REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IA26/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence Artificielle (IA)**

Conception et réalisation d'un prototype de véhicule autonome connecté

Par :

BOUZEKRI AFAF

Soutenu le 28/06/2022 devant le jury composé de :

KAZAR Okba	Pr	Président
ZOUAI Meftah	MAB	Rapporteur
BEN DAHMENE Asma	MAB	Examinateur

Année universitaire 2021-2022

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة محمد خيضر بسكرة

تصريح شرفي

(خاص بالالتزام بقواعد النزاهة العلمية لإنجاز بحث)

أنا الممضي أسفله،

السيد (ة): **بوزكري عفاف** الصفة: **طالبة**

الحامل لبطاقة التعريف الوطنية رقم: **208007657** والصادرة بتاريخ: **08/06/2022**

المسجل بكلية: **العلوم الدقيقة وعلوم الطبيعة والحياة**

قسم: **الإعلام الآلي**

و المكلفة بإنجاز مذكرة تخرج في الماستر عنوانها:

Conception et réalisation d'un prototype de véhicule autonome connecté.

أصرح بشرفي أني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات المهنية والنزاهة الأكاديمية المطلوبة في إنجاز البحث المذكور أعلاه.

التاريخ: .....**2022/06/20**.....

توقيع المعني:

Remerciements

Louanges à Dieu, qui m'a accordé le succès et le remboursement et m'a donné la persévérance et m'a aidé à terminer ce travail.

En premier lieu, j'adresse mes sincères remerciements et ma gratitude à mon superviseur ***Dr Meftah ZOUAI****, pour ses efforts considérables et ses précieux conseils tout au long d'une période de la réalisation de ce projet.*

Ensuit je tiens à remercier le chef département ,et tous les enseignants de l'informatique surtout les enseignants de l'IA.

Afaf BOUZEKRI

Dédicaces

Je dédie ce modeste travail :

À mes chers parents pour leur soutien, leur fatigue et leurs sacrifices pour moi en tout temps et tout au long de mon parcours universitaire,

À mes chers frères et soeurs pour leur soutien et leurs encouragements,
À mon défunt frère qui m'a toujours soutenu et encouragé,
À mon collègue ***Mohamed Younes SENOUSSI****, mes chers amis* ***Imane KABISSI****,* ***Chaima DJEFFAL****,*
Mahdia Fadhila Adjal*,* ***Nadjat Khergag***
À tous mes amis.

ملخص

تعتبر السيارة وسيلة ضرورية في الحياة اليومية للإنسان، حيث نجد في بعض الأحيان كل أفراد العائلة الواحدة تقريبًا يمتلكون سياراتهم الخاصة بالرغم من أنهم لا يستخدمونها في أغلب الأحيان، بينما يمكنهم استبدال هذا العدد من السيارات بالسيارة التي يمكن أن تؤدي المهام التي تطلب منها مثل إيصال البالغين إلى العمل ثم أخذ الأطفال إلى المدرسة دون الحاجة إلى وجود سائق في السيارة، مما يقلل من تكاليف النقل والوقت ويحسن السلامة على الطريق بفضل أنظمة السيارة الذكية.

في مشروعنا، نظرا لتكلفة القيام بإنشاء سيارة ذاتية القيادة حقيقية لاستخدامها في أعمال بحثية وتعريف الطلبة بأنظمة القيادة المدمجة الذكية، قمنا بصنع نموذج أولي لسيارة ذاتية القيادة يحتوي على أهم الأجهزة الموجودة في السيارات ذاتية القيادة، حيث يمكن لهذا النموذج المصغر استشعار البيئة المحيطة به، كما يمكنها اكتشاف الطريق والعوائق وعلامات المرور بالاعتماد على التعلم العميق لتأمين حركة السيارة.

كلمات مفتاحية: سيارة ذاتية القيادة، التعلم العميق، استشعار المسافة، اكتشاف الطريق، العوائق، علامات المرور.

Résumé

La voiture est un moyen nécessaire dans la vie quotidienne de l'être humain, car on trouve parfois que presque tous les membres d'une même famille possèdent leur propre voiture, bien qu'ils ne l'utilisent pas la plupart du temps, alors qu'ils peuvent remplacer ce nombre de voitures avec une voiture capable d'effectuer les tâches que vous lui demandez, comme amener les adultes au travail puis emmener les enfants à l'école sans avoir besoin d'un chauffeur dans la voiture, ce qui réduit les coûts et le temps de transport et améliore la sécurité sur la route grâce à smart systèmes automobiles.

Dans notre projet, compte tenu du coût de la création d'une véritable voiture autonome à utiliser dans les travaux de recherche et de l'initiation des étudiants aux systèmes de conduite intégrés intelligents, nous avons réalisé un prototype de voiture autonome qui contient les dispositifs les plus importants trouvés dans l'auto- conduire des voitures, car ce modèle miniature peut détecter l'environnement, ainsi qu'il peut détecter les routes, les obstacles et les panneaux de signalisation en s'appuyant sur l'apprentissage en profondeur pour sécuriser le mouvement du véhicule.

Mots-clés :voiture autonome, apprentissage en profondeur, détection de distance, détection de route, obstacles, panneaux de signalisation.

Abstract

The car is a necessary means in the daily life of the human being, as we sometimes find almost all members of the same family own their own car, although they do not use it most of the time, while they can replace this number of cars with a car that can perform the tasks you ask of it, such as delivering adults to work and then take children to school without the need for a driver in the car, which reduces transportation costs and time and improves safety on the road thanks to smart car systems.

In our project, given the cost of creating a real self-driving car to be used in research work and introducing students to smart integrated driving systems, we made a prototype of a self-driving car that contains the most important devices found in self-driving cars, as this miniature model can sense the surrounding environment, as well as It can detect road, obstruction and traffic signs by drawing on deep learning to secure the vehicle's movement.

Keywords :self-driving car, deep learning, distance sensing, road detection, obstacles, traffic signs.

Table des matières

Table des figures

Liste des Algorithmes

Liste des codes sources

Introduction Générale

Après le grand développement qu'a connu Internet à la fin des années quatre vingt, il a été utilisé pour communiquer entre les objets (**IoT**), l'Internet des Objets a conduit aux progrès de la communication, ce qui a poussé les constructeurs automobiles à chercher à développer le modèle des véhicules autonomes, ce qui a conduit à l'émergence de ce que l'on appelle l'Internet des véhicules(**IoV**), qui à son tour a conduit ces derniers temps à l'accélération du développement des véhicules autonomes.

Les véhicules autonomes sont définis comme des véhicules capables de se déplacer et de se conduire sans intervention humaine grâce à de nombreuses technologies de détection intelligentes.

Problématique

À l'ère actuelle, la voiture est devenue un moyen nécessaire dans la vie quotidienne de l'individu, car nous trouvons une famille qui nécessite presque tout le monde qui possède sa propre voiture, bien qu'il ne l'utilise pas la plupart du temps, alors que nous pouvons remplacer ce nombre de voitures avec une seule voiture, qui peut effectuer les tâches que vous lui demandez. , comme amener les parents au travail puis les enfants à l'école, sans avoir besoin d'un chauffeur dans la voiture, ce qui réduit les coûts de transport, fait gagner du temps et améliore la sécurité routière,

grâce aux systèmes intelligents contenus dans la voiture.

Objectif du travail

Dans ce projet et en raison du coût élevé de la création d'une véritable voiture autonome, nous avons essayé de créer un modèle miniature de ce que contient la voiture autonome afin de mener des expériences de recherche dessus, ce qui permet d'économiser du temps et de l'argent. Ce projet consiste à développer un modèle théorique ainsi qu'un prototype de voiture équipé d'une caméra et des capteurs ultrasons et d'un ordinateur de bord. La voiture a la capacité de se déplacer librement avec une détection d'obstacles et un système d'aide pour se garer, détection de la route, détection des feux et des panneaux de signalisation.

Après l'introduction générale, la mémoire sera divisée en quatre chapitres comme suit :

Chapitre 01 : Internet des objets. nous aborderons le concept de l'internet des objets,son fonctionnement, de s'architecturer et de ses domaines d'application, etc.

Chapitre 02 : Internet des véhicules. Dans ce chapitre, nous aborderons quelques concepts de base de l'Internet des véhicules, des caractéristiques de l'Internet des véhicules, des niveaux d'autonomie et du fonctionnement et de l'évolution du véhicule autonome, etc et enfin nous mentionnerons les travaux connexes à notre projet **Chapitre 03 : Conception.** Dans ce chapitre, nous décrirons le système que nous développerons et expliquerons le fonctionnement de chaque partie de celui-ci.

Chapitre 04 : Implimentation. Ce chapitre présentera les outils de mise en œuvre et les détails du code en plus de discuter des résultats à obtenir et enfin de comparer notre travail avec des travaux connexes.

Conclusion générale et perspective.

Chapitre I

Internet des Objets

I.1 Introduction

Nous assistons à l'aube d'une nouvelle ère de l'Internet des objets de manière générale, l'IoT fait référence à l'interconnexion en réseau d'objets du quotidien, souvent équipés d'une intelligence omniprésente. L'IoT augmentera l'ubiquité d'Internet en intégrant chaque objet pour l'interaction via des systèmes embarqués, ce qui conduit à un réseau hautement distribué d'appareils communicants avec les êtres humains ainsi qu'avec d'autres appareils.

Grâce aux progrès rapides des technologies sous-jacentes, l'IoT ouvre d'énormes opportunités pour un grand nombre d'applications nouvelles qui promettent d'améliorer la qualité de nos vies. Ces dernières années, l'IoT a attiré l'attention des chercheurs et des praticiens du monde entier. Il vise à créer et à améliorer des opérations dans divers secteurs tels que les villes intelligentes, la santé et les voitures connectées et autonomes. dans ce chapitre, nous étudierons en détail l'internet des objets, ses caractéristiques et tout ce qui rapporte.

I.2 Définitions

Dans cette section, nous définissons quelques termes de base dans ce chapitre, comme suivant :

I.2.1 Internet des objets (IoT)

L'Internet des objets (IoT en anglais) est une extension d'Internet à tous les objets qui communiquent directement ou indirectement par le biais d'appareils électroniques connectés au Web. Ce nouvel aspect soulève des enjeux techniques, économiques et sociaux majeurs, sans oublier ceux liés à la gouvernance[36].

I.2.2 Objet connecté

Un objet connecté qui est une devise son objectif principal n'est pas les systèmes informatiques ou les interfaces d'accès Web. Vous permet d'intégrer votre connexion Internet dans OC Améliorez-le en matière de fonctionnalité et d'interaction avec l'environnement, il devient un OC enrichi (ECO), Les CO peuvent interagir indépendamment du monde physique sans intervention humaine. Il existe certaines limitations telles que le stockage, la bande passante, la consommation, etc. L'énergie etc... Il doit être accepté pour être utilisé, il a une certaine forme d'intelligence, la capacité de le faire Envoyez et recevez des données avec un logiciel qui utilise des capteurs embarqués. Un objets connectés sont précieux lorsqu'ils sont connectés à d'autres objets ou briques logicielles. Voici un exemple : les montres connectées n'ont d'intérêt que dans un écosystème centré sur la santé/bien-être qui va bien au-delà de la connaissance du temps[32] [29] .

I.2.3 M2M

la transmission machine à machine oriente l'association incontinentes technologies pour l'information et pour la transmission (abréviation Tic) sans incontinents objets dits intelligents et communicants et ceci sur l'objectif pour apporter de ces derniers les moyens d'interagir dépourvu action humaine sans le principe d'information ce dernier-né peut convenir indistinctement de une coordination soit de une société[4].

I.3 Fonctionnement de l'IoT

Pour avoir l'Internet des objets, tous les éléments doivent être fournis : capteurs, matériel et communication, traitement des données et interface utilisateur. Les capteurs(Pression, température, nombre de cycles, vibration, temps d'utilisation... Ils existe un type de capteur pour chaque usage.) collectent des données autour d'eux (l'environnement), pour envoyer ces données vers le cloud par plusieurs voies différentes (même fonction), dont les satellites, LPWAN..., ou via une connexion directe à Internet. Chaque méthode a ses compromis entre la consommation d'énergie, la portée et la bande passante et une fois que les données atteignent le cloud, elles sont traitées par le logiciel de traitement de données pour être ensuite utilisées d'une manière ou d'une autre par l'utilisateur[25].

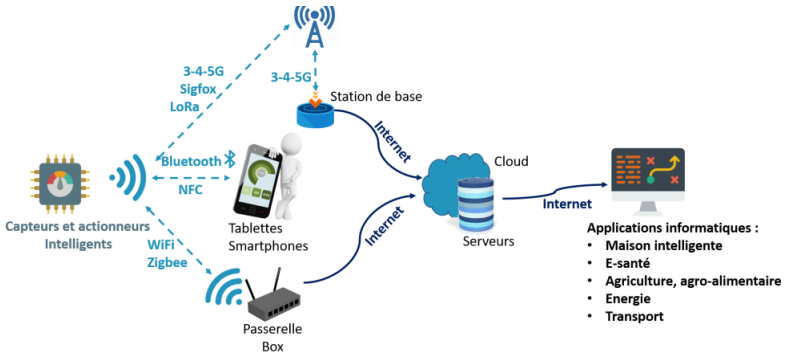

FIGURE I.1 – Fonctionnement de l'IoT [6].

I.4 Architecture de l'IoT

L'architecture du système Internet des objets se compose de plusieurs niveaux imbriqués les uns dans les autres pour relier le monde numérique (virtuel) au monde physique (tangibles) des réseaux et du cloud. L'architecture diffère d'un projet à l'autre car elle n'a pas une structure formellement identique, cependant un cheminement des données peut être prévu [30] [40].

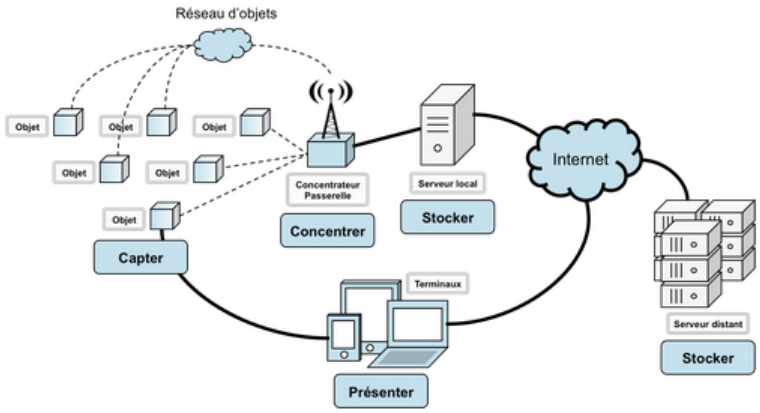

FIGURE I.2 – L'architecture de l'IoT[33].

I.5 Domaines applicatifs de l'IOT

Il existe plusieurs domaines d'application d'internet des objets[5], nous mentionnons ce qui suit :

1. Les transports : une voiture soit capable de communiquer de façon autonome avec d'autres véhicules, véhicules autonomes (sans conducteur) capables de se déplacer sans aucune intervention humaine.
2. La santé : il existe plusieurs d'applications permettant à un patient et à son docteur de recevoir des informations sans communication directe.
3. La domotique : La domotique regroupe l'ensemble des techniques permettant l'automatisation des équipements d'un habitat.
4. Agriculture : l'agriculture intelligente pour a contribué à la sécurité alimentaire en intégrant le besoin d'adaptation et le potentiel d'atténuation dans les stratégies de développement de l'agriculture durable.
5. Ville intelligente : circulation routière intelligente, transports intelligents, collecte des déchets, cartographies diverses.
6. Environnements intelligents : prédiction des séismes, détection d'incendies, qualité de l'air, etc.
7. Sécurité et gestion des urgences : attentats, radiations, explosions.
8. Logistique : aller plus loin que les approches actuelles.
9. Contrôle industriel : prédiction des pannes, mesure, dépannage à distance, pronostic.
10. Applications ludiques...

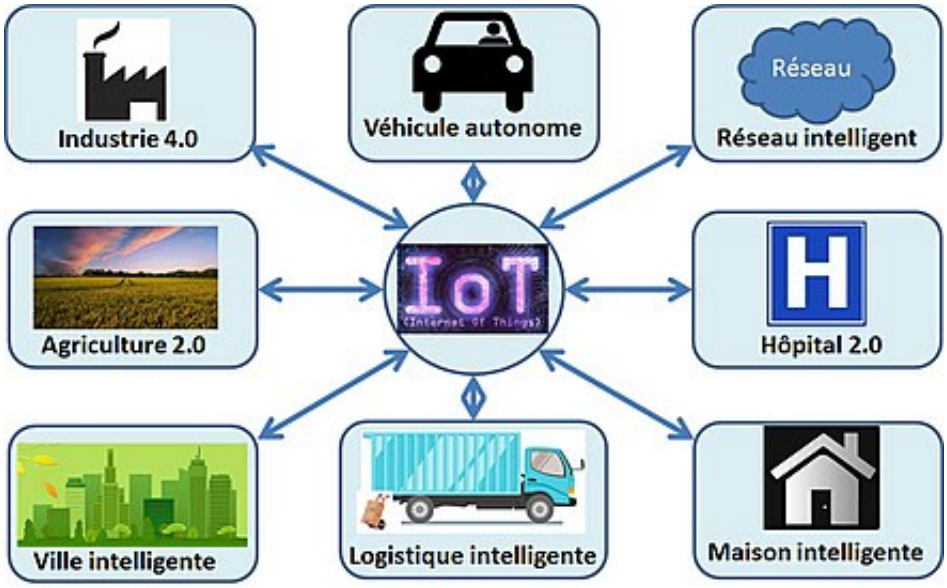

FIGURE I.3 – Domaines d'application de l'IoT[31].

I.6 Quatre piliers d'une solution IoT

La solution IoT est représentée par cette pyramide pour faciliter sa compréhension et faciliter la recherche d'un moyen de créer de la valeur immédiate[10].

1. **Device :** le hardware est un ensemble de choses, ce sont les principaux composants physiques des objets connectés : les capteurs.
2. **Connectivité :** étant donné que les hardwares sont les objets de l'Internet des objets, la connexion représente l'Internet des objets, elle est donc considérée comme un pont entre le monde physique et numérique, car elle est le noyau de la solution Internet des objets. L'objectif principal de la connectivité est de télécharger les données collectées par l'appareil vers le cloud.

3. **Data :** les capacités d'un capteur de la transformation et l'analyse et l'interprétation des données qu'il collecte lui a valu une grande valeur dans la création d'une solution IoT.
4. **Valeur :** Créer de la valeur pour l'utilisateur final est l'objectif de la solution Internet des objets.

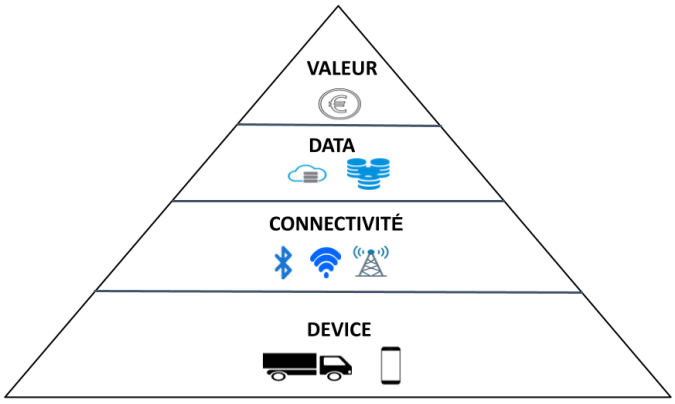

FIGURE I.4 – Pyramide de l'IoT[10].

I.7 Sécurité de l'IoT

La sécurité de l'IoT est l'une des questions les plus importantes, car c'est le domaine spécialisé dans la protection des réseaux et des appareils connectés à l'IoT. Négliger la sécurité conduit attaque des appareils à moins qu'ils ne soient entièrement protégés. Par conséquent, la mise en œuvre des mesures de sécurité nécessaires est essentielle pour assurer la sécurité des réseaux auxquels les appareils IoT sont connectés[1].

I.7.1 Défis de l'IoT en matière de sécurité

De nombreux problèmes sont considérés comme un défi qui empêche la sécurité complète des appareils Internet dans l'environnement de l'Internet des objets[34]. Parmi ces problèmes, il existe plusieurs problèmes majeurs communs aux appareils Internet, notamment :

1. Utiliser des mots de passe cryptés qui peuvent être exposés à une faille de sécurité même s'ils ont été modifiés.
2. Appareils restreints avec ressources et manque de ressources de calculs nécessaires pour mettre en œuvre la sécurité, donc beaucoup d'appareils pour fournir des fonctionnalités de sécurité avancées (hardcore). Par exemple, les capteurs qui surveillent l'humidité et la température ne peuvent pas gérer le cryptage avancé et d'autres mesures de sécurité.
3. Absence de normes de sécurité acceptées par l'industrie malgré les cadres IoT existants. Par conséquent, il n'y a pas de cadre convenu entre les grandes entreprises et les entreprises industrielles.

I.7.2 Industries les plus vulnérables aux menaces de sécurité IoT

Les failles de sécurité IoT peuvent survenir dans n'importe quel secteur : maison intelligente, voiture connectée... Le système individuel est grandement affecté par les données et les informations qu'il contient. L'attaque pourrait être la désactivation des freins d'une voiture connectée ou d'un appareil de santé connectée (pompe à insuline) entraînant une menace pour la vie de l'individu[9].

FIGURE I.5 – Industries de sécurité IoT[16].

I.8 Technologies clés génériques

L'Internet des objets fonctionne grâce à plusieurs technologies telles que : RCSF, Cloud computing, systèmes embarqués, nanotechnologie, etc.[27].

1. **Les réseaux de capteurs sans fil RCSF (WSN) :** est la technique la plus importante, et consiste d'un ensemble de noeuds capteurs. Elles sont organisées en champs qui ont des fonctionnalités de capturer et traiter et transmettre les données. Les nœuds capteurs permettent la représentation des caractéristiques dynamiques des objets ou du champ de déploiement.
2. **Cloud Computing :** Il fournit un espace de stockage des données IoT, et fournit également divers autres services et ressources en ligne (visualiser des données, analyser...).
3. **Big Data :** Analysez les données avec des outils avancés collectés par les ob-

jets IoT.

4. **Les protocoles de communication :** Il assure la communication entre les objets et les applications, les protocoles spécifiant le format des données, la taille des paquets, l'adressage et le routage.
5. **Les systèmes embarqués :** les composants de base des objets Connectés sont des cartes à microcontrôleur qui intègrent un microprocesseur, ainsi que des mémoires, des ports d'entrée et de sortie pour connecter des capteurs.
6. **Identification par radio fréquence (RFID) :** Identification est suivie des objets, représentation des informations statiques des objets. Identificateur, description, date d'expiration etc.

I.9 Écosystème des technologies IoT

L'écosystème des techniques IoT se contient de 4 couches [17] suivants : appareils, données, métier et utilisateurs.

1. **couche appareils :** un ensemble de capteurs, de matériel, d'actionneurs et de logiciels qui constituent un appareil qui communique et interagit avec un réseau.
2. **couche données :** données collectées, traitées, transmises, stockées, analysées, présentées et utilisées dans un cadre professionnel.
3. **couche métier :** Fonction métier de la technologie internet des objets.
4. **couche utilisateur :** Les personnes qui interagissent avec les techniques et les appareils IoT.

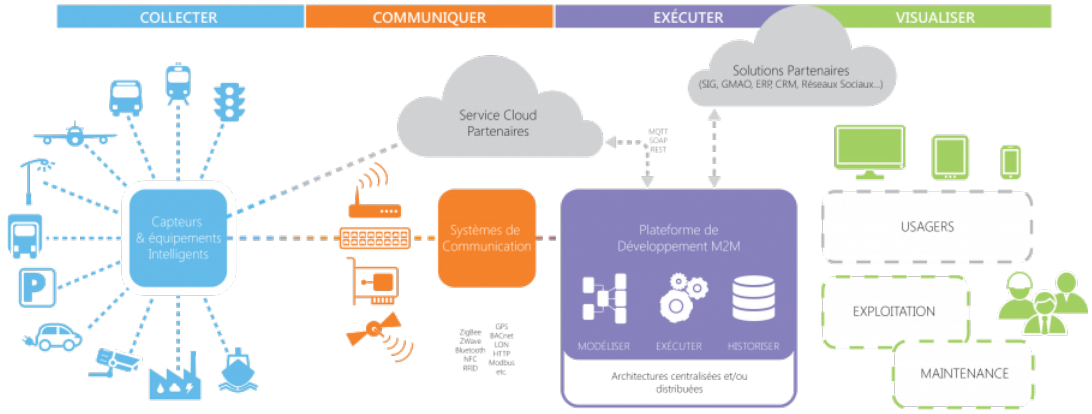

FIGURE I.6 – Écosystème de l'IoT[18].

I.10 Pile de technologies IoT

Dans cette section , nous allons présenter deux parties esséntielles : apparels IoT et les protocoles, la connectivité de l'IoT comme suivant :

I.10.1 Appareils IoT

L'Internet des objets a beaucoup varié, et les plus importants de ces appareils sont :

1. **Actionneurs :** ce sont des dispositifs qui convertissent des informations numériques en phénomènes physiques. Ils effectuent des actions physiques lorsqu'ils reçoivent des instructions du centre de contrôle.
2. **Systèmes incorporés :** ces systèmes reposent sur un microprocesseur ou un microcontrôleur qui gère une fonction particulière dans un système plus vaste.

3. **Appareils intelligents :** les appareils intelligents sont des appareils électroniques qui sont connectés à des appareils ou à des réseaux via divers protocoles sans fil, ont la capacité de calculer et incluent souvent un microcontrôleur.
4. **Microcontrôleur (MCU) :** les micro-ordinateurs sont intégrés dans de minuscules puces contenant des processeurs, de la RAM et de la ROM, et les microcontrôleurs sont plus limités en puissance que les microprocesseurs.
5. **Microprocesseur (MPU) :** les microprocesseurs remplissent les fonctions de processeurs sur un ou plusieurs circuits intégrés. Bien que les microprocesseurs aient besoin de périphériques pour effectuer des tâches, ils réduisent considérablement les coûts de traitement car ils ne disposent que d'un processeur.
6. **Appareils non informatiques :** les appareils transmettant des données et communiquent uniquement, ils n'ont pas le pouvoir de calculer.
7. **Transducteurs :** les transducteurs sont des dispositifs qui convertissent des formes d'énergies d'une forme à une autre. Dans les dispositifs IoT, il s'agit notamment de capteurs et d'actionneurs internes qui transmettent des données lorsque les dispositifs interagissent avec leur environnement.
8. **Capteurs :** les capteurs détectent les changements dans leur environnement et génèrent des impulsions électriques pour communiquer, les capteurs détectent par exemple les changements de température, de composition chimique et de condition physique.

I.10.2 Protocoles et connectivité IoT

Dans cette section, nous parlerons de deux choses importantes dans l'internet des objets[20], à savoir respectivement la connexion des appareils IoT et les passerelles

IoT comme suit :

1. **Connexion des appareils IoT :** l'un des aspects clés de la planification de projet IoT consiste à définir les protocoles pour les appareils IoT, c'est-à-dire comment les appareils communiquent et se connectent dans la famille de technologies IoT, ils communiquent via des fonctionnalités intégrées ou des passerelles.
2. **les passerelles IoT :** les passerelles font partie de la technique IoT et servent à connecter les appareils IoT au cloud. La station de base joue le rôle d'une passerelle, elle est Interface liant le réseau de capteurs avec l'internet. Elle est la seule entité connectée à Internet dans le RCSF. Des solutions propriétaires, développées en ad hoc, pour supporter la communication au sein du réseau de capteurs les noeuds capteurs sont indirectement intégrés à Internet.

I.11 Conclusion

Dans ce chapitre, nous avons abordé le concept de l'Internet des objets et conclu qu'il s'agit d'une extension d'Internet au monde réel des objets qui nous entourent et qu'il consiste en une manière simplifiée de connecter les objets, et la sécurité dans l'IoT plus important pour la confidentialité des objets. L'IoT offre un potentiel inestimable dans le domaine d'intelligence artificielle. Grâce à des capteurs connectés, il est possible de superviser l'ensemble des systèmes qui y sont installés.

Dans le chapitre suivant, nous découvrirons en détail l'Internet des voitures et tout ce qui s'y rapporte.

Chapitre II

Internet des véhicules

II.1 Introduction

La technique Internet des véhicules (IoV) a évolué au fil du temps à partir du réseau de véhicules ad hoc classique (VANET), qui fait référence à un réseau d'entités disparates dans un véhicule à roues.

Avec l'accélération de l'innovation, il n'est pas difficile d'avoir une voiture autonome dans ce monde. Les voitures autonomes et les voitures connectées sont des techniques complémentaires car elles partagent certaines technologies et en complètent d'autres. Les véhicules connectés utilisent la technologie sans fil, l'infrastructure, les smart phones, etc. En cas de besoin, les voitures autonomes utilisent des capteurs et des ordinateurs pour analyser leur environnement et conduire sans intervention humaine.

Nous aborderons plus en détail l'internet des véhicules (véhicules autonomes et connectés) et découvrirons les travaux connexes qui y sont liés dans ce chapitre.

II.1.1 Internet des véhicules(IoV)

L'internet des véhicules (IoV) est un réseau distribué qui prend en charge l'utilisation des données générées par les véhicules connectés et les réseaux de véhicules autorégulés (VANET). Un objectif important de l'IoV est de permettre aux véhicules de communiquer en temps réel avec les conducteurs, les piétons, les autres véhicules, l'infrastructure routière et les systèmes de gestion de flotte[37].

II.1.2 Véhicule connectée

Un véhicule connecté oriental un véhicule disposant pour systèmes pour communication embarqués lequel permettant incontinente communication dépourvu cours sans l'environnement sur certaines éventualités une jonction sans Internet existe[12].

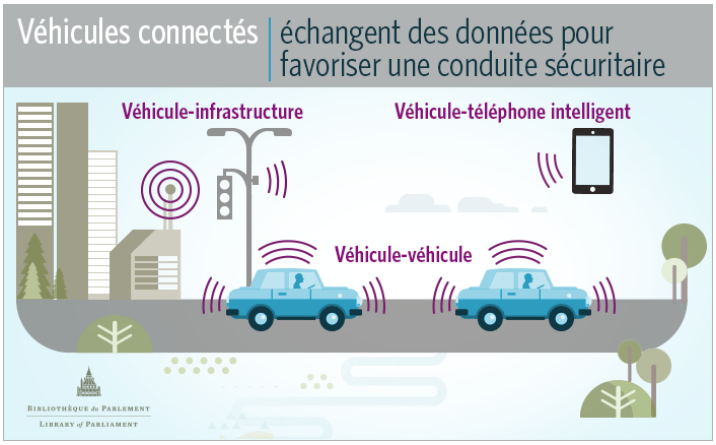

FIGURE II.1 – Véhicule connecté.

II.1.3 Véhicule autonome

Ce que l'on appelle une voiture autonome est un véhicule able de rouler sans d'un être humain. Grâce à de nombreux capteurs et à un logiciel de calcul particulièrement

élaboré, elle est able de se déplacer dans le trafic et de prendre des décisions toute seule, sans l'apport d'un conducteur [22].

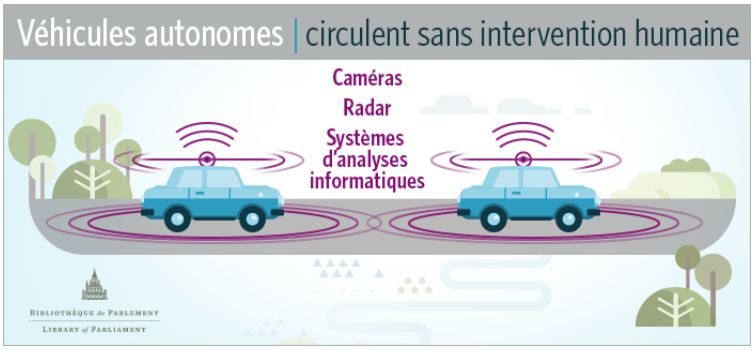

FIGURE II.2 – Véhicule autonome.

II.2 Caractéristiques de l'IoV

Il existe plusieurs caractéristiques spécifiques de l'IoV [21]sont :

- Évolutivité élevée : un réseau évolutif et à grande échelle est nécessaire pour créer un réseau véhiculaire.
- topologie dynamique : de nombreux composants différents et hétérogènes de l'IoV interagissent et se déplacent rapidement les uns avec les autres, ce qui entraîne une modification de la structure du réseau.
- densité de réseau non uniforme : la densité du réseau Internet des véhicules varie en fonction de différentes conditions telles que les routes, la taille des villes, la situation géographique...
- Complexe de communication :la densité du réseau IoV varie d'un scénario à l'autre. l'IoV est un réseau de communication complexe, il doit donc être fiable.

- Énergie et capacité de traitement :Un réseau d'IoV est un groupe de véhicules qui ont suffisamment de puissance et d'espace pour inclure la puissance de traitement afin qu'ils ne manquent pas de puissance de traitement ou de capacité de mémoire.
- Communication géographique :le réseau véhiculaire utilise la géo-connexion pour transmettre des paquets d'un nœud à un autre.

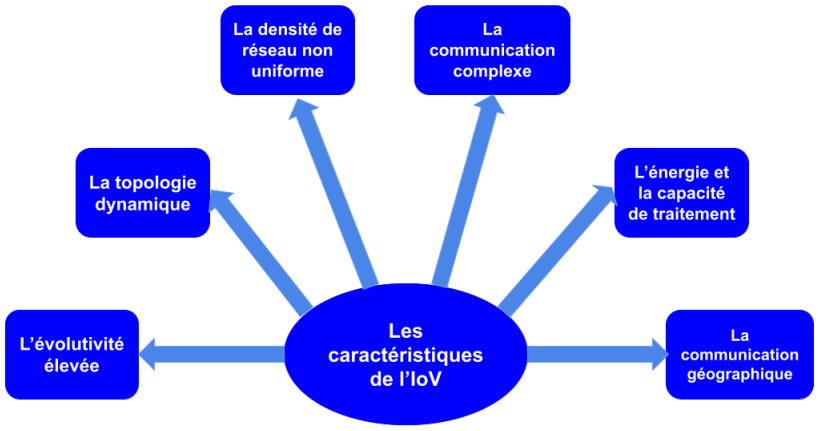

FIGURE II.3 – Les caractéristiques de l'IoV[21].

II.3 Niveaux d'automatisation

Le développement technologique dans le secteur automobile a conduit à la présence de six niveaux allant de la conduite entièrement humaine à la conduite automatisée [11], et les véhicules sont classés selon les niveaux suivants :

II.3.1 Nivaux 0 : il n'y a pas d'automatisation

Ce niveau est considéré comme le niveau le plus bas car le véhicule est équipé uniquement de systèmes d'avertissement. Ainsi, il n'y a pas d'autonomie, ce qui signifie que le conducteur est seul responsable de la conduite de la voiture.

II.3.2 Nivaux 1 : assistance à la conduite

A ce niveau, la voiture est capable d'interagir dynamiquement avec son environnement, mais le conducteur doit contrôler en permanence la conduite de la voiture.

II.3.3 Nivaux 2 : autonomie partielle

A ce niveau, la voiture fournit une assistance à la direction, à l'accélération, à la décélération et au freinage, tandis que le conducteur reste vigilant et surveille la conduite de la voiture pour pouvoir la contrôler.

II.3.4 Nivaux 3 : autonomie conditionnelle ou semi-autonome

A ce niveau, le conducteur peut tout faire pendant que la voiture est en mouvement, comme feuilleter un journal... tout en restant vigilant à tout moment pour reprendre le contrôle de la voiture dans les situations nécessaires.

II.3.5 Nivaux 4 : autonomie totale

Le système de la voiture ne nécessite pas d'intervention du conducteur à ce niveau (il peut dormir sur le volant). La voiture est capable de prendre les bonnes décisions pour éviter les situations dangereuses auxquelles elle est confrontée sur son chemin et donc la direction, la pédale, l'embrayage et le frein sont expérimentaux.

II.3.6 Nivaux 5 : Autonomie absolue

La voiture est totalement indépendante (idéal) et roule sur diverses routes partout et par tous les temps sans la présence du conducteur.

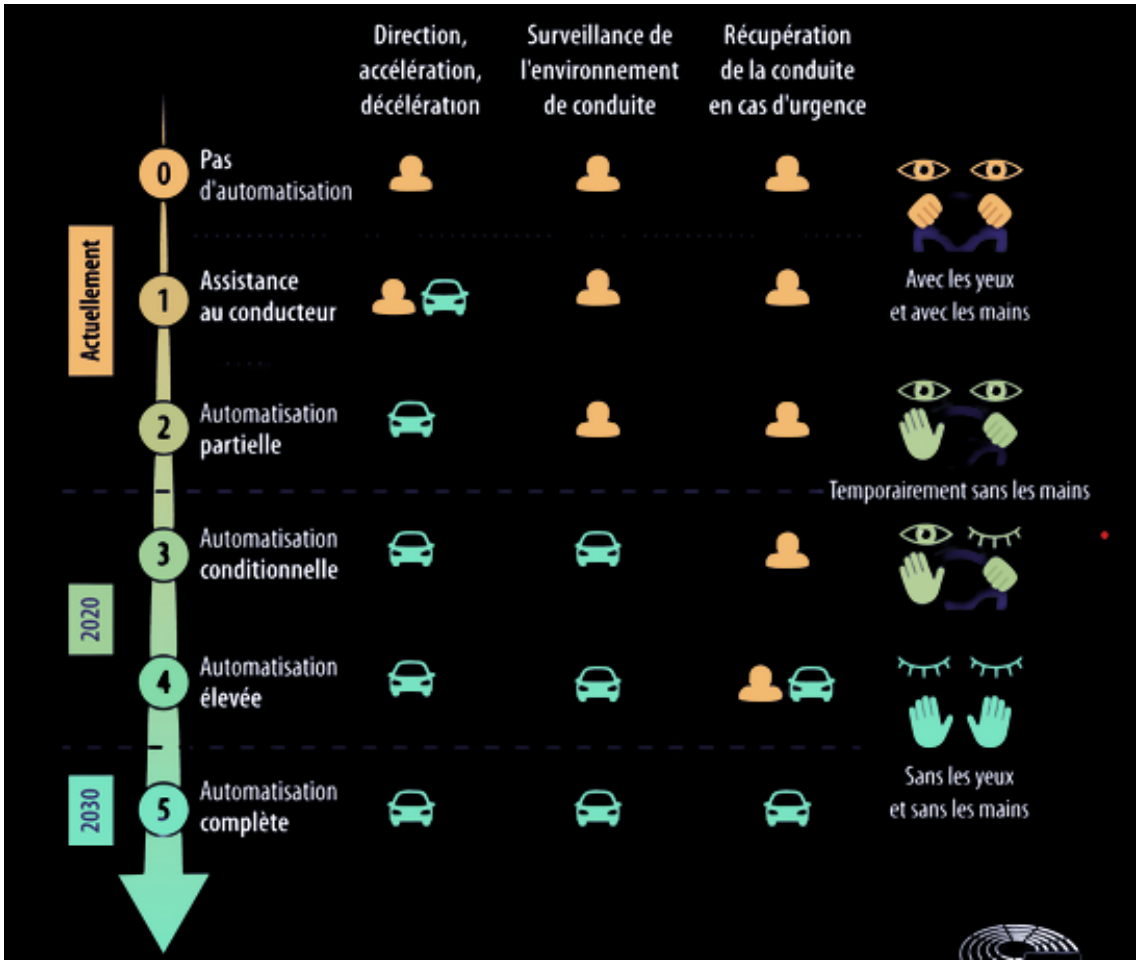

FIGURE II.4 – Les niveaux d'automatisation[19].

Comparaison entre les nivaux d'automatisation :

Dans le tableau suivant une petite comparaison entre les différents niveaux d'automatisation des véhicules par rapport l'exécution de la direction et de l'accélération

et décélération, surveillance de l'environnement de conduite, la performance de repli de la tâche de conduite dynamique et la capacité du système, [14] comme suivant :

Niveau	Exécution de la direction et de l'accélération et décélération	Surveillance de l'environnement de conduite	Performances de repli de la tâche de conduite dynamique	Capacité du système (modes de conduite)
0	conducteur	conducteur	conducteur	n/a
1	conducteur et le système	conducteur	conducteur	certains modes de conduite
2	système	conducteur	conducteur	certains modes de conduite
3	système	système	conducteur	certains modes de conduite
4	système	système	système	certains modes de conduite
5	système	système	système	tousles modes de conduite

II.4 Fonctionnement d'une voiture autonome

Les voitures autonomes dépendent de leurs propres caméras, radars et capteurs pour fonctionner. Il produit des cartes haute résolution afin que la voiture soit consciente de tous les éléments de son environnement pendant la conduite[13]. Les

décisions de conduite dépendent des informations envoyées au véhicule autonome via des radars, des capteurs et des caméras.

- **Des lidars :** une méthode de télédétection, qui effectue une étude tridimensionnelle de l'environnement qui l'entoure. Habituellement, il y en a deux à l'avant et à l'arrière de la voiture.
- **Une caméra trifrocale :** équipé d'une longue portée, d'une portée moyenne et d'une porté court, il se trouve également au niveau du tableau de bord.
- **Un radar frontal :** il mesure la distance entre le véhicule et les véhicules dans le même environnement, caractérisé par une longue portée.
- **Des capteurs ultrasons :** en plus de sa capacité à détecter les obstacles, il est de courte durée.
- **Des radars d'angle :** il est placé aux quatre coins du véhicule pour détecter les mouvements autour de celui-ci.
- **Des caméras 180° :** ces caméras sont placées sur les plaques d'immatriculation (avant et arrière), ainsi que sous les rétroviseurs.
- **Odomètre :** il calcule la distance pour vérifier suffisant espace du garer voiture ou non.
- **Antenne GPS :** pour suivre les coordonnées d'un véhicule.

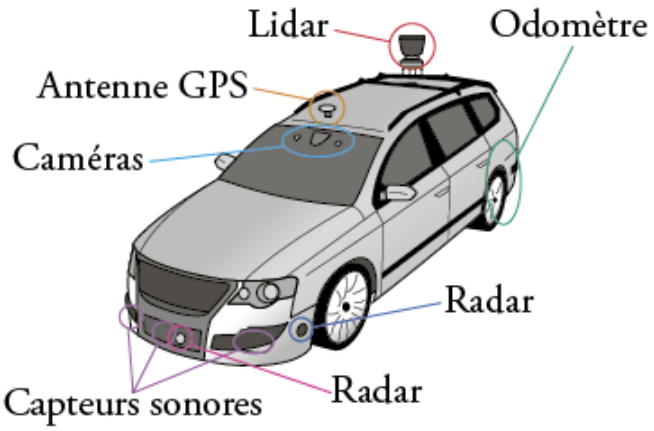

FIGURE II.5 – Les appareils nécessaire par voiture autonome[8].

II.5 Développement des véhicules connectés autonomes

En raison de l'expansion intense du marché automobile, de nouvelles stratégies de développement ont émergé, ce qui a conduit à l'émergence de voitures connectées et autonomes pour le transport. Internet et d'autres entreprises jouent un rôle important et essentiel dans le développement des véhicules connectés et autonomes. Ainsi, le premier véhicule connecté de Chine a été présenté à l'auto china en 2016, le système d'interaction vocale intellige a été annoncé lors de la conférence créative des développeurs Baidu Al 2017[7], la plate-forme de conduite intelligente sur laquelle ils se concentrent et les centres de données etc.

II.6 Défis avec les voitures autonomes

Les défis posés par les voitures autonomes varient d'un environnement à l'autre[24], notamment :

- **Lidar et Radar :** Considérés parmi ces défis, il est coûteux et encore en développement d'atteindre un équilibre entre portée et précision.
- **Conditions météorologiques :** lorsque la neige tombe, les séparateurs de voie disparaissent, il est donc difficile pour les caméras et les capteurs de suivre les marquages au sol. à analyser uniquement.

II.7 Trois éléments de la route de 5ème génération

Il existe trois éléments principaux dans la route de cinquième génération [23], comme suit :

II.7.1 Route automatisée

- Ce défi est représenté de manière automatisée pour tirer parti des lacunes des techniques qui gèrent le trafic.
- la première étape consiste à mettre en œuvre des stratégies qui gèrent le nouveau trafic, qui à son tour dépend sur l'augmentation de la connectivité et de l'automatisation des véhicules, pour améliorer le système routier, ses performances et l'assurance voyage, il peut améliorer l'utilisation des infrastructures à partir des convois de véhicules ou des systèmes de transport intelligents. Les mises à jour actuelles sont la combinaison de mesures collectives de gestion du trafic et la coexistence avec des véhicules motorisés et traditionnels et des services nomades individuels.

- La prochaine étape consiste à déployer des systèmes de suivi d'itinéraire qui intègrent des données de capteurs embarqués dans des infrastructures embarquées dans le but de pouvoir exploiter les systèmes de capteurs existants tout en intégrant de nouvelles générations pour obtenir des données de trafic plus précises et moins coûteuses [39].

II.7.2 Route adaptable

— Ce défi consiste à fournir un système d'itinéraires rentable et flexible (itinéraire adaptatif), capable de s'adapter à toutes les exigences de déplacement futures.

— Les principales solutions qui permettent à la route d'être plus extensible en :

- Développer des façons nouvelles et différentes de construire des routes, ces techniques offrent des solutions alternatives et rapides au cas où certains tronçons seraient endommagés.
- Développer des formations offrant une solution intermédiaire entre la résistance au roulement, la portance de l'eau, l'adhérence et le bruit de roulement.
- Service sur la route pour pouvoir intégrer des systèmes de ramassage et des extensions de route renouvelables si nécessaire, pour fournir des équipements tels que l'éclairage général, les véhicules de recharge et le conseil.

II.7.3 Route résiliente au changement climatique

La route devrait être plus flexible face aux effets du changement climatique, et de nombreuses mesures sont à prendre, notamment :

- Déterminer et définir les besoins des gestionnaires en matière de climat.
- coordination des données climatiques.
- Répartir les effets sociaux et économiques en cas de turbulences.
- Connaître (identifier) les faiblesses du réseau.
- Identifier les techniques qui contribuent aux objectifs d'atténuation.
- Déterminer les futurs niveaux de services du réseau.

II.8 Sécurité des véhicules autonomes connectés

Il a récemment commencé à remarquer des attaques sophistiquées apparaissant sur les systèmes des véhicules, puisque les véhicules autonomes fourniront une connectivité plus élevée à l'avenir, ils devront se renseigner sur leur environnement que les pirates peuvent manipuler pour déterminer leur chemin.Les véhicules peuvent servir de cible s'ils sont d'une grande importance d'un point de vue informatique ou financière, et en raison de leur nature critique et de leur grande valeur, ils deviendront la cible d'actes illégaux[26] [38].

II.9 Problèmes de sécurité des architectures actuelles

Dans cette section on va voir deux parties [28](attaques usuelles et preuve de concepts) comme suivant :

II.9.1 Attaques usuelles

De nos jours, les attaques sont généralisées sur les systèmes des véhicules avec deux motivations principales :

II.9.1.1 Activation d'options

Pour activer les options non payantes, ses composants doivent être physiquement présents dans la voiture, l'attaque peut se faire en retéléchargeant les fichiers du programme de gestion du moteur pour augmenter sa puissance. Des micrologiciels tiers peuvent également être trouvés sur Internet.

II.9.1.2 Vol de véhicules

La voiture est volée de multiples façons, par exemple en activant des attaques sur la serrure de porte ou sur le système d'exploitation... La voiture peut également être volée lors du démarrage ou de l'ouverture de la porte de manière sans fil, car ces systèmes sans fil sont sensibles en termes d'attaques.

II.9.2 Preuve de concepts

Les attaques mettent en évidence des faiblesses importantes à différents niveaux de la carrosserie : capteurs, communications internes et externes. De nombreuses attaques sont effectuées en se connectant au port patch OBD du système embarqué.

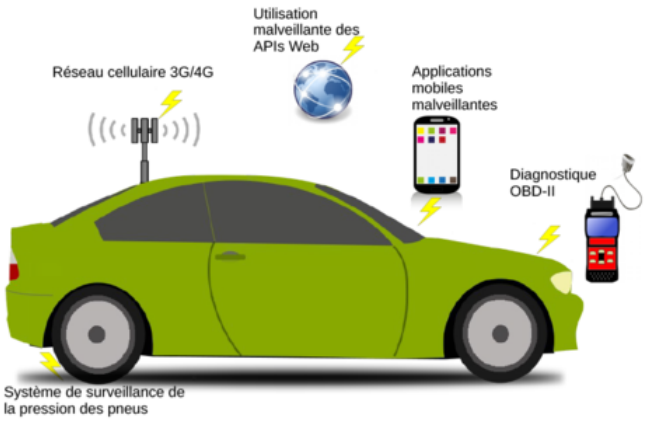

FIGURE II.6 – Attaques démontrées sur des véhicules actuels[3].

II.10 Travaux connexes

Dans cette section, nous parlerons brièvement sur des travaux connexes à notre projet et les comparerons comme suit :

dans [35], les auteurs proposent un modèle de voitures autonomes est construit sur la vision monoculaire. Cette voiture a été entraînée sur des données d'images de course combinées à un angle de braquage synchrone généré manuellement. Il a été conduit sur une route elliptique et une route en forme de 8 avec une piste bordée de feux de circulation. Il a une vitesse d'environ 5 à 6 km/h dans diverses conditions de conduite.

Ce papier [2] présente un prototype d'AGV qui contrôle les commandes de direction et d'accélérateur qui a été conçu à l'aide de modèles de réseaux de neurones, ce modèle peut prédire mais pas de manière significative car il peut analyser l'angle de braquage.

Les auteurs dans [15], présentent un prototype de voiture autonome pouvant circuler sur différents trajets (routes courbes, droites et routes droites suivies de virages). Son logiciel combine la technique de détection de voie (et la détermination des marqueurs) et la surveillance des véhicules pour améliorer la sécurité du conducteur.

II.11 Conclusion

Le véhicule autonome constitue un défi Méthodique et sociétal sans précédent dans le domaine de la mobilité. C'est un objet qui dans sa forme là aussi aboutie est d'une très grande complexité et qui nécessite à la fois d'importants efforts en recherche et développement et d'importants soutiens financiers. Si l'avenir semble à peu près tracé pour les niveaux d'autonomie 1 à 3, il subsiste reprise quelques incertitudes sur le niveau 4 et de réels doutes quant à la faisabilité du niveau 5 pour un coût abordable. De nombreux défis procédures et non stratégies restent à relever. Ils omen sur la sûreté de fonctionnement, la baisse des coûts de génération, les évolutions nécessaires de l'infrastructure et des métiers des exploitants routiers, la prévention des cyberattaques, la définition des administrations utiles et socialement acceptables pour les usagers, etc.

Chapitre III

Conception du système

III.1 Introduction

Dans ce chapitre nous allons présenter la conception de notre projet, nous commençons par les architecture générale et détaillée. ensuite nous mettons en évidence le côté conceptuel de notre application qui constitue une étape fondamentale qui précède l'implémentation du système, permet de détailler les différents scénarios à implémenter dans le chapitre suivant, nous expliquerons chaque partie de notre système.

III.2 Architecture générale

Dans cette section, nous allons présenter l'architecture générale de notre système , qui est basée sur deux parties principale :

la premier partie c'est la voiture, la deuxième partie, la station de calcul et de contrôle de la voiture.

- La voiture perçoit l'environnement a travers les capteurs (images et télémétries) et la caméra. Ensuite, la voiture envoi ses informations à la station de calcul et de contrôle.
- Station de calcul et de contrôle traite et analyse les données reçues par le véhicule et les renvoie sous forme des commandes en temps réel.

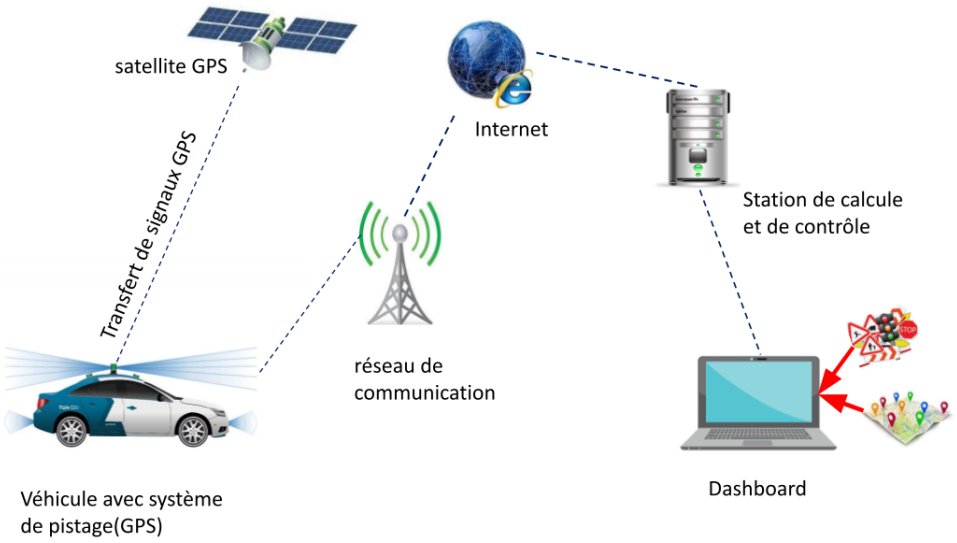

FIGURE III.1 – Architecture générale.

La figure suivante illustre comment le faire fonctionner de manière facile et simple :

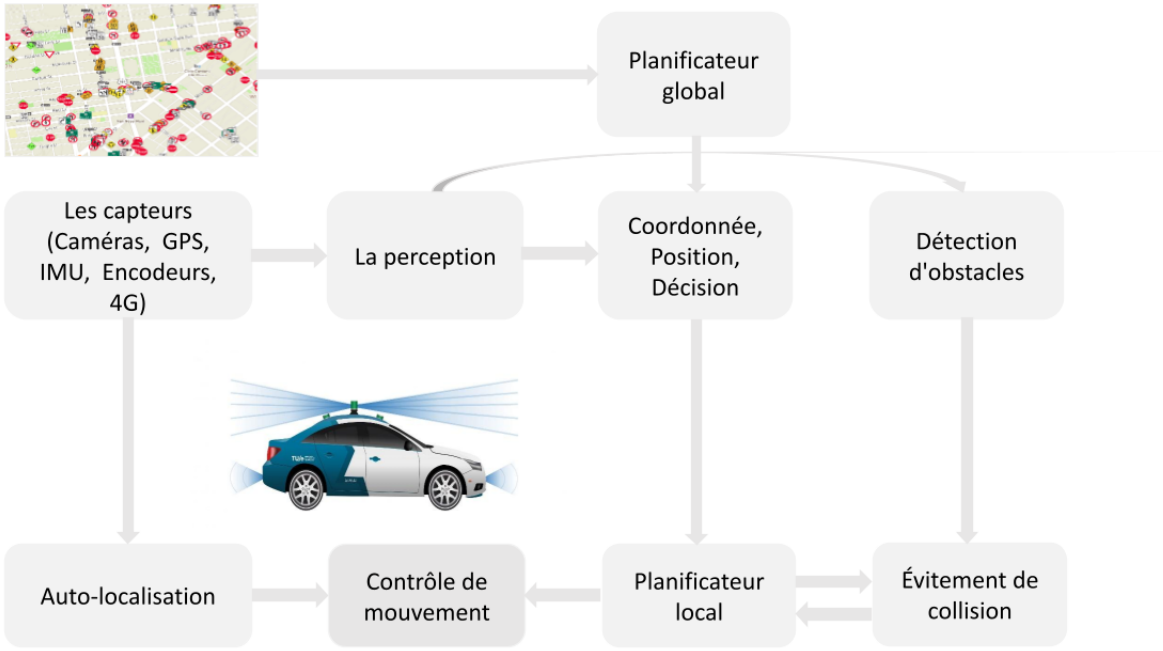

FIGURE III.2 – Fonctionnement de système des véhicules autonome.

III.3 Architecture détaillée

Dans cette section,nous détaillons les trois parties de notre architecture (le véhicule, la station de calcul et de contrôle,le dashboard) :

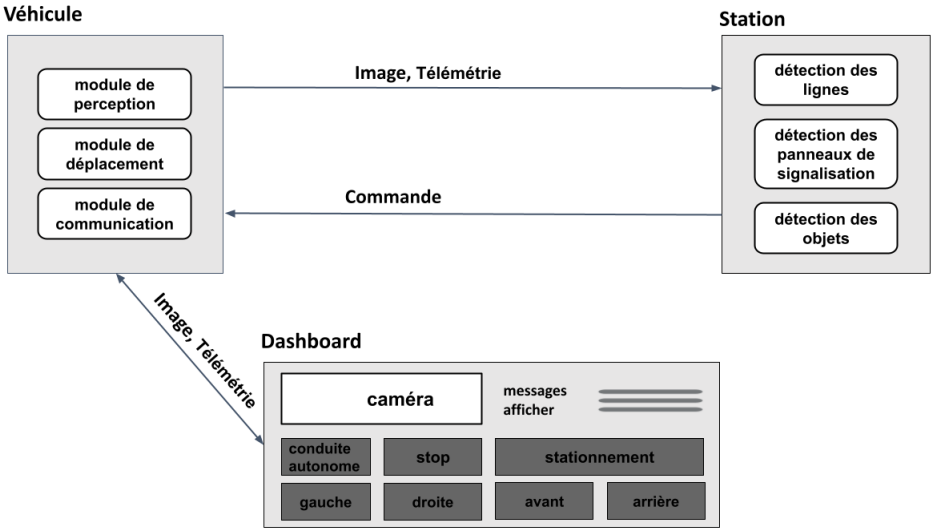

FIGURE III.3 – Architecture détaillée de notre systéme .

III.3.1 Véhicule

le véhicule est la première partie essentielle, il se compose de trois modules principaux :

III.3.1.1 Module de perception

Les voitures autonomes doivent être équipées d'un ensemble de capteurs pour devenir autonomes afin de mieux percevoir (détecter et d'interpréter) leur environnement (la route, les autres véhicules, les piétons et les signaux du code de la route) de manière fiable :

Notre voiture est dotée de deux types des capteurs :

1. **Une caméra :** la voiture équipée par une caméra. elle permet de surveiller le champ de vision. Elle offre une vue qui permet d'apprécier les feux de

circulation et d'apprécier les objets et les piétons.

Les performances des caméras ne changent pas avec le temps, ce qui en fait un élément indispensable du travail des véhicules autonomes.

2. **Capteur ultrasonique :** plus de la caméra, nous avons également équipé le véhicule de quatre capteurs ultrasons. Ce type de capteur utilise l'énergie électrique pour envoyer et recevoir de l'énergie mécanique sous forme d'ondes sonores qui peuvent être utilisées pour mesurer la distance afin que le véhicule puisse se garer ou s'arrêter dans des situations critiques pour éviter les collisions lors de la détection des objets devant elle.

III.3.1.2 Module de déplacement

Une voiture autonome peut aller partout où va une voiture conventionnelle et peut faire tout ce qu'un conducteur humain fait. Mais il est très nécessaire de le former correctement et donc l'une des étapes qui sont suivies lors de la formation d'une voiture autonome est de savoir comment changer le sens de la marche tout en naviguant sur la voie. Par conséquent, ce module aide le véhicule à se déplacer d'un endroit à un autre , en avançant dans une direction directe ou en reculant, comme peut le véhicule peut également changer le sens de sa marche vers la droite ou vers la gauche selon la direction de la route qu'il emprunte lors de son déplacement.

III.3.1.3 Module de communication

Le véhicule communique avec la station de calcul et de contrôle, le dashboard par l'intermédiaire du module de communication qui est un réseaux sans fil et un programme exécute dans la voiture que permet d'échange des commandes soient échangées entre eux [41].

III.3.2 Station de traitement

La station de traitement, traite les images envoyées et captées par la voiture sont traitées par les trois algorithmes ci-dessous afin de les renvoyer au véhicule sous forme de commandes.

III.3.2.1 Détection des lignes de la route

Pour que le véhicule autonome détecte le chemin, il recourt à l'aide de la vidéo prise à partir de la caméra placée dessus. Donc nous allons concevoir un algorithme qui permettra de détecter les voies sur la route et d'estimer la courbure de la route pour prédire le virage du véhicule.

L'algorithme suit un pipeline simple avec plusieurs étapes, comme indiqué dans le diagramme d'activité suivante :

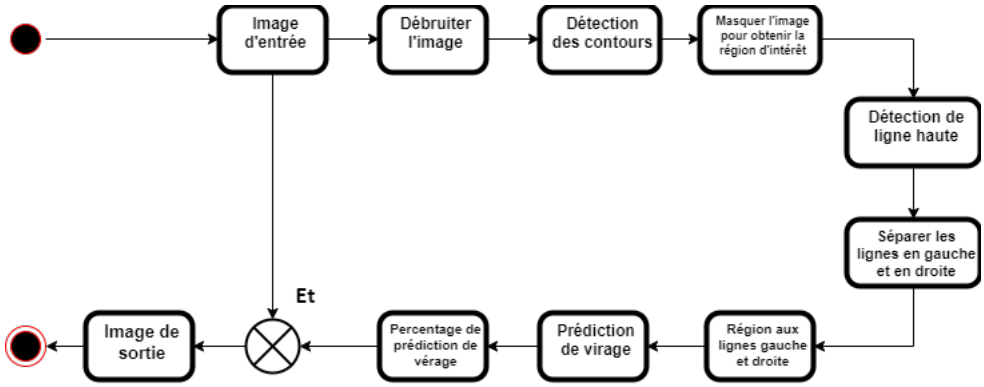

FIGURE III.4 – Diagramme d'activité.

L'algorithme a été créé pour montrer le fonctionnement d'un système de détection de la route, sur les véhicules équipés d'une caméra frontale. Ce système est un élément essentiel dans les systèmes avancés d'aide à la conduite (ADAS) utilisés dans les véhicules autonomes/semi-autonomes. Cette fonction est responsable de la détection

des voies, de la mesure du rayon de la courbe (resserrement d'une courbe) et surveille le décalage par rapport au centre. Avec ces informations, le système améliore considérablement la sécurité en s'assurant que le véhicule est centré à l'intérieur des lignes de voies, ainsi qu'en ajoutant du confort s'il est également configuré pour contrôler le volant pour prendre des virages doux sur les autoroutes sans aucune intervention du conducteur.

L'algorithme applicable contient plusieurs étapes et phases successives qui sont décrites ci-dessous comme suivant :

Algorithme 1 Algorithme de détection de la route et de prédiction de virage

$image \leftarrow readVideo()$;
while $True$ **do**
$frame \leftarrow readImage()$;
$birdView \leftarrow perspectiveWarp(frame)$;
$thresh \leftarrow processImage(birdView)$;
$hist \leftarrow plotHistogram(thresh;)$
$ploty, left\_fit, right\_fit, left\_fitx, right\_fitx \leftarrow slideWindowSearch(thresh, hist)$;
$draw\_info \leftarrow generalSearch(thresh, left\_fit, right\_fit)$;
$curveRad, curveDir \leftarrow measureLaneCurvature(ploty, left\_fitx, right\_fitx)$;
$meanPts, result \leftarrow drawLaneLines(frame, thresh, minverse, draw\_info)$;
$deviation, directionDev \leftarrow offCenter(meanPts, frame)$;
$finalImg \leftarrow addText(result, curveRad, curveDir, deviation, directionDev)$;
display video ;
end while

Fonctionnement d'un algorithme :

- La première fonction est : readvideo() pour accéder aux vidéos qui se extraire dans la caméra en temps réel.
- Processus Image() : cette fonction exécute certaines techniques de traitement pour isoler les lignes de voix blanches et les préparer à être analysées plus en détail par les fonctions à venir. Fondamentalement, il applique le filtrage des couleurs HLS pour filtrer les blancs dans le cadre, puis le convertit en niveaux de gris qui sont ensuite appliqués un seuillage pour se débarrasser des détections inutiles autres que les voies, deviennent floues et enfin les bords sont extraits avec la fonction cv2.Canny().
- perspectiveWarp() : maintenant que nous avons l'image que nous voulons, une déformation de perspective est appliquée. 4 points sont placés sur le cadre de sorte qu'ils n'entourent que la zone où les voies sont présentes , puis le mappent sur une autre matrice pour créer un regard d'oiseau sur les voies. Cela nous permettra de travailler avec une image beaucoup plus fine et d'aider à détecter les courbures des voies.

FIGURE III.5 – Différentes phases de la trame en cours de traitement.

- plotHistogram() : tracer un histogramme pour la moitié inférieure de l'image est une partie essentielle pour obtenir les informations sur l'endroit exact où commencent les voies de gauche et de droite.

- slideWindowSearch() : une approche de fenêtre glissante est utilisée pour détecter les voies et leur courbure. Il utilise les informations de la fonction d'histogramme précédent et place une boîte avec une voie au centre. Place ensuite une autre boite sur le dessus en fonction des positions des pixels blancs de boîte précédente et se place en conséquence j'us qu'en haute du cadre. De cette façon, nous avant les informations pour faire des calculs. Ensuite, un ajustement polynomial du second degré est effectué pour avoir un ajustement de courbe dans l'espace des pixels.
- generalSearch() : après avoir exécuté la fonction slideWindowSearch(), cette fonction generalsearch() est maintenant capable de remplir une zone autour de ces voies détectées, applique à nouveau le polyfit du deuxième degré pour ensuite tracer une ligne jaune qui chevauche les voies assez précisément. Cette ligne sera utilisée pour mesurer le rayon de courbure qui est essentiel pour prédire les angles de braquage.

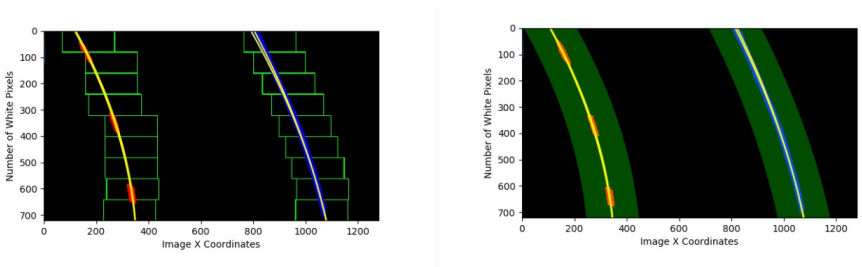

FIGURE III.6 – Recherche générale visualisée.

- measureLaneCuvarture : avec les informations fournies par les deux fonctions précédentes, la fonction np.polyfit() est à nouveau utilisée mais avec les valeurs multipliées par les variables xmperpix et ymperpix pour les convertir de l'espace pixel à l'espace mètre.

- drawLaneLines() : sont appliquées certaines les méthodes pour visualiser les voies détectées et d'autres informations à afficher pour l'image finale. Cette fonction particulière prend les voies détectées et remplit la zone à l'intérieur avec une couleur verte. Il visualise également le centre de la voie en prenant la moyenne des listes left_fitx et right_fitx et en les stockant dans la variable ptsmean, cette variable est utilisée pour calculer le décalage du véhicule de chaque côté ou de celui-ci est centrée dans la voie.
- offCenter() : utilise la variable ptsmean pour calculer la valeur de décalage et l'afficher dans l'espace mètre.
- addText() : enfin en ajoutant du texte sur l'image finale pour connaître la direction de la route et le pourcentage de prédiction de avérage.

maintenant on peut résumer l'algorithme précédent dans la figure suivante qui représente la technique utilisée dans la détection des lignes d'une route.

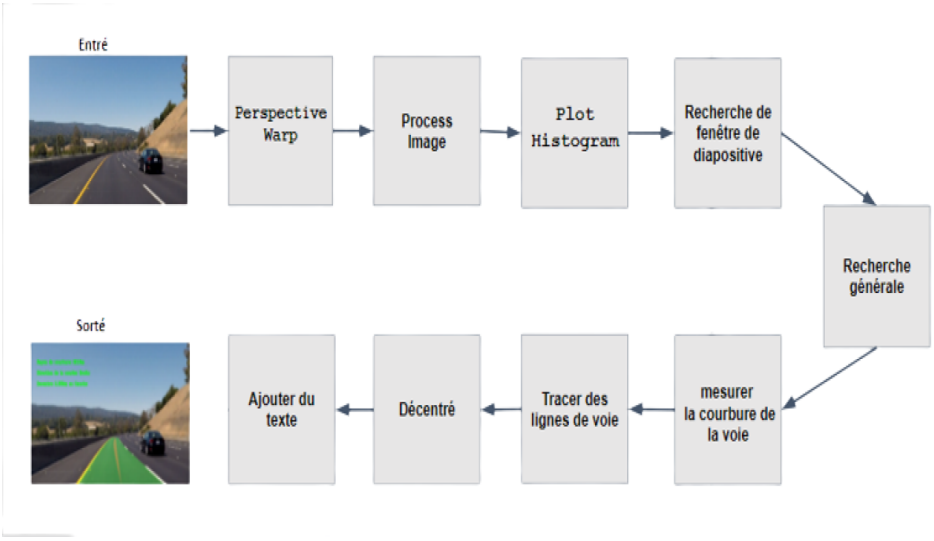

FIGURE III.7 – Technique de détection de voie pour les voitures autonomes.

III.3.2.2 Détection des objets

Pendant que le véhicule autonome est sur la route, il peut y avoir des obstacles devant lui, il doit donc les détecter, nous avons donc travaillait sur l'algorithme de détection d'obstacles par deep learning.

La figure suivante explique brève menti comment détecter un objet par deep learning (algorithme DNN).

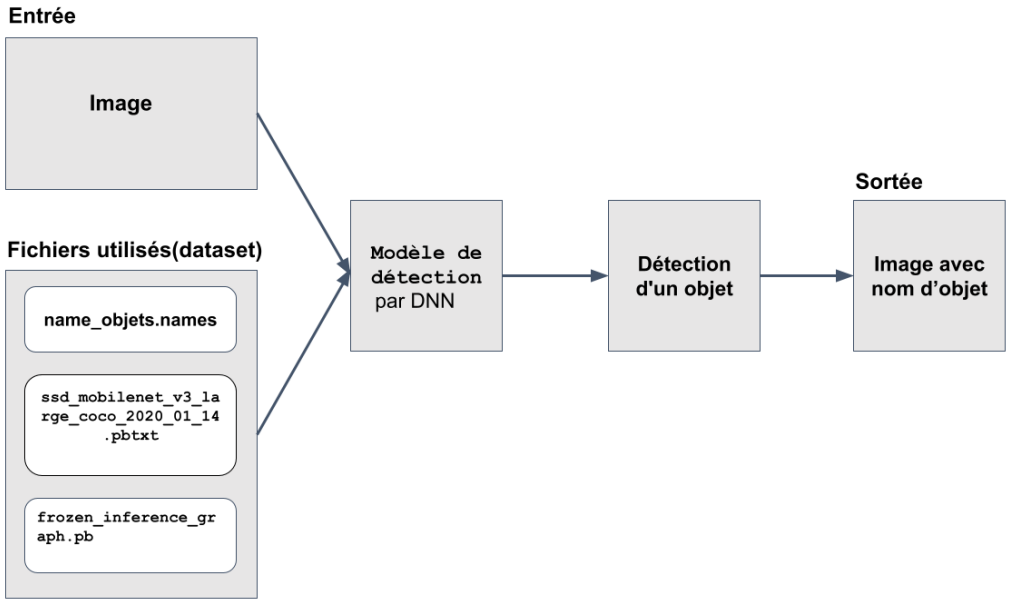

FIGURE III.8 – Détection d'obstacle par deep learning.

Maintenant on va voir les étapes utilisées dans un algorithme suivant :

Algorithme 2 Algorithme de détection des objets

T : tableau taille 3 entier ;
ovrire fichier name_objets.names ;
$conf\_path \leftarrow ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt$;
$weight\_path \leftarrow frozen\_inference\_graph.pb$;
$model \leftarrow detectionModel(conf\_path, weight\_path)$;
while $True$ **do**
  $image \leftarrow readImage()$;
  $T \leftarrow model.detect(image)$;
  if taille(idis) $\neq 0$ **then**
    déssine rectangle sur l'objet ;
    création d'un nom de l'objet affécté par le fichier name_objets.names ;
end while

Les images sont traitées à l'aide d'openCV, donc dans cette partie, je vais montrer comment détecter des objets en temps réel.

Fonctionnement d'un algorithme :

On utilisant opencv pour la détection d'objets est une méthode qui équilibre vitesse et précision en temps réel basée sur Deep Learning avec OpenCv, donc nous devrons à l'accéder à la caméra de du véhicule et appliquer la détection d'objets à chaque image.

La première étape de cet algorithme consiste à lire la vidéo à travers la caméra du véhicule en temps réel et à la redimensionner car nous aurons besoin de sa longueur et de sa largeur. Transformez ensuite la trame à l'aide de dnn, puis définissez un point considéré comme une entrée du réseau de neurones, puis alimentez l'entrée via

le réseau qui donne la découverte. Une fois les obstacles détectés, il vérifie les valeurs de confiance pour déterminer si le cadre doit être dessiné, puis ajoute le nom de l'objet.

III.3.2.3 Détection des pannaux signalisation

Afin d'assurer la sécurité routière du véhicule autonome, celui-ci doit appliquer le code de la route, nous avons donc travaillait à rendre le véhicule capable de détecter et de connaître les pannaux de signalisation. Par conséquent, nous avons créé un algorithme dont le but principal est de découvrir et de connaître les pannaux de signalisation.

L'algorithme que nous avons créé analyse le panneau de signalisation afin qu'il puisse le distinguer des autres panneaux de signalisation à l'aide d'openCv, et à partir de celui-ci, donc l'algorithme utilisé pour découvrir les panneaux de signalisation comme suit :

Algorithme 3 Algorithme de détection des pannaux signalisation.

```
success, frame ← readCameraCapture();
while success and not clicked do
    waitKey(1);
    success, frame ← readCameraCapture();
    gray ← grayColor(frame);
    img ← medianBlur(gray);
    circles ← HoughCircles(img);
    if not circles is None then
        circles ← unit16(circles); // Entier non signé (0 à 255)
        max_r = 0, max_i = 0;
        for i in lenth(circles[ :, :,2][0]) do
            // [debut,fin,pas=2]
            if circles[:, :, 2][0][i] > 50 and circles[:, :, 2][0][i] > max_r then
                max_i ← i;
                max_r ← circles[:, :, 2][0][i];
            end if
        end for
        x, y, r ← circles[:, :, :][0][max_i];
        if y > r and x > r then
            square ← frame[y − r : y + r, x − r : x + r];
            dominant_color ← get_dominant_color(square, 2);
            Algorithme affécté pannaux ;
        end if
    end if
end while
```

Algorithme 4 Algorithme affécté pannaux

```
if dominant_color[2] > 100; then
    afficher(stop);
else if dominant_color[0] > 80 then
    z_0 ← calcule_sqr();
    z_0_c ← get_dominant_color(z_0, 1);
    z_1 ← calcule_sqr();
    z_1_c ← get_dominant_color(z_1, 1);
    z_2 ← calcule_sqr();
    z_2_c ← get_dominant_color(z_2, 1);
    if z_1_c[2] < 60 then
        if som(z_0_c) > som(z_2_c) then
            afficher(gauche);
        else
            afficher(droite);
        end if
    else
        if som(z_1_c) > som(z_0_c) and som(z_1_c) > som(z_2_c) then
            afficher(avant);
        else if som(z_0_c) > som(z_2_c) then
            afficher(avant et gauche);
        else
            afficher(avant et droite);
        end if
    end if
end if
```

Fonctionnement d'un algorithme 3 :

- readCameraCapture() : cette fonction pour lire l'image en temps réel, qu'il existe dans la bibliothèque opencv.
- grayColor() :pour transformer l'image à la color gry.
- medianBlur() : l'objectif de cette fonction est lissé l'image, elle transforme chaque canal en une image multicanale.
- HaughCircles() : rechercher des cercles dans l'image en grayScal à l'aide de la transformation Haugh pour la trouver à l'aide de modifier la transformation Haugh.
- Ensuite, il est vérifié si le non-cercle est null, si la condition est satisfaite, le cercle est converti en entier non signé entre 0 et 255. Après tout ça, on recherche si la valeur d'un des éléments de la matrice du circuit est supérieure à 50 et supérieure à la valeur de max_r. On remplace max_i par une valeur d'i et max_r par la nouvelle valeur qui satisfait les deux conditions Puis on appelle la fonction get_dominant_color() si la condition avant elle est remplie, pour venir le rôle de l'algorithme 4.
- get_dominant_color() : cette fonction pour retourner la colour dominante d'une image

Fonctionnement d'un algorithme 4 :

- La première étape consiste à vérifier si la valeur de dominante_color[2] est supérieure à 100, si la condition est remplie, le signe est le panneau stop, et si sa valeur pour le premier indice(0) est supérieure à 80, il calcule les trois valeurs de z par la fonction calcule_sqr() suivi de deux autres conditions par

lesquelles chaque signe est affiché à gauche et à droite. la dernière condition qui distingue chacun des signes avant et gauche, avant et droite.

- calcule_sqr() : cette fonction retournée la valeur de centre signe pour afficher sur la forme square.

III.3.3 Dashboard

Son objectif est de surveiller et de contrôler le système. Il se compose d'un écran pour voir l'environnement que voit l'objectif de la caméra placé dans le véhicule, en plus de plusieurs boutons au cas où nous voudrions contrôler le véhicule.

III.4 Conclusion

Dans ce chapitre, nous avons présenté les architectures générales et détaillées du système de notre projet et expliqué chacune d'entre elles séparément. En plus nous avons également expliqué en détail les algorithmes utilisés à chaque étape de l'architecture détaillée du système. Dans le chapitre suivant, nous expliquerons exactement ce que nous avons fait à chaque étape.

Chapitre IV

Implémentation du système

IV.1 Introduction

Dans le chapitre précédent, nous avons fourni une description de la conception du système et expliqué chaque étape séparément.Dans ce chapitre, nous définirons les programmes et les outils, les langages de programmation et les bibliothèques utilisés dans la mise en œuvre du projet. puis a représenté le matériel utilisé après on va présenter quelques codes. Enfin nous présenter quelques résultats.

IV.2 Environnement de développement

Dans cette section on parlera sur deux côtés : 1re côté est les différents frameworks, libraries et les langages de programmation et la 2ème coté : les outils de développement ont utilisant dans notre projet.

IV.2.1 Langages de programmation et Framework

Dans cette section nous parlerons sur les softwares utilisant dans notre projet comme les langages de programmation, les bibliothèques.

IV.2.1.1 Python

Python est le langage de programmation open source la plus populaire parmi les programmeurs. C'est un langage de programmation orienté objet (POO) polyvalent, facile à apprendre. Il est largement utilisé dans de nombreux domaines tels que le développement de pages Web, la création de logiciels indépendants, etc.

FIGURE IV.1 – Logo python

IV.2.1.2 OpenCV

Il s'agit d'une bibliothèque gratuite qui se concentre principalement sur le traitement d'images en temps réel. C'est aussi une bibliothèque d'association logicielle dont l'objectif principal est de développer la vision par ordinateur, développée par Intel Corporation, puis soutenue respectivement par Willow Garge pour la robotique et Itseez, et enfin par Intel à nouveau après avoir acquis Itseez depuis 2016.

FIGURE IV.2 – Logo opencv

IV.2.1.3 Numpy

Il s'agit d'une bibliothèque logicielle open source utilisé pour gérer de grands tableaux et des champs à plusieurs niveaux. Il fournit une bibliothèque de fonctions mathématiques de haut niveau pour travailler à la fois dans les champs et les tableaux. Cette bibliothèque est également spécialisée dans le calcul scientifique en Python et contient une variété d'outils et de techniques qui peuvent être utilisées pour résoudre des problèmes mathématiques en ingénierie et en science.

FIGURE IV.3 – Logo numpy

IV.2.1.4 Matplotlib

Il s'agit d'une bibliothèque de schémas Python qui fournit une API orientée objet, destinée à fournir un package de mise en page des données pour le langage python.

FIGURE IV.4 – Logo matplotlib

IV.2.2 Outils de développement

Il existe de nombreux outils différents qui sont utilisés pour lire les langages de programmation, dans ce projet, nous avons utilisé un seul outil qui est Pycharm

IV.2.2.1 Pycharm

Il s'agit d'un environnement de développement intégré utilisé dans la programmation informatique, en particulier la programmation en langage Python. Produit par Jetbrains, il aide à analyser le code et à détecter les erreurs De plus, ce logiciel est multiplate-forme, car il peut être utilisé sur divers systèmes d'exploitation.

FIGURE IV.5 – Logo pycharm

IV.3 Schéma éléctronique

Ici, nous clarifions comment les appareils électroniques ont été connectés les uns aux autres lors de la préparation du véhicule à travers la figure suivante :

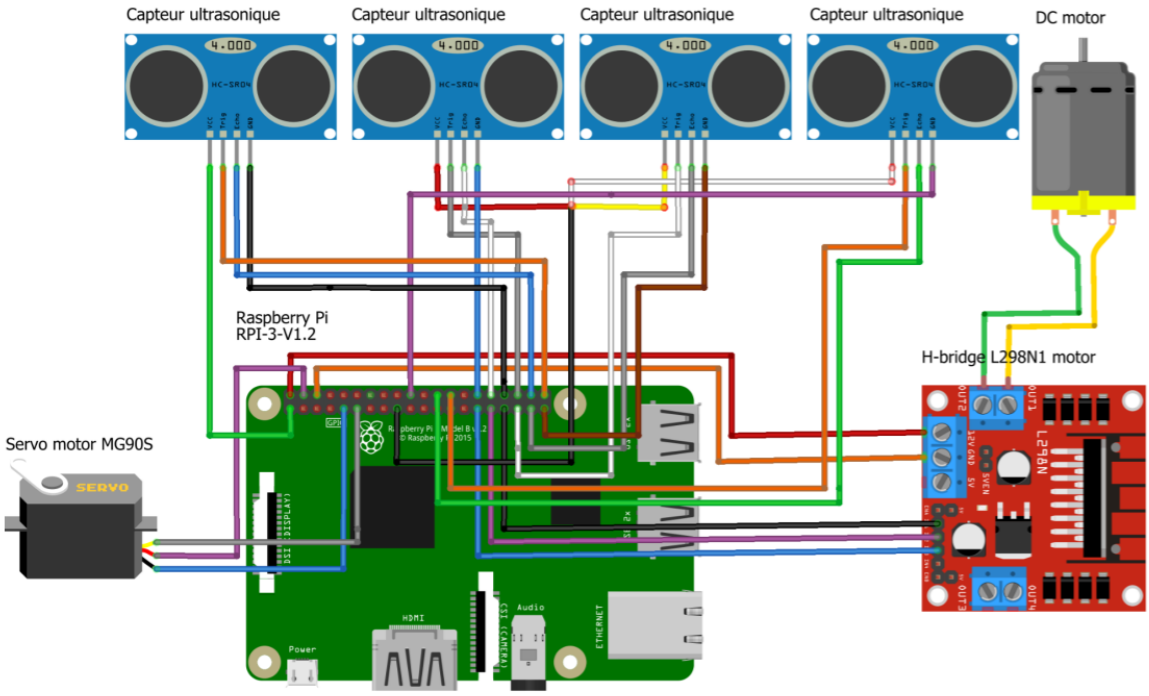

FIGURE IV.6 – Schéma éléctronique

IV.3.1 Raspberry Pi

Le raspberry Pi est un petit ordinateur qui peut être connecté à un moniteur, un clavier et une souris pour le contrôler. C'est aussi un appareil Linux open source qui s'appuie sur Python comme langage de programmation officiel pour sa programmation.

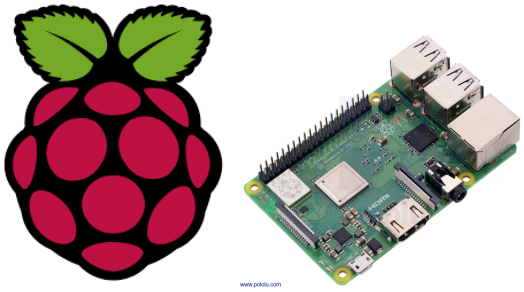

FIGURE IV.7 – Raspberry pi

IV.3.2 Capteur Ultrasonique

Un capteur à ultrasons est un appareil électronique qui mesure la distance à un objet en transmettant des ultrasons (fréquences supérieures au son qu'une personne peut entendre) et en recevant les ondes réfléchies par l'objet. Ce capteur se compose de deux parties principales, l'émetteur et le récepteur.L'émetteur envoie une impulsion d'ultrasons à travers un cristal piézoélectrique.

FIGURE IV.8 – Capteur ultrasonique

IV.3.3 L298N Driver

Un moteur a double pont en H (H-Bridge L298N) qui permet de contrôler simultanément la direction et la vitesse de deux moteurs à courant continu, l'unité peut

entraîner des moteurs à courant continu de tensions comprises entre 5 et 35 volts avec un courant de crête pouvant atteindre 2 ampères.

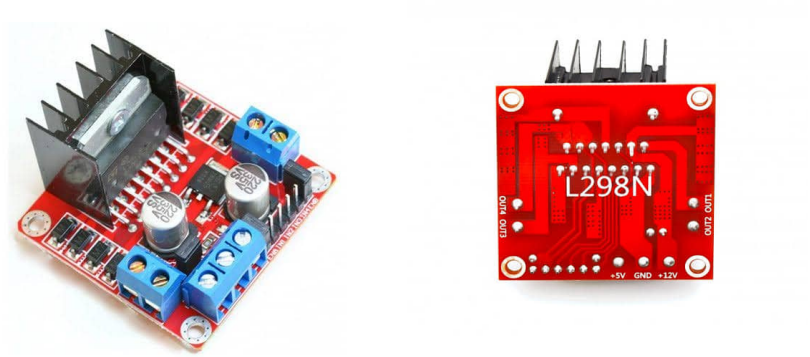

FIGURE IV.9 – H-bridge L298N.

IV.3.4 DC motor

Un moteur à courant continu (DC motor) est un moteur électrique rotatif qui convertit le courant électrique continu (DC) en énergie mécanique.

FIGURE IV.10 – DC motor.

IV.3.5 Servo Motor MG90S

Servomoteur MG90S. Il s'agit d'un petit appareil adapté aux systèmes de direction de petites voitures autonomes en tant que prototype ou d'autres petits véhicules

télécommandés. Principalement caractérisé par sa puissance et sa grande maniabilité, ce moteur permet un réglage précis de la position en virage, de l'accélération et de la vitesse.

FIGURE IV.11 – Servo motor MG90S

IV.4 Installation du matériel

Dans cette section on va présenter l'installation du matériel et les capteurs utilisés dans notre projet, on va détailler les parties fondamentales pour attendre l'objectif de notre projet "la réalisation d'un prototype de voiture autonome et connectée".

La voiture est constituée d'un capteur de stationnement, qui est le capteur latéral droit, et elle est également équipée de deux radars : le radar de recul aide à garer la voiture placée à l'arrière et un autre radar à l'avant, ces deux radars mesurent la distance entre la voiture et l'obstacle en cas de détection d'un objet afin d'envoyer un signal d'arrêt. Il se compose également d'un pont en H (H-bridge) qui l'aide à contrôler la vitesse.

Le raspberry Pi dans la voiture est une assistance technologique intégrée entre plusieurs composants électroniques du dashbord qui lui permet de fournir directement des informations et des données liées à la voiture.

- La figure suivante présenter l'installation d'un radar de recul avant de la voiture et le servomoteur.

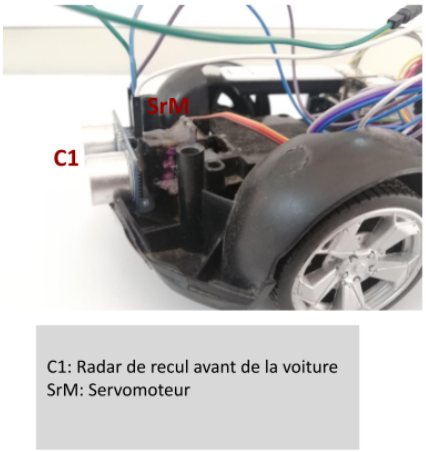

FIGURE IV.12 – Positionnement de servor motor.

- La figure suivante présenter l'installation d'un radar de recul arrière de la voiture, le moteur DC avec le pont en H (H-bridge).

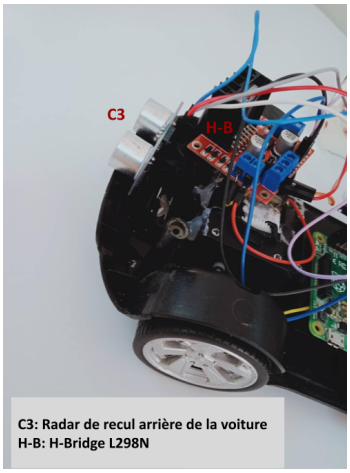

FIGURE IV.13 – Positionnement de le pont en H(H-bridge)

- La figure suivante présenter l'installation de tout le matériel nécessaire dans notre projet.

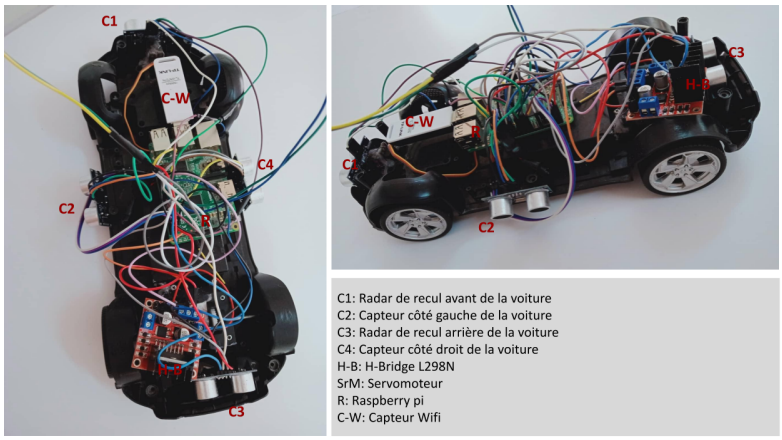

FIGURE IV.14 – Positionnement de tous les pièces d'un voiture.

- Enfin, la figure suivante montre la forme extérieure du projet (voiture indépendante et connectée).

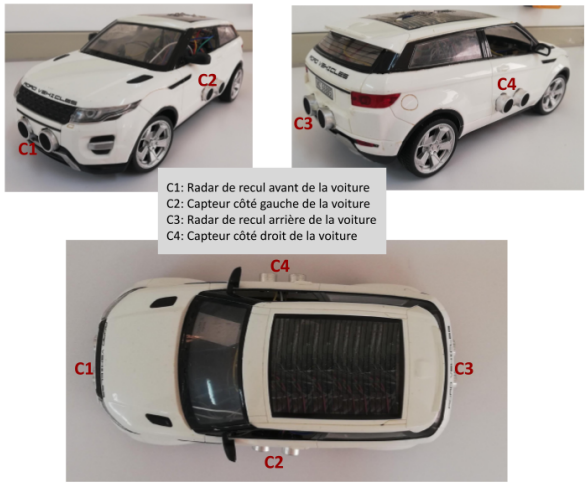

FIGURE IV.15 – Positionnement des capteurs de voiture.

IV.5 Systèmes de pilotage et contrôle de véhicule

Nous montrerons comment nous avons travaillé à travers présenter des parties sur le code utilisé pour développer et programmer le véhicule.

IV.5.1 Véhicule

Nous allons montrer les fonctions les plus importantes dans cette partie du système.

1. **Les bibliothèques nécessaires** pour programmer un véhicule

```
# IMPORTER LES BIBLIOTHQUES NCESSAIRES
import RPi.GPIO as GPIO
import RPi.GPIO as gp
import time
from gpiozero import DistanceSensor
```

codes source IV.1 – Les bibliothèques nécessaires pour le véhicule

2. **Détection la distance :** Ici, la distance est calculée par les capteurs gauches, droite, et radar de recul avant et recule arrière.

```
# les quatres capteurs existant d'un voiture
sensor_gauche = DistanceSensor(trigger=21, echo=20)
sensor_droite = DistanceSensor(trigger=19, echo=26)
sensor_arriere = DistanceSensor(trigger=16, echo=12)
sensor_avant = DistanceSensor(trigger=7, echo=8)

def sensing(sensor):
    distance =round(sensor.distance*100, 2)
    print("distance: {} cm".format(distance))
```

```
10      return distance
```

codes source IV.2 – fonction capteurer la distance

3. **Stop :** Ici, la distance est calculée par le capteur, et si la distance calculée est inférieure ou égale à 20 cm, le véhicule s'arrêtera afin d'éviter de heurter des obstacles.

```
1  Motor1A = 13
2  Motor1B = 6
3  Motor1E = 5
4  def setup():
5      GPIO.setwarnings(False)
6      GPIO.setmode(GPIO.BCM)              # GPIO Numbering
7      GPIO.setup(Motor1A,GPIO.OUT)  # All pins as Outputs
8      GPIO.setup(Motor1B,GPIO.OUT)
9      GPIO.setup(Motor1E,GPIO.OUT)
10 def Stop():
11    setup()
12    GPIO.output(Motor1A,GPIO.LOW)
13    GPIO.output(Motor1E,GPIO.LOW)
14    GPIO.output(Motor1B,GPIO.LOW)
```

codes source IV.3 – fonction stop

4. **Capteur cote_droit :** la fonction côté_droite() pour vérifier l'espace de stationnement de voiture suffisant (elle se stationne sur le côté droit)ou non. Nous remplissons un tableau d'une longueur de "la voiture * 3 " avec les valeurs de la distance calculée à partir par le capteur droit, puis on va vérifier une partie de tableau si elle est équivalente aux longueurs de voiture", la voiture sera garée automatiquement.

```
history_dist = []
car_length = 150
car_width = 25
def cot_droite():
    global history_dist, car_length, car_width, c3, c4
    compteur = 1
    global stat
    dsr = 0
    while stat:
        lenth_distances = len(history_dist)
        if c3 < car_width:
            history_dist.clear()
            dsr = c3
        else:
           if lenth_distances < car_length*3:
               history_dist.append(c3)
               compteur += 1
               p = c3 + car_width
               if compteur >= car_length:
                  Stop()
                  gauche()
                  for i in range(35):
                       dmarage_avant()
                  centre()
                  d =  (math.sqrt(pow(car_length*2, 1.6) + pow(p, 2)))/2
                  while c4 <= d:
                        dmarage_arrire()
                  droite()
                  while c4 <= 10
                      dmarage_arrire()
```

```
31                    stat = False
```

codes source IV.4 – fonction coté_droite()

5. **Avant :** la fonction demarage_avant() pour faire avancer le véhicule.

```
1 def dmarage_avant():
2   setup()
3   GPIO.output(Motor1A,GPIO.LOW)
4   GPIO.output(Motor1B,GPIO.HIGH)
5   GPIO.output(Motor1E,GPIO.HIGH)
6   GPIO.setup(11, GPIO.OUT)
7   GPIO.output(11, 1)
8   sleep(0.002)
9   Stop()
```

codes source IV.5 – Fonction demarage_avant()

6. **Arriere :** la fonction demarage_arriere() pour déplacer le véhicule en marche arrière.

```
1 def  dmarage_arrire ():
2  setup()
3  GPIO.output(Motor1A,GPIO.HIGH)
4  GPIO.output(Motor1B,GPIO.LOW)
5  GPIO.output(Motor1E,GPIO.HIGH)
6  sleep(0.002)
7  Stop()
```

codes source IV.6 – Fonction demarage_arriere()

7. **Tourner droit :** la fonction droite() afin d'ajuster la roue de véhicule à la direction droite.

```
MIN_DUTY = 2
MAX_DUTY = 12
def deg_to_duty(deg):
    return (deg - 0) * (MAX_DUTY- MIN_DUTY) / 180 + MIN_DUTY
#vehicule tourner a droite
def droite():
  angle_stat = 90
  if angle_stat!=180:
    centre()
    for angle_stat in range(181):
              duty_cycle = deg_to_duty(angle_stat)
              Set_Angle(duty_cycle)
```

codes source IV.7 – Fonction droite()

8. **Tourner gauche :** la fonction gauche() afin d'ajuster la roue de véhicule à la direction gauche.

```
#vehicule tourner a gauche
def gauche():
   angle_stat = 90
   if angle_stat!=0:
    centre()
    while angle_stat > 0:
          angle_stat -= 1
          duty_cycle = deg_to_duty(angle_stat)
          Set_Angle(duty_cycle)
```

codes source IV.8 – Fonction gauche()

9. **Tourner centre :** la fonction centre() afin d'ajuster la roue de véhicule à la

direction directe.

```
#vehicule tourner a centre
def centre():
  angle_stat = 90
  if angle_stat!=90:
    if angle_stat<91:
       for angle_stat in range(91):
          duty_cycle = deg_to_duty(angle_stat)
          Set_Angle(duty_cycle)
    elif angle_stat>90 :
       while angle_stat >90:
          angle_stat -= 1
          duty_cycle = deg_to_duty(angle_stat)
          Set_Angle(duty_cycle)
```

codes source IV.9 – Fonction def centre()

IV.5.2 Station de traitement

On verra les fonctions la plus importante dans les codes des trois algorithmes décrits précédemment.

IV.5.2.1 détection d'objets

1. **Bibliothèques nécessaires** pour détecter les obstacles, nous avons besoin d'une bibliothèque, qui est OpenCV

```
# IMPORTER LES BIBLIOTHQUES NCESSAIRES
import cv2
```

codes source IV.10 – Les bibliothèques nécessaires pour la détection des obstacles

2. **détection par DNN :** l'obstacle est détecté par une fonction définie dans la bibliothèque OpenCV appelée :

   cv2.dnn_DetectionModel(conf_path, weight_path)

```
configuration_path = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
weight_path = 'frozen_inference_graph.pb'

model_detecte = cv2.dnn_DetectionModel(configuration_path, weight_path)
```

codes source IV.11 – Détection des objets par DNN

3. **Affecter le nom à l'objet après sa détection :** pour affecter le nom d'obstacle, nous utilisons la fonction cv2.PutText() a la ligne 18, à l'aide du fichier name$_o bjets.Names$.

```
cam = cv2.VideoCapture('http://192.168.43.27:8080/video')
cam.set(3, 740)
cam.set(4, 580)
names = []
file = 'name_objets.names'

with open(file, 'rt') as f:
    names = f.read().rstrip('\n').split('\n')
while True:
    _, image = cam.read()
    idis, confs, bx = model_detecte.detect(image, confThreshold=0.5)

    if len(idis) != 0:
        for idn, conf, box in zip(idis.flatten(), confs.flatten(), bx):
            cv2.rectangle(image, box, color=(0, 255, 0), thickness=2)

            cv2.putText(image, names[idn-1], (box[0]+10, box[1]+20),
            cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), thickness=2)
```

```
19
20              name_object = names[idn-1]
21              print(name_object)
22
23              cv2.putText(image, str(round(confidence * 100, 2)),
24              (box[0] + 200, box[1] + 30),
25              cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
```

codes source IV.12 – Affectation le nom d'objet

IV.5.2.2 détection des lignes de la route

1. **les bébliothèques nécessaires** pour la détection des lignes sont présentées dans le code suivant :

```
1 # IMPORTER LES BIBLIOTHQUES NCESSAIRES
2 import cv2
3 import numpy as np
4 import os
5 from matplotlib import pyplot as plt
```

codes source IV.13 – Les bibliothèques nécessaires pour la détection des lignes

2. **code d'une fonction processImage() comme suivant :**

```
1 def processImage(inpImage):
2
3     # Appliquez le filtrage de couleur HLS pour filtrer les lignes de voies
      blanches
4     hls = cv2.cvtColor(inpImage, cv2.COLOR_BGR2HLS)
5     lower_white = np.array([0, 160, 10])
6     upper_white = np.array([255, 255, 255])
```

```
    mask = cv2.inRange(inpImage, lower_white, upper_white)
    hls_result = cv2.bitwise_and(inpImage, inpImage, mask=mask)
    # Convertir l'image en niveaux de gris, appliquer un seuil, flouter et
    extraire les bords
    gray = cv2.cvtColor(hls_result, cv2.COLOR_BGR2GRAY)
    ret, thresh = cv2.threshold(gray, 160, 255, cv2.THRESH_BINARY)
    blur = cv2.GaussianBlur(thresh, (3, 3), 0)
    canny = cv2.Canny(blur, 40, 60)
    return image, hls_result, gray, thresh, blur, canny
```

codes source IV.14 – Fonction processImage()

3. code d'une fonction measureLaneCurvature

```
def measureLaneCurvature(ploty, leftx, rightx):

    leftx = leftx[::-1]  # Inverser pour correspondre de haut en bas en y
    rightx = rightx[::-1]  # Inverser pour correspondre de haut en bas en y
    # Choisissez la valeur y maximale, correspondant au bas de l'image
    y_eval = np.max(ploty)

    # Ajuster de nouveaux polynmes  x, y dans l'espace mondial
    left_fit_cr = np.polyfit(ploty*ym_per_pix, leftx*xm_per_pix, 2)
    right_fit_cr = np.polyfit(ploty*ym_per_pix, rightx*xm_per_pix, 2)

    # Calculer les nouveaux rayons de courbure
    left_curverad = ((1 + (2*left_fit_cr[0]*y_eval*ym_per_pix + left_fit_cr
    [1])**2)**1.5) / np.absolute(2*left_fit_cr[0])
    right_curverad = ((1 + (2*right_fit_cr[0]*y_eval*ym_per_pix +
    right_fit_cr[1])**2)**1.5) / np.absolute(2*right_fit_cr[0])

    # Maintenant, notre rayon de courbure est en mtres
```

```
    # print(courbe_gauche, 'm', courbe_droite, 'm')
    # Dcidez s'il s'agit d'une courbe  gauche ou  droite
    if leftx[0] - leftx[-1] > 60:
        curve_direction = 'Gauche'
    elif leftx[-1] - leftx[0] > 60:
        curve_direction = 'Droite'
    else:
        curve_direction = 'Direct'
    return (left_curverad + right_curverad) / 2.0, curve_direction
```

codes source IV.15 – Fonction measureLaneCurvature()

4. **code d'une fonction perspectiveWarp()**

```

def perspectiveWarp(inpImage):

    # Obtenir la taille de l'image
    img_size = (inpImage.shape[1], inpImage.shape[0])
    # Points de perspective  dformer
    src = np.float32([[590, 440], [690, 440], [200, 640], [1000, 640]])
   # Fentre    afficher
    dst = np.float32([[200, 0], [1200, 0], [200, 710], [1200, 710]])
    # Matrice pour dformer l'image
    matrix = cv2.getPerspectiveTransform(src, dst)
    # Matrice inverse pour dformer l'image pour la fentre finale
    minv = cv2.getPerspectiveTransform(dst, src)
    birdseye = cv2.warpPerspective(inpImage, matrix, img_size)

    height, width = birdseye.shape[:2]

    birdseyeLeft = birdseye[0:height, 0:width // 2]
```

```
    birdseyeRight = birdseye[0:height, width // 2:width]

    return birdseye, birdseyeLeft, birdseyeRight, minv
```

codes source IV.16 – Fonction perspectiveWarp()

5. **code d'une fonction offCenter()**

```
def offCenter(meanPts, inpFrame):

    # Calcul de l'cart en mtres
    mpts = meanPts[-1][-1][-2].astype(int)
    pixelDeviation = inpFrame.shape[1] / 2 - abs(mpts)
    deviation = pixelDeviation * xm_per_pix
    direction = "Gauche" if deviation < 0 else "Droite"

    return deviation, direction
```

codes source IV.17 – Fonction offCenter()

IV.5.2.3 détection des pannaux signalisation

1. **Les bibliothèques nécessaires** pour la détection des pannaux signalisation sont présentées dans le code suivant :

```
# IMPORTER LES BIBLIOTHQUES NCESSAIRES
import cv2
import numpy as np
from scipy.stats import itemfreq
```

codes source IV.18 – Les bibliothèques nécessaires pour la détection des pannaux signalisation

2. **Code d'une fonction get_dominant_color()**

```
def get_dominant_color(image, n_colors):

    pixels = np.float32(image).reshape((-1, 3))
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 200, .1)
    flags = cv2.KMEANS_RANDOM_CENTERS
    flags, labels, centroids = cv2.kmeans(pixels, n_colors, None, criteria,
    10, flags)
    palette = np.uint8(centroids)

    return palette[np.argmax(itemfreq(labels)[:, -1])]
```

codes source IV.19 – Fonction get_dominant_color()

3. **Les conditions nécessaires pour la détection des pannaux signalisation** sont présentées dans le code suivant :

```
dominant_color = get_dominant_color(square, 2)
if dominant_color[2] > 100:
    print("stop")
elif dominant_color[0] > 80:
    zone_0 = square[square.shape[0] * 3 // 8:square.shape[0]
    * 5 // 8, square.shape[1] * 1 // 8:square.shape[1] * 3 // 8]
    cv2.imshow('Zone0', zone_0)
    zone_0_color = get_dominant_color(zone_0, 1)
    zone_1 = square[square.shape[0] * 1 // 8:square.shape[0]
     * 3 // 8, square.shape[1] * 3 // 8:square.shape[1] * 5 // 8]
    cv2.imshow('Zone1', zone_1)
    zone_1_color = get_dominant_color(zone_1, 1)
    zone_2 = square[square.shape[0] * 3 // 8:square.shape[0]
    * 5 // 8, square.shape[1] * 5 // 8:square.shape[1] * 7 // 8]
```

```
    cv2.imshow('Zone2', zone_2)
    zone_2_color = get_dominant_color(zone_2, 1)

    if zone_1_color[2] < 60:
        if sum(zone_0_color) > sum(zone_2_color):
            print("gauche")
        else:
            print("droite")
    else:
        if sum(zone_1_color) > sum(zone_0_color)  and sum(zone_1_color) >
        sum(zone_2_color):
            print("avant")
        elif sum(zone_0_color) > sum(zone_2_color):
            print("avant et gauche")
        else:
            print("avant et droite")
else:
    print("N/A")
```

codes source IV.20 – Les conditions nécessaires pour détecter pannaux signalisation

IV.5.3 Dashboard

Nous avons créé le dashbord pour résumer ce que la voiture peut faire grâce aux boutons à commande manuelle. Il contient plusieurs boutons pour contrôler la voiture manuellement et un autre bouton pour lui donner le droit à la conduite autonome en plus de surveiller son mouvement grâce à la caméra.

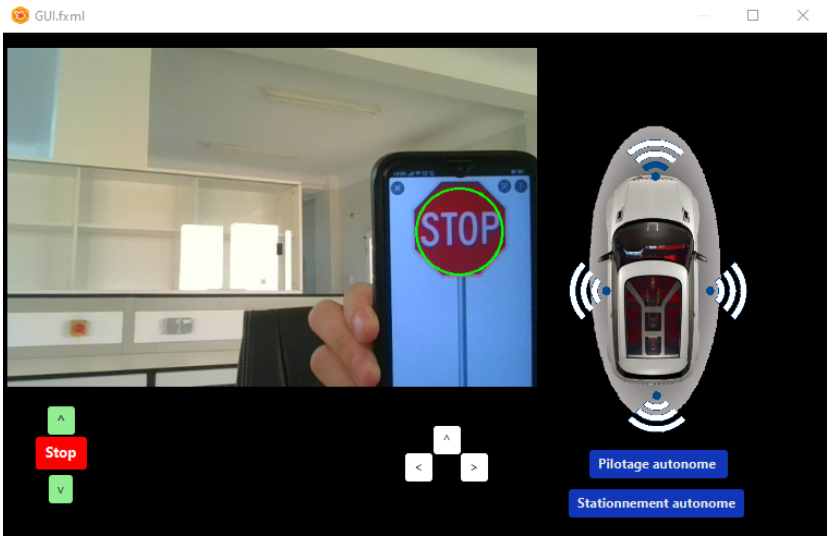

FIGURE IV.16 – Dashboard

- pour l'interconnexion entre les trois phases présédent (véhicule, station de traitement et le dashboard) ont utilié le protocole de communication "UDP" parce qu'il est permis une communication des données rapides sans délai.

IV.6 Résultats et discussion

Nous présenterons et discuterons certains des résultats que le véhicule obtiendra lors de son déplacement.

- **Résultats pour la détection des lignes**
  - Si la direction obtenue est à gauche, comme sur la figure ci-dessous, le véhicule fait tourner ses roues dans le sens gauche et continu sa route.

FIGURE IV.17 – Direction gauche

- Si la direction obtenue est une direction directe (tout droit), comme sur la figure ci-dessous, le véhicule ramène ses roues en position centrale et continue sa route.

FIGURE IV.18 – Direction direct

- Mais si la direction obtenue est la droite direction, comme sur la figure ci-dessous, le véhicule fait tourner ses roues dans la droite direction et

continue sa route.

FIGURE IV.19 – Direction droite

- **Résultats pour la détection d'obstacles**
  - Lorsque vous trouvez une voiture roulant devant elle, laissez la distance de sécurité tout en vous déplaçant ou arrêtez-vous si la voiture est à l'arrêt.

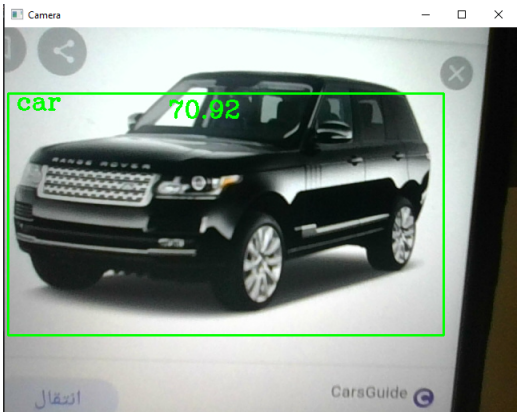

FIGURE IV.20 – Détection la voiture

  - Lorsque vous trouvez un bus roulant devant elle, laissez la distance de sécurité tout en vous déplaçant ou arrêtez-vous si un bus est à l'arrêt.

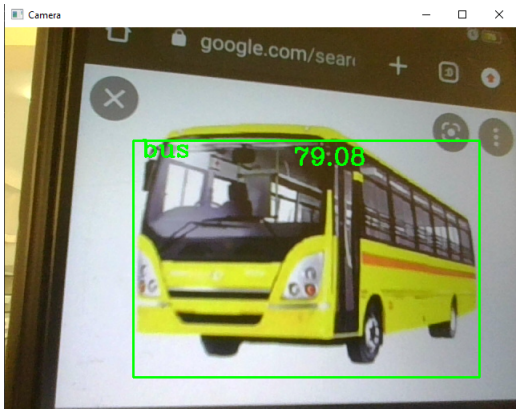

FIGURE IV.21 – Détection des bus

- Lorsque vous voyez une moto avancer, gardez une distance de sécurité tout en vous déplaçant et arrêtez-vous si la moto s'arrête.

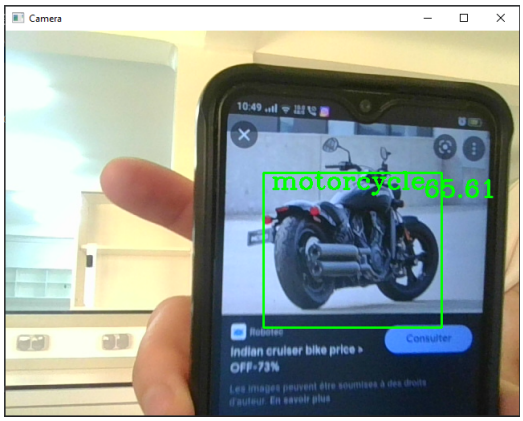

FIGURE IV.22 – Détection des motos

- Le véhicule s'arrête lorsqu'une personne passe devant lui en laissant la distance de sécurité.

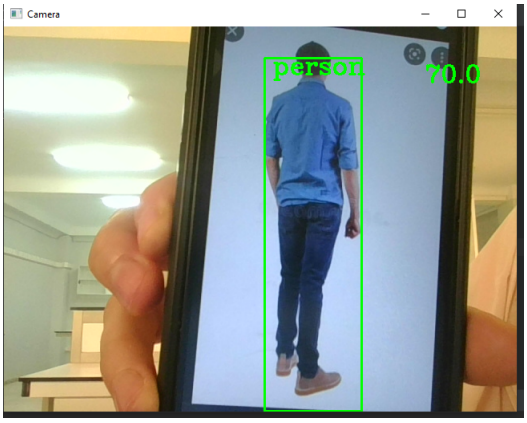

FIGURE IV.23 – Détection des personnes

- **Résultats pour la détection des pannaux signalisation**
  - Lorsque le véhicule détecte la présence d'un panneau "obligé d'avancer", il est obligé d'avancer, il doit continuer sa route car il n'y a pas d'autre chemin que lui.

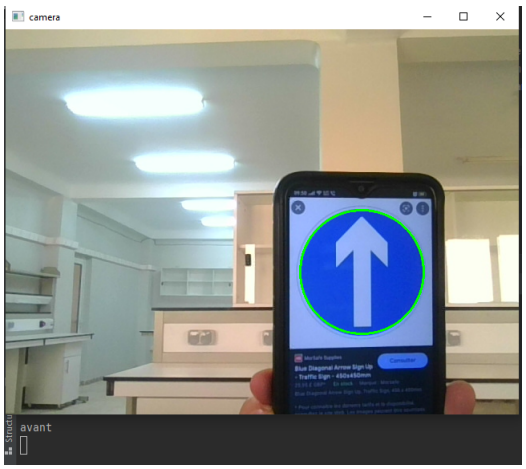

FIGURE IV.24 – Pannau avant

  - Lorsque le véhicule détecte la présence d'un panneau "obligé d'avancer ou

gauche", il doit choisir entre les deux routes et poursuivre sa route car il n'y a pas d'autre chemin que celles-ci.

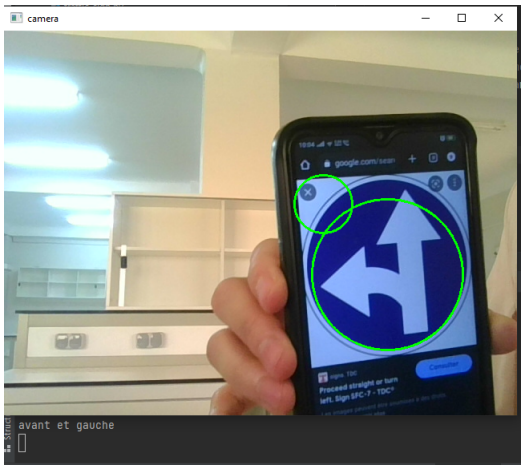

FIGURE IV.25 – Pannau avant et gauche

- Lorsque le véhicule détecte la présence d'un panneau "obligatoire à gauche", il doit continuer sa route gauche car il n'y a pas d'autre route que lui.

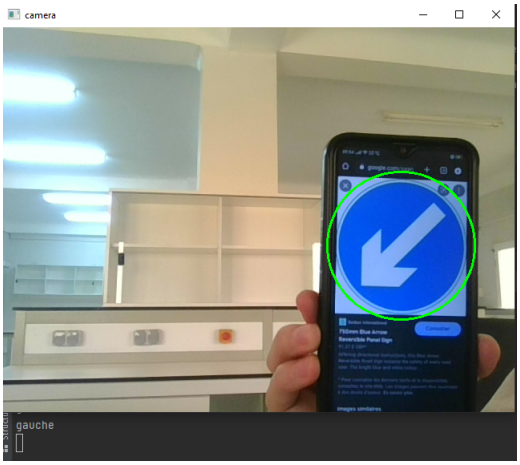

FIGURE IV.26 – Pannau gauche

- Lorsque le véhicule détecte la présence d'un panneau "obligatoire à droite", il doit continuer sa route droite car il n'y a pas d'autre route que lui.

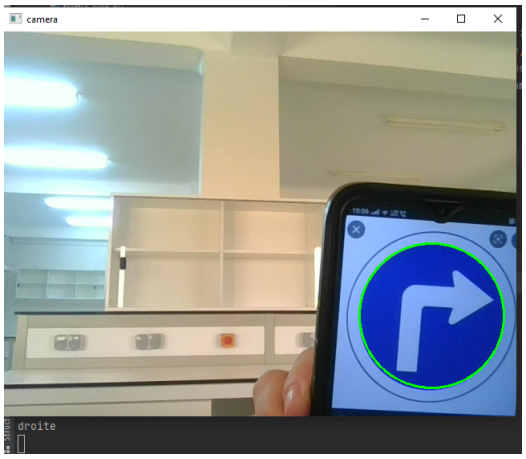

FIGURE IV.27 – Pannau droite

- Lorsque le véhicule détecte la présence d'un panneau "stop", il doit s'arrête.

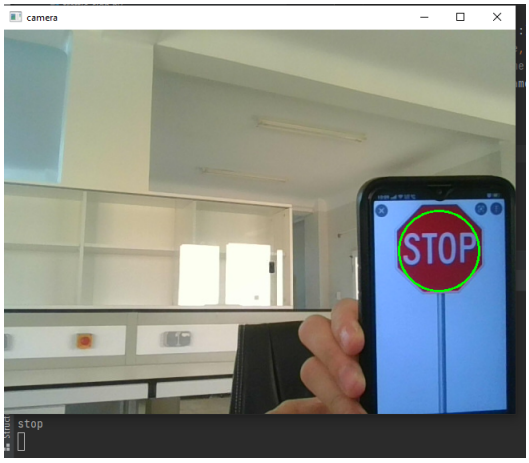

FIGURE IV.28 – Pannau stop

IV.7 Conclusion

Dans ce chapitre, nous avons défini les outils, logiciels et matériels utilisés pour développer notre système, et nous avons également montré les morceaux de code que nous avons créés à chaque étape de l'architecture du système, ainsi que la manière de connecter des appareils électroniques les uns aux autres pour créer un modèle de voiture autonome et définir chacun des appareils séparément. Enfin, nous avons également présenté certains des résultats obtenus à l'issue du développement de notre système.

Conclusion Générale

Dans ce mémoire, nous avons présenté quatre chapitres, le premier chapitre portait sur l'IoT et certains concepts connexes, son fonctionnement, ses architectures et ses domaines, et le deuxième chapitre portait sur l'Internet des véhicules dans lequel nous avons présenté les niveaux d'autonomie et le fonctionnement des voitures autonomes et quelques concepts de base etc. Et dans le troisième chapitre nous avons montré l'architecture générale et détaillé de notre système et expliqué comment il fonctionne, et enfin nous avons présenté le schéma électronique du système et l'interface principale(dashboard) du système et de ses composants et nous avons discuté des différents resultats que nous avons obtenus.

Dans le future, nous doterons notre système à la fois le Lidar pour améliorer la visibilité et capteur GPS pour suivre la voiture , en plus du radar et de l'odomètre pour augmenter la sécurité du mouvement de la voiture.

Bibliographie

[1] Younes Abbassi and Habib Benlahmer. Un aperçu sur la sécurité de l'internet des objets (iot). In *Colloque sur les Objets et systèmes Connectés-COC'2021*, 2021.

[2] Avishkar Seth Alice James and S. C. Mukhopadhyay. Autonomous ground vehicle for off-the-road applications based on neural network. pages 285–287.

[3] Ludovic Apvrille and Letitia W Li. Sécurite des véhicules connectés et/ou autonomes. *MISC Multi-System & Internet Cookbook*, (87) :56–65, 2016.

[4] Yassine Banouar. *Gestion autonomique de la QoS au niveau middleware dans l'IoT*. PhD thesis, Université Paul Sabatier-Toulouse III, 2017.

[5] Badr BENMAMMAR. Optimisation des performances de l'iot une approche basée : sur la radio intelligente. *La gestion et le contrôle intelligents des performances et de la sécurité dans l'IoT*, page 57, 2022.

[6] Damien BRULIN, Chantal LABAT, and Fabrice PEYRARD. Objets connectés : des radiofréquences aux réseaux.

[7] Lucile Buisson, Jean-Pierre Nicolas, and Nathalie Gouget. L'expérimentation des véhicules autonomes : quelle place dans le processus d'innovation ? *RTS-Recherche Transports Sécurité*, 2021 :13, 2021.

[8] Antoine CAMUSAT, Anas KATIM, Marion MESNIL, Pan RUIDONG, Yuxuan SONG, and Abdelaziz BENSRHAIR. La stereovision pour la perception du vehicule autonome.

[9] Nadia Chaabouni. *Détection et prévention des intrusions pour les systèmes IoT en utilisant des techniques d'apprentissage*. PhD thesis, Bordeaux, 2020.

[10] Sunil Cheruvu, Anil Kumar, Ned Smith, and David M Wheeler. *Demystifying internet of things security : successful iot device/edge and platform security deployment*. Springer Nature, 2020.

[11] Hugo Cusanno, Christine Vidal-Gomel, and Sophie Le Bellu. Comment guider les genèses instrumentales pour la prise en main d'un véhicule automatisé ? In *55ème Congrès de la SELF, L'activité et ses frontières. Penser et agir sur les transformations de nos sociétés.*, pages 336–341, 2021.

[12] Eric Enregle. Aux sources du véhicule connecté. *Question (s) de management*, (4) :135–145, 2016.

[13] Aurélia FONTAINE, Alexia GROSS, Julien LECLERC, Thileepan UMAPATHIPILLAI, Huajin WANG, Tonglin YAN, and Abdelaziz BENSRHAIR. Véhicule autonome : Les systèmes de perception embarqués.

[14] Michel Freyssenet. La" production réflexive", une alternative à la" production de masse" et à la" production au plus juste" ? *Sociologie du travail*, pages 365–388, 1995.

[15] N. Baba Shayeer B. Rakesh G. Babu Naik, Prerit Ameta and S. Kavya Dravida. Convolutional neural network based on self-driving autonomous vehicle(cnn). pages 929–930.

[16] Sarath Chandu Gaddam and Mritunjay Kumar Rai. A comparative study on various lpwan and cellular communication technologies for iot based smart ap-

plications. In *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)*, pages 1–8. IEEE, 2018.

[17] David Gaidioz. Implémentation de synthèse de fréquence très faible consommation, pour applications iot, en technologie 28fdsoi. 2021.

[18] Dominique Gaïti. *Architecture et protocoles applicatifs pour la chorégraphie de services dans l'Internet des objets*. PhD thesis, Université Paris-Est, 2013.

[19] Clara Gandrez, Fabrice Mantelet, Améziane Aoussat, Francine Jeremie, and Eric Landel. Modele comportemental d'identification de situations limites en conduite autonome.

[20] Marie GRIFFON, Clément MAUGET, Corentin GUYOT, Elena SPOTO, Victor MOREL, and Peiwen YANG. Internet des objets pour la smartcity.

[21] NASSIRA GUERFI. La navigation des véhicules autonomes par un algorithme génétique.

[22] Jean-Baptiste Haué, Sophie Le Bellu, and Cécile Barbier. Le véhicule autonome : se désengager et se réengager dans la conduite. *Activités*, (17-1), 2020.

[23] Nicolas Hautière. La route du futur. *Annales des Mines-Enjeux Numériques*, (7) :pp–15, 2019.

[24] Nicolas Hautiere, Hélène Tattegrain, and Michèle Guilbot. Véhicules connectés et autonomes : quels enjeux technologiques, juridiques et de sécurité routière ? *Hygiène et Sécurité du Travail*, (246) :pp–100, 2017.

[25] Nagamalai Jeyanthi. Internet of things (iot) as interconnection of threats (iot). In *Security and Privacy in Internet of Things (IoTs)*, pages 23–42. CRC Press, 2016.

[26] Gérard Le Lann. Anonymat, non-traçabilité et sécurité-innocuité dans les réseaux de véhicules autonomes connectés. In *8ème Atelier sur la Protection de la Vie Privée (APVP'17)*, 2017.

[27] Philippe Legault. Tendances de la servitisation dans le secteur manufacturier : comparaison de la valeur des solutions analytiques génériques et spécialisées basées sur l'iot avec topsis. 2019.

[28] Carsten Maple, Matthew Bradbury, Anh Tuan Le, and Kevin Ghirardello. A connected and autonomous vehicle reference architecture for attack surface analysis. *Applied Sciences*, 9(23) :5101, 2019.

[29] Guadalupe Ortiz, Meftah Zouai, Okba Kazar, Alfonso Garcia-de Prado, and Juan Boubeta-Puig. Atmosphere : Context and situational-aware collaborative iot architecture for edge-fog-cloud computing. *Computer Standards & Interfaces*, 79 :103550, 2022.

[30] Keyur K Patel, Sunil M Patel, et al. Internet of things-iot : definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, 6(5), 2016.

[31] Rabeb Saad. *Modèle collaboratif pour l'Internet of Things (IoT)*. PhD thesis, Université du Québec à Chicoutimi, 2016.

[32] Imad Saleh. Internet des objets (ido) : Concepts, enjeux, défis et perspectives. *Revue Internet des objets*, 2(10.21494), 2018.

[33] Tidiane Sylla. *Sécurité et vie privée centrées sur l'utilisateur dans l'IoT*. PhD thesis, Bordeaux, 2021.

[34] Tidiane Sylla. *Sécurité et vie privée centrées sur l'utilisateur dans l'IoT*. PhD thesis, Bordeaux, 2021.

[35] Quoc-Vu Dang Truong-Dong Do, Minh-Thien Duong and My-Ha Le. Real-time self-driving car navigation using deep neural network. *International Conference on Green Technology and Sustainable Development (GTSD)*, (4) :1–2, 2018.

[36] Mathieu Weill and Mohsen Souissi. L'internet des objets : concept ou réalité ? In *Annales des Mines-Réalités industrielles*, number 4, pages 90–96. Eska, 2010.

[37] Fangchun Yang, Jinglin Li, Tao Lei, and Shangguang Wang. Architecture and key technologies for internet of vehicles : a survey. *Journal of Communications and Information Networks*, 2(2) :1–17, 2017.

[38] Meftah Zouai, Okba Kazar, Guadalupe Ortiz Bellot, Belgacem Haba, Nadia Kabachi, and M Krishnamurhty. Ambiance intelligence approach using iot and multi-agent system. *International Journal of Distributed Systems and Technologies (IJDST)*, 10(1) :37–55, 2019.

[39] Meftah Zouai, Okba Kazar, Belgacem Haba, Guadalupe Ortiz, and Nadia Kabachi. New approach using an iot robot to oversight the smart home environment. *ISTEOpenScience–Published by ISTE Ltd. London, UK–openscience. fr*, 2019.

[40] Meftah Zouai, Okba Kazar, Belgacem Haba, and Hamza Saouli. Smart house simulation based multi-agent system and internet of things. In *2017 International Conference on Mathematics and Information Technology (ICMIT)*, pages 201–203. IEEE, 2017.

[41] Meftah Zouai, Guadalupe Ortiz, and Okba Kazar. Paving the way to industry 4.0 : an approach based on multi-agents system and complex event processing.