



Université Mohammed V de Rabat - Rabat
École Nationale d'Informatique et d'Analyse des
Systèmes



RAPPORT DE PROJET DE FIN D'ANNÉE 2023/2024

FILIÈRE

GÉNIE LOGICIEL

SUJET:

Prévention d'accidents par intelligence artificielle en temps réelle visuelle et sonore

Réalisé par :

M.Ismail AHAKAY

Jury :

Prof. M. Karim BAÏNA

M.Ayman ANNA

Prof. Mme. BERRADA Bouchra

Encadré par :

Prof. M. Karim BAÏNA

M. Mohamed IMAM

Année Universitaire 2023-2024

Remerciements

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer par des remerciements à ceux qui m'ont beaucoup appris au cours de cette année.

Nous sommes reconnaissants envers nos chers professeurs, en particulier nos encadrants, **Prof. M. Karim BAÏNA, Ph.D. Candidate M. Mohamed IMAM** pour leurs conseils éclairés, leur expertise et leur soutien constant.

Nous tenons de remercier les membres de jurys, **Prof. M. Karim BAÏNA**, et **Prof Mme. Bouchra BERRADA** pour leurs conseils éclairés et leur expertise .

Que tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce projet trouvent l'expression de mes remerciements les plus sincères.

Enfin, nous tenons à exprimer notre reconnaissance à tous ceux qui nous ont accordé leur confiance et nous ont fait confiance. Nous espérons que ce rapport répondra à leurs attentes.

Encore une fois, merci à tous pour votre précieuse contribution.

Résumé

La prévention de collision est une technologie essentielle dans de nombreuses applications, telles que la robotique et les véhicules autonomes. Ce projet se concentre sur le développement d'un système de prévention de collision en temps réel . Initialement, nous avons commencé par créer des modèles en temps réel, marquant notre première incursion dans le domaine de l'intelligence artificielle.

Notre exploration s'est ensuite étendue au modèle "Depth Anything", qui a servi de base à notre méthodologie de perspective trapézoïdale. En utilisant l'algorithme YOLO (You Only Look Once), nous avons mis en œuvre la détection d'objets pour identifier les obstacles potentiels. Par la suite, nous avons développé un algorithme pour détecter les collisions entre les boîtes englobantes et la perspective trapézoïdale définie.

La phase finale de notre projet a consisté à optimiser le système pour qu'il fonctionne en temps réel, garantissant une prévention des collisions rapide et précise. Ce projet améliore non seulement notre compréhension des techniques d'IA et de vision par ordinateur, mais fournit également un cadre robuste pour les avancées futures dans la technologie de prévention des collisions.

Mots clés : prévention des collisions, Intelligence Artificielle, vision de l'ordinateur,détection d'objets,temps réel, perspective trapézoïdale .

Abstract

Collision prevention is a critical technology with widespread applications in robotics and autonomous vehicles. In our project, we embarked on creating a real-time collision avoidance system, marking our initial foray into artificial intelligence. Our approach began with understanding and implementing real-time models, essential for ensuring the system's responsiveness and efficiency. By leveraging these models, we laid a strong foundation for developing more advanced AI-based techniques.

Central to our methodology was the "Depth Anything" model, which we used to establish a trapezoid perspective for assessing potential collisions. To detect objects within this perspective, we employed the YOLO (You Only Look Once) algorithm. YOLO's capability to identify and locate objects with high accuracy and speed made it an ideal choice for our system. By integrating YOLO, we could reliably detect obstacles, marking them as bounding boxes within the defined trapezoid perspective. This integration allowed us to focus on accurately detecting and predicting collisions by analyzing the spatial relationships and movements of these bounding boxes within the trapezoid area.

The final phase of our project involved optimizing the entire system to ensure real-time operation. This optimization was crucial for maintaining the system's accuracy and speed, essential for applications requiring immediate collision avoidance and response. Through iterative testing and refinement, we achieved a robust and efficient collision prevention system. This project not only enhanced our understanding of AI and computer vision but also provided a solid framework for future advancements in collision avoidance technology. Our work demonstrates the potential of combining state-of-the-art AI models with innovative perspectives, paving the way for more sophisticated and reliable collision avoidance systems in various domains.

Keywords : collision prevention, artificial intelligence, computer vision, Object detection, real-time, trapezoidal perspective .

Table des matières

Remerciements	1
Résumé	2
Abstract	3
Introduction générale	7
1 Contexte général du projet	8
1.1 L'état de l'art	8
1.1.1 Prévention des collisions basée sur les capteurs	8
1.1.2 Avancées en IA et vision par ordinateur pour la prévention des collisions	9
1.1.2.1 Faster R-CNN (Region Convolutional Neural Network)	9
1.1.2.2 RetinaNet	9
1.1.2.3 DeepSORT (Simple Online and Realtime Tracking)	9
1.2 Sujet	10
1.3 Problématique	10
1.4 Solution	11
1.5 Gestion de projet	11
1.6 Besoins fonctionnels	11
1.7 Besoins non fonctionnels	12
1.8 Objectifs	12
2 Conception Globale du Projet	14
2.1 Analyse théorique	14
2.2 Conception globale	14
2.2.1 Conception globale	15
2.3 Conception détaillée	15
2.3.1 Diagramme BPMN	16

3 Développement du Système et Performances obtenues	18
3.1 Langage de Programmation et Technologies Utilisées	18
3.1.1 Python :	18
3.1.1.1 Introduction à Python :	18
3.1.1.2 Explication du choix de Python :	19
3.1.1.3 Utilisation des fonctionnalités de Python :	19
3.1.1.4 Avantages et bénéfices de l'utilisation de Python :	19
3.1.1.5 Difficultés rencontrées et solutions apportées :	19
3.1.2 TensorFlow :	19
3.1.2.1 Introduction à TensorFlow :	19
3.1.2.2 Explication du choix de TensorFlow :	19
3.1.2.3 Utilisation des fonctionnalités de TensorFlow :	20
3.1.2.4 Avantages et bénéfices de l'utilisation de TensorFlow :	20
3.1.2.5 Difficultés rencontrées et solutions apportées :	20
3.1.3 PyTorch :	20
3.1.3.1 Introduction à PyTorch :	20
3.1.3.2 Explication du choix de PyTorch :	20
3.1.3.3 Utilisation des fonctionnalités de PyTorch :	20
3.1.3.4 Avantages et bénéfices de l'utilisation de PyTorch :	20
3.1.3.5 Difficultés rencontrées et solutions apportées :	21
3.1.4 OpenCV :	21
3.1.4.1 Introduction à OpenCV :	21
3.1.4.2 Explication du choix d'OpenCV :	21
3.1.4.3 Utilisation des fonctionnalités d'OpenCV :	21
3.1.4.4 Avantages et bénéfices de l'utilisation d'OpenCV :	21
3.1.4.5 Difficulties rencontrées et solutions apportées :	21
3.2 Accomplissement du Projet - Version Finale	22
3.2.1 Modèles d'IA intégrés	22
3.2.1.1 Depth Anything Model	22
3.2.1.2 YOLOv8	24
3.2.2 Méthodologie de perspective Trapézoïdale	26
3.2.2.1 Résultats et Commentaires	28
3.2.3 Traitement en Temps Réel	29
3.2.3.1 Flux de Traitement en Temps Réel	29
3.2.4 Intégration et Réalisation du projet	30
3.2.4.1 Combinaison de la Perspective Trapézoïdale	30
3.2.4.2 Détection des Personnes avec YOLO	30

3.2.4.3	Traitement en Temps Réel	30
3.2.4.4	Gestion des Intersections	31
3.2.4.5	Génération des Alertes	31
4	Discussion	32
4.1	Avantages de la Solution Proposée	32
4.2	Inconvénients de la Solution Proposée	33
5	Conclusion	34

Introduction générale

La prévention des collisions est une technologie cruciale dans de nombreux domaines, allant de la robotique aux véhicules autonomes, en passant par la réalité virtuelle et les jeux vidéo. Son importance réside dans la capacité à prévenir les accidents, améliorer l'interaction homme-machine, et assurer la sécurité dans des environnements divers et variés.

Ce rapport se concentre sur le développement d'un système de prévention des collisions en temps réel. Notre projet a débuté avec la création de modèles en temps réel, marquant ainsi notre première incursion dans le vaste domaine de l'intelligence artificielle (IA). La première phase de notre exploration a impliqué l'utilisation du modèle "Depth Anything", qui a posé les bases de notre méthodologie de perspective trapézoïdale.

L'algorithme YOLO (You Only Look Once) Pose a été implémenté pour la détection d'objets, nous permettant d'identifier avec précision les obstacles potentiels. Sur cette base, nous avons développé un algorithme capable de détecter les collisions entre les boîtes englobantes et la perspective trapézoïdale définie, améliorant ainsi la précision de notre système de prévention.

La phase finale du projet a été dédiée à l'optimisation du système pour qu'il puisse fonctionner en temps réel. Cela a permis de garantir une prévention des collisions rapide et précise, essentielle pour les applications nécessitant une réactivité immédiate.

Ce projet a non seulement approfondi notre compréhension des techniques d'IA et de vision par ordinateur, mais a également fourni un cadre robuste pour les avancées futures dans la technologie de prévention des collisions. En développant cette technologie, nous ouvrons la voie à des systèmes plus sûrs et plus efficaces dans divers domaines d'application.

Mots clés : prévention des collisions, intelligence artificielle, computer vision, YOLO Pose, perspective trapézoïdale, temps réel .

Chapitre 1

Contexte général du projet

Dans ce chapitre, nous présentons le contexte général du projet, en abordant le sujet et son contexte, la problématique à résoudre, la solution envisagée, la gestion du projet mise en place, ainsi que les besoins fonctionnels, les besoins non fonctionnels et les objectifs du projet.

1.1 L'état de l'art

La prévention des collisions est une technologie essentielle dans de nombreuses applications, telles que la robotique et les véhicules autonomes. Son importance réside dans la capacité à prévenir les accidents, améliorer l'interaction homme-machine et assurer la sécurité dans des environnements divers et variés. Avec l'essor rapide de ces domaines, la nécessité de systèmes de prévention des collisions précis et en temps réel devient de plus en plus pressante.

1.1.1 Prévention des collisions basée sur les capteurs

Historiquement, les systèmes de prévention des collisions se sont fortement appuyés sur des capteurs tels que les ultrasons, le radar et le LIDAR. Ces technologies ont été essentielles dans des applications allant des systèmes de sécurité automobile à la robotique. Par exemple, les capteurs ultrasoniques ont été largement utilisés pour des tâches de détection à courte portée, comme l'assistance au stationnement dans les véhicules. Les systèmes radar, offrant des portées de détection plus longues et une robustesse sous diverses conditions météorologiques, ont joué un rôle crucial dans la mise en œuvre du régulateur de vitesse adaptatif (ACC). Le LIDAR, avec ses données spatiales de haute résolution, a été crucial pour cartographier les environnements pour les véhicules autonomes.

Malgré leur utilisation répandue, les systèmes basés sur les capteurs rencontrent des défis significatifs. Ceux-ci incluent des coûts élevés, une portée de détection limitée, une susceptibilité aux influences environnementales et des problèmes de surcharge de données où le traitement de vastes quantités de données de capteurs peut conduire à des temps de réponse accrus.

1.1.2 Avancées en IA et vision par ordinateur pour la prévention des collisions

Les progrès récents en intelligence artificielle (IA) et vision par ordinateur ont permis de développer des solutions plus sophistiquées et efficaces pour la prévention des collisions. Voici trois exemples de ces solutions et les défis associés à chacune :

1.1.2.1 Faster R-CNN (Region Convolutional Neural Network)

- **Avantages** : Faster R-CNN est réputé pour sa précision et sa capacité à détecter et segmenter des objets avec une grande exactitude, ce qui le rend particulièrement utile dans des applications nécessitant une analyse détaillée, comme les systèmes de sécurité avancés et l'inspection industrielle.
- **Inconvénients** : Toutefois, il est plus lent par rapport à d'autres solutions comme SSD, ce qui peut poser des problèmes dans des scénarios nécessitant une réponse en temps réel. En outre, il exige des ressources computationnelles importantes, rendant son déploiement difficile sur des dispositifs à faible puissance.

1.1.2.2 RetinaNet

- **Avantages** : Utilise une architecture de réseau de neurones profonde pour améliorer la précision de détection, surtout pour les objets de petite taille, ce qui est crucial pour des applications où la précision est essentielle, comme la surveillance urbaine ou la navigation autonome.
- **Inconvénients** : Peut être gourmand en ressources et nécessiter des temps de traitement plus longs, limitant son utilisation dans des applications nécessitant une prévention des collisions ultra-rapide.

1.1.2.3 DeepSORT (Simple Online and Realtime Tracking)

- **Avantages** : Technique de suivi des objets en temps réel qui complète les algorithmes de détection en assurant le suivi des objets à travers les frames successives d'une vidéo, très utile dans les systèmes de surveillance et les véhicules autonomes, où le suivi précis des objets en mouvement est crucial.
- **Inconvénients** : Peut rencontrer des difficultés avec des objets très proches les uns des autres ou dans des environnements très dynamiques où les objets changent rapidement de direction. De plus, il nécessite des capacités de calcul significatives pour maintenir le suivi en temps réel.

Ces solutions offrent des capacités avancées pour la prévention des collisions, mais elles présentent également des limitations importantes dans des environnements réels. Le défi principal consiste à

équilibrer la précision et la vitesse, ainsi qu'à optimiser l'utilisation des ressources computationnelles pour garantir des performances fiables et adaptées aux besoins spécifiques des applications. La prévention des collisions reste un domaine en évolution rapide, avec des recherches continues pour surmonter ces défis et améliorer l'efficacité des systèmes existants.

1.2 Sujet

Notre projet se concentre sur le développement d'un système de prévention des collisions en temps réel. Le projet a débuté avec la création de modèles en temps réel, marquant notre première incursion dans le domaine de l'IA. L'objectif principal est de développer un système capable de détecter les collisions de manière précise et rapide, répondant aux exigences des environnements dynamiques et variés.

Pour atteindre cet objectif, nous avons exploré le modèle "Depth Anything", qui a servi de base pour notre méthodologie de perspective trapézoïdale. Cette approche nous permet de visualiser et d'analyser l'espace en trois dimensions de manière plus efficace. Nous avons également implémenté l'algorithme YOLO (You Only Look Once) Pose pour la détection d'objets, afin d'identifier avec précision les obstacles potentiels.

Le projet a été structuré de manière rigoureuse, avec une gestion détaillée incluant la définition des rôles et des responsabilités, l'établissement d'un calendrier précis, et la mise en place de points de contrôle réguliers pour suivre l'avancement. Les besoins fonctionnels du projet incluent la capacité à détecter les collisions en temps réel, la précision de la détection des objets, et l'intégration pour une réponse immédiate. Les besoins non fonctionnels portent sur la robustesse du système, sa capacité à fonctionner dans des conditions variées et son efficacité en termes de ressources.

La phase finale du projet a été dédiée à l'optimisation du système pour qu'il puisse fonctionner en temps réel, garantissant une prévention des collisions rapide et précise. Les objectifs principaux étaient de développer une solution qui améliore la sécurité et l'interaction dans divers environnements, tout en fournissant une base robuste pour les futures avancées technologiques dans la prévention des collisions.

En conclusion, ce projet a non seulement approfondi notre compréhension des techniques d'IA et de vision par ordinateur, mais a également fourni un cadre solide pour les innovations futures dans ce domaine essentiel.

1.3 Problématique

Comment peut-on développer et optimiser un système de prévention des collisions en temps réel utilisant des techniques avancées de computer vision et d'intelligence artificielle, pour améliorer la précision et la rapidité de la détection dans des applications comme la robotique et les véhicules autonomes ?

1.4 Solution

Le projet de développement du système de prévention des collisions propose une solution innovante qui répond aux défis de détection rapide et précise dans des environnements dynamiques. Ce système offre une intégration avancée, combinant le modèle "Depth Anything" avec l'algorithme YOLO Pose pour une détection d'objets précise. Les fonctionnalités incluent une visualisation en temps réel des collisions, des outils avancés d'analyse et de gestion des données, et une interface conviviale pour une utilisation efficace par les utilisateurs finaux. Notre engagement est d'assurer une gestion transparente, équitable et sécurisée du système de prévention des collisions, offrant ainsi une expérience utilisateur de haute qualité pour toutes les parties prenantes.

1.5 Gestion de projet

La réalisation se compose essentiellement de 3 parties :

- Analyse des besoins
- Conception du projet
- Développement (codage)

1.6 Besoins fonctionnels

Voici les principaux besoins fonctionnels identifiés pour le système de prévention des collisions en temps réel :

1. **prévention des collisions en temps réel** : Le système doit être capable de détecter les collisions entre les objets en temps réel, avec une précision élevée et une réactivité rapide.
2. **Visualisation des collisions** : Les utilisateurs doivent pouvoir visualiser les collisions détectées en temps réel à travers une interface conviviale, permettant une analyse approfondie des événements.
3. **Analyse des données de collision** : Le système doit fournir des outils d'analyse avancés pour interpréter les données de collision, y compris la trajectoire des objets et la gravité des collisions.
4. **Gestion des paramètres de détection** : Les administrateurs doivent avoir la capacité de configurer et de modifier les paramètres de détection en fonction des besoins spécifiques de l'environnement.
5. **Alertes en temps réel** : Le système doit être capable de générer des alertes en temps réel lorsqu'une collision est détectée, permettant une action immédiate pour prévenir les accidents.
6. **Intégration avec d'autres systèmes** : Le système doit pouvoir s'intégrer avec d'autres systèmes, tels que les systèmes de contrôle de véhicules autonomes ou les systèmes de surveillance industrielle, pour une utilisation dans des applications complexes.

En répondant à ces besoins fonctionnels, le système de prévention des collisions en temps réel sera en mesure de fournir une solution efficace et fiable pour prévenir les accidents et assurer la sécurité dans divers environnements.

1.7 Besoins non fonctionnels

Voici les principaux besoins non fonctionnels identifiés pour le système de prévention des collisions en temps réel :

- **Performance** : Le système doit offrir des performances élevées pour garantir une prévention des collisions rapide et précise. Les algorithmes de détection doivent être optimisés pour minimiser les temps de traitement et assurer une réponse en temps réel, même dans des environnements complexes.
- **Fiabilité** : Le système doit être fiable et stable, en minimisant les risques de défaillance ou de panne. Des tests rigoureux doivent être effectués pour valider la robustesse du système et garantir son bon fonctionnement dans des conditions variées.
- **Extensibilité** : Le système doit être conçu de manière à être facilement extensible, permettant l'ajout de nouvelles fonctionnalités et l'intégration avec d'autres systèmes à l'avenir. Cela garantira la pérennité et l'adaptabilité du système face aux évolutions technologiques et aux besoins changeants.
- **Portabilité** : Le système doit être compatible avec différentes plateformes et architectures, assurant ainsi sa disponibilité sur une large gamme de dispositifs et de configurations matérielles. Il doit fonctionner de manière optimale sur les ordinateurs de bureau, les appareils mobiles et autres dispositifs compatibles.
- **Maintenance** : Le système doit être facile à entretenir et à mettre à jour pour garantir sa stabilité et sa performance à long terme. Une documentation claire et exhaustive ainsi que des bonnes pratiques de développement doivent être fournies pour faciliter la maintenance et la résolution des problèmes éventuels.

En répondant à ces besoins non fonctionnels, le système de prévention des collisionssera en mesure de fournir une solution fiable, performante et sécurisée pour répondre aux exigences des environnements dynamiques et variés.

1.8 Objectifs

Les objectifs du projet de développement du système de prévention des collisions en temps réel, intégrant l'utilisation de données audio pour les avertissements, sont les suivants :

1. Développer un système de prévention des collisions performant et réactif, capable de détecter les collisions avec une grande précision dans des environnements dynamiques et variés, et d'émettre des avertissements audio en cas de danger imminent.
2. Garantir des performances élevées du système, en minimisant les temps de traitement et en assurant une réponse en temps réel même dans des environnements complexes, tout en maintenant une qualité audio optimale.
3. Concevoir le système de manière à être facilement extensible, permettant l'ajout de nouvelles fonctionnalités audio et visuelles et l'intégration avec d'autres systèmes à l'avenir.
4. Assurer la compatibilité du système avec différentes plateformes et architectures, garantissant ainsi sa disponibilité sur une large gamme de dispositifs et de configurations matérielles, avec une prise en charge optimale du traitement audio.

Ces objectifs visent à développer un système de prévention des collisions efficace, fiable et sécurisé, utilisant des données audio pour renforcer la prévention des accidents et la sécurité des utilisateurs dans divers environnements.

Chapitre 2

Conception Globale du Projet

Dans cette section, nous allons explorer la conception globale de notre projet de développement du système de prévention des collisions en temps réel. Nous mettrons en lumière les différentes phases du processus de conception ainsi que les décisions techniques que nous avons prises pour atteindre nos objectifs.

2.1 Analyse théorique

L'analyse conceptuelle marque le début de notre exploration approfondie du système de prévention des collisions en temps réel. Cette phase initiale revêt une importance cruciale car elle nous permet de définir clairement les concepts et les principes fondamentaux sur lesquels reposera notre projet.

Durant cette analyse, nous avons examiné en détail les différents éléments constitutifs du système de prévention des collisions. Nous avons étudié les concepts de détection d'objets en temps réel, d'intégration et de traitement des données, afin de comprendre les défis et les opportunités qu'ils présentent.

De plus, nous avons effectué une étude approfondie des technologies existantes dans le domaine de la prévention des collisions, en évaluant leurs avantages et leurs limitations. Cela nous a permis de faire des choix éclairés quant aux outils et aux méthodes à utiliser dans notre projet.

En analysant conceptuellement le système de prévention des collisions, nous avons posé les bases nécessaires pour une conception et un développement réussis. Cette compréhension approfondie des concepts sous-jacents nous permettra de prendre des décisions éclairées tout au long du processus de développement, afin de garantir la qualité et la pertinence de notre solution.

2.2 Conception globale

Après avoir minutieusement analysé les besoins et les attentes de notre projet de prévention des collisions en temps réel, nous abordons dans cette section une phase cruciale : la conception. Cette étape revêt une importance capitale pour le succès du projet, car elle vise à établir une structure

solide pour notre système. Notre objectif principal est de définir avec précision les tâches à accomplir et de préparer le terrain pour la phase de réalisation.

Dans cette première partie, nous nous concentrerons sur la conception globale de notre système. Nous définissons les grandes lignes de son architecture, en mettant l'accent sur les éléments clés qui le composeront. Cette vision d'ensemble nous permettra de garantir une cohérence globale dans notre approche.

Par la suite, dans la deuxième partie, nous plongeons plus en profondeur en utilisant les diagrammes appropriés. Ces diagrammes nous permettent de représenter de manière visuelle et précise la structure et les interactions au sein de notre système. Ils serviront de guide précieux pour les développeurs lors de la phase de mise en œuvre.

La phase de conception marque un pas important vers la concrétisation de notre projet de prévention des collisions en temps réel . Elle nous permet de transformer les besoins en une structure claire et cohérente, posant ainsi les bases d'un système performant et fonctionnel.

2.2.1 Conception globale

Notre projet de prévention des collisions en temps réel vise à développer un système robuste capable de détecter les collisions avec précision et de fournir des avertissements audio en cas de danger imminent. Notre architecture repose sur trois principaux composants :

- Un module de traitement visuel chargé de capturer et d'analyser en temps réel les données vidéo pour détecter les collisions.
- Un moteur d'alerte audio intégré au système pour émettre des avertissements sonores en cas de prévention des collisions.
- Une interface utilisateur conviviale permettant aux utilisateurs de visualiser en temps réel les collisions détectées et de recevoir des alertes audio.

Cette architecture spécifique guide notre conception pour garantir un fonctionnement fluide et une expérience utilisateur optimale dans le contexte de notre projet de prévention des collisions en temps réel .

2.3 Conception détaillée

Dans cette phase cruciale de notre projet de prévention des collisions en temps réel, nous nous attelons à une conception approfondie qui concrétisera notre vision. Nous utiliserons des diagrammes BPMN pour modéliser la remontée d'événements de l'edge vers le fog, l'escalade visuelle et sonore, ainsi que pour définir les actions d'alerte. Notre système évaluera la criticité des événements en fonction de la distance et de la gravité, ce qui permettra une alerte graduelle et adaptée.

2.3.1 Diagramme BPMN

Dans cette section, nous utiliserons des diagrammes BPMN (Business Process Model and Notation) pour représenter de manière détaillée la remontée des événements de l'edge vers le fog. Ces diagrammes permettront également de modéliser l'escalade visuelle et sonore ainsi que les actions d'alerte qui seront déclenchées en fonction de la criticité des événements détectés. Le système évaluera minutieusement la distance et la gravité de chaque événement afin de déterminer le niveau d'alerte approprié. Cette approche graduelle garantira une réaction adaptée à chaque situation, assurant ainsi une gestion efficace des collisions en temps réel.

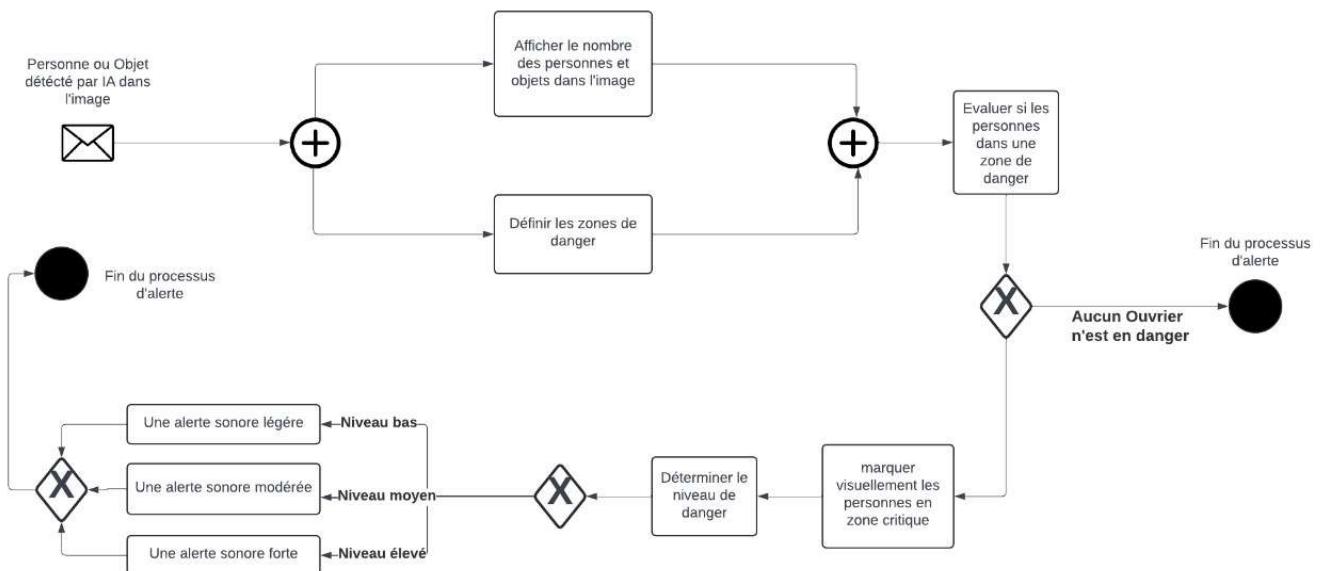


FIGURE 2.1 – Diagramme BPMN

1. Détection de Personnes ou Objets : Le processus débute avec l'utilisation de l'algorithme YOLO Pose pour la détection en temps réel de personnes ou d'objets dans une image capturée. Ce choix technologique permet une reconnaissance efficace et rapide, essentielle pour les applications nécessitant une réaction immédiate.
2. Affichage du Nombre de Personne Déetectés : Après la détection, le système affiche le nombre de personnes ou objets détectés. Cette information est essentielle pour la surveillance et permet d'évaluer l'efficacité de la détection en fonction de l'environnement opérationnel.
3. Évaluation de la Situation de Danger : Utilisant les données fournies par le modèle "Depth Anything", le système évalue si les objets détectés se trouvent dans une perspective trapézoïdale de danger, une méthodologie développée pour estimer la proximité et le potentiel de collision des objets par rapport à des zones sensibles.
4. Détermination du Niveau de Danger : Le niveau de danger est déterminé en fonction de l'emplacement et du mouvement des objets dans la perspective trapézoïdale. Les niveaux sont classifiés

- comme bas, moyen ou élevé, guidant les réponses du système en termes d'alertes .
5. Marquage Visuel et Alertes Sonores : En cas de détection d'une situation critique, des alertes sonores sont générées et un marquage visuel est appliqué sur l'image pour indiquer clairement les objets à risque. Ces alertes sont différencierées par le niveau de danger, avec des sons spécifiques pour chaque niveau.
 6. Conclusion du Processus : Le processus se termine avec l'arrêt des alertes si aucune collision n'est imminente, ou avec la continuation des interventions nécessaires jusqu'à la résolution de la situation. Ce système robuste garantit une détection et une réaction rapides face à des situations potentiellement dangereuses, améliorant ainsi la sécurité globale.

Chapitre 3

Développement du Système et Performances obtenues

Dans cette section, nous discutons de l'implémentation de notre système de prévention des collisions en temps réel et présentons les résultats obtenus, illustrés par des captures d'écran.

3.1 Langage de Programmation et Technologies Utilisées

Notre projet intègre des technologies de pointe en programmation et en traitement de données pour le développement d'un système robuste et efficace. Nous avons sélectionné Python, TensorFlow, PyTorch, et OpenCV en raison de leurs capacités exceptionnelles dans le domaine de l'intelligence artificielle et de la vision par ordinateur.

3.1.1 Python :



FIGURE 3.1 – Logo de Python.

3.1.1.1 Introduction à Python :

Python est choisi pour son langage clair, sa vaste bibliothèque de ressources, et son soutien robuste pour le traitement de données et l'apprentissage automatique. Il rend le développement de notre

application à la fois flexible et puissant.

3.1.1.2 Explication du choix de Python :

Python offre une syntaxe intuitive et est largement adopté dans la communauté scientifique, ce qui facilite le partage de code et la collaboration. Sa compatibilité avec de nombreux frameworks et bibliothèques d'IA est essentielle pour notre projet.

3.1.1.3 Utilisation des fonctionnalités de Python :

Nous utilisons Python pour scripter le comportement de notre système, manipuler les données de capteurs, et intégrer des algorithmes complexes de prévention de collisions.

3.1.1.4 Avantages et bénéfices de l'utilisation de Python :

Python accélère le développement, facilite la maintenance et l'évolution de notre système, et permet une intégration transparente avec des technologies complémentaires.

3.1.1.5 Difficultés rencontrées et solutions apportées :

La gestion de la performance en temps réel était un défi en raison de la nature interprétée de Python. Nous avons optimisé notre code et utilisé des bibliothèques Python spécifiques pour améliorer les performances.

3.1.2 TensorFlow :



FIGURE 3.2 – Logo de TensorFlow.

3.1.2.1 Introduction à TensorFlow :

TensorFlow est appliqué pour le développement et l'amélioration continue du modèle d'IA sous-jacent, assurant une prédiction précise et fiable des collisions potentielles.

3.1.2.2 Explication du choix de TensorFlow :

Nous avons choisi TensorFlow pour ses capacités avancées dans la modélisation et le déploiement d'algorithmes d'apprentissage profond, indispensables pour notre analyse prédictive.

3.1.2.3 Utilisation des fonctionnalités de TensorFlow :

TensorFlow est utilisé pour construire, entraîner et déployer des modèles d'apprentissage profond qui identifient et prévoient les comportements d'objets dans des environnements complexes.

3.1.2.4 Avantages et bénéfices de l'utilisation de TensorFlow :

TensorFlow offre une scalabilité, des performances robustes dans le traitement des données en temps réel, et une intégration facile avec les API Python.

3.1.2.5 Difficultés rencontrées et solutions apportées :

La complexité de la configuration initiale de TensorFlow a nécessité une compréhension approfondie. Nous avons organisé des sessions de formation pour nos développeurs pour surmonter cet obstacle.

3.1.3 PyTorch :



FIGURE 3.3 – Logo de PyTorch.

3.1.3.1 Introduction à PyTorch :

PyTorch est utilisé pour ses fonctionnalités avancées en matière d'apprentissage automatique, contribuant à la formation, à l'évaluation, et à l'implémentation efficaces du modèle d'IA.

3.1.3.2 Explication du choix de PyTorch :

Nous avons sélectionné PyTorch en raison de sa flexibilité et de son orientation vers la recherche, ce qui nous permet d'expérimenter et d'innover rapidement dans le développement de nos modèles.

3.1.3.3 Utilisation des fonctionnalités de PyTorch :

Dans notre projet, PyTorch facilite la prototypage rapide et efficace de nouvelles idées en apprentissage automatique, en particulier pour les modèles qui nécessitent un ajustement dynamique lors de l'apprentissage.

3.1.3.4 Avantages et bénéfices de l'utilisation de PyTorch :

L'architecture de PyTorch et son support pour le calcul dynamique offrent une grande flexibilité et accélèrent le cycle de développement des fonctionnalités d'IA.

3.1.3.5 Difficultés rencontrées et solutions apportées :

Le passage de la phase de prototype à la production avec PyTorch peut être complexe. Nous avons utilisé des outils spécifiques pour faciliter cette transition et garantir la stabilité de notre application en production.

3.1.4 OpenCV :



FIGURE 3.4 – Logo d'OpenCV.

3.1.4.1 Introduction à OpenCV :

OpenCV est intégré pour le traitement en temps réel des flux visuels, permettant à l'application de capturer, d'analyser, et de réagir aux données visuelles de manière dynamique et réactive.

3.1.4.2 Explication du choix d'OpenCV :

OpenCV est choisi pour ses puissantes capacités de traitement d'image et de vidéo, essentielles pour la capture et l'analyse précises des environnements dans lesquels notre système est déployé.

3.1.4.3 Utilisation des fonctionnalités d'OpenCV :

Nous utilisons OpenCV pour traiter les données de la caméra en temps réel, détecter les objets, et suivre leur mouvement, ce qui est crucial pour la prédiction et la prévention des collisions.

3.1.4.4 Avantages et bénéfices de l'utilisation d'OpenCV :

OpenCV améliore considérablement la réactivité de notre système et supporte les exigences de traitement intensif de notre application.

3.1.4.5 Difficultés rencontrées et solutions apportées :

L'intégration initiale d'OpenCV avec d'autres composants logiciels a posé des défis, résolus grâce à notre collaboration étroite en tant que binôme, ce qui a permis d'assurer une compatibilité complète et une intégration fluide des différentes technologies au sein de notre système.

3.2 Accomplissement du Projet - Version Finale

3.2.1 Modèles d'IA intégrés

3.2.1.1 Depth Anything Model

Dans le domaine de la vision par ordinateur, l'estimation de profondeur à partir d'images monoculaires (c-à-dire prises avec une seule caméra) représente un défi majeur et un domaine de recherche actif. La capacité à déduire la troisième dimension, la profondeur, à partir d'une image 2D est cruciale pour de nombreuses applications technologiques, telles que la robotique autonome, la navigation de véhicules autonomes, la réalité augmentée, et plus encore. Le modèle "Depth Anything", développé pour répondre à ce besoin, utilise des techniques avancées d'apprentissage automatique pour interpréter et transformer des images planes en cartes de profondeur détaillées.

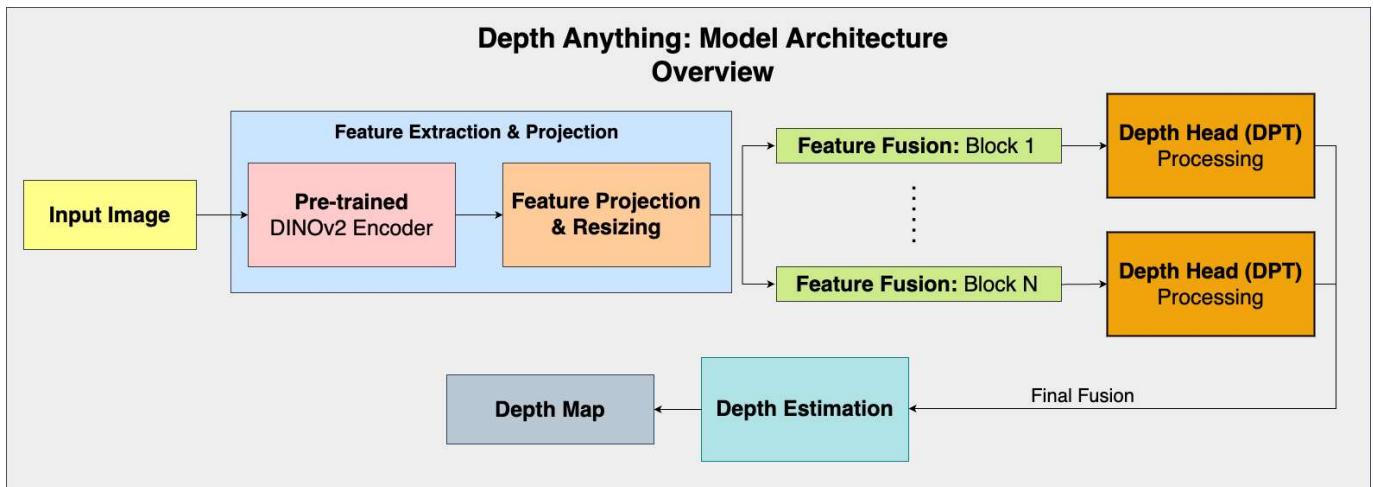


FIGURE 3.5 – Depth Anything Model Architecture

- Entrée d'Image :** L'étape initiale du modèle consiste en la réception d'une image d'entrée, qui peut être une photographie ou une capture vidéo provenant de divers dispositifs tels que des caméras de smartphones ou des systèmes embarqués dans des véhicules autonomes. Cette image sert de base pour l'extraction de données visuelles qui seront analysées pour la détermination de la profondeur.
- Extraction et Projection de Caractéristiques :** Cette phase du processus utilise un encodeur pré-entraîné, le DINov2, un type de réseau de neurones convolutionnel (CNN) conçu pour identifier et extraire des caractéristiques pertinentes à partir de l'image d'entrée. Ces caractéristiques incluent des éléments visuels tels que les contours, les textures et les couleurs, qui sont cruciaux pour l'estimation de la profondeur.
- Projection et redimensionnement des caractéristiques :** Après l'extraction, les caractéristiques sont projetées dans un nouvel espace de caractéristiques et redimensionnées pour norma-

liser les dimensions des données. Cette normalisation est nécessaire pour assurer la compatibilité et l'efficacité des étapes suivantes, particulièrement lors de la fusion des caractéristiques.

4. **Fusion de Caractéristiques** : Les caractéristiques projetées subissent un processus de fusion dans plusieurs blocs séquentiels (de Block 1 à Block N). Chaque bloc fusionne les caractéristiques en utilisant des techniques avancées telles que la sommation pondérée ou la concaténation. Cette étape est essentielle pour intégrer les caractéristiques à différentes échelles et contextualiser les données spatiales, ce qui enrichit la précision de la carte de profondeur finale.
5. **Traitement des Têtes de Profondeur (Depth Head Processing)** : Chaque tête de profondeur traite les caractéristiques fusionnées pour estimer la profondeur des différentes régions de l'image. Ce traitement utilise soit des réseaux de neurones à convolution, soit des réseaux entièrement connectés pour calculer la distance relative des objets détectés par rapport au point de vue de la caméra. L'optimisation des prédictions de profondeur à cette étape est cruciale pour la précision globale du système.
6. **Estimation de Profondeur Finale** : La phase finale du modèle consiste en la fusion finale de toutes les estimations de profondeur issues des différentes têtes de profondeur. Cette fusion compile les diverses estimations en une carte de profondeur cohérente, utilisant des techniques sophistiquées pour résoudre les incertitudes et harmoniser les prédictions.
7. **Carte de Profondeur** : La carte de profondeur résultante représente graphiquement la distance de chaque pixel ou région de l'image par rapport à la caméra, avec des gradations de couleur ou d'intensité pour indiquer différentes distances. Cette carte est fondamentale pour des applications nécessitant une perception précise de l'espace tridimensionnel, comme la navigation autonome ou la robotique avancée.

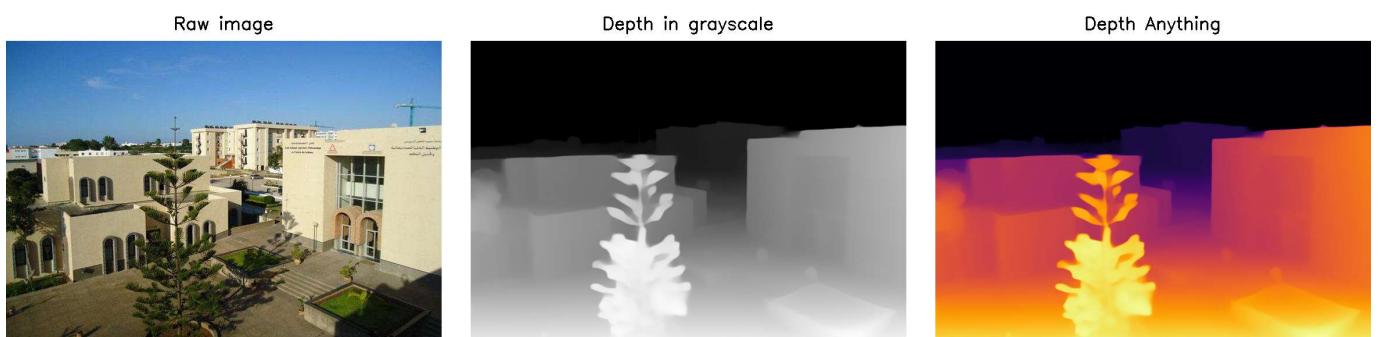


FIGURE 3.6 – Depth Anything Example

Avantages du Modèle Depth Anything

1. **Précision** : Le modèle offre une estimation de profondeur extrêmement précise grâce à l'utilisation de techniques avancées d'apprentissage automatique et de fusion de caractéristiques.
2. **Compatibilité** : Il est compatible avec diverses sources d'images, y compris les caméras de

smartphones et les systèmes embarqués dans les véhicules autonomes, rendant son utilisation flexible et adaptable.

3. **Efficacité** : Les étapes de normalisation et de fusion des caractéristiques assurent une gestion efficace des données, ce qui améliore la performance globale du système.

Inconvénients du Modèle Depth Anything

1. **Complexité** : Le modèle est complexe et nécessite des ressources de calcul significatives, ce qui peut être un obstacle pour certaines applications à faible budget.
2. **Dépendance aux Données d’Entraînement** : La performance du modèle dépend fortement de la qualité et de la diversité des données d’entraînement, ce qui peut limiter sa généralisation à des environnements non vus.
3. **Temps de Traitement** : Le temps de traitement peut être élevé, surtout pour les images haute résolution ou les séquences vidéo, ce qui peut ne pas convenir aux applications en temps réel.

Pour maximiser les bénéfices apportés par Depth Anything, il est essentiel de transformer les informations de profondeur en représentations visuelles faciles à interpréter. C'est ici qu'intervient le concept de la segmentation de l'image en zones distinctes. L'une des méthodes efficaces pour réaliser cette segmentation est l'utilisation de trapèzes semi-transparents superposés sur l'image.

3.2.1.2 YOLOv8

Dans notre projet sur la prévention des collisions en temps réel, YOLOv8 émerge comme une technologie cruciale, adaptant l'efficacité de l'algorithme YOLO à l'analyse des poses humaines pour renforcer la surveillance et l'interaction homme-machine. Cette partie détaille comment YOLOv8 améliore notre système en fournissant une reconnaissance rapide et précise des mouvements humains, une fonction essentielle pour prévenir les collisions en temps réel. En explorant son architecture et ses performances, nous soulignons son impact significatif sur l'avancement des systèmes de reconnaissance de mouvements, crucial pour la sécurité et l'efficacité des environnements dynamiques où coexistent humains et machines.

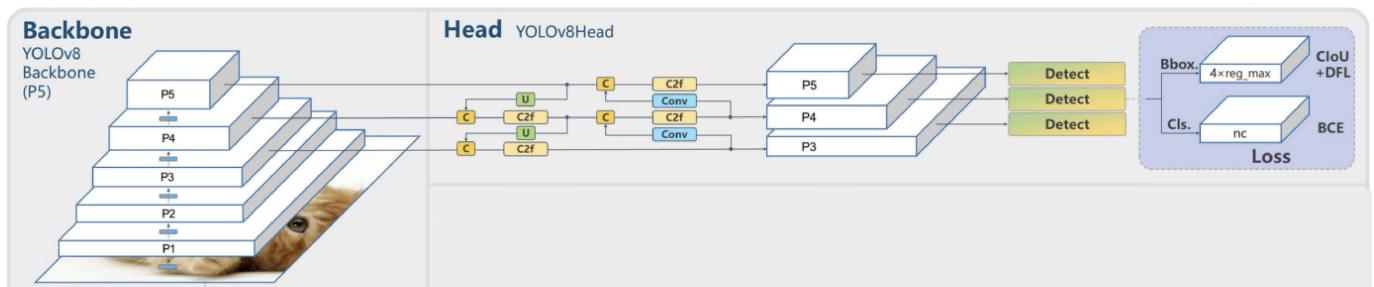


FIGURE 3.7 – Yolov8 Architecture

1. Backbone : Le backbone est responsable de l'extraction des caractéristiques des images d'entrée. Il s'agit généralement d'un réseau de neurones convolutionnel profond. Dans YOLOv8, ce backbone est appelé "YOLOv8 Backbone" et le diagramme le dénote en utilisant des niveaux de pyramide P1 à P5. Ces niveaux représentent différentes échelles d'extraction de caractéristiques, où :

P_1 est l'échelle la plus petite avec les caractéristiques de plus haute résolution, utile pour détecter de petits objets.

P_5 est l'échelle la plus grande avec les caractéristiques de plus basse résolution, utile pour détecter de grands objets.

2. Neck : Le cou est une série de couches utilisées pour raffiner les caractéristiques extraites par le backbone et les préparer pour la détection d'objets précise. Dans les architectures YOLO, cela implique généralement des couches convolutionnelles supplémentaires et des couches qui ajustent l'échelle des cartes de caractéristiques pour les rendre plus adaptées à la détection d'objets à différentes échelles.

3. Head : La tête du modèle est l'ensemble final de couches qui prédit les boîtes englobantes et les probabilités de classe pour chaque objet. La tête de YOLOv8 semble utiliser plusieurs couches de détection, chacune correspondant à une échelle différente de cartes de caractéristiques (P3, P4, P5). Ces couches se composent de :

C (Couches Convolutionnelles) pour traiter les caractéristiques.

U (Upsampling, suréchantillonnage) pour correspondre à l'échelle de différentes couches de caractéristiques.

$C2F$ (peut-être une couche ou un module personnalisé spécifique à YOLOv8) pour affiner davantage les caractéristiques avant la détection.

4. Détection : Chaque bloc de détection traite la carte de caractéristiques pour prédire :

Boîtes englobantes (Bbox), qui sont les coordonnées du rectangle entourant chaque objet détecté.

Probabilités de classe des objets (Cls.), qui prédisent la probabilité de chaque classe présente dans la boîte englobante.

Calcul de perte, qui utilise l'entropie croisée binaire (BCE) pour les probabilités de classe, et une perte combinée IoU et apprentissage des caractéristiques en profondeur (Clou + DFL) pour les boîtes englobantes.

5. Loss Function : BCE (Binary Cross-Entropy, Entropie Croisée Binaire) : Utilisée pour calculer la perte sur les prédictions de classe. Clou + DFL : Il s'agit probablement d'une combinaison de "Complete IoU" (une mesure de chevauchement entre les boîtes englobantes prédites et réelles) et d'un mécanisme "Depth-wise Feature Learning" (apprentissage des caractéristiques en profondeur) qui aide à affiner la précision de localisation.



FIGURE 3.8 – Image avec détection d'objet YOLOv8

3.2.2 Méthodologie de perspective Trapézoidale

La segmentation d'une image en utilisant des trapèzes permet de diviser l'espace visuel en trois zones distinctes : la zone proche, la zone moyenne et la zone éloignée. Chaque zone est définie en fonction de la distance relative des objets par rapport à la caméra, déterminée grâce aux données de profondeur. Cette méthode offre plusieurs avantages :

1. **Clarté Visuelle** : Les trapèzes semi-transparents de différentes couleurs (vert pour proche, jaune pour moyenne, rouge pour éloignée) permettent de visualiser clairement les différentes zones de l'image. Cette visualisation aide à comprendre rapidement la disposition des objets dans l'espace.
2. **Aide à la Prise de Décision** : En distinguant les objets en fonction de leur distance, les systèmes automatisés, tels que les véhicules autonomes ou les robots, peuvent prendre des décisions plus éclairées pour éviter des obstacles ou naviguer dans un environnement complexe.
3. **Analyse et Interprétation Améliorées** : La segmentation en zones facilite l'analyse de l'image en permettant de se concentrer sur des parties spécifiques en fonction de leur distance. Cela peut être particulièrement utile dans les domaines de la surveillance, de la navigation et de la robotique. Fonctionnement du Code de Trapèze

Le code que nous avons développé permet de dessiner des trapèzes semi-transparents sur une image pour représenter ces trois zones. Voici les étapes principales du fonctionnement du code :

1. Préparation de l'Image :

- (a) Initialisation : Une copie de l'image d'origine est créée pour le traitement afin de préserver l'image originale. Cette étape est cruciale car elle garantit que l'image initiale reste intacte, permettant ainsi d'effectuer des modifications et des analyses sans altérer les données brutes. La copie de l'image est ensuite utilisée pour toutes les opérations suivantes, ce qui permet de manipuler les pixels et d'appliquer des transformations sans risque de perdre les informations originales.

- (b) Calcul des Coordonnées : Les dimensions de l'image sont utilisées pour calculer les coordonnées des sommets du trapèze de base. Cela implique de déterminer la largeur et la hauteur de l'image, puis d'utiliser ces valeurs pour définir les points qui formeront les sommets du trapèze. Ces coordonnées initiales servent de base pour la détection des couleurs et sont ajustées en fonction des résultats de l'analyse des pixels. Le trapèze de base est essentiel pour segmenter l'image en zones distinctes et pour faciliter la visualisation des différentes distances.

2. Détection des Couleurs :

- (a) Analyse des Pixels : Chaque ligne de pixels à partir du milieu de l'image est analysée pour détecter les couleurs jaunes et rouges. Cette analyse commence à partir du point médian de l'image et se poursuit vers le bas, en examinant chaque pixel pour déterminer s'il correspond aux critères de couleur spécifiés. Les pixels sont évalués en fonction de leur intensité et de leur teinte pour identifier ceux qui sont jaunes ou rouges. Cette étape est cruciale pour la segmentation de l'image en zones de distance, car elle permet de localiser précisément les régions contenant les couleurs d'intérêt.
- (b) Mise à Jour des Coordonnées : Si des pixels jaunes ou rouges sont détectés, les coordonnées des sommets du trapèze sont ajustées pour inclure ces pixels. Lorsque des pixels correspondant aux couleurs spécifiées sont trouvés, les points des sommets du trapèze sont déplacés pour englober ces zones colorées. Cela permet de définir des zones spécifiques (jaune pour la zone moyenne, rouge pour la zone proche, et verte pour la zone éloignée) basées sur la détection des couleurs. Les coordonnées ajustées assurent que les trapèzes dessinés sur l'image reflètent fidèlement la distribution des couleurs et, par conséquent, la segmentation des distances.

3. Dessin des Trapèzes :

- (a) Fonction de Dessin : Une fonction dédiée est utilisée pour dessiner les trapèzes semi-transparents sur l'image. Cette fonction prend en entrée les coordonnées des sommets du trapèze et la couleur souhaitée, puis utilise ces informations pour remplir la zone définie par le trapèze avec la couleur spécifiée. La fonction utilise des bibliothèques de traitement d'image pour dessiner les formes de manière précise et efficace, garantissant que les trapèzes sont correctement positionnés et dimensionnés sur l'image.
- (b) Superposition : Les trapèzes sont superposés sur l'image en utilisant une transparence pour permettre de voir à travers et conserver une bonne visibilité de l'image originale. L'utilisation de la transparence permet de superposer les trapèzes colorés sur l'image sans masquer complètement les détails sous-jacents. Cela crée une visualisation efficace où les zones colorées indiquent les distances relatives tout en laissant visibles les caractéristiques importantes de l'image originale. Cette technique de superposition améliore la clarté et l'utilité des informations affichées, facilitant ainsi l'interprétation visuelle.

3.2.2.1 Résultats et Commentaires



FIGURE 3.9 – Exemple 1 du trapèze appliqué à Depth Anything.

Sur cette illustration, nous visualisons l’application de notre modèle de profondeur ‘Depth Anything’, qui a permis de créer une perspective en trapèze dynamique adaptative basée sur les données de profondeur obtenues. Cette perspective est segmentée en trois zones distinctes, chacune représentant un niveau différent de risque ou de proximité par rapport à la source de détection. Ce découpage en zones est particulièrement visible ici grâce à la distance notable des personnes par rapport à la source, ce qui permet une évaluation claire et structurée du risque de collision. L’image de droite montre clairement comment ces zones, colorées différemment, facilitent l’interprétation rapide et efficace des données pour une action préventive immédiate.

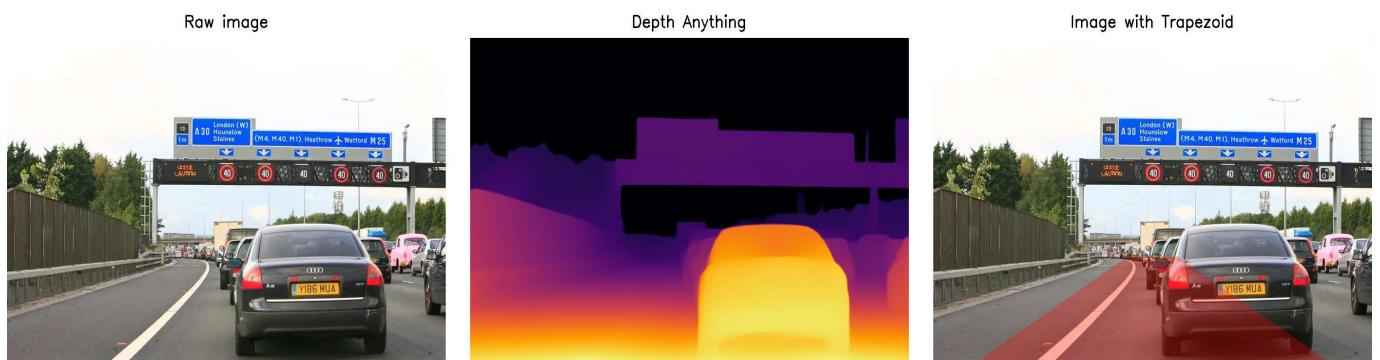


FIGURE 3.10 – Exemple 2 du trapèze appliqué à Depth Anything.

Alors que dans cette image, on voit que le trapèze tout rouge justifié par la présence du véhicule très proche du système de détection, à l’aide de YOLO et de l’algorithme de l’intersection, on pourra prévenir l’accident dans ce cas .



FIGURE 3.11 – Exemple 3 du trapèze appliqué à Depth Anything.

Dans cette image, la fille qui se trouve au milieu est très proche du système de détection ce qui rend le trapèze tout en rouge montrant le degré de danger.

3.2.3 Traitement en Temps Réel

Le traitement en temps réel fait référence à la capacité d'un système à traiter les données instantanément à mesure qu'elles sont reçues, assurant une réponse immédiate. Cette capacité est cruciale dans des domaines comme la robotique, les véhicules autonomes, où les délais peuvent avoir des conséquences importantes.

Pour accomplir le traitement en temps réel, plusieurs éléments sont essentiels :

1. **Algorithmes Optimisés** : Conçus pour traiter les données rapidement et efficacement.
2. **Matériel Haute Performance** : Utilisation de GPU, ou autres accélérateurs matériels.

3.2.3.1 Flux de Traitement en Temps Réel

Le flux de traitement en temps réel comprend plusieurs étapes clés, depuis l'entrée vidéo jusqu'à l'affichage des résultats. Voici un schéma simplifié et les étapes détaillées.

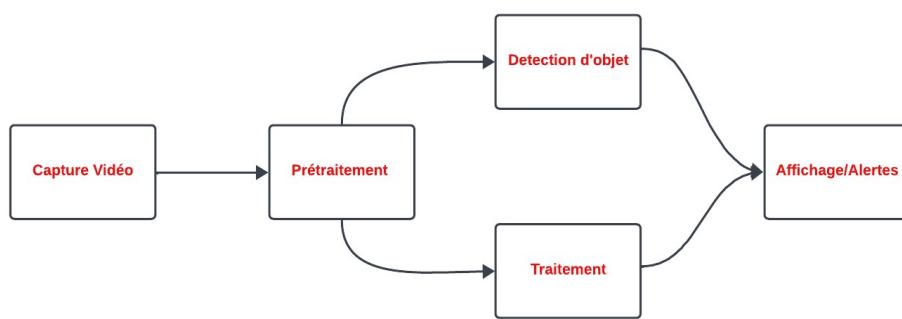


FIGURE 3.12 – Flux de Traitement en Temps Réel

1. **Capture Vidéo** : Les caméras ou capteurs capturent le flux vidéo en direct.
2. **Prétraitement** : Les données vidéo sont nettoyées et préparées pour le traitement.

3. **Détection d'Objets** : Les algorithmes, comme YOLO Pose, identifient les obstacles potentiels.
4. **Traitement** : Le système évalue la criticité des événements basés sur la distance et la gravité.
5. **Affichage/Alertes** : Les résultats sont affichés en temps réel avec des alertes visuelles ou sonores.

3.2.4 Intégration et Réalisation du projet

Dans cette sous-section, nous détaillons la réalisation du projet en mettant en lumière les éléments clés qui ont permis d'atteindre nos objectifs en matière de prévention des collisions en temps réel.

3.2.4.1 Combinaison de la Perspective Trapézoïdale

Nous avons utilisé le modèle "Depth Anything" pour générer une perspective trapézoïdale, définissant ainsi une zone de prévention des collisions en trois dimensions. Cette technique nous a permis de tenir compte de la profondeur et de la distance des objets dans la scène. Le trapèze ainsi formé représente les différentes distances par rapport à la caméra, permettant une analyse précise des positions relatives des objets.

3.2.4.2 Détection des Personnes avec YOLO

En parallèle, nous avons implémenté l'algorithme YOLO (You Only Look Once) pour la détection des personnes. Cet algorithme est particulièrement efficace pour identifier rapidement les objets et dessiner des rectangles englobants autour des personnes détectées. Ces rectangles englobants facilitent le suivi des mouvements des personnes et leur position dans l'espace.



FIGURE 3.13 – YOLO et Trapèze

Dans cette figure, on voit qu'on a pu combiné les deux approches : YOLOv8 et perspective trapezoidale, et on peut détecter ainsi chaque objet et personne à partir de son *Bounding box* de savoir dans quelle zone du trapèze il se trouve et poursuite déclencher des alertes .

3.2.4.3 Traitement en Temps Réel

L'un des défis majeurs de notre projet était de garantir un traitement en temps réel des données. Nous avons optimisé nos algorithmes pour qu'ils puissent analyser en continu les flux vidéo et détecter

les collisions sans latence significative. Cette réactivité est cruciale pour des applications telles que la robotique, les véhicules autonomes.

3.2.4.4 Gestion des Intersections

Nous avons développé une méthode pour analyser les intersections entre les rectangles englobants des personnes et la perspective trapézoïdale. Chaque intersection est évaluée en fonction de la distance par rapport à la caméra, en priorisant les zones les plus proches. Cette priorisation permet de générer des alertes de manière cohérente, en mettant l'accent sur les collisions potentielles les plus immédiates et critiques.

3.2.4.5 Génération des Alertes

En fonction des intersections détectées, le système génère des alertes visuelles et sonores. Ces alertes sont adaptées à chaque zone de détection, avec des niveaux d'urgence variables en fonction de la proximité et de la gravité de la collision potentielle. Par exemple, une collision imminente dans une zone proche déclenche une alerte sonore et visuelle plus intense qu'une collision possible dans une zone plus éloignée.

Grâce à cette combinaison de technologies et à notre approche méthodique, nous avons réussi à créer un système de prévention des collisions en temps réel performant et fiable. Ce système est capable de fournir des alertes précises et immédiates, contribuant ainsi à la sécurité et à l'efficacité dans divers environnements d'application. Notre travail sur ce projet a également permis d'améliorer notre compréhension des techniques d'intelligence artificielle et de vision par ordinateur, tout en posant les bases pour des avancées futures dans la technologie de prévention des collisions.

Chapitre 4

Discussion

Dans ce chapitre, nous discuterons en détail la solution proposée pour la prévention des collisions en temps réel, en analysant ses avantages ainsi que ses inconvénients.

4.1 Avantages de la Solution Proposée

1. **Précision de la Détection :** Grâce à l'utilisation combinée du modèle "Depth Anything" pour générer des perspectives trapézoïdales et de l'algorithme YOLO pour la détection des personnes, notre système bénéficie d'une précision accrue. Cette combinaison permet de prendre en compte à la fois la position et la profondeur des objets, offrant une détection fiable et exacte des collisions potentielles.
2. **Traitements en Temps Réel :** L'optimisation de nos algorithmes pour le traitement en temps réel est un avantage majeur. Cela permet une réponse immédiate aux événements, crucial pour les applications où chaque milliseconde compte, comme dans les véhicules autonomes.
3. **Alertes Graduellement Adaptées :** La priorisation des alertes en fonction de la distance et de la gravité de l'intersection entre les rectangles englobants et la perspective trapézoïdale permet une gestion efficace des situations critiques. Les alertes visuelles et sonores sont adaptées à la proximité et à l'urgence des collisions potentielles, assurant ainsi une réponse appropriée et immédiate.
4. **Polyvalence et Extensibilité :** Le système est conçu pour être facilement extensible, permettant l'ajout de nouvelles fonctionnalités et l'intégration avec d'autres systèmes. Cette flexibilité assure que notre solution peut évoluer avec les besoins futurs et s'adapter à différents environnements et applications.

4.2 Inconvénients de la Solution Proposée

1. **Complexité de Mise en Œuvre :** La combinaison de technologies avancées comme "Depth Anything" et YOLO requiert une expertise technique élevée. Cette complexité peut poser des défis lors de la mise en œuvre et nécessiter des ressources importantes en termes de temps et de compétences.
2. **Dépendance à l'Infrastructure Matérielle :** Pour fonctionner efficacement, notre système nécessite une infrastructure matérielle robuste, incluant des capteurs de haute qualité et des processeurs performants pour le traitement en temps réel. Cette dépendance peut augmenter les coûts et limiter l'accessibilité de la solution pour certaines applications ou environnements moins bien équipés.
3. **Sensibilité aux Conditions Environnementales :** La performance du système peut être affectée par des conditions environnementales variables, telles que l'éclairage ou les obstacles non prévus dans le champ de vision. Ces facteurs peuvent diminuer la précision de la détection et nécessiter des ajustements supplémentaires pour maintenir une performance optimale.

En conclusion, notre solution de prévention des collisions en temps réel offre de nombreux avantages en termes de précision, réactivité et adaptabilité. Cependant, elle présente également des défis, notamment en termes de complexité, de dépendance matérielle et de sensibilité aux conditions environnementales. Une évaluation continue et des ajustements seront nécessaires pour maximiser les avantages tout en minimisant les inconvénients, assurant ainsi une application efficace et fiable dans divers contextes.

Chapitre 5

Conclusion

Dans ce projet modeste, nous avons réussi à développer un système de prévention des collisions et de prévision d'accidents en temps réel utilisant l'intelligence artificielle, marquant notre première incursion dans le domaine de l'IA. L'accent mis sur les modèles en temps réel était essentiel, garantissant la réactivité et l'efficacité du système, qui est devenu la base pour l'intégration de techniques d'IA avancées.

Au cœur de notre méthodologie se trouvait le modèle "Depth Anything", qui a créé une perspective trapézoïdale essentielle pour évaluer avec précision les collisions potentielles. Nous avons intégré l'algorithme YOLO (You Only Look Once), connu pour sa capacité à détecter rapidement et précisément les objets. Cela a été crucial pour notre capacité à identifier efficacement les obstacles et à les encadrer dans des boîtes englobantes, améliorant ainsi grandement notre précision dans la prédiction des collisions en analysant les relations spatiales et les mouvements au sein de ces zones définies.

Nos efforts ont abouti à un processus d'optimisation rigoureux pour maintenir les normes opérationnelles en temps réel, atteignant un équilibre entre vitesse et précision. Grâce à un raffinement et à des tests continus, nous avons développé un système qui non seulement répond, mais dépasse les attentes actuelles pour les applications de l'IA dans les technologies de sécurité.

Futurs Améliorations :

À l'avenir, plusieurs améliorations stratégiques pourraient encore augmenter les capacités de notre système de prévention des collisions et de prévision d'accidents :

- 1. Implémentation d'algorithmes d'apprentissage automatique pour la détection d'anomalies** : Utiliser l'apprentissage automatique pour détecter les modèles et les anomalies dans les données de conduite pourrait permettre à notre système d'anticiper les collisions potentielles avant que les risques évidents ne se manifestent, améliorant ainsi les capacités préventives.
- 2. Apprentissage et adaptation en temps réel** : En permettant au système d'apprendre à partir de nouvelles données en temps réel, nous pouvons améliorer continuellement sa précision

prédictive, assurant une adaptabilité aux nouveaux scénarios et conditions sans nécessiter de mises à jour manuelles.

3. **Déploiement sur des dispositifs de calcul de pointe plus puissants :** L'utilisation de matériel informatique avancé peut réduire la latence et augmenter la puissance de traitement, garantissant des prédictions plus rapides et plus précises, cruciales pour les applications en temps réel.

Ces développements futurs promettent non seulement de raffiner l'efficacité et l'efficience de notre système, mais aussi d'élargir son champ d'application, ayant un impact significatif sur la garantie de solutions de navigation plus sûres sur diverses plateformes.

Bibliographie

- [1] Depth-Anything . , lien : <https://start.spring.io/>
- [2] YOLO Pose Model *YOLO*, lien : [https://docs.ultralytics.com/tasks/pose//](https://docs.ultralytics.com/tasks/pose/)
- [3] RADAR and Lidar sensors for ADAS . lien : <https://www.foresightauto.com/an-overview-of-autonomous-sensors-lidar-radar-and-cameras/>
- [4] Impact of AI in object detection . lien : <https://www.researchgate.net/publication/345378039> *research*, *net*
- [5] Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks . lien : <https://arxiv.org/abs/1506.01497>
- [6] Deep SORT : Simple Online and Realtime Tracking . lien : <https://arxiv.org/abs/1703.07402>
- [7] TensorFlow Lite . lien : <https://www.tensorflow.org/lite/guide?hl=fr>

Nous exprimons notre gratitude envers les auteurs et les chercheurs dont les travaux ont été une source précieuse d'information pour mon projet.

© AHAKAY ISMAIL & ANNA AYAMAN 2024
All rights reserved.