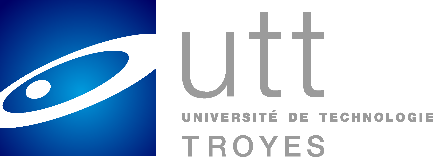
Une image contenant noir, capture d’écran, conception, noir et blanc

Description générée automatiquementUne image contenant ciel, nuage, brouillard, bâtiment

Description générée automatiquement

# Genero

Rapport d’UE CS06 A23

Odyssé LOIZEAU, Kléden ROUSSEAU, Fabien Gautier

Semestre

Table des matières

[[UE - intitulé exact] 1](#_Toc155899095)

[1. Présentation du projet 2](#_Toc155899096)

[1.1. Contexte et objectifs 2](#_Toc155899097)

[1.2. Cahier des charges 3](#_Toc155899098)

[1.3. Gestion de projet 5](#_Toc155899099)

[1.4. Équipe de projet 9](#_Toc155899100)

[2. Création de l’application web 14](#_Toc155899101)

[2.1. Présentation du framework Django 14](#_Toc155899102)

[2.2. Mise en place 15](#_Toc155899103)

[2.3. Présentation de l’application 17](#_Toc155899104)

[Modélisation 3D 19](#_Toc155899105)

[3. Intelligence Artificielle 22](#_Toc155899106)

[3.1. Création et équilibrage du dataset 22](#_Toc155899107)

[3.2. Machine Learning 24](#_Toc155899108)

[4. Limites du projet 29](#_Toc155899109)

[4.1. Améliorations et ajouts 29](#_Toc155899110)

[4.2. Difficultés 29](#_Toc155899111)

[4.3. Critiques et perspectives 31](#_Toc155899112)

[5. Conclusion 33](#_Toc155899113)

[5.1. Apports personnels 33](#_Toc155899114)

[5.2. Conclusion générale 34](#_Toc155899115)

## 1. Présentation du projet

### 1.1. Contexte et objectifs

L’émergence de l’intelligence artificielle est devenue incontestable. Ce domaine de recherche, qui n’était encore qu’une niche il y a quelques années, est même parvenu à toucher le grand public par une percée fulgurante. L’une des raisons qui explique cet engouement, c’est notamment la facilité d’accès à de très puissants modèles d’IA génératifs. Parmi les plus connus, on peut citer *ChatGPT*, *Midjourney* ou encore *DALL-E*. Ces innovations, aujourd’hui accessibles à portée de clavier, ouvrent des perspectives encore insoupçonnées.

Sur un autre plan, le développement de la robotique et des technologies qui y sont liées est aussi remarquable. L’apparition de standard de développement multi-plateforme (ROS), l’amélioration et la miniaturisation des capteurs/actionneurs ainsi que les progrès effectués sur les structures mécaniques des robots permettent de décupler le champ des possibles. La robotique devient de plus en plus agile et ses enjeux sont multiples : adaptabilité de l’automatisation, substitution de l’humain pour des tâches pénibles, dangereuses ou dans des milieux inhospitaliers. En bref, des problématiques concrètes qui, si elles sont résolues, amélioreront sensiblement les conditions de vie des êtres humains.

C’est à l’intersection de ces deux domaines que nous est venue notre idée de projet. Si je lui demande correctement, ChatGPT est capable de m’écrire une petite histoire dans le style des fables de La Fontaine. Aussi, si je demande à Midjourney de me générer une image d’un ours blanc qui participe au Tour de France, le modèle sera en mesure de me fournir un résultat convaincant.

D’un autre côté, si je souhaite développer un robot qui aura pour mission principale d’inspecter les égouts, cela risque de me prendre un peu de temps ainsi qu’un peu d’argent avant d’obtenir un produit réellement efficace. Ensuite, imaginons que je souhaite à nouveau développer un robot d’inspection, cette fois-ci dédié à l’inspection des façades de bâtiments. Pour mener à bien ce nouveau projet, je pourrais repartir sur la base de mon précédent robot et l’adapter à sa nouvelle mission, ce qui certes me fera économiser temps et de l’argent, mais m’en prendra quand même un peu. Néanmoins, ne serait-il pas possible d’automatiser ce processus de création ? La base du robot restant plus ou moins la même, juste l’environnement et quelques autres paramètres changent.

Nouvelle hypothèse : un ami souhaite lui aussi développer un robot d’inspection, spécialisé cette nouvelle fois dans l’analyse des structures en polymère. Cet ami est expert en polymère et pourra très précisément décrire les missions du robot. Cependant, il n’a que très peu de notions en robotique et même avec mes plans et schémas, il ne pourra aller bien loin…

Ce qui lui faudrait, ce serait un outil capable de l’assister et le guider dans sa conception. Cet outil pourrait fournir une liste cohérente de composants selon de nombreux critères à définir (le contexte, les divers besoins, la gamme de prix, etc). Également, l’outil pourrait fournir des éléments de modélisation 3D du robot qui pourront être simulés, exportés et testés virtuellement. Par ailleurs, il pourrait aussi apporter des éléments de recadrage lorsque les exigences spécifiées semblent irréalistes ou incomplètes. En bref, créer un outil intelligent d’aide à la conception de robots. En outre, l’un des objectifs corollaire de ce projet est aussi de découvrir et/ou monter en compétences dans des domaines aujourd’hui en pleine expansion, entre autres : l’IA, la robotique, la CAO et le développement web.

### 1.2. Cahier des charges

Au départ, il nous a fallu fixer le cadre de notre projet. Le but étant d'implémenter au travers d’une application différentes caractéristiques d’assistance à la conception de robots pour les utilisateurs.

#### 1.2.1. Application

L’application sera une application web. Elle offrira des avantages tels que la visualisation dynamique des résultats (liste de composants, visualisation 3D), la facilité d’accès au travers de n’importe quel navigateur internet et depuis n’importe quel terminal, ainsi que la conception modulaire permettant d’utiliser des bibliothèques et ressources préexistante, nous facilitant ainsi considérablement le développement.

Dans la même idée de simplification du développement de l’application web, on se basera sur le framework python de développement web Django, bien connu pour sa facilité de maintenance, ses ressources et sa versatilité. Autre avantage non-négligeable, la création en arrière-plan de la base de données du site web.

#### 1.2.2. Interprétation du texte par l’IA

Un modèle d’IA destiné à l’interprétation du texte sera implémenté. Son objectif sera de renvoyer une liste cohérente de composants adaptés aux besoins spécifiés par l’utilisateur. D’abord, il devra être en mesure de fournir une liste simple de composants aptes à fonctionner ensemble. Puis, on pourra imaginer l’ajout de critères pertinents tels que la prise en compte de l’environnement dans lequel évoluera le robot, mais aussi les coûts induits par les différents composants, la durabilité de l’assemblage, les contraintes mécaniques qui pourraient s’appliquer, etc.

Plusieurs possibilités seront à explorer notamment avec l’utilisation d'algorithmes, la création de modèles ou la ré-utilisation de modèles existants (Fine-Tuning). L’interprétation de texte par un modèle d’IA LLM (Large Language Model) peut-être effectué à travers différentes méthodes, mais la plus efficace reste la tokenisation, dont le principe consiste à associer un identifiant à chaque mot et à vectoriser le texte avec ces ID afin de comparer des ensembles d’entiers à travers un modèle entraîné pour ce besoin.

#### 1.2.3. Modélisation 3D

L'application devra proposer une fonction d’export des fichiers 3D pour qu’ils puissent être exploitables dans d’autres applications et même éventuellement offrir une option de visualisation de certains composants du robot. On pourrait imaginer imprimer en 3D certains des éléments modélisés ou tout simplement les tester au travers d’outils de simulation, comme Gazebo ou Solidworks.

Il faudra aussi déterminer le type d'extension de fichier 3D le plus adapté à notre projet. Soit quelque chose de simplement descriptif et fonctionnel facile à implémenter, ou bien quelque chose de plus lourd mais qui prendra en compte les aspects techniques et physiques, permettant par la suite de faire des tests d’autant plus réalistes et précis.

#### 1.2.4. Expérience utilisateur

L’interface devra être claire et simple d’utilisation. D’un côté, on retrouvera une case à cocher permettant de spécifier l’un des trois types de robot (transport, manipulation, inspection) pris en charge par l'application. Aussi, il y aura un champ de texte qui permettra d'acquérir les exigences (plus ou moins précises) des utilisateurs.

En dessous, on retrouvera une case dédiée aux résultats qui seront fournis par le modèle d’IA, à savoir la liste des composants ainsi que potentiellement d’autres caractéristiques comme le coût par exemple. Il pourrait être intéressant de fournir une option de téléchargement qui renverrait le détail des informations obtenues sous la forme d’un court rapport dans un fichier texte par exemple.

De l’autre côté, on imaginera la fenêtre de visualisation du modèle 3D. Soit d’un élément, d’un assemblage ou d’une tentative de représentation du robot fourni par l’IA. Possiblement, la fenêtre pourra être interactive et bénéficiera également d’une option de téléchargement du fichier 3D dont l’extension sera à définir.

Ci-dessous, vous trouverez un premier schéma de ce à quoi pourrait ressembler l’interface utilisateur finale. Cette dernière sera très certainement amenée à évoluer selon les avancées, les changements de critères et les difficultés techniques rencontrées lors des développements successifs de l’application web.

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

*Figure X : Premier schéma d’interface pour l’application web*

#### 1.2.5. Rendu final

À l'issue de ce projet, nous présenterons un “proof of concept” de l’ensemble de notre travail. Idéalement, le rendu devra implémenter un maximum des caractéristiques détaillées dans ce cahier des charges, en gardant à l’esprit cependant que nous serons probablement confrontés à des difficultés techniques, de moyens et/ou de capacités.

### 1.3. Gestion de projet

#### 1.3.1. Méthodologie agile (adaptée)

Sur ce projet, nous n’étions pas à temps plein, ce qui explique pourquoi nous avons dû mettre en place une version adaptée de la méthodologie de gestion de projet SCRUM. Parmi les principaux aspects sélectionnés et adaptés, on retrouve :

* Réunions bihebdomadaires courtes d’au maximum un quart d’heure, permettant de tenir l’ensemble de l’équipe au courant des avancées, réalisations et blocages de chacun.
* Réunions mensuelles qui permettent de faire l’état des lieux de l’avancement du projet. Point étape pour discuter en détail des progrès, définir ou redéfinir certaines priorités et corriger certains dysfonctionnements.
* Développement continu de sorte à stimuler l’avancement du projet. Nous nous sommes imposé une consigne simple : pour chaque séance de travail, nous devions produire quelque chose et ce, même s’il y avait des imperfections. Adopter cette méthodologie de travail nous a poussé à être actif tout au long du projet ainsi qu’à éviter de perdre des séances à faire de la documentation à rallonge.

#### 1.3.2. Outils et logiciels

Pour mener à bien le projet, nous nous sommes appuyés sur plusieurs outils et logiciels ayant chacun leurs propres spécificités.

* **GitHub :** Une plateforme en ligne utilisée pour le stockage, le partage et la gestion de code source dans le cadre de projets de développement logiciel. Elle utilise le système de contrôle de version Git, nous permettant de suivre et de gérer les modifications apportées au code au fil du temps. GitHub facilite la collaboration entre les développeurs, permettant à plusieurs personnes de travailler ensemble sur le même projet, en fusionnant leurs modifications et en résolvant les conflits de code. La plateforme offre également des fonctionnalités telles que la gestion des bugs, la demande de tirage (pull requests), la révision de code et l'intégration continue. Elle est largement utilisée par les programmeurs individuels, les équipes de développement et les grandes entreprises pour héberger et gérer leurs projets logiciels.

Nous nous sommes également appuyés sur l’extension ***GitHub Desktop*** qui est un outil permettant de simplifier l'utilisation de GitHub en offrant une interface graphique intuitive pour gérer les dépôts (ou "repositories"). Il nous a permis de cloner, créer des branches, commit et push les modifications sur GitHub sans avoir à utiliser des commandes en ligne de commande. De plus, GitHub Desktop nous a facilité la collaboration sur le projet en nous permettant de synchroniser facilement nos travaux et de visualiser les différences entre les versions de fichiers.

* **Visual Studio Code (VS Code) :** Un éditeur de code source gratuit et puissant développé par Microsoft. Il est conçu pour soutenir le développement de logiciels en offrant des fonctionnalités telles que le support de nombreux langages de programmation (principalement Python, Javascript et HTML/CSS dans notre cas), le débogage intégré, la complétion de code intelligente, la gestion des versions avec Git, et une personnalisation étendue grâce à des extensions. VS Code est apprécié pour son interface utilisateur intuitive, sa performance rapide et son support multi-plateforme, fonctionnant sous Windows, macOS et Linux. Nous l’avons très largement utilisé pour écrire, modifier et déboguer notre code dans le cadre de la partie développement web de ce projet.
* **Google Drive :** Un service de stockage et de partage de fichiers en ligne développé par Google. Il nous a permis de sauvegarder l’ensemble de nos fichiers et documents dans le cloud, d'y accéder depuis n'importe quel appareil connecté à Internet. En plus du stockage, Google Drive intègre également les applications de la suite Google Workspace, comme Google Docs, Sheets et Slides, permettant la création et l'édition en ligne de documents, de feuilles de calcul et de présentations. Ces caractéristiques nous ont permis de collaborer en temps réel sur les divers aspects du projet, facilitant ainsi notre travail d'équipe et notre productivité.
* **Google Colab :** Un service gratuit offert par Google qui permet aux utilisateurs d'écrire et d'exécuter du code Python via le navigateur. Il est particulièrement utile pour le *machine learning*, l'analyse de données et l'éducation. Google Colab fournit un environnement de notebook Jupyter hébergé dans le cloud, où nous avons pu écrire et exécuter du code, sauvegarder et partager nos analyses, et accéder à des ressources de calcul puissantes. Ce service nous a grandement facilité la collaboration en temps réel ainsi que le partage de projets. Il s'intègre facilement avec Google Drive pour le stockage des notebooks et des données. Google Colab est largement utilisé dans la communauté de la science des données pour son accessibilité et sa facilité d'utilisation.
* **MindView :** Uun logiciel de mind mapping et de gestion de projet conçu pour aider les utilisateurs à organiser, visualiser et présenter leurs idées de manière structurée et interactive. Il nous a permis de créer des cartes mentales, qui sont des représentations graphiques des idées et des concepts, facilitant ainsi la planification et l'organisation des pensées. MindView offre de nombreuses fonctionnalités telles que l'intégration avec Microsoft Office, des options de personnalisation des cartes, des outils de gestion de tâches et de suivi de projet, ainsi que des capacités de présentation et de collaboration en équipe. Il est couramment utilisé dans les domaines de l'éducation, des affaires et de la gestion de projet pour améliorer la productivité et la communication.

#### 1.2.3. Diagrammes et graphes

###### Diagramme bête à cornes :

Le diagramme bête à cornes est un outil de conception qui identifie l'utilisateur principal d'un produit ou service et ses besoins principaux, aidant à orienter le développement vers ce qui est vraiment nécessaire. De par sa simplicité, il peut aussi servir d’outil de présentation d’un projet.

Une image contenant texte, Police, capture d’écran, diagramme

Description générée automatiquement

*Figure X : Diagramme bête à corne de notre projet*

###### Mind map :

Une mind map est un outil visuel pour organiser et représenter des idées. Elle commence par une idée centrale et se déploie en branches et sous-branches, intégrant mots-clés, images, et couleurs pour faciliter la compréhension et la mémorisation. C'est un moyen efficace pour brainstormer, planifier, et structurer des pensées de manière créative et intuitive.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

*Figure X : Mind map de notre projet*

###### Diagramme de Gantt :

Un diagramme de Gantt est un outil de gestion de projet qui représente visuellement le planning d'un projet. Il affiche les tâches à réaliser sur un axe horizontal représentant le temps, et montre leur durée, leur chevauchement, et leur séquence. Cela permet de visualiser facilement le calendrier du projet et l'avancement des différentes tâches.

Une image contenant capture d’écran, texte, ligne, nombre

Description générée automatiquement

*Figure X : Diagramme de Gantt de notre projet (voir annexe 1)*

Cependant, bien que le diagramme de Gantt soit un excellent outil pour faciliter l’organisation et la visualisation des tâches à réaliser, il est à noter qu’il s’applique mieux aux projets où les membres sont impliqués à 100% de leurs capacités. Dans notre cas, nous avions en parallèle nos cours ainsi que d’autres projets, ce qui explique certaines durées de tâches soit aussi anormalement longues.

### 1.4. Équipe de projet

#### 1.4.1. CV des membres

Une image contenant texte, capture d’écran, Site web

Description générée automatiquement

Une image contenant texte, capture d’écran, Visage humain, document

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement

#### 1.4.2. Rôles & motivations personnelles

Désignation des rôles, assorti de domaines de compétence sur lesquels chaque membre opérera.

**Fabien GAUTIER**

###### Motivation personnelle :

Ayant déjà réalisé des projets impliquant du machine learning ou du développement Web, je voulais pouvoir en réaliser un en approfondissant ces domaines. Nous avons alors convenu avec Odyssé et Kléden de travailler sur un projet liant ceux-ci dans un contexte impliquant des thématiques d’avenir comme la robotique. Pouvoir réaliser un projet impliquant des connaissances développées en cours et développer des compétences en réalisant un projet est vraiment intéressant dans le cadre de nos études.

###### Rôles au sein du projet :

Mon rôle au sein de ce projet a été de développer l’application Web et de trouver des solutions afin de développer les fonctionnalités du projet et de les afficher à l’utilisateur. J’ai eu un rôle assez polyvalent dans la réalisation de ce projet me permettant d’agir sur la plupart des compartiments de l’application. J’ai aussi été chargé de développer les solutions autour des modèles 3D et d’implémenter les différents éléments développés dans l’application. J’ai aussi réalisé un travail important de recherche et de documentation pour mettre en place des solutions principalement pour la génération de modèles 3D.

**Odyssé LOIZEAU**

###### Motivation personnelle :

L’idée du projet est venue parallèlement avec la découverte et l’utilisation des IA génératives et notamment DALL-E et ChatGPT. à partir du langage humain, et sans compétences particulières, de pouvoir créer quelque chose de plus complexe comme du code ou une image a démontré le début d’une nouvelle méthode de travail, ce qui allait être mis en avant était moins la compétence et plus la créativité et la capacité à innover. Partant de ce principe, et en discutant avec des acteurs importants du domaine durant mon stage (premier semestre 2023), notamment Marc Carrel-Billiard (Global Lead Innovation chez Accenture), trois technologies sont ressorties comme étant aux prémices d’un développement exponentiel. L’intelligence artificielle générative multimodale (ce qu’est devenu aujourd’hui ChatGPT-4 ou Google Gemini), les jumeaux numériques complexes (corps humains, villes, usine complète) et la robotique au sens large. Il semblait donc intéressant de faire un projet mêlant ces trois domaines dans lesquels nous serions amenés à travailler, et voir comment nous pouvions prendre en compétence tout en essayant de produire un POC d’un projet qui pourrait s’avérer utile. Nous avons eu la confirmation que ce projet avait du sens après avoir pu échanger avec des acteurs de chez Nvidia (Sandra Skaff) ayant admis travailler sur un projet similaire, mais bloquants actuellement sur des problématiques que nous avons d’ailleurs également rencontrées.

###### Rôles au sein du projet :

D’un point de vue organisationnel, j’ai eut le rôle d’un SCRUM master, ayant pour but de faire comprendre la méthodologie Agile avec le framework SCRUM, reposant sur le travail en autonomie des membres techniques tout en faisant remonter les problématiques rencontrées durant le travail de façon régulière, afin de faire rapidement remonter les points bloquants, tout en produisant un produit fonctionnel à chaque étape.

D’un point de vue opérationnel, j’ai travaillé en premier lieu sur la veille technologique de la génération d’objets 3D, puis je suis ensuite passé sur la création/récupération de données avant de basculer sur la partie création/affinage du modèle permettant la sélection des composants à partir du texte de description du robot. Passer par l’ensemble de ces domaines aura permis de prendre en compétence sur chacun de ces domaines, en adéquation avec le framework SCRUM, dont le but est d’avoir une équipe complète et prenant en compétence collectivement sur l’ensemble des technologies du projet.

**Kléden ROUSSEAU**

###### Motivation personnelle :

Dès le début, ce projet a capté mon intérêt car j'y ai immédiatement vu une très bonne occasion d'explorer et d'approfondir mes connaissances dans des domaines techniques encore relativement nouveaux pour moi. Ce qui m'a particulièrement attiré, c'est son caractère pluridisciplinaire, mélangeant à la fois la robotique, l'intelligence artificielle et le développement web. Ces domaines, à la fois distincts et complémentaires, offrent un terrain de jeu fertile pour l'innovation et le développement de nouvelles technologies.

En outre, la pertinence du sujet est renforcée par sa capacité à s'accorder aux dernières avancées et découvertes dans ces champs respectifs. Cela représente une occasion unique non seulement d'acquérir des compétences techniques, mais aussi de se mettre à jour au sujet des initiatives de recherche et de développement à la pointe des évolutions technologiques actuelles.

###### Rôles au sein du projet :

D’un point de vue agile, le rôle que j’ai eu tend à se rapprocher d’une sorte de Product Owner, j’ai été chargé de définir la vision du produit en comprenant les besoins des utilisateurs et les formaliser pour le développement. Mon rôle implique de prioriser les fonctionnalités pour maximiser la valeur et l'impact du produit. J’ai beaucoup collaboré avec Fabien sur les aspects d’ergonomie du site web et d’expérience utilisateur. Ma mission principale a été de guider l'équipe vers la création d'un produit qui répond aux exigences de notre cahier des charges.

D’un point de vue plus concret, mon rôle a d'abord consisté à actualiser nos connaissances dans plusieurs des domaines essentiels de ce projet, notamment en intelligence artificielle, en représentation 3D de robots, et en développement web. Cette phase initiale était cruciale pour aligner notre équipe avec les dernières avancées technologiques. Par la suite, j'ai eu un rôle d'assistance technique, en apportant mon soutien à Fabien sur le développement web, et à Odyssé pour approfondir des points d’intelligence artificielle, avec un accent particulier sur le machine learning. Ce rôle m'a permis de contribuer activement à différents aspects du projet tout en renforçant la synergie au sein de notre équipe. J'ai également eu un rôle important dans le cadrage et la rédaction de ce rapport de projet.

## 2. Création de l’application web

### 2.1. Présentation du framework Django

Afin de fournir à l’utilisateur un moyen simple d’utilisation et accessible, nous avons choisi de développer un site web où nous présentons notre application d’aide à la conception robotique. Il nous fallait alors un moyen de mettre en relation une partie de front-end (ce que l’utilisateur voit), et une partie back-end (le fonctionnement caché du site web). Nous avons alors choisi d’utiliser le framework Django. Cet outil est utilisé avec Python et permet une approche assez instinctive du développement web permettant d’inclure des éléments de front-ends en JavaScript, HTML et CSS, ainsi qu’un back-end en langage Python.

Django suit une architecture Modèle-vue-contrôleur qui est un style d’architecture très utilisé pour développer des interfaces graphiques. De nombreuses grandes entreprises utilisent Django, c’est notamment le cas d’Instagram, Spotify ou encore Pinterest. Le motif MVC comporte trois compartiments majeurs :

Une image contenant capture d’écran, texte

Description générée automatiquement

Figure : Fonctionnement de l’architecture Modèle Vue Contrôleur

* ***Le modèle :*** Celui-ci contient la structure des données, il fait le lien entre les bases de données utilisées et l’affichage de celles-ci.
* ***La vue :*** Cet élément reçoit les données du modèle et est chargé de l’affichage, la manipulation et la récupération de celles-ci sur l’interface utilisateur. La vue traite les requêtes de l’utilisateur et lui fournit l’affichage correspondant.
* ***Le contrôleur :*** Dans Django, on utilise un système de templates pour la couche de présentation, similaire à la Vue dans d'autres implémentations du modèle MVC. Les templates définissent comment afficher les données récupérées par la vue.

### 2.2. Mise en place

Tout d’abord, pour utiliser Django, il faut avoir une version de Python installée sur son environnement de travail. On peut ensuite installer la bibliothèque django comme suit :

|  |
| --- |
| pip install django |

On initialise ensuite le site web avec la commande suivante :

|  |
| --- |
| django-admin startproject mysite |

Cette commande permet de créer l’architecture django comme suit :

Une image contenant texte, diagramme, capture d’écran, conception

Description générée automatiquement

Figure : Initialisation de l'application Web

* ***manage.py :*** C'est un fichier python prenant en argument des de commandes permettant de faire des actions importantes sur le serveur et le site web. On peut utiliser celui-ci pour par exemple, la création de migrations, l'exécution de tests, ou encore le démarrage du serveur de développement.
* ***init.py :*** Ce fichier est présent dans chaque répertoire Python pour indiquer que le répertoire doit être traité comme un package.
* ***asgi.py*** *(ASGI : Asynchronous Server Gateway Interface)* ***:*** Il s'agit d'un point d'entrée pour les serveurs d'application asynchrones. Il est utilisé pour déployer des applications Django avec des serveurs prenant en charge l'asynchronisme, ce qui permet une gestion plus efficace des connexions simultanées.
* ***settings.py :*** Ce fichier contient les paramètres de configuration de votre application Django. Des variables telles que les clés secrètes, les paramètres de base de données, les applications installées, les paramètres de langue y sont définies
* ***urls.py :*** Ce fichier définit les URL de l'application.Il permet d’organiser la structure de l'application et à gérer la navigation.
* ***wsgi.py*** *(WSGI : Web Server Gateway Interface)* ***:*** Il fournit un point d'entrée pour les serveurs web compatibles avec WSGI.

On peut alors ensuite étoffer cette architecture en créant une application :

|  |
| --- |
| python manage.py startapp projectApp |

On obtient alors :

Une image contenant diagramme, capture d’écran, Plan, ligne

Description générée automatiquement

Figure : Architecture de l'application Web

* ***admin.py :*** Ce fichier est utilisé pour configurer l'interface d'administration de Django.
* ***apps.py :*** Il permet de configurer des métadonnées pour l’application, telles que le nom convivial ou la configuration des signaux.
* models.py : C'est l'endroit où sont définis les modèles de données pour votre application.
* ***tests.py :*** Les tests unitaires sont décrits dans ce fichier. Les tests sont essentiels pour assurer le bon fonctionnement de l’application et détecter d'éventuels problèmes.
* ***views.py :*** Ce fichier contient les vues de l’application. Les vues sont ici des fonctions ou des classes Python qui traitent les requêtes HTTP et renvoient des réponses. Elles gèrent les interactions entre les modèles et les templates.
* ***urls.py :*** Comme précédemment, ce fichier définit les URL spécifiques à l’application.
* ***migrations :*** Ce répertoire est créé lorsque des commandes de gestion des migrations sont exécutées. Il contient des fichiers qui représentent les changements de schéma de base de données au fil du temps. Les migrations permettent de mettre à jour la base de données de manière cohérente avec l'évolution du modèle de données.

Ce que nous avons pu voir ici correspond donc à l’architecture de base d’un site web développé avec le framework django. Il est ajouté à l’application de nombreuses ressources et répertoires comme les pages HTML et CSS, les codes Javascripts, les fichiers 3D à télécharger par l’utilisateur et les codes de Back-end.

En effet, une fois l’application créée, on peut commencer à éditer le site web à l’aide de pages HTML, fichiers CSS et codes JavaScripts dans un répertoire qui leur est propre. Lors de la navigation d'une personne sur internet, des codes HTML, CSS, et parfois Javascript sont envoyés à son navigateur lors du chargement de la page web qu'elle choisit de consulter. Ces langages de programmation permettent de structurer (HTML), de styliser (CSS), et d'ajouter des fonctionnalités interactives (Javascript) à la page web. Les actions de l'utilisateur sur cette page sont capturées, permettant de déclencher l'envoi et l'affichage des codes adaptés en fonction du fonctionnement spécifique du site.

Chaque page Web constitue une vue qui sert d’interface entre le back-end et l’utilisateur. Lorsque l’utilisateur interagit avec une page, il faut donc pouvoir récupérer les actions de celui-ci et lui proposer les réponses adaptées à ses actions.

### 2.3. Présentation de l’application

Dans le cas de cette application, la page par défaut est composée du formulaire à remplir afin de décrire le projet robotique à réaliser. Nous avons choisi de décomposer les choix de robot en trois parties : l’inspection, le transport et la manipulation. On exploite les données saisies dans les champs “inspection” et “manipulation” avec un algorithme d’intelligence artificielle pour fournir des résultats correspondant aux besoins de l’utilisateur. Voici la page principale de l’application :

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

Figure : Page principale de l'application Web

L’utilisateur rentre dans la partie inspection un texte qui décrit ce que le robot effectuera en tâches d’inspection (par exemple : robot inspectant des canalisations). Dans la partie transport on retrouve un menu déroulant adaptatif qui permet à l’utilisateur de saisir quel type de mobilité le robot aura besoin d’avoir. Dans la partie Manipulation, l’utilisateur décrit les manipulations que le robot doit effectuer (par exemple : saisie et collecte d’ordures dans un environnement urbain).

Lorsque le formulaire est soumis, une page “confirmation” est chargée avec les résultats obtenus pour l’aide à la conception du robot décrit par l’utilisateur. Les textes entrés dans celui-ci sont enregistrés puis passés à travers des algorithmes d’intelligence artificielle. Les choix de l'utilisateur de la section transport permettent de faire une requête API pour générer une pièce 3D à proposer. Le résultat obtenu est ensuite transmis à la vue de confirmation, et ceux-ci sont affichés.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure : Page de résultats

On propose tout d’abord des recommandations sur les capteurs puis les actionneurs en fonction du besoin de l’utilisateur. Celui-ci peut également télécharger une pièce 3D proposée selon les critères qu’il a pu renseigner. Le modèle proposéi correspond à l’actionneur faisant avancer le robot choisi.

## Modélisation 3D

La génération d’un modèle 3D à partir d’un texte est une tâche très complexe surtout lorsque l’on doit être précis sur celles-ci. La complexité d’un fichier 3D et le manque de données ou dataset sur ce sujet ajoutent de la difficulté à l’algorithme d’intelligence artificielle à réaliser pour créer un modèle 3D à partir d’un texte. Nous avons alors réalisé un grand travail de documentation pour pouvoir trouver un moyen répondant au mieux à notre cahier des charges. C’est une fonctionnalité assez innovante sur laquelle on ne trouve pas beaucoup d’informations du fait de sa complexité et des moyens nécessaires pour mettre en place un tel système. De nombreuses entreprises du secteur de l’industrie développent actuellement des outils similaires mais ne proposent pour la plupart, pas d’application en Open Source ou d’outil assez performant pour être utilisé avec un usage professionnel.

Nous avons alors commencé par analyser l’application Onshape permettant de réaliser une pièce 3D avec un programme Python en utilisant une API. La pièce générée par Onshape permettait alors d’obtenir un fichier en format URDF facilement utilisable dans un environnement tel que ROS, et visualisable avec Gazebo. Cette piste-ci était alors très adaptée à notre cahier des charges orienté robotique pour la modélisation 3D. Cependant, la complexité de pouvoir générer du texte pour réaliser un code suffisamment précis et performant pour réaliser la pièce voulue a été un frein pour notre utilisation d’Onshape.

Nous avons ensuite orienté nos recherches vers des outils comme RobotDK proposant un grand nombre de modèles de robots industriels 3D simulables, avec une API et de nombreux usages possibles. Les robots n’étant pas en lien avec la direction que nous voulions pour notre projet ainsi que les modèles proposés nous ont finalement poussés à chercher dans une autre direction.

Après nous être documentés sur de nombreuses pistes et solutions notamment la plateforme HuggingFace d’intelligence Artificielle, nous sommes finalement rendus compte que la génération d’un modèle 3D à partir d’un texte était une tâche très complexe et qu’il fallait résoudre notre problématique différemment.

Nous avons alors utilisé l’API kittycad avec la plateforme [zoo.dev](https://zoo.dev/) pour pouvoir générer des pièces 3D par rapport aux besoins utilisateurs remplis dans l’application Web. Le projet développé par cette entreprise permet de générer des modèles 3D dans différents formats (STL, GLTF, STEP,...) à partir de saisie textuelle et propose différentes fonctionnalités liées à celle-ci :

Une image contenant capture d’écran

Description générée automatiquement

Figure : Exemple de modèle 3D généré sur Zoo.dev

Pour utiliser l’API, on récupère un token, pour créer un client et s’identifier. On peut ensuite envoyer des requêtes avec le prompt rentré par l’utilisateur et lui fournir une pièce à télécharger en sortie. L’objectif est de fournir un modèle pour le châssis et un pour l’actionneur de déplacement (roues, hélices, chenilles etc). Le principal inconvénient de cette méthode est de récupérer ce que l’utilisateur remplit sur la page et de générer la pièce ensuite. Un temps d’attente variable et parfois excédant une dizaine de secondes est alors à envisager lors de l’utilisation de l’application Web. Nous avons choisi de ne fournir qu'une seule pièce de à l’utilisateur afin de réduire ce temps d’attente et améliorer la fiabilité des pièces fournies.

Nous avons alors pu obtenir un résultat correct sur la génération des pièces grâce à ce système avec des points très satisfaisant et d’autres perspectives d’améliorations.

Pour commencer, la taille que l’utilisateur veut pour l’actionneur du robot est très bien prise en charge, nous obtenons de très bons résultats à ce sujet. En revanche, la génération de pièces 3D à partir d’un texte étant une tâche très complexe, les pièces générées sont souvent inutilisables ou irréalistes bien que la forme et l’idée transmise par l’utilisateur soit respectée:

Une image contenant texte, capture d’écran, cercle, conception

Description générée automatiquement

Figure : Roue de 8cm de diamètre générée avec l'application Web

On peut expliquer cela par le fait que l’API d’une application est rarement d’un aussi bon niveau que ce qui est utilisable sur cette même application pour éviter tout souci lié à la concurrence. Les modèles générés ne sont également pas toujours valides et résultent parfois d’erreurs de fichiers lorsque la demande de l’utilisateur n’est pas gérée par l’outil de *Zoo.dev*. Cette technologie étant de plus très récente, les résultats ne sont pas encore satisfaisants mais nous sommes convaincus que cela s’améliorera sûrement dans le futur.

## 3. Intelligence Artificielle

### 3.1. Création et équilibrage du dataset

Les données d'entraînement sont l’un des aspects les plus essentiels de l’apprentissage supervisé (machine learning). Pour entraîner notre algorithme sur une bonne base, il est primordial de lui fournir un dataset de qualité. Ces données doivent être fiables et les plus représentatives possible des situations réelles auxquelles sera confronté notre programme.

L’idée était de trouver un dataset contenant des phrases descriptives de robots, chacune associée à un ou plusieurs composants (capteur, actionneur, élément de structure) constitutifs de ce dernier. Sur internet, il existe un grand nombre de sites webs proposant des datasets libres de droits, parmi les plus connus, on retrouve Kaggle, Hugging Face, Google Dataset Search ou encore GitHub. Ces sites proposent un grand nombre de ressources en lien avec la robotique, pour faire de l’optimisation de navigation ou de l’automatisation. Cependant, pour ce qui concerne des descriptions détaillées de robots, nos pistes se sont assez vite avérées inefficaces. Du fait du caractère trop spécifique de nos recherches, nous n’avons pas réussi a trouvé de dataset exploitable.

Néanmoins, il est intéressant de constater qu'actuellement, le manque de dataset fiable et exploitable est une problématique sérieuse qui ralentit considérablement les développements d’applications, notamment dans le secteur de la robotique. Raison pour laquelle la création de données synthétiques (Synthetics Data) tend à se développer de plus en plus.

C’est pourquoi, pour pallier ce manque, nous avons décidé de construire notre propre dataset. L’objectif était d’écrire nous-mêmes les phrases descriptives et d’y associer, à chaque fois, les composants adéquats. Tâche laborieuse s’il en est une, cette méthode présente néanmoins l’avantage d’être sur mesure. Autrement dit, nous équilibrons les données qui composent le dataset comme nous le souhaitons.

Dans un premier temps, nous avons décidé de prendre en considération un cas simple et un nombre limité de capteur/actionneur. Ce premier dataset sera composé de phrases descriptives de robots spécialisés dans l’inspection de tout type d’environnements. Que ce soit de l’inspection de canalisation, à l’exploration sous-marine et spatiale, en passant par l’inspection de structures et de bâtiments ou encore de zones sinistrés, nous avons essayé d’être les plus exhaustifs possible de sorte à ce que notre programme soit apte à fournir une réponse cohérente peu importe le cas de figure. Pour que les performances du modèle soient bonnes, le dataset d’apprentissage devra être composé d’un nombre suffisant de datas. Aussi, il est important d’avoir des variations sur les structures de texte pour que le modèle soit préparé à tous les types de descriptions.

L’idée de base était de fournir chacun 150 descriptions de robots associés à des capteurs. Pour accélérer le processus de génération de phrases, nous nous sommes appuyés sur l’outil ChatGPT. En entrée, nous lui donnions une simple phrase et nous lui demandions de nous retourner un certain nombre de variantes de cette dernière, en changeant l’ordre des mots, en utilisant des synonymes, etc. Ce qui permettait d'accroître considérablement le nombre de descriptions.

*Figure X : illustration de l’utilisation de chatGPT pour la génération des phrases*Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Ensuite, chaque phrase fut associée à un ou plusieurs composants adaptés à la réalisation de la mission du robot. La liste des capteurs/actionneurs disponibles est la suivante : Caméra RGB, Caméra Infrarouge, Caméra Thermique, Caméra Profondeur, Capteur Ultrason, LED et Laser télémètre.

Les composants sont associés à chaque description de robot sous forme de matrice 1x7 colonnes, composées uniquement de 0 (absence du composants) ou de 1 (présence du composants). Par exemple, un robot ayant pour mission de détecter des sources chaudes sera généralement accompagné d’une caméra thermique. Dans le même sens, un robot d’inspection de structure pourra être équipé d’une caméra RGB ainsi que d’un capteur à ultrason.Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Aussi, de sorte à simplifier le processus d’ajout des datas, nous avons créé un très simple programme python. Ce dernier est composé d’un encart permettant de recueillir le texte et de la liste des composants associés aux cases à cocher, qui générera un fichier .csv que l’on utilisera ensuite pour entraîner notre modèle.

*Figure X : vue de l’interface de création des données*

Une image contenant texte, capture d’écran, nombre, ligne

Description générée automatiquement

*Figure X : extrait des données du fichier .csv*

Suite au traitement initial des données, incluant l'élimination des éléments redondants et des cas susceptibles de fausser le modèle, notre ensemble de données s'est réduit à 350 exemples. En comparaison, les ensembles de données les plus restreints disponibles sur HuggingFace contiennent habituellement au moins 20 000 entrées, réparties entre les phases d'entraînement, de validation et de test. Notre défi consistait donc à produire un travail de qualité malgré la limitation quantitative de nos données.

### 3.2. Machine Learning

#### 3.2.1. Différentes méthodes

Le but de notre application est de fournir une aide à la conception de robots. Cette aide se présente, dans un premier temps, comme une liste de capteurs adaptés à la mission du robot à concevoir. Autrement dit : En entrée, l’application acceptera un texte descriptif plus ou moins précis d’un robot à concevoir. En sortie, il fournira une liste de capteurs pertinents, selon les spécificités du robot considéré. Entre ces deux phases, c’est justement là qu’intervient notre modèle d’intelligence artificielle.

Un modèle en machine learning est un système conçu pour apprendre à partir de données variées (des textes, des images, des chiffres). Ce processus commence par la collecte et le prétraitement des données pour les rendre exploitables, précisément ce que nous avons fait lors de la phase de création et d’équilibrage du dataset. Ensuite, le modèle entame sa phase d’apprentissage, étape cruciale puisqu’elle consiste en l’ajustement de ses paramètres internes pour renvoyer des résultats cohérents, démarche qui dans notre cas sera supervisée. Après ça, le modèle est évalué pour vérifier sa performance, notamment en le testant sur des données nouvelles ou réservées. Finalement, une fois formé et validé, notre modèle pourra être utilisé pour réaliser la tâche pour laquelle il a été conçu avec efficacité et précision.

Il existe de nombreux types de modèles, chacun présentant des caractéristiques et propriétés spécifiques. De sorte à déterminer le modèle le plus pertinent dans notre cas de figure, nous en avons testé plusieurs. Les deux premiers types sont des modèles basés principalement sur des algorithmes, tandis que les deux derniers sont basés sur des réseaux de neurones.

###### Méthode “K nearest neighbors (KNN)” :

Le KNN est une méthode simple mais puissante en apprentissage automatique, utilisée pour la classification et la régression. Le principe de base est de prédire la catégorie (en classification) ou la valeur (en régression) d'un nouvel élément en se basant sur les données les plus similaires déjà présentes dans le jeu de données.

Le paramètre 'k' représente le nombre de voisins les plus proches à prendre en compte. Pour déterminer ces voisins, KNN utilise une mesure de distance, souvent la distance euclidienne, pour calculer la proximité entre les éléments. En mode classification, l'étiquette finale attribuée à un nouvel élément est déterminée par un vote majoritaire parmi les k voisins, tandis qu'en régression, c'est la moyenne de leurs valeurs qui est utilisée. Un aspect clé de KNN est sa sensibilité à la mise à l'échelle des caractéristiques, car les distances sont directement affectées par l'échelle des données. Bien que KNN soit simple à comprendre et à mettre en œuvre, il peut être moins efficace pour les grands ensembles de données ou ceux avec un grand nombre de caractéristiques, en raison de la complexité croissante du calcul des distances.

Dans l'application pratique, ce modèle a affiché des performances décevantes, principalement dues à sa capacité limitée à offrir une seule classification en sortie. Or, dans notre situation, il était souvent nécessaire de gérer plusieurs capteurs, donc plusieurs sorties étaient requises. Cette limitation, combinée à un volume restreint de données, a rendu le modèle peu adapté et peu fiable pour notre application spécifique.

###### Méthode “Random Forest” :

La méthode Random Forest est un algorithme populaire en machine learning qui combine plusieurs arbres de décision pour améliorer la fiabilité et l'exactitude des prédictions. Chaque arbre est construit en utilisant un échantillon aléatoire des données et des caractéristiques, ce qui contribue à réduire le risque de surajustement des arbres de décision individuels.

Lors de la prédiction, chaque arbre donne son verdict et la prédiction finale est déterminée par un vote majoritaire en classification, ou par la moyenne en régression. Les avantages de Random Forest incluent sa robustesse face aux grands ensembles de données et sa capacité à traiter des caractéristiques complexes, tout en gérant bien les données manquantes. Cependant, il peut être moins interprétable qu'un arbre de décision unique et peut nécessiter plus de temps de calcul pour de très grands jeux de données.

En pratique, les résultats obtenus étaient légèrement meilleurs, mais ils comportaient encore une part d'aléatoire. Ce modèle souffrait du même problème que celui des k plus proches voisins : il ne pouvait fournir qu'une seule sortie. Ainsi, lorsqu'il était nécessaire d'utiliser deux capteurs, le modèle ne faisait pas toujours le meilleur choix. De plus, dans les situations où l'utilisation des deux capteurs était requise, le modèle commettait une erreur.

Nous nous sommes ensuite essayés aux réseaux de neurones (Neural Network). Les réseaux de neurones sont des modèles informatiques inspirés du fonctionnement du cerveau humain, utilisés en machine learning pour reconnaître des modèles complexes dans les données. Ils se composent de couches de "neurones" artificiels qui reçoivent des entrées, les traitent à travers des fonctions mathématiques, et transmettent les sorties à d'autres neurones. Chaque connexion entre les neurones a un "poids" qui s'ajuste au cours de l'apprentissage, permettant au réseau d'apprendre et de faire des prédictions ou classifications précises à partir de données d'entrées variées. Ces réseaux sont particulièrement efficaces pour des tâches comme la reconnaissance d'images, la traduction automatique, ou la détection de tendances dans de grandes quantités de données.

###### Méthode “Neural Network (NN) from scratch” :

Le principe de construction d'un réseau de neurones "from scratch" implique de créer un réseau de neurones sans utiliser de bibliothèques d'apprentissage automatique prédéfinies, afin de comprendre en profondeur son fonctionnement. Cela commence par l'initialisation des poids et des biais de manière aléatoire. Ensuite, on définit les couches du réseau, souvent composées de neurones qui effectuent des calculs basés sur leurs entrées, poids, et une fonction d'activation.

Le processus d'apprentissage se fait via l'algorithme de rétropropagation, où l'on ajuste les poids et les biais en fonction de l'erreur calculée entre la sortie prédite et la sortie réelle, souvent à l'aide de la descente de gradient. Ce processus itératif améliore progressivement la précision du modèle. Construire un NN from scratch est particulièrement intéressant lorsqu’il s’agit de comprendre les mécanismes internes des réseaux de neurones, mais c'est une tâche souvent plus complexe et qui, parfois, peut s’avérer moins efficace que d’utiliser des bibliothèques spécialisées.

Nous avons donc créé notre propre modèle basé sur notre ensemble de données. Il s'agit d'un classificateur de texte multi-étiquettes (Multi Label Text Classifier), ce qui signifie qu'il prend un texte en entrée et associe à chaque mot un "token" (jeton), généralement sous forme d'un nombre entier. Cette numérisation permet à l'ordinateur de convertir la séquence de nombres en vecteurs, qui sont ensuite associés aux labels (capteurs) correspondants. Le modèle expérimente diverses configurations de neurones pour affiner au mieux sa réponse en fonction de l'entrée, un processus connu sous le nom d'entraînement.

Ce modèle s'est distingué en étant le premier à nous fournir des résultats satisfaisants, grâce notamment à l'ajustement du "threshold" (seuil de confiance). Le threshold est un paramètre clé dans le fonctionnement des modèles de machine learning. Il représente la limite de confiance nécessaire pour qu'une réponse soit considérée comme valide par le modèle. Cette valeur varie de 0 à 1, où un threshold de 0 signifie que le modèle affichera toutes les réponses possibles, tandis qu'un threshold de 1 indiquera que seules les réponses pour lesquelles le modèle a une certitude absolue de leur exactitude seront présentées.

Une caractéristique intéressante de ce modèle est que plus il est bien entraîné, et plus le jeu de données sur lequel il est basé est robuste et de qualité, plus il est possible de fixer un threshold élevé. Dans notre situation, le threshold était réglé autour de 0,25. Cette valeur est notable compte tenu du fait que le modèle a été entraîné sur un ensemble relativement restreint de 350 données. Ce niveau de threshold indique un équilibre entre prudence et efficacité, permettant à notre modèle de fournir des réponses fiables sans être trop restrictif dans ses prédictions. Cette configuration nous a permis de maximiser l'utilité du modèle tout en maintenant un niveau de confiance équilibré dans les résultats qu'il produisait.

###### Fine-Tuning d’un LLM:

Des suites de notre rencontre avec un doctorant spécialisé en Intelligence Artificielle, présenté par Mme Saddem, il nous a été suggéré de faire le fine-tuning d'un Large Language Model (LLM).

Le fine-tuning est une technique qui consiste à modifier et adapter un modèle d'IA préexistant pour qu'il réponde à nos besoins spécifiques. L'un des principaux avantages de cette approche est la possibilité de tirer parti de modèles déjà très performants pour des tâches spécifiques, auxquels nous n'aurions normalement pas accès. Cependant, l'inconvénient réside dans le besoin de ressources informatiques conséquentes, et le fait que le modèle est déjà spécialisé pour une certaine tâche.

Dans notre cas, étant donné que l'analyse de texte était un élément clé, l'utilisation d'un LLM nous a été recommandée. Ces modèles sont particulièrement adaptés pour analyser des textes complexes et peuvent produire en sortie d'autres textes, dont la complexité peut varier en fonction des paramètres que nous définissons. En somme, cette approche nous offrait la possibilité d'exploiter des capacités avancées d'analyse de texte tout en adaptant le modèle à notre contexte spécifique.

Pour correctement réaliser le fine-tuning, un prétraitement des données particulièrement méticuleux a été requis. Nous devions obtenir un ensemble de données sans fautes de frappe et formaté en textes complets, plutôt qu'en simples labels. Ce processus rigoureux de vérification et de correction a assuré une qualité et une fiabilité élevées des données, essentielles pour la précision et l'efficacité du modèle.

Nous avons utilisé le modèle Falcon 7B, très compétent dans la compréhension du français et reconnu pour sa capacité à produire des résultats de qualité. Bien que les résultats étaient intéressants, ils se sont révélés inadaptés à notre situation spécifique. En effet, au lieu de générer une liste de capteurs requise, le modèle produisait un texte explicatif détaillant les contextes d'utilisation possibles du robot. Cette fonctionnalité peut effectivement être intéressante, mais ne répondait pas à nos besoins spécifiques.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

###### e) supplément : Fine-Tuning d’un NLP en text classification Multi-Labels (Bert):

Nous n’avons pas eu le temps d’explorer cette méthode. Elle consiste à faire un mixe entre l’utilisation d’un LLM et celle d’un text classifier. Le principe repose sur la “compréhension” du texte par la machine puis ensuite la classification de ce texte dans une catégorie. Une des utilisations les plus répandues actuellement est la classification des commentaires sur les réseaux sociaux afin d’en déterminer l’émotion perçue (bienveillance, haine, colère, remerciement, encouragement,etc). La méthode classe le commentaire dans une catégorie ce qui permet par la suite d’effectuer une modération ou de mettre en avant le commentaire selon la nature de celui-ci.

D’abord, il nous a fallu dans un premier temps trouver un modèle multi-label. C’est-à-dire, un modèle capable d’associer un texte à potentiellement plusieurs catégories plutôt qu’à simplement une seule catégorie. Lorsque ce modèle fut trouvé, la quantité de données nécessaires à l'entraînement de ce modèle était trop faible (généralement plusieurs milliers de lignes uniquement pour l'entraînement). La puissance de calcul nécessaire et de RAM est particulièrement importante afin de mettre en mémoire le modèle.

Cependant, il est fort probable que ce modèle nous aurait apporté des résultats de très bon niveau de par le fait qu’il soit pré-entraîné sur un très large éventail de textes. Toutefois, le risque aurait été d’avoir eu un entraînement sur des données totalement extérieures au monde de la robotique et donc d’avoir un effet de sur-entraînement durant la phase de fine-tuning.

## 4. Limites du projet

### 4.1. Améliorations et ajouts

Par la nature de ce projet, nous n’avons pas forcément pu réaliser tout ce que nous souhaitions comme nous le souhaitons. Effectivement, l’application web est encore très largement perfectionnable, notamment les fonctionnalités déjà implémentées. Trois axes importants peuvent être améliorés de façon certaine :

1. La recommandation et sélection des composants nécessaires au robot. Dans notre projet, le modèle d'entraînement pour les capteurs ou actionneurs, propose sept éléments électroniques correspondants à la demande de l’utilisateur. Il serait possible de creuser plus en profondeur des cas pratiques en faisant une veille technologique auprès de multiples acteurs de la robotique afin d’affiner au maximum quels capteurs/actionneurs à utiliser et dans quel cas, allant même jusqu’à la sélection de la méthode de mise en mouvement.
2. Les modèles 3D fournis par l'application sont des fichiers .step répandus en CAD mais pas forcément en robotique, là où le format URDF est généralement préféré. Actuellement, les modèles de génération d’objets 3D restent extrêmement spécialisés ou peu performants (à l’image de Shap-E d’OpenAI pourtant leader dans les domaines de l’IA générative). La technologie n’est pas encore mature pour la 3D, mais le deviendra avec son lot d'applications qui iront avec.
3. La possibilité de générer du code pour tester le comportement du robot en simulation. Le but ultime serait de pouvoir mettre en relation le robot et ses composants avec un software afin de le faire se tester et s’améliorer par lui-même. On voit aujourd’hui qu’il est possible avec ChatGPT-4 ou Copilote et faire des projets de plus en plus complexe presque sans intervention humaine, il y a donc fort à parier que ces technologies vont se perfectionner dans les prochains mois/années, jusqu’à atteindre un niveau de développement très pointu, la limite étant, pour l’heure, la quantité de programmes disponibles pour l'entraînement de ces modèles d’IA.

### 4.2. Difficultés

Au cours du projet, nous avons dû faire face à diverses difficultés, prévisibles ou non, qui ont matériellement affecté notre progression :

* **Absence d’un dataset adapté :** L’une des premières difficultés que nous avons rencontrée est sans aucun doute celle du manque de données fiables. En intelligence artificielle et a fortiori en machine learning, les données sont essentielles, c’est le paramètre premier qui déterminera la qualité du modèle en sortie. Or, comme expliqué précédemment, nos exigences sur les données étant trop spécifiques, nous avons été contraints de créer notre propre dataset (synthetic) avec les défauts que cela charrie (nombre réduit de données, biais sur les cas pratiques). C’est le problème le plus récurrent en apprentissage automatique. Certains grands acteurs du milieu travaillant sur des problématiques similaires à la nôtre ont aussi été confrontés à ce type de problème, c’est notamment le cas de Nvidia pour leur plugin Isaac SIM.
* **Production de code par l’IA :** Initialement, nous avions pour ambitions d’essayer de produire du code par le biais d’une IA, mais cela s’est vite avéré bien plus complexe que ce que nous avions préalablement estimé. En effet, les subtilités et exigences des langages permettant la description de robots virtuels rendent la tâche particulièrement ardue. Pour être en mesure de construire un modèle capable de produire un tel résultat, il nous aurait fallu un certain nombre de prérequis que nous n’avions tout simplement pas (connaissances avancées du langage, large dataset spécifiquement adapté, grosse capacité de calcul pour l’apprentissage du modèle). Finalement, nous nous sommes rabattus sur une solution de secours, plus simple à implémenter à notre échelle, mais tout de même exigeante, celle consistant à fournir dans un premier temps, une liste de composants adaptés aux besoins du robot à concevoir.
* **Recours à l’API ChatGPT :** Concevoir notre propre modèle pour construire du code étant trop complexe, nous avons envisagé d'utiliser l'API ChatGPT pour produire du code en langage URDF, conscient de son potentiel pour améliorer notre travail. Cependant, les coûts associés à son utilisation nous ont très vite fait abandonner l’idée, nous obligeant ainsi à privilégier des alternatives moins onéreuses. Le modèle 3.5 donnant des résultats peu concluants, pour un prix raisonnable (environ 0,01 centime/requête), la modèle 4 donnait des résultats beaucoup plus intéressants, mais chaque requête via l’API coûtait 0,10 centime. Additionné au fait qu’il est nécessaire d’effectuer plusieurs itérations (requêtes) avant d’obtenir un modèle URDF correct, chaque génération coûterait rapidement trop cher.
* **Complexité des systèmes robotiques :** La nature intrinsèquement complexe des systèmes robotiques a ajouté une couche supplémentaire de difficulté à notre projet. Pour produire un résultat efficace, l’application devait théoriquement être apte à gérer un ensemble de paramètres complexes et interdépendants (les composants, leurs interactions, la structure, programmation du robot, implications environnementales, etc). Néanmoins, nos difficultés personnelles sont à nuancer. En effet, il est aujourd’hui possible de simuler des systèmes de plus en plus complexes à des niveaux de plus en plus précis. La démocratisation des connaissances techniques n’a jamais été aussi développée qu’actuellement, notamment avec les LLM. Nombreux sont les domaines qui ont récemment vu l’arrivée d’assistant de très bon niveau permettant d’accompagner les utilisateurs dans leur travail. Il est donc logique que cette révolution qu’est l’IA vienne s’appliquer à des concepts de plus en plus complexes pour les populariser et aider les ingénieurs, comme les non-ingénieurs, à développer des produits en phase avec les besoins actuels et futurs. Les difficultés que nous avons rencontrées étaient en grande partie techniques, cependant, il y a fort à parier que des entreprises spécialisées dépasseront rapidement certaines de nos problématiques pour arriver à un produit fini d’excellent niveau.
* **Mise à niveau de nos connaissances :** Au fil du projet, nous avons acquis de nombreuses compétences et connaissances par le biais de diverses sources (wiki, forums spécialisés, youtube, GitHub, ChatGPT). Cela nous a permis d’obtenir les bases de plusieurs domaines complexes tels que la création de modèles textuels, le fine-tuning de modèles existants, la gestion de prompts via des API redirigeant vers des modèles de génération 3D. Cependant, ces technologies étant encore à leur balbutiement et tendant à se démocratiser, la disponibilité de ressources/documentations reste limitée. Additionné au fait que ces sujets sont complexes et qu’ils doivent être expliqués de manière très pédagogique afin d’en saisir un maximum de préceptes, la compréhension de ces derniers repose donc en grande partie sur les ressources disponibles en ligne et leur qualité, parfois peu ou trop descriptives pour un débutant.

Malgré tout, ces obstacles reflètent l’ambition portée par notre sujet. L’un des objectifs de cet enseignement étant de se confronter à des thématiques complexes, nous en retirons beaucoup de positifs puisque nos compétences et notre savoir ont été considérablement enrichis.

### 4.3. Critiques et perspectives

De nos jours, la notion d’industrie 4.0 tend à promouvoir des systèmes polyvalents et de plus en plus adaptables plutôt que hautement spécialisés. De ce constat naît une question qui mérite d’être posée : est-il vraiment judicieux de continuer à vouloir développer des robots extrêmement spécialisés et qui seront forcément coûteux (ne pouvant pas bénéficier des économies d'échelles) ?

Finalement, produire un robot par situation (selon contexte spécifique) est une façon de faire peut-être trop artisanale. Cependant, pour des industriels cherchant à développer des produits commercialisables à grande échelle, cet outil pourrait s'avérer utile, permettant d'accélérer le processus de création et rendant le produit final plus compétitif. En revanche, il est peu probable qu'un utilisateur unique opte pour une telle solution, on l’imagine facilement préférer un système plus réadaptable, lui évitant d’investir beaucoup dans un produit potentiellement plus rapidement obsolète car étant trop spécialisé.

Il serait donc plus pertinent que notre application vise à produire des systèmes réadaptables plutôt que parfaitement spécialisés. Ce changement de ligne directrice pouvant ainsi offrir de meilleures potentialités économiques à notre projet. Avec cette critique en tête, on pourrait aussi envisager un outil conçu pour réadapter des systèmes déjà existants avec pour but, par exemple, de minimiser les coûts induits par ces réorganisations. Cette nécessité se justifie d'autant plus aujourd'hui avec les exigences croissantes de modernisation ainsi que de mise aux normes des systèmes robotiques et des lignes de production, se rapprochant de services comme Cognite.

Un autre point à considérer serait l’aspect écologique de ce projet. Le sujet des matériaux rares, du recyclage et de l’énergie prend de plus en plus de place dans notre société. Il pourrait donc être intéressant de questionner la nécessité d’avoir des robots spécialisés nécessitant individuellement plus de matériaux, là où un agent multi-tâches nécessiterait moins de matériaux et supposerait une meilleure optimisation de sa consommation/configuration pour effectuer un maximum de tâches en minimisant son impact énergétique/environnemental. Plus la diversification robotique sera grande, plus le besoin en composants différents sera large, compliquant la ré-utilisation/recyclage des composants.

Enfin, d’un point de vue psychologique, une entreprise serait-elle prête à laisser un logiciel générer entièrement un robot? La question de laisser une application agir en autonomie complète afin de répondre à un problème aussi complexe que celui-ci pourrait susciter des craintes à l'utiliser sans l’intervention d’un humain. Or l’intervention d’un humain viendrait contredire le principe même du modèle de cette application. Au même titre qu’à l’arrivée d’Internet, la véracité des propos disponibles a été remise en cause et à l’heure ou ChatGPT est capable de fournir des informations d’excellente qualité, il n’est pas certains que l’humain soit prêt à laisser une machine gérer un projet à 100%, sans son intervention et son contrôle.

Une fois toutes ces critiques posées, il est important de rappeler que l’humain n’a cessé de tenter d’optimiser son temps de travail, ce qui a donné lieu à l’industrialisation connue aujourd’hui, cherchant toujours plus à optimiser les flux, d’approvisionnement, de production, de livraison, etc. Dans cette logique de quête d’efficacité et de simplicité, la robotique aura un rôle important à jouer et les métiers qui s'articulent autour de ce secteur, seront le fruit d’outils pensés et réfléchis, de façon à ce que la machine soit la plus autonome possible afin de limiter au maximum l’intervention humaine. L’idéal étant que la machine devienne complètement autonome.

## 

## 5. Conclusion

### 5.1. Apports personnels

###### Fabien GAUTIER

Ce projet m’a permis d’approfondir des connaissances et thématiques que j’avais pu voir par le passé. Il m’a permis d’avoir une réelle expérience avec l’utilisation et le développement d’un outil d’intelligence artificielle. Je voulais absolument réaliser un projet impliquant de l’IA afin de savoir si travailler sur de tels sujets me correspondraient dans ma vie professionnelle. De plus, j’ai pu continuer à approfondir mes connaissances en Python en utilisant des outils ou des méthodes que j’ai pu utiliser dans le passé, ou alors en développer de nouvelles pour répondre à mes besoins.

La programmation et l’informatique sont des sujets qui me passionnent, j’ai donc beaucoup aimé mon expérience avec notre groupe qui nous a permis de travailler et approfondir un sujet qui nous intéressaient tous les trois. J’ai également beaucoup aimé travailler en équipe avec mes camarades, notre groupe était très adapté à ce type de projet car nous pouvions compter les uns sur les autres pour nous diviser le travail et nous faire réellement confiance sur les sujets que chacun traitait.

Finalement, le travail de recherche que nous avons eu afin d’essayer de créer un outil novateur m’a fortement motivé à travailler dans des secteurs innovants. Ce projet m’aura été très utile dans les divers questionnements que j’ai pu avoir au sujet de mon projet professionnel.

###### Odyssé LOIZEAU

Sur plusieurs aspects, ce projet est instructif.

* Le premier est la connaissance technique acquise dans des domaines qui seront utilisés sans aucun doute dans l’entreprise de demain et le monde de l’innovation. Arriver dans le milieu professionnel avec un minimum de connaissances techniques à parfaire permettra de pouvoir rapidement être mis sur des tâches en relation avec ces compétences intéressantes, et donc avoir accès à des missions pour lesquelles je porte un intérêt personnel plus fort. Par ailleurs, la possibilité d’adapter et modifier rapidement un modèle déjà existant me permettra, pour certaines applications, d'obtenir des résultats de très bonne qualité en peu de temps.
* Sur un plan plus organisationnel, ce projet aura permis de mettre en évidence l’importance d’avoir une méthode de suivi du travail effectué et de l’utilisation de certaines méthodes par rapport à d’autres, plus adaptées ou non. Par exemple, il sera plus simple d’utiliser une méthode en cascade ou en V lorsque les tâches à effectuer sont connues et ont simplement besoin d’être légèrement adaptées à un contexte. Dans notre cas, cela se rapprochait plus de l’innovation/recherche, nous n’avions donc sur le papier pas de feuille de route, utiliser une méthode Agile était donc l’une des meilleures options disponible, afin d’avancer de façon structurée et efficace.
* Pour terminer, je dirais que ce projet aura permis de voir la différence entre l’idéation d’un projet et sa réalisation. Lors de l’idéation, il est simple de planifier dans sa tête les technologies à utiliser et comment l’ensemble du projet va se dérouler. On pense avoir saisi l’ensemble des aspects, même parfois techniques, des contraintes et on les anticipe. Cependant, une fois en réalisation, on prend conscience d’autres problématiques (surtout dans l’innovation), qui sont parfois basiques mais complexes à surmonter. Cela permet de prendre en expérience sur cette partie projet, qui me semble être la plus importante lorsque l’on initie un projet novateur: être capable de rapidement fixer un cadre, avec des objectifs et des limites, même dans un domaine non maîtrisé. C’est, à l’heure actuelle, la compétence que je souhaite le plus développer.

###### Kléden ROUSSEAU

Il ne serait pas sincère de ma part de dire que je n'étais pas inquiet au début du projet, mes appréhensions n’ont d’ailleurs fait que grandir au fur et à mesure que nous progressions dans nos recherches. J'étais conscient de la difficulté du sujet que nous abordions. L'ambition du projet était certes motivante, mais engendrait aussi une forme de stress. En fin de compte, cette préoccupation était justifiée car nous n'avons pas réussi à mettre en œuvre notre idée de départ.

Néanmoins, cela m'a aidé à mettre les choses en perspective. En tenant compte des délais et des ressources disponibles, il était irréaliste de penser que nous pourrions réaliser parfaitement ce que nous avions envisagé. Toutefois, je ne considère pas cela comme un échec, bien au contraire. Notre montée en compétences nous a permis de découvrir les failles de notre idée originale. Il est essentiel de commencer par de bonnes idées, mais l'étape suivante consiste à les adapter à la réalité, ce que nous avons fait en produisant cette PoC. Bien qu’il reste encore bon nombre d’aspects à perfectionner, nous sommes parvenus à produire une application web concrète d’aide à la conception de robot, en faisant preuve de résilience vis-à-vis des contraintes rencontrées ici et là.

Ce projet m'a permis de gagner en expérience. J'ai compris l'importance de la gestion de projet, de l'équilibre entre apprentissage et réalisation, et de la nécessité de revoir régulièrement nos avancées. J'ai acquis des connaissances dans des domaines que je ne connaissais que de nom ou de loin : l'intelligence artificielle, le machine learning, la robotique et le développement web, des compétences que je peux désormais valoriser. Grâce à cette expérience, je suis maintenant bien mieux préparé pour prendre part à de futurs projets.

### 5.2. Conclusion générale

Bien que nous ayons rencontré plus de difficultés que ce à quoi nous nous attendions, ce projet a tout de même abouti à des résultats et des réflexions assez prometteuses. L'objectif initial était de produire une *proof of concept* d’un outil d’aide à la conception de robot, ce que nous avons réalisé, même si de nombreuses améliorations restent à apporter. Compte-tenu du temps imparti, de nos moyens ainsi que de nos connaissances au commencement du projet, nous sommes satisfait du travail présenté.

Les sujets sur lesquels nous avons travaillé sont d’actualités et tendent à se développer (IA, robotique, 3D) et prendre une part significative dans le monde de l’industrie et s’étendre à d’autres domaines. On peut citer par exemple Nvidia avec *Isaac Sim*, logiciel de simulation capable de reproduire un environnement l’ensemble de sa physique, ou *Ecostruxure* de Schneider Electric ayant pour objectif d’éliminer au maximum la ré-ingénierie des chaînes de production et ainsi concentrer les efforts d'ingénierie sur des problèmes plus complexes. La robotique peut être perçue comme la possibilité à un ordinateur d’effectuer des actions dans le monde physique de manière autonome. L’essor de l’Intelligence Artificielle et l’accélération qu’elle prend peut laisser présager un très fort développement de la robotique dans les prochaines décennies, en automatisant notamment de nombreuses tâches pénibles ou dangereuses. L’IA à la possibilité de réduire le besoin d’expertise de l’humain, la robotique pourrait avoir le rôle de réduire le besoin de main d'œuvre humaine avec toutes les modifications que cela impliquerait (législatives, sociales et même environnementales).

Ce projet a été une expérience pratique précieuse, nous mettant dans la peau de futurs ingénieurs confrontés à des défis réels, aussi bien techniques qu’organisationnels. Il a nécessité de mettre en place une planification, une gestion efficace du projet, l'utilisation de nos connaissances existantes et l'apprentissage de nouvelles compétences dans les domaines nécessaires. Nous avons également dû nous informer sur les dernières tendances et innovations en intelligence artificielle et en robotique, des domaines où nous avons désormais acquis une bonne base. Cette expérience a été enrichissante, nous préparant pour notre future carrière en ingénierie.

**ANNEXE 1 : DIAGRAMME DE GANTT**

Une image contenant capture d’écran, texte, ligne, nombre

Description générée automatiquement