MAGICIAN

AUTONOMOUS DEFECTS DETECTION AND REPAIR IN MANUFACTURING

PROJECT NO

101120731

PROJECT ACRONYM

MAGICIAN

PROJECT TITLE:

IMMERSIVE LEARNING FOR IMPERFECTION DETECTION AND REPAIR THROUGH HUMAN-ROBOT INTERACTION

CALL/TOPIC:

HORIZON-CL4-2022-DIGITAL-EMERGING-02-07

START DATE OF PROJECT:

01.10.2023

DURATION:

48 MONTHS

DELIVERABLE D3.2

Second delivery of perception systems

DUE DATE OF DELIVERABLE:

30.06.2025

ACTUAL SUBMISSION DATE:

15.07.2025

Work Package	**WP3 - Data acquisition and skills learning**
Associated Task	**T3.2 - Perception System, data acquisition and processing**
Deliverable Lead Partner	**FORTH**
Main author(s)	**Ammar Qammaz, Nikolaos Vasilikopoulos, Luigi Palopoli, Iason Oikonomidis, Antonis Argyros, Daniele Fontanelli, Annemarijn Blom, Matteo Dalle Vedove, Elena Basei, Gionata Salvietti, Michele Pompilio, Domenico Prattichizzo**
Internal Reviewer(s)	**Geert Driessen**
Version	**1.0**

DISSEMINATION LEVEL		
PU	Public	X
SEN	Sensitive - limited under GA conditions	

CHANGE CONTROL

DOCUMENT HISTORY

VERSION	DATE	CHANGE HISTORY	AUTHOR(S)	ORGANISATION
0.1	04/07/2025	First Draft	Ammar Qammaz, Nikolaos Vasilikopoulos, Luigi Palopoli, Iason Oikonomidis, Antonis Argyros, Daniele Fontanelli, Annemarijn Blom, Matteo Dalle Vedove, Elena Basei, Gionata Salvietti, Michele Pompilio, Domenico Prattichizzo	FORTH, UNITN, IIT, PIPPLE
0.2	11/07/2025	Internal Review	Geert Driessen	PIPPLE
1.0	13/07/2025	Final Version	Antonis Argyros	FORTH

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

This deliverable is part of a project that has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101120731.

EXECUTIVE SUMMARY

This deliverable presents the progress made in the development of the perception systems within the MAGICIAN project in the duration of time after D3.1. It focuses on the detection of imperfections using visual and tactile perception modules, human motion detection, and the learning of defect detection skills from human operators. The document provides a comprehensive overview of the methodologies employed, the results obtained, and the challenges and limitations encountered during the development process, setting the foundation for future work. A summary of the progress of the perception system is the following:

Defect Sensing module:

- The vision sensor has been redesigned using 6x programmable polarised LED light sources making the sensor robust to surfaces that occluded the original single light source.
- The tactile system has been integrated with the vision system with respect to both hardware and software components.
- The microcontroller used for the project has been optimised from an Arduino MEGA to an Arduino Nano.
- The camera system has switched to a 16 mm lens instead of the 12 mm used initially providing higher fidelity images that allow for better M/L accuracy.
- 3x Time of Flight laser range sensors have been added to the sensor providing passive range finding measurements (3D plane) to the robot.
- The physical design of components has been carried out using KiCad for the electronics board and OpenSCAD ensuring reproducibility of the sensor head.
- We have recorded data using data from FORTH and IIT shipments with recording the final shipment to UNITN happening during the second integration meeting scheduled for June 2025.
- The neural network computational workload has improved from 8Hz to 23Hz matching the maximum framerate of the camera.
- Sensing packages have been integrated with ROS and can communicate with the rest of the system (Figure 1.2).
- The current prototypes of the vision and tactile defect sensing modules were tested during the 1st integration meeting in IIT mounted and used with the MAGICIAN sensing robot.

Motion Detection module:

- We have successfully tracked persons from security camera recordings in the TOFAS factory.
- We have successfully tracked persons during the IIT integration meeting while also running the defect sensing stack in real-time.
- Human Pose estimation packages communicate with the rest of the robot using ROS and emit ROS messages and TF2 coordinates.

DEVIATIONS

No deviation is foreseen from the planned path.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

ACRONYM	DESCRIPTION
D	Deliverable
EC	European Commission
WP	Work package
WT	Work task
CR	Cleaning robot
SR	Sensing robot
2D	Two dimensional
3D	Three dimensional
BMI	Body Mass Index
CAD	Computer Aided Design
CMOS	Complementary metal oxide semiconductor
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DMP	Dynamic Motion Primitives
GDPR	General Data Protection Regulation
GigE	Gigabit Ethernet
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HMR	Human Mesh Recovery
KPI	Key Performance Index
LED	Light Emitting Diode
MOCAP	Motion Capture
MP	Megapixel
NN	Neural Network
PIR	Passive infrared

PSD	Power Spectral Density
RAM	Random Access Memory
RGB	Red, Green, Blue
RGBD	Red, Green, Blue + Depth
ROS	Robot Operating System
SDK	Software Development Kit
ViTs	Vision Transformers
WP	Work package
WT	Work Task
YOLO	You Only Look Once

1 INTRODUCTION

This deliverable provides a comprehensive report of the progress of development of the MAGICIAN sensing modules for data acquisition and skills learning. The main sensing modules of the MAGICIAN platform are (a) the visual perception system and (b) the tactile perception system for imperfection detection. These two fundamental modules involve physical sensors and hardware design (Figure 1.1) and are destined to complement each other helping the MAGICIAN robot successfully tackle the defect detection tasks that are currently performed by humans but will, in the future, be assigned to the MAGICIAN platform. Two more perception modules are (a) the human motion detection module and (b) the learning defect detection and learning working skills from humans' modules. These are software-defined and capitalize on recent AI methods to endow the platform with the capabilities required to sense humans and their actions.

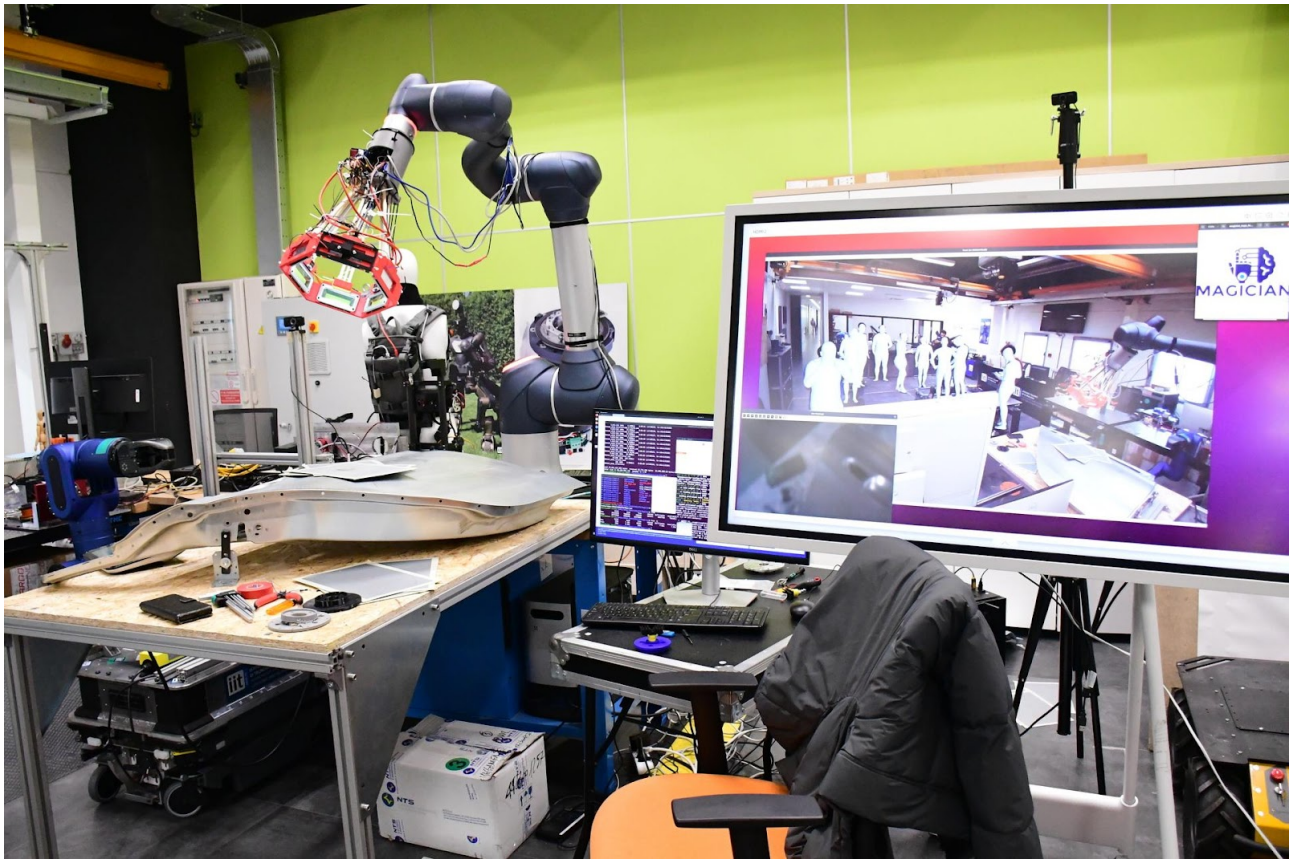


Figure 1.1. The Hardware & Software prototypes of the perception system integrated with the Doosan robot during the MAGICIAN 1st integration meeting in IIT premises.

1.1 PURPOSE AND SCOPE

The primary objective of this deliverable is to report the progress in the development and integration of the visual and tactile perception systems in the MAGICIAN project,

with focus on their role in human classification and skills learning. These systems are central to enabling the automated identification and categorization of manufacturing defects and serve as the foundation for subsequent robotic reworking operations.

This document presents the second release of the algorithms and software components that support human classification and the learning of reworking skills from human demonstrations. The emphasis is on the perception side, particularly on how multi-modal sensory inputs (visual, tactile) are processed and interpreted to inform defect classification.

While early steps toward defect reworking have been initiated, this deliverable prioritizes the maturity and capabilities of the perception modules. These include enhancements to the visual module's detection accuracy, improved annotation tools and strategies, and extended tactile sensing capabilities. Furthermore, the document outlines the mechanisms through which human demonstrations are captured, processed, and integrated with the perception systems for the purpose of learning reworking policies.

As with D3.1, both perception modalities—visual and tactile—are designed to complement each other, offering robust classification capabilities across a wide range of defect types and severities. This deliverable also describes updated methods for the visual observation of human operators, particularly as they perform classification tasks, which now play a critical role in enabling skill transfer and learning for robotic reworking.

This second iteration builds upon the foundational work described in D3.1 and marks the transition toward perception systems that are not only capable of detecting and classifying defects but also beginning to interpret and model human classification behaviour and workflows.

1.2 CONTRIBUTION TO PROJECT OBJECTIVES

Similarly to D3.1, the developments described in this deliverable contribute to multiple scientific, technological, and societal objectives of the MAGICIAN project. In particular, D3.2 supports:

Scientific and Technological Objectives:

- **O1**: A robotic perception module integrating visual and tactile sensors for defects analysis and classification. Specifically, D3.2 provides refined algorithms and improved data acquisition strategies that increase robustness and accuracy in visual-tactile defect classification. The modules have also been extended to support learning from human classification behaviours, creating a tighter feedback loop between data annotation, perception, and learning.
- **O2**: A robotic cleaning module with a specialised end-effector for defect reworking. This deliverable presents preliminary work towards learning-based policy acquisition for defect reworking is included, particularly where perception data and human annotations guide initial motion policy design.

- **O3**: A software robotic platform integrating services for perception and cleaning modules. This deliverable presents updates to the perception stack, including its integration within the broader robotic architecture and its interfaces to future reworking modules.

Social Sciences and Humanities (SSH) Objectives

- **O6**: A human-centred approach to human-robot collaboration, promoting usability, safety, and trustworthiness. The perception systems integrate human-derived classification data, supporting trust, transparency, and explainability in how robots interpret defects. Human-in-the-loop tools for annotation and demonstration have been further developed.

Demonstration Objectives

- **O7**: Demonstration of the prototypes in operational scenarios, and **O8**: Expansion of MAGICIAN scope and applicability via Financial Support to Third Parties (FSTP). D3.2 supports the long-term demonstration objectives by maturing the perception backbone of the system, which will be central in the upcoming integration and validation phases. Moreover, the updated tools and methods in this deliverable pave the way for external testing and potential third-party contributions under the FSTP scheme.

Overall, this deliverable demonstrates progress toward the perception-related goals of the MAGICIAN project. The visual and tactile modules have evolved beyond isolated sensing components to become integrated elements of a human-informed perception pipeline. While the core emphasis remains on accurate and explainable classification of manufacturing defects, early steps have been taken to support the learning of reworking policies through perceptual observation of human actions. These developments form an important stepping stone toward the overall ambition of MAGICIAN: to enable robust, safe, and human-aware automation of defect handling.

1.3 RELATION TO OTHER WORK PACKAGES

The perception components described in the present deliverable are at the heart of most of the project's activities, and they have an understandably strong relation with many of its work packages. The synthetic list of the most important relations remains identical to the one of D3.1 and is offered next for the purpose of document self-containment:

- **WP2 – Use case definition and platform design**: even if the perception solutions developed in the WP claim for a certain level of generality, the initial idea and the main design choices are connected to the specific requirements of the automotive use case identified as the main driver of the project's research activities. In particular, the unique combination requirements on the perception

system's accuracy, on its integration within a robust robotic platform, and on the final cost of the solution pose several formidable challenges that we are facing in the activities in WP3 and that are succinctly reported in this report.

- **WP4 – Robotic platform and interfaces:** WP3 and WP4 are the two main pillars producing the technological assets at the heart of the system components. The activities of the two WP are deeply intertwined. Specifically, the planning and scheduling components (T4.3) take their decisions based on the results of the defect analysis and on the prediction of the possible motion of human operators. The motion control and active sensing component make a direct use of the Information processed through the perception pipeline. This information is also used for T4.5 (closed-loop defect analysis). On the other hand, also the Inverse Information flow (from WP4 to WP3) is extremely important. Knowing the possible motion performance and strategies of the robot arm where the perception system is mounted sets the background for the development of perception strategies (e.g., the velocity of the motion and the accuracy of the distance between the perception system and the car plays an important role the decision of the optical system and of the visual processing pipeline).
- **WP5 - Integration and performance analysis**: The components developed in WP3 and described in this document will be integrated in the final platform (T5.1). Most of them will be part of the demonstrator (T5.2) and their performance will contribute substantially to the project's KPIs.
- **WP6 – Cascade funding management**: since the perception component will be used in the subprojects stemming from the cascade funding scheme, the project's findings will be crucial to offer support and technical assistance (T6.4).

1.4 STRUCTURE OF THE DOCUMENT

The document maintains the structure of D3.1. Its main structure comprises of seven chapters, addressing key components of the perception systems developed for the MAGICIAN project, with the last section devoted to the outlook and planning for the future integration of the modules and their fusion in a multi-modal system.

After this introductory Chapter, Chapters 2 and 3 focus on the visual and tactile perception systems, respectively, for imperfections detection. Each chapter details the introduction, state of the art, objectives, methodologies, and preliminary findings, and ends by highlighting the open challenges of these systems. Chapter 4 addresses human motion detection, emphasizing its importance for human-robot collaboration. This chapter follows a similar structure to the previous ones.

Chapter 5 describes the module for learning defect reworking skills from humans, which is crucial for enhancing automation in defect management. This chapter also includes state-of-the-art reviews, system objectives, methodologies, and findings. Chapter 6 looks ahead, outlining plans for further development and integration of the perception modules, with a focus on multi-modal fusion and active sensing. The document

concludes in Chapter 7, summarizing the key outcomes and setting the stage for the next steps in the MAGICIAN project.

1.5 PERCEPTION SYSTEM OVERVIEW

This section outlines the key requirements and system architecture of the MAGICIAN perception module, which supports synchronized, real-time data acquisition across multiple sensing modalities.

1.5.1 REQUIREMENTS AND SPECIFICATIONS

The requirements for the perception sensor remain the same and are closely tied to the key performance indicators (KPIs) set by the project (Section 1.2 and Table 2.1). The initial prototype version of the perception system proved a good initial approximation of the perception solution needed and provided the consortium with a very early platform to work, experiment and validate the project's required performance and accuracy analysis goals defined in the D2.1 "Use Case Definition" document.

1.5.2 SYSTEM ARCHITECTURE

The MAGICIAN perception system uses ROS (Robot Operating System) as its core development platform. ROS allows seamless integration of many software nodes (in our case visual defect detection, tactile defect detection, human tracking, force sensing, reworking and learning from demonstration), is compatible with code written in both interpreted (Python) and compiled languages (C/C++) and provides a stable and mature message communication API. It not only enables a robust and easy way for the MAGICIAN modules to interact and coordinate but it also makes the platform compatible with most other robot platforms and software that may use the MAGICIAN codebase. In a similar manner ROS ensures compatibility with partners introduced to the project during the OC1 and OC2 project stages.

ROS is geared towards a large variety of robotics projects and features versatile node communication primitives. The main such features are ROS Messages, ROS Topics and ROS Services. Systems that need to accommodate low bitrate image and lidar streams can be perfectly satisfied by the ROS platform. However, in contrast with most ROS applications MAGICIAN requires a high-capacity system architecture. The large volume and strict synchronization requirements of data recorded and transported from the vision sensor to the host system plays a crucial role for the accurate and timely execution of the defect sensing stack and any throughput delays and overheads will inevitably lead to dropped data. This will in turn adversely affect defect detection accuracy given the limited available time and very small physical size of the defects.

The module of the perception system that reads incoming data from the sensor is the MAGICIAN grabber. To deal with performance problems at their root, the MAGICIAN grabber node offers a data bypass by using the Linux Shared Memory Kernel capabilities.

Mapping a contiguous RAM memory region and populating it with incoming data offers a zero-copy approach with respect to the processes running on the MAGICIAN host system. Client applications such as the Python (Pytorch/Tensorflow) neural network classifiers then directly map these regions and only exchange semaphore signals to gracefully lock the buffers for reading and writing. The above stated architecture is summarized in Figure 1.2. Having processed the large amount of input data in a consistent, synchronized and optimized manner, the image/tactile classifiers can then emit lower bitrate ROS topic classification events that do not saturate the ROS system and allow other ROS-Nodes to operate with better performance due to the better efficiency of this approach.

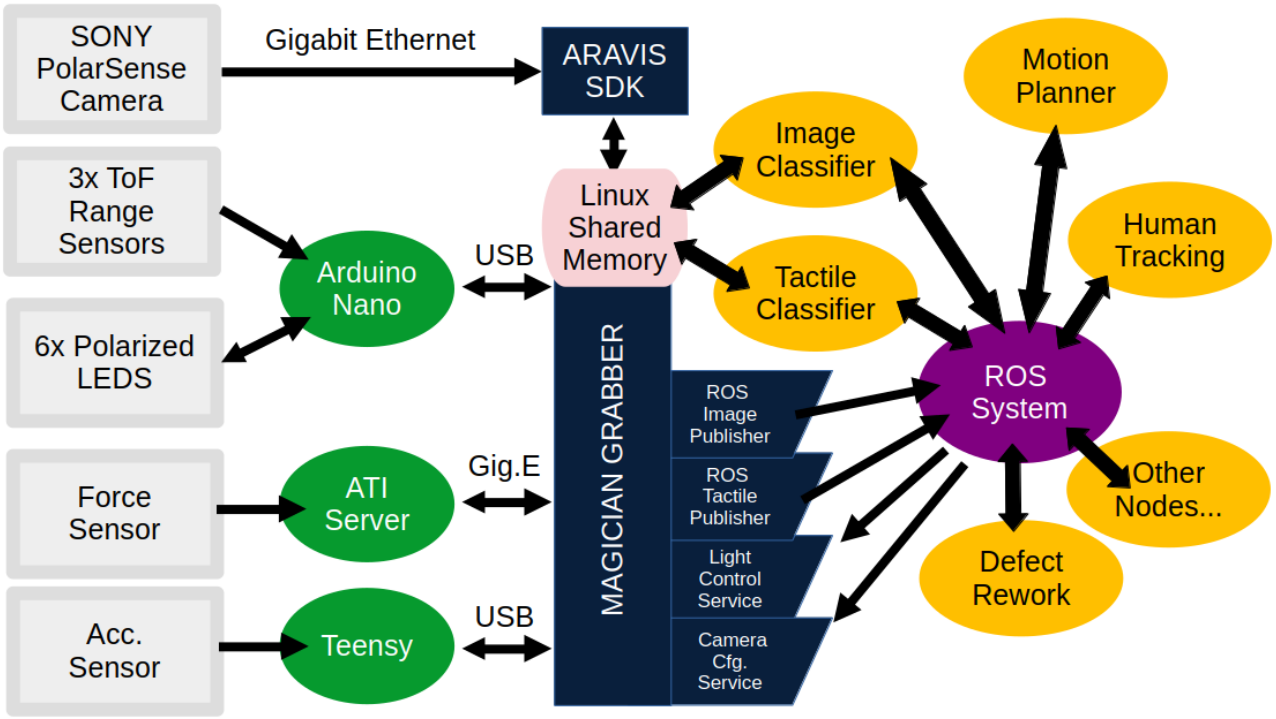


Figure 1.2. Overview of the perception system architecture. Gray boxes mark physical sensors. Green devices mark embedded controllers attached to the MAGICIAN host PC. To avoid congestion and synchronisation issues at the ROS level, the MAGICIAN Grabber (https://github.com/magician-project/magician_grabber) maps regions of system memory which can be accessed in a zero-copy fashion from other processes on the host PC using Linux Shared Memory. At the same time, it provides ROS messages and services to ensure easy operation for less demanding data streams.

2 VISUAL PERCEPTION FOR IMPERFECTIONS DETECTION

The visual perception system for imperfection detection is a critical component for the success of the project since vision is a primary sensing modality which factory workers utilize for imperfection detection. Mirroring the way humans operate, the scanning speed and accuracy of the visual perception module of the MAGICIAN robot directly impacts the main Key Performance Indicators (KPIs) of the project and therefore significantly influences the project outcome.

Following the visit to the industrial plant of TOFAS in January 2024, during April 2024, FORTH received a large shipment of annotated defective material that was instrumental in forming a very early proof of concept prototype that was shown to the project partners during October of 2024. Given their constructive feedback as well as experimental trials, we developed the 1st version visual perception module that will be presented in detail in this section. Through much coordinated effort this first version of the visual perceptions was first tested in the premises of Istituto Italiano di Tecnologia (IIT) during March of 2025 (Figure 2.1) performing detections in defective metal samples from the shipment sent to IIT that it had never seen before (since these samples were not part of the training dataset sent to FORTH). The current version of the sensor is able to operate in real-time (Figure 2.3) and perform Positive/Negative Dent classification for Class A defects (Figure 2.2). It is made of 3D printed parts, and features 6x LED illuminators.

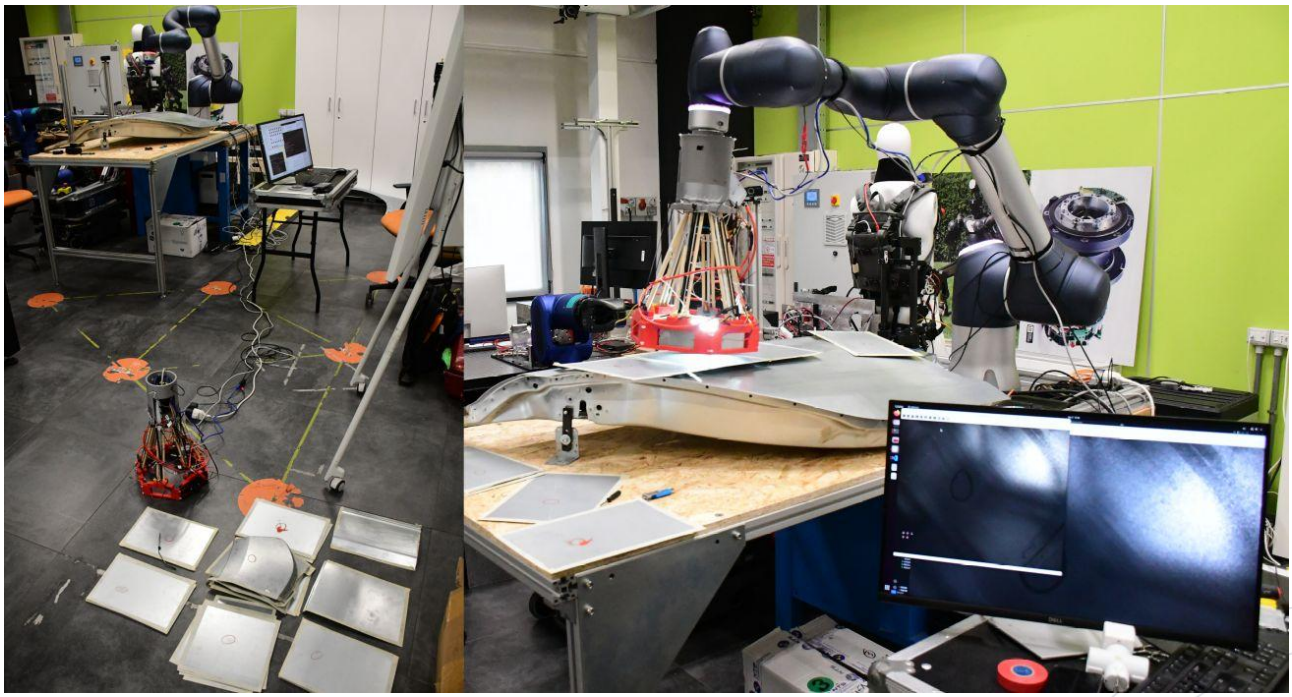


Figure 2.1. During the integration meeting the collection of defective samples were recorded to be included in the database of training samples for training the neural networks of the sensor.

2.1 INTRODUCTION

The project use cases described in D2.1 for automotive manufacturing present significant challenges. The MAGICIAN system will need to deal with various car chassis, each comprising hundreds of uniquely processed metal sheets with diverse shapes, edges and contours. Sensing is required to detect a broad variety of imperfections, namely

- positive/negative dents
- weld spatters
- sealing residuals,
- deformations
- material defects

each of which uniquely affects the involved surfaces, as illustrated in Figure 2.2.

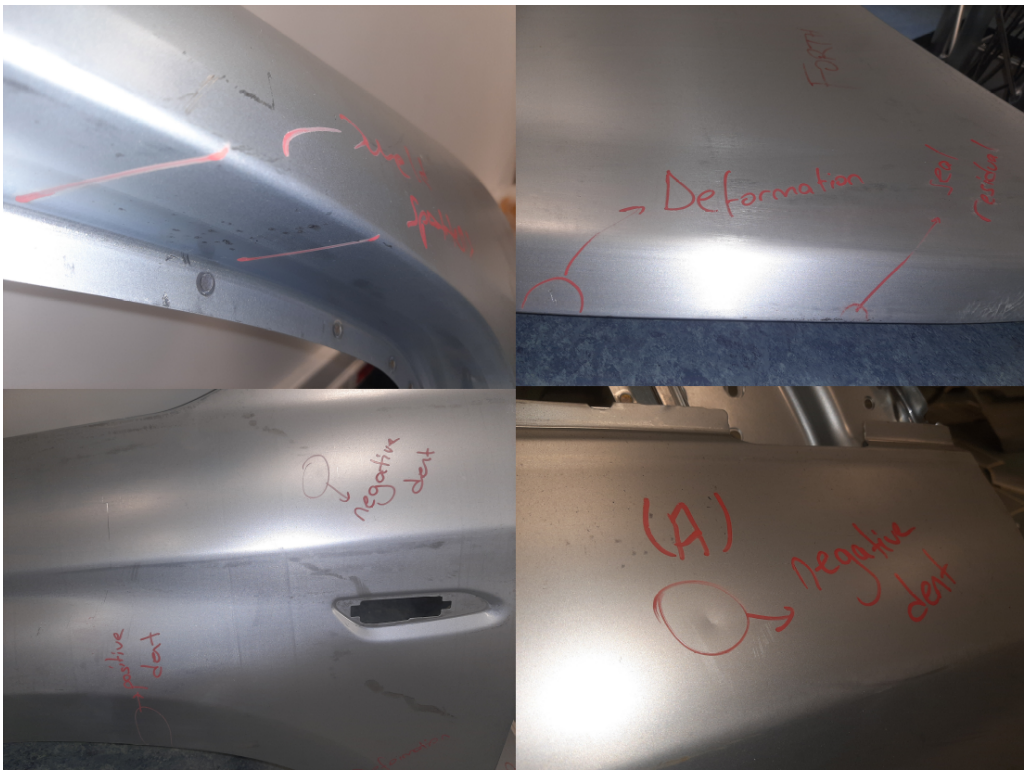


Figure 2.2. Annotated defect examples from the shipment of TOFAS to FORTH, of welding spots (top left), material deformation and seal residuals (top right), positive / negative dents (bottom row). Defects range in severity, size and location.

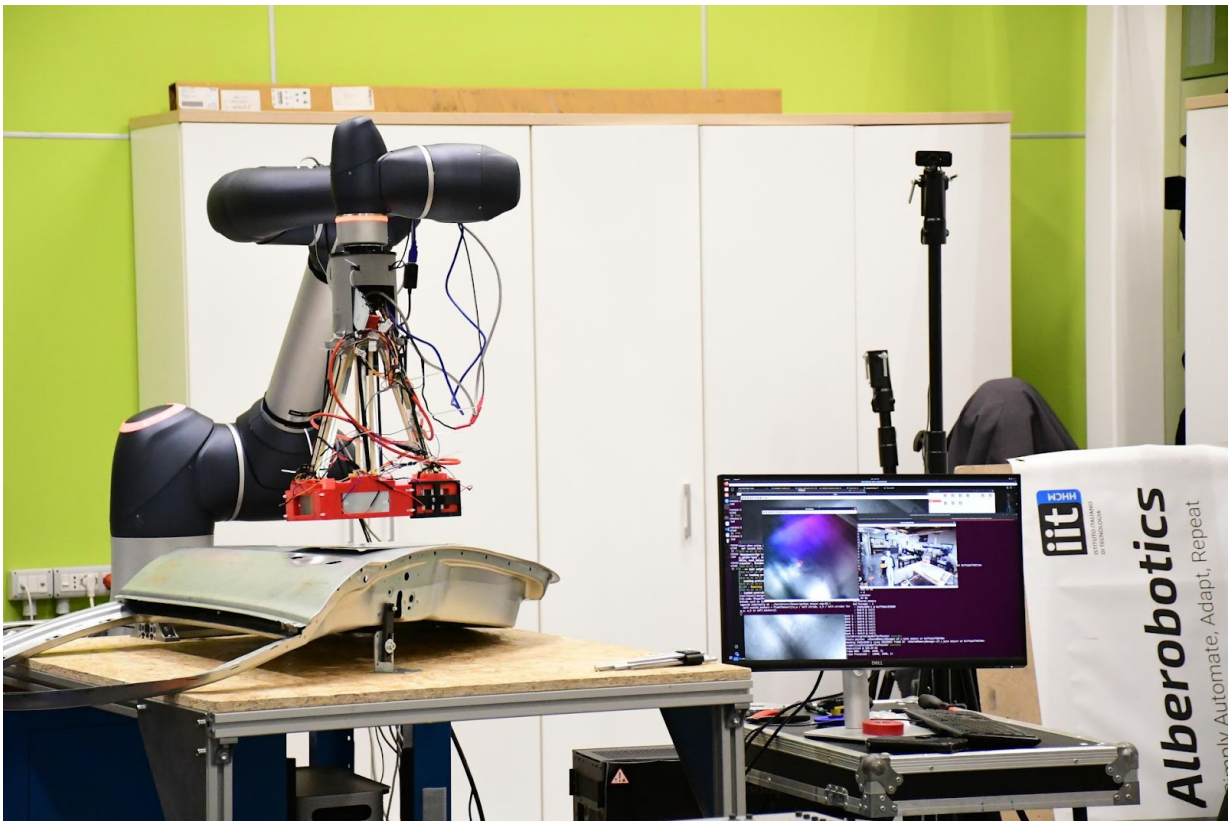


Figure 2.3. The MAGICIAN robot scanning surfaces and performing real-time defect classification during the 1st integration meeting in IIT premises.

As a result, a vision system capable of recording such defects requires exceptional optical clarity, with high resolution being crucial to ensure that defects produce a sufficiently large and detectable signature on the sensor. However, with increased resolution, more pixels are occupied for the same surface area in the data transfer between the sensor and the computer. This increase in pixel data also raises the processing demand, which can become a limiting factor for real-time operation.

2.2 OBJECTIVES AND REQUIREMENTS

Hereafter we report the KPIs, as described in the Deliverable D2.1 - “Use Case Definition”, related to the Visual Perception for Imperfections Detection. These KPIs (Table 2.1) define the constraints within which the developed visual perception solution must perform.

Scientific and technological objective	KPI ID	KPI definition	After MAGICIAN
(O1) A robotic perception module integrating visual and tactile sensors. The module will be	O1-KPI-SR1	Smallest size of defect that can be sensed/detected by the perception module.	≤0.3mm

embedded in a robotic sensor module (the SR, hereafter) and will be used for defects analysis and classification. The SR will replicate the skills of human workers through a learning scheme.	O1-KPI-SR2	Detection success rate vs humans.	False positives: ≤120% Skipped defects: ≤110%
	O1-KPI-SR3	Car-body scan time compared vs humans on a benchmark set.	≤110%

Table 2.1. KPIs related to the Visual Perception for Imperfections Detection

2.3 CAMERA SYSTEM

The MAGICIAN camera system for imperfections detection is based on a SONY XCG-CP510 sensor equipped with a global shutter polarization 5.1 Megapixel CMOS sensor and GigE interface. The sensor captures a polarized image where each individual pixel of each captured frame features one of four different linear polarization filters. This enables simultaneous capture of four polarization images at 0°, 45°, 90° and 135° with perfect synchronization. Since each of the 6 LED illuminators featured on the MAGICIAN sensor emit light with a specific polarization, defective areas cause light to be reflected in a non-uniform manner. The initial camera optics were chosen using a camera configuration simulator developed to calculate an effective selection of lens to accommodate the requirements of our project. The initial 12mm lens was selected to provide a balance between accuracy and scanning speed since a less zoomed-in view requires less time to complete the scanning operation helping us to comply with O1-KPI-SR3. After deploying this configuration however and despite the polarization camera that effectively provides 4 measurements for each pixel it became apparent that in order to accurately classify detects in near the minimum size limit (O1-KPI-SR1) better optics would provide a more clear view of the defects and thus allow the deploy neural networks to learn cleaner features, and perform classification with higher accuracy. The lens of the camera system was thus updated to a 16mm lens that reduced the effective view area to 14 cm x 10 cm when hovering 4.5cm over a surface.

The SONY PolarSense camera features connectors that expose the shutter state to external devices. Using them and an ADC converter the PolarSense sensor can be coupled with the Arduino microcontroller that controls the LED illuminators to ensure proper synchronization of light pulses with each camera frame for optimized lighting of the surface area. Although hardware synchronization would provide the ultimate possible precision in lighting control, it also comes with risks since it involves bridging electronics components from different circuits operating with different voltages and in the possible case of a short-circuit creates the risk of damaging the expensive PolarSense camera.

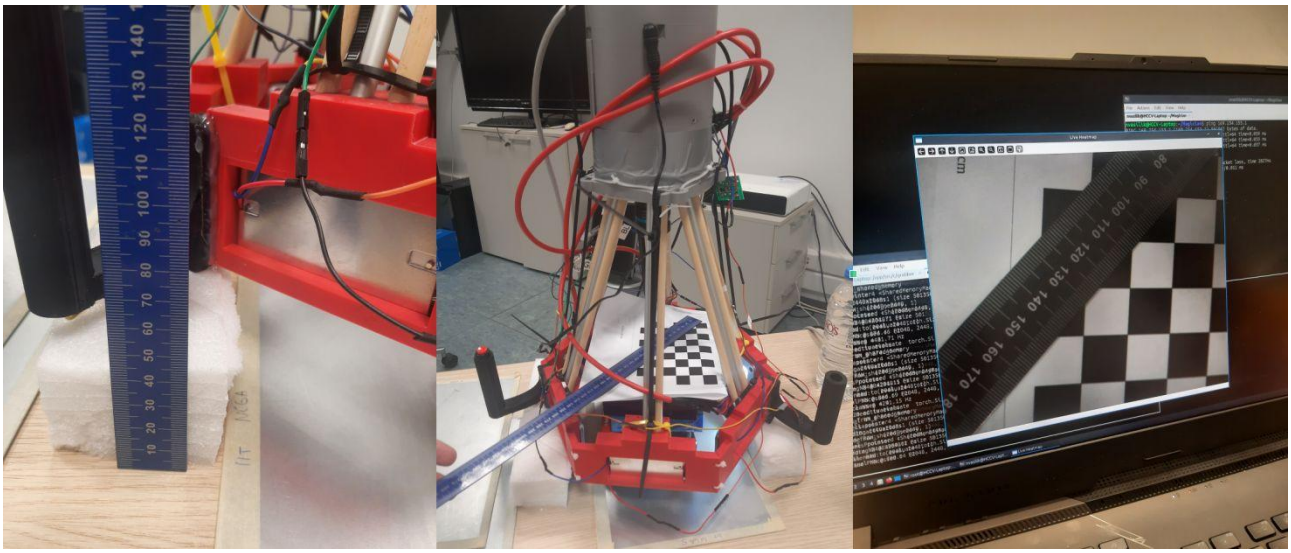


Figure 2.4. Left: A very pronounced negative dent with a diameter of 2 mm. Right: a negative dent with a 0.3 mm diameter (matching the 300micron KPI). The camera system described uses a 12 mm lens to optimize for the expected minimum defect size while accommodating the largest field of view that minimizes scan time.

The first naive implementation of the camera controller operated lights using a round-robin schema and operated with a clock that was completely unsynchronized with the camera. This approach sometimes led to surfaces being partly illuminated by more than one LED lights, especially when operating with high exposures. Although this also created some unique lighting patterns that could even prove useful in the future it was a simple proof-of-concept solution. Since the camera snaps & receives images at a maximum speed of 23Hz, this means that between two subsequent captured frames there is effectively a ~43 millisecond window to use to prepare the lighting for the next frame. We thus implemented a software-based synchronization routine built-in inside our grabber executable that gracefully handles lighting by switching LED state after reception of a frame. This provides an acceptable middle ground between no synchronization and hardware synchronization and ensures illumination from a single LED on each frame making polarization patterns more predictable.

Figure 2.5. Left: A diagram of the Vision Sensor and its various components. Right: visualization of different polarization images retrieved using the SONY PolarSense XCG-CP510 camera sensor.

The initial microcontroller selected for controlling the low-level peripherals such as Buttons, Accelerometers, ToF sensors and LED lights was Arduino MEGA 2560 (Figure 2.7). The main reason for its choice was its 4096-byte EEPROM (program memory) which was needed to populate data structures to interface with 3x instances of the vl53lxx-v2 range sensor code using the official API for the sensor provided by STMicroelectronics. Trying to use an Arduino NANO (1024-byte EEPROM) initially seemed infeasible. After optimization of the STMicro code and omission of API features that where not used compilation became possible on an Arduino NANO, however available SRAM (2KB in Arduino Nano vs 8KB in Arduino Mega) became the bottleneck resulting in situations where during execution and depending on the load of the microcontroller the stack pointer would surpass the heap pointer leading to memory corruption and a restart of the microcontroller. A written from scratch implementation for VL53LXX-V2 (https://github.com/pololu/vl53l0x-arduino) coupled with significant effort made Arduino NANO a viable choice for the camera system freeing up valuable physical space (Figure 2.7) allowing further miniaturization of the whole sensor.

The perception system 3D printed electronics housing (Figure 2.6) offers a precisely built slot that accepts the camera which is in turn also secured in place using the screwed in lens assembly. The light hexagon that provides polarized illumination to the camera is once again 3D printed and features 6x symmetric light holder pieces that can be meshed to form a hexagon. The lights used are based on a DC Torch assembly that feature a heatsink, a reflector and a transparent plexiglass for the LED where the polarization sheet is applied. The advertised power consumption of the LED cobs is 10W and they operate using 3V. Structural connection of the lighting hexagon with the electronics box is done using 18 beams with a Φ8 diameter and a 282mm length.

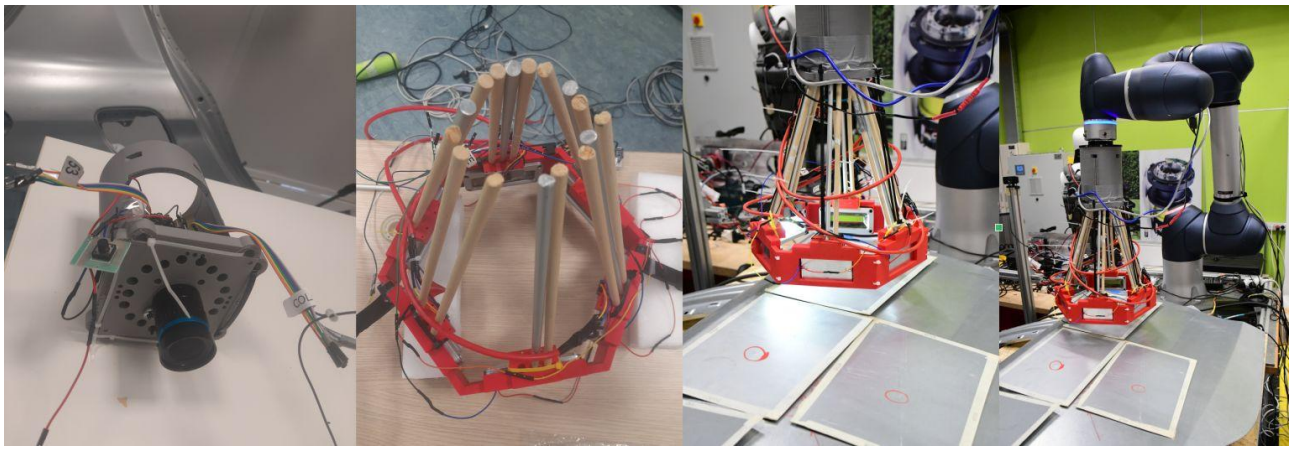


Figure 2.6. From left to right: The camera is currently attached on a 3D printed cylinder that also houses the rest of the electronics; to reduce weight while also providing good support a series of aluminium and wooden rods connect to the electronics box. The Vision Sensor can be mounted on the Doosan robot using a 3D printed mounting mechanism that allows the sensor to be used to scan metal surfaces for defects.

Six rods are made using aluminium providing lateral rigidness to the structure and the rest are made from wood to stabilize its rotation and reduce vibrations while also minimizing the sensor weight which is currently 1.85 kg.

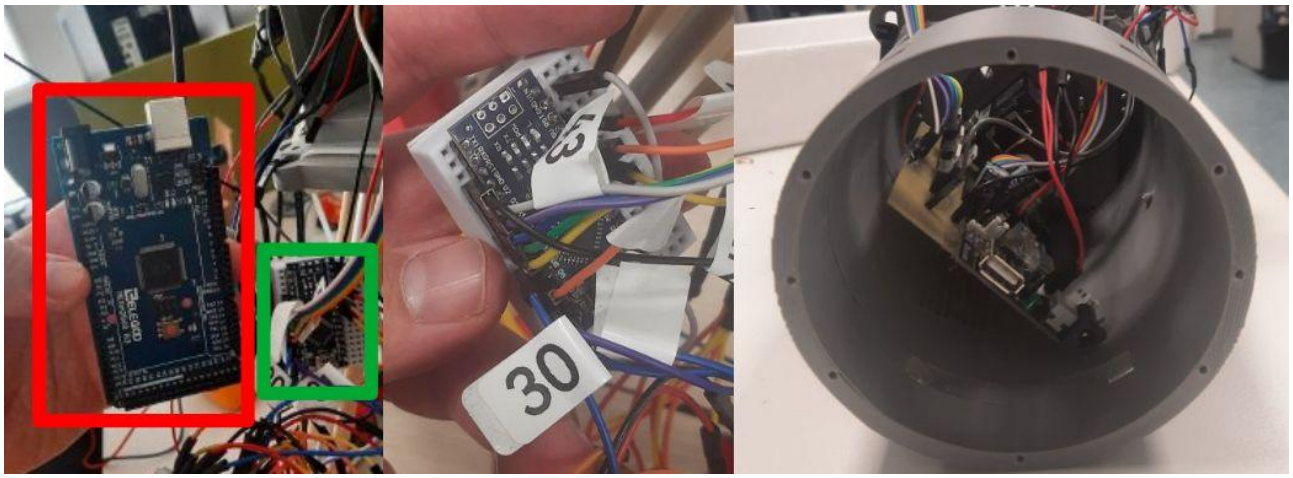


Figure 2.7. Left: The camera microcontroller was optimized from an Arduino MEGA (red rectangle) to an Arduino Nano (green rectangle) after successfully fitting the VL53LXX-V2 software to the very limited EEPROM/SRAM space of the smaller microcontroller. Right: The reduced footprint leaves much more free space in the electronics compartment, which is mainly occupied by the light power stage, allowing for a possibly smaller sensor size.

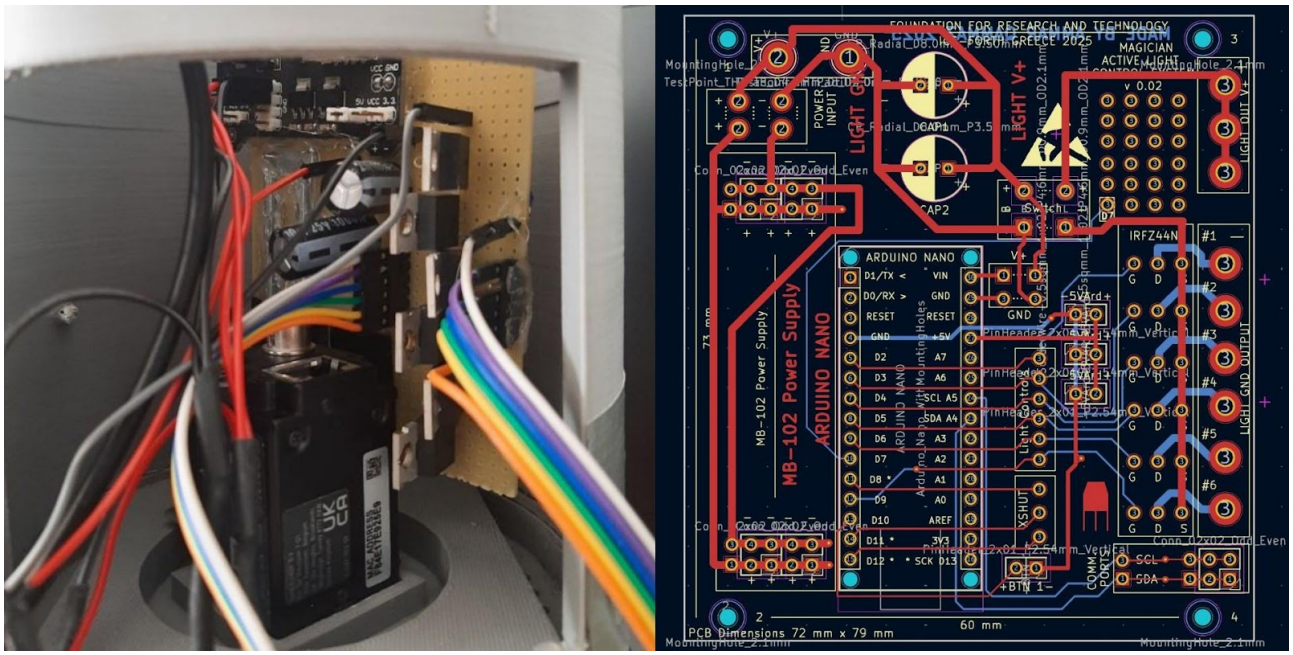


Figure 2.8. Left: The current temporary electronics switching board is implemented on a bread board and mounted next to the camera. Connections with various electronics (such as the Arduino) are facilitated using jumper wires. Right: The planned circuit for the camera system in its current state. Features such as the MB-102 may be removed outside of the sensor to further reduce required space.

The electronics circuit (Figure 2.8) during the initial early experimentation phase was made using a breadboard and uses IRFZ44N MOSFETs, capacitors and a MB-102 breadboard 5V/3V power supply to provide sufficient voltage to the Arduino and the LED lights. It's design has been robust enough to allow the capture of existing defect samples to kickstart the development of the neural network models, while its modularity allowed it to be adapted on the fly to changes to microcontrollers (Arduino MEGA → NANO), a variety of input voltages (7V – 12V) and was robust enough to handle the motions and vibrations of the Doosan robot even when operating at speeds of 20 cm/sec which are close to the 25 cm/sec which is the maximum velocity target of the robot. The finalization of the electronics design and move to an integrated PCB will ensure maximum robustness of its physical assembly and standardization that will allow easier production of the vision sensor.

2.4 DATA ACQUISITION AND ANNOTATION

2.4.1 DATA ACQUISITION

The MAGICIAN Grabber utility (Figure 2.9) has gradually evolved as a mature centralized software interface that can simultaneously handle all the data streams of the MAGICIAN sensor in a synchronised, transparent and easily programmable way. Although this

section, which builds upon the line depicted in D2.3, is part of the visual perception chapter, the MAGICIAN Grabber is a shared component responsible for acquiring both visual and tactile data streams. Tactile-specific outputs mentioned here are described in more detail in Chapter 3. The utility can either record data to disk and/or stream data to client applications. It is written from scratch with a multi-threaded approach in mind, dedicating a thread (possibly pinned to a distinct CPU core) to each data source for. It can also be compiled in 3 different configurations. Its first and simplest binary form is the *magician_grabber* which is entirely written in C and handles all input streams (including tactile), however without performing any processing of the input streams. Its second and more complicated binary is *magician_grabber_tactile* which also includes bindings to libfftw3 and the IIR library (https://github.com/berndporr/iir1). This version of the library processes tactile force and accelerometer data in real-time calculating friction, acceleration spikiness, and power spectral density (PSD) for the exerted forces and accelerations experienced by the sensor. Calculating these representations of the incoming force data simplifies the requirements of the tactile classifier by locally performing computations in a context where the data is closely synchronized. The final binary generated is the *ros_magician_grabber* which includes the dependencies to the ROS library and can interface the data streams using ROS topics and services. All topics broadcasted by the ROS version of the *magician_grabber* are published under the */magician_grabber/* prefix to make them easily distinguishable from other ROS topics. In more detail: Force data is broadcast under */magician_grabber/wrench_sens*ed using a *WrenchStamp*ed geometry message. This allows force data to be used by the robot control nodes to also estimate contact forces during manipulation. Accelerometer data are broadcast at the *magician_grabber/accel_sensed* topic using an *AccelStamped* message. It is worth noting that *AccelStamped* features both a linear and angular component, however the data produced by the MAGICIAN sensor does not contain angular information, so this component is set to zero. Distance values from the time-of-flight sensors are emitted as Float32 values in millimetres in the */magician_grabber/distance1-3* topics. Active LED lights of the sensor can be intercepted using the */magician_grabber/light1-6 topics* and */magician_grabber/button* broadcasts the state of the physical button that is used for the annotation of tactile data.

All versions of the grabber accept the same command line parameters and can be configured using the same commands. Issuing the command line parameter *–help* displays a list of available commands and parameters as seen in Figure 2.9.

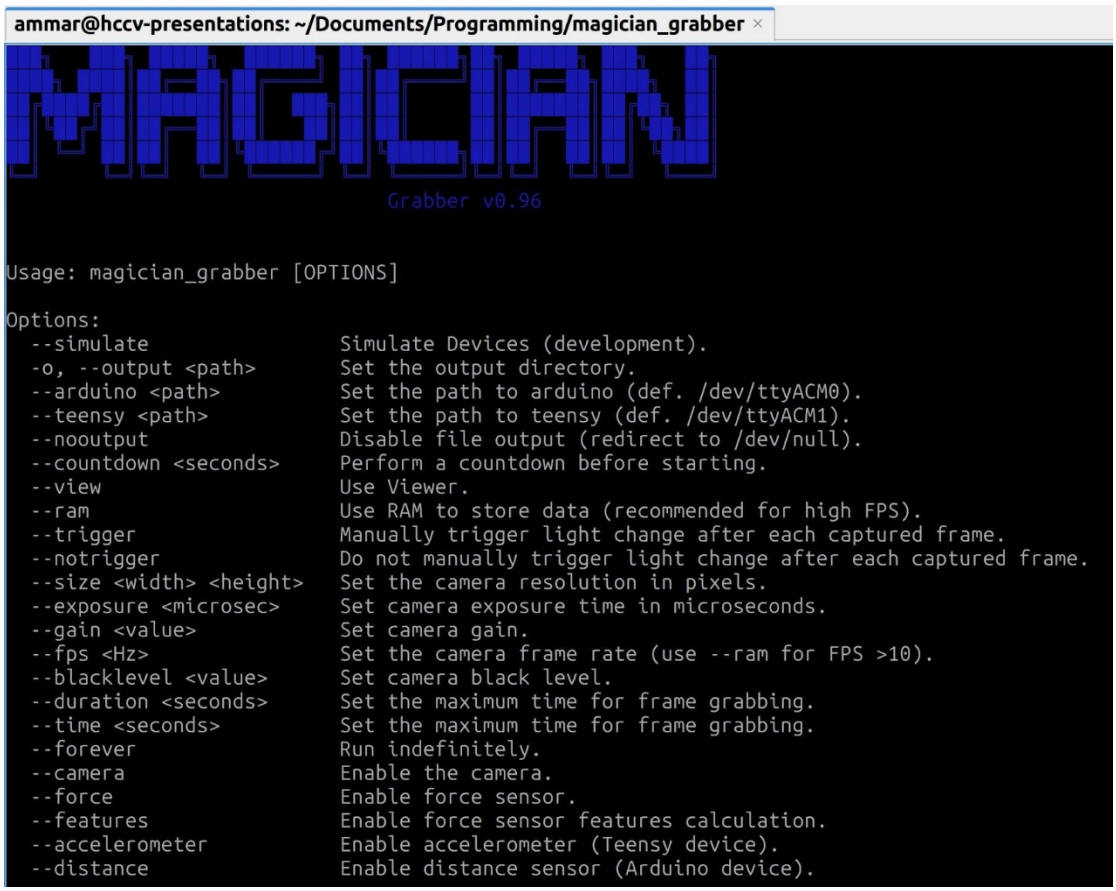


Figure 2.9. The MAGICIAN grabber utility listing all available commands when started using *–help*.

A data capture session can be accommodated using the following command:

./magician_grabber_tactile --all --size 2448 2048 --exposure 4500 --fps 23 --time 45 --countdown 10 --output ***ExperimentName***

This will provide 10 seconds of lead time for the user to take the sensor in his hands and position it over the area with defective samples, grab using all available streams (camera, force sensor, accelerometer, etc.) with an exposure of 4500 microseconds acquiring a video stream of 23 Hz for 45 seconds. This is a typical configuration when grabbing datasets.

A ROS broadcasting session involves just running *./ros_magician_grabber* that automatically enables the *--all, --nooutput* and *--stream* commands.

A dataset generated on disk features files of three different file formats, JSON, CSV and PNM. **ExperimentName/info.json** is the first file generated that contains the grabber configuration for the specific dataset. **ExperimentName/colorFrame_0_xxxxx.pnm** files contain the recorded images using a lossless representation, with each image occupying 4.8MB on disk. **ExperimentName/controller.csv** contains information about peripherals connected on the Arduino NANO (Figure 2.7). The CSV header contains

System Timestamp (Unix Time), *dev_timestamp* (Arduino loop time in msec), Button state, Distance1-3 (VL53LXX-V2 ToF laser range sensor reading in millimetres), Light1-6 LED power switch state with values 1 or 0. Values follow the capture framerate of the Polarized Camera to align LEDs and distances to the incoming frames. Sampling rate for messages is typically 10 - 23 Hz depending on how fast frame acquisition is set up (command line argument -- *fps XX*).

ExperimentName/tactile/force.csv records the force sensor readings from the ATI Net F/T force sensor recording timestamp (Unix timestamp when the message was received in microseconds), *fX, fY, fZ* force data and *tX, tY, tZ* torque data. Sampling rate is 7 kHz but can be reduced using the ATI Web interface.

ExperimentName/tactile/accelerometer.csv records timestamp (Unix timestamp when the message was received by the host system in microseconds), *dev_timestamp* (Teensy microcontroller loop time in milliseconds) and *accX, accY, accZ* acceleration measurements. Sampling rate is 4 kHz but can be reduced by re-flashing the teensy microcontroller.

Finally, if the *magician_grabber_tactile* binary is the one used, it will automatically emit the following files:

ExperimentName/tactile/acceleration_psd.csv

ExperimentName/tactile/acceleration_spikeness.csv

ExperimentName/tactile/force_psd.csv

ExperimentName/tactile/friction.csv

Each of these files contains a pair of values, the first is a timestamp which has been synchronized between the acceleration and force data (even if they operate at different rates), and the second column of data contains the processed value for each specific measurement. These measurements can be directly used by the tactile classification module to simplify calculations

2.4.2 DATA ANNOTATION

To annotate the data, we developed a tool based on the cross-platform *WxWidgets* library, specifically its *wxPython* wrapper. Upon execution the tool opens a GUI window that accepts a path to a dataset and proceeds to open it and allow visual inspection of recorded samples and given a few clicks highlight the defect type and classification which will in turn be used by the neural network stack to perform learning and successfully tackle the task.

Data annotation is a time consuming and repetitive task that consumes a lot of human resources that could be dedicated to other parts of the MAGICIAN project. Furthermore, physical details in the lighting system, camera lens, focus, and physical geometry of the sensor invalidate previous training efforts giving rise to new patterns and requiring new data acquisition and data annotation to be properly supported. Since the start of the

project, we have consistently tried to improve this aspect and facilitate data annotation in a way that yields time savings with a cascading positive effect to the rest of the project. After experimentation with the Segment Anything (SAM) foundation model we found it did not output consistent results with respect to automatic segmentation of the defective areas. However, we continuously monitor the literature for new methods that could be applied to our data annotation task to improve it.

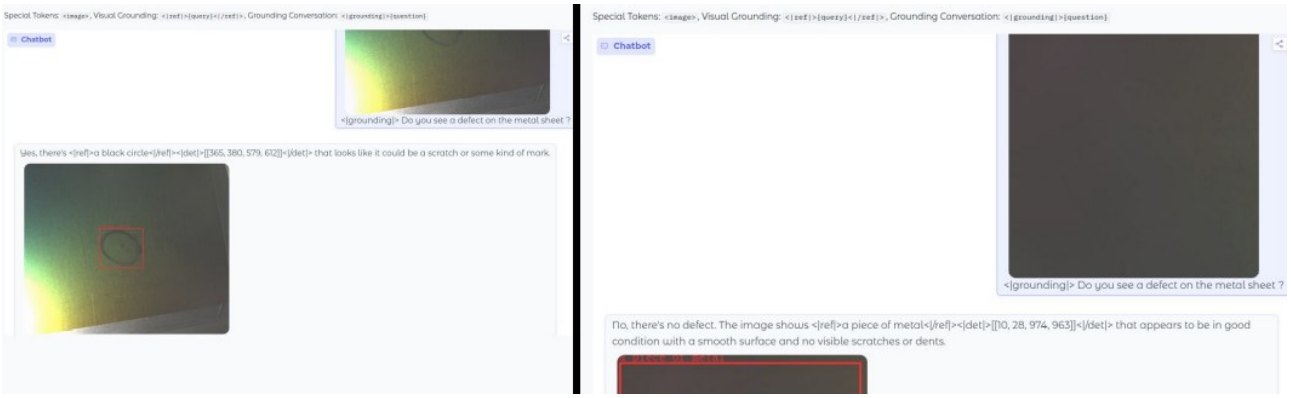


Figure 2.10. By using the DeepSeek VL2, vision language model (VLM) we can automate questions using physical human language to facilitate some degree of dataset sorting to optimize the time needed by the human annotator.

A tool that recently became available to the computer vision community is large vision language models (VLMs) inference. They first became available through Meta's LLAMA (https://arxiv.org/abs/2407.21783) that featured the capacity to process both language and image inputs and respond with textual tokens, combining both image perception and a good degree of problem-solving logic. However, due to legal issues with the European Union, LLAMA was not made available in the EU therefore making it a non-applicable choice with respect to the MAGICIAN project. Shortly after however the Chinese DeepSeek company gave Deep Seek VL-2 to the public.

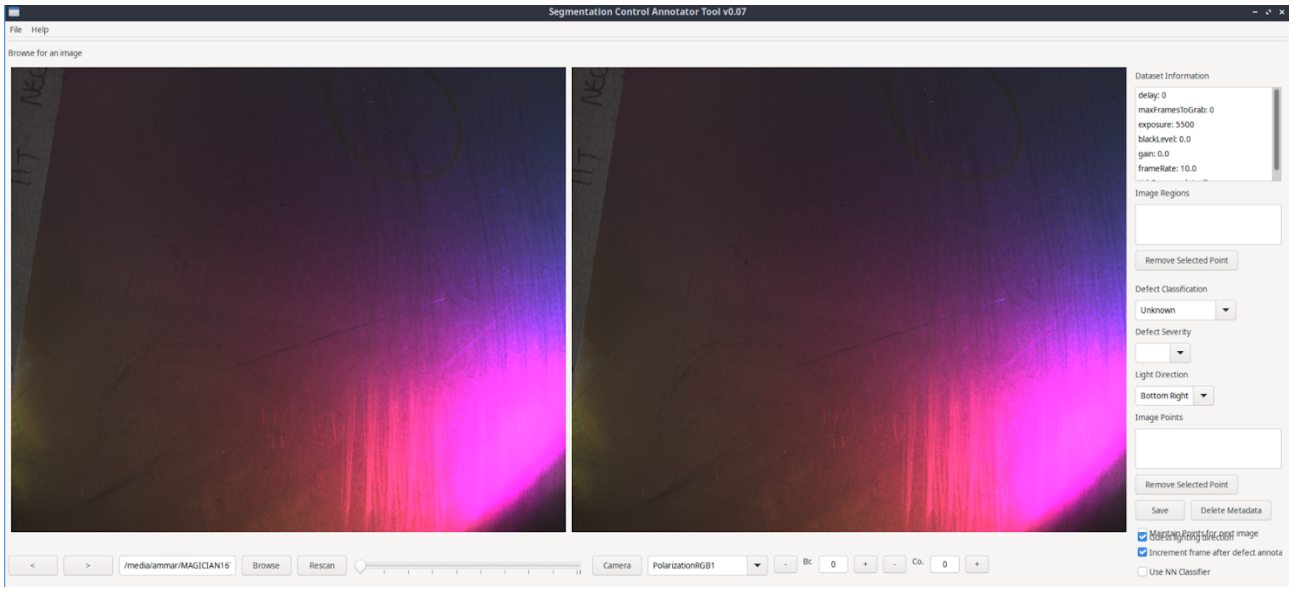


Figure 2.11. The annotation tool graphical user interface in its current state while processing a recorded dataset.

The permissive MIT license of the DeepSeek VL2 code combined with the very low requirements of its DeepSeek-VL2-tiny quantization that can be executed in real-time on computers with GPUs featuring 16GB VRAM made it an ideal candidate for experimentation. DeepSeek VL2 has been trained on over 6 billion images and can tackle a wide variety of problems with excellent accuracy with respect to its size. Questions to the model take the form of tokens accompanied by images. The network can perform tasks such as Optical Character Recognition (OCR), image summarization, document and graph understanding as well as visual grounding. Utilizing the visual grounding capability and performing well defined questions that have clear-cut answers that are easily elaborated we can take advantage of their capabilities for annotation. By creating queries that feature the image grounding tokens, as seen in Figure 2.10 we can extract bounding boxes for defect types that can then be semi-automatically annotated. The network is capable of also providing coordinates on the image encoding the locations of the areas of interest. Given a recent computer system, queries to the VLM can be served at a rate of 0.45Hz. This speed is comparable to the time required by a human to perform annotation, however sometimes the response of the network may involve a bounding box that also contains the marking of the defect (see left image on Figure 2.10). Due to the very high importance of correct annotation data annotation is still performed manually, however working on this automation capability may also prove useful during the cascade funding that will introduce more use cases to the project.

2.5 METHODOLOGIES EMPLOYED

Developing the prototype system presented here involved usage and testing of various methodologies. Both our algorithm and the neural network classifier are now ported in PyTorch Lightning and PyTorch. The algorithm logic is mostly the same but re-written

in PyTorch to use the GPU. Also, a new component was added into the algorithm, a majority vote between the tiles after the neural network prediction. The neural network components are now changed. What is more, the loss function which used to be a Categorical Cross Entropy loss is now changed to Focal Loss. Focal Loss performs better when there is data imbalance and when it is difficult to distinguish between easy and hard examples. In our case there is a significant data imbalance between metal samples that are normal and those that have a defect. There is also difficulty in distinguishing the negative dent defects. Therefore, Focal Loss was a straightforward choice in our update method.

We experimented with recent convolutional neural networks (CNNs) and transformers. More specifically, ResNet-18, ResNeXt-50-32x4d, ConvNeXt Small, EfficientNet V2 S and Swin V2 T (Transformer). In all such models, the first layer was modified for input with 4 channels instead of 3 (RGB), 1 channel represents each polarization. Moreover, the final fully connected layer was modified to output 3 classes, one representing positive dents, another for negative dents and the third one for clean metal samples. In addition, our updated method is containerized using Docker. Consequently, it is now easier to deploy it without requiring a specific operating system or hardware, if it is supported by Docker.

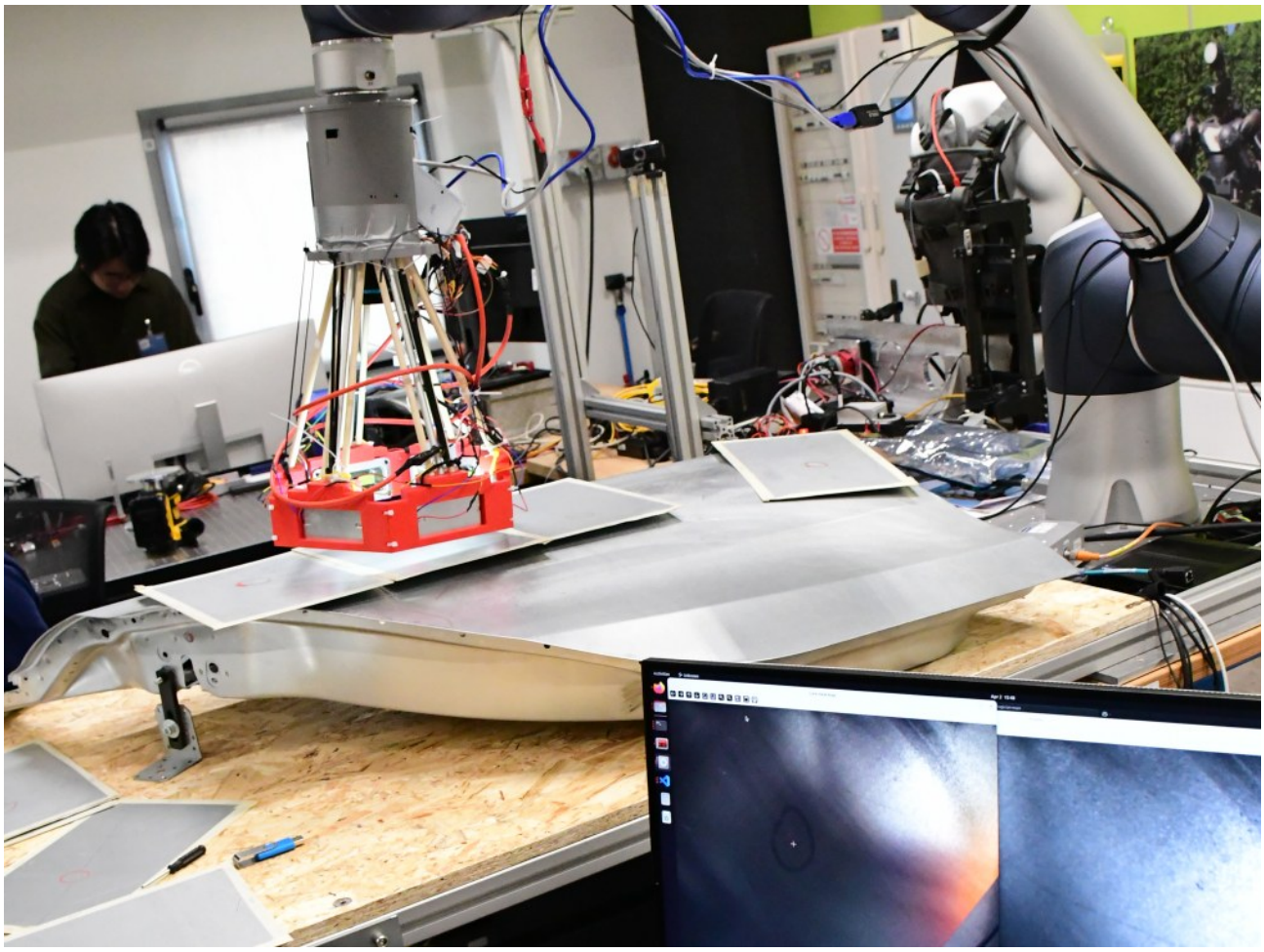


Figure 2.12. Defect detection running in real-time on the MAGICIAN platform and detecting a negative dent defect during the 1st integration meeting.

2.6 RESULTS AND FINDINGS

As described above, we experimented with different models. The models described are trained only using the metal samples delivered to FORTH. The same train/validation split, and the same seed were used in all experiments.

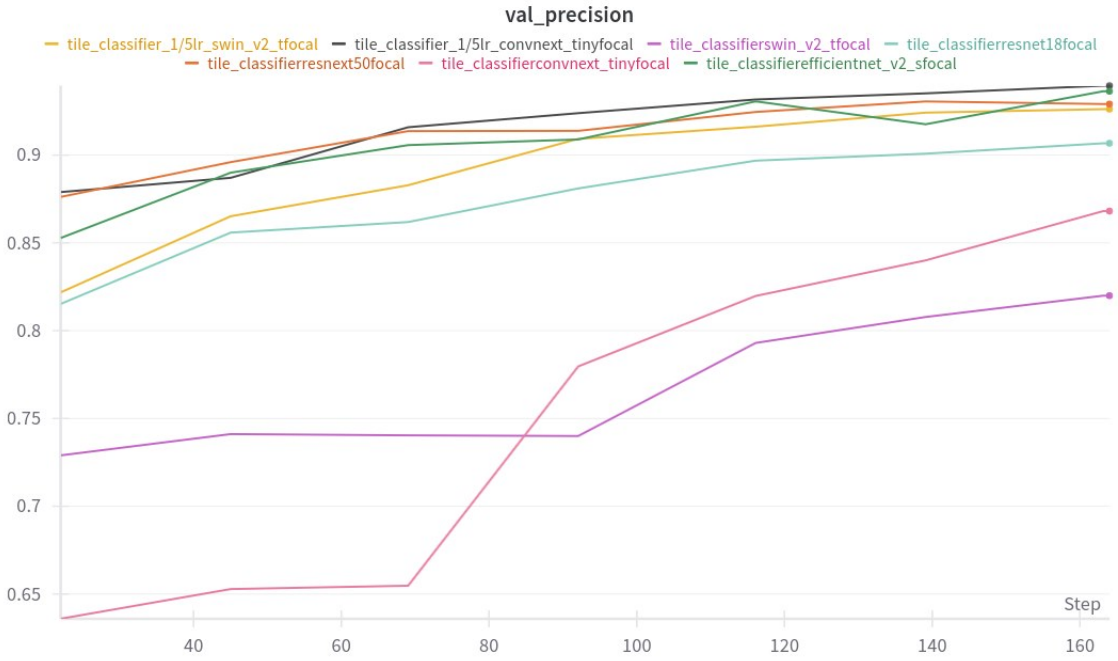


Figure 2.13. The plot shows the precision of each model in the validation set.

Figure 2.13 demonstrates precision over 90% from step 120.

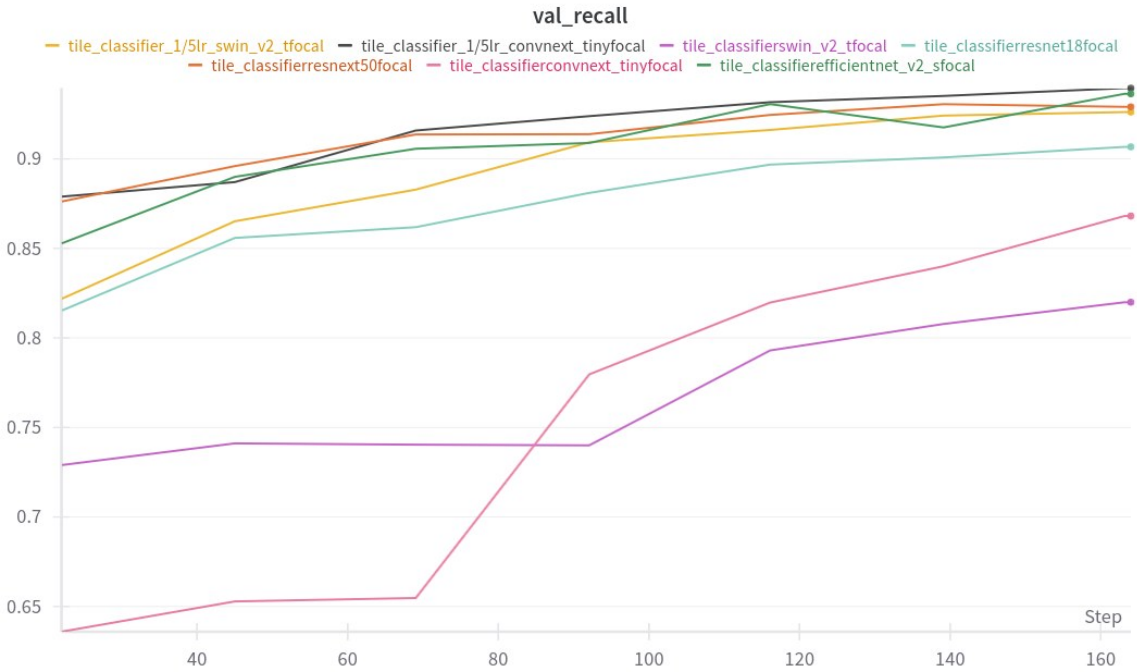


Figure 2.14. The plot shows the recall of each model in the validation set.

Figure 2.14 shows the recall of the models which in our case indicates that all models

have a low false negative percentage which means that they do not miss the positive and negative dents on the metal samples.

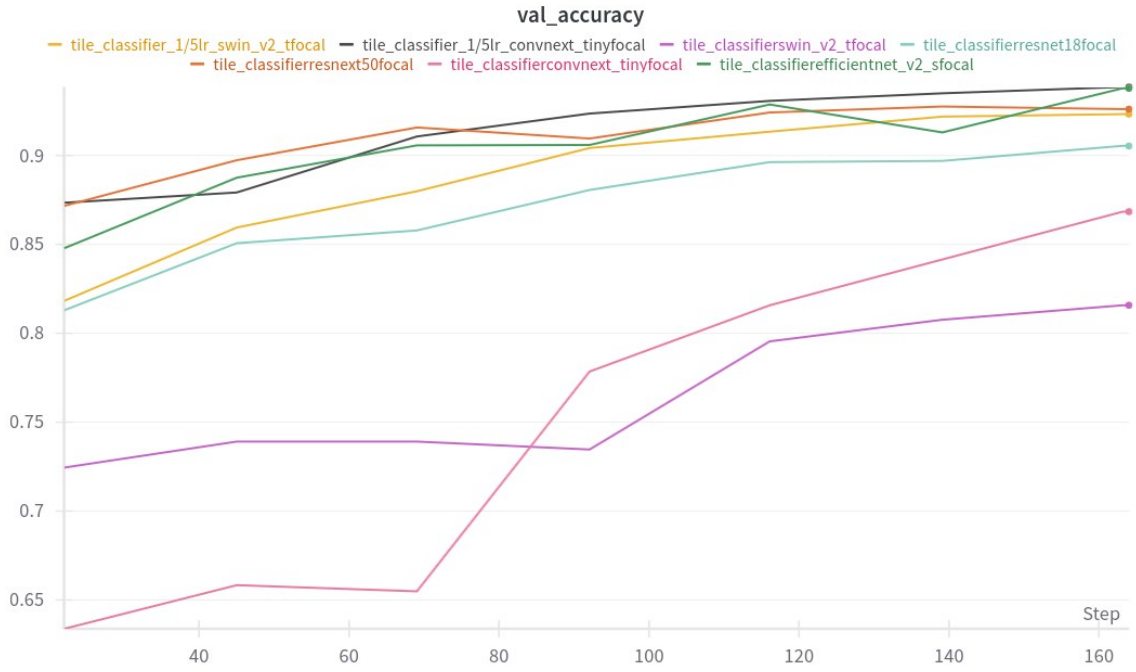


Figure 2.15. The plot shows the accuracy of each model in the validation set.

Figure 2.15 shows over 90% accuracy after 100 training steps.

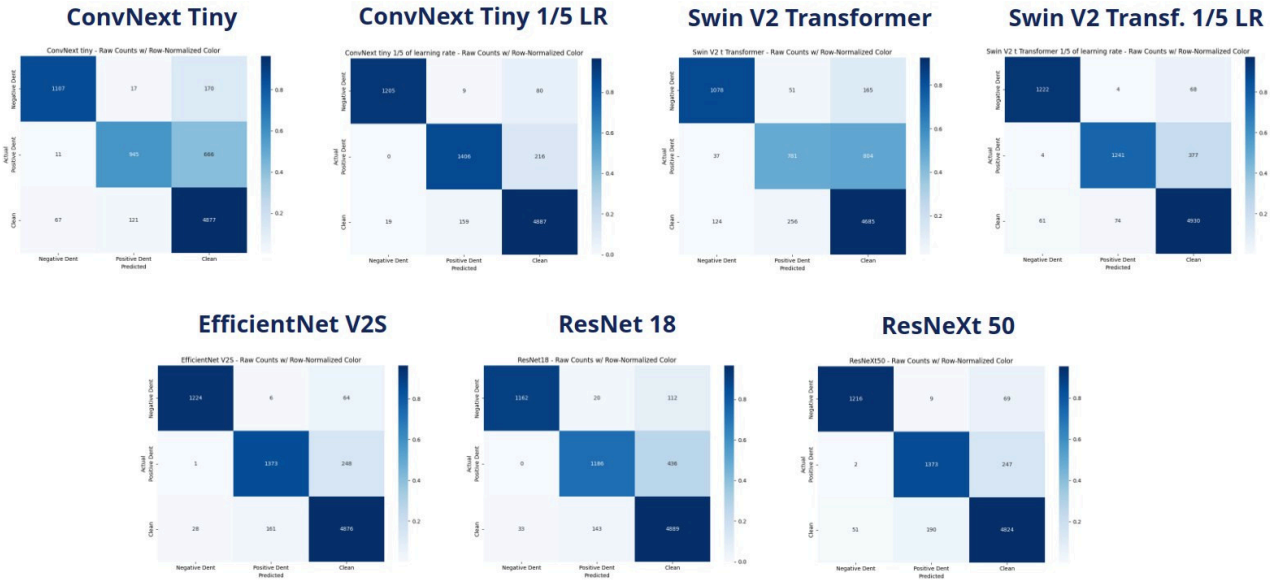


Figure 2.16. Confusion matrices for each of the developed visual defect classification models.

As shown in Figure 2.16, all the models perform well with small false positive rates.

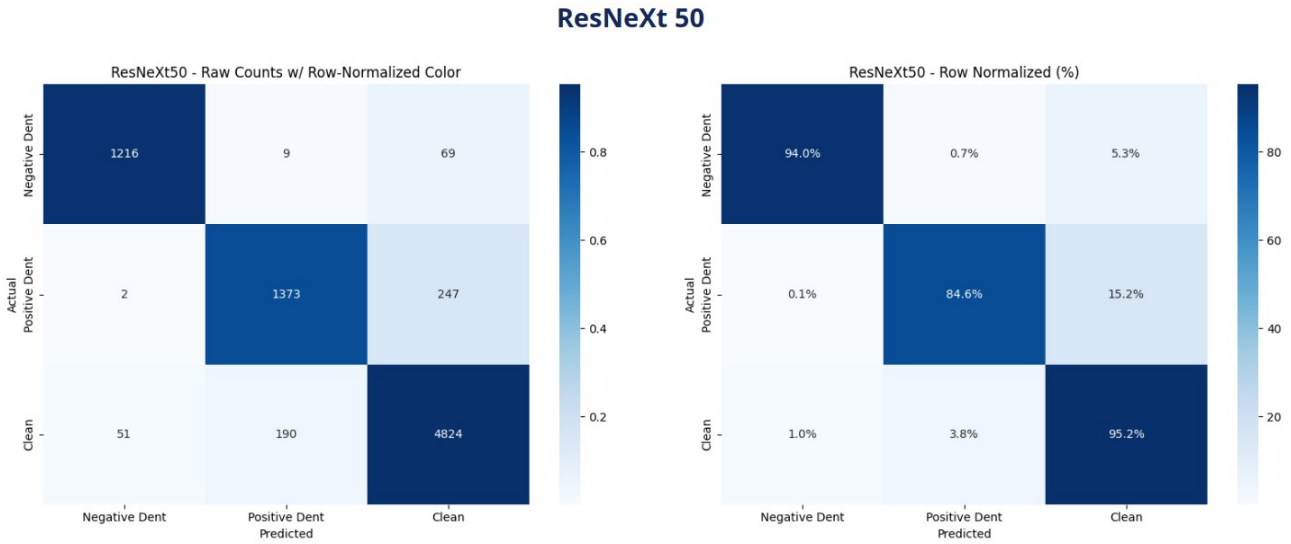


Figure 2.17. Confusion matrices for the best configuration experimentally derived.

Figure 2.17 shows our best performing model ResNeXt-50. The right confusion matrix is the same with the left one but normalized. This experiment proves that our model has minimal false positives, and few false negatives seem to appear in positive dents.

2.7 CHALLENGES AND LIMITATIONS

Given the initial considerations our development attempted to treat the challenges and limitations identified in D3.1. By increasing the number of LED illuminators to 6x, introducing laser range sensors, improving lens magnification and creating a 3D printed chassis for the vision sensor we have a much more robust sensor, that however has new identified challenges and limitations that will need to be improved in the future steps of development of the system:

1. **Sensor Hardware and dependencies**

- The use of a polarization camera has been consolidated since it allows for a rich source of patterns on the observed surfaces. Furthermore, by using unpolarized light and combining all 4 polarization channels we can in principle have a regular monochrome camera stream in case of a use-case during the cascade funding where light polarization is not beneficial to an application. By creating a standard 3D printed arrangement we have managed to record consistent datasets.

- During the IIT integration week, Altinay partners expressed the need for a smaller sensor size that hovers at a higher distance with respect to the scanned surface.

This is because the current sensor's size may not allow the robot arm to reach all visible areas due to possible collisions with the car chassis.

- The current prototype sensor has a weight of 1.85kg and a height of 48.5 cm. Its 3D printed materials however cause noticeable vibrations to the sensor chassis especially when the robot is moving close to the maximum velocity of 25 cm / sec. Manufacturing a metal version of the sensor will improve of this however the weight of the sensor must be kept in check since the maximum weight that can be handled by the robot is 25 kg.

2. Illumination Challenges

- Although we implemented a 6-directional LED light system using an advertised 10W LED torch, ambient illumination can vary wildly in scenarios with direct sunlight or other lighting sources. In cases with severe ambient light the polarized light becomes less pronounced as a signal. The solution to this problem is the use of even stronger illuminators operated using pulses synchronized to the camera shutter. This technique will allow lower exposures (e.g. 650 microseconds) that will make ambient light disappear even in cases where it is very severe.

3. Focus and Distance Sensing

- The camera sensor has a fixed focus lens. There are auto-focus solutions that can automate focus; however, these continuously and incrementally alter focus leading to a high percentage of blurred frames on fast moving cameras, something which is unacceptable for our application.

- Robot trajectories currently have a distance that makes the sensor hover approximately 4.5 cm above the scanned surfaces. However, the presence of hard-to-reach surfaces may prompt different distances if the sensor size is not reduced. The existence of distance sensors helps with passively maintaining a correct hovering height.

4. Dataset Size and Deep Learning

- Despite TOFAS shipping hundreds of kilograms of materials, these are ultimately few samples for deep learning standards. For example, since all the door frames we have received have welding spatters, it is possible for a neural network to associate the contours of doors with welding spatters, leading to a method that will not generalize well in the actual tasks.

- The use of the tiled image approach in combination with multiple lighting sources and the scanning of samples shipped to other partners mitigates this problem as well as the problem of learning the markings instead of the defects to the best of our ability.

3 TACTILE PERCEPTION SYSTEM FOR IMPERFECTIONS DETECTION

3.1 INTRODUCTION

The tactile perception system constitutes the second fundamental component within the MAGICIAN project, complementing the vision system in the detection and classification of surface imperfections. The core challenge addressed by this module is to replicate the refined manual dexterity of human operators, who rely heavily on tactile feedback to discern and categorize defects on car body surfaces. Following the industrial visit to TOFAS in January 2024, IIT received a substantial shipment of annotated defective materials, which proved crucial for assembling the first tactile dataset. This dataset includes force and acceleration measurements recorded during scanning procedures, leveraging the sensing technologies and methodologies previously established in D3.1.

3.2 STATE OF THE ART

In this phase of the project, the state-of-the-art review provided in D3.1 remains fully valid and applicable. The technologies, methods, and research directions previously analyzed continue to represent the scientific and technical foundation for the ongoing development activities within Work Package 3. The references to tactile sensing technologies, sensor fusion strategies, haptic feature extraction, and data-driven classification frameworks, as discussed in D3.1, remain relevant for both the acquisition and processing pipelines adopted in this deliverable.

As the project progresses into more advanced validation and system integration phases, future updates of the state of the art may include comparative studies with more recent publications or emerging techniques in deep tactile learning and multimodal defect detection. However, at the current stage, the foundational knowledge established in D3.1 continues to guide and support the development of the MAGICIAN tactile perception module.

3.3 OBJECTIVES AND REQUIREMENTS

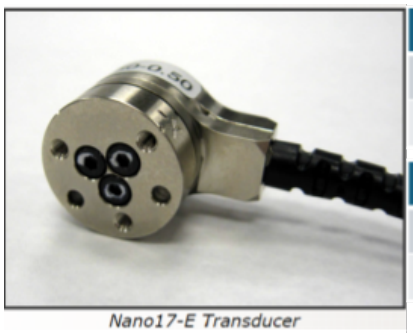
The objectives defined for the tactile perception system remain unchanged from those outlined in D3.1. The Key Performance Indicators (KPIs) originally reported, as established in D2.1 – "Use Case Definition," continue to represent the reference metrics for evaluating the performance of the system under development. As in the case of visual perception, these KPIs (Table 3.1) define the operational constraints and expected capabilities of the tactile perception solution for imperfection detection.

Scientific and technological objective	KPI ID	KPI definition	After MAGICIAN
(O1) A robotic perception module integrating visual and tactile sensors. The module will be embedded in a robotic sensor module (the SR, hereafter) and will be used for defects analysis and classification. The SR will replicate the skills of human workers through a learning scheme.	O1-KPI-SR1	Smallest size of defect that can be sensed/detected by the perception module.	≤0.3mm
	O1-KPI-SR2	Detection success rate vs humans.	False positives: ≤120% Skipped defects: ≤110%
	O1-KPI-SR3	Car-body scan time compared vs humans on a benchmark set.	≤110%
	O1-KPI-LRN SR1	Misclassification rate with respect to humans.	≤10%
	O1-KPI-LRN SR2	Time to convergence.	Observation time ≤ 15h to achieve KPI LRN-SR1

Table 3.1. KPIs related to the Tactile Perception System for Imperfections Detection.

3.4 TACTILE SENSORS

The tactile sensors employed in this phase remain consistent with those described in D3.1. The sensing unit integrates a triaxial force sensor (ATI Nano17) and a triaxial accelerometer (Analog Devices ADXL335). The force sensor offers high sensitivity and resolution, enabling the detection of subtle surface irregularities during contact-based exploration. The accelerometer complements the system by capturing dynamic signals associated with rapid changes in surface texture and contact events.



Nano17-E Transducer

SENSING RANGES			
Fx, Fy	Fz	Tx, Ty	Tz
0-12 N	17 N	120 Nmm	120 Nmm

RESOLUTION			
Fx, Fy	Fz	Tx, Ty	Tz
1/320 N	1/320 N	1/64 Nmm	1/64 Nmm

SENSING RANGES
Ax, Ay, Az
-3 to 3 g

RESOLUTION
Ax, Ay, Az
6 mg

Figure 3.1. The ATI Nano17 force sensor and ADXL335 accelerometer, along with their respective resolutions and sensing ranges.

These two sensors can be effectively utilized for detecting surface defects when integrated into an exploration tool. Specifically, IIT integrated the proposed sensor set with diverse end effector designs in the MAGICIAN tactile perception system, enabling the exploration of a broad range of solutions. The devices built to incorporate these sensors, along with the various types of end-effectors, are detailed in D4.1.

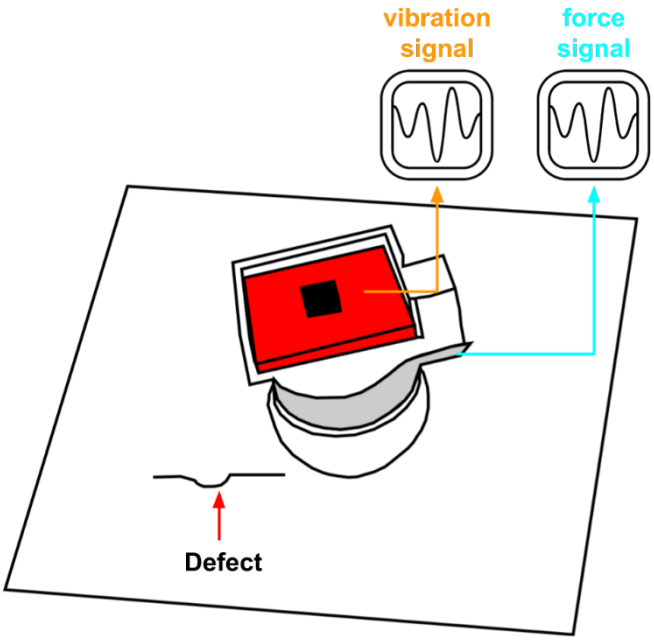


Figure 3.2. Schematic overview of the force and acceleration sensors mounting on the tactile sensor probe. The proximity to the probe ensures minimal signal attenuation during data acquisition. When the probe is in contact with the surface being scanned, corresponding force and acceleration signals are recorded, enabling the detection of potential defects through the collected data.

3.5 DATA ACQUISITION AND ANNOTATION

In this second phase of the project, the tactile data acquisition pipeline was extended and refined to support synchronized, multi-modal data collection on real car body frames. The updated campaign involved scanning 28 real car body frames, each annotated by TOFAS operators for defect type and severity. These frames were selected to include a variety of relevant imperfections (e.g., positive/negative dents) covering different severity levels, thereby enhancing the representativeness of the dataset.

Each acquisition trial was performed by 8 users, who explored the car body surfaces using the custom-developed tactile device, already documented in D4.1, instrumented with a triaxial force sensor (ATI Nano17) and a triaxial accelerometer (Analog Devices ADXL335). These sensors provided raw data streams at high temporal resolution: the force sensor was sampled at 7 kHz, while the accelerometer was sampled at 4 kHz. This high-rate acquisition ensures sufficient resolution for detecting transient contact events and subtle dynamic variations during defect exploration.

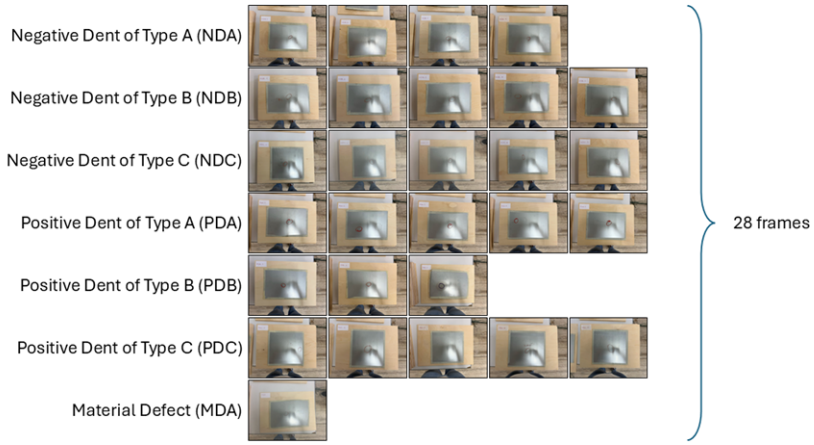


Figure 3.3. The 28 car body frames used for data acquisition. Each frame was mounted on identically sized wooden panels to ensure stable positioning on the desk during recording. This setup guaranteed consistent defect locations within the VICON reference frame, enabling automatic labelling of signals corresponding to defect presence.

To integrate positional information, a Vicon motion tracking system was employed, providing real-time 6-DoF tracking of the tactile probe relative to the car body with sub-millimeter accuracy. The synchronization between the tactile sensing data and the Vicon tracking data was achieved through a redesigned data acquisition architecture based on LabVIEW. The LabVIEW application, originally developed for standalone sensor acquisition, was modified to support synchronous data collection across all sensing modalities. Specifically, the updated LabVIEW software includes:

- A unified acquisition loop triggered by a shared system clock, ensuring temporal alignment between tactile and motion data.
- Direct interfacing with the Vicon DataStream SDK via dedicated LabVIEW nodes to ingest and timestamp pose data at a nominal rate of 100 Hz.
- Real-time buffering and export routines to record synchronized data streams in a structured format for subsequent processing.

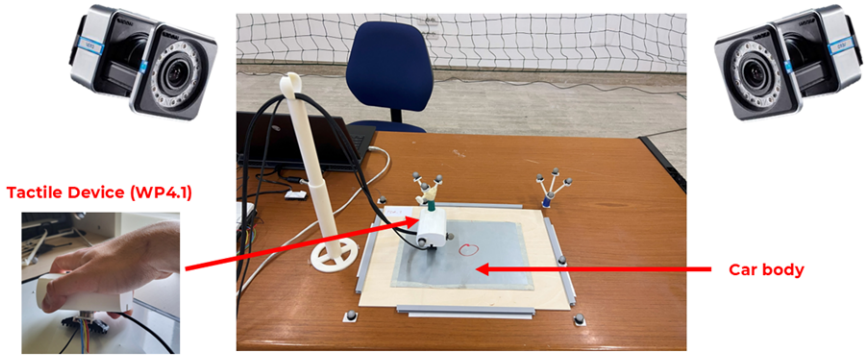


Figure 3.4. Acquisition setup. During data acquisition, the car body is placed on a desk while the user scans its surface using a handheld device equipped with a tactile sensor. Simultaneously, the VICON tracking system records the device's position.

During each trial, users were instructed to maintain consistent contact with the surface, scanning across the entire panel area while naturally reacting to surface texture and defects. The Vicon tracking system made it possible to automatically label the tactile data by determining when the probe passed over a predefined defect region. A binary flag was appended to each measurement frame to indicate the presence or absence of a defect, based on spatial correspondence between the recorded trajectory and TOFAS annotations.

A total of 280 valid acquisition sessions were recorded, each lasting on average 20 seconds. The resulting dataset consists of time-aligned streams of:

- 3-axis force (7 kHz),
- 3-axis acceleration (4 kHz),
- 6-DoF pose (Vicon, 100 Hz),
- Binary defect presence flag (generated offline via trajectory matching).

All trials are organized in a hierarchical folder structure by user, defect type, and probe, following the same naming conventions introduced in D3.1. Each acquisition includes raw time series (.csv) files and metadata describing trial conditions (e.g., probe type, defect annotation, user ID). This enriched dataset represents a critical milestone in WP3, enabling not only improved training of tactile-based classifiers, but also facilitating the study of user exploration strategies, thanks to the synchronized positional data. The integration of tactile sensing and motion tracking in a unified acquisition framework lays the foundation for advanced learning-from-demonstration methods in later project phases.

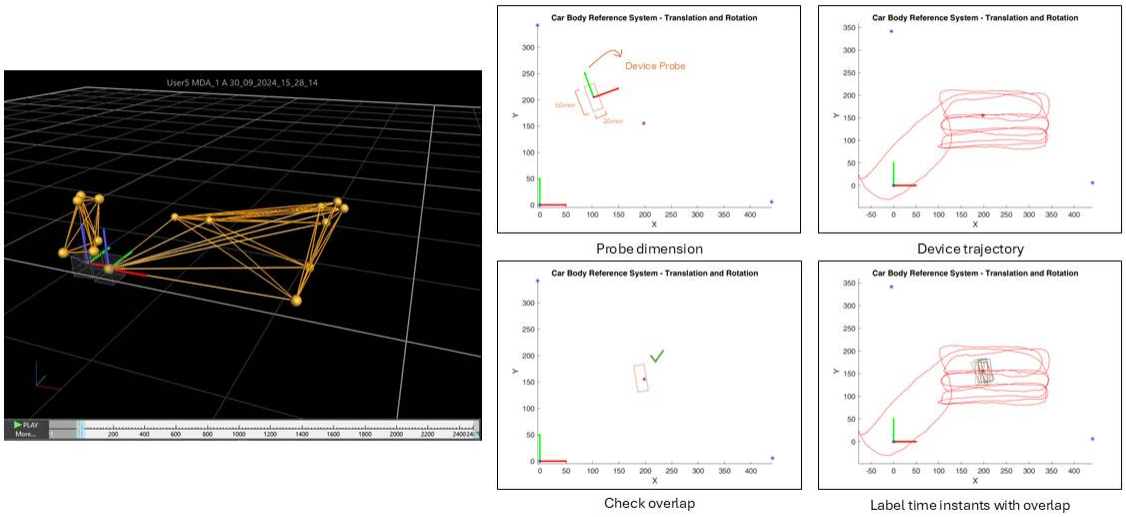


Figure 3.5. Device position tracking relative to the car body. The VICON system tracks the position of the handheld device as the user explores the surface, enabling reconstruction of the device's trajectory. When the device passes over a known defect location, the corresponding sensor signals are automatically labelled.

3.6 METHODOLOGIES EMPLOYED

Thanks to this new setup, a new data collection campaign was carried out, enabling the construction of an updated tactile dataset in which all tactile signals are automatically labelled whenever they correspond to the presence of a defect.

System Calibration

To ensure accurate spatial alignment between tactile measurements and known defect locations, a rigorous calibration procedure was developed and applied prior to data acquisition. This process establishes a common reference frame for each Car Body panel and allows all measurements, both positional and tactile, to be expressed within a consistent coordinate system. The result is a dataset in which each defect is not only labeled, but also precisely localized in 3D space relative to the inspected surface. The calibration begins with the physical preparation of each Car Body panel. All extraneous labels, dirt, and adhesive residues are removed from both the front and rear sides of the metal surface. The panel is then mounted onto a dedicated calibration rig using standardized alignment bars, ensuring that every panel is positioned in the same nominal pose across different acquisitions. This repeatable setup defines a baseline for establishing a local coordinate frame. To define this frame, three lightweight rods, each equipped with a Vicon marker cluster, are clamped to the **lower-left (L_1)**, **lower-right (L_2)**, and **upper-left (L_3)** corners of the panel. These three non-collinear points form a local reference system in accordance with the right-hand rule. The origin is set at L_1, the X-axis is defined by the vector from L_1 to L_2, the Y-axis by the projection from L_1 to L_3,

and the Z-axis is computed as their cross product, pointing outward from the panel surface. This results in an orthonormal coordinate frame {**X**, **Y**, **Z**} ⊂ $\mathbb{R}^3$ that captures the physical orientation of the panel in space. Next, the defect of interest is marked on the surface with a red circle and a dedicated Vicon marker is placed at its geometric center, hereafter denoted as $p_{defect}^{Vicon} \in \mathbb{R}^3$. The tactile inspection device is also equipped with its own Vicon marker cluster to enable tracking of its 6-DoF pose during exploration.

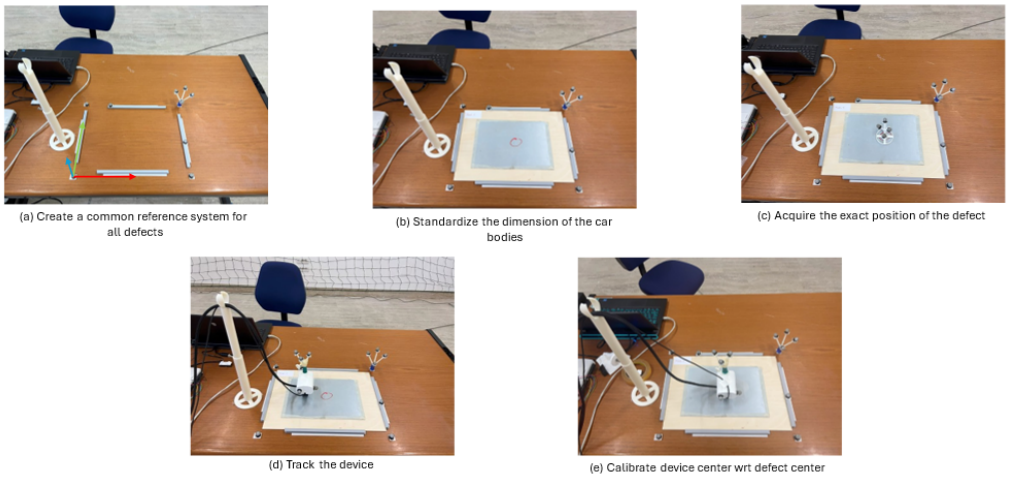


Figure 3.6. The process begins with the creation of a common reference system to ensure consistent localization across all defects (a). Next, the dimensions of the car body panels are standardized to provide a uniform working area (b). Using a calibration target, the exact position of each defect is acquired (c). The handheld inspection device is then tracked in real time to monitor its position during the task (d). Finally, the device is calibrated so that its center aligns precisely with the center of the detected defect (e).

The calibration measurements are carried out using a custom MATLAB script that connects to the **Vicon DataStream SDK** in client-pull mode. For each valid frame, the system records:

- The global position and orientation of the Car Body: $T_C^{Vicon} \in \mathbb{R}^3$, $Q_C^{Vicon} \in \mathbb{H}$
- The global position and orientation of the defect: $T_D^{Vicon} \in \mathbb{R}^3$, $Q_D^{Vicon} \in \mathbb{H}$
- The 3D coordinates of the three corner markers: $L^{1Vicon}, L^{2Vicon}, L^{3Vicon} \in \mathbb{R}^3$

Frames in which any marker is occluded are discarded, and a statistical stability check ensures that only those frames with standard deviation σ < 0.5 mm for all marker positions are retained. The final valid frame defines the nominal rigid body transforms.

To express any 3D point in the Car Body coordinate frame, we apply a translation and a rotation as follows:

$$p^{CB} = R_C^{-1} \cdot \left(p^{Vicon} - T_C^{Vicon}\right),$$

where:

- $p^{CB} \in \mathbb{R}^3$ is the point in the Car Body frame,
- $p^{Vicon} \in \mathbb{R}^3$ is the same point in the Vicon global frame,
- $T_C^{Vicon} \in \mathbb{R}^3$ is the Car Body's translation,
- R_C is the rotation matrix corresponding to $Q_C^{Vicon} \in \mathbb{H}$.

The orientation of the defect with respect to the Car Body is given by quaternion composition:

$$Q_D^{CB} = \left(Q_C^{Vicon}\right)^{-1} \otimes Q_D^{Vicon}$$

Where:

- $Q_D^{CB} \in \mathbb{H}$ is the orientation of the defect in the Car Body frame,
- $\otimes$ denotes the Hamilton product,
- All quaternions belong to the space of unit quaternions $\mathbb{H}$.

All computed data, including the relative positions of the corner markers and defect center in the Car Body frame, as well as the composite quaternions, are stored in structured .mat files for use in downstream processing. This enables automatic defect labeling of tactile signals by spatial alignment of the tracked sensor probe with the known defect locations.

Repeating this procedure for all panels and defect types yields a robust and consistent calibration across the dataset. The observed localization errors remain within ±2 mm along all axes.

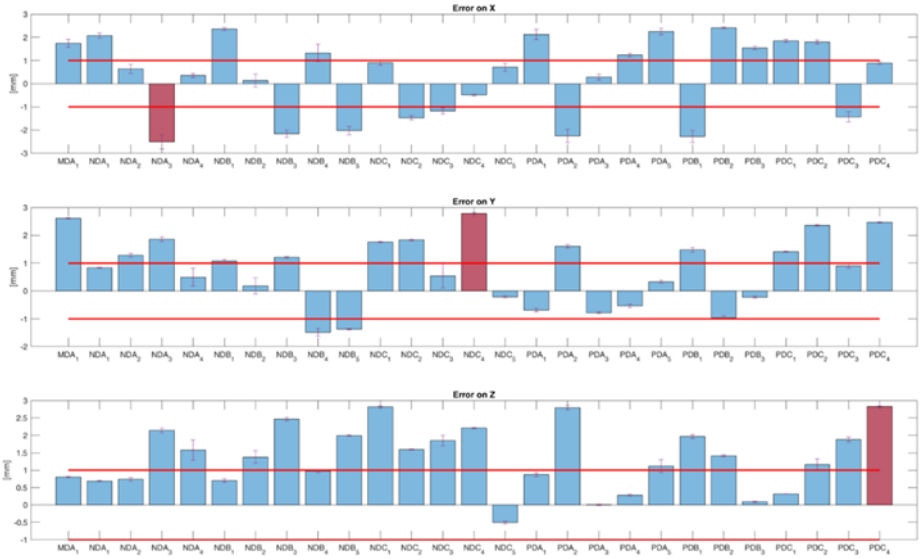


Figure 3.7. These plots show the calibration error obtained by positioning the device exactly above each defect and comparing the expected and actual positions. The errors are reported separately for the X (top), Y (middle), and Z (bottom) directions. Each bar corresponds to a specific defect, identified by its label (e.g., MDA_1, NDA_2, etc.).

This calibration pipeline forms the backbone of the dataset construction, ensuring synchronization between tactile and positional signals and enabling fully automated and spatially accurate defect labeling.

Data Labelling

Following the calibration and data acquisition stages, an automatic labelling procedure was implemented to determine whether the tactile probe was in contact with a known surface defect. This pipeline enables the precise annotation of tactile data by combining spatial calibration, positional tracking, and temporal synchronization between the tactile sensors and the Vicon motion capture system.

Each user trial is recorded within a dedicated folder that includes:

- A *.system* metadata file, from which we extract the total number of frames (denoted as *FramesCaptured*);
- An *.xcp* XML file, from which we read the acquisition start timestamp.

At runtime, a MATLAB script parses these files to retrieve the expected frame count and acquisition parameters. The script then opens a Vicon DataStream client in *ClientPull* mode and initiates a live streaming session. For every acquired frame, the following data are recorded:

- The Car Body's global position $T_C^{Vicon} \in \mathbb{R}^3$ and orientation $Q_C^{Vicon} \in \mathbb{H}$;
- The positions of the three Car Body corner markers $L^{1Vicon}, L^{2Vicon}, L^{3Vicon} \in \mathbb{R}^3$;
- The global coordinates of the defect marker $T_D^{Vicon} \in \mathbb{R}^3$;
- The global pose of the tracked tactile probe $T_P^{Vicon} \in \mathbb{R}^3, Q_P^{Vicon} \in \mathbb{H}$.

To ensure that the acquisition is complete and temporally consistent, the Vicon frame counter is monitored to confirm that the number of frames matches *FramesCaptured*. Streaming ends only after two full passes through the first frame, ensuring that the recorded sequence fully aligns with the original acquisition window. A *matchFrameCounter* array is built to verify a one-to-one mapping between internal and Vicon frames; if any frame is dropped or mismatched, the script safely aborts.

Once acquisition is complete, the previously computed calibration data, specifically, the corner positions and the defect center location in the Car Body frame, are loaded. The transformation described in the **System Calibration** section is used to express all recorded 3D points in the Car Body coordinate system.

The defect center is denoted as $(x^0, y^0) \in \mathbb{R}^2$, and is used to define an elliptical defect region with horizontal and vertical semi-axes a and b. This representation is motivated by the specific design of the tactile device, which uses a comb-like end-effector consisting of multiple thin prongs. Due to the spacing between the prongs, it is possible for the defect to be geometrically beneath the probe without being physically contacted

by any prong. For this reason, a soft-contact region is defined using an ellipse centered at the defect, capturing the area within which contact is likely, even if not guaranteed by the rigid geometry. The values used for the ellipse are based on the nominal dimensions of the probe: width $w = 5\,mm$ and height $h = 3.5\,mm$, used to define the semi-axes of the ellipse $a = \frac{w}{2} = 2.5\,mm$ and $b = \frac{h}{2} = 1.75\,mm$. For each frame, the pose of the probe in the Car Body reference frame is used to compute the 2D projection of its four corner vertices onto the surface plane. These vertices are then tested against the standard ellipse equation:

$$\frac{((x_i - x^0)^2)}{a^2} + \frac{((y_i - y^0)^2)}{b^2} \leq 1$$

If at least one vertex (x_i, y_i) of the probe lies inside the ellipse, the frame is labeled as overlapping, that is the probe is over the defect.

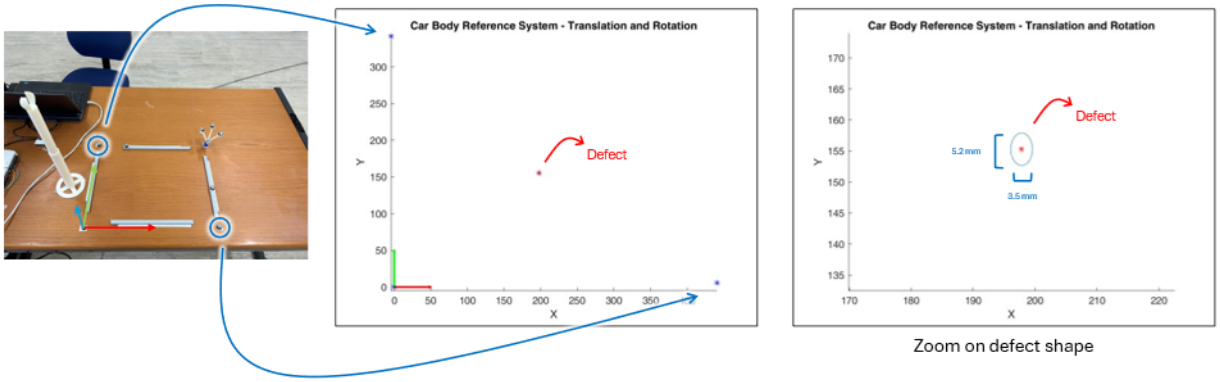


Figure 3.8. Representation of the defect in the coordinate system. Each defect is expressed in the car body reference frame and modeled using an uncertainty ellipse, which accounts for the non-ideal contact behavior of the scallop-shaped probe.

As a complementary verification step, a set of sample points is drawn along the ellipse perimeter, and MATLAB's *inpolygon* function is used to determine whether any of those points fall within the polygon defined by the projected probe. This dual-check strategy provides robustness against edge cases introduced by the comb geometry and possible minor tracking jitter.

The result of the automatic labeling process is a binary vector *isOverlapping*, which flags each frame as either overlapping or not. At the end of each trial, the script exports:

- A CSV file containing one row per frame with the structure $[x, y, z, q_w, q_x, q_y, q_z, overlapFlag]$
- A metadata file reporting the trial's start time and average frame rate;
- A console output summarizing the total number of frames where contact was detected and the overall duration of the trial.

This automated pipeline ensures that all tactile data is labeled with high spatial and temporal precision, enabling the development and evaluation of machine learning models for defect detection. The use of an elliptical contact region based on the probe's real geometry and functional behavior ensures that the labeling remains faithful to physical interaction constraints while maintaining algorithmic robustness.

Classification

For the tactile detection and classification, several models were implemented and investigated as described previously in D3.1. The same models are used in the classification at this moment, however, we made some changes and improvements based on new data including the preprocessing of the data.

Compared to the data just in the previous deliverable, there are some important changes. First of all, the previous data was fake generated data and now we have obtained real data. In contrary to the previous version, we have not only considered the feature-engineered data but also the raw data files of acceleration and force. With the focus on the implementation for the first integration, we also considered the raw data files since if those give similar (or better) results as the feature-engineered data, the processing of the raw data is not necessary, which will save us time in the real setting of classifying the defects. For the current data, the results between raw and feature-engineered data are negligible small, so for now we advice to use the raw data as much as possible. The training time is longer for this data, but since the training is not something we execute very often, that is no issue.

Furthermore, with this data, we split the data into time windows. Two approaches are implemented, fixed and rolling time windows. Fixed windows divide the data into non-overlapping segments, which is computationally efficient and used by default in the current implementation. Rolling windows create overlapping segments and can capture more detailed temporal transitions, but increase complexity and risk overfitting. For the current phase, fixed time windows are used due to their simplicity and suitability for the available data. For the time windows, we also set a percentage of the time measured as defect, to state if the time window is a defect or not.

Also two approaches were explored for handling different types of defects within the data set. The first approach treated the problem as a multi-class classification task, where "no defect" was included as one of the classes. The second approach involved first classifying the data as binary (defect/no defect) and then categorizing the different types of defects separately. Our results showed that the multi-class classification approach yielded better performance, with higher accuracy. The multi-class approach showed better performance and is used as the default strategy moving forward.

We noticed the models sometimes predicted too often no defect in cases where there was a defect. For this we developed a method to set a threshold, that if there is a chance of say 30% predicted that it is a defect, we already set it as a defect. This percentage can be adjusted as preferred.

Also the possiblity to set the frequency of the data as desired with some resampling, or drop a part of the no defect lines to make the data more balanced, or reduce noice bij smoothing the top and bottom percentiles of the data, are implemented.

3.7 RESULTS AND FINDINGS

In the previous deliverable, we have described the different models that were investigated and implemented. There were two directions: combination LSTM & CNN, and ensemble methods as Random Forest, Gradient Boosting and Bagging. These models are still used and if you want to read more about these, we recommend to read the previous deliverable.

Tactile dataset

Upon completion of the data labelling procedure, all tactile sensor data acquired during the user trials are processed and organized to form the final dataset. The objective of this phase is to synchronize the tactile signals (force and acceleration) with the positional labels generated by the Vicon-based labelling pipeline, ensuring that each tactile measurement is associated with a precise spatial interpretation.

The tactile data includes:

- Triaxial force signals $F(t) = [F_x(t), F_y(t), Fz(t)]$, sampled at $7\ kHz$
- Triaxial acceleration signals $A(t) = [A_x(t), A_y(t), Az(t)]$, sampled at $4\ kHz$

These time-series are aligned with the Vicon positional data based on shared timestamps and frame indices. For each instant t, if the position of the tactile probe is labeled as $isOverlapping = 1$, the corresponding force and acceleration samples $F(t)$ and $A(t)$ are also labeled as defect contact. This results in a binary annotation appended to each sample of the tactile streams.

In addition to raw signals, the dataset includes a set of **tactile features** derived from the time-series, already defined in Deliverable D3.1 and detailed in Section 3.6. These features include:

- **Power Spectral Density (PSD)** of both force and acceleration, computed in short windows to analyze frequency content associated with surface textures and discontinuities. The PSD is defined as $PSD(f) = |\mathcal{F}\{ s(t) \}|^2$ where $\mathcal{F}\{\}$ is the Fourier Transform, and $s(t)$ is either the force or acceleration signal.
- **Acceleration Spikeness**, measuring the impulsiveness of the signal, estimated using statistical descriptors such as kurtosis or peak density over a window, highlighting sudden changes indicative of defects.
- **Friction Index**, computed from the force signal as a function of tangential force

components.

All features are computed over temporal windows, and then assigned a binary label based on whether their window overlaps with a defect-labelled region from the positional data. Formally, for a window w_k centered at time t_k, the label is:

$$label(w_k) = \begin{cases} 1 \ \textbf{\textit{if}} \ \exists\, t \in w_k \ such\ that\ isOverlapping(t) = 1 \\ 0 \ otherwise \end{cases}$$

Each trial is saved in structured and *.csv* files, organized by user ID, part ID/defect type, sensor configuration and probe type. Data includes both the raw time series (force, acceleration) with binary defect labels and the computed features with matching window-level labels.

This stable version of the dataset represents a key advancement over that described in Deliverable D3.1. It offers synchronized, spatially aware, and semantically labeled tactile data, well-suited for training and evaluating machine learning models for robotic defect detection and human-inspired exploration strategy learning.

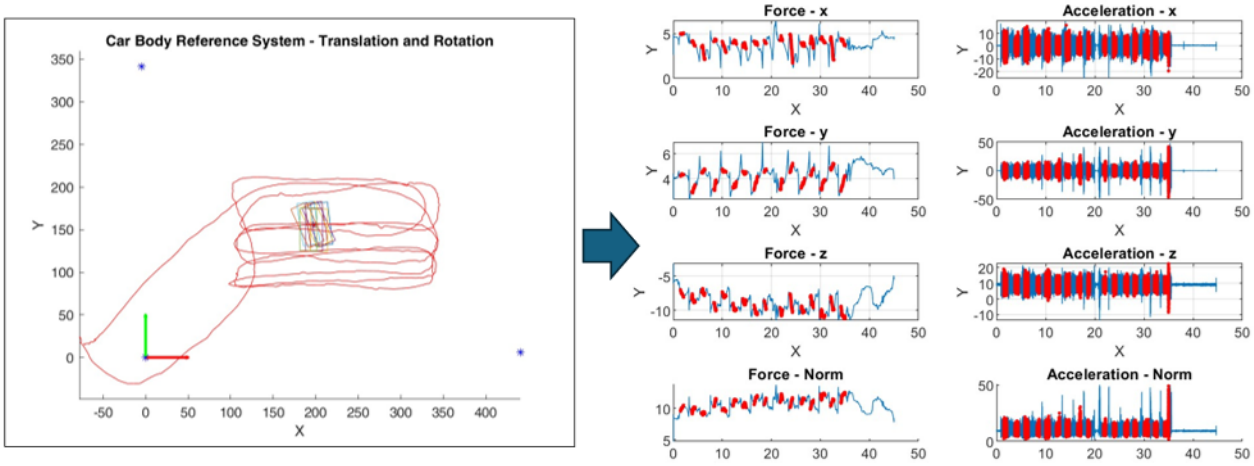


Figure 3.8. Representation of how positional data are used to label tactile data. When the device passes over a defect—illustrated as a rectangle in the figure—the corresponding time window in the force and acceleration signals is labeled as a defect. This association is visually represented by red highlights in the force and acceleration plots.

Classification

As described in the previous section, we have a lot of methods to preprocess the data, and a lot of parameters coming with this to adjust and tune. It is important to note that the current results are based on data collected using a handheld version of the tactile probe operated by a human. In the final setup, the data will be gathered by the robotic system, which may lead to differences in signal characteristics. As such, we refrain from drawing strong conclusions at this stage about the performing results. Once new data from the robotic setup becomes available, it will be essential to re-evaluate the model performance and update the approach accordingly.

In the preliminary results we now gathered, the Random Forest model mostly

performed the best. The accuracies are, depending on the parameter settings, motly achieved between 70-85%. In general, we can say that the model is good in detecting defects, but not always in the classification of positive or negative dent.

An example of the classification results are illustrated in Figure 3.9. In this case, we used raw data, random forest model with fixed time windows of 1 second.

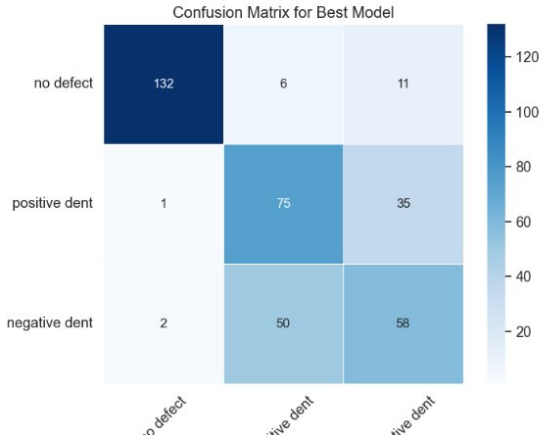


Figure 3.9. Confusion matrix of classification results.

3.8 CHALLENGES AND LIMITATIONS

The advancements presented in this phase of the project—particularly the development of an automated data acquisition and labelling pipeline—have addressed several limitations highlighted in D3.1, like the construction of a stable tactile dataset based on real car-body components. Nonetheless, new challenges have emerged due to the increased complexity of the experimental setup, the integration of multimodal sensors, and the requirement for high spatial and temporal precision.

Data Acquisition

One of the main technical challenges lies in the accurate synchronization between the Vicon motion capture system and the high-frequency tactile sensors. Although the acquisition software was extended to align all data streams using common timestamps, small temporal misalignments may still occur, particularly at the beginning or end of acquisitions. These require careful post-processing to avoid labelling errors in the tactile signal. A second limitation relates to the geometry of the tactile probe, which employs a comb-like end-effector. Due to the spacing between the prongs, it is possible for the probe to pass over a defect without establishing physical contact. To mitigate this, a soft-contact model was introduced using an elliptical approximation of the probe's contact area. While this solution increases labelling robustness, it may introduce false positives in cases of marginal overlap or in the presence of nearby defects.

Classification

In the previous deliverable we stated that the training time would be a challenge, however we think that is no issue anymore. The training will happen in the beginning a few times, but for the real time detection there is then already a model available. The real time detection is very quick.

With the available data, we only classify on positive/negative dent. Since the model is not always correctly predicting the type of defect, we are not yet classifying on the severity A/B/C. This can become a challenge later in the proces, since the desired outcome will be to also predict the severity. However, with new data becoming available, we hope to improve our model and if that is the case, we can try to also classify of the severity.

4 HUMAN MOTION PERCEPTION

4.1 INTRODUCTION

Human Motion Perception describes the ability of computer systems to sense human presence and motions using electronic sensors. Perceiving humans can be tackled with various technological solutions ranging from inexpensive (~1€) Passive Infrared (PIR) human motion detectors that provide 1 bit of data (Motion/No motion), to commercial MOCAP systems that quickly exceed in cost the hundreds of thousands of Euros, require specialized suites with reflective markers and have millimetre precision for all human joints along with an inverse kinematics solution for the human skeleton. When framed within the context of computer vision, a commonly posed problem that can aid in detecting human motion is that of Human Pose Estimation. Human Pose Estimation refers to the task of estimating the pose, in an appropriate representation, of the observed human(s) in the scene using visual input. Since humans prefer not to wear specialized clothes with sensors on them, Human Pose Estimation is commonly tackled using cameras that can observe users in a non-intrusive way. The representation of the estimated pose can be in 2D, by localizing bounding boxes, segmentation masks, or key points on the input image. For human presence and/or pure motion detection, 2D landmarks usually suffice. Most methods, however, focus on 3D human pose estimation that also recovers the 3D depth of each of the landmarks. This is especially useful when the 3D position of the human plays a role in occupational safety like the scenarios we are tackling in MAGICIAN. 2D and 3D human pose estimation can include the body, hands, face and gaze, all of which are subsets of the problem, with methods that try to tackle all of them being referred to as Holistic or Total Capture methods. Finally, as made evident in the following state of the art section, there are methods that not only recover key point positions but also estimate the human shape, including biometric parameters such as height, BMI, among others, thus also providing a comprehensive 3D mesh model of the tracked humans. These are “Human Mesh Recovery” (HMR) class methods and can provide pinpoint precision for the whole human body surface.

4.2 OBJECTIVES AND REQUIREMENTS

Collaborative robotics is often seen as the cornerstone of next-generation manufacturing solutions, commonly referred to as "Industry 5.0." However, distinguishing between robotic applications that qualify as "collaborative" and those that do not, is not always straightforward. Rather, it is easier and potentially more informative to identify a spectrum of possible scenarios. At one end of the spectrum, we find stand-alone, classical robotic stations, which are classified as “collaborative” to simplify the physical layout of the production line (collaborative robots do not need to be segregated from humans). At the opposite end, we find truly collaborative

applications in which humans and robots engage in a direct and physical collaboration (e.g., a robot can hand over tools to humans or help them move heavy loads). In the MAGICIAN project, we find ourselves in an intermediate scenario: robots and humans share the same workspace, but they do not directly collaborate. The stations where the sensing robots identify defects and the cleaning robots remove them are shared areas, where humans work alongside the robots to supervise their operation or handle particularly complex tasks that exceed the robots' capabilities. In this setting, our problem is to ensure a safe coexistence without sacrificing productivity. We can identify two situations which can lead to potential problems:

1. A mobile robot is moving along a possible collision course with a human,
2. A robotic manipulator is executing an activity following a path that can collide with some part of the body of the human operator and/or trap them.

Our approach to deal with these potential problems hinges on human-aware motion planning. A general overview of the approach is illustrated in Figure 4.1. In this figure we assume the presence of a robot, which must execute a given set of tasks (such as scanning the surface of a car body in search of defects). Appropriate environment sensors detect the presence of humans in the scene and observe their motion. Based on this observation, a system produces a prediction of the human motion for a time horizon of 1.5 to 2 seconds. Based on this prediction and the knowledge of the task that the robot must execute, the human-aware motion planner decides a trajectory that minimises the risk of accidents while guaranteeing a satisfactory level of performance.

In this section we are focusing on ways to produce an acceptable prediction for the human motion. The problem takes on a different form depending on whether we are dealing with a mobile robot or a manipulator. In the first case, we need a prediction on the position of the human body as a whole (for instance the centroid of the point cloud associated with the human). This topic was explored in previous projects, and we will build upon the results obtained at that time [Ant23]. For the second case, the robot and the human need to work at a close distance. Therefore, we need to consider the position of the different parts of the body since these fine details can prove useful. For instance, for a human with open and stretched arms, it is possible for the robot to use the space between the two arms. This motion prediction will be one of the outcomes of the project and it must meet the following requirements:

1. Accuracy: the acceptable margin of error is in the order of a few centimetres.
2. Time horizon: to be useful for motion planning, the prediction must be reliable for a time horizon of approximately 2 seconds.
3. Efficiency: the system must demonstrate sufficient reactivity, meaning the prediction should be delivered within a few tenths of a second after new data is collected.
4. Multi-scenario: in cases where uncertainty remains about the person's possible movements, the system can generate multiple scenarios, each associated with a probability level.

We can think of human motion perception as the interplay between two different conceptual modules. The first (elaborated in Section 4.3.1) deals with pattern recognition in the RGB level, with the task of correctly extracting the pose of observed humans regardless of their various optical appearance differences. The second module (Section 4.3.2) deals with pattern recognition in the pose coordinate level (regardless of their RGB appearance), trying to calculate, observe and predict patterns of motion in human joint coordinate trajectories. Both modules form a common mechanism and play an important role for a successful motion prediction framework. A pose estimation module without good accuracy cannot provide accurate data for reliable motion predictions. Similarly, an accurate but slow rate of pose estimation predictions will not provide enough time resolution for detailed understanding of motion, thus resulting in skewed and unrealistic, erratic motion predictions. Furthermore, even with very good pose estimation having a powerful motion prediction technique is essential for a system that can properly model and anticipate the complex and intricate human motions that can be encountered in an industrial environment like the one we target.

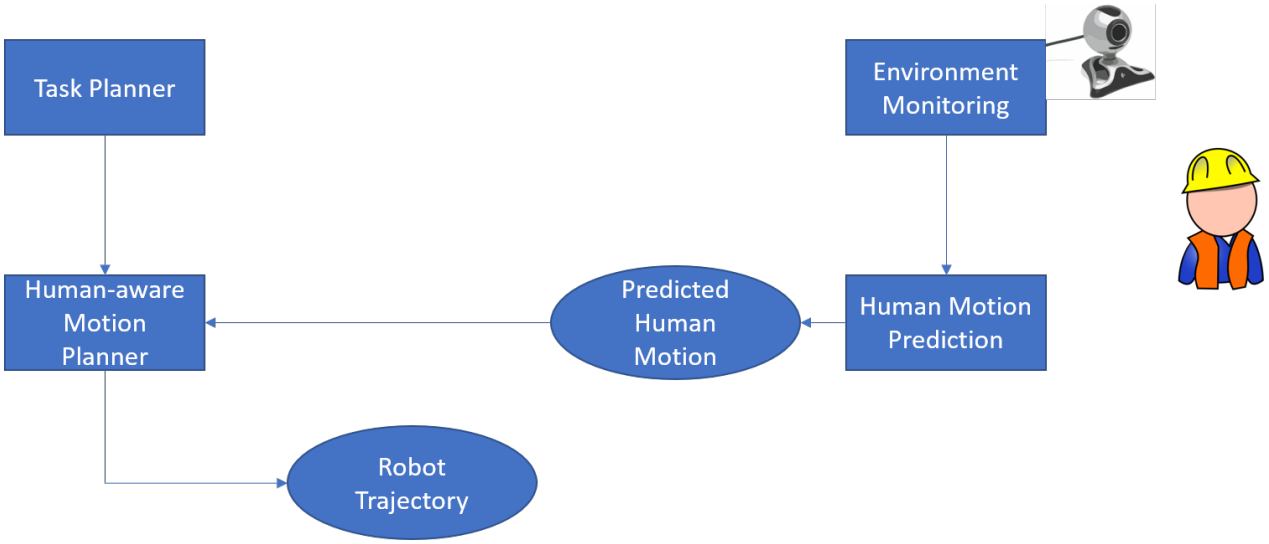


Figure 4.1. The framework of Human-Aware Motion Planning.

4.3 METHODOLOGIES EMPLOYED

To meet the demanding requirements of the human motion predictor outlined above, the MAGICIAN team has thoroughly evaluated the best state-of-the-art solutions that are assessed in Sections 4.3.1 and 4.3.2. For pose detection, we are developing a novel U-NET based architecture that regressed 2D pose, depth and normal vectors in real-time to facilitate the motion prediction task while also providing depth perception to the MAGICIAN Cobot. The 2D pose can also be augmented using a MocapNET that performs inverse kinematics regression to provide higher level data. For motion prediction, after careful assessment, we narrowed our options to two alternatives: applying one of the latest deep learning approaches [Yan2024, Tian2024] or using classic clustering

techniques informed by our understanding of the specific process.

4.3.1 TECHNIQUES FOR POSE DETECTION

We will begin by describing the pose detection sub-problem, which in general consists of the task of receiving an RGB image featuring persons, identifying them and the joints of their skeleton and providing this data as high-level output for use by other modules. Tackling the task is very challenging since the appearance of humans in an image widely ranges when they are recorded as 2D projections of red, green and blue light intensities. The human body is very flexible with many configurations, human appearance is very varied, parts of the scene may be occluded by obstacles, cameras suffer from lens deformations, thermodynamic noise, motion blur, vibrations and other potential artifacts, observations might have multiple explanations and differences in lighting given the low dynamic range of typical camera CCDs pose significant challenges that need to be systematically overcome. Deep learning approaches have recently managed to tackle the very high dimensional space of RGB images successfully performing pose estimation and the next sections will briefly describe the various involved techniques adopted for use in the context of the MAGICIAN project.

4.3.1.1 DATA ACQUISITION AND ANNOTATION

Data acquisition for pose estimation methods presents significant challenges in the European Union. Collecting images that contain individuals with identifiable and potentially privacy-sensitive information is both costly and difficult under European law and GDPR provisions. Additionally, the consequences of a person withdrawing consent for inclusion in a training set are unclear. Removing such data from an already trained model without retraining the model from scratch remains an unresolved research problem. Moreover, training a method unbiased in gender, race, and appearance is highly challenging. Naively collecting data can result in a neural network that is significantly biased against certain demographic groups, even if it appears to perform well for individuals who are well-represented in the training set. To this end, we use well-established open datasets such as COCO17 [Lin14], MPII [Andriluka14], EXLPose [Lee23], AIChallenger [Wu17], and the AM-2K [Li22] and BG-20K [Li22] datasets as our primary sources. These provide a solid baseline onto which we can incorporate our own data. To effectively augment the openly available datasets, we use generative AI-generated data (Figure 4.2), which are programmatically created to better suit the intended industrial application while aiming to represent all worker categories in an unbiased manner. Additionally, data from other sources can be easily incorporated after being processed through a series of segmentation and ground truth extraction steps for training.

Figure 4.2. Using generative AI, namely score-based diffusion techniques, we can programmatically create synthetic scenes that loosely resemble our target application. This way we can provide a richer source of samples while bypassing the legal, ethical and practical complexities of collecting actual data from real workers.

Complementary to the defect annotation tool presented in Section 2.4.2 of this deliverable, we also developed a Human Pose Annotation counterpart. Using the same underlying GUI frameworks and a similar visual language, this tool (Figure 4.3) allows users to annotate captured human pose data and prepare it for training, as shown in Figure 2.10 and Figure 2.11. Depth and 3D normal vectors are automatically provided using Depth Anything 2 [Yang24], while segmentation masks are extracted using Detectron 2 [Yuxi19] and DPText [Ye23]. Finally, 2D pose estimation is initialized using the latest version of HR-Net to automatically annotate the 2D landmarks. After these procedures are completed, the user can review and correct human joint landmarks, which may be miscalculated in challenging images with many people and occlusions.

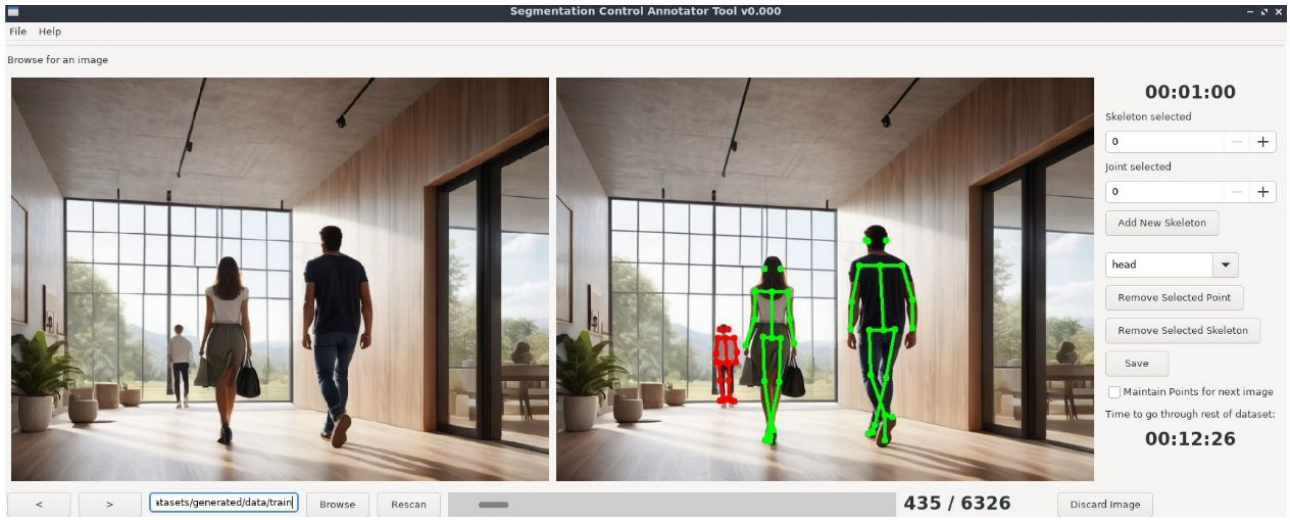


Figure 4.3. The Annotation tool developed while used to annotate synthetic data generated using generative AI to be included in our model's training.

4.3.1.2 HUMAN POSE DETECTION METHOD

Real time performance and high 3D accuracy are crucial in a production line. We developed D-PoSE (Depth as an Intermediate Representation for 3D Human Pose and Shape Estimation), Recent works use larger models with transformer backbones and decoders to improve the accuracy in human pose and shape (HPS) benchmarks. D-PoSE proposes a vision-based approach that uses the estimated human depth-maps as an intermediate representation for HPS and leverages training with synthetic data and the ground-truth depth-maps provided with them for depth supervision during training. Although trained on synthetic datasets, D-PoSE achieves state-of-the-art performance on the real-world benchmark datasets, EMDB and 3DPW. Despite its simple lightweight design and the CNN backbone, it outperforms ViT-based models that have several parameters that is larger by almost an order of magnitude. Therefore, D-PoSE runs real time 3D pose and shape estimation for multiple people with conventional GPUs.

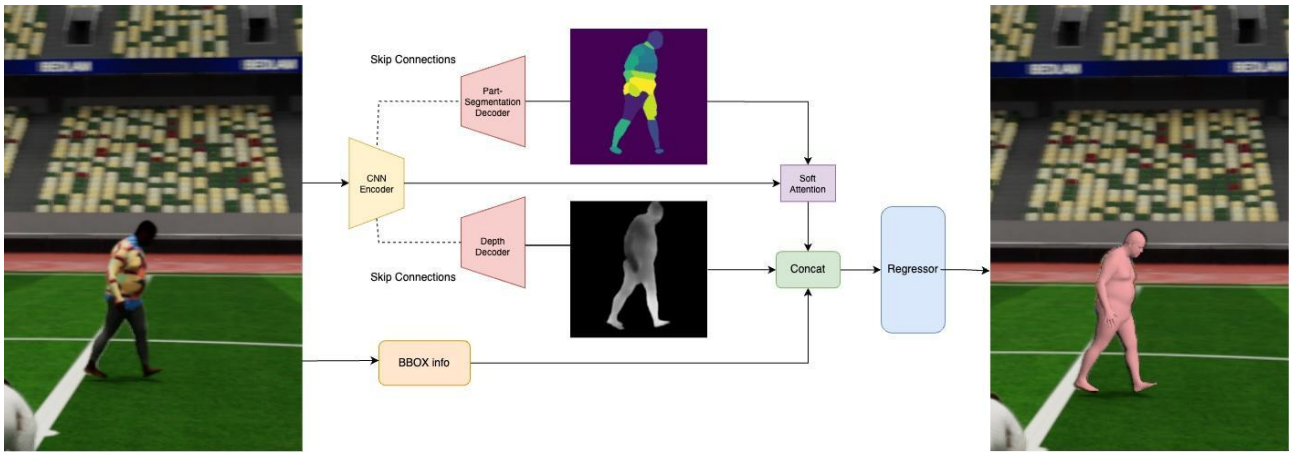


Figure 4.4. The architecture of D-PoSE. Given an input image, features are extracted using a CNN. With these feature maps, a human depth map and a part-segmentation map are estimated. The original features pass through a soft-attention mechanism which uses part-segmentation maps. The final features are concatenated with the bounding-box information and the depth features and are given as input to the regressor which estimates the 3D human pose and shape.

An overview of the architecture of D-PoSE is provided in Figure 4.4. Given an input image, features are extracted using a CNN. With these feature maps, a human depth map and a part-segmentation map are estimated. The CNN features pass through a soft-attention mechanism which uses the part-segmentation maps. The final features are concatenated with the bounding-box information and the estimated human depth map and are given as input to the regressor which estimates the 3D human pose and shape. Below, we provide further details on each and every of the aforementioned modules and representations.

To predict the 3D human mesh, we utilize the SMPL-X body model, which consists of N = 10,475 vertices and K = 54 joints, including those for the neck, jaw, eyeballs and fingers. The SMPL-X model is represented by the function $M(\theta, \beta, \psi)$, where θ denotes pose parameters, β captures shape parameters, and ψ represents facial expression parameters.

	Training Datasets	Method	EMDB [20]			3DPW [47]		
			MVE	MPJPE	PA-MPJPE	MVE	MPJPE	PA-MPJPE
HRNet	SD	PARE	-	-	-	97.9	82.0	50.9
	SD	CLIFF	-	-	-	87.6	73.9	46.4
	BL	BEDLAM-HMR	-	-	-	93.1	79.0	46.4
	BL	BEDLAM-CLIFF	113.2	97.1	61.3	85.0	72.0	46.6
	BL	**D-PoSE (Ours)**	**99.0**	**85.5**	**53.2**	**81.4**	**68.9**	**43.6**
ViT	BL	HMR2.0	106.6	90.7	51.3	88.4	72.2	45.1
	BL	TokenHMR	106.2	89.6	**49.8**	85.7	71.6	44.0
HR Net	BL	**D-PoSE (Ours)**	**99.0**	**85.5**	53.2	**81.4**	**68.9**	**43.6**

Figure 4.5. Comparison of HPS errors on the EMDB and 3DPW datasets. SD denotes standard realistic datasets, while BL denotes training exclusively with synthetic datasets (BEDLAM and AGORA).

In Figure 4.5 we compare our method to the current state of the art methods. In order to evaluate our method, we convert 3DPW and EMDB SMPL meshed to SMPL-X. In both 3DPW and EMDB we report Mean Vertex Error (MVE) using the vertices obtained from the SMPL mesh, Mean Per Joint Position Error (MPJPE) of the human 3D joints, and Procrustes-Aligned Mean Per Joint Position Error (PA-MPJPE) between the predictions and the ground-truth. All metrics are reported in mm. The results in Figure 4.5 show that our model in 3DPW reduces PA-MPJPE by 3.0 mm, MPJPE by 3.1 mm and MVE by 3.6 mm when compared with BEDLAM-CLIFF (HR-Net backbone). In EMDB reduces PA-MPJPE by 8.1 mm, MPJPE by 11.6 mm and MVE by 14.2 mm when compared with BEDLAM-CLIFF (HRNet backbone). When compared with TokenHMR (ViT backbone) in 3DPW reduces PA-MPJPE by 0.4 mm, MPJPE by 2.7 mm and MVE by 4.3 mm. In EMDB reduces MPJPE by 4.1 mm and MVE by 7.2 mm.

Method	GFLOPs	# Params	Infer. time
TokenHMR	254.31	681.0 M	441 ms
D-PoSE (ours)	**64.70**	**81.2 M**	**265 ms**

Figure 4.6. Comparing complexity of TokenHMR and D-PoSE with respect to GFLOPs, number of parameters and inference time (CPU).

In Figure 4.6 we present the floating-point operations, the number of model parameters and the inference time of D-PoSE. All reported figures suggest that compared to the state-of-the-art model TokenHMR, D-PoSE is lighter by a large margin, having 83.8% less parameters. The reason that our model is significantly smaller is that we use a CNN backbone instead of ViT while also using lightweight decoders and regressor.

Figure 4.7. Each image block represents: the input image (left); the part-segmentation estimation as an intermediate representation (middle-top); the human depth map as an intermediate representation (middle-bottom); the 3D HPS estimation of our method (right). The figure illustrates results from the 3DPW dataset (top left block) the EMDB test set (top right), synthetic image sampled from the BEDLAM validation set (bottom left) and from the RICH dataset (bottom right).

Our qualitative results provide evidence on the effectiveness of our method across a diverse set of challenging scenarios. Table consolidated results from four key datasets, illustrating the versatility and robustness of our approach across a variety of environments and challenges. The top-left section of Figure 4.7 showcases the 3D HPS estimation capabilities of D-PoSE on the 3DPW dataset, along with intermediate representations of depth and part segmentation. Despite the challenges posed by realistic outdoor scenes and occlusions, our method exhibits strong generalization, effectively transferring from synthetic training data to real-world environments. Its robustness is further evidenced by maintaining accuracy even in heavily occluded scenes, a common issue in real-world human pose estimation (HPS) applications. Figure 4.7 top-right presents the results from the EMDB dataset, highlighting our method's performance in a scene with a challenging pose. In the bottom-left of Figure 4.7, results from the synthetic BEDLAM dataset illustrate our method's ability to maintain high accuracy, validating its efficacy across both real and synthetic environments. Finally, the bottom-right of Figure 4.7 presents the results from the RICH dataset, which features complex human poses.

Figure 4.8. Successfully tracking workers performing defect repairs on TOFAS premises.

We plan to experiment with different backbones such as ViT to provide a larger (in terms of parameters) version of D-PoSE.

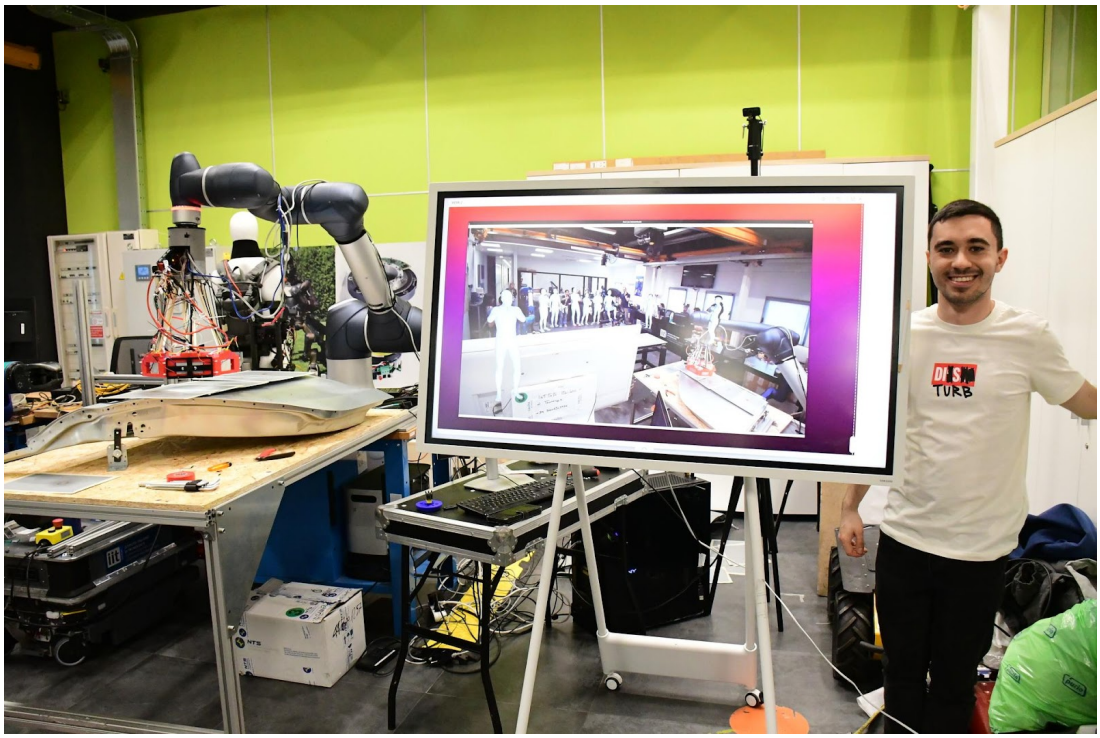


Figure 4.9. Successfully tracking 8 people in real-time using D-Pose during the 1st integration meeting in IIT.

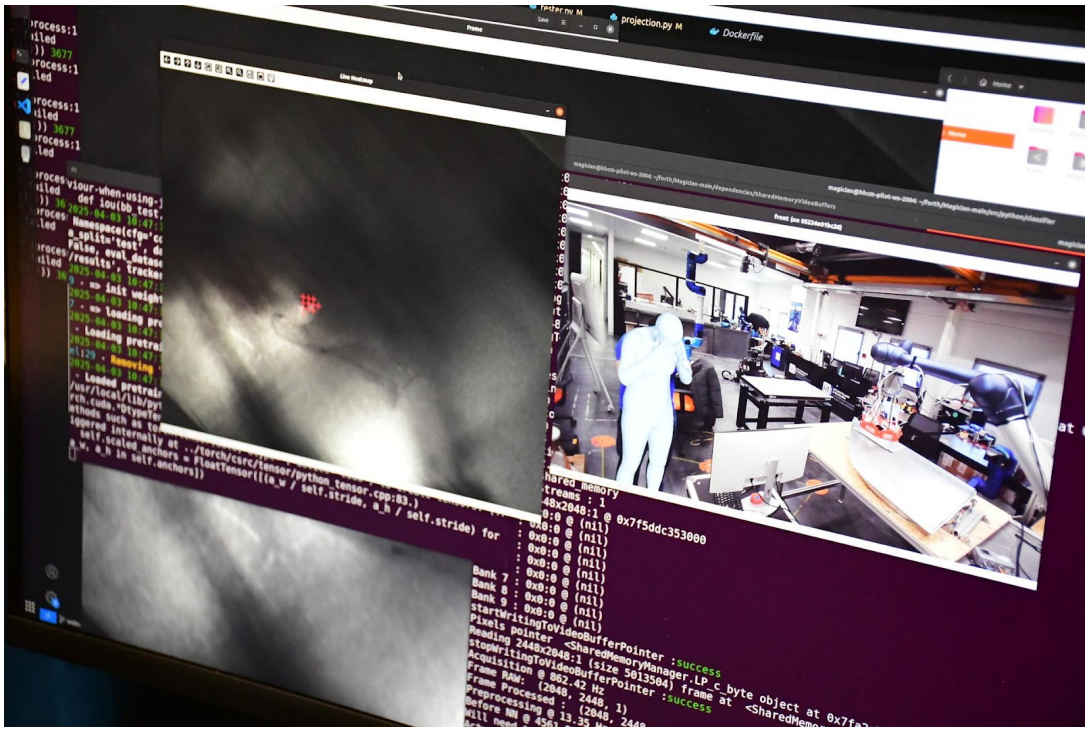


Figure 4.10. Successfully running D-Pose while also performing defect detection from the same computer system host during the 1st integration meeting in IIT.

4.3.1.3 SENSING HUMANS AND THEIR CONTEXT

D-Pose provides state of the art accuracy and understanding of the 3D human form and pose using a low cost RGB camera even in very challenging scenes with multiple visible humans (Figure 4.8 and Figure 4.9). It is also excellent with respect to its computational efficiency allowing it to be executed in parallel with the defect detection node (Figure 4.10) on a single GPU. By its design, however, it is not built to provide any information on the spatial context of humans. To address this aspect of the observed scene we have developed a Y shaped Multi Attribute Prediction neural network that in short is called Y-MAP Net (https://arxiv.org/abs/2411.10334). It offers a combination of outputs that address all visible parts of the scene (including objects).

Y-MAP Net main train corpus is the COCO 15 and COCO 17 datasets. As a result, the network learns internal representations for a variety of categories including Persons, Vehicles, Animals, Common Objects, Furniture, Appliances, Materials, Obstacles and Buildings. Using generative AI, we also supplement the training data with industrial scenes in an attempt to make the network behave better in settings like the ones targeted by MAGICIAN. The network can provide short keywords describing the camera observations that could be used by the robot to easily program behaviour switching based on string matching. The network produces segmentation maps for the people present in the scene which could be used to provide bounding boxes for the execution

of D-Pose instead of the YOLO or MaskRCNN networks. Finally, the vehicle and person segmentation maps along with the depth and normal maps can be beneficial as a general collision avoidance technique.

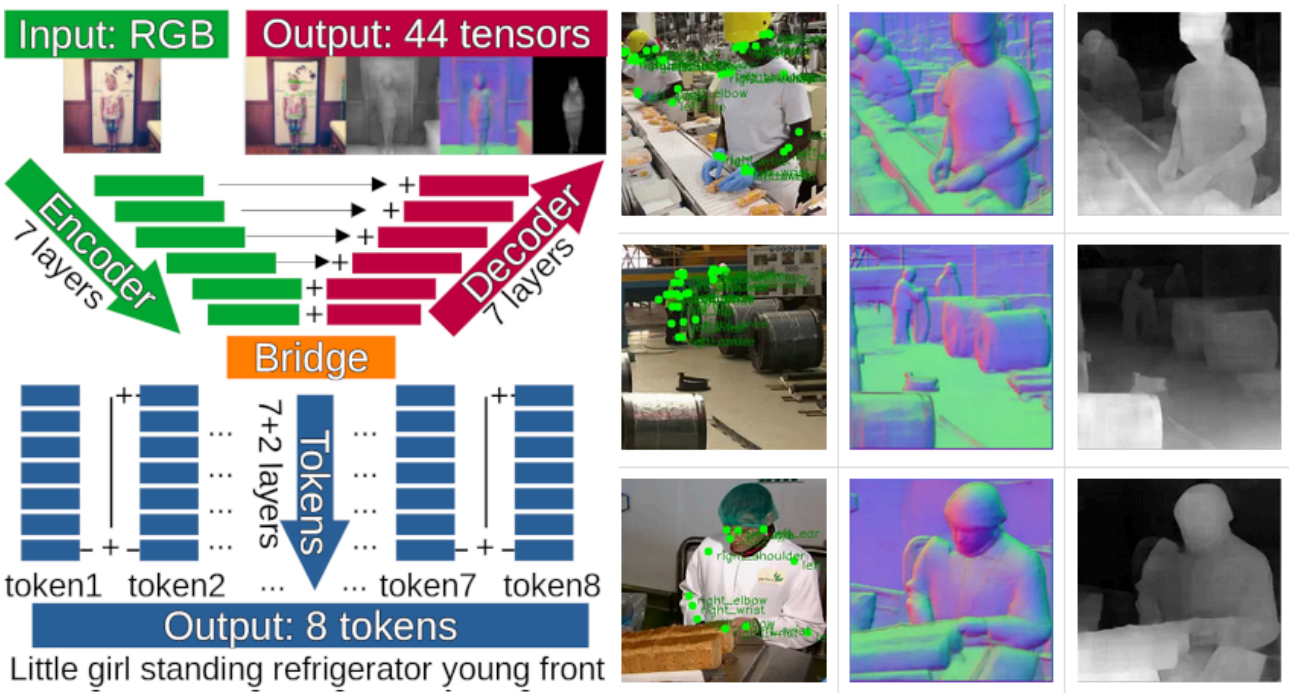


Figure 4.11. Left: The Y-MAPNet (https://arxiv.org/abs/2411.10334) network architecture. Right: The network extracts 2D pose, normal vectors and depth maps for the whole scene from monocular RGB something useful in the context of industrial applications.

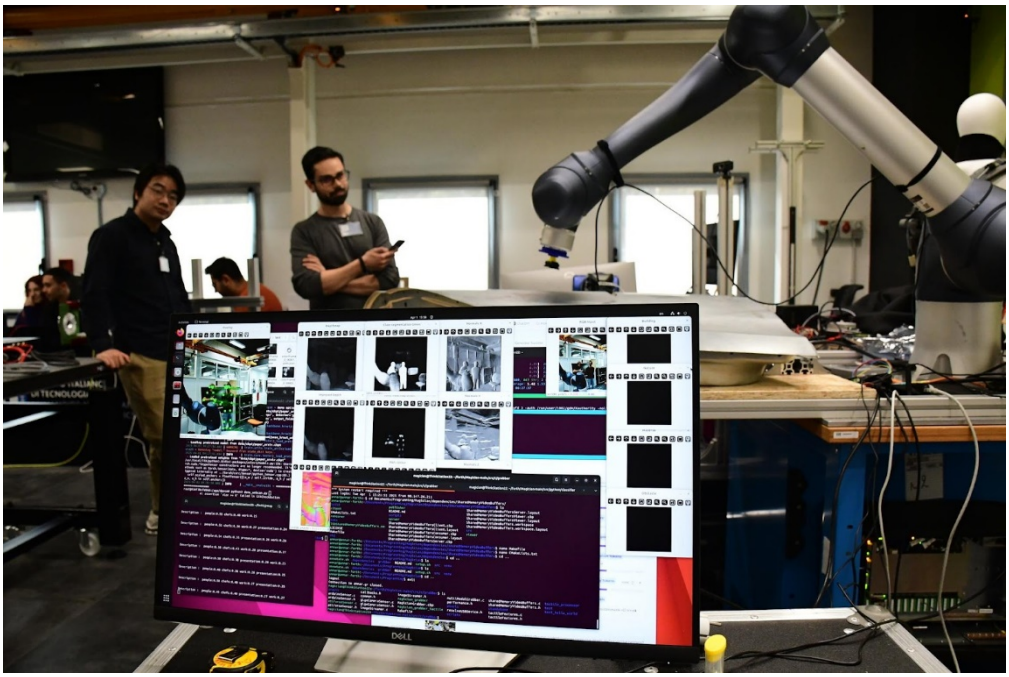


Figure 4.12. Successfully using Y-MAP Net (https://arxiv.org/abs/2411.10334) to extract 2D pose, depth maps, normal vectors and segmentation maps in the observed scene.

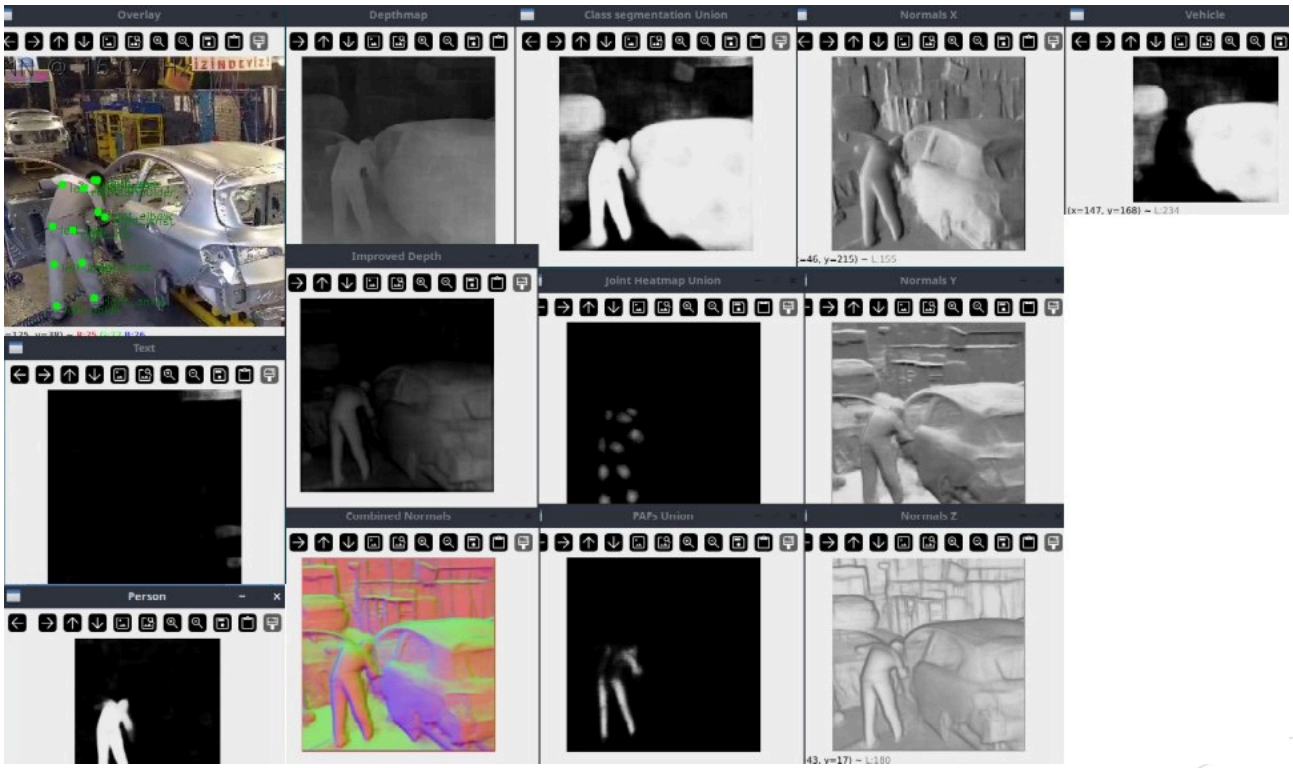


Figure 4.13. Employing Y MAP Net on monocular videos from a security camera located in the TOFAS factory and extracting 2D joints, person and vehicle segmentation masks, Depth map, Normal maps and Part Affinity Fields using one-pass evaluation in real-time (19Hz @ RTX 4070 GPU).

4.3.1.4 RESULTS AND FINDINGS

We can successfully track human-related datasets from the factory lines of the TOFAS plant. By using real-time segmented depth streams directly extracted from RGB, along with a minimum safety distance policy, we can establish a baseline human-aware motion planning module. Providing the system with additional annotated data will further enhance results. Additionally, depending on the available computational resources, by adding more layers and increasing the capacity of the neural networks they can yield even cleaner output, based on our needs following experimental evaluation of the solution.

4.3.2 TECHNIQUES FOR HUMAN MOTION PREDICTION

The estimation of human pose is a preliminary activity needed to predict human motion. As already mentioned, we adapted two different families of techniques: deep neural networks and classic clustering.

4.3.2.1 NEURAL BASED TECHNIQUES FOR HUMAN MOTION PREDICTION

We have tested two state-of-the-art algorithms for human motion prediction. As detailed next, we had to do some adaptation to meet the challenging real—time requirements of our application.

The first is *Adaptive Spatial-Temporal Graph-Mixer [Yang2024].* The input to the network is the 3D pose sequence. The first step of this solution is pose embedding through an Adaptive Spatial Graph Convolution. This step is needed to map the pose sequence to a higher dimensional space. Then the spatial and temporal dependencies are modelled separately with Spatial and Temporal Graph-Mixers using 3 different adjacency matrices each. The 3 matrices are: predefined, learnable, and adaptive. A prediction head then outputs the predicted future 3D pose sequence. We modified this method by using only one adjacency matrix (the predefined one with dependencies equal to the structure of the skeleton) to speed up the real—time performance. We tested both versions are tested with 12 poses as input and a prediction of 30 poses with a frame rate of 30 fps. The average error at the wrist with three adjacency matrices was 102 mm, while the single matrix delivered an average error of 103 mm with similar joint-level errors.

The second technique is the transformer-based diffusion model [Tiang2024]. The input 3D pose sequence is padded and transformed into the frequency domain using Discrete Cosine Transform (DCT). The denoiser network, composed of several Transformer layers, generates multiple predictions, which are mapped back to the temporal domain using Inverse DCT. We tested the method with input sequences of 6, 12, and 15 poses, predicting 30 or 60 future poses at 30 fps. The average prediction error was 110 mm, with

wrist-level errors of 135 mm and 147 mm. This method, however, cannot achieve real-time predictions due to a 0.1s processing time for each prediction.

4.3.2.2 CLUSTERING-BASED TECHNIQUES FOR HUMAN MOTION PREDICTION

This method is based on two phases: 1. we leverage a clustering technique to group similar human gestures, 2. during the prediction phase we retrieve the centroid of the closest cluster, and we use it for prediction (see Figure 4.14). Below we discuss the segmentation process and the specific clustering model used.

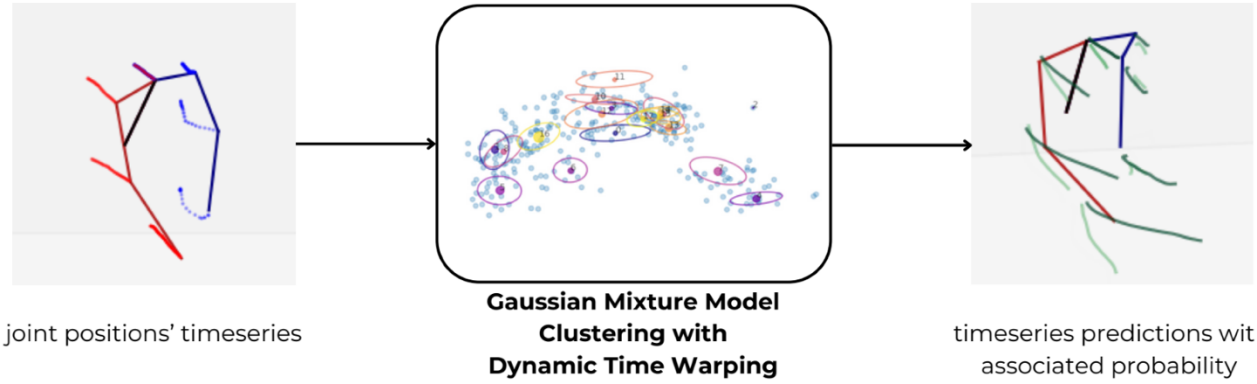


Figure 4.14. Segmented joint position time-series data clustered using GMM with DTW, resulting in time-series predictions with associated probabilities.

Segmentation. The first phase is to use segmentation to isolate the different gestures. The input are the skeleton data using a ZED camera and the official SDK. The skeleton data is collected over time, and the time-series of the skeleton is segmented based on changes in the direction of the terminal velocities of the hands. A change in direction typically indicates a change in gesture. For this initial implementation, we opted for a very simple segmentation logic based on these velocity changes. However, this approach is intended to be a starting point and can be replaced in the future with more sophisticated segmentation techniques as our methods and requirements evolve.

Clustering of Gestures. To cluster the human motion, we implemented a Gaussian Mixture Model (GMM) based on Dynamic Time Warping (DTW) to compare time-series samples. Each cluster is represented by a multivariate Gaussian and can approximate the cluster's members in the prediction phase. As new samples of current joint positions in 3D space are collected, we recompute the weight of each Gaussian in generating the time-series. This process provides a probability vector indicating cluster membership.

Performance. The performance of the clustering solution has been tested on a dataset [Cicirelli2022] produced on an assembly task, which is reminiscent of the type of tasks considered in MAGICIAN. We have considered a selection of gestures scoring similar result. A representative example is the error at the wrist, for which we represent the absolute mean error and the variance in the following example plot. On the *x* axis we

report the interval of time elapsed, which is proportional to the number of acquired samples.

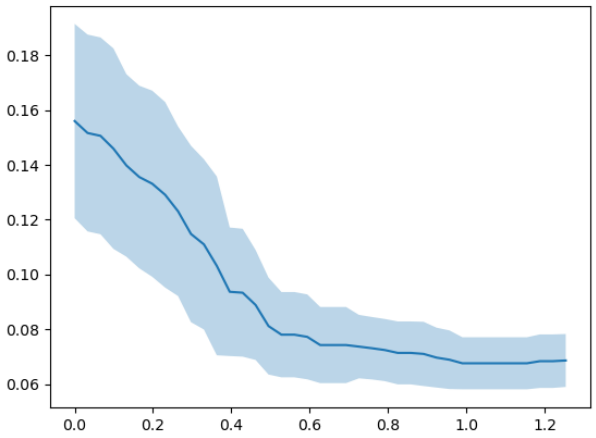


Figure 4.15. Reduction of wrist position error over time, showing the decrease in mean error and variance as the prediction horizon extends.

As we can see the error decrease both in terms of mean and variance as the prediction horizon progresses. This perfectly understandable since, with a small number of samples, there can be a few clusters with non-null probability. As time passes and more samples are acquired, we can remove the ambiguity and identify a single cluster with high probability. Still, even with a long-time horizon, the prediction accuracy is sufficient to be used by the motion planner.

4.4 CHALLENGES AND LIMITATIONS

The presented methods for addressing the problem of human motion detection are designed to tackle a highly complex challenge that, until recently, was considered infeasible to solve in real-time without specialized sensors due to its high dimensionality. By leveraging modern neural network techniques, we are developing a novel real-time module. Preliminary tests with input data that resemble the expected conditions on the deployed factory floor indicate that the module is performing adequately.

That said, for the system to achieve the best possible results, it is crucial to incorporate actual data from the specific use case, which we currently lack. To compensate, we are using a large collection of openly available generic training data, generative AI techniques designed to resemble industrial applications, and datasets from real factory floors like our target environment. As a result, we rely on the developed solution's generalization ability to address the specific use case. However, recording and using data from a robot equipped with a camera matching the specifications of the target system

and capturing images from the actual factory floor would have a very positive impact on the module's accuracy.

Another significant challenge in the development of these large neural network-based methods is the sheer computational intensity of performing back-propagation over hundreds of millions of weights using hundreds of thousands of training samples. Despite any optimizations applied, this process remains incredibly time-consuming. Given our current computational resources, we are limited to only four neural network training iterations per month. While rigorous validation techniques and a well-defined test set give us a good sense of overall accuracy, identifying and resolving issues is slow and difficult. This is because any update to the neural network architecture requires retraining. Additionally, the black-box nature of neural networks is a key limitation in systems like this, posing a challenge for this module.

Despite having candidate methods that can successfully facilitate human pose estimation, motion detection, and prediction, we must always bear in mind that these are fundamentally affected by practical issues such as occlusions, camera position, field of view, brightness, and contrast. For example, if a camera is mounted on a robot in a position where its view becomes occluded when the robot arm moves in certain ways, the system may be unable to correctly assess the situation, not due to a technical flaw in the neural networks, but because of the physical limitation of not being able to properly observe the scene. Similarly, if a camera sensor for human monitoring is positioned on the sensor head inspecting surfaces for defects, its rapid movements and proximity to the scanned object may result in improper exposure, blurred images, or dark observations, making it unsuitable for accurate human detection.

A final challenge for the developed system arises from the real-time requirements of the module. Most high-accuracy 3D human pose estimation methods use heavy transformer-based neural network architectures, often enhanced by stochastic optimization loops, which deliver highly detailed output. However, robot motion planning requires controllers to receive high-frequency input to operate effectively, creating increased demands for the computational performance of the human perception loop, which conflicts with the need for high accuracy. Both methods examined for this task have been designed with this trade-off in mind. In particular, MocapNET uses sparse 2D key points as input to enable real-time operation. However, this approach introduces certain limitations, such as multiple 3D solutions corresponding to the same 2D projections, geometric symmetries that may cause ambiguity, and the potential negative impact of occlusions or 2D noise on accuracy.

5 LEARNING DEFECT WORKING SKILLS FROM HUMANS

5.1 INTRODUCTION

The learning defect working skills from humans module represents a crucial component in the development of the MAGICIAN project, and it must be capable of performing high-precision manufacturing tasks. The primary challenge in this area is to replicate the intricate manual skills that human operators employ to detect, identify, and rectify defects on complex surfaces, such as those found on car bodies. This requires the robotic system to possess a high degree of flexibility and precision to mimic human-like adaptability. Additionally, the system must be equipped with algorithms that can accurately capture the nuances of human actions with the surface, at speeds comparable to human performance, with high accuracy, and real-time responsiveness. Furthermore, the system needs to be applicable to different parts of the car body, making necessary the capability of generalization.

For these reasons, we chose to employ Dynamic Motion Primitives (DMPs), which are a cornerstone in robotic learning and control, particularly for applications requiring the replication of complex, human-like movements. The core idea behind DMPs is to encapsulate motion patterns in a mathematically tractable form that can be modulated in real time to adapt to new situations. Originally proposed for simple reaching and grasping tasks, DMPs have been expanded to address more sophisticated scenarios, such as defect detection and correction in manufacturing environments, where robots work alongside humans. These tasks demand high levels of dexterity, adaptability, and safety, qualities that DMPs are inherently capable of providing.

However, traditional DMPs operate in Euclidean space, which may not always be the most natural or efficient representation for the robot's task or configuration space. Riemannian manifolds, which generalize the concept of curved surfaces to higher-dimensional spaces, offer a more suitable mathematical framework for capturing the complex, nonlinear structures inherent in robotic motion. For instance, representing a robot's joint configurations or sensor data as points on a Riemannian manifold allows for more natural interpolation, smoothing, and learning of motion trajectories. This approach enables the robot to better understand and replicate human-like motions that are critical for tasks such as defect detection and correction in manufacturing, where the robot needs to handle irregularities and uncertainties.

In this context, we aim to integrate DMPs with Riemannian manifolds to learn and execute defect working skills from humans in a collaborative setting. This integration allows for the natural representation of complex motion trajectories and enhances the robot's ability to generalize from a limited set of human demonstrations. The goal is to create a system that can efficiently learn from human operators, adapt to new and

unforeseen defects (Figure 5.1), and work safely alongside humans on factory floors.

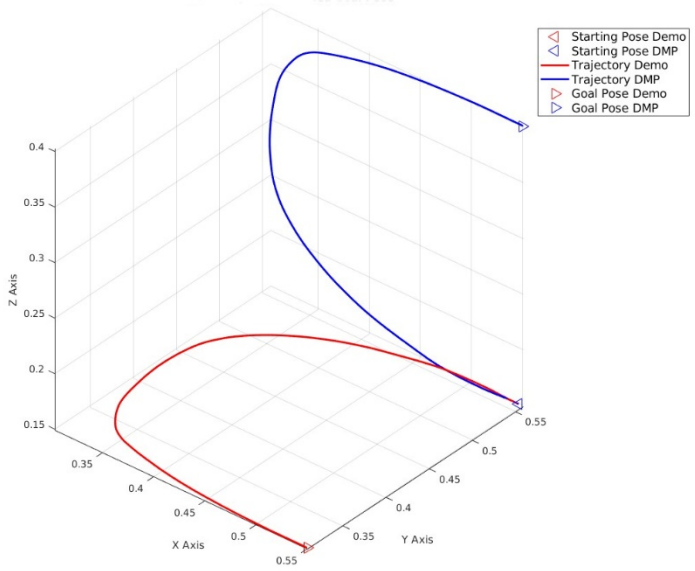


Figure 5.1. Example of generalization ability of DMPs together with Riemannian metrics. Changing the goal pose, the DMP is able to generate the same trajectory from the starting pose, without losing any information.

5.2 STATE OF THE ART

The field of Dynamic Motion Primitives (DMPs) has seen substantial development since its inception. DMPs are a framework that simplifies the generation of complex robotic movements through a combination of nonlinear differential equations and attractor dynamics. Originally introduced by Ijspeert et al. [Ijspeert2013], DMPs have been used extensively for trajectory learning, where they offer robustness to perturbations and the ability to modulate learned motions in real time based on environmental feedback.

Recent advancements in DMPs have focused on addressing their limitations, such as their inability to handle complex, high-dimensional, and nonlinear task spaces. Modifications like goal switching, obstacle avoidance, and multi-dimensional DMPs have been proposed to tackle these issues. Khansari-Zadeh and Billard [Khansari2011] introduced Stable Estimator of Dynamical Systems (SEDS), a method that learns stable nonlinear dynamical systems for trajectory generation, which improves upon standard DMPs by ensuring global asymptotic stability. Furthermore, work by Pastor et al. [Pastor2009] extended DMPs to allow for real-time obstacle avoidance by modifying the attractor landscape.

In addition to these extensions, Riemannian geometry has recently emerged as a powerful tool to enhance DMPs. The introduction of Riemannian Motion Policies (RMPs) by Ratliff et al. [Ratliff2018] has paved the way for learning and controlling movements on manifolds. RMPs provide a way to encode complex, task-relevant motions using

Riemannian metrics, which respect the intrinsic geometry of the underlying manifold. This framework allows for more natural and efficient learning of motions, especially when dealing with articulated robots or tasks defined on curved spaces.

Integrating DMPs with Riemannian manifolds has proven to be a promising approach for enhancing robot learning and control. Riemannian manifolds provide a flexible mathematical structure that can represent the curved and nonlinear nature of many robotic tasks. For instance, learning on the space of rotations (SO(3)) or the space of positive definite matrices (SPD(n)) is naturally handled using Riemannian geometry. Jacquier et al. [Jacquier2020] demonstrated the effectiveness of DMPs on Riemannian manifolds for imitation learning tasks, where the task space is non-Euclidean.

Research by Allenspach et al. [Allenspach2024] has focused on leveraging Riemannian metrics to learn and adapt motions on a robot's configuration space, significantly enhancing its ability to handle real-world complexities such as joint limits and dynamic obstacles. This integration is particularly advantageous in tasks requiring precise manipulation and force control, such as defect detection and correction in collaborative human-robot environments.

The use of Riemannian DMPs in human-robot collaboration has become increasingly relevant in manufacturing and other industrial applications. In these environments, robots are required not only to learn from human demonstrations but also to adapt to new defects and anomalies on the fly. The application of DMPs in these settings has shown promise considering the ability to generalize from a few examples to unseen situations is greatly enhanced by the geometric properties of the Riemannian space.

Additionally, advances in reinforcement learning (RL) and imitation learning (IL) have been combined with DMPs on Riemannian manifolds to further improve robot adaptability and robustness. Methods such as Geometric Reinforcement Learning (GRL) by Zhang et al. [Zhang2015] exploit Riemannian structures to reduce sample complexity and improve convergence rates in policy learning, particularly in environments that are dynamic and highly uncertain.

5.3 OBJECTIVES AND REQUIREMENTS

The primary objective of this deliverable is to develop a robotic system capable of learning and adapting defect working skills from human demonstrations in real time. This involves creating motion planning and control algorithms based on Riemannian DMPs. The system aims to achieve the following objectives:

1. High Adaptability: The robot must generalize learned skills to new parts not encountered during training. The system will employ Riemannian metrics to adapt DMPs dynamically to novel situations.
2. Safety and Efficiency: Collaborative tasks require a high level of safety and efficiency. The robot must predict and avoid collisions with human operators while performing defect correction tasks. Riemannian DMPs provide a more accurate and flexible representation of the task space, which is crucial for

maintaining safety in dynamic environments.

3. Real-Time Operation: The system must operate in real time to ensure seamless integration into a human-robot collaborative setting. This requires optimizing the motion planning algorithms to handle sensor data and adapt motions within a few milliseconds.
4. Robust Learning from Limited Data: Since acquiring extensive real-world data is often impractical due to safety and privacy concerns, the system must efficiently learn from a small number of demonstrations. To this end, DMPs are well fit given that they require just few demonstrations to be able to learn and generalize motion data captured from humans.

The system's KPIs include achieving a convergence time of ≤15 hours for learning new skills, ensuring the system can learn and adapt efficiently, and ≤10 hours of observations to converge to a satisfactory policy. Moreover, similarity measures between the learnt and the human policies make the adoption of DMPs a good choice for this type of task.

Hereafter we report the KPIs (Table 5.1), as described in the Deliverable D2.1 - "Use Case Definition", related to the Learning defect working skills from humans.

Scientific and technological objective	KPI ID	KPI definition	After MAGICIAN
(O1) A robotic perception module integrating visual and tactile sensors. The module will be embedded in a robotic sensor module (the SR, hereafter) and will be used for defects analysis and classification. The SR will replicate the skills of human workers through a learning scheme.	O1-KPI-LRN-SR1	Misclassification rate with respect to human.	≤10%
	O1-KPI-LRN-SR2	Time to convergence.	Observation time ≤ 15h to achieve KPI-LRN-SR1
(O2) A robotic cleaning module attached to a robotic platform (the CR hereafter) equipped with a specialized end-effector to rework defects. The system will learn the necessary skills by observing humans.	O2-KPI-LRN-CR1	Reduction of measurement uncertainty.	RMSE ≤ 5%
	O2-KPI-LRN-CR2	Time synchronisation error among data coming from different sources.	≤ 0.1 ms
	O2-KPI-LRN-CR3	Number of samples to converge to a satisfactory policy.	≤ 10h of observations
	O2-KPI-LRN-CR4	Similarity measures between the learnt and the human policies.	position error ≤ 1mm; orientation error ≤1°; force error ≤ 5N; moment error ≤ 2Nm

Table 5.1. KPIs related to the Learning defect working skills from humans.

5.4 DATA ACQUISITION AND ANNOTATION

Data acquisition for learning defect working skills from humans involves capturing visual data streams from human demonstrations. This process is essential for training the robot to recognize and react to different types of car body on factory floors.

Data collection involves real-world data generation. Real-world data is collected using high-resolution cameras recording human actions. For this deliverable of MAGICIAN, the data adopted is derived from the Human Motion Detection module, considering the necessary information for this module is the trajectory of the human hand with respect to the car body part of interest. No other sensors are employed to learn human motion data, as the use of a cartesian impedance controller will optimally control the interactions between the robot and the chassis when reproducing human-learned trajectories.

On the other hand, the annotation process is a critical step in preparing the dataset for supervised learning. We use a combination of automated and manual techniques to ensure high-quality annotations. State-of-the-art computer vision models like MocapNET are employed to automate the annotation of human poses and actions, while manual oversight is provided to correct any inaccuracies, particularly in challenging scenarios with occlusions or complex poses. Another fundamental part of manual annotation is necessary to define the car part for which the motion is learned. In this way, we can define a motion primitive for each car part, and then to generalize even in cases where the car model is changed. This dataset is essential for training machine learning models that can exploit the geometric properties of DMPs together with Riemannian manifolds for efficient learning and adaptation.

5.5 METHODOLOGIES EMPLOYED

The methodologies employed in this research involve a novel integration of Dynamic Motion Primitives (DMPs) with Riemannian manifold learning to achieve adaptive and generalizable robot manipulation skills for defect detection and correction tasks in collaborative human-robot environments. The core framework leverages human demonstrations to learn variable impedance manipulation skills, which are then generalized to new scenarios by exploiting the geometric properties of Riemannian spaces. The approach can be broadly divided into four key components: Trajectory Collection and Preprocessing, Riemannian-based Skill Learning, Skill Generalization using Extended DMPs, and Real-Time Robotic Control. While the extraction of the skill is performed on MATLAB, robotic control has been developed in C++ and communication is obtained using ROS topics.

Trajectory Collection and Preprocessing

The first step involves collecting multiple trajectories of human demonstrations for various defect correction tasks, such as defect detection or surface finishing. As can be

seen in Figure 5.2, these trajectories, consisting of both positional and orientation data of the human hand, are captured using external cameras. For each demonstration, the position p(t)∈R3 and orientation q(t)∈S3 (unit quaternion) of the human hand are recorded. Each trajectory is represented as:

$$O_i(t) = \{p_i(t), q_i(t)\},\ i = 1,2,\dots,N$$

where N is the number of demonstrations. The recorded trajectories are time-aligned to a common time frame $[0, T]$ to ensure consistency. This alignment is achieved using a linear time warping function:

$$O_i(t) = O_i\left(\frac{T(t - t_0)}{(t_1 - t_0)}\right),\ i = 1,2,\dots,N,$$

where t_0, t_1 are the start and end times of each demonstration.

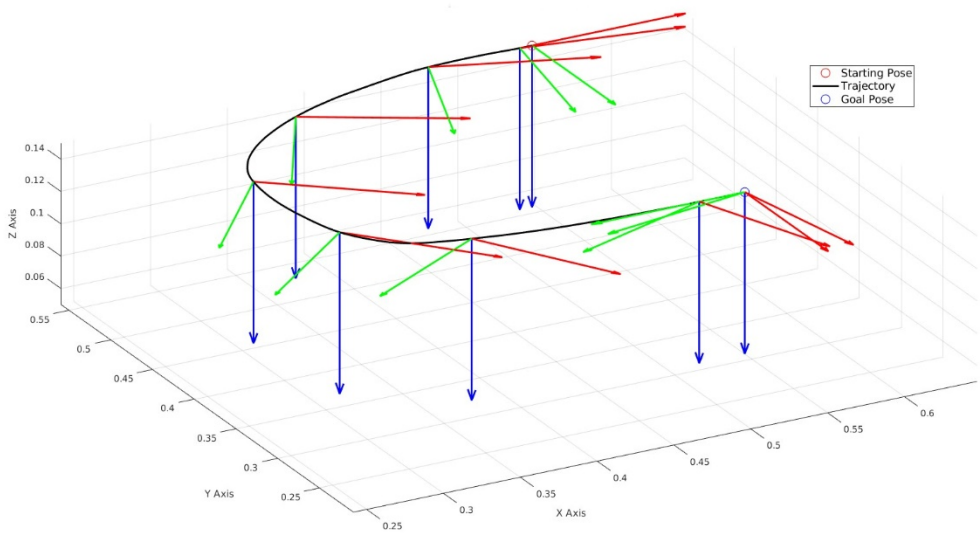


Figure 5.2. Example of a recorded trajectory performed by a human during the sensing phase. The orientation in this case is expressed in form of quaternion.

After having aligned all the trajectories for each car part, we transformed the obtained pose from the camera reference frame to a specific initial frame recognized on the car part. In this way, we can later generalize the obtained trajectories in case the component is changed with one of different dimensions, by recognizing an initial and a final pose on the car part.

Both positional and transformed orientation data are then encoded using a Gaussian Mixture Model-Gaussian Mixture Regression (GMM-GMR) framework, which captures the variability in the demonstrations and allows for the estimation of the mean trajectory. The encoded mean trajectory is obtained through regression of the demonstrated data, providing a robust representation for skill reproduction.

Riemannian-Based Skill Learning

To handle the nonlinearities and complexities of the robot's configuration space, the demonstration data is encoded on a Riemannian manifold. The orientation data, represented as quaternions, is converted to an axis-angle quaternion, such that the angle is learned through a classical Cartesian DMP along with the cartesian position while the axis is mapped to a tangent space using the Quaternion Logarithmic Mapping Function.

This transformation allows the axis to be treated as a decoupled 3D vector in the tangent space, simplifying the process of trajectory encoding and generalization. Once having extracted the initial and final target orientations, we proceed to evaluate the geodesic between two consecutive orientations, i.e. the distance on the Riemann manifold expressed as the arccosine of the angle between them. In this way, we compressed a 3-dimensional data (axis) as a unique value, the geodesic (Figure 5.3).

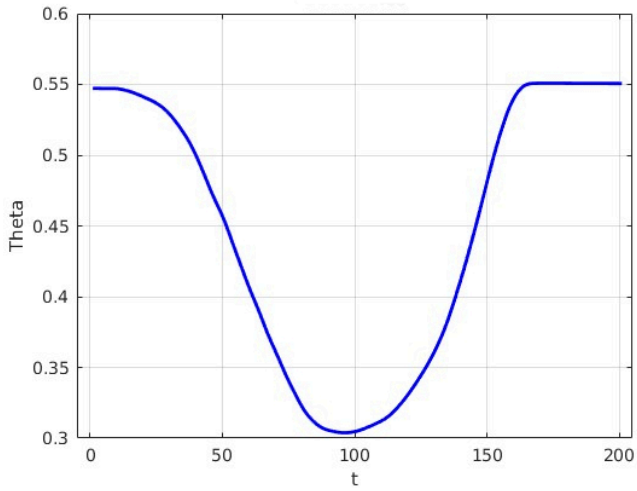


Figure 5.3. Geodesic between two consecutive orientations. The angle of difference is expressed in radians. This representation is especially beneficial for generalization and to compress data.

Apart from the difference in the evaluation of the distance between two consecutive vectors, the rest of the Riemannian DMP is exactly the same as a classical one, where, using a combination of a stable, attractor-based system and a flexible, non-linear function, the attractor system drives the motion towards a goal, while the non-linear function adds adaptability, allowing the robot to adjust the motion in response to changes in the environment. Once a new trajectory is integrated through the learned DMP, we apply the Rodriguez formula to generalize on different initial and final orientations.

Surface-based Learning

Based on Riemmanian DMP, we extended the definition of the required differential operators to work on surfaces described as polyhedral meshes. The resulting framework, named *MeshDMP*, enables the learning of motion policies which can be easily

transferred to various surfaces wit without constraining the geometry topology and curvature [DalleVedove2025].

As shown in Figure 5.4, the MeshDMP framework can easily retarget motion policies on different surfaces by preserving the overall shape of the initial demonstration. MeshDMP can so be used to learn end-effector position trajectories on surfaces, and we envision to learn orientation profiles relative to the surface. By doing so, we can learn skills that not only can generalise to unforeseen defect on the car body but can be transferred to new production processes with little to nonadditional training.

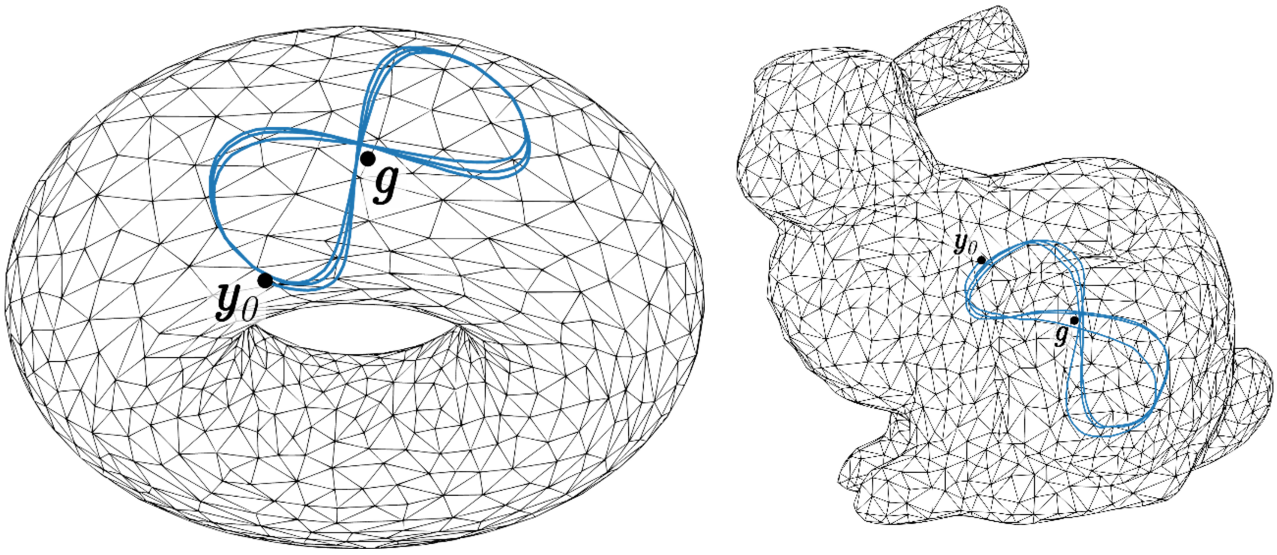


Figure 5.4. Example of motion retargeting on different meshes. In this case, an 8-shaped trajectory is learned from synthetic data from a flat plane, and the corresponding policy is retargeted onto other surfaces, such as the torus and the Stanford bunny.

Real-Time Robotic Control

The calculated pose trajectories are then used to compute control commands based on a Cartesian Impedance Control framework. Operating in Cartesian space allows for the modulation of the manipulator's compliance in specific directions, enabling more convenient and flexible handling of physical interactions. For example, in a grinding or polishing task where the manipulator needs to move along a rigid surface, the stiffness can be reduced in the direction normal to the surface. This ensures consistent contact with the object while maintaining high precision along the desired path in the other directions.

The proposed approach has been validated in real-world experiments with a 6-DoF robot manipulator, showing that the integration of DMPs with Riemannian manifolds leads to more adaptable and robust skill acquisition and reproduction, particularly in environments that involve possible changes in the initial and final poses.

5.6 RESULTS AND FINDINGS

The preliminary results from the integration of Dynamic Motion Primitives (DMPs) with Riemannian manifolds have demonstrated promising outcomes in terms of learning defect working skills from human demonstrations. Initial experiments were conducted to evaluate the system's capability to learn and adapt to various defect detection and correction tasks, using visual data.

Using a dataset of human demonstrations collected through high-resolution cameras, the system was trained to replicate human-like defect detection and correction behaviours. The integration of DMPs with Riemannian metrics allowed the robot to generalize well across different types of defects and surface geometries. The Riemannian DMP framework achieved a mean of **3%** of error in reproducing the trajectory regarding the position (Figure 5.5) and a mean of **8%** regarding the orientation (Figure 5.6). This suggests that the Riemannian representation effectively captures the underlying geometric and physical properties of the task space, resulting in robust learning outcomes.

The preliminary findings also indicated that the Riemannian DMPs were able to interpolate and extrapolate motions more smoothly than their Euclidean counterparts.

For example, the system successfully learned to adapt to new locations on different parts of car body panels (Figure 5.7), achieving a task success rate of **96%** when evaluated on a benchmark set of test cases, which included defects of various sizes and types.

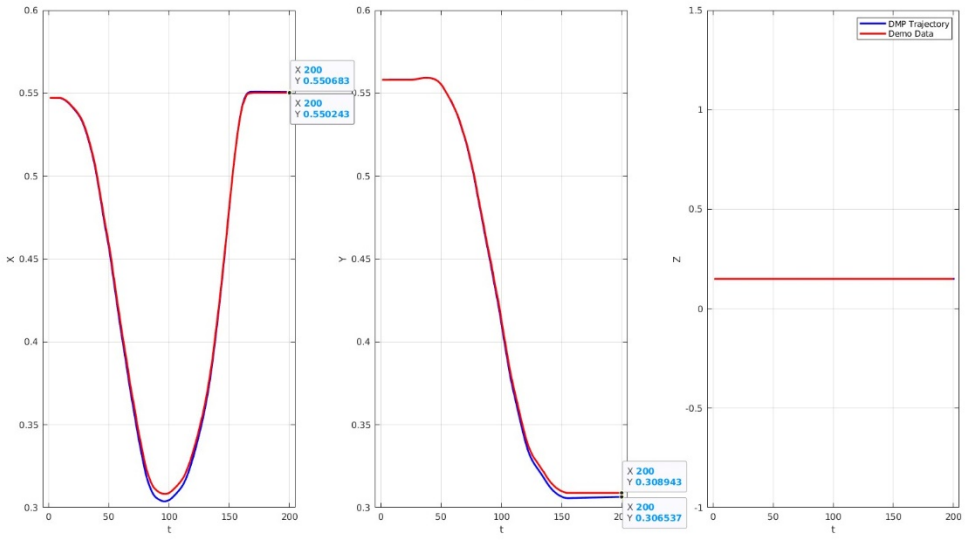


Figure 5.5. Demo human trajectory (red) and trajectory reproduced by the Cartesian DMP (Blue) of the position *(x, y, z)*. As it can be seen, the error is very low, and it can still be improved with finer tuning of the parameters belonging to the DMPs.

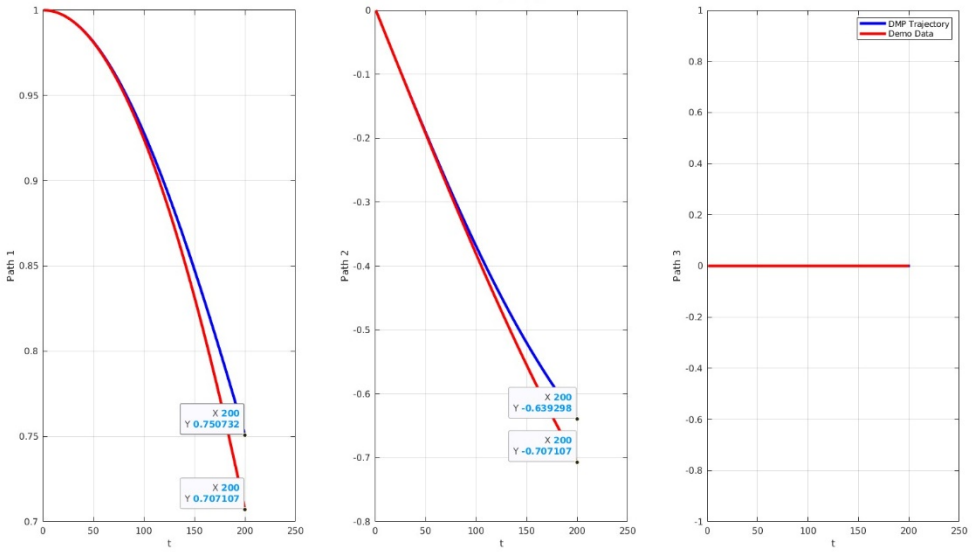


Figure 5.6. Demo human trajectory (red) and trajectory reproduced by the Riemannian DMP (Blue) of the orientation (expressed as axis). As it can be seen, the error is higher than the position but the performances when generalizing are superior to what can be obtained with the classical DMP.

The developed system still needs to be optimized to reach near real-time performance, with a motion planning and adaptation latency of approximately 1 to 2 s. This still does not meet the real-time requirements for collaborative human-robot environments where rapid adjustments are necessary for safety and efficiency.

Preliminary trials in simulated and real-world environments showed that the robot's ability to learn from human demonstrations while maintaining safe interaction distances needs to be enhanced. The Riemannian DMPs enabled smoother motion transitions and more predictable behaviour, reducing the likelihood of unintended collisions. However, in tasks where the robot had to operate near human operators, such as collaborative defect detection and correction, safety metrics, such as the average stopping distance from human collaborators, need to be implemented. Nevertheless, the application of this learning technique together with a cartesian impedance controller provides a first degree of compliance with human collaborators.

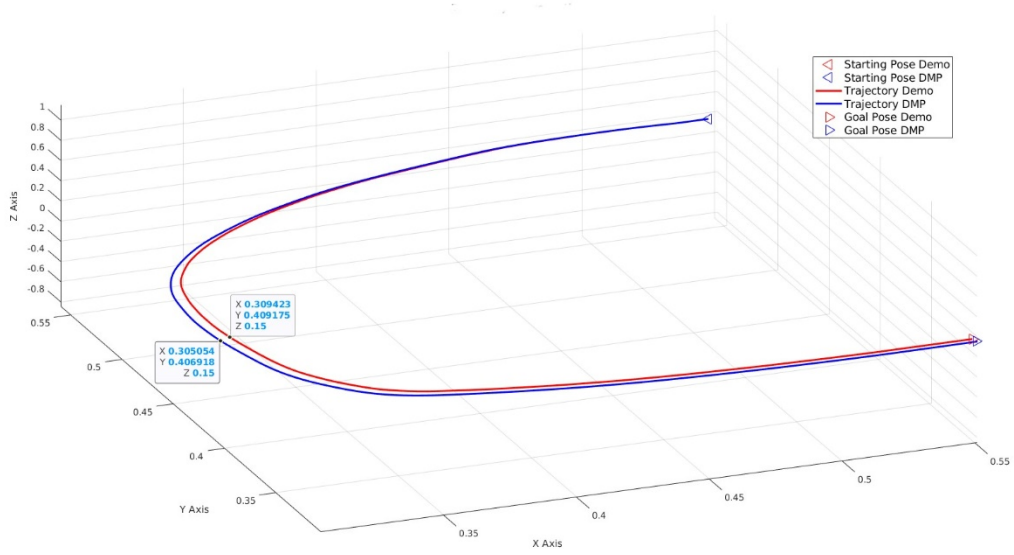


Figure 5.7. Complete trajectory of the demo and the DMP trajectories. As can be noticed, the motion of the human is preserved when replicated, and the final error is very low.

5.7 CHALLENGES AND LIMITATIONS

Despite the promising preliminary results, several challenges and limitations were identified in the integration of Dynamic Motion Primitives (DMPs) using Riemannian manifolds for learning defect working skills from humans. These challenges need to be addressed to fully realize the potential of this approach in real-world applications.

One of the primary challenges encountered is the computational complexity associated with the Riemannian manifold calculations, particularly in high-dimensional task spaces. While the use of GPUs has mitigated some of these issues, the system's scalability remains a concern. For instance, when scaling up to more complex tasks that involve multiple types of defects and larger workspaces, the computation time for Riemannian distance calculations can become a bottleneck. This can limit the system's ability to maintain real-time performance, especially in highly dynamic environments where quick adaptations are crucial.

While the Riemannian DMP framework has shown improved adaptability to new environments, its robustness is still limited by the inherent variability and uncertainty in real-world settings. For instance, changes in environmental conditions such as lighting, surface reflectance, or temperature can adversely affect the robot's performance. Developing more robust perception algorithms that can dynamically adjust to these variations is essential for the system's deployment in real-world manufacturing scenarios.

The integration of Riemannian DMPs also introduces complexity in terms of system usability for non-expert operators. While the approach offers significant performance

benefits, the learning and adaptation processes are not yet fully transparent to human operators. This lack of interpretability can hinder user trust and acceptance, particularly in collaborative settings where humans and robots must work together seamlessly. Simplifying the user interface and providing intuitive feedback mechanisms are crucial steps toward improving usability.

Addressing these challenges will involve several future research directions, including:

- Optimization of Riemannian Calculations: Developing more efficient algorithms and approximation methods to reduce the computational overhead associated with Riemannian metrics.
- Advanced Data Augmentation: Leveraging generative models and simulation environments to create more diverse and representative training datasets.
- User-Centric Design: Focusing on human factors engineering to create more intuitive and user-friendly interfaces that facilitate better human-robot collaboration.

By tackling these challenges, the system can be made more robust, efficient, and user-friendly, ultimately enhancing its applicability in industrial settings where defect detection and correction are crucial.

6 CONCLUSIONS

This deliverable reported the development activities in the year following D3.1, which aims to develop the required perception systems that will provide the rest of the system with the necessary information to carry out the required tasks. Specifically, conducted research, developed prototypes, experiments and preliminary results are presented in detail in all the relevant perception areas, include defect detection using visual and tactile input, human presence detection and pose estimation, and further processing of this lower-level input for tasks such as learning defect reworking. The presented work has already yielded encouraging preliminary results and useful insights and lays the groundwork for further development of the perception systems and their integration in the MAGICIAN platform.

6.1 FURTHER DEVELOPMENT OF THE VISUAL PERCEPTION MODULE

We can summarize the next steps for each of the components with the following list.

Defect Sensing module:

- Switch to a metallic fabricated sensor construction to make it sturdier and more robust compared to the current 3D printed end-effector prototype.
- Reduce the physical dimensions of the sensor to make it more versatile in tight corners and crevices of the car body.
- Upgrade LEDs to higher wattage COBs that will enable lower exposures and mitigate ambient light providing stronger polarization patterns.
- Use light pulses with hardware synchronization to the camera shutter instead of continuous light operation with software synchronization to the camera.
- Continue improving software including NNs, annotation tools and ROS nodes.
- After finalizing the new metallic sensor design with stronger lights, record finalized data and train the network with the target physical dimensions and exposure values to yield the best model possible.
- Use passive range finding provisions of the sensor with data from Altinay factory setup to provide an initial transformation of the car with respect to the robot position to account for any positioning errors of the car.

Motion Detection module:

- Improve human pose estimation by incorporating data from our use case.
- Combine perception data in a common coordinate frame w.r.t the robot using ROS

6.2 FURTHER DEVELOPMENT OF THE TACTILE PERCEPTION MODULE

The next phase of development for the Tactile Perception Module will focus on creating a metallic version of the sensor with a reduced physical footprint to enable easier reaching of internal surfaces of the car chassis.

6.3 INTEGRATED SENSING

As described above, we are developing two families of solutions for defect detection: vision-based techniques and tactile techniques. We are planning to implement a synergistic application of the two techniques. This will be possible by deploying the two sensors on the same end-effector. More specifically, we can classify in the following classes:

- Class 1: Defects that for their type and/or position can be efficiently detected only by vision sensors
- Class 2: Defects that for their type and/or position can be efficiently detected only by tactile sensors
- Class 3: Defects that can be analysed by both techniques.

For defects belonging to Class 3, it is possible to adopt some type of sensor fusion to reduce the probability of false positives or false negatives.

6.3.1 MULTI-MODAL FUSION

Given the nature of the application and the knowledge on the main MAGICIAN use case, we have to proceed with a combined application of the two different sensing methods discussed in Section 2 and Section 3. Indeed, there are car body parts and type of defects that can be more suitable for the vision system rather than the tactile one, and the other way around. Nonetheless, this information comes with a probabilistic description that is a function of different factors, e.g., welding gun currents/voltages used during the welding process, type and imperfections of the materials adopted, probabilistic location of the defect, etc., hence the tools described in D4.2 consider this stochasticity for each sensing methodology in isolation. What is foreseen in this respect in WP3 is the synergistic use of these two stochastic modalities to probabilistically optimise the scheduling of the two sensing systems and decrease the perception time, as already depicted in D3.1. Therefore, this multi-modal fusion is postponed further in the project

execution once sufficient data evidence is collected from WP5 first tests.

7 REFERENCES

[Ant23] Antonucci, A., Bevilacqua, P., Leonardi, S., Paolopoli, L., & Fontanelli, D. (2023). Humans as path-finders for mobile robots using teach-by-showing navigation. Autonomous Robots, 47(8), 1255-1273.

[Yan2024] S. Yang, H. Li, C.-M. Pun, C. Du, and H. Gao, "Adaptive spatial-temporal graph-mixer for human motion prediction," IEEE Signal Processing Letters, vol. 31, pp. 1244–1248, 2024.

[Tian2024] S. Tian, M. Zheng, and X. Liang, "Transfusion: A practical and effective transformer-based diffusion model for 3D human motion prediction," IEEE Robotics and Automation Letters, vol. 9, no. 7, pp. 6232–6239, 2024.

[Lin14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in Computer Vision–ECCV 2014, vol. 8693, pp. 740–755, 2014.

[Andriluka14] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3686–3693, 2014.

[Lee23] S. Lee, J. Rim, B. Jeong, G. Kim, B. Woo, H. Lee, S. Cho, and S. Kwak, "Human pose estimation in extremely low-light conditions," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 704–714, 2023.

[Wu17] J. Wu, W. Zhang, C. Huang, and K. Huang, "AI Challenger: A large-scale dataset for going deeper in image understanding," arXiv preprint arXiv:1711.06475, 2017.

[Li22] Li, J., Zhang, J., Maybank, S.J. et al. Bridging Composite and Real: Towards End-to-End Deep Image Matting. Int J Comput Vis 130, 246–266 (2022).

[Yang24] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth Anything v2," arXiv preprint arXiv:2406.09414, 2024.

[Yuxi19] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," Facebook AI Research, 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[Ye23] M. Ye, J. Zhang, S. Zhao, J. Liu, B. Du, and D. Tao, “DPText-DETR: Towards better scene text detection with dynamic points in transformer,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 3, pp. 3241–3249, 2023.

[Ijspeert2013] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: Learning attractor models for motor behaviors,” Neural Computation, vol. 25, no. 2, pp. 328–373, 2013.

[Khansari2011] S. M. Khansari-Zadeh, and A. Billard, "Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models," in IEEE Transactions on Robotics, vol. 27, no. 5, pp. 943-957, Oct. 2011.

[Pastor2009] P. Pastor, H. Hoffmann, T. Asfour and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009, pp. 763-768.

[Ratliff2018] Nathan D. Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield and Dieter Fox, “Riemannian Motion Policies”, ArXiV, 2018.

[Jaquier2020] N. Jaquier, L. Rozo and S. Calinon, "Analysis and Transfer of Human Movement Manipulability in Industry-like Activities," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 11131-11138.

[Allenspach2024] M. Allenspach, M. Pantic, R. Girod, L. Ott, and R. Siegwart, “Task Adaptation in Industrial Human-Robot Interaction: Leveraging Riemannian Motion Policies”, ArXiV, 2024.

[Zhang2015] B. Zhang, Z. Mao, W. Liu, J. Liu, “Geometric Reinforcement Learning for Path Planning of UAVs.”, Journal of Intelligent and Robotic Systems, 77, 391–409 (2015)

[DalleVedove2025] M. Dalle Vedove, F.J. Abu-Dakka, L. Palopoli, D. Fontanelli, and M. Saveriano, "MeshDMP: Motion Planning on Discrete Manifolds using Dynamic Movement Primitives, 2025 IEEE-RAS International Conference on Robotics and Automation (ICRA)