

Intégration de modèles connexionnistes dans un prototype de substitution visuelle



Thèse de Bachelor présentée par

Damian BOQUETE COSTA

pour l'obtention du titre Bachelor of Science HES-SO en

**Informatique et systèmes de communication avec orientation
Informatique logicielle**

Septembre 2023

Professeur-e HES responsable

Docteur Guido BOLOGNA

Légende et source de l'illustration de couverture : Génération par l'intelligence artificielle Midjourney correspondant à son interprétation du sujet en quelques mots clés, www.midjourney.com.

TABLE DES MATIÈRES

Remerciements	vi
Énoncé du sujet	vii
Résumé	viii
Liste de acronymes	ix
Liste des illustrations	x
Liste des tableaux	xii
Liste des annexes	xiii
Introduction	1
1 Chapitre 1 : Analyse de la problématique, de l'existant et introduction aux concepts généraux	3
1.1 La cécité	3
1.2 Les projets similaires	4
a <i>Be My Eyes</i>	4
b <i>TapTapSee</i>	4
c <i>Real-Time CV Based Autonomous Navigation System</i>	4
d Prototype de substitution de la vision par le toucher	5
e Synthèse	5
e.1 Le modèle de détection	5
e.2 Les fonctionnalités	5
1.3 L'intelligence artificielle	6
a Généralités	6
b Le Machine Learning	7
c Le Deep Learning	7
2 Chapitre 2 : Approfondissement des concepts	8
2.1 Généralités	8
2.2 Les images	9
2.3 Les réseaux de neurones convolutifs	10
2.4 Anatomie d'un réseau de neurones convolutifs	10
a La couche convulsive	11
b La couche d'agrégation	11
c La couche de détection d'objets	12
2.5 Les Transformers	14
2.6 Les Vision Transformers (ViT)	14
2.7 Anatomie d'un ViT	15
a Pré-traitement des données en entrée	16
b L'encodeur	16

c	Le mécanisme d'attention	18
d	La tête de détection	19
3	Chapitre 3 : Les technologies et ressources utilisées	21
3.1	Généralités	21
3.2	Matériel physique	21
3.3	Langage informatique	23
3.4	Bibliothèques logicielles	23
a	PyTorch	23
b	Hugging face	24
c	txtai	24
d	Open Computer Vision	24
e	NumPy	24
f	Autres bibliothèques	25
3.5	Jeux de données	25
4	Chapitre 4 : L'étude comparative	26
4.1	Généralités	26
4.2	Métriques utilisées	27
a	Score	27
b	Intersection over Union	27
c	Temps d'exécution d'une inférence	28
d	Confiance	28
e	Threshold	28
4.3	Méthodologie utilisée	29
4.4	Résultats et échantillons	31
a	Nombre de détections correctes	31
b	Score Intersection over Union (IoU)	32
c	Score de confiance	33
d	Temps d'exécution	34
e	Échantillons visuels	35
4.5	Décision	36
5	Chapitre 5 : L'anatomie du système	37
5.1	Généralités	37
5.2	Fonctionnement	37
5.3	L'architecture du système	39
a	Le dossier <i>src</i>	39
b	Le dossier <i>models</i>	40
c	Le dossier <i>utils</i>	40
5.4	Fonctionnalités	41
a	Le mode couleur	41
b	Le mode distance	41
c	Le mode de détection centrale	42
d	Le mode de détection verticale	43
e	Le mode de détection par cadrants	44
f	Les fonctionnalités annexes	45

5.5	Tests utilisateurs	45
a	Les tests #1 et #2	46
b	Le test #3	46
c	Les tests #4 et #5	47
d	Synthèse	47
5.6	Les difficultés rencontrées	48
Conclusion		49
Annexes		50
Références documentaires		52

REMERCIEMENTS

J'aimerais exprimer ma gratitude envers les personnes sans lesquelles ce travail n'aurait pu prendre forme. Tout d'abord, je tiens à remercier le Docteur Guido BOLOGNA pour son savoir et son soutien inestimable dans le domaine de l'intelligence artificielle, qui ont contribué à la réalisation de ce travail. Je souhaite également exprimer ma reconnaissance envers ma famille et mes amis proches, qui m'ont soutenu et encouragé tout au long de mes études et de mes différents projets.

Un grand merci à Ruben ANDRÉ RAMOS, Maria del Mar ORDOÑEZ MOLANO, Fabian TROLLER, Xavier PERRET et Inès MAYA pour leur précieuse contribution en participant aux essais utilisateurs. Leurs retours et leurs suggestions ont permis d'améliorer significativement le prototype. Je tiens également à remercier Lorenzo RITUCCI pour sa relecture attentive de cette thèse, qui a grandement contribué à l'amélioration de sa qualité.

Enfin, je souhaite citer les noms des Messieurs GRABAU, BIRNER, TROLLER, PEIRY, PERRET, MONTAVON, SIREY et ANTONIJEVIC. Sans qui le cours des événements n'aurait probablement pas abouti au même résultat.

**INTEGRATION DE MODELES CONNEXIONNISTES DANS UN
PROTOTYPE DE SUBSTITUTION VISUELLE****ORIENTATION : INFORMATIQUE LOGICIELLE****Descriptif :**

À l'échelle mondiale, parmi les 8 milliards de personnes, on estime approximativement que 50 millions sont aveugles. Dans ce projet, nous désirons augmenter les facultés perceptives des personnes aveugles. Nous proposons un prototype d'aide à la mobilité qui encode des pixels colorés par des sons d'instruments de musique spatialisés, afin de représenter et de souligner la couleur et l'emplacement des entités visuelles dans l'environnement proche. La couleur n'étant pas toujours suffisante pour identifier un objet, nous proposons un module cognitif doté de modèles connexionnistes profonds. L'identification des objets sera transmise par synthèse vocale. Ces cinq dernières années des progrès substantiels ont été accomplis avec les architectures neuronales profondes. Une fois entraînées, elles sont en mesure de reconnaître avec une grande précision un nombre important de classes d'objets. Une des questions que l'on se pose dans ce projet est de savoir si en utilisant un prototype de substitution visuelle ce serait possible de bouger dans un environnement réel ou de rechercher des objets.

Travail demandé :

- Recenser les travaux associant les techniques de Machine Learning à la substitution visuelle pour les personnes aveugles.
- Comprendre les architectures neuronales profondes pour la reconnaissance d'objets.
- Evaluer plusieurs architectures neuronales, dont les Transformers et le modèle Yolo pour la reconnaissance des objets.
- Intégrer dans un prototype le codage de la couleur par les sons d'instruments de musique et l'identification des objets.
- Imaginer des expériences de substitution visuelle, les réaliser et faire une synthèse des résultats.
- Rédaction du rapport.

Candidat :

BOQUETE COSTA DAMIAN

Filière d'études : ISC

Professeur responsable :

BOLOGNA GUIDO

En collaboration avec : ---

Travail de bachelor soumis à une convention de stage en entreprise : **non**Travail de bachelor soumis à un contrat de confidentialité : **non**

RÉSUMÉ

L'intelligence artificielle est au cœur de la plupart des débats en 2023, que ce soit en son encontre ou en sa faveur. La puissance de certaines intelligences artificielles abouties, telles que Midjourney, ChatGPT entre autres, fait ressurgir les sujets les plus prometteurs sur la table de l'évolution technologique humaine. Le domaine médical constituant un thème phare, les prouesses montrées par ces intelligences artificielles (ré)introduisent la possibilité d'innover les solutions existantes. La vision étant un des cinq sens les plus importants dans le quotidien d'une personne, l'en priver crée une situation de handicap non négligeable et réduit considérablement la qualité de vie de l'individu. La cécité constitue donc un handicap majeur pour lequel nous n'avons pas encore de solution miracle. Cela étant, lesdites prouesses pourraient nous aider à étudier un moyen d'aider ces personnes. De ce fait, l'objectif de cette étude est de réaliser une comparaison entre deux modèles distincts de reconnaissance d'objets dans le but de déterminer leur efficacité respective. Cette évaluation est réalisée en vue de sélectionner le modèle le plus performant afin de l'incorporer dans le développement d'un système de compensation destiné aux individus atteints d'une déficience ou d'une absence de vision. Le logiciel intègre une caméra et un système auditif permettant de traduire les informations visuelles telles que la distance et la couleur, ainsi que la détection et l'identification d'objets, en informations sonores. Grâce à cette transformation, l'utilisateur de ce système est en mesure de construire une représentation spatiale de son environnement en se basant sur ces données sonores.



Candidat-e :

DAMIAN BOQUETE COSTA

Filière d'études : ISC

Professeur-e(s) responsable(s) :

DOCTEUR GUIDO BOLOGNA

En collaboration avec : -

Travail de bachelor soumis à une convention de stage
en entreprise : non

Travail soumis à un contrat de confidentialité : non

LISTE DE ACRONYMES

- bbox** Bounding Box. 12, 13, 25, 27, 35, 42, 43
- COCO** Common Objets in Context. 25, 26, 29, 40
- DL** Deep Learning. 7
- HEPIA** Haute école du paysage, d'ingénierie et d'architecture de Genève. 1, 40, 41
- IA** Intelligence artificielle. 3, 6, 7
- IoU** Intersection over Union. iv, x, xii, 27, 32, 36
- ML** Machine Learning. 6, 7, 23
- MLP** Multi-Layer Perceptron. 17, 19
- OpenCV** Open Computer Vision. 24
- ReLU** Rectified Linear Unit. 11
- RNC** Réseaux de neurones convolutionnels. x, 8, 10, 11, 13, 15, 26
- RNP** Réseaux de neurones profonds. 6
- SSD** Single Shot MultiBox Detector. ix, x, xii, 13, 26
- SSDLite** Single Shot MultiBox Detector (SSD) Lite 320 MobileNet V3. xii, 23, 26, 27, 31, 32, 33, 34, 35, 36, 38, 40, 45, 46
- ViT** Vision Transformers. iii, x, 8, 14, 15, 16, 17, 26
- YOLO** You Only Look Once. x, 4, 5, 12, 13
- YOLOS-Tiny** You Only Look at one Sequence, tiny version. xii, 23, 24, 26, 31, 32, 33, 34, 35, 36, 38, 40, 42, 43, 44, 45, 46

LISTE DES ILLUSTRATIONS

1.1 Statistiques sur les cas de cécité en Suisse	3
2.1 Structure de donnée d'une image RGB	9
2.2 Réseau de neurones	10
2.3 Couche d'agrégation	12
2.4 Couche YOLO	12
2.5 Couche SSD	13
2.6 Architectures You Only Look Once (YOLO) et SSD	13
2.7 Architecture d'un Transformer	14
2.8 Comparaisons de performances entre ViT et Réseaux de neurones convolutionnels (RNC)	15
2.9 Architecture d'un ViT	15
2.10 Segmentation d'une image en série de <i>patchs</i>	16
2.11 Architecture d'un encodeur de ViT	17
2.12 Diagramme d'un neurone artificiel	19
2.13 Différentes fonctions d'activation	20
2.14 Diagramme de la tête de détection	20
 3.1 Photo du matériel utilisé	 22
4.1 Indice de recouvrement	28
4.2 Couche d'agrégation	29
4.3 Statistiques des scores de détection	31
4.4 Statistiques des IoU	32
4.5 Statistiques des scores de confiance	33
4.6 Statistiques des temps d'exécution	34
4.7 Échantillons d'inférences	35
 5.1 Diagramme du programme	 39
5.2 Inférence en mode de détection centrale	42
5.3 Inférence en mode de détection verticale	43
5.4 Inférence en mode de détection par cadran	44
5.5 Échelle de couleurs utilisée pour le test de fiabilité	48

Références des URL

- URL01 ucba.ch/fileadmin/pdfs/Forschung/Forschung_FR/Forschungsberichte/UCBA_-_Cecite_etc_-_Evolutions_en_Suisse_-_Calculs_2019._docx.pdf
- URL02 e2eml.school/convert_rgb_to_grayscale.html
- URL03 miro.medium.com/v2/resize:fit:1946/1*au63ByDxkZYaRZpLdcfK1Q.png
- URL04 indoml.files.wordpress.com/2018/03/pooling-layer3.png?w=624&h=185
- URL05 pjreddie.com/darknet/yolo/
- URL06 arxiv.org/abs/1512.02325
- URL07 arxiv.org/abs/1706.03762
- URL08 arxiv.org/abs/2010.11929

- URL09 [researchgate.net/figure/Common-activation-functions-in-artificial-neural-networks-NNs-that-introduce_fig7_341310767](https://www.researchgate.net/figure/Common-activation-functions-in-artificial-neural-networks-NNs-that-introduce_fig7_341310767)
- URL10 pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection
- URL11 [hcocodataset.org/#download](https://cocodataset.org/#download)

LISTE DES TABLEAUX

2.1	Matrices de convolution	11
3.1	Spécificités de l'ordinateur utilisé	21
3.2	Caractéristiques de la caméra Intel® RealSense™ D455	22
4.1	Spécificités du modèle You Only Look at one Sequence, tiny version (YOLOSTiny)	26
4.2	Spécificités du modèle SSD Lite 320 MobileNet V3 (SSDLite)	27
4.3	Comparaison des scores de détection	31
4.4	Comparaison des scores d'IoU	32
4.5	Comparaison des scores de confiance	33
4.6	Comparaison des temps d'exécution	34
5.1	Option d'exécution de programme	38

Références des URL

- URL01 www.lenovo.com/ch/en/p/laptops/thinkbook/thinkbook-series/lenovo-thinkbook-15-iml/xxtbxm8001
- URL02 www.intelrealsense.com/depth-camera-d455/
- URL03 www.pytorch.org/vision/stable/models.html
- URL04 www.arxiv.org/pdf/2106.00666

LISTE DES ANNEXES

Système de substitution visuelle	51
Programme d'essais et statistiques	51

INTRODUCTION

En rédigeant cette thèse, nous nous situons au milieu de l'année 2023, ce qui implique que de nombreux développements en matière d'intelligence artificielle sont déjà survenus. La popularité de ces outils continue de croître de manière significative, suscitant à la fois fascination et inquiétude. Leurs capacités se sont tant améliorées qu'il est difficile de ne pas être impressionné, voire effrayé. L'avancée technique dans ce domaine fait de l'intelligence artificielle un outil pertinent pour l'étude de différentes méthodes visant à résoudre les problèmes persistants qui entravent le développement de l'humanité.

L'une des entraves majeures est la cécité. Ce terme désigne l'absence totale d'un des cinq sens humains, la vision. Bien que la médecine moderne ait fait des progrès considérables, ce handicap reste incurable. Les personnes atteintes de cécité subissent des conséquences considérables sur leur quotidien, leur habileté à mener des activités quotidiennes étant entravées par leur incapacité visuelle. Par conséquent, ce qui peut sembler être une tâche simple pour une personne dotée d'une vision saine peut être difficile, voire impossible pour une personne souffrant de cécité. Ce sujet de travail de Bachelor, émis par le Dr G. Bologna dans le cadre de mes études à la [Haute école du paysage, d'ingénierie et d'architecture de Genève \(HEPIA\)](#), se focalise sur l'application de l'intelligence artificielle en vue d'améliorer la qualité de vie des individus présentant une déficience visuelle partielle ou totale. À cet effet, un système de substitution visuelle a été développé afin d'analyser sa pertinence et sa viabilité en tant que solution pour accomplir cette tâche spécifique.

Ce travail adopte une méthodologie formée de plusieurs étapes. Dans un premier temps, une analyse approfondie des projets de recherche existants et des produits commerciaux répondant à la problématique spécifique a été effectuée dans le but de mieux comprendre l'état de l'art et d'établir des critères d'acceptation pertinents. Par la suite, une recherche a été menée pour répondre à diverses questions concernant le fonctionnement des différents modèles de détection d'objets pré-entraînés. Cette recherche s'est appuyée sur l'exploration de la littérature existante sur ce sujet. Suite à cela, une phase de sélection des technologies a été réalisée afin de choisir les outils répondant aux exigences d'un tel système, impliquant l'évaluation de différentes bibliothèques logicielles et technologies spécifiques, en incluant la mise en place d'un environnement spécifique pouvant les accueillir. Des mesures de performances ont été effectuées pour

évaluer divers aspects des deux types de modèles étudiés. Les statistiques obtenues à partir de cette étude sont ensuite interprétées, afin d'en extraire des informations pertinentes. Finalement, les phases de développement et de tests commencent par une conception schématique simple du fonctionnement du programme et de son développement itératif. Lors d'ajout de fonctionnalités, de correction de bogues ou de changements de comportements majeurs, le système est soumis à des tests utilisateurs.

La structure de cette thèse est organisée en plusieurs parties. Dans un premier temps, nous procéderons à une analyse approfondie de la problématique, de l'analyse de l'existant ainsi qu'à l'exposition des concepts fondamentaux associés. Par la suite, nous aborderons les choix technologiques en les décrivant et leur sélection sera justifiée. Ensuite, nous présenterons les résultats de nos recherches et les statistiques relatives aux différents types de modèles de reconnaissance d'objets. Cette thèse se poursuivra en exposant la partie conceptuelle et fonctionnelle du système, suivi de la présentation des divers retours utilisateurs. Enfin, une conclusion offrira une réflexion critique sur le travail accompli, ainsi qu'une exploration des pistes d'amélioration à envisager pour l'avenir.

CHAPITRE 1 : ANALYSE DE LA PROBLÉMATIQUE, DE L'EXISTANT ET INTRODUCTION AUX CONCEPTS GÉNÉRAUX

1.1. LA CÉCITÉ

La cécité est une absence de vision totale¹ qui touche un grand nombre de personnes à travers le monde. Selon une étude suisse datant de 2019, le nombre de personnes atteintes de cécité est estimé à environ 377'000 et devrait continuer d'augmenter au fil des années.



ILLUSTRATION 1.1 – Nombre de personnes malvoyantes, aveugles ou sourdaveugles en Suisse.
Source : tiré de UCBA 2019, p. 1 ref. URL01

L'absence de vision entraîne des conséquences significatives dans la vie quotidienne, obligeant les personnes atteintes à adopter un mode de vie différent de celui des individus voyants. Par exemple, lorsqu'il s'agit de chercher un objet quelconque à travers différentes pièces de leur domicile, un individu aveugle peut éprouver des difficultés en raison de son handicap. Tandis qu'une personne voyante vivant seule n'éprouvera pas de difficulté à localiser l'objet en question, dans la mesure où celui-ci se trouve à un emplacement ne sortant pas de l'ordinaire. Ce simple exemple met en évidence la pertinence de ce projet. Notre objectif consiste à explorer les aptitudes des différents algorithmes du domaine de la vision numérique assistée par Intelligence artificielle (IA) pour surmonter les obstacles liés à la recherche d'objets du quotidien dans un environnement restreint, en vue d'aider les personnes souffrant de cécité.

1. *Cécité*, 2023.

1.2. LES PROJETS SIMILAIRES

La problématique présentée précédemment n'est pas nouvelle. De ce fait, de nombreuses personnes et institutions se sont déjà penchées sur ce domaine dans le but d'apporter des alternatives et/ou des solutions plus ou moins viables et originales. L'analyse de différents travaux traitant du même sujet nous permettrait de consolider notre vision concernant les objectifs à atteindre et les questions méritant une réponse. Ainsi, certains travaux et produits commercialisés ont été sélectionnés afin d'en extraire des informations pertinentes.

a. *Be My Eyes*

Derrière ce nom se cache une application² gratuite disponible sur smartphone. Cette application prétend pouvoir aider les personnes atteintes de déficience visuelle dans diverses tâches quotidiennes. Ceci est réalisé en utilisant ladite application pour prendre une photo de ce que l'on souhaite analyser. Ensuite, l'application se charge d'analyser la photo et de générer une description de son contenu, sous forme de texte émis par une voix synthétique.

b. *TapTapSee*

Le concept de ce produit³ s'apparente sensiblement à celui de Be My Eyes. Il s'agit également d'une application mobile utilisant la caméra du smartphone pour analyser la scène se présentant devant l'objectif. L'utilisateur déclenche la détection d'objets en effectuant deux tapotements succincts sur l'écran. L'application se charge ensuite de générer une description de l'image grâce à une voix de synthèse.

c. *Real-Time CV Based Autonomous Navigation System*

Ce projet⁴ présente le prototype d'un système pouvant être similaire au nôtre. Il se compose d'une caméra et d'un appareil auditif connectés à une unité de calcul capable de détecter les obstacles se dressant dans le champ de vision. On y trouve une comparaison des performances entre les modèles de détection d'objets nommés **YOLO** et **SSD MobileNet V2** sous différents angles.

2. *Be My Eyes - See the world together*, 2012.

3. *TapTapSee - Blind and Visually Impaired Assistive Technology*, 2017.

4. Hasan et al., 2022.

d. Prototype de substitution de la vision par le toucher

Apparu dans l'émission *Underscore_*, ce projet⁵ développé par une entreprise française vise à remplacer la vision par le toucher grâce à une matrice de picots placés sur le bas du dos de l'utilisateur. La matrice serait en mesure de retranscrire une image en deux dimensions en actionnant les mouvements des picots, envoyant des signaux interprétés par le sens du toucher de l'utilisateur. Lors de cette émission, il est annoncé qu'une première version sera commercialisée d'ici fin 2023 ou début 2024, au prix de 3000 euros.

e. Synthèse

Tous ces projets présentent des qualités en termes de fonctionnalités et adoptent des approches spécifiques pour résoudre le problème. À la suite de ces présentations, nous pouvons conclure qu'il serait intéressant d'examiner les aspects suivants :

e.1. Le modèle de détection

Parmi les projets qui divulguent ouvertement les modèles de détection d'objets utilisés, on retrouve **YOLO** sous sa troisième version. Ce modèle est réputé comme l'un des meilleurs dans son domaine. Cependant, cette version spécifique commence à prendre de l'âge et de nouvelles versions et variantes sont disponibles. Par conséquent, il serait intéressant d'explorer la gamme de possibilités offertes par ces avancées et d'observer les éventuelles améliorations qu'elles proposent.

e.2. Les fonctionnalités

Les projets qui utilisent des captures d'écran pour décrire la scène représentée démontrent leur efficacité en termes de transmission d'informations. Pour notre vision du projet, il semble plus pertinent d'opter pour un système fonctionnant en temps réel, afin de permettre une recherche active d'objets, plutôt que de se baser sur des instantanés potentiellement plus coûteux à traiter. De plus, en offrant plusieurs sources d'informations telles que la distance et la couleur de lors d'une détection d'objet dans le champ de vision, nous pourrions compléter la palette de possibilités pour l'utilisateur.

5. *Underscore_*, 2023.

D'autres aspects moins influents mais néanmoins pertinents à prendre en considération pour le développement futur du projet incluent la minimisation des coûts et la réduction de la taille physique d'un tel système. Les personnes atteintes de déficience visuelle ne constituent pas un public cible défini par leurs préférences, mais plutôt par les contraintes imposées par leur handicap. Par conséquent, il est éthiquement juste de proposer une solution à moindre coût.

En ce qui concerne les dimensions physique du prototype, il est possible qu'il se présente sous une forme brute avec une ergonomie améliorable. Ainsi, réduire sa taille en minimisant au maximum l'impact sur les performances semble être un point important à aborder pour assurer la continuité du projet.

1.3. L'INTELLIGENCE ARTIFICIELLE

a. Généralités

Afin de s'approcher au mieux d'un résultat exploitable et convaincant, ce travail se base sur un domaine, aujourd'hui largement exploité, de l'informatique : L'**IA**. Celui-ci regroupe un ensemble de techniques visant à simuler des comportements semblables à ceux d'êtres vivants dotés d'intelligence sur plusieurs axes⁶. Le premier de ces axes est celui de l'apprentissage. L'**IA** doit être capable, à partir de données et de tests, de s'adapter et de s'auto-corriger en fonction des informations qu'elle reçoit. Le deuxième axe est celui du raisonnement. Grâce à son apprentissage, l'**IA** peut agir selon la logique qui lui est inculquée et adapter ses prises de décision en conséquence. Le troisième axe est sa capacité à résoudre des problèmes. L'**IA** est de plus en plus exploitée en raison de son habileté à trouver des solutions efficaces, là où l'être humain peut rencontrer des difficultés ou échouer en raison de diverses contraintes.

En ce qui concerne le raisonnement d'une **IA**, un des points d'intérêt est sa perception. Les êtres humains utilisent leurs sens pour scanner leur environnement et en extraire des informations sur ce qui les entoure. Certaines **IA**, telles que les Réseaux de neurones profonds (RNP), sont spécialement conçues pour reproduire cette capacité d'analyse perceptive, leur permettant de détecter et de classifier les informations reçues. Ce point nous rapproche d'un des concepts incontournable de l'**IA** : le Machine Learning (ML).

6. *Artificial intelligence | Definition, Examples, Types, Applications, Companies, & Facts*, 2023.

b. Le Machine Learning

Selon IBM⁷, "le **ML** est une des branches de l'**IA** et des sciences informatiques qui se focalise sur l'utilisation de données et d'algorithmes dans le but d'imiter l'apprentissage graduel de l'humain, améliorant sa précision". Ce concept nous intéresse, car celui-ci nous introduit un autre concept plus précis concernant les outils utilisés dans cette thèse : le **Deep Learning (DL)**.

c. Le Deep Learning

Le **DL** est le dernier point de la hiérarchie quant aux concepts abstraits de l'**IA** abordée dans cette thèse. Celui-ci reprend la définition de **ML** par héritage en ajoutant une précision sur celle-ci. Le **DL**, selon IBM⁸, englobe les réseaux de neurones constitués de trois couches ou plus (justifiant l'appellation "deep", profond en anglais). Ceux-ci reprennent donc le principe d'apprentissage calqué sur l'humain en précisant la taille minimale d'un modèle. Cette propriété a comme avantage de pouvoir exploiter des jeux de données de grande taille lors de la phase d'apprentissage. De ce fait, les modèles décrits par le **DL** sont plus adaptés aux cas d'utilisation nécessitant la capacité d'identification et de différenciation d'objets plus variés.

Dans le contexte de cette étude, il est pertinent de considérer la compétence mentionnée précédemment comme l'avantage le plus approprié et pertinent pour proposer une méthode offrant une solution relativement efficace à la problématique présentée. Ainsi, l'exploration des outils et des concepts offerts par le domaine du **DL** constitue une étape logique permettant de progresser vers une solution concrète potentielle.

7. *What is Machine Learning ?, [s. d.]*

8. *What is Deep Learning ?, [s. d.]*

CHAPITRE 2 : APPROFONDISSEMENT DES CONCEPTS

2.1. GÉNÉRALITÉS

Afin de guider ce projet dans la bonne direction, il est impératif d'explorer et de comprendre les différents concepts impliqués dans la construction d'un modèle complexe capable de localiser et de distinguer des objets dans un espace limité. À cette fin, plusieurs algorithmes sont disponibles pour atteindre de telles capacités, parmi lesquels la famille des **RNC** est la plus populaire. Ces modèles ont largement contribué à l'avancement de l'apprentissage profond.

Cependant, l'article intitulé *An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale*⁹, publié en 2020, explique qu'il est possible d'utiliser un autre type d'algorithme pour obtenir des résultats similaires, voire supérieurs. Ces algorithmes sont connus sous le nom de **ViT**. Ce type de modèle est une adaptation de l'architecture originale des *Transformers* (introduite en section 2.5), qui était initialement principalement utilisée dans le domaine du traitement automatique du langage naturel. Ce domaine englobe des tâches telles que la génération de texte ou la traduction d'un texte vers une langue spécifique.

En prenant cela en considération, il est pertinent d'entreprendre une étude approfondie et comparative de ces deux algorithmes en sélectionnant un modèle représentatif de chaque. L'objectif est de déterminer lequel serait le plus approprié en tant que système de substitution visuelle pour notre programme.

9. Dosovitskiy et al., 2021.

2.2. LES IMAGES

Ce chapitre nécessite une introduction à un élément fondamental sans lequel ce travail perdrait tout son sens : les images. Dans le domaine de la vision numérique, les images jouent un rôle central, car elles constituent les données sur lesquelles les modèles d'apprentissage se "nourrissent". En d'autres termes, les images sont l'équivalent du carburant pour une voiture. De ce fait, sans elles, un modèle de détection d'objets n'a aucune utilité pratique.

D'un point de vue technique, une image est représentée sous la forme d'une matrice à deux ou trois dimensions. Si l'image est en nuances de gris, elle est représentée par une matrice à deux dimensions. Si l'image contient des informations de couleur, elle est représentée par une matrice à trois dimensions, comportant des canaux pour le rouge, le vert et le bleu.

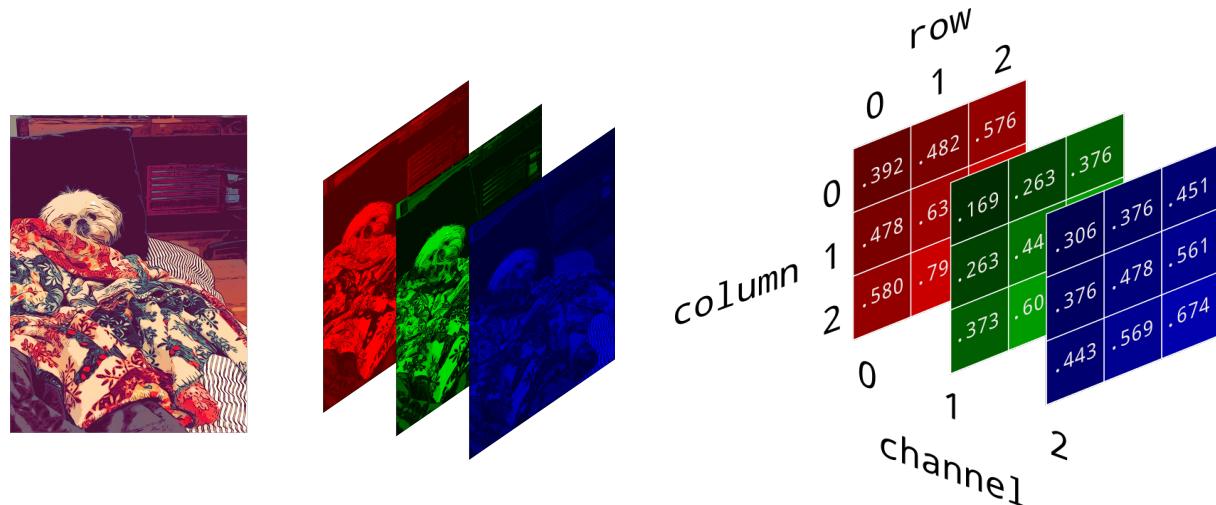


ILLUSTRATION 2.1 – Représentation de la structure de donnée correspondant à une image de couleur. Source : tiré de www.e2eml.school, ref. URL02.

Une connaissance approfondie de l'anatomie d'une image est importante pour comprendre les différentes opérations auxquelles elles sont constamment soumises. Cela permet une meilleure visualisation du fonctionnement du processus de détection et de reconnaissance d'objets.

2.3. LES RÉSEAUX DE NEURONES CONVOLUTIFS

Les **RNC** sont un type de modèle d'apprentissage profond communément utilisé dans le domaine de la vision numérique.

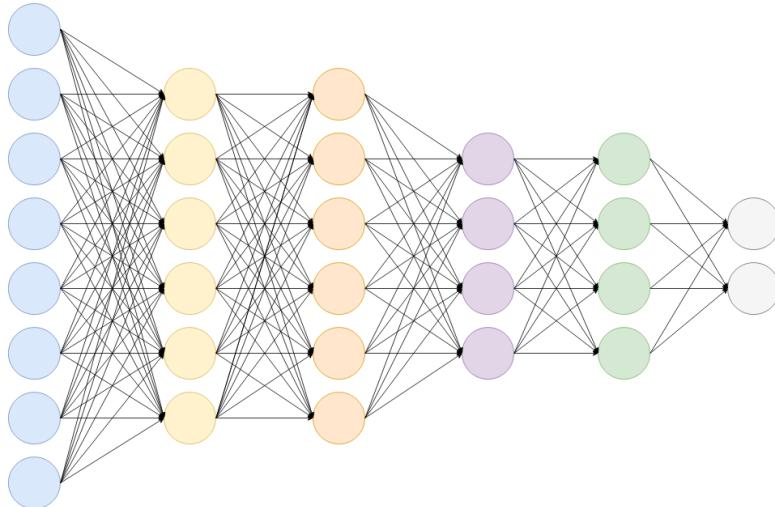


ILLUSTRATION 2.2 – Représentation simplifiée d'un réseau de neurones. Source : tiré de miro.medium.com ref. URL03.

En interne, les **RNC** sont une interconnexion de couches de neurones artificiels capable de traiter des informations reçues pour mettre en évidence certaines propriétés de ladite information par le biais d'opérations mathématiques. Les multiples couches d'un **RNC** jouent toutes un rôle précis quant au traitement de l'information et génèrent des sorties pouvant être exploitées par la couche suivante. Cet enchaînement d'opérations nous permet de partir d'une image et de nous retrouver avec un résultat sous forme de classification ou de détection.

2.4. ANATOMIE D'UN RÉSEAU DE NEURONES CONVOLUTIFS

Comme énoncé précédemment, il existe de multiples couches d'interconnexion de neurones capables de réaliser une opération mathématique précise sur l'ensemble des données d'entrée. Ces couches permettent d'obtenir différents résultats, qui peuvent être transférés à la couche suivante. Voici une liste non exhaustive de couches existantes dans un **RNC**.

a. La couche convulsive

Cette couche est la pierre angulaire d'un RNC. Elle applique un ensemble de filtres de convolution à l'image d'entrée pour extraire des caractéristiques importantes. Les filtres sont des matrices qui sont appliquées à des régions de l'image d'entrée, produisant ainsi une matrice de sortie. Il existe différents types de noyaux de convolutions, tels que le *flou moyen*, *flou Gaussien*, *Laplacien* et *Sharpen* entre autres. Voici une représentation plus visuelle de ces noyaux :

	Identité	Flou moyen	Flou Gaussien	Laplacien	Sharpen
Matrices	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
Résultats					

TABLEAU 2.1 – Matrices de convolution et leurs effets. Source : tiré du cours des réseaux convolutifs, Dr. Guido Bologna 2022, p. 13 à 21

Cette couche est souvent accompagnée de la fonction Rectified Linear Unit (ReLU), celle-ci permet de corriger la sortie de la couche convulsive en remplaçant toute valeur négative par zéro. Toute valeur positive ou égale à zéro restera tel quel.

$$ReLU(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

b. La couche d'agrégation

La couche d'agrégation, (aussi appelée couche de *pooling* en anglais) a comme objectif de réduire la taille d'une matrice donnée afin d'obtenir une généralisation de la matrice de départ. Elle accompagne généralement les couches de convolution d'un modèle, car elle permet de réduire le nombre d'opérations tout en gardant les caractéristiques importantes de l'entrée, libérant ainsi la puissance de calcul de la machine.

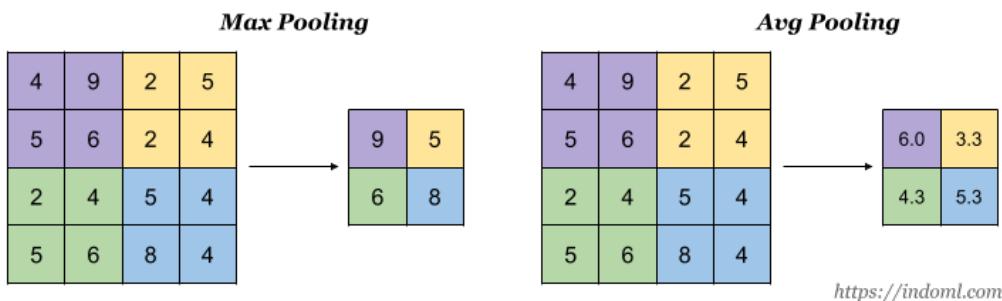


ILLUSTRATION 2.3 – Représentation de l'opération d'agrégation. Source : tiré de www.indoml.com ref. URL04.

Il existe plusieurs types d'opérations d'agrégation, telle que la moyenne, obtenant la valeur moyenne d'une zone ou encore la maximale, permettant de ressortir les valeurs maximales d'une zone de la matrice.

c. La couche de détection d'objets

Cette couche, pouvant être un ensemble de plusieurs couches, est conçue pour prédire les cadres englobant un objet (appelées Bounding Box (bbox)) ainsi que les scores de confiance correspondants. Elle remplace la couche entièrement connectée, présente dans les modèles de classification d'images, pour palier à la haute utilisation de mémoire et de puissance de calcul générées par celles-ci.

Il existe plusieurs types de couches différentes. **YOLO** est un modèle de détection d'objets à part entière utilisant son propre type de couche de détection. Celui-ci base son fonctionnement¹⁰ sur la segmentation de l'image donnée en régions de taille définie. Sur chacune de ces régions va se produire une prédiction par génération des bbox accompagnée du taux de confiance, jouant un rôle décisif quant au résultat final.

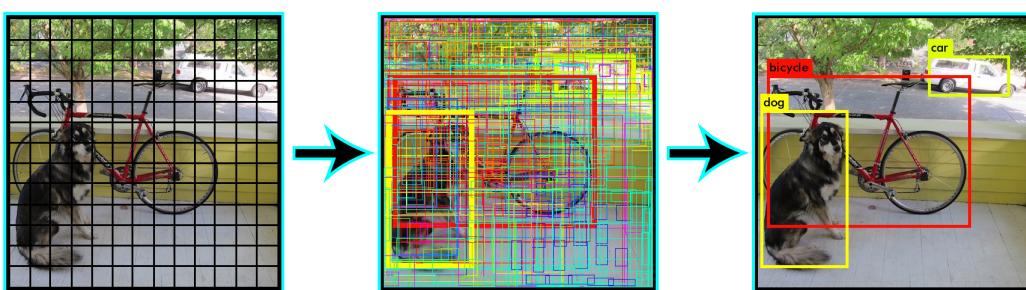


ILLUSTRATION 2.4 – Représentation du fonctionnement de **YOLO**. Source : tiré de www.pjreddie.com ref. URL05.

10. Redmon et al., 2018.

Un autre type de couche concerne le **SSD**. Celui-ci utilise des filtres de convolution de différentes tailles pour générer une carte de caractéristiques (généralement appelée *feature map* en anglais). Celle-ci est ensuite utilisée prédire des **bbox** ainsi que leur score de confiance pour chaque classe d'objets¹¹.

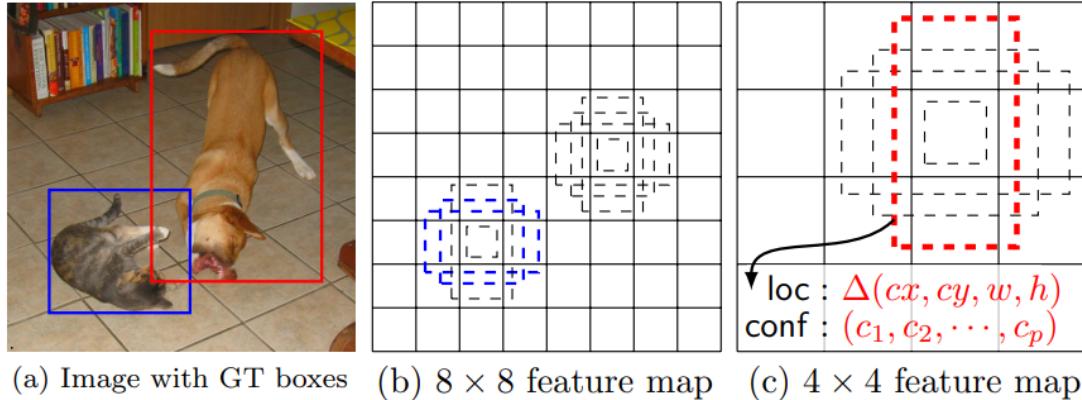


ILLUSTRATION 2.5 – Représentation du fonctionnement du **SSD**. Source : tiré de www.arxiv.org ref. URL06.

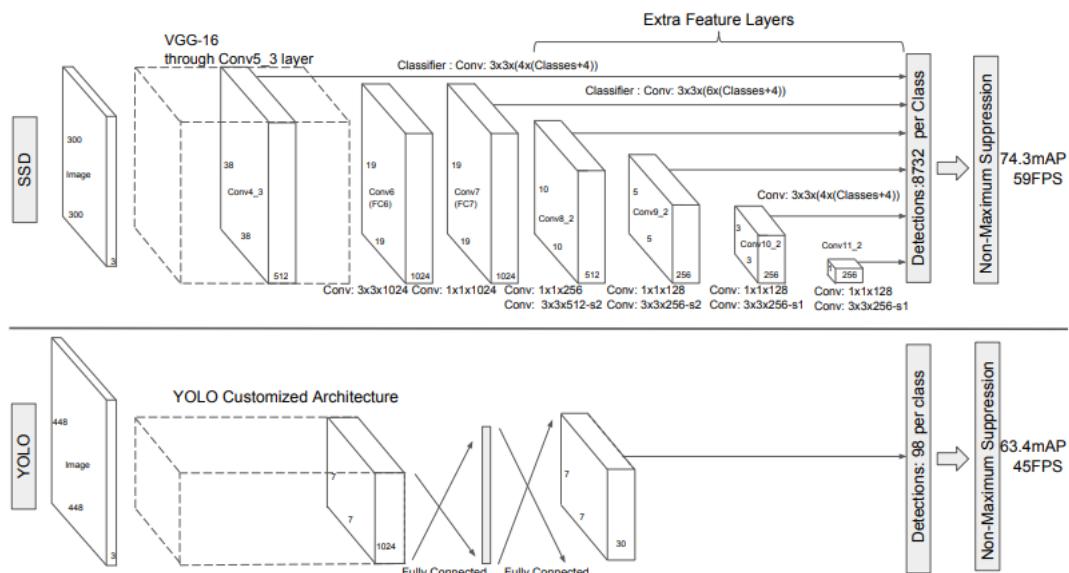


ILLUSTRATION 2.6 – Architecture des modèles **YOLO** et **SSD**. Source : tiré de www.arxiv.org ref. URL06.

En utilisant ces composants, entre autres non cités ici pour raison de concision, il est possible de construire des **RNC** capables de prendre une image en entrée et d'y générer des **bbox** en spécifiant un taux de certitude.

11. Liu et al., 2016.

2.5. LES TRANSFORMERS

Un Transformer est un modèle d'apprentissage profond apparu pour la première fois dans l'article datant de 2017 nommé *Attention is all you need*¹². Dans celui-ci, il est expliqué qu'un modèle basant son architecture sur deux composants, un encodeur et un décodeur, est capable d'atteindre des performances supérieures à celles des modèles considérés comme représentants de l'état de l'art dans le domaine du traitement automatique du langage naturel.

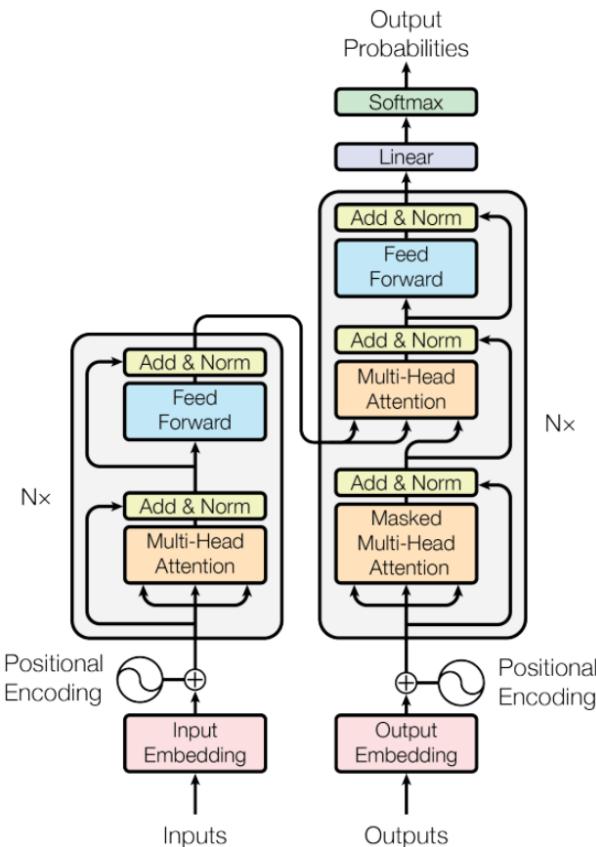


ILLUSTRATION 2.7 – Représentation schématisée d'un Transformer. Source : tiré de www.arxiv.org ref. URL07.

2.6. LES ViT

Une variante du modèle précédemment mentionné a été spécifiquement développée pour le domaine de la vision par ordinateur, visant à accomplir des tâches telles que la détection et la classification d'objets. Cette variante, appelée *Vision Transformer*, est basée sur le modèle Transformer et a suscité un intérêt considérable. Dans l'article présentant les ViT¹³, les auteurs ont affirmé que le ViT surpassait les modèles de pointe à cette époque.

12. Vaswani et al., 2017.

13. Dosovitskiy et al., 2021.

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.55 ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	90.72 ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	99.50 ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	—
CIFAR-100	94.55 ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	—
Oxford-IIIT Pets	97.56 ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	—
Oxford Flowers-102	99.68 ± 0.02	99.74 ± 0.00	99.61 ± 0.02	99.63 ± 0.03	—
VTAB (19 tasks)	77.63 ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

ILLUSTRATION 2.8 – Tableau des comparaisons entre les ViT (ViT-XXX) et les RNC (ResNet et EfficientNet). Source : tiré de www.arxiv.org ref. URL08.

Par conséquent, des expérimentations ultérieures ont été menées pour évaluer les performances du ViT et d'un RNC dans notre cas d'utilisation spécifique. L'objectif principal de ces expériences est de déterminer quel type de modèle est le plus adapté à notre problématique.

2.7. ANATOMIE D'UN ViT

Le ViT, en tant l'alternative basée sur le modèle Transformer, est composé de plusieurs composants interconnectés qui traitent l'information et génèrent des prédictions. Ce modèle utilise également un mécanisme d'attention similaire à celui observé chez des êtres vivants.

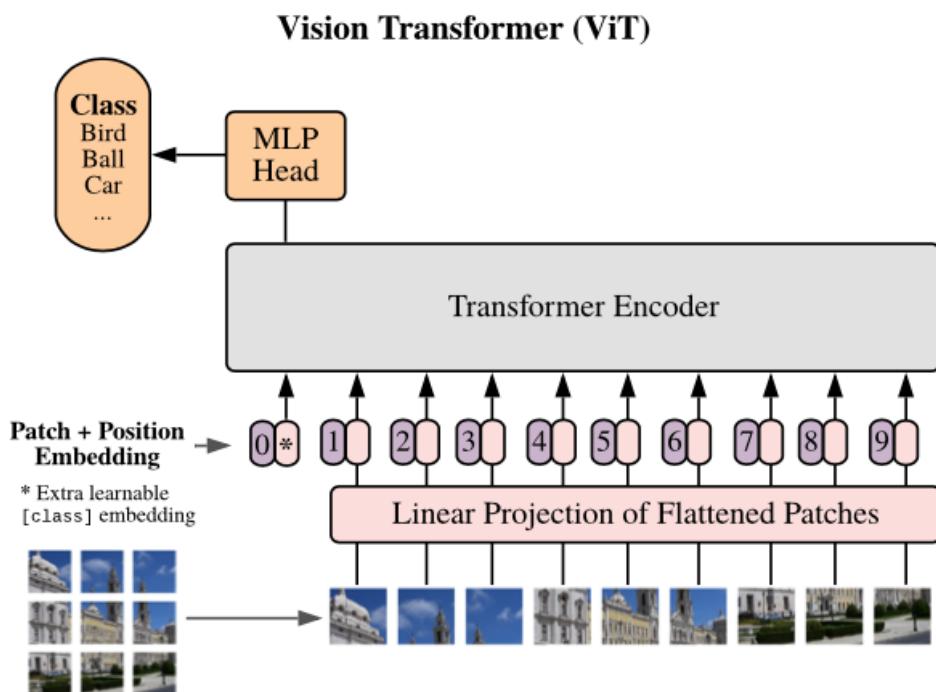


ILLUSTRATION 2.9 – Architecture d'un ViT. Source : tiré de www.arxiv.org ref. URL08.

a. Pré-traitement des données en entrée

Avant de pouvoir utiliser les compétences du modèle, il est nécessaire de modeler les images à traiter afin que celles-ci soient exploitables par le modèle en question. De ce fait, cette phase commence par diviser l'image en une série de sous-images (appelés *patches*) de dimension $patch_x \times patch_y$.



ILLUSTRATION 2.10 – Segmentation d'une image en série de *patches*. Source : tiré de www.arxiv.org ref. URL08.

Ceux-ci sont ensuite tous passés à travers une fonction de projection linéaire qui s'occupe "d'aplatir" les *patches*. C'est-à-dire, de passer des dimensions $patch_x \times patch_y$ à un vecteur de dimension $1 \times (patch_x \times patch_y)$.

Finalement, ces vecteurs sont conjoints d'un indice permettant d'indiquer la position du *patch* dans l'image originale. Cet indice, appelé indice de position, est ajouté dans le but de fournir une information complémentaire qui s'avère utile lors de l'inférence. Il permet notamment d'encoder la localisation spatiale des différents *patches* dans l'image, ce qui peut aider le modèle à comprendre la structure et la relation entre les différents éléments visuels.

b. L'encodeur

Ce composant constitue la plus grosse partie du fonctionnement du **ViT**. Il a pour but d'extraire des informations à partir des données pré-traitées précédemment. Cette extraction désigne une tache de capture de caractéristiques visuelles significatives d'une image donnée. Cette tâche est opérée par le mécanisme d'attention, présenté dans la sous-section suivante. Ce composant fonctionne en utilisant plusieurs couches identiques empilées les unes sur les autres (noté $L \times$ sur la figure 2.11).

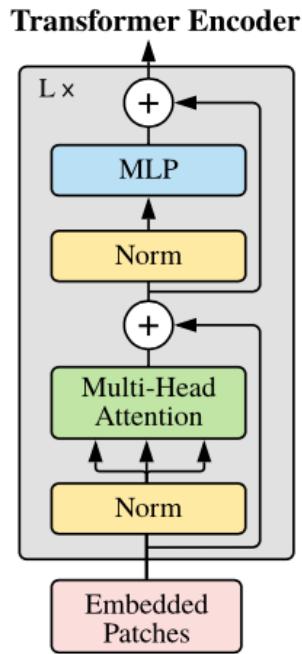


ILLUSTRATION 2.11 – Architecture d'un encodeur de ViT. Source : tiré de www.arxiv.org ref. URL08.

Son flux d'exécution commence par un module qui applique une normalisation aux vecteurs passés en entrée pour ensuite être transmis au mécanisme d'attention (sous-section c). Une fois le processus d'attention accompli, les sorties générées par cette étape sont additionnées par les entrées originales non normalisées. Cette addition garantit la cohérence de l'information traitée, car les données subissent des transformations significatives pouvant les déformer. Grâce à celle-ci, il nous est possible de s'assurer que le modèle se "souvienne" de ce qu'il apprend.

Après l'étape d'addition des sorties d'attention aux entrées, les vecteurs résultants sont soumis à une nouvelle normalisation, puis dirigés vers le module Multi-Layer Perceptron (MLP). Ce module consiste en une interconnexion de neurones, similaire à ce qui est illustré dans la figure 2.2. Celui-ci introduit une étape de transformation qui permet au modèle de capturer des motifs et des relations entre les caractéristiques extraites. Cette transformation vise à adapter les représentations des vecteurs afin de mieux représenter les caractéristiques de l'image, ce qui facilite la tâche de classification. Finalement, une dernière addition est effectuée entre les données produites par le MLP et celles de l'addition performée avant les deux dernières étapes mentionnées. Ces données finales sont ensuite relayées à la tête de détection (sous-section d), dernier composant du ViT.

c. Le mécanisme d'attention

Le mécanisme d'attention est un élément clé du fonctionnement de l'encodeur. Il permet de se focaliser sur les parties pertinentes d'une image lors de l'identification d'un objet. L'attention permet ainsi de détecter les zones d'intérêt en attribuant des pondérations à tous les *patchs* prédefinis de l'image traitée.

En interne, le mécanisme d'attention reçoit chaque *patchs* pré-traités et sous la forme de vecteurs de dimension $1 \times (patch_x \times patch_y)$. Ces vecteurs sont ensuite projetés trois fois à travers le mécanisme d'attention. La première projection transforme les *patchs* en une représentation vectorielle appelée Q (signifiant *query* ou requête en français). La deuxième projection les transforme en une représentation vectorielle dans l'espace K (signifiant *key* ou clé), tandis que la troisième projection les transforme en une représentation vectorielle dans l'espace V (signifiant *value* ou valeur).

Ces projections distinctes permettent de calculer le score d'attention. Pour ce faire, on effectue un produit scalaire entre les vecteurs de l'espace Q et K , ce qui génère une matrice de scores d'attention. En effectuant ce produit entre les vecteurs de ces deux espaces, on mesure donc la similarité entre les *patchs*. Ces scores sont ensuite normalisés pour obtenir des valeurs dans l'intervalle $[0, 1]$. Ces valeurs normalisées sont ensuite utilisées pour pondérer les vecteurs et ainsi générer un résultat utilisable pour le prochain composant de l'encodeur. Pour ce faire, la matrice de scores d'attention est injecté dans un second produit scalaire avec les vecteurs de l'espace V .

d. La tête de détection

Ce dernier composant a comme objectif de pouvoir générer des prédictions sur toutes les classes d'objets connues par le modèle. Ceci est effectué grâce aux propriétés du **MLP**. Cet algorithme, succinctement présenté dans la section b, consiste en une interconnexion de *neurones artificiels*, aussi appelés *perceptrons*.

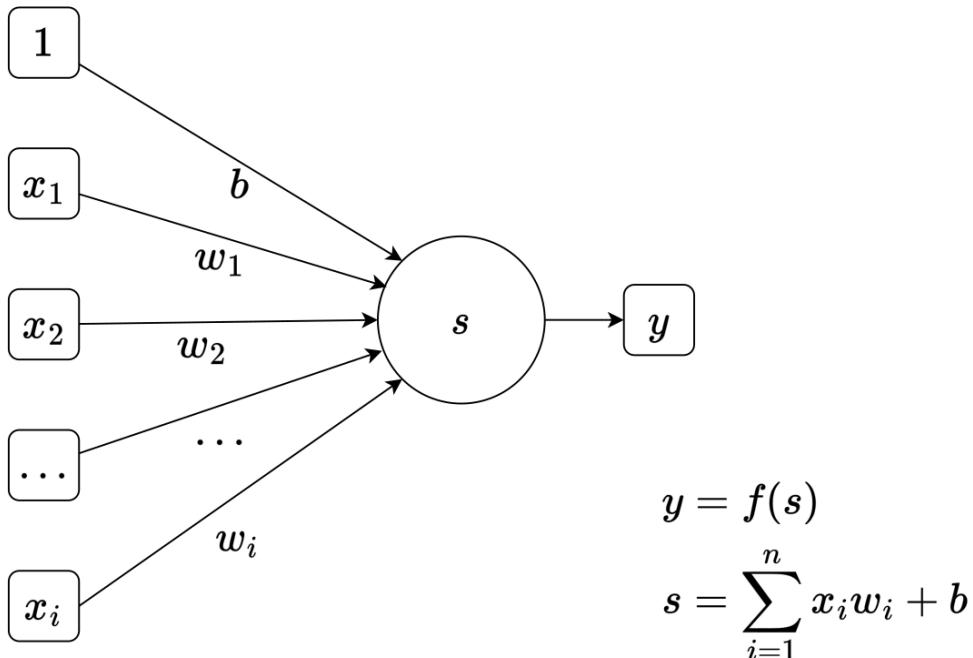


ILLUSTRATION 2.12 – Diagramme d'un neurone artificiel. Source : Damian Boquete Costa

Ces neurones reçoivent i entrées x et calculent une somme de celles-ci, chacune pondérée par un poids w . Un biais b vient compléter le calcul de la somme n . Les poids représentent l'importance d'une entrée. C'est en adaptant ceux-ci que le modèle est capable d'apprendre, il adapte ce qu'il perçoit comme pertinent d'un point de vue mathématique et abstrait pour l'humain. Le biais b ajuste le seuil d'activation du neurone, permettant ainsi de contrôler sa sensibilité et d'améliorer sa capacité à apprendre et généraliser à partir des données d'entrée n . Une somme s de chaque entrée multipliée par son poids et additionnée avec le biais est effectuée. Le résultat en découlant est finalement injecté dans une fonction d'activation $f(s)$. Cette fonction d'activation a pour but de transformer la somme s en une sortie non linéaire. Elle introduit une non-linéarité dans le neurone, ce qui lui permet de modéliser des relations entre les entrées et les sorties.

La fonction d'activation peut être de différents types :

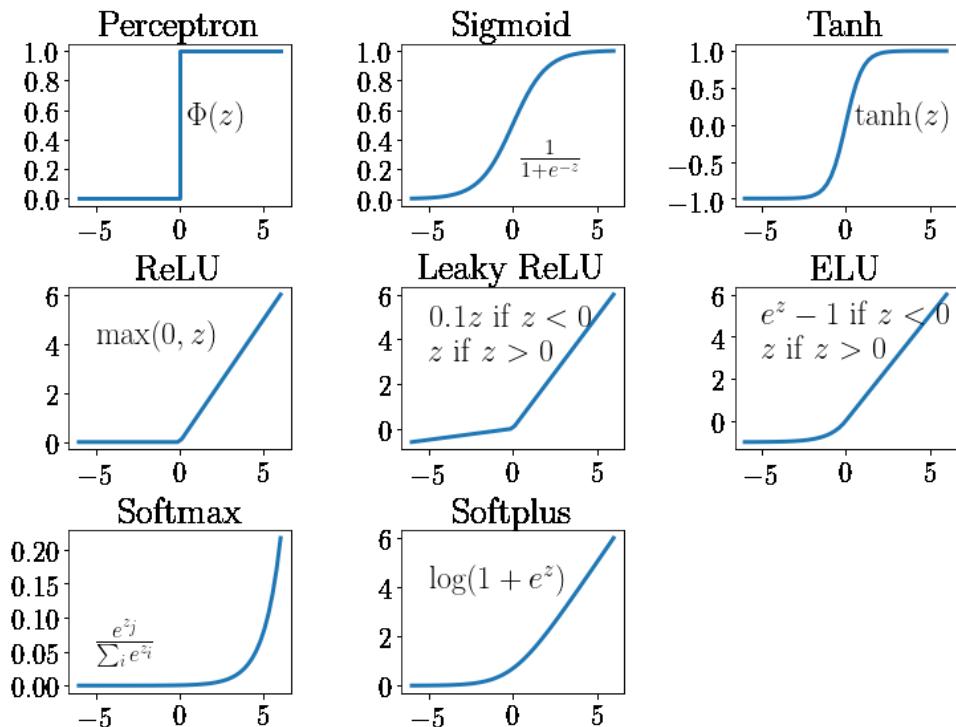


ILLUSTRATION 2.13 – Tableau non exhaustif de fonctions d'activation. Source : tiré de www.researchgate.net ref. URL09.

La tête de détection utilise les sorties générées par l'encodeur et les fait passer à travers plusieurs couches de neurones artificiels afin de produire une valeur non linéaire représentant une estimation de correspondance pour chaque classe d'objets connue.

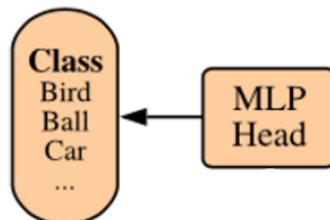


ILLUSTRATION 2.14 – Diagramme de la tête de détection. Source : tiré de www.arxiv.org ref. URL08.

CHAPITRE 3 : LES TECHNOLOGIES ET RESSOURCES UTILISÉES

3.1. GÉNÉRALITÉS

Pour créer notre système ainsi que notre programme d'essais d'un point de vue technique, il est nécessaire de définir les technologies et outils qui nous permettront d'atteindre au mieux un résultat convenable pour répondre aux besoins.

3.2. MATÉRIEL PHYSIQUE

Le développement ainsi que les essais de ce système sont effectués en faisant usages de machines et composants ayant leur propres spécificités et offrent diverses fonctionnalités ainsi que des performances qui leur sont propres, le choix de ce matériel s'est fait, principalement, en fonction de ce qui était en notre possession ou qui était mis à disposition. De ce fait, échanger ceux-ci, au stade actuel du projet, peut entraîner des performances différentes ainsi que des problèmes de compatibilité. Voici une description de ce qui a été utilisé dans le cadre de ce travail :

Lenovo ThinkBook 15 IML est l'ordinateur sur lequel le programme a été développé.

Celui-ci fonctionne avec le système d'exploitation *Linux*, sous une de ses nombreuses distributions nommée *Mint*.

Catégorie	Valeur
Système d'exploitation (OS)	Linux 5.15.0-72-generic
Distribution	Linux Mint 21.1 Vera
Processeur (CPU)	Intel Core i5-1035G1
Cadence CPU	1.00 à 3.60 GHz
Processeur graphique (GPU)	Intel UHD Graphics
Type de GPU	Intégré au CPU
Mémoire volatile (RAM)	16 GB

TABLEAU 3.1 – Spécificités du Lenovo ThinkBook 15 IML. Source : tiré de www.lenovo.com ref. URL01

Microsoft Numpad est un simple pavé numérique utilisé pour pouvoir interagir avec le système une fois démarré. Il s'avère utile, car le PC ne sert que de machine offrant une puissance de calcul. La machine n'est donc pas considérée comme un outil avec lequel l'utilisateur sera amené à interagir avec.

Intel RealSense D455 est une caméra conçue pour les applications de vision par ordinateur et de perception 3D ainsi pour la capture de profondeur et sa compatibilité avec les systèmes informatiques. Celle-ci est équipée de deux capteurs d'image et d'une technologie de vision stéréo qui lui permet de capturer des images en trois dimensions avec précision. Dans le cadre de ce projet, elle est utilisée pour recevoir le flux vidéo et calculer les distances.

Caractéristique	Valeur
Résolution de profondeur	Jusqu'à 1280x720
Résolution couleur	Jusqu'à 1280x800
Champ de vision diagonal	Supérieur à 90°
Images par secondes profondeur	jusqu'à 90 FPS
Images par secondes couleur	jusqu'à 90 FPS
Plage de distance	0,4 m à plus de 6 m

TABLEAU 3.2 – Caractéristiques de la caméra Intel® RealSense™ D455. Source : tiré de www.intelrealsense.com ref. URL02



ILLUSTRATION 3.1 – Photo du matériel utilisé, la main sert d'échelle de grandeur. Source : réalisé par Damian Boquete Costa

3.3. LANGAGE INFORMATIQUE

Le langage de programmation utilisé dans ce projet est Python¹⁴, dans sa version 3.9. Ce choix a été motivé par la simplicité de Python et le vaste catalogues bibliothèque logicielle disponible, celles-ci sont souvent développées par diverses organisations et par la communauté Python. Il est largement utilisé dans le domaine de l'apprentissage automatique (**ML**) en raison de sa popularité et de son écosystème riche.

Ce langage représente 100% des lignes de code produites pour les différents scripts liés à ce travail. Bien que Python puisse être considéré comme relativement lent par rapport à d'autres langages tels que le C¹⁵ ou Rust¹⁶, de nombreuses bibliothèques logicielles exploitent des langages plus bas niveau pour effectuer des calculs intensifs de manière efficace.

Ces bibliothèques, généralement écrites en langages plus primitifs, permettent d'exploiter les performances optimales des machines tout en offrant une interface conviviale et facile à utiliser en Python. Ainsi, Python reste un choix courant pour les applications liées à l'apprentissage automatique, grâce à celles-ci.

3.4. BIBLIOTHÈQUES LOGICIELLES

Comme mentionné précédemment, un certain nombre de bibliothèques logicielles sont exploitées pour permettre au programme de performer plus facilement et plus efficacement diverses tâches.

a. PyTorch

PyTorch¹⁷ est un *framework* (ensemble structuré de bibliothèques, d'outils et de conventions qui offrent un cadre de développement facilité) d'apprentissage automatique open-source. Il permet de développer des applications d'intelligence artificielle avec facilité. Celui-ci propose des bibliothèques et des modules pour diverses tâches d'apprentissage automatique, dont l'application de la vision numérique. De plus, il bénéficie de mises à jour régulières et d'une documentation abondante. Dans notre contexte, il permet l'instanciation et les inférences des modèles pré-entraînés **SSDLite** et **YOLOs-Tiny**.

14. www.python.org

15. [en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language))

16. www.rust-lang.org

17. *PyTorch*, [s. d.]

b. Hugging face

Hugging face¹⁸ est une entreprise spécialisée dans le développement d'outils et de bibliothèques pour l'apprentissage automatique. Leur plateforme open-source offre un large éventail de modèles pré-entraînés permettant aux chercheurs et aux développeurs d'accéder facilement à des fonctionnalités avancées. Le modèle YOLOS-Tiny utilisé pour les comparaisons et pour substitut visuel principal provient de cette plateforme. Grâce à leur bibliothèque logicielle compatible avec Python.

c. txtai

Txtai¹⁹ est une bibliothèque open-source qui offre des fonctionnalités avancées de recherche, de traitement du langage naturel (NLP) et de génération de résumés. Elle permet d'exploiter des modèles de pointe pour analyser et extraire des informations à partir de grandes quantités de texte. Celle-ci offre également un système de synthèse vocale qui permet de convertir du texte en audio à l'aide d'une voix synthétique féminine. Dans le contexte de notre projet, cette fonctionnalité est utilisée pour la retranscription des détections.

d. Open Computer Vision

Open Computer Vision (OpenCV)²⁰ est une bibliothèque open-source largement utilisée dans le domaine de la vision par ordinateur et du traitement d'images. Elle offre une vaste gamme de fonctionnalités pour la capture, la manipulation et l'analyse d'images et de vidéos. Elle dispose d'une documentation détaillée, d'exemples de code et des mises à jour régulières. Elle est utilisée dans notre programme pour manipuler le flux vidéo continu provenant de la caméra, l'interaction de utilisateur ainsi diverses tâches plus simples comme l'écriture et le dessin sur image.

e. NumPy

NumPy²¹ est une bibliothèque en Python pour le calcul scientifique et numérique. Elle fournit des structures de données efficaces pour représenter et manipuler des tableaux multidimensionnels.

18. *Hugging Face – The AI community building the future.* [s. d.].

19. Mezzetti, 2020.

20. *OpenCV*, [s. d.].

21. *NumPy*, [s. d.].

mensionnels, ainsi qu'une collection de fonctions mathématiques pour effectuer des opérations sur ceux-ci. NumPy est reconnue pour sa rapidité et son efficacité, ce qui en fait un choix privilégié pour les calculs numériques intensifs. De ce fait, elle est utilisée durant l'exécution du programme pour traiter les images reçues.

f. Autres bibliothèques

Webcolors ²² est un module initialement créé pour travailler avec les définitions de couleurs utilisées dans les standards du net. Dans notre cas, nous exploitons uniquement les définitions des couleurs sous la forme clé-valeur. Le script contenant les noms des couleurs ainsi que leurs valeurs encodées en RGB (Rouge, Vert, Bleu) est utilisé pour traduire une valeur en nom de couleur. Cette fonctionnalité facilite l'identification ainsi l'énonciation des couleurs par la voix synthétique.

PyRealSense2 ²³ est un module permettant l'utilisation simplifiée de la caméra Intel RealSense D455.

L'installation de toutes ces dépendances logicielles constituent un environnement conséquent pour la machine hôte. Pour éviter que ceux-ci ne viennent polluer la mémoire de l'hôte sur le long terme, des environnements virtuels sont créés et utilisés.

3.5. JEUX DE DONNÉES

Pour alimenter les modèles en images afin de pouvoir mener les essais dans le cadre de la comparaison entre eux, une source de données conséquente nous est nécessaire. Dans le but de combler ce besoin, la base de données **Common Objects in Context (COCO)** ²⁴. Celle-ci contient plus de 200 000 images annotées, couvrant 80 classes d'objets différentes. Les annotations fournies incluent les coordonnées **bbox** pour chaque objet détecté, ainsi que les catégories d'objets correspondantes. Elle est largement utilisée dans le domaine de la vision par ordinateur pour entraîner et tester des modèles spécialisés dans la vision numérique.

22. Bennett, 2023.

23. *IntelRealSense/librealsense*, 2023.

24. Lin et al., 2015.

CHAPITRE 4 : L'ÉTUDE COMPARATIVE

4.1. GÉNÉRALITÉS

Lors de la présentation des différentes architectures de modèles de détection d'objets, la pertinence d'une comparaison a été mentionnée comme étant le précurseur concernant développement du système complet. De ce fait, les deux concourants pré-sélectionnés sont :

YOLOS-Tiny²⁵ est une alternative du célèbre modèle YOLO. Celui-ci arbore une architecture basée sur les **ViT**, différente de ses congénères dotés d'une architecture **RNC**. Le document exposant cette version de l'ensemble *You Only Look at one Sequence* affirme que celui-ci a obtenu un score de précision moyenne de 28.7% sur les différentes catégories d'objets disponibles sur le jeu de données **COCO**. Ce modèle a été sélectionné par nos soins, car celui-ci se base sur une architecture récente qui mérite d'être observée de plus près, surtout dans notre cas d'utilisation. De plus, celui-ci prétend faire preuve d'une efficacité remarquable tout en étant relativement peu lourd à exploiter sous cette version *Tiny*.

Métrique	Résultat
Paramètres	5'700'000
FLOPs ²⁶	1'200'000
Précision moyenne	28.7%

TABLEAU 4.1 – Spécificités du modèle YOLOS-Tiny. Source : tiré de www.pytorch.org ref. URL03.

SSDLite²⁷ est un **RNC** composé entre l'algorithme **SSD** en version allégé (d'où l'extension *Lite* ou léger en anglais) et le squelette de l'architecture du **RNC MobileNet** dans sa troisième version. Celui-ci est donc une combinaison de plusieurs éléments décrits comme conçus pour être efficace dans l'application de la détection d'objets en temps réel. La bibliothèque logicielle le mettant à disposition classe sa précision moyenne à 21.3% sur les différentes catégories d'objets disponibles sur le jeu de données **COCO**.

25. Fang et al., 2021.

26. en.wikipedia.org/wiki/Floating-point_arithmetic

27. *Everything You Need To Know About Torchvision's SSDlite Implementation*, [s. d.].

Métrique	Résultat
Paramètres	3'400'000
FLOPs	580'000
Précision moyenne	21.3%

TABLEAU 4.2 – Spécificités du modèle **SSDLite**. Source : tiré de www.arxiv.org ref. URL04.

4.2. MÉTRIQUES UTILISÉES

a. Score

Le score est un terme générique choisi pour indiquer le nombre de prédictions correctes lors d'une inférence en fonction d'une limite de confiance fixée (appelé "threshold"). Cette valeur indique l'efficacité générale du modèle sous la contrainte exprimant l'interprétation du modèle quant à la qualité de sa propre détection. Celui-ci peut être exprimé en fraction ou en pourcentage.

b. Intersection over Union

L'indice de recouvrement²⁸ ou plus communément appelé **IoU** dans le domaine, cette métrique a pour but d'indiquer la précision du placement de la zone de détection (appelé **bbox**) réalisée par le modèle en comparant sa précision avec la zone prédéfinie représentant la réponse considéré comme étant la plus juste. La zone partagée entre la prédiction et la réponse attendue génère un score exprimé en pourcentage relatif à la zone totale de la réponse attendue. De ce fait, plus la zone de prédiction chevauche la zone attendue, plus le pourcentage sera haut. Un pourcentage **IoU** est considéré excellent lorsque $IoU \geq 90\%$, bon lorsque $IoU \geq 75\%$, satisfaisant lorsque $IoU \geq 60\%$ et insatisfaisant lorsque $IoU < 60\%$. Cette métrique peut être exprimée en pourcentage ou en valeur normalisée $x \in [0, 1]$.

28. *Jaccard index*, 2023.



ILLUSTRATION 4.1 – Représentation visuelle de l’indice de recouvrement. Source : tiré de www.pyimagesearch.com ref. URL10.

c. Temps d’exécution d’une inférence

Lorsqu’un modèle de détection d’objets reçoit une image en entrée, celui-ci va la traiter en exécutant une multitude d’opérations propres à l’architecture du modèle en question. En fonction de sa grandeur et complexité, le temps d’exécution d’une inférence peut sensiblement varier. Dans le cadre du développement de notre système, le principe d’exécution en temps réel est importante et se doit d’être respecté en choisissant un modèle capable de fonctionner dans un intervalle de temps raisonnable. Cette métrique se mesure en millisecondes (ms).

d. Confiance

La confiance est un score, exprimé en pourcentage ou en valeur normalisée $x \in [0, 1]$, produit par la détection d’un ou plusieurs objets lors d’une inférence. Cette métrique indique la qualité selon l’interprétation du modèle lors de cette détection.

e. Threshold

Le *threshold* désigne une limite selon laquelle les détections sont triées en fonction de leur score de confiance. De ce fait, seuls les détections possédant un score plus haut ou égal à ce palier seront pris en compte lors de la génération des données en sorties. Cette métrique s’exprime en pourcentage ou en valeur normalisée $x \in [0, 1]$.

4.3. MÉTHODOLOGIE UTILISÉE

Les essais sont automatisés grâce à plusieurs scripts, écrits avec le langage de programmation Python. Ceux-ci sont agencés selon le diagramme suivant :

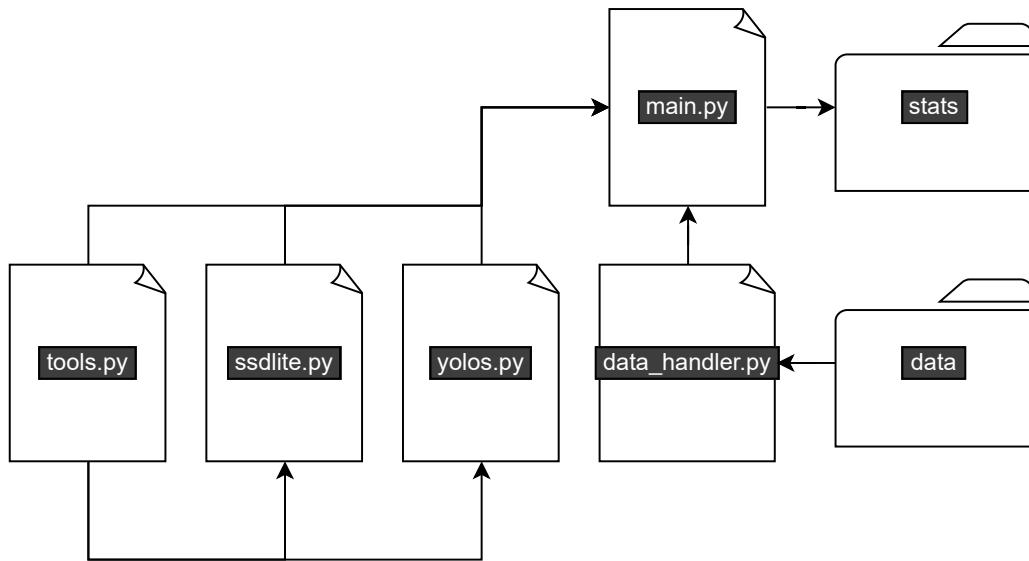


ILLUSTRATION 4.2 – Représentation de l'ensemble du programme de comparaison. Source : réalisé par Damian Boquete Costa

Le programme principal nommé *main.py* s'occupe de lancer la batterie de tests selon des options spécifiées lors du lancement du programme en ligne de commande. Celui-ci s'occupe de lancer quatre tours de tests sur quatre limites de confiance différentes : 0%, 50%, 75% et 90%. Lors de l'exécution du programme par ligne de commande, il est possible de spécifier un nombre d'images utilisées pour l'inférence lors d'un tour de test. De ce fait, si l'on spécifie un nombre d'images à 10, alors chaque tour de test s'exécutera sur 10 images différentes. La deuxième et dernière option permet de visualiser les images ainsi que les inférences réalisées sur celles-ci.

Lors de son lancement, la première tâche effectuée concerne le jeu de données utilisé pour la batterie de test. Le programme relaye donc son fonctionnement au fichier *data_handler.py* qui s'occupe de télécharger le jeu de données COCO localement si cela n'est pas déjà fait. Ensuite, il s'occupe de modeler le jeu de données pour qu'il puisse être compatible avec les modèles de détection pré-entraînés. Les modèles mentionnés, provenant de *yolos.py* et *ssdlite320.py*, sont ensuite instanciés et exécutés avec le jeu de données fraîchement mis en place.

Pour tout ce qui concerne les fonctionnalités internes au programme, tel que la création des graphiques et autres calculs relatifs aux métriques, sont gérés par le fichier *tools.py*. La jointure de tous ces fichiers, comme illustré ci-dessus, permettent de calculer et obtenir les résultats et échantillons présentés à continuation.

4.4. RÉSULTATS ET ÉCHANTILLONS

Les expériences ont été menées pour évaluer la pertinence des modèles en tant que substituts visuels. Chaque itération a utilisé un ensemble de 1000 images.

a. Nombre de détections correctes

En ce qui concerne la capacité des modèles à identifier les objets, on observe un écart de performance significatif entre les deux modèles.

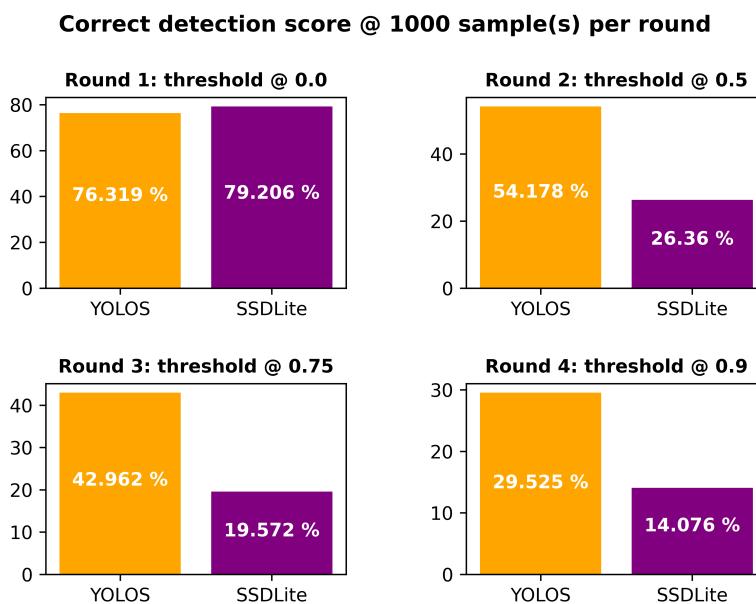


ILLUSTRATION 4.3 – Statistiques montrant les scores de détection. Source : réalisé par Damian Boquete Costa

Palier de confiance (Threshold)	Score YOLOS-Tiny	Score SSDLite
0%	5498/7204 (76.32%)	5706/7204 (79.21%)
50%	3903/7204 (54.18%)	1899/7204 (26.36%)
75%	3095/7204 (42.96%)	1410/7204 (19.57%)
90%	2127/7204 (29.53%)	1014/7204 (14.08%)

TABLEAU 4.3 – Comparaison des scores de détection. Source : réalisé par Damian Boquete Costa

Le modèle YOLOS-Tiny présente une précision plus élevée lorsque des seuils de confiance sont appliqués, tandis que le modèle SSDLite connaît une baisse significative de performance dans ces conditions. Cela peut indiquer un manque de précision du modèle SSDLite.

b. Score IoU

Contrairement à la supposition énoncée lors de l'analyse statistique précédente, il est possible ici d'observer que le modèle SSDLite est plus performant en termes de précision de placement lors d'une détection.

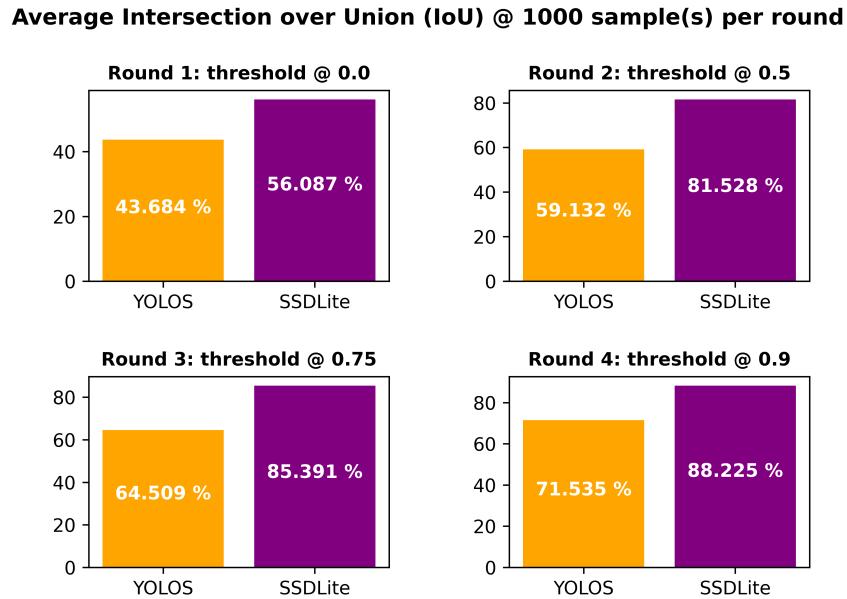


ILLUSTRATION 4.4 – Statistiques montrant les scores d’IoU. Source : réalisé par Damian Boquete Costa

Palier de confiance (Threshold)	Score YOLOS-Tiny	Score SSDLite
0%	43.68%	56.09%
50%	59.13%	81.53%
75%	64.51%	85.39%
90%	71.54%	88.23%

TABLEAU 4.4 – Comparaison des scores d’IoU. Source : réalisé par Damian Boquete Costa

Cependant, en termes de nombre de détections par itération, le modèle SSDLite génère beaucoup moins de détections que le modèle YOLOS-Tiny. Par conséquent, on peut penser que le score IoU plus élevé du modèle SSDLite est dû au fait qu'il détecte moins d'objets que le modèle YOLOS-Tiny.

c. Score de confiance

Dans l'ensemble, les deux modèles ont des performances similaires lorsqu'une limite de confiance est appliquée. Cependant, il est important de noter que le modèle YOLOS-Tiny produit environ deux fois plus de prédictions lorsqu'une limite est imposée. En tenant compte de ce paramètre, le modèle YOLOS-Tiny est capable de générer davantage d'informations tout en maintenant un score de confiance comparable à celui du modèle SSDLite.

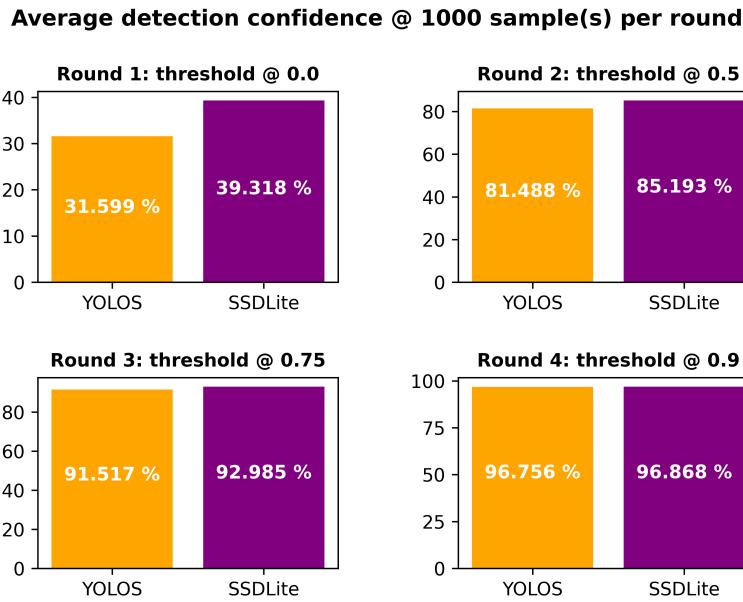


ILLUSTRATION 4.5 – Statistiques montrant les scores de confiance. Source : réalisé par Damian Boquete Costa

Palier de confiance (Threshold)	Score YOLOS-Tiny	Score SSDLite
0%	31.6%	39.32%
50%	81.49%	85.19%
75%	91.52%	92.99%
90%	96.76%	96.87%

TABLEAU 4.5 – Comparaison des scores de confiance. Source : réalisé par Damian Boquete Costa

d. Temps d'exécution

En analysant ces statistiques, on constate que le modèle **SSDLite** affiche des temps d'exécution nettement inférieurs à ceux du modèle **YOLOS-Tiny** pour toutes les limites de confiance considérées. En effet, le modèle **SSDLite** présente une moyenne de temps d'exécution comprise entre 5.942 et 6.11 millisecondes, tandis que le modèle **YOLOS-Tiny** affiche une moyenne de temps d'exécution allant de 30.518 à 32.238 millisecondes.

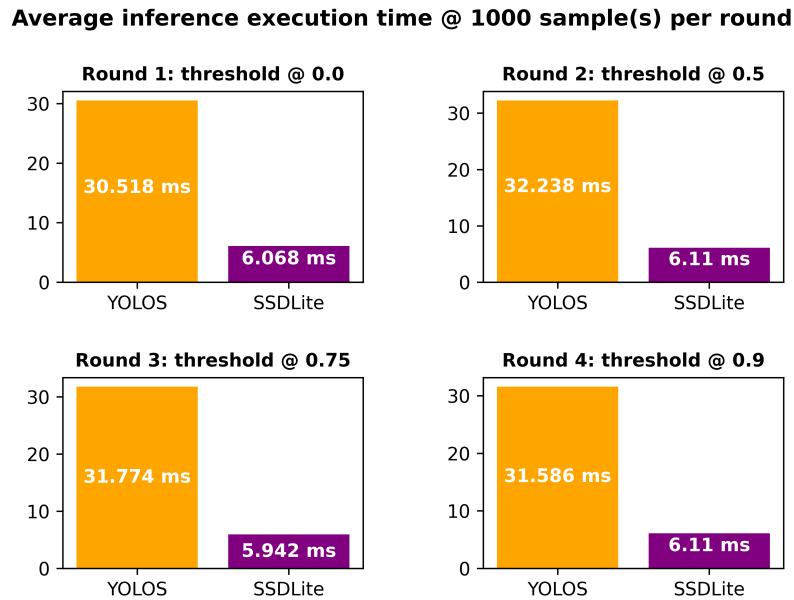


ILLUSTRATION 4.6 – Statistiques montrant les temps d'exécution. Source : réalisé par Damian Boquete Costa

Palier de confiance (Threshold)	Score YOLOS-Tiny	Score SSDLite
0%	30.518 ms	6.0678 ms
50%	32.2383 ms	6.1101 ms
75%	31.7738 ms	5.9417 ms
90%	31.5865 ms	6.1104 ms

TABLEAU 4.6 – Comparaison des temps d'exécution. Source : réalisé par Damian Boquete Costa

e. Échantillons visuels

Les statistiques présentées jusqu'ici offrent une vision globale des performances des deux modèles mis en concurrence. Afin de se faire une idée des capacités de ces deux-ci, quelques échantillons visuels sous formes d'images sur lesquelles des inférences ont été réalisées et enregistrées.



ILLUSTRATION 4.7 – Échantillons d'inférences avec une limite de confiance de minimum 75%.
Source : tiré du jeu de données disponible sur www.cocodataset.org ref. URL11.

Les illustrations ci-dessus arborent deux types de bbox, les bbox vertes représentent les détections attendues et les bleues représentent les détections inférées par les modèles. En s'appuyant sur ces images, on remarque que, comme l'ont énoncé les statistiques précédentes, le modèle YOLOS-Tiny est capable de générer plus de détections que son adversaire SSDLite

4.5. DÉCISION

L'affrontement entre ces deux modèles de détection met en évidence plusieurs points d'attention. Premièrement et sans doute le plus important, le score de détection. On remarque que **YOLOS-Tiny** gagne avec une bonne marge sur son concurrent **SSDLite**. Il nous est important d'avoir un modèle performant sous la main capable de performer un maximum de détection.

Le second point est l'**IoU**, il nous est nécessaire d'avoir un outil capable de savoir situer correctement une détection dans l'espace, afin de pouvoir s'en servir lors d'une énumération d'objets selon positionnement spatial. À ce niveau-là, les deux modèles montrent des statistiques intéressantes. Cependant, **YOLOS-Tiny** et son nombre de détections équivalent au double de son concurrent gagnent cette manche.

Pour finir, le temps d'exécution d'une inférence se doit d'être minime afin de permettre au système de fonctionner avec un temps de latence minime dû à la contrainte du fonctionnement en temps réel. Concernant cela, il est indéniable que le modèle **SSDLite** l'emporte avec ses temps d'exécution approximativement cinq fois plus courts que ceux de **YOLOS-Tiny**. C'est avec cette dernière métrique que l'on remarque un fait indéniable : la précision a un coût considérable sur la performance.

Sur la base de ces observations, le modèle de détection d'objets **YOLOS-Tiny** sera adopté en tant que principal. Cependant, au cours de ces expérimentations, l'idée d'incorporer la possibilité de changer de modèle en cours de fonctionnement a été prise en considération. Par conséquent, le modèle **SSDLite** sera également implémenté dans le programme.

CHAPITRE 5 : L'ANATOMIE DU SYSTÈME

5.1. GÉNÉRALITÉS

Dans ce chapitre, nous fournirons une description détaillée du système principal implémentant le programme de substitution visuelle. Nous commencerons par une vue d'ensemble de son architecture. Ensuite, nous expliquerons en détail les fonctionnalités mises en œuvre par les différentes composantes du système. Des tests utilisateurs réalisés tout au long du développement du programme ainsi que leur synthèse seront exposés. Finalement, les difficultés rencontrées durant le développement de ce système seront énumérées, montrant les limitations de ce travail en passant.

5.2. FONCTIONNEMENT

Lorsque le programme est exécuté, la voix synthétique accueille l'utilisateur et indique la touche correspondant au mode d'aide. Ensuite, le programme attend passivement l'entrée de l'utilisateur via le pavé numérique. Différents modes sont disponibles, comme expliqué dans la section 5.4.

Ces modes peuvent fonctionner de deux manières distinctes : dépendant ou indépendant des entrées de l'utilisateur. Par exemple, le mode annonçant les distances ne nécessite pas de signal provenant de l'appui sur une touche du pavé numérique. En revanche, le mode de détection verticale n'activera pas l'inférence ni la description spatiale à moins que l'utilisateur n'interagisse avec la touche *0* du pavé.

Une fois que le programme est dans un mode sélectionné, ce mode peut être quitté en appuyant sur n'importe quelle touche, à l'exception des modes nécessitant l'interaction de l'utilisateur. Dans ces cas, la touche *0* ne peut pas servir de fonction de sortie.

Enfin, l'utilisateur peut éteindre le programme en appuyant sur le bouton de sortie *0* s'il est en mode d'attente passive.

L'exécution du programme peut se faire en modifiant plusieurs paramètres, voici la liste complète :

Option	Variante	Description
-h	--help	Affiche l'aide concernant l'utilisation du programme.
-s	--speecharate	Définit la vitesse de la voix synthétique.
-m	--model	Définit quel modèle, entre YOLOS-Tiny et SSDLite , est actif.
-cr	--color_resolution	Définit la résolution du flux vidéo normal utilisé par la caméra.
-dr	--depth_resolution	Définit la résolution du flux vidéo de profondeur utilisé par la caméra.
-f	--fps	Définit le nombre d'images par seconde auquel la caméra sera soumise.
-yv	--yolos_version	Définit la version de YOLOS-Tiny utilisée.
-d	--display	Indique si le flux vidéo est affiché à l'écran.
-c	--color_mode	Indique si le mode couleur fonctionne avec la voix synthétique ou avec les sons d'instruments.

TABLEAU 5.1 – Options disponibles lors de l'exécution du programme. Source : réalisé par Damian Boquete Costa

Par défaut, le programme peut se lancer sans avoir besoin de spécifier aucune option. Dans ce cas-là, les paramètres par défaut sont appliqués.

5.3. L'ARCHITECTURE DU SYSTÈME

Le programme, d'un point de vue technique, se décompose en plusieurs scripts Python organisés en dossiers. Ces fichiers s'interconnectent entre eux, permettant l'articulation des fonctionnalités. Le répertoire racine, contenant l'entièreté des dossiers et fichiers sources se nomme *src*. Pour plus de précisions sur les détails d'implémentation, les fichiers sources commentés sont disponibles sur le dépôt distant (voir annexes).

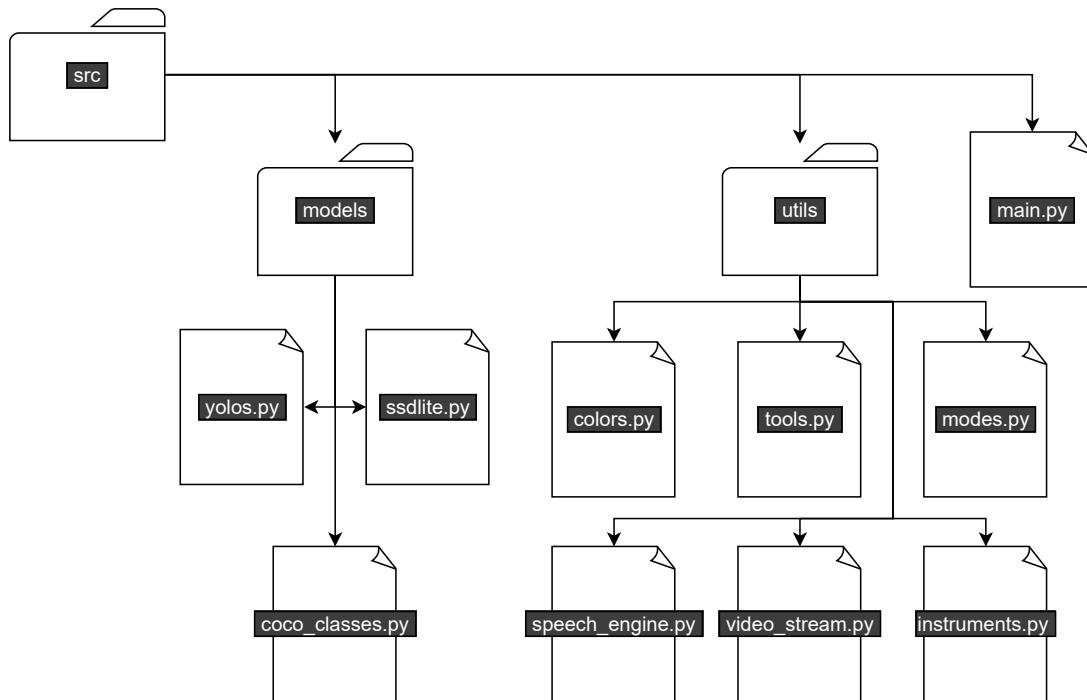


ILLUSTRATION 5.1 – Diagramme du programme de substitution visuelle. Source : réalisé par Damian Boquete Costa

a. Le dossier *src*

models est le répertoire regroupant l'implémentation de tous les modèles de détection d'objets utilisés.

utils est le répertoire contenant les divers scripts régissant les fonctionnalités majeures.

main.py est le script principal par lequel le démarrage ainsi que le comportement dicté par les entrées utilisateurs en temps réel.

b. Le dossier *models*

yolos.py définit l'implémentation et l'utilisation de YOLOS-Tiny.

ssdlite.py définit l'implémentation et l'utilisation de SSDLite.

coco_classes.py répertorie toutes les classes d'objets détectables par tout modèle pré-entraîné sur le jeu de données COCO.

c. Le dossier *utils*

sound_engine.py est le script permettant l'utilisation parallélisée de la voix synthétique servant de guide tout au long de l'exécution du programme. Celui-ci gère aussi la sortie des sons d'instruments si le mode couleur l'utilise.

instruments.py est un script modifié provenant d'un projet similaire, crée par deux assistants de l'[HEPIA](#). Celui-ci gère la traduction des couleurs en sons.

video_stream.py assure le bon fonctionnement du flux vidéo en continu de la part de la caméra.

modes.py contient l'implémentation des différents modes mis à disposition à l'utilisateur du programme.

tools.py fournit des fonctions utiles pour faciliter l'implémentation de divers comportements et fonctionnalités utilisés dans le programme.

colors.py définit une liste de correspondances entre les noms de couleurs et leurs valeurs hexadécimales, ainsi qu'une fonction pour trouver le nom de la couleur la plus proche à partir d'une valeur hexadécimale donnée.

5.4. FONCTIONNALITÉS

En ce qui concerne les capacités du programme, celui-ci offre plusieurs modes capables de fournir différentes informations.

a. Le mode couleur

Le mode couleur permet d'annoncer la couleur se trouvant au centre du champ de vision. Deux options sont disponibles pour cette annonce. La première option utilise la voix synthétique pour énoncer le nom des couleurs détectées. La deuxième option utilise des sons d'instruments pour représenter les nuances de couleur. Ces deux options suivent le même principe de calcul de la moyenne des valeurs des 5×5 pixels situés au centre de l'image traitée. Ensuite, cette valeur est transmise au script de l'option choisie pour être traitée et générer la sortie correspondante.

Dans le cas de l'option de voix synthétique, une version modifiée de la bibliothèque *Webcolors* est utilisée pour trouver la correspondance la plus proche entre la valeur de couleur donnée et les valeurs prédéfinies dans le dictionnaire de couleurs. Pour l'option utilisant les sons d'instruments, une fonctionnalité modifiée du projet *sound-color-detobj* réalisé par les assistants d'*HEPIA*, *Ludovic Pfeiffer* et *David Gonzalez*, est utilisée. Cette fonctionnalité traite la couleur en déterminant ses propriétés dans l'espace de couleur HSL²⁹ et en lui attribuant une catégorie de couleur. Cette catégorie est ensuite associée à un son d'instrument correspondant, dont la tonalité est modifiée en fonction de la distance entre la couleur détectée et l'utilisateur.

b. Le mode distance

Ce mode permet d'annoncer la distance séparant l'entité se trouvant en face de l'utilisateur et lui-même. Pour ce faire, la caméra utilise le principe de stéréoscopie³⁰. Cette technique superpose la même image, provenant de deux caméras placées à deux endroits différents de l'appareil. Ainsi, il est possible de reproduire un effet en trois dimensions et d'en déduire une distance. Ce principe est fortement inspiré du fonctionnement du cerveau et du placement des yeux chez l'humain.

Ainsi, la caméra peut donc directement nous offrir l'accès à son interprétation de la distance. Ce mode s'occupe de récupérer cette information et de l'annoncer grâce à la voix synthétique.

29. en.wikipedia.org/wiki/HSL_and_HSV

30. *Stéréoscopie*, 2023.

c. Le mode de détection centrale

Le mode de détection centrale utilise le flux vidéo en temps réel comme donnée d'entrée pour inférer le contenu de chaque image à l'aide du modèle de détection choisi. Il indique les objets détectés dans la région centrale de l'image traitée. Le choix de l'objet pris en compte dépend du fait que la surface de la **bbox** de celui-ci doit se situer dans la partie centrale (verticale uniquement) de l'image.

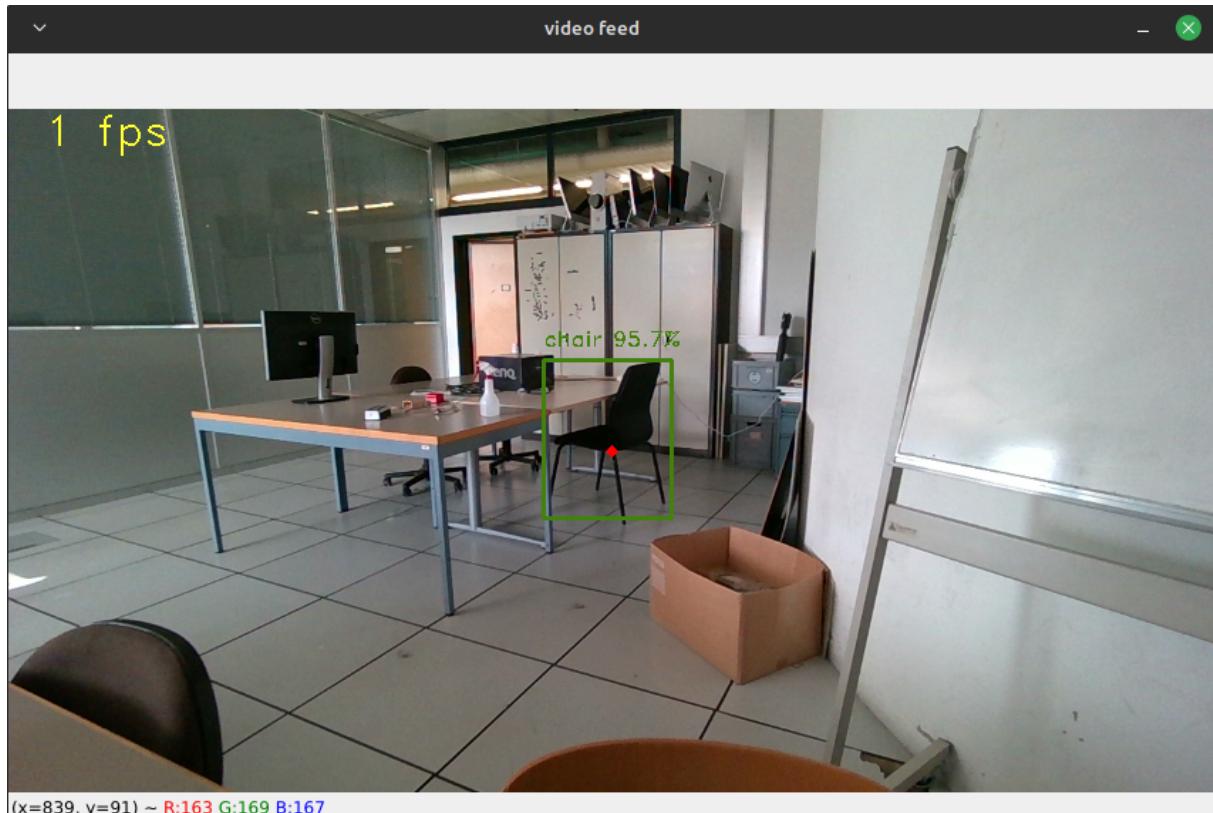


ILLUSTRATION 5.2 – Exemple d'inférence en mode de détection centrale. Source : réalisé par Damian Boquete Costa

Sur la figure 5.2 il est possible de constater qu'une chaise noire à été correctement détecté au centre de l'image, provenant du flux vidéo en temps réel, par le modèle YOLOs-Tiny.

d. Le mode de détection verticale

Le mode de détection verticale fonctionne de manière similaire au mode de détection centrale. Cependant, il annonce tous les objets de gauche à droite. Une bande centrale est délimitée pour distinguer les objets se trouvant de chaque côté. Ce mode est plus coûteux en termes de temps d'exécution en raison de l'énumération lente de la voix synthétique. De ce fait, l'inférence est déclenchée par l'utilisateur en appuyant sur le bouton 0.

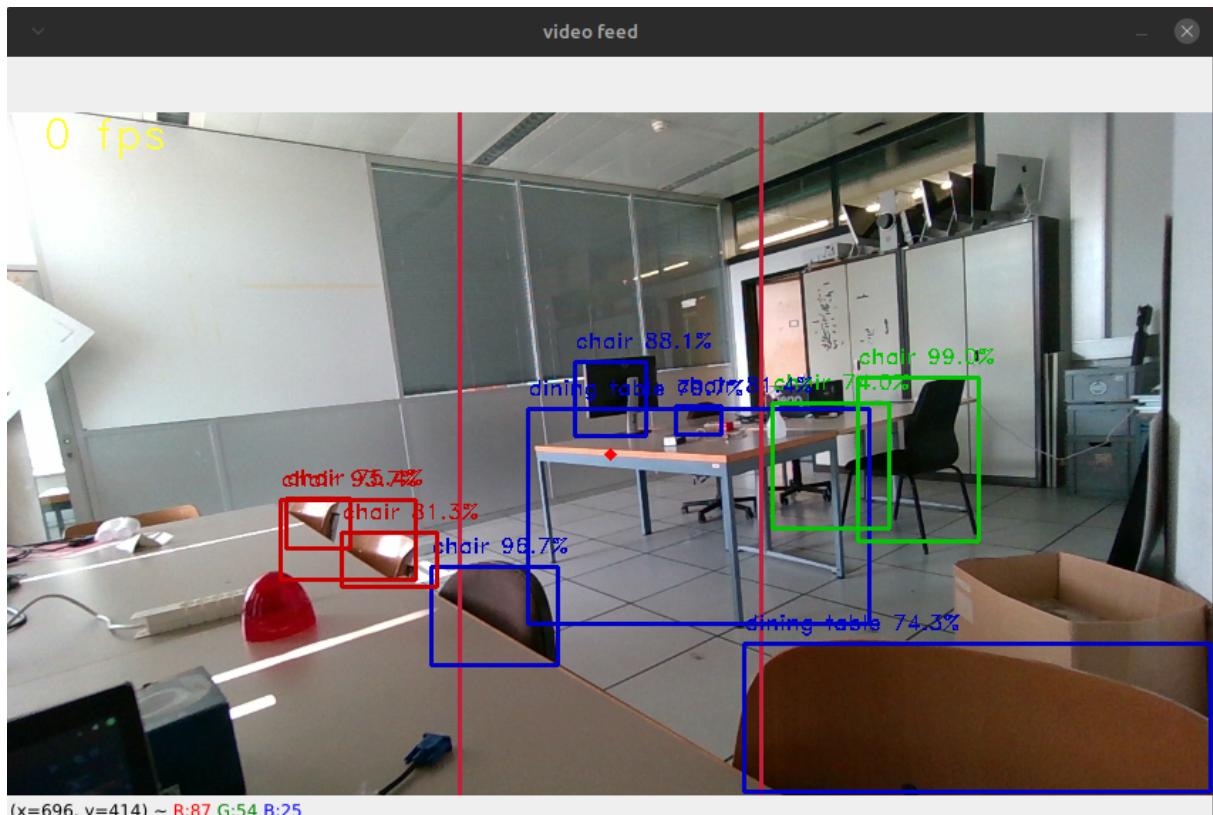


ILLUSTRATION 5.3 – Exemple d'inférence en mode de détection verticale. Source : réalisé par Damian Boquete Costa

Sur la figure 5.3, on peut observer neuf objets qui ont été correctement détectés par le modèle YOLOS-Tiny. Ces inférences sont segmentées par des lignes rouges qui délimitent les côtés gauche, droit et le centre de l'image. Chaque **bbox** est associée à une couleur différente, indiquant celles qui seront énoncées comme étant situées à gauche, au centre ou à droite.

e. Le mode de détection par cadrants

Le mode de détection par cadrants est une extension du mode de détection verticale. Il reprend le comportement du mode de détection verticale en ajoutant une détection par cadran. Ainsi, la voix synthétique annonce les détections de gauche à droite, puis en parcourant le reste par cadran, du coin supérieur droit au coin supérieur gauche en passant par le bas, dans l'ordre des aiguilles d'une montre. Ce mode est encore plus coûteux en termes de temps d'exécution que le précédent et est également déclenché par l'interaction de l'utilisateur.

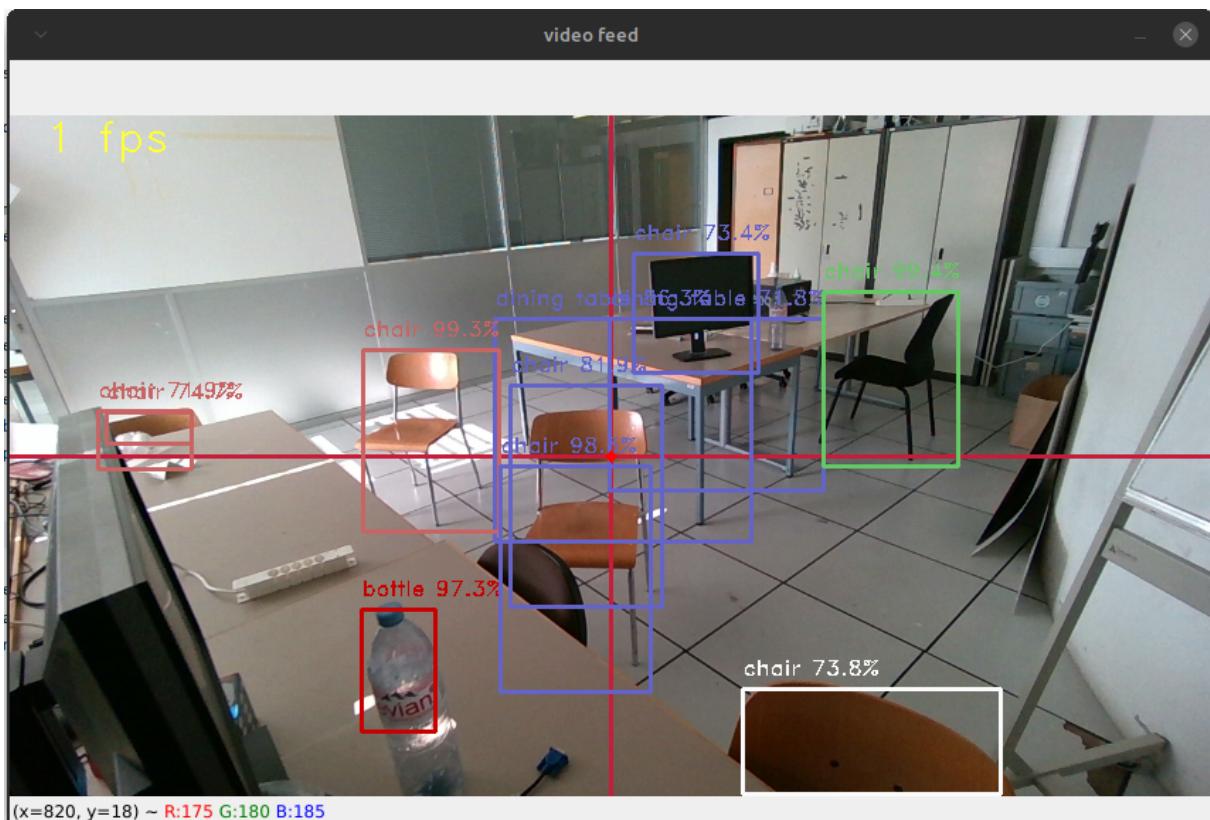


ILLUSTRATION 5.4 – Exemple d’inférence en mode de détection par cadran. Source : réalisé par Damian Boquete Costa

Sur la figure 5.4, un nombre important de détections sont effectuées par le modèle YOLOS-Tiny. De la même manière que pour le mode présenté en section d, les objets détectés sont colorés différemment en fonction de leur position spatiale.

f. Les fonctionnalités annexes

Le système est par ailleurs doté de modes supplémentaires, qui sont moins importants que les autres. Le premier est le mode permettant de changer le modèle de détection d'objets actif. Grâce à ce mode, il est possible d'utiliser les modes d'inférence avec le modèle YOLOS-Tiny ou SSDLite. Le deuxième et dernier mode de la liste concerne l'aide. Ce mode utilise la voix synthétique pour énoncer le nom de tous les modes disponibles ainsi que la touche qui les active.

5.5. TESTS UTILISATEURS

La réalisation d'un tel système dans son intégralité est un travail conséquent qui requiert de nombreuses heures de développement. Comme pour la rédaction d'un long document, il est parfois difficile de prendre du recul sur son propre projet et de détecter ses éventuels points faibles ou lacunes. C'est pourquoi l'intervention de personnes externes est essentielle pour identifier les éventuelles failles de ce système.

Dans le cadre de tests utilisateur, il est possible de créer un scénario propice à l'évaluation du programme. Plusieurs personnes ont généreusement offert leur temps et leur participation pour se prêter à cet exercice. Le scénario consiste à placer une personne équipée du prototype complet dans un espace restreint, comme une pièce d'appartement, et de lui confier la mission de trouver l'un des trois objets spécifiques préalablement disséminés dans cet espace. La personne effectuant le test est obligatoirement bandée au niveau des yeux et est désorientée en effectuant des rotations sur elle-même dans différentes directions à intervalles aléatoires.

Les objets utilisés sont généralement un ordinateur portable, un clavier et une grande bouteille d'eau en plastique. Il est possible de varier ces objets en les remplaçant par un sac à dos ou autre, dans le but d'observer, en parallèle de l'expérience utilisateur, les capacités des modèles de détection. La personne est ensuite observée et chronométrée jusqu'à ce qu'elle atteigne son objectif.

Ces tests permettent d'évaluer les performances et l'efficacité du système de substitution visuelle dans des situations réelles, en simulant les défis auxquels les utilisateurs pourraient être confrontés. Les résultats de ces tests fournissent des informations précieuses pour améliorer le système et optimiser son fonctionnement afin de répondre au mieux aux besoins des utilisateurs.

a. Les tests #1 et #2

Les deux premiers essais se sont révélés peu concluants en termes de performance, mais très utiles en ce qui concerne l'expérience utilisateur. Les deux premières personnes à avoir participé m'ont toutes deux indiqué que les inférences du modèle YOLO-Tiny étaient lentes et qu'il peinait à détecter quoi que ce soit. Celles-ci ont aussi proposé l'idée de combiner le mode de distance avec les modes de détection. L'inconfort généré par l'échange constant entre le mode de distance et les modes de détection était bien trop handicapant pour les utilisateurs. De plus, cette session de tests a dévoilé un problème conséquent de consommation de batterie ainsi que de surchauffe. Ces constatations proviennent du fait que, durant ces deux essais, le pourcentage de batterie est passé de 100% à 46% en 17 minutes et les températures indiquées par les senseurs du processeur indiquaient une valeur alarmante de 94°C.

Suite à ces retours, des mises à jour du système ont été effectuées. Cela comprend l'ajout du mode de distance dans tous les modes de détection. L'ajout du mode d'exécution synchrone du programme de voix synthétique, réduisant les temps d'inférence du modèle YOLO-Tiny, ainsi que l'ajout de plusieurs options d'exécution permettant une meilleure personnalisation.

b. Le test #3

Suite aux premiers essais, un troisième test a été mené dans le but de valider les ajouts et modifications apportés suite aux problèmes rencontrés lors de la dernière session. Le résultat de celui-ci s'avère toujours négatif en ce qui concerne la performance du système à remplir l'objectif principal. Cependant, celle-ci est légèrement meilleure qu'en son état initial. L'utilisateur a rencontré des problèmes persistants au niveau du potentiel de détection du modèle YOLO-Tiny ainsi qu'une détection de couleur très souvent erronée. Cependant, le modèle SSDLite montre une capacité légèrement meilleure à aider le détenteur du système. Les améliorations des modes de détection par le biais de l'ajout du mode de distance ont été bien accueillis et beaucoup utilisés par le testeur.

En ce qui concerne les temps d'exécution d'inférences, ceux-ci se sont améliorés d'environ 40% grâce au mode séquentiel. De plus, la consommation de batterie s'est avérée nettement moins élevée avec une consommation de 6% en 13 minutes. Les températures de la machine aussi ont été bien moins élevées en fin de séance, avec un relevé indiquant 75°C.

Avec l'acquisition de ces informations, le mode couleur a été amélioré grâce à l'utilisation des constantes de couleurs offertes par la bibliothèque *Webcolors*. Ces constantes ont simplement été abstraites en changeant les noms de couleurs originales, souvent trop précises et peu connues par tous.

c. Les tests #4 et #5

Les deux derniers tests, ont été réalisés par deux personnes différentes, dont l'une provenant d'un domaine étranger à l'informatique. Ces tests ont permis de recueillir des critiques et des avis sur l'appréciation globale du programme. Plusieurs constatations ont émergé de ces essais.

Tout d'abord, il a été noté une légère amélioration des performances des modèles de détection. Cela suggère que les ajustements et les mises à jour apportés au système ont eu un impact positif sur leur fonctionnement.

Ensuite, il a été observé que le champ de vision de la caméra était trop restreint, ce qui pouvait affecter la précision des détections. Cela indique la nécessité d'explorer des solutions pour élargir le champ de vision et améliorer la capture des objets, tout en maintenant des performances acceptables.

Par ailleurs, des problèmes ont été identifiés au niveau du fonctionnement de la voix synthétique. Il semble qu'elle ne s'active pas correctement, probablement en raison d'un conflit de source audio généré par le second processus utilisé dans le programme. Ces problèmes ont été résolus grâce aux retours des utilisateurs et aux ajustements effectués.

Enfin, lors des tests, une personne a proposé une idée intéressante concernant le placement de la caméra. Elle suggère de la positionner au niveau de la ceinture plutôt que sur la tête, ce qui permettrait une meilleure détection des objets situés sur les tables.

d. Synthèse

En résumé, grâce aux tests réalisés et aux retours des utilisateurs, le système a pu être amélioré de manière itérative en prenant en compte leurs expériences et leurs préférences d'utilisation. Ces critiques ont été précieuses pour orienter le développement futur du projet et visent à rendre le système plus adapté et efficace pour les potentiels utilisateurs finaux.

5.6. LES DIFFICULTÉS RENCONTRÉES

L'élaboration de ce projet n'a pas été un chemin linéaire et a été ponctuée de diverses difficultés, erreurs et limitations qui ont influencé son développement et ses résultats. Voici les principales difficultés auxquelles nous avons été confrontés :

La voix synthétique : Bien que l'implémentation de la voix synthétique ait été relativement simple, nous avons rencontré des problèmes de latence lors de son utilisation. En effet, l'exécution de la voix synthétique peut être gourmande en temps, ce qui peut entraîner des retards dans l'exécution du programme.

Le parallélisme : L'utilisation de plusieurs processus a posé divers problèmes et a nécessité du temps de débogage. Il peut être difficile de détecter les erreurs d'exécution lorsque le programme n'est pas séquentiel. De plus, l'utilisation de plusieurs processus peut entraîner des incompatibilités avec certaines bibliothèques externes. Par exemple, nous avons rencontré un conflit de périphériques audio lors de l'allocation par une bibliothèque externe, ce qui nous a empêché d'utiliser la même sortie audio que celle allouée par défaut dans le processus principal, même si ce dernier n'utilisait pas de sortie audio.

La traduction couleur-texte : La traduction des couleurs en texte a été un défi en raison de la diversité des espaces colorimétriques, la plupart du temps en trois dimensions. Il est difficile d'interpréter avec précision une couleur à l'aide d'un ordinateur de la même manière que le fait l'œil humain. Par conséquent, cette fonctionnalité reste limitée en termes de fiabilité.



ILLUSTRATION 5.5 – Échelle de couleurs utilisée pour le test de fiabilité du mode de détection de couleur. Source : réalisé par Damian Boquete Costa

Les temps d'inférence des modèles : Les temps d'inférence des modèles utilisés sont très sensibles aux variations de l'exécution. Par exemple, l'utilisation de *threads*³¹ peut entraîner des interférences avec les temps d'inférence, ce qui rallonge considérablement leur durée d'exécution. De ce fait, l'utilisation de la concurrence a été abandonnée au profit du parallélisme.

Ces difficultés ont représenté des défis importants dans le développement du projet, mais elles ont également été des opportunités d'apprentissage et d'amélioration pour surmonter ces obstacles et aboutir à un système plus performant.

31. [en.wikipedia.org/wiki/Thread_\(computing\)](https://en.wikipedia.org/wiki/Thread_(computing))

CONCLUSION

Durant ce travail, plusieurs tâches ont été réalisées pour parvenir à l'état actuel du système. Tout d'abord, une série de recherches a été effectuée pour comprendre le problème principal qui justifie l'existence d'un tel programme. Ensuite, des recherches ont été menées sur les projets et produits existants visant à résoudre cette problématique. Une exploration approfondie des concepts entourant l'outil principal utilisé pour développer ce projet a été réalisée. Une liste exhaustive de ressources techniques a été établie afin de fournir les outils nécessaires pour mener à bien le projet. Un programme complet d'essais des deux types de modèles de détection d'objets a été conçu et utilisé pour générer des statistiques permettant de prendre des décisions éclairées sur la pertinence de ces deux algorithmes. Enfin, un prototype intégrant toutes ces étapes a été développé, bénéficiant de l'aide précieuse de personnes ayant consacré leur temps pour identifier les points faibles et les failles du système.

Ce projet, dans son état du 12 juillet 2023, ne peut être considéré comme suffisamment mature. Celui-ci manque de précision et ne performe pas encore de façon assez efficace pour le considérer comme un outil viable pouvant aider des personnes en difficultés. Cependant, il constitue un bon point de départ. Il possède une base solide sur laquelle il est possible de s'appuyer pour continuer le développement et peut-être parvenir à une version évoluée et plus complexe qui améliorerait réellement le quotidien des personnes dans le besoin.

Si le développement de ce prototype devait se poursuivre, plusieurs améliorations potentielles pourraient être envisagées. Étant donné que le modèle de détection d'objets est la pierre angulaire de ce système, il serait bénéfique d'explorer et d'en tester une plus large gamme. La création d'un modèle personnalisé pourrait également être envisagée, bien que cela soit moins réaliste étant donné que les entraînements de modèles, étant aujourd'hui disponibles en format pré-entraînés, ont souvent bénéficié d'une puissance de calcul considérable que seules les entités les plus puissantes du marché peuvent se permettre d'obtenir.

Des essais sur systèmes embarqués, équipés de processeurs graphiques dédiés, pourraient mener vers un prototype plus compact et portable. Les ordinateurs portables, bien qu'utiles pour le développement, ne fournissent peut-être pas la meilleure expérience utilisateur en termes de mobilité et d'immersion. C'est avec de telles idées que l'avenir de ce projet pourrait se dessiner, le propulsant vers une solution viable et accessible au grand public.

ANNEXES : LIENS GITLAB DES PROGRAMMES

githepia.hesge.ch/damian.boquetec/bachelor-project/-/tree/main/my_detector

githepia.hesge.ch/damian.boquetec/bachelor-project/-/tree/main/tests/vit%vs%cnn

RÉFÉRENCES DOCUMENTAIRES

- Artificial intelligence | Definition, Examples, Types, Applications, Companies, & Facts*, 2023 [en ligne]. [visité le 2023-03-09]. Disp. à l'adr. : <https://www.britannica.com/technology/artificial-intelligence>.
- Be My Eyes - See the world together*, 2012 [en ligne]. [visité le 2023-06-22]. Disp. à l'adr. : <https://www.bemyeyes.com/>.
- BENNETT, James, 2023. *ubernostrum/webcolors* [en ligne]. [visité le 2023-07-11]. Disp. à l'adr. : <https://github.com/ubernostrum/webcolors>. original-date : 2013-09-11T06:45:17Z.
- Cécité*, 2023 [en ligne]. [visité le 2023-07-10]. Disp. à l'adr. : <https://fr.wikipedia.org/w/index.php?title=%C3%A9c%C3%A9t%C3%A9&oldid=204998859>. Page Version ID : 204998859.
- DOSOVITSKIY, Alexey ; BEYER, Lucas ; KOLESNIKOV, Alexander ; WEISSENBORN, Dirk ; ZHAI, Xiaohua ; UNTERTHINER, Thomas ; DEHGHANI, Mostafa ; MINDERER, Matthias ; HEIGOLD, Georg ; GELLY, Sylvain ; USZKOREIT, Jakob ; HOULSBY, Neil, 2021. *An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale* [en ligne]. arXiv [visité le 2023-07-05]. Disp. à l'adr. : <http://arxiv.org/abs/2010.11929>. arXiv :2010.11929 [cs].
- Everything You Need To Know About Torchvision's SSDlite Implementation*, [s. d.] [en ligne]. [visité le 2023-07-11]. Disp. à l'adr. : <https://pytorch.org/blog/torchvision-ssdlite-implementation/>.
- FANG, Yuxin ; LIAO, Bencheng ; WANG, Xinggang ; FANG, Jiemin ; QI, Jiyang ; WU, Rui ; NIU, Jianwei ; LIU, Wenyu, 2021. *You Only Look at One Sequence : Rethinking Transformer in Vision through Object Detection* [en ligne]. arXiv [visité le 2023-07-11]. Disp. à l'adr. : <http://arxiv.org/abs/2106.00666>. arXiv :2106.00666 [cs].

HASAN, Md. Zahidul ; SIKDER, Shovon ; RAHAMAN, Muhammad Aminur, 2022. Real-Time Computer Vision Based Autonomous Navigation System for Assisting Visually Impaired People using Machine Learning. In : *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*, p. 1-6. Disp. à l'adr. DOI : [10.1109/STI56238.2022.98103268](https://doi.org/10.1109/STI56238.2022.98103268).

Hugging Face – The AI community building the future. [s. d.] [en ligne]. [visité le 2023-07-06]. Disp. à l'adr. : <https://huggingface.co/>.

IntelRealSense/librealsense, 2023 [en ligne]. Intel® RealSense™ [visité le 2023-07-06]. Disp. à l'adr. : <https://github.com/IntelRealSense/librealsense>. original-date : 2015-11-17T20:42:18Z.

Jaccard index, 2023 [en ligne]. [visité le 2023-03-12]. Disp. à l'adr. : https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=1144140835. Page Version ID : 1144140835.

LIN, Tsung-Yi ; MAIRE, Michael ; BELONGIE, Serge ; BOURDEV, Lubomir ; GIRSHICK, Ross ; HAYS, James ; PERONA, Pietro ; RAMANAN, Deva ; ZITNICK, C. Lawrence ; DOLLÁR, Piotr, 2015. *Microsoft COCO : Common Objects in Context* [en ligne]. arXiv [visité le 2023-07-11]. Disp. à l'adr. : <http://arxiv.org/abs/1405.0312>. arXiv :1405.0312 [cs].

LIU, Wei ; ANGUELOV, Dragomir ; ERHAN, Dumitru ; SZEGEDY, Christian ; REED, Scott ; FU, Cheng-Yang ; BERG, Alexander C., 2016. SSD : Single Shot MultiBox Detector. In : [en ligne]. T. 9905, p. 21-37 [visité le 2023-06-27]. Disp. à l'adr. DOI : [10.1007/978-3-319-46448-0_2](https://doi.org/10.1007/978-3-319-46448-0_2). arXiv :1512.02325 [cs].

MEZZETTI, David, 2020. *txtai* [en ligne]. [visité le 2023-07-06]. Disp. à l'adr. : <https://github.com/neuml/txtai>. original-date : 2020-08-09T19:14:59Z.

NumPy, [s. d.] [en ligne]. [visité le 2023-07-06]. Disp. à l'adr. : <https://numpy.org/>.

OpenCV, [s. d.] [en ligne]. [visité le 2023-07-06]. Disp. à l'adr. : <https://opencv.org/>.

PyTorch, [s. d.] [en ligne]. [visité le 2023-07-06]. Disp. à l'adr. : <https://www.pytorch.org/>.

REDMON, Joseph ; FARHADI, Ali, 2018. YOLOv3 : An Incremental Improvement. *arXiv*.

Stéréoscopie, 2023 [en ligne]. [visité le 2023-07-07]. Disp. à l'adr. : <https://fr.wikipedia.org/w/index.php?title=St%C3%A9r%C3%A9oscopie&oldid=205773502>. Page Version ID : 205773502.

TapTapSee - Blind and Visually Impaired Assistive Technology, 2017 [en ligne]. [visité le 2023-06-22]. Disp. à l'adr. : <https://taptapseeapp.com/>.

UNDERSCORE_, 2023. *Ce robot va tout changer* [en ligne]. [visité le 2023-06-22]. Disp. à l'adr. : <https://www.youtube.com/watch?v=CizFMRHlzI4>.

VASWANI, Ashish; SHAZER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Lukasz; POLOSUKHIN, Illia, 2017. *Attention Is All You Need* [en ligne]. arXiv [visité le 2023-07-05]. Disp. à l'adr. : <http://arxiv.org/abs/1706.03762>. arXiv :1706.03762 [cs].

What is Deep Learning ?, [s. d.]. *What is Deep Learning ? | IBM* [en ligne]. [visité le 2023-03-16]. Disp. à l'adr. : <https://www.ibm.com/topics/deep-learning>.

What is Machine Learning ?, [s. d.]. *What is Machine Learning ? | IBM* [en ligne]. [visité le 2023-03-09]. Disp. à l'adr. : <https://www.ibm.com/topics/machine-learning>.