POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Digital Twin and Machine Learning solutions for the Manufacturing Environment

Supervisors	Candidate
Prof. Andrea SANNA	Giorgio GIACALONE

July, 2021

Abstract

The advent of Industry 4.0 has brought manufacturing realities to become more flexible and prone to reconfiguration in order to adapt to unexpected events and clients needs. Smart Manufacturing wants to encourage the usage of innovative technologies to promote the digital transformation, especially exploiting the possibilities offered by cyber physical systems and virtual environments (VEs). Digital Twins (DTs) have been widely adopted to virtually reproduce the physical world and to integrate real environments with their digital counterpart.

The development of a DT solution for a production line can be used for monitoring activities, to assess limitations and costs of the real counterpart and to simulate enhancements before implementing possible solutions in the real world. Also, the big amount of data that flow between from the physical assets to their virtual replica can be used to train machine learning systems. Machine Learning (ML) is widely accepted as a relevant technology for Industry 4.0 but it requires large datasets for training. Moreover, most ML methods require labelling, which often has to be manually entered in case of real-world data. DTs can provide a powerful instrument for training ML systems, since a simulation can generate a huge amount of data that can be automatically labelled, thus reducing the user's effort during the training dataset preparation phase.

This work investigates the creation of a Digital Twin of a real production line for assembling skateboards. The proposed use case represents a complex system, which requires both the creation of a VE and the usage of a ML system. The VE has been developed with Unity 3D as an interactive environment that can be experienced through immersive virtual reality for training activities as well as for inspection or analysis activities. The DT of the line has been enhanced with YOLO (You only look once), a state-of-the-art, real-time object detection algorithm deployed on Darknet, an open-source neural network framework written in C and CUDA. The ML system has been trained with a synthetic dataset automatically generated and labelled with Blender. The proposed system allows plant designers to evaluate the benefits of introducing a novel, extremely fast and accurate object detection system on the assembly line. Moreover, the pose detection system performance enables its usage in the VE for real-time users training.

Table of Contents

List of Figures

List of Tables

Acronyms

2D bi-dimensional.

3D three-dimensional.

AI Artificial Intelligence.

ANN Artificial Neural Network.

API Application Programming Interface.

AR Augmented Reality.

CPS Cyber Physical System.

CPU Central Processing Unit.

DT Digital Twin.

FK Forward Kinematics.

FPS Frames Per Second.

GPU Graphics Processing Unit.

GUI Graphical User Interface.

IK Inverse Kinematics.

IoT Internet of Things.

ML Machine Learning.

UI User Interface.

VE Virtual Environment.

VR Virtual Reality.

Glossary

framerate The frequency (rate) at which consecutive images called frames appear on a display.

framework A universal, reusable software environment that provides particular functionality to facilitate the development of software applications.

headset A support framework with attached electronic devices that is worn on the head.

mesh A collection of vertices, edges and faces that defines the shape of a polyhedral object.

rendering The process of generating a photorealistic or non-photorealistic image from a 2D or 3D model by means of a computer program.

shader A type of computer program, running on a GPU, used to calculate rendering effects in 3D scenes.

singleton A software design pattern that restricts the instantiation of a class to one single instance.

synthetic dataset An organized collection of data that is created algorithmically and it is used to train machine learning models.

texture A bitmap image applied to a surface in computer graphics.

virtual reality A simulated immersive experience that can be similar to or completely different from the real world.

Introduction

The process of digitization and the advance of technological progress is bringing more and more technologies in every day life. Since the beginning of the century we are facing a new process of industrialization better known as Industry 4.0. The world of industry has been primarily affected by this revolutionary current because all the industrial processes are transforming to promote more flexibility, productivity and overall quality. The manufacturing sector is moving towards this direction by remodelling their architecture and frameworks to embrace new technologies such as Internet of Things (IoT), Cyber Physical Systems (CPSs), Big Data and Artificial Intelligence (AI). This trend will lead to the construction of a new form of industrial production called Smart Manufacturing.

In this context, the adoption of Digital Twin (DT) solutions is becoming a noteworthy resource to increase the competitiveness of the Manufacturing Environment. In most definitions, a DT model means creating a virtual representation of any production system that interacts with the physical counterpart and provides intelligence for evaluation, optimization, prediction, simulation, etc. DT applications can support manufacturing industries over the entire product lifecycle, from the design stage to the maintenance services. In fact, the virtual environment can be used to improve the requirements, simulate the process, evaluate the performance of the production system before implementing a real prototype. This pattern leads to build a more reliable framework at minor cost, thus fulfilling the goals of Smart Manufacturing.

In addition, DT systems can embrace the employment of various technologies widespread in the context of Industry 4.0; IoT sensors installed on physical objects, for example, can exchange data over the internet and update the virtual counterpart with important information regarding its status or even predict system faults; Virtual Reality (VR) applications can be deployed to offer an immersive experience into the production line and also to train technicians managing every machine; data generated at every stage of the production and exchanged via internet can be used to train Artificial Neural Network;

Artificial Intelligence (AI) applications based on Machine Learning (ML) are drawing the attention of manufacturing sector. They can be used to support manufacturers to facilitate certain task such as the detection of system faults or to help make determinate decision after analyzing previous data. However, ML methods require large volume of training datasets that often have to be manually

labelled in case of real-world data. DTs can provide a powerful instrument for training ML systems, since a simulation can generate a huge amount of data that can be automatically labelled, thus reducing the user's effort during the training dataset preparation phase.

Many example of DT frameworks can be found in literature, but there are few real use cases. Alexopoulos et al. [1] proposed a framework for implementing a DT-driven approach for developing ML models. It explore the possibility to create a synthetic dataset from the DT model of the production line, thus reducing cost and time to prepare a suitable training dataset for ML models. The training dataset is generated automatically and used to train an ANN for vision-based recognition of parts' orientation using simulation of DT models, which in turn can be used for adaptively controlling the production process.

The thesis project

This work starts in collaboration with CIM 4.0 - Competence Industry Manufacturing 4.0 - an Italian organization that aims to provide the strategic and operative support instruments for manufacturing-oriented enterprises toward the digital transformation of industrial process, accordingly to the Industry 4.0 vision.

CIM 4.0 propose itself as an integrated reference point in the technological sectors and industrial areas of the Piedmont region; CIM 4.0 has set up, in its actual place, two different pilot lines or manufacturing demonstration lines that allows to illustrate all the innovative technologies that can be developed and integrated to support the manufacturing companies.

This project's goal is therefore to provide a DT solution for CIM 4.0 Digital Pilot line. The proposed system is a novel use case implementing a DT which is used for both the training of the ML system with synthetic data and for experiencing a digital production line through immersive virtual reality.

The thesis is split into 5 chapters, whose contents are briefly described in the following paragraphs.

In the first chapter, the concept of DT has been analyzed starting from its initial employment and how it has been evolving until now. It contains a description of the applications and benefits for the manufacturing sector, and its correlation with other innovative technologies.

The second chapter will contain an overview of the project requirements, along with a description of the major technologies and tools used throughout the development of this thesis, including the Unity game engine and the ML model adopted.

The third chapter of the thesis will focus entirely on the design process and the development of the DT model using two different software, providing details about how the synthetic dataset has been created to train an Object Detection model and then successfully integrated in the DT application.

Results of this work will be presented in the fourth section, followed by a comparison between the two different software used to build the DT model. This

chapter will also contain an analysis of user feedback and the data that was collected through user tests.

The last chapter of the thesis will explore potential improvements and new features that could be implemented in the future.

Chapter 1

State of the Art

In the recently 10 years, the world of industry has taken a big step forward connectivity and the upgrading of information systems. With the rapid development of new technologies, such as cloud computing, Internet of Things (IoT), Big data analytics, and Artificial Intelligence (AI), the Smart Manufacturing era has arrived. Industrial companies from all over the world, have been using those technologies to increase flexibility in manufacturing, along with mass customization, better quality, and improved productivity. Smart manufacturing plays an important role in Industry 4.0. Typical resources are converted into smart objects so that there are able to collect data, process them and behave within a smart environment. Digital twin (DT) is introduced to develop a smarter manufacturing system with higher efficiency and reliability.

1.1 Industry 4.0

The Industry 4.0 is a new industrial automation trend introduced at the Hanover Fair in 2001. Its announcement had, and still has, a dual objective: to define a development trend in the technological field for companies and systems involved in industrial production, defining its boundaries with a systematic approach; set a goal to be pursued for future innovations in the same area.

The term is meant to imply the advent of the fourth industrial revolution. Just as the first three revolutions have originated from the conjuncture of technological innovation, new market demands and the need to consolidate entire industrial sectors in competition with each other, also the fourth industrial revolution follows this direction.

The First Industrial Revolution is generally considered to be the steam machine and water power; the Second Industrial Revolution is instead seen as the application of electricity to mass production, especially in the new automotive industry; Third Industrial Revolution, also known as Digital Revolution, is mainly linked to extensive use of electronics and information technology to automate production.

The Fourth industrial revolution is about connectivity and big data: industrial

frameworks are evolving from closed system to interconnected systems through the use of internet. It thus enables companies to cope with the challenges of producing increasingly individualized products with a short-lead time to market and higher quality. Today, we are on the cusp of the Fourth industrial revolution and manufacturing industry is in first place affected by this revolutionary current. Industry 4.0 attempts to respond to five main requirements that increasingly stand out as the corner stones for competitiveness among companies worldwide:

- **Increased productivity** thanks to a decrease in set-up times and a reduction in errors and machine down-time.
- **Increased flexibility** in the production of small, possibly customised batches based on customer requirements.
- **Increased speed** in developing new products thanks to rapid prototyping and reconfiguration.
- **Increased quality** through the analysis of the entire production in real time thanks to sensors and data collection processed.
- **Increased competitiveness** thanks to high value products.

1.1.1 Smart Manufacturing

Smart Manufacturing is an emerging form of industrial production integrating manufacturing assets and technologies with sensors, computing platforms, communication technology, control, simulation, modelling and predictive engineering. It utilises the concepts of cyber-physical systems spearheaded by IoT, Cloud computing, AI and data science. Once implemented and widespread, these concepts and technologies would make Smart Manufacturing the hallmark of the next industrial revolution.

Following this paragraph is an overview of the most common technologies covered by Smart Manufacturing (see figure 1.1). However, the main theme that will be discussed is DT solution and its connection with the mentioned technologies.

Cyber-Physical Systems Cyber-Physical Systems (CPS) can be considered as one of the key enabling technologies of the fourth industrial revolution. It can be defined as a framework where all the physical components have their own representation in the digital world, to be integrated with assets with computing, storage and communication capabilities, and to be networked together. This definition introduces to the Digital Twin: physical components, even entire systems, reproduced in the digital world in order to reproduce their behaviour and study their performance.

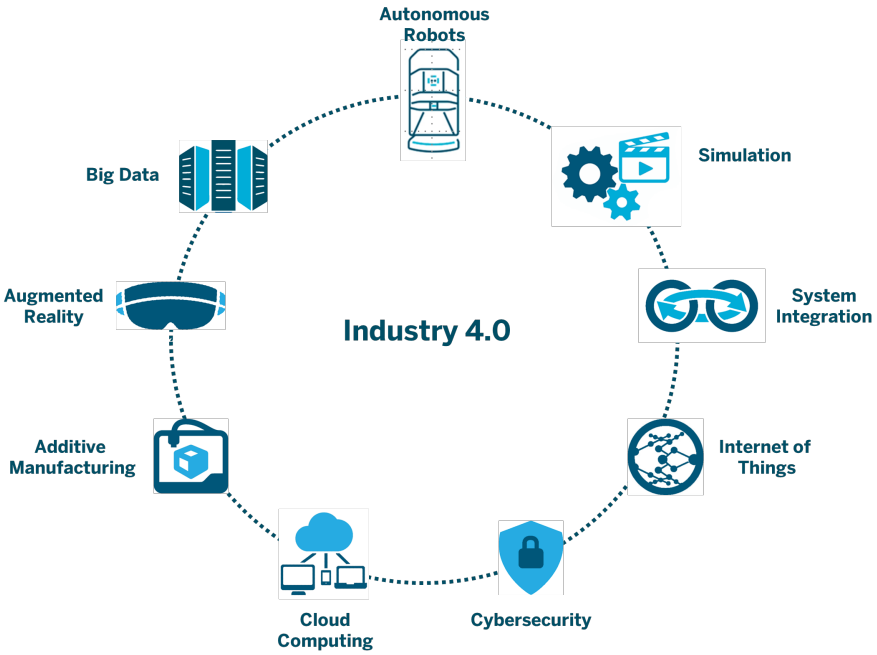


Figure 1.1: Smart Manufacturing enabling technologies and concepts

Internet of Thing The concept of IoT (Internet of Thing) exhaustive fits in the direction of Industry 4.0. Physical objects that are embedded with sensors, software and other technologies create a network for the purpose of interaction and exchanging data over the internet. The data generated is collected, analyzed and cross-processed by computing systems (often installed in a Cloud) which is then redistributed to all those involved in the productive chain to maximise production efficiency.

Big Data and Artificial Intelligence In manufacturing sector, big data can include all the data that are collected at every stage of production, including data from machines, devices, and operator. However, the main problem with big data is that there is too much of it and storing it is not a trivial challenge due to the additional cost. This is where AI and big data can work together. The only way to efficiently deal with this amount of data is to manage it with data-labelling and to use AI software algorithms. In fact, AI becomes better, the more data it is given.

Cloud Computing Connectivity is already a pillar of efficient manufacturing. Cloud Computing aims to the delivery of computing services, such as servers, storage resources, databases, networking, software and intelligence, via the internet ("the Cloud"). Compared to traditional technology that uses individual computers to have updated software for information processing, cloud computing solution offers several advantages such as reliability, cost savings, scalability and centralized management.

1.2 Digital Twin for Smart Manufacturing

One of the most relevant purpose of Smart Manufacturing is providing the necessary tools to build a framework that aims at cost reduction and performance improvements. The simplest way to achieve this goal is taking into account the more parameters as possible, simulate the process and make adjustments until it reaches the desired behaviour. Once the framework is capable of satisfying all the requirements set, it is possible to transfer the acquired knowledge to the real world. Today, it is possible to create this particular workflow by developing a Digital Twin solution.

1.2.1 Concept of Digital Twin

The concept of the digital twin (DT) was first used by NASA to describe a digital replica of physical systems in space maintained for diagnosis and prognosis. During the NASA's Apollo program, they had two real identical space vehicle. One of them was launched to perform the mission, while the other stayed on Earth, allowing engineers to study the performance and mirror the condition of the launched one. DT was then introduced into the aerospace industry and mainly applied to for monitoring and optimize the performance of the airspace vehicles by using a digital mirror model. From 2013, the term Digital Twin has been used for different sectors and it has been linked to various aspects. A detailed overview of the Digital Twin history is depicted in [2]. The initial intended use in literature is to monitor and planning of physical system. Then the focus has shifted to provide virtual representations of systems along their life-cycle. Eventually, DT have been proposed to support decision making through engineering and statistical analyses.

Aside from the field of application the definition of DT is associated to a virtual representation that interacts with the physical object and provides intelligence for evaluation, optimization, prediction, simulation, etc. In particular, the concept of DT was defined by three different elements, including a physical entity, a digital counterpart, and a connection that ties two parts together. In the context of Industry 4.0, being able to build a DT model of a manufacturing product pilot line, fulfils the concept of Smart Manufacturing. In fact, having a smart infrastructure means being able to make smart decision through real-time communication and cooperation with humans, machine, sensors, and so forth. The goal is to achieve flexible, smart, and reconfigurable manufacturing processes in order to address a dynamic and global market.

Nowadays, the advent of technologies such as IoT or Cloud computing can help the development of a DT and in particular the communication layer between the cyber and physical worlds. The huge amount of data that come from each intelligent resource travels through internet and updates the simulation in real time. On-the-fly changes and updates to the manufacturing process can be firstly tested in the virtual environment and subsequently implemented in the real world by evaluating the performances.

1.2.2 Architecture

The architecture for building a basic framework for a DT solution consists of: a) the real world, b) the virtual world and c) the connections of information associating the virtual with the real world, with the DT serving as a digital controller of the real-world manufacturing system.

The real world is recognized as three-dimension physical space that comprehends many entities such as employees, machinery, hardware, and infrastructures with physical features that identify the element (shape, mass, structure, size, etc.). The entities have practical functions and can be organized in an orderly way to complete certain tasks considering constraints on time, cost, quality, etc. Data from the entities are analog data with smoothly varying values.

The virtual world is where the DT truly lives. It is made of a 3D digital mirror model for each entity, which is similar to the physical counterpart not only in geometric expression but also in physical properties and behaviors. The environment build is able to replicates the physical entities and their interaction during their lifetime, featured by real-time synchronization, significant correspondence and high fidelity.

Physical world and its virtual replica in the cyber space are linked through connections made up by a flow of continuous data. On one side, real-time data are collected from sensors of the physical entity and transmitted to the virtual replica for model updating and calibration, and on the other side, significant information generated from virtual simulation is sent back to the physical space to guide and optimize the equivalent physical entity [3]. The DT forms a closed loop from the physical space to the virtual space and back to the physical space again.

Recently, Tao et al.[4] from Beihang University proposed an extended five-dimension definition for the DT. Figure 1.2 illustrates the entities involved in this framework. Compared with the previous basic framework, this innovative architecture incorporates two new entities: the DT data and services entity. The first one combines all the data that are exchanged during the simulation with the intent of more comprehensive and accurate information capture, while the services entity want to explore new functions linked to the DT model (e.g. detection, judgement and prediction).

Although DT concept plays an important role in promoting Smart Manufacturing, its implementation is very complex as it involves several technologies in dynamic evolution. Towards this direction, ISO[1] is developing a standard “Digital Twin Framework for Manufacturing” to provide a generic guideline and a reference architecture for case-specific digital twin implementations (ISO 2020) [5].

[1]ISO: International Organization for Standardization, an international standard-setting body composed of representatives from various national standards organizations.

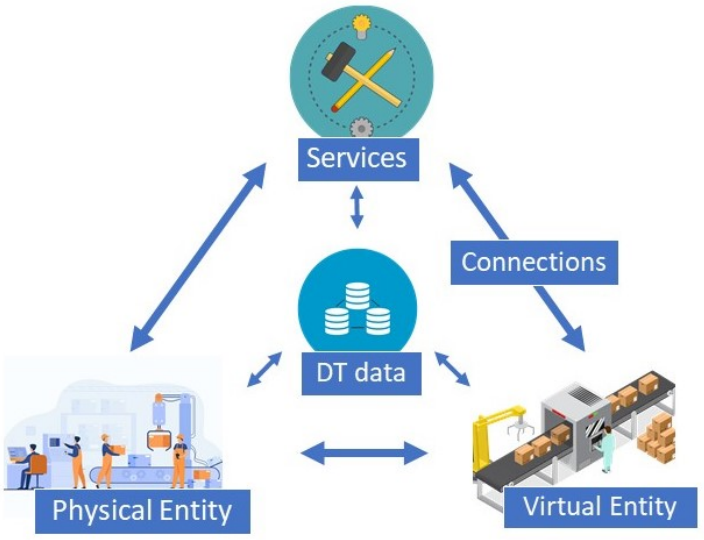


Figure 1.2: Five-Dimensional Digital Twin Architecture

1.2.3 Value of Digital Twin

DT solutions for manufacturing sector can cover various applications such as monitoring, diagnostic, prediction and control. However, the architecture of a DT model itself shows other significant features that makes this solution more influential.

As the market changes rapidly, industrial companies has to deal with market demand and adapt to it in a short time. This leads to become more flexible and prone to reconfiguration. The DT model can provide a chance to develop insights about how a product behaves even before it is completed through the simulation in the cyber world. It is possible to reconfigure the whole behaviour of the industrial framework, eliminate potential failures and adjust the physical resources accordingly. Moreover, as the virtual models in the DT are kept connected to the physical counterparts in the physical world, this connection offers a possibility to analyze how the physical entity performs under different conditions in real-time, and thus makes in-time adjustments to ensure it works exactly as planned.

Another important aspect is related to the reduction of energy consumption and maintenance cost. As the states of the physical asset can be analyzed in real-time, the degraded components can be replaced in a timely manner to avoid additional energy consumption, or for the same intent, it is possible to start or close a determinate machine in a orderly way. In addition, from the analysis of data that flows between real and cyber world, it is possible to predict a breakdown or maintenance event in advance to largely reduce downtime and maintenance costs.

Finally, the DT can fuse different information technologies, including machine learning algorithms, IoT, big data, etc., to accomplish complex tasks. For example, the proposed DT solution integrates an Artificial Neural Network that is capable

to make real-time object detection during the assembly line process.

1.3 Applications of Digital Twin

In the global economy and in global business operations, we have witnessed that there has been a need for Industry 4.0 to dramatically increase the overall level of industrialisation, informatisation and manufacturing digitisation to achieve greater efficiency, competency, and competitiveness [6]. Digital twin solution is a key enable for such purposes, as its common applications includes monitoring, diagnostic, prediction and control. These activities are already used in manufacturing context but in the most cases after the framework has been built. Having a DT model instead permits to simulate, monitor and apply changes the production line in a virtual environment, thus reducing the overall cost and lifting up the efficiency.

This section wants to analyze how DT models can be integrated into the product lifecycle and provide some use cases.

1.3.1 Digital Twin in Product Lifecycle

Manufacturing industries can adopt DT solution during the three principal stages of the entire product lifecycle, which are: design stage, production stage, and service stage. Figure 1.3 illustrates a schematic view of the entire process and the activities linked to each phase.

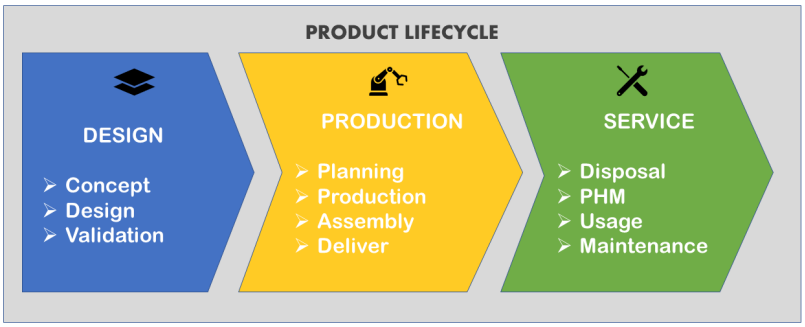


Figure 1.3: Product lifecycle Process

Digital Twin in Design Stage

The initial phase of a product lifecycle is the process of designing parts, components and resources that will be installed in the production line. A general workflow could be the following: a designer makes a sketch of the process, then tries to create some physical prototypes for testing and finally assembles the system. However, using physical prototypes to test the design is very costly.

DT models can overcome this obstacle by simulating the performance of a model in the virtual space. Virtual tests are performed for detecting interference

among various components of the equipment, assessing ergonomics, and prediction equipment. As the simulations and test are conducted only in the virtual space, a DT model in design phase reduces the time for detecting design errors and evaluating the system performance. All the data generated from the simulation can also be used to further improve the requirements and make adjustments in order to improve the reliability of the final product.

Developing a DT framework that reflects and observes the real-world Smart Manufacturing System (SMS) behavior and use it in all design procedures is challenging. Jiewu Leng et al [7] propose an innovative Function-Structure-Behavior-Control-Intelligence-Performance (FSBCIP) framework to review how digital twins technologies are integrated into and promote the SMS (Smart Manufacturing system) design based on a literature search in the Web of Science database. A key feature in this work is the application of the 5D-DT conceptual model and the integration of innovative technologies such as IoT and virtual/augmented reality.

Digital Twin in Production Stage

Production is the key stage of the product lifecycle, it puts on practise all the design choices and it is responsible for the profits and sales of a manufacturing industry. It is a complex process as it involves materials, energy, and information that interact synchronously with each other.

DT models in production stage is used to further enhance the performance and efficiency of the system. IoT and Big Data technologies are the tools to achieve better performance and optimize the production line. In fact, all the data that come from sensor and are sent through internet can be used to update the digital twin that identically behaves like its real counterpart. This allows to control the behavior of the assets and to obtain validated tests through virtual testing. There is also the possibility to evaluate the integration of new automation technologies without disturbing plant output.

S.M. Jeon et al. [8] presents a case study focused on the DT solution to validate the performance optimization of production lines. The virtual model is created using Siemens Tecnomatix Plant Simulation software and it is properly aligned with the real production line by a virtual PLC[2] programmed through Siemens SIMATIC TIA Portal.

As a result, the DT model can offer a bridge to link the physical and virtual space together. It can make the virtual environment mirror the operations of the assembly line in a timely manned and control behaviors of the physical assets in real-time.

[2]PLC: Programmable logic controller, an industrial mainframe used for industrial control systems.

Digital Twin in Service Stage

Service stage comprises all the activities that are adopted to maintain and monitor the health of a product after its disposal. In this context, Prognostics and Health Management (PHM) is the process that performs diagnosis and prognosis, and provides design rules for maintenance strategies. DT solutions can provide useful tools for maintenance and achieve higher efficiency and accuracy for PHM. By simulating the system behavior in the virtual space, it is possible to monitor the health or state of a machine and predict system faults in the early stages of the process. The field of data analytics can provide value to the PHM community by analyzing those large amount of data, enabling maintenance decisions to be made and increasing operational efficiency.

1.3.2 Virtual Commissioning

It has been discussed how design and implementation of a new production solution is often a time-consuming and costly process. After the design phase is finalized and the resources are installed, the next phase is project commissioning. It involves all the procedures to check, inspect and test every operational component of the system: from individual functions up to complex simulation. Also, operators are trained on new equipment, new processes or revised procedures. The main purpose of commissioning is to perform as many test and procedures to guarantee the operability of the production system in terms of performance, reliability and safety. So, it is really challenging to plan and can lead to delays in production and even lost business.

By implementing a DT solution in the early stage of the production, all the tests and procedures can be performed in a simulated virtual environment. This enables:

- Simulation of production line and monitor of the behavior.
- Prediction of system faults or incorrect implementations.
- Simulation of the impact of new machinery on the existing operation.
- Training of supervisors and operators in a virtual reality application.
- Evaluate performances and make adjustments according to the requirements.

Therefore, Digital Twin is a key solution in the direction of Virtual commissioning as it allows manufacturers to monitor and control their installation, as well as to optimize the process - in both startup and the maintenance phase - before implementing in the physical world. As is evident, Virtual commissioning plays an important role in Smart Manufacturing because it contributes the development of more efficient production systems with fewer costs and in a shorter time.

1.4 Digital Twin and Virtual Reality

The concept of Virtual reality has started spreading since the beginning of the 20th century [9]. However, the first real implementations emerged only between 1970 and 1990, thanks to the rapid growth of 3D computer graphics; during these years, the first devices had a rudimentary form and the VR industry mainly provided them for medical, flight simulation and military training purposes. Augmented reality (AR) can be viewed as a variation of VR. In contrast to VR, which creates a completely virtual environment, AR overlays virtual information on the real world to augment the reality rather than replace it. VR and AR technologies have therefore always been showing a lot of potential for transforming many fields of research and engineering, but since the '90s the commercial focus of these devices shifted heavily towards the entertainment industry. It is only in recent years that these tools began to be used in a more widespread way for scientific and engineering applications.

The DT concept and the related architecture aims to the fusion between the physical and virtual worlds. So, the VR and AR technologies can play important roles because their introduction can further enhance the interaction in the virtual (VR) and real (AR) world. Also, it is possible to define new higher quality services in design, planning, guidance and training.

1.4.1 Virtual representation of the manufacturing process

A general workflow is analyzed to efficiently integrate these technologies into the five-dimension DT to provide users with more interactive and immersive services. Some key elements in the assembly process are introduced as follows:

- Physical assembly scene: It contains the real assembly machines, tool, operators, resources, which are organized following a orderly workflow.
- 3D geometric models: All the assets in the real environment have an associated 3D model (CAD or fbx). The virtual environment can be defined using a software characterized by a graphic engine (Unity or Tecnomatix Process Simulate).
- VR and AR devices: Devices such as HMDs, gloves or tracksuit are used to collect data from the physical world and to give the same experience in terms of interaction and sensorial feedback.
- Virtual models, data and services: All the data collected from the VR and AR simulation, together with the 3D models of the involved assets manage to create a visual experience to describe, support and monitor the assembly process. Simulated data from the visual experience and real data from the physical entities can be exploit to optimize the assembly process or to test new features.

Therefore, developing a DT-based solution in combination with VR and AR for an assembly process can clearly increase the performance of the application in terms of human interaction. DT's virtual space becomes immersive and can simulate the physical space through senses and perception, thus realizing full body immersion. The visual experience leads to a more accurate perception of the production line itself and can be used for training purposes. In a similar way, DT's physical space can be featured with AR to show descriptive interfaces or realistic models during the exploration of the industrial site. Given the advantages brought by VR and AR, joining with these technologies will be a prevalent and beneficial trend for the DT.

1.5 Digital Twin and Big Data

The introduction of CPSs, IoT devices, Cloud Computing as novelty technologies in Industry 4.0 has increased the amount of data generated and sent through internet. This huge amount of information is vital for the AI area. In particular, Machine Learning (ML), a subset of AI, is a group of computer techniques that focus on extracting useful knowledge and letting the ML component deciding. This can be obtained via learning or training process where large volume of data are required. Nevertheless, the large amount of data generated everyday during the testing of real machines and systems is raw and not suitable for AI-ML application. In fact, ML approaches, such as supervised Artificial Neural Networks (ANN), are in need of proper quality and quantity dataset and more important those dataset have to be labelled for training purposes. Even though manufacturing companies usually have large amount of data, gathering and cleaning them is a time-consuming process. The process of labelling a dataset is necessary to provide knowledge over all the variables that should be considered by the model. Manual labelling is prone to error and it is labour an time-intensive as the manufacturing environments changes frequently. These limitation may arrest the development of successful AI applications in several manufacturing cases as both time and cost required become an obstacle.

Towards that direction, DT models can be utilized to help the training phase in ML by creating a suitable training datasets as well as by automatic labelling. This is achievable because DT virtual models contains all the information about each single asset (position, orientation, shape, color) and also all the data that is generated can be analyzed and filtered, thus alleviating user's involvement during training.

A framework for implementing a DT-driven approach for developing ML models is presented by [1]. DT models can create a virtual (synthetic) dataset trough simulation tools and utilize it for training ML models in a cost and time-effective way. Section 3.3.1 will describe the process that lead to the creation of a synthetic dataset, using Blender software and then use this dataset to train an ML model.

1.6 Summary

The advent of Industry 4.0, as the expression of the fourth industrial revolution, is encouraging the insertion and development of new technologies in the manufacturing sector with the purpose of increasing flexibility, mass customization, quality and improving productivity. Smart Manufacturing is an emerging form of industrial production that integrates new technologies such as IoT, AI, CPS, Cloud Computing, and Big Data.

In this context, Digital Twin (DT) systems provide a virtual representation of industrial frameworks along their lifecycle that permits to simulate, monitor and evaluate performances even before implementing them in the real world. In the last 10 years, researchers and companies are studying the fields of application for a DT model and its synergism with other emerging tecnologies.

Table 1.1 below shows some of the definitions and developments of the DT appeared in literature.

No.	Ref	Year	Description
1	[10]	2015	DT as a digital controller of a real-world manufacturing system
2	[11]	2015	Investigate the combination of simulation models with real data through a DT to predict future states of the real system
3	[12]	2018	DT framework to optimize the planning and commissioning of production process through simulation
4	[13]	2018	DT applications applied in manufacturing processes to reduce costs and to improve performance
5	[14] [15]	2021	Application of an intelligent DT integrated with ML for bearing anomaly detection
6	[16]	2020	Deployment of a Dynamic DT for system optimizations using ML model
7	[17]	2020	DT technology for LEDs lifetime analysis exploiting ML method for fault detection
8	[1]	2020	Proposed framework for implementing DT models for training ML models
9	[8]	2020	DT solution to validate the performance optimization of production lines
10	[7]	2021	Framework to review how DT technologies are integrated into and promote the SMS (Smart Manufacturing system)

Table 1.1: Applications of Digital Twin in literature

Chapter 2

Technologies and Tools

2.1 Project requirements

It has been discussed how the development of Digital Twin solutions for a manufacturing company is a key enable to achieve greater efficiency, competency an competitiveness as well as to promote the application of new technologies that are spreading in the Smart Manufacturing era. Towards this direction, this work wants to explore those benefits by presenting a DT solution for CIM 4.0 demonstrative production line for assembling skateboards.

The main requirements and goals of this work are:

- **Inspection of the production line:** a Virtual Reality application permits to navigate through a virtual replica of the production line, interact with the assets and inspect the main steps for assembling a skateboard.
- **Training:** as the production line requires the interaction of a technician with some of the assets, the VR experience can be used to train a generic user to correctly use and operate the different technologies and their interaction interfaces.
- **Testing of new features:** a virtual environment permits to test new functionalities and evaluate their performance prior to implementing them in the real world. Here it has been tested how the production workflow can change with the introduction of a Machine Learning model for Object Detection.

2.2 CIM 4.0 Digital Pilot Line

The main goal of this work is to create a DT solution for CIM 4.0's demonstrative production line that is being designed to assemble skateboards.

The physical production line consists of three main assets which are shown in figure 2.1: 1) a Modular Intralogistic Organizer (MIO) by Comau, which operates as an automated warehouse 2) a Virtual Guidance Interactive Learning (Vir.GIL)

Figure 2.1: CIM 4.0 Digital pilot line: The Racer 5 and conveyor belt (bottom right), the production pallet (upper right), the MIO (left) and the Vir.GIL workbench (center)

system by Comau, a complex system which combines together different technologies to provide digital guidance to the user 3) a Racer 5 by Comau, a 6-axis articulated robotic manipulator with a 5 kg payload, designed to ensure both industrial efficiency while providing safe, barrier-free operations.

All the process is managed by a Manufacturing Execution Process (MES), a proprietary software that can control the PLC. It sends the inputs to go through each step of the production line and collects the feedback from each state. The MIO rotates its eight shelves providing, one at a time, all the part required to assemble the skateboard: boards, trucks and wheels of different colors. The Vir.GIL system guides the operator through the preparation of the production pallet: it has a a visual system able to track the operator's position and body orientation; also has a laser pointer and a vocal system that gives hints to suggest where and how to correctly place the components. For the third step of the procedure, the pallet is manually moved by the operator from the Vir.GIL workbench to a conveyor belt where the Racer 5 is installed. It is a collaborative robot so it can co-operate with human interaction. In fact, it is equipped with a sensor that tracks the operator position and adapt its working speed accordingly. After a manual input from the operator, the cobot starts to pick each component and to assembly the skateboard. Since the skateboard's component has been placed to a specific point of the pallet, the cobot knows where it has to find the correct part. A proper visual system is used to facilitate the picking process. It is not able to distinguish among different

component, but it searches for a certain shape in the acquired frame using an image processing algorithm. For example, if it has to pick a wheel it will search for a circle shape on the captured frame, and when precision is above a certain threshold, it is ready to pick the component and place it to the correct position. Finally, after the assembly process is completed the pallet is transported to the end of the conveyor belt, collected by the operator, and placed again to the Vir.GIL station that will guide the operator through the process of fasten the screws and complete the mounting sequence.

During the preparation of the production pallet the Vir.GIL system indicates the operator to place the four wheels with their face up, as it is possible to see in figure 2.1 (upper right). This is realized because the Racer 5 cobot has been programmed for picking the wheels only if they are faced up and it will fail the pick and place task if this particular requirement is not satisfied. To improve this task it has been created a custom object detection module that is able to predict the position of the wheel (up, back or side), thus increasing the flexibility of the cobot's vision system. These new feature has been then tested in the virtual environment to evaluate how the process can be further optimized.

2.3 Blender

Blender [18] is a free open source software that provides all the tools for 3D modelling, rigging, animation, compositing and rendering. It is a cross-platform and runs equally well on Linux, Mac OS X, and Windows on 32-bit and 64-bit platform. The graphical user interface (GUI) is very intuitive, as it resembles an animator's production workflow and also provides a Python application program interface (API) for scripting.

Blender creation suite has been adopted for the creation of a DT application for two main reasons: it allows to create complex 3D models that will represent the virtual replica of physical assets for a generic production line; thanks to its render engine it is possible to generate a syntetic dataset that will be used to train an object detection model. The modelling interface (figure 2.2) includes various editing methods: the most important are *Object Mode* and *Edit Mode*; the former is used to modify a single object properties (e.g. position, rotation) while the latter modifies the geometry of the mesh by adding vertex, faces, and so on. Materials, lights and backgrounds are all defined using a network of shading nodes. These nodes output values, vectors, colors and shaders. The result is being able to reproduce the same appearance of any existent material by creating the so-called PBR[1] materials. Other Blender's feature includes UV unwrapping, texturing, modifiers, sculpting and particle simulation.

[1]PBR: Physically Based Rendering materials allow to simulate almost any existing material with a single unified format.

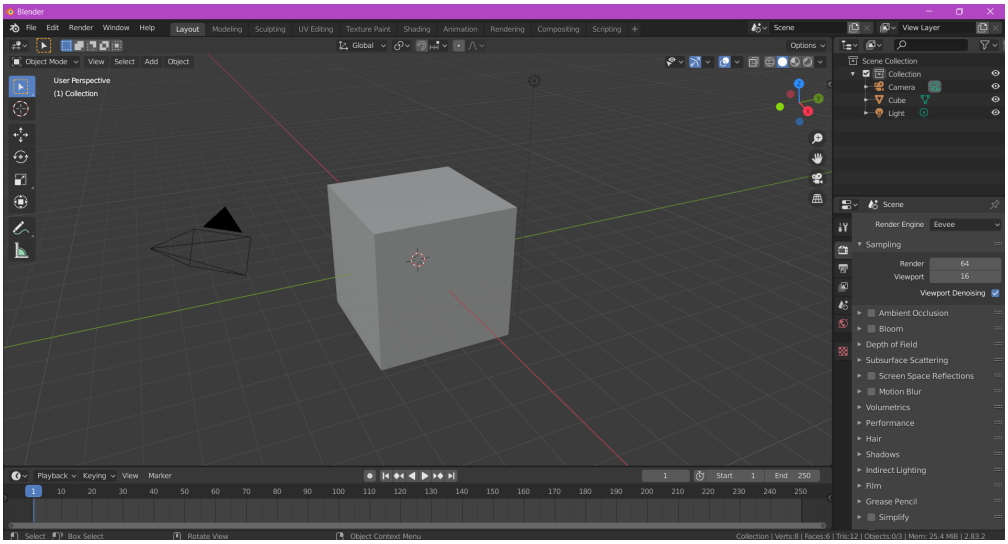


Figure 2.2: Screenshot of the Blender modelling interface

Once the 3D model's appearance is similar to the real counterpart, it might be featured with an animation that wants to replicate the asset's behaviour. In particular, Blender provides the tools to create an animation and attach it to the object. An animation can be created by changing specific parameters over time, such as the object's position, color, material or even its shape. However, if the object that has to be animated is too complex, as for example an human body or a robotic arm, first it is necessary to create a virtual armature and then define constraints for each existent joint. This process is called *rigging* and it is a common technique in computer animation. Section 3.2.1 will describe how the cobot's 3D model has been properly rigged in order to replicate its behaviour and how its animation is being imported in the Unity environment.

Blender version 2.8 introduced a new rendering engine called Eevee. It is a render engine built using OpenGL[2] focused on achieving high speed and interactivity during the 3D pipeline process. Eevee uses a process called *rasterization* via OpenGL 3.3, which estimates the way light interacts with objects and materials using numerous algorithms. Although the performance are better (in terms of speed) the accuracy is lower compared to Cycles, which is a raytrace render engine. Cycles uses NVIDIA's OptiX ray-tracing renderer to deliver more realistic effects like subsurface scattering and motion blur.

The creation of a synthetic dataset can be addressed to a Python script. In fact, Blender has an embedded Python interpreter and provides two modules (bpy and mathutils) that are able to access to every Blender's data, classes and functions. So, it is possible to run a script that takes the camera, select the desired render engine and produces as many rendering as needed for the creation of the synthetic dataset.

[2]OpenGL: cross-platform API for rendering 2D and 3D vector graphics [19]

The selection of the render engine is related to the needs of the ML model that is being trained. Generally, if the renderings needs to be more realist as possible, because the scene consists of transparent objects or global illumination has an high impact, it may be necessary to use Cycles render engine. On the other hand, if the creation of the dataset has to favor performance over accuracy, Eevee render engine could offer a fair tradeoff.

2.4 Unity

Unity [20] is a cross-platform game engine (figure 2.3) that can be used to create 2D, 3D, Virtual Reality and Augmented Reality applications such as games, simulations and other experiences. It combines several tools and libraries to make the development of said applications easier, and provides support for deploying the final product to over 25 different platforms. The engine itself has been written using the C++ programming language, but the development of applications using Unity is done through C# scripting.

The Unity game engine has been chosen for the development of a DT application because it provides a simple and flexible all-in-one framework for creating complex VR interactive experience. Also Unity provides access to an Asset Store, where members of the community can share their own packages (free or paid) which implement custom solution or contain pre-made content that can be easily reused. In addition, 3D models created in other software can be easily imported by converting them into a FBX format. As the main field of application is video games, Unity game engine is suitable to recreate ad-hoc interaction that are typical for the manufacturing process such as grab object, place into the correct place or even simulate the whole process. In addition it is possible to incorporate elements of online games, such as points, leaderboards, and badges into non-game context, in order to improve engagement with both employees and consumers (Gamification [21]).

Each element on the Unity's scene is considered as a separate GameObject with its components such as Transform, Box collider, Mesh rendered, or Rigid Body that are unique. All the interactions between the user and the objects of the environment or between the objects can be configured through a C# script that is attached to an object as one of the other components. Each script creates a connection with the internal workings of Unity by implementing a class which derives from the built-in class called MonoBehaviour. There are two main functions that can be defined in a script: the **Update** function is responsible to handle the frame update for the GameObject. Here it is possible to define all the actions, triggers and behaviours that an object should handle over time during gameplay; the **Start** function instead is called during startup and is responsible of the initialization of local or global variables. An important aspect is the possibility of activate/deactivate any GameObject during the execution of the application itself. In this way it is possible to test functionality or even add other game objects as the application is running,

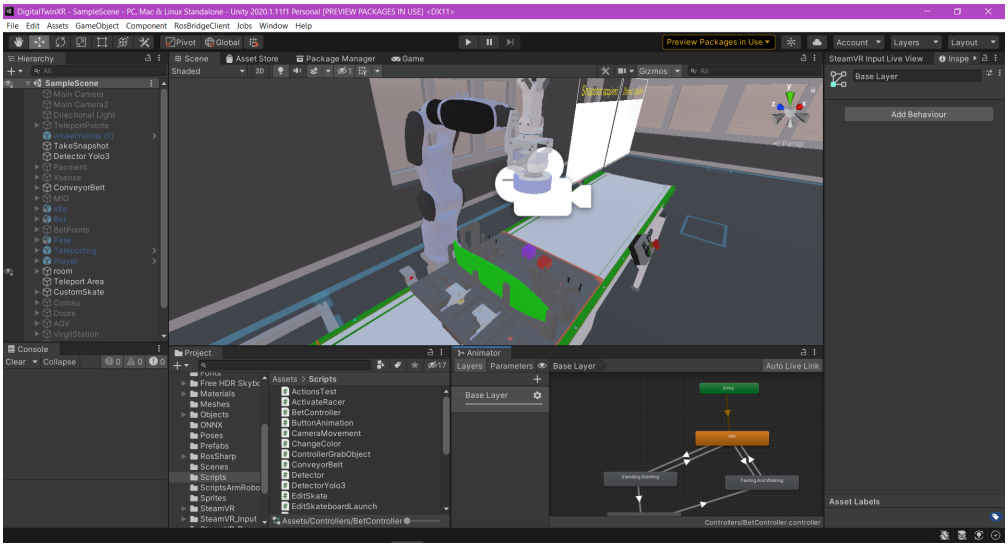


Figure 2.3: Screenshot of the Unity 3D Editor

thus enhancing the flexibility of the application during its developing.

Of course, using a general purpose game engine like Unity to build a DT solution also has some drawbacks: collecting data from IoT sensors and transfer them to the digital model in the virtual environment is not trivial and requires a specific communication protocol; by simulating the behavior of the system it is difficult to retrieve information such as usury of machines or predict when a system fault occurs. However, an immersive VR experience developed in Unity can be used to inspect the production line and for training purposes. Section 3.2.2 will describe the process for developing a DT solution for a real production line using Unity game engine. The VR experience developed focuses on training technicians to correctly interact with certain assets and also to test a possible upgrade to the production vision system by introducing a ML Object Detection.

2.5 Tecnomatix Process Simulate Software

Process Simulate is one of the Tecnomatix application suites proprietary to Siemens company [22]. It is a digital manufacturing solution for manufacturing process verification in a 3D environment. It requires a license server configuration to being installed and can only run on Windows platform. Figure 2.3 shows the GUI of the sofware.

The Process Simulate suite is one of the best solution to develop a complex DT solution for a manufacturing process. In fact, it provide capabilities to design, analyze, simulate and optimize manufacturing processes from the factory level down to lines and work cells. The 3D models of a manufacturing site has to be imported on a CAD format and then it is possible to configure their behavior and interaction with the whole system. Process Simulate provides also other features such as

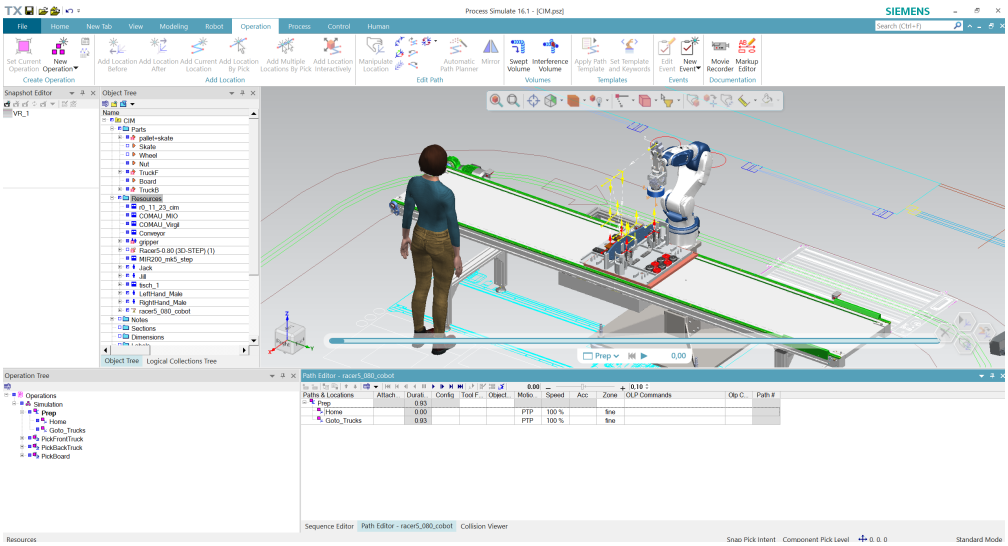


Figure 2.4: Screenshot of the Process Simulate Editor

ergonomics analysis, performance analysis, VR experience, analyze kinematics, robot movements and human motion, as well as determine the optimal tool, robot or device needed for the assembly process.

As a result, developing a DT solution on Process Simulate software for a manufacturing process can provide: detection of design errors at early planning stages; reduction of the number of physical prototypes; increase of overall system performance. However, the usage of the software itself and all the related features is not self-explanatory and requires an ad-hoc formation by Siemens in order to fully exploit the capabilities of the software. In this work, it has been set up a virtual environment on Process Simulate that comprises the main assets of the production line analyzed. It is possible to explore and inspect the production line through an immersive VR application and test the cobot inverse kinematic (IK) solver.

2.6 Darknet framework

The ML system was deployed on Ubuntu using Darknet, an open source neural network framework written in C and CUDA [23]. Darknet is mainly used for Object detection, and have different architecture, features than other deep learning frameworks. It can be used on CPU but it reaches maximum performance on CPU computation. To this end, it is necessary to install the NVIDIA CUDA Toolkit and NVIDIA cuDNN, which are GPU-accelerated libraries of primitives for deep neural networks. In addition, OpenCV library is needed for data augmentation of the dataset during the training phase of the neural network.

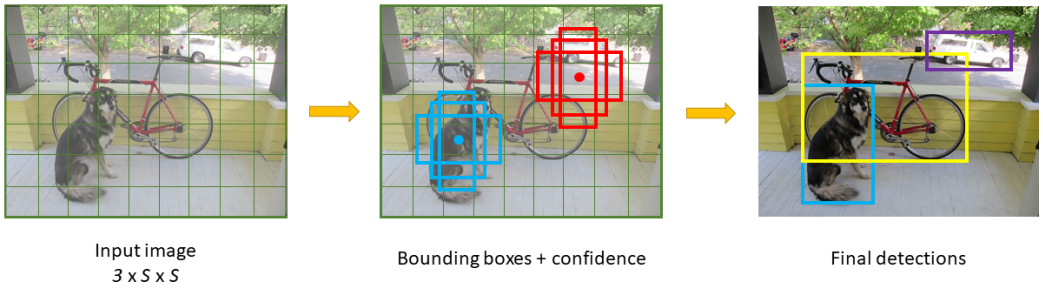


Figure 2.5: YOLO model

2.6.1 YOLOv3 real-time object detection system

The Darknet ANN was deployed to use You Only Look Once (YOLO), a state-of-the-art, real-time object detection system [24]. An Object detection system provides the tools to find all the objects in an image and draws the so-called **bounding boxes**, which is an area defined to surround an object. YOLO algorithm is based on *regression*, a technique of image processing that predicts classes and bounding boxes for the whole image in one run of the algorithm. This is a common approach for real-time object detection as, in general, trades a bit of accuracy for large improvements in speed.

Figure 2.4 shows the steps of the YOLO algorithm for object detection. As input YOLO requires an image $3(RGB) \times 416px \times 416px$. The model segments an image into a S x S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. The bounding boxes is composed with 5 values: (dx, dy, dw, dh, c): dx and dy are the coordinates of the bounding box center relative to the grid cell; dw and dh are respectively the width and the height of the bounding box; c is the confidence value than an object exist within the bounding box. These prediction are then encoded as an $S \times S \times (B * 5 + C)$ tensor[3]. Most of these cells and bounding boxes will not contain an object. Therefore, the model predicts the value c, which serves to remove boxes with low object probability and bounding boxes with the highest shared area in a process called non-max suppression. Finally, the bounding boxes with higher confidence are drawn into the original image.

[3]Tensor: container that store data in N-dimensions.

Chapter 3

Design and Development

3.1 Architecture

The main goal of this work, as already mentioned, is to create a Digital Twin solution for CIM 4.0 demonstrative production line for assembling skateboards. The proposed DT was developed and deployed on a workstation equipped with an Intel Core i7 CPU, an NVIDIA GeForce Quadro 4000 Graphic Card and 128 GB RAM with dual boot for either Windows 10 or Ubuntu 20.04 LTS. Figure 3.1 shows the software layers. The DT was developed in Unity 3D on Windows: the SteamVR plugin enables the application to run on the HTC VIVE Pro (2018) immersive Virtual Reality headset, whereas the Barracuda plugin is used to integrate the Neural Network, exported in the Open Neural Network Exchange (ONNX) format, into Unity 3D. The ML system was deployed on Ubuntu using Darknet, an open source neural network framework written in C and CUDA. The Darknet neural network was deployed to use You Only Look Once (YOLO), a state-of-the-art, real-time object detection system, to recognize the pose of the wheels for the pick and place task. To this end, it was necessary to install the NVIDIA Cuda Toolkit, the Cuda Deep Neural Network library and the OpenCV library. The synthetic dataset used to train the Neural Network was generated through Blender, an open source 3D modeling software.

3.2 Unity DT application

The DT model had to provide the tools to navigate and inspect the industrial production line, to train a generic user to correctly use each component of the system and finally to test new features in the virtual environment. Towards this direction, it has been developed a complex VR interactive experience using Unity game engine.

Figure 3.1: Software layer overview

The overall design of the application can be subdivided into three main phases:

1. **Asset Modelling:** all the 3D models that are involved in the production system have been created using Blender, featured with animations to represent their behavior

2. **Design of the Virtual Environment:** set up of a Unity scene and the integration with SteamVR plugin that enables the application to run on the HTC VIVE Pro, immersive Virtual Reality headset

3. **Interaction design:** definition of the application's flow of execution and C# scripts for implementing interactions between the user and the environment

3.2.1 Asset Modelling

The first phase takes into account the generation of a virtual replica for each physical component of the production line. Most of the component can be retrieved from the vendor's web site, where it is possible to download the specific CAD file. Unity assets needs to be a FBX format and the conversion is done by most of the

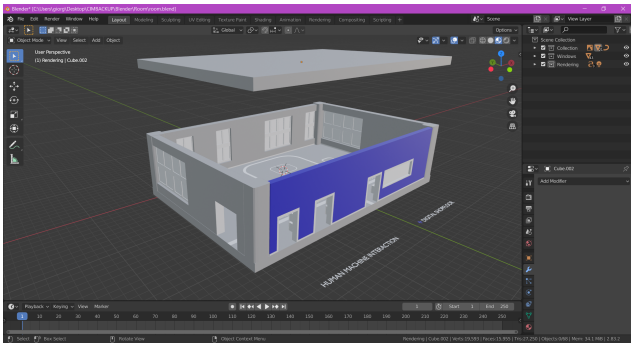


Figure 3.2: 3D model of the production line site

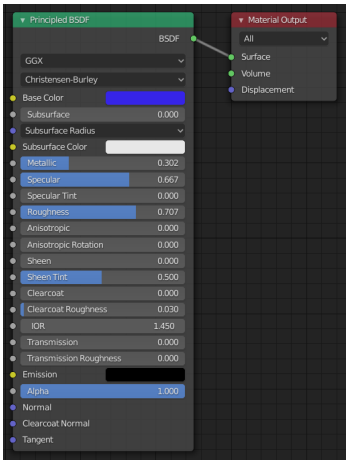


Figure 3.3: Shader editor in Blender

3D modelling software. However, some vendors do not provide the model for some resources. It is possible to overcome this lack by reproducing the model in a 3D modelling software like Blender. It provides all the tools to create high-detailed 3D model starting from primitive mesh shapes such as: plane, cube, circle, UV sphere, etc. With the combination of the two main editing method: *Object mode* and *Edit mode* it is possible to modify the object properties (shape, rotation) and the geometry of the mesh to accomplish a digital replica of the desired asset.

The first 3D model created was the site of the production line (figure 3.2). In order to preserve the real scale of the object it has been imported a 2D map of the real site as a background image. Also, Blender has a functionality in *Edit mode* that shows the length of each edge of the object in meters, useful to have an indication about the dimensions in real scale.

After the modelling phase it is necessary to define a material for each component of the 3D model. Blender provides a network of shading nodes to produce materials using the Shader Editor (figure 3.3). The Principled BSDF node combines multiple layers into a single easy to use node. By tuning parameters some of the parameter on this node (Specular, Roughness, IOR, Transmission) it is possible to create PBR materials, thus produce a specific material which is similar to the real ones. Also, combining "Image texture" nodes with Principled BSDF node can attach a specific texture to an object. Generally the level of detail (LOD) of the 3D model together with the generation of a specific material, has to be compliant with quality of visual experience it wants to offer. The more realistic is the 3D model created, the more immersive the VR experience will be.

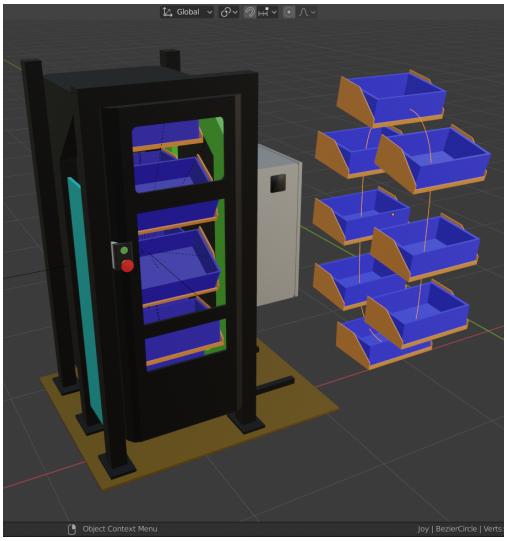


Figure 3.4: 3D model of MIO

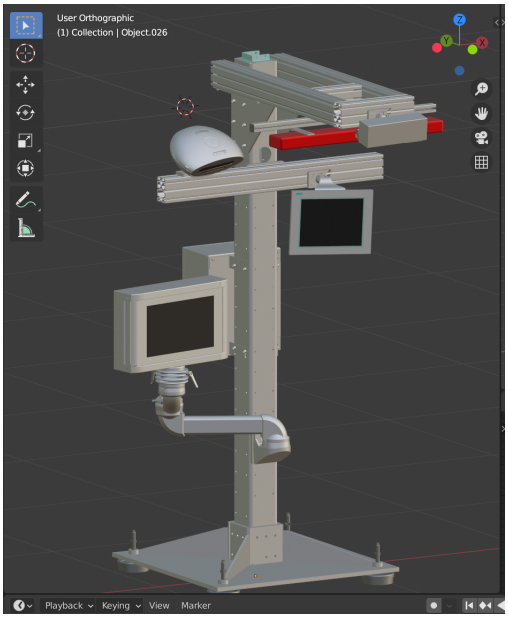


Figure 3.5: 3D model of Vir.GIL

The physical production line consists of three main assets (figure 2.1): 1) a Modular Intralogistic Organizer (MIO) by Comau, which operates as an automated warehouse 2) a Virtual Guidance Interactive Learning (Vir.GIL) system by Comau 3) a Racer 5 by Comau together with a conveyor belt. Only the MIO has been created completely in Blender, the other models has been retrieved from vendor website and converted from CAD file to FBX format. As both the MIO and Racer 5 perform always the same task, their 3D models have been featured with a custom animation using Blender.

The MIO has eight shelves that rotates into a circular shape and provides, one at a time, all the components that are needed for assembling a skateboard (board, trucks and wheels). The animation created is simple: each shelf has an *Object Constraint* to an unique Bezier circle, used as target but with different offset (figure 3.4). As a result each shelf will follow the Bezier circle, thus fulfilling its real behavior. The bezier circle is then animated for 100 frames. Also each single shelf, during its circular path, has been featured with four different keyframes that are needed to reproduce its rotation as soon as it approaches to the bottom and the top of the MIO's structure.

Simulate the robot's behaviour is instead more complex, as it is made of six different joints and each one of them has its own kinematic. The solution explored is to provide a rig for the robot. The rigging process attaches an armature to the robot, composed of 6 bones (figure 3.6). Each one of the bones has a *Bone Constraint* of Limit rotation accordingly to its technical specifications provided by Comau web site [25] and it controls a specific part of the robot. The robot

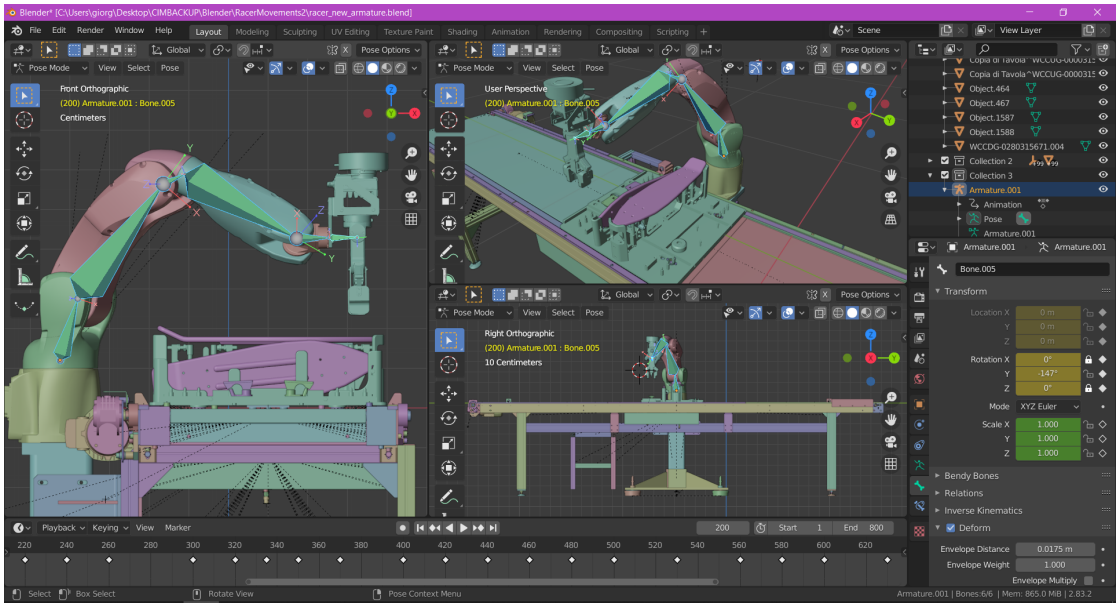


Figure 3.6: Blender interface for 3D modelling of Racer 5

was animated using Forward Kinematics (FK), thus each single bone has been manipulated individually based on parent-child relationships. Every time the robot assumes a different pose a new keyframe is created and the armature is modified to achieve the desired robot pose. This solution comes from the lack of the IK (Inverse Kinematics) solver of the Racer 5, so the movements of the robot will not adapt dynamically to the position of the components in the production pallet. The animation will instead simulate the exact behaviour of the robot in the production process.

Vir.GIL's behavior cannot be resolved with an animation. It is a digital assistant that combines speech and gesture to guide the worker or teach itself new sequences for new tasks. Its behaviour will be implemented directly into the Unity application (see section 3.2.3).

3.2.2 Design of the Virtual Environment

The next phase for the creation of the DT model involves the setup of the Virtual Environment (VE). The Unity project of the proposed application consist of a single scene, which contains all the 3D models that have been created in the previously step exported in a FBX format. Every model imported in the Unity scene is displayed in the Hierarchy and handled as a GameObject (figure 2.3). The *Inspector* panel on the right of the GUI is used to view and edit properties and setting for all the GameObjects in the Unity editor, but also assets, materials and in-Editor settings and preferences. The VE is then prepared using the Editor tools that Unity provides such as Rotate, Move, and Scale Tool that operates for the single GameObject.

Once the scene is ready and all the components of the production line are at the correct position, it was necessary to download and install the SteamVR plugin from the Unity Asset store. The SteamVR plugin enables the application to run on the HTC VIVE Pro (2018) immersive Virtual Reality headset from VALVE company. This plugin provides to developers an integration for a variety of controllers and input devices in a VR application. A proper window (see figure 3.7) will load a default SteamVR Input JSON[1] file that defines the actions and bindings for the VR controllers. It is possible to customize these bindings and create a new configuration using the binding UI and then save them to generate a new JSON file. Also, SteamVR plugin provides virtual examples of interacting with the virtual environments along with prefabs and scripts to develop VR applications. The prefabs employed in the DT application are: *Player* and *Teleporting*. The first one identifies the user of the VR experience. It has attached a script that implement a singleton class representing the local VR player with methods for getting the player's hand, head and tracking origin. The *Teleporting* prefab binds a button of the controller to the action of virtually teleport from one point of the room to another. Teleporting is allowed only in a certain area that can be defined using a plane GameObject and assign the *TeleportingArea* script to it. When the

[1]JSON: JavaScript Object Notation is a file format suitable for data interchange

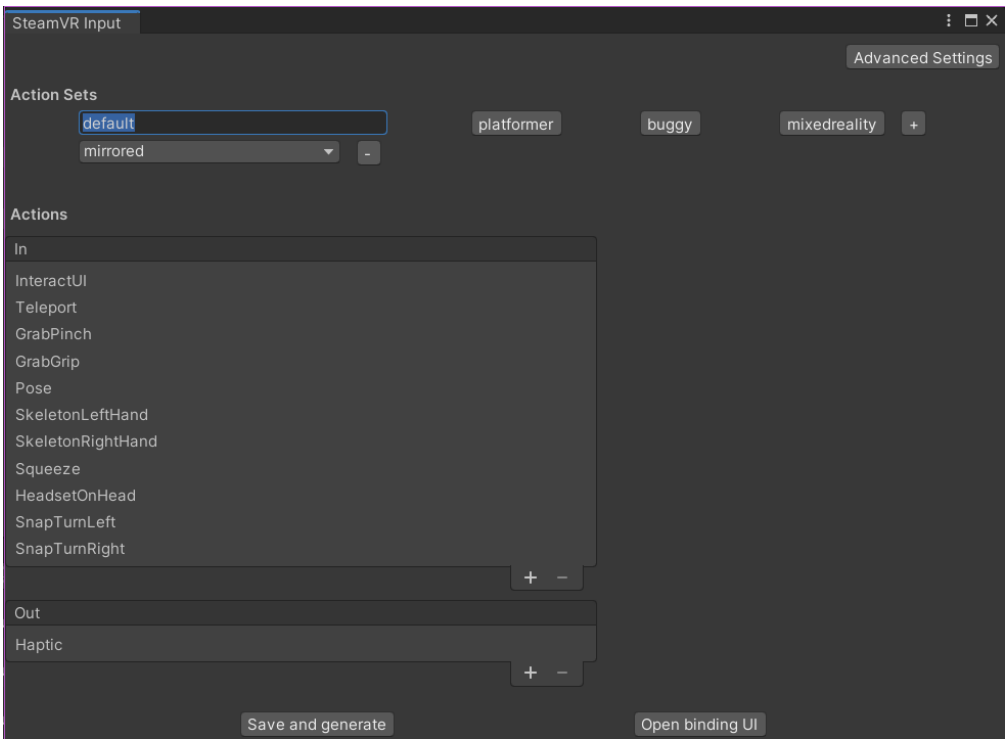


Figure 3.7: SteamVR plugin settings inside Unity

teleport button is pressed it shows a parabolic shape laser that points to the new destination. If the laser color turns from green to red color, it indicated that the pointed destination is outside the room or an invalid position.

3.2.3 Interaction Design

So far, the user/player of the VR experience is capable of navigating through the site of CIM 4.0 production line, inspect each component and teleport from one point to another. The next step for the development of a DT model is giving the user the opportunity to interact with the virtual environment and the resources available.

The virtual environment has been organized into three different area that simulates the workflow of the production line: 1) customization of the skateboard 2) preparation of the production pallet 3) assembly of the skateboard .

Customization of the skateboard

To begin with, the first task that the user has to perform is to customize the skateboard accordingly to its preferences.

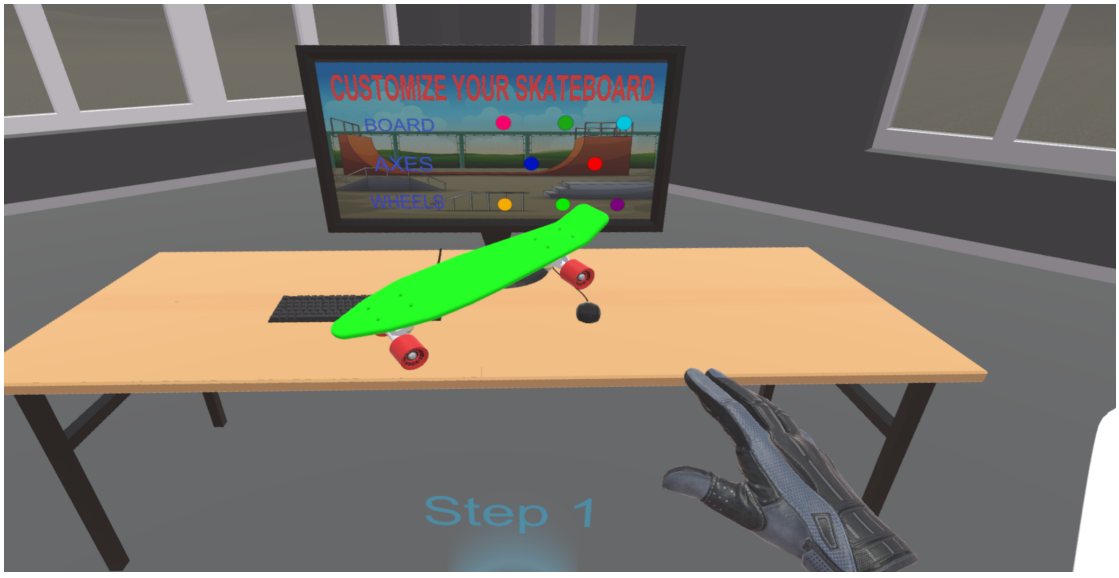


Figure 3.8: Screenshot of the VR application during the customization of the skateboard

Figure 3.8 shows the point of view of the user while performing the first step. It is possible to grab the skateboard GameObject with one hand and use the other hand to push certain buttons that dynamically change the color of each component of the skateboard. The action of grabbing a GameObject is possible thanks to a script attached to the object called *Interactable* (see figure 3.10), provided by SteamVR plugin. At run-time, when the hand move close to any GameObject with

an *Interactable* component, the script will highlight that it is possible to perform a *grab* action. If the user triggers the corresponding button on the controller that is binded with *grab* action, the object will be picked up and then dropped once the controller button is released. Also, in order to simulate the physic of throwing an object, two other scripts are being used: *Throwable* and *Velocity Estimator*.

The action of pushing a button is another important interaction that has been implemented. Each button is a different GameObject with an *Interactable* script attached to it, as it is triggered likewise the grabbing action. But it also has a *UI Element* script (see figure 3.9). This script will run a specific function to be called when the game engine identify a *grab* action for that particular button. Here, when the button is pressed it will call a proper function called *changeColor* with a parameter that indicates the color that will be given to the object.

Preparation of the production pallet

During the preparation of the production pallet there are two entities involved: the MIO and the Vir.GIL. The first one is an automated warehouse that contains all the components needed to assembly the skateboard in different coloration, organized in eight different shelves; the latter is instead a digital assistant that combines speech and gesture to guide the worker and to teach him new sequences of operations to resolve specific tasks. Its goal in this specific production line is to guide an operator to take the skateboard components into the MIO and place them to a particular point of the production pallet.

The same behavior has been recreated in Unity with a combination of two

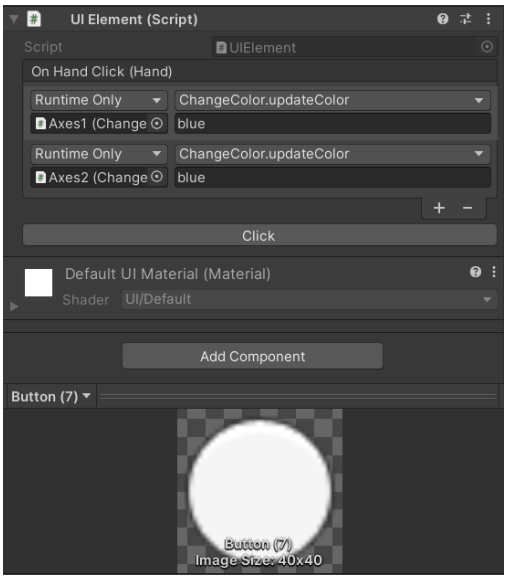


Figure 3.9: Editor settings of an interactable button

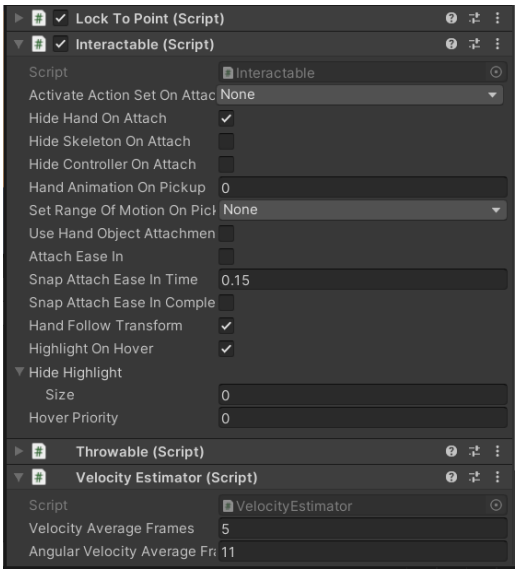


Figure 3.10: Editor settings of an interactable object

different visual feedbacks (see figure 3.11). The first one are UI windows that pop up and describe with a text the operation to accomplish, while the other are bouncing arrows that point to the exact position where the action has to be performed. UI windows are realized with Unity Canvas GameObject, one for each step of the process. A Canvas object is shown as a rectangle in the Scene View of Unity and UI elements such as *Text* or *Image* must be children of such a Canvas. In order to define an ordered sequence of steps, UI windows and arrows are controlled by a specific script and the user can switch from one step to another using a proper button triggered with the *grab* gesture.

The sequence of operation that has to be performed by the user to correctly prepare the production pallet are the following:

1. Pick a board from the MIO and place it in the green area
2. Pick two supports from the MIO and place them in the green area
3. Pick four wheels from the MIO and place them in the green area
4. Pick four screws from the boxes and place them in the green area

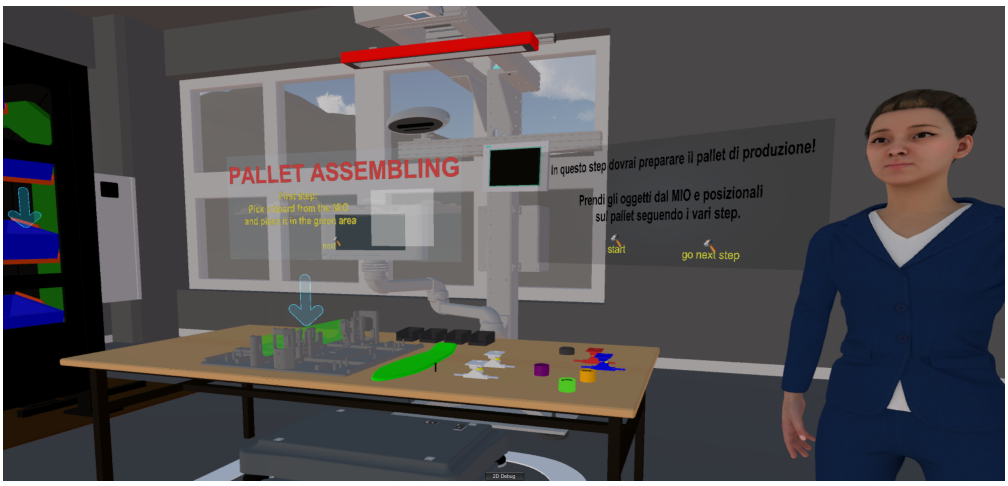


Figure 3.11: Screenshot of the VR application during the preparation of the pallet

The green area, is a 3D cube with a transparent green material that helps the user to identify the location where to put the object. As for the bouncing arrows, these object are shown and hidden by the script that controls the steps for the preparation of the production pallet.

The behavior of the MIO have been developed in Blender with a custom animation that simulate the rotation of its eight shelves (see section 3.2.1). The animation is handled in Unity with the creation of an *Animation controller* (see figure 3.13). This asset manages various animation clips of a specific GameObject (as for the MIO) using a State Machine Diagram, which describes the sequence of

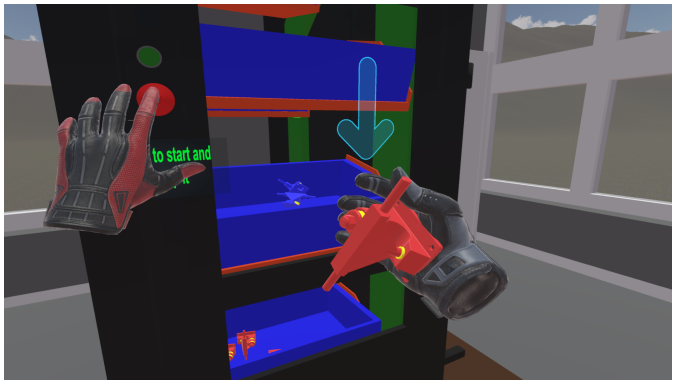


Figure 3.12: Interaction with the MIO asset seen from the VR application

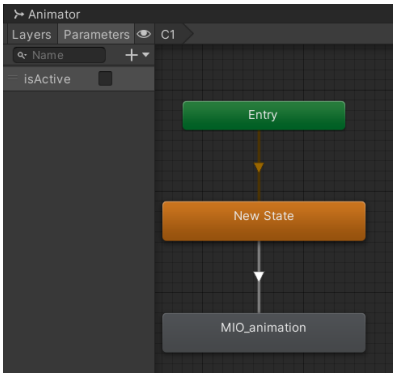


Figure 3.13: Animation controller for the MIO

events that an object goes through during its lifetime in response to events. The event that triggers the start of the MIO's animation is the pushing of the red button, placed in the front of the MIO object (see figure 3.12). When the button is pushed, a script will access to the *Animation Controller* and by setting a boolean parameter (*isActive*) it will start the animation of the MIO. The subsequent pressure of the same button will stop the animation.

Assembly of the skateboard

In the final phase of the production line the production pallet is moved to a conveyor belt and shifted beneath the Racer 5 collaborative robot, that is responsible of assembling the skateboard. The 3D model imported in Unity has been animated to simulate the cobot movements (see section 3.2.1). So, the GameObject has an Animation Controller, similar to the one implemented for the MIO, that will start and stop the animation as soon as the user pushes the green or red button on the conveyor belt (see figure 3.14). A collaborative robot adjusts its speed depending on the operator distance from the operative area. The same function is realized in Unity with a proper script. A parallelepiped is added into the scene for the corresponding operative area of the cobot. When the game engine detects a collision between the user position (Transform component) and the parallelepiped, it adjusts the animation's speed to a reasonable value.

The animation realized in Blender cannot pick the object in the Unity environment but only simulate the cobot movement. The pick and place operations has been implemented by exploiting the *Fixed Joint* component, that restricts an object's movement to be dependent upon another object. Therefore, all the objects that are needed to be picked up by the cobot's arm (wheels, trucks and board) have a *Fixed Joint* component along with a proper *Box collider*. When the animation starts the field "Connected Body" of the *Fixed joint* component is empty, but as soon as the game engine detects a collision between the cobot's arm

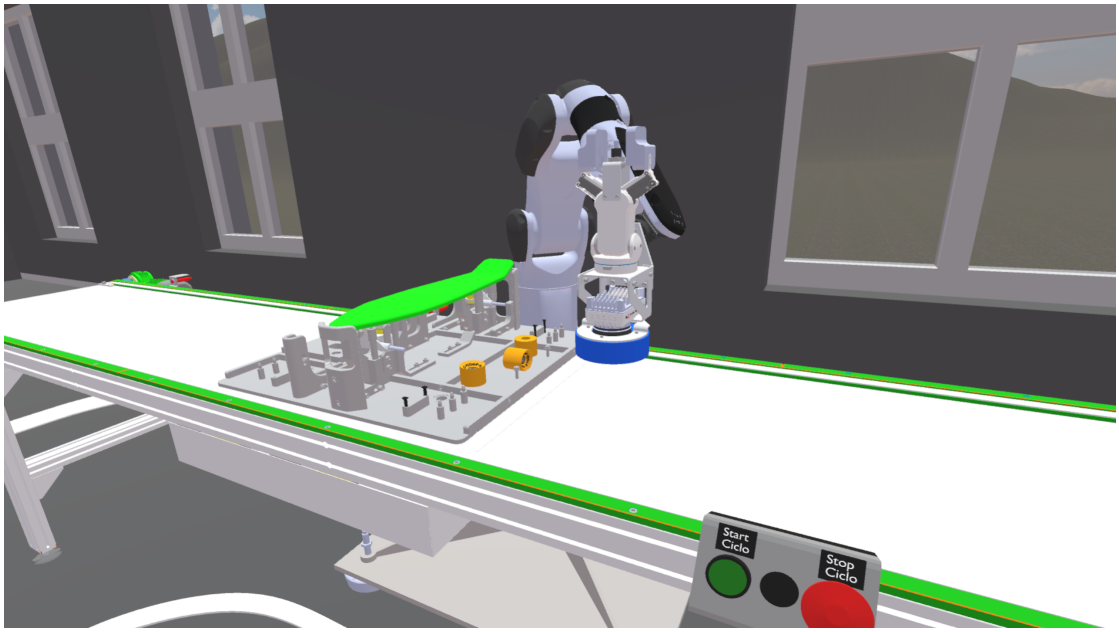


Figure 3.14: Screenshot of the VR application during assembly phase

and the object to be picked up (e.g. a truck), a proper script will fill that empty field with the *Rigid body* of the cobot GameObject. As a result that object will follow the movement of the cobot's arm as it is parented to it. In addition when the truck reaches the point where it should be released, the script will delete the "Connected Body" field and the object will be no more parented to the cobot's arm. Another script (Lock To Point) is responsible to snap the object to the correct location; at run time the script computes the distance of the object from the target location and if the distance is below a certain threshold the object will be snapped accordingly.

The solution implemented to simulate the cobot's behaviour has some limitation as it is not flexible and it relies on finding the skateboard component always in the same location. An upgrade of this solution can be to simulate an IK solver in Unity so that the cobot's arm will adjust his movement accordingly to the component's position. However, such a solution is not trivial and should exploit external libraries such as ROS or MoveIt.

Finally, the application has been featured with a humanoid avatar as a virtual assistant. The goal is to guide the user through each step of the production line, giving also detailed information about the components. The avatar model and the animations have been retrieved from Mixamo [26] a digital library of 3D characters and animations provide by Adobe. The avatar is able to walk into the virtual space and to vocally describe by a synthetic voice the steps of the assembly process. A script controls the avatar behavior, activating the right animation and reproducing the correct sentence at the right time. In this case, the Animator Controller is more complex (see figure 3.16) because the avatar has to switch between four different

states: idle, greeting, talking and walking. Along with the audio feedback, there are also UI windows that pop up describing the task that needs to be done (see figure 3.15).

Figure 3.15: Virtual avatar in the VR application

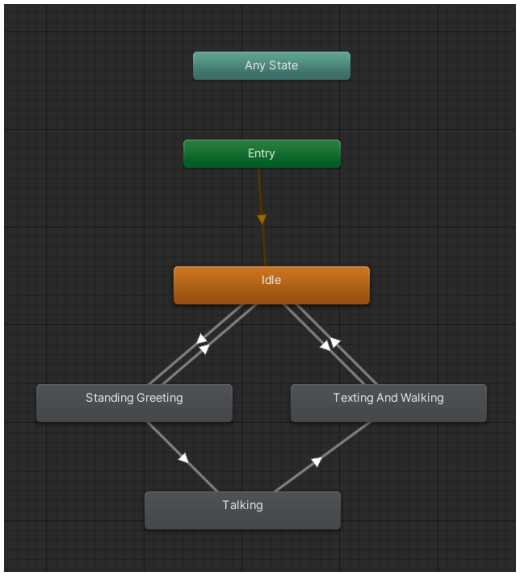


Figure 3.16: Animation controller for the virtual avatar

3.3 Object Detection model

DT models can greatly benefit production lines when it comes to testing new functionalities before implementing them in the real-world environment. Therefore the DT solution proposed in this work want to investigate new features and upgrades for the physical assembly line.

The last phase of CIM 4.0 production line involves the pick and place task performed by the Racer 5 collaborative robot to assembly the final skateboard. Before picking each component from the production pallet, the current vision system mounted on the cobot tries to detect the position searching their shape in the framed image. However, the system cannot distinguish if the wheels are faced up, down or they lean sideways. In addition, the Vir.GIL does not integrate a vision system able to detect the correctness of the operations performed by the user. As a result, when the pallet is placed under the cobot the wheels may be on the wrong side. To improve the pick and place task and make it flexible to this type of errors, an Object Detection module has been deployed to correctly detect the orientation of the wheels.

The following sections describes all the steps performed to create a synthetic

dataset that will be used to train a custom YOLO Object Detection module and finally how to integrate this system into the DT application.

3.3.1 Synthetic Dataset Creation

Machine Learning methods usually requires large volumes of quality training images to perform well. Synthetic datasets are an emerging trend in the context of supervised ML because the process of generating and labelling images is faster and more reliable. The dataset used for this work was generated with Blender, using the wheel 3D digital model created for the DT.

The object detection model has been thought to be deployed on the Racer 5 vision system, so the Blender scene is composed of the production pallet and the wheels. Figure 3.17 shows the corresponding Blender project: on the right the 3D Viewport containing the assets, while on the left it is shown the Text Editor to load and modify Python scripts. The selection of material and the lightning are important parameters in setting up the rendering simulation and should simulate the same condition of the real environment. In fact, the renderings produced during this phase will be used to train the Object Detection module so photorealistic pictures of the real setting will make the vision system more reliable. Also, for the same reason the resolution and focal length of the virtual camera has been set up to resemble the behaviour of the real camera mounted on the robotic system.

The *Training Dataset* is generated by a proper script. YOLO should be able to recognize three classes: wheel top, wheel back and wheel side. Thus, the script creates 150 renderings for each class, for a total of 450 images with resolution 960 x 540 pixels (two samples are depicted in figure 3.18). To make the detection system more reliable and flexible, variations have been introduced for each class in term of colors of the wheels and cameras perspective (orthographic and perspective). In fact, Blender provides Python APIs to access every scene's object and change their properties such as position, orientation and even the associated material. Also, the dataset is not composed by only images; the supervised ML algorithm needs annotations for each image of the dataset that indicate where each object is located in the image and provide a class label for each object. The script exploits a proper function (see code in Appendix A.1.1) that returns camera space bounding boxes of the mesh object. The bounding box is expressed as combination of four values: the x-axis and y-axis position coordinates and the width and height for each object detected in a given frame. These values, along with the label for the class (integer value from 0 to 3) are written in a 'txt' file corresponding for the rendering analyzed.

The created *Training Dataset* is then used as input for the YOLO Object Detection system.

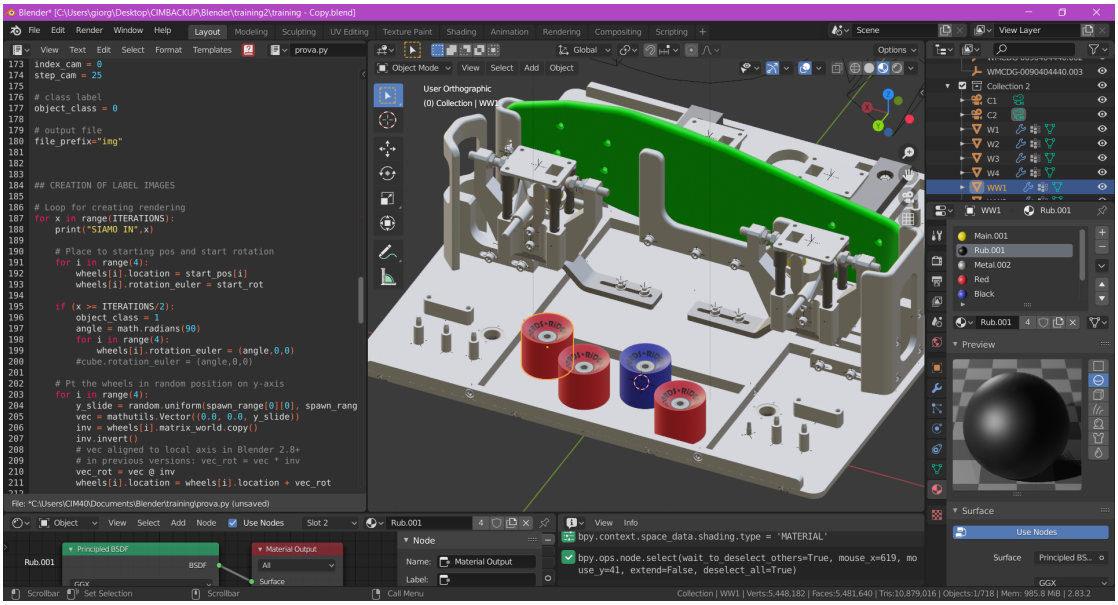


Figure 3.17: Blender scene for the creation of the synthetic dataset

Figure 3.18: Samples from the training dataset

3.3.2 Neural Network Training and Testing

The training phase requires the definition of three important files:

- **Configuration file**: contains the definition of the ANN properties such as the batch size, the subdivisions and the number of epochs, filters and classes.
- **Data file**: needed to define the folder path of the dataset and to specify the path for saving new weights
- **Pre-trained weights:** this file contains the initial weights for the YOLOv3 architecture.

The Darknet provides some guideline for the definition of the configuration file. YOLO has to predict between three different classes: wheel-top, wheel-back and

wheel-side. Consequently, the number of epochs for the training phase has been set to 6000 ($2000 \times number of classes$), while the number of filter for the convolutional layers is set to 24, following the expression: $(number of classes + 5) \times 3$.

The batch size represents the number of samples processed before the model is updated, while the subdivisions are how many mini batches the model splits the batch in. The value of batch size (64) and subdivisions (8) parameters have been fine-tuned through several tests on the specific hardware to achieve better performances. So, YOLO will load 64 images per interaction, split the batch into 8 "mini-batches" and sent them to the GPU for process.

Also, YOLO performs a data augmentation step which enlarges the training dataset by a large number of variations obtained through cropping, scaling and other visual artifacts and transformations applied to the original images.

The training process is started through a shell prompt and it takes about twelve hours to complete using the cuDNN acceleration for NVIDIA GeForce Quadro 4000 graphic card. Once the training phase was completed, the model was tested with both images and a video of the real production line. The video was provided as a parameter to the YOLO system via a shell prompt. The YOLO system is able to grab each frame of the video, make the prediction and show the result in real-time as a new, labelled video. Figure 3.19 shows an example of real-time detection on the real production pallet.

Figure 3.19: Detection sample of YOLO model for a real image

The accuracy of the YOLO object detector is estimated by calculating the mean average precision (mAP) during the training phase. The mAP compares the ground-truth bounding box to the detected box and returns a score. The higher the score, the more accurate the model is in its detection. For this purpose, it has been set up a *Validation Dataset* of 55 images taken from the real world and manually labelled using YOLO mark [27]. Every 4 epoch of the training phase the average precision is being calculated using images from validation dataset.

Figure 3.20 shows the loss function in blue and mAP trend in red. As it is evident, loss function is very high at the very beginning and rapidly goes down as the model approaches to its true value. Otherwise, the mAP increases during the epochs and picks its best value as it approaches to the 2400 epoch.

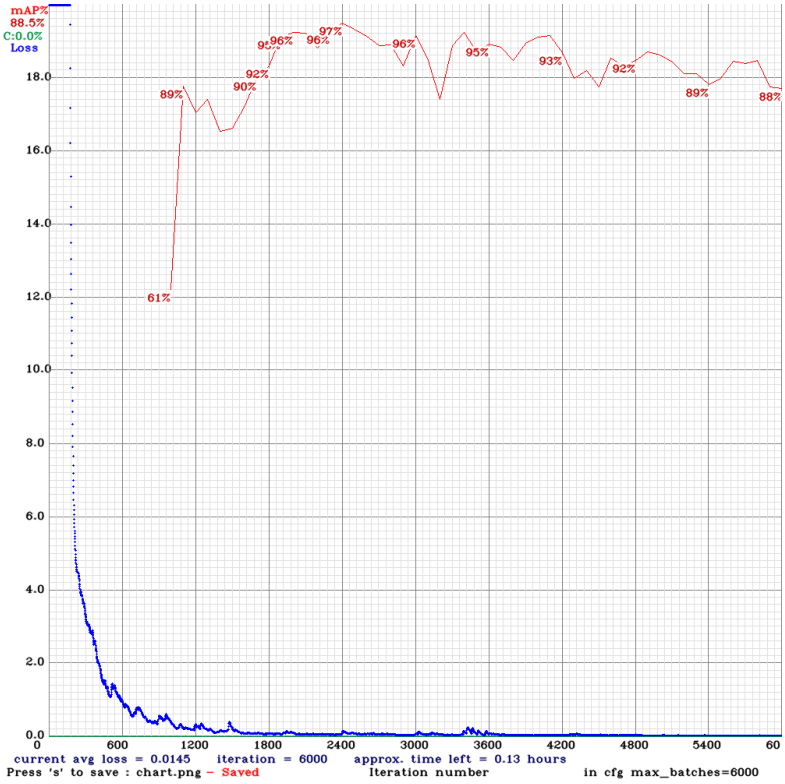


Figure 3.20: Performances of the detection system in terms of mean Average Precision (red line) and loss function (blue line)

Validation dataset and the corresponding trend of *mAP* are useful to select proper parameters of the system. In this case, the *mAP* indicates that the best weights are around the 3000 step of the training phase. Therefore, the model has been tested in the real environment using those specific weights with an industrial video camera and it achieves an accuracy of 85%.

3.3.3 Unity Barracuda and ONNX

To use the trained neural network in the DT Unity application it was necessary to export it to the ONNX format. ONNX (Open Neural Network Exchange) is an open format for ML models which allows to easily interchange models between various ML frameworks and tools. The conversion process was possible following the process described in one of the many GitHub repository discussing these conversions. It was necessary to provide the configuration file and the file containing the weights of the ANN to generate a file with onnx extension.

This particular file has been imported in the proposed DT application using Barracuda [28], a lightweight and cross-platform Neural Net inference library for Unity that permits to run ANN on both GPU and CPU. Through a control script it is possible to perform the object detection inside the DT using the ONNX model

on a snapshot of the wheels taken with an invisible camera in the VE, at the same position and perspective of the real one. The result of the detection will be a tensor, a container that stores data in N-dimensions. The trained YOLO system generates an output of dimension $24 \times 52 \times 52$ containing the results of the detection, given by the bounding boxes and the confidence of the detection. For each object detected in the input image, the output tensor saves the x-axis coordinate, the y-axis coordinate, width, and height. After filtering these results by selecting the ones with higher confidence, it is possible to draw the corresponding bounding box to the snapshot used as input, using different colors to distinguish the three classes (top, back, side).

Inside the VE, a GUI panel displayed near the cobot shows at each frame both the acquired snapshot and the detection result, whereas the user can restart the detection by a button (e.g., after correcting the pose of a misplaced wheel). The ONNX module is able to run YOLO algorithm and give back the results in about 2 seconds, which is a reasonable delay for testing the behavior of the cobot inside the DT. Figure 3.21 shows an example of the detection output inside the VR application.

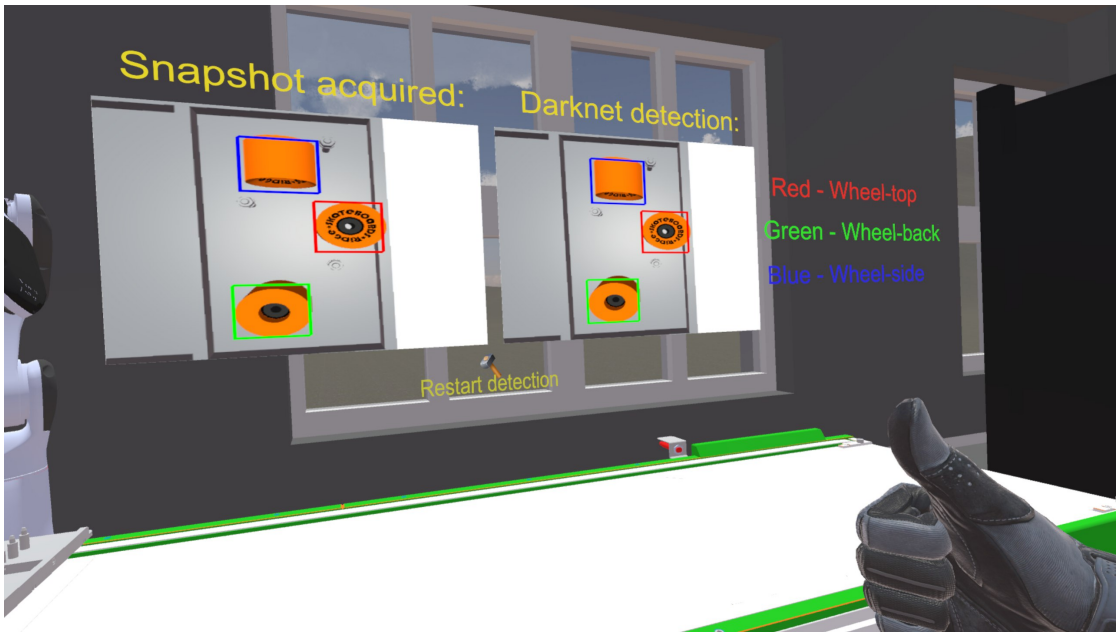


Figure 3.21: Screenshot of YOLO detection result inside the VR application

3.4 Process Simulate DT application

Alongside the Virtual Reality experience developed in Unity, this work presents a native DT model for the same production line developed on Tecnomatix Process Simulate software by Siemens. Process Simulate is one of the best solution to

develop a complex DT solution for a manufacturing process due to its capabilities to analyze ergonomics for human workstation, recreate robot movements and moreover to simulate the entire process by collecting data from physical assets. The application developed for this work does not implement this features yet, but it focuses of providing a VR experience of the production line to navigate and analyze the main assets and to test the Inverse Kinematics (IK) solver of the Racer 5 robot.

3.4.1 Design of the Virtual Environment

The virtual environment has been set up by importing the CAD files for each component of the production line; a 2D map of the real site has been used to replicate the exact position and proportion of the assets. Figure 3.22 illustrates how the Process Simulate GUI is organized: the VE comprehends the MIO automated warehouse, the Vir.GIL digital assistant and the Racer 5 robot installed above the conveyor belt; there are two virtual human models that represent the operators responsible for the interaction with the Vir.GIL and the cobot; the *Object Tree* panel on the left contains a hierarchy of the objects in the scene, organized in different directories; the upper part displays many panels (Robot, Process, Operation) that permits to access to specific features and properties of the software.

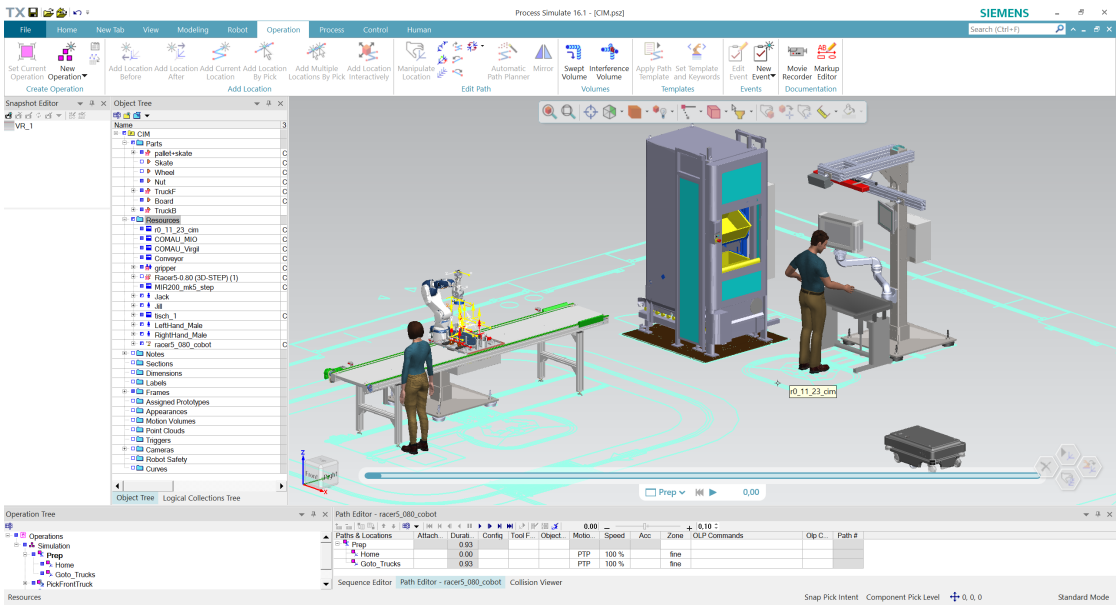


Figure 3.22: Screenshot of the DT model in Process Simulate

When the Racer 5 CAD model was imported, it was identified as a Robot component; the Robot panel on the top of the GUI provides all the tools for editing and validating robotic tasks in a 3D engineering environment. The first step was to define its kinematics through the *Kinematics Editor* panel (see figure 3.23). The Racer 5 is composed of 6 joints; by consulting the datasheet on the vendor's website it is possible to retrieve the rotations limits for each joint and set them accordingly. To perform the pick and place operation, the Racer 5 has a gripper mounted on the last joint; after importing the specific CAD model the *Mount tool* panel enables to install this tool for the robot; the gripper also has his specific kinematics to open and close itself for the pick and place operations. Once completed, the robot has been successfully set up and can be manipulated through the *Robot Jog* window (see figure 3.24). The *All Joints* area enables to adjust the values of each robot's joint in Forward Kinematics (FK), while the *Manipulation* area permits to exploit the Inverse Kinematics (IK) solver. Therefore, each translation or rotation of the gripper (end effector) will adjust each joint of the Racer 5 accordingly.

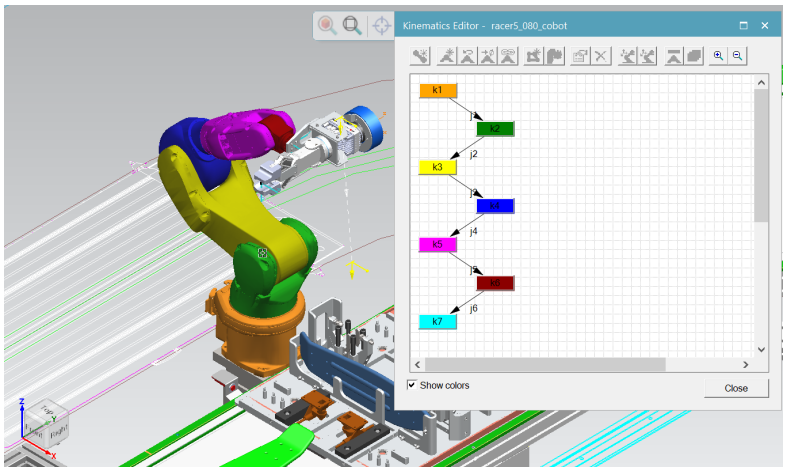


Figure 3.23: View of the Process Simulate Kinematics Editor

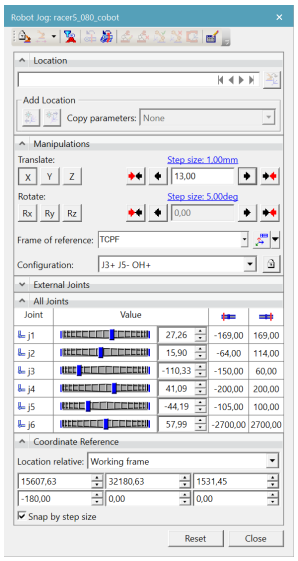


Figure 3.24: View of the Process Simulate Robot Jog Panel

3.4.2 Interaction Design

This stage of the DT model refers to the development and simulation of each asset's behaviour including the human tasks and the interaction with the machines' GUI. In particular, this section will describe how the Racer 5 movement have been recreated after setting up its kinematics.

Process Simulate provides tool to create a digital simulation of the production

line. Figure 3.25 shows the two panels involved: *Operation Tree* and *Path Editor*. From the former panel a new *Operation* is defined. In this case the simulation wants to illustrate the operations conducted by the Racer 5 robot to pick and place the trucks and the board. The movements of the robot are defined from the *Robot Jog* panel exploiting the IK solver and being registered in the *Path Editor* that provides an easy way to visualize and manipulate path data by displaying detailed information about paths and locations. It is possible to specify the *Motion Type* for the interpolation between two position (PTP, LINEAR) and also define some macros to control the tool mounted on the robot. For example, the #Drive CLOSE macro will close the gripper when the robotic arm is approaching to the component to be picked up.

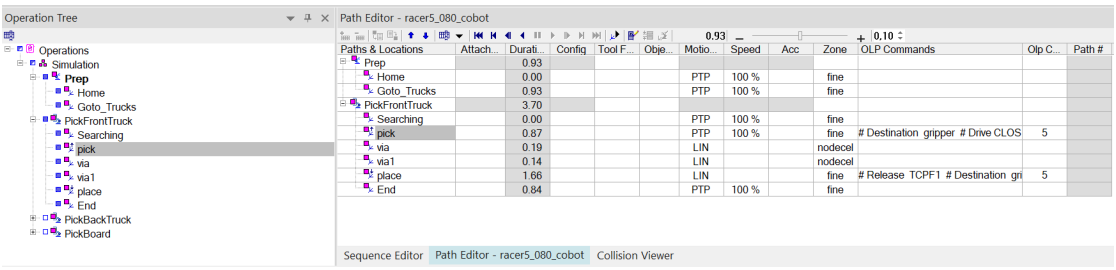


Figure 3.25: Process Simulate panels for setting up a simulation

In addition, Process Simulate offers the possibility to load the virtual environment created into a Virtual Reality world. It supports only the HTC VIVE Pro headset and the SteamVR plugin, the same used for the Unity application. In particular, after setting up the robot kinematics it is possible to interact with it and simulate the IK solver by grabbing the end-effector. Another important feature allows to generate a *VR Invitation* to collaborate in a virtual reality session, event if the guest machine does not run Process Simulate. The collaboration can begin after the host submits an invitation file. After the guests open the invitation file, they share the same virtual environment of the host and can interact with the same assets.

Chapter 4

Results and Analysis

4.1 Performance analysis

Considering that the proposed DT solution provides a real-time Virtual Reality experience developed in Unity 3D, performance characteristics play a key role. In fact, low **framerate** or excessive memory usage would affect the immersive gameplay and will end to not meet the desired requirements. As already mentioned the main goals of the proposed VR experience are: to navigate and analyze the production line; train the user to correctly perform a certain task; test a new functionality modifying the current vision system mounted on the Racer 5 robot.

For a good VR experience that feels responsive and doesn't cause any sickness, the framerate should be at least 60 FPS. Two aspect can be responsible of a serious drop of FPS during the VR experience: 1) the number of polygons of the 3D objects imported in the Unity scene 2) the complexity of the code in the C# script attached to the GameObjects.

When running the VR application and approaching to a virtual asset that has a large number of polygons, the system may experience a significant drop of FPS. To overcome this problematic is necessary to simplify the mesh of that particular 3D object. Blender provides a decimate tool to reduce the polygon count by merging vertices based on their relative distance. If correctly used the mesh would appear almost the same and the Unity engine will render the corresponding object without further troubles.

The second issue that may decrease the framerates can be easily resolved by analyzing the structure of a C# script in Unity. When running the application, the Unity pipeline will perform the *Update* methods before displaying each frame in the game. If the VR experience runs at 60 FPS the *Update* function will run 60 times per second. Therefore, it is necessary to avoid heavy nested loops in the *Update* function with unnecessary calculations else the delay times will become exponential. To prevent this behaviour, it can be useful to run these processes in a co-routine to have the processing spread over many frames.

4.2 Comparison between the proposed solution

This work proposes two different implementation for a DT model made with Unity 3D (section 3.2) and Tecnomatix Process Simulate (section 3.4). The real difference between the two solution lies in the type of application that they can offer. In fact, Unity 3D is a cross-platform game engine that is mainly used to create games and complex 3D application, while Process Simulate is a digital manufacturing solution for process simulation in a 3D environment. Therefore, it is evident that the best solution to develop a complete Digital Twin solution is to use Process Simulate. However, as the requirement of this project heads towards the creation of a complex VR interactive experience, Unity 3D fully satisfies those expectations.

Manufacturing companies can evaluate to develop a DT application using Unity 3D or Process Simulate by taking into account various aspect. Unity 3D is a free software and allows to set up a VE for any production line in shorter time; virtual models can be created and animated using a third-part 3D modelling software and imported into the Unity scene; the design of the interactions for the VR application can be implemented by a programmer with experience in C# programming language. On the other hand, the usage of Process Simulate requires a licence and moreover an ad-hoc formation by Siemens in order to fully exploit the capabilities of the software.

However, it has been analyzed that the key factor of a DT model comprehends the exchange of data between the physical assets and its virtual counterpart to update the simulation and retrieve significant information about the whole process. This aspect can be covered using Tecnomatix Process Simulates suite, as mentioned by [8]. Moreover, the Siemens product also provides the tools to evaluate human ergonomics and to extract important variables that can optimize the performance of the production line.

To sum up, both Unity 3D and Process Simulate are valid solutions for the creation of a DT model for a production line, but the level of detail that they can offer is very different. A manufacturing company should prefer one solution over the other according to its needs and expectation.

4.3 Subjective evaluation tests

After the development of the DT solution in Unity 3D, a group of 12 people was asked to test the VR experience in order to obtain a subjective quality assessment of the application and collect feedbacks for further improvements.

The test group was composed of 8 male and 4 female subjects, with an age between 21 a 44 years, selected to have a wide variety of skills and different professional backgrounds. The application was tested in CIM 4.0 digital site, where the real production line is located, so the subjects have had a chance to examine the real assets and their appearance; this allowed to better evaluate the digital representation as compared to the real site.

The subjects were asked to answer a questionnaire (see A.2), that collects the user's opinion and impression about the system usability, intuitiveness of the navigation tools, realism of the VE as compared to the real production line and the perceived performance characteristics.

The survey consists of three parts: the first one collects general information about the test subject, such as age, sex and previous VR experiences; for the second part the used is asked to perform 2 different tasks and reply to few questions taken from the NASA TLX tool [29] for assessing the perceived workload of the proposed actions and check if the user have suffered from any nausea or sickness during the VR experience. The first task consists in launching the application and explore the Virtual Environment of the production line using the teleportation function, inspect the virtual assets by checking if the dimensions, the materials and the lightning resemble the real site and perform the first step of the simulation (customize the skateboard); the second task requires the user to complete the simulation by performing the training task to prepare the production pallet, followed by the assembly phase of the skateboard. During the execution of the tasks the user can meet all the features that have been developed such as the interaction with the UI windows, the presence of the humanoid avatar and the interaction with objects and buttons.

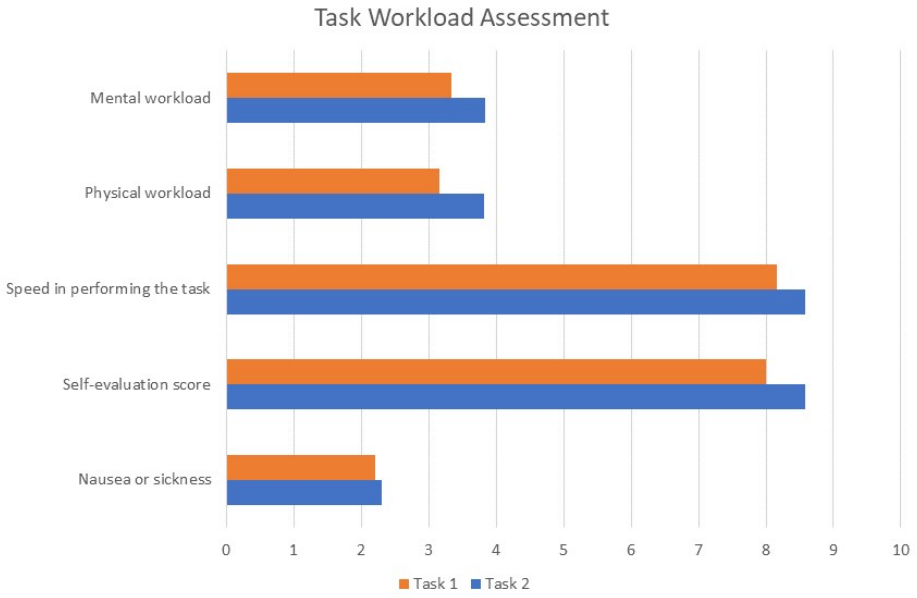


Figure 4.1: Average results for perceived workload, self-evaluation and sickness in both tasks

After completing each task, the user has to record his impression answering 5 question about the physical and mental workload, time to get used to the commands and complete the task, success in performing the actions and sickness or nausea felt. The answers are expressed by choosing a value between 1 and 10. Figure 4.1 reports the average result for each of these questions, showing positive results for both task. A noteworthy outcome is the absence of nausea or sickness experienced by test subjects (probably due to the good performance of the VR headset and the optimizations performed), while the increase of physical and mental workload score of task 2 can be reasonable because the user had to follow visual instructions and complete the training procedure to prepare the production pallet.

The third and last section of the survey, contains questions about the system usability and quality assessment of the VR experience. The user is asked to rate the intuitiveness, usability, realism and performance of the system when performing the tasks. Each question can be answered with a scale of five values, from strongly disagree to strongly agree to evaluate different aspects of the user experience.

At the end, the user also has the chance to suggest any improvements for further increase the overall experience of the DT application.

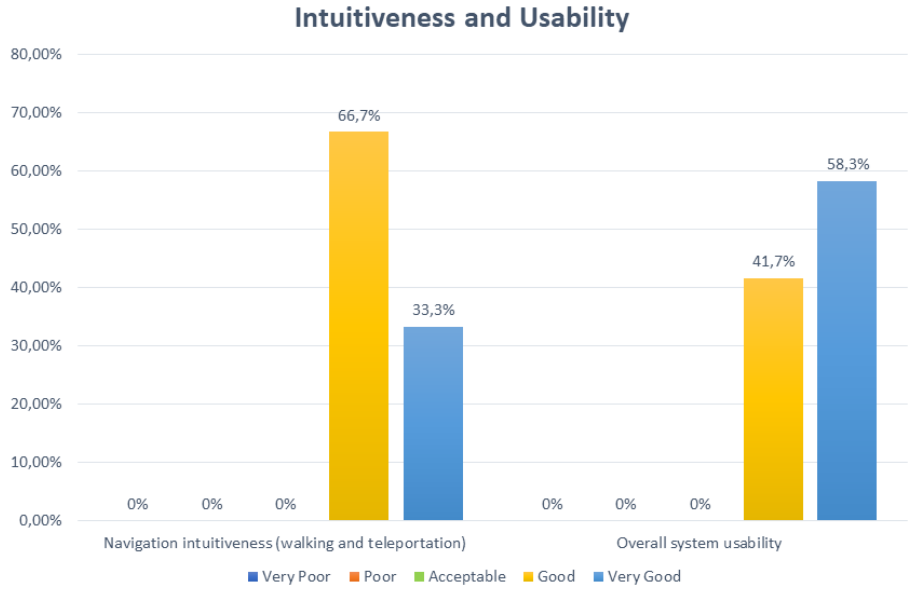


Figure 4.2: User evaluation of navigation intuitiveness and system usability

The graph in figure 4.2 shows the scores assigned by users to the intuitiveness of the teleportation tool in the VE, as well as a usability score which refers to the application as a whole. Test subjects were very satisfied with the navigation tools (66.7% "Good" and 33.3% "Very Good"), so the teleportation seems to be a valid tool to navigate around the VE, and also the overall system usability was appreciated by the mayority of the users (58.3% "Very Good" and 41.7% "Good").

When questioned about the realism of the 3D models as compared to the real

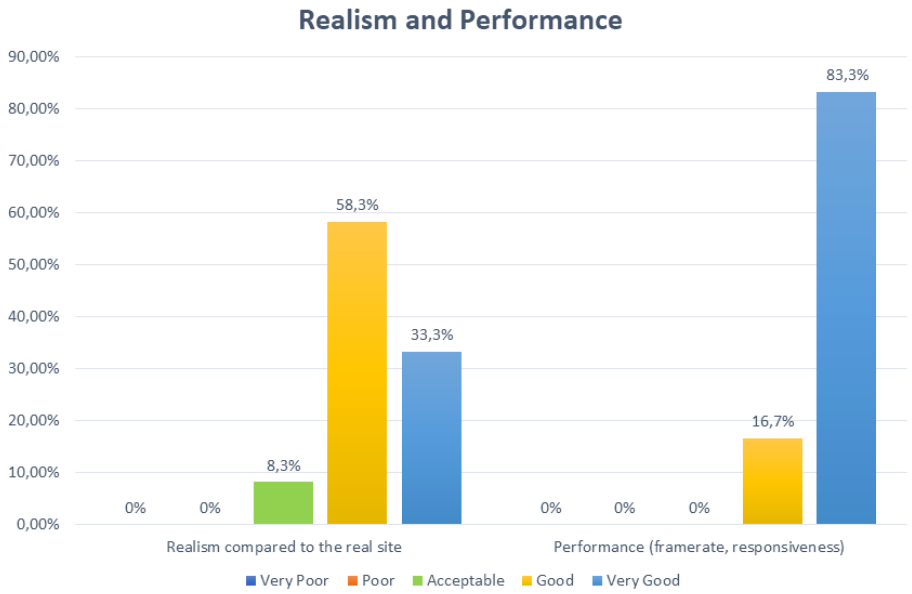


Figure 4.3: Evaluation of the perceived performance and realism compared to the real production line

assets, and the application's performance (framerate and responsiveness) users also answered rather positively. Figure 4.3 shows that test subjects were very satisfied with the application's performance (83.3% "Very Good" and 16.7% "Good" answers), while the realism of the virtual assets was also mostly positive (58.3% "Good" and 33.3% "Very Good") but also included a 8.3% of lower "Accepatble" scores. In fact from the questionnaire emerged 2 suggestions to further improve the ambient details and the textures of the objects.

Other important aspects that were asked to rate were the effectiveness of the training task and the accuracy of the Object Detection module, which is integrated in the VR experience. Figure 4.4 highlights a very positive result: the majority of the users (66.7%) found the accuracy of the detection "Good" and the remaining 33.3% answered "Very Good", so the delay for computing the prediction did not preclude the visual experience. Also the training task was considered "Very Effective" from the 66.7% of the users, with a lower 8.3% "Acceptable" and 25% "Good" scores. Probably some of the steps should be explained more in detail.

Finally, the users were asked to judge the intuitiveness of the grabbing actions performed to collects virtual objects and also to interact with some buttons, together with the usefulness of the humanoid virtual assistant (vocal feedback and UI windows). As it can be seen in figure 4.5, the vast majority of the users (83.3%) didn't have trouble to interact with the virtual objects and the buttons, and also the 66.7% of the users felt that the virtual assistant was "Very Useful". So, the choice of guiding the user through each step of the simulation turned out to be good, even though a lower 8.3% found it "Acceptable", suggesting to increase the

voice commands of the virtual assistant by providing a virtual UI interface for asking some default questions.

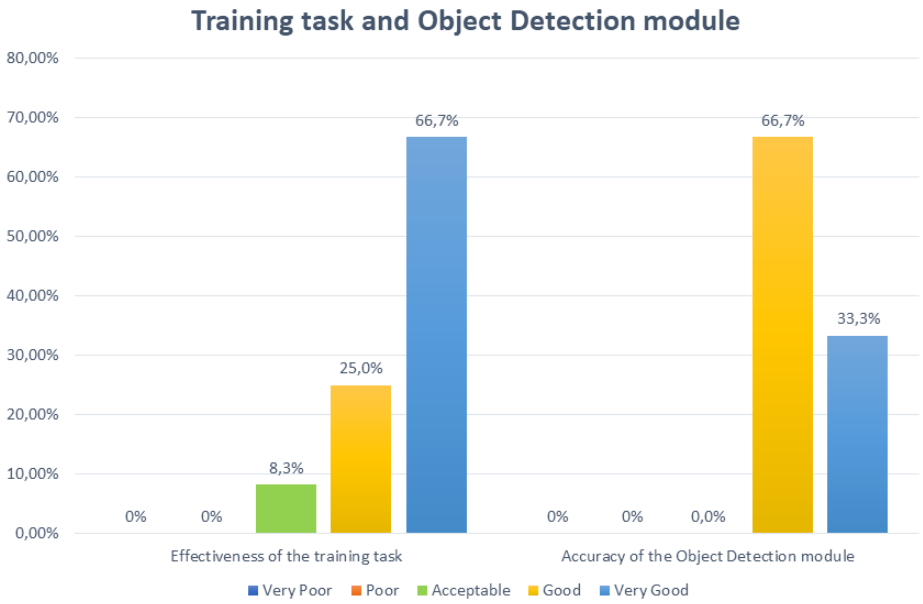


Figure 4.4: Subjective scores evaluating the training task and the accuracy of the Object Detection model

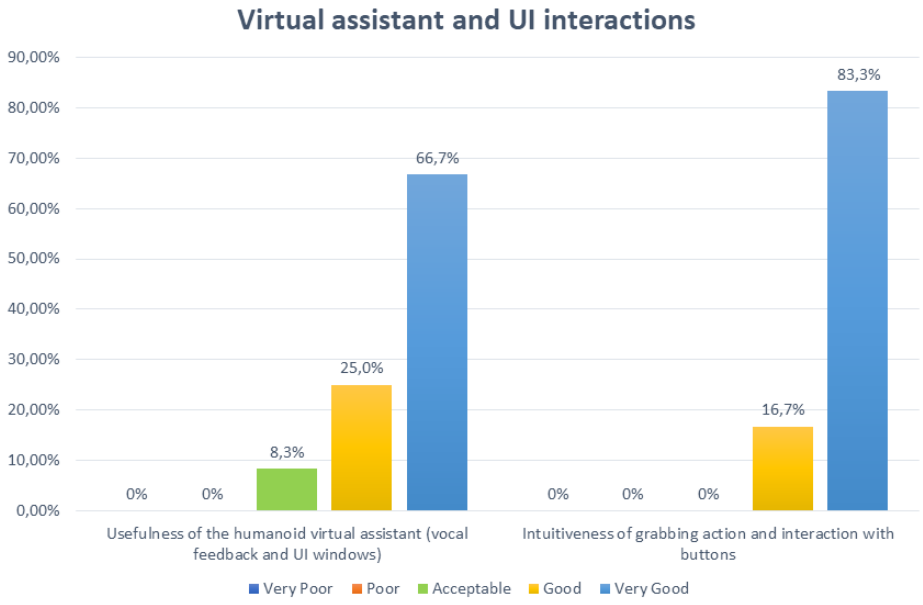


Figure 4.5: Evaluation of the usefulness of the humanoid virtual assistant and the intuitiveness of UI interactions

Chapter 5

Conclusions and Future Work

The proposed research describes the development and deploy of a Digital Twin of a production line for assembling skateboards, deployed at the Competence Industry Manufacturing 4.0 (CIM 4.0) Center. The production line combines together different recent technologies such as collaborative robots, an intelligent warehouse system, digital guidance for the operator and image-processing algorithms, especially to assist a cobot in a pick and place task. Since this step of the procedure rely on a detection approach prone to errors, the DT was exploited to simulate and evaluate a possible upgrade for the production line. To this end, the DT integrates a recent image-processing system based on a ML convolutional neural network trained on a synthetic dataset. Moreover, the neural network can provide real-time performances, thus the DT can be used not only for automatic simulations but also for real-time immersive virtual reality sessions. Overall, all the technologies provided in the physical line have been successfully digitalized in the DT, resulting in a compelling and valuable tool that will be used to visualize, navigate and inspect the production line through immersive VR and train technicians to correctly interact and operate the different technologies and their interfaces. The proposed upgrade to the image-processing system clearly enhances the system reliability and will lead to a physical update of the real-world system.

Future works will be aimed at further testing and evaluating possible upgrades to the physical line in the VE prior to implementing them in the real world. Further digitalizing physical upgrade of the line may be necessary: e.g., an automated ground vehicle (AGV) is currently been integrated in the physical line to move the pallet from the Vir.GIL workbench to the Racer 5 assembly line.

Regarding the Unity 3D application, it would be useful to develop new features aimed to make the VR experience more immersive, for example:

- investigate the digitalization of the IK solver for the Racer 5 into the DT, since it could lead to higher level of collaboration between the cobot and the

final user, which could be tested in a simulated, immersive environment.

- integrate a motion tracking technology to stream the movements of the user body, including the hands and evaluate how the experience may increase the value of the training task
- develop a framework to make the application suitable to support the collaboration between two different user in the same virtual reality session and communicate among them with a vocal or textual channel

Further improvements will also aimed to upgrade the current version of the application developed in Process Simulate in order to evaluate the ergonomics for the human workstation and also to integrate the YOLO algorithm within the simulation.

Finally, the YOLO development has recently moved to Ultralytics, thus, it would be interesting to update and test a more recent version of YOLO into the DT.

Appendix

A.1 Synthetic Dataset Creation

A.1.1 Calculate Bounding Box function

```
1
2  def calculateBoundingBox(scene, camera_object, mesh_object):
3      # Get the inverse transformation matrix
4      matrix = camera_object.matrix_world.normalized().inverted()
5      # Create a new mesh data block, using the inverse transform matrix to undo any
        transformations
6      mesh = mesh_object.to_mesh()
7      mesh.transform(mesh_object.matrix_world)
8      mesh.transform(matrix)
9
10     # Get the world coordinates for the camera frame bounding box
11     frame = [-v for v in camera_object.data.view_frame(scene=scene)[:3]]
12
13     lx = []
14     ly = []
15
16     for v in mesh.vertices:
17         co_local = v.co
18         z = -co_local.z
19
20         if z <= 0.0:
21             #Vertex is behind the camera; ignore it
22             continue
23         else:
24             # Perspective division
25             frame = [(v / (v.z / z)) for v in frame]
26
27         min_x, max_x = frame[1].x, frame[2].x
28         min_y, max_y = frame[0].y, frame[1].y
29
30         x = (co_local.x - min_x) / (max_x - min_x)
31         y = (co_local.y - min_y) / (max_y - min_y)
32
33         lx.append(x)
34         ly.append(y)
35
36     min_x = np.clip(min(lx), 0.0, 1.0)
37     max_x = np.clip(max(lx), 0.0, 1.0)
38     min_y = np.clip(min(ly), 0.0, 1.0)
39     max_y = np.clip(max(ly), 0.0, 1.0)
40
41     return min_x, max_x, min_y, max_y
```

A.2 User survey

Digital Twin application evaluation questionnaire

*Campo obbligatorio

Age *

La tua risposta

Sex *

- Male
- Female

Have you ever had a Virtual Reality experience? *

- Yes
- No

Task 1

After launching the application, explore the Virtual Environment using the teleportation function, inspect the virtual assets and complete the first step of the workflow (customize the skateboard).

How mentally demanding was the task? *

	1	2	3	4	5	6	7	8	9	10	
Very Low	○	○	○	○	○	○	○	○	○	○	Very High

How physically demanding was the task? *

	1	2	3	4	5	6	7	8	9	10	
Very Low	○	○	○	○	○	○	○	○	○	○	Very High

How fast did you manage to get used to the commands? *

	1	2	3	4	5	6	7	8	9	10	
Very Slow	○	○	○	○	○	○	○	○	○	○	Very Fast

How successful were you in accomplishing what you were asked to do ? *

	1	2	3	4	5	6	7	8	9	10	
Failure	○	○	○	○	○	○	○	○	○	○	Perfect

How much nausea or sickness did you feel when performing the task ? *

	1	2	3	4	5	6	7	8	9	10	
None	○	○	○	○	○	○	○	○	○	○	Very High

Task 2

After the customization of the skateboard, perform the remaining two steps: preparation of the production pallet and assembly of the skateboard.

How mentally demanding was the task? *

	1	2	3	4	5	6	7	8	9	10	
Very Low	○	○	○	○	○	○	○	○	○	○	Very High

How physically demanding was the task? *

	1	2	3	4	5	6	7	8	9	10	
Very Low	○	○	○	○	○	○	○	○	○	○	Very High

How fast did you manage to get used to the visual feedbacks ? (UI windows and bouncing arrows) *

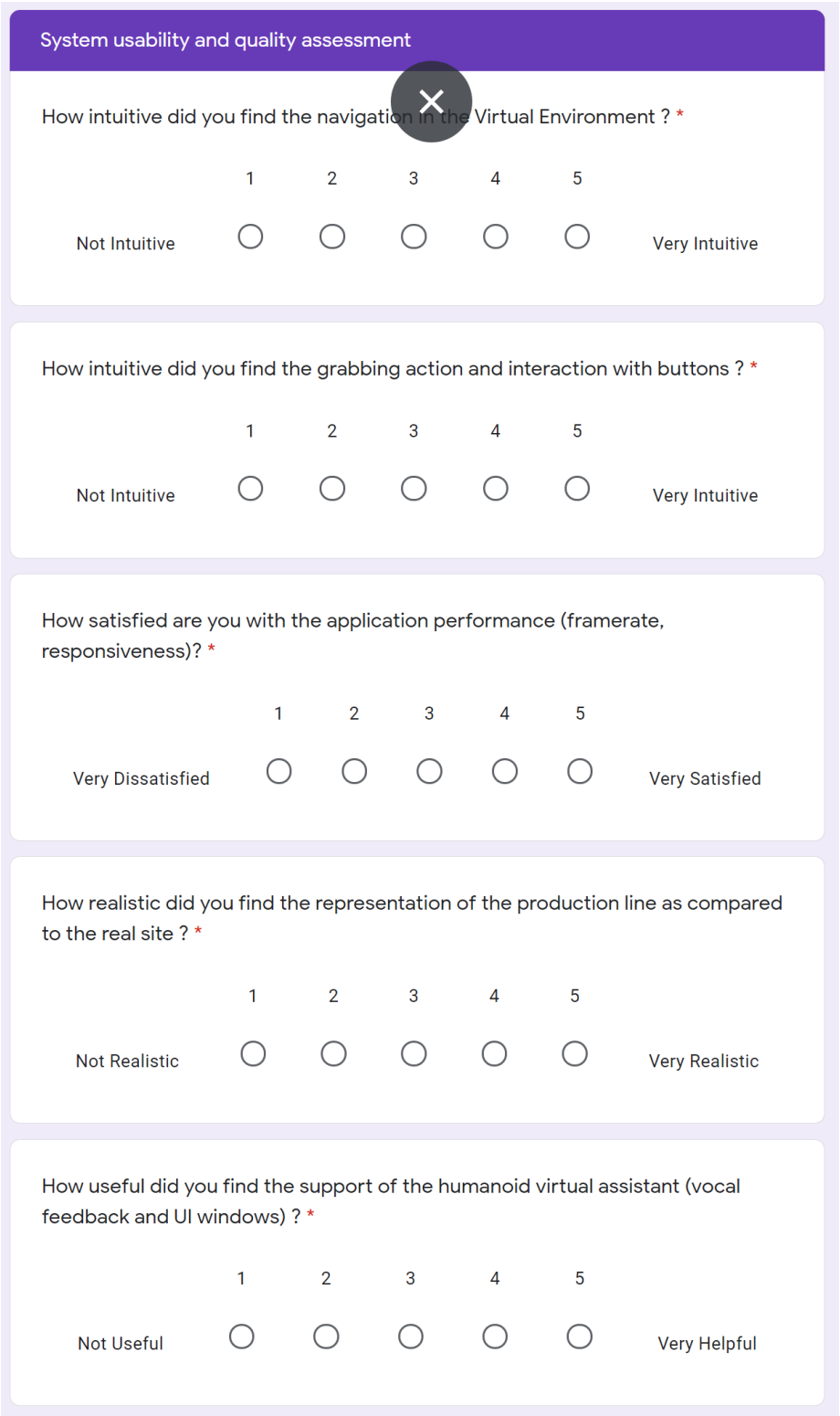
	1	2	3	4	5	6	7	8	9	10	
Very Slow	○	○	○	○	○	○	○	○	○	○	Very Fast

How successful were you in accomplishing what you were asked to do ? *

	1	2	3	4	5	6	7	8	9	10	
Failure	○	○	○	○	○	○	○	○	○	○	Perfect

How much nausea or sickness did you feel when performing the task ? *

	1	2	3	4	5	6	7	8	9	10	
None	○	○	○	○	○	○	○	○	○	○	Very High



System usability and quality assessment

How intuitive did you find the navigation in the Virtual Environment ? *

	1	2	3	4	5	
Not Intuitive	○	○	○	○	○	Very Intuitive

How intuitive did you find the grabbing action and interaction with buttons ? *

	1	2	3	4	5	
Not Intuitive	○	○	○	○	○	Very Intuitive

How satisfied are you with the application performance (framerate, responsiveness)? *

	1	2	3	4	5	
Very Dissatisfied	○	○	○	○	○	Very Satisfied

How realistic did you find the representation of the production line as compared to the real site ? *

	1	2	3	4	5	
Not Realistic	○	○	○	○	○	Very Realistic

How useful did you find the support of the humanoid virtual assistant (vocal feedback and UI windows) ? *

	1	2	3	4	5	
Not Useful	○	○	○	○	○	Very Helpful

How would you rate the effectiveness of the training task for the preparation of the production pallet ? *

	1	2	3	4	5	
Not Effective	○	○	○	○	○	Very Effective

How would you rate the accuracy of the Object Detection module during the assembly task ? *

	1	2	3	4	5	
Very Poor	○	○	○	○	○	Excellent

Overall system usability score *

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Suggestions

La tua risposta

Bibliography

[1] Kosmas Alexopoulos, Nikolaos Nikolakis, and George Chryssolouris. «Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing». In: *International Journal of Computer Integrated Manufacturing* 33.5 (2020), pp. 429–439. DOI: 10.1080/0951192X.2020.1747642. eprint: https://doi.org/10.1080/0951192X.2020.1747642. URL: https://doi.org/10.1080/0951192X.2020.1747642 (cit. on pp. 2, 14, 15).

[2] Elisa Negri, Luca Fumagalli, and Marco Macchi. «A Review of the Roles of Digital Twin in CPS-based Production Systems». In: *Procedia Manufacturing* 11 (2017). 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy, pp. 939–948. ISSN: 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2017.07.198. URL: https://www.sciencedirect.com/science/article/pii/S2351978917304067 (cit. on p. 7).

[3] Fei Tao, Meng Zhang, and A.Y.C. Nee. *Digital Twin Driven Smart Manufacturing*. San Diego, USA: Elsevier Science and Technology, 2019 (cit. on p. 8).

[4] Fei Tao, Meng Zhang, Yushan Liu, and A.Y.C. Nee. «Digital twin driven prognostics and health management for complex equipment». In: *CIRP Annals* 67.1 (2018), pp. 169–172. ISSN: 0007-8506. DOI: https://doi.org/10.1016/j.cirp.2018.04.055. URL: https://www.sciencedirect.com/science/article/pii/S0007850618300799 (cit. on p. 8).

[5] *Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles*. URL: https://www.iso.org/standard/75066.html (cit. on p. 8).

[6] Li Da Xu, Eric L. Xu, and Ling Li. «Industry 4.0: state of the art and future trends». In: *International Journal of Production Research* 56.8 (2018), pp. 2941–2962. DOI: 10.1080/00207543.2018.1444806. eprint: https://doi.org/10.1080/00207543.2018.1444806. URL: https://doi.org/10.1080/00207543.2018.1444806 (cit. on p. 10).

[7] Jiewu Leng, Dewen Wang, Weiming Shen, Xinyu Li, Qiang Liu, and Xin Chen. «Digital twins-based smart manufacturing system design in Industry 4.0: A review». In: *Journal of Manufacturing Systems* 60 (2021), pp. 119–137. ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2021.05.011. URL: https://www.sciencedirect.com/science/article/pii/S0278612521001151 (cit. on pp. 11, 15).

[8] S. M. Jeon and S. Schuesslbauer. «Digital Twin Application for Production Optimization». In: *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 2020, pp. 542–545. DOI: 10.1109/IEEM45057.2020.9309874 (cit. on pp. 11, 15, 45).

[9] *History of VR - Timeline of Events and Tech Development*. URL: https://virtualspeech.com/blog/history-of-vr (cit. on p. 13).

[10] Michael Grieves. «Digital Twin: Manufacturing Excellence through Virtual Factory Replication». In: (Mar. 2015) (cit. on p. 15).

[11] Roland Rosen, Georg von Wichert, George Lo, and Kurt D. Bettenhausen. «About The Importance of Autonomy and Digital Twins for the Future of Manufacturing». In: *IFAC-PapersOnLine* 48.3 (2015). 15th IFAC Symposium onInformation Control Problems inManufacturing, pp. 567–572. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.06.141. URL: https://www.sciencedirect.com/science/article/pii/S2405896315003808 (cit. on p. 15).

[12] Nikolaos Nikolakis, Kosmas Alexopoulos, Evangelos Xanthakis, and George Chryssolouris. «The digital twin implementation for linking the virtual representation of human-based production tasks to their physical counterpart in the factory-floor». In: *International Journal of Computer Integrated Manufacturing* 32.1 (2019), pp. 1–12. DOI: 10.1080/0951192X.2018.1529430. eprint: https://doi.org/10.1080/0951192X.2018.1529430. URL: https://doi.org/10.1080/0951192X.2018.1529430 (cit. on p. 15).

[13] Fei Tao, He Zhang, Ang Liu, and A. Y. C. Nee. «Digital Twin in Industry: State-of-the-Art». In: *IEEE Transactions on Industrial Informatics* 15.4 (2019), pp. 2405–2415. DOI: 10.1109/TII.2018.2873186 (cit. on p. 15).

[14] T.G. Ritto and F.A. Rochinha. «Digital twin, physics-based model, and machine learning applied to damage detection in structures». In: *Mechanical Systems and Signal Processing* 155 (Jan. 2021), p. 107614. ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2021.107614. URL: http://dx.doi.org/10.1016/j.ymssp.2021.107614 (cit. on p. 15).

[15] Farzin Piltan and Jong-Myon Kim. «Bearing Anomaly Recognition Using an Intelligent Digital Twin Integrated with Machine Learning». In: *Applied Sciences* 11.10 (2021). ISSN: 2076-3417. DOI: 10.3390/app11104602. URL: https://www.mdpi.com/2076-3417/11/10/4602 (cit. on p. 15).

[16] *Increasing Facility Uptime Using Machine Learning and Physics-Based Hybrid Analytics in a Dynamic Digital Twin*. Vol. Day 3 Wed, May 06, 2020. OTC Offshore Technology Conference. D031S032R002. May 2020. DOI: 10.4043/30723-MS. eprint: https://onepetro.org/OTCONF/proceedings-pdf/20OTC/3-20OTC/D031S032R002/2340518/otc-30723-ms.pdf. URL: https://doi.org/10.4043/30723-MS (cit. on p. 15).

[17] Mesfin Seid Ibrahim, Jiajie Fan, Winco K. C. Yung, Alexandru Prisacaru, Willem van Driel, Xuejun Fan, and Guoqi Zhang. «Machine Learning and Digital Twin Driven Diagnostics and Prognostics of Light-Emitting Diodes». In: *Laser & Photonics Reviews* 14.12 (2020), p. 2000254. DOI: https://doi.org/10.1002/lpor.202000254. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/lpor.202000254. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/lpor.202000254 (cit. on p. 15).

[18] *Blender website*. URL: https://www.blender.org/ (cit. on p. 18).

[19] *OpenGL website*. URL: https://www.opengl.org (cit. on p. 19).

[20] *Unity game engine website*. URL: https://unity.com (cit. on p. 20).

[21] Luís Filipe Rodrigues, Abílio Oliveira, and Helena Rodrigues. «Main gamification concepts: A systematic mapping study». In: *Heliyon* 5.7 (2019), e01993. ISSN: 2405-8440. DOI: https://doi.org/10.1016/j.heliyon.2019.e01993. URL: https://www.sciencedirect.com/science/article/pii/S240584401935618X (cit. on p. 20).

[22] *Siemens Tecnomatix Process Simulate website*. URL: https://www.engusa.com/it/product/siemens-tecnomatix-process-simulate (cit. on p. 21).

[23] *Darknet framework website*. URL: https://github.com/AlexeyAB/darknet (cit. on p. 22).

[24] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. «You Only Look Once: Unified, Real-Time Object Detection». In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640 (cit. on p. 23).

[25] *Comau website*. URL: https://www.comau.com/en/our-competences/robotics/robot-team/racer-5-0-80 (cit. on p. 27).

[26] *Mixamo website*. URL: https://www.mixamo.com/ (cit. on p. 34).

[27] *Yolo Mark Website*. URL: https://github.com/AlexeyAB/Yolo_mark (cit. on p. 38).

[28] *Unity Barracuda Website*. URL: https://docs.unity3d.com/Packages/com.unity.barracuda@1.0/manual/index.html (cit. on p. 39).

[29] *NASA Task Load Index (TLX) Website*. NASA. URL: https://humansystems.arc.nasa.gov/groups/TLX/ (cit. on p. 46).

Acknowledgements