur. Dans le prochain chapitre, nous allons détailler l'implémentation de notre système et présenter les résultats obtenus.

Chapitre 04

Implémentation

1. Introduction

Dans ce chapitre, nous abordons en détail le système que nous avons introduit dans le chapitre précédent, en examinant tous les aspects de son implémentation et de sa réalisation. Ce système va utiliser le modèle d'Image Captioning et de détection d'objets comme présenté et discuté au niveau du chapitre 3. En effet, notre système se présente sous forme d'une application mobile dédiée aux personnes malvoyantes qui utilise ces deux modèles principaux.

Le chapitre commence par illustrer les aspects d'implémentation de notre système, qui implique la réalisation de cette application mobile. Nous commençons par présenter l'environnement de travail utilisé lors du processus de développement, soit matériels ou logiciels. Puis, nous expliquons en détail les différentes étapes inclus dans l'implémentation des modèles de détection d'objets et d'Image Captioning.

2. Environnement de travail

Dans cette section, nous allons explorer l'environnement de travail dans laquelle cette application a été développée et déployée. Plus précisément, l'environnement matériel et l'environnement logiciel.

2.1. Environnement matériel

Les différents composants matériels qui sont utilisés lors du développement et de l'exécution de notre application sont :

Tableau 2: Environnement matériel.

Matériel	Caractéristique
PC Dell Inspiron3542	Système d'exploitation : Windows 8.1 Mémoire (RAM) : 8Go Processeur : Intel(R) Core(TM) i5-4210 CPU @ 1.70GHz Carte Graphique : Intel(R) HD GraphicsFamily.
Medion erazer P6661 (MD99976)	Système d'exploitation : Windows 10 Mémoire (RAM) : 12Go Processeur : Intel(R) Core(TM) i5-6200 CPU @ 2.30GHz Carte Graphique : Nvidia GTX 950M

2.2. Environnement logiciel

Cette partie présente les outils logiciels, les bibliothèques de Deep Learning, et les éditeurs de code utilisés lors du développement de notre application.

2.2.1. Tensorflow

TensorFlow est une bibliothèque d'Apprentissage Automatique open source développée par Google basée sur Python. Elle est destinée à accélérer le développement et le déploiement de modèles d'Apprentissage Automatique, notamment dans le domaine des réseaux de neurones. [48]



2.2.2. Android Studio

Android Studio est l'environnement de développement intégré (IDE) officiel des applications Android. Basé sur le puissant outil de développement et d'édition de code d'IntelliJ IDEA, Android Studio offre encore plus de fonctionnalités qui améliorent votre productivité lorsque vous créez des applications Android. Voici une liste non exhaustive de ces fonctionnalités :



- Un système de compilation flexible basé sur Gradle
- Un émulateur rapide et riche en fonctionnalités
- Un environnement unifié pour un développement sur tous les appareils Android
- La modification en temps réel pour mettre à jour les composables dans les émulateurs et les appareils physiques en temps réel
- Des modèles de code et l'intégration GitHub pour vous aider à compiler des fonctionnalités d'application courantes et à importer des exemples de code
- Des outils et frameworks de test complet
- Des outils Lint permettant de détecter les problèmes de performances, d'ergonomie, de compatibilité des versions, etc.
- Compatibilité C++ et NDK
- Compatibilité intégrée avec Google Cloud Platform, qui facilite l'intégration de Google Cloud Messaging et App Engine. [45]

2.2.3. Google Colab

Colaboratory, souvent raccourci en "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté à la Machine Learning, à l'analyse de données et à l'éducation. En termes plus techniques, Colab est un service hébergé de notebooks Jupyter qui ne nécessite aucune configuration et permet d'accéder sans frais à des ressources informatiques, dont des GPU. [46]



2.2.4. Google Drive

Google Drive est un service de stockage en ligne fourni par Google. Il permet aux utilisateurs de sauvegarder, de partager et d'accéder à des fichiers et des dossiers à partir de n'importe quel appareil disposant d'une connexion Internet. [49]



2.2.5. LabelImg

LabelImg est un outil open source écrit en Python qui est utilisé pour annoter des images et créer des ensembles de données pour l'Apprentissage Automatique, en particulier pour les tâches de détection d'objets. Il fournit une interface utilisateur graphique facile à utiliser qui permet aux utilisateurs de dessiner des rectangles autour des objets dans une image, et d'attribuer des annotations à ces objets.



Les annotations sont enregistrées sous forme de fichiers XML au format PASCAL VOC, le format utilisé par ImageNet. De plus, il prend également en charge les formats YOLO et CreateML. [50]

3. Implémentation de modèle de détection d'objets

Dans la première étape de développement de cette application, nous présentons une méthodologie complète pour l'implémentation d'un modèle de détection d'objets. Nous détaillons tous les processus que nous avons suivis pour réaliser ce modèle en suivant plusieurs étapes.

3.1. Collecte des images

Tout d'abord, nous devons collecter et rassembler un ensemble d'images représentatives du domaine dans lequel le modèle sera utilisé. On doit assurer que les images couvrent différents scénarios, angles de vue et conditions d'éclairage.

Pour les besoins de ce projet, nous avons rassemblé des images des 10 objets les plus nécessaires pour les malvoyants, à savoir :

1.Person; 2.Remote; 3.Stairs; 4.Door; 5.Bed; 6.Table; 7.Chair; 8.Cup;9.Laptop; 10.Phone.

Notre application est principalement utilisée dans des environnements fermés, comme la maison par exemple. C'est pourquoi toutes les images que nous avons collectées représentent des objets situés à l'intérieur de ces environnements domestiques, et non à l'extérieur.

La collecte des données a été effectuée en téléchargeant manuellement des images à partir du Web, en combinant celles-ci avec nos propres images personnelles.

Initialement, nous avons commencé par collecter 150 images par objet. Cependant, après l'entraînement initial, nous avons constaté que la précision de certains objets était un peu faible. Donc, nous avons augmenté le nombre d'images pour ces objets à 180 images chacun. Finalement, nous avons obtenu cette distribution des images par objet, comme illustré dans la figure suivante.

L'objet	image	Nbr d'images	L'objet	image	Nbr d'images
	Table	180		Person	180
	Chair	180		Remote	150
	Cup	150		Stairs	180
	Laptop	150		Door	180
	Phone	150		Bed	180

Figure 36: Nombre d'images collectées par objet.

3.2. Annotation des images

Après la collecte des images, l'étape suivante consiste à annoter l'ensemble des données, c'est une tâche fatigante et qui prend beaucoup de temps mais elle est très importante pour le prétraitement des images.

Pour annoter les images, nous avons utilisé l'outil LabelImg qui facilite le dessin des boîtes englobantes (les rectangles) souhaitées autour des objets, comme illustré dans la figure suivante.

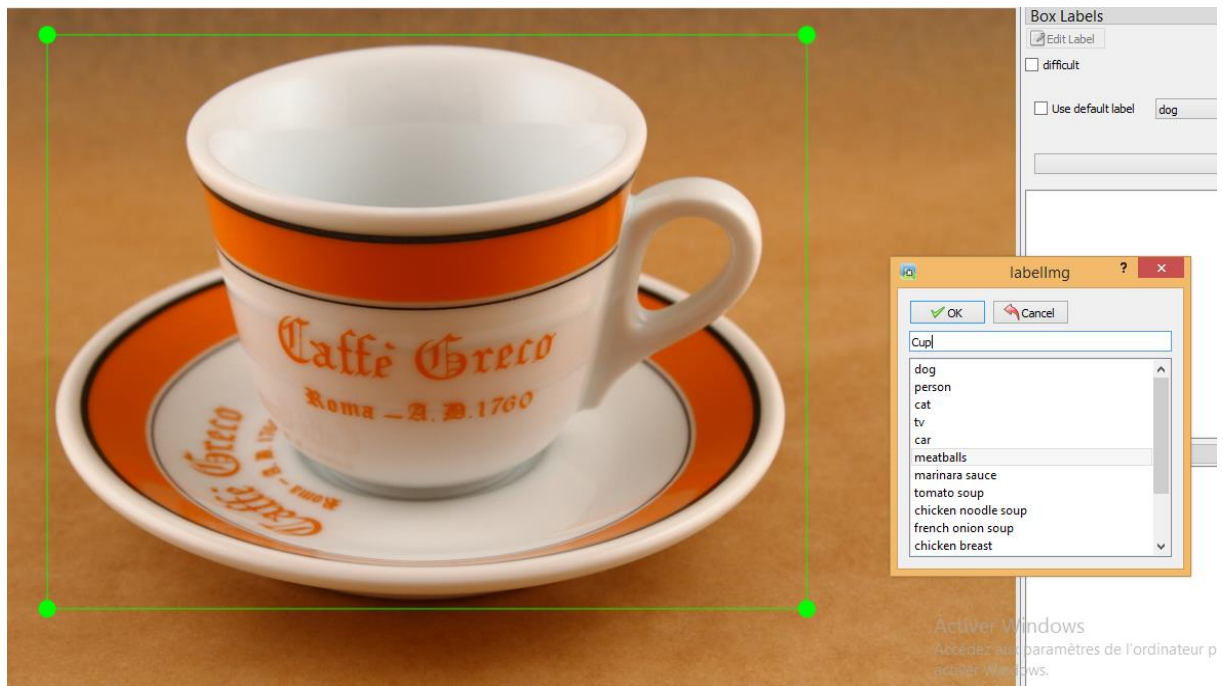


Figure 37: Annotation d'une image avec LabelImg.

LabelImg nous offre la possibilité d'enregistrer les annotations dans différents formats tels que PASCAL VOC, YOLO et CreateML. Chaque format utilise sa représentation spécifique des coordonnées de la boîte englobante. Puisque nous avons utilisé le modèle YOLO dans notre système, nous utilisons son format. Après avoir dessiné la boîte englobante et ajouté l'annotation, le programme crée automatiquement un fichier (TXT) portant le même nom que l'image où les informations sur les annotations dans l'image sont stockées. Le nombre de lignes de ce fichier texte indique le nombre d'objets présents dans une image. Chaque ligne a cinq paramètres.

Par exemple, dans la figure suivante qui représente le fichier d'annotation de l'image de la figure 37, il y a une seule ligne, donc un seul objet dans l'image. Chaque ligne est représentée dans le format suivant :

```
<object-class-ID> <coordonnéeX_min> <coordonnéeY_min> <coordonnéeX_max>
<coordonnéeY_max>
```

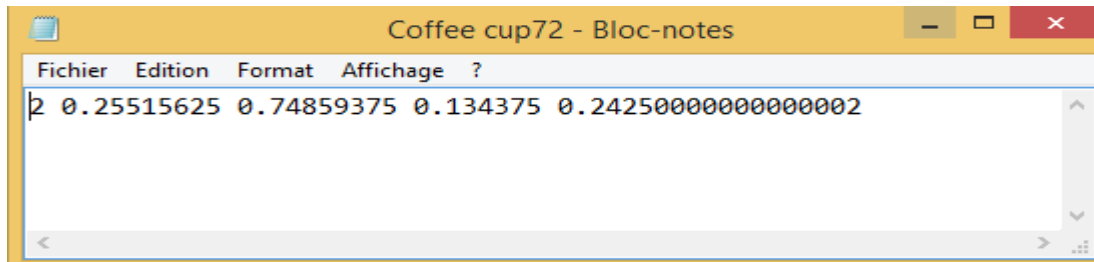


Figure 38: Exemple d'un fichier d'annotation au format YOLO.

Les coordonnées et les dimensions de la boîte englobante sont normalisées entre zéro et un en pourcentage des dimensions de l'image.

3.3. Division de l'ensemble des images

Une fois que l'annotation est terminée, on va diviser notre ensemble d'images en ensembles d'entraînement et de test. L'ensemble d'entraînement est utilisé pour entraîner le modèle de détection, L'ensemble de test est utilisé pour tester les performances du modèle avec de nouvelles données non vues auparavant. Pour cela, nous sélectionnons pour chaque objet 30 images pour le test, et le reste est utilisé pour l'entraînement. Ensuite, nous créons un répertoire appelé "dataset" où nous ajoutons toutes les images d'entraînement et de test, ainsi que leurs fichiers d'annotation correspondants, respectivement dans les répertoires "train" et "test".

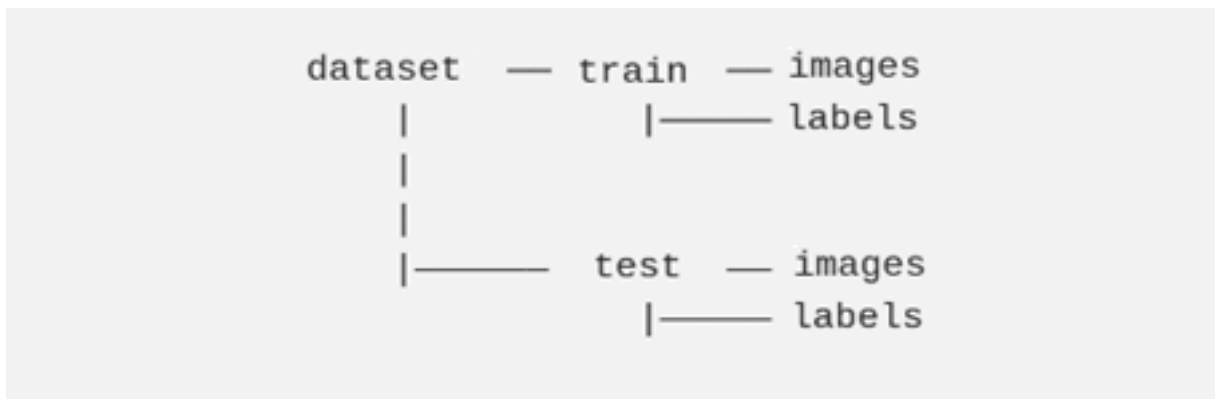


Figure 39: Structure de répertoire de l'ensemble de données.

3.4. Entraînement de modèle

Pour l'entraînement de modèle, nous avons décidé d'utiliser Google Colab pour accélérer l'entraînement. En utilisant les fonctionnalités des GPU offerts par Google Colab, nous avons pu accélérer considérablement le processus d'entraînement.

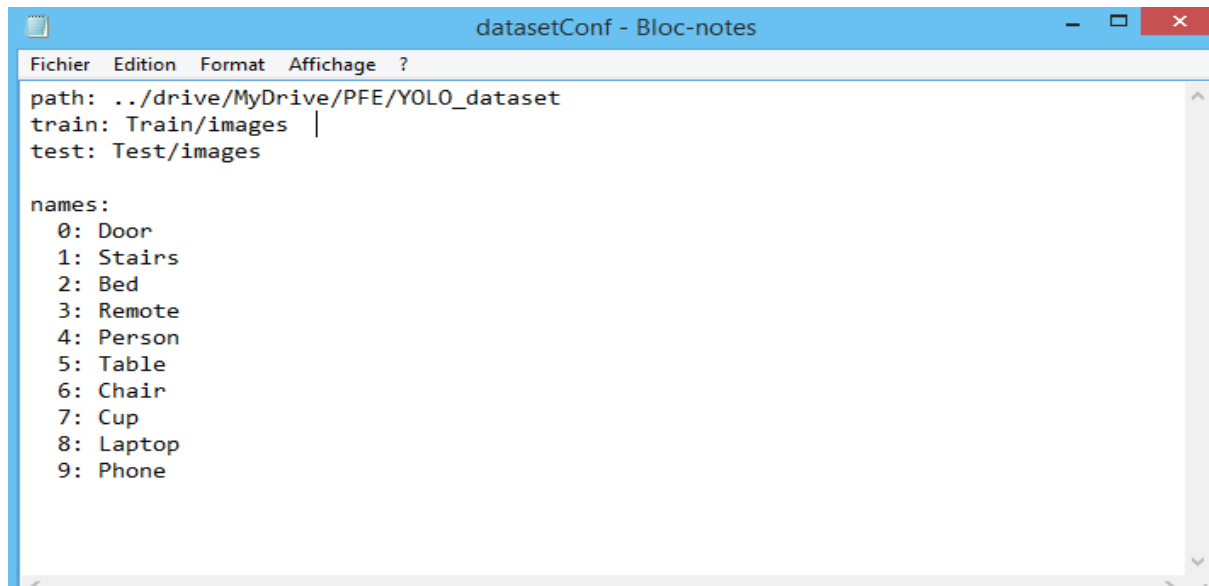
De plus, nous avons utilisé Google Drive pour héberger notre ensemble de données et importer tous les outils nécessaires dans l'entraînement. On peut même sauvegarder les poids

et les paramètres du modèle dans Google Drive une fois que notre modèle a été entraîné. Cela nous a permis de les récupérer facilement pour une utilisation future.

En utilisant Google Colab et Google Drive, on peut bénéficier des capacités de calcul puissantes des plateformes Cloud, en facilitant l'accès aux ensembles de données et accélérer le processus d'entraînement.

3.4.1. Fichier de configuration des données

Après avoir importé notre ensemble de données sur Google Drive, il est nécessaire de créer un fichier de configuration des données et l'importer lui aussi au Google Drive. Ce fichier est au format YAML, il contient des informations essentielles pour préparer et organiser les données d'entraînement et de test utilisées par le modèle YOLO. Ce fichier généralement inclus les détails suivants : le chemin vers les fichiers d'entraînement et de test, les classes des objets et les noms des classes.



```

datasetConf - Bloc-notes
Fichier Edition Format Affichage ?
path: ../drive/MyDrive/PFE/YOLO_dataset
train: Train/images
test: Test/images

names:
  0: Door
  1: Stairs
  2: Bed
  3: Remote
  4: Person
  5: Table
  6: Chair
  7: Cup
  8: Laptop
  9: Phone
  
```

Figure 40: Fichier de configuration de données.

3.4.2. Téléchargement du modèle pré-entraîné

Étant donné que notre ensemble de données n'est pas trop grand, l'apprentissage par transfert devrait produire de meilleurs résultats que l'entraînement à partir de zéro car la création d'un réseau de neurones complet à partir de zéro nécessite beaucoup de temps, de capacité de stockage et de puissance de calcul.

Donc, cette étape consiste à télécharger le modèle pré-entraîné YOLO plus précisément YOLOv5 qui est entraîné sur la base de données COCO. En utilisant ce modèle nous bénéficions de l'avantage de commencer avec des poids initiaux déjà ajustés. Enfin, après le

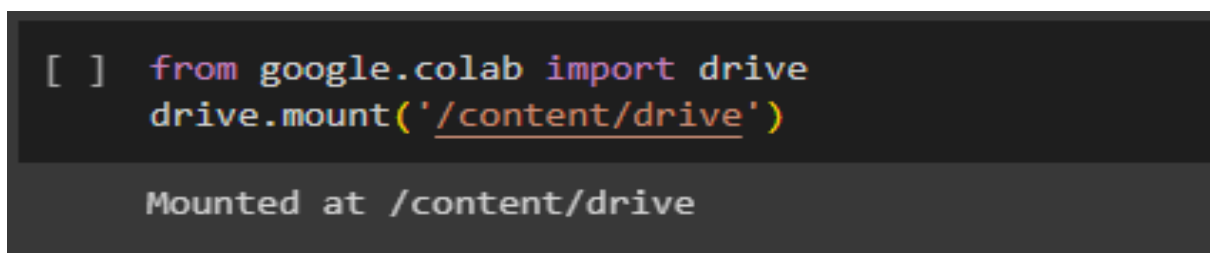
téléchargement, il faut placer le modèle dans le répertoire que nous avons déjà créé sur Google Drive.

Nous avons choisi de travailler avec l'algorithme YOLO pour plusieurs raisons :

- Vitesse : YOLO améliore la vitesse de détection, car il permet une détection en temps réel des objets. Cette caractéristique en fait un choix adapté pour notre application, qui requiert une détection rapide des objets en temps réel, notamment pour les personnes malvoyantes.
- Haute précision : La précision élevée est une autre caractéristique clé de YOLO. En tant que méthode prédictive, elle offre des résultats précis avec peu d'erreurs de fond.
- Capacités d'apprentissage : Il est capable d'apprendre les représentations des objets à partir des données d'entraînement et d'appliquer ces connaissances à la détection d'objets.

3.4.3. Lancement d'entraînement

Pour entraîner le modèle, nous avons créé dans Google Colab un notebook Python dédié à l'entraînement du modèle de détection. Nous avons importé les bibliothèques et les dépendances nécessaires, telles que TensorFlow et PyTorch. Ensuite, nous avons connecté Google Drive à notre environnement de Colab afin de pouvoir accéder à notre ensemble de données en exécutant le code de la figure 41.

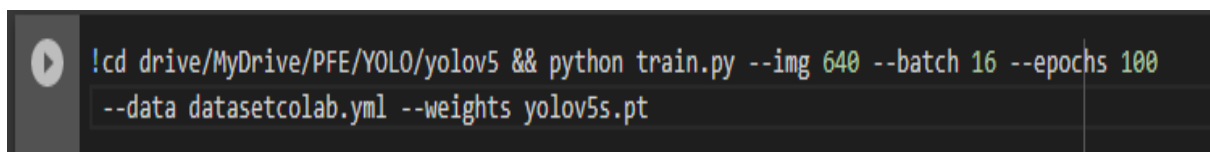


```
[ ] from google.colab import drive
    drive.mount('/content/drive')

Mounted at /content/drive
```

Figure 41: Connexion au compte Google Drive.

Enfin on peut Lancer l'entrainement en exécutant cette ligne de code :



```
!cd drive/MyDrive/PFE/YOLO/yolov5 && python train.py --img 640 --batch 16 --epochs 100
  --data datasetcolab.yml --weights yolov5s.pt
```

Figure 42: Commande d'entrainement de modèle YOLO.

Les paramètres de l'entraînement de modèle sont :

- **"img"** : Ce paramètre fait référence à la taille des images d'entrée lors de la détection d'objets.
- **"batch"** : La taille du lot utilisé lors de l'entraînement du modèle.
- **"epochs"** : Le nombre d'itérations effectuées lors de l'entraînement du modèle.
- **"data"** : Chemin d'accès au fichier de configuration des données.
- **"Weights"** : Chemin d'accès au poids du modèle pré-entraîné utilisés.

3.5. Évaluation du modèle

Après que l'entraînement du modèle a été terminé avec 100 époques (itérations), nous pouvons afficher les métriques de pertes et de précision qui sont enregistrées dans le fichier "results.png".

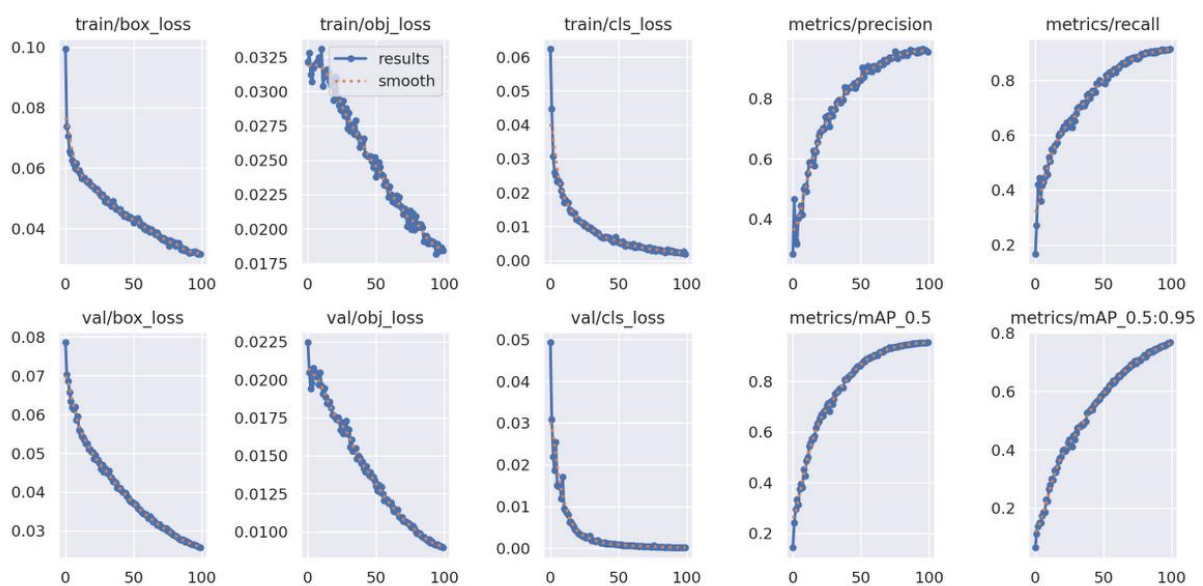


Figure 43: Métrique de perte et de précision du modèle YOLO.

Voici une explication de chaque métrique :

- **P (Précision)** : La précision mesure la proportion d'objets détectés correctement par rapport au nombre total d'objets détectés.
- **R (Rappel)** : Le rappel mesure la proportion d'objets détectés correctement par rapport au nombre total d'objets réels présents dans l'image.
- **box_loss (perte des boîtes englobantes)** : La perte des boîtes englobantes mesure l'écart entre les boîtes englobantes prédites par le modèle et les boîtes englobantes réelles présentes dans les images d'entraînement.
- **obj_loss (perte d'objets)** : La perte d'objets mesure l'exactitude de la prédiction de la présence ou de l'absence d'objets dans une boîte englobante.
- **cls_loss (perte de classification)** : La perte de classification mesure l'exactitude de la prédiction de la classe de l'objet contenu dans une boîte englobante. [47]

On peut aussi voir toutes ces métriques de précision et de perte pour chaque objet de l'ensemble de données comme présenté dans la figure suivante :

Class	Images	Instances	P	R	mAP50	mAP50-95
all	1380	3005	0.956	0.914	0.954	0.767
Door	1380	202	0.926	0.863	0.935	0.712
Stairs	1380	199	0.98	0.978	0.987	0.82
Bed	1380	185	0.926	0.962	0.98	0.808
Remote	1380	156	0.995	0.981	0.992	0.904
Person	1380	670	0.925	0.761	0.877	0.585
Table	1380	262	0.957	0.954	0.974	0.736
Chair	1380	794	0.92	0.705	0.832	0.553
Cup	1380	178	0.985	0.994	0.995	0.88
Laptop	1380	179	0.965	0.966	0.987	0.823
Phone	1380	180	0.983	0.975	0.978	0.852

Figure 44: Résultats de l'apprentissage obtenu pour chaque objet.

3.5.1. Test et résultat

Nous avons effectué des tests sur des images sélectionnées de manière aléatoire, ce qui nous a permis d'obtenir les résultats suivants :



Figure 45: Résultat obtenu de détection par yolov5 entraîné.

4. Implémentation de modèle d'Image Captioning

La deuxième étape de développement de cette application est l'implémentation d'un modèle d'Image Captioning pour la génération automatique des descriptions textuelles des images. Cependant, avant d'opter pour cette approche, nous avons exploré une autre méthode dans laquelle nous avons tenté d'utiliser les résultats de notre modèle de détection YOLO pour établir des relations entre les objets détectés.

Par exemple, on a défini une règle selon laquelle si la coordonnée y_2 de l'objet A se situe entre les coordonnées y_1 et y_2 de l'objet B, et que les coordonnées x_1 et x_2 de l'objet A se situent entre les coordonnées x_1 et x_2 de l'objet B, alors l'objet A est considéré comme étant en dessous de l'objet B.

Malheureusement, les résultats obtenus avec cette méthode n'ont pas été à la hauteur de nos attentes en ce qui concerne la génération automatique de descriptions d'images. Les descriptions générées étaient souvent incorrectes ou peu cohérentes. Les relations spatiales entre les objets ne sont pas toujours suffisantes pour capturer la signification et le contexte d'une image.

C'est pourquoi nous avons choisi le modèle d'Image Captioning, qui utilise des réseaux neuronaux profonds pour analyser l'image et générer des descriptions basées sur le mécanisme de transformateur. Nous allons détailler les étapes de préparation et du prétraitement du jeu de données, ainsi que l'architecture spécifique du modèle transformateur utilisée.

4.1. Choix de l'ensemble de données

Pour former et évaluer des modèles d'Image Captioning, il est essentiel d'avoir accès à des ensembles de données appropriés. Parmi les ensembles de données les plus populaires, il y a Flickr et COCO qui sont tous sous la forme [image \rightarrow caption] comme nous avons présentés dans le chapitre précédent. Au début, nous avons choisi Flickr8k en raison du nombre réduit d'images dans l'ensemble de données (8 000) par rapport à Flickr30k (30 000) et COCO (82 000). Cette décision a été prise afin d'accélérer le processus d'entraînement. Cependant, nous avons constaté un inconvénient de l'ensemble de données Flickr, à savoir que toutes les images de ce jeu de données représentent des êtres vivants, par exemple : "a man is walking...", "a cat is sitting...", "a dog is running...", "a woman is...". Donc, dans la phase de prédiction même si on passe au modèle une image ne contenant pas d'être vivant, le résultat sera toujours similaire aux exemples précédents.

Finalement, nous avons choisi d'utiliser le jeu de données COCO (Common Objects in Context) qui couvre une grande variété de scènes et d'objets plus que l'ensemble de donnée Flickr. Ce jeu de données est composé de 82 000 images, chacune ayant au moins 5 descriptions différentes. Cependant, nous n'avons pas utilisé l'ensemble de données complet. Nous avons plutôt sélectionné aléatoirement un sous-ensemble de 10 000 images à partir du jeu de données

COCO initial comme présenté dans la figure 46, car l'utilisation de l'ensemble de données complet nécessite beaucoup de temps, de capacité de stockage et de puissance de calcul.

```
[ ] indices=[]
for a in data['annotations']:
    indices.append(a['image_id'])
indices = random.sample(list(set(indices)), 10000)
len(indices)

10000
```

Figure 46: Sélection d'un sous-ensemble de dataset.

À la fin de cette étape, nous disposons d'un sous-ensemble aléatoire de 10 000 images provenant du jeu de données COCO chaque image de cet ensemble de donnée à 5 descriptions (Caption) différentes, qui vont être utilisés pour l'entraînement du modèle d'Image Captioning.



Figure 47: Image avec sa description de l'ensemble de données COCO.

4.2. Préparation de l'ensemble de données

Pour entraîner un bon modèle d'Image Captioning, il est important d'avoir un ensemble de données bien préparé qui comprend plusieurs aspects clés, tels que le prétraitement des images et du texte des descriptions (les captions) et la division de l'ensemble de données en un ensemble d'apprentissage et de test.

4.2.1. Prétraitement des images

Le prétraitement des images est une étape clé dans la génération de descriptions d'images automatiques pour garantir de bons résultats et de meilleures performances du modèle. Le prétraitement transforme les images brutes en un format adapté à l'analyse et à l'Apprentissage Automatique. Pour cette raison, nous avons appliqué plusieurs techniques de prétraitement à l'ensemble de données, telles que le décodage, la normalisation et le redimensionnement d'image, pour l'adapter au format du modèle.

```
[ ] def load_data(img_path, caption):  
    img = tf.io.read_file(img_path)  
    img = tf.io.decode_jpeg(img, channels=3)  
    img = tf.keras.layers.Resizing(299, 299)(img)  
    img = img / 255.  
    caption = tokenizer(caption)  
    return img, caption
```

Figure 48: Prétraitement d'images pour la tâche d'Image Captioning.

Tout d'abord, nous avons décodé l'image à partir de son format brut (JPEG) pour obtenir une représentation au format tenseur. Ensuite, nous avons redimensionné l'image aux dimensions de 299x299 pixels. Comme mentionné précédemment, nous utilisons le modèle InceptionV3 en tant qu'un réseau de neurones convolutifs (CNN) pour extraire les caractéristiques des images. Et pour s'adapter à ce modèle, les images doivent être redimensionnées en 299x299 pixels. C'est pourquoi nous avons redimensionné les images à cette taille.

Ensuite, nous avons normalisé les valeurs des pixels de l'image redimensionnée en les divisant par 255. Cette opération permet de mettre toutes les valeurs des pixels sur une échelle comprise entre 0 et 1. La normalisation facilite l'entraînement des modèles en assurant que les caractéristiques ont des échelles similaires.

Nous avons également effectué une augmentation des images en appliquant des déformations géométriques. Tout d'abord, nous avons effectué une inversion aléatoire horizontale des images. Ensuite, nous avons effectué une rotation aléatoire des images, ce qui permet de varier légèrement l'orientation des objets présents dans les images. Enfin, nous avons

appliqué une augmentation aléatoire du contraste des images, cela permet de rendre les images plus dynamiques en ajustant les niveaux de luminosité et de contraste. L'ensemble de ces opérations d'augmentation d'images permet aux modèles de mieux distinguer les caractéristiques invariantes et d'augmenter le nombre d'images présentant les mêmes concepts et descriptions.

```
[ ] image_augmentation = tf.keras.Sequential(
    [
        tf.keras.layers.RandomFlip("horizontal"),
        tf.keras.layers.RandomRotation(0.2),
        tf.keras.layers.RandomContrast(0.3),
    ]
)
```

Figure 49: Augmentation de l'ensemble des images.

4.2.2. Prétraitement du texte (les descriptions)

Après avoir préparé et prétraité les images, nous devons faire de même pour le texte qui correspond aux descriptions de ces images. Cette étape est fondamentale pour former un bon modèle d'Image Captioning pour obtenir une description sémantiquement précise d'une image. Le traitement des données textuelles nécessite plusieurs étapes.

La première étape du prétraitement consiste à nettoyer le texte. Tout d'abord, nous avons converti le texte en minuscules pour normaliser la casse. Ensuite, nous avons supprimé les caractères indésirables. Les espaces supplémentaires sont ensuite réduits à un seul espace pour normaliser la séparation entre les mots, et les espaces de début et de fin sont supprimés de la chaîne. Enfin, nous ajoutons des balises spéciales ("[start]" et "[end]") pour marquer respectivement le début et la fin du texte.

```
def preprocess(text):
    text = text.lower()
    text = re.sub(r'^\w\s', '', text)
    text = re.sub('\s+', ' ', text)
    text = text.strip()
    text = '[start] ' + text + ' [end]'
    return text
```

Figure 50: Nettoyage du texte.

Une autre étape importante que nous utilisons dans le traitement de texte est la tokenisation, qui divise le texte en unités plus petites appelées "tokens". Le modèle est ensuite configuré avec un nombre maximal de tokens et une longueur de séquence de sortie définie.

Ensuite, nous ajustons le modèle pour lui permettre d'apprendre le vocabulaire à partir des données textuelles fournies. Cela permet de créer un vocabulaire basé sur les tokens uniques trouvés dans nos données, ce qui permet au modèle de comprendre et de générer des séquences de texte cohérentes et pertinentes.

```
tokenizer = tf.keras.layers.TextVectorization(
    max_tokens=VOCABULARY_SIZE,
    standardize=None,
    output_sequence_length=MAX_LENGTH)

tokenizer.adapt(captions['caption'])
```

Figure 51: Tokenisation et Création du vocabulaire del'ensemble de textes.

4.2.3. Division de l'ensemble de données

Enfin, il est important de diviser l'ensemble de données en ensembles d'apprentissage et de validation. Cela nous permet d'évaluer les performances des modèles d'Image Captioning. Comme mentionné précédemment, nous disposons d'un ensemble de 10 000 images, chacune avec 5 descriptions différentes, ce qui donne un total de 40 000 images dans l'ensemble de données. Nous divisons cet ensemble en utilisant une répartition de 80% pour l'ensemble d'apprentissage et 20% pour l'ensemble de validation, et donc la répartition de chaque ensemble sera comme présentée dans la figure suivante. Ainsi, nous pouvons entraîner notre modèle sur l'ensemble d'apprentissage et évaluer sa performance sur l'ensemble de validation afin de mesurer sa capacité à générer des descriptions précises et cohérentes pour de nouvelles images.

```
[19] len(train_imgs), len(train_captions), len(val_imgs), len(val_captions)

(40012, 40012, 10009, 10009)
```

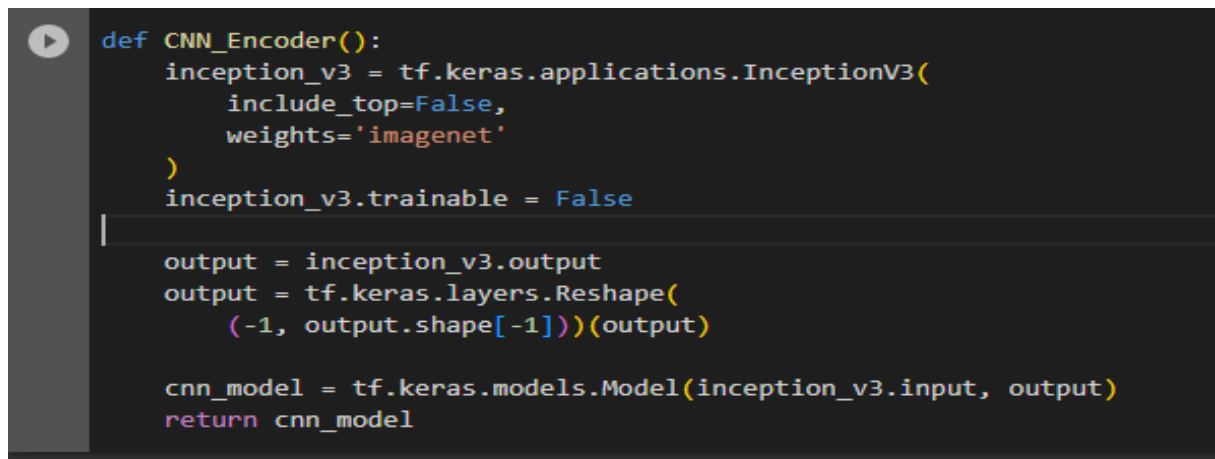
Figure 52: Répartition de chaque ensemble de données.

4.3. Entraînement de modèle

Étant donné que notre ensemble de données est assez grand, l'entraînement du modèle sur nos propres machines va prendre beaucoup de temps. Nous allons suivre la même approche que nous avons utilisée pour entraîner le modèle de détection d'objets, qui consiste à utiliser les capacités de calcul puissantes des plateformes Cloud telles que Google Colab et Google Drive. Cela facilite l'accès aux ensembles de données et accélère le processus d'entraînement. Donc, nous allons répéter les mêmes étapes pour préparer l'environnement de travail en se connectant à Google Drive avec Colab, puis télécharger l'ensemble de données dans Colab et importer les bibliothèques nécessaires.

4.3.1. Construction du modèle

Le processus de création d'un modèle d'Image Captioning commence généralement par l'utilisation d'un CNN pré-entraîné sur un grand ensemble de données. Dans notre projet nous avons utilisé InceptionV3 comme un CNN pour l'extraction des caractéristiques des images. Comme montré dans la figure 53.



```
def CNN_Encoder():
    inception_v3 = tf.keras.applications.InceptionV3(
        include_top=False,
        weights='imagenet'
    )
    inception_v3.trainable = False

    output = inception_v3.output
    output = tf.keras.layers.Reshape(
        (-1, output.shape[-1]))(output)

    cnn_model = tf.keras.models.Model(inception_v3.input, output)
    return cnn_model
```

Figure 53: Modèle InceptionV3 pour l'extraction des caractéristiques.

Une fois les caractéristiques visuelles extraites du CNN, elles sont fournies au modèle transformateur encodeur. Le transformateur encodeur est une architecture basée sur le mécanisme d'attention qui permet de capturer la relation entre les caractéristiques visuelles et les mots des descriptions. Cette architecture nous permet de mieux capturer les relations complexes entre différentes parties d'images et de mots lors de la génération de descriptions.

Le transformateur encodeur est ensuite suivi d'un transformateur décodeur. Le transformateur décodeur est chargé de générer les séquences de mots qui composent la description de l'image. Il prend en entrée les caractéristiques visuelles encodées par le transformateur encodeur et les mots générés précédemment. À chaque étape, le transformateur décodeur utilise l'attention pour se focaliser sur les parties importantes des caractéristiques visuelles et génère le mot suivant dans la séquence de description.

La combinaison du transformateur encodeur et décodeur permet de créer un modèle capable de comprendre les caractéristiques visuelles des images et de générer des descriptions cohérentes et sémantiquement précises.

- Lancement de l'entraînement :

```
[ ] history = caption_model.fit(
    train_dataset,
    epochs=100,
    validation_data=val_dataset,
)
```

```
Epoch 1/100
5030/5030 [=====] - ETA: 0s - loss: 3.6889 -
Epoch 1: saving model to /content/drive/MyDrive/PFE/transformer_model
5030/5030 [=====] - 1611s 315ms/step - loss:
Epoch 2/100
```

Figure 54: Commande d'entraînement de modèle d'Image Captioning.

4.3.2. Prédiction

Une fois que le modèle d'Image Captioning est entraîné, il peut être utilisé pour effectuer des prédictions sur de nouvelles images. Le processus de prédiction consiste à fournir une image en entrée au modèle et de la faire passer à travers le CNN pour extraire les caractéristiques visuelles. Ces caractéristiques sont ensuite transmises au transformateur encodeur, qui encode les relations entre les caractéristiques visuelles et les mots.

Pour générer une prédiction de description, on utilise le transformateur décodeur. Au début, une séquence de démarrage est fournie, qui est le jeton spécial "[start]". Ensuite, le modèle génère un mot à la fois en utilisant l'attention pour se concentrer sur les parties pertinentes de l'image. Le mot généré est ensuite utilisé comme entrée pour la prédiction du mot suivant. Ce processus est répété jusqu'à ce qu'a la séquence de fin spéciale "[end]" soit générée ou jusqu'à ce qu'une longueur maximale prédéfinie soit atteinte.


```
def generate_caption(img_path):
    img = load_image_from_path(img_path)
    img = tf.expand_dims(img, axis=0)
    img_embed = caption_model.cnn_model(img)
    img_encoded = caption_model.encoder(img_embed, training=False)

    y_inp = '[start]'
    for i in range(MAX_LENGTH-1):
        tokenized = tokenizer([y_inp])[:, :-1]
        mask = tf.cast(tokenized != 0, tf.int32)
        pred = caption_model.decoder(
            tokenized, img_encoded, training=False, mask=mask)

        pred_idx = np.argmax(pred[0, i, :])
        pred_word = idx2word(pred_idx).numpy().decode('utf-8')
        if pred_word == '[end]':
            break

        y_inp += ' ' + pred_word

    y_inp = y_inp.replace('[start] ', '')
    return y_inp
```

Figure 55: Méthode de prédiction d'une description d'image.

4.3.3. Test et résultat

Nous avons effectué des tests sur des images sélectionnées de manière aléatoire, ce qui nous a permis d'obtenir ces résultats suivants :



Figure 56: Résultat obtenu de prédiction par le modèle d'Image Captioning.

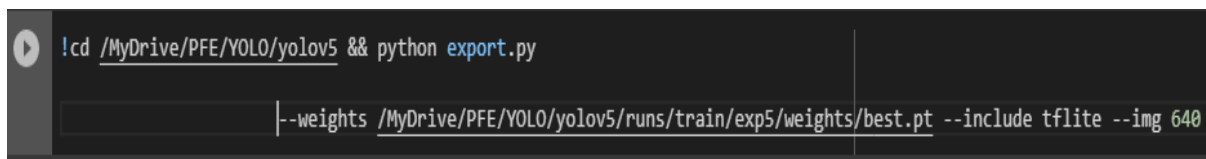
5. Conversion en TFLITE

Après que les deux modèles de détection d'objets et d'Image Captioning ont été entraînés, une étape importante pour les implémenter dans notre application mobile consiste à les convertir en TensorFlow Lite.

La conversion en TensorFlow Lite est essentielle car elle permet de compresser et d'optimiser les modèles afin qu'ils puissent être exécutés efficacement sur des appareils mobiles. TensorFlow Lite est une version allégée du framework TensorFlow, spécifiquement conçue pour les dispositifs avec des ressources limitées, tels que les smartphones et les tablettes.

5.1. Conversion de modèle de détection d'objets

Après avoir entraîné notre modèle de détection YOLOv5, les poids du modèle sont enregistrés au format PyTorch (".pt"), pour convertir ce modèle en TensorFlow Lite (TFLite) on peut utiliser cette commande :



```
!cd /MyDrive/PFE/YOLO/yolov5 && python export.py
--weights /MyDrive/PFE/YOLO/yolov5/runs/train/exp5/weights/best.pt --include tflite --img 640
```

Figure 57: Conversion du modèle YOLOv5 au TFLITE.

Voici une description des paramètres de ce code :

"**--weights**" : C'est un argument pour spécifier le chemin d'accès aux poids du notre modèle YOLOv5 entraîné.

"**—include tflite**" : C'est un paramètre pour indiquer l'inclusion du format TFLite lors de l'exportation du modèle. Cela signifie que le modèle sera également converti en format TFLite en plus des autres formats possibles.

"**--img 640**" : C'est un argument pour spécifier la résolution d'image à utiliser lors de la conversion du modèle. Dans notre exemple, la résolution d'image est fixée à 640 pixels.

5.2. Conversion de modèle d'Image Captioning

Après l'entraînement du modèle d'Image Captioning, nous avons enregistré les poids du modèle au format H5 dans le répertoire Google Drive. Pour convertir ce modèle, il faut le charger puis le convertir avec la commande suivante:

```
[ ] model.load_weights('/MyDrive/PFE/transformer_model/image_captioning_transformer_weights.h5')
    converter= tf.lite.TFLiteConverter.from_keras_model(model)
    tflite_model = converter.convert()
    with open("caption.tflite","wb") as f:
        f.write(tflite_model)
```

Figure 58: Conversion de modèle d'Image Captioning au TFLITE.

On charge le poids de modèle d'Image Captioning basé sur le transformateur à partir d'un fichier h5, puis on convertit le modèle Keras en un modèle TFLite, puis on enregistre le modèle TFLite converti dans un fichier "caption.tflite".

6. Choix du modèle

Après la conversion des deux modèles en TensorFlow Lite, nous pouvons maintenant les intégrer à Android Studio pour mettre en œuvre notre application mobile Android. La première étape de cette application consiste à permettre aux utilisateurs de choisir le type de modèle qui leur convient le mieux. Lorsqu'un utilisateur malvoyant prononce le nom d'une catégorie, soit "Find Object" ou "Describe", le modèle correspondant sera activé.

L'algorithme du choix de modèle, est présenté dans la figure suivante :

```
public void onResults(Bundle results) {
    ArrayList<String> matches = results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
    if (matches != null && !matches.isEmpty()) {
        String spokenText = matches.get(0);
        if (spokenText.equalsIgnoreCase( anotherString: "describe")) {
            Caption( view: null);
        } else if (spokenText.equalsIgnoreCase( anotherString: "find object")) {
            Yolo( view: null) ;
        }
    }
}
```

Figure 59: Code de choix de modèle.

7. Conversion aux résultats vocaux

Pour faciliter la compréhension des personnes malvoyantes des résultats du traitement, nous avons ajouté deux méthodes pour convertir les résultats en audio. Une méthode est dédiée aux résultats de détection d'objets, tandis que l'autre est spécifique aux résultats de génération

de descriptions d'images. Ces méthodes vont aider les utilisateurs malvoyants, car ils dépendent principalement de l'audio pour percevoir l'information.

- Méthode de conversion pour la détection d'objets : Pour les résultats de la détection d'objets, nous avons créé la méthode "playSound()" qui génère une notification sonore lorsqu'un objet est détecté.

```
10 usages
private void playSound() {
    mediaPlayer = MediaPlayer.create(context, R.raw.sound);
    mediaPlayer.start();
}
```

Figure 60: Méthode de conversion aux résultats vocaux pour la détection d'objets.

- Méthode de conversion pour l'Image Captioning : Pour les résultats de l'Image Captioning, nous avons créé la méthode "ConvertTextToSpeech()" qui convertit le texte généré par le modèle d'Image Captioning en voix.

```
private void convertTextToSpeech(String text) {
    if (textToSpeech != null && !text.isEmpty()) {
        textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, params: null, utterancelid: null);
    }
}
```

Figure 61: Méthode de conversion aux résultats vocaux pour l'Image Captioning.

8. Interfaces et scénario d'exécution

Cette section permet de simuler l'exécution de l'application mobile par un utilisateur tout en présentant les différentes interfaces qui composent l'application. Nous allons simuler le choix entre la détection d'objet et l'Image Captioning, décrire les scènes avec l'Image Captioning et détecter des objets tout en affichant les résultats sur les interfaces de l'application mobile.

Au niveau de la page d'accueil, nous pouvons choisir la détection d'objet en appuyant sur le côté droit de l'écran, ou l'Image Captioning en appuyant sur le côté gauche de l'écran (Voir figure 62).

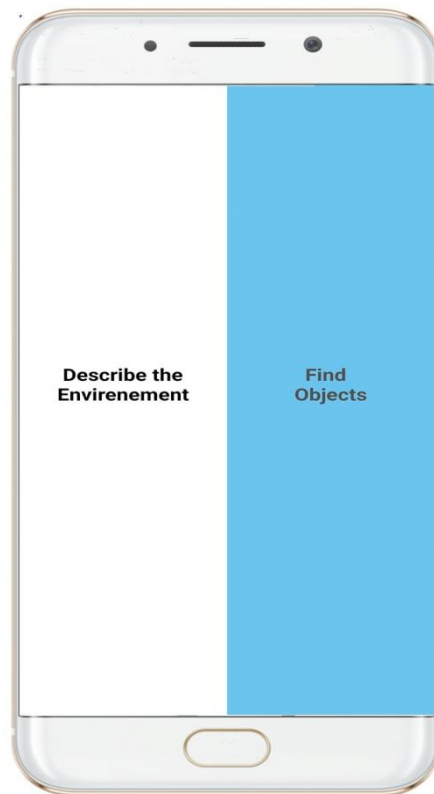


Figure 62: Page d'accueil.

Au lieu d'appuyer sur la droite ou la gauche. L'utilisateur peut aussi maintenir pression sur l'écran, en demandant oralement « Find Object » si l'utilisateur veut aller sur la page de la détection d'objet, ou dire « Describe » pour accéder à la page de l'Image Captioning (Voir Figure 63).

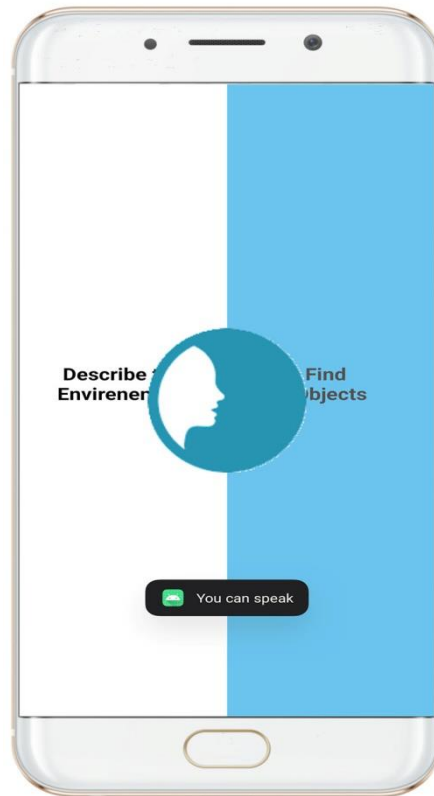


Figure 63: Page d'accueil avec commande orale.

Si l'utilisateur a décidé d'aller sur la page de description, il peut presser sur le bouton de capture pour prendre une photo, ou il peut rester presser sur l'écran et l'application va lui donner la main pour parler, il suffit de dire « Take Picture » et après deux secondes l'application va lui donner une description orale. Dans la figure 64, l'application va donner comme une description « A laptop computer sitting on the top of a desk ».

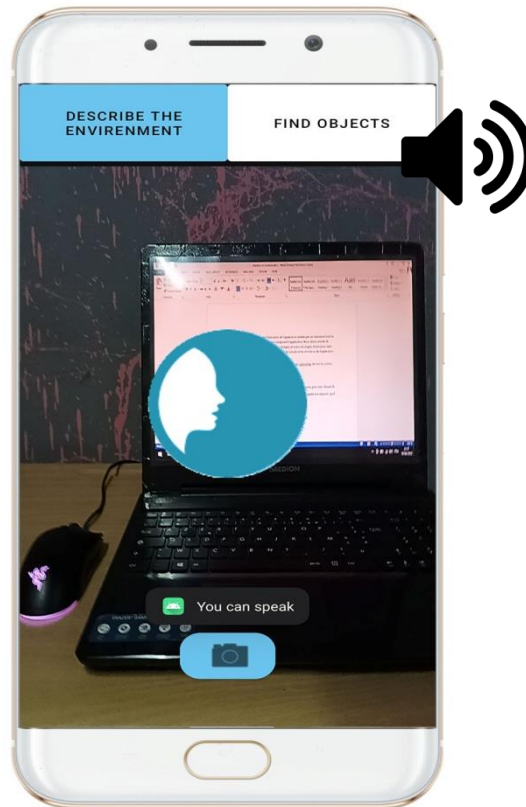


Figure 64: Génération d'une description vocale de l'image.

Dans le cas où l'utilisateur souhaite retrouver un objet qu'il a perdu et qu'il ne peut pas le voir correctement, ou pour une personne non-voyante qui recherche quelque chose, l'utilisateur peut cliquer sur le côté droit de l'écran ou appuyer et maintenir n'importe où sur l'écran, puis dire « Find Object » pour accéder à la page de détection d'objet. Sur cette page de détection (voir Figure 65).

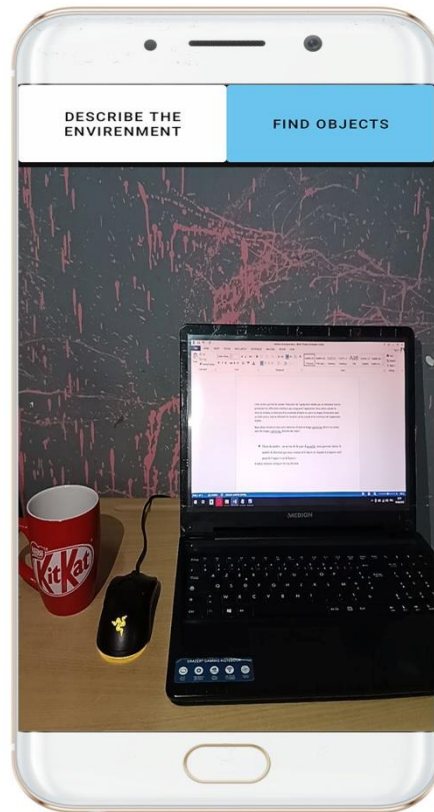


Figure 65: Interface pour la détection d'objets.

Une fois de plus, l'utilisateur doit maintenir l'appui et l'application donnera la possibilité à l'utilisateur de prononcer le nom de l'objet qu'il souhaite trouver. Dans la figure 66, par exemple, l'utilisateur a dit "LAPTOP", ce qui permet à l'application d'entrer en mode LIVE. Si l'utilisateur repère l'objet lors de la navigation, l'application le signalera avec un son (bip) pour alerter l'utilisateur.

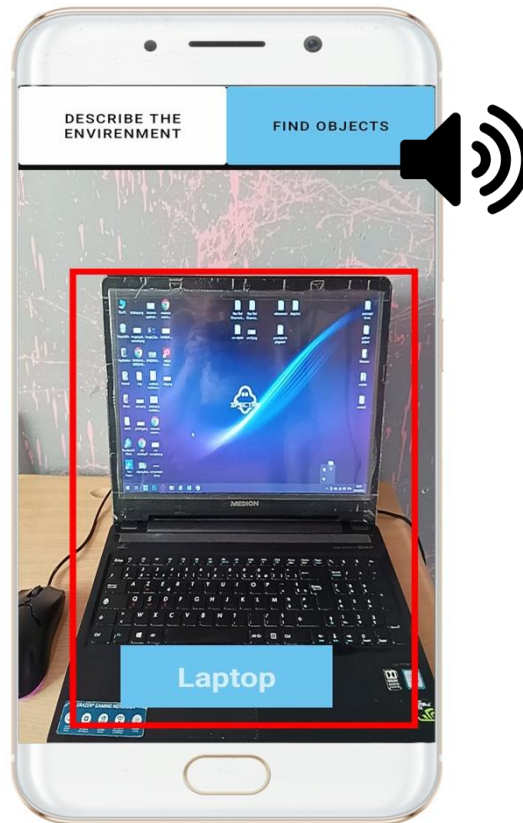


Figure 66: Notification quand un objet est trouvé.

9. Conclusion

Dans ce dernier chapitre, nous avons couvert tous les aspects essentiels du développement de notre système. Nous avons commencé par une introduction aux outils utilisés, puis nous avons abordé l'implémentation des différents modèles et les fonctionnalités de l'application mobile développée. Enfin, nous avons conclu en présentant un scénario d'exécution simulant l'utilisation du système par un utilisateur.

Conclusion générale

La vision humaine est un sens fondamental qui contribue grandement à notre compréhension du monde qui nous entoure. Notre perception et notre expérience quotidienne dépendent de notre capacité à voir et à interpréter visuellement les informations visuelles. La perte de ce sens provoque un grand changement pour ces personnes, et rend leur vie très difficile et dépendante de l'aide des autres. C'est dans ce but que les récentes solutions utilisent l'Intelligence Artificielle et plus précisément la Vision par Ordinateur comme moyen d'imiter le fonctionnement de l'œil humain.

Dans ce sens, notre architecture consiste dans un premier temps d'entraîner un modèle de détection d'objets pour pouvoir trouver des objets perdu ou difficile à trouver pour les personnes malvoyantes. Puis dans un second temps, d'entraîner un modèle de description automatique des images, pour pouvoir décrire des scènes qui se trouvent devant cette personne.

En résumé, ce projet a contribué à améliorer notre capacité d'analyse des problèmes et à développer une réflexion plus efficace pour les résoudre, tout en respectant les délais. De plus, il nous a permis d'acquérir de nouvelles connaissances sur plusieurs concepts et technologies, surtout dans l'Intelligence Artificielle et l'Apprentissage Profond. En outre, ce projet nous a donné la possibilité d'apprendre un nouveau langage très populaire et apprécié : le Python.

Nous avons réussi à atteindre les principaux objectifs fixés au départ. Nous avons développé une application mobile qui est utilisable par les malvoyants et qui offre les fonctionnalités spécifiques suivantes :

- L'assistance des personnes malvoyantes dans la découverte de l'environnement qui les entoure par une description de l'environnement.
- L'assistance des personnes malvoyantes dans la détection des objets spécifiques.

Ce travail offre encore des possibilités d'amélioration et d'approfondissement. Plusieurs vues d'améliorations et de perspectives sont à énumérer, dont nous citons quelques-uns :

- Augmenter le pourcentage de précision des modèles utilisés.
- Augmenter le nombre objets détectés.
- Mettre les modèles dans le Cloud et utiliser des API.

Références

- [1] mondiale de la Santé, O. (2020). Rapport mondial sur la vision.
- [2] Magalie, G. R. E. G. O. I. R. E. Le développement de la fonction visuelle chez l'enfant: étude bibliographique.
- [3] Dr Laurent Leininger. Publié le 10/06/2013. L'œil : l'organe de la vision.
- [4] Cambois, E., & Robine, J. M. (2003). Concepts et mesure de l'incapacité: définitions et application d'un modèle à la population française. *Retraite et société*, (2), 59-91.
- [5] Pr Dominique Brémond-Gignac, Pr Matthieu Robert .PRISE EN CHARGE DE LA BASSE VISION.Edimark.
- [6] Cécité et déficience visuelle. (s.d.). Consulté le 30, 2023, sur Organisation mondiale de la Santé:<https://www.who.int/fr/news-room/fact-sheets/detail/blindness-and-visual-impairment#:~:text=Les%20principales%20causes%20de%20d%C3%A9ficience,les%20personnes%20de%20tous%20%C3%A2ges.>
- [7] SciencesPo. Décembre 2017 .Handicap visuel : fiche technique à visée informative et pédagogique.
- [8] Robert, P. Y. (2018). L'accompagnement de la déficience visuelle chez l'adulte, de l'organe au fonctionnel. *Revue Francophone d'Orthoptie*, 11(1), 41-43.
- [9] Maladies des yeux. (s.d.). Consulté le 6 2, 2023, sur The International Agency for the Prevention of Blindness: <https://www.iapb.org/fr/learn/knowledge-hub/eye-conditions/>
- [10] VUE, L. GLAUCOME.
- [11] Torossian, M. (2021). Classification déficience visuelle. *Revue Francophone d'Orthoptie*, 14(3), 102-103.
- [12] Robert, P. Y. (2017). Déficiences visuelles: Rééducations et Réadaptations-Rapport de la Société française d'ophtalmologie. Elsevier Health Sciences.
- [13] Vittrant, D., & Le Cunff, E. (2020). Améliorer la mobilité des personnes malvoyantes et non-voyantes: focus sur la canne blanche électronique. *Revue Francophone d'Orthoptie*, 13(1), 46-50.
- [14] ROUTON, M., & FERRANT, M. Aides à la lecture et déficience visuelle.
- [15] Maumet, L. (2007). L'accès à l'écrit des personnes déficientes visuelles.
- [16] Tessier, M., & Guigou, S. Applications santé connectées: que conseiller à nos patients?.

- [17] Union Nationale des Aveugles et Déficients Visuels. (s.d.). Les principales pathologies de la vue. Consulté le 6 juin 2023, sur <https://www.unadev.com/le-handicap-visuel/les-principales-pathologies/>
- [18] Haiech, J. (2020). Parcourir l'histoire de l'intelligence artificielle, pour mieux la définir et la comprendre. *médecine/sciences*, 36(10), 919-923.
- [19] Benyekhlef, K., & Zhu, J. (2018). Intelligence artificielle et justice: justice prédictive, conflits de basse intensité et données massives. *Intelligence*, 30(3).
- [20] MRHARI, A., & DINAR, Y. (2018). INTELLIGENCE ARTIFICIELLE: QUEL AVENIR POUR LE MARKETING?. *PUBLIC & NONPROFIT MANAGEMENT REVIEW*, 3(1).
- [21] Ryax Technologies. (s.d.). Quelles différences entre intelligence artificielle et machine learning ? Consulté le 27 mai 2023, sur <https://ryax.tech/fr/differences-intelligence-artificielle-machine-learning/>
- [22] MEHIEDDINE, O., & HERIZI, S. (2021). Développement d'un système intelligent pour prédire la satisfaction envers une agence touristique en se basant sur les avis des clients (Doctoral dissertation, FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE DÉPARTEMENT D'INFORMATIQUE-Spécialité: Informatique Décisionnel et Optimisation).
- [23] Labiad, A. (2017). Sélection des mots clés basée sur la classification et l'extraction des règles d'association (Doctoral dissertation, Université du Québec à Trois-Rivières).
- [24] Bellahmer, H. (2020). Implémentation et évaluation d'un modèle d'apprentissage automatique pour l'estimation de la valeur marchande de propriétés immobilières (Doctoral dissertation, Université Mouloud Mammeri).
- [25] Martinez, C., Ramasso, E., Perrin, G., & Rombaut, M. (2019, August). Apprentissage par renforcement profond pour la classification précoce de séquences temporelles. In *GRETSI 2019-XXVIIème Colloque francophone de traitement du signal et des images*.
- [26] Kelleher, J. D. (2019). *Deep learning*. MIT press.
- [27] Rusk, N. (2016). *Deep learning*. *Nature Methods*, 13(1), 35-35.
- [28] Baouche, R. (2015). Prédiction des Paramètres Physiques des Couches Pétrolifères par Analyse des Réseaux de Neurones et Analyse Faciologique (Doctoral dissertation, université M'hamed Bougara. Boumerdès).
- [29] Benchikha. 2022. Cour De Système Décisionnels M1 STIW. Elearning Univ Constantine 2
- [30] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, 77, 354-377.

- [31] Melki, R. (2019). Apprentissage des réseaux de neurones MLP par une méthode hybride à base d'une métaheuristique.
- [32] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
- [33] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31(7), 1235-1270.
- [34] Yahiaoui, N. Un système basé sur la vision par ordinateur pour la prévention des virus respiratoires (Cas du Covid-19) (Doctoral dissertation, UNIVERSITE KASDI MERBAH OUARGLA).
- [35] Boillet, M. (2023). Détection d'Objets dans les documents numériques par réseaux de neurones profonds. arXiv preprint arXiv:2301.11753.
- [36] sur le blog d'IMAIOS, B. PRATIQUE QUOTIDIENNE.
- [37] Radhakrishnan, P. (2017, 29 septembre). Image Captioning in Deep Learning. Consulté le 2 juin 2023, sur: <https://towardsdatascience.com/image-captioning-in-deep-learning-9cd23fb4d8d2>.
- [38] Mlle. ABDELLI Wafa. Juin 2019. Image Captioning . Mémoire Master Université Dr. TAHAR MOULAY SAIDA FACULTÉ : TECHNOLOGIE DÉPARTEMENT : INFORMATIQUE.
- [39] SAIDJ, S. D. (2022). Techniques de NLP pour la détection des fausses nouvelles (Doctoral dissertation, Université Ibn Khaldoun-Tiaret-).
- [40] Castilla-Arquillo, R., Pérez-del-Pulgar, C., Paz-Delgado, G. J., & Gerdes, L. (2022). Hardware-accelerated Mars Sample Localization via deep transfer learning from photorealistic simulations. *IEEE Robotics and Automation Letters*, 7(4), 12555-12561.
- [41] KHELALEF, A. (2021). Reconnaissance d'activités humaines en utilisant les descripteurs spatio-temporels 2D/3D (Doctoral dissertation, Université de Batna 2).
- [42] Randrianarivony, M. I. (2018). Détection de concepts et annotation automatique d'images médicales par apprentissage profond (Doctoral dissertation, Université d'Antananarivo).
- [43] Pitpitt. (n.d.). Annotation des données — DataFranca. Datafranca.org. Consulté le 4 juin 2023, sur: https://datafranca.org/wiki/Annotation_des_donn%C3%A9es
- [44] Ranjan, A. (2022, June 13). Image Captioning with an End-to-End Transformer Network | Python in Plain English. Medium. Consulté le 12 juin 2023 sur : <https://python.plainenglish.io/image-captioning-with-an-end-to-end-transformer-network-8f39e1438cd4>

[45] intro. (s.d.). Consulté le 05 2023, 24, sur developer:
<https://developer.android.com/studio/intro>

[46] colaboratory. (s.d.). Consulté le 05 mai 2023, sur research:
<https://research.google.com/colaboratory/faq.html?hl=fr#:~:text=Colaboratory%2C%20souvent%20raccourci%20en%20%22Colab,donn%C3%A9es%20et%20%C3%A0%20l'%C3%A9ducation.>

[47] Gad, A. F. (2020). Evaluating object detection models using mean average precision (mAP). PaperspaceBlog.

[48] tensorflow-definition-tout-savoir. (s.d.). Consulté le 05 2023, 24, sur lebigdata:
<https://www.lebigdata.fr/tensorflow-definition-tout-savoir>

[49] searchmobilecomputing. (s.d.). Consulté le 05 2023, 24, sur techtarget:
<https://www.techtartget.com/searchmobilecomputing/definition/Google-Drive#:~:text=Google%20Drive%20is%20a%20free,mobile%20devices%2C%20tablets%20and%20PCs.>

[50] yolo-custom-1. (s.d.). Consulté le 05 2023, 24, sur datacorner:
<https://www.datacorner.fr/yolo-custom-1/>