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 D3.1

First delivery of perception systems

DUE DATE OF DELIVERABLE:

30.09.2024

ACTUAL SUBMISSION DATE:

30.09.2024

Work Package	**WP3 - Data acquisition and skills learning**
Associated Task	**T3.1 - Perception System, data acquisition and processing**
Deliverable Lead Partner	**FORTH**
Main author(s)	**Ammar Qammaz, Iason Oikonomidis, Luigi Palopoli, Antonis Argyros, Geert Driessen, Roos van Dongen, Gionata Salvietti, Nicole D'Aurizio, Domenico Prattichizzo**
Internal Reviewer(s)	**Daniele Fontanelli**
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	25/08/2024	First Draft	Ammar Qammaz, Iason Oikonomidis, Luigi Palopoli, Gionata Salvietti	FORTH, UNITN, IIT
0.2	25.09.2024	Internal Review	Daniele Fontanelli	UNITN
1.0	30.09.2024	Final Version	Antonis Argyros	FORTH

MAGICIAN
AUTONOMOUS DEFECTS DETECTION AND REPAIR IN MANUFACTURING

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. 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, preliminary results obtained, and the challenges and limitations encountered during the development process, setting the foundation for future work.

More specifically, this deliverable offers a detailed description of the perception system's components, including their requirements and specifications as defined by the industrial partners, and the overall system architecture designed to meet these needs. It furnishes a thorough understanding of the operations and methodologies foreseen in the subsequent stages of the project, ensuring alignment with the project objectives and facilitating improvements in industrial operations.

The visual perception module is described first, starting with an introduction to the state of the art in visual defect detection. It includes the objectives and requirements identified in collaboration with partners such as TOFAS and CRF, the design and implementation of the camera system, and the data acquisition and annotation processes. The methodologies employed involve tailored sensor design and machine learning techniques aimed at accurately detecting imperfections in automotive components. Preliminary results demonstrate promising capabilities, while challenges such as the finalization of the sensor design, the precise control of observation conditions, and the acquisition of more training data are discussed to outline areas needing further effort.

Similarly, the tactile perception system for imperfections detection is presented in detail. This section introduces the tactile sensors utilized, the data acquisition and annotation methods, and the specific methodologies applied for processing recorded data. The objectives and requirements are again defined in collaboration with industrial partners to ensure relevance and applicability. Preliminary findings highlight the effectiveness of tactile sensing in identifying surface defects, with discussions on challenges like data

variability, sensor sensitivity, and the need for extensive datasets to train robust models.

The deliverable also covers human motion detection, emphasizing its importance in capturing and interpreting human movements within manufacturing tasks. It reviews the state of the art, defines objectives and requirements, and describes the data acquisition and annotation strategies. Preliminary results indicate the feasibility of accurately modelling human motions, although challenges such as real-time processing constraints and occlusions are identified.

An important aspect of the project is the learning of defect detection skills from humans. The relevant section outlines the challenges and objectives of transferring human expertise to robotic systems using Dynamic Motion Primitives (DMPs) enhanced with Riemannian manifolds. By capturing the complex, nonlinear motion patterns of human operators, the system seeks to generalize these skills to various car parts and surfaces. Preliminary results indicate promise in mimicking human performance, though challenges remain in achieving real-time responsiveness and handling the subtleties of human decision-making.

In the Conclusions, the deliverable outlines the next steps for further development of the perception modules. Plans include enhancing the visual and tactile perception systems, integrating multi-modal sensing for more robust detection, and exploring active sensing strategies to improve efficiency and accuracy. Future work will address the identified challenges, refine the methodologies, and focus on validating the systems within real-world industrial environments.

The submission of this document (D3.1) is part of Work Package WP3 (Perception System Development) and is the outcome of the three Tasks comprising it, Tasks T3.1, T3.2 and T3.3, delivered at the end of Month 12 of the project.

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
AoLP	Angle of linear polarization
BMI	Body Mass Index
CAD	Computer Aided Design
CMOS	Complementary metal oxide semiconductor
CNN	Convolutional Neural Network
D	Deliverable
DCT	Discrete Cosine Transform
DMP	Dynamic Motion Primitives
DoLP	Degree of linear polarization
EC	European Commission
GDPR	General Data Protection Regulation
GigE	Gigabit Ethernet
GPGPU	General Purpose Processing Unit
GPU	Graphics Processing Unit
GUI	Graphical User Interface
EC	European Commission
HMR	Human Mesh Recovery
KPI	Key Performance Index
LED	Light Emitting Diode
MOCAP	Motion Capture
MP	Megapixel
NN	Neural Network
ONNX	Open Neural Network Exchange
PDR	Paintless Dent Repair

PIR	Passive infrared
POSIX	Portable operating system interface
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
SfP	Shape from Polarization
STFT	Short-Time Fast Fourier Transform
UDP	User Datagram Protocol
ViTs	Vision Transformers
WP	Work package
WT	Work Task
YOLO	You Only Look Once

1 INTRODUCTION

This deliverable provides a comprehensive report of the preliminary state of the MAGICIAN sensing modules for data acquisition and skill 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 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.

1.1 PURPOSE AND SCOPE

The primary objective of this deliverable is to provide a comprehensive overview of the visual and tactile perception systems, which constitute the core technological components for defect detection and classification in the MAGICIAN project. These systems are designed to facilitate the identification and categorization of manufacturing defects, thereby enabling automated reworking processes. This document describes how these perception systems are being researched, designed, and implemented during the first year of MAGICIAN.

The visual perception module includes hardware design and computer vision techniques to detect and classify defects in car components. The tactile perception module employs sensors and data analysis for the detection of surface irregularities and other tactile defects that are not easily visible. Overall, the idea behind the selection of these two sensing modalities is to complement each other. Besides these two sensing modules, this document outlines methods for the visual observation of human operators for human-robot collaboration and learning modules that enable robots to acquire defect reworking skills from human demonstrations. The deliverable sets the groundwork for subsequent iterations and refinements that will lead to a fully operational perception system integrated within the manufacturing line.

1.2 CONTRIBUTION TO PROJECT OBJECTIVES

The progress reported in this deliverable contributes in several ways to the broader goals of MAGICIAN, which span scientific, technological, social sciences, and demonstration Objectives. Specifically, the stated Objectives of MAGICIAN are

Scientific and Technological Objectives:

- O1: A robotic perception module integrating visual and tactile sensors for

defects analysis and classification.

- O2: A robotic cleaning module with a specialised end-effector for defect reworking.
- O3: A software robotic platform integrating services for perception and cleaning modules.
- O4: A closed-loop defect detection and avoidance system for robotic and welding processes.
- O5: Development of two TRL 7 integrated prototypes for defect analysis and reworking.

Social Sciences and Humanities (SSH) Objectives

- O6: A human-centred approach to human-robot collaboration, promoting usability, safety, and trustworthiness.

Demonstration Objectives

- O7: Demonstration of the prototypes in operational scenarios.
- O8: Expansion of MAGICIAN's scope and applicability via Financial Support to Third Parties (FSTP).

The perception systems detailed in this deliverable form the backbone of the MAGICIAN platform's perception capabilities, enabling accurate defect detection, supporting reworking strategies, and facilitating effective human-robot collaboration. These systems provide essential data and insights that drive the platform's core functions, aligning with the project's ambition to enhance automation in defect handling through a robust, adaptable, and human-centred approach.

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. A synthetic list of the most important relations is offered next.

- **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 is structured into seven main 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

1.5.1 REQUIREMENTS AND SPECIFICATIONS

The sensing components described in this document rely on the needs and requirements pointed out during the analysis carried out in the WP2 and reported in the D2.1 – "Use Case Definition", where the identified automotive Use Cases are described, constituting the fundamental playground for the ongoing MAGICIAN research and activities. As pointed out the Section 1.3, the specifics of these Use Cases lead to demanding constraints on the perception system's features and accuracy, on its integration within a robust robotic platform, and on the final cost of the solution. These topics will be further explored and discussed in Sections 2.3 of this report, where KPIs related to the Visual Perception are pointed out, in Section 3.3, with KPIs related to Tactile Perception, in Section 5.3, with KPIs related to the Learning.

1.5.2 SYSTEM ARCHITECTURE

The system architecture of the MAGICIAN perception systems is designed with modularity and flexibility at its core, utilizing ROS (Robot Operating System) as the primary integration framework. This architecture allows for seamless communication between various perception modules and supports the development and deployment of automated defect detection and classification solutions in car manufacturing lines.

The architecture comprises key perception modules, including visual defect detection, tactile defect detection, and human detection, alongside supporting sensing modalities such as human tracking and force sensing for learning defect reworking by demonstration. These modules are designed to operate independently yet interconnect through ROS wherever required, facilitating data exchange and coordinating actions across the system.

The perception modules are containerized to enhance isolation and modularity, leveraging platforms such as Docker or equivalent alternatives that offer simplicity and robust isolation capabilities. This containerized approach allows each module to be developed, tested, and deployed independently, simplifying integration and maintenance while ensuring consistent performance across different environments.

Although specific hardware and detailed communication protocols are yet to be defined, the high-level architectural approach prioritizes scalability and adaptability, enabling the system to evolve as new requirements and insights emerge during the project. The initial considerations also include basic performance, scalability, and fault tolerance aspects, which will be further refined in subsequent development phases.

As the system development progresses, the architecture will be iteratively updated to incorporate more detailed specifications, refined interfaces, and comprehensive performance metrics, ensuring alignment with the overall objectives of the MAGICIAN project.

2 VISUAL PERCEPTION FOR IMPERFECTIONS DETECTION

The visual perception system for imperfections 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.

The visit to the industrial plant of TOFAS in January 2024 provided hands on experience and gave all technical partners access to the first metal sheet samples with imperfections that where crucial to study and understand the requirements of the problem. The relevant use cases and considerations were described in Deliverable 2.1 and, more specifically, in Sections 2.1.3.1.1 and 2.2.3.1.1 of that Deliverable.

In April 2024, FORTH received a large shipment of annotated defective material (shown at the FORTH premises in Figure 2.1) that has been instrumental in forming a prototype solution for the visual perception module that will be presented in detail in this section.

Figure 2.1. The shipment of defect samples deployed at the dedicated space for the project in the Foundation for Research and Technology Hellas (FORTH).

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 and
- material defects,

each of which uniquely affects the involved surfaces, as illustrated in Figure 2.2. Metal sheets feature varying textures from part to part. Often, they exhibit imperfections, such as scratches, that may seem like defects to an untrained observer. However, these marks generally pose no issues for the manufacturing process, as they are covered without problems once paint is applied and should therefore be considered non-defective. Nevertheless, true defects are typically small, with the minimum targeted defects being around 300 microns in size, and they can occur over large areas of several square meters across the car's surface.

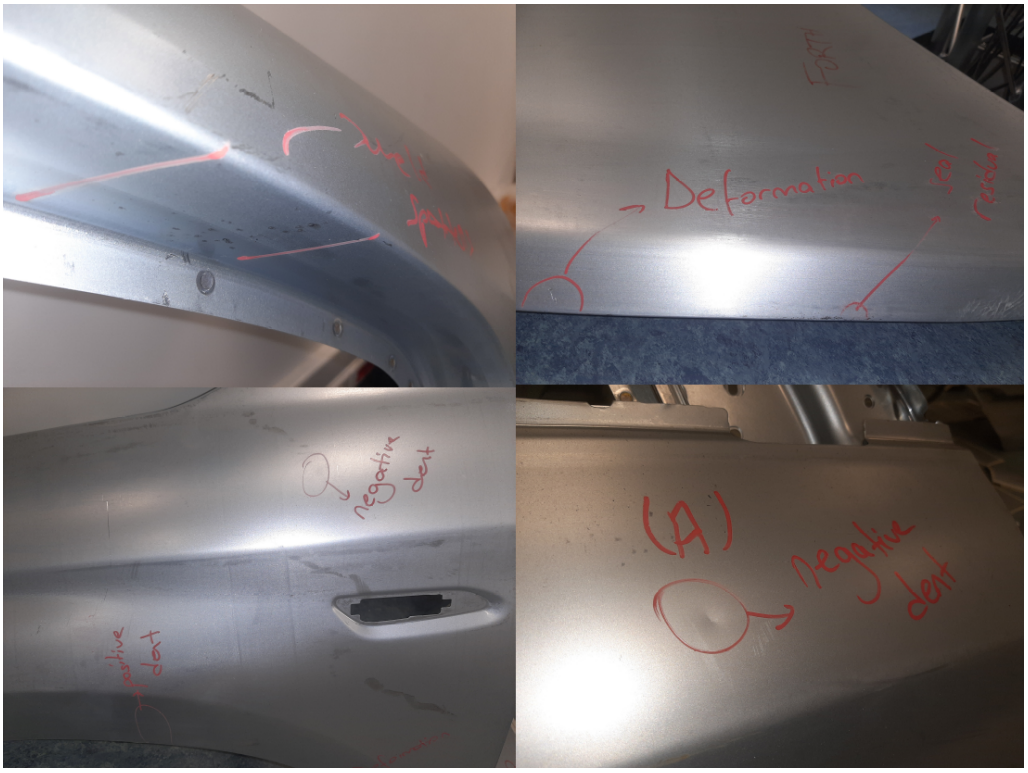


Figure 2.2. Annotated defect examples from the shipment of TOFAS to FORTH, of welding spots (top left), material deformation and seal residuals (top right), positive / negative dents (bottom row). Defects range in severity, size and location.

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.

Figure 2.3. After experimentally testing various sensors, we decided to base the camera system on the SONY XCG-CP510 sensor that can detect light polarization. Coupled with a light source with known polarization, this camera provides 4 angle readings for each camera pixel at 0°, 45°, 90° and 135°, thus providing a rich source of observations for the neural network defect classifier.

2.2 RELATED WORK AND STATE OF THE ART

Figure 2.4. Figure 2.4. Commercial aftermarket tools for dent detection tasks include linen reflective markers with black stripes of varying width (Left), or LED based striped lights that can individually be turned on and off with alternating black bands (Right). These "structured-light" techniques are geared towards painted smooth surfaces. By altering the observer's gaze and the pattern reflection on the car body in different angles, dents become more noticeable compared to constant ambient light.

Paintless Dent Repair (PDR) is the aftermarket automotive industry term describing detecting and repairing minor dents and imperfections on painted car surfaces. The term paintless describes that using tools like plungers and suction cups, without repainting the surface, dents can be reworked to become more subtle. The specialized lights used for PDR utilize high-contrast, often striped, LED or fluorescent lighting (Figure 2.4) to create reflections on the painted vehicle's glossy and shiny surface, making small dents and deformations more visible to the human eye. The key advantage of PDR lights is their ability to accentuate surface irregularities that are otherwise hard

to detect under normal lighting conditions. By adjusting the position of the light and the angle of observation, aftermarket technicians can more clearly see the depth and extent of a dent.

Unfortunately, detecting dents and imperfections on an unpainted surface is a much harder task. The rough texture of raw metal sheets, before applying paint, makes dents and imperfections much harder to detect and thus is a much more challenging scenario. Regardless of the lack of direct application to our use case, these widely used tools and techniques highlight the importance of projecting structured light on surfaces with imperfections to help the technician facilitate detection (even if the technician is human, and the human brain is uniquely equipped to perform pattern recognition tasks). Keeping these techniques in mind, adopting structured light, using machine learning for classification, and utilizing light polarization as an additional modality that can help us observe defects, we can begin to survey the literature to build a visual perception system for defect detection.

Figure 2.5. Left: Convolutional neural networks perform convolutions of the image gradually transforming it to a representation that is easy to classify. Middle: Vision transformers take this approach one step further by having an attention mechanism, where different attention "heads" help create a more robust model with better understanding of the observed scene. Unfortunately, the examined problem cannot be naively tackled through the above methodologies. Defects can appear in any part of a metal sheet and are very local, thus requiring an approach independently targeting each part of the image.

The field of visual perception for defect detection has undergone significant advancements, particularly with the adoption of machine learning and deep artificial neural network techniques. Traditional methods, which relied on manual inspection or basic image processing techniques like thresholding and edge detection, were often limited in their ability to detect subtle imperfections across varying textures and shapes of industrial components. These methods also required extensive calibration and were highly sensitive to environmental conditions, which could lead to inconsistent results in complex manufacturing settings.

Convolutional Neural Networks (CNNs) [Alom18] revolutionized the way defects are detected on surfaces by automatically learning features from large datasets. CNNs excel at identifying patterns in images and can be trained to recognize a wide range of objects. Alexnet [Kriz12] pioneered convolutional network classification setting a new state of the

art on the Image Net benchmark. More recently, Vision Transformers (ViTs)[Khan22] have emerged as a promising alternative. ViTs utilize an attention mechanism that allows the model to focus on different parts of the image, effectively capturing both global and local features. This makes ViTs particularly well-suited for complex classification tasks. However, the complexity of the attention mechanism and the large computational resources required for ViTs can make them impractical for real-time applications, especially in an industrial setting where high-speed processing is crucial.

Another approach that has gained traction is the use of hybrid models that combine the strengths of CNNs and ViTs. These models leverage the hierarchical feature extraction capability of CNNs while incorporating the attention mechanisms of ViTs to achieve a more robust and accurate defect detection system. Despite the potential of these hybrid models, they still face challenges in terms of computational efficiency and the need for large, annotated datasets.

The de-facto state of the art standard in real-time image classification is the YOLO (You Only Look Once) family of methods. Since the original publication of the method [Redm16] a series of incremental updates have been proposed [Wang24], with recent versions also supporting extractions of per pixel masks (Ultralytics YOLO V8 segmentation model). Due to its popularity, the YOLO architecture seems to be a valid candidate for the task. After careful consideration though, it exhibits several drawbacks when considered in the context of the defect detection scenario, especially when trained with a very small number of samples, and even more so when considering their dimensions and other, irrelevant metal markings.

Given the specific challenges of detecting very small and local defects on metal sheets in an automotive manufacturing environment, traditional CNN and ViT-based approaches alone are insufficient (Figure 2.5). The defects in question are often as small as 300 microns and can appear in any part of the metal sheet, making the detection task highly localized. Therefore, a more specialized approach is required, one that can handle high-resolution images and focus on very small areas without overly relying on contextual information. In the context of the MAGICIAN project, we are exploring advanced techniques that go beyond conventional CNNs and ViTs. This includes the development of a specialized neural network architecture tailored for high-resolution, localized defect detection. Our approach integrates multi-scale feature extraction and local tile-based mechanisms, enabling the system to focus on minute details while maintaining the ability to process large surface areas efficiently through image batching and GPGPU parallelization. Additionally, the use of polarization imaging, as facilitated by the SONY XCG-CP510 sensor, provides additional data channels that can be leveraged to enhance the detection accuracy of small defects.

Overall, while the state of the art in visual perception for defect detection has made significant strides, the specific requirements of detecting small, localized imperfections in automotive metal sheets necessitate a novel approach that combines high-resolution imaging with advanced, targeted neural network architectures.

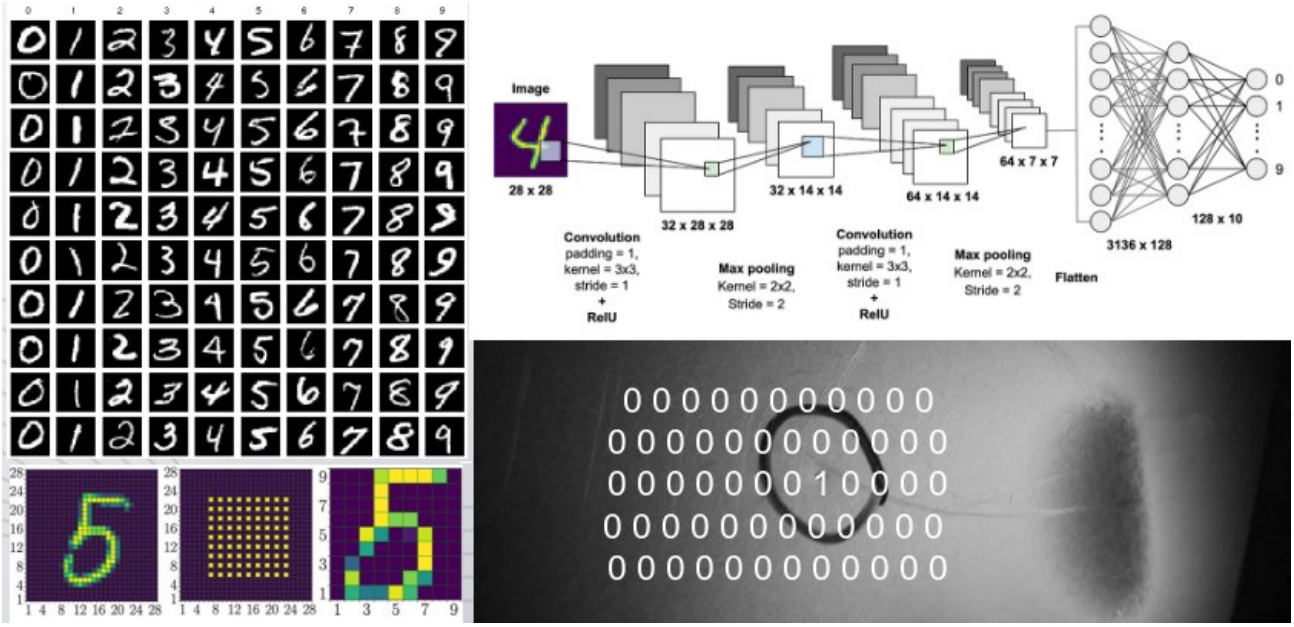


Figure 2.6. The problem of defect detection on metal sheets can be considered closer to the digit recognition problem. Each local neighbourhood of pixels needs to be independently classified.

More specifically, for polarization techniques, the repository https://github.com/tkuri/Awesome-Polarization offers an extensive list of the papers and combining deep-learning approaches with cameras that can distinguish light polarization. Works like [Ding21] focusing on defect detection on composite laminates, and multiple works such as [Ba20, Desc21, and Lei22] also managing to extract 3D shape (Shape-from-Polarization SfP) from this input. The state of the art is [Mugli23] that combines an event camera with polarized lights for SfP. Event cameras allow for incredibly fast acquisition framerates in Kilo Hertz speeds. This means that even a very zoomed lens that covers a very limited area becomes viable since there are fewer limitations in terms of data processing. We considered such an approach, only "recording events" when there are polarization discrepancies between the observed and an ideal surface. This could allow a solution with a polarized light, event driven technique. However, unfortunately, event cameras are still not mature and such a camera system choice would entail many uncertainties, forcing us to select a regular global shutter system where the neural network will always have all the frame information to ensure the correct operation of the final system.

2.3 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 1.1) 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 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%

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

2.4 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 with each shot and each individual pixel features one of four different linear polarization filters which enables four polarization images at 0°, 45°, 90° and 135° to be captured simultaneously.

The acquisition framerate is 21 Hz, yielding a total of 105.283.584 measurements per second. The camera has a C-Mount lens. After trying different lens combinations, we decided to base the experimental setup on a ½" 12mm / F 1.4 lens that is in focus when positioned at approximately 30 cm from the observed object. The shutter speed of the sensor can range from 60 to 1/100,000 seconds. Given a 600 lumens light source, the minimum exposure time for a clear image is 3000 microseconds which is equivalent to a 333.3 Hz acquisition rate. This is important since, despite sampling the sensor at a rate of 21Hz, the visual fidelity of each of the recovered frames will not suffer from motion blur, vibrations that may originate from the robot arm, or other similar problems.

The 2448x2048 retrieved image that features all 4 polarizations can observe an area of approximately 14.8 cm x 12.4 cm or 183.52 cm^2 or 18352 mm^2 (Figure 2.7). Given that the largest car dimensions of the Stellantis LCV platform (see Section 1.1.2.1 of D2.1) are 4826,39 mm length, 1887,45 mm width and 1681,52 mm height we can proceed to calculate the theoretical time to scan this car platform.

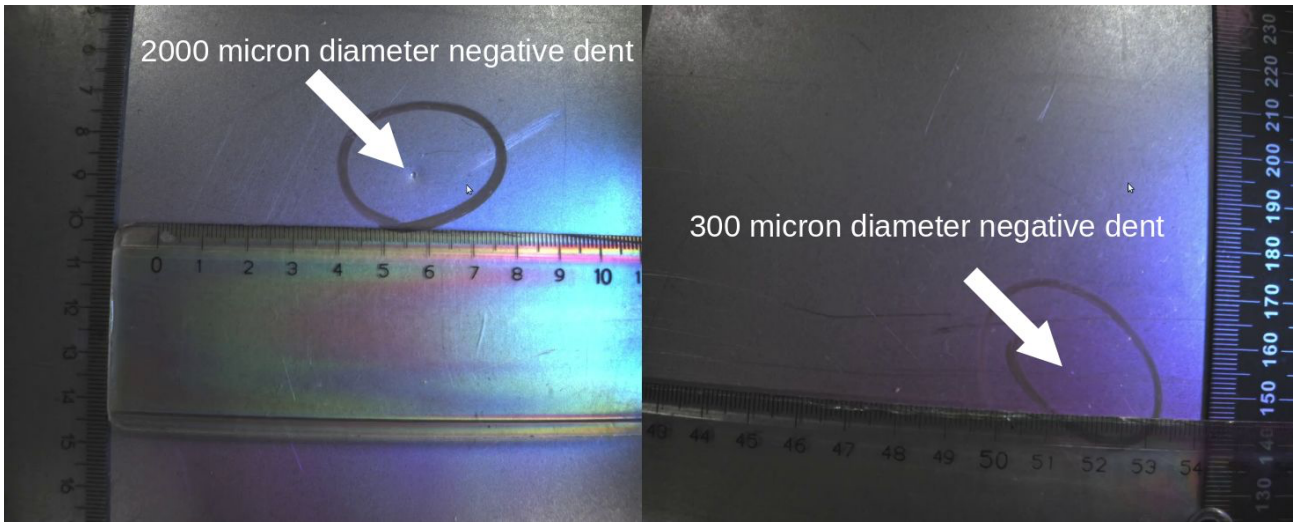


Figure 2.7. Left: A very pronounced negative dent with a diameter of 2 mm. Right: a negative dent with a 0.3 mm diameter (matching the 300micron KPI). The camera system described uses a 12 mm lens to optimize for the expected minimum defect size while accommodating the largest field of view that minimizes scan time.

We can assume that the scanning area of the car is a 3D rectangular prism without its bottom side. Its surface area can be calculated using the formula:

Surface = 2 x (length x height + width x height) + length x width = 31,688,482.27 mm²

Using the proposed camera system acquiring 21 frames per second and positioned by the robot arm, to record consecutive areas of the car surface and the field of view of 18,352 mm², the theoretical minimum number of frames required to scan the whole 3D solid would be approximately 1726 frames which would, in turn, take 82,22 seconds.

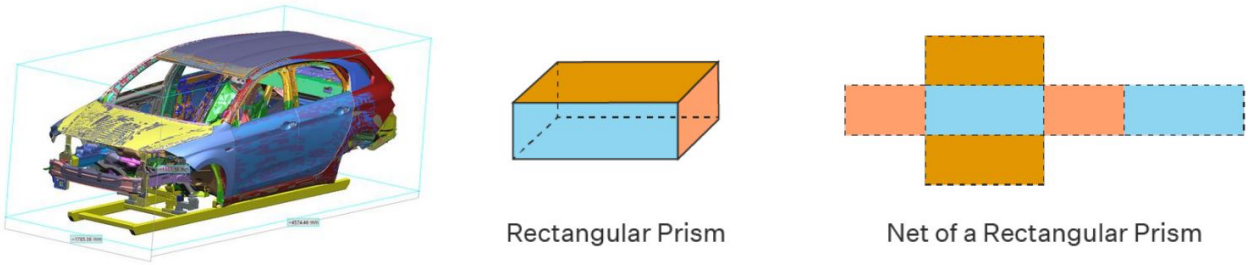


Figure 2.8. Considering a 3D rectangular prism, we can approximate the theoretical time to scan a car.

In reality (as seen in Figure 2.8) the car scanning area is much less since the windows, wheels and especially the frontal view of the car is not populated. On the other hand, given the robot platform's physical capabilities we expect to have a slight overlap between subsequent images, thus also reducing the effective area scanned. Although the current camera system in theory seems to be close to the requirements of the project, a viable strategy to improve the scanning speed would be to involve more than one camera in the sensing robot arm. In this case a slightly higher zoom lens (for example 13 mm) could be preferable to ensure even better fidelity images and better handling of very small defects (Figure 2.22).

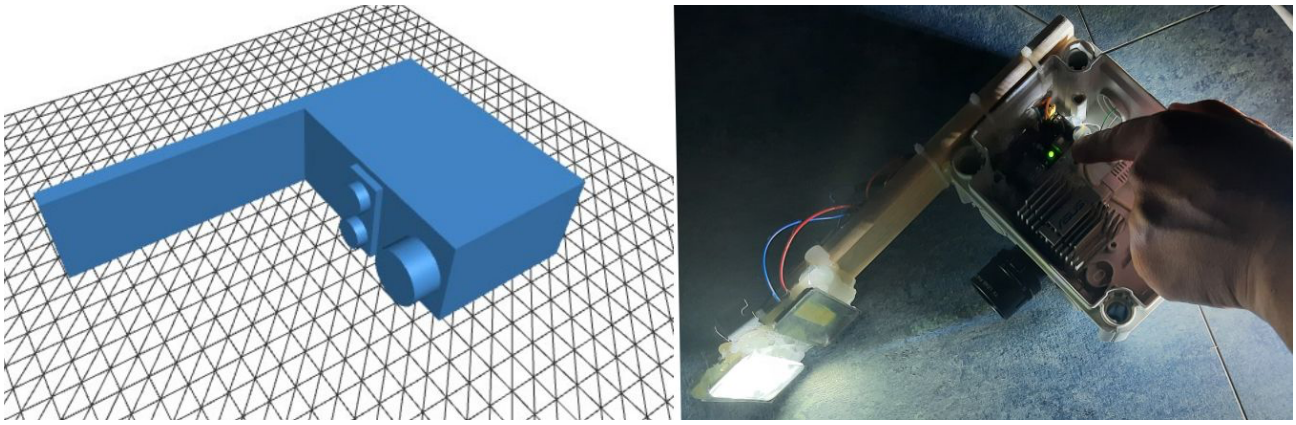


Figure 2.9. Left: A CAD concept of the initial camera system design. Right: The actual prototype camera system developed that also incorporates two light sources of 600 lumens with experimentally fine-tuned topology for better illumination.

The camera system produces a stream of images encoded using the GigE Vision open standard. SONY provides an SDK for the camera that supports Degree of Linear Polarization (DoPL), polarization direction (Surface Normals) and stress and distortion (retardation) extraction. The SDK, however, is only available for Windows using the Visual Studio 2015 and 2017 platforms and has licensing limitations binding each copy of the software to specific hardware installations.

```
resolution                = (1232, 1028)  # resolution of the sensor (Polar sensor needs 4 readings so is actually half size !)
pixel_size_microns        = 3.45  # physical size of each pixel in microns
diagonal_size_mm          = 11.1  # diagonal size of the sensor in millimeters
defect_size               = 300  # defect size in microns
light_angle               = 30  # angle of the light in degrees
focal_length              = 12000  # 12mm focal length
object_distance           = 30000  # Distance of object from camera lens
distance_to_plane         = object_distance + focal_length  # distance from the camera sensor to the plane in millimeters
focal_length_mm           = focal_length / 1000  # convert focal length to millimeters
exposure_time_microseconds = 5000 # 5000 microseconds for each frame (Used later for vibration calculations)
framerate_hz              = 23   # Transport bandwidth for each frame (Gig.Eth)
framerate_microseconds    = 1000000 * (1/framerate_hz)
target_time_microseconds  = 60 * 1000000 # 60sec target time
cost_per_sensor_euros     = 500 #500 Euros
surface_size_mm           = 1000  # 1m = 1000mm, surface size in millimeters

percentage_coverage, num_tiles_width, num_tiles_height, field_of_view = calculate_defect_coverage_for_a_given_camera_system(resolution, pixel_size_microns, diagonal_size_mm, defect_size, light_angle, distance_to_plane,
    focal_length_mm, surface_size_mm)

print(f"The defect + shadow at {light_angle} degrees covers approximately {percentage_coverage:.2f}% of the sensor.")
print(f"To cover the surface, you need {num_tiles_width} tiles in width and {num_tiles_height} tiles in height.")

totalTimeMicroSec = (num_tiles_width*num_tiles_height * framerate_microseconds)
print(f"To cover the ",surface_size_mm/1000,"m² surface, we need ", totalTimeMicroSec/1000000," sec , fov is ",field_of_view )
print(f"We are using an exposure time that is ",100*exposure_time_microseconds/framerate_microseconds," % of the camera transport bandwidth" )

numberOfSensorsNeeded = math.ceil(totalTimeMicroSec / target_time_microseconds)
print(f"We need ",numberOfSensorsNeeded," cameras that cost a total of ",numberOfSensorsNeeded*cost_per_sensor_euros," euros to solve problem  in ",target_time_microseconds/1000000," seconds")
```

Figure 2.10. To decide on the parameters of the camera system we create a utility that performs optics calculations and, given the available KPI constraints select a camera/lens configuration that best satisfies our accuracy requirements while also allowing for the fastest scan time possible. We experimentally validate our calculations using different camera/lens/lighting combinations as seen in Figure 2.3.

2.5 DATA ACQUISITION AND ANNOTATION

2.5.1 DATA ACQUISITION

To facilitate data acquisition, after briefly experimenting with the proprietary closed-source SDK of SONY, we opted to instead adopt the ARAVIS opensource video acquisition toolkit (https://github.com/AravisProject/aravis). After a brief initial troubleshooting period and patching the driver with the help of the ARAVIS developer

community to correctly acknowledge the SONY-XCG-CP510 sensor (https://github.com/AravisProject/aravis/issues/836), the ARAVIS SDK successfully and consistently acts as a dependable transport layer for the image streams of the camera. It implements both the GigE and USB3 protocols used by industrial cameras. Being open source allows integration of the Camera System with any operating system or PC software. At the same time, it allows low-level interception of the image immediately after receiving it from the network interface, thus being very efficient and not adding any computational overhead. ARAVIS is written in C and is application-agnostic. When used in conjunction with the Polarsense cameras, it provides the raw polarization readings and allows for configuration of all the exposed camera configuration parameters like exposure, black level, gain, buffer sizes etc. To extract Polarization metrics like Degree of Linear Polarization (DoLP), Angle of Linear Polarization (AoLP), perform demosaicing, analysis of stokes vectors and Mueller Matrix computations, all of which are analysis techniques for polarization image processing, we use the polanalyser package (https://github.com/elerac/polanalyser). However, these tools are mostly used for preview of the received polarized light from different directions of the light compared to the observed surfaces and the camera. As we will further elaborate in Section 2.6 where we describe the employed methodology, our goal is to provide the Neural Network classifier with the raw data from the sensor for each polarization orientation. Thus, the ARAVIS SDK fits this task perfectly with minimal overheads and allows transparency in the image transport layers due to its open-source implementation. This may prove useful in tackling problems that might arise from the deployed network topology.

To integrate ARAVIS with the developed classification prototype, a streaming utility was developed[1], that utilizes shared memory video buffers[2] based on the mmap mechanism of POSIX-compliant UNIX systems. The streamer executable is linked against the ARAVIS SDK that processes incoming UDP packets in the background and fills the last available incoming image. Upon receiving the signal for a completely received and available image, the streamer executable directly copies and maps it to a video buffer on system RAM. This mmaped image is also protected with a mutex for correctly defined access to the image, preventing potential over-writing from the next frame while it is being consumed. The python classifier is in turn using the Shared Memory Video Buffer python wrapper[3] and can map the same memory area and expose it to the python runtime as a numpy object that is immediately compatible with all deep learning frameworks and libraries. The data acquisition pipeline is thus very lean in terms of lines of code, it relies on completely open-source components and leverages existing kernel mechanisms like

[1] (https://github.com/AmmarkoV/aravis-c-examples/blob/main/07-streamer.c
[2] https://github.com/AmmarkoV/SharedMemoryVideoBuffers
[3] (https://github.com/AmmarkoV/SharedMemoryVideoBuffers/blob/main/src/python/client_downstream.py)

mmap. In total, the data from the camera get copied once to fill the buffer in the ARAVIS layer; after being assembled they are copied once more on the mmaped memory that is common for the streaming executable and python classifier, and they are copied one last time to the GPU in order perform hardware accelerated NN inference to achieve a fast framerate.

Streaming the acquired data to the real-time classifier will be a very important task for the deployed robot and the above-described architecture has been designed from scratch to handle this task in the best way possible. A much easier task than production grade real-time image transport is to collect acquired data on disk to create offline training datasets for the method. To perform this task a separate executable "grabber"[4] was developed that encodes incoming images as portable anymap files (PNM or commonly called netpbm). Data Acquisition sessions are typically recorded at 10 Hz with a 3000-microsecond exposure time to contain crisp and spatially variable snapshots. Recording sessions are limited to 1500 recorded frames which take two and a half minutes to record and approximately 1 hour to annotate using the GUI annotation tool we have developed.

A Data acquisition session starts with the following command:

./06-grabber --size 2448 2048 --exposure 3000 --fps 10 --maxFrames 1500 -o ***name***

This yields a **name**/info.json file containing the capture parameters and 1500 files with paths **name**/colorFrame_0_xxxxx.pnm . The PNM is a lossless format so there is no degradation (such as lossy compression) of the received data, with each image occupying 4.8MB on disk. After the files are stored on disk, a secondary script is automatically executed, converting the raw data to AoLP, AoLP_s, AoLP_v, and DoLP formats for visualization purposes. After this operation is complete, the data is compressed in an archive of 2.3GB for each dataset.

2.5.2 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

[4] https://github.com/AmmarkoV/aravis-c-examples/blob/main/06-grabber.c

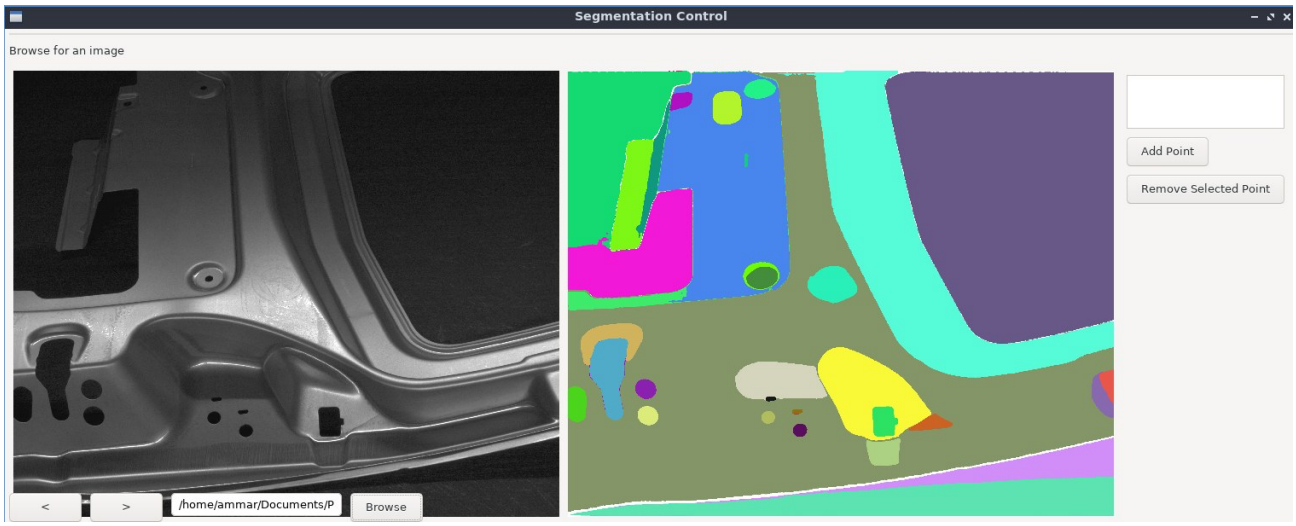


Figure 2.11. Combining the polarization channels in a single image yields a monochrome frame. Using the Segment Anything (https://github.com/facebookresearch/segment-anything) foundation model we can successfully identify and annotate different parts of the metal assembly of the car. This can help with tracking annotated defects from one frame to the next, however this technique is only viable when using lens with a broad field of view, that unfortunately do not allow enough resolution for the challenging defects we want to tackle.

the captured frames. The tool has integration with the Segment Anything (SAM) foundation model [Kiri23] (as seen in Figure 2.11). This was initially considered in the hope that it could potentially help automate the annotation effort. Unfortunately, SAM is not capable of segmenting defects on the metal sheets using the zoomed lens. When used on a camera system with lens with a wider field of view, it can produce meaningful segmentation. However, this segmentation is not useful since, at this scale, defects become imperceivable due to their very small size.

The tool automatically generates a .json file for each .pnm image in the same directory. It contains the MD5 hash of the image to guard against potential storage corruption that might subsequently affect the training effort as well as the dimensions of the image, the defects present on the image and their type and classification. Defects can be added to the image by selecting a defect type from the drop-down menu on the right and then clicking on the left or right images (Figure 2.11). The arrow keys of the keyboard or the buttons and seek bar of the GUI can be used to progress to the next frame. By clicking on an already existing defect, it can be selected and have its type changed. Furthermore, if the annotator mistakenly adds a defect, it is easy to remove it using the delete button or the appropriate GUI button. Using the combined acquisition/annotation toolkit described a total of 28 datasets of 1500 frames each have been recorded, for a total of 42.000 frames. However, it is worth noting that physical changes on the camera, polarization orientation or light position and rotation invalidate previously recorded data since the intrinsic configuration of the experimental setup diverges.

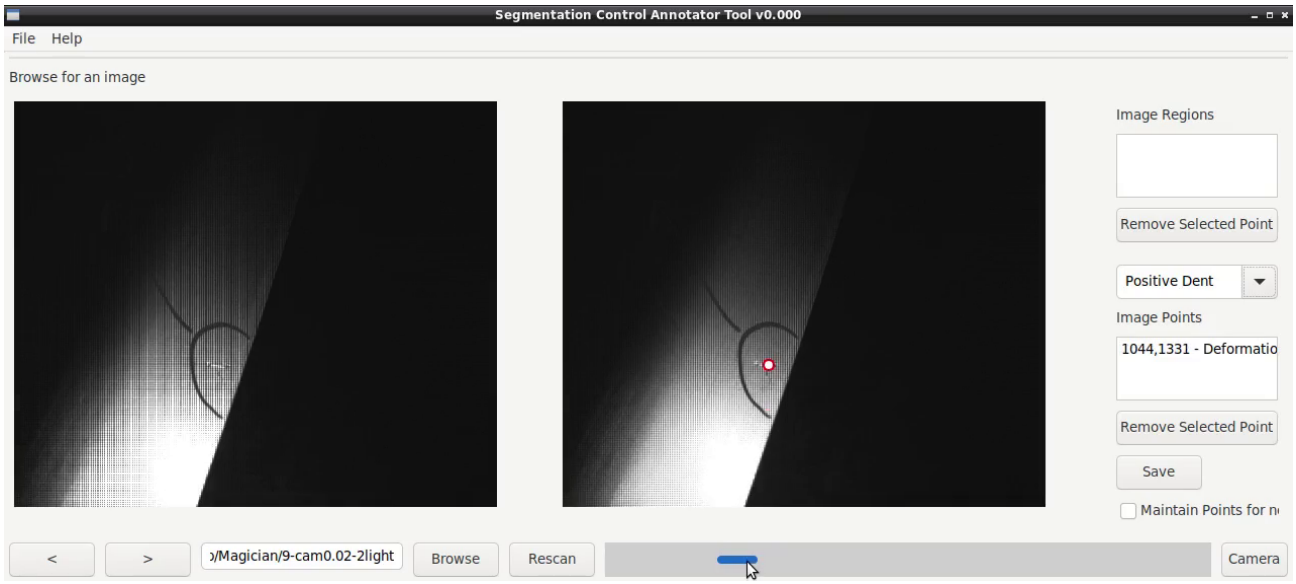


Figure 2.12. The annotation tool graphical user interface while processing a recorded dataset.

The grabber and annotation tools developed streamline dataset acquisition and preparation and provide a high-quality solution to these problems. The lossless recording ensures no image degradation after acquisition from the sensor, while MD5 hashes provide passive protection from data corruption in storage that might silently affect and degrade training. The grabber utility can execute recording using a single command, while the annotation tool requires a few clicks and one keystroke per frame which only take a few seconds for the user. That said, due to the volume of data, this is still a very time-consuming and tedious task that needs to be manually performed with care to avoid mistakes.

Once the camera system and classification neural network is finalized, efforts will be made to integrate them in the annotation tool to speed up and automate the annotation process. Samples that can be correctly automatically annotated by an existing classifier network are already covered by the existing training, so in essence we expect such automation to provide a moderate speed up in the annotation procedure. This in turn will allow only the most interesting new data not to be automatically annotated and thus requiring manual annotation.

2.6 METHODOLOGIES EMPLOYED

Developing the prototype system presented here involved usage and testing of various methodologies. After the beginning of the project, and before having any available metal sheet samples, we began preparation work by experimenting with simulated 3D rendered defects. Since most rendering ray-tracing engines do not account for the polarization of light waves, but just for its intensity, finding a suitable engine for the task was difficult. We were finally able to simulate a hyperspectral rendering pipeline using the Mitsuba 3 3D renderer (Figure 2.13). Despite being able to programmatically arrange

the camera, lens and light in novel ways through software, after receiving the first metal samples it quickly became apparent that the “ideal” surfaces of the 3D models, even when textured with metal gradients had a significant gap from actual observations. We abandoned this methodology in favour of collected samples from the developed camera system. This allowed to eliminate the synthetic image discrepancies from actual data as an accuracy factor, a problem commonly referred to as domain gap.

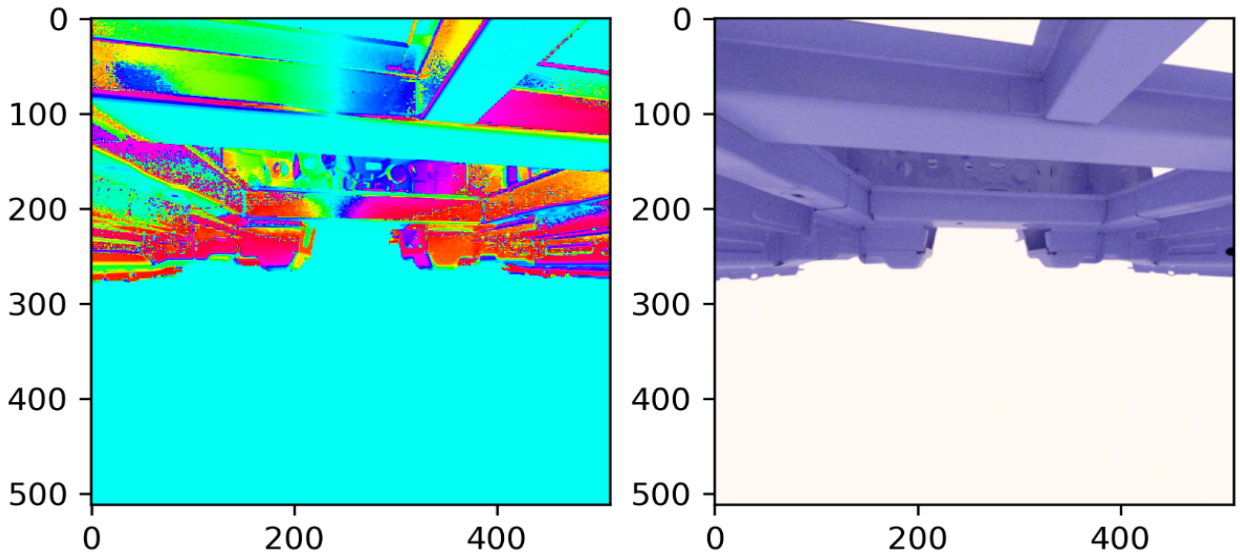


Figure 2.13. Simulating rendering a car chassis with polarization enabled 3D rendering in Mistuba 3.

Another method used to further understand and tackle the problem was the use of a commercial industrial metrology 3D scanner, namely the Artec Space Spider (Figure 2.14). This device uses a structured light system with 5 cameras, a 6 LED array with 3D point accuracy up to 500 microns, a framerate up to 7.5 fps and a working area of up to 900cm^3. With this scanner providing 3D geometry on a very fine level, the task of defect detection could be transformed to plane fitting of the CAD parts compared to the observed 3D structure. Unfortunately, this is not possible for multiple reasons, the first of which is that despite the very high detail of the sensor, it is still not enough for the very demanding target accuracy, as specified in the relevant KPIs of the project. A second issue we can identify is the following: going through the steps of 3D reconstruction involves significant computational overhead for the localization and mapping of the different volumes that are not contributing anything towards the actual problem we are attempting to solve. Making matters worse, the size of the observed defects is so small that the actual defects may be drown in the noise of the cumulative discrepancies of the 3D model compared to the 3D scanned surface, making it very hard to detect them through 3D fitting. The reconstructed surface contains useful 3D and texture features that be combined with the 3D rendering techniques to be used for a synthetic training approach, however this still generates a domain gap, compared to data recorded from the actual sensor.

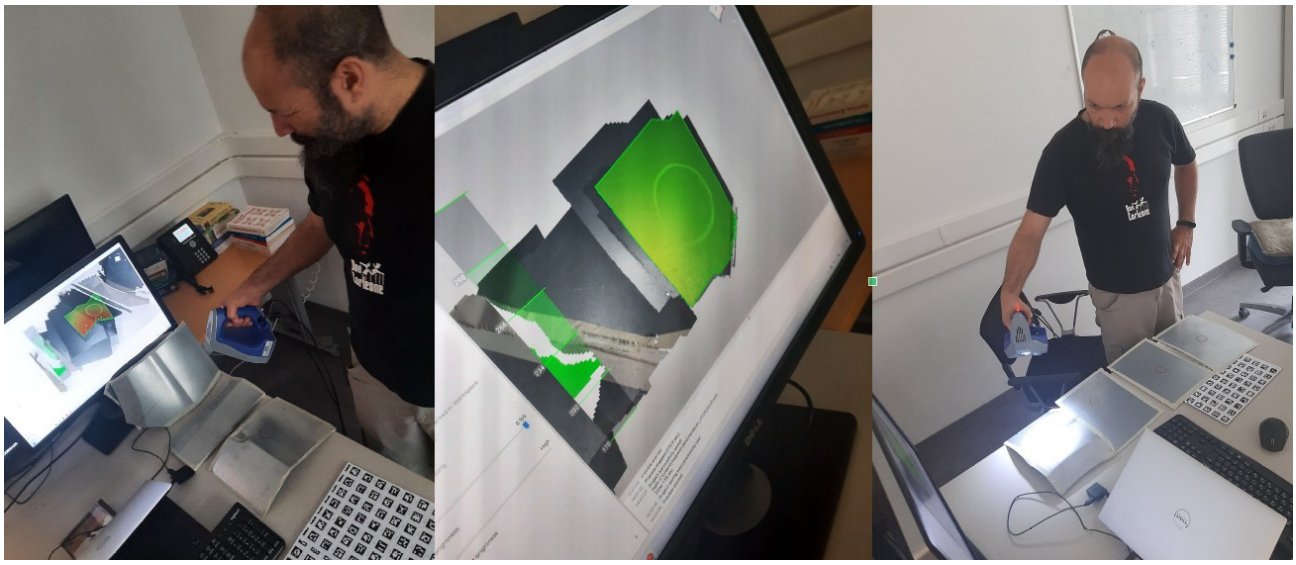


Figure 2.14. Artec Space Spider industrial 3D scanner benchmarks on the defect sensing task.

We experimented with off-the-shelf RGB sensors as a readily available methodology for image capture. Metal sheets have no colour, and thus the RGB information of a colour camera is only useful in the case of sealing material residuals, due to their characteristic sealant dark-blue colour. Introducing a controllable RGB light source (Figure 2.15) to the experimental setup however we can use the three wavelength bands (corresponding to red, green and blue) to excite the different channels of each coloured pixel. This way we can effectively record three different measurements from three different physical origins on the sensor.

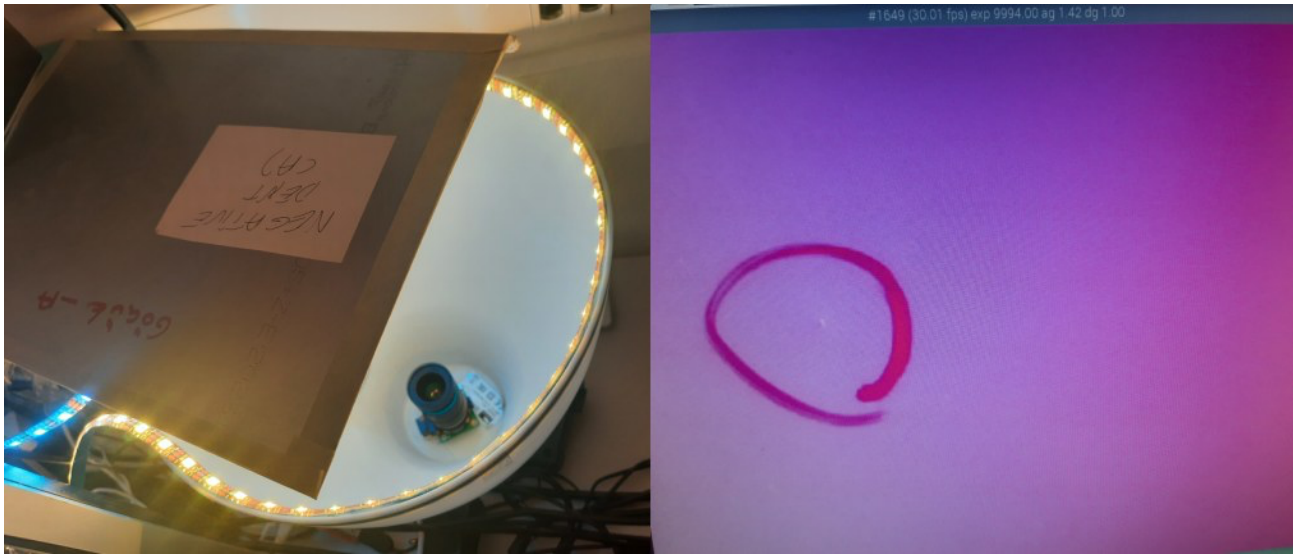


Figure 2.15. Early experiments using global shutter RGB sensor with 13mm lens and addressable RGB LEDs.

Although this approach is very versatile since it does not require a sophisticated and expensive sensor, the main problem given the low exposure time we are targeting (3000 μs), this setup results in underexposed images, unless using unreasonably bright light

sources, which is undesirable due to other constraints, such as power draw.

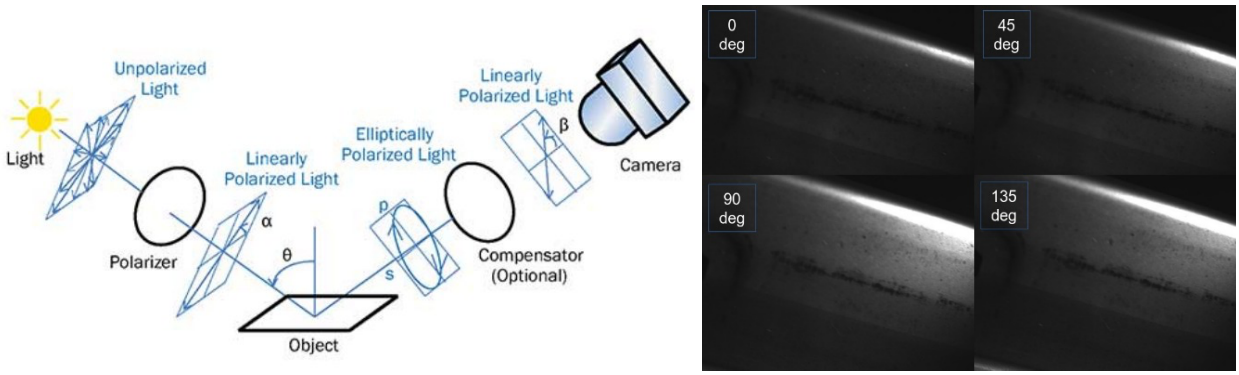


Figure 2.16. Left: Using polarized light, we can estimate the light angle of attack θ for every pixel recorded by our camera. Right: The camera records 4 polarization orientations that provide fine grained texture on the metal surface.

As hinted by our camera specification, the best performing methodology of those we attempted to employ involved the usage of cameras that can sense different light polarizations and thus provide a composite of four readings per pixel (Figure 2.16), yielding a rich source of information. Using polarized light enables us to have a much stronger light source that is slightly dimmed through a polarizing filter. The polarization filter absorbs the other light orientations thus dimming the overall light reaching the surface. However, using a 600 lumens LED light we can have very good observations even at extremely low exposure times such as the target 3000 μs.

We base our classifier neural network on the Keras 3.0 framework, using the Tensorflow 2.16.0 back-end that at the time of writing is the latest stable version. Keras is compatible with Tensorflow, PyTorch and JAX backends, thus providing high flexibility for the developed NN methodology. We have also developed bindings for ONNX porting of the network, thus making it compatible with all major NN ecosystems.

The neural network methodology employed is a Conv-NET architecture with input tiles of 64x64x4 size, four convolutional layers featuring dropout, max-pooling and batch normalization. These in turn are followed by two densely connected layers, following the principles of the architecture shown in Figure 2.6. It is worth noting that despite the neural network being relatively small, having a larger network can quickly lead to overfitting in our problem. Furthermore, the incoming image is split in batches and the neural network needs to be executed thousands of times to cover each incoming image. The XCG-CP510 sensor yields 2448x2048 images that consist of 4 polarizations of 1224x1024 pixels each. Tiling them in 64x64 tiles yields 19x16 tiles that at minimum we need to execute the neural network 304 times per frame to cover the whole image. Depending on the available hardware and given the 21Hz rate of incoming images, this means that the size of the network must also be constrained by our computational capacity to ensure timely classification. We perform training for 10 epochs using a batch size of 16, a learning rate of 10^{-4} using the ADAM optimizer, a categorical cross entropy loss, and utilizing early stopping. We perform training validation using either a 20%

validation split on the training data, or by using a different selection of the recorded samples for training and validation. Due to the low number of metal samples however, this strategy might not be optimal for the future, and after finalizing the neural network architecture through experiments with an increased validation set, until more samples are made available, it will be important to use the whole dataset for training extracting the best possible performing model from the available data.

A final methodological consideration employed is balancing sample classes before training. As one can easily understand, the metal samples available to us (see Figure 2.1, 2.2) have a disproportionately high ratio of clean surface compared to defects. This is useful because clean metal sheets are also important for training, serving as the negative class ("no defect present"). However, assuming that for each 100 recorded tiles, only one exhibits a defect, naively training a network directly on all these tiles entails the danger of overfitting, always classifying a surface as "clean" and achieving "99% accuracy" when naively measuring the accuracy on the task. This is the well-known problem of class imbalance, commonly encountered in classification tasks. To tackle this problem, during conversion from the annotated input images to the training tiles there is probabilistic filtering to ensure that there is no bias in the number of samples for a specific class. Furthermore, over-exposed areas (where all three channels are saturated), or where the standard deviation of intensities is below a certain threshold can also be omitted since they do not offer meaningful information and can be trivially excluded as defects even without the use of a Neural Network.

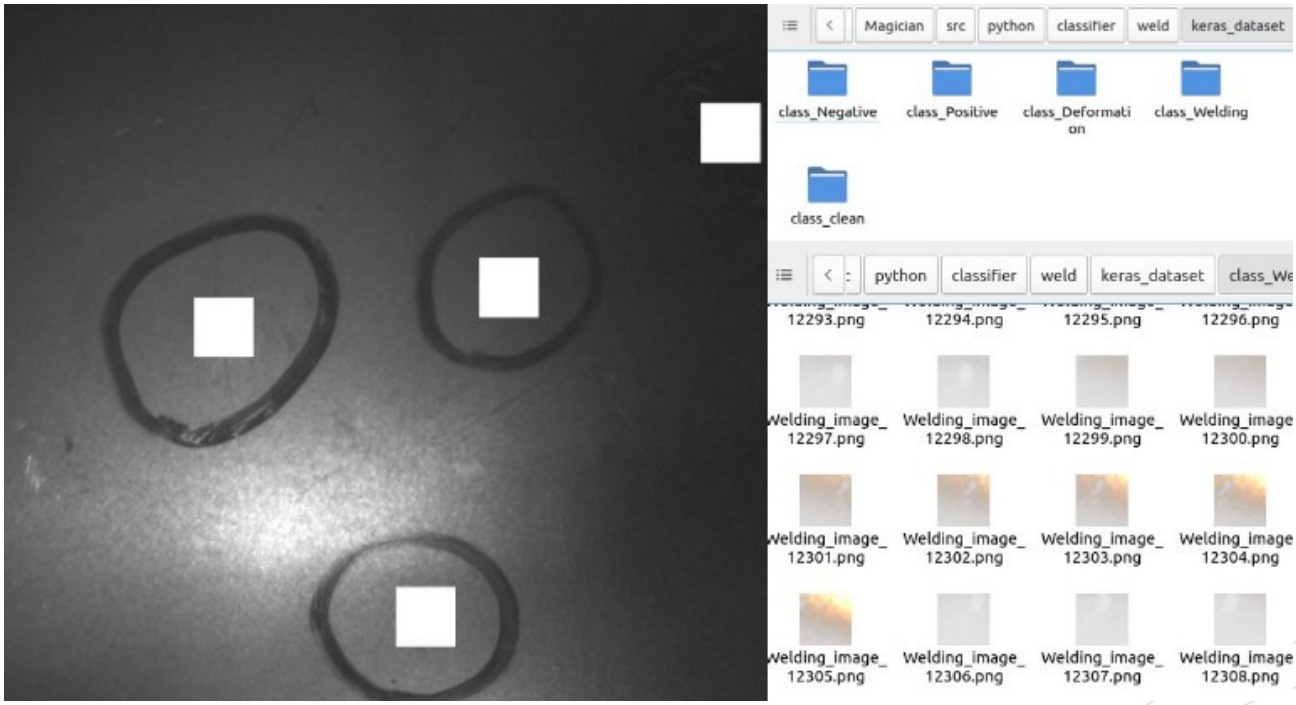


Figure 2.17. Left: Using a fixed tile size for defects we decouple the very visible markings from the actual defects, thus not allowing the NN to "cheat" by learning to detect the markers instead of the actual defects. This way also, despite a limited number of metal samples we can gather tens of thousands of samples required to thoroughly train a NN. Training data

is organized via keras.utils.image_dataset_from_directory conventions making it compatible with an existing code to handle and stream the data.

2.7 PRELIMINARY RESULTS AND FINDINGS

As described above, we have successfully built a working, proof-of-concept, Visual Perception for Imperfections Detection system. Although the system is still a prototype, it showcases the strengths of the employed sensors and techniques to tackle the problem. It uses light polarization as a primary imaging technique through the use of the SONY PolarSense sensor and a polarized light source (Figure. The selected lens has a 12 mm focal length, providing enough visual clarity to make the smallest possible defects have a 4x4 fingerprint in the 5MP sensor. The camera acquisition method is based on the opensource ARAVIS SDK, and we employ an optimized grabber that uses the mmap Linux Kernel mechanism to perform zero-copy transport of the image data to conserve resources. To annotate the datasets, we developed a GUI annotation tool and experimented with foundational models for segmentation. The neural network classifier uses a proven and well-studied Conv-NET architecture and is based on tiles, working around the low number of available samples, and also not overfitting to the marker annotations. We successfully train a network using 10K+ samples from each class (Figure 2.17) and have a preliminary system that in relatively unoptimized python code can perform classification at rates ranging from 4Hz to 14Hz depending on the density of tiling on the image, as seen in Figure 2.19.

As already mentioned, the currently developed sensor prototype is still just a proof of concept. We nevertheless performed a rigorous preliminary quantitative analysis of the developed system. We perform a validation/training split using 20% of recorded data using the sensor described (Figures 2.9 and 2.16). In the available recorded data, we have overwhelmingly more positive and negative dent training samples, followed by welding spots, then material deformations and finally only 3 samples for sealing residuals (Figure 2.2). Therefore, to have a meaningful analysis, we perform the test on 3 classes, namely positive/negative/clean surfaces. This ensures the results reflect the model's accuracy and are not influenced by the number of available samples. As seen in Figure 2.18, after performing training for 200 epochs using a batch size of 16, a learning rate of 10e-4, using model checkpointing, and early stopping watch that triggers on epoch 197, the model achieves a 98.35% training accuracy with validation accuracy at 84.29% for the 2 fundamental dent + non-defective classes. We expect improving the optics of the system (increasing the camera lens zoom) to make defects more pronounced and thereby improve these results. We will also use an active vision system with dynamic illumination to also improve errors that are accumulated due to over / under exposed areas of the image. Moreover, having a larger number of samples and richer training corpus will help bring the validation curve of Figure 2.18 closer to the training score in the same graph.

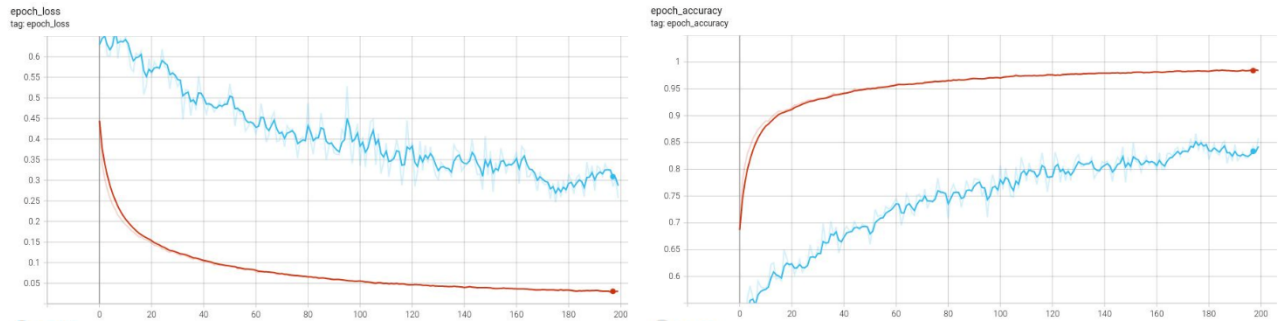


Figure 2.18. Training summary for 200 epochs using a learning rate of 10e-4 for classifying Positive/Negative dents and non-defective areas. After collecting the dataset, we perform a 0.2 validation split. Left: Plot for training loss (orange) and validation (cyan) sets. The model achieves 98.35% accuracy on training data with 0.0303 loss and 84.29% accuracy on validation data with 0.3089 loss achieved after the 197th epoch.

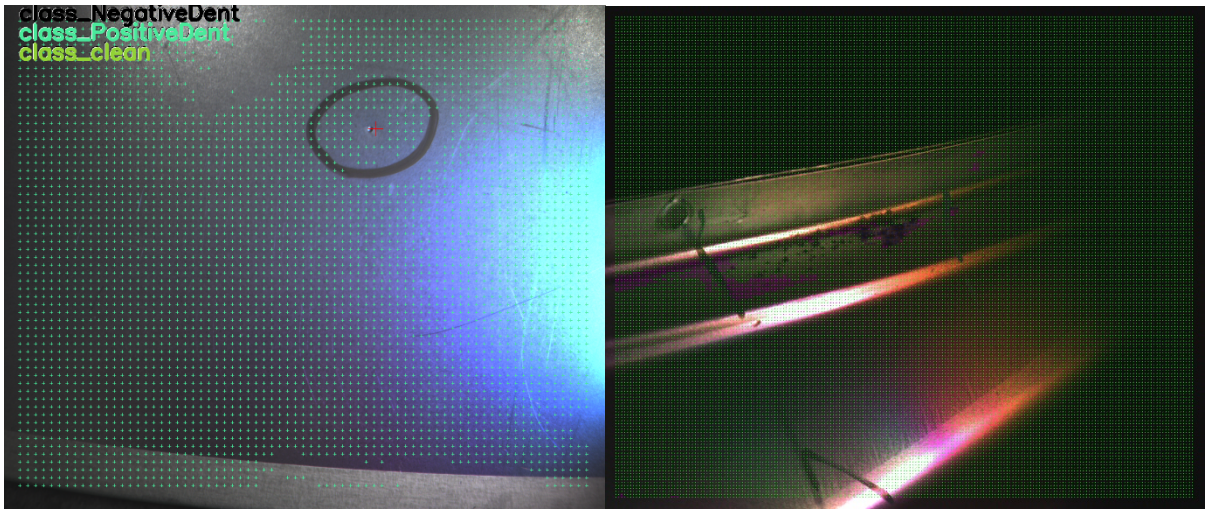


Figure 2.19. Using the methodology proposed we have a proof-of-concept system that can perform rudimentary classification. The proposed architecture is very flexible and depending on the tile size, density of batching on input images and available computational resources can be tailored to less dense (Left) or even per pixel classification results (Right).

Qualitative classification results can be seen in Figures 2.20 and 2.21, where with green crosses the model highlights areas that are considered non defective. Areas without a green cross are suspect candidates of a possible defect, and areas with a red cross are areas where the system has triggered a defect classification. Overall, the system seems to identify pronounced defect cases. This experimental finding is also very promising since by increasing the optic zoom of the area we can make even smaller defects more pronounced thus showing as a viable way to improve accuracy.

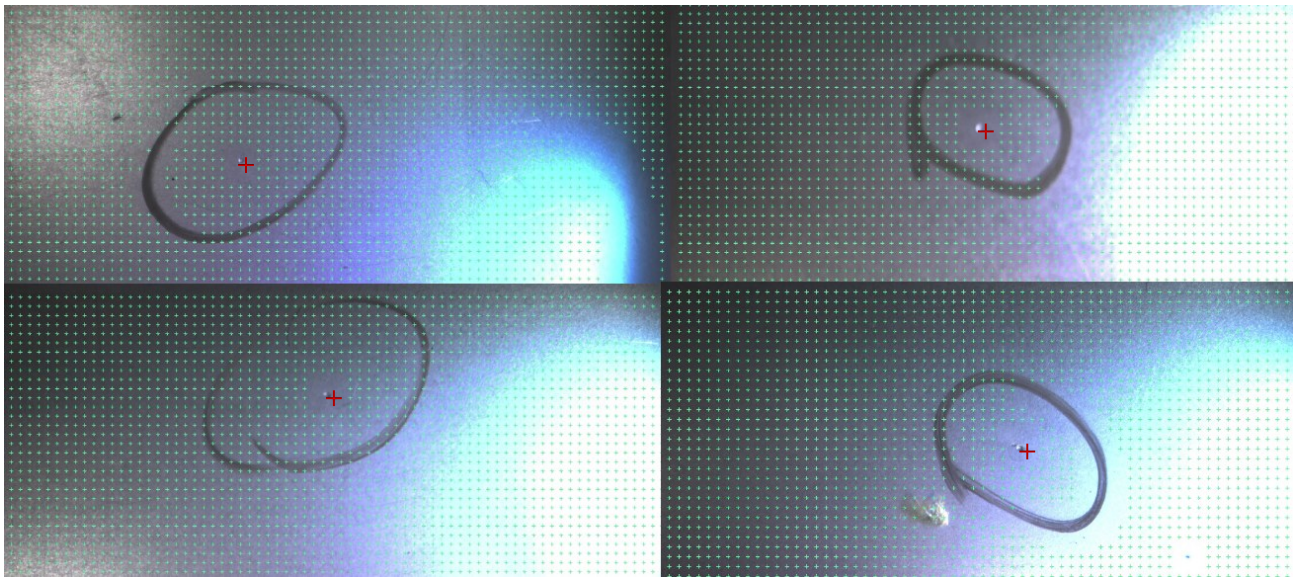


Figure 2.20. Various samples are currently correctly handled by the developed system when running the camera against an incoming stream of images from the sensor and performing classification on the fly.

Failure cases of the system are constituted by out of focus images, since the camera is currently manually operated and thus often does not stay in the optimal distance range away from the metal surface making small dents disappear from the sensor. The current setup has a single light source direction that depending on the angle of the metal sheet and camera system might occlude lighting or overexpose parts of the image due to specular reflections to the detriment of accuracy. Finally, the texture of the metal surfaces is very rich and features scratches, dust, particulate matter and various other marks and features. We observe that sometimes the system may identify scratches as defects (Figure 2.21 bottom right). These are false positive classifications, since the experts do not consider scratches as problems since they get gracefully covered in paint without causing problems for consumers.

Figure 2.21. Failure cases during our experimental evaluation include very small dents close to our minimum KPI (Top Left), a combination of very small (<700 micron) dents when being more than 2 cm out of focus (Top Right/Bottom Left). Finally, large scratches and on the materials that are not labelled as a defect by our experts are sometimes detected (Bottom Right). Having only one light source constitutes a problem for the current system since depending on the angle of the camera in relation to the metal surface specular reflections may overexpose the parts of the image closest to the light source. A significant finding is that our tiling strategy successfully manages not to overfit the NN on the marker annotations and thus this makes the model correctly aligned to the task at hand.

2.8 CHALLENGES AND LIMITATIONS

As mentioned in the introduction of this section, the vision-based imperfections detection task is very challenging due to the large surface area that needs to be scanned, the very small size of defects, and the short time available to complete the scan. Given these constrains, we have developed a system that leverages light polarization and an optimized neural network to tackle this challenging problem. However, several challenges have been identified in the preliminary system:

1. **Hardware dependence**
    - The most prevalent challenge to this preliminary system we developed is that, in contrast to other systems, it also has a hardware aspect that can make or break the performance. We have to carefully design the prototype camera systems, and we cannot rely on off-the-shelf hardware, which would allow to focus our efforts on the software and Machine Learning challenges.
    - Making matters even more challenging, the employed visual systems in the context of the project should be designed and assembled with very small tolerances. Furthermore, many hardware details, such as the camera sensors, lens,

lighting devices, polarizers, capacitors, electronics must be decided and fixed early on, because deviations will cause degraded performance in the final classification, stemming from hardware factors that are very difficult to control. Changes to the camera system invalidate all recorded and annotated data, making this a wasteful process. Designing the correct camera system and finalizing it is thus a very important limitation and a challenge of paramount importance for the timely execution of the MAGICIAN Project.

2. **Illumination Challenges**
   - Although we have experimented with multiple light sources (Figure 2.12), the initial prototype (Figure 2.9) for practical reasons that have to do with electronics and physical construction simplicity only has polarized light coming from one direction. This however can be problematic with complex surfaces that might occlude light, causing it to stop shining the whole observed surface, and thus constraining the available data.
   - In such cases an active light system should illuminate the occluded area using a light from a different direction, thus correctly tackling the problem and utilizing the whole sensor area at all times. This fact might also prompt us to the revisit the addressable light scenarios, possibly emitting white light instead of RGB, for higher luminosity.
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.
   - Having a fixed focus lens that requires the camera to be positioned at around 30 cm above the observed surface effectively transfers the focus task to the robot motion planning control loop. As an extra measure we are considering adding a closed-loop passive distance sensing circuit to the camera head that will constantly provide feedback to the robot, ensuring correct focus. Our experiments show that even cheap ultrasonic sensors can provide good distance accuracy at refresh rates of 100Hz. We will also experiment with infrared and laser time of flight range sensors, that however may cause problems by emitting light that interferes with the camera sensor.
4. **Dataset Size and Deep Learning**
   - Despite TOFAS shipping us 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 bypasses this problem as well as the problem of learning the markings instead of the defects, however limitations on the

number of available samples lead to limited performance of the final classifier.

5. **Optics/Lens Selection**
    - We are currently basing our system on a 12 mm lens, to best satisfy the given KPIs (Figure 2.10) and balance scanning time and resolution. Defects however occupy a very small area on the sensor compared to non-defects, and thus a slightly more aggressive zoom lens might be preferrable to improve accuracy on defects close to the KPI limit of 300 microns. Essentially in this case we would be trading better observation resolution for longer overall scan time.
6. **Network and Hardware Integration**
    - The utilized camera system is based on Gigabit Ethernet and can also be powered over ethernet. This is very convenient since even a 20m CAT6 ethernet cable can be used, however it requires a gigabit switch and depending on where the camera will be plugged, network traffic on the switch might negatively affect image transport speeds.
    - This hardware limitation, along with the need to include GPU hardware acceleration for the neural network execution must be taken into consideration to prevent problems during integration of the camera system with the robot.

Overall, the development of the vision-based defect detection system involves navigating several key challenges. The precise design of hardware components, effective light management, and focus control are all critical to the system's performance. Additionally, the limitations in sample sizes for deep learning and the complexities of integrating network and hardware elements must be addressed as we move forward.

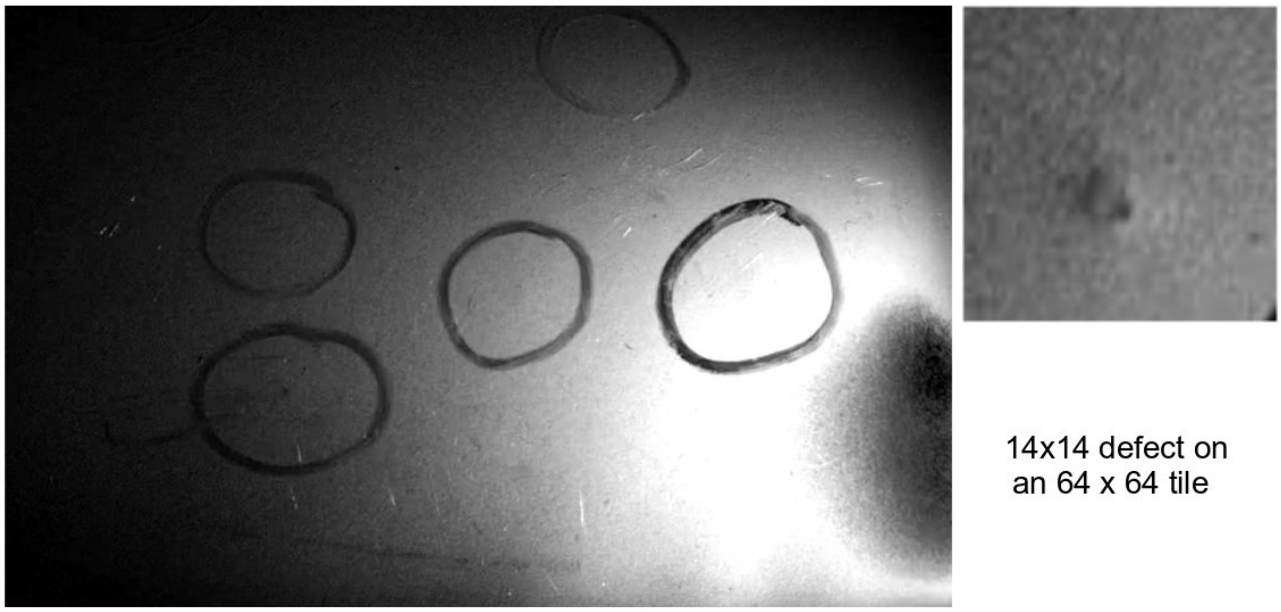


Figure 2.22. Left: A metal sheet seen using the SONY XCG-CP510 sensor, a 12mm lens and doing a Degree of Linear Polarization visualization. The sample features 5 negative dents highlighted with a marker by experts. We observe that

the metal has many visible abnormalities that however do not constitute defects since they are gracefully covered by paint. The system will need to be able to deal with such artifacts to suppress false positives. Right: Raw values from the sensor of a 64x64 pixel area of one polarization channel showing an actual defect up-close.

3 TACTILE PERCEPTION SYSTEM FOR IMPERFECTIONS DETECTION

3.1 INTRODUCTION

The tactile perception system represents within MAGICIAN the second essential component, alongside the vision system, in the detection of imperfections. The primary challenge of this module lies in emulating the sophisticated manual skills of human operators, who rely on their sense of touch to identify and differentiate between defects on the car's surface. Unlike visual inspection, tactile perception entails direct contact with the car body, which presents complex and often irregular surfaces, such as curved profiles or tight spots. These surfaces require the tactile system to have a high degree of flexibility and precision to detect imperfections that might otherwise go unnoticed. Additionally, the sensors chosen for this module must be capable of acquiring data that accurately reflect the interaction with the surface. This data acquisition needs to occur at a speed comparable to human scanning, with high accuracy and precision, and a high sampling frequency to facilitate real-time defect detection.

3.2 STATE OF THE ART

Recent research has demonstrated that acceleration and force signals can be effectively used to identify the material of a surface and detect potential defects. When a rigid tool is stroked over a surface, variations in applied force and scan velocity significantly influence the acquired signals, which are crucial for robust surface classification systems. To address the challenge of accounting for variable scan parameters, studies in automatic texture recognition typically rely on controlled exploration trajectories or predefined scan times, often achieved with the assistance of robots, to ensure consistent signal acquisition. When a human operator runs a rigid tool over the surface of an object, the force is applied, and the scanning speed usually varies throughout the exploration and between different sessions. These variations in scan-time parameters have a significant impact on the acceleration signals that are captured [Kuchenbecker2011], [Romano2012]. To overcome these challenges, reliable acceleration-based features have been modelled for distinguishing between different materials by mitigating the dependency on force and velocity [Strese2016]. Additionally, dynamic friction force, which also depends on scan-force and scan-velocity, provides another dimension of analysis that complements acceleration data. The combination of these signals, as acceleration, force, and friction, has proven effective not only in classifying surface

materials, but also in detecting surface defects, making them valuable tools in advanced material inspection and defect detection systems.

3.3 OBJECTIVES AND REQUIREMENTS

Hereafter we report the KPIs, as described in the Deliverable D2.1 - "Use Case Definition", related to the Tactile Perception System for Imperfections Detection. Similarly, and in parallel to the visual perception case, these KPIs (Table 3.1) define the constraints within which the developed tactile 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 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.

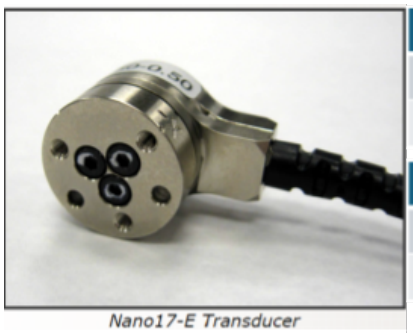
3.4 TACTILE SENSORS

To effectively replicate human tactile perception, it is crucial to mimic the function of mechanoreceptors, which play a key role during hand-surface interaction. This can be achieved by integrating both force and acceleration sensors into the tactile perception system developed by IIT within MAGICIAN (see Figure 3.2). These sensors work together to capture the complex dynamics of contact, allowing the system to detect and analyse the subtle variations in force and movement that are essential for identifying surface imperfections, much like human mechanoreceptors do. Force sensors are designed to detect variations in pressure applied to a surface, whereas accelerometers can measure oscillations or vibrations typically perceptible through human touch.

To acquire these data accurately, a force sensor, specifically the ATI Nano17, and an accelerometer, the ADXL335, have been employed (see Figure 3.1). The ATI Nano17 Network Force/Torque (Net F/T) sensor system is equipped with EtherNet/IP and CAN

bus communication interfaces, ensuring compatibility with standard Ethernet networks. Its web browser interface simplifies configuration and setup via Ethernet connection on all NetBox models. The option for a PROFINET interface adds flexibility. Additionally, the Net CAN OEM interface is designed for integration into small robot arms, offering CAN Bus and RS-485 serial interfaces for communication with a host computer.

The ADXL335 is a 3-axis accelerometer with signal conditioned voltage outputs. The sensor measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. Different bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis, which are suitable to mimic the human mechanoreceptor bandwidth since the latter respond to stimuli in a frequency range (~20 Hz to 1 kHz).



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. Preliminary studies conducted by IIT have shown that different patterns on the contact tip of the tactile perception module enhance detection by highlighting tactile features related to defects in car body parts. Moreover, multiple contact tip designs can be developed to meet specific needs; for example, a more precise tip could be used for accessing challenging geometries such as edges, while a broader contact area end-effector could maximize the scanned surface

and improve overall scan efficiency. The devices built to incorporate these sensors, along with the various types of end-effectors, are detailed in D4.1.

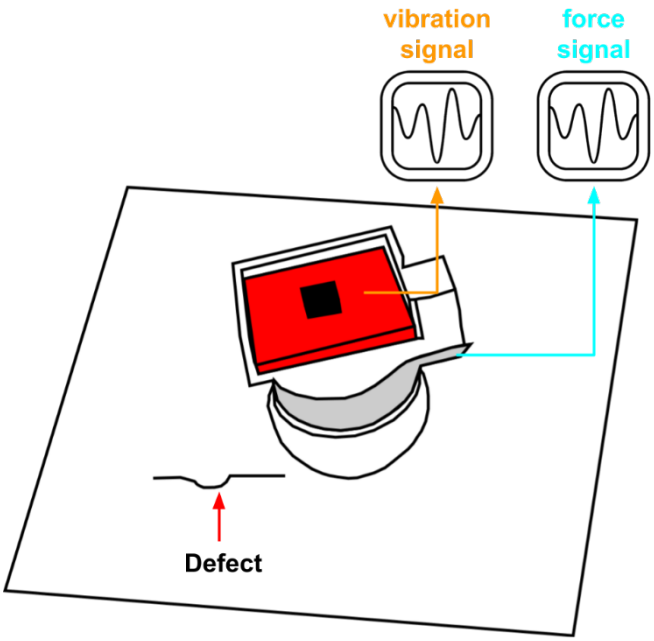


Figure 3.2. Schematic overview of the force and acceleration sensors mounting on the tactile sensor probe. The proximity to the probe ensures minimal signal attenuation during data acquisition. When the probe is in contact with the surface being scanned, the corresponding force and acceleration signals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

3.5 DATA ACQUISITION AND ANNOTATION

In the initial development stages, a software layer was created to facilitate data collection for offline analysis. The process involved using custom-made car bodies, specifically designed for this purpose, with defects comparable to those described by TOFAS to gather and label data for training and refining. The resulting dataset serves as a foundation for training, refining and fine-tuning the algorithms to identify and differentiate surface imperfections.

Multiple users were intentionally included in the data collection process to introduce variability in the acquired measurements. This variability is essential for building more robust and accurate machine learning models for defect classification, as one of the main challenges with tactile data is that it can vary depending on how each user performs the scan. By capturing data from different users, the dataset not only enhances model generalization but also allows for the identification of optimal scanning behaviours, ultimately improving the efficiency and consistency of future data acquisitions.

During a single acquisition, the user moves the tactile perception device across the

surface, ensuring that the sensing probe contacts the area of interest, whether it is a defect or not, depending on the aim of the acquisition (see Figure 3.3). The sensors embedded in the tactile perception device are placed to have their *x,y*-axis coplanar to the surface and their *z*-axis completing the right-handed coordinate system. There are no fixed constraints on the scanning method regarding speed, trajectory, or force. The starting and ending points of each scan may also differ between acquisitions, providing flexibility in the scanning process. The only system constraint is the saturation values of the sensors, which are, however, significantly higher than the force and acceleration typically used in this task (see Figure 3.1).

The recorded acceleration data have a sampling frequency of 4 kHz to ensure compatibility with the full sensor bandwidth, while the force data are sampled at a frequency of 7 kHz. Each acquisition is saved in three separate text files: one containing force values, another containing acceleration values, and a third file that is used to realign the two time series, accounting for the different sampling frequencies of the two sensors.

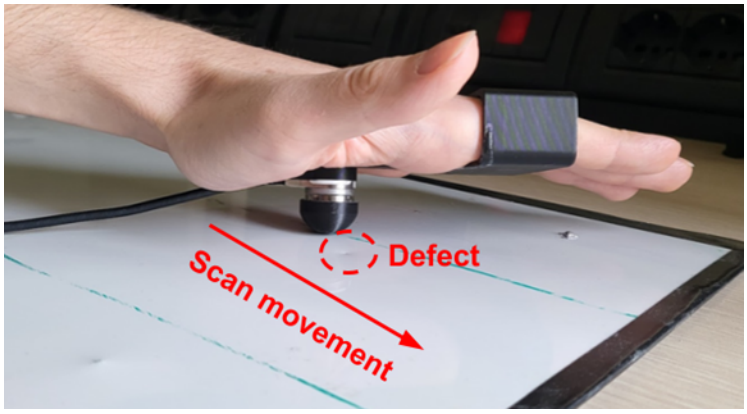


Figure 3.3. Example of a data acquisition process: the user moves the tactile sensor probe across the car part's surface while maintaining continuous contact with it.

Data are collected using a custom-developed LabVIEW user interface (see Figure 3.4). This software facilitates the labelling process by enabling the user to easily assign labels to the data being acquired. In particular, data are labelled and organized into a hierarchical tree folder structure based on three main factors: the user performing the acquisition, the type of defect being examined (including 'no defect' as an additional category), and the specific sensing probe used for scanning. Each acquisition is indexed with an incremental trial number. This systematic approach ensures that all data are well-organized and easily accessible for analysis.

After the tree folder structure is structured, a custom MATLAB script is used to post-process the data, formatting and saving them in a .csv file while adding the post-computed tactile features described in Section 3.4. The final directory structure and data formatting process are illustrated in Figure 3.5.

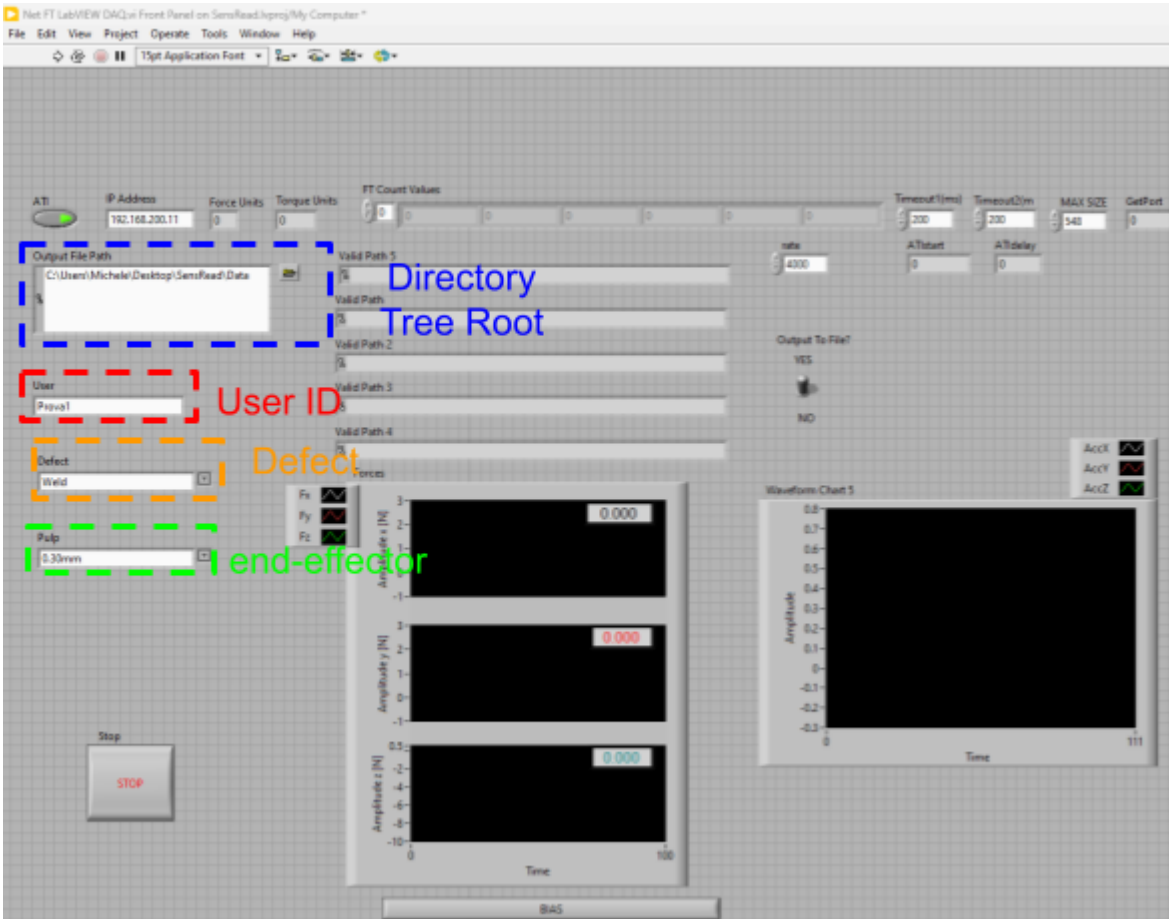


Figure 3.4. Before starting an acquisition, the user configures their ID, selects the type of surface defect, and specifies the sensor probe being used. Once these parameters are set, the user initiates the scan by moving the sensor probe in contact with the surface. After completing the exploration, the software can be stopped. The user may then either adjust the parameters for new conditions or retain the current settings for additional trials.

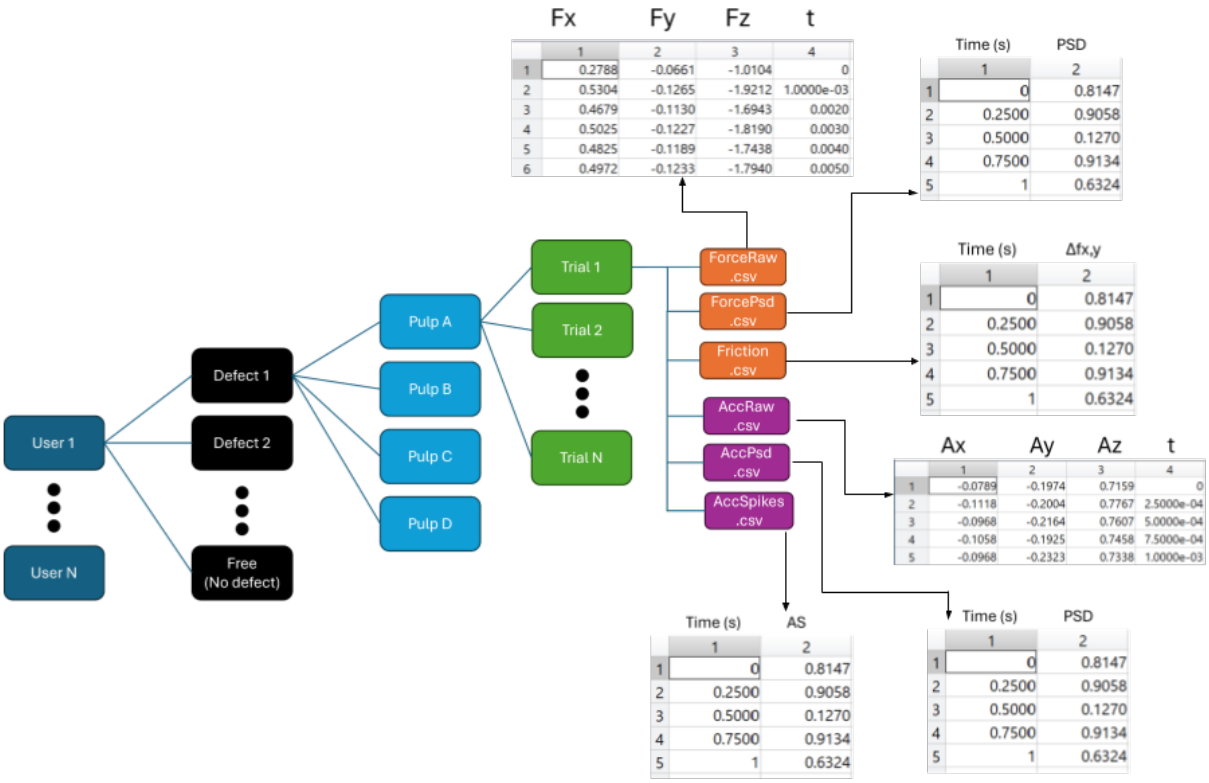


Figure 3.5. At the end of the dataset collection, the data is organized into a hierarchical directory structure. Starting at the root, there is a folder for each user. Within each user's folder, sub-folders are designated for each defect label assigned during acquisition. These defect-labelled folders contain additional sub-folders corresponding to the sensor probes used. At the lowest level of the directory tree, folders contain data from the different trials. After post-processing, each trial is described by 6 .csv files containing raw time series and the extracted features. The labels for defect classification are based on those provided on the car parts by TOFAS, while the labels for different sensor probes follow a sequential letter sequence. Any acquisitions performed on areas without defects are labelled as "Free".

In this initial phase, data was collected from four users across four types of defects (positive dent, negative dent, scratch, and weld spatter), plus areas without defects. For each defect type, five acquisitions were made with four different end-effectors.

The final dataset comprises 400 acquisitions, each containing the following data: 3-axis acceleration signals, 3-axis force signals, power spectral density of acceleration signals, power spectral density of force signals, differential friction signal, and acceleration spike signals. These were obtained from 5 trials conducted with 4 different sensor probes on 4 defect types (plus a defect-free area), by 4 users.

3.6 METHODOLOGIES EMPLOYED

Force and acceleration data are utilized to extract features that can identify the presence of defects in a tactile signal acquisition.

Features Extraction

Through preliminary studies, three key tactile features have been modelled from the acquired data:

1. Power Spectral Density (PSD): Both force and acceleration measurements are used to compute the Power Spectral Density (PSD) of the signals. More in detail, all the three axes of both the acceleration and the force signal are combined into one using the Euclidean norm (see [Landin2010]). This approach preserves the spectral characteristics of the recorded signals and reduces the data dependency of the device inclination toward the surface [Kuchenbecker2011]. Single-axis signals would suffer from the fact that the device is not constantly held perpendicular toward the surface during human freehand exploration of surfaces. The PSD is calculated using MATLAB for both the acceleration and force signals, based on their Short-Time Fast Fourier Transform (STFT). Since the STFT algorithm provides several tuning parameters, such as window size, window overlap length, and window type, preliminary studies were conducted to optimize these settings. As a result, the algorithm was configured to use a Hamming window with a size of 0.25 seconds and a 90% overlap. The PSD computed from the obtained STFT will be a three-dimensional representation, showing how the signal's energy is distributed across different frequencies at each time instant, as determined by the window size. The final total PSD provides valuable information about the spectral distribution of a signal's energy, highlighting the dominant spectral components at frequencies where the PSD values are higher, and indicating less significant components at lower PSD values. Additionally, the PSD derived from the STFT offers insight into how the spectral energy of a signal is distributed over the scan time. By summing the energy of all signal frequencies at each time instant, the total PSD of the force and acceleration signals over time reveals energy peaks as the device's probes pass over a defect (Figure 3.6).

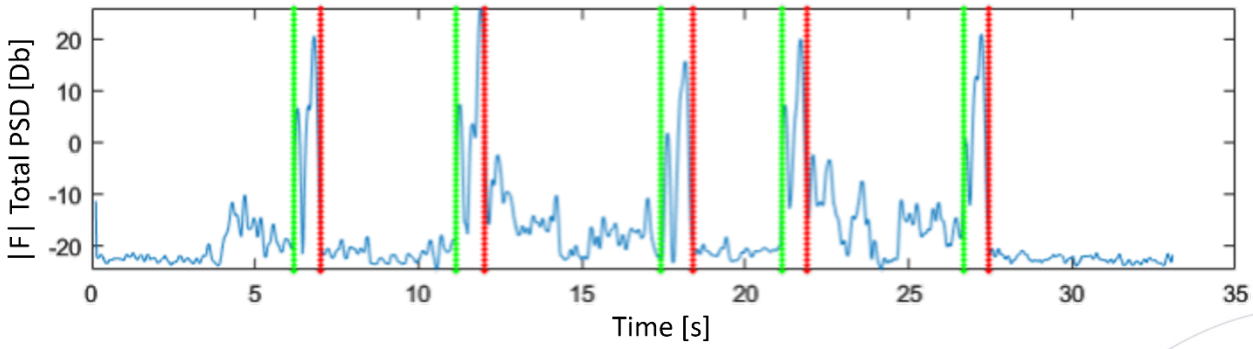


Figure 3.6. Total PSD plot of the force signal norm used to assess the feature's suitability for defect detection. The sensor probe was passed over the defect five times, resulting in energy peaks in the PSD at the time points corresponding to each pass over the defect.

2. Acceleration Spikiness: Spikes in otherwise smooth acceleration time-domain signals serve as strong indicators of bumpy surfaces or meshes where the recording device may occasionally get stuck. Also, for this feature the three axis the acceleration signal are combined to one using the Euclidean norm. An algorithm designed to detect these spikes has been modelled from existing studies [Strese2016]. Given the acceleration signal $\mathbf{x}$, first a low-pass filter with a cutoff frequency of 100 Hz is applied. To further reduce the effects of increasing scan force or velocity due to the operator's hand movements, a Simple Moving Average (SMA) is calculated over 2000 data points ($\overline{x}_{2000}$), corresponding to a 0.5-second time window based on the acquisition sampling frequency of 4 kHz. Next, a threshold vector $\mathbf{x}_{th} = 2^{*}\sigma(\mathbf{x}) + \overline{x} + \overline{x}_{2000}$ is calculated, where $\sigma(\mathbf{x})$ represent the standard deviation of the acceleration signal $\mathbf{x}$. Finally, the signal vector representing the acceleration spikes feature is derived as the difference vector $\mathbf{x}_{\Delta} = \mathbf{x} - \mathbf{x}_{th}$, where all the negative values are set to zero.

3. Friction: Friction is a significant tactile dimension relevant to surface classification [Romano2014] that can be estimated from the force signal. Stickier surfaces exhibit a wider range of $\mathbf{f}_x$ and $\mathbf{f}_y$ values, which can be associated with dynamic friction. Here, x and y represent the axes of the plane parallel to the sensing surface of the force sensor. Variations in these values may indicate the presence of a defect on the scanned surface. The differential friction force values are calculated as $\Delta\mathbf{f}_{x,y} = |\mathbf{f}_x - \mathbf{f}_y|$, providing further insight into surface characteristics and potential defects.

Classification

With the tactile data available, classification models will be explored to be able to detect and classify the defects. The features that are collected are time series. Various potential models have been examined to use for time series classification. Two different directions for classification were investigated and the implemented models will be detailed below. First with a short introduction of the models, followed by some more details on the implementation.

- LSTM and CNN combined
    - Convolutional Neural Networks (CNNs) use convolutional layers to capture spatial hierarchies in data. They are good at capturing local patterns and features in the data.
    - Long Short-Term Memory networks (LSTMs) are an advanced type of RNN that uses gates to better capture long-term dependenncies.
    - Combining CNNs and LSTMs leverages the strength of CNNs in extracting local features and the ability of LSTMs to capture temporal dependencies, making it particularly effective for time series data where both spatial and temporal patterns are important.
- Ensemble learning
    - Random Forest leverage multiple decision trees to capture diverse temporal patterns and reduce overfitting in time series data.

- Gradient Boosting sequentially builds models that correct previous errors, effectively capturing complex relationships in time series.
- Bagging enhances the stability and accuracy of time series classification by averaging predictions from multiple models trained on different data subsets.

For the two different approaches, different data is used as input. The LSTM-CNN combination uses the time series as input, while the ensemble learning method needs features as input. The advantage of the LSTM-CNN model is that it automatically learns relevant features from the time series data and that it is well-suited for handling sequential data and can be adapted for multivariate time series. However, the disadvantage is that it is computationally intensive and needs large amounts of labeled data to generalize well, otherwise there is a higher risk of overfitting. The advantage of the ensemble models is that these are more interpretable, less prone to overfitting and effective with small data. In these methods, custom feature extraction and engineering is allowed which can be useful to incorporate domain knowledge in the features. Disadvantages are that these models might not capture complex patterns in the data as effectively, and information about the temporal dependencies can get lost because features are treated independently.

For the incorporation of both model types, three signals are used: AccelerationPsd, ForcePsd and Friction. The code for the classification is made in Python. First, it is observed that the time frequency of the Psd data (timesteps of 0.025) compared to the friction data (timesteps of 0.001) is not the same, which is desired while combining the timeseries in one model. For this, the option is to up- or downsample. For the purpose of keeping the training time short, the choice is made to downsample the Friction data for now by rounding the time index to the nearest 0.025 seconds, grouping by this rounded time, and then taking the mean of each group.

For training and testing the models, in scikitlearn the train_test_split function is used, with a training set of 70% and a test set of 30%. The stratify parameter is used to make sure the train and test set have the same proportion of each class as in the original dataset. A random state is set to ensure the reproducibility of the data split.

1. LSTM and CNN combined: The first approach consists of two models, for the two different order options: first LSTM, then CNN or the other way around. For these models, there are multiple hyperparameters defined which can be tuned to find the best hyperparameter combination for the model. The hyperparameters are:

- Number of CNN layers
- Number of LSTM layers
- Filter size of the CNN layers
- Unit size of the LSTM layers
- Epochs
- Batch sizes
- Drop rates
- Unit size of the Dense layer

There are some values that are not yet hyperparameters but for the future can be adjusted. Those are the learning rate, metric, pool size, kernel size and the patience.

2. *Ensemble learning*: To use the ensemble learning models, features are extracted from the time series. For this, up- or downsampling is not necessary. Multiple features are computed per signal, and then combined to use in the classifier. The features that are incorporated are the following, and these features are computed for all three signals AccelerationPsd, ForcePsd and Friction:

- Mean
- Median
- Standard Deviation
- Variance
- Skewness
- Kurtosis
- Minimum
- Maximum
- Quantile 25
- Quantile 75
- IQR
- Range
- Peak count
- Valley count
- Zero Crossing Rate
- Entropy
- Autocorrelation lag 1
- Autocorrelation lag 5

Features with zero variance are removed, and further feature selection can be considered in the future. With these features, the three ensemble models can be applied: Random Forest, Gradient Boosting and Bagging. For all three, the number of estimators is set at 100 and a random state is set.

Next to these developed models the Acceleration Spikes are considered, to see if those could be used those to first classify data into whether or not it was a defect, and then determine for the defects which defect it exactly was. For the classification of the spikes, it is wished to apply the same models as before but then for binary classification. However, the LSTM-CNN combinations seem to perform incredibely slow in this case, for which an explanation will be searched. Because of this, in the preliminary results for now LSTM-CNN (with 10 epochs) is included and not CNN-LSTM.

For all of the above, first the data from one pulp (A) is used, with all the trials and users incorporated. With the real data, there will be looked into the differences between the classification results if multiple pulps, multiple users, or other combinations of signals would be used.

3.7 PRELIMINARY RESULTS AND FINDINGS

Features Extraction

By collecting the Power Spectral Density (PSD) data for various defects and different probe textures, an analysis of the crest factor of the PSD signal was conducted for each acquisition. It was observed that, in general, the crest factor of the PSD signal is higher when the device passes over a defect compared to when no defect is present. Furthermore, different probe textures can be employed depending on the type of defect, allowing for the optimization of the relative crest factor for improved defect detection (Figure 3.7).

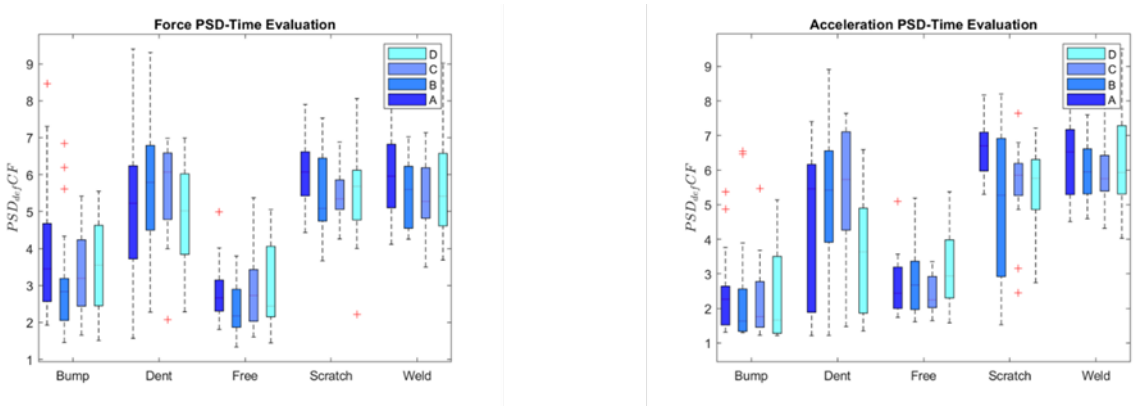


Figure 3.7. Box plots showing the distribution of the Total PSD crest factor of the force and acceleration signals for the entire data acquisition. The PSD crest factors are grouped by defect, and within each defect category, they are further grouped by the different sensor probes used. It is noticeable that the median of the Total PSD crest factors is generally higher when a defect is present during the acquisition. Additionally, this information can be leveraged to determine which sensor probes are more effective at identifying specific types of defects.

By leveraging the acceleration spikiness feature, it was observed that the acceleration signals exhibit significant peaks when the scanned surface contains a defect, compared to when no defect is present. These acceleration peaks can therefore serve as reliable indicators of the presence of a defect and can also be used to characterize the type of defect encountered (Figure 3.8).

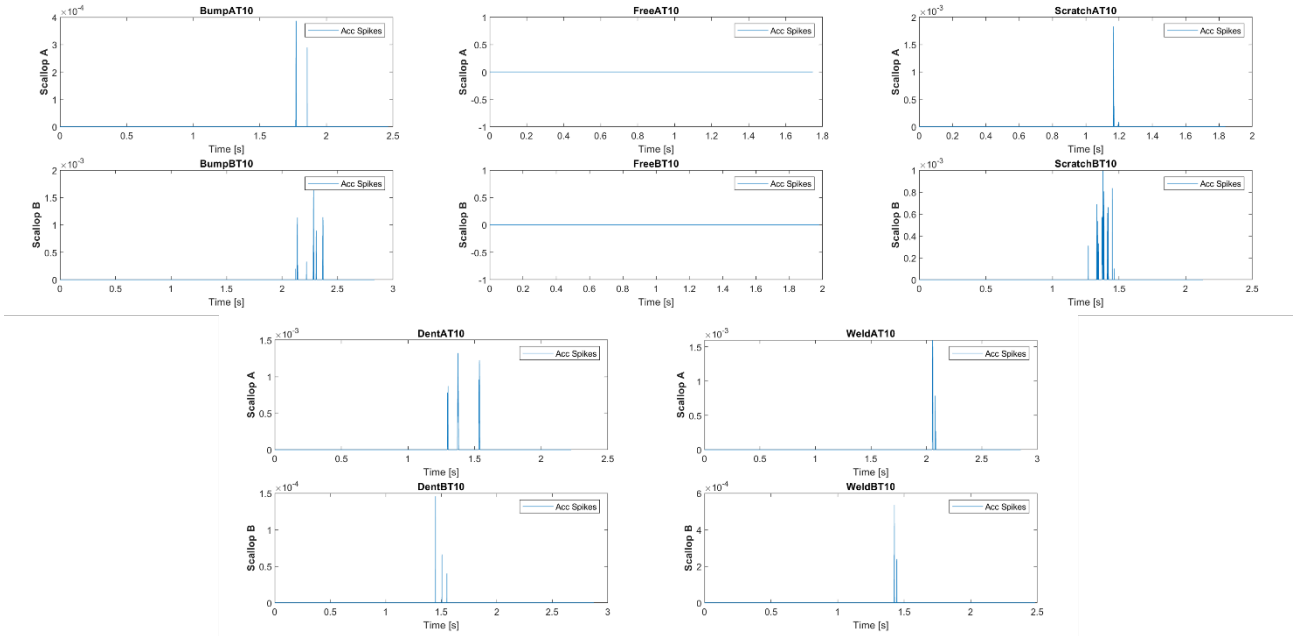


Figure 3.8. Acceleration Spikes feature for a single trial performed by one user. For each defect, the acceleration spikes feature is displayed for each probe used. It is evident that significant peaks in acceleration spikes occur when a defect is present, while no notable peaks are observed in the other case.

Similarly to the acceleration spikiness feature, friction data can be compared between acquisitions with and without defects. By examining these comparisons, it becomes evident that dynamic friction exhibits a notable peak when a defect is present on the scanned surface. This increase in friction serves as an additional indicator of defect presence, reinforcing the findings from the acceleration spikiness analysis (Figure 3.9).

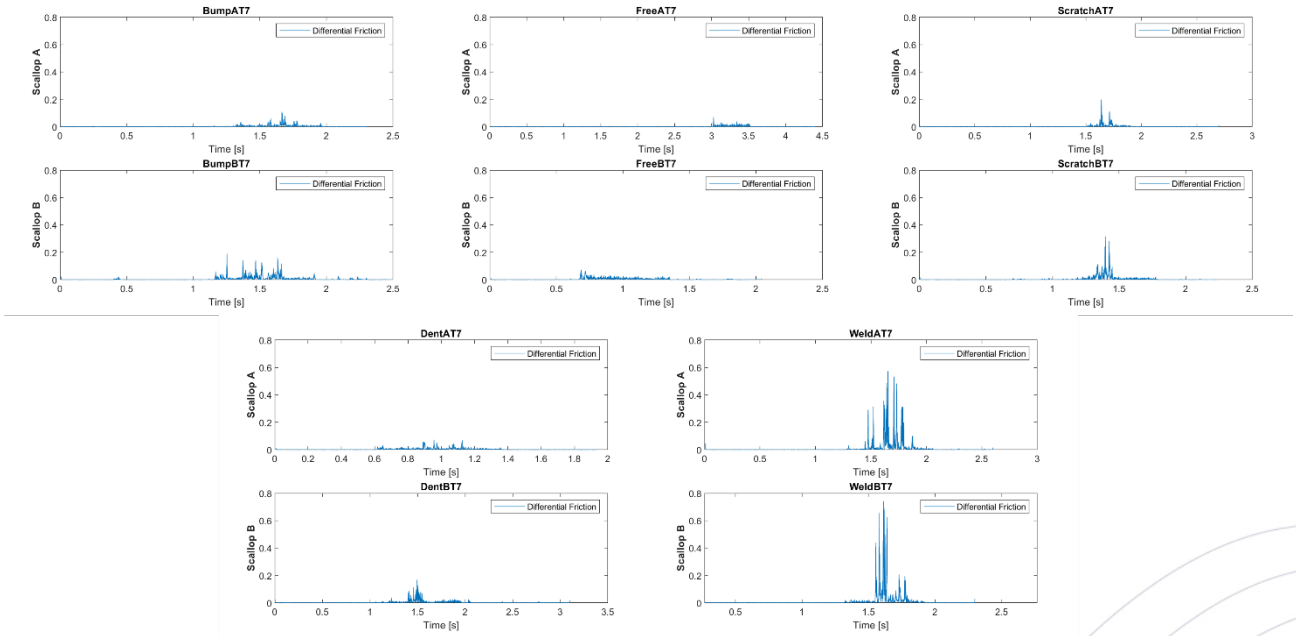


Figure 3.9. Differential Friction feature for a single trial performed by one user. For each defect, the differential friction feature is displayed for each probe used. Also in this case, even if less evident with respect to the acceleration spikes feature, it is evident that significant peaks in the feature occur when a defect is present, while no notable peaks are observed in the other case.

Classification

For all of the five classification models, some preliminary results were gathered on the test data available. Since this is not real data, conclusions are not drawn yet. It gives an indication of the performance of the models. The code for the models is prepared such that the real data can be tested quickly when becoming available.

For the preliminary findings, we begin to look at the introduced models. For the LSTM-CNN combinations, the code has been runned three times to see if they differ a lot, and for the results we give the average score. The results are:

- First CNN, then LSTM 42,22%
- First LSTM, then CNN 62,22%
- Random Forest 70%
- Gradient Boosting 63%
- Bagging 70%

Next to the overall accuracy, it is interesting to see for these models despite which defect it has to be, how much of the time is the fact that whether or not it is a defect, predicted correctly? For LSTM-CNN this is on average for the three runs 82,22% and for CNN-LSTM it is 72,22%. For the ensemble models, it is in all cases 96,67% of the time predicted correctly.

If the acceleration spikes with a random forest are considered, it results in an accuracy into defect yes or no of 73,33%. With the LSTM-CNN model, that is 80%. The separate classification with the spikes does not seem to work that well, but with the real data we will check these results again.

3.8 CHALLENGES AND LIMITATIONS

One of the key limitations at this stage of the project is the use of data acquired from custom-made parts, which may not fully represent actual car-body components. This highlights the need to validate the system's performance on real car-body parts to confirm the validity of the current results.

Data Acquisition

A significant limitation is the large volume of data required to effectively train the models. The current dataset size is insufficient for developing a machine learning algorithm with finely tuned parameters capable of delivering robust and reliable results. Expanding the dataset is crucial to improve the algorithm's performance and ensure its accuracy in detecting defects under real-world conditions. Additionally, since the current data acquisition is performed with surfaces in a horizontal position, it may differ from the

vertical orientation used in the assembly line. To account for this, tests will be conducted with surfaces placed vertically to determine if there are any differences in the scanning process. Moreover, in this initial phase the participants collecting the data are not the actual operators, which could lead to discrepancies in the results. To address this issue, we are evaluating the possibility of conducting data acquisition sessions at TOFAS, allowing us to work with real operators. This will help us gather more representative data and better reflect the actual working conditions in a production.

Classification

Since the real data is not available, conclusions from the stated preliminary results cannot be drawn yet. When this becomes available, the available code will be executed and based on those results, we choose a path to continue on. We will continue and explore other combinations of signals and look at the results for different pulps.

Expected challenges that might still come our way are:

- Training time: especially the LSTM-CNNs are not that quick, for a combination of signals for one pulp, to check all hyperparameter combinations once takes around 20 minutes. If we want to run it several times, for different combinations of signals and pulps, and maybe add some more hyperparameters, this will probably increase. However, after training, it does not take long to classify the results. Depending on the requirements for the training time, it can become a challenge to deal with if the training time has to be shortened. Gated Recurrent Unit (GRUs) are an alternative to consider because those have similar functionality but are less computationally intensive.

Data format of the real data: Now the data is gathered per defect, and the data on the real test parts will be too. It is expected that when the tactile sensing data will be collected in the real final case, it will be data from all around the car and it can be the case that we have a time series where there are multiple defects contained. In this case first the defects have to be detected before they can be classificated, and this will be something to think about how to do this in a correct way.

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 STATE OF THE ART

The advent of neural networks revolutionized 3D human pose estimation that was previously possible only using specialized RGBD cameras [Oiko11, Bhol14]. OpenPose [Qiao17] was the first method that really pushed the state of the art allowing full body, hands and facial 2D pose estimation in real-time through GPGPU acceleration for multiple persons using generic “in-the-wild" uncalibrated video streams. After OpenPose, research pushed in two main directions: the first was towards optimizing accuracy with works such as HRNets [Wang20] that are among the recent state of the art 2D detection-based methods. The second direction included methods focusing on real-time performance such as BlazePose [Baza20, Mroz21], that are suitable for execution on computationally constrained hardware.
Moving to 3D human pose estimation, early methods typically handled multi person scenarios by iteratively applying the same single person method on different areas of the image. Techniques can be also classified as one-stage or two-stage depending on whether they first involve an RGB to 2D pose estimation step or if they extract RGB to 3D in a single step, such as applying a single, monolithic network to the input image. Notable techniques in this category include X-Nect [Meht20] and ZoomNAS [Xu22]. Gradually research also included mesh reconstruction aspects, with methods such as Monocular Total Capture [Xian19], Expressive body capture [Pavl19], proHMR [Kol21] and DiffPose [Gong23]. It's worth noting that the top performing recent methods (Figure 4.1) in terms of accuracy all use a transformer-based architecture. Two of the most prominent such methods are MotionBERT [Zhu23] and ViTPose [XuY22]. Non-

transformer architectures however continue to dominate real-time applications due to their superior computational performance.

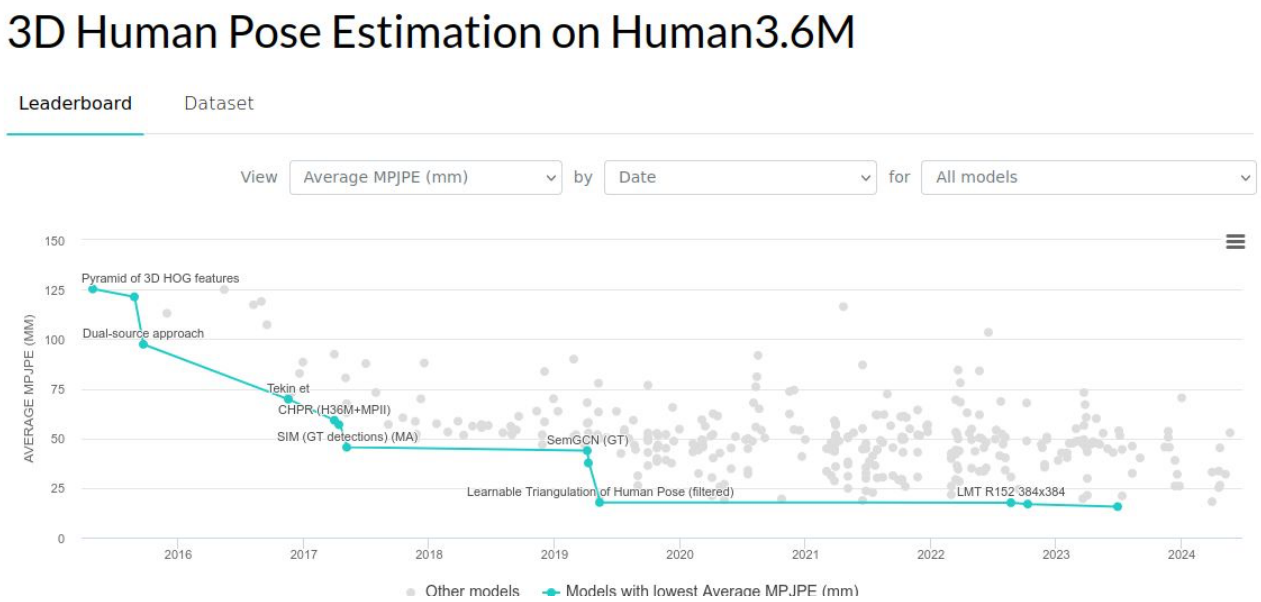


Figure 4.1. Leaderboard of state-of-the-art 3D Mean Per Joint Error (MPJPE) in the very commonly used Human 3.6M dataset [Ione13] in the last years from https://paperswithcode.com/sota/3d-human-pose-estimation-on-human36m

Very recently, during the last year, mature foundation models that allow Depth estimation from RGB appeared with the notable examples of Marigold [Ke24] and DepthAnything [Yang24]. These, coupled with models for segmentation such as MaskRCNN [He17], Detectron 2 [Yuxi19] or Segment Anything [Kiri23], provided a new way to tackle multi-person 2D, 3D and Mesh recovery. Even more recently, Meta announced the "Sapiens" [Khir24] foundation model[5] for human perception that is the new state of the art in the field.

[5] https://about.meta.com/realitylabs/codecavatars/sapiens/

Figure 4.2. The Sapiens foundation model [Khir24] is the state of the art providing pose, segmentation, depth and 3D normals in unprecedented detail for humans observed by regular RGB cameras.

4.3 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 4.3. 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 [Ant21]. 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 4.4.1) 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 4.4.2) 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.

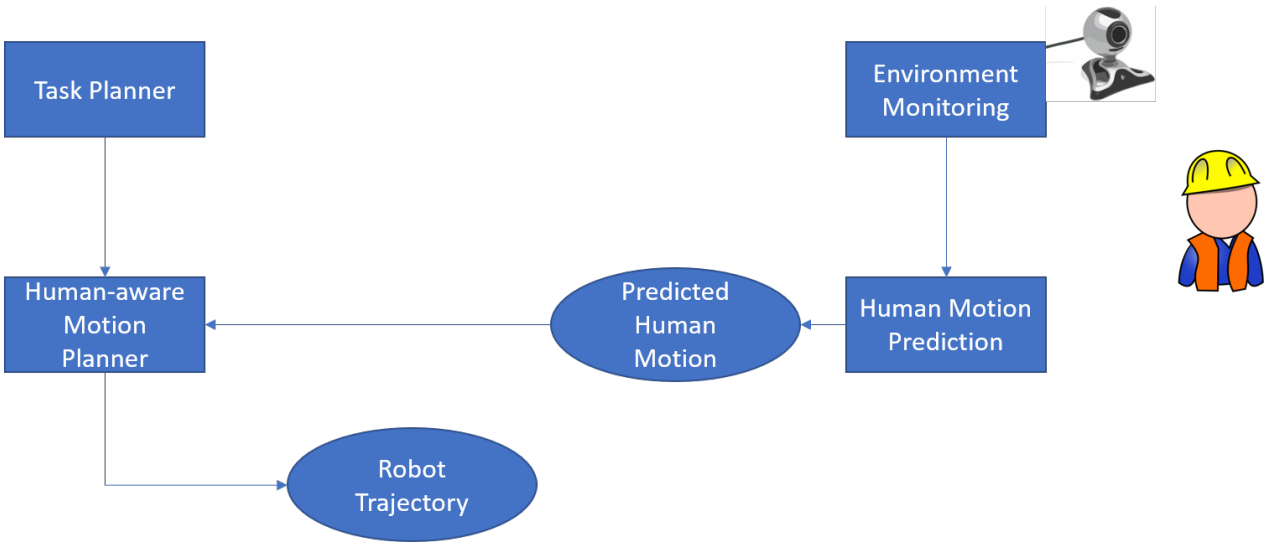


Figure 4.3. The framework of Human-Aware Motion Planning.

4.4 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 4.4.1 and 4.4.2. For pose detection we are developing a novel U-NET based architecture that regressed 2D pose, depth and normals 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.

4.4.1 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.

4.4.1.1 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 4.4), 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 4.4. 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 2.5 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 4.5) allows users to annotate captured human pose data and prepare it for training, as shown in Figures 2.8 and 2.9. Depth and 3D Normals 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 OpenPose to automatically annotate the 2D landmarks. After these procedures are completed, the user can review and correct human joint landmarks, which are often miscalculated in challenging images with many people and occlusions. The introduction of the Sapiens Foundation Model [Khir24] (Figure 4.2) offers a potential future solution for initializing annotations, helping to reduce the time required for annotators by providing a more accurate starting point for each image.

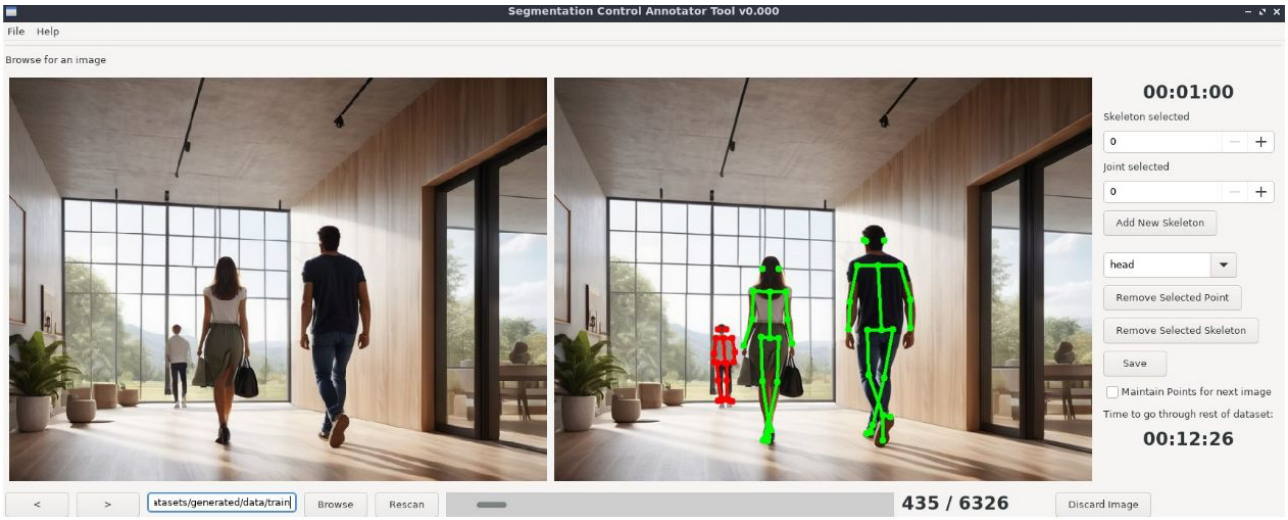


Figure 4.5. The Annotation tool developed while used to annotate synthetic data generated using generative AI to be included in our model's training.

4.4.1.2 POSE DETECTION METHOD

For the neural network architecture, the priority is real-time performance. Particularly in production line settings, it is impractical to deploy a complex pose estimation method that cannot deliver results at interactive frame rates (>10Hz). The neural network we developed utilizes the same toolkit as the defect detection module to ensure shared dependencies and full software compatibility.

To ensure the neural network is robust to adversarial conditions it might encounter, a series of augmentations are applied to the training data. These augmentations include brightness and contrast adjustments, stochastic uniform Gaussian noise corruption, burned pixel simulation, as well as pan, zoom, and rotation transformations. These techniques help the neural network develop internal representations that are resilient to common input artifacts, ensuring proper generalization across various adversities it may face during deployment.

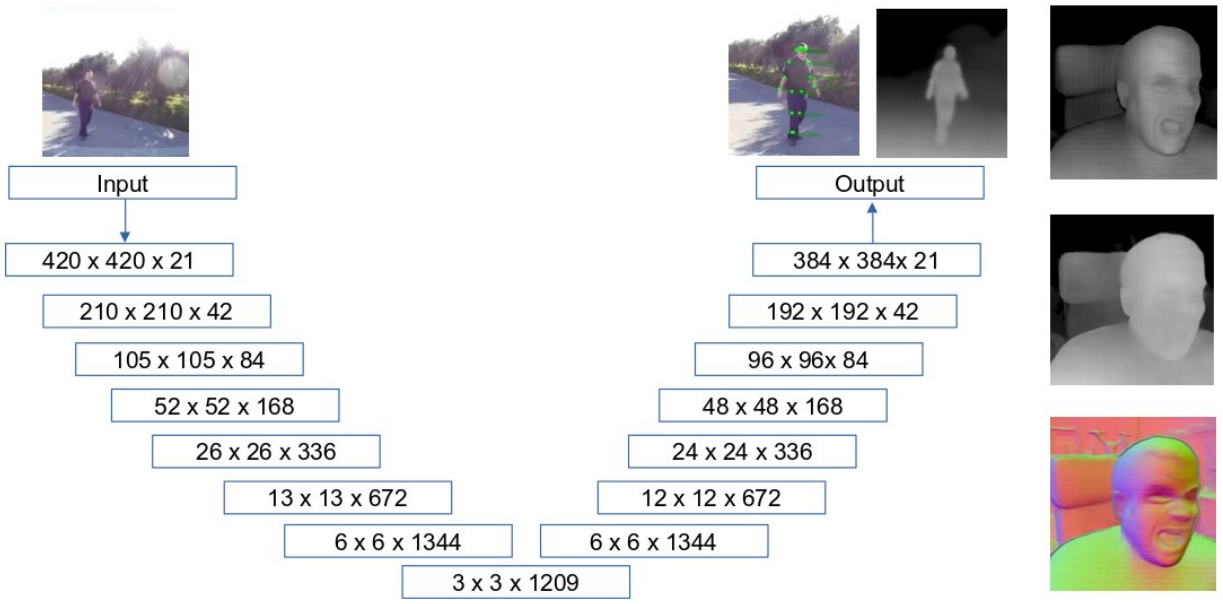


Figure 4.6. The architecture employed is based on U-NET, receives an RGB input and outputs 2D key points, a depth map, normals as well as segmentation data, providing ample data for the human motion detection task and allowing the cobot to maintain a good understanding of human presence and motion. The network works at a 15Hz framerate at mid-tier graphics card (GTX1060) in order not to monopolize the available computing power on-board the robot. It's worth noting that during the time of writing this deliverable, the "Sapiens" foundation model by Facebook/Meta (Figure 4.2) beat us to publication, now being the first published work made available leveraging the idea of regression of key points, depth maps, segmentation masks and normals.

Due to the computational complexity of training large neural networks, we employ a multi-core optimized low-level data loader written in C that is thoroughly optimized for optimal use of the available computational resources.

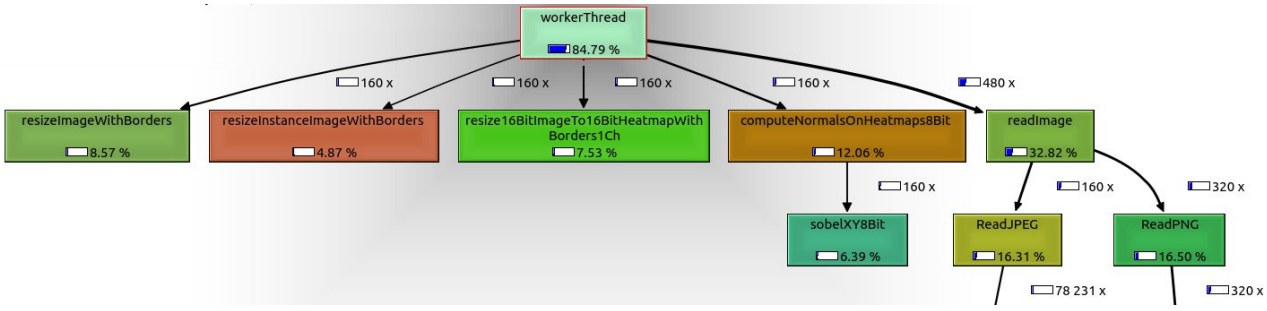


Figure 4.7. We employ rigorous profiling using valgrind to optimize the data loader architecture for our NN training code. Having an optimized training methodology available from the start of the effort will ensure the maximum amount of time-savings through the training lifetime of this project.

For the task of human motion detection from RGB images, the trained neural network we propose converts RGB inputs into depth, normals, key points, and segmentations, providing a wealth of data. However, further human pose estimation data can be extracted using a method based on inverse kinematics estimation. If needed, we plan to

utilize the MocapNET [Qamm19, Qamm20, Qamm21] methodology (Figures 4.8, 4.9, 4.11, 4.12) as a back-end solution to provide this data in real-time.

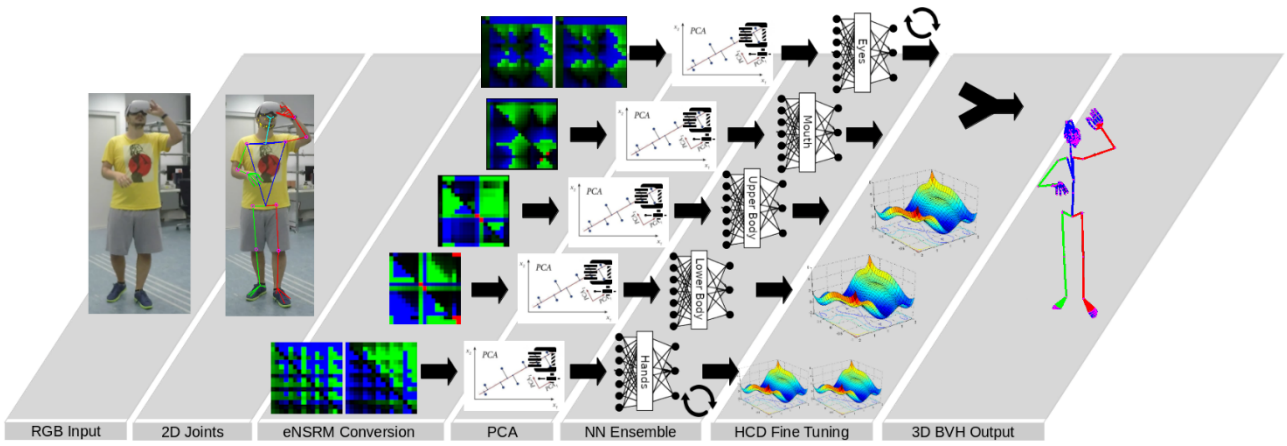


Figure 4.8. MocapNET [Qamm19, Qamm20, Qamm21] is a 2-stage method that can perform inverse kinematics from a cloud of 2D points using an ensemble of neural networks. The output of the method is a Bio Vision Hierarchy (BVH) MOCAP skeleton output that is similar to data acquired by systems with motion capture suits and markers.

As mentioned in Section 4.2, traditional human motion detection and pose estimation methods typically focus on the body, excluding the face and hands. This is mainly due to sensor resolution limitations and the fact that in full-body image streams, hands often occupy only a few pixels, making it challenging to discern hand poses. However, with modern high-definition video streams, holistic capture methods can now address the combined problem of body and hand motion. The two methods we propose also account for hand poses, provided a clear, high-resolution stream is available. The U-Net we developed (Figure 4.6) provides depth information for the body, hands, face, and objects in the scene (Figure 4.10). Additionally, [Qamm21], when given 2D landmarks for hand joints, can offer an inverse kinematics solution for observed hand poses, as shown in Figure 4.8 and Figure 4.9. It is worth noting that the MocapNET method also supports facial and gaze tracking [Qamm23]. However, since each body sub-hierarchy requires additional computational resources, we expect the facial landmark capabilities will not be used, especially because factory workers often wear protective glasses and masks that cover their eyes and face.

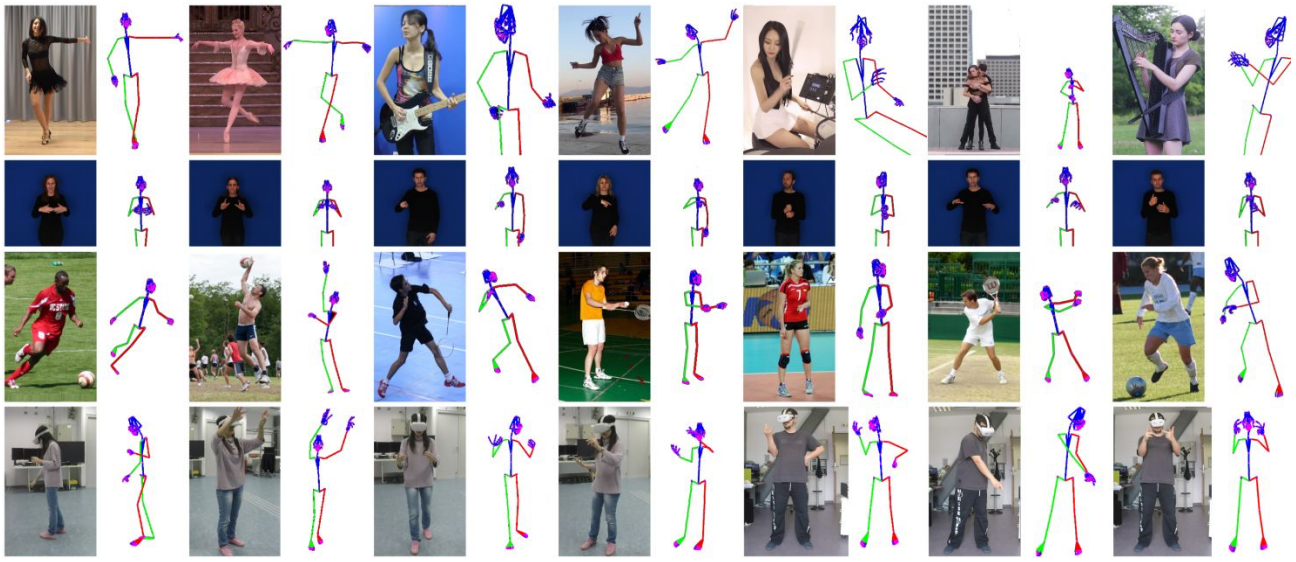


Figure 4.9. Qualitative results for body+hands using MocapNET [Qamm21] ensembles for various types of input data.

By using multiple camera sources and applying the same techniques to each incoming video stream, we can scale the human perception pipeline to a multi-view scenario. With more than one candidate output pose, we can average the recovered results across observations, accounting for limb visibility to help reduce noise and pose jitter. However, it is important to note that processing multiple streams in real-time is computationally expensive, as resource demands scale linearly with the number of video streams. Ensuring high-quality 3D human pose estimation even in a monocular RGB scenario allows us to handle cases with additional streams more efficiently.

4.4.1.3 PRELIMINARY RESULTS AND FINDINGS

Although we do not currently have any human-related datasets from the factory lines of the project, we have obtained encouraging preliminary results by training the proposed methods on open datasets and qualitatively testing them on data recorded in compliance with EU procedures. This testing was conducted within the framework of the EU H2020 SustAGE Project (no. 826506).

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, modifying the neural network in Figure 4.6 by adding more layers and increasing its capacity can yield even cleaner output, based on our needs following experimental evaluation of the solution.

Figure 4.10. Depth stream acquired in real-time (20Hz) from RGB images using the developed U-NET of Figure 4.6. The proposed framework provides a dense depth map of the scene thus hopefully mapping closely to the Human-Aware Motion Planning task by allowing the robot to avoid impacts with humans and objects on the scene, while also providing enough information for finding contact points of the person with objects to facilitate the motion prediction task. Dataset provided by the EU H2020 SustAGE Project (no. 826506).

Using just the 2D landmarks of the observed human and a neural network capable of performing real-time inverse kinematics, like MocapNET [Qamm19, Qamm20, Qamm21], we can extract a 3D skeleton. This approach works even without having a solution that provides a dense depth map of the scene in which the robot operates. The resulting 3D skeleton not only contains the 3D positions of each observed joint but also the 3D orientation of each. This transforms the human skeleton from a list of 2D or 3D points into 3D rotations for each degree of freedom of the human body, allowing us to analyse their motion, acceleration, and velocity (Figure 4.12). These insights can then be used to anticipate human movements and integrate them into the robot's motion planning module.

However, since the input to the inverse kinematics step is the human 2D joints detected in the RGB image, this technique is more susceptible to occlusions and noisy 2D skeleton outputs. Also, the robot's field of view plays a crucial role in this task's success. If large parts of the skeleton are not visible to the system, solving the inverse kinematics chain becomes impossible.

We believe that combining these two methods addresses the problem from both a bottom-up (scene depth) and a top-down (human inverse kinematics) approach. The intersection of these methodologies will ultimately enable a high-confidence motion planning module, ensuring safe interaction between robots and factory workers.

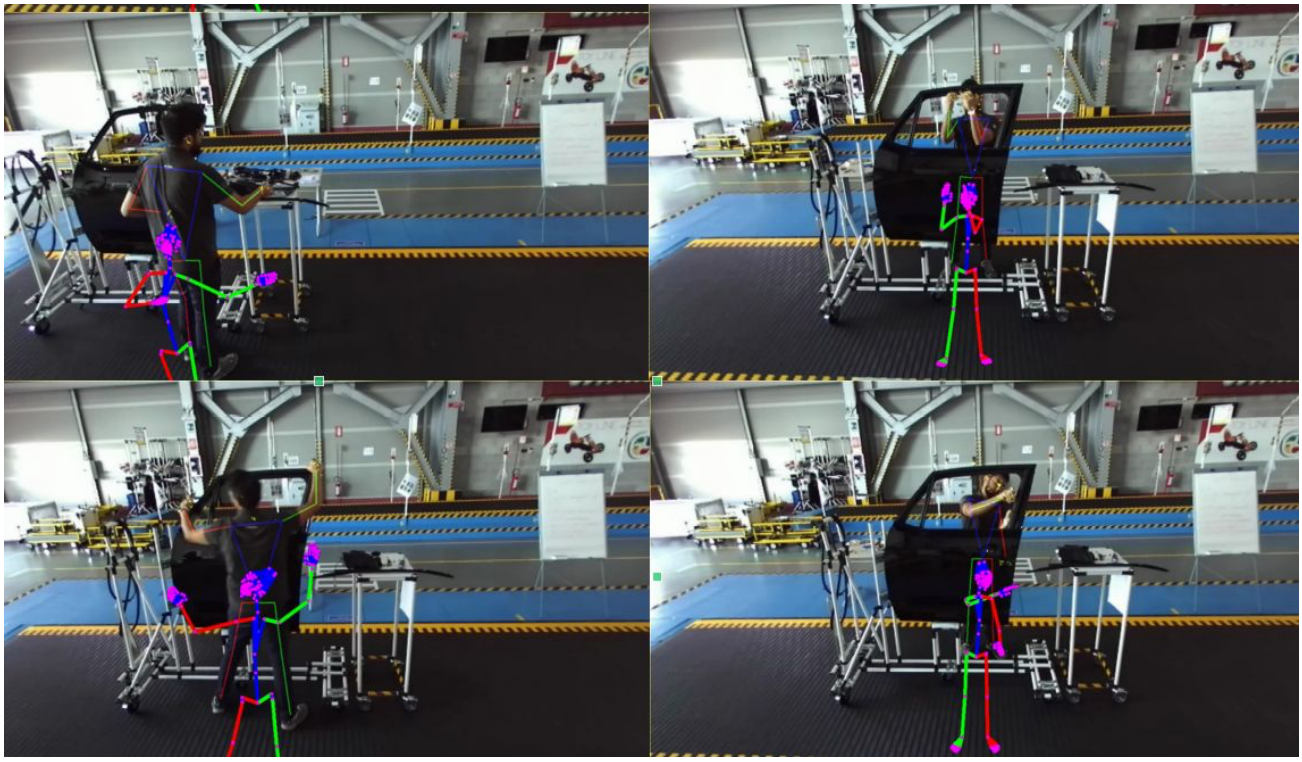


Figure 4.11. Given 2D joint key points for an observed human, MocapNET [Qamm19, Qamm20, Qamm21] can provide a 3D inverse kinematics solution for the skeleton in real-time, to provide high-level pose data to the motion prediction module. Dataset provided by the EU H2020 SustAGE Project (no. 826506).

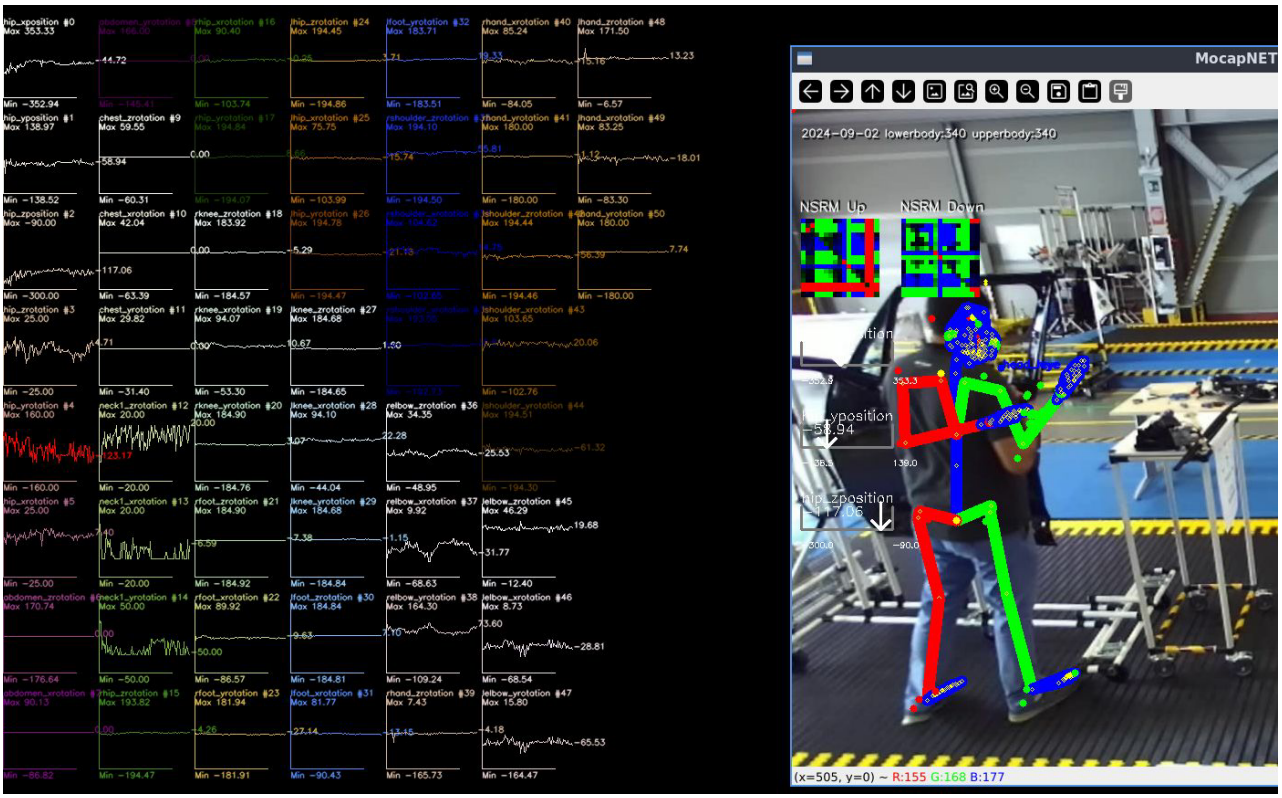


Figure 4.12. MocapNET real-time 3D kinematic solution for each observed joint plotted in 1D graphs for each of the joints tracked. Using this high-level as input, the state and acceleration of the various limbs of the human can be studied to create policies altering the robot trajectory according to the predictions of the human motion. Dataset provided by the EU H2020 SustAGE Project (no. 826506).

4.4.2 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.

4.4.2.1 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 30fps. The average error at the wrist with three adjacency matrices was 102 mm, while the single matrix delivered an average error of 103mm 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 30fps. 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.

4.4.2.2 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 4.13). Below we discuss the segmentation process and the specific clustering model used.

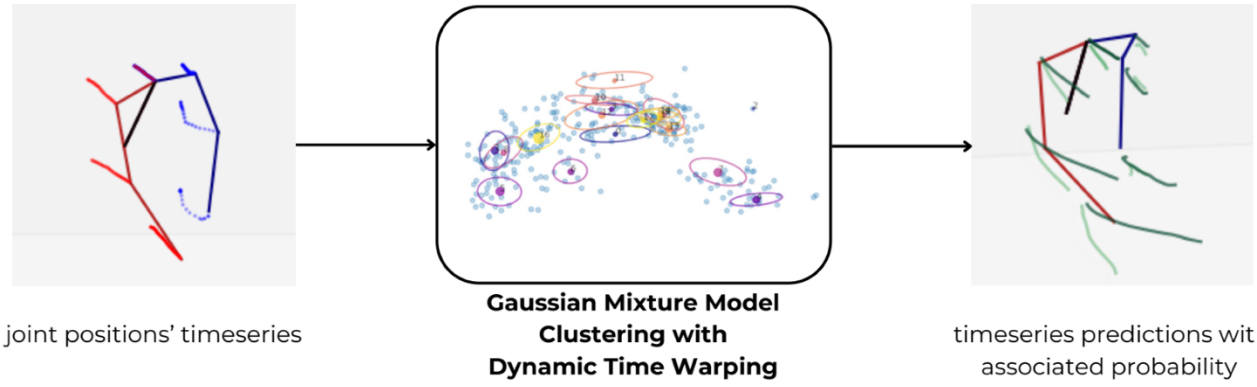


Figure 4.13. Segmented joint position time-series data clustered using GMM with DTW, resulting in time-series predictions with associated probabilities.

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.

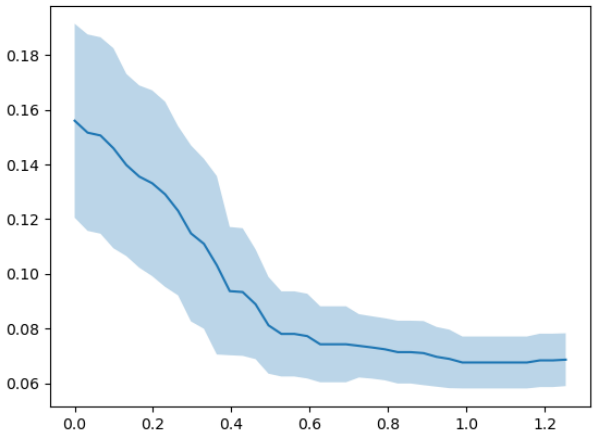


Figure 4.14. Reduction of wrist position error over time, showing the decrease in mean error and variance as the prediction horizon extends.

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 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.

4.5 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. As shown in Figure 4.15, with our current resources, we are limited to only four neural network 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 close 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.

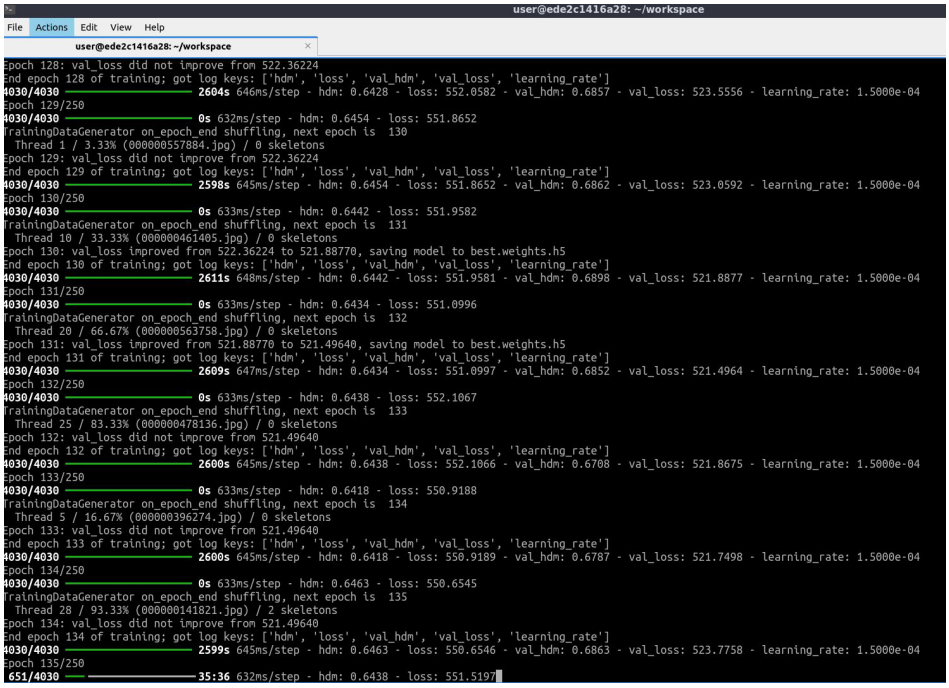


Figure 4.15. Despite the various training optimizations (Figure 4.7), training a 200M model with 120K training samples on a workstation equipped with an NVIDIA RTX A6000, 512GB RAM and a 16 core/32 thread CPU takes 2600 sec / epoch, or approximately one week for a full training session. This makes iterations and improvements on the model very time consuming, with a slow development time.

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.

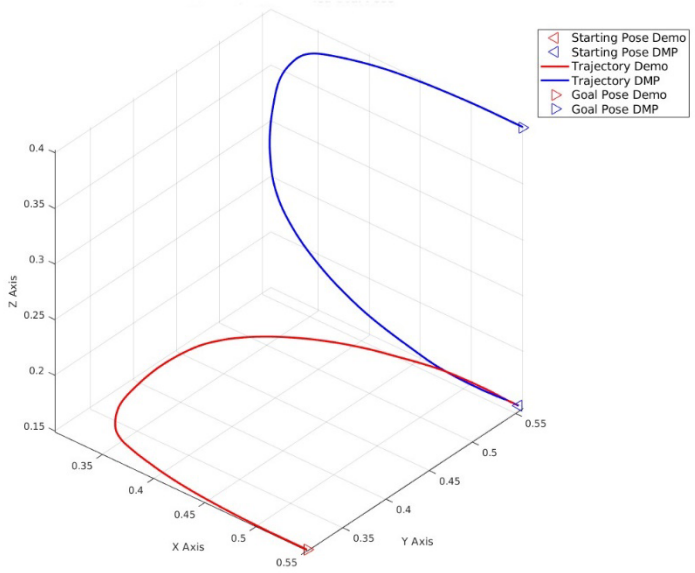


Figure 5.1. Example of generalization ability of DMPs together with Riemannian metrics. Changing the goal pose, the DMP is able to generate the same trajectory from the starting pose, without losing any information.

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. [Ijspeert2002], 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 through the use of 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.

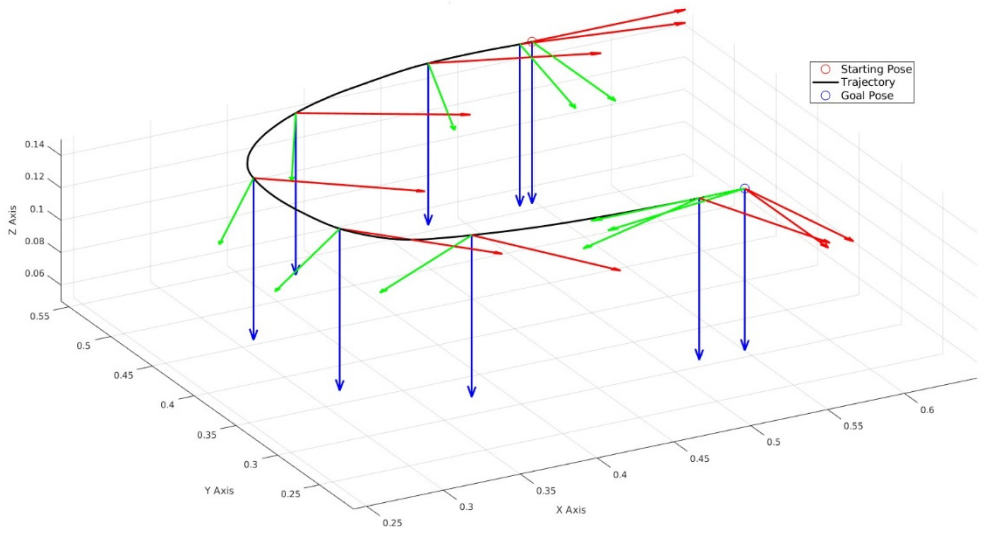


Figure 5.2. Example of a recorded trajectory performed by a human during the sensing phase. The orientation in this case is expressed in form of quaternion.

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 are able to 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).

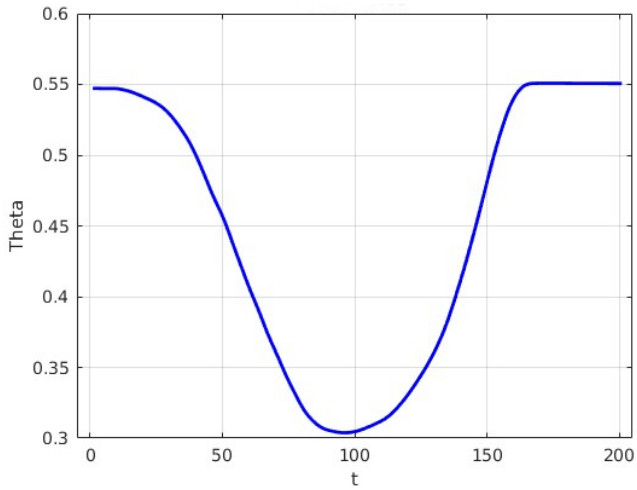


Figure 5.3. Geodesic between two consecutive orientations. The angle of difference is expressed in radians. This representation is especially beneficial for generalization and to compress data.

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.

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 PRELIMINARY 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 5.4) and a mean of **8%** regarding the orientation (Figure 5.5). 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 5.6), achieving a task success rate of **96%** when evaluated on a benchmark set of test cases, which included defects of various sizes and types.

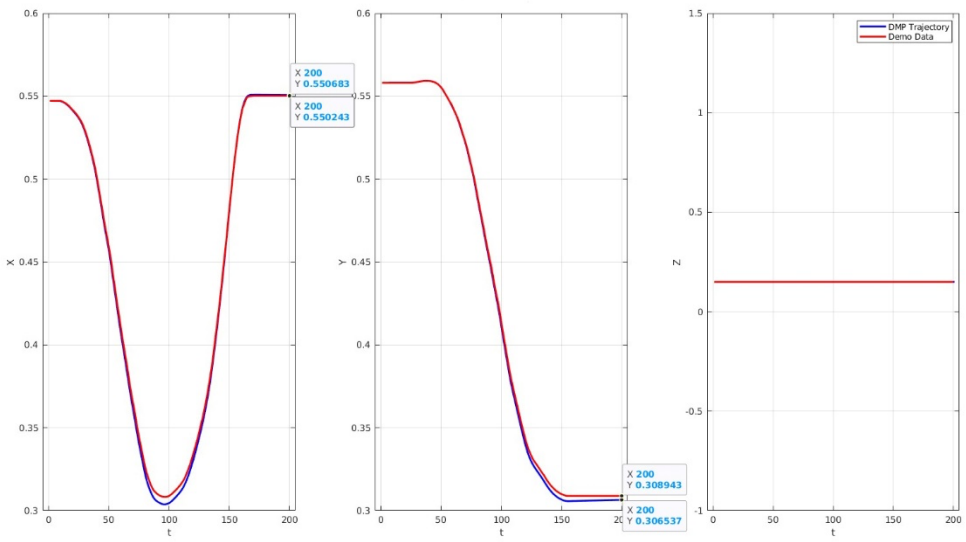


Figure 5.4. Demo human trajectory (red) and trajectory reproduced by the Cartesian DMP (Blue) of the position (x,y,z). As it can be seen, the error is very low, and it can still be improved with finer tuning of the parameters belonging to the DMPs.

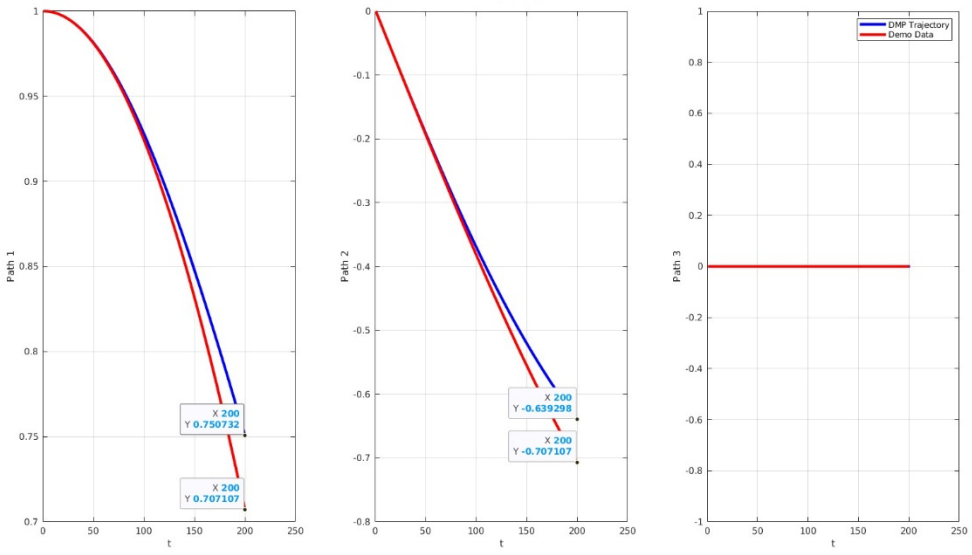


Figure 5.5. Demo human trajectory (red) and trajectory reproduced by the Riemannian DMP (Blue) of the orientation (expressed as axis). As it can be seen, the error is higher than the position but the performances when generalizing are superior to what can be obtained with the classical DMP.

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 in close proximity to 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.

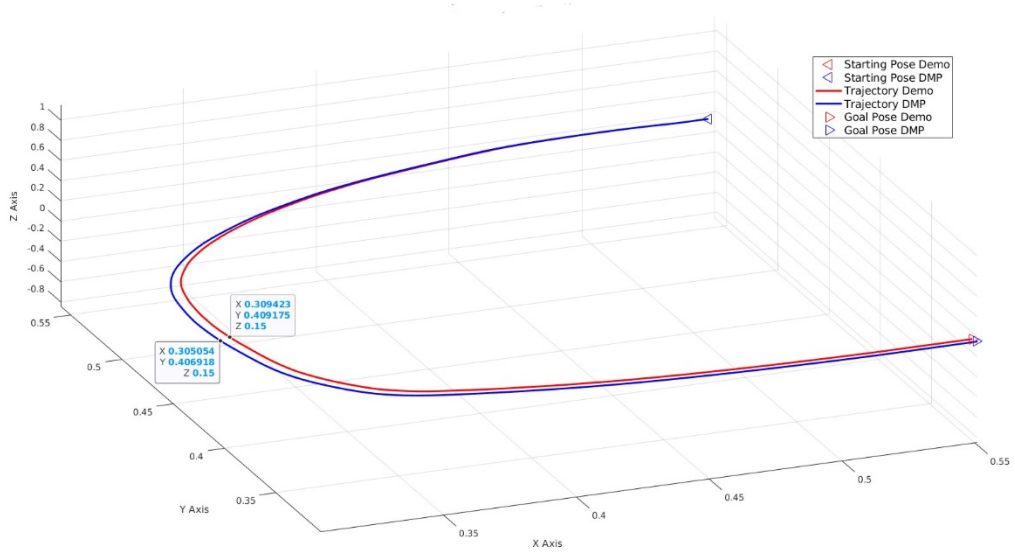


Figure 5.6. Complete trajectory of the demo and the DMP trajectories. As can be noticed, the motion of the human is preserved when replicated, and the final error is very low.

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 activities of the first year in the MAGICIAN WP3, 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

The proof-of-concept visual perception module we developed serves as the baseline for its further development. We can summarize the next steps for each of the components with the following list.

Defect Sensing module:

- Redesign with multiple programmable light sources for surfaces that currently occlude the single light source currently used.
- Possibly switch to 13 mm lens instead of 12 mm for better accuracy if we can reduce scanning area.
- Add passive range finding provisions to the sensor to be used by the robot in combination with the fixed focus lens.
- Have a rigorous CAD/electronics design to ensure reproducibility of the sensor head in later work packages / between partners / during actual deployment / if the working sensor becomes damaged by a collision in the actual robot.
- After finalizing the sensor design and construction, record finalized data using data from all partners and train network with most data possible.
- Optimize computational workload considering integration to the rest of the platform
- Integrate packages with ROS / rest of the robot

Motion Detection module:

- Decide on where the human pose estimation camera will be placed, its field of view etc.
- Improve human pose estimation by incorporate data from our use case.
- Integrate packages with ROS / rest of the robot

6.2 FURTHER DEVELOPMENT OF THE TACTILE PERCEPTION MODULE

The next phase of development for the Tactile Perception Module will focus on refining data acquisition protocols and maximizing the amount of collected data. This will involve improving the efficiency of the acquisition process and expanding user participation in data collection to better generalize defect exploration methods. Specifically, IIT will utilize a Vicon system to track the Tactile Perception Module, incorporating tracking data into the collected dataset. This will enable the continuous data stream to be managed and facilitate automatic labelling of multiple defects within a single scan during post-processing. Indeed, in the initial development stage, data was labelled asking the participant to pass over a specific defect. However, future acquisitions will include

precise tracking of the tool's position and orientation, along with the exact defect location on the surface. Moreover, the new protocol will ensure that users are unaware of defect locations by removing any indicator on the car bodies (e.g. circles made with markers), eliminating potential biases in their scanning movements. This will result in more reliable force and acceleration measurements. These improvements will enhance the accuracy of the scanning process and defect classification by ensuring precise coverage of the tool's movement.

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, we proceed by first using the vision sensor (which does not require to come into contact with the car-body and requires a smaller time), and then move to the tactile sensor only if the confidence in the outcome of the detection algorithm is below a certain threshold.

Consider a candidate area and let:

- D be the event: “defect in the area”;
- V be the event: “vision sensor notifies the presence of a defect”
- T be the event: “tactile sensor notifies the presence of a defect”
- The overline notation denote the complement of an event (e.g., $\overline{D}$ means absence of defect in an area).

Based on the previous measurement, we have a statistical evaluation of the presence of the defects. In other words, when the car-body arrives, for the considered location, we know P(D). Clearly the convenience of a measurement is there if:

$$P(D|\overline{\mathrm{TV}}) \leq P(D|\overline{\mathrm{V}}) \leq P(D) \leq P(D|V) \leq P(D|VT)$$

In other words, if the visual sensor notifies the presence of a defect, it’s the resulting conditional probability has to be greater that then prior probability, and conversely if the visual sensor signals the absence of a defect, the conditional probability has to be smaller. As we discussed in D4.1, it is possible to see that the convenience of

a measurement requires:

$$P(V|D) \geq P\left(V\middle|\overline{D}\right),$$

and the convenience of a tactile confirmation requires that:

$$P(T|D) \geq P\left(T\middle|\overline{D}\right).$$

The two equations above reflect the intuitive notion that in order for a sensor to convey valuable information, its accuracy has to be greater than the probability of a false positive.

The convenience of the combined use requires that *P(D|VT) > P(D|V)*. The exact threshold on *P(D|V)* to decide the use of the tactile sensor is a tuning variable, as explained in D4.1.

Input from the sensing components is $P(V|D), P\left(V\middle|\overline{D}\right), P(T|D)$ and $P\left(T\middle|\overline{D}\right)$, which will have to be provided based on a precise characterisation of the solutions proposed above.

7 REFERENCES

[Chinello2012] Chinello, Francesco, et al. "A three DoFs wearable tactile display for exploration and manipulation of virtual objects." 2012 IEEE Haptics Symposium (HAPTICS). IEEE, 2012.

[Pacchierotti2017] Pacchierotti, Claudio, et al. "Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives." IEEE transactions on haptics 10.4 (2017): 580-600.

[Kappassov2015] Kappassov, Z., Corrales, J. A., & Perdereau, V. (2015). Tactile sensing in dexterous robot hands. Robotics and Autonomous Systems, 74, 195-220.

[Seminara2019] Seminara, Lucia, et al. "Active haptic perception in robots: a review." Frontiers in neurorobotics 13 (2019): 467142.

[Pape2012] Pape, L., Oddo, C. M., Controzzi, M., Cipriani, C., Förster, A., Carrozza, M. C., & Schmidhuber, J. (2012). Learning tactile skills through curious exploration. Frontiers in neurorobotics, 6, 6.

[Prattichizzo2013] Prattichizzo, D., Chinello, F., Pacchierotti, C., & Malvezzi, M. (2013). Towards wearability in fingertip haptics: a 3-dof wearable device for cutaneous force feedback. IEEE Transactions on Haptics, 6(4), 506-516.

[Laurenzi2023] Arturo Laurenzi, Davide Antonucci, Nikos G. Tsagarakis, Luca Muratore, “The XBot2 real-time middleware for robotics,” Robotics and Autonomous Systems, Volume 163, 2023, 104379, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2023.104379.

[Muratore2020] L. Muratore, A. Laurenzi, E. Mingo Hoffman and N. G. Tsagarakis, "The XBot Real-Time Software Framework for Robotics: From the Developer to the User Perspective," in IEEE Robotics & Automation Magazine, vol. 27, no. 3, pp. 133-143, Sept. 2020, doi: 10.1109/MRA.2020.2979954.

[Laurenzi2019] A. Laurenzi, E. M. Hoffman, L. Muratore and N. G. Tsagarakis, "CartesI/O: A ROS Based Real-Time Capable Cartesian Control Framework," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 591-596, doi: 10.1109/ICRA.2019.8794464.

[Hogan1985] N. Hogan, “Impedance control: An approach to manipulation: Part I-theory". Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 107(1). https://doi.org/10.1115/1.3140702

[Muratore2023] L. Muratore, A. Laurenzi, A. De Luca, L. Bertoni, D. Torielli, L. Baccelliere, E. Del Bianco, N. G. Tsagarakis, “A Unified Multimodal Interface for the RELAX High-Payload Collaborative Robot” Sensors 2023, 23, 7735. https://doi.org/10.3390/s23187735

[Bertoni2022] L. Bertoni, L. Muratore, A. Laurenzi and N. G. Tsagarakis, "Task Driven Online Impedance Modulation," 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), Ginowan, Japan, 2022, pp. 865-872, doi: 10.1109/Humanoids53995.2022.10000215.