



Université Hassan I  
Faculté des Sciences et Techniques  
Département informatique



Master Réseaux et système informatique

Projet de fin d'année

**APPLICATION DE RECONNAISSANCE  
DES OBJETS**

Encadré Par :

**M.abdellah ezzati**

Réalisé Par :

**LAFRAOUZI Mouhssine**

**Soutenu le 27 juillet 2024 devant les membres de jury :**

M.Abdellah ezzati

M.Abderrahim Marzouk

Professeur à la FST Settat

Professeur à la FST Settat

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# REMERCIEMENTS

Ce mémoire n'aurait pas pu être réalisé sans le soutien inestimable de nombreuses personnes que je tiens à remercier sincèrement pour leur aide tout au long de mon parcours.

Je tiens à exprimer ma profonde gratitude au Professeur **EZZATI Abdellah**, Chef de la filière, pour son engagement exemplaire envers ses étudiants. Sa bienveillance et son dévouement ont créé un environnement d'apprentissage stimulant et propice à notre développement

Je remercie également chaleureusement **les membres du jury**, dont les conseils avisés et l'expertise ont enrichi la qualité de mon mémoire. Leur présence et leur soutien sont une véritable reconnaissance de mon travail, et je leur en suis profondément reconnaissant.

Je souhaite exprimer toute ma reconnaissance à **ELKADDARI Hamza** et **NUIH Omar** pour leur soutien indéfectible. Leur disponibilité et leurs conseils précieux ont été d'une aide inestimable tout au long de mon parcours académique.

Un grand merci également à **tous mes professeurs** de la filière RSI pour leur enseignement de qualité et leur accompagnement attentif. Leurs cours ont été essentiels pour acquérir des connaissances solides et développer mes compétences.

Je n'oublie pas non plus **mes camarades de promotion**, avec qui j'ai partagé des moments précieux et enrichissants. Leur amitié et leur soutien ont été une source d'inspiration tout au long de cette expérience.

# DÉDICACES

## À MES CHERS PARENTS

Je dédie ce modeste travail et mes profondes gratitude à mes chers parents pour l'éducation qu'ils m'ont prodiguée, avec tous les moyens et au prix de tous les sacrifices qu'ils ont consentis à mon égard, pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance.

## À MA FAMILLE PROCHE

J'espère sincèrement que ce travail est une façon de témoigner ma profonde gratitude envers ma famille pour leur amour et leur soutien inconditionnel tout au long de ces années. Aucun mot ne peut véritablement exprimer toute l'appréciation que je ressens à leur égard.

## À MES AMIS

Je tiens à exprimer ma profonde gratitude à mes amis pour tous les moments inoubliables que nous avons partagés ensemble et pour leur aide et leur soutien. Je leur suis reconnaissant pour leur présence à mes côtés et pour les instants précieux que nous avons partagés.

## À MES PROFESSEURS

Pour votre encadrement, vos conseils, vos critiques, et vos jugements, je vous remercie sincèrement. Votre soutien a été inestimable.

# RÉSUMÉ

Ce rapport explore en profondeur les approches et technologies utilisées dans le domaine crucial de la reconnaissance des objets, relevant de l'intelligence artificielle et de la vision par ordinateur. Dans le premier chapitre, nous examinons les concepts essentiels de la détection et du suivi des objets, ainsi que leurs nombreuses applications dans divers secteurs.

Le second chapitre présente un état de l'art (SOTA en anglais, pour "State Of The Art"), décrivant les meilleures techniques et algorithmes disponibles. Il aborde les principales catégories d'algorithmes, le fonctionnement des réseaux de neurones, ainsi que les divers modèles utilisés tels que R-CNN, YOLO et SSD, en plus des frameworks de Deep Learning associés.

Le troisième chapitre se concentre sur l'analyse approfondie et la conception du système, en identifiant les besoins spécifiques et en proposant une architecture appropriée.

Enfin, le quatrième chapitre détaille l'implémentation pratique de l'application, en couvrant les choix technologiques, les étapes d'implémentation ainsi que les résultats obtenus.

Ce rapport vise à offrir une compréhension approfondie des techniques modernes de reconnaissance des objets et de leur application concrète, mettant en lumière les défis rencontrés et les solutions apportées tout au long du processus.

# ABSTRACT

This report explores in depth the approaches and technologies used in the crucial field of object recognition, which is part of artificial intelligence and computer vision. In the first chapter, we examine the essential concepts of object detection and tracking, as well as their many applications in various sectors.

The second chapter presents a State of the Art (SOTA), describing the best available techniques and algorithms. It covers the main categories of algorithms, the functioning of neural networks, as well as the various models used such as R-CNN, YOLO and SSD, in addition to the associated Deep Learning frameworks.

The third chapter focuses on the in-depth analysis and design of the system, identifying specific needs and proposing an appropriate architecture.

Finally, the fourth chapter details the practical implementation of the application, covering the technological choices, the implementation steps as well as the results obtained.

This report aims to provide an in-depth understanding of modern object recognition techniques and their practical application, highlighting the challenges encountered and the solutions provided throughout the process.

# TABLE DES MATIERES

|  |    |
|--|----|
| REMERCIEMENTS .....  | 1  |
| DÉDICACES.....   | 2  |
| RÉSUMÉ .....   | 3  |
| ABSTRACT .....   | 4  |
| TABLE DES MATIERES .....   | 5  |
| LISTE DES FIGURES.....   | 6  |
| LISTE DES ABRÉVIATIONS .....                                       | 7  |
| INTRODUCTION .....   | 8  |
| CHAPITRE 1 : La reconnaissance des objets .....                    | 9  |
| Introduction.....  | 9  |
| I. Détection et suivi d'objets .....                               | 9  |
| II. Domaines d'applications.....                                   | 10 |
| III. L'Intelligence Artificielle et la détection des objets .....  | 13 |
| Conclusion .....   | 13 |
| CHAPITRE2 : Etat de l'art .....                                    | 14 |
| Introduction.....  | 14 |
| I. Catégories d'algorithmes.....                                   | 14 |
| II. Fonctionnement d'un réseau de neurones .....                   | 15 |
| III. Les différents types de modèles des réseaux de neurones ..... | 16 |
| IV. La fonction d'activation .....                                 | 20 |
| V. La famille de modèles R-CNN .....                               | 20 |
| VI. La famille de modèles YOLO (You Only Look Once) .....          | 22 |
| VII. SSD (Single Shot Detector).....                               | 27 |
| VIII. Les frameworks de Deep Learning .....                        | 28 |
| Conclusion .....   | 29 |
| CHAPITRE 3 : Analyse et conception .....                           | 30 |
| Introduction.....  | 30 |
| I. Analyse des besoins du système .....                            | 30 |
| II. Conception .....   | 31 |
| III. Architecture du système .....                                 | 38 |
| Conclusion .....   | 39 |

|   |    |
|---|----|
| CHAPITRE 4 : Implémentation de l'application..... | 40 |
| Introduction.....                                 | 40 |
| I.    Choix des outils et technologies.....       | 40 |
| II.   Comment exécuter le code ?.....             | 45 |
| III.  Résultats.....                              | 46 |
| Conclusion .....                                  | 48 |
| CONCLUSION .....                                  | 49 |
| REFERENCES .....                                  | 50 |

# LISTE DES FIGURES

|  |    |
|--|----|
| Figure 1 : Infrastructures de transport .....  | 10 |
| Figure 2 : OD dans la fabrication des produits .....                                     | 11 |
| Figure 3 : Voitures autonomes .....  | 11 |
| Figure 4 : OD dans l'agriculture .....   | 12 |
| Figure 5 : OD dans le domaine de santé.....  | 12 |
| Figure 6 : Un réseau neuronal convolutionnel typique .....                               | 17 |
| Figure 7 : Réseau de neurones récurrents.....  | 18 |
| Figure 8 : Architecture YOLO .....   | 22 |
| Figure 9: exemple YOLO de football .....   | 23 |
| Figure 10 : Blocs résiduels .....  | 24 |
| Figure 11 : Régression des boîtes englobantes (1) .....                                  | 25 |
| Figure 12 : Régression des boîtes englobantes (2).....                                   | 25 |
| Figure 13 : Intersection sur les Unions ou IOU .....                                     | 26 |
| Figure 14 : Logo StarUML .....   | 31 |
| Figure 15: Diagramme des cas d'utilisation.....  | 32 |
| Figure 16 : Diagramme de séquence pour l'inscription .....                               | 33 |
| Figure 17 : Diagramme de séquence pour l'authentification .....                          | 34 |
| Figure 18 : diagramme de séquences concernant les vidéos YouTube .....                   | 35 |
| Figure 19 : diagramme de séquences concernant la détection des objets en temps réel..... | 36 |
| Figure 20 : Diagramme de séquence pour la détection des objets.....                      | 37 |
| Figure 21 : Structure des répertoires et des fichiers de l'application Web .....         | 39 |
| Figure 22 : Logo HTML, CSS et JS .....   | 40 |
| Figure 23 : Logo python.....   | 41 |
| Figure 24: Logo Stremlit .....   | 41 |
| Figure 25 : Logo YOLO.....   | 43 |
| Figure 26 : Page de connexion.....   | 46 |
| Figure 27 : Page d'inscription .....   | 46 |
| Figure 28 : exception lors de non Connexion de la page HOME .....                        | 47 |
| Figure 29 : Page About .....   | 47 |
| Figure 30 : page HOME.....   | 48 |



# LISTE DES ABRÉVIATIONS

| Abréviation | Signification                              |
|-------------|--|
| BDD         | Base de données                            |
| BPTT        | Backpropagation Through Time               |
| CNN         | Réseaux de neurones convolutifs            |
| CNTK        | Microsoft Cognitive Toolkit                |
| CSS         | Cascading Style Sheets                     |
| FCOS        | Fully Convolutional One-Stage Detector     |
| FPN         | Feature Pyramid Networks                   |
| FST         | Faculté des sciences et techniques         |
| GAN         | Generative Adversarial Networks            |
| GRU         | Gated Recurrent Unit                       |
| HTML        | HyperText Markup Language                  |
| HTTP        | Protocole de transfert hypertexte          |
| IA          | artificial intelligence                    |
| JS          | JavaScript                                 |
| LSTM        | Long Short-Term Memory                     |
| OD          | Object detection                           |
| ResNet      | Residual Networks                          |
| ReLU        | Rectified Linear Unit                      |
| RoI         | Regions of Interest                        |
| RPN         | Region Proposal Networks                   |
| R-CNN       | Region-based Convolutional Neural Networks |
| RNN         | Réseaux de neurones récurrents             |
| RSI         | Réseau et système informatique             |
| SOTA        | State Of The Art                           |
| SSD         | Single Shot Detector                       |
| tanh        | Tangente hyperbolique                      |
| UML         | Unified Modeling Language                  |
| VAE         | Variational Autoencoders                   |
| VGG         | Very Deep Convolutional Networks           |
| WSGI        | Web Server Gateway Interface               |
| YOLO        | You Only Look Once                         |

# INTRODUCTION

La détection d'objets est une discipline cruciale de la vision par ordinateur et de l'intelligence artificielle, visant à identifier et localiser des objets spécifiques dans des images ou des vidéos. Ce domaine trouve des applications diverses et croissantes dans des domaines tels que la sécurité, la surveillance, la robotique, l'automobile autonome, et bien d'autres. En effet, la capacité des machines à reconnaître et à comprendre leur environnement visuel est fondamentale pour leur interaction efficace et intelligente avec le monde physique.

La détection d'objets revêt une importance croissante dans notre société numérique moderne. Des applications telles que la sécurité urbaine avec la surveillance vidéo, la gestion intelligente des stocks dans les entrepôts, la conduite autonome dans l'industrie automobile, et même l'assistance à la chirurgie médicale bénéficient toutes des avancées dans ce domaine. En permettant aux machines de percevoir leur environnement de manière semblable à celle des humains, la détection d'objets ouvre la voie à des technologies plus sûres, plus efficaces et plus autonomes.

Ce projet vise à développer une application de détection d'objets en utilisant Python, en explorant diverses méthodes et technologies disponibles pour atteindre cet objectif, notamment l'utilisation de bibliothèques populaires telles que OpenCV pour le traitement d'images, et TensorFlow ou PyTorch pour l'apprentissage profond. Nous examinerons également des modèles de détection d'objets pré-entraînés comme YOLO (You Only Look Once) et Faster R-CNN, tout en évaluant la possibilité de créer nos propres modèles adaptés à nos besoins spécifiques.

# CHAPITRE 1 :

## La reconnaissance des objets

### Introduction

La reconnaissance d'objets en vision par ordinateur vise à identifier et localiser des entités spécifiques dans des images ou des vidéos, crucial pour des applications telles que la sécurité, la surveillance et la robotique.

### I. Détection et suivi d'objets

#### a. Détection d'objets

La détection d'objets est un processus utilisé pour identifier et localiser des objets dans une image ou une vidéo. Cette tâche consiste non seulement à reconnaître le type d'objet présent, mais également à dessiner un cadre de délimitation autour de celui-ci, ce qui permet de déterminer son emplacement exact.

Les algorithmes de détection d'objets utilisent généralement des modèles d'apprentissage profond, tels que YOLO, Faster R-CNN et SSD, pour détecter des objets dans des images ou des images vidéo.

Ces modèles utilisent des réseaux neuronaux convolutionnels (CNN) pour extraire des caractéristiques des données d'entrée, puis utilisent ces caractéristiques pour classer et localiser les objets.

#### b. Suivi d'objets

Le suivi d'objets, quant à lui, consiste à estimer l'état d'un objet, comme sa position, sa vitesse et son orientation, au fil du temps.

Cela nécessite d'analyser les informations temporelles de la vidéo et d'associer les objets détectés d'une image à l'autre. Le suivi d'objets est particulièrement important dans les applications impliquant des objets en mouvement, comme la surveillance ou les véhicules autonomes.

## II. Domaines d'applications

La détection d'objets peut être utilisée pour tout problème où vous devez savoir si une image ou une vidéo contient un ou plusieurs objets spécifiques et où ces objets se trouvent. Autrement dit utiliser la détection d'objets lorsque vous devez vérifier si quelque chose est présent dans une image ou une vidéo.

En raison de cette large utilité, la vision par ordinateur a trouvé des utilisations dans de nombreux secteurs ;

### a. Infrastructures de transport

Les fournisseurs de transport peuvent utiliser la détection d'objets pour assurer la sécurité sur leurs réseaux, par exemple, en identifiant des obstacles sur une voie ferrée ou pour vérifier si quelqu'un est présent dans une zone restreinte sur un chantier. En effet, la détection d'objets et la vision par ordinateur plus largement peuvent jouer un rôle clé dans la sécurité des transports.

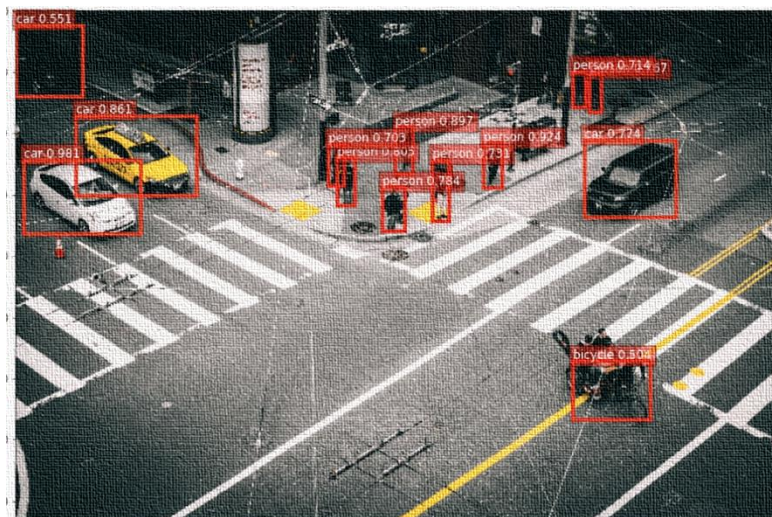


Figure 1 : Infrastructures de transport

### b. Surveillance et sécurité

Identification des personnes, des véhicules et des activités suspectes dans les flux vidéo des caméras de surveillance.

### c. Fabricants de produits alimentaires

Les fabricants de produits alimentaires peuvent utiliser la détection d'objets pour garantir l'intégrité des produits avant qu'ils ne soient emballés et expédiés. Par exemple, vous pouvez utiliser la vision pour vous assurer que la crème glacée n'a pas fondu sur un tapis roulant pendant la production.

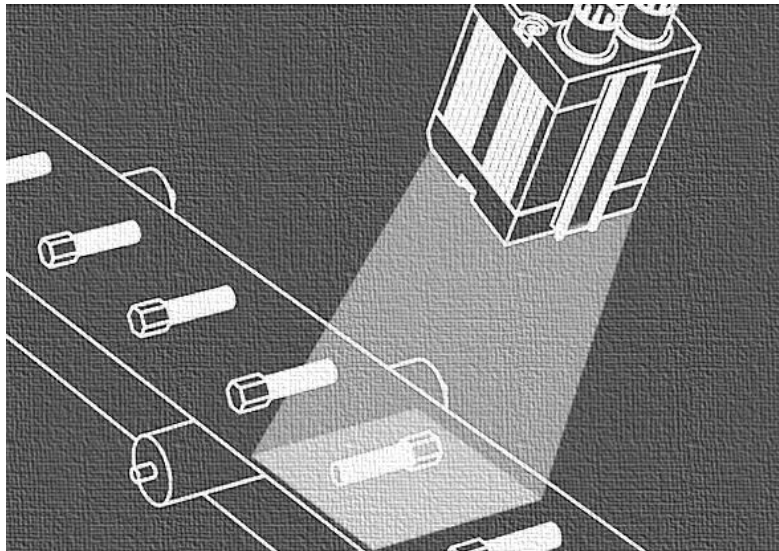


Figure 2 : OD dans la fabrication des produits

#### d. Véhicules autonomes

Au cours de la dernière décennie, des travaux importants ont été réalisés pour créer des voitures qui se conduisent toutes seules en toute sécurité. Derrière cette innovation se cache la vision par ordinateur. Les voitures doivent être capables de détecter certains objets (piétons, feux de circulation, cônes de signalisation, par exemple) afin de pouvoir prendre des décisions sur ce qu'elles doivent faire et où aller.

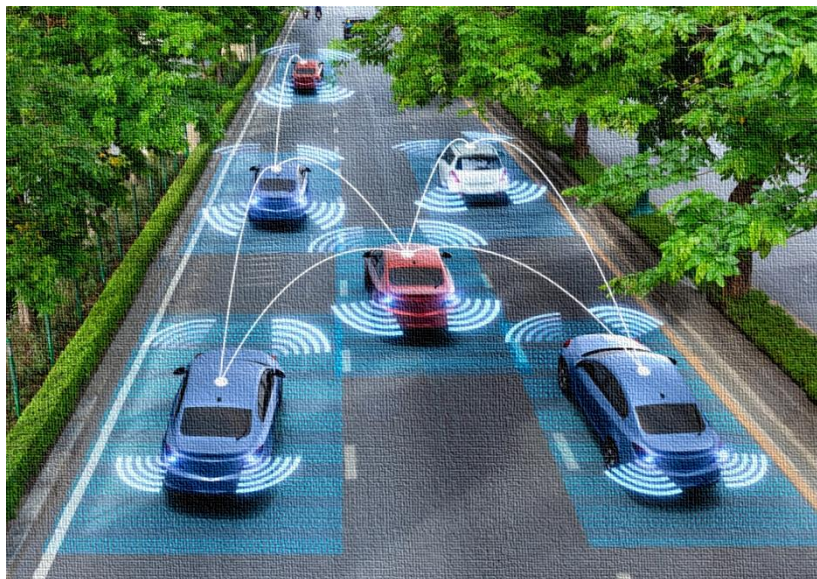


Figure 3 : Voitures autonomes



## e. Agriculture

L'intelligence artificielle et la robotique jouent un rôle majeur dans l'agriculture moderne. Les robots de récolte sont des robots basés sur la vision qui ont été introduits pour remplacer la cueillette manuelle des fruits et légumes.

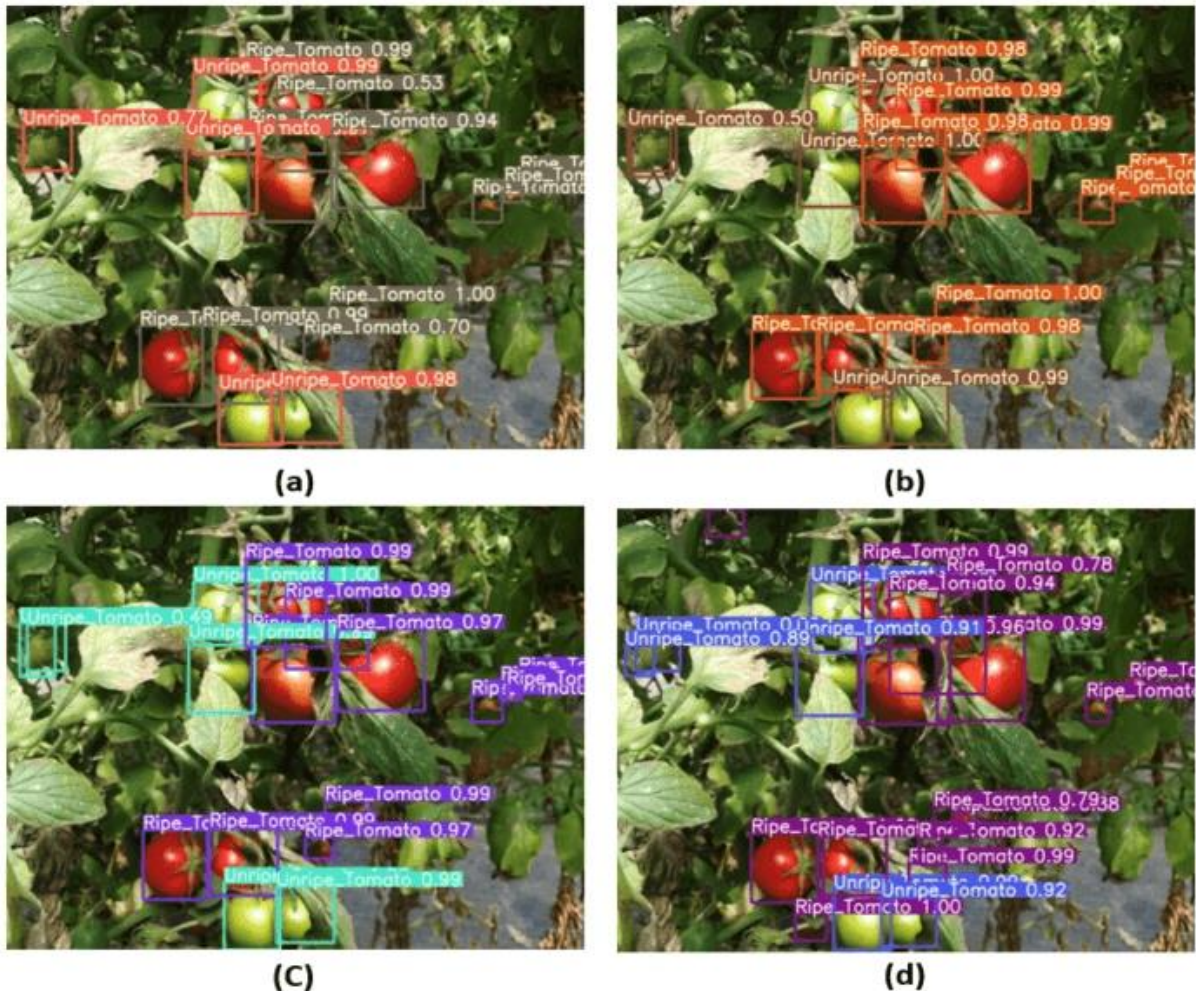


Figure 4 : OD dans l'agriculture

## f. Imagerie médicale

Localisation et classification des anomalies dans les images médicales telles que les radiographies, les IRM ou les scanners CT.

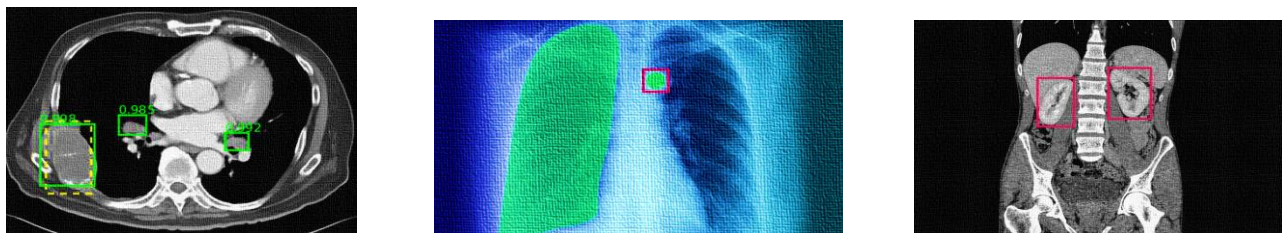


Figure 5 : OD dans le domaine de santé

### III. L'Intelligence Artificielle et la détection des objets

L'intelligence artificielle (IA) offre diverses approches pour la reconnaissance d'objets, incluant des techniques de segmentation d'images, de description d'images et d'apprentissage profond.

#### 1. Segmentation d'images

Cette méthode divise une image en segments ou en régions pour identifier des objets distincts. Des techniques comme la segmentation sémantique et la segmentation d'instance sont utilisées pour améliorer la précision et la granularité de la détection d'objets.

#### 2. Description d'images

Pour extraire des caractéristiques pertinentes d'une image, des techniques telles que l'extraction de caractéristiques locales (par exemple, des coins ou des contours) et globales (par exemple, des histogrammes de couleurs ou des textures) sont utilisées. Ces caractéristiques sont essentielles pour entraîner des modèles de détection d'objets à reconnaître et localiser des objets spécifiques.

#### 3. Deep Learning

L'apprentissage profond est une approche puissante et moderne pour la reconnaissance d'objets, où des réseaux neuronaux profonds sont entraînés sur de vastes ensembles de données annotées pour apprendre automatiquement à identifier des objets dans des images ou des vidéos. Des architectures comme les réseaux de neurones convolutionnels (CNN) et des modèles spécifiques comme YOLO (You Only Look Once) et Faster R-CNN sont largement utilisés pour des tâches de détection d'objets complexes.

### Conclusion

Ce chapitre a introduit la reconnaissance d'objets comme cruciale pour la vision par ordinateur et l'intelligence artificielle, en explorant les fondements de la reconnaissance des formes et les techniques telles que la segmentation d'image et la description d'image qui sont des processus importants qui alimentent le deep learning en fournissant des données préparées et des caractéristiques pertinentes, ce qui permet aux réseaux neuronaux d'apprendre efficacement à détecter et à reconnaître des objets dans les images.

# CHAPITRE 2 : Etat de l'art

## Introduction

La détection d'objets est essentielle en vision par ordinateur, bénéficiant largement des avancées du Deep Learning. Ce chapitre explore l'état de l'art des algorithmes, cruciaux pour identifier et localiser précisément des objets dans images et vidéos. Nous examinons le fonctionnement des réseaux de neurones, en particulier les CNN et les RNN, ainsi que des modèles avancés comme R-CNN, YOLO, et SSD. En outre, nous analysons les fonctions d'activation clés et comparons les frameworks comme TensorFlow, PyTorch et autres, soulignant leur rôle dans l'avancement des technologies de détection d'objets.

## I. Catégories d'algorithmes

### 1. Les détecteurs à deux étapes

Les détecteurs à deux étapes, tels que Fast R-CNN, Mask R-CNN, FPN (Feature Pyramid Networks), etc., suivent une approche en deux phases distinctes pour la détection d'objets.

**Phase 1 - Propositions :** Un modèle est utilisé pour générer des régions d'intérêt (RoI - Regions of Interest) potentielles dans l'image. Ces régions sont souvent générées par des méthodes comme Selective Search ou des réseaux spécialisés (RPN - Region Proposal Networks) et sont basées sur des caractéristiques visuelles.

**Phase 2 - Refinement :** Les régions d'intérêt sont ensuite utilisées pour extraire des caractéristiques spécifiques à chaque région et effectuer la classification et la localisation finales des objets à l'intérieur de ces régions.

**Avantages :** Cette approche permet généralement une meilleure précision dans la localisation des objets grâce à une deuxième phase de raffinement des prédictions. Elle est souvent préférée dans les tâches où la précision est critique.

**Limitations :** Plus complexes et plus lentes que les détecteurs à une étape, les détecteurs à deux étapes nécessitent généralement plus de ressources en termes de calcul et de mémoire.



## 2. Les détecteurs à une étape

Les détecteurs à une étape, comme RetinaNet, SSD (Single Shot Multibox Detector), FCOS (Fully Convolutional One-Stage Detector), YOLO (You Only Look Once), etc., se distinguent par leur capacité à effectuer la détection d'objets en une seule passe de l'image à travers le réseau de neurones.

**Approche :** Ces détecteurs considèrent l'ensemble de l'image simultanément plutôt que de diviser le processus de détection en plusieurs étapes distinctes. Ils prédisent directement les boîtes englobantes (bounding boxes) et les classes des objets dans une grille d'ancres ou à travers une prédiction dense de l'image.

**Avantages :** Généralement plus rapides et efficaces en termes de calcul, les détecteurs à une étape sont bien adaptés aux applications nécessitant une détection en temps réel ou à grande échelle.

**Limitations :** Ils peuvent parfois manquer de précision dans la localisation fine des objets, en particulier pour les objets de petite taille ou les scènes complexes avec des chevauchements.

## II. Fonctionnement d'un réseau de neurones

### 1. Principes de base des réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des modèles informatiques inspirés du fonctionnement du cerveau humain, conçus pour apprendre à partir de données et effectuer des tâches complexes. Leur fonctionnement repose sur des unités interconnectées appelées neurones, organisées en couches. Chaque neurone reçoit des entrées, applique une fonction d'activation à la somme pondérée de ces entrées, et transmet le résultat à la couche suivante. Cette architecture permet au réseau de neurones de capturer des modèles et des relations complexes dans les données, ce qui est crucial pour la détection d'objets.

### 2. Architecture typique d'un réseau de neurones pour la détection d'objets

L'architecture typique d'un réseau de neurones pour la détection d'objets commence généralement par une couche **d'entrée** où les images sont introduites sous forme de pixels. Ces données passent ensuite à travers plusieurs couches de traitement, telles que des couches **convolutives** pour extraire des caractéristiques visuelles, des couches de **pooling** pour réduire la dimension des données, et des couches **entièrement connectées** pour la classification finale des objets détectés.

### III. Les différents types de modèles des réseaux de neurones

#### 1. Les réseaux de neurones convolutifs (CNN)

##### a. Inspiration biologique et évolution des CNN

Les réseaux de neurones convolutifs (CNN) sont inspirés du fonctionnement du cortex visuel biologique et ont été développés pour traiter efficacement des tâches de reconnaissance d'objets complexes.

##### b. Opération de convolution et son rôle dans la détection d'objets

L'opération de convolution est au cœur des réseaux de neurones convolutifs (CNN). Elle consiste à appliquer des filtres (ou kernels) à travers une image pour extraire des caractéristiques locales importantes, telles que les bords, les textures et les motifs. Cette technique permet au CNN de détecter des motifs récurrents et significatifs dans différentes parties de l'image, facilitant ainsi la reconnaissance et la localisation d'objets dans des scènes complexe

##### c. Couches convolutives et leur importance

Les couches convolutives sont essentielles dans un CNN car elles sont responsables de l'extraction des caractéristiques à différentes échelles spatiales. Chaque couche convolutive utilise plusieurs filtres pour générer des cartes d'activation qui capturent des informations spécifiques de l'image, ce qui permet au réseau de construire une représentation hiérarchique des données visuelles. Ces couches jouent un rôle critique dans la capacité du CNN à reconnaître des motifs et des structures complexes dans les images, améliorant ainsi la précision de la détection d'objets.

##### d. Couches de pooling pour la réduction de dimension

Les couches de pooling sont utilisées dans les CNN pour réduire la dimension spatiale des cartes d'activation générées par les couches convolutives.

Typiquement, le pooling se fait à travers des opérations telles que MaxPooling ou AveragePooling, qui agrègent l'information en conservant les caractéristiques les plus importantes et en réduisant le nombre de paramètres. Cette réduction de dimensionnalité permet de rendre le réseau plus efficace en termes de calcul tout en préservant les informations essentielles pour la détection d'objets.

### e. Utilisation des perceptrons dans les CNN

Les perceptrons, ou neurones entièrement connectés, sont utilisés dans les couches finales des CNN pour effectuer la classification des objets détectés. Chaque neurone dans cette couche reçoit en entrée les caractéristiques extraites par les couches convolutives et les utilise pour calculer la probabilité que l'image d'entrée appartienne à chaque classe d'objets prédéfinie. Cette étape finale de classification permet de transformer les caractéristiques visuelles apprises par le réseau en décisions précises sur la présence et la localisation des objets dans une image.

### f. Exemples de réseaux convolutifs célèbres

Quelques exemples célèbres de réseaux convolutifs incluent VGG (Very Deep Convolutional Networks), connu pour sa profondeur et sa simplicité architecturale, et ResNet (Residual Networks), qui introduit des connexions résiduelles pour faciliter l'entraînement de réseaux encore plus profonds en réduisant le problème de la disparition du gradient.

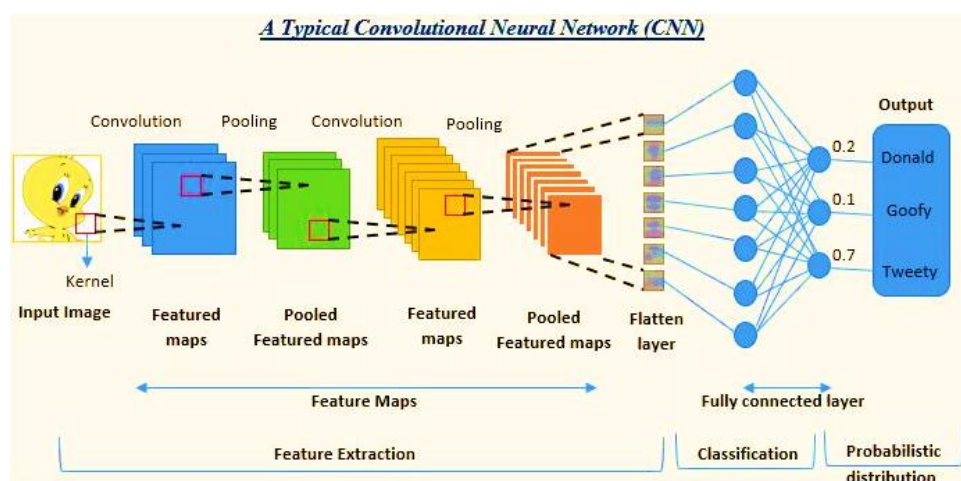


Figure 6 : Un réseau neuronal convolutionnel typique

## 2. Réseaux de neurones récurrents (RNN)

### a. Définition et principe de fonctionnement des RNN

Les réseaux de neurones récurrents (RNN) sont une classe spéciale de réseaux de neurones conçus pour traiter des données séquentielles où l'ordre et la dépendance temporelle sont importants. Contrairement aux réseaux de neurones feedforward traditionnels, les RNN possèdent des connexions récurrentes qui leur permettent de maintenir un état interne, ou mémoire, qui évolue au fur et à mesure que de nouvelles entrées sont introduites. Cela leur permet de modéliser des séquences de données et d'exploiter la structure temporelle des données, ce qui est particulièrement utile pour la détection d'objets dans des vidéos et des séquences d'images.

Les RNN sont caractérisés par leur capacité à traiter des données d'entrée séquentielles, où chaque étape de traitement dépend non seulement de l'entrée actuelle, mais aussi de l'état caché (hidden state) provenant de l'étape de traitement précédente. Cette boucle récurrente permet aux RNN de capturer les dépendances à long terme dans les données séquentielles, ce qui est essentiel pour des tâches telles que la prédiction et la classification de séquences temporelles.

#### b. Mécanismes d'apprentissage dans les RNN

L'apprentissage dans les RNN se fait principalement par rétropropagation à travers le temps (Backpropagation Through Time, BPTT), où le réseau est entraîné à minimiser une fonction de perte en ajustant les weights des connexions récurrentes et des connexions feedforward. Cependant, les RNN traditionnels peuvent souffrir du problème du gradient qui disparaît ou explose sur de longues séquences, limitant ainsi leur capacité à capturer des dépendances à long terme. Pour surmonter cette limitation, des architectures plus avancées comme les LSTM (Long Short-Term Memory) et les GRU (Gated Recurrent Unit) ont été développées, introduisant des mécanismes de portes pour mieux gérer l'information à travers le temps.

#### c. Applications des RNN dans la détection d'objets

Les RNN sont largement utilisés dans la détection d'objets dans des vidéos et des séquences d'images, où la compréhension contextuelle et la dépendance temporelle sont essentielles. Par exemple, dans le suivi d'objets, les RNN peuvent être utilisés pour prédire la trajectoire future d'un objet en mouvement en se basant sur les observations précédentes. De plus, les RNN peuvent être intégrés dans des architectures hybrides avec des CNN pour améliorer la précision des systèmes de détection d'objets dans des environnements dynamiques et complexes.

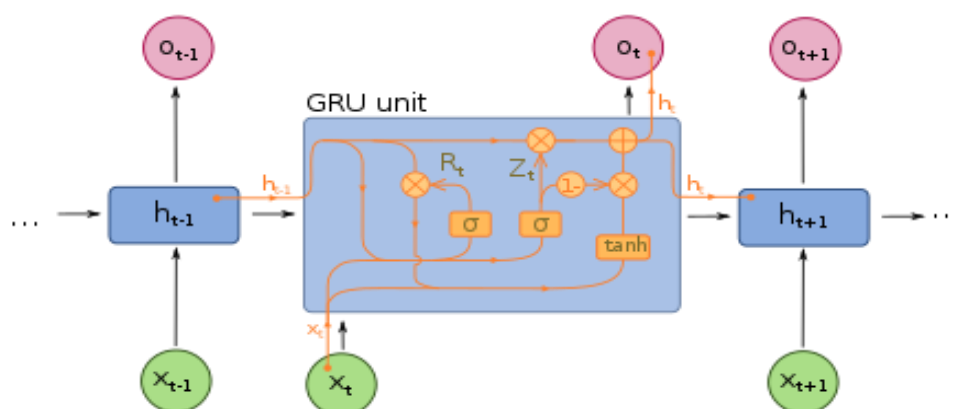


Figure 7 : Réseau de neurones récurrents

### 3. Modèles génératifs profonds

Les modèles génératifs profonds sont des architectures de réseau de neurones capables de générer de nouvelles données qui ressemblent à celles d'un ensemble d'apprentissage donné. Deux des modèles génératifs profonds les plus couramment utilisés sont les GAN (Generative Adversarial Networks) et les VAE (Variational Autoencoders).

#### a. Présentation des modèles génératifs profonds

##### **-GAN**

Les GAN se composent de deux réseaux neuronaux adversariaux : un générateur et un discriminateur. Le générateur tente de créer des données réalistes pour tromper le discriminateur, qui lui cherche à distinguer les données générées des données réelles. Ce processus d'entraînement concurrentiel permet au générateur de produire des données de sortie de plus en plus réalistes au fil du temps.

##### **-VAE**

Les VAE, quant à eux, sont des modèles probabilistes basés sur l'idée d'un autoencodeur, où le réseau est entraîné pour apprendre une représentation latente des données en maximisant la probabilité que les données générées soient similaires aux données d'entrée. Les VAE introduisent une distribution probabiliste sur l'encodage latente, permettant une génération de données plus contrôlée et diversifiée par rapport aux GAN.

#### b. Utilisation potentielle pour la détection d'objets

Les modèles génératifs profonds, comme les GAN et les VAE, ont un potentiel significatif pour la génération et la détection d'objets. Les GAN peuvent être utilisés pour générer des images réalistes d'objets à partir de descriptions textuelles ou de schémas de données, tandis que les VAE peuvent être utilisés pour apprendre des représentations latentes des objets qui peuvent ensuite être utilisées pour la classification ou la détection d'objets dans des scènes complexes.

Bref, les modèles génératifs profonds offrent une approche novatrice pour la génération et la détection d'objets en vision par ordinateur, exploitant la capacité des réseaux de neurones à générer des données réalistes et à apprendre des représentations robustes et diversifiées des objets dans les données visuelles.

## IV. La fonction d'activation

### 1. Importance des fonctions d'activation dans les réseaux de neurones

Les fonctions d'activation sont essentielles car elles introduisent la non-linéarité dans les couches cachées des réseaux de neurones. Cela permet aux réseaux de neurones d'apprendre des représentations complexes des données, ce qui est crucial pour des tâches comme la détection d'objets où les relations entre les caractéristiques visuelles peuvent être hautement non linéaires. En l'absence de fonctions d'activation, les réseaux de neurones se réduiraient à une série de transformations linéaires, limitant ainsi leur capacité à modéliser efficacement des données complexes et à généraliser.

### 2. Caractéristiques recherchées et rôles spécifiques dans la détection d'objets

Pour la détection d'objets, les fonctions d'activation doivent être capables de gérer des informations variées et complexes extraites des données visuelles. Une fonction d'activation efficace doit donc présenter plusieurs caractéristiques désirables, telles qu'une non-linéarité importante pour capturer des relations non triviales, une capacité à stabiliser l'entraînement du réseau en évitant le problème du gradient qui disparaît (vanishing gradient), et une efficacité computationnelle pour des réseaux de grande taille utilisés dans les architectures modernes de détection d'objets.

## V. La famille de modèles R-CNN

### 1. Evolution et améliorations des modèles R-CNN pour la détection d'objets

Les modèles R-CNN (Region-based Convolutional Neural Networks) ont révolutionné la détection d'objets en introduisant l'apprentissage profond dans ce domaine. Depuis leur introduction en 2013, ils ont connu une évolution remarquable grâce à des améliorations continues, les rendant plus précis, plus rapides et plus efficaces.

*Les étapes clés de cette évolution incluent :*

#### **-R-CNN (2013) :**

Le modèle original proposait une approche en deux étapes : Génération de régions candidates par recherche sélective, puis classification et localisation des objets au sein de ces régions par un réseau neuronal convolutif (CNN).

**-Fast R-CNN (2015) :** Cette version partageait la couche convolutive entre les étapes de génération de régions et de classification, ce qui accélérail considérablement le processus.

**-Faster R-CNN (2015) :** Un réseau neuronal à régions (RPN) a été introduit pour générer des régions candidates de manière plus efficace, menant à une détection d'objets encore plus rapide et précise.

**-Mask R-CNN (2017) :** Cette extension a permis la segmentation d'instances, délimitant les contours précis de chaque objet détecté.

**-Développements ultérieurs :** Des recherches continues ont donné naissance à des variantes comme FPN (Feature Pyramid Networks) et Cascade R-CNN, qui améliorent encore la précision et la robustesse de la détection d'objets.

## 2. Présentation de R-CNN, Fast R-CNN et Faster R-CNN

### a. R-CNN

**-Architecture :** Deux étapes distinctes : génération de régions candidates et classification/localisation des objets.

**-Avantages :** Précision initiale élevée.

**-Limites :** Vitesse lente due à la recherche sélective.

### b. Fast R-CNN

**-Amélioration :** Partage de la couche convolutive entre les étapes, ce qui accélère le processus.

**-Avantages :** Gain de vitesse significatif par rapport au R-CNN.

**-Limites :** Génération de régions candidates moins précise que Faster R-CNN.

### c. Faster R-CNN

Introduction du RPN (Region Proposal Network) : Génère des régions candidates de manière plus efficace et précise.

**-Avantages :** Vitesse et précision accrues par rapport aux versions précédentes.

**-Limites :** Plus complexe à implémenter que les modèles R-CNN et Fast R-CNN.





L'architecture fonctionne comme suit :

1. Redimensionne l'image d'entrée en 448x448 avant de passer par le réseau convolutionnel.
2. Une convolution 1x1 est d'abord appliquée pour réduire le nombre de canaux, qui est ensuite suivie d'une convolution 3x3 pour générer une sortie cubique.
3. La fonction d'activation sous le capot est ReLU, à l'exception de la couche finale, qui utilise une fonction d'activation linéaire.
4. Certaines techniques supplémentaires, telles que la normalisation par lots et le dropout, régularisent respectivement le modèle et l'empêchent de sur-adapter.

### 3. Comment ça marche ?

Imaginez que vous avez créé une application YOLO qui détecte les joueurs et les ballons de football à partir d'une image donnée.

Comment obtenir l'image (B) à partir de l'image (A) ?

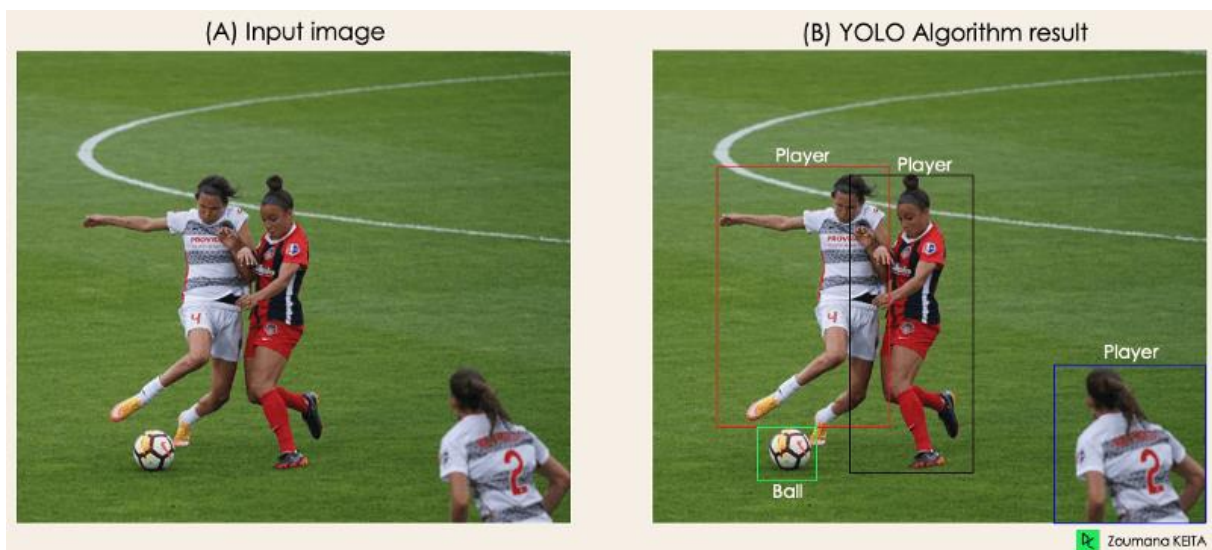


Figure 9: exemple YOLO de football

L'algorithme fonctionne sur la base des quatre approches suivantes :

- a. Blocs résiduels**
- b. Régression de la boîte englobante**
- c. Intersection Over Unions ou IOU en abrégé**
- d. Suppression non maximale.**

Examinons de plus près chacune d'entre elles.

### a. Blocs résiduels

Cette première étape commence par diviser l'image originale (A) en cellules de grille NxN de forme égale, où N dans notre cas est 4 comme indiqué sur l'image de droite. Chaque cellule de la grille est chargée de localiser et de prédire la classe de l'objet qu'elle couvre, ainsi que la valeur de probabilité/confiance.

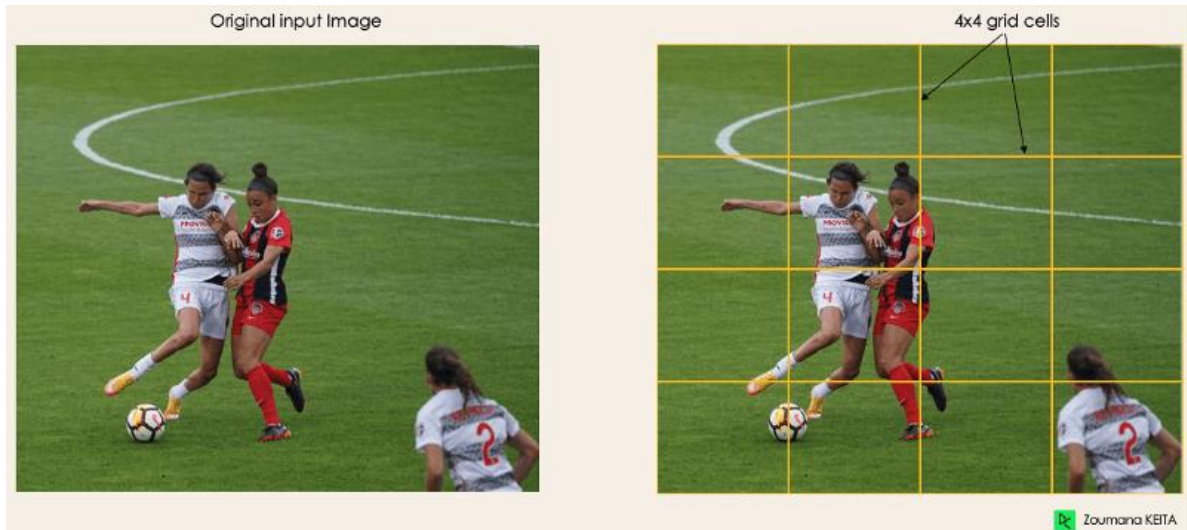


Figure 10 : Blocs résiduels

### b. Régression des boîtes englobantes

L'étape suivante consiste à déterminer les boîtes englobantes qui correspondent à des rectangles mettant en évidence tous les objets de l'image. Nous pouvons avoir autant de boîtes englobantes qu'il y a d'objets dans une image donnée

YOLO détermine les attributs de ces boîtes englobantes à l'aide d'un seul module de régression au format suivant, où Y est la représentation vectorielle finale de chaque boîte englobante.

$$Y = [pc, bx, by, bh, bw, c1, c2]$$

Ceci est particulièrement important lors de la phase d'apprentissage du modèle.

pc correspond au score de probabilité de la grille contenant un objet. Par exemple, toutes les grilles en rouge auront un score de probabilité supérieur à zéro. L'image de droite est la version simplifiée puisque la probabilité de chaque cellule jaune est nulle (non significative).

bx, by sont les coordonnées x et y du centre du cadre de délimitation par rapport à la cellule de grille enveloppante.

c1 et c2 correspondent aux deux classes Player et Ball. Nous pouvons avoir autant de classes que votre cas d'utilisation l'exige.



✦ Bounding box centers

Figure 12 : Régression des boîtes englobantes (2)

### c. Intersection sur les Unions ou IOU

La plupart du temps, un seul objet dans une image peut avoir plusieurs candidats de grille pour la prédiction, même si tous ne sont pas pertinents. Le but de l'IOU (une valeur comprise entre 0 et 1) est d'éliminer ces cases de grille pour ne conserver que celles qui sont pertinentes.

#### **Voici la logique qui sous-tend cela :**

L'utilisateur définit son seuil de sélection d'IOU, qui peut être, par exemple, 0,5.

Ensuite, YOLO calcule l'IOU de chaque cellule de la grille qui est la zone d'intersection divisée par la zone d'union.

Enfin, il ignore la prédiction des cellules de la grille ayant un  $IOU \leq \text{seuil}$  et considère celles ayant un  $IOU > \text{seuil}$ .

Vous trouverez ci-dessous une illustration de l'application du processus de sélection de grille à l'objet en bas à gauche. Nous pouvons observer que l'objet avait à l'origine deux candidats de grille, puis que seule la « Grille 2 » a été sélectionnée à la fin.

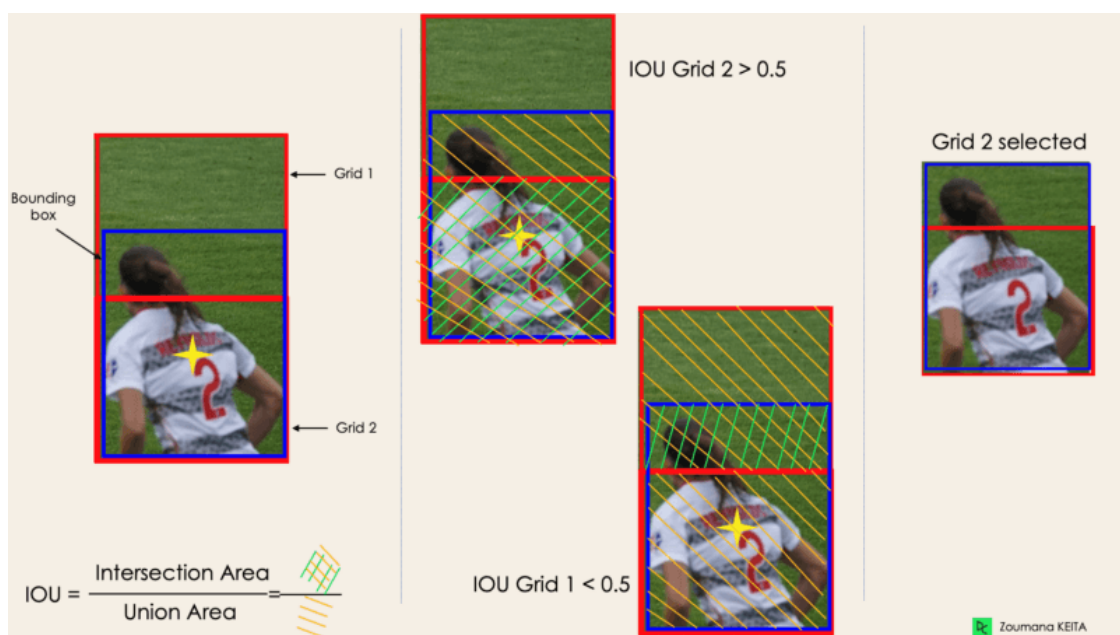


Figure 13 : Intersection sur les Unions ou IOU

### d. Suppression non maximale ou NMS

La définition d'un seuil pour l'IOU n'est pas toujours suffisante car un objet peut avoir plusieurs cases avec un IOU au-delà du seuil, et laisser toutes ces cases peut inclure du bruit. C'est ici que nous pouvons utiliser NMS pour ne conserver que les cases avec le score de probabilité de détection le plus élevé.



## VII. SSD (Single Shot Detector)

### 1. Présentation de l'approche SSD pour la détection d'objets

SSD (Single Shot Detector) est une architecture de réseau neuronal convolutif (CNN) pour la détection d'objets qui se distingue par sa capacité à réaliser une détection en une seule passe. Contrairement aux approches par régions comme R-CNN qui s'appuient sur des propositions de régions candidates, SSD traite l'image entière simultanément pour prédire à la fois la classe et la localisation des objets.

L'approche SSD s'articule autour de deux éléments clés :

**Un réseau neuronal convolutif de base** : Extrait des caractéristiques d'images à différentes échelles spatiales.

**Couches de détection** : Prédissent les classes et les localisations des objets pour chaque cellule de la grille à différentes échelles spatiales.

Cette architecture permet à SSD de réaliser une détection d'objets d'une manière efficace et unifiée, ce qui lui confère plusieurs avantages :

**Vitesse** : SSD est beaucoup plus rapide que les approches par régions car il n'a pas besoin de générer de propositions de régions candidates.

**Précision** : SSD peut atteindre une précision comparable aux approches par régions, en particulier avec des architectures et des techniques d'entraînement plus récentes.

**Simplicité** : L'architecture de SSD est plus simple et plus facile à mettre en œuvre que les approches par régions.

### 2. Avantages par rapport aux approches traditionnelles basées sur des régions

Les approches basées sur des régions, telles que R-CNN, génèrent d'abord des propositions de régions candidates pour les objets potentiels, puis classifient et affinent les localisations au sein de ces régions. Cette approche en deux étapes peut s'avérer lente et inefficace.

SSD, en revanche, réalise la détection d'objets en une seule passe en prédisant directement les classes et les localisations des objets à partir de l'image entière. Cela offre plusieurs avantages :

**Vitesse accrue :** SSD peut traiter des images beaucoup plus rapidement que les approches par régions, ce qui le rend idéal pour les applications en temps réel.

**Meilleure détection de petits objets :** SSD est plus efficace pour détecter de petits objets que les approches par régions, car il n'est pas limité par la taille des propositions de régions candidates.

**Détection d'objets sans occlusion :** SSD peut détecter des objets même lorsqu'ils sont partiellement occlus par d'autres objets, ce qui est un défi pour les approches par régions.

## VIII. Les frameworks de Deep Learning

### 1. Présentation des principaux frameworks

Les frameworks de Deep Learning sont des outils essentiels pour le développement et le déploiement d'algorithmes d'apprentissage profond. Voici une présentation des principaux frameworks utilisés aujourd'hui :

#### a. Theano

- Theano est l'un des premiers frameworks de calcul numérique dédiés au Deep Learning.
- Développé par l'Université de Montréal, il offre une efficacité et une flexibilité considérables pour la construction de modèles complexes.

#### b. TensorFlow

- Développé par Google, TensorFlow est devenu l'un des frameworks les plus populaires pour le Deep Learning.
- Il offre une architecture flexible pour la création de réseaux neuronaux à grande échelle et prend en charge une large gamme de déploiements, y compris sur des systèmes distribués.

#### c. Keras

- Keras est une interface haut niveau permettant de construire et d'entraîner des modèles Deep Learning.
- Initialement conçu pour fonctionner au-dessus de TensorFlow, Keras offre une facilité d'utilisation et une rapidité de prototypage tout en maintenant une grande puissance.

#### d. PyTorch

- PyTorch est un framework développé par Facebook qui combine souplesse et rapidité.
- Connu pour sa facilité d'utilisation et sa capacité à construire des graphes computationnels dynamiques, PyTorch est largement utilisé pour la recherche et la production.

#### e. CNTK (Microsoft Cognitive Toolkit)

- Développé par Microsoft, le CNTK est un framework open-source qui offre des performances élevées grâce à son optimisation pour les systèmes distribués.

#### f. Torch

- Torch est un framework utilisé principalement pour le Deep Learning et l'apprentissage automatique.
- Connu pour sa rapidité et son efficacité, il offre une base solide pour la recherche en intelligence artificielle.

#### g. Darknet

- Darknet est un framework open-source développé pour effectuer des tâches de vision par ordinateur et de reconnaissance d'objets.
- Il est reconnu pour sa rapidité et sa capacité à gérer des réseaux de neurones convolutionnels (CNN) efficacement.

### Conclusion

En conclusion, ce chapitre a dressé un panorama détaillé de l'état de l'art en détection d'objets par le biais des réseaux de neurones, en mettant en lumière les avancées clés et les techniques essentielles. Nous avons exploré les fondements des réseaux de neurones, les différents types de modèles adaptés à la détection d'objets, ainsi que les familles de modèles R-CNN, YOLO et SSD. Comprendre ces avancées est crucial pour développer efficacement des solutions dans ce domaine complexe et en rapide évolution, préparant ainsi le terrain pour les méthodes spécifiques abordées dans les prochains chapitres.

# CHAPITRE 3 :

## Analyse et conception

### Introduction

Ce chapitre présente l'analyse des besoins et la conception du système. L'objectif est de définir les fonctionnalités, les exigences et les spécifications techniques du système.

### I. Analyse des besoins du système

#### 1. Besoins fonctionnels

Les besoins fonctionnels de l'application de détection d'objets sont centrés sur les fonctionnalités essentielles que le système doit fournir pour répondre aux attentes des utilisateurs.

Ces besoins incluent :

**Détection des Objets et spécification :** L'application doit être capable de détecter et de reconnaître différents types d'objets dans des images ou des vidéos en temps réel et même reconnaître un objet spécifique .

**Classification des Objets :** Une fois détectés, les objets doivent être correctement classifiés en fonction de leur type (par exemple, véhicules, personnes, animaux, etc.).

**Localisation Précise :** Il est essentiel de fournir des informations précises sur la localisation spatiale des objets détectés afin de permettre une interaction efficace avec ces objets.

**Suivi des Objets :** Capacité à suivre les objets détectés dans le temps, même lorsque ceux-ci se déplacent dans le champ de vision de la caméra.

**Interface Utilisateur Intuitive :** Une interface conviviale et intuitive est nécessaire pour permettre aux utilisateurs de visualiser les résultats de la détection et d'interagir avec les objets détectés de manière efficace.

**Compatibilité Multiplateforme :** L'application doit pouvoir fonctionner sur différentes plateformes matérielles et logicielles couramment utilisées.

**Performance en Temps Réel :** La détection et la classification des objets doivent être effectuées en temps réel pour assurer une réponse rapide et efficace.



## 2. Besoins non fonctionnels

Les besoins non fonctionnels définissent les critères de performance, de sécurité, de convivialité et d'autres aspects qui ne sont pas directement liés aux fonctionnalités spécifiques du système.

Pour cette application de détection d'objets, les besoins non fonctionnels comprennent :

**Performance** : La détection des objets doit être rapide et efficace, avec des délais de traitement minimaux même lors de la détection simultanée de plusieurs objets.

**Précision** : La précision de la détection et de la classification des objets doit être élevée pour minimiser les erreurs et assurer une reconnaissance fiable des objets.

**Évolutivité** : Le système doit être conçu pour pouvoir évoluer et gérer efficacement une augmentation du nombre d'utilisateurs et de la complexité des scénarios de détection.

**Convivialité** : L'interface utilisateur doit être conviviale, avec des fonctionnalités intuitives pour faciliter l'utilisation par des utilisateurs de différents niveaux de compétence.

**Fiabilité** : Le système doit être robuste et fiable, minimisant les temps d'arrêt et assurant un fonctionnement continu dans des conditions variées.

**Intégration** : Capacité à s'intégrer facilement avec d'autres systèmes ou technologies existantes, tels que des plateformes de traitement de données ou des solutions de gestion.

## II. Conception

### 1. Logiciel de conception



**StarUML** est un logiciel de modélisation UML (Unified Modeling Language) qui permet aux développeurs de créer des diagrammes de modèles pour concevoir des systèmes logiciels. Il offre une interface conviviale et des fonctionnalités puissantes pour représenter visuellement les différents aspects d'un système logiciel, y compris la structure, le comportement, les interactions et les processus.

Figure 14 : Logo StarUML

## 2. Diagramme du cas d'utilisation

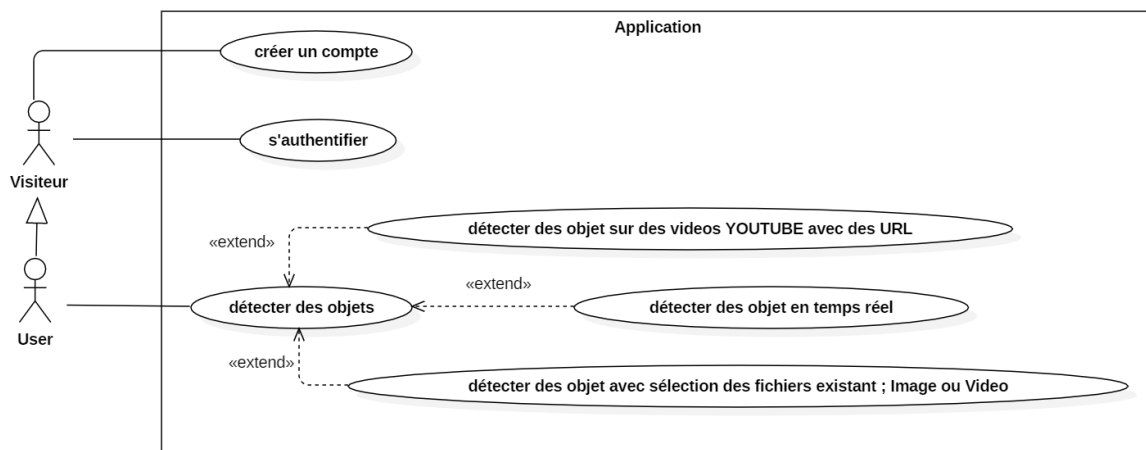


Figure 15: Diagramme des cas d'utilisation

### Acteurs :

**Visiteur** : La personne qui souhaite utiliser le système pour détecter des objets.

**Utilisateur** : Une personne authentifiée qui peut utiliser toutes les fonctionnalités du système.

### Cas d'utilisation :

**Créer un compte** : Ce cas d'utilisation permet à un visiteur de créer un compte utilisateur.

**S'authentifier** : Ce cas d'utilisation permet à un visiteur de s'authentifier dans le système.

**Détecter des objets en temps réel** : Ce cas d'utilisation permet à un utilisateur de détecter des objets en temps réel à l'aide de la caméra de son appareil.

**Détecter des objets avec sélection des fichiers existants** : Ce cas d'utilisation permet à un utilisateur de détecter des objets à partir de fichiers existants sur son appareil ; Image ou Vidéo.

**Détecter des objets sur un vidéos YOUTUBE avec URL** : Ce cas d'utilisation permet à un utilisateur de détecter des objets dans un vidéo YOUTUBE en saisissant un URL.

### 3. Diagramme de séquence

#### a. Création du compte

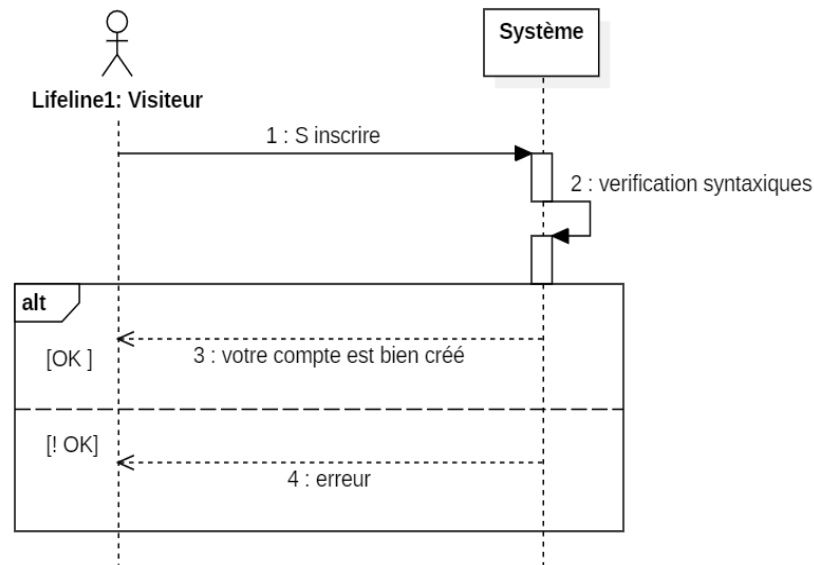


Figure 16 : Diagramme de séquence pour l'inscription

*Les participants impliqués dans ce processus sont :*

**Visiteur** : Représente l'être humain qui souhaite s'inscrire dans le système.

**Système** : Représente notre application avec la BDD.

*Étapes du processus d'inscription :*

**1-Le visiteur saisit son adresse e-mail et son mot de passe.** L'adresse e-mail est utilisée comme identifiant unique pour le compte de l'utilisateur, tandis que le mot de passe sert à protéger l'accès à ce compte.

**2-Le système vérifie la syntaxe de l'adresse e-mail et du mot de passe.** Assurer que l'adresse e-mail est correctement formatée et que le mot de passe répond aux exigences de sécurité du système.

**3-Si la vérification syntaxique est réussie, le système crée le compte de l'utilisateur.** Cela implique de stocker les informations de l'utilisateur dans une base de données et de lui attribuer un identifiant unique.

**4-Le système envoie un message de confirmation à l'utilisateur.** Ce message indique que le compte a été créé avec succès et fournit des instructions sur la façon de se connecter.

**3' -Si la vérification syntaxique échoue, le système affiche un message d'erreur à l'utilisateur.** Ce message indique la nature de l'erreur et fournit des instructions sur la façon de la corriger.

## b. La connexion

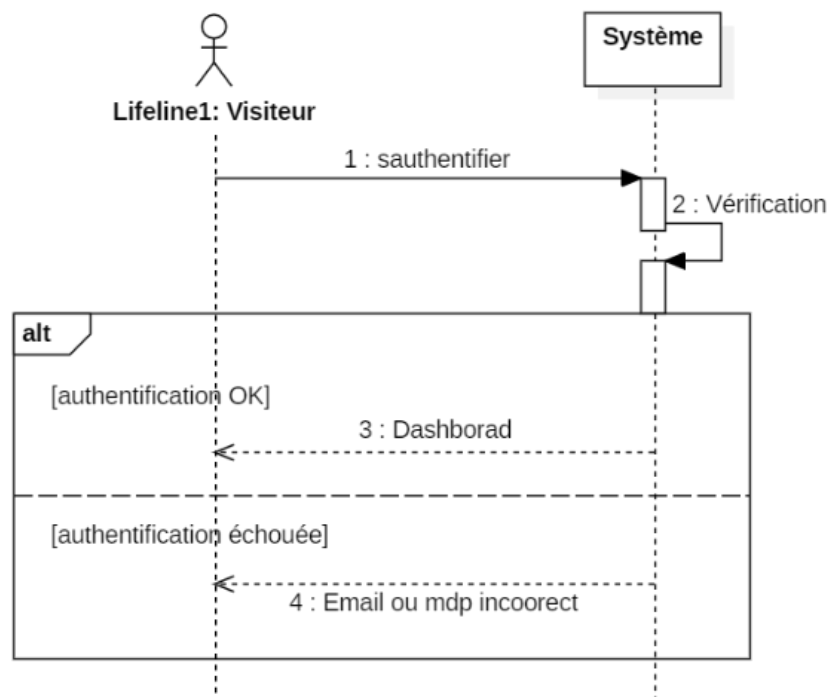


Figure 17 : Diagramme de séquence pour l'authentification

*Les participants impliqués dans ce processus sont :*

**Visiteur** : Représente l'être humain qui souhaite s'authentifier dans le système.

**Système** : Représente notre application avec la BDD.

*Étapes du processus d'authentification :*

**1-Authentification** : Le visiteur initie le processus d'authentification en envoyant une requête au système. Cette requête contient les identifiants de l'utilisateur, tels que l'email utilisateur et le mot de passe.

**2-Vérification** : Le système reçoit la requête d'authentification et vérifie les identifiants de l'utilisateur. Il compare le nom d'utilisateur et le mot de passe saisis aux informations stockées dans sa base de données.

**3-Résultat de l'authentification** : Si les identifiants de l'utilisateur sont corrects, le système envoie une réponse à l'utilisateur indiquant une authentification réussie. L'utilisateur est alors autorisé à accéder aux ressources du système.

**3'-Échec de l'authentification** : Si les identifiants de l'utilisateur sont incorrects, le système envoie une réponse à l'utilisateur indiquant un échec d'authentification. L'utilisateur est invité à réessayer en saisissant les bons identifiants.

### c. Détection d'objets des vidéos YOUTUBE

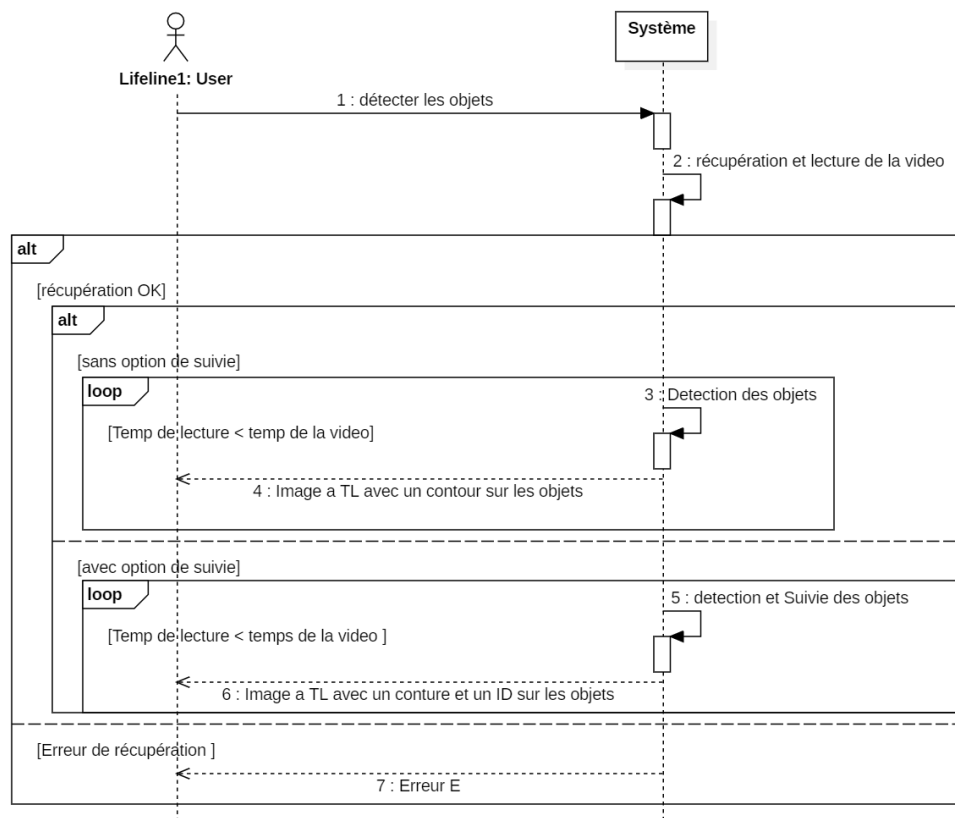


Figure 18 : diagramme de séquences concernant les vidéos YouTube

*Les participants impliqués dans ce processus sont :*

**Utilisateur** : Représente l'être humain authentifié dans le système.

**Système** : Représente notre application.

*Étapes du processus de la détection des objets dans des vidéos YouTube :*

**1-Récupération de la vidéo** : Le code commence par récupérer l'URL de la vidéo YouTube spécifiée à l'aide de la bibliothèque `pytube`.

**2-Lecture de la vidéo** : Une boucle commence où chaque image de la vidéo est lue successivement à l'aide de `vid\_cap.read()` jusqu'à ce que la vidéo soit entièrement lue ou qu'une erreur survienne.

**3-Resultat** : Les résultats de la détection ou du suivi sont affichés à l'aide de `st.image` tel que il existe des boites englobantes les objets détectés superposées à l'image de la vidéo, et si le suivi est activé, les objets suivis sont marqués avec leurs identifiants.

**3'-Erreur** : Si un problème est identifié, tout le processus est arrêté et un message d'erreur s'affiche.

#### d. Détection d'objet en temps réel

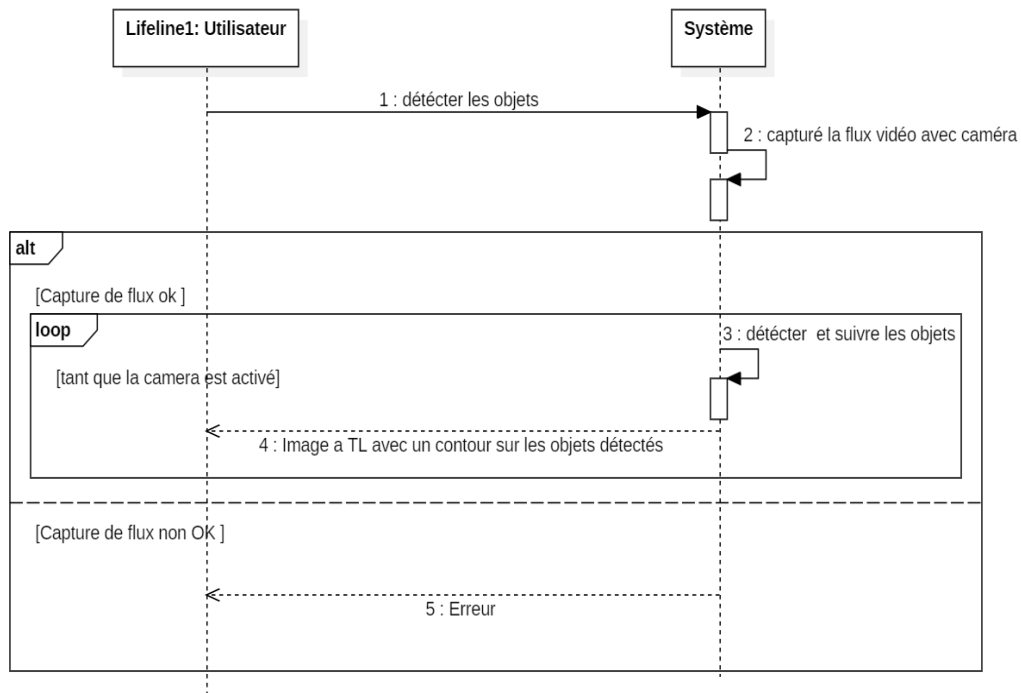


Figure 19 : diagramme de séquences concernant la détection des objets en temps réel

*Les participants impliqués dans ce processus sont :*

**Utilisateur** : Représente l'être humain authentifier dans le système.

**Système** : Représente notre application.

*Étapes du processus de la détection des objets en temp réel :*

**1-Lecture de la Vidéo** : Une fois la source vidéo ouverte et le flux vidéo capturé, une boucle commence où chaque image du flux est capturée et traitée.

**2-Détection et suivre les objets** : Le modèle YOLOv8 est utilisé pour prédire ou suivre les objets dans l'image en fonction des paramètres sélectionnés par l'utilisateur (niveau de confiance, suivi d'objets).

**3-Les résultats** : L'image résultante avec les objets détectés ou suivis est affichée dans l'interface Streamlit en temps réel à l'aide de `st.image``.

**3'-Les Erreurs** : Les erreurs sont affichées dans l'interface Streamlit pour informer l'utilisateur de tout problème survenu pendant la détection en temps réel.

## e. Détection d'objets avec la sélection des fichiers existants

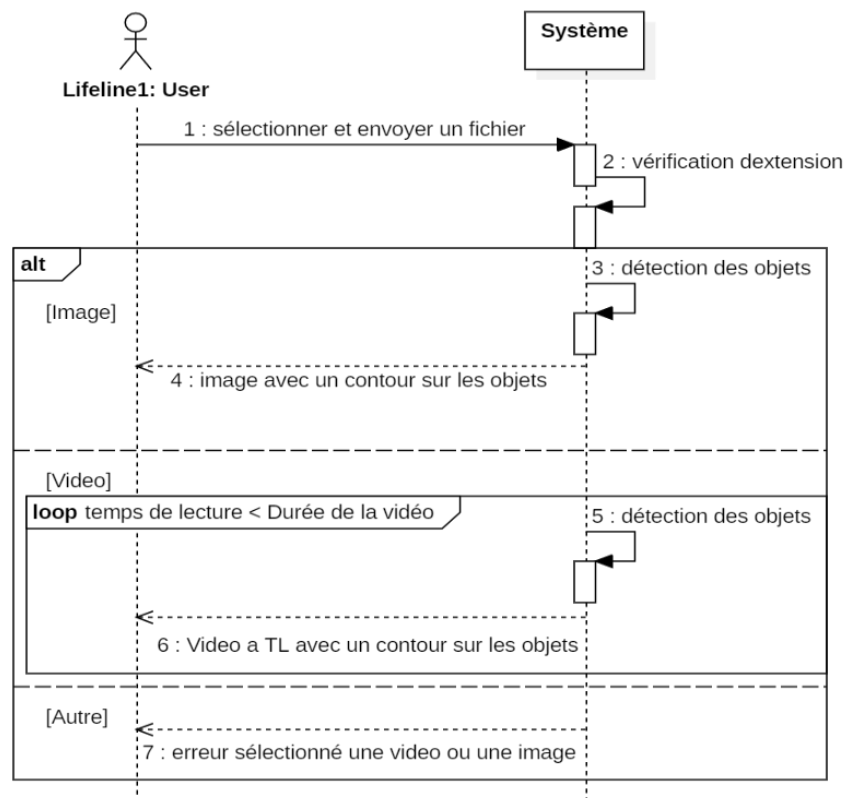


Figure 20 : Diagramme de séquence pour la détection des objets

*Les participants impliqués dans ce processus sont :*

**Utilisateur** : Représente la personne qui souhaite détecter des objets.

**Système** : Représente le système de détection d'objets.

*Étapes du processus de détection d'objets :*

**1-Sélection de l'entrée** : L'utilisateur sélectionne une image ou une vidéo comme entrée pour le processus de détection d'objets.

**2-Vérification d'extension** : Le système vérifie si le fichier sélectionné est une image d'extension « 'jpg', 'jpeg', 'png', 'bmp', 'webp' » ou une vidéo d'extension « .mp4 »

**3-Détection et suivre les objets** : Si oui le modèle YOLOv8 est utilisé pour prédire ou suivre les objets dans l'image ou la vidéo.

**3'-Erreur** : Si non le système affiche une erreur à l'utilisateur s'il existe.

### III. Architecture du système

#### a. Répertoire de weights

Le répertoire nommé « weights » Contient le fichier du modèle Yolov8 téléchargé. Nous utiliserons ce modèle pour la détection d'objets et la tâche de suivi. Nous devons simplement ajouter le chemin de ce fichier dans notre fichier settings.py.

#### b. Répertoire de vidéos

Le répertoire nommé « vidéos » fait pour stocker tous les fichiers vidéo. Nous sélectionnerons ces vidéos pour les tâches de détection et de suivi d'objets.

#### c. Répertoire d'images

Le répertoire nommé « images » est utilisée pour stocker toutes les images sur lesquelles vous souhaitez détecter des objets.

#### d. Main.py

Le fichier `main.py` utilise Streamlit pour créer une interface web multifonctionnelle avec une barre latérale de navigation.

#### e. About.py

Ce fichier contient une présentation sur le projet de détection d'objets utilisant YOLOv8.

#### f. Authenticator.py

Ce fichier contient des fonctions permettant l'authentification des utilisateurs à travers des requêtes API REST. Il facilite l'inscription, la connexion, et la réinitialisation du mot de passe, tout en gérant l'état de session pour garantir une expérience utilisateur interactive et sécurisée.

#### g. home.py

Il s'agit du fichier principal du projet, qui contient l'application de détection des objets. Il définit la mise en page de l'application, qui comprend un téléchargeur de fichiers, un lecteur vidéo, un curseur de seuil de confiance et une liste déroulante de sélection d'objets.

Il définit également la logique de l'application, qui comprend le chargement du modèle YOLOv8, la détection d'objets dans l'image ou les images vidéo téléchargées et l'affichage des objets détectés.



#### h. settings.py

Ce fichier contient toutes les constantes et paramètres de configuration requis pour le projet. Il définit le chemin d'accès au modèle YOLOv8, le seuil de confiance, le seuil de suppression non maximal et les noms des objets à détecter. Il contient également des paramètres liés à l'application Streamlit, tels que les URL d'image et de vidéo par défaut.

#### i. helper.py

Ce fichier contient des fonctions d'assistance utilisées dans le projet. Il comprend des fonctions pour charger le modèle YOLOv8, prétraiter l'image ou les images vidéo d'entrée et post-traiter les cadres de délimitation de sortie et les étiquettes de classe.

#### j. bytetrack.yaml et botsort.yaml

Les deux fichiers «.Yaml» sont des fichiers de configuration spécifiques pour deux types de suivi d'objets différents, utilisant deux bibliothèques ou implémentations distinctes : BoT-SORT et ByteTrack.

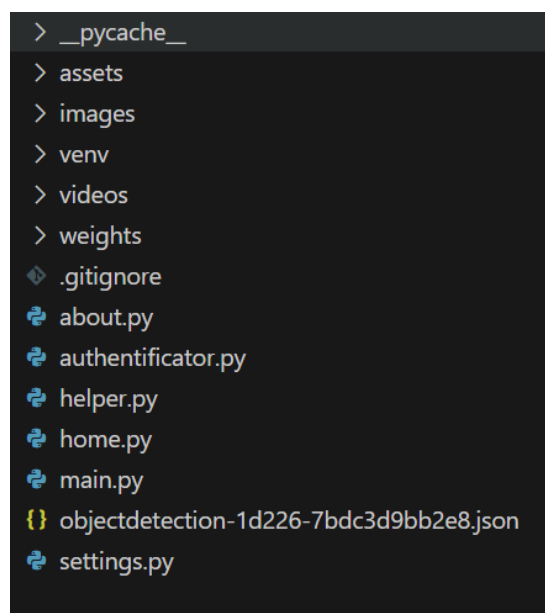


Figure 21 : Structure des répertoires et des fichiers de l'application

## Conclusion

Ce chapitre a présenté l'analyse des besoins et la conception du système. Les résultats de cette analyse et de cette conception seront utilisés pour la mise en œuvre du système dans le chapitre suivant.

# CHAPITRE 4 :

## Implémentation de l'application

### Introduction

Ce chapitre décrit l'implémentation de l'application, y compris les choix technologiques, l'architecture de l'application et les résultats obtenus.

### I. Choix des outils et technologies

#### 1. Logiciels

##### a. HTML, CSS et JS



Figure 22 : Logo HTML, CSS et JS

**HTML** (HyperText Markup Language) est le langage de balisage standard utilisé pour structurer le contenu des pages web, tandis que **CSS** (Cascading Style Sheets) est utilisé pour contrôler leur présentation visuelle en spécifiant les styles, tels que la couleur, la taille et la disposition. Ensemble, HTML et CSS fournissent la base de la conception web. **JavaScript**, quant à lui, est un langage de programmation côté client qui permet d'ajouter des fonctionnalités interactives et dynamiques aux pages web, telles que des animations, des effets visuels et des fonctionnalités de traitement des données. Ces trois langages sont largement utilisés ensemble pour créer des expériences web riches et interactives.

## b. Python



Figure 23 : Logo python

Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions, est un langage de programmation qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. C'est l'un des langages les plus utilisés dans le domaine de l'informatique quantique et de l'intelligence artificielle, ainsi que pour le traitement des Big Data et du Machine learning.

## c. Streamlit



Figure 24: Logo Streamlit

Streamlit est un framework open source qui simplifie le processus de création d'applications Web en Python, il facilite la création d'interfaces utilisateur Web pour les applications d'apprentissage automatique, permettant aux data scientists et aux développeurs de partager leur travail avec des intervenants non techniques.

Dans ce projet, Streamlit est utilisé pour créer une application Web qui affiche le flux vidéo traité, permettant aux utilisateurs d'ajuster divers paramètres du modèle ML tels que le seuil de confiance de détection et l'algorithme de suivi.

## 2. Libraires

### a. OpenCV (cv2)

OpenCV est une bibliothèque open-source spécialisée dans le traitement d'images et la vision par ordinateur. Elle est utilisée pour la manipulation d'images et la capture vidéo dans notre application.

### b. TensorFlow

TensorFlow est une bibliothèque de machine learning développée par Google. Dans notre cas, nous l'utilisons pour intégrer des modèles de détection d'objets basés sur YOLO (You Only Look Once).

### c. Werkzeug

Werkzeug est une bibliothèque WSGI (Web Server Gateway Interface) utilisée par Flask. Elle fournit des utilitaires utiles pour gérer les téléchargements de fichiers et les réponses HTTP.

### d. Requests

Requests est une bibliothèque HTTP pour Python qui facilite l'envoi de requêtes HTTP, utilisée ici pour communiquer avec d'autres services ou APIs si nécessaire.

### e. Ultralytics YOLO

Ultralytics YOLO est une implémentation de YOLO développée par Ultralytics, utilisée spécifiquement pour la détection d'objets en temps réel dans nos images et vidéos.

## 3. Modèle pré-entraînés

Les meilleurs modèles de détection d'objets sont formés sur des dizaines, voire des centaines, de milliers d'images étiquetées. De plus, les ensembles de données d'images eux-mêmes sont intrinsèquement coûteux en termes de calcul. Former un modèle de détection d'objets à partir de zéro nécessite beaucoup de temps et de ressources qui ne sont pas toujours disponibles. Former plusieurs modèles de détection d'objets à des fins de comparaison nécessite encore plus de temps et de ressources. Heureusement, nous n'avons pas à le faire. Au lieu de cela, nous pouvons utiliser un modèle pré-entraîné.

## f. YOLOv8

Le modèle YOLOv8 est un réseau neuronal profond qui a été formé sur un grand ensemble de données pour détecter des objets dans des images et des vidéos. Pour utiliser ce modèle pour la détection et le suivi d'objets, nous devons télécharger les pondérations YOLOv8 pré-entraînées.

YOLOv8 est un modèle de détection d'objets qui utilise des réseaux neuronaux profonds pour détecter des objets dans des images ou des vidéos. YOLOv8 est l'une des dernières versions de la série YOLO et offre une précision améliorée et des vitesses de détection plus rapides.

YOLOv8 est un modèle de pointe (SOTA) qui s'appuie sur le succès des versions YOLO précédentes et introduit de nouvelles fonctionnalités et améliorations pour améliorer encore les performances et la flexibilité. YOLOv8 est conçu pour être rapide, précis et facile à utiliser, ce qui en fait un excellent choix pour une large gamme de tâches de détection et de suivi d'objets, de segmentation d'instances, de classification d'images et d'estimation de pose.

Il utilise un seul réseau neuronal pour diviser une image ou une vidéo d'entrée en une grille de cellules, et chaque cellule est responsable de la détection des objets dans cette région. Le réseau prédit également les cadres de délimitation, les scores de confiance et les probabilités de classe pour chaque objet détecté.



*Figure 25 : Logo YOLO*

### **Pour quoi YOLO pour la détection d'objets ?**

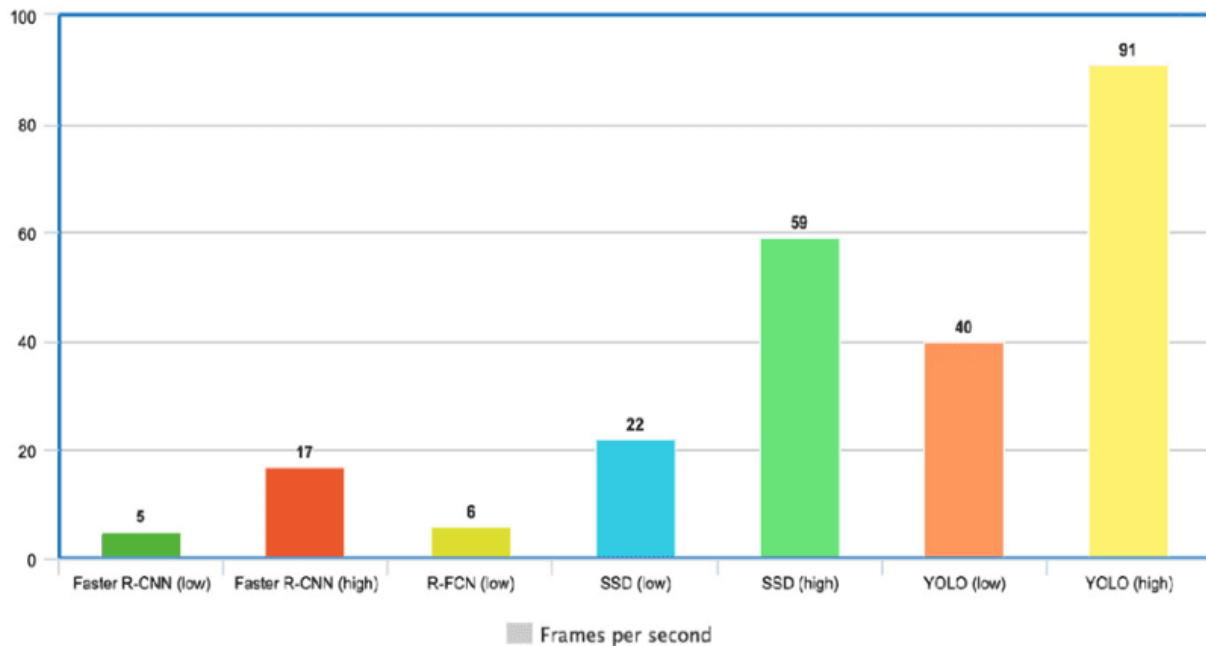
Parmi les raisons pour lesquelles YOLO est en tête de la concurrence, on peut citer :

1. Rapidité
2. Précision de détection
3. Bonne généralisation
4. Open-source

## **-Vitesse**

YOLO est extrêmement rapide car il ne gère pas de pipelines complexes. Il peut traiter des images à 45 images par seconde (FPS). De plus, YOLO atteint plus de deux fois la précision moyenne (mAP) par rapport aux autres systèmes en temps réel, ce qui en fait un excellent candidat pour le traitement en temps réel.

D'après le graphique ci-dessous, nous observons que YOLO est bien au-dessus des autres détecteurs d'objets avec 91 FPS.



## **-Haute précision de détection**

YOLO est bien au-delà des autres modèles de pointe en termes de précision avec très peu d'erreurs de fond.

## **-Meilleure généralisation**

Cela est particulièrement vrai pour les nouvelles versions de YOLO, qui seront abordées plus loin dans l'article. Grâce à ces avancées, YOLO a poussé un peu plus loin en fournissant une meilleure généralisation pour de nouveaux domaines, ce qui le rend idéal pour les applications reposant sur une détection d'objets rapide et robuste.

Par exemple, l'article Détection automatique du mélanome avec les réseaux neuronaux convolutionnels profonds de Yolo montre que la première version YOLOv1 a la précision moyenne la plus faible pour la détection automatique du mélanome, par rapport à YOLOv2 et YOLOv3.

## -Open source

Le fait de rendre YOLO open source a conduit la communauté à améliorer constamment le modèle. C'est l'une des raisons pour lesquelles YOLO a apporté autant d'améliorations en si peu de temps.

## II. Comment exécuter le code ?

### a. Prérequis

Python 3.7 ou supérieur et plusieurs packages Python tels que NumPy, OpenCV, PyTorch et Streamlit.

### b. Environnement virtuel

Installation : `pip install virtualenv`

Création : `virtualenv myenv`

Activation : `source myenv/bin/activate`

**Remarque :** Créez l'environnement virtuel dans le chemin de votre projet.

### c. Installation de Torch et Torchvision compatibles CUDA

- ✓ Télécharger et installer CUDA via le site : <https://developer.nvidia.com/>
- ✓ Ensuite, accédez au site Web officiel de PyTorch (<https://pytorch.org/>) et sélectionnez les options d'installation appropriées pour votre système.

**Remarque :** Vous devrez sélectionner la version appropriée de PyTorch qui prend en charge CUDA.

- ✓ Une fois que vous avez sélectionné les options appropriées, copiez la commande d'installation et collez la dans une fenêtre de terminal pour installer PyTorch.

### Cela peut ressembler à ceci :

```
➤ pip install torch==1.7.0+cpu torchvision==0.8.1+cpu -f  
<https://download.pytorch.org/whl/cu102/torch_stable.html>
```

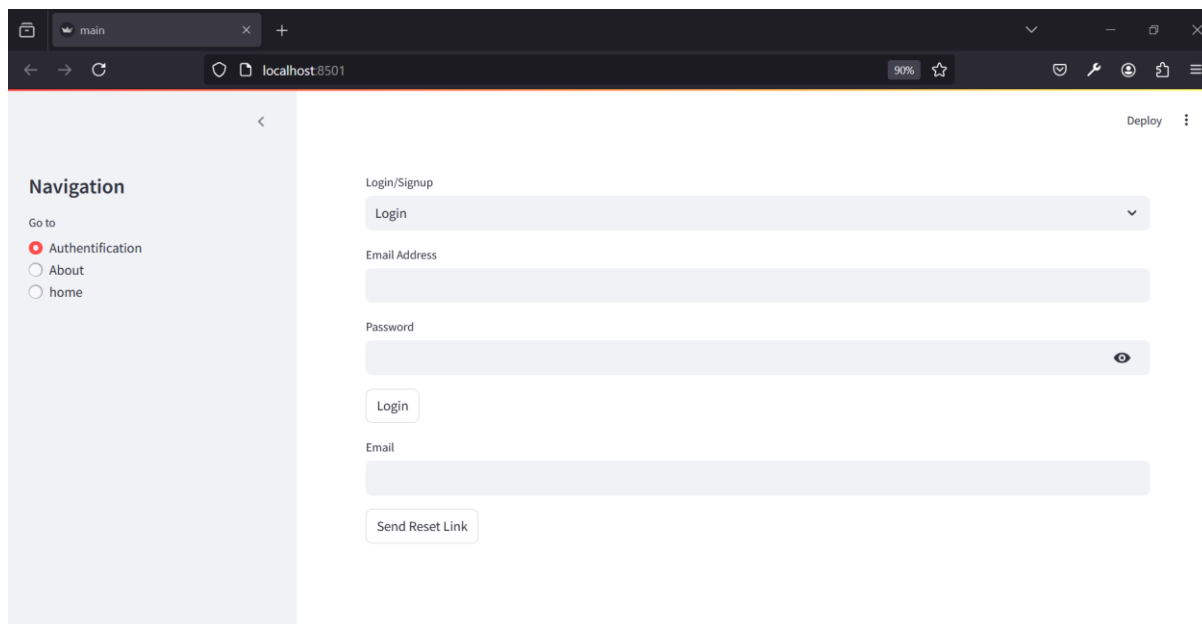
### d. Installation des packages

- ✓ Lancer la commande : `pip install -r requirements.txt`

### e. Lancement du projet

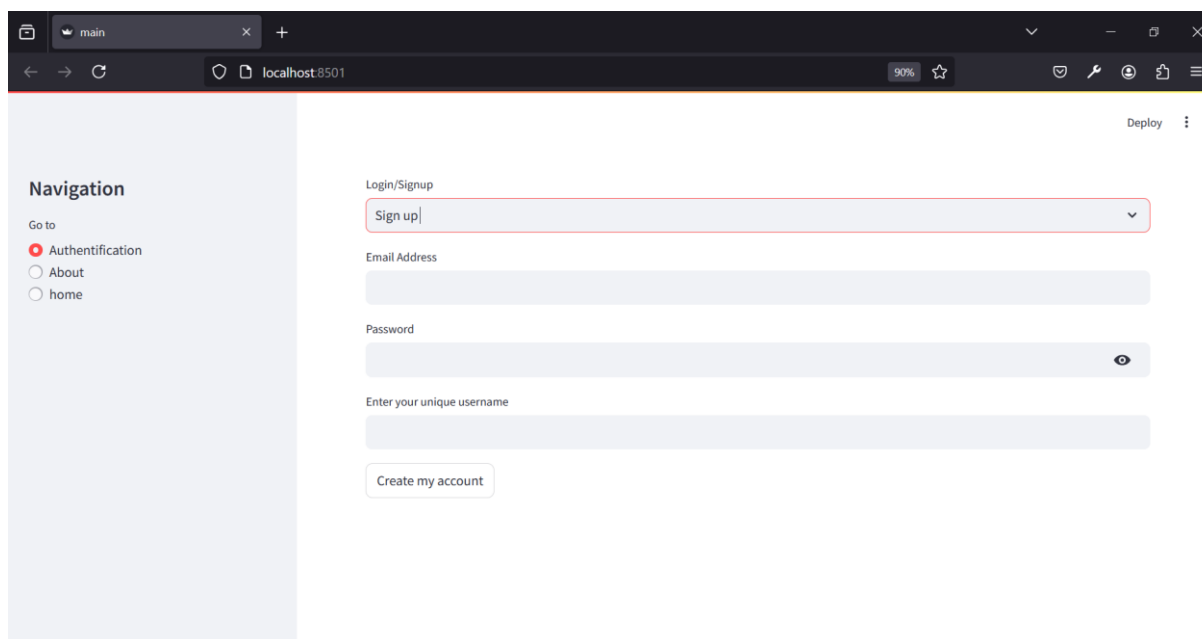
- ✓ Une fois tout est installé lancer le projet à l'aide de la commande :  
`Streamlit run main.py`

### III. Résultats



The screenshot shows a web browser window with the address bar displaying 'localhost:8501'. The page has a dark theme. On the left, there is a 'Navigation' sidebar with a 'Go to' section containing three radio buttons: 'Authentication' (selected), 'About', and 'home'. The main content area is titled 'Login/Signup' and features a dropdown menu currently set to 'Login'. Below this are input fields for 'Email Address' and 'Password' (with a toggle for visibility). There are 'Login' and 'Send Reset Link' buttons. A 'Deploy' button is visible in the top right corner.

Figure 26 : Page de connexion



The screenshot shows the same web browser window, but the page is now the registration form. The 'Login/Signup' dropdown menu is now set to 'Sign up'. The input fields are for 'Email Address' and 'Password'. Below these is a new field labeled 'Enter your unique username'. There is a 'Create my account' button. The 'Navigation' sidebar and 'Deploy' button remain the same.

Figure 27 : Page d'inscription



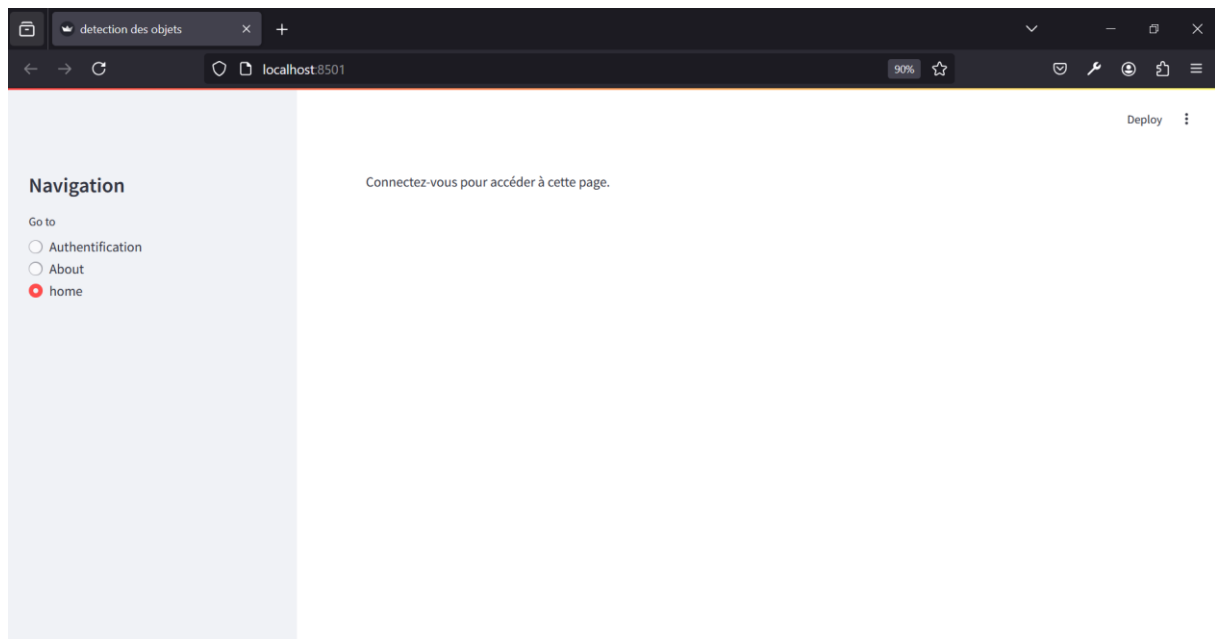


Figure 28 : exception lors de non Connexion de la page HOME

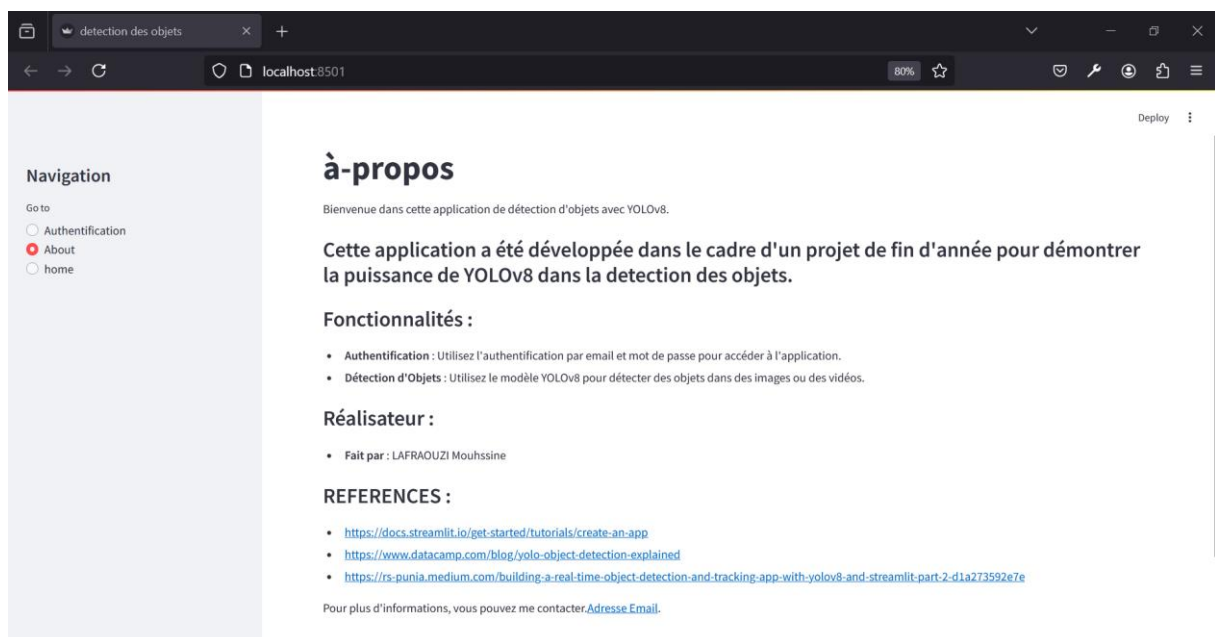


Figure 29 : Page About

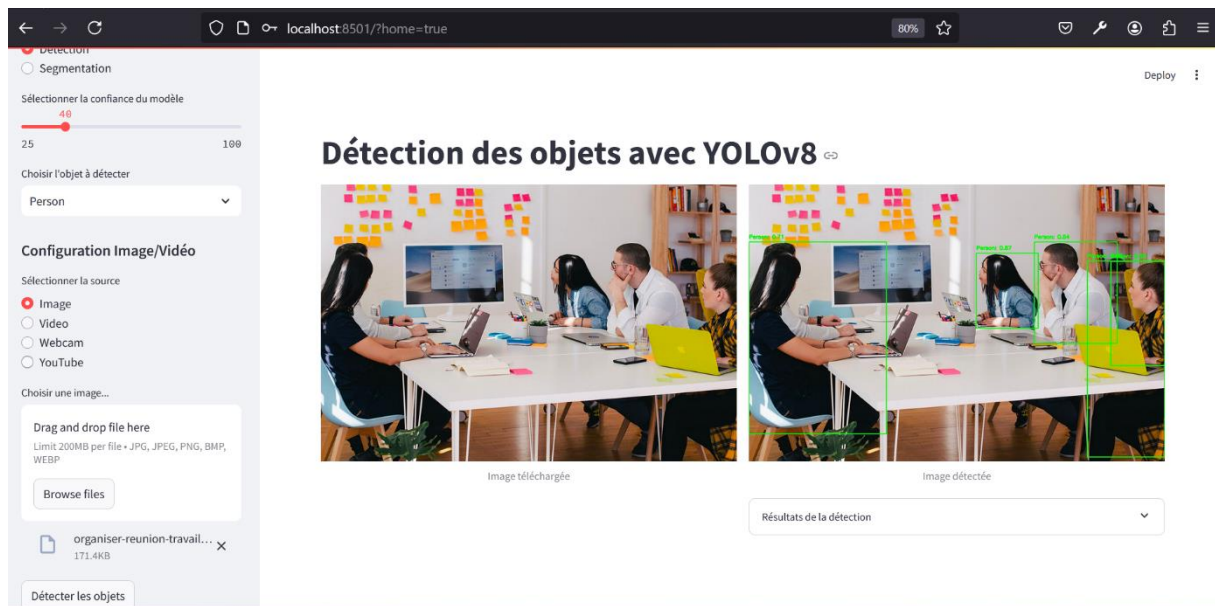


Figure 30 : page HOME

## Conclusion

Ce chapitre a présenté l'implémentation de l'application.

Les choix technologiques, l'architecture de l'application et les résultats obtenus.

# CONCLUSION

Nous avons examiné les concepts fondamentaux de la détection et du suivi d'objets, ainsi que les technologies et algorithmes les plus avancés tels que R-CNN, YOLO et SSD. L'étude de l'état de l'art a mis en lumière les progrès significatifs réalisés dans ce domaine et a souligné l'importance croissante de ces techniques dans des applications variées telles que la surveillance, l'automatisation industrielle et la conduite autonome.

Les résultats obtenus à travers l'implémentation pratique ont démontré la faisabilité et l'efficacité des approches étudiées. Ils ont également mis en évidence les défis persistants tels que la précision des détections en temps réel et l'adaptabilité aux conditions environnementales variables. Ces avancées ont des implications directes sur le développement de solutions technologiques plus intelligentes et plus sûres pour divers secteurs industriels et commerciaux.

Ce projet a été une occasion précieuse d'approfondir notre compréhension des systèmes de reconnaissance d'objets et des technologies associées. Nous avons appris l'importance cruciale de la sélection appropriée des algorithmes en fonction des besoins spécifiques du projet, ainsi que la nécessité d'une architecture système bien conçue pour garantir des performances optimales. De plus, nous avons acquis une expérience précieuse dans l'évaluation rigoureuse des résultats et dans l'optimisation continue des modèles déployés.

En conclusion, ce rapport a non seulement offert une vue d'ensemble détaillée des techniques modernes de reconnaissance des objets, mais il a également souligné leur potentiel transformateur et les défis technologiques associés. L'avenir de la reconnaissance des objets promet des avancées continues, propulsées par l'innovation dans le domaine de l'intelligence artificielle et de la vision par ordinateur, et ce travail constitue une étape importante dans cette direction.

# REFERENCES

**[1] Documentation Streamlit :**

<https://docs.streamlit.io/get-started/tutorials/create-an-app>

**[2] Explication de la détection d'objets YOLO :**

<https://www.datacamp.com/blog/yolo-object-detection-explained>

**[3] Création d'une application de détection et de suivi d'objets en temps réel :**

<https://rs-punia.medium.com/building-a-real-time-object-detection-and-tracking-app-with-yolov8-and-streamlit-part-2-d1a273592e7e>

**[4] Qu'est-ce que la détection d'objets ? Le guide ultime :**

<https://blog.roboflow.com/object-detection/>

**[5] Comment implémenter la détection d'objets à l'aide de l'apprentissage profond :**

<https://www.augmentedstartups.com/blog/how-to-implement-object-detection-using-deep-learning-a-step-by-step-guide>

**[6] La détection d'objets :**

<https://visionplatform.ai/fr/yolov9-le-premier-modele-induit-par-transformateur/>

**[7] Segmentation d'image :**

<https://www.ibm.com/fr-fr/topics/semantic-segmentation>