Internship Report

Creation of a database of human movements for transfer to robot manipulators

Constitution d'une base de données de mouvements humains pour le transfert vers des robots manipulateurs

Presented by

Gabriel de Souza Marcos

July 2025

Company tutor: **Sébastien LENGAGNE, Antonio Claudio DE SOUSA DOS SANTOS FILHO**

Academic tutor: **Romuald AUFRERE**

Jury: **Romuald AUFRERE, K. El Khamlichi**

Polytech Clermont - Université Clermont Auvergne INP
Department S3ER Systèmes Embarqués, Energie Electrique et Robotique.

Abstract

This report presents a study on dataset creation, including supervised learning applications. The goal is to support the doctoral thesis of Antonio Claudio de Sousa dos Santos Filho, who is investigating various Transfer learning (TL) methods, and to develop a human motion capture system for cube manipulation that enables the transfer of human skills to one or more robotic systems.

The report is organized into an introduction, description of the objectives, presentation of the research organization, internship development, methodology, results and conclusion.

Keywords: human movement, database, transfer learning, Mediapipe, Aruco, Supervised learning, Ros2.

Résumé

Ce rapport présente une étude sur la création d'ensembles de données, y compris les applications d'apprentissage supervisé. L'objectif est de soutenir la thèse doctorale d'Antonio Claudio de Sousa dos Santos Filho, qui étudie diverses méthodes de transfert de connaissances, et de développer un système de capture de mouvement humain pour la manipulation de cubes qui permet le transfert de compétences humaines vers un ou plusieurs systèmes robotiques.

Le report est organisé en introduction, description des objectifs, présentation de l'organisme de recherche, stage développement, méthodologie, résultats et conclusion.

Mots-clés : mouvement humain, base de données, transfert de connaissances, Mediapipe, Aruco, apprentissage supervisé, Ros2.

Acknowledgement

Contents

List of Figures

Acronymes

ML Machine learning. 7, 26, 27

RL Reinforcement learning. 13, 14

ROS2 Robot Operating System 2. 14

TL Transfer learning. 1, 7, 10, 12, 14, 26, 27

UNN Universal Notice Network. 12, 13, 19

Introduction

The automation of robotic systems through Machine learning (ML) offers numerous advantages, such as increased operational efficiency, reduced human error, cost savings, improved workplace safety, and the ability for humans to focus on more complex and creative tasks. In this context, ML plays a crucial role by enabling robots to learn from data and autonomously adapt to dynamic environments.

One approach to applying ML is through TL, which involves preserving and reusing the expertise of a previous human operator to train a new robotic system. However, a key challenge in this method lies in efficiently training these intelligent systems to perform tasks with the same naturalness as a human.

The challenge faced is mainly because some tasks that are simple for humans can be surprisingly difficult for robots to execute. As explained by Moravec's paradox, everyday tasks that humans perform effortlessly can be extremely complex for robots. The paradox, articulated in the 1980s[3]—primarily by Hans Moravec—highlights the difficulty of replicating forms of knowledge that have evolved over hundreds of thousands of years of human development, in contrast to more recent cognitive skills, such as playing chess or solving algebraic problems.

Given the current human technological level, it is more efficient to use humans to demonstrate certain tasks, thereby enabling a more effective transfer of knowledge. There are several advantages to using human motion, especially when dealing with high-level instructions, as it allows for greater flexibility and naturalness in task execution. This approach enables the robot to learn not only the sequence of actions but also the subtle nuances of human movement, which can be crucial for executing complex or dynamic tasks.

In this context, the current internship aims to explore and apply this method by collecting a dataset of a human performing a task, in order to develop a training module, named module UNN (Universal Neural Network), capable of transferring that human knowledge to one or more robots.

Chapter 1

Internship Framework

This chapter presents an overview of the research organization where the internship was conducted, as well as the motivation and objectives that guided the work.

1.1 Company/Organization

The Pascal Institute is an interdisciplinary research laboratory affiliated with the University of Clermont Auvergne. It focuses on strategic areas within Engineering and Systems Sciences, including Process Engineering, Mechanics, Robotics, Information Science, and Health. [4]

The institute is a member of Clermont Auvergne INP, which brings together three engineering schools: ISIMA, POLYTECH Clermont, and SIGMA Clermont. It develops innovative and intelligent systems through a systemic, multi-scale approach, covering research fields such as imaging, mechanics, robotics, and more.

In addition, the institute is also involved in three joint laboratories: FACTOLAB, BIOEM, and BIOGASLAB. It is a member of two competitiveness clusters: Céréales Vallée, which focuses on the cereal industry, and ViaMéca, which aims to accelerate innovation in the mechanical industry. The institute collaborates with several industrial partners, including major companies such as Renault, Michelin, and EDF. Furthermore, the Pascal Institute is overseen by three main bodies: CNRS, the University of Clermont Auvergne, and SIGMA Clermont.

The institute has a substantial research staff, consisting of 137 teacher-researchers (UCA, Clermont Auvergne INP), 5 CNRS researchers, 28 P/MCU-PH (UCA/CHU), 23 hospital practitioners (PH) (CHU), 5 PRAG/PAST, 15 technical and administrative staff (BIATSS) (UCA), 11 CNRS technical staff (ITA), 147 PhD students, and 28 fixed-term contract researchers (CDD).

Figure 1.1: Laboratoire de recherche Institut Pascal.[1]

The research at the Pascal Institute is organized into five main departments, each with specific research areas, as illustrated in Fig. 1.2:

- **M3G:** Mécanique, matériaux, structures.

- **PHOTON:** Photonique, ondes, nanomatériaux.

- **GePEB:** Énergie et biosystèmes.

- **TGI:** Thérapie par l'image.

- **ISPR:** Systèmes de perception, robotique.

I carried out my internship within the ISPR department. More specifically, I was part of the Modeling, Autonomous, and Control of Complex Systems (MACCS) team. This research group primarily focuses on the modeling and control of mobile and manipulative robots, robot vision, and visual servoing—topics that are closely related to the subject of this report.
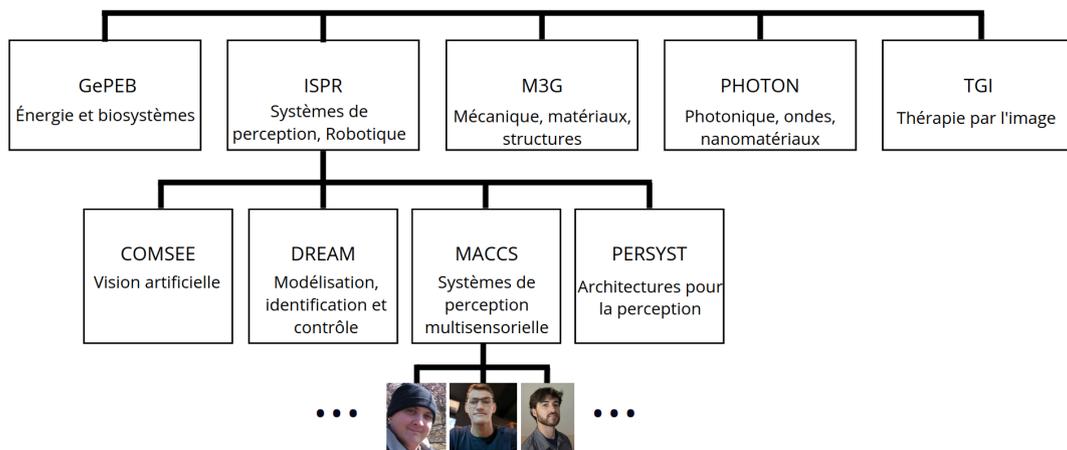

Figure 1.2: Organizational structure.

1.2   Context

When facing the challenge of solving a task with a specific robot, two main approaches can be adopted: the model-based approach and the learning-based approach. The first one, which is classical in nature, requires considerable human effort to plan how the task should be executed in order to generate the commands that will be sent to the robot to control it. This method has proven useful and effective for simple tasks; however, it presents significant limitations when dealing with complex and dynamic tasks.

On the other hand, the learning-based approach has stood out for its effectiveness in solving complex tasks — such as, for example, the manipulation of a Rubik's Cube by a robotic hand[5]. This approach uses artificial intelligence techniques to learn control policies directly from interaction with the environment. Despite its advances, training these models still requires high computational and time costs. Furthermore, when one wishes to apply the same learned behavior to robots with different morphologies, it is necessary to restart the training process, which further increases the associated costs.

A promising solution to this problem is the use of transfer learning techniques, which aim to reuse the knowledge acquired by one robot during the training of a task to accelerate the learning process in other robots that need to perform the same task. Despite its potential, this approach is not trivial, as it involves the challenge of dealing with different robot morphologies.

Motivated by recent advances in this area, the PhD student Antonio Claudio de Sousa dos Santos Filho is conducting research on the development of new strategies for performing transfer learning between robots with different physical structures. An interesting extension of this idea is to explore the transfer of knowledge not only between robots but also from humans to robots. With that in mind, the present internship complements Antonio's thesis by focusing on a specific, yet equally important, aspect: the

transfer of knowledge from a human to a robot. In other words, this work investigates the application of transfer learning in robots using human motion data. It is important to note that this study addresses only the initial step of this approach, using a simple task that is easy for both a robotic agent and a human to perform.

Despite the advantages of this application— as discussed in this report introduction— several challenges must be considered. One major issue is the difference in body structure between humans and robots. Moreover, human motion data often lacks precise joint-level information, such as torque or joint velocity, which are critical for robotic control. Robots also rely on predefined control models, making it difficult to reproduce the smooth and intuitive characteristics of human motion.

To address these challenges, this internship uses a human motion sensor to capture data from each body articulation. With this data, the task module is built using supervised learning techniques. This implementation improves the robot's interpretation of the human dataset, enabling it to learn meaningful patterns and motion trajectories from labeled data. By associating specific human gestures with corresponding robot actions, the robot can gradually generalize and reproduce the task, as illustrated in Fig. 1.3, taking into account its own kinematic and dynamic constraints.
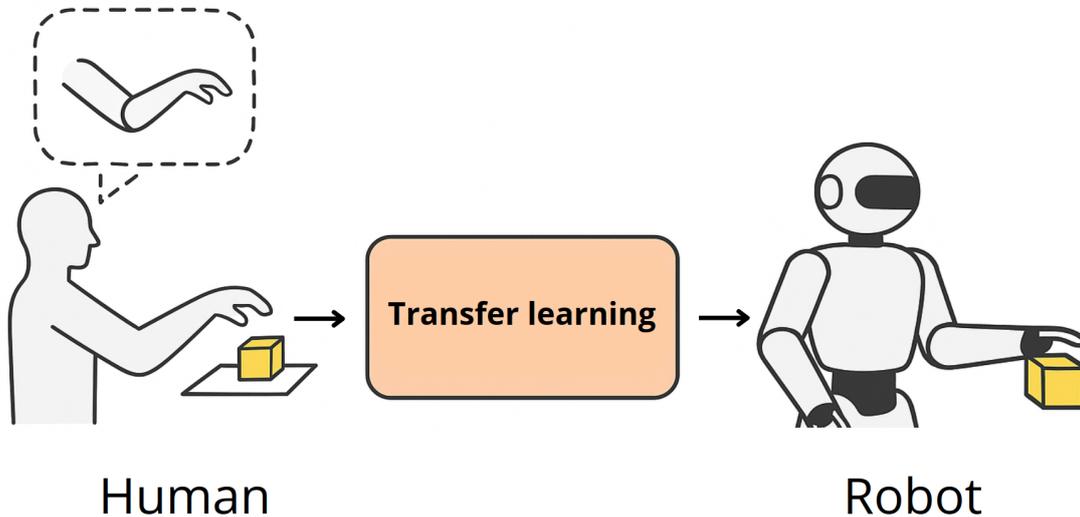
Figure 1.3: Implementation of a supervised learning technic to perform a pick and place task based on human movements.

1.3 Objectives

The overall objective of this internship is to set up a human motion capture system for cube manipulation, allowing the transfer of human skills to one or more robotic systems. The specific objectives are outlined below:

- Provide framework to collect a dataset containing the movements of a human and a robot performing the same primitive task (for example, a simple motion involving moving the effector in circles).

- Provide framework to collect a second dataset containing only human movements solving the desired task that the robot is expected to learn.

- Apply supervised learning algorithms and TL methods to train the robotic agents.

- Evaluate the performance and efficiency of the trained robotic systems.

These objectives are presented in a specifications document (see Table. 1.4), which provides informations about the tasks to be completed, as well as the priority level of each one. The low-priority tasks were not requested by the client and are considered supplementary in the context of this internship.

Requirement	Criterion	Criterion Level	Tasks priority
Motion capture program	Camera image	Display the representation of the detected markers and landmarks	Mandatory
Human-Robot Data Collection	Number of data rows	at least 10k	High
Human-Only Data Collection	Number of data rows	at least 10k	High
Transfer Learning Application	Success rate	70% of attempts	Low
Performance Evaluation with robot Ned²	Success rate	70% of attempts	Low

Figure 1.4: Specifications document.

Chapter 2

Background and Related Work

The objective of this chapter is to explain the main theoretical concepts applied during this internship to perform TL between two agents. In addition, the concepts and tools used to construct the dataset with human and robot data are introduced.

2.1 Transfer Learning

TL, in the robotic context, is a way to transfer knowledge between two or more robots. This concept is essential for reusing control policies across different robots, avoiding the high computational and energy costs of training each new robot from scratch.

One of the major challenges in robotic TL is morphological heterogeneity: robots with different physical structures (such as number of joints and links lenghts) do not share the same state or action spaces. This incompatibility makes it impractical to directly reuse learned models from one robot to another—even when the task remains identical. For this reason, several studies have addressed this issue.

One example is the Ph.D. thesis of Mr. Mehdi Mounsif [6]. This work validated the performance of the Universal Notice Network (UNN)—a modular architecture designed to decouple task knowledge from robot-specific control policies. The idea is to encapsulate task knowledge into a separate component—the task module—which can be transferred across different robots.

Motivated by this work, Samuel Beaussant [2]proposed an alternative pipeline for the UNN module: an enhanced method called the Latent-Space Universal Notice Network (LS-UNN), as shown in Fig.2.2. The key innovation is the replacement of manually defined interfaces with a learned, robot-agnostic latent space, constructed from time-aligned demonstrations across multiple agents.

In order to achieve this new interface, he created a input and a output bases, denoted respectively as B_i and B_o. These bases, specific to each robot, can be learned by training a model on a primitive task (e.g., a reaching task). They function as transformation modules that enable a shared understanding between different agents within a common representation space.

The bases B_i and B_o are created by following a base training procedure, illustrated in Fig. 2.1. In this procedure, the training module must approximate the representation space (encoded data) of both agents ("Robot 1" and "Robot 2")—with a human replacing "Robot 1" in the context of my stage. This approach assumes that the encoded data is obtained from both agents performing the same task, which justifies the use of a primitive task, i.e., one that is simple to execute. The output base, in turn, approximates the encoded data into commands that result in the same desired end-effector position and orientation for each agent.
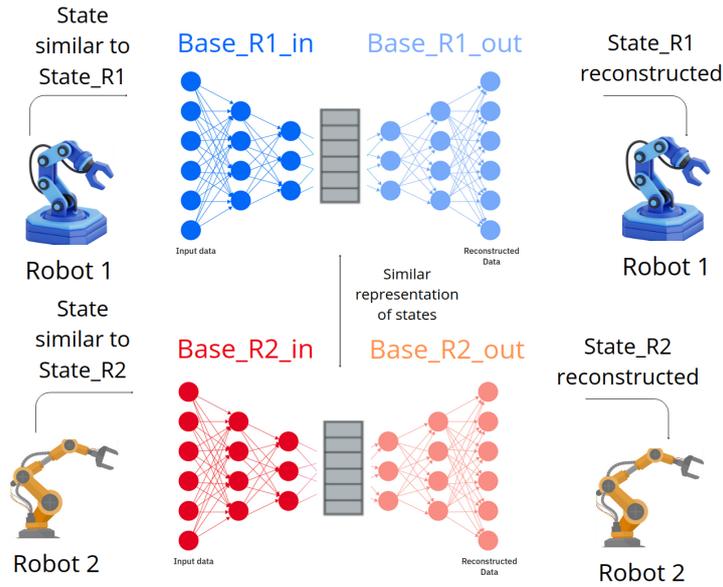
Figure 2.1: Base training procedure.

The latent space, that is produced by this base training procedure, goal a unified representation that abstracts away robot morphology.

Once this space is constructed, a task module can be trained within it and reused by any robot, regardless of its structure or control space. The task module is trained to take the current robot state in the latent space and transform it into the desired target state, using frozen input and output bases to evaluate whether the task is being successfully performed or if additional training is needed.
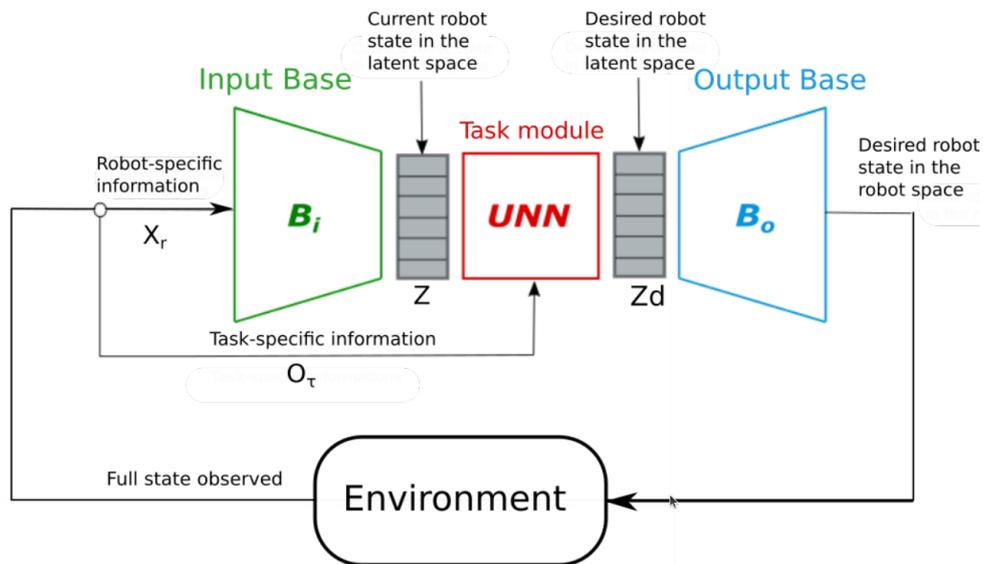
Figure 2.2: Schematic representation of the UNN latent space pipeline and its three different modules.[2]

In the context of this internship, where motion is transferred from humans to robots, the LS-UNN method is particularly suitable. It addresses both the morphological gap and the complexity of human movement, requiring an abstract domain to interpret data from both agents. This facilitates a more natural and scalable integration between human and robot execution.

The approaches previously mentioned relied on Reinforcement learning (RL) to train the UNN and LS-UNN modules. However, in this internship, a supervised learning approach was adopted. RL involves agents learning through trial and error, guided by a reward signal that evaluates the effectiveness of their

actions within an environment. While powerful, this method can be complex, particularly when designing suitable reward functions for tasks involving the transfer of human motion to robots.

In contrast, supervised learning is a machine learning paradigm in which models are trained using labeled input-output pairs, enabling them to learn direct mappings from inputs to desired outputs. This approach was chosen for its simplicity and directness, especially in scenarios where labeled data is available and designing a robust reward function for RL would be challenging or impractical.

2.2   Motion capture

The dataset forms an important part of TL, as it is essential for the construction of the agent bases and the training of the UNN module. The dataset includes information of the position and orientation of both the agents' movements (human and robot) and the manipulated cube.

2.2.1   Constraints

To ensure high-quality data acquisition, the choice of motion capture libraries is critical. Several constraints must be considered:

- Compatibility with the systems used, particularly with robotic frameworks such as Robot Operating System 2 (ROS2).

- Robustness under real-world recording conditions, including lighting variations, reflections, and camera angles.

- Accuracy of detection, both for object position and orientation and for human joint estimation.

- Easy integration and availability of documentation, to support a faster and more effective implementation.

2.2.2   Object motion capture

Given the constraints, several options were studied according to the dataset requirements: AprilTag, ARToolKit, Chilitags, and Aruco for capturing the motion of the cube.

AprilTag [7] is a very robust library that detects markers—specific binary patterns (tags) with well-defined borders. It also provides reliable pose information (position and orientation) for systems and is widely used in augmented reality (AR) applications. Similarly, the ARToolKit library is also known for its use in AR, as it allows for the detection of simple markers [8] and real-time camera pose estimation, enabling virtual objects to be overlaid on real-world video. However, the ARToolKit project has been officially discontinued in its main repository for several years, and updates have become rare, limiting its compatibility with modern systems such as ROS2.

Another library studied is Chilitags [9], which offers satisfactory performance for basic use, such as moderate frame rates (FPS) and compatibility with simple hardware platforms. It does not rely on heavy dependencies and provides many out-of-the-box functions for initialization, detection, and pose estimation with just a few lines of code. However, it lacks robustness and may be limited in precision due to its sensitivity to filming conditions.

In contrast, a more suitable option is the Aruco library [10]. Pose estimation for each marker is performed using detection algorithms, for example PnP algorithms, and refined using the corners of the markers. Reprojection error can be controlled through good camera calibration and marker definition, resulting in greater robustness. Additionally, Aruco [11] can detect multiple markers simultaneously in a single image, each with a unique ID, enabling representation of different objects or reference points in a scene.

Since cube detection and robot control are directly integrated with OpenCV and ROS2, and because a considerable level of robustness is required to mitigate issues in recording quality, the chosen library for this internship project was Aruco. It also stands out for its large community and extensive support materials, which ease the learning and implementation process. A comparative table of the mentioned libraries is presented in Fig.2.3.

Attribute	AprilTag	ARToolKit	Chilitags	Aruco
Pose	Accurate	Good	Basic	Accurate
Robustness	High	Medium	Low	High
ROS2	Partial	Low	Limited	High
Community	Active	Weak	Small	Strong
Notes	Used in AR	No longer maintained	Lightweight	Widely supported

Figure 2.3: Comparative table to object motion capture.

2.2.3 Human motion capture

It is also necessary to record the data from each joint of the human body, so some options were studied: OpenPose, MMPose, AprilTag, OpenCV (with DNN), and Mediapipe.

The OpenPose library [12] detects the position of body joints—including the body, hands, face, and feet—from static images or live video, using convolutional neural networks (CNNs). Although it offers high accuracy, this library requires powerful hardware to run in real time due to its high computational cost.

Another alternative is MMPose [13], a modular and therefore flexible framework for human pose estimation. Its wide variety of integrated models allows better adaptation to different project needs. However, its use requires prior knowledge, making implementation more complex[14].

A more popular option is OpenCV (with DNN) [15], which already offers several ready-to-use tools, enabling the construction of complete pipelines—such as image capture, edge detection, perspective correction, and noise filtering—all within a single workflow. However, its precision and the variety of available models are not as comprehensive as those in dedicated frameworks like OpenPose or MMPose. Furthermore, there are no built-in hand detection models in OpenCV's DNN module, limiting it for full-body motion capture.

On the other hand, the Mediapipe library [16] uses its own models such as BlazePose (for pose) and BlazeHand (for hands), enabling the detection of 32 different landmarks across the body and 30 on the hands. The models are optimized using TensorFlow Lite, allowing for high efficiency and speed in detection. Additionally, it is easy to use, because of the extensive documentation and active community [17].

The main interest in capturing human motion lies in the arms, torso, and hands. Moreover, robustness and speed in motion capture are necessary, as filming conditions, as mentioned earlier, can introduce noise. Therefore, the chosen library was Mediapipe, which possesses these characteristics and provides accessible documentation for implementation. A comparative table of the mentioned libraries is provided below 2.4

Attribute	OpenPose	MMPose	OpenCV (DNN)	Mediapipe
Body	Full	Configurable	Limited Full	Full
Accuracy	High	High	Medium	High
Hardware	High	Medium/High	Low	Low
Simplicity	Medium	Low	High	Very High
Hands	Yes	Yes	No	Yes

Figure 2.4: Comparative table to human motion capture.

Chapter 3

Approach and Implementation

This chapter presents the practical work, which process are described in section 2.1, carried out for the development of the UNN module along with the results obtained. The activities are divided into three main stages. First, datasets are collected for both a primitive task and a pick-and-place task, involving both human and robot executions. Second, the collected data is used to train the input and output bases. Once this process is complete, the task module can be developed.

3.1   Dataset construction

The dataset consists of the position and orientation of both the agents' movements and the cube. Specifically, it includes the 3D position (x, y, z) and quaternion orientation (qx, qy, qz, qw) of the cube, as well as the 3D position of each agent (human and robot).

   This section discusses the motivation behind capturing datasets of the cube and human motion in different contexts—namely, a primitive task and a pick-and-place task. It also describes the data collection process and presents the results obtained.

3.1.1   Primitive task dataset

As indicated in Section 2.1, the first step is to collect similar states (or trajectories) for both the human and the robot. The goal is to establish a common input-output basis that enables a shared representation of task executions.

   To achieve this, we capture the trajectories of both the human and the cube. From these trajectories, we then compute the robot's joint movements that would result in the same cube trajectory. In other words, the position data of the cube, recorded during human execution, serves as a reference to generate the corresponding robot joint movements. This ensures that both the human and the robot perform the task with time-aligned cube trajectories.

   This process can be illustrated with a clear diagram Fig.3.1 : first, a video recording provides, on one side, the human's position and, on the other, the cube's position. Next, a calculation is performed to determine the robot's trajectory that would produce the same cube trajectory.
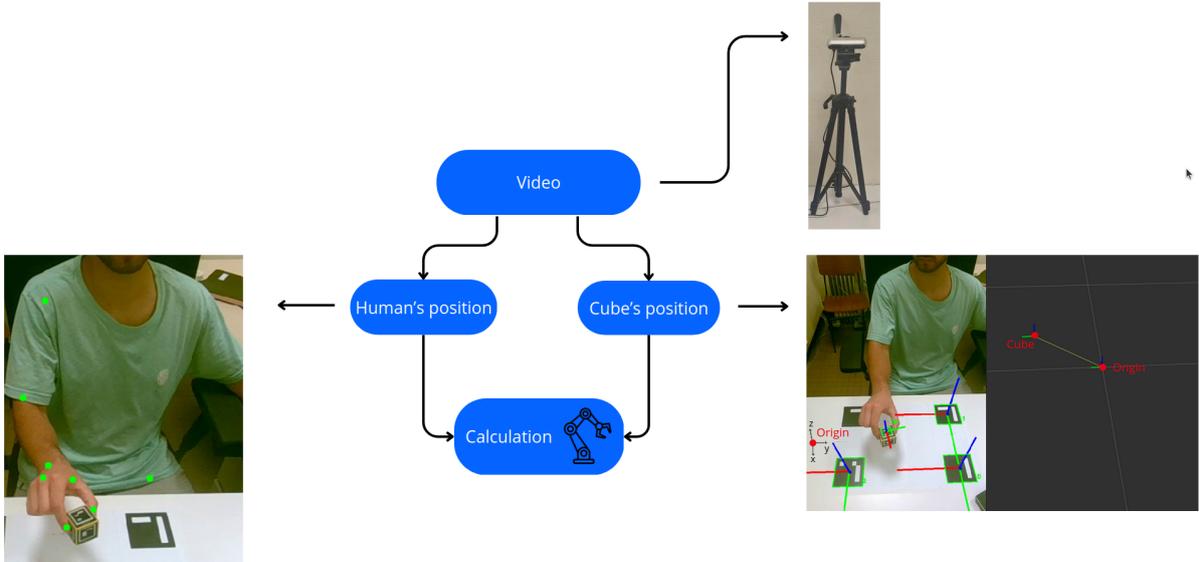
Figure 3.1: Procedure to collect robot, cube and human trajectories.

Data collection was carried out using an automatic segmentation process, which divided the recordings into shorter trajectories of approximately one minute each. This segmentation was triggered whenever landmarks of the left hand were detected in the frame, allowing for the isolation of relevant motion sequences.

3.1.2 Pick and place task dataset

The importance of this dataset lies in its use for training the UNN module. The module aims to determine the necessary commands to perform the task, and thus requires a dataset with diverse demonstrations of how the task can be accomplished.

The pick-and-place task used to construct the dataset involved grasping the cube from a random position and moving it to a fixed point on the table. This approach allows for a wide variety of motion possibilities, enabling the robot to learn multiple strategies for performing the task. In addition, similar to the primitive task dataset, an automatic segmentation process was applied. However, in this case, the data was segmented each time the task was completed.

3.2 Bases construction

Once the primitive task dataset is prepared, it becomes possible to construct the input and output bases. These bases serve as transformation modules that allow different agents to interpret and express motion within a shared latent space. This is essential for constructing the UNN module, as it is designed to utilize only the shared information between agents to accomplish the task.

In this internship, the bases are designed to model the relationship between human and robot movement, rather than the relationship between the movement of different robots, as explored in Samuel Beaussant's work [2]. The input base translates human motion into the latent space, producing a representation that corresponds to the robot's motion. The output base then takes this latent representation and reconstructs both the robot's and the human's motions, as illustrated in Fig. 3.2. Both bases are trained using supervised learning, as explained in section 2.1, to support this translation between agents.
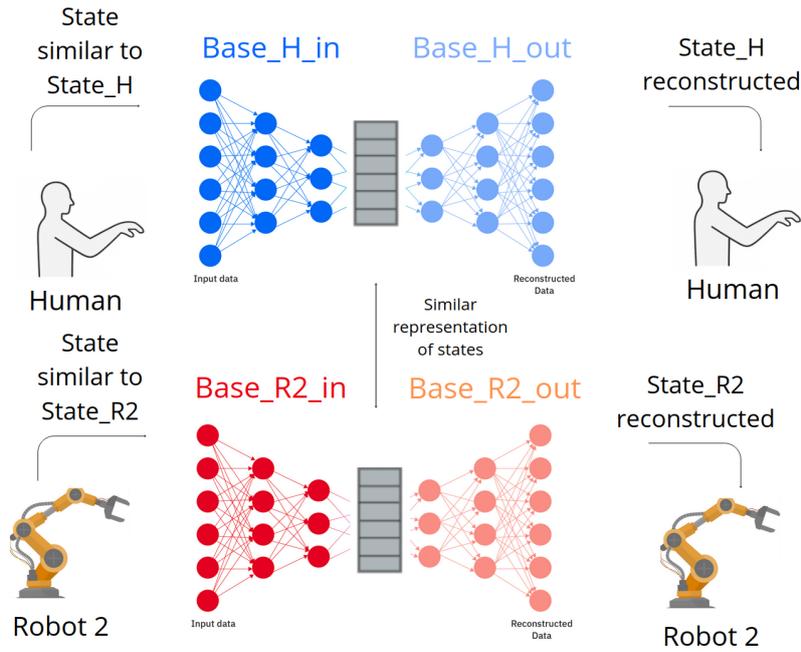
Figure 3.2: Bases training procedure with agent 1 as a human.

3.3 Module UNN Construction

After constructing the input and output bases for each agent, the next step is to develop the UNN module, which is responsible for learning and executing tasks. Importantly, these bases are kept frozen during the training of the UNN, meaning that their parameters are not updated.

Figure 3.3 illustrates this process: after extracting both the human's and the cube's positions during the pick-and-place task (as described in Section 3.1.2), these trajectories are used to train the UNN module (see Fig. 3.3a). The processed features, obtained by projecting the data onto the bases, are then fed into the module UNN.

At each time step t, the frozen UNN module takes the projected features as input and outputs the action to be taken by the robot at time $t + 1$ (see Fig. 3.3b). This output provides the specific commands required for the robot to replicate the human's pick-and-place actions, as observed in the dataset. Throughout this phase, the bases remain fixed (frozen).

During this internship, the UNN module was trained using supervised learning.

(a) UNN module training.
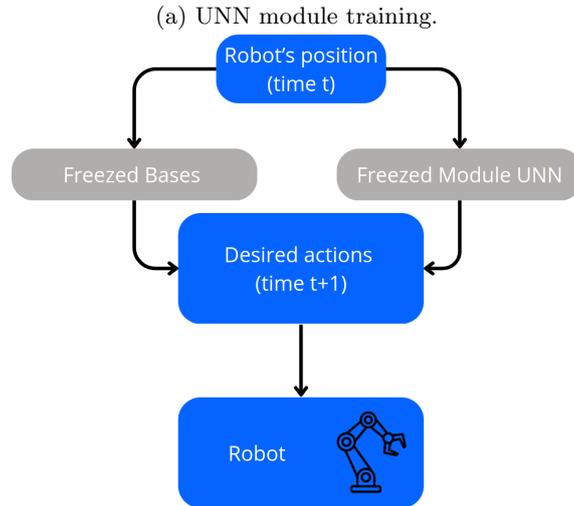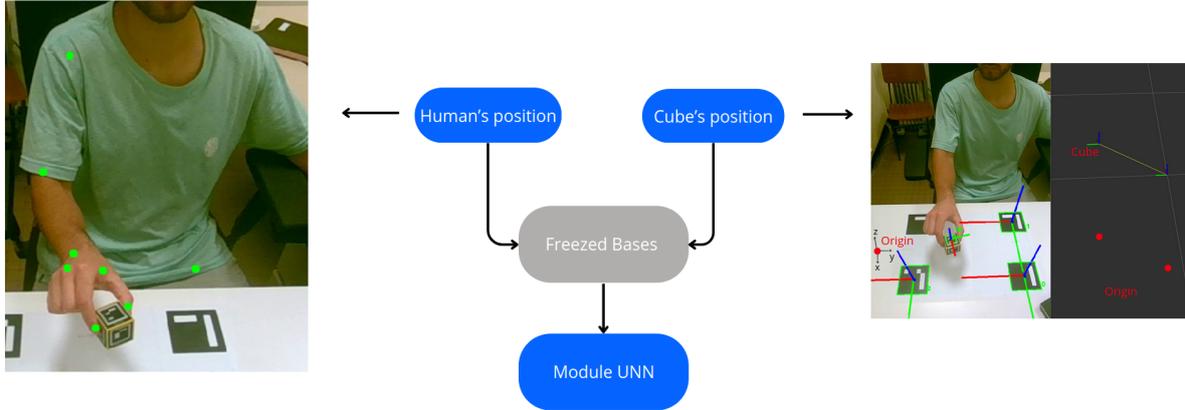
(b) UNN module utilization (action prediction at $t + 1$).

Figure 3.3: Schemas of the UNN module procedure.

3.4 Results and Analysis

The following section presents the organization and development of the work carried out during the internship. First, the architecture of the program will be detailed, illustrating the structure and interaction between its main modules. Next, the methods used for the automatic segmentation of sequences will be discussed, as well as the generation of the robot sequences. The organization of the work will then be presented through a timeline, highlighting the distribution of tasks over time. Finally, an overview of the progress made to date will be presented, including a table showing the percentage of completion of the internship.

3.4.1 Program Architecture

To begin, it is necessary to describe the overall architecture of the program. The diagram in Fig. 3.4 illustrates the structure and interaction between the main components, providing a clear overview of how the system operates. The robotics framework ROS 2 is used alongside a data capture system to generate

the datasets, which serve as the foundation for constructing the input and output bases required for the development of the task module.
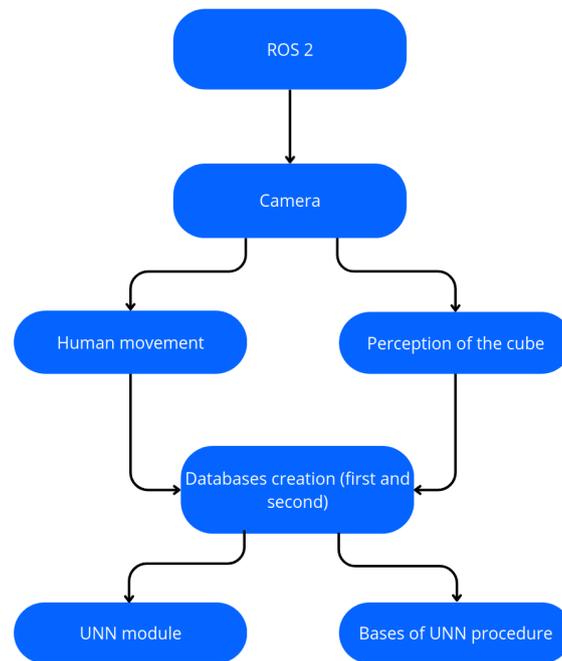

Figure 3.4: Architecture of the program.

3.4.2 Automatic Sequence Segmentation

A better approach to record the human and cube movement data was to segment them into several sequences, each containing trajectories with a controlled number of data. To achieve this, the initial method was to use detection of a specific part of the body to determine when to split the data and when to continue aggregating them. The selection of the body part was based on its need for the composition of the human movement dataset; the left hand was chosen because only the right hand was used to move the cube. So every time the left hand was detected, the data was segmented, and when it wasn't visible, indicated that the data sequence was not yet complete, then the data continued to be added. This process is illustrated in Fig. 3.5.

(a) Command to divide the data.

(b) Continuing to add data.

Figure 3.5: Automatic Sequence Segmentation

3.4.3 Generation of Robot Sequences

In addition to the human sequences, sequences for the robots were also generated during the development process. The robot movements were based directly on the dataset collected from the cube manipulation performed by the human. As a result, the robot sequences correspond exactly to the same segmented sequences recorded for the human body and the cube, since these data had already been divided and prepared for use in ROS 2. Using ROS 2 as the main framework, the end effector of the robot was controlled to replicate the trajectories extracted from the dataset, ensuring a reproduction of the demonstrated movements.

3.4.4 Work organization

The organization and planning of the work was made following a Gannt chart, Fig.3.6, that is used to visualize the distribution of tasks and the timeline followed throughout the project.
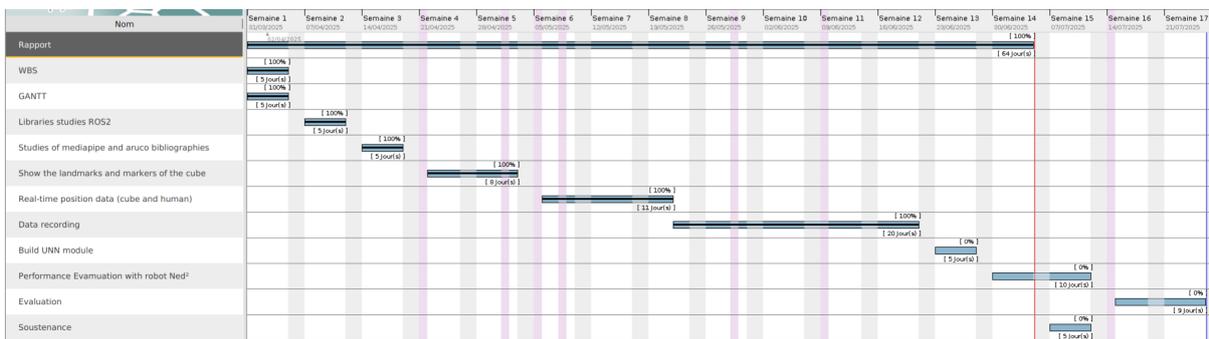

Figure 3.6: Updated Gantt at the end of the internship.
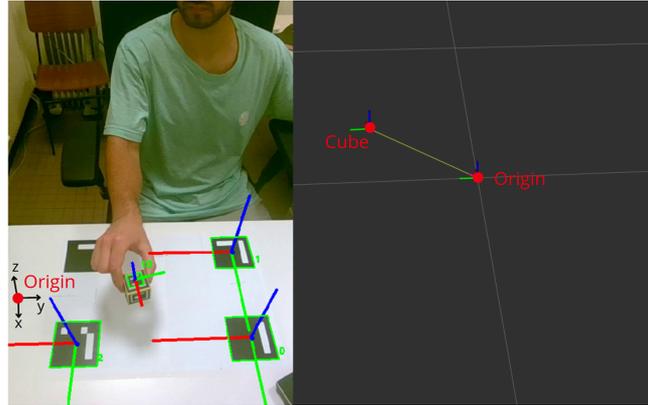
3.4.5 Realization

The main results achieved, as illustrated in Fig. 3.7, includes:

- Primitive task dataset: Five hours of cube manipulation were recorded on video, resulting in approximately 150,000 rows of data.
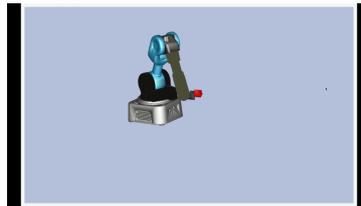
- Pick-and-place task dataset: Two hours of cube manipulation were recorded on video, resulting in approximately 30,000 rows of data.

(a) Human motion capture.

(b) Cube motion capture.

(c) Robot motion capture.

Figure 3.7: Dataset capture.

In addition to these results, the Gantt charts shown in Section 3.4.4 illustrate the project planning and its progression throughout the internship. The majority of tasks were successfully completed, with two tasks remaining to be finalized: "Build UNN module" and "Performance Evaluation with robot Ned²." This is also reflected in the specifications document, which shows in Fig. 3.8 that 100% of the primary objectives of the internship were achieved, and a total of 60% of all objectives were completed.

The main delay encountered during the internship was in the recording of the primitive task dataset. This phase required more time than initially planned due to multiple issues related to data quality during capture sessions. Several recordings had to be redone because of noise, inaccuracies in tracking, or inconsistencies in the marker detection systems, especially when using the ArUco library and MediaPipe simultaneously.

Despite this, the project schedule was adjusted accordingly, and most of the planned objectives were reached as intended. The current state of progress follows the updated Gantt chart, and the remaining work is expected to be completed by the end of the internship.

Requirement	Criterion	Criterion Level	Tasks priority	Concluded
Motion capture program	Camera image	Display the representation of the detected markers and landmarks	Mandatory	👍
Human-Robot Data Collection	Number of data rows	at least 10k	High	👍
Human-Only Data Collection	Number of data rows	at least 10k	High	👍
Transfer Learning Application	Success rate	70% of attempts	Low	👎
Performance Evaluation with robot Ned²	Success rate	70% of attempts	Low	👎

Figure 3.8: Updated Specifications document at the end of the internship.

Chapter 4

Cultural Differences

During this internship, I was immersed in the French work culture, which, as a Brazilian, provided a valuable opportunity to broaden both my personal and professional horizons. In addition, the experience of living in another country proved to be both interesting and enriching. In this chapter, I will discuss some of the cultural differences I encountered, both in everyday life.

4.1   Sports

In Brazil, sports are generally associated with warm-weather activities such as football, running, and volleyball, which are very popular and well-suited to the country's tropical climate. However, during my time in France, I was introduced to a variety of sports that were new to me and more adapted to colder climates. I had the opportunity to try skiing and ice skating—two activities that are not commonly practiced in Brazil.

Additionally, I discovered the strong local enthusiasm for rugby in Clermont-Ferrand, the city where I did my internship. Rugby is very popular there and plays an important role in the city's identity. Attending matches and learning about the sport helped me better understand the community and connect with local colleagues and friends.

4.2   Gastronomy

One of the most noticeable cultural differences I experienced was in the area of gastronomy. French meals, especially lunch, often consist of several courses, which was quite different from the simpler meals I was used to in Brazil. I also noticed the widespread use of cheese and butter in many dishes. Cheese, in particular, is a staple of the French diet and is often enjoyed after the main course. I was also very impressed by the quality and variety of products available in French bakeries. The fresh bread, croissants, and other pastries were not only delicious but also a daily part of life, reflecting the importance of artisanal food in French culture.

DDRS

During my internship at the Institut Pascal, I had the opportunity to contribute not only to technological advancement but also to aspects related to sustainable development. The main focus of the project was the automation of robotic systems using ML, in order to reduce the training time of robots and, consequently, the energy consumption associated with long learning periods. This approach, which involves the reuse of human knowledge through TL techniques, allows to train intelligent systems more quickly and efficiently, which reduces the use of computational and energy resources. Also, in the laboratory context, although there was no air-conditioning system installed - only fans, which can be inefficient on very hot days -, I noticed an effort to minimize energy consumption: the lighting of the laboratory was adequate, with low-power LED lamps, and no machine was turned on unnecessarily.

These factors show a concern with sustainability, even if there is still room for improvements in building infrastructure, such as thermal insulation and air conditioning. Thus, the internship contributed both to the advancement of intelligent automation and to practical reflections on how to align technological research with principles of energy efficiency and environmental responsibility.

Conclusion

This chapter presents the final considerations of the work carried out during the internship. It is divided into two parts: the first summarizes the technical results obtained and the next steps for continuing the project; the second highlights the personal and professional development achieved throughout the internship experience.

Conclusion and Future Work

This work focused on the study of TL applied to supervised learning. With this in mind, the research involved the use of the ArUco library and the MediaPipe toolbox to collect a dataset aimed at following a ML pipeline, and to validate TL from a human to a robot. The study centered on the application of supervised learning in the field of robotics and aimed to contribute to the doctoral thesis of Antonio Claudio de Sousa dos Santos Filho.

As a result, the dataset was successfully collected, achieving the primary objective of the internship. The supplementary objectives — namely, the implementation of the TL model and its testing on a real robot — are still in progress. The work will therefore continue until the end of the internship in order to fulfill all proposed goals.

There are several possibilities for continuing this project. With the existing human movement dataset and the creation of additional robot motion datasets, it will be possible to implement new tasks to validate the operation of the TL.

Personal Conclusion

Throughout this internship, I was able to develop several technical skills, including: ROS 2, MediaPipe, ArUco library, supervised learning, TL, programming, object and human motion capture, project management, and technical report writing.

In addition, I improved my soft skills by collaborating in a team environment to achieve results and solve challenges.

Being part of a French research institution such as Institut Pascal provided me with valuable insights into a professional research environment, thanks to the shared culture, teamwork, and the entirely new life experience it offered. This internship has been a meaningful step in my academic and professional development.

Bibliography

[1] Institut Pascal. Institut pascal - chu (tgi). `http://www.institutpascal.uca.fr/index.php/fr/plan-d-acces`. Accessed: 2025-07-04.

[2] Samuel Beaussant. *Transfer learning between robots with state abstraction*. Phd thesis, Université Clermont Auvergne, 2023. ffNNT : ff. fftel-04598434.

[3] Wikipedia contributors. Moravec's paradox. `https://en.wikipedia.org/wiki/Moravec%27s_paradox`, 2025. Accessed: 2025-06-06.

[4] Institut Pascal. Presentation – institut pascal. `http://www.institutpascal.uca.fr/index.php/en/the-institut-pascal/presentation`, 2024. Accessed: 2025-06-06.

[5] OpenAI. Solving rubik's cube with a robot hand. `https://openai.com/index/solving-rubiks-cube/`. Accessed: 2025-07-05.

[6] Mr. Mehdi Mounsif. *Exploration of Teacher-Centered and Task-Centered paradigms for efficient transfer of skills between morphologically distinct robot*. Phd thesis, Université Clermont Auvergne, 2020.

[7] April Robotics Laboratory. Apriltag: Project page. `https://april.eecs.umich.edu/software/apriltag`, 2025. Accessed: 2025-06-11.

[8] April Robotics Laboratory. Apriltag: Official repository. `https://github.com/AprilRobotics/apriltag`, 2025. Accessed: 2025-06-30.

[9] Chili - EPFL. Chilitags: Documentation. `https://chilitags.github.io/`, 2025. Accessed: 2025-06-12.

[10] OpenCV Contributors. Aruco: Official documentation. `https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html`, 2025. Accessed: 2025-06-12.

[11] OpenCV Contributors. Aruco: Original repository. `https://github.com/opencv/opencv_contrib/tree/master/modules/aruco`, 2025. Accessed: 2025-06-12.

[12] Carnegie Mellon Perceptual Computing Lab. Openpose: Project page. `https://cmu-perceptual-computing-lab.github.io/openpose/`, 2025. Accessed: 2025-06-12.

[13] OpenMMLab. Mmpose: Documentation. `https://mmpose.readthedocs.io/en/latest/`, 2025. Accessed: 2025-06-12.

[14] OpenCV Contributors. Opencv: Official repository. `https://github.com/opencv/opencv`, 2025. Accessed: 2025-06-13.

[15] OpenCV Contributors. Opencv: Deep neural networks (dnn) module documentation. `https://docs.opencv.org/4.x/d6/d0f/group__dnn.html`, 2025. Accessed: 2025-06-13.

[16] Google Research. Mediapipe: Project page. `https://mediapipe.dev/`, 2025. Accessed: 2025-06-13.

[17] Google Research. Mediapipe: Official repository. `https://github.com/google/mediapipe`, 2025. Accessed: 2025-06-13.

Appendix A - Check list

Couverture	⊘	Logos, titre, prénoms + noms des étudiants, prénoms+noms des tuteurs, mention « rapport de stage » ou « rapport de projet », année, département ; nom de l'école en entier (Polytech Clermont)
Résumé en francais et en anglais, mots clés	⊘	Résumé (200 mots maximum) + mots clés Abstact + key words Entreprise, problématique, méthode, résultats, analyse des résultats Les 2 résumés sont sur une seule page
Remerciements	⊘	Ordre (en gras ce qui est obligatoire) : **tuteur entreprise**, personnels entreprise, **tuteur école**, enseignants école, autres « Mr. » = Mister ; Monsieur = « M. » Attention à l'accord des participes passés
Sommaire	⊘	2 niveaux de titres
Table des figures et des tableaux	⊘	Toutes les figures et tous les tableaux sont répertoriés. Classement dans l'ordre d'apparition des figures et tableaux dans le texte « Figure 1 : Titre, numéro de page » Toutes les figures et tableaux doivent être numérotées, titrées, sourcées et référencées dans le texte.
Glossaire	⊘	Les mots du domaine présentés en ordre **alphabétique** + définition En fin de lexique : une mention indiquant que « tous les mots suivis d'un astérisque sont définis dans le lexique » Dans le texte : un astérisque à chaque mot défini dans le lexique.
Table des abréviations	⊘	Obligatoire s'il y a utilisation d'abréviations Présentées par ordre alphabétique
Introduction	⊘	Accroche, sujet, problématique, entreprise, enjeux, méthode de travail, annonce du plan
Contexte	⊘	Pour évaluer votre travail, il faut clairement définir d'où vous partez et ce que avez à accomplir (de manière chiffrée, par un cahier des charges par exemple)
Corps du rapport	⊘	Cette partie présente votre travail
Résultats	⊘	Il faut également définir quels sont vos résultats (quel pourcentage du cahier des charges a été réalisé ?).
Différences culturelles	⊘	Présence d'une partie sur les différences culturelles (en fin de rapport, avant la conclusion)
DDRS	⊘	Présence de 2 à 4 pages sur le Développement Durable et Responsabilités Sociétales, comme indiqué dans le document disponible sur l'ENT.
Conclusion	⊘	Rappel de la problématique, rappel synthétique des résultats, distance critique par rapport à ces résultats, ouverture vers un autre sujet ou une autre problématique
Bilan	⊘	Prise de recul, analyse des compétences acquises
Bibliographie Sitographie	⊘	Application des consignes de présentation des items ne pas confondre bibliographie et sitographie (sites webs d'entreprises par exemple) ATTENTION à bien citer dans le texte **toutes** les références
Table des matières	⊘	Tous les niveaux de titres + tous les « objets » du rapport (les tables, les remerciements etc)
Annexes	⊘	Table des annexes (« Annexe 1 : TITRE, numéro de page ») les annexes sont numérotées et classées en ordre d'apparition dans le texte, elles sont TOUTES appelées dans le texte
Fiche du tuteur	⊘	Présence de la fiche de validation du stage par le tuteur. (c'est une annexe)

Appendix B - Report validation form signed by the tutor