UNIVERSITÉ DU LUXEMBOURG

PhD-FSTM-2023-108
The Faculty of Science, Technology, and Medicine

DISSERTATION

Defence held on 28/09/2023 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN SCIENCES DE L'INGÉNIEUR

by

Alexej SIMETH
Born on 7 July 1990 in Cologne, Germany

AI-BASED COMPUTER VISION TO ENABLE ROBOTIC AUTOMATION IN HIGH MIX LOW VOLUME ASSEMBLY

Dissertation defence committee

Prof. Dr.-Ing. Peter Plapper, Dissertation Supervisor
Professor, Université du Luxembourg

Prof. Dr.-Ing. Karl Hofmann-von Kap-herr
Professor, Hochschule Trier

Prof. Dr.-Ing. Slawomir Kedziora,
Professor, Université du Luxembourg

Prof. Dr.-Ing. Rainer Müller
Professor, Universität des Saarlandes,
ZeMA Zentrum für Mechatronik und Automatisierungstechnik

Prof. Dr.-Ing. Markus Schäfer, Chairman
Professor, Université du Luxembourg

Prof. Dr.-Ing. Uzair Khaleeq Uz Zaman
Professor, National University of Sciences and Technology Pakistan,
CEME College of Electrical & Mechanical Engineering

Acknowledgements

I would like to express my sincere gratitude to all the individuals who have supported me throughout my journey to completing this PhD thesis. Their support, encouragement, and contributions have been invaluable, and I am deeply thankful to each one of them.

First and foremost, I extend my heartfelt appreciation to my esteemed supervisor, Prof. Plapper, for the opportunity to conduct this thesis. His support and continuous encouragement have been instrumental in shaping my PhD research. I am grateful to the esteemed examiners, Prof. Kedziora, Prof. Schäfer, Prof. Müller, Prof. Hofmann-von Kap-herr, and Prof. Zaman, for their commitment and dedication during the evaluation and the defence. Their constructive feedback has significantly enriched the quality of this thesis.

I would also like to thank my colleagues, especially Dr. Sri Kolla and Dr. Jeff Mangers, for their support and camaraderie, the meticulous review of my manuscript, as well as their friendship outside the university. Furthermore, I would like to thank Dr. Atal Kumar, Dr. Meysam Minoufekr, Thierry Decker and all other colleagues and student assistants for the valuable discussion, the support, the dedication, and the splendid time during my PhD.

I am deeply grateful to my parents, Jasmin and Jürgen, for their unending love, encouragement, and sacrifices. Their belief in my abilities has been a guiding light throughout my academic journey. Furthermore, I would like to thank my sister, Nastassja, for her support and encouragement and for always being there for me.

I am deeply grateful to my fiancé, Larissa, and her unwavering presence during my PhD. She stood by my side during the highs and lows and patiently endured my emotional ups and downs. Her unwavering support has been a continuous source of inspiration and motivation, and I am truly fortunate to have her as my tower of strength.

To everyone who has played a part, big or small, in completing this thesis, I offer my heartfelt appreciation. Your support has been pivotal, and I am honoured to have been surrounded by such wonderful people.

Alexej Simeth — Luxembourg, September 2023

Declaration

I, Alexej Simeth, declare that this thesis titled

> "**AI-based Computer Vision to Enable Robotic Automation in High Mix Low Volume Assembly**"

and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Luxembourg, the 30th of September, 2023

Place and Date

Alexej Simeth

Signature

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADAM	Adaptive Moment Estimation
AE	AutoEncoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Average Precision
AR	Augmented Reality
BBox	Bounding Box
CAD	Computer-aided Design
CET	Comité d'Encadrement de Thèse (Thesis Supervision Committee)
CHT	Circular Hough Transform
CMYK	Cyan, Magenta, Yellow, Key (Black)
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CONV	Convolutional
CS	Case Study
CV	Computer Vision
DFKI	Deutsches Forschungszentrum für künstliche Intelligenz (German Research Center for Artificial Intelligence)
DIN	Deutsches Institut für Normung (German Institute for Standardisation)

Dist	Distance
DL	Deep Learning
DOI	Digital Object Identifier
DSSD	Deconvolutional Single Shot Detector
EC	European Commission
EU	European Union
FAIR	Facebook AI Research
FC	Fully Connected
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
FPS	Frames per second
GUI	Graphical User Interface
HMI	Human-Machine Interface
HMLV	High mix low volume
HRC	Human-Robot Collaboration
HSV	Hue, Saturation, Value
IoU	Intersection over Union
ISBN	International Standard Book Number
ISEF	Infinite size Symmetric Exponential Filter
LED	Light Emitting Diode
Line-likel.	Line-likeliness
LfD	Learning from Demonstration

ln	Natural logarithm function
LoG	Laplacian of Gaussian
mAP	Mean Average Precision
MCS	Monte Carlo Simulation
MFEE	Multi Functional End Effector
ML	Machine Learning
MLP	Multi Layer Perceptron
MP	Mega Pixels
MTM	Methods-Time Measurement
MQTT	Message Queuing Telemetry Transport
P1-P4	Panel 1, ..., Panel 4
PhD	Philosophiae Doctor
PLC	Programmable logic controller
px	Pixel
QA	Quality Assurance
R-CNN	Region based Convolutional Neural Network
ResNet	Residual Network
RFID	Radio Frequency Identification
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue, Depth
ROI	Region of Interest
SGD	Stochastic Gradient Decent
SME	Small and Medium-sized Enterprise

SSD	Single Shot Multibox Detector
Std. Dev.	Standard Deviation
SVM	Support Vector Machine
TCP	Tool Center Point
TCP/IP	Transmission Control Protocol/Internet Protocol
TN	True Negative
TP	True Positive
TRL	Technology Readiness Level
VDI	Verein Deutscher Ingenieure (Association of German Engineers)
YOLO	You Only Look Once

1 Introduction

1.1 Background & Problem Statement

Small and Medium-sized Enterprises (SMEs) play an important role in the EU manufacturing ecosystem (Muller et al., 2022, p. 16). Looking at the Grande Region Luxembourg and the Luxemburgish industry, SMEs account for two-thirds of persons employed and value-added (non-financial industry) (European Commission, 2022b). In Germany, the situation is similar, but the shares of employment (58%) and value-added (42%) are lower (European Commission, 2022a). In such high-wage countries, SMEs need to remain competitive in a global market and face the difficulty of balancing cost-efficiency with maintaining high-quality products or services. Exactly those SMEs rely on a significant amount of manual operations, and the degree of automation is often low or non-existent (Downs et al., 2021; Karaulova et al., 2019; Kleindienst and Ramsauer, 2015). However, automation can lead to increased productivity, higher quality, increased delivery reliability, or reduced rework and scrap (Johansen et al., 2021a; Kinkel, 2009).

One reason for the lack of automation is the increasing product customisation and shortening of product life cycles, resulting in high product variability (Hu, 2013; Lu et al., 2020). In this high-mix and low-volume (HMLV) manufacturing environment, the product batch sizes are small or may equal one. This is especially true for the final assembly since the complete variance of every product must be covered. Thus, the business case for automation is often not given because the HMLV and the associated product changeovers decrease the refinancing potential (Franzkowiak et al., 2014; Gan et al., 2023). For large batch sizes, tools, positions, and other relevant process parameters are adapted to fit a specific product. However, this procedure is not economical for small volumes with high product variance and processes are often conducted manually (Burggräf et al., 2019). Hence, the SMEs in HMLV production scenarios require developing automation solutions compatible with multiple products, which maintain overall production flexibility (Downs et al., 2021; Flannigan et al., 2014; Gan et al., 2023; Grube Hansen et al., 2017; Johansen et al., 2021b; Kusuda, 2010; Schlette et al., 2019).

Furthermore, uprising challenges faced by SMEs intensify the need for new automation solutions. According to the European Commission, the scarcity of skilled staff and the costs of production or labour are rated as the most important by SMEs throughout EU-27 (Muller et al., 2022, p. 128). For SMEs, flexible automation can free employees, make them available for other tasks, and reduce the burden of personnel shortage.

For the flexible assembly automation of manual operations for highly individual products in HMLV, replacing human observations at various points with digital systems to start, execute, terminate, or monitor processes is necessary. Especially Artificial Intelligence (AI) and Computer Vision (CV) are expected to enable flexible automation since it is possible to deduct decisions from unknown multidimensional correlations in sensor data, which is critical for manufacturing highly customised products (Akerkar, 2019, pp. 19-32; Ren et al., 2022). Inspired by the vast development in both domains, this research focuses on applying learning-based CV methods to enable the automation of industrial HMLV processes focusing on joining in assembly.

1.2 Objective & Approach

The pursuit scientific objective of this research is to develop a multidisciplinary approach to leverage learning-based CV methods to automate assembly processes in SMEs operating in an HMLV environment. The aim is to integrate production engineering methodologies with CV and machine learning (ML) and deep learning (DL) techniques to enable the application of automation in this context. This PhD project is industry-initiated and conducted together with a partner from the Greater Region Luxembourg. The partner is producing HMLV products manually, and the main challenge is the high product variance ordered by its customers. The multidisciplinary approach shall facilitate the overall target of the industrial project, i.e., to enable flexible automation of the assembly of lightweight panels.

A systematic procedure is proposed to achieve the research objectives involving the following steps. The first step analyses generic manual assembly tasks to identify the relevant parameters requiring visual determining. Key characteristics and requirements can be identified by studying the manual processes, laying the foundation for subsequent assembly process automation development. In the next step, learning-based CV models are selected and employed to determine the identified process-relevant parameters critical for automation with sufficient accuracy. These models leverage ML and DL techniques to automatically derive decision rules based on data. By training the CV models on relevant datasets, they can adapt and generalise well to handle variations and challenges encountered in assembling HMLV products. Then, the robust and flexible CV models are specifically adapted to the identified process parameters. These models accurately estimate the desired parameters based on visual inputs. The CV models are refined through optimisation to achieve the required accuracy and reliability in diverse assembly scenarios and environmental conditions. Their performance and robustness are examined in different experiments and compared to conventional CV methods commonly applied in the existing industrial processes. Finally, the procedure and the developed models are validated with the industrial partner's assembly processes on a technology demonstrator.

1.3 Structure of the Report

After this introductory chapter, the remainder of the thesis is structured as follows. Chapter 2 summarises the fundamentals of the three domains assembly, CV, and AI, with a focus on ML and DL. In Section 2.1, relevant terms of assembly, assembly tasks and automation are introduced, and the most relevant definitions for HMLV, changeability, and flexibility are provided. Image representation, processing and filtering techniques to enhance and extract specific image features are covered in Section 2.2, while Section 2.3 focuses on ML and DL methods for image analysis. Finally, Section 2.4 present an overview of the state of the art for the development and application of learning-based CV models targeting assembly automation in HMLV and identifies existing research gaps.

In Chapter 3, the research objectives are concretised in Section 3.1 based on the identified gaps. Then, the applied research methodology in this thesis is detailed in Section 3.2. Furthermore, the scope of the analysis and procedure development and their requirements are specified in Section 3.3.

Chapter 4 covers the procedure development to identify process-relevant parameters and to determine these by applying learning-based CV models. Based on the analysis of typical joining motions in Section 4.2, the process-relevant parameters critical for automation are defined in Section 4.3. Then, the actual procedure is detailed in Section 4.4.

Different types of process-relevant parameters are identified, and learning-based CV models are developed in Chapters 5 and 6 following the defined procedure. While Chapter 5 covers the analysis of location-dependent parameters, Chapter 6 includes time-dependent parameters. The industry partner's assembly processes are used as a reference and are defined in Sections 5.1 and 6.1, respectively. The different process-relevant parameters for the reference processes are identified in Sections 5.2 and 6.2. Finally, Sections 5.3, 5.4, and 5.5 include the experimental development and analysis of different conventional and learning-based CV models for location-dependent parameters. The equivalent for time-dependent parameters is presented in Sections 6.3 and 6.4.

In Chapter 7, the models are integrated into a technology demonstrator. The process-relevant parameters are determined with several learning-based CV models, and the whole assembly process is automated using an industrial robot with different tools. Furthermore, the procedure is applied to additional industrial processes from other companies.

Finally, Chapter 8 summarises the thesis and provides an outlook for further research.

2 Fundamentals and State of the Art

This chapter targets to establish a mutual understanding of the fields of consideration for this thesis. Therefore, the relevant topics and terms are presented and put into context. In the beginning, an overview of assembly and assembly automation with a focus on High-Mix Low-Volume (HMLV) and its implications and different types of flexibility is given in Section 2.1. The following sections describe the subject areas of Computer Vision (CV), Machine Learning (ML), and Deep Learning (DL), which are frequently combined in both research and industry. Section 2.2 defines the foundations of CV, the main tasks relevant to this research and different concepts of image processing and filtering. ML- and DL-based concepts for image analysis are specified in Section 2.3, and the most prominent learning-based models are explained. After setting the scene, current approaches to enabling and implementing automation in HMLV assembly processes are compared to identify the research need in Section 2.4.

2.1 Assembly in High-Mix Low-Volume

2.1.1 Assembly & Assembly Automation

Assembly, according to the VDI Guideline 2860, is defined as the set of all processes that serve the assembly of geometrically defined bodies (VDI 2860, 1990). Industrial products are typically composed of individual parts joined with various production technologies at different points in time and assembled into a product of higher complexity during the assembly process (Lotter, 2012b, p. 1). The main functions of assembly include joining (DIN 8593), material handling (VDI 2860), inspection (VDI 2860), adjustment (DIN 8580), and specific special operations such as oiling or deburring (Lotter, 2012b, p. 2). Müller et al. expand the main functions with the category commissioning to consider the increasing share of mechatronic functionalities of assembled products (Müller et al., 2011). Wiendahl et al. classify assembly processes as all procedures that serve to join parts, assemblies, and final products (Wiendahl et al., 2009, p. 158). They are divided into primary processes that add value during assembly and secondary processes comprising all other efforts required without adding value to the product. According to Lotter, the main primary operations are the joining processes, given in Figure 2.1 (Lotter, 2012b, p. 49). Overall, the assembly is an essential part of the production, as the assembly process can account for up to 70% of the manufacturing costs for many products (Lotter, 2012b, p. 5).

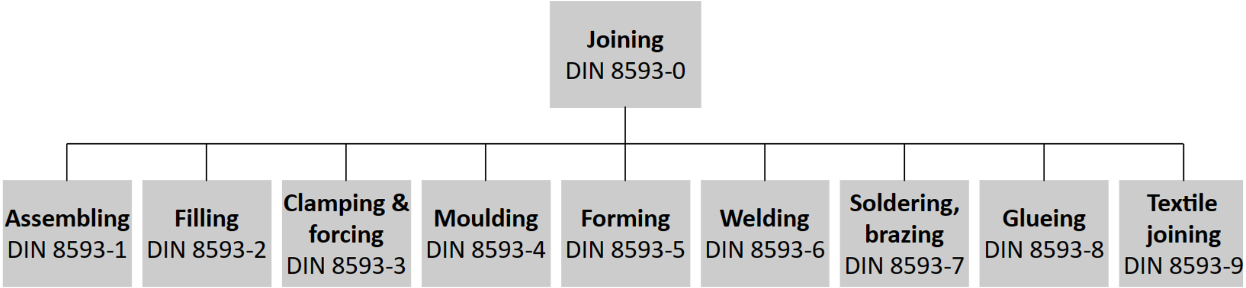


Figure 2.1: Joining tasks as defined in DIN 8593.

The organisational structure and arrangement of individual workstations determine an assembly system's structural layout and functioning. Various work organisation principles for assembly systems exist. Generally, work organisation is attributed to five organisational forms, with different variations of these forms found in companies based on their specific requirements. These organisational forms include single-station and job shop assembly, group assembly, line and paced assembly, and flow assembly. A key differentiating factor is the movement characteristic of the assembly objects and workstations. In the first two forms, the assembly object remains stationary while the assembly object is moved from line assembly onwards. Workstations are stationary in single-station, job shop, and line and paced assembly. In contrast, the workstations are mobile in group and flow assembly, requiring workers to change assembly stations. In producing customer-specific products, certain technical and organisational prerequisites are essential. This thesis focuses on the technological aspects. (Eversheim, 1989, p. 176; Fischer, 2013, p. 604)

The selection of an organisational form and assembly system highly depends on the number of units of a product or variant intended to be produced over time. A distinction is made between single-unit, batch, and mass production. However, there are no universal thresholds to differentiate between these forms, and the boundaries between them are vague, as the classification also depends on the specific product. Generally, single-unit production refers to a product's design and one-time manufacturing without planning for its repeated production. However, the product may be produced intermittently at longer intervals. Depending on the frequency, single-unit producers and order-based producers are distinguished. Order-based producers refer to cases where production occurs based on specific orders rather than maintaining inventory (small batches). Small and medium-sized enterprises (SMEs) predominantly represent both categories. (Steinbauer, 2012, p. 30)

Assembly automation
The process of progressively replacing human activities with functions performed by artificial systems (automata) is called automation (Hesse, 2000, p. 2). By automation, facilities are equipped to operate fully or partially without human intervention (DIN V 19233, 1998). The degree of automation represents the proportion of automated functions among all functions (Hesse, 2000, pp. 2–12). An automation task is solved by extending a technical system with an automation device, which combines various process interfaces, such as sensors and actuators, with communication systems and an automation controller equipped with suitable hardware and software as depicted in Appendix A1.1 (Jelali, 2013). A sensor converts physical, chemical, or biological quantities into electrical signals, amplified and normalised as a standard unit signal to be processed by a microcontroller. The signal processing in the controller results in the execution of actions, such as open- or closed-loop control tasks by an actuator. These automation devices aim to selectively influence process parameters by changing energy, mass, and information flows. (Jacques and Hansen, 2010, pp. 103–162)

The difficulty of automating assembly, in terms of technical implementation, primarily stems from the complexity of handling operations and joining processes involved. Factors such as accessibility to the part, part geometry, joining location, material stiffness, or the potential for interlocking of assembly parts influence this difficulty (Hesse, 2012, pp. 9–10). Joining processes that require a simple primary movement is considered to be easily automatable. However, the decisive criteria for assembly automation still are product volume and product variance (Feldmann, 2013, p. 20). In addition, the decision to automate assembly is influenced by time duration, reliability, accuracy, and cost of assembly operations. Overall, the degree of automation in assembly is relatively low, especially for SMEs (Jeschke, 2015; Hesse, 2012, pp. 195–196).

A range of hybrid forms exists between manual and fully automated assembly systems. An assembly system is called a hybrid when both manual and automated assembly steps are present. The hybrid system with a combination of manual workstations and automated

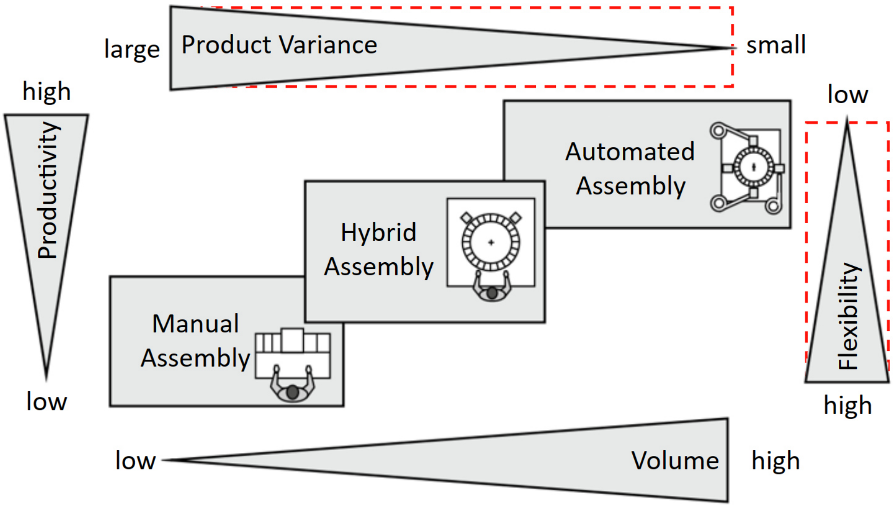


Figure 2.2: Application areas of manual, hybrid, and automated assembly concepts adapted from (Lotter, 2012c, p. 168). The dashed red lines indicate the required increased flexibility and product versatility for automated systems.

stations is positioned between manual and fully automated assembly regarding volume, productivity, product variance, and flexibility (s. Figure 2.2) (Lotter, 2012c, p. 167). Different products or product variants are assembled in any order by integrating programmable handling devices, joining equipment, and inspection devices. Typically, industrial robots are used as programmable handling or joining devices in flexible assembly systems. According to Lotter, flexible assembly systems must possess the ability to assemble a complete product family, adapt to other products, and be reusable. However, the costs of the basic equipment of flexible assembly systems, such as industrial robots, control systems, and workbenches, constitute only a portion of the required investments. Depending on the degree of automation and product complexity, the costs of the periphery, including feeding elements, fixtures, material transport, and gripping systems, can exceed the basic costs by far. As a result, semi-automation or hybrid systems often present an economical alternative to full automation. (Lotter, 2012a, pp. 273–274)

2.1.2 High-Mix Low-Volume

HMLV manufacturing describes a production involving many diverse products in relatively small quantities. High-mix refers to the variety of products produced, while low-volume refers to the small quantity of each product produced. No uniform definition exists for HMLV or when production reaches the state of HMLV. Most commonly, HMLV describes a scenario in which companies are confronted with an increasing demand for customised products (Hu, 2013; Lu et al., 2020). The variety of products with variable demands leads to small batch sizes with low repetition. In consequence, the volumes and lifecycles of each individual product decrease. Overall, the higher amount of parts variations increases the complexity of assembly for the players in such an HMLV environment. (Downs et al., 2021; Gan et al., 2023; Johansen et al., 2021a; Karaulova et al., 2019; Kusuda, 2010; Nagashima and Katsura, 2014; Schlette et al., 2019)

Another aspect connotated with HMLV is the type of affected companies. The majority of enterprises active in an HMLV environment are typically SMEs (Downs et al., 2021; Grube Hansen et al., 2017; Karaulova et al., 2019; Ong et al., 2020; Onstein et al., 2020). As mentioned in the introduction of this thesis (s. Chapter 1), in Europe and specifically in the Grand Region, SMEs represent >99% of all companies and account for >60% of employment and >50% of overall generated value-added (Muller et al., 2022, pp. 16–22). Moreover, these SMEs are often specialised and operate in a niche. One of the challenges of HMLV manufacturing is the need to manage many unique products and processes efficiently while maintaining high levels of quality and customisation. Therefore, a prominent assembly organisation is the job shop assembly (Bohnen et al., 2013; Gan et al., 2023), combined with a significant amount of manual operations. (Downs et al., 2021; Grube Hansen et al., 2017; Karaulova et al., 2019; Nagashima and Katsura, 2014; Schlette et al., 2019)

This is especially challenging for SMEs in high-wage countries, as these companies face a trade-off between the increased output and productivity of automated assembly systems and the flexibility and versatility of manual assembly (Johansen et al., 2021a; Schlette et al., 2019). This dilemma is depicted in Figure 2.2, and the red lines indicate the required increase in flexibility and product versatility for automated systems. Brecher et al. define this more general as the polylemma of production with the opposing targets of economies of scale vs scope. The objective is to resolve the dichotomy by increasing flexibility in product and production while simultaneously maintaining the series manufacturing costs level (s. Appendix A1.2) (Brecher et al., 2011, pp. 21–23).

Further aspects limit the application of automated systems in HMLV assembly. First, the assembly is usually the last step in manufacturing and must cover the whole range of variants, including their tolerances from prior processes. The small product volumes and high program changes do not achieve reasonable payback periods for initial investments. Another prohibiting factor is the costs and skills associated with reprogramming systems to new or changed products. Lastly, many SMEs struggle with adopting Industry 4.0 and digitisation technologies which can be a prerequisite for the targeted automation (Kolla, 2022). Overall, many factors impact the introduction of automated systems in HMLV assembly, and maintaining the required flexibility is one key challenge. (Downs et al., 2021; Grube Hansen et al., 2017; Karaulova et al., 2019)

2.1.3 Flexibility & Changeability

The ability of a company or a production system to respond to changes in external and internal influences and adapt its business processes, market performance, and production capabilities is commonly referred to as changeability. Within the context of production systems, flexibility and adaptability are often distinguished, which share a general understanding but have multiple definitions. (Wiendahl et al., 2009, pp. 115–129)

Nyhuis et al. define flexibility as the ability to adapt quickly and with minimal effort to changing influencing factors. The range of action, or flexibility corridor, for these short-term reactions is limited and predefined (Kluge, 2011, p. 16). For instance, flexible assembly systems can assemble various products within a product family without reconfiguring or accommodating changes in production volumes within a predetermined range. In contrast to flexibility, adaptability describes the ability to enable responses to changes outside the flexibility corridor (Müller et al., 2011, pp. 425–426). Ensuring adequate responsiveness of a production system requires specific characteristics, referred to as adaptability enablers. These include universality (e.g., variant flexibility), mobility (e.g., machines on wheels), scalability (e.g., flexible working time models), modularity (e.g., plug & produce modules), and compatibility (e.g., standardised software interfaces) (Wiendahl et al., 2009, p. 126). (Nyhuis et al., 2008, pp. 24–27)

According to Wiendahl et al., it is helpful to classify the changeability of a production company into different classes that follow the classical hierarchy of a factory and its products from the factory planning perspective. Accordingly, different levels of production are assigned different types of changeability. In this context, adaptability refers to the tactical ability of an entire factory to reconfigure and/or change production capacity to switch to another (usually similar) product family. For this thesis, relevant cell and workstation levels require flexible, reconfigurable systems with change-over ability. It can be achieved through the design of the product or the process, which are, however, not independent of each other.

The latter refers to the ability of a machine/workstation to perform defined work operations for known products at any desired time with minimal effort and in the shortest possible time (Plapper et al., 2011, p. 6). (Wiendahl et al., 2009, p. 132)

The terms are, however, primarily applied in the German literature. Regarding automation in HMLV, the term flexibility is most prominent but used with various meanings. De Ginste et al. identified 15 different types of flexibility on the workstation level. The authors define workstation flexibility as a multidimensional concept and divide the flexibility into top-level dimensions (product, modification, process and volume flexibility) and base dimensions (labour, machine, material handling, material and operation flexibility). The latter is linked to basic resources such as personnel, equipment, and material. An assembly workstation is thus considered more flexible if it can produce a range of products and new variants simultaneously, is unaffected by product modifications, and can change or adapt the process to the product mix. This flexibility needed in HMLV poses a challenge to the industrial implementation of automated assembly (Johansen et al., 2021b). (De Ginste et al., 2019)

In this thesis, the term flexibility is also used on the workstation level in the sense of the definition by De Ginste et al. In the following sections, an overview of CV and AI in image analysis is provided.

2.2 Computer Vision & Digital Image Processing

This section introduces the subjects of CV and digital image processing. The latter represents the theoretical basics for any image analysis. The application of Artificial Intelligence (AI)-based techniques to solve CV tasks has rapidly evolved in recent years. However, it is discussed later in Section 2.3.

2.2.1 Introduction to Computer Vision

CV is an interdisciplinary scientific field that encompasses the analysis of images or videos to extract meaningful information to comprehend and represent the underlying physical world. By leveraging computational algorithms and techniques, CV aims to mimic human visual perception and cognition. The application of CV algorithms is diverse and encompasses tasks such as stereo matching, object tracking, and image restoration. While advancements in AI, particularly DL and convolutional neural networks (CNNs), have demonstrated superior performance in specific areas such as object detection and recognition, there are traditional CV techniques that continue to excel in specific domains, including panoramic vision and 3D reconstruction. These techniques provide valuable insights and solutions for various practical challenges in CV research and applications. (Huang et al., 2021; Szeliski, 2022, pp. 3–22)

For this thesis, the relevant CV tasks belong to the application area of object detection. The primary tasks are given in the following and visualised in Figure 2.3.

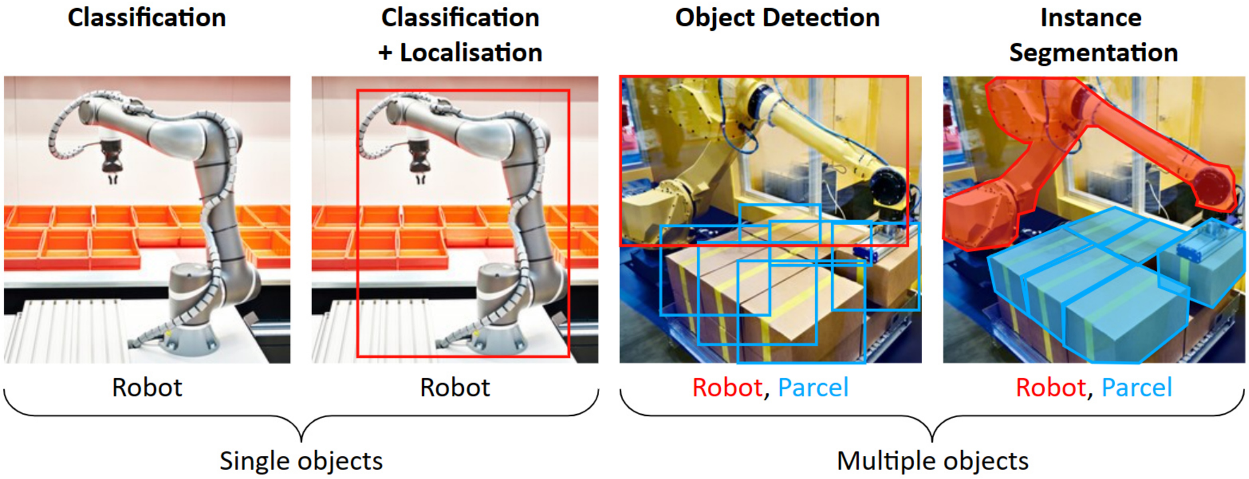


Figure 2.3: Comparison of the CV tasks image classification, object detection, and instance segmentation for single and multiple objects on an image (Huang et al., 2021; Meena et al., 2020). Image source: (Adobe, 2023).

Image classification describes the classification of an observed image into one category from a set of two or more previously defined categories. The main target is to find a mathematical decision rule that performs the classification based on the properties given by a training data set. Besides classification with a single class per image (s. Figure 2.3), multilabel classification is also possible. For example, assigning multiple diseases to a single image is necessary for classifying medical images. (Huang et al., 2021; Meena et al., 2020)

Object detection combines an image classification task and the localisation of a classified object on the image. The latter finds the area on the image where the object is located. A boundary, the so-called bounding box, is drawn around it to highlight the region. Multiple objects of different classes are detected on a single image depending on the task. (Huang et al., 2021; Meena et al., 2020)

Instance segmentation relates to the classification of individual objects within an image or video at the pixel level into a group of objects in a given picture. It provides a unique segmentation mask for each instance of an object, enabling precise localisation and differentiation of multiple objects of the same class. (Huang et al., 2021; Meena et al., 2020)

The basis for the CV tasks mentioned above is the processing of digital images, detailed in the next subsection.

2.2.2 Digital Image Processing

Image representation

The processing of image data resulting in either images or several features is formally known as Digital Image Processing. The term image is used in different areas. In math, for instance, an image describes the set of output values of a function. In photography, an image is the optical reproduction of the environment reduced to two dimensions. Combining both areas leads to a mathematical description of an image as a function, which assigns to every element in a domain a colour value of the colour space. (Bredies and Lorenz, 2011, pp. 1–2)

The domain may be discrete or continuous. To describe an image, vector graphics or raster images are applied depending on image complexity. A raster graphic is suitable for describing a more complex image, such as a photograph. Scanning is the process of converting an analogue, continuous image into a digital raster graphic. This form of digitisation is characterised by the domain's rasterisation and the codomain's quantisation. The domain is a raster, given by the arrangement of the image points, so-called pixels (px), in regular grids. Among others, rectangular, triangular, or hexagonal grids are suitable for this purpose since these shapes are characterised by an equal number of edges and immediate neighbours in a grid arrangement. However, the rectangular grid has prevailed for image processing due to its clearer mathematical description as a matrix. The elements' range in the codomain depends on the information assigned to the pixels, i.e., the colour. (Bredies and Lorenz, 2011, pp. 1–2; Süße and Rodner, 2014, pp. 3–15)

Colour space

The pixel's colour information is taken from a colour space, which can be understood as a representation of a colour model. Colour models serve as the practical representation and quantification of the characteristics of colour in an abstract colour system such as RGB, CYMK or HSV. The RGB colour model is based on additive colour mixing and is built upon three primary colours: Red (R), Green (G), and Blue (B). This model is utilised in fields where the generation of colour involves the emission and superposition of light with different wavelengths. In this context, darkness is the absence of light in any form. When light consists of equal proportions of the emitters of the three primary colours, it appears white. An alternative to the RGB colour model is the HSV (Hue, Saturation, Value) space that represents colours in terms of their hue (colour tone), saturation (intensity or purity of colour), and value (brightness or lightness). Digital colour images are commonly stored in the RGB space with three colour channels. Grayscale images use one channel for the intensity of each pixel. A black-and-white image can be described in binary terms. Here, the pixels are assigned either a 0 for the colour black or a 1 for white. (Bredies and Lorenz, 2011, pp. 3–5; Brown, 2019, pp. 86–106; Szeliski, 2022, pp. 87–96)

Image preprocessing & filtering

Image preprocessing aims to eliminate redundant information from images while preserving the relevant details and enhancing other important attributes essential for subsequent processes. Examples of image data processing into a form suitable for further analysis include noise reduction, sharpening and smoothing, brightness and colour correction, resizing or geometric transformations. The design of image processing stages is crucial in achieving satisfactory outcomes for various CV applications. (Harakannanavar et al., 2019; Szeliski, 2022, p. 109; Thanh et al., 2019)

Image filtering refers to modifying or enhancing an image by applying a mathematical operation, known as a filter, to each pixel or neighbourhood of pixels in the image. The filter alters the pixel values based on predefined rules or calculations, allowing for various types of image transformations. Image filtering techniques are applied for preprocessing as well as feature extraction. Standard filters in preprocessing are the Gaussian filter (Deng and Cahill, 1994), average filter (Qinlan and Hong, 2009), Laplacian filter (Shen et al., 2006), median filter (Eng and Ma, 2000), Gamma transform (Solomon and Breckon, 2011, Cha. 4), or binarisation (Otsu, 1979). Besides filtering in the spatial domain (pixel domain), images are also analysed in the frequency domain by applying a Fourier transform. (Gonzalez and Woods, 2018, Chapters 3-4; Szeliski, 2022, Chapter 3)

2.2.3 Image Features & Feature Extraction

Distinctive and meaningful patterns in images are known as image features. They represent local or global properties of the image, which can be amplified and extracted using, for instance, the discussed preprocessing and filtering methods. Some image feature examples are depicted in Figure 2.4 and explained in the following.

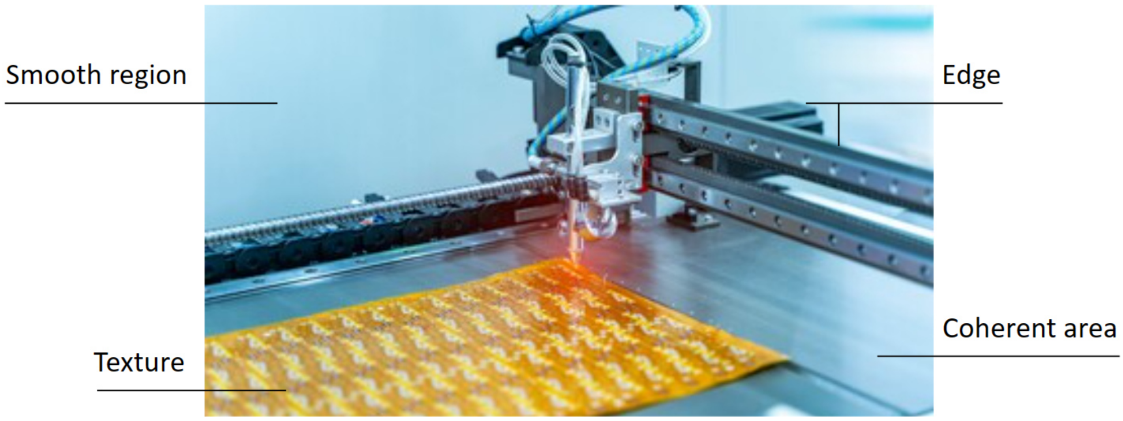


Figure 2.4: Features of an image. Image source: (Adobe, 2023).

An ideal edge can be expressed mathematically by the discontinuity of the spatial grey value function of the image plane (Burger and Burge, 2015, p. 125). Discontinuity is understood as a large change of intensity in one direction, which the gradient of the grey value function can detect. If the gradient is high in more than one direction, there is a high probability that a corner is present. Reflection edges show properties of the imaged object, while illumination edges are caused by lighting setup (Jähne, 2002, pp. 333–355). If an area appears uniformly coloured, it is called a smooth region. Such regions are separated from others by edges. A continuous gradient of brightness under uniform illumination remains a smooth region and allows considerations of the curvature of the imaged object. A periodic texture is characterised by its regularity in one or more directions. Textures such as the fine texture of woven fabrics can exhibit such periodicity. Areas of the same orientation, such as wood grain, are called coherent. In this case, approximately parallel lines occur more frequently. (Bredies and Lorenz, 2011, p. 5; Szeliski, 2022, p. 434)

Gradient edge detection

Derivative filters are used to detect discontinuities in images. The filters are designed to give no response in smooth regions of an image and to return significant values at points of abrupt changes in intensity level or texture, indicating an end or a start of a region. Such identification of jumps in intensity values is also known as edge detection. Classical gradient edge detectors apply the matching of local image segments with specific edge patterns. Edges are detected by searching for maxima or minima in the image's first derivative (gradient). Popular edge detection operators are Prewitt, Sobel, or Kirsch, all built of a 3x3 Kernel with different weights. An example of edge detection with a Sobel kernel is shown in Figure 2.5 b) with the input image being a). They are applied directly to identify edges or within an algorithm such as Canny or ISEF, which include additional filters or logic to enhance the detection result. However, the gradient edge operator remains the basis for extracting the edge feature. For further information, see Appendix A1.3. (Chaple et al., 2015; Joshi and Koju, 2012; Sharifi et al., 2002)

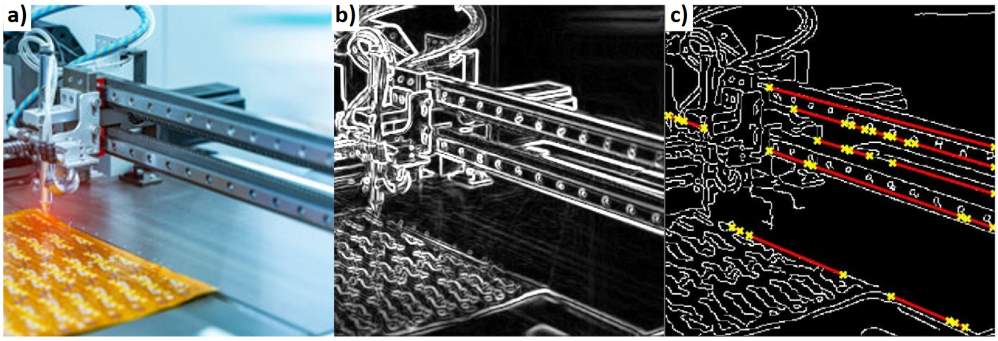


Figure 2.5: a) Input image. b) Gradient Edge detection with Sobel operator. c) Linear Hough transform based on Canny edge detection with detected lines highlighted in red. Image source: (Adobe, 2023).

Hough transform

The Hough transform is another technique to extract geometric shapes as lines or circles used in CV and image analysis. It is also effective in high noise, discontinues, or partial occlusion scenarios. The core of the Hough transform is converting an input image into a parameter space, the Hough space, where each point represents a potential model parameter. Model parameters depend on the shape of the objects to be detected. For example, parameters for detecting lines are slope and intercept, and for circles (circular Hough transform), radius and centre coordinates. By accumulating votes from edge pixels that align with the selected model parameters, the Hough transform identifies significant peaks in the parameter space. The peaks indicate the presence of a geometric object in the input image. Hough transform implementations usually include an RGB-to-greyscale conversion followed by Gaussian blur and edge detection. The resulting binary edge image is the input to the actual Hough transform. Additional parameters define thresholds for the accumulator, defining how many votes are required to consider a peak as an object. Figure 2.5 c) shows the output of a linear

Hough transform with the identified edges highlighted in red. Due to the choice of thresholds, some visible straight edges are not counted as linear objects. For further information, see Appendix A1.4. (Ballard et al., 2016; Duda and Hart, 1972; Illingworth and Kittler, 1987; Mukhopadhyay and Chaudhuri, 2015)

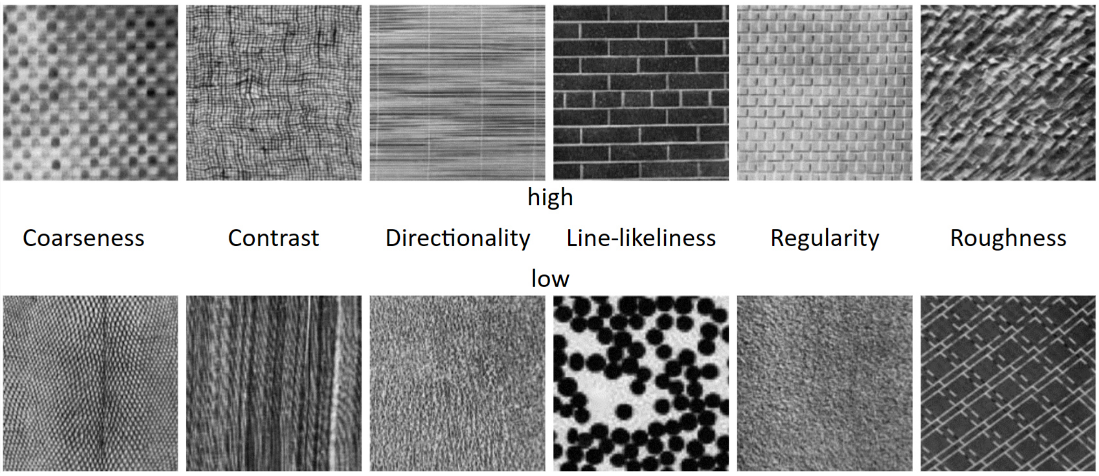


Figure 2.6: Images examples with a high (upper) and low (lower) value for the Tamura texture descriptor based on (Chi et al., 2019).

Tamura textural features

Textural feature descriptors play an important role and are widely applied in CV tasks. The textural feature descriptors proposed by Tamura et al. correlate highly with human visual perception and are proven to describe texture human-like most accurately and efficiently (Tamura et al., 1978). By using statistical measures, the Tamura features compute the spatial variations, intensity distributions, and structural patterns within an image (Chi et al., 2019). The descriptors are based on six textural features: coarseness, contrast, directionality, line-likeness, regularity, and roughness (Tamura et al., 1978). Example images with strong and weak expressions for each feature are given in Figure 2.6. Coarseness represents the image granularity and is a measure of the micro-texture of an image. Contrast measures the variation of grey-level intensity of the pixels and is a measure of image quality. Directional patterns, like horizontal, vertical, or diagonal lines, are characterised by feature directionality. Line-likeness is characteristic of texture that is composed of lines. A group of edge pixels is considered a line when the pixels have the same edge direction. The descriptor regularity characterises the sum of texture variation in an image. It is measured as the sum of the variation of the four previously listed Tamura features. Roughness emphasises the effects of coarseness and contrast and is approximated as the sum of both features. (Chi et al., 2019; Karmakar, 2017; Tamura et al., 1978)

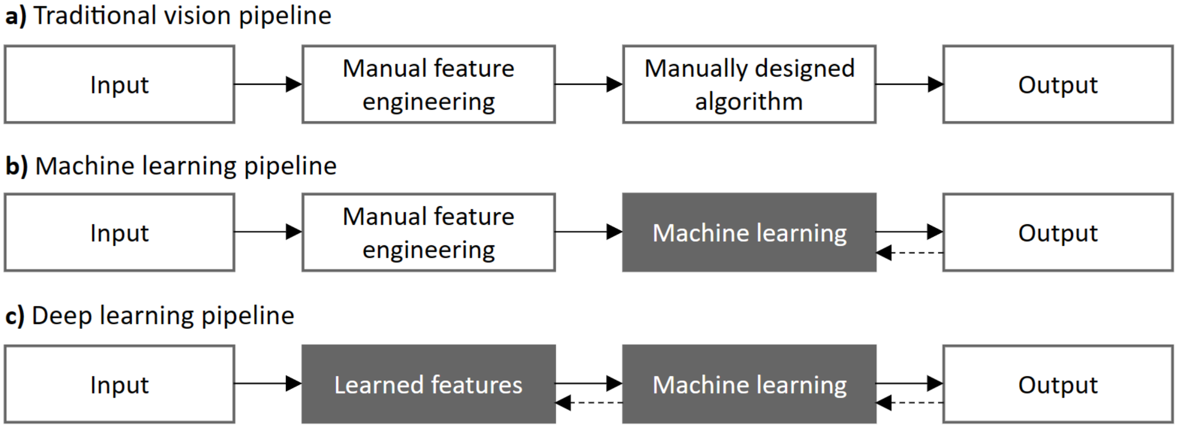

a) Traditional vision pipeline
Input
Manual feature engineering
Manually designed algorithm
Output
b) Machine learning pipeline
Input
Manual feature engineering
Machine learning
Output
c) Deep learning pipeline
Input
Learned features
Machine learning
Output

Figure 2.7: Differences between traditional, machine learning, and deep learning vision pipeline. Adapted from (Goodfellow et al., 2016, p. 10; Szeliski, 2022, p. 238).

Differences in feature extraction

Image features can be classified based on pixel properties such as brightness, gradient, colour, or texture. The selection, transformation, combination, and manipulation of different features are the foundation for image analysis and the extraction of meaningful information. Figure 2.7 highlights the main differences between a traditional CV pipeline and pipelines for ML- and DL-based CV. In the classic pipeline, all processing steps are designed manually. Feature and algorithm engineering requires domain knowledge and a deep understanding of the CV problem to identify relevant variables, extract useful information, and derive new features that capture the underlying patterns. The ML pipeline still requires engineered features but solves the data analysis and processing by automatically recognising the underlying structural relationships within the data. The recognition process relies on algorithmic techniques that acquire knowledge from the dataset. With DL, the whole pipeline from input pixels to output information consists of learning algorithm components, including mid-level representations, directly from the training data. End-to-end learning indicated by dashed arrows in Figure 2.7 from input pixels to output information is part of the vast success of the DL pipeline since it omits the large amounts of human intervention and ingenuity necessary to develop both features and algorithms (Glassner, 2021, p. 4). Additionally, the more sophisticated features learned by the model increase its robustness, i.e., they make the model less affected by noise. (Akerkar, 2019, pp. 19–32; Szeliski, 2022, pp. 10, 237)

The basic concepts of ML and DL are presented in the following sections, focusing on their application in CV.

2.3 Machine Learning & Deep Learning

2.3.1 Overview

The German Research Center for Artificial Intelligence (DFKI) defines AI as the property of an IT system to exhibit human-like, intelligent behaviour. This includes imitating human abilities such as logical thinking, learning, planning, and creativity (DFKI, 2017, pp. 28–29). Figure 2.8 displays the relation between AI and relevant areas for this thesis. Whereas ML and DL are subdomains of AI, Robotics and CV apply AI methods (Huang et al., 2021). With the advancement in computational power and AI algorithms, vision-based approaches have improved tremendously to solve complex problems that have had no effective solution in the past (Jiao et al., 2019).

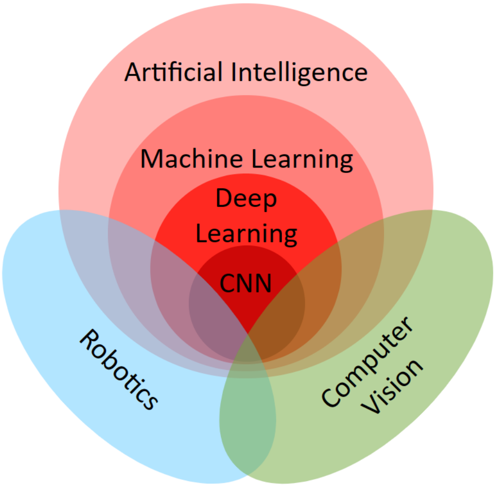


Figure 2.8: Relation between Computer Vision, Artificial Intelligence, Machine Learning, and Deep Learning. Adapted from (Goodfellow et al., 2016, p. 9; Huang et al., 2021).

In general, ML is one of the areas of AI which involves the development of computational approaches to make sense of data automatically. These algorithmic methods automatically recognise structures in given data sets. The technology leverages the insight that learning is a dynamic process made possible through examples and experiences instead of predefined rules. Like a human, a machine can retain information and improve over time. Common tasks in ML are clustering, dimensionality reduction, classification, and regression. (Akerkar, 2019, pp. 19–32; Glassner, 2021, pp. 3–13)

DL is a subfield of ML that has contributed to most of the recent success of the ML field.

The architecture of DL algorithms is generally based on applying multi-layer Artificial Neural Networks (ANN). ANNs comprise a network of synthetic (digital) neurons that process input data and send the resulting output data to other neurons. With multiple processing layers, ANNs learn representations of data with multiple levels of abstraction by adapting the neurons' weights. The backward propagation of errors from the output to the input layer using the Backpropagation algorithm is the fundamental process behind the weight optimisation of ANNs (Rumelhart et al., 1986). The success of DL is mainly rooted in the fact that DL networks adopt a more general learning principle characterised by multiple levels of composition, which has the key advantage of automatic feature extraction (Huang et al., 2021). Especially with the publication of KRIZHEVSKY ET AL., the application of deep convolutional neural networks for vision tasks increased drastically, and it is now one of the preferred architectures for multiple CV problems (Krizhevsky et al., 2012). Further information on ANNs is provided in Appendix A1.6. (Akerkar, 2019, pp. 33–40; LeCun et al., 2015; Nielsen, 2015)

A distinction can be made between the following types of learning:

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

Supervised Learning describes the learning from training data. The training dataset consists of a set of data whose class assignment is known, so-called labelled or sorted data. In unsupervised learning, an algorithm finds groups or classes in the data without specifying any classification criteria. This automatic finding of similarities in the data structure is used for cluster analysis or dimensionality reduction. The mixed-form semi-supervised learning uses a combination of a small portion of labelled data and a large portion of data whose class membership is unknown for learning. The default of assigning a few sample data to classes leads to higher accuracy than unsupervised learning at a lower time cost than supervised learning. Reinforcement learning, on the other hand, describes how an abstract entity, i.e., an agent, learns to make optimal decisions through reinforcing or contradicting feedback. The goal is to develop a long-term strategy to maximise a utility/reward function. Therefore, the agent acts in an environment and learns through trial and error to discover the most effective strategies or policies for achieving its goals. (Akerkar, 2019, pp. 19–32; Glassner, 2021, pp. 3–13; LeCun et al., 2015)

2.3.2 Convolutional Neural Networks

One type of ANN is convolutional neural networks (CNN), widely applied in image analysis and vision tasks. A CNN-like structure was initially published by Fukushima and popularised by LeCun et al. (Fukushima, 1980; LeCun et al., 1998). Today, they are the backbone of most state-of-the-art object detection and segmentation models. The central concept of CNNs is convolving filters of different sizes over the entire image to capture local patterns and spatial relationships. This enables extracting and recognising complex visual patterns, objects, or structures. (Jiao et al., 2019; LeCun et al., 2015)

In principle, CNNs consist of single or multiple blocks of convolutional (conv) layers and pooling layers followed by "single or multiple blocks of fully connected layers and an output layer" (Ghosh et al., 2020, p. 3). A standard architecture of a CNN with two conv layers, two pooling layers and two fully connected layers is depicted in Figure 2.9. The input image is represented as a 2D matrix where each element represents a pixel of the image. For example, an RGB image with three colour channels has three matrices, and its representation is often referred to as a 3D tensor. The height and width of the image must fit the CNN input size. (Akerkar, 2019, p. 37; Glassner, 2021, pp. 429–453; Szeliski, 2022, pp. 291–296)

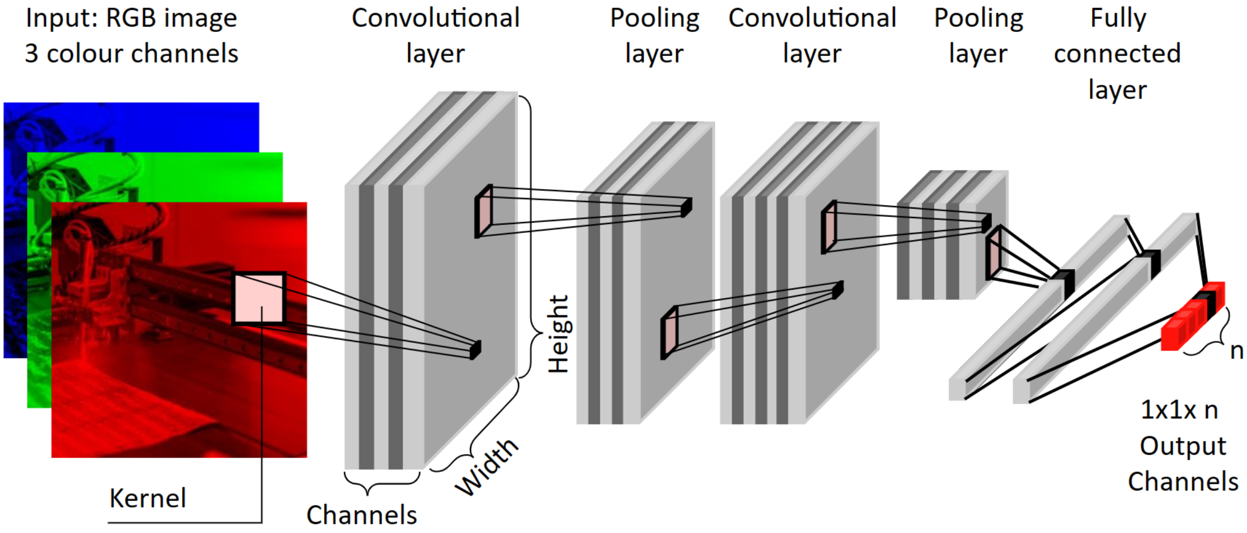


Figure 2.9: Architecture of a convolutional neural network.

Convolutional layer

The elementary building block of a CNN is the convolutional layer, which consists of a set of filters or kernels. The filters convolve channel-wise over the input and generate feature maps. Each conv layer is characterised by the size of the kernel and the height, width, and number of generated feature maps. Convolutional layers are the main learnable compounds responsible for learning representations of visual features by updating filter weights and biases during training. This enables the network to understand complex patterns in images. The CNN has the advantage of sharing the weights of the filters across the whole input image, which saves memory. Layers at the beginning of the CNN extract low-level features such as edges, whereas layers towards the end recognise high-level features. Examples of such features are the learned characteristics of ears, eyes, and nose in face detection. (Akerkar, 2019, p. 37; Almabdy and Elrefaei, 2019; Glassner, 2021, pp. 429–453)

Pooling layer

Pooling layers downsample the feature maps by summarising or reducing their spatial dimensions. The most common pooling technique is max pooling which selects the maximum value within each pooling region (Ghosh et al., 2020, p. 9). According to SCHERER ET AL., max pooling results in faster model convergence, thus increasing the learning speed (Scherer et al., 2010). But also other forms of pooling as average pooling, stochastic pooling or spatial pyramid pooling, exist. Pooling helps to reduce computational complexity, extract invariant features, and improve the network's robustness to small spatial variations. In addition, pooling layers also contribute to the network's ability to generalise and reduce the risk of overfitting by discarding some spatial information. (Akerkar, 2019, p. 37; Almabdy and Elrefaei, 2019; Glassner, 2021, pp. 429–453)

Fully connected layer

Fully connected layers, also called dense layers, are the traditional neural network layers found towards the end of CNN architectures. These layers connect every neuron in the previous layer to every neuron in the current layer and convert the previous 2D feature maps into a 1D array. Fully connected layers aggregate and process the features learned by convolutional and pooling layers, capturing high-level representations. They are typically used for classification or regression tasks, allowing the network to make predictions based on the learned features. (Almabdy and Elrefaei, 2019; Glassner, 2021, pp. 429–453; Goodfellow et al., 2016, pp. 335–339)

Pre-trained CNN & transfer learning

For the training of a CNN to solve a specific problem, enough training data is required. By using CNNs trained on a large-scale dataset, the overall training process and the amount of additional labelled training data can be reduced significantly. First, they capture general visual patterns and features that are transferrable to other related tasks. This reduces the need for extensive training on limited datasets and helps overcome the limitation of insufficient training data. Additionally, pre-trained CNNs have already learned to recognise various visual concepts, enabling them to extract higher-level representations and provide useful features for downstream tasks. Transfer learning is the process of selecting a pre-trained network and finetuning it to a specific problem at hand. It usually achieves better performance and faster convergence than training a network with randomly initialised weights from scratch. (Glassner, 2021, p. 542; Tan et al., 2018; The MathWorks, 2021)

Hyperparameters
Some parameters must be defined before training a DL model because they are not obtained from training data. The so-called hyperparameters determine the behaviour and characteristics of the model and influence its learning process significantly. Model-specific hyperparameters define the configuration of the CNN, such as layer number or activation function (Bochinski et al., 2017). However, they are given to a large extent when applying pre-trained models. The main optimiser-specific parameters determining the learning algorithm are introduced briefly in the following.

The **optimiser** targets to identify the CNN weights that minimise a loss function. Stochastic Gradient Descent (SGD) or Adaptive Momentum Estimation (ADAM) are commonly applied optimisers. In a comparison of the performance of the same DL models on three different datasets, including ImageNet, SGD achieves higher accuracy on all sets (Gupta et al., 2021). However, there is no one-fits-all approach, and the faster ADAM optimiser is often the default. (Glassner, 2021, pp. 387–421)

The **learning rate** defines the step size at which a DL model updates its weights during training. It controls how the model parameters are updated in each iteration, impacting the speed and stability of the learning process. A higher learning rate can lead to faster convergence but risks overshooting the optimal solution. In comparison, a lower learning rate may result in slower convergence but potentially more exact adjustments. (Glassner, 2021, pp. 389–391; Szeliski, 2022, p. 288)

The **batch size** defines the number of images passed to the network in an iteration. After processing a batch, the model weights are updated. It affects the trade-off between computation efficiency and the quality of the gradient estimate. A larger batch size can provide a more accurate gradient estimate but requires more memory and computational resources, which is often the limiting factor. A smaller batch size increases the runtime but can result in better generalisation. (Feurer and Hutter, 2019; Goodfellow et al., 2016, p. 276)

The **number of epochs** refers to a single pass-through of the entire training set during the training process. It includes presenting the training examples, calculating the loss, and updating the model parameters. Therefore, it directly affects the training time. Both too large and too small numbers of epochs can result in suboptimal training results such as underfitting or overfitting. Early-stopping is a technique to avoid overfitting by ending the training process automatically if the validation error does not decrease for a defined amount of epochs (Ying, 2019). An **iteration** is the complete pass-through of a batch of images, including loss calculation and updating the model parameters. (Bochinski et al., 2017)

2.3.3 Classification & Detection models

After discussing the main concepts of ML and DL at the beginning of this section, this subsection briefly introduces ML and DL algorithms and models applied in this thesis. At first, classification algorithms are presented before selected object detection and instance segmentation models are explained.

Support Vector Machine

Support Vector Machines (SVMs) are a class of supervised ML algorithms used for classification and regression tasks. SVM is a maximum margin classifier that aims to find an optimal decision boundary or a hyperplane that best separates different classes in the input data mapped into a high-dimensional feature space. SVMs use support vectors, a subset of training data points closest to the decision boundary, to define the hyperplane. By maximising the margin between the support vectors, SVMs provide robust and generalisable models that can handle complex datasets and achieve good classification performance. Furthermore, once trained, only the support vectors are stored, which reduces memory and facilitates fast predictions. Further details are provided in Appendix A1.5. (Stanevski and Tsvetkov, 2005; Szeliski, 2022, pp. 250–254)

ResNet50

The residual network 50 is a 50-layer deep CNN with 49 conv layers and one fully connected layer at the end. The basic idea of residual networks is to skip blocks of convolutions to allow the network to perform identity mappings. Therefore, feature maps flow directly from the previous layer to the subsequent layer. In ResNet50, for example, the output of the first conv block (three conv layers) is the input for the second conv block (three conv layers) and is added directly to the output of the second conv block, skipping three layers. The implementation of residual layers enables the training of deep networks with improved accuracy and convergence compared to traditional CNN architectures. In 2015, ResNet was the first CNN to beat a human in the *ImageNet Large Scale Visual Recognition Challenge* (Fei-Fei and Deng, 2017). (He et al., 2016)

One-stage & Two-stage object detectors

DL-based object detection algorithms have gained much interest from researchers worldwide in the last decade. They are divided into one-stage and two-stage detectors. The latter processes the images in two consecutive steps. At first, the detection architecture generates an object region proposal (region of interest, ROI) with conventional or DL methods, such as selective search or region proposal networks (RPN). Then, the object classification follows based on features extracted from the proposed region with bounding box regression. Two-stage methods have the highest detection accuracy but are typically slower. The most common two-stage object detection models are Region-based CNNs (R-CNN) (Girshick et al., 2014), fast R-CNN (Girshick, 2015), and faster R-CNN (Ren et al., 2017). (Jiao et al., 2019; Zhao et al., 2019)

The second kind is the one-stage detectors that predict bounding boxes over the images without the region proposal step, thus consuming less time and more suited for real-time applications. These detectors divide the image into a grid of predefined anchor boxes and simultaneously predict the class labels and bounding box offsets for each anchor box. One-stage detectors are generally faster but may sacrifice some accuracy compared to two-stage detectors. The most common one-stage detectors belong to the Single Shot MultiBoxDetector (SSD) (Liu et al., 2016) or You Only Look Once (YOLO) (Redmon et al., 2016) detector families. (Jiao et al., 2019)

Single Shot MulitBox Detector

Liu et al. proposed a novel method of detecting objects in images using a single deep neural network and called this detection method the Single Shot MultiBox Detector (SSD). This detection method is simple, easy to train, and very straightforward. The SSD detector directly performs convolutional prediction on the underlying feature extraction network and extracts feature map characteristics at different scales. The SSD predicts a set of default bounding boxes and confidence scores for defined locations in the image for each class. During training and evaluation, the identified bounding boxes are matched with the ground truth using intersection over union (IoU) (Liu et al., 2016). Several improved versions of SSD algorithms can be found, introducing a feature pyramid in DSSD (Fu et al., 2017), feature fusion in FSSD (Li and Zhou, 2017), and attention mechanism in ASSD (Yi et al., 2019). However, since 2019, no model improvement has been published.

You Only Look Once

Another widely used one-stage detector was proposed by Redmon et al., known as YOLO, an acronym for You Only Look Once. It uses regression to solve the object detection task, where a single CNN directly predicts the bounding box coordinates and class probabilities of multiple objects in a single step. Therefore, the YOLO algorithm divides the input image into an $S \times S$ grid of cells. Then, it predicts a fixed amount of anchor boxes for each cell and returns the bounding box coordinates relative to the cell's position, width, height, and class probabilities for each anchor box. (Redmon et al., 2016)

The initial YOLO model encouraged many researchers to improve the model. Until the beginning of 2023, at least the updated versions YOLOv2 (Redmon and Farhadi, 2016), YOLOv3 (Redmon and Farhadi, 2018), YOLOv4 (Bochkovskiy et al., 2020), PP-YOLO (Long et al., 2020), PP-YOLOv2, YOLOX (Ge et al., 2021), YOLOR (Wang et al., 2021), YOLOv5 (Jocher et al., 2022), YOLOv6 (Li et al., 2022), YOLOv7 (Wang et al., 2022), and YOLOv8 (Jocher and Chaurasia, 2023) have been published. This emphasises the huge research interest in the YOLO detection model family. The newer models improved many aspects, from model architecture over data augmentation to better hardware implementation and usability.

Detectron2

Detectron2 is an open-source CV library developed by Facebook AI Research (FAIR). It offers a wide range of state-of-the-art object detection and segmentation algorithms, including Faster R-CNN, Mask R-CNN, and RetinaNet, with pre-trained models and a large set of configurable components offering a convenient DL platform. In contrast to the original algorithms, Detectron2 offers additional features and options. For instance, while Faster R-CNN in the original framework uses a ResNet Conv4 backbone with Conv5 head and a single feature map, Detectron2 enhances this algorithm by incorporating the Feature Pyramid

Network (FPN), achieving optimal speed and higher accuracy on the COCO dataset. (Wu et al., 2019)

2.3.4 Performance Metrics

Measuring the performance of predictions is part of the evaluation during the training process and an essential step in developing and optimising ML and DL models. The primary metrics applied for performance evaluation are presented in the following.

The basis for many metrics is the **confusion matrix**, which is depicted for binary classification in Figure 2.10 a). The predictions of the models are compared to the ground truth class. The diagonal elements contain the True Positive (TP) and True Negative (TN) values where the predicted label is correct and equals the actual label. False Positive (FP), i.e., the prediction is positive for a negative result, and False Negative (FN) for the opposite case, is used for false predictions. (Dalianis, 2018; Ting, 2016)

The **accuracy** is the ratio of all positive predictions divided by all predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.1}$$

The **recall** or sensitivity represents the true positive rate, the proportion of positive data samples evaluated correctly.

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \tag{2.2}$$

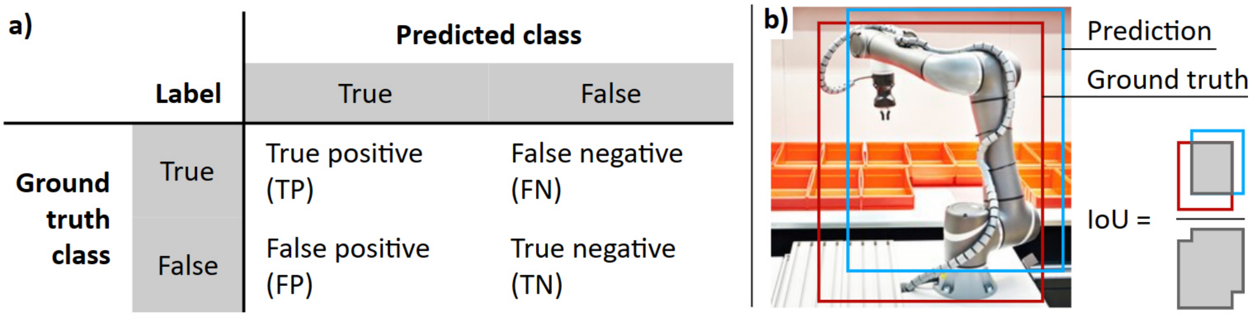
The **precision** is the ratio of positive results predicted correctly divided by all positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{2.3}$$

The **F1-score** is the harmonic mean of both precision and recall and is used to measure test accuracy. The accuracy may be high in imbalanced datasets, although the minority class is not classified well ($TN >> TP$). Precision, recall, and F1-score measures are more suitable for such cases since they do not include TN in the calculation.

$$\text{F1-score} = 2 \cdot \frac{\text{precsion} \cdot \text{recall}}{\text{precision + recall}} \tag{2.4}$$

The **Intersection over Union** (IoU) evaluates the localisation quality of a prediction in object detection. Therefore, the overlap between the ground truth bounding box G and the



a)

		Predicted class	
	Label	True	False
Ground truth class	True	True positive (TP)	False negative (FN)
	False	False positive (FP)	True negative (TN)

Figure 2.10: Confusion matrix adapted from (Ting, 2016). b) Intersection over Union (IoU) adapted from (Elgendy, 2020). Image source: (Adobe, 2023).

predicted bounding box P is divided by their union (s. Figure 2.10 b). The IoU value ranges between 0 (no overlap) and 1 (100% overlap). A prediction is counted as TP if the IoU value exceeds a defined threshold, usually 0.5.

$$\mathrm{IoU}\,(G, P) = \frac{G \cap P}{G \cup P} \tag{2.5}$$

The most frequent evaluation metric for object detection is **average precision** (AP). The AP describes the area under the precision-recall curve obtained by computing precision and recall for every confidence threshold. The **mean average precision** (mAP) is the average of the APs computed for each class. The mAP of classification and detection benchmarks differ in terms of IoU thresholds. Whereas in Pascal VOC, a threshold of 0.5 is used, the COCO benchmark applies an average of mAPs over a set of IoU thresholds, IoU $\in \{0.5, 0.55, ..., 0.95\}$. Furthermore, the evaluation of different-sized objects is part of the COCO benchmark. In this thesis, mAP50 denotes the mAP at an IoU threshold of 0.5, mAP75 represents the mAP at an IoU of 0.75, etc. The mAP50-95 is the average of mAPs as defined in the COCO benchmark. (Everingham et al., 2010, p. 8; Lin et al., 2014; Szeliski, 2022, p. 381)

Lastly, the **training time** and the **inference time** are recorded. Training time refers to the total time taken to fully train the model on the training data set with the selected hyperparameters. The inference denotes the time a model needs from data input to return a prediction result.

2.4 Discussion of State of the Art

This thesis aims to leverage the potential of learning-based CV models to enable assembly automation in HMLV environments. The Venn diagram in Figure 2.11 presents the relevant subject areas introduced in the previous sections and their overlap. The core subject of this thesis is positioned in the intersection of the three areas HMLV, learning-based CV, and assembly automation (4). Literature, which is assigned to that area, is considered highly relevant for this thesis. The additional consideration of intersections 1, 2, and 3 completes the analysis and indicates research gaps.

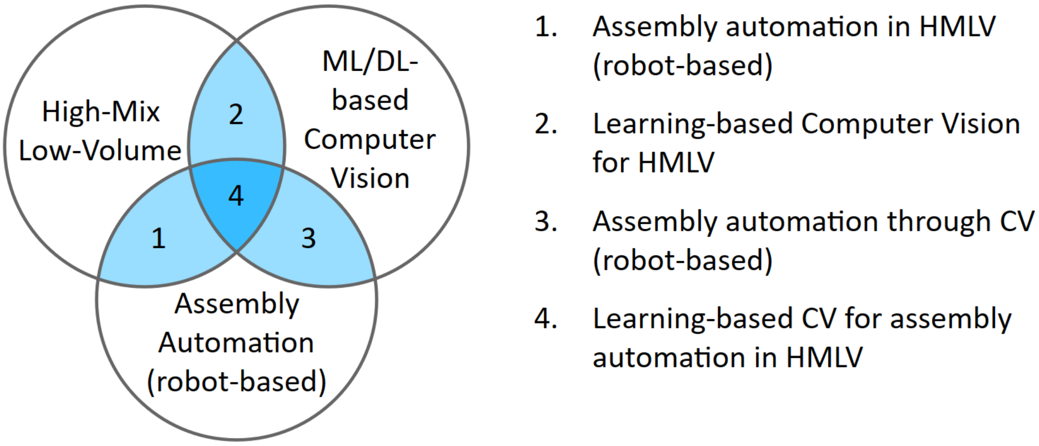

High-Mix Low-Volume
ML/DL-based Computer Vision
Assembly Automation (robot-based)
1
2
3
4
1. Assembly automation in HMLV (robot-based)
2. Learning-based Computer Vision for HMLV
3. Assembly automation through CV (robot-based)
4. Learning-based CV for assembly automation in HMLV

Figure 2.11: Venn diagram representing the focus of the literature research.

2.4.1 Overview

Table 2.1 provides an overview of the key research conducted in the areas highlighted by the Venn diagram, and the first column relates to the enumeration. In HMLV manufacturing, researchers have focused on understanding the characteristics and challenges of this domain. They have also identified roadmaps with technologies likely to be adopted in the industry in the coming years. A significant emphasis is placed on developing flexible automation solutions for the HMLV environment. Increasing productivity and reducing costs in high-wage countries are essential for maintaining competitiveness in the global market. However, previous automation efforts have faced limitations in terms of flexibility, resulting in a significant proportion of manual operations. Furthermore, SMEs are underrepresented in the research landscape and struggle to grasp and implement the generated knowledge. These issues emphasise the need for flexible and reconfigurable robotic systems in HMLV, focusing on SMEs. Enabling technologies in this regard are AI and human-robot collaboration (HRC).

Another prominent research area is the layout and planning of manufacturing and assembly systems suitable for the HMLV environment. The primary focus is maintaining flexibility to adapt to changing product volumes and variant demands. The exploration of simulation and digital twinning techniques targets to increase technical and financial planning confidence.

Table 2.1: Overview of the research areas and their research focus.[1]

Area	Subject	Topic	Authors
HMLV	Overview	Technology-Roadmapping, challenges	Flannigan et al., 2014; Gan et al., 2023; Grube Hansen et al., 2017; Johansen et al., 2021b; Karaulova et al., 2019; Tahmina et al., 2023
		Flexibility	previous
		Automation & HRC	previous and Falco et al., 2021
		Demographic change	Acharya et al., 2019
	Manufacturing Layout	Organisation	Telgen et al., 2014
		Cellular production system	Hoshino et al., 2008; Kusuda, 2010
		Reconfigurability	Doan and Lin, 2017
		Simulation	Caccamo et al., 2022; Herps et al., 2022
	Planning & Scheduling	ML-based, agent-based optimisation	Kocsi et al., 2020; Lin et al., 2020; Zhou et al., 2021; Zhu et al., 2017
(1)	HRC	Task allocation	Chen et al., 2014; Chen et al., 2011; Malik and Bilberg, 2019
		Intention estimation	Cramer et al., 2018
		Parameter learning	Sorensen et al., 2018
	Programming	LfD	Kito et al., 2017; Ko et al., 2016; Eissa et al., 2020
		CAD based	Hu et al., 2020; Pane et al., 2020
		AR, Telemanipulation	Ong et al., 2020; Weng et al., 2020; Yuan et al., 2020
	Material handling	Grasping	Lager et al., 2019; Xie et al., 2022
		Gripper design	Chen et al., 2015
(2)	Inspection	Defects	Apostolopoulos and Tzani, 2022; Gamal et al., 2021; Lang et al., 2022; Li et al., 2018; Mazzetto et al., 2020; Nwankpa et al., 2021; Singh and Desai, 2023; Vu et al., 2023; Zheng et al., 2021a; Zheng et al., 2021b
		Monitoring, Quantity & Existence	Hu et al., 2021; Ma and Peng, 2019; Malburg et al., 2021; Zancul et al., 2020
		Implementation framework	Hridoy et al., 2022
	Grasping & path planning	Unknown objects, unknown poses, bin picking	Chu et al., 2018; Deng et al., 2015; Ishiyama and Lu, 2022; Onstein et al., 2020; Sharma and Valles, 2020
(3) & (4)		see Section 2.4.2.	

Scheduling is another important topic for HMLV manufacturers. ML-based or agent-based models are applied to optimise overall efficiency in manufacturing and assembly systems. Integrating Industry 4.0 technologies such as cyber-physical systems and RFID tags enhances situational awareness in various aspects of the plant (e.g., product status, machine status, warehouse status), enabling better response to unforeseen events like machine breakdowns and facilitating decision-making. Improved planning and scheduling lead to workload balancing, increased utilisation and throughput, faster delivery, and overall cost reduction.

In the combined area of assembly automation and HMLV (s. Table 2.1, (1)), three highly researched topics are identified: HRC, robot programming, and material handling/grasping. HRC, which is also highlighted in technology roadmaps, is expected to have a significant impact on HMLV assembly. Task allocation, i.e., assigning tasks to human workers and robots in an optimal manner, is a common problem in HRC. Assembly tasks are classified based on complexity factors, such as handling, mounting, human safety, and part feeding, to optimise assembly time and cost. Human intention estimation is another aspect of research in HRC, which involves identifying object affordances or employing algorithms such as filter-based state estimation. The goal is to determine the optimal moment for transitioning a task between humans and robots, facilitating natural and efficient collaboration. Additionally, the research investigates enabling robotic systems to find reliable solutions for tasks with inherent process uncertainties. This involves searching for an optimal set of process parameters using statistical or learning-based techniques such as reinforcement learning.

Research in robot programming aims to reduce the effort required for reprogramming robotic systems. Learning from Demonstration (LfD) is an intuitive approach where robots are programmed based on force or visual feedback. The robot arm is guided directly along the desired path, or the motions of teaching tools are tracked to capture the 3D pose and calculate the trajectory. The resulting program is optimised directly by the system or human involvement (human-in-the-loop). Another technique is to generate robot trajectories directly from CAD files. Assembly tasks are derived by matching defined tasks to geometric constraints in the CAD model, e.g., a concentric constraint of two parts may require an insertion. The parametrisation of the task enables the generation of a path that the robot executes with skilled motions, such as force-controlled alignment of surfaces. Alternatively, operators can select features through a Human-Machine Interface (HMI), such as edges or faces, which serve as the basis for proposing a trajectory. Furthermore, telemanipulation or

[1] Non-exhaustive overview.

the integration of Augmented Reality (AR) devices allows operators to interact with robots for programming without the need to write source code.

Flexible handling of different objects is another challenge in HMLV environments. Research has focused on developing new gripper designs that enable flexible grasping, such as anthropomorphic grippers that offer higher functionality and adaptability to different product characteristics like shape, material, or hardness. Alternatively, researchers have explored improving grasping capabilities with universal grippers by leveraging learning-based models to predict grasping positions. These systems utilise both visual and tactile feedback. Reinforcement learning is a common approach where robot systems learn grasping through trial and error. The emphasis is on models that learn grasping rather than the implemented CV models.

The research field of learning-based CV in HMLV environments (s. Table 2.1,(2)) primarily involves inspection tasks. Many published articles focus on the classification or detection of surface defects, particularly on metallic surfaces. Monitoring and quality control for specific process steps are also areas of interest, where the presence or count of certain features is checked. While most researchers present specific problem instantiations or review CV techniques, a framework proposed by Hridoy et al. addresses the implementation of learning-based classification in industrial environments. The framework is discussed further in the next section. However, in these publications, the spatial information of detected objects in images is typically used for visualisation only. Social or mobile robots use the additional information from detection or segmentation for object avoidance or grasping point estimation. The training of the models involves public datasets with common objects, e.g. a pen or cup, without any industrial context. Additionally, the research on CV models examines grasping and path planning. In bin-picking tasks, CV models identify objects, and the RoI is then mapped with depth information to determine the object's pose. 3D model reconstruction from canonical views is performed for unknown objects and serves as input for grasping point prediction.

The most relevant topics from (3) and (4), along with selected works from the overview, are discussed in the next section.

2.4.2 Comparative Analysis

The comparative analysis presented in this section focuses on the most relevant publications pertaining to the thesis topic. Based on the three primary subject areas outlined in Figure 2.11, the analysis encompasses literature that exhibits substantial overlap with the defined problem statement and objective, namely the utilisation of learning-based CV models for enabling assembly automation in HMLV environments. The selected publications are reviewed and discussed below.

Hridoy et al. focus on developing an industrial inspection system framework using DL. Although not directly linked to assembly, the framework development aligns with the targeted procedure and is thus included for analysis. The presented framework uses CNN for defect detection and classification of products as either defective or non-defective. The framework comprises three main steps: data collection, model training, and defect detection. The data collection includes collecting product image data, processing, and labelling. In the model training, a proposed CNN is trained on the dataset for binary defect classification. The defect detection step applies the same image acquisition and processing as the data collection and uses the trained CNN to the new image. The classification result is passed to a µ-controller that sends signals to a servo motor to filter out the defective parts. The authors apply the framework to an inspection task of hexagonal nuts. They generate a balanced dataset of 4000 images with defective and non-defective parts. Regarding the CNN, they conducted several trials with different standard architectures and hyperparameters to identify an optimal CNN. However, the framework does not include any model selection process. The proposed CNN is trained on the custom and public casting material datasets. The achieved classification accuracy is in both cases >99.5% and, according to the authors, outperforming existing methods. (Hridoy et al., 2022)

The proposed framework is rather simple and tailored to the problem at hand. The preprocessing and ROI extraction are static, and a predefined patch is cropped from the image. A dynamic ROI proposition is essential for inspection tasks in changing HMLV processes but is not considered. Furthermore, despite conducting experimental analyses to identify the optimal architecture, the authors do not include the CNN model selection in their framework, thus lacking consistency. The framework further limits the inspection tasks to a binary classification into non-defective or defective. The validation datasets contain only one product, either the hexagonal nuts or casted metal parts, without any fluctuation or noise, that would resemble the changing environments encountered in HMLV scenarios. Overall, the framework is very specific to the authors' task and lacks wider applicability. It is static, focuses on binary classification only, and is validated on basic datasets.

Shao et al. published another framework depending on a learning-based CV system. They present a location instruction-based end-to-end motion generation framework for sequential robotic manipulation tasks. The proposed framework consists of three modules: the AutoEncoder (AE), the location detection network, and the motion generation network. The AE receives colour and depth images and processes the information into small-scale feature maps. The location detection network proposes the target object's location based on the feature maps and an input image of the target object. The motion generation network generates the entire motion trajectory to finish the specified manipulation task. Here, the task is to build a tower of three coloured cubes by stacking them at a defined location in the robot's workspace. The overall experiment includes 15 differently coloured cubes, of which five are used in a single experiment. Out of the five, the model must detect three specified cubes. The system achieves a 90% success rate in real-world experiments. (Shao et al., 2020)

Overall, the proposed model by Shao et al. can generate trajectories for sequential tasks based on visual inputs for the given use case. The selected sequential task is very limited and encompasses the stacking of regular parts that are always oriented. The task is neither challenging for grasping, manipulation, nor detection. However, the novelty lies in predicting sequential collision-free trajectories. For the proposed framework, the CV system is essential as well. Regarding the detection problem, there are always coloured cubes against a well-defined background pictured perpendicular from a ceiling-mounted camera, and colour thresholding is probably sufficient. The detection problem is far away from any realistic industrial detection scenario. Still, it remains a core element for the whole motion generation.

Gauthier et al., Schlette et al., and Drigalski et al. describe flexible robotic systems designed for HMLV tasks. While Gauthier et al. focus on reducing programming and automatic learning of new objects for assembly, Schlette et al. and Drigalski et al. present very sophisticated robotic systems that competed successfully in robotic assembly challenges. For all three systems, visual inputs are essential.

Gauthier et al. propose a programming-free robotic system for assembly tasks, allowing a robot to learn new objects and tasks from non-expert users without the need for traditional programming. The authors present three main modules: object teaching, task teaching, and task execution, which heavily rely on visual inputs. In the object teaching module, the system autonomously generates RGB images of a new object to train a Faster R-CNN object detector. The system employs a visual exploration routine to create a training dataset from augmented canonical views of a new part. For task teaching, a human demonstrates the assembly tasks manually, and the camera system utilises the trained detector to track hand and object motions. Probabilistic logic rules recognise task primitives based on relative hand and object motions. By decomposition into these task primitives, a sequence of actions is generated. The task execution module incorporates multimodal perception from RGB-D cameras and a tactile sensor. The trained Faster R-CNN identifies the locations of the parts for assembly in the workspace. The system obtains the real-world coordinates with a registered depth camera. The robot picks the parts at the gripping point, defined as the centre of the detected bounding box with a tactile gripper. The overall robotic system is implemented and validated on an insertion task. (Gauthier et al., 2021)

The authors present a compelling case for a complete system from object learning to task execution. Considering the CV system, several weaknesses are observed. For learning, the new object must be in a defined location to execute the exploration routine. In the experiments, all parts are orange or grey on a white background with strong contrast. All specimens are 3D printed and not genuine industrial-grade parts. The model's ability to generalise on more complex problems is questionable, given the relatively simple detection task. Furthermore, the authors provide no information on the system's object learning process duration and accuracy. The authors use the Faster R-CNN only for object detection. A combined model based on CHT and SVM detects the assembly targets, i.e., bores in a housing. Thus, the learning process does not apply to other assembly tasks with different product characteristics and assembly targets. Finally, it is unclear how and why the DL-based CV model is selected. Regarding the model's intended use, the publication lacks the consideration of other models available at the time of publication that are more performant, faster, and easier to train.

The paper of Schlette et al. describes developing a novel robotic assembly cell design for HMLV production scenarios. The proposed assembly cell won the World Robot Challenge (WRC) in 2018. The cell design comprises two mid-size industrial robots attached to a central, reconfigurable worktable that connects and redistributes power, compressed air, ethernet, and Safety-PLC. For the assembly challenge, the authors 3D-print nearly all required tool holders, fixtures, suction nozzles and a specific tool for scooping to keep flexibility at a high level. For example, the robots grasp each part with part-specific fingertips attached to the grippers. Furthermore, the authors avoid bin picking by using a scooping tool that collects and orients a set of loose parts so that the other robot can pick them at a defined location. Unsorted trays carry the parts for the assembly challenge. A CV system is necessary for determining the uncertain poses of the objects to be picked. Based on known CAD data, the convex hull of the object is extracted and used to calculate stable orientations of the object on a planar surface. The system presents the orientations sorted by stability to the user who selects the case in the specific assembly scenario. The part orientation is then determined by template matching to the virtual model in the same pose. This CV system enables fast setup times for new parts since it only requires the CAD model without training. (Schlette et al., 2019)

The demonstration of the concept of a reconfigurable and flexible robot assembly cell is impressive. The main contribution is the system architecture description with the aspects of its implementation regarding robot control, programming, and CV. Again, the CV system is critical to identify the assembly parts. The presented approach is, however, limited to known objects with available CAD data provided on a planar surface. Furthermore, a human must select the orientation to generate the virtual template for the matching algorithm. Despite its sufficiency for the discussed challenge, its potential for an industrial assembly is low.

Drigalski et al. introduce a two-armed robot system designed for the automatic HMLV assembly that achieved third place at the WRC 2020. The authors implement no parts-specific jigs or fingertips but use general-purpose grippers, hand-held tools, and two RGB-D wrist cameras. Four main modules comprise the system: an assembly database, a symbolic planner, a CV system, and a hybrid force-position controller of the robot arms. The assembly database stores all assembly CAD models, assembly sequences, and part gripping points, generating target positions for each part. The planner is a core of the system and responsible for the uncertainty-aware manipulation of objects. The authors implement different action sequences to reorient the object and accurately determine the position, such as centring by multiple grasping the object in x and y directions. The CV system detects the parts in the workspace following a coarse-to-fine approach. The SSD object detector predicts the locations on the 2D image of the parts. The accurate position is then determined based on the object contour for small or flat parts or via template matching of the CAD data with the part. The assembly tasks are achieved with force-control and compliant tools (spring loaded for compensation). (Drigalski et al., 2022)

The presented robotic system achieves high performance and can assemble various objects. The uncertainty-reduction procedures are especially impressive as they significantly increase the grasping accuracy with simple motions. Regarding the vision system, the authors combine colour and depth information. Although the implemented SSD detector is rather outdated at the time of the publication, it is the most robust part of the CV system and often outperforms the 3D point cloud-based recognition routines, as stated by the authors. Besides detecting the objects, identifying and locating certain product characteristics, such as bores or threads, is also necessary. Instead of integrating this task in the SSD model, the authors rely on classical edge detection, which is far more sensitive to changing lighting conditions, posing a challenge for the system. Furthermore, the system requires complete CAD documentation of the assembly to generate the assembly targets.

Mo et al. implement the YOLOv3 algorithm for identifying and detecting the position of solder joints in automotive door panels. The authors aim to improve an existing automotive door assembly line by automating the soldering process exposed to frequent product changes in unstable environmental conditions. The basis is a robust and flexible detection algorithm. Therefore, the authors create a dataset of 553 door panel images with approximately 60 solder joints per image of three different types utilised for training the YOLOv3 model. It achieves an AP50 of 0.78-0.96 for the three joint types. The mAP50 is 0.85 and comparable to the existing system. However, the YOLOv3 model distinguishes the joint type and is 50 times faster than the original conventional system. With less than 200 ms per prediction, the model achieves near real-time sufficient for many industrial processes. (Mo et al., 2019)

The industrial case study presented by Mo et al. is a good example of the potential of learning-based detection models in the industry, but the results are limited. Although the authors state the difficulty of changing environmental conditions for the vision system, this is not visible in the dataset. Furthermore, the considered products change in geometry, but the solder joint appearances stay constant. Thus the relevant product characteristics to be detected exhibit a low variation for all considered products. Consequently, the proposed model will likely decrease performance on new products with a more significant change. Finally, the paper ends with the plain detection model results without implementation, although the authors intend to use spatial information for the joining process. It is unknown whether the bounding box predictions are sufficient or if further measures are necessary, which probably are.

2.4.3 Summary

The literature analysis reveals a significant number of publications that address the topics of HMLV, learning-based CV, and assembly automation, highlighting their current relevance and significance in academia and industry. Integrating ML and DL methods to enable flexible automation tasks in HMLV manufacturing is becoming increasingly important, with learning-based CV systems playing a central role in many of these approaches. Notably, a substantial amount of research focuses on robotics and inspection applications utilising learning-based CV models. However, these approaches and methods often exhibit a high level of specificity tailored to a certain problem at hand. Moreover, much of the research remains confined to laboratory settings or non-industrial use cases, lacking real-world implementation. The application of such models in an industrial context today is limited primarily to inspection tasks rather than direct integration into the assembly process itself. The detailed analysis of six highly relevant publications for this thesis reveals that proposed frameworks for developing and implementing learning-based CV systems are often either problem-specific or non-existent. Despite the presentation of sophisticated robotic solutions by some authors, there is a lack of explanation regarding the design and selection of CV models for specific problems in assembly. Consequently, there is a need for a universal approach to effectively harness the potential of learning-based CV models and address the challenges associated with industrial products and assembly processes in the HMLV domain.

3 Objectives & Research Methodology

Based on the knowledge and the identified gaps from the state of the art, the research objectives are refined in Section 3.1 of this chapter. From there, the research methodology is proposed along with research leading questions in Section 3.2. Lastly, the scope of research and the requirements are stated in Section 3.3.

3.1 Research Objectives

Current research directly linked to automation in HMLV environments focuses on the following aspects. Literature in the HMLV domain primarily addresses organisational topics such as line layout and scheduling. They also highlight the need for flexible automation solutions, but the proposed approaches are generic and limited. In the assembly domain, the literature concentrates on improving robot programming and path generation using techniques such as Learning from Demonstration, Augmented Reality, telemanipulation, and proposing paths directly from CAD data for faster setup of flexible production equipment. Another area of focus is grasping unknown objects for flexible material handling. While CV systems are mentioned, they are not the primary focus, and there is a lack of explanation regarding how and why specific models are adapted to the problem.

In the CV and ML/DL domain, extensive research has been conducted, leading to significant advancements in recent years. However, when examining industrial applications, the majority of publications address quality control and surface inspection tasks. The limited publications discussing CV application in assembly tasks focus on specific problems that do not apply to other scenarios. Despite learning-based CV being recognised as a critical technology in flexible robotics and automation, there is currently no universal approach for effectively utilising it in assembly automation in the context of HMLV products.

The research objective of this thesis is to develop a procedure that leverages learning-based CV methods to enable the automation of assembly tasks, particularly in SMEs operating in HMLV environments. The research addresses the challenges associated with assembling, specifically joining, various products exhibiting high variance. To achieve this, generic joining tasks are analysed, and critical process-relevant parameters essential for task automation are defined. These parameters, categorised as location-dependent and time-dependent (s. Section 4.3), are applicable across different joining tasks. The procedure involves identifying the process-relevant parameters for a given set of products and their manual joining process, followed by developing a robust and flexible CV model for accurately determining the parameters.

Ultimately, the developed procedure aims to enable the flexible automation of the industrial partner's lightweight composite panel assembly, considering changing products, product characteristics, and dynamic environmental conditions.

3.2 Research Methodology

This thesis adheres to the research methodology for applied sciences as defined by Ulrich, which consists of seven steps. It begins by identifying and illustrating practice-relevant problems (Step 1) and reviewing existing literature on the subject (Steps 2 & 3). The next step involves gaining a comprehensive understanding of the specific application context within the industry (Step 4), followed by the development of general methods or models (Step 5). These methods or models are then tested in the application context, accompanied by practical consultations (Steps 6 & 7). This approach ensures the research is grounded in real-world problems and oriented towards practical implementation. (Ulrich, 1984, p. 193)

Starting in practice by initiating an industrial research project and ending with solving real-world problems for the project partner, this research suits well to the described guideline. The methodology of this thesis is derived from the defined objectives and is illustrated in Figure 3.1, together with the research leading questions.

Chapter 4 focuses on the development of the procedure, which involves decomposing typical joining tasks into basic motions and analysing the dimensionality of the joining process. Based on the motion and geometry taxonomy, process-relevant parameters for the joining tasks are defined. Two types of parameters are derived: location-dependent parameters, which relate to specific positions during the joining process, and time-dependent parameters, which vary with time during the joining execution. After defining the process-relevant

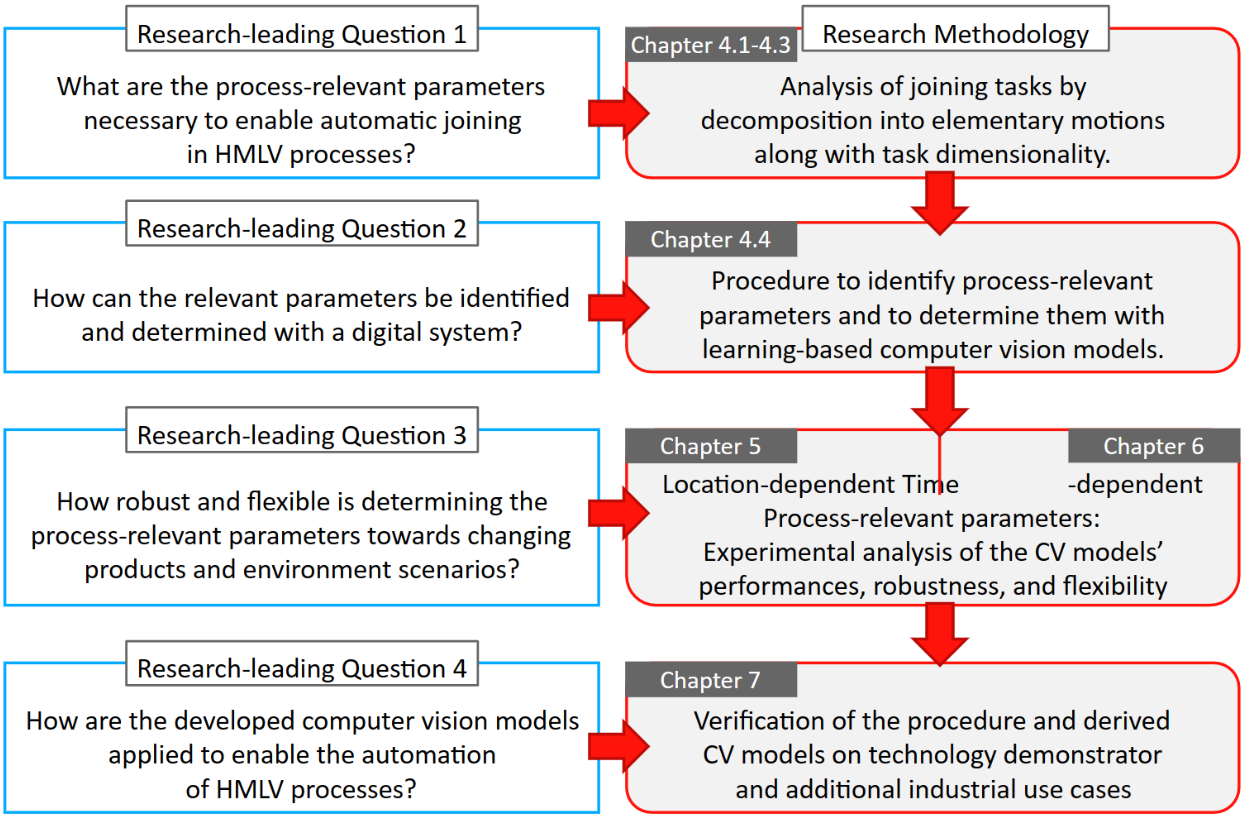

Research-leading Question 1
What are the process-relevant parameters necessary to enable automatic joining in HMLV processes?
Research-leading Question 2
How can the relevant parameters be identified and determined with a digital system?
Research-leading Question 3
How robust and flexible is determining the process-relevant parameters towards changing products and environment scenarios?
Research-leading Question 4
How are the developed computer vision models applied to enable the automation of HMLV processes?
Research Methodology
Chapter 4.1-4.3
Analysis of joining tasks by decomposition into elementary motions along with task dimensionality.
Chapter 4.4
Procedure to identify process-relevant parameters and to determine them with learning-based computer vision models.
Chapter 5
Chapter 6
Location-dependent Time-dependent
Process-relevant parameters:
Experimental analysis of the CV models' performances, robustness, and flexibility
Chapter 7
Verification of the procedure and derived CV models on technology demonstrator and additional industrial use cases

Figure 3.1: Research concept and structure of the thesis.

parameters, the proposed procedure outlines how to identify and determine these parameters using a learning-based CV model.

Chapter 5 and 6 apply the procedure to the joining processes of the industry partner to identify the process-relevant parameters. Chapter 5 focuses on location-dependent parameters, while Chapter 6 explores time-dependent parameters. Multiple conventional and learning-based CV models are developed, compared, and tested in various experiments to analyse their general applicability, performance, flexibility, and robustness in HMLV environments with changing products and dynamic conditions.

In Chapter 7, the developed models are implemented on a technology demonstrator, where the output of the models is utilised to automate insertion and glueing processes. Additionally, the procedure is further validated on two additional industrial processes without technical implementation. These chapters contribute to the practical application and validation of the developed procedure and models.

3.3 Scope of the Thesis

This thesis focuses explicitly on automating assembly processes, with a particular emphasis on the joining task, within the context of SMEs operating in HMLV environments. The primary objective is to address the challenges associated with assembling diverse products that exhibit significant variance.

In this context, the main challenge lies in coping with the high product variance rather than the complexity of the joining tasks. Therefore, any limitations should not be attributed to the difficulty of replicating specific joining motions or the inaccessibility of assembly locations. The focus of this thesis does not involve developing new solutions for complex assembly tasks or material handling. Instead, the research assumes that there are existing methods to conduct the required motions automatically, and tools used by human workers can be adapted for use by automation systems, such as mounting them to robot flanges as end effectors.

Within the assembly process, the targeted task is the actual joining process. Although other tasks like material handling are important, this analysis assumes that parts and materials can be manipulated to and from the joining location. The starting point for the proposed procedure is the existing manual assembly process in an HMLV manufacturing scenario. This means that assembled products and corresponding assembly processes are available as required input data for the analysis.

The approach involves implementing CV models to obtain the essential parameters for automation, referred to as process-relevant parameters. Therefore, it is necessary for the sensor, typically a camera, to have a direct line of sight to the joining locations or any other critical product features. The determinant features must be visibly accessible to the sensor hardware to obtain the required visual information.

Overall, the scope and requirements of this research can be summarised as follows:

- Focus on SMEs operating in HMLV environments
- Emphasis on automating the joining task

- Product variation is the primary challenge, not assembly complexity
- Motion execution should be achievable automatically
- Availability of information on manual processes and products.

After detailing the objectives, the methodology, and the scope of this thesis, the next chapter contains the procedure development.

4 Procedure to Identify & Determine Process-Relevant Parameters

The technical target of this thesis is deploying robust and flexible computer vision models (CV) that detect different product characteristics and process states intending to enable the automation of manual assembly processes in high-mix, low-volume (HMLV) environments. Therefore, developing an appropriate procedure is the focus of this chapter.

At the beginning of this chapter, the requirements and boundary conditions for the following analysis and procedure development are defined in Section 4.1. Then, types of process-relevant parameters are derived based on common assembly tasks in Sections 4.2 & 4.3. So far, physical process properties define the category of joining tasks, e.g. joining by welding vs joining by clamping and forcing (cf. Figure 2.1). Alternatively, they are broken down into basic human actions, such as reaching or moving. However, neither is suitable for defining the required information to enable automation. Hence, analysing the motion of frequent joining tasks in the assembly builds the basis for the necessary information, i.e., the process-relevant parameters. Lastly, this chapter provides a procedure for identifying the process-relevant parameters of product-process combinations and obtaining them using learning-based CV models in Section 4.4.

4.1 Requirements for the Analysis

The assembly comprises various tasks and processes used to assemble geometrically defined bodies (VDI 2860, 1990). In this standard, the VDI divides the assembly into processes of joining (DIN 8593, 2003), manipulation, control, alignment, and other special operations, which are defined in additional standards. Others break down the assembly process into basic motions. For example, up to 85% of fully controllable work in assembly consists of the five basic movements *reach, grasp, move, position*, and *release*, according to the German MTM association (MTM, 2011, p. 33).

Similarly, Awad proposes four assembly task categories: *separate, manipulate, position*, and *join* (Awad, 2017). Primary and secondary activities further organise the different processes or tasks independently of the categorisation. Primary operations comprise all activities to complete a product that add value during assembly. According to Lotter, the main primary operation in assembly is the actual joining process (Lotter, 2012b, p. 49). Hence, this research focuses on enabling the automatic conduction of joining processes in assembly.

However, not all joining processes defined in DIN 8593 are typically listed in assembly but in manufacturing, e.g. joining by moulding. Thus, a discrepancy exists between physical joining and joining, considered as an assembly activity. Typical assembly joining tasks are, for instance, joining by putting together (assembling), screwing, nailing, riveting, glueing, or similar processes (MTM, 2011, pp. III 15-53; Pfeiffer, 2013, p. 144). Therefore, only common joining processes in the assembly are selected, which are manually executable, with or without a tool.

4.2 Motion-based Taxonomy of Joining Processes in Assembly

The joining processes specified in the standard DIN 8593 are distinguished by the physics of the bond between two or more joined parts. By this, it is possible to specify, e.g., the strength of the joint or other mechanical properties. However, the motion to join components may be similar or the same for different joining processes. The joining motion and complexity are highly relevant for an automation system, but the actual joining process, e.g. screwing or riveting, may not be the decisive criteria. The target is identifying parameters that can be determined accurately and used across different joining processes. Therefore, grouping the different joining activities based on their motion is reasonable.

Table 4.1 provides an overview of different joining processes with their joining motion. The selection of joining processes specified in DIN 8593 is based on fundamental and standard processes defined by MTM linked to joining (cf. MTM, 2011, pp. II, III). It is visible that all joining processes grouped in joining by assembling consist of either a linear translation or a combination of translations and rotations along the same axis. The simplest form is joining by placing. Here, the joined parts are fixed in one direction by their contact surface with each other and held in place by gravity. The joining motion is along a single axis by placing one part onto another. The same is valid for the insertion of rotationally symmetrical parts. For inserts with not rotationally symmetric joining faces, the insert must rotate around the motion axis into the correct position before insertion. Mounting a hook assembly usually needs linear translations in two directions. Setting a bayonet connection consists of a defined rotation around the motion axis after inserting the connector with a linear translation. All motions require no or small joining forces.

Table 4.1: Joining processes and their basic joining motion.

Joining (DIN 8593)	Joining motion	Joining force
Joining by assembling		
Placing	linear translation	no
Inserting	linear translation, rotation	low-medium
Hook assembly	linear translations	low
Setting	linear translation, rotation	low
Snap fitting	linear translation	low
Joining by clamping and forcing		
Bolt, screw	linear translation, rotation	low-medium
Engineering fit	linear translation, rotation	low-high
Nail	linear translation	medium-high
Joining by forming		
Clinching	linear translation	medium-high
Rivet	linear translation	medium-high
Joining by welding		
Fusion welding	point, freeform translation	no
Joining by glueing		
Glueing	point, freeform translation	no
Joining by Filling		
Filling	no motion, positioning of tool	no

Linear translation and rotation around the translation axis are the motion components of all clamping, forcing and forming joining processes. Screws or bolts are attached with a combined translation and rotation of the same axis. Depending on the screw type, the joining force along the motion axis changes. Whereas machine screws require a low joining force in the direction of translation, thread-cutting screws are joined with significantly higher forces. The tightening torque depends on the screw or bolt type and the joint specification. The joining motion of engineering fits and nails are also linear translations. However, prior rotation may be necessary if the fit parts are not rotationally symmetric. The force depends on the joined materials or the type of engineering fit (clearance, transition). Clinching and riveting have linear translation motions. The assembly requires tools, e.g., a rivet gun or screwdriver, that are all existing and applicable in automated systems.

Joining by bonding consists of two consecutive assembly tasks. First, the adhesive is applied to the parts, and then the parts are joined by one of the previous processes. For the application, an adhesive dispenser must move to a single position, to multiple positions, or apply the glue along a path. The path can be on a freeform; thus, translation and rotation on multiple axes may be necessary. The application can be contactless, and the dispenser is always at a set distance from the joined parts. Similarly, a welding torch must conduct motions along multiple axes for fusion welding if the unwelded seam is a freeform.

Joining by filling is an extraordinary case. Here, cavities of a part are filled with a liquid, gas, paste or powder. If the filling is contactless, e.g. rinsing a liquid from above, there is no actual joining motion. A dispenser must be positioned so that the liquid can enter the part. Otherwise, a nozzle or similar device must be connected to the filled part, usually with a joining-by-assembling process.

All discussed motions are summarised in Figure 4.1, and a) shows the simplest case, where a tool must be positioned in the workspace relative to the joined parts. The motion can be a linear translation. An application example is the positioning of a nozzle above a cavity for filling. Figure 4.1 b)-d) are all combinations of a linear translation and a rotation around the translation axis. The rotation happens either before, during, or after translation. The given examples are a peg-in-hole (b), a hexagonal bolt (c), and a male bayonet connector (d). Assembling a hook requires translations in two directions (e). In contrast to the illustration, the directions are not necessarily perpendicular. An L-insertion is motion-wise similar but has a rotation between both translations. The rotation axis is normal to the plane containing both motion axes. The bent sheet metal structure (f) has an unwelded seam. In the depicted case, the seam is a line. Thus, linear translation is sufficient. However, more complex paths may result in a freeform motion.

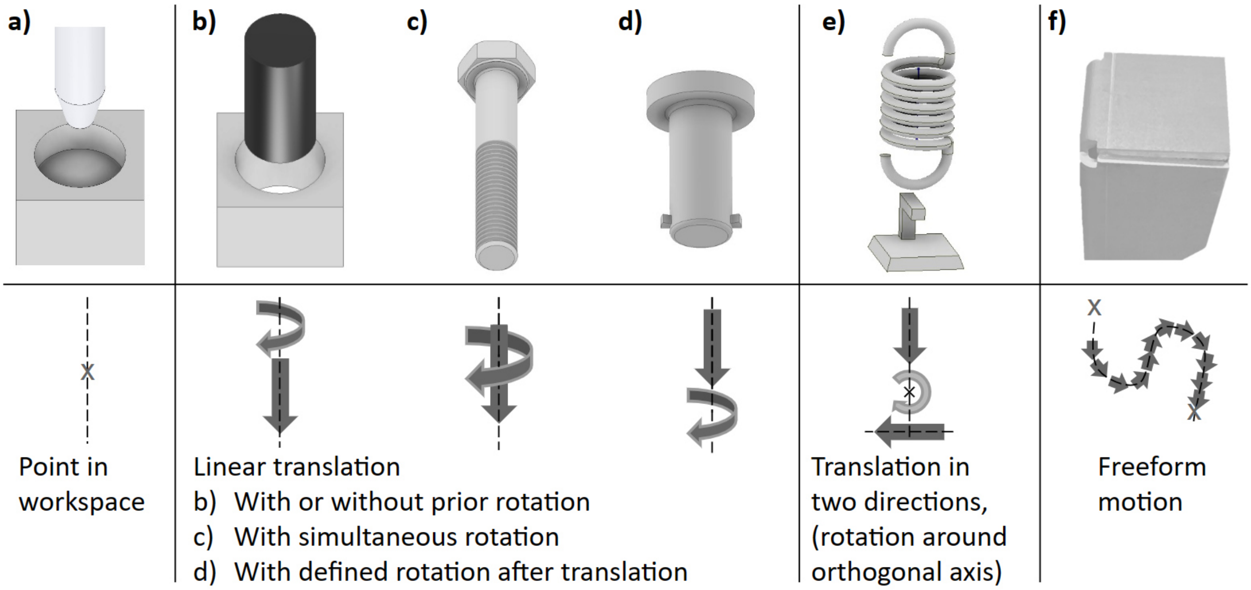

a)
b)
c)
d)
e)
f)
Point in workspace
Linear translation
b) With or without prior rotation
c) With simultaneous rotation
d) With defined rotation after translation
Translation in two directions, (rotation around orthogonal axis)
Freeform motion

Figure 4.1: Joining taxonomy based on basic assembly motions.[2]

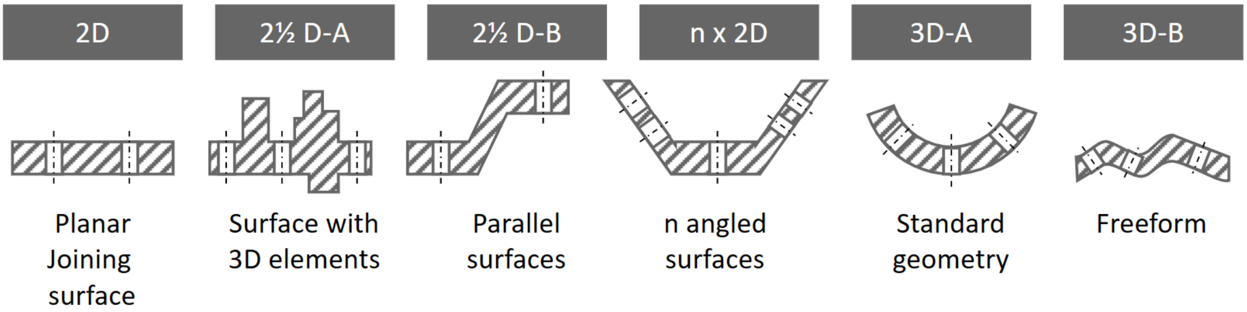


Figure 4.2: Geometric classification inspired by Krautheim et al., 1997.[3]

After categorising the basic assembly motions, it is necessary to define the geometry of the joined parts. For accurate geometric classification, the definition of the different spatial features is reasonable. Therefore, the arrangement and shape of the process and joining faces are considered. Figure 4.2 illustrates the geometrical classification. If all assembly targets, i.e., the positions of the assembly tasks on a part symbolised with a through-hole, are in a single plane, the assembly is in the dimension of 2D. If additional 3D elements exist outside the joining plane, the geometry is defined as 2½D-A. A geometry with multiple parallel joining planes is called 2½D-B. Parts of the dimension n x 2D have several process planes that are arranged at an angle to each other. Geometries classified as 3D have either

[2] The non-exhaustive taxonomy is based on common assembly processes described in this chapter. Other cases may exist.

[3] The non-exhaustive geometric classification is based on common assembly tasks described in this chapter. Other cases may exist.

joining faces belonging to standard shapes, such as a half cylinder or are freeform shapes. Based on the defined taxonomies, process-relevant parameters are defined in the following.

4.3 Definition of Process-Relevant Parameters

Specific parameters must be determined for the different assembly motions and joining geometries to start and execute the joining process with an automated system. The process-relevant parameters shall be defined in a way applicable across multiple assembly tasks. Determining them is conducted with a CV system. For the definition of these process-relevant parameters, certain assumptions are relevant. As mentioned in Sections 3.3 and 4.1, the baseline of the assembly is a manual process. So far, a static workspace is considered with joining parts at rest and roughly positioned to a reference, such as a mechanical end stop, a universal fixture, or just by gravity. Therefore, an approximate position of the parts is available. Alternatively, parts and their pose can be automatically recognised, e.g., with fiducial markers (Garrido-Jurado et al., 2014), the part geometry (point cloud) (Koga et al., 2022), and other techniques not considered in this thesis. Additionally, it is assumed that several part information is available in digital form, such as basic part geometry, assembly task and specifications (e.g., thread position on part and tightening torque for screwing).

Several combinations of the assembly motions and their spatial execution are possible. The cases b)-d) depicted in Figure 4.1 only have motions along one axis. The motion can be translation, rotation or a combination. Furthermore, the axis of motion can be perpendicular to the joining surface or angled. For any geometry, the assembly location must be known to enable the starting of the task. If a prior rotation along the motion axis is necessary to enable the joining, the assembly target's orientation must be determined before the execution. However, this can be determined as well systemically by the process design, e.g., parts are always provided in the correct orientation. The product data specify the angle between the motion axis and the joining plane. The length of the assembly motion can be specified by the process design or based on sensor feedback. Distance, force or torque are input variables to determine the end of a motion. For instance, rivets are driven into the assembly location until the factory head touches the part surface and screws are tightened until a specified torque is reached. Force-controlled, skilled motions are implemented to enable a peg-in-hole motion (Wang et al., 2020). Thus, most variables can be parametrised, and the motion conducted once the starting position is located sufficiently accurate.

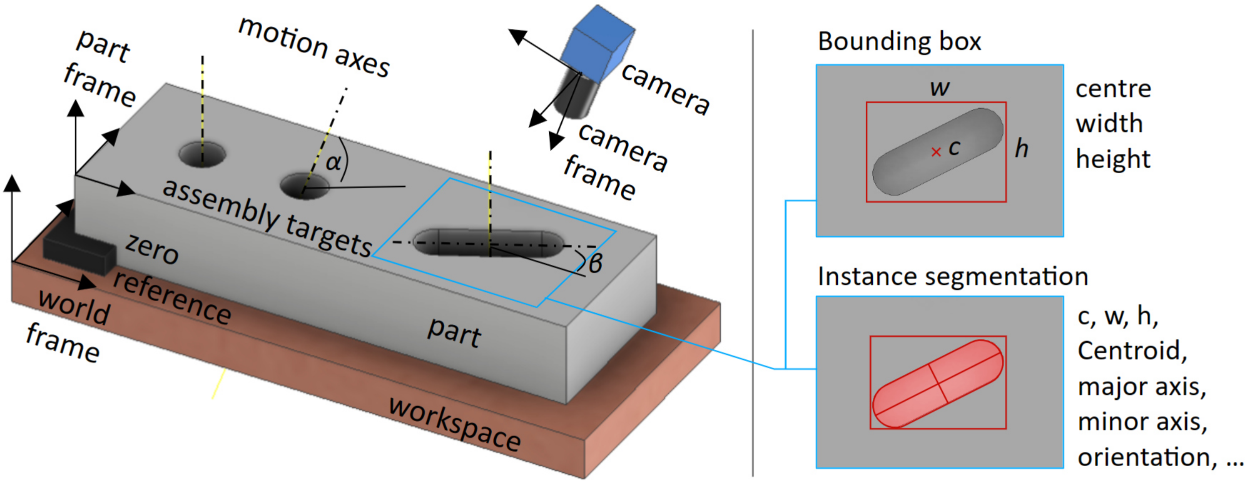


Figure 4.3: Detection (upper right) and segmentation (lower right) of slot assembly target on a 2D assembly part.

In manual processes, determining the missing position of the assembly target is solved by human vision. In this thesis, the target is to replace human decisions and observations with automated systems; thus, a CV system is applied. The required location data can be generated using object detection or instance segmentation (cf. Section 2.2.1). For clarity, a simple 2D part, i.e., all assembly targets lie on the same plane, with different joining motion axes, is depicted in Figure 4.3. Each motion only consists of translation with or without rotation along the same axis. If the camera frame and the joining plane are known in world frame coordinates, the image pixels can be converted into world coordinates cf. (Müller et al., 2019). With instance segmentation, more information on the found objects can be extracted. However, bounding boxes may be sufficient for the missing information depending on the assembly task. The assembly target itself can be determined directly and found by the CV system. Alternatively, other distinctive part features, such as edges or textures, can be used to obtain the data if the assembly target's location is known relative to the selected part features. The accuracy of the visual detection needs to be aligned with the assembly process requirements.

Considering assembly geometry, the locations of the joining planes are given by the part specification in the cases 2D and 2½D-A/B. The planes are defined by the height in the z-direction. Therefore, calculating the distance between the work object and the camera is possible, and the image coordinates can be converted without measuring the distance between the camera and the work object. For the other cases, it is still possible to calculate the joining planes based on part specification. However, a deviation in the x or y direction may change the camera distance. For parts of higher geometric dimensionality, combining the camera image with an additional sensor input, such as a distance measurement, is beneficial.

The same scheme applies to case e) of Figure 4.1. First, the initial position must be determined to start the first linear translation. Then, the directions of the eventual rotation and second translation are necessary. They are given relative to the part frame. Thus, the directions can be derived from the part information. Alternatively, the feedback of the CV system can determine the parameters. Skilled motions exist for the discussed case enabling the execution (Pane et al., 2020). Even in the case of a freeform task (f), initial coordinates are necessary to execute the motion. The path for the subsequential freeform motion can be approximated with support locations detected. Instance segmentation can locate the complete freeform, which can be used to propose the freeform path. In combination with force-controlled contour or surface following, the task can be conducted.

The joining by filling is a particular case in several respects. On the one hand, a filling device must be positioned in the non-contact filling so that material can penetrate the part to be filled (s. Figure 4.1 a). On the other hand, the filling level changes during the filling process. Visual monitoring is required if the level of the filled material cannot be measured directly, e.g., volumetrically or gravimetrically. Since there is no relative movement or contact with the part to be joined, the classic variables of force, displacement, and torque cannot be used. Visually determining such fluid levels is still challenging, and the presented methods in other research are not very robust or flexible (Simeth et al., 2021). With a CV system, the region of interest can be detected and then continuously monitored. Depending on the required return value, classification or regression models are suitable.

In conclusion, it is necessary for all discussed joining cases to directly or indirectly determine the position of the assembly targets, i.e., *location-dependent parameters*. In addition, there are joining processes where certain process states must be recognised. The states are *time-depend* and change during process execution. Hence, the process-relevant parameters are:

- **Location-dependent parameters**
  Parameters relating to specific positions during the joining process
- **Time-dependent parameters**
  Parameters varying with time during the joining execution

Determining the process-relevant parameters for different products with changing characteristics in changing environmental conditions is necessary. Even if further information is required for specific joining processes, recognising the above-mentioned process-relevant parameters using suitable CV models forms the basis. The following part of this chapter presents a procedure for identifying the process-relevant parameters and determining them robustly and flexibly through learning-based CV models.

4.4 Procedure

The basis for developing a suitable learning-based CV model is the knowledge of the process-relevant parameters, which are decisive for the automatic execution of the joining task. They depend on the underlying products to be assembled and the assembly task. Therefore, the developed procedure is split into three categories with five steps, as depicted in Figure 4.4.

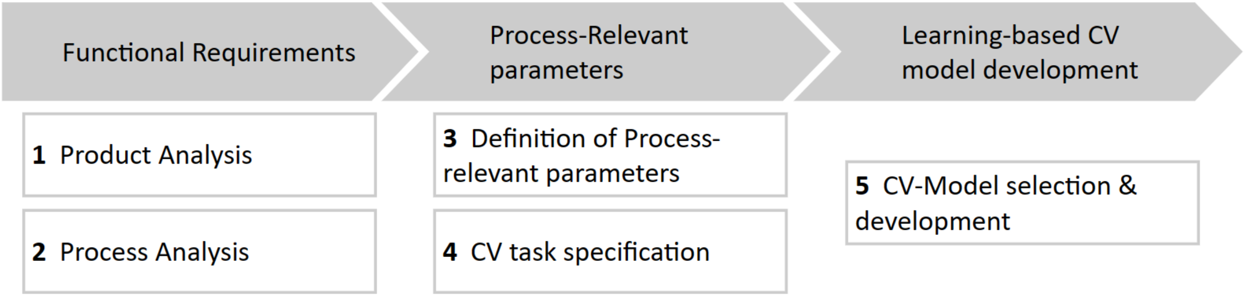

Functional Requirements
Process-Relevant parameters
Learning-based CV model development
1 Product Analysis
2 Process Analysis
3 Definition of Process-relevant parameters
4 CV task specification
5 CV-Model selection & development

Figure 4.4: Procedure to identify process-relevant parameters and to determine them using learning-based CV models.

The starting point of the procedure is the analysis of the product and process. From there, functional requirements that specify the functions required for an automated system can be defined. Furthermore, the data on products and processes unveils additional drivers for variation. With the functional breakdown of the joining process, the process-relevant parameters can be defined in the next step. Knowing the parameters, a suitable CV task is selected. Further, the prediction quality is specified, e.g., the spatial accuracy. Finally, a suitable leaning-based CV model is selected and developed for the defined product-process combination in the last step.

The procedure is currently used in several industrial research projects to develop a robust and flexible CV system, which is the enabling element of planned automation solutions. A selection of the applications is presented in Chapters 5, 6, and 7. The individual procedure steps are discussed in the following sections.

4.4.1 Step 1: Product Analysis

In the product analysis, all relevant product-defining parameters are recorded, for which the term *product characteristic* is used in the following. The result is relevant data about the product variants to be assembled, including all product characteristics that can influence the assembly and a CV system. These characteristics include dimensions, geometry, spatial arrangement, material, surface, structure, and product composition. It is essential to consider all product variations selected for the assembly. The product characteristics can be visualised in a product structure, where dependencies between the product characteristics get more prominent. For example, the position of a particular characteristic on a component may depend on the location of another characteristic. Figure 4.5 shows an example of a generic product structure with characteristics and dependencies. Additionally, it is important to consider future changes expected in the relevant characteristics and incorporate them into the data collection, e.g., upcoming product changes. Depending on the characteristics and the product variants, a morphological box may be suitable to record the characteristics for all product variants.

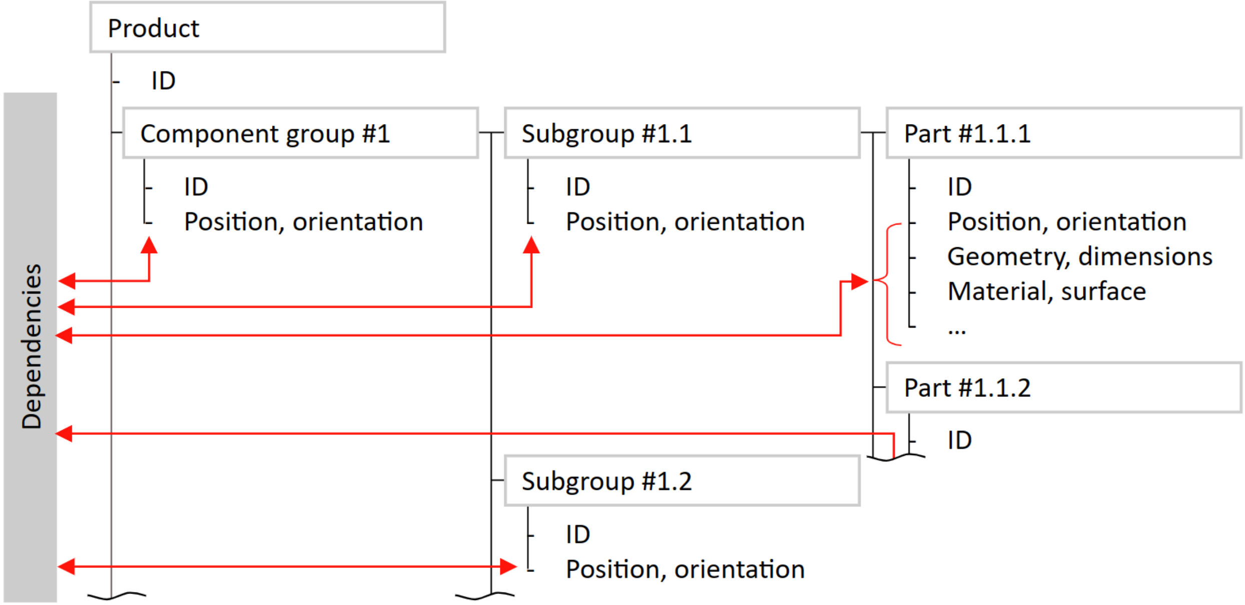
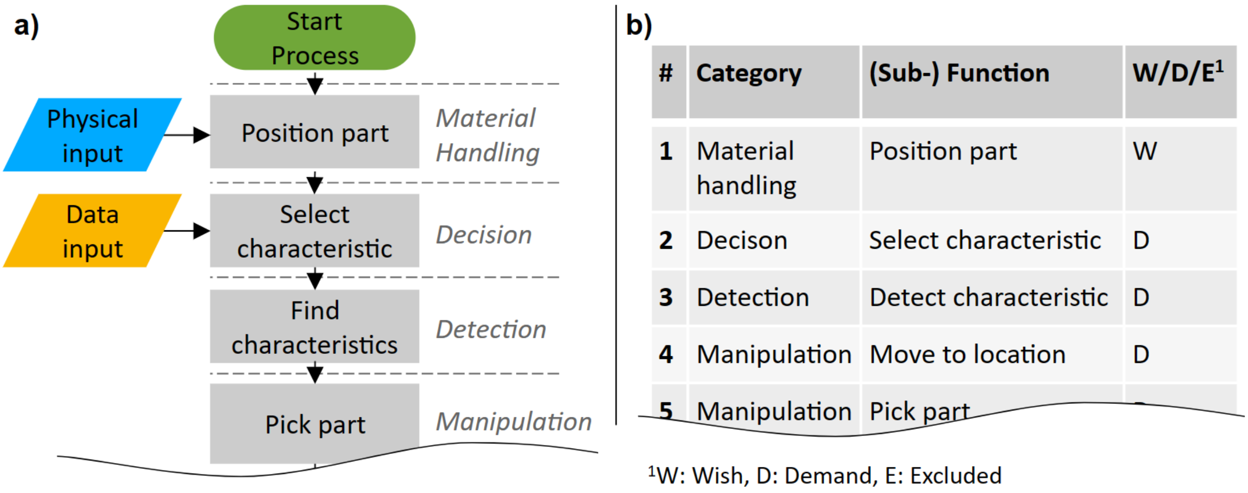


Figure 4.5: Generic Product Structure adapted from (Vajna et al., 2009, p. 45).

4.4.2 Step 2: Process Analysis

Recording the existing manual assembly process is the starting point for the process analysis. In order to cover all process variations, the assembly steps and tasks for the entire product range need to be documented. Subsequently, a generalised assembly process is established, encompassing the entire documented process variation. The individual steps should be described in a solution-neutral and functional manner. Based on the functional analysis defined in VDI 2803, functions are described using a verb and an object, for example, *tighten screw*. The abstract, functional description is independent of the technical implementation and allows various technical solutions. The functional assembly process can be represented in a flowchart for an overview (s. Figure 4.6 a). Different application areas can be identified directly by grouping functions according to defined categories. For example, functions can be assigned to function groups separating physical, visual, and cognitive functions, e.g., manipulation, detection, and decision. Particularly, steps where a human worker takes decisions, retrieves information from instructions or technical drawings or locates and identifies components are highly relevant. These tasks must be performed with sensors in an automated system if the information cannot be provided directly.



#	Category	(Sub-) Function	W/D/E[1]
1	Material handling	Position part	W
2	Decison	Select characteristic	D
3	Detection	Detect characteristic	D
4	Manipulation	Move to location	D
5	Manipulation	Pick part	

[1]W: Wish, D: Demand, E: Excluded

Figure 4.6: a) Functional process flow chart with inputs/outputs and function groups. b) Functional requirements list.

After analysing the functions of the assembly process, the next step is to specify the automation functions. The automation objective always depends on the specific case, and not all functions may be targeted for the automated system. Considering the existing assembly infrastructure, the selected functions need to be examined to determine what is required for their implementation. If necessary, the functions can be subdivided into sub-functions, such as the division of *tightening screw* into *locating screw head*, *positioning screwdriver*, and *tightening screw*. Subdividing the functions is continued until, for the considered process, a suitable degree of abstraction is achieved (Müller et al., 2013). With the complete list of functions, the functional requirements are derived. These requirements are then listed in a requirement catalogue, specifying whether they are mandatory, desired, or if certain functions are excluded from the automation. For example, a requirement list for the flowchart in Figure 4.6 a) is given in b). The function *Position part* is categorised as *material handling* function. The *W* indicates that it is not a fixed request to automate this function. Alternatively, manual material handling is possible. To conduct a picking task depending on certain product characteristics, these must be selected (step 2) and detected (step 3) before the actual function *pick part* can be executed. The picking task itself is further split into two subfunctions. The first subfunction is the movement to the location depending on the detected product characteristic. The second task is the actual picking. Separating the picking from the movement by subdivision is useful if, for instance, one function remains the same for all product variants and another needs to be specifically adapted.

4.4.3 Step 3: Process-Relevant Parameters

The product and process analysis and further specification of the automation functions are the input data for identifying the process-relevant parameters necessary to determine using a learning-based CV system. Each function is categorised in the requirements list (s. Figure 4.6 b). The categories of decision and detection are in focus because every decision and observation made by human workers must be replaced with a digital system.

There are several possibilities for making automated decisions. First, an automated system can decide directly based on available information. This scenario is explained with an assembly where different sequences are permitted and are not specified for the manual process. For example, multiple consecutive functions *screw part* may lead to the same result independently of their sequence; thus, it is not defined for manual operation. If the information is available on how many screws are attached, the sequence can be specified systematically, e.g. by ordinary number in drawing, minimising tool path, randomly, and others. Alternatively, the information is unavailable to the automated system. For decisions depending on product characteristics, further data are required, i.e., process-relevant parameters and an additional function of the group *detection* is necessary. Ideally, all functions are already specified in the necessary level of detail so that all *detection* functions are already identified.

Principally, all functions of the group *detection* can be process-relevant parameters. As outlined in Section 4.3, relevant parameters belong either to location- or time-dependent parameters that must be obtained using a CV system. Detecting product characteristics on a part belongs to location-based parameters. The tolerances of prior processes or clearances between parts may require accurate detection of product characteristics. A manual material handling of the positioning function in Figure 4.6 b), row 1, represents one of such prior processes. However, the information from prior detections may still be valid if certain product characteristics must be determined multiple times in different assembly steps. Furthermore, the detection of product characteristics depending on other characteristics may be obsolete. Overall, there can be multiple process-relevant parameters for a single joining task.

4.4.4 Step 4: Computer Vision Task Definition

Once the process-relevant parameters are identified, their determining method is specified. It is essential to understand what problem shall be solved with the learning-based CV system. Clearly defining the CV task is the baseline to identify the appropriate model. Section 2.2.1 presents the main CV tasks, such as image classification, object detection or semantic segmentation. Each requires specialised models with distinct architectures and capabilities, and defining the type is the first step. Therefore, the required output, that is, the return value of the CV system, is further specified. Everything related to the existence of an object, e.g. a product characteristic, can be solved with classification models. For example, the output of an image classification task is a class label or a probability distribution over multiple classes. The models are usually the simplest among the CV tasks and faster than detection or segmentation models. Thus, they are generally suitable for recognising time-dependent states, although it depends on the specific problem.

Detection and segmentation models provide localisation information of objects in the image, which is necessary for location-dependent parameters. Bounding box prediction is less complex than identifying objects on the pixel level (segmentation) and is preferable if sufficient for the task. The bounding box parameters (centre coordinates, height, width) may facilitate calculating object position and size for symmetric objects such as threads, bores, or holes. Also, the location based on centre coordinates may satisfy the requirements if conclusions to the original shape drawn from an enclosing rectangle are valid. Combining the bounding boxes of multiple objects can provide further information about the whole part, such as type, position, or orientation. With instance segmentation, significantly more data about the instances found is generated. Overall, the model's spatial accuracy must fit the assembly task's technical requirements. In general, the CV model is not a limitation of accuracy when implemented properly. Standard industrial robots usually feature position repeatability in the 0.01-0.1 mm range.[4] Correctly dimensioning the resolution, the CV system performs on a robot accuracy level.[5]

Besides the model output, further aspects of the CV task definition strongly influence the model performance. By defining the problem to the required granularity, the model performance can be influenced positively. A binary classification, e.g., whether a characteristic exists in an image or not, may be sufficient instead of using multiple classes. Furthermore, detecting similar product characteristics, such as different kinds of threads, can be more effective to treat as the same object class than to distinguish each thread type with the CV model. Finally, any practical constraints specific to the application must be considered, such as model size, computational resources, or latency requirements. Once the CV task is defined, the model is developed, as described in the following procedure step.

4.4.5 Step 5: Computer Vision Model Development

The advantage of learning-based CV models is the ability to process images and return selected information without being explicitly programmed. Irrespective of the defined CV task, the iterative workflow to develop the model remains the same. The workflow is depicted in Figure 4.7 and detailed in the following.

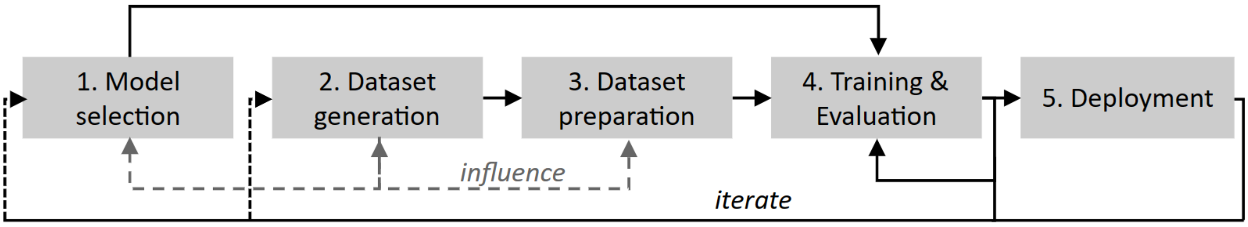


Figure 4.7: Learning-based CV model development workflow based on (Akerkar, 2019, p. 22).

[4] Based on leading OEMs industrial robots portfolio with 1-3 m reach and <200 kg payload.

[5] ABB, Cognex, and Keyence industry experts recommend using an image resolution of 10-20 pixel/mm to achieve robot accuracy level.

Model selection
At first, a suitable learning-based CV model is selected based on the task requirements defined in the previous section. Several criteria are proposed to select the appropriate model. The model's performance is a crucial parameter, and it has increased tremendously with models published in recent years. For classification, detection, and segmentation, challenges exist with large datasets making the different models comparable (e.g., ImageNet (Deng et al., 2009), MS COCO (Lin et al., 2014)). The models are usually compared in the dimensions of accuracy and inference time. Besides the benchmarks, models published in specific research with a task similar to the task at hand can indicate how well a model fits the problem.

Secondly, the popularity in the research community is a good indicator. For popular models, more material to build on and access to support from both developers and users exists. This can simplify the development process. Furthermore, integrating pre-trained models leverages knowledge from large-scale datasets, reduces the required amount of data and is usually much faster than training a network from scratch with randomly initialised weights. Finally, the availability must be verified for each model. Specific environments, operating systems, or hardware requirements may limit the choice of models. Hardware restrictions, for instance, may prohibit the implementation of large models or increase their inference time so that they are practically not usable. However, CV models with conventional feature extractions shall be explored if product characteristics and environment are not subject to change, i.e., the tasks are very similar for all products from a visual perspective.

Dataset generation
The basis for training the model is a sufficiently large dataset. In principle, the more data available, the better. One approach is to collect images from existing sources such as online image repositories, public datasets, or domain-specific databases. For some industrial applications, datasets or images may exist, e.g., for metal surface defects (Zhang et al., 2020). However, there is a high chance that the available data is limited, and a representative dataset of images must be generated. Therefore, factors such as resolution, colour space, background, lighting conditions, object diversity, and any specific visual feature or attribute relevant to the task must be considered. For example, a wrong resolution can result in very small product characteristics in the image regarding their size in pixels, complicating their recognition with a CV model.

Real-world images are collected ideally in the industrial environment, where the CV system shall be implemented. Otherwise, experimental test stands are a solution for creating a similar industrial scenario. Images or extracted keyframes from videos are utilised for the dataset. Alternatively, images can be synthesised with computer graphics techniques. This involves creating images using 3D models, rendering engines, and simulation tools. Synthetic data allows precise control over object placement, backgrounds, lighting, and other factors. However, it may require domain-specific knowledge and tools.

Dataset preparation

After generating the initial set of images, they need to be annotated. The annotation must match the defined CV task and the selected model since different models require different types of labelled training data. Class labels, bounding boxes, or pixel-level masks are added to the ground truth, which is a tedious and time-consuming step. Nevertheless, the accurate association of annotations with the data sample is essential to ensure a high-quality dataset. The image annotation process can be accelerated with specific tools (cf. De Gregorio et al., 2020) or crowd workers. If a model already exists at this time, it can be used to make predictions for unlabelled data. The labelled dataset is then analysed and cleaned. This includes tasks like identifying outliers, inconsistencies and missing values, removing duplicates, correcting errors, and dealing with inconsistent formatting.

By applying data augmentation techniques, the dataset size is increased. Synthetic images are generated with image transformations such as rotation, scaling, cropping, flipping, adding noise, changing brightness, contrast, or saturation, and other methods. The augmentation process adds dataset variability and improves model generalisation. However, only augmentation shall be added that resembles the authentic images. It is unreasonable to generate images that are unlikely to occur during the assembly process. Finally, the dataset is balanced, i.e., levelling the number of images per object class in the dataset and split into training, test, and validation sets so that different kinds of images are evenly distributed across the sets. Common splits are 70-80% for training and the remainder for validation and testing.

Model training & testing

At the end of the previous steps, selected CV models and a meaningful and correctly formatted dataset are available. The training set is used to train the model, and the progress is validated after a defined number of iterations with the validation set. Monitoring the training progress by tracking training and validation metrics identifies signs of overfitting, underfitting or unstable learning. As visualised in Figure 4.7, the training is an iterative process. The training process is conducted with different hyperparameters (s. Section 2.3.2) to optimise the model's performance, evaluated on the test set. Evaluation metrics are used to assess the performance and the ability to generalise on previously unseen data (s. Section 2.3.4). Furthermore, a qualitative analysis shall be conducted by inspecting model predictions and exploring any errors or limitations. Training the model is repeated until the performance on the test set is sufficient for the specified CV task. However, if no satisfying results are achieved, an iteration back to model selection and dataset generation/preparation may also be necessary.

Model deployment

The trained model is then put into production. Therefore, the model's compatibility with the existing environment must be given. For example, adapting the model regarding format, size, inference, or memory requirements may be necessary. However, ensuring compatibility should be considered already during model selection. Integrating the model into the deployment environment, whether it involves a web application, mobile device, or edge computing platform, is critical in ensuring seamless functionality. Additionally, continuous monitoring and updating of the deployed model should be implemented to incorporate new data and improvements, maintaining the model's performance and relevance over time.

4.5 Intermediate Summary

In this chapter, a 5-step procedure is proposed to identify and determine parameters critical for enabling the automation of joining processes in HMLV assembly. The definition of the process-relevant parameters involves analysing typical joining motions in the first step and combining this analysis with the geometric dimensionality of the joining task. Two essential types of process-relevant parameters are identified: *location-dependent* parameters, which relate to specific positions during the joining process, and *time-dependent* parameters, which vary with time during the execution of the joining process. The following procedure is proposed for identifying and determining the process-relevant parameters for a product-process combination.

In Step 1, all relevant product-defining data is recorded and analysed to identify the product characteristics impacting the assembly process or the CV system. Step 2 covers analysing the existing manual assembly process with the target to derive functional requirements for the automation system. The process-relevant parameters are deducted from the requirements list in Step 3. The focus is functions belonging to decisions or observations taken by a human, which a digital system must replace. Step 4 specifies the CV task that can determine the identified process-relevant parameters with sufficient accuracy for the application. Using a learning-based CV model development workflow, the development of suitable CV models is conducted in Step 5.

To use the developed models, they must be robust and flexible enough to meet the requirements set by an HMLV production environment. This is studied in the following Chapter 5 for location-dependent parameters and Chapter 6 for time-dependent parameters.

5 Location-Dependent Parameters

The aim of this chapter is to identify process-relevant parameters and to determine them with suitable Computer Vision (CV) methods focusing on location-dependent parameters with the in Chapter 4 presented procedure. At first, the reference products and assembly process are introduced (s. Section 5.1). Then, based on an analysis of the products and the assembly process, functional requirements for the automation are defined and further specified, and process-relevant parameters are derived in Section 5.2. After introducing the experimental methodology in Section 5.3, different conventional and deep learning (DL)-based CV models are developed and compared in a pre-study regarding their general applicability in the described HMLV assembly scenario (Section 5.4). Finally, the best-performing model is examined regarding its product applicability, flexibility, and robustness in different experiments in Section 5.5.

5.1 Reference Process: Assembly of Lightweight Panels

The flexible assembly of individualised lightweight panels is the selected reference process to showcase the detection of location-dependent parameters. The status quo of the process is manual. Figure 5.1 depicts the assembled parts. The right shows the lightweight panel, into which specific components are inserted, the so-called inserts. The panels vary in dimension, geometry, outer and core material, and, most importantly, the number of bores of different diameters and depths. Each bore fits a specific type of insert. The left of Figure 5.1 shows a standard insert type. As depicted, the component group insert consists of the actual inserted part and a metallic cover, the tab.

The manual assembly consists of the following steps. First, based on a production order with a technical drawing, the worker places the ordered panel on the workbench and commissions all required insert types next to the bench. A specific insert is defined on the drawing for each bore on the panel. Next, the worker selects the corresponding insert for each bore and places it manually in the bore. The worker pushes the insert into the bore until the tab

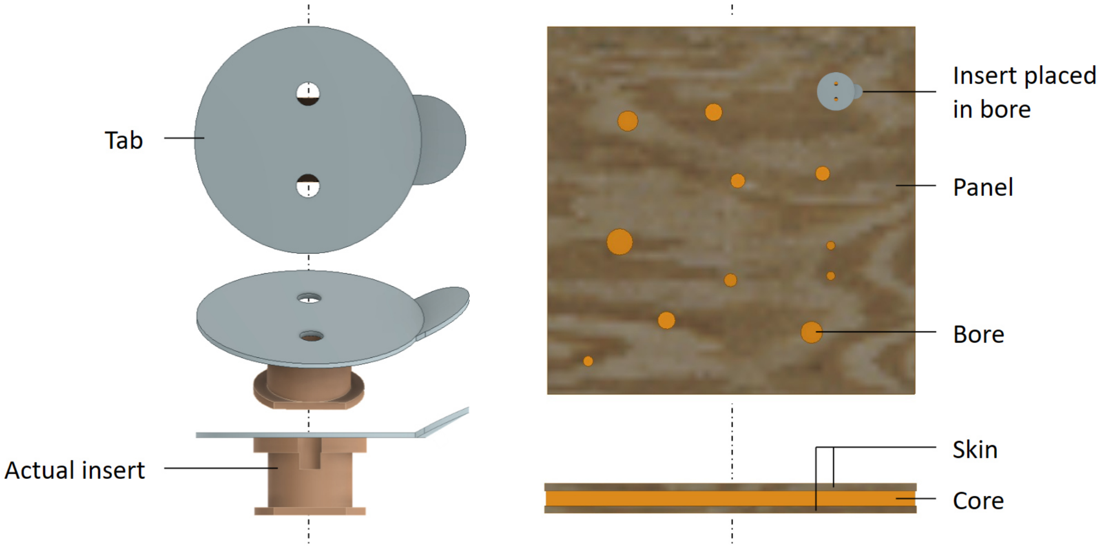


Figure 5.1: Left: Insert assembly with tab and actual inserted part. Right: Panel with bores and a placed insert.

evenly sits on the panel skin. This step is repeated until all specified inserts are placed into the corresponding bores. After inserting all inserts, the panel is ready for the subsequent assembly process (s. Section 6.1).

5.2 Identification of Process-Relevant Parameters

In this section, the product is analysed, and important product characteristics are described. A functional analysis of the process is conducted, and the automation functions are specified. Based on functional requirements, the process-relevant parameters are defined.

5.2.1 Step 1: Product Analysis

The target outcome of the product analysis is all the relevant product-defining data that may affect the assembly process or the CV system. As depicted in Figure 5.1, the assembly process joins multiple inserts with a panel by inserting them into bores. Table 5.1 contains the characteristics of both, the panels (left) and the inserts (right). The panels consist of three layers, i.e., the skin layers and the core. For both layers, different materials exist. The material and finishing strongly impact the appearance of the panel and the bore. The panels are always flat but have different geometries (rectangular, circular, and other) in varying sizes. Each panel can have up to 200 bores, into which inserts must be placed during assembly. Their location can be anywhere on the panel as long as they do not exceed the edge of the panel. The bore diameter and depth depend on the insert type. It is assumed that the bore diameter is 0.5 mm larger than the specified insert diameter.

Table 5.1: Left: Panel characteristics. Right: Insert characteristics.[6]

Panel	
Geometry	Flat, different outlines and curvatures
Footprint	confidential
Height	<50 mm
Skin material	Resin, aluminium, carbon with different coatings
Core material	Glass fibre, aramid, aluminium
Bores	< 200 bores per panel
Bore position	Anywhere on panel but rim
Bore diameter	< 30 mm
Bore depth	5-45 mm

Insert	
Tab diameter	> bore diameter
Tab thickness	0.5 mm
Surface quality	Smooth
Hole diameter	2 mm
Hole distance	5-15 mm
Insert geometry	Cylindrical base shape in different forms
Insert height	< panel height
Insert diameter	< bore diameter
Insert material	Polymer, metal
Weight	< 100g

The inserts exist in over 250 different types, and the characteristics are summarised in Table 5.1 on the right. They consist of an inserted part and a tab made from metal or polymer with a smooth surface. The tabs have two holes necessary for a subsequent glueing process (s. Section 6.1). The distance between the holes depends on the insert type and diameter. The actual inserted part features a cylindric shape. It can have different radii or anchor-like elements. The overall insert weight is light and less than 100 g. The inserts are provided on product carriers named trays, with multiple pieces of the same kind, where the inserts are placed in simplified negative forms with the shape of the tab. The inserts are loosely held in position, and the recess for the flap limits a rotation around the longitudinal axis. The clearance between the tab diameter and the negative form in the tab is estimated to be 2 mm.

The structure of the assembled panel and its components is visualised in Appendix A2.1. The product's basic hierarchy, together with different product characteristics, is given. The product characteristics themselves may relate to each other. For example, each panel bore's dimension depends on the insert belonging to that specific bore. Likewise, the inserts rely on the position information of the panel bores and need to be associated with a particular bore.

[6] Dimensions are changed and may not represent the actual values from the industry partner.

5.2.2 Step 2: Process Analysis

After detailing the products *panel* and *insert*, their joining process is analysed. The target is to utilise a solution-neutral, functional description of the individual assembly steps. The focus is on joining tasks, not handing material to and from the workstation[7]. The functional flowchart of the pick-and-place process is given in Figure 5.2, which is valid for every considered panel. It shows the functions and the required input on a physical or information level. The functions are grouped into categories in the chart to distinguish material handling, joining, and supporting activities. The first step is placing all parts in the workspace. This step belongs to material handling and is, in consequence, not the focus of this thesis. Once the parts are inside the workspace, the worker must select the bore(s) for the next joining step. As input, the worker uses a technical drawing or a work instruction, where the bore positions are specified and insert types are allocated to specific bores. The selection of bores and gathering necessary locations and insert information belongs to the category decision. The next tasks are to identify the correct insert tray, pick the right insert, and fit it into the selected bore, which must be located on the panel. Here, the tasks belong to either manipulation or detection. To ensure correct placement, the worker must push the insert into the bore until the tab fully touches the panel skin, which closes the bore. These steps are repeated until all specified inserts are inserted.

Specification of automation functions

From the described process, the functions *select bore*, *get insert type*, *locate insert on the tray*, *pick up the insert*, *locate the bore on the panel*, and *place insert* form the joining process and shall be automated. No product-specific carrier for the panels is assumed to exist. The automation system can provide the functions *select bore* and *get insert type*. It is defined by the production order. However, the information which insert belongs in which bore must be provided. The insert type cannot be determined by bore diameter and depth since multiple insert types may have the exact outer dimensions but different characteristics. The assembly sequence is set based on the bore and tray locations. It is required that the automated system can detect the insert in the tray and reach the position with a gripping tool. Due to the clearance between the insert and tray, either a sufficiently accurate detection and gripping or a later position correction must be conducted. A corrective measure is, for example, a mechanical alignment of the gripper and insert axis or a recalculation of the target position, including the insert offset. Depending on the selected gripping technology, the contact area

[7] Flexible handling of changing products is also one big challenge in automating HMLV processes

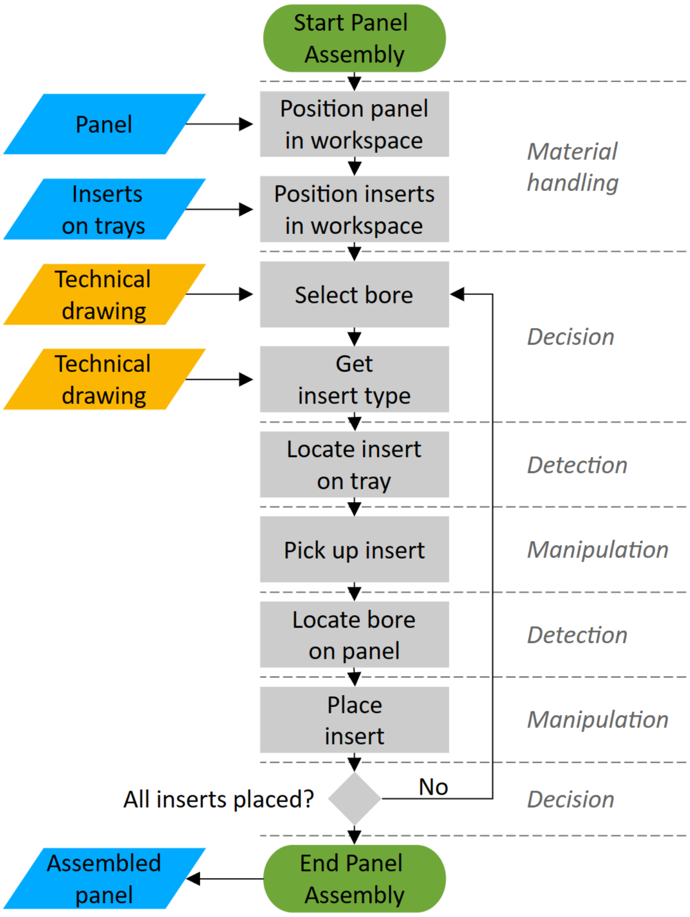


Figure 5.2: Functional flowchart of the manual Pick-and-Place process with function categories. Blue parallelograms indicate physical and orange information inputs/outputs.

between the gripper and tab must avoid the holes, e.g. for vacuum gripping. In this case, the detection must be precise enough for correct gripper positioning.

After successfully picking up the specified insert, the next step is to place the insert in the correct bore on the panel. For this purpose, the precise location of the bore must be detected. Due to the assumption that there is no product-specific product carrier, either the rough position and orientation of the panel in the workspace or another panel reference must be known. A reference is necessary to start the bore detection process and find the targeted bore with high confidence. Once the bore is detected, the insert can be inserted into the panel, assuming that the insert pick-up is accurate enough or that the corresponding error is known or has been corrected. The case where two bores on the panel are so close that the tabs would overlap and jeopardise the bore's closure is excluded from the automation task.

From the discussed process, the following subfunctions result:

1. Get panel data
2. Select bore for assembly
3. Select the corresponding insert
4. Get the position of the insert tray
5. Move to insert tray position
6. Detect insert position on the tray
7. Move to insert position
8. Pick up insert with the tool
   If necessary: correct pick up or determine the offset
9. Determine the position and orientation of the panel in the workspace
10. Detect the bore position on the panel
11. Move the tool with the insert to the bore position
12. Place the insert in the bore on the panel
13. Check correct placement

Functional Requirements

The functional requirements of the automation system result from the specified subfunctions above. They are summarised in the requirements list (s. Table 5.2). For the sake of completeness, also the subfunctions excluded from this research are added to the list. This ensures that the assumptions made for implementing the subfunctions do not collide with the implementation of the other modules. To distinguish those functions, they are labelled as “wish”, “demand”, or “excluded”. The subfunction category mentioned in Figure 5.2 and the required physical devices or information sources are indicated in additional columns.

As discussed earlier, the material handling system is excluded from this analysis. Therefore, it is assumed that the panel and the insert trays are placed according to a known reference. In consequence, the approximate position information of the parts is determined and available. All parts must be reachable by the gripper system. Furthermore, it is assumed that digital datasets exist for each panel. The digital information includes at least basic panel dimensions, the positions of the bores on the panel and their assigned insert type. The panel is not moved during the pick-and-place process and is fixed in the workspace, either manually or

Table 5.2: Functional requirements of the reference pick-and-place assembly process.

No	Category	Subfunction	W/D/E[a]	Source, resource
1	Material handling	Position panel in workspace	E	Material handler, worker
2	Material handling	Position trays in workspace	E	Material handler, worker
3	Decision	Get panel data	D	Production order, product information
4	Decision	Select bore for assembly	D	Path planning based on product information
5	Decision	Get insert type	D	Product information
6	Decision	Get insert tray position	E	Position defined by material handler/worker
7	Manipulation	Move to tray position	D	Manipulator, e.g. robot
8	Detection	Detect insert on tray	D	CV system
9	Manipulation	Move to insert position	D	Manipulator, e.g. robot
10	Manipulation	Pick up insert	D	Gripping tool, manipulator
11	Detection	Determine pose of panel in workspace	W	Positioning of panel to a reference, CV system
12	Detection	Detect bore position on panel	D	CV system
13	Manipulation	Move tool to bore position	D	Manipulator, tool
14	Manipulation	Place insert	D	Manipulator, tool
15	Decision	Check correct placement	W	Physically, visually, dedicated QA

[a] W: Wish, D: Demand, E: Excluded.

automatically, using clamping devices or vacuum tables, for instance. Finally, it is assumed that a manipulation device equipped with a gripper can conduct the required joining motion and insert handling.

5.2.3 Step 3: Process-Relevant Parameters

The product and process analysis and further specification of the automation functions are the basis for identifying the process-relevant parameters necessary to obtain using a CV system. From the functional requirements list in Table 5.2, a digital system must replace the functions in the categories *decision* and *detection*. The decisions in rows three to six can be taken with data available in the system. The production order defines the panel, and in the product information or technical drawing used for the work instructions, the bore positions and related insert types are specified. Reference tray locations in the workspace enable known approximate tray positions. However, the exact location of the insert is unknown due to the clearance between the tray and the insert and must be detected. As visualised in Figure 5.3, the geometry is 2.5D (cf. Figure 4.2) with all assembly targets in plane-parallel surfaces, and the joining motion is linear along the vertical axis (cf. dashed blue arrows). For this task, manipulation systems exist. To enable the motion, also the final placement position is required. The bore position is not determined by the approximate panel positioning and must be detected as well. Checking the correct placement (row 15) after the insertion is marked as a wish. Multiple technical solutions for this quality assurance step do not require a CV system, e.g., path and force-controlled motion to ensure correct placement. However, the step is not necessary to conduct the assembly and thus is not further discussed in this research.

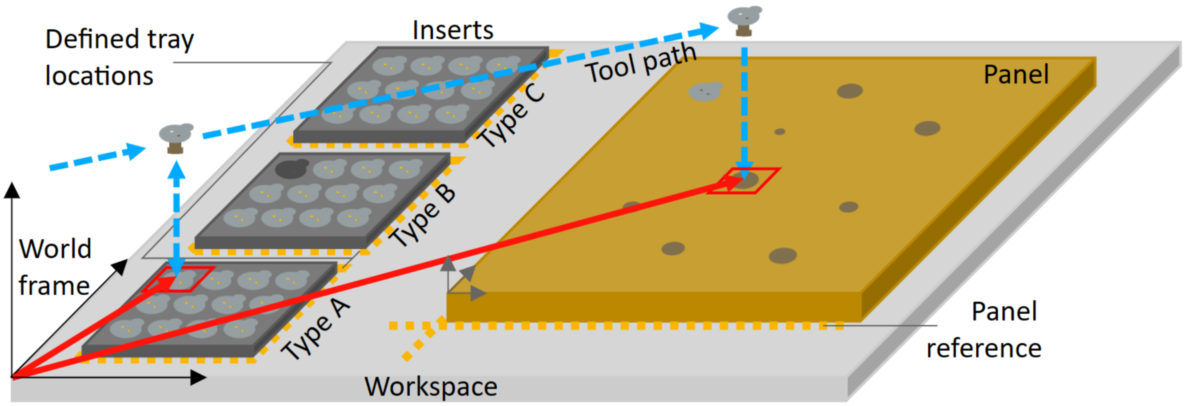


Figure 5.3: Schematic illustration of the pick-and-place process. Red arrows symbolise the location vectors of the insert and bore. Dashed blue arrows indicate the tool path. The dotted orange lines represent reference positions in the workspace.

Overall, there are two assembly steps where the required information cannot be deducted from existing data: the position detection of the insert and the bore. Both are location-dependent parameters. Figure 5.3 also indicates this by the red squares and location vectors. A CV system shall replace the observations done by a human worker. Comparing the two detection problems, bore detection is significantly more challenging. The tabs are standardised parts used across multiple panels with less variation in their appearance. Instead, the panels and their bores change significantly from panel to panel due to different materials of skin and core, surface finishings, etc. In consequence, the detection of the bore is selected for further analysis in the following sections.

5.2.4 Step 4: Computer Vision Task Definition

The location of bores and tabs are defined as process-relevant parameters and bore detection is selected for further analysis. The location of the bore is required to allow a manipulator to move to the bore. The assembly motion is in 2.5D, and positions in the image space (pixels) can be converted into real-world coordinates with the distance between the camera and the object's surface. Thus, it is sufficient to determine the bores in the images and extract their pixel location to calculate their positions in the workspace. From a vision perspective, specific objects, i.e., the bores, must be located in an input image. The bores are always circular and symmetric when observed from above, perpendicular to the panel surface. Two different CV methods are applicable to find the bore positions in the images (cf. Section 2.2.1). By object detection, the CV model predicts the area on the image where the object is located and returns a bounding box enclosing the object. Due to the symmetry of the bore, the bounding box centre represents the bore's centre point location in the image if predicted precisely. With instance segmentation, each pixel of the bore is identified in the image and returned. From the pixel group, the central position can be estimated.

5.3 Experimental Methodology: Detection of Location-Dependent Parameters

Different experiments are conducted to showcase the suitability of object detection models for their application in HMLV processes. Therefore, the following capabilities are investigated:

1. **Applicability** of conventional and learning-based CV models in the HMLV process with changing products.
2. **Flexibility** - Detection performance on changing and new products
3. **Robustness** - Detection performance in different lighting conditions
4. **Accuracy** - Spatial accuracy of the object localisation

The following subsection presents the experimental setup, the experiments, the used datasets and the performance metrics selected for evaluation.

5.3.1 Experimental Setup and Experiments

The overall setup is depicted in Figure 5.4 a). On an ABB IRB 120 industrial robot, a multifunctional end effector (MFEE) is installed. The MFEE comprises a camera module, a glue module, and two ring light modules (s. Figure 5.4 b). The installed industrial camera, type DFK 33GX264 from TheImagingSource, is a 5MP colour camera with a fixed 8 mm lens. The working distance between the camera and the panels is fixed during the individual experiments. The camera axis is perpendicular to the surface of the workspace and runs in automatic mode. In this operation mode, according to the current lighting situation, the camera adjusts parameters automatically, such as exposure time, gain, etc. The camera system is calibrated with a chessboard pattern following the method presented by Zhang (Zhang, 2000). For the calibration, 20 images of a chessboard pattern fixed in the workspace are taken from different angles.

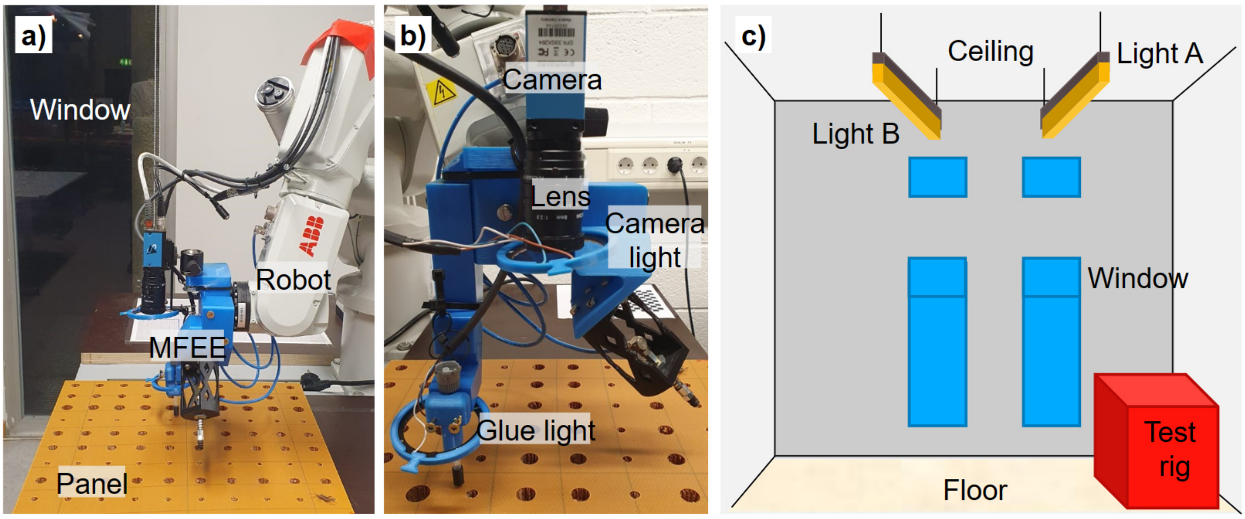


Figure 5.4: (a) Side view of the overall experimental setup with the robot, MFEE, panel. (b) MFEE with camera module and two ring lights. The chessboard at the right back of the test rig is used for calibration. (c) Schematic of the laboratory with light sources. The red cube represents the robot setup. The window in the bottom right is the same window depicted in the image (a).

The ring light concentric with the camera axis is referred to as the camera light, and the other as the glue light. Each ring light consists of 24 RGB-LEDs controllable in intensity ranging from 0-255 for each colour channel. In conducted experiments, the ring lights always have white light with the same intensity for each channel. The experimental setup is placed in the corner of the laboratory next to a window, a natural light source (Figure 5.4 c). In addition, the laboratory has two light sources on the ceiling. The light source closer to the test rig is named room light A, and the other is room light B. The light in the room can be turned on or off.

Four different panels are selected for the experiments (s. Figure 5.5). The panels have different skin and core materials. In consequence, colour, texture, appearance, and reflectivity change between the panels. Additionally, the bores in the panels differ due to the change in material and geometric specification of the core.

The first experiment is conducted to identify suitable CV models that work on changing products. The target is to analyse whether a model can or cannot find bores in changing products. Images of all four panels are used in this prestudy. Different conventional and learning-based models are developed and tested on the images. For the conventional models, features are engineered manually. Therefore, multiple pre- and postprocessing methods are implemented. The DL models are trained on the image data. The performances of all models are compared using the metrics specified in the following section. Finally, the best-performing model in the prestudy is selected for further experiments in the main study.

The main study includes three different experiments. In experiment A, the general model performance is examined. Furthermore, the results from experiment A function as the reference for the following experiments in this study. The second experiment examines the model performance on changing products. Here, the model is exposed to panels not included in the training dataset. The last experiment in the main study analyses the impact of drastic changes in lighting on the model's prediction accuracy. Therefore, different lighting scenarios are created with the light sources in the laboratory, as shown in Figure 5.4 c). A detailed description of each experiment is provided in the prestudy and main study sections.

5.3.2 Datasets

Throughout the PhD project, three different datasets of panel images are developed in the setting depicted in Figure 5.4. The pictures are taken with the camera at altering times of the day, resulting in different lighting situations since the amount and intensity of daylight through the window change significantly. In consequence, indoor lighting is also adapted. It is assumed that external lighting conditions may not be controllable in a potential assembly situation. Therefore, the datasets shall resemble images as they may occur in reality. The images of each dataset are taken perpendicular to the panel surface using the setup depicted in Figure 5.4 b). The camera runs in automatic mode and provides a 5MP colour image. The working distance between the panel and the camera is constant for each individual dataset but changes between the datasets.

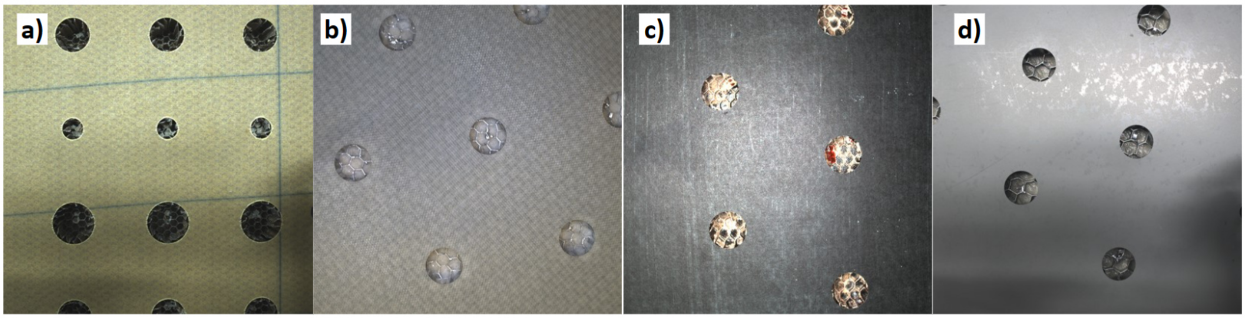


Figure 5.5: Images of the different panels 1-4 used in the experiments.

Dataset A - Initial dataset

The first dataset created in this PhD project comprises all four panels shown in Figure 5.5. However, due to the availability of the products, the amount of images per panel differs. The dataset has 212 images of panel 1, 128 images of panel 2, 182 images of panel 3, and 135 images of panel 4. In total, 657 colour images were taken. Between each image of a panel, the panel's position is altered by translation and rotation. A central square is cropped from the original image, and the resulting 1526 × 1526 × 3 image is saved for further usage to reduce the data size. The images contain a varying number of bores. The lighting situation for each panel is relatively constant and does not vary significantly within each panel subset. The bores in the images are annotated with bounding boxes. Due to a requirement set by the project partner, the labelling was conducted using MATLAB Image Labeler (The MathWorks, 2023b) with an academic license.

Dataset B – Main dataset

Dataset B contains a total of 1200 images. For each panel given in Figure 5.5, a series of 300 images is taken. Between each image, the position of the panel changes (translation, rotation). For each 5MP image, a central square is cropped in $768 \times 768 \times 3$ pixels to reduce the data size. The bores depicted on each cropped image are then annotated with bounding boxes using the Python programme Label Studio (Tkachenko et al., 2022). All bores in all panels have the same label category.

Dataset C – Lighting dataset

The third dataset consists of images of panel 1 only (s. Figure 5.5 a). The four light sources (s. Figure 5.4 c) are changed systematically, and five images are taken for each configuration. Due to the automatic camera mode, the taken images in each configuration change until the camera has fully adapted to the new lighting situation. The panel's position is unchanged throughout the entire process. All images contain the exact same scenery of panel 1 in the workspace but in 24 different lighting situations. Examples of this dataset are provided in Figure 5.6. In total, 980 images of panel 1 are generated. A detailed description of the image acquisition procedure and the experiment is given in section 5.5.2. This dataset is named dataset C.

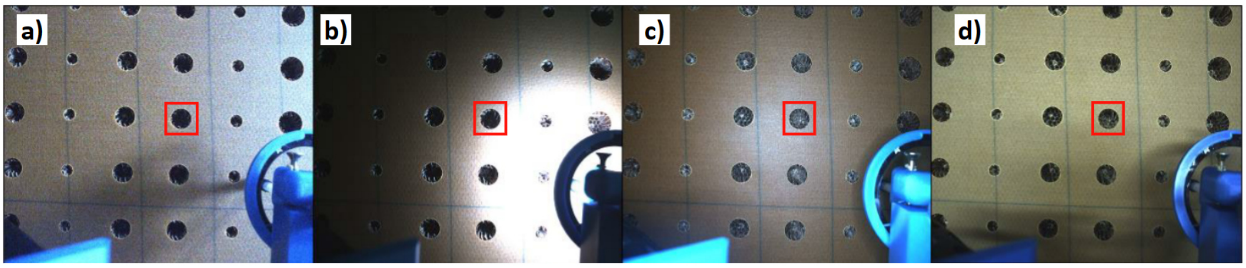


Figure 5.6: Example images of panel 1 in different lighting situations (dataset C). The red bounding box indicates the most central bore used to compare the results.

5.3.3 Performance Metrics

The results of the bore detection and localisation are evaluated with different metrics. The basis for all calculations is the confusion matrix (cf. Subsection 2.3.4). In these experiments, TP is the number of actual positive cases. That is, a bore is detected as a bore at the correct position. FP describes the number of positions or objects incorrectly detected as a bore. FN is the number of bores not detected as such but as background. Cropped bores on the image edge are not counted into TP or FN. TN is not considered since no classification or detection of background exists. Examples for each metric regarding the conventional approaches are given in Figure 5.7.

Calculating the metrics for the conventional approaches is based on a visual inspection of every detection result. The learning-based models are evaluated automatically. A detection is considered as TP if the IoU of the predicted bounding box is equal to or larger than 0.5. With the values for TP, FP, and FN, the metrics precision (P), recall (R), and F1-score are calculated. In addition, the inference time for all models is recorded. Only for the learning-based models, the average IoU is computed. Furthermore, the mAP value at different IoU thresholds is used for performance evaluation.

The primary metrics to determine the localisation accuracy in the main study are calculated

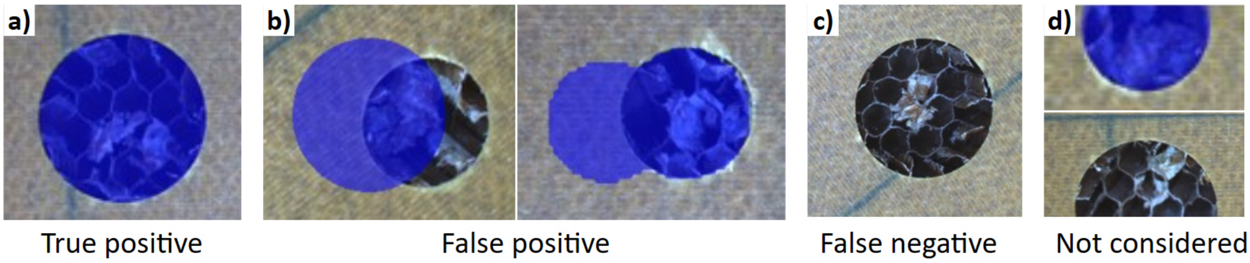


Figure 5.7: Example of a) true positive, b) false positive, c) false negative (no detection), d) cut-off bores on the image edge.

from the bounding box information of the most central element detected on each image. The four parameters of the bounding box are the centre coordinates, the width, and the height (x, y, w, h). Based on x and y, the spatial accuracy of the bounding box on the image is determined by calculating the Euclidean distance between the reference bounding box centre and the predicted centre.

5.4 Prestudy: Applicability of Conventional and Learning-based Computer Vision Models

The prestudy aims to identify CV models suitable for application in HMLV processes, specifically for the selected insertion task with changing products. Therefore, several conventional and DL-based CV models are tested on the detection task to find the bores in different panels and compared, explained in the following sections.[8]

5.4.1 Design of Experiments

The first experiment examines the general applicability of different CV models to detect bores in panels. For the conventional approaches, features must be engineered to predict the bore locations. Based on initial tests, the Circular Hough Transform (CHT) and a gradient edge detector are identified as suitable methods for the bore detection task. However, other tested algorithms indicated a lower performance and were not considered for further evaluation, e.g. Canny edge detection (Canny, 1986) and Gabor filters (Razak and Taharim, 2009).

The general approach for the conventional filters is the same. At first, suitable preprocessing methods are identified and tested. Then, the actual CV method is fine-tuned to enhance the

[8] The results of this section are partly based on a master thesis of Sergio Duarte.

output for the postprocessing step. The models return different return values. The gradient edge detector calculates the changes in intensity values. It returns a greyscale image, while the CHT identifies circles directly and provides centre positions in pixels and the radii of the circles. In the last step, the responses are post-processed to generate a binary mask with bores in the foreground and the rest in the image's background. The DL-based models are selected in the next section.

Dataset A is used for this experiment. The conventional models are set up for each panel type separately. They are implemented, and the parameters are adjusted on 25 images. The performance is then measured based on the prediction of 100 additional images per panel. For the DL-based models, dataset A is split for each panel in a train and test set in the ratio of 80%:20%. The implemented object detectors are trained on the training set, and the performance is calculated based on the prediction results of the test set. Finally, the results of all models are compared using the metrics P, R, F1-score, and inference time. Furthermore, the IoU-score of the learning-based model predictions is analysed.

Augmentation techniques are applied to overcome drawbacks due to an insufficient amount of training data. Dataset A, with 657 images of four different panels, is increased in size with data augmentation to enhance the training outcome of the learning-based models. Out of each image, five artificial images are generated by reflection, rotation, and jittering contrast, brightness, and saturation. The augmented dataset is split into training and testing in the ratio of 80%:20% for each panel. The DL-based models are trained on each panel subset to create panel-specific detectors. Additionally, they are trained on all panels resulting in five detectors per selected DL architecture. Then, the trained models are tested on the related test sets (only one panel or all panels) and analysed using the abovementioned metrics.

5.4.2 Step 5: Model Selection

The learning-based CV model development workflow in Step 5 of the procedure starts with selecting suitable models for the CV task. As defined in the problem definition section, the model must locate circular objects on a plane. Object detection models with bounding box prediction and instance segmentation models with pixel-wise prediction may be applied. The latter models are more complex and require annotation of the ground truth data on a pixel level. This implies a much higher effort on labelling. Therefore, object detection models are tested.

Two different types of object detectors exist, namely, one-stage and two-stage detectors. While one-stage detectors are typically faster and structurally simpler, two-stage detectors offer a higher precision in bounding box prediction. As outlined in Section 2.3.3, one-stage detectors based on the families SSD and YOLO are the most researched models with high performance and good documentation. Thus, detectors of both families are selected for this study. Other models tested are aggregated channel features and cascade detector, which did not yield good results in initial tests and are not considered any further (Hermawati et al., 2018). Improved versions of SSD and YOLO have been published when conducting the prestudy. However, based on the requirements of the study defined by the industrial partner, i.e., to have the algorithms available in MATLAB, earlier versions of one-stage detectors, specifically YOLOv2 and the SSD algorithms, are selected for the work.

Both models use different feature extraction networks. Their backbones are equalised and exchanged with the pre-trained CNN ResNet50. This network achieved a higher prediction accuracy and faster inference time when trained and tested on the ImageNet database compared to the initial CNNs (The MathWorks, 2021). Thus, the exchange is expected to improve the overall detection result of the selected detectors. Furthermore, using the same backbone enables a more compelling comparison of YOLOv2 and SSD because training starts at the same point, and inference time is unaffected by different backbone CNN sizes. Due to the limited training data, a pre-trained version of the ResNet50 is implemented. It has been trained on the ImageNet dataset with over 14 million images and 1000 object classes.

The following subsections explain the implementations of the conventional models CHT and Gradient and the DL-based models SSD and YOLOv2. Then, their detection results on dataset A are compared. Finally, the best models are selected for the main study.

5.4.3 Circular Hough Transform

The first approach in this prestudy is the CHT. To apply the CHT filter, the procedure in Figure 5.8 is followed. Figure 5.9 gives an example of the input image and the output after each step conducted for panel 4.

The first step is the selection of preprocessing filters. The goal of preprocessing an image is to remove redundant data from the images and enhance the features the CHT shall identify.

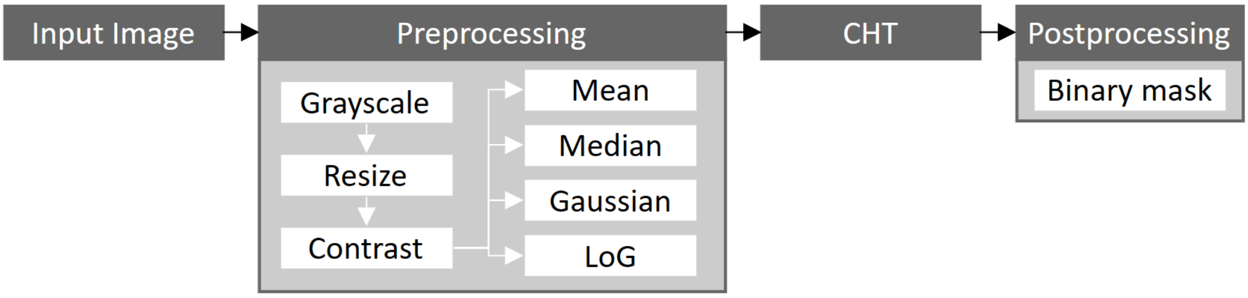


Figure 5.8: Circular Hough Transform approach.

The preprocessing applied for the CHT consists of multiple steps. At first, the input image is transformed to grayscale, reduced in size, and the pixel intensities are adjusted to increase the contrast. The image is resized to decrease the data size and increase the processing speed. Additionally, reducing the image size removes small details due to interpolating the new pixel values. The tested sizes are 25%, 50%, and 100% of the original image. The second part of the preprocessing consists of applying different smoothening filters. Implemented filters are mean filter, median, Gaussian, Laplacian of Gaussian (LoG), and combinations. The standard deviation and the kernel size of the Gaussian are altered as well. The other filters have a fixed kernel size. The output of the preprocessing step is a smoothened grayscale image (cf. Figure 5.9 b). After preprocessing, the details inside the bores are removed, and the contrast of the bore outline is intensified.

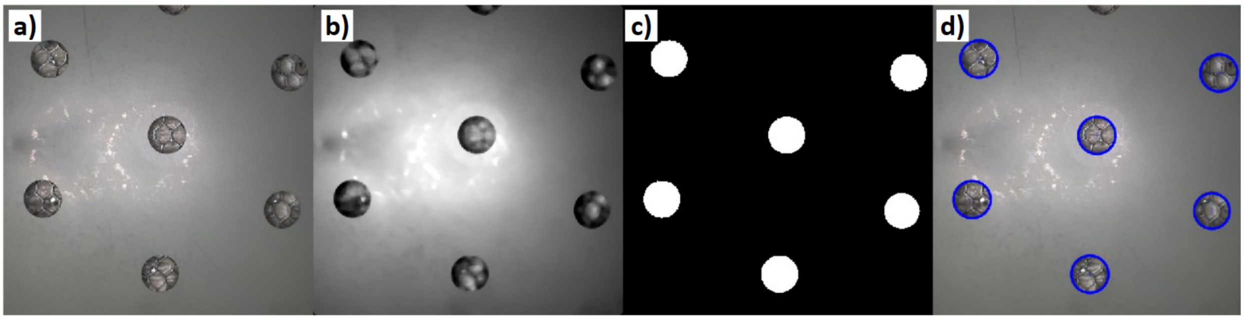


Figure 5.9: Examples for the input image (a), the output after preprocessing (b), the binary mask resulting from the CHT (c), the overlayed mask on the original image (d).

The first part of the CHT is Canny edge detection and binarisation. The binarised image is transformed into the Hough space, where the CHT searches for edge pixels on a circle. The main parameters of the CHT are the size of the circles searched and the sensitivity. The CHT tries for every defined radius to identify circles in the image – the larger the radius range, the longer the processing time. Thus, a well-defined radius range leads to identifying circles in the target size range only and increases the processing speed. The sensitivity defines how many counts on the accumulator are required to accept a value as a circle. A change in the threshold leads to a different amount of circles detected. In this implementation, a higher sensitivity corresponds to fewer circles predicted. With the response of the CHT, a binary mask is created in the postprocessing step. The parameter variation of the whole CHT implementation is given in Table 5.3. The actual bore sizes determine the radius range for each panel. To the minimum and maximum identified bore radius, a safety margin of 10 pixels is added. The range is always adjusted along with the image size in the same ratio. The sensitivity is gathered experimentally and defined for each panel individually. The different image sizes and preprocessing filter combinations are tested for each panel. The setting achieving the highest performance on the 25 setup images is selected for the final evaluation on the 100 additional test images.

Table 5.3: Parameter variation of CHT.

Step	Parameter	Variation			
Preprocessing	Resize	100%	50%	25%	
	Filter	Mean	Median	Gaussian	Laplacian
	Kernel	5x5	5x5	5x5-50x50	3x3
	Sigma			0.5-5	
	Combinations	Single: Mean, Median, Gaussian Combination: Mean & Gaussian, Median & Gaussian, LoG & Mean, LoG & Median			
CHT	Rmin	Panel 1: 45*resize		Panel 2-4: 77*resize	
	Rmax	Panel 1: 135*resize		Panel 2-4: 107*resize	
	Sensitivity	0.90-0.97			

Table 5.4: CHT bore detection results.

Panel	Precision	Recall	F1-score	FPS	Image size	Preprocessing
1	0.986	1	0.993	4.97	50%	Gaussian
2	1	0.980	0.990	14.49	25%	Median
3	0.995	0.954	0.974	0.56	100%	LoG & Median
4	1	1	1	14.31	25%	Median
Avg.	0.995	0.984	0.983	8.58	-	-

The results of the CHT bore detection for each panel are summarised in Table 5.4, with the selected preprocessing method outlined in the last column. The overall performance on every panel is high. For panel 4, a perfect result is achieved – every bore is found without any false prediction. For panels 1 and 2, the F1-score is as well > 0.99, indicating that almost no misclassifications happened. Only for panel 3 does the recall drop, indicating that not all bores are identified. One reason is the distinct structures inside the bores with high contrast, which do not get removed by the smoothening filters. The remaining edges of the structures affect the bore search of the CHT. Still, the performance on panel 3 is high, with an F1-score of 0.974. Comparing the speed, the models for panel 2 and 4 are significantly faster than the others with > 14 FPS. The larger image size necessary for panel 1 and 3 slows down the processing time significantly. Despite the high quality of the results, each panel has a different solution. It is not possible to reuse a model of one panel for another panel. Even a change in the input image size of the same panel may drastically reduce performance. Consequently, a new model setup must be experimentally identified for each change in product or circumstances.

5.4.4 Gradient Edge Detection

The second implemented conventional model is gradient edge detection. In contrast to CHT, the Gradient only identifies changes in intensity values. Thus, the identification and computation of circular objects are subject to pre- and postprocessing. Figure 5.10 shows the Gradient approach.

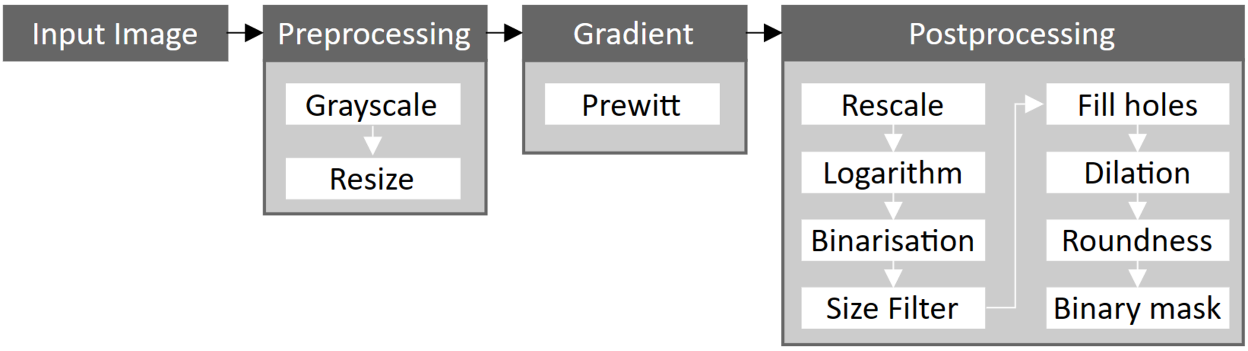


Figure 5.10: Gradient Edge Detection approach.

The preprocessing converts the colour input image into greyscale and reduces the size by four. Without further filtering, the grayscale image's gradient magnitude is calculated using the Prewitt operator. Other tested kernels delivered inferior results. Engineering the bore feature is the main task in the postprocessing step. At first, the intensity values are rescaled to the interval $[1, 2]$ to facilitate a transform with the natural logarithm function (ln). The ln maps the values on the interval $[0, \ln(2)]$ and amplifies small values. Then, the image is binarised using Otsu's method, which finds the optimal threshold based on the histogram (Otsu, 1979). Specifying whether the foreground is darker or brighter than the background is essential and considered in the parameter polarity. Since this changes with the panels, the binarisation must be adapted. In this experiment, foreground pixels for panels 1, 3, and 4 are brighter, and for panel 2, they are darker.

Several morphological operations are conducted on the binary image. A size filter removes all connected foreground pixels smaller than a defined size. Hole filling turns every background pixel surrounded by foreground pixels into the foreground. By dilation, background pixels are expanded into the foreground based on their neighbourhood pixels. The remaining regions in the binary image are checked for their roundness. The final mask is generated by approximating circles for the detected regions and calculating their centroids. An example output image for every step of the Gradient approach is given in Figure 5.11.

Table 5.5: Gradient bore detection results (Simeth et al., 2022).

Panel	Precision	Recall	F1-score	FPS	Image size	Polarity
1	0.964	0.942	0.953	16.64	25%	bright
2	0.887	0.821	0.852	16.19	25%	dark
3	0.929	0.989	0.957	17.27	25%	bright
4	0.934	0.975	0.954	17.81	25%	bright
Avg.	0.928	0.932	0.929	16.89	25%	-

The results for all panels for the Gradient approach are given in Table 5.5. The performance on panels 1, 3, and 4 is also very high, with F1-scores > 0.95. For panel 1, most predictions are correct, but not all bores are detected. For panels 3 and 4, it is the opposite. A larger R shows that almost all bores are identified at the cost of more FP. The model performance deteriorates substantially on panel 2. Here, P and especially R are reduced; thus, more bores are not detected on panel two than on other panels. A reason for the compromised results on panel 2 may be the high similarity of the skin surface and the bottom of the bore, resulting in reduced contrast. The inference time for this detection model is 16-18 FPS and fast enough for many industrial applications. Overall, the only change between the panels is the binarisation's polarity. However, this information can be determined by the panel type and given as an input parameter, making the same model applicable to different panels.

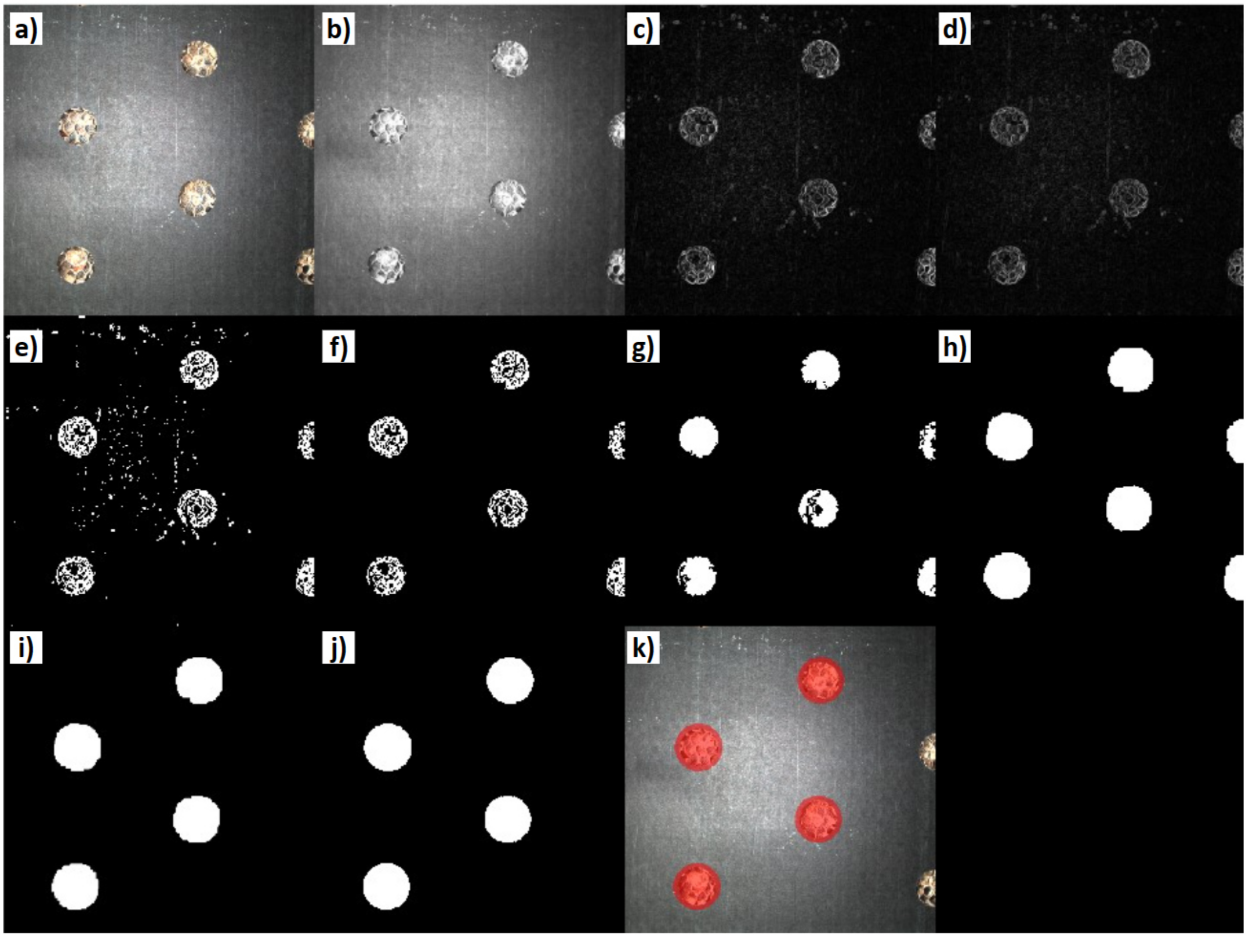


Figure 5.11: a) input image, b) after greyscale and resize, c) gradient magnitude, d) rescale and ln function, e) binarisation, f) size filter, g) hole filling, h) dilation, i) roundness filter, j) binary mask, k) original image with the overlayed mask.

5.4.5 Single Shot MultiBox Detector

The results of the implemented SSD detector with the modified backbone to pre-trained ResNet50 are given in Table 5.6. The parameters P, R, F1-score, and inference time are provided for the positive detection threshold of IoU = 0.5. Similar to the conventional approaches, P, R, and F1-score indicate how well the bores are predicted on each panel. The IoU-score describes the quality of the bounding box prediction. The higher the score, the more accurately the bounding box matches the ground truth annotation. For the single panel detectors, the F1-score is, on average, >95%, and both P and R are on a high level. The worst prediction is achieved on panel 2 because R drops below 0.9. The predictions on panel 1, 2, and 4 are exact, with P close to 1. On panel 1 and 3, most bores are identified. The location accuracy and the inference time are similar for all individual detectors and lie at 0.88-0.9 and 32-33 FPS, respectively. Two different detectors are trained on complete training dataset A, i.e., a category-agnostic with one class for all bores and a detector with one class for each panel's bores. The category-agnostic detector achieves similar results compared to the panel-individual detectors. The IoU-score is improved by a slight margin. The small reduction in inference time is probably linked to the higher number of predictions. The results of the SSD detector with four bore classes deteriorated significantly, and this detector is not considered any further.

Table 5.6: SSD bore detection results (Simeth et al., 2022).

Panel	Precision	Recall	F1-score	FPS	IoU
1	0.976	0.970	0.973	33.02	0.891
2	0.986	0.886	0.933	32.24	0.879
3	0.943	1	0.938	32.79	0.886
4	0.973	0.905	0.938	32.42	0.894
Avg.	0.969	0.940	0.944	32.62	0.888
All, one class.	0.917	0.972	0.944	34.81	0.898
All, four classes.	0.970	0.345	0.509	34.01	0.898

5.4.6 You Only Look Once

In this study, YOLOv2 with a modified backbone is applied, trained, and tested like the SSD detector. Table 5.7 contains the results of the individual detectors and the detectors for all panels. R of the individual detectors is always close to one and more significant than P, meaning that more or less all bores are identified, but some FPs exist. Only for panel 3, R somewhat decreases, but paired with the significantly higher P, the YOLOv2 detector achieves the best performance on panel 3 in terms of the F1- and IoU-score. All values for P, R and F1-score are > 0.9. The bounding box localisation is lowest at panel 2. Overall, all individual detectors are performing well. The detector trained on all panels with different bore categories achieves a similar performance very close to the average of the individual ones with improved IoU-score and FPS. The localisation of the detector with four bore classes is 0.891 and higher than the average of the individual ones. The class-agnostic detector performs significantly worse and is not further considered. The inference time is 33-36 FPS and similar for all detectors.

Table 5.7: YOLOv2 bore detection results (Simeth et al., 2022).

Panel	Precision	Recall	F1-score	FPS	IoU
1	0.900	1	0.948	35.54	0.861
2	0.935	1	0.964	35.31	0.835
3	0.978	0.989	0.984	33.14	0.900
4	0.929	1	0.963	35.87	0.874
Avg.	0.936	0.997	0.965	34.97	0.867
All, one class.	0.822	0.888	0.854	35.78	0.740
All, four classes.	0.928	0.999	0.962	36.01	0.893

5.4.7 Comparison

All presented approaches achieve good results on the selected dataset A. The classical approaches, CHT and Gradient, are evaluated manually based on visual inspection. On the other hand, the learning-based detection models SSD and YOLOv2 automatically assess the performance using image annotations. The metrics F1-score and FPS are utilised for the comparison since they are the same for all models. The F1-scores across all panels are given in Figure 5.12. Regarding the learning-based models, *single* refers to detectors trained only on one panel. The term *one class* describes the detector trained on all panels with only one category for all bores, thus the category agnostic detector. The detector with a bore category for each panel is named *four classes*.

All selected models perform well on the chosen dataset. The average and median F1-scores of all models are above 0.92. The CHT achieves the best results in terms of F1 reaching the highest average and median and the smallest range of variation in performance on each panel. For the gradient edge detector and the SSD trained on all panels, it is visible that they don't perform well on every panel. While the median is close to 0.95, similar to the other detectors, the performance for one panel drops below 0.87. The results of the single SSD and YOLOv2 detectors and the YOLOv2 trained on all panels do not differ significantly. However, the YOLOv2 models achieve a slightly higher F1-score with less scattering than the SSD.

To ensure correct training of the SSD and YOLOv2 models, each appearance of the target objects, i.e. the bores, is labelled. This includes the incomplete bores at the image edges as well. Otherwise, there exists the risk of introducing FNs to the models. Therefore, these cut

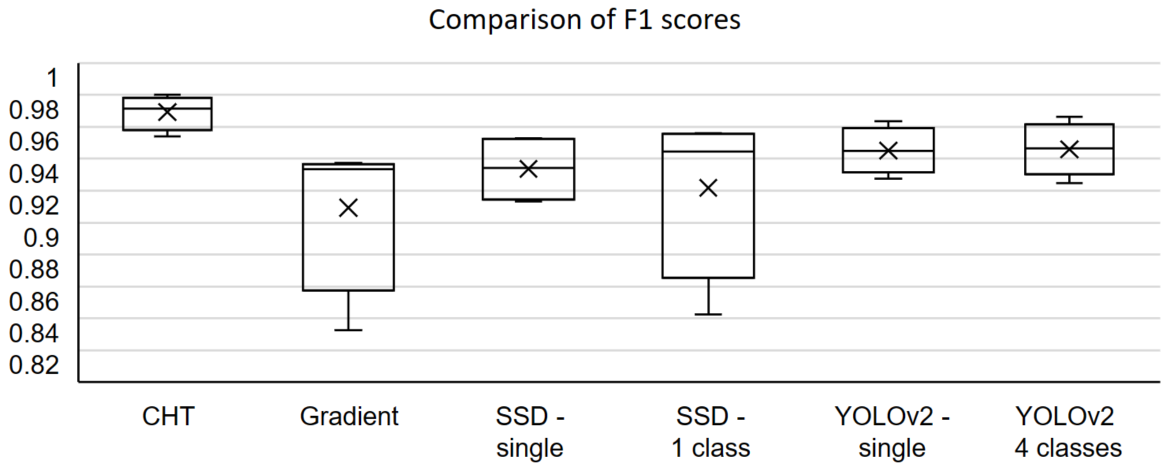


Figure 5.12: Comparison of the F1-scores of all detectors based on all panels.

bores are also included in the evaluation of SSD and YOLOv2, which is opposite to the assessment of the classical models (cf. Figure 5.13). A random sample visual inspection of the SSD and YOLOv2 detection results unveiled that mispredictions occur almost exclusively on these rim bores. Considering this, the learning-based models perform better as depicted if the cut-off holes would be neglected for evaluation. As explained, this is due to the training process not being recommended.

Comparing the inference time, SSD and YOLOv2 outperform conventional models by far. They can run at a rate of >33 FPS, with YOLOv2 being the fastest. This is about twice as fast as the conventional models. The gradient edge detection achieves 17 FPS and the CHT 14 FPS. The most significant impact has the input image size. Since the CHT does not run for all panels on the compressed image, the processing speed reduces to less than 1 FPS if the original image is input.

The detection of the bores consists of the classification of objects and their localisation. The localisation precision has an essential role in the overall prediction quality of the detector. The localisation of the conventional models is evaluated by visual inspection. For the SSD and YOLOv2 detector, the localisation is assessed using the average IoU-score of all correct predictions. This includes as well the cropped bores at the sides. The localisation results show that for both SSD and YOLOv2, the models trained on the complete training set have a higher localisation accuracy and a reduced variation over the four panels. As a result, they are less affected by a change in a panel and provide more robust predictions. Comparing SSD to YOLOv2, the results indicate a better localisation performance of the SSD detectors.

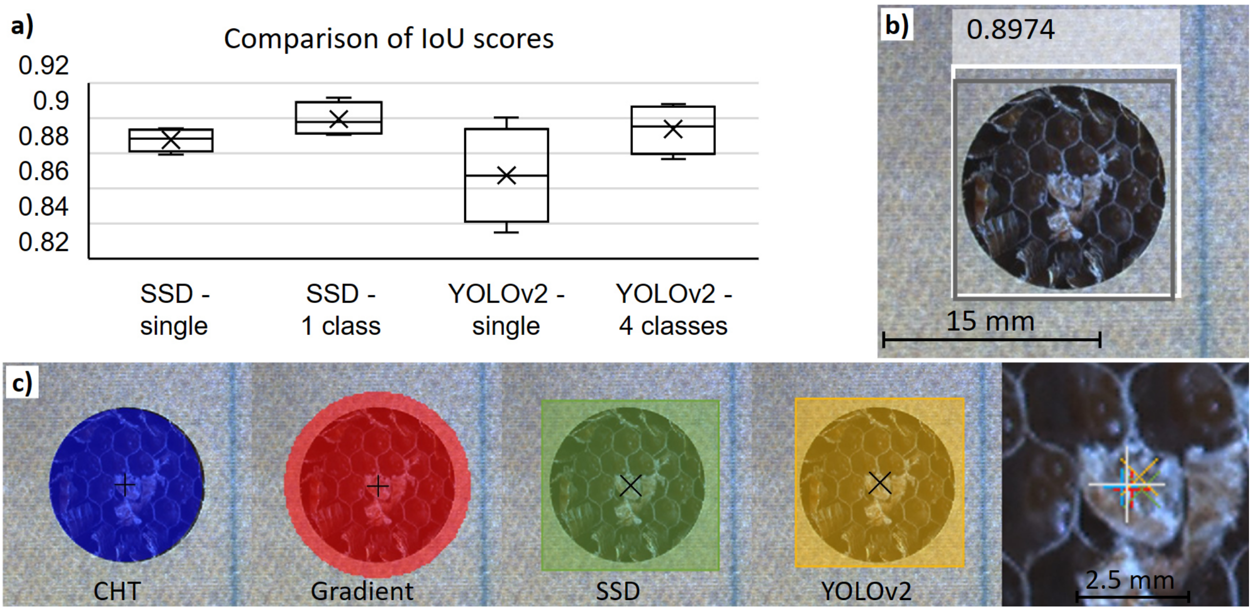


Figure 5.13: a) Comparison of the IoU-scores of the SSD and YOLOv2 detectors. b) Example average bounding box prediction (white) with an IoU of 0.897 compared to ground truth (grey). c) Comparison of CHT, Gradient, SSD, and YOLOv2 bore detection of the same bore. The last image highlights the centre positions of the predictions compared to the measured centre (large white plus).

However, considering the models trained on all panels, their average IoU values are between 0.89-0.9, and the difference is insignificant. An example of an average predicted bounding box with an IoU of 0.897 is given in Figure 5.13 b).

Figure 5.13 c) shows the detection results of all four approaches for the same bore. For each model, the binary mask or the bounding box is overlayed with the original image and the centre of the object is indicated with a plus for classical models and a cross for learning-based models. The last image of the series is a close-up of the bore centre to highlight the predicted centre positions. All predictions are within 7 pixels from the measured centre point indicated by the large white plus. For this specific example, the centre point predictions of the conventional models are more accurate. However, there is no evaluation of the conventional models' overall localisation quality since it is only based on a visual inspection. The IoU values for the learning-based models match the average for the predictions. Thus, most SSD and YOLOv2 predictions probably have similar localisation precision.

The final evaluation of all models and their fit for the selected detection task is shown in Figure 5.14. All models classify and localise the bores well on each panel, with a slight ad-

	CHT	Gradient	SSD	YOLOv2
Classification	↑	↖	↑	↑
Localisation	↑	↑	↖	↖
Inference	↙	←	↑	↑
Adaptability	↓	←	↖	↑
Pre/Post-processing	↓	↙	↑	↑
Data preparation	↖	↖	↓	↓

Legend:

↑	Excellent
↖	Good
←	Average
↙	Poor
↓	Very poor

Figure 5.14: Final evaluation of the models fit for the detection task.

vantage towards localisation for the classical models. However, SSD and YOLOv2 struggle mainly on the cut-off bores close to the image edge. Exactly these bores are not considered for CHT and Gradient evaluation. Most important for the task is the flexibility of the models to a product change. To enable the application of the CHT and Gradient, several parameters in preprocessing, primary processing, and postprocessing must be adapted for every model. This feature engineering is obsolete for SSD and YOLOv2. Additionally, a change in input image size may lead to no CHT or Gradient predictions, while SSD and YOLOv2 are indifferent. The main disadvantage of the learning-based models is the generation of dataset annotations. Overall, the SSD and YOLOv2 better fit the given task. Considering that already newer, improved versions exist, the learning-based detection approaches are selected for the main study in the next section.

5.5 Main Study: Flexible & Robust Detection using YOLO

Neural network-based object detection models show a superior fit for the application in HMLV processes. In this section, such a detection model is investigated in several experiments which shall resemble the actual HMLV production scenario. The following model capabilities are investigated:

- **Flexibility** - Performance on changing and new products
- **Robustness** - Performance in uncontrolled lighting environments
- **Accuracy** - Spatial accuracy of the object detection

Based on the prestudy, a new, state-of-the-art learning-based detection model for the defined assembly process is implemented following the procedure explained in Section 4.4.

5.5.1 Step 5: Model Selection

In general, one-stage detection models are getting increasing attention due to their competitive performance, real-time capability, and reduced complexity over two-stage models. For the initially implemented models, SSD and YOLOv2, newer versions exist. The last update of the SSD model was published in 2019 by Yi et al. (Yi et al., 2019), whereas the latest YOLOv8 was launched in early 2023. However, the models YOLOv6, v7, and v8 were just published at the time of experiment conduction. In consequence, they are not considered for the model selection. Available models were YOLOv3, YOLOv4, YOLOv5, YOLOX, YOLOR, FCOS, and others, which all outperform the models YOLOv2 and SSD from the prestudy to a large extent. Wang et al. show in a benchmark that the YOLO family has the best performance-inference ratio (Wang et al., 2022). On MS COCO 17 dataset, the best YOLO models YOLOv5 and YOLOR have similar performances with mAP of 55-56% and framerates of 34-38 FPS for the large models. Due to the superior resources and support provided by Jocher et al. for the model YOLOv5, it is selected for the main study (Jocher et al., 2022).

The object detection model YOLOv5 is implemented in version v7.0-70-g589edc7, and the pre-trained weights yolov5x.pt are used as the basis for the training of object detectors (Jocher et al., 2022). All experiments are run in Google Colabs using the PyTorch framework and GPU (Tesla T4). A stochastic gradient descent optimiser is selected with a learning rate of 0.01 (default). Each model is trained for 100 epochs with a batch size of 16. The images are passed to the model in size 640x640x3. Each image is augmented whenever it is loaded. The augmentation includes translation, scaling, vertical reflection, image mosaicking, and jittering of the hue, saturation, and value. After each training epoch, the model fitness is calculated. The fitness is the weighted average of the metrics mAP50 and mAP50-95 in the ratio 10%:90% calculated on the validation dataset. The model with the highest fitness is selected for the experiments.

5.5.2 Design of Experiments

The functionality of the state-of-the-art object detector YOLOv5 is investigated in three experiments. First, the models are trained in Experiment A, and their general performance is examined. Then, their performance and localisation accuracy is analysed in Experiment B for different products and in Experiment C in changing lighting conditions.

Experiment A: General Performance
The first experiment is designed to assess the general performance of the selected CV model on dataset B. The objective is to generate a performance reference and to identify if any challenges exist with the model training. Therefore, dataset B is divided into training, validation, and test sets in the ratio 80%:10%:10% evenly over the four panels. Then, five different reference models are trained. The first model is trained on the complete training set with all four panels. The other four models are trained on three panels, with one of the four selected panels removed from the dataset B. The performance of the models is determined with the validation sets.

Experiment B: Product Flexibility
The second experiment examines the models' product flexibility by testing the performance of the trained models from Experiment A on changing and new products. Here, the objective is to identify the model's ability to generalise and make predictions on unknown and new data. Therefore, the following steps are conducted:

1. The model trained on all panels is tested on the test set from dataset B.
2. The models trained on three panels only are tested on a reduced test set containing only the three panels used for training.
3. The models trained on three panels are tested on all panel images of the fourth panel.

Furthermore, the precision of the object localisation of the trained models on different subsets of dataset B is investigated. First, the aim is to quantify the performance of YOLOv5 on new products when trained on existing products only. Furthermore, the possibility of using the information from the bounding box for subsequent manufacturing processes, i.e., the insertion of the insert, shall be showcased. Therefore, the model predictions on the test sets of dataset B subsets are compared with the ground truth annotations of the test sets. The parameters IoU of the bounding boxes of the most central bore and the Euclidean distance between their centre points are used for comparison.

Experiment C: Robustness
The third experiment simulates working in changing lighting conditions. Images of the exact same scenery with changed lighting configuration are analysed by the model trained in experiment A. This experiment evaluates the impact of changing lighting situations on the ability of the model to identify product features, in this case, the bores, and to predict their location precisely. In contrast to experiments A & B, full-resolution images (2448 × 2048

pixels) are fed to the model this time. The detection results for the most central object are used for comparison (s. Figure 5.6).

Four different lighting cases can be created by alternating the room light status. Additionally, the ring lights can be turned off or turned on at a defined light intensity. The light intensity of the ring lights is incremented by 25 for each channel, totalling ten different intensities, i.e. $\{(25, 25, 25), (50, 50, 50), \ldots, (250, 250, 250)\}$. An overview of the lighting scenarios used is given in Table 5.8. The images were taken with the following procedure:

Image generation procedure for Dataset C

```
For each Case 1-4:              # altering room lights
    For each Subcase 1-6:           # altering ring lights
        Do until specified intensity reached:
            Take five images
            Increment ring light(s) intensity by 25 units
```

YOLOv5 object detector is run on each image, and the most central bore's bounding box parameters are compared to a reference bounding box from the annotation. The model trained on the entire dataset B is applied for object detection.

Table 5.8: Overview of the lighting scenarios by lighting case.

Lighting case	Room A	Room B	Camera	Ring	# images
Case 1	Off	Off	-	-	245
Case 2	On	Off	-	-	245
Case 3	Off	On	-	-	245
Case 4	On	On	-	-	245
Subcase 1	-	-	Off	Off	5
Subcase 2	-	-	25-250[a]	Off	50
Subcase 3	-	-	Off	25-250	50
Subcase 4	-	-	25-250	25-250	50
Subcase 5	-	-	On (=250)	25-225	45
Subcase 6	-	-	25-225	On (=250)	45

[a] 25-250 and 25-225 indicate incrementing the intensity from 25 in 25-steps to 250 or 225.

5.5.3 Experiment A: General Performance

In experiment A, all object detection models used in the main study are trained and validated. From dataset B, five different training subsets are created and applied to train five YOLOv5 object detectors. The models and datasets are named based on the panels used for training. For example, dataset D1234 indicates that images from all four panels are included. Therefore, the model M1234 is trained on dataset D1234, including all 1200 images of dataset B. The dataset D134 consists of the images of panels 1, 3, and 4, in total 900, and is used to train model M134. The specification of the datasets is given in Table 5.9. Although 300 images per panel are taken, the number of instances on panel 1 is higher than the others. Due to a different bore pattern, panel 1 features approximately ten bores per image. The other panels 2, 3, and 4, have six on average.

Table 5.9: Datasets used in Experiment A drawn from dataset B.

Dataset	Total images	Instances	Panel 1	Panel 2	Panel 3	Panel 4
D1234	1200	8562	X	X	X	X
D123	900	6676	X	X	X	
D124	900	6747	X	X		X
D134	900	6712	X		X	X
D234	900	5551	X	X	X	

All dataset have the split training-validation-testing: 80%:10%:10%.

Table 5.10: Training results of YOLOv5 object detectors.

Model	Validation		Best	Time	Precision	Recall	mAP50	mAP50-95
name	images	instances	epoch	[min]				
M1234	120	841	98	122	**0.991**	0.992	0.994	0.978
M123	90	648	100	97.6	0.989	0.996	0.993	0.975
M124	90	660	100	98.6	0.990	0.991	0.993	0.974
M134	90	648	100	97.7	**0.991**	**0.997**	0.994	**0.98**
M234	90	567	**87**	**94.7**	0.988	0.996	0.995	**0.98**

Model: YOLOv5 v7.0-70-g589edc7. Weights: yolov5x.pt. Augmentation: hyp.scratch-med.yaml. Python version: 3.8.16. Framework: torch-1.13.0 + cu116. GPU: Tesla T4.

The training results are given in Table 5.10. All models are trained with GPU for 100 epochs. The training took 122 minutes on the complete dataset B training set. For the reduced training sets with three panels, the training times are around 98 minutes. The model with the highest fitness is selected for later application. For the models M1234 and M234, the best results are achieved after epochs 98 and 87, respectively. The others achieve the highest performance after 100 epochs of training. The results of all trained models show excellent performance on precision, recall and mAP50, which are all larger than 0.988. Almost all objects are found and predicted correctly. In addition, the models also detect objects at high IoU thresholds accurately. All detectors have an mAP50-95 equal to or larger than 0.974. The inferior predictions occur on split bores at the image edge, similar to the mispredictions in the prestudy.

The learning curves for all models are given in Figure 5.15. Each model's precision, recall, mAP, box loss, and object loss are plotted for each epoch. The curves are close to each other and show a similar profile. The box and object loss decrease with a high gradient until epoch 12. Then, the curves become flatter. By then, precision, recall, and mAP50 are already close to 1 and do not increase significantly throughout the remaining epochs. Both losses decrease slower until epoch 100, while the mAP50-95 continuously improves. All curves indicate that there are no issues throughout the training process. After successful implementation and training with promising results, the models are ready for application in the following experiments.

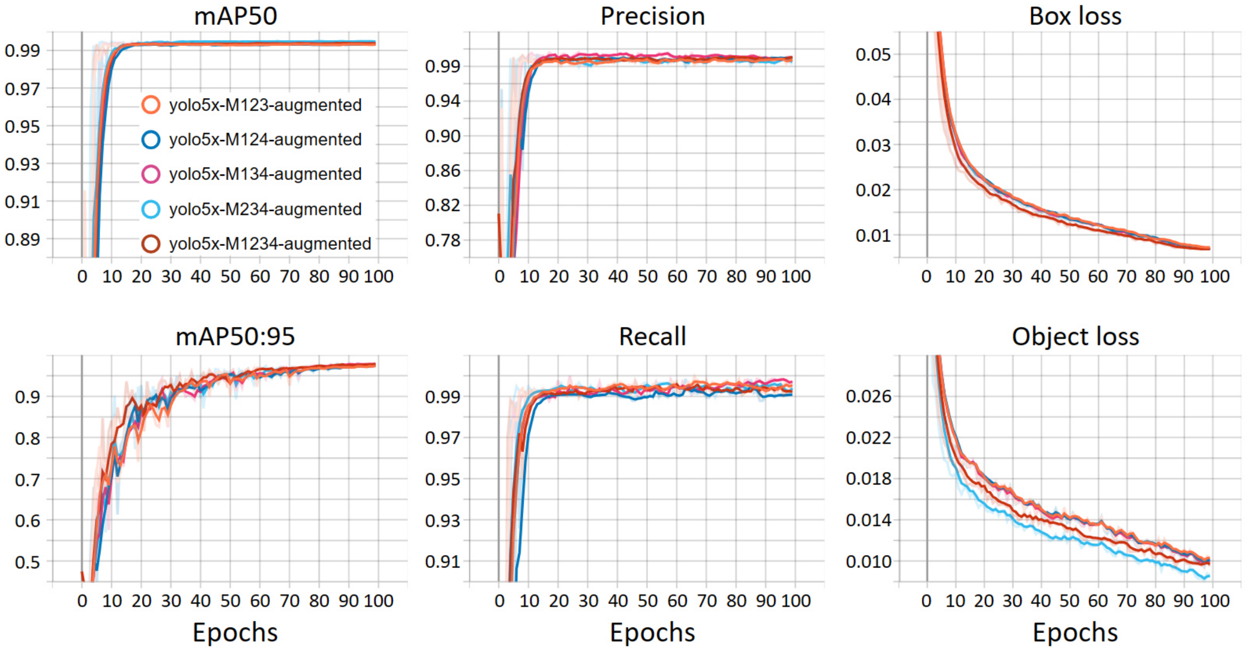


Figure 5.15: Learning curves (with smoothing) for the parameters mAP50, mAP50-95, precision, recall, box loss, and object loss for all models.

5.5.4 Experiment B: Performance on Changing Products

The ability of the models to generalise is analysed by running them on new, unknown data. Therefore, the performance on the test sets is determined. Additionally, the models trained on only three panels are tested on the complete set of images of the panel removed from training data (s. Table 5.11). The performances of all models are again high. The models produce similar results on both the validation and test sets. Models trained only on three panels indicate no bore detection issues on the unknown fourth panel. The metrics P, R, and mAP50 are nearly unchanged on the panel images not exposed during training. Only at high IoU thresholds a marginal drop in mAP50-95 is visible. Still, mAP50-95 results equal at least 0.945 when predicting bores on the unknown panel. The inference time varies between 50 and 56.8 ms.

Table 5.11: Testing results of YOLOv5 object detectors.

Model	Set	Images	Instances	Precision	Recall	mAP50	mAP50-95	Inference
M1234	D1234	120	836	0.99	0.999	**0.995**	**0.979**	54.0 ms
M123	D123	90	652	0.989	0.997	**0.995**	0.978	53.0 ms
	D4	300	1886	**0.993**	0.996	**0.995**	0.976	53.5 ms
M124	D124	90	651	0.989	**1**	**0.995**	0.977	54.3 ms
	D3	300	1815	0.985	0.994	**0.995**	0.965	**50.0** ms
M134	D134	90	641	0.992	0.996	**0.995**	0.975	56.8 ms
	D2	300	1850	0.989	0.994	**0.995**	0.945	55.2 ms
M234	D234	90	564	**0.993**	**1**	**0.995**	0.978	54.7 ms
	D1	300	3011	0.987	0.996	0.994	0.958	54.0 ms

The localisation accuracy is determined by comparing the predicted and the ground truth bounding box data of the central bore of each image. IoU and centre distance characteristics are depicted in the box plot charts in Figures 5.16 and 5.17. The results are close to each other for all tested model test set combinations. Regarding the IoU, the five models trained and tested on three or four panels do not vary much. Boxes, whiskers, and outliers are in the same range, with minimal IoU values at 0.93. The upper limit for each observation is a perfect bounding box prediction with 100% overlap. Models trained on three panels and tested on the fourth panel not included in the training data show a higher variation in IoU. The number of outliers and their significance is stronger pronounced in those results.

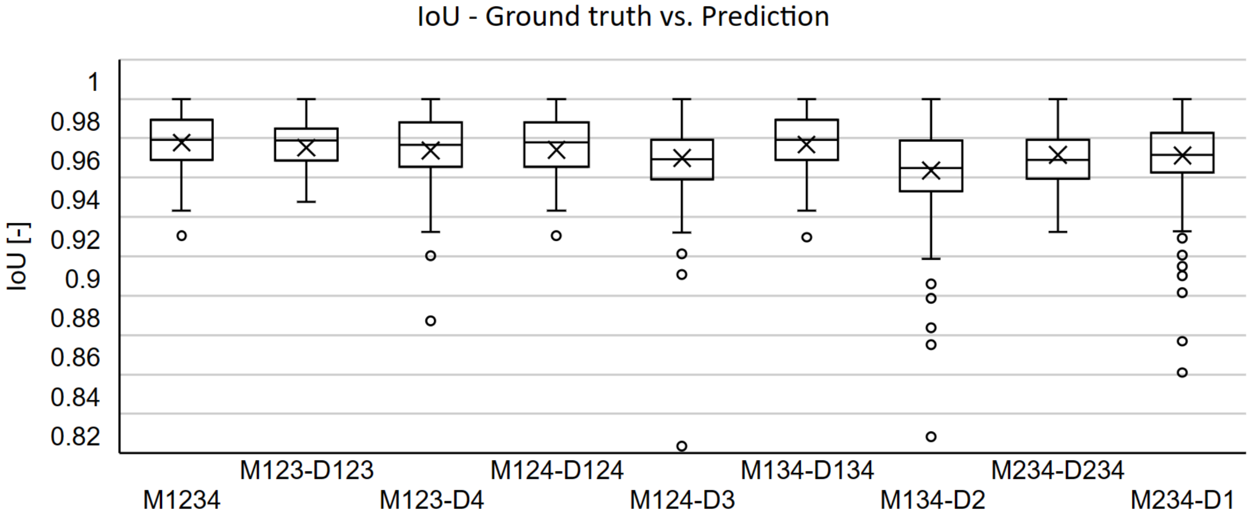


Figure 5.16: Comparison of IoU-calculations for the central bore predictions for all model and test set combinations.

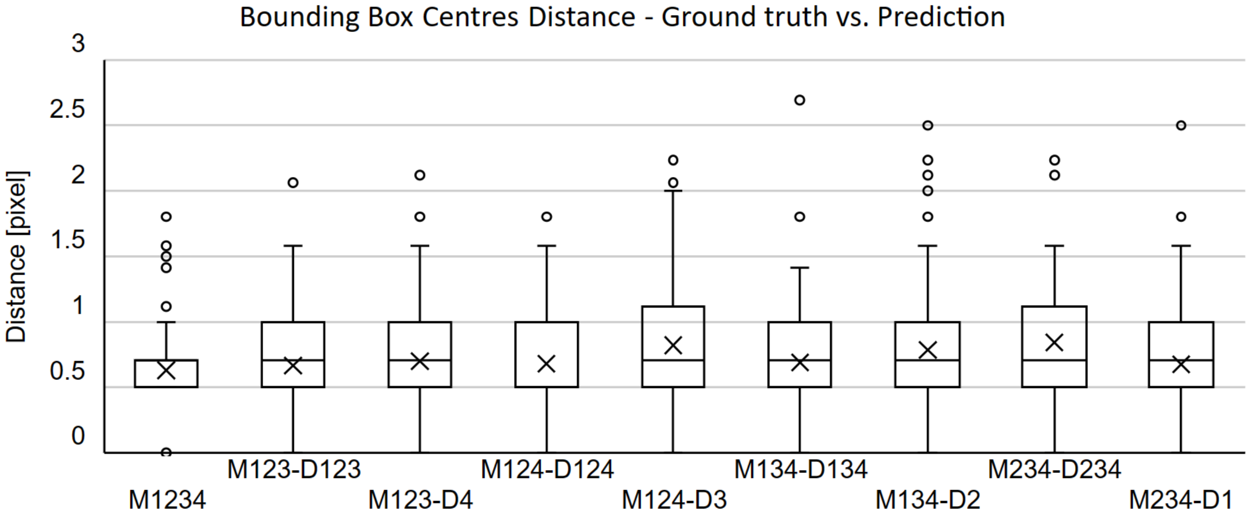


Figure 5.17: Comparison of distance calculations in pixels for central bore predictions for all model and test set combinations.

The average IoU across all predictions of all combinations tested is 0.971. An example of bounding boxes with this IoU is given in Figure 5.18 on the left. Since all images in one test set are different with different locations of the central bore, a grey-filled circle is drawn to represent an average bore. In the same Figure 5.18, the worst prediction in terms of IoU is given. The image to this prediction is from dataset D3 and is not sharp, but it shows a panel in motion with a blurred bore. The same applies to the two outliers of M134-D2 observations with the lowest IoU. Again, the translation and rotation process was not entirely finished, and the image capture started during the panel repositioning. In these images, the predicted bounding box is smaller than the annotation. The soft, blurry edges are not included in the prediction. These images are part of the 300 images per panel only and do not belong to test sets, i.e., the 10% selected for testing.

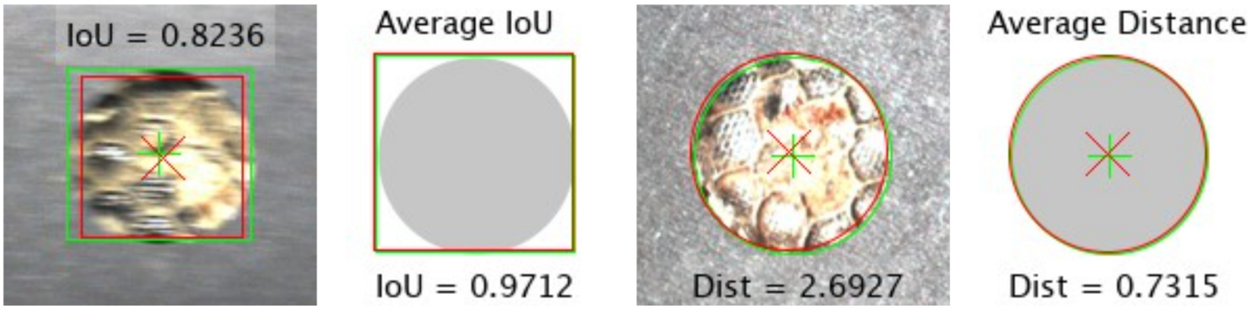


Figure 5.18: Worst and average prediction result for IoU (left) and centre distance (right). Predictions are indicated with the red line and the cross marker, and annotation with the green line and the plus marker.

The measurements of the distance between the predicted and ground truth bounding box centres are given in Figure 5.17. The overall variation without outliers is low. All observations range from 2 pixels to 0, i.e., no distance between centres. Model M1234 is the most precise, and the box of the plot is distributed in the interval [0.5, 0.7]. For most predictions, the x and y coordinates differ from the annotation by not more than 0.5 pixels. In the results of the other models, there is no significant difference. Remarkably, the behaviour of models trained on three panels is unchanged regarding the centre distance for known and unknown panels. Although the blurry images are still in the individual panel sets (D1-D4), this does not have the same effect as it has on the IoU. The most significant distance is calculated on a sharp image drawn from the panel 3 dataset D3 (Figure 5.18, right). The longest distance is approximately 2.7 pixels, which is in the setting of experiment B 0.41 mm. The average predicted bounding box centre is 0.73 pixels or 0.11 mm away from the annotated centre. In the example given for the mean distance, the centre markers and the drawn circles align to a substantial degree.

5.5.5 Experiment C: Performance in Different Lighting Conditions

For Experiment C, 980 images of the same scene are taken under different lighting conditions. An image with relatively uniform lighting of the central bore is selected to create the reference annotation. The reference image is done in Case 1, Subcase 1, with both ring lights turned off (s. Figure 5.19). The bounding box is square with an edge length of 164 pixels, proportional to a 14.7 mm bore diameter. The bounding box parameters (centre coordinates, width, height) are equal for all images from dataset C. The predictions for the central bounding box for all pictures of dataset C are generated with the M1234 model trained on the dataset B training data with all panels and are compared with the reference (s. Table 5.9).

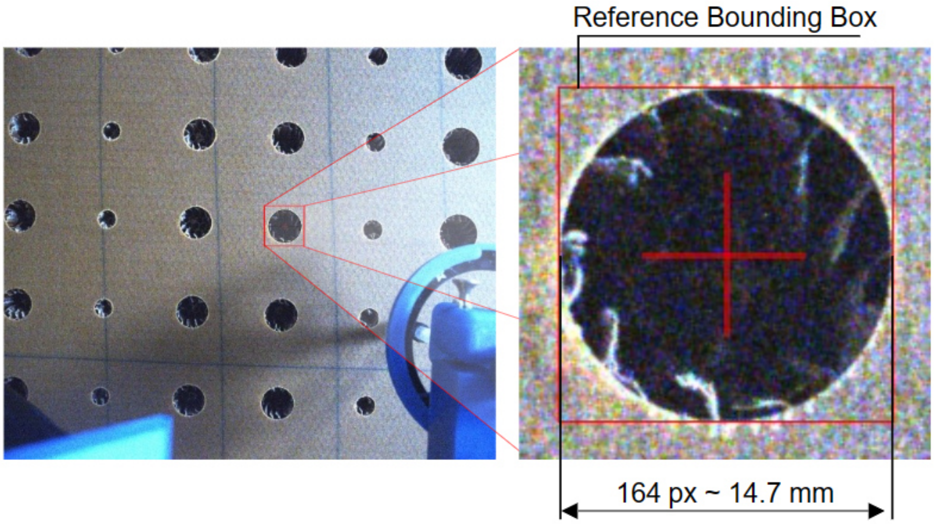


Figure 5.19: Reference image for experiment C and a close-up of the central bore. The reference bounding box and the centre are highlighted in red.

Four main lighting cases and six subcases have been defined to investigate the influence of different lighting on the prediction results. However, the results of the lighting cases do not differ significantly. Figure 5.20 compares all predicted bounding boxes with the reference bounding box. For all images of dataset C, exactly one instance is predicted for the central bore. Thus, R and P equal one since the target bore is always detected at the correct location exactly once. The difference in the x-coordinates between the prediction and the reference ranges from 0 to -1.5 pixels. The negative sign means a translation along the x-axis to the left. The median line is close to the upper box line, equal to -0.5 pixels. The average for the x-centre coordinate is at approximately -0.87 pixels. Overall, the x-coordinate prediction is exact, with low variation in every lighting scenario. The y-coordinate prediction is again very precise, with all values in a range of 1.5 pixels, but less accurate. The localisation error slightly increases due to a larger difference between the predicted and reference y-coordinate values. The average difference is at -3.18 pixels, and the median is at -3 pixels. Again, the results are similar for all lighting cases. The predicted bounding box centre is shifted to the top left of the image. The maximum Euclidean distance between reference and prediction is 4.27 pixels, which is, in the setup of experiment C, less than 0.4 mm.

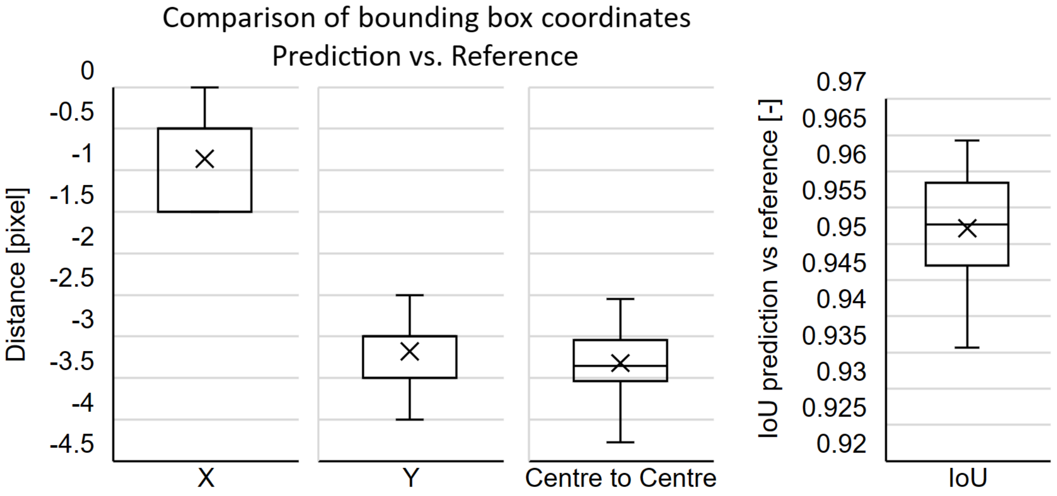


Figure 5.20: Difference between the predicted and reference bounding box coordinates, distance, and the IoU-scores.

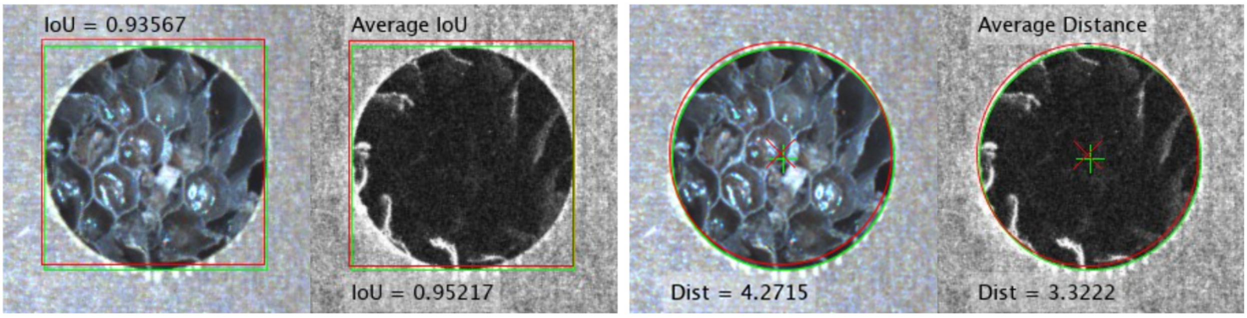


Figure 5.21: Worst and average prediction results for IoU (left) and centre distance (right). Predictions are indicated with the red line and the cross-marker and reference with the green line and the plus-marker.

The overlap of the predicted bounding box with the reference is measured using the IoU. The corresponding box plot shows a minimum IoU of greater than 0.935. Most predictions have an IoU above 0.95, commonly the highest threshold for accepting a positive prediction. Overall, the overlap of the bounding boxes is significant. Figure 5.21 shows the worst and average predictions regarding centre distance and IoU. The reference has a green outline and a plus-marker indicating the bore centre point. The prediction is highlighted in red with a cross-marker. A circle is plotted based on the height and width of the reference bounding box to emphasise the centre distance. Overall, the detection results are high quality, with only minor deviations from the reference. The example plots (box, circle, markers) in Figure 5.21 can be drawn on complete pixels only. The calculations and later transformation into Cartesian coordinates are on the sub-pixel level, increasing the localisation accuracy.

In Experiment C, the input image size is the full resolution of 5MP. 28 bores are visible on each image. As a result, the inference time increases to 81.8 ms.

5.6 Intermediate Summary

The applicability of learning-based CV models in HMLV assembly is studied in this chapter, focusing on location-depending parameters. Therefore, the models' performance, flexibility, and robustness are examined in several experiments. The assembly of lightweight panels conducted at the industry partner serves as the reference process. The developed procedure presented in Chapter 4 is applied to identify the process-relevant parameters. Two different location-dependent parameters are discovered: the location of bores on panels and the location of inserts in the tray. Among these, bore detection is the most challenging task and is thus selected for the experiments. The CV task is accurately predicting the bore's location on four panels with different properties.

The conducted Prestudy aims to identify CV models suitable for application in HMLV processes using the reference process as an example. The results show that learning-based models perform well on all panels, while conventional models require adaptation and are highly affected by noise. The best-performing model is selected for further analysis.

Three different experiments assess the general performance, flexibility, robustness, and spatial accuracy of the selected learning-based object detection model YOLOv5. Experiment A examines the general performance, and all trained YOLOv5 models indicate a high performance after training with mAP50:95 > 0.97 on the validation set.

Experiment B examines the performance on changing and new products (product flexibility) and the spatial accuracy. Despite not being trained on certain panel types, the models can predict similar product features on these unknown panels. The behaviour of models trained on three panels remains consistent regarding the centre distance for known and unknown panels, highlighting the model's product flexibility. On average, the predicted bore centres are less than one pixel or approximately 0.11 mm away from the annotated bore centre.

Experiment C evaluates the model's performance in changing environments (robustness) and the spatial accuracy. The model demonstrates high precision and exhibits no deterioration of the predictions despite significant changes in lighting. Overall, the mean offset of the bore centres is <0.3 mm, indicating the model's robustness against noise caused by lighting.

These experiments validate the effectiveness, flexibility, and robustness of the YOLOv5 model for detecting the location-dependent parameter *bore* in varying conditions and for unknown panel types. The next chapter examines the performance of learning-based CV models for time-dependent parameters.

6 Time-Dependent Parameters

Similar to Chapter 5, this chapter covers identifying process-relevant parameters and implementing different CV models to gather the parameters. However, another process-relevant parameter type is considered, namely the time-dependent parameters. First, the reference product and process are introduced and analysed at the beginning of this chapter in Section 6.1. Then, following the procedure from Chapter 4, functional requirements are defined, and the process-relevant parameters are determined in Section 6.2. Knowing the parameters, suitable CV models are selected, implemented, and tested in Sections 6.3 & 6.4.

6.1 Reference Process: Glueing of Lightweight Panels

Each inserted part is bonded to the panel by glueing, which is the subsequent step of the pick-and-place process. The glueing is an example of a joining process where different states must be identified during execution to control the process. The states change during execution, which makes them time-dependent. The status quo of the glueing process is manual and consists of the following steps. After placing the assembled panel with inserts in the workspace, the worker selects an insert and starts the glueing. The glueing process is visualised in Figure 6.1, and a) shows the cross-section of a placed insert inside the panel. To fill in the glue through one of the two ventilation holes, the worker places the nozzle of a glueing device on one of the two holes so that the nozzle seals the hole. Which hole the worker chooses is irrelevant. The hole is called the inlet hole (s. Figure 6.1 b).

Then, the worker starts pumping the glue through the nozzle into the panel. The adhesive fills the hollow space around the insert in the panel (s. Figure 6.1 c). While filling the glue, the worker continuously monitors the other ventilation hole (outlet). The worker stops the glueing process when the glue exits the outlet hole (s. Figure 6.1 d). The glue must exit the outlet hole to ensure enough glue is filled into the panel. If the amount of adhesive is too little, rework is necessary. However, it must not spill over the tab on the panel skin. Contaminated panel skin with glue causes rework or, in the worst case, the panel is not

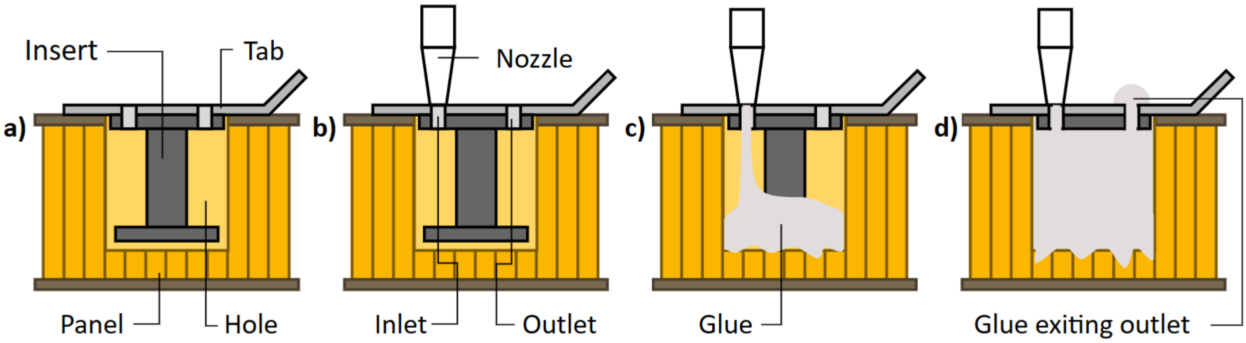


Figure 6.1: Reference glueing process. a) Cross section of the panel and insert, b) glue nozzle placed on inlet hole, c) filling glue into panel, d) glue exits the ventilation hole.

usable and put to scrap. Due to the panel core properties, the volume of glue changes for every hole. Additionally, temperature influences the viscosity of the glue. More liquid glue will settle faster into the cavities inside the panel core, changing the filled glue volume. The glueing is repeated for every placed insert in the assembled panel. After each insert is glued, the panel is moved to storage, where the glue cures.

6.2 Identification of Process-Relevant Parameters

This section focuses on the functional analysis of the reference process and the automation functions specification. Based on functional requirements, the process-relevant parameters are defined. Since the reference process is the subsequent assembly step of Chapter 5, only the product features essential for the glueing process are discussed.

6.2.1 Step 1: Product Analysis

The main product characteristics are given in Table 5.1. In contrast to Chapter 5, the inserts are now inserted into the holes of the panel. Thus, there is only one part, i.e., the assembled panel, where the tabs are visible. As for the holes, the tab location can be anywhere on the panel if they do not exceed the edge area. The inserts may be placed in different orientations depending on the pick-and-place process. Figure 6.2 shows an example of an assembled board with five inserts in different orientations. As a result, the ventilation hole positions change relative to the hole position.

Although there are far fewer tabs than inserts, the tab and its pattern change. Additionally, the ventilation hole appearance depends on the tab-insert combination. The variation of

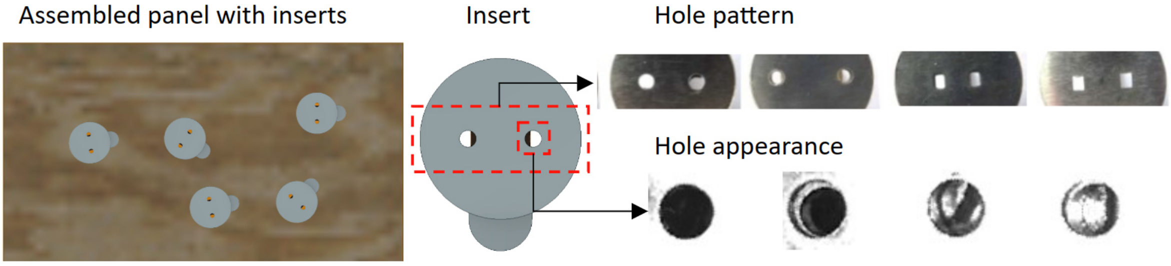


Figure 6.2: Assembled panel with inserts. The insert's hole pattern with different hole types and distances (upper right) and the hole appearance from different insert-tab combinations (lower right).

hole pattern and appearance is shown in Figure 6.2. The hole pattern consists of the hole type and the hole distance. The holes are either circular or square and have a 5-15 mm distance. The hole diameter or edge length is 2 mm. The zoomed ventilation holes highlight the changes in the hole's appearance. Depending on the insert type, the hole colour changes from entirely dark to bright and reflective with metallic structures.

6.2.2 Step 2: Process Analysis

The functional description and analysis of the glueing process are conducted in this subsection. Again, the description of the individual steps is solution-neutral, and the focus lies on the joining task, not the material handling. An overview of the single steps with their function group is given in the adapted functional flowchart in Figure 6.3. The first step is placing the panel in the workspace, which belongs to material handling. An insert is selected in the second step to start the actual glueing. Once the selected insert is located on the panel, the ventilation holes must be detected on the tab. The sequence of glueing the inserts or the ventilation hole for filling the glue does not matter regarding the glueing results. Once the inlet hole is defined and the location is determined, the glueing nozzle is placed on the hole. The nozzle and the tab seal the inlet hole so that the adhesive cannot leak from the inlet hole during glueing. The single tasks belong to detection and manipulation. The next step is filling the adhesive into the panel. Here, the worker pumps the glue until it exits the uncovered ventilation hole. This task is named glueing. While glueing, the worker must monitor the outlet hole and decide whether enough adhesive is filled, and the glueing process can end. These steps are repeated for every placed insert until all parts are bonded with glue.

Specification of automation functions

All functions of the function groups decision, detection, manipulation, and glueing shall be

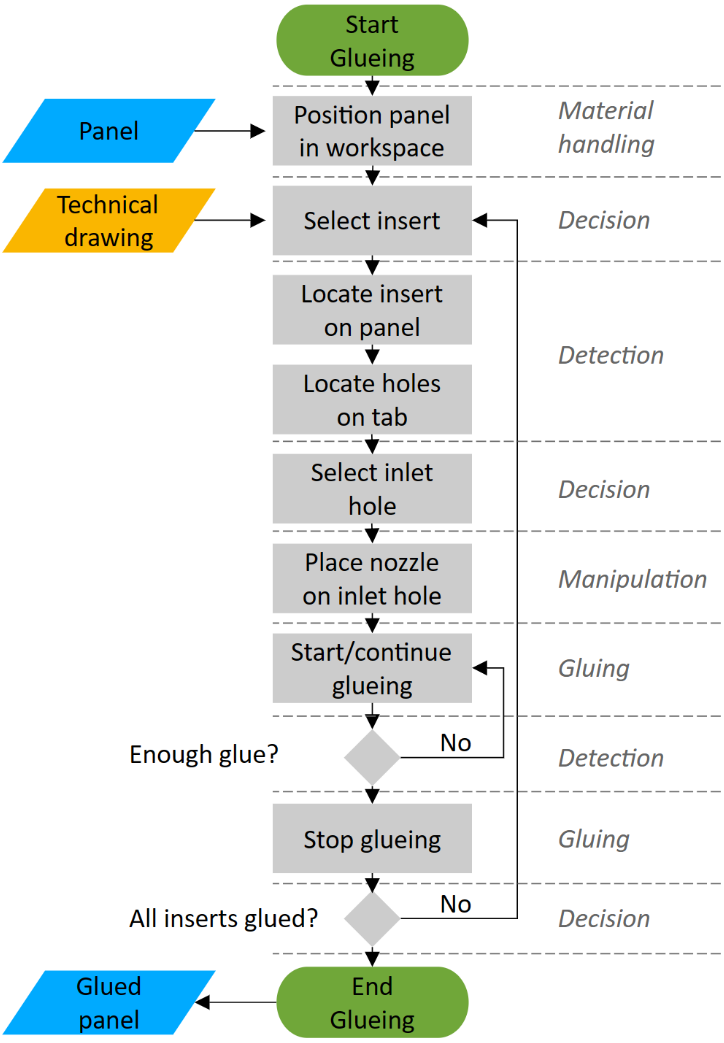


Figure 6.3: Functional flowchart of the manual glueing process with function categories. Blue parallelograms indicate physical and orange information inputs/outputs.

automated (material handling is excluded). The automation system provides information about the insert-hole combination and its position on the panel. The hole information is already available if the panel is not moved between the pick-and-place and glueing processes or if the translation and rotation between the processes are known. Otherwise, the selected insert must be detected. On the insert, the detection of the ventilation holes is necessary. One of the holes is defined as an inlet, and the other as an outlet hole. This decision must ensure that the outlet hole is monitorable and not obstructed by any part or device. The localisation of the holes must be precise enough to position the nozzle correctly, independent of the hole pattern. Furthermore, the manipulation device must be able to position the nozzle so that glue can enter leakage-free through the inlet and exit only from the outlet hole.

Monitoring the outlet hole enables sensor-based determining of the glue-filling level. A volumetrically or gravimetrically controlled filling process is not possible due to the panel's properties. Therefore, the filling level indication is essential for a reliable glueing process. Too little adhesive produces rework, and too much causes rework or scrap. Although scrap as an incorrect glueing consequence is worse than rework, both insufficient and overfull glue error states shall be avoided. After finishing one insert glueing, the steps are repeated until all inserts are glued. From the discussed process, the following subfunctions result:

1. Get panel data
2. Select insert-hole assembly for glueing
3. Determine the position and orientation of the panel in the workspace
4. Detect the insert position on the panel
5. Detect the ventilation holes
6. Select the inlet hole
7. Move glueing nozzle to the inlet hole
8. Monitor the outlet hole
9. Start the glueing process
10. Stop the glueing process when the target glue level is reached

Functional requirements
Table 6.1 summarises the functional requirements of the automation system resulting from the specified subfunctions above in the requirements list. Also, the subfunctions excluded from this research are added to the list. This ensures that the assumptions made for implementing the subfunctions do not collide with the implementation of the other modules. The labels *wish*, *demand*, or *excluded* distinguish those subfunctions. The subfunction category (s. Figure 6.3) and the required physical devices or information sources are indicated in additional columns.

The glueing as the principal joining task is in focus, and the material handling is excluded. It is assumed that the panel is either positioned to a reference or that the actual panel and hole locations are still available from the previous assembly step (s. Section 5.2). The same digital product information is required as for the pick-and-place process, including basic

Table 6.1: Functional requirements of the glueing process of inserts and lightweight panels.

No	Category	Subfunction	W/D/E[a]	Source, resource
1	Material handling	Position panel in workspace	E	Material handler, worker
2	Decision	Get panel data	D	Production order, product information
3	Decision	Select insert-hole assembly	D	Path planning based on product information
4	Detection	Determine pose of panel in workspace	W	From previous step, position referenced, CV system
5	Detection	Detect insert tab on panel	D	CV system
6	Detection	Detect ventilation holes	D	CV system
7	Decision	Select inlet hole	D	System setup
8	Manipulation	Move nozzle to inlet	D	Manipulator, e.g., robot
9	Manipulation	Place nozzle on inlet hole	D	Manipulator, e.g., robot
10	Detection	Find and monitor outlet hole	D	CV system
11	Glueing	Start glue filling	D	Glue dispensing system
12	Detection	Determine glue level	D	CV system
13	Glueing	Stop glueing when target glue level reached	D	Glue dispensing system, CV system

[a] W: Wish, D: Demand, E: Excluded.

panel dimensions and the hole positions relative to the panel. The glueing device must be within reach of all inlet holes. Additionally, it must be possible to cover the complete inlet hole area with the nozzle and seal the contact area between the nozzle and tab. Finally, each outlet hole must be in clear sight of the CV system so that outlet hole monitoring is possible.

6.2.3 Step 3: Process-Relevant Parameters

The basis for defining the process-relevant parameters is the conducted product and process analysis and specifying the automation functions. A digital system must replace the function groups decision and detection given in Figure 6.3, as a worker obtains this information manually. The glueing sequence and the upcoming insert to be glued can be defined with the product information and the hole pattern on the panel. The insert position is approximately known if the panel is at least placed to a reference. In the best case, the panel position is not altered, and the hole detections from the pick-and-place process are reusable. The detection of the inserts is similar to the detection in Table 5.2, row 8. However, the insert is not located in the tray but on the panel. Since the insert orientation might be unknown, the ventilation holes must be detected. The setup of the automation system defines the inlet and outlet holes. The inlet must be reachable with the glue nozzle and the outlet visible for the CV system. As long as both requirements are fulfilled, the inlet hole selection is valid. After the selection, the positions of the inlet and outlet holes are defined. As visualised in Figure 6.4 a), the motion of the glueing tool is in 2D with all assembly targets in one plane. The contact between the nozzle and tab is established with a linear joining motion along the vertical axis. For this task, manipulation systems exist. To enable successful glueing, the CV system must be able to give a response on the glue level. The glueing itself requires an electronically controllable glue dispensing device.

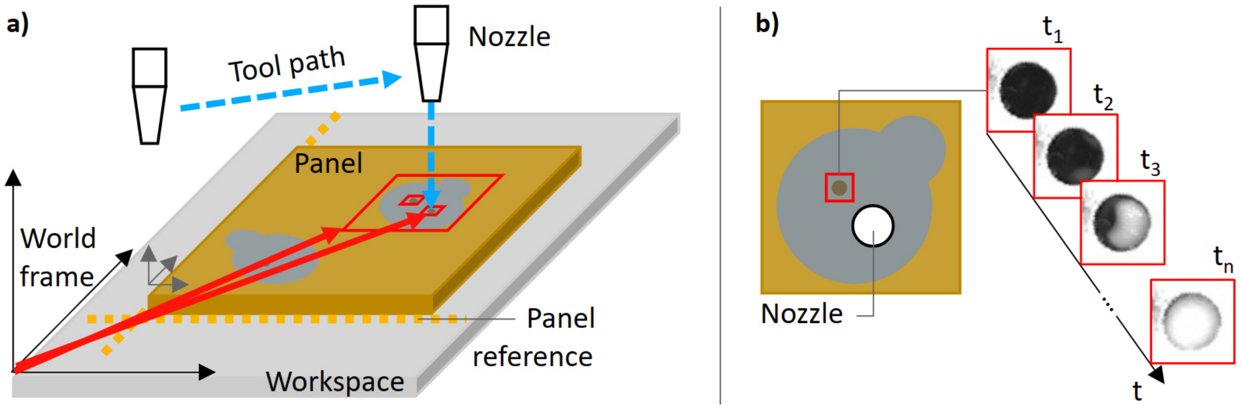


Figure 6.4: Schematic illustration of the glueing process. a) Motion of the glue nozzle; red arrows represent location vectors, dashed blue arrows indicate the tool path. b) Change in glue level over time at the outlet hole during glueing.

Overall, there are three subfunctions where the required information is not available from existing data: the detection of the insert, the detection of the ventilation holes, and the status of the glue level. The red squares and location vectors in Figure 6.4 a) indicate the missing location information. For the first two subfunctions, location-dependent parameters must be detected. Until the target location is known, there is no motion or change in the workspace. However, the detection of location-dependent parameters is the same as the hole detection discussed in Chapter 5.

In contrast, identifying the current state of the glueing process is different (s. Figure 6.4 b). Here, the state of the glue level changes while filling the adhesive into the panel. Correct and timely identification of the assembly state is crucial for successfully conducting the joining process. Additionally, the identification of the glueing state must be fast enough so that the glueing process can be controlled within specified limits. Therefore, the following sections analyse the time-dependent classification of assembly states using the glueing process as a reference.

6.2.4 Step 4: Computer Vision Task Definition

For this joining task, the location of tab and ventilation holes is necessary to place a manipulator with the glueing device to the inlet hole (s. Figure 6.4 a). The motion is in 2D, and it is sufficient to determine the pixel positions of the objects since they can be converted into real-world coordinates. It is a similar problem as defined in Section 5.2.4.

During the joining task, different situations must be assigned to specified groups to identify the assembly state that changes. As illustrated in Figure 6.4 b), the glue level in the outlet hole changes with the amount of adhesive filled into the panel. Therefore, the system must analyse the region of interest (ROI), i.e., the outlet hole, and respond with feedback indicating whether the glueing process must continue or stop. The feedback can be either discrete by classifying the input into discrete classes, e.g., empty, 50%, full, or by determining continuous values for the glue level using regression. For the glueing process, the minimal information required is if the glueing shall be continued or stopped and is based on the glue level in the outlet hole. Thus, binary classification of the outlet hole into the categories *empty* and *full* is sufficient for this task. However, other models are also applicable to identify the amount of glue in the outlet hole. Since binary classification is the simplest solution for the defined binary problem, it is selected for further analysis.

6.3 Experimental Methodology: Classification of Time-Dependent Parameters

Several experiments are conducted to create realistic glueing process data and to test several models on the defined binary classification problem. First, the experiments investigate the suitability of different models for the selected HMLV process. Then, the flexibility and robustness of the identified model are analysed by exposing it to new data in changing lighting conditions.[9]

[9] The results of this section are partly based on a master thesis of Jessica Plaßmann.

6.3.1 Experimental Setup

The experiments and datasets are conducted and generated on different setups depicted in Figure 6.5. The initially designed test rig uses aluminium profiles as the main structure. Initially, a smartphone camera is mounted parallel to the surface on the horizontal profile. The resolution of the camera is $6,000 \times 8,000$ pixels (48MP). Binning reduces the image size to 12 MP. Based on the dimensions of the glueing nozzle utilised in the manual process, a nozzle is constructed and created using 3D printing. Water-soluble and solvent-free wooden glue with similar colour and viscosity replaces the original industrial adhesive. This has the advantage of more effortless cleaning of the test samples and is significantly less cost-intensive. The glue is pumped manually into the panel.

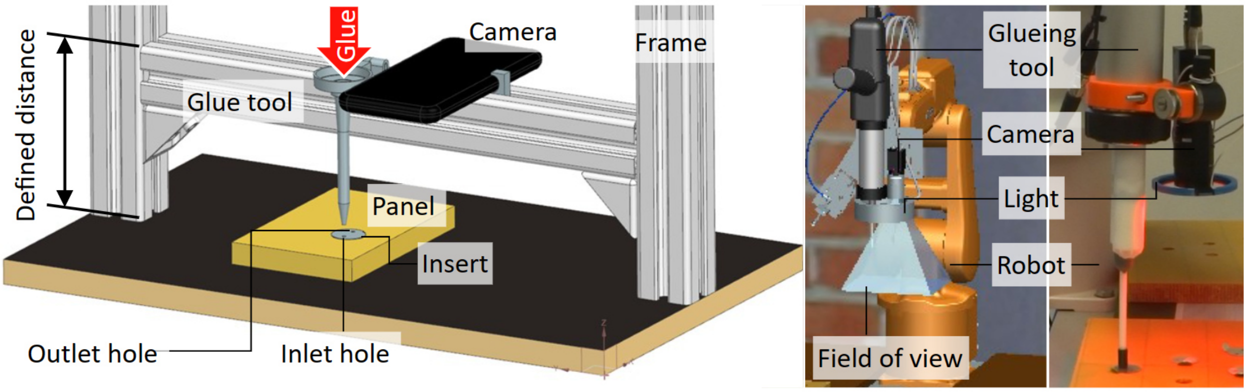


Figure 6.5: a) Initial test stand. b) Robot-based setup with industrial-grade glueing pump and camera. Image source: (Simeth et al., 2021).

To generate a more realistic setup, another test stand is designed. The new experimental setup uses an industrial grade 5 MP camera with a concentric ring light and an electric glue pump. All devices are mounted to the flange of an industrial robot. The nozzle and camera axes are parallel and perpendicular to the panel surface placed in the robot's workspace. An Arduino One can start, change the volume flow, or stop the electric glue pump. The µ-controller is connected via Ethernet to a computer, which enables the direct control of the pump. The industrial camera also sends image data via Ethernet to the main computer. In this setup, it is possible to use live video feeds of glueing processes for the analysis.

6.3.2 Glueing Dataset

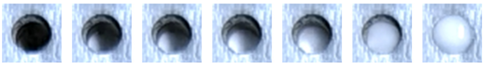
The basis for the analysis is the ground truth dataset generated on the initial test stand (s. Figure 6.5 a). The glue level is time-dependent and increases throughout the glueing process. The target is to simulate and observe several glueing processes to capture multiple time series of the glue level. Therefore, an assembled panel is placed on the ground plate of the test stand inside the camera's field of view. The glueing tool is arranged so that it covers the inlet hole. While manually pumping glue through the nozzle until it exits the outlet, the camera films the scenery and monitors the outlet hole. The videos are taken with a framerate of 30 FPS. In total, 18 videos were created. The algorithm shall be applied in an industrial process. Therefore, it is necessary for the dataset to resemble possible disturbance variables as realistically as possible. For image sensors, such disturbance variables are, for example, uneven exposure ratios, blurring, reflections or contamination of the components and lenses. Accordingly, these variables, especially the changing light exposure conditions, are included in creating the videos.

The single frames are extracted for each video, and a rectangular patch of $41 \times 41 \times 3$ pixels is cropped, covering the outlet. Each frame is assigned to the category *empty* or *full*. The two labels represent the states where either insufficient glue is filled into the panel, and the glueing must continue, or the level is sufficient, and the glueing must stop. The process owner at the industry partner defines the thresholds for *empty* and *full*. However, the border is rather vague and not well-defined. In Table 6.2, examples are given for each label. More examples are depicted in Appendix A3.1. There are significantly more data points for the class *empty*. No glue is visible at the outlet hole at the beginning of each glueing process simulation, and the level gradually increases. However, the transition from the state *empty* to *full* is short, and not many images can be generated before the glueing must stop to avoid spilling glue across the panel. In total, 380 images of the label *empty* and 225 of the label *full* are selected for the initial dataset.

Data Augmentation

To train learning-based CV models large amount of data is essential. Therefore, the dataset is increased following the importance sampling method. Copies of selected frames are manually processed and re-added to the dataset to create sufficiently large data from the small number of observations. Consequently, the model is deliberately exposed more frequently to data points that have a more significant influence on classification (Arouna, 2004). The glue dataset achieves this effect by dividing the existing labels into sub-labels containing images close to the initial label change (s. Figure 6, row sub-label). Only the images of the sub-labels *close to full* and *just full* are augmented. Seven artificial images are generated from each original image by applying rotation and reflection. After the selective augmentation,

Table 6.2: Overview of the glueing dataset based on (Simeth et al., 2021).

Data	Frames from 18 videos taken on test rig			
Label	Empty (380 images)		Full (225 images)	
				
Sub-label	*Divide into more and less influential data for importance sampling*			
	Clearly empty	Close to full	Just full	Overfull
		augment	augment	
Dataset	Empty: 1830 images		Full: 1170 images	

the total dataset contains 3, 000 data points with 1, 830 images of the class *empty* and 1, 170 images of the class *full*.

6.3.3 Step 5: Model Selection

In liquid-level detection, most of the methods introduced by other researchers apply conventional CV models to identify the surface, the silhouette or the colour of a liquid and compare it against predefined thresholds or references (Simeth et al., 2021). For this study, a similar low-dimensional model and further models with more elaborated decision rules are selected. The first model is also based on a single input dimension. The colour of the applied adhesive is white. Thus, the overall ROI brightness is expected to increase with the glue level in the outlet hole. The average brightness is calculated by converting the image patch from RGB to grayscale and taking the average of the intensity values of the pixels. At a certain brightness threshold, the process is stopped. The threshold is defined with the videos created for the dataset.

Secondly, a multidimensional detection rule is applied based on several textural image features. Texture feature descriptors are important and widely used in CV classification tasks. Tamura's texture features (Tamura et al., 1978) correlate highly with human visual perception and are proven to describe texture human-like most accurately and efficiently (Chi et al., 2019). Tamura features are selected since an automated glue-level classification system shall replace human vision. Tamura's texture descriptors are based on six textural features (s. Section 2.2.3). A support vector machine (SVM) image classifier is implemented to find a classification rule automatically.

The third proposed model to robustly classify the glue level is a hybrid model based on a pre-trained CNN and an SVM, presented by Simeth et al. (Simeth et al., 2021). The CNN is applied to extract image features, which are used for classification. Then, similar to the second approach, the features are forwarded to train an SVM image classifier. An SVM is selected because it can yield similar performance and accuracy for binary problems compared to classification directly by CNN while reducing complexity and computational effort (Kumar T.K. et al., 2019).

The last model uses only the pre-trained CNN ResNet50 for the complete classification. Therefore, the fully connected layer and the classification layer of the selected CNN are replaced to fit the two classes *empty* and *full*. The complete CNN is then re-trained on the glue dataset.

6.3.4 Design of Experiments

Two experiments are conducted to identify a suitable model for the time-dependent classification task and to test its flexibility and robustness. While the first one solely uses the images from the glueing dataset, the second experiment is performed on live streams of simulated glueing processes. The glueing dataset and the baseline videos are used to determine the general performance of the introduced CV models in the first experiment. The brightness model threshold is determined with the 18 baseline videos used for glueing dataset generation. For each video, the frames are identified where the state changes from *empty* to *full*. The outlet hole patch brightness is calculated, and threshold ranges for every video are derived from which a global threshold is defined.

The other models use learning-based classifiers and are trained to find the optimal classification rule. The first step is balancing the two classes by randomly drawing images from the *empty* class to match the two label counts. Then, the glue dataset is randomly split into training, validation, and testing in the ratio of 70%:10%:20%. Each set contains the same number of images of both labels. The SVM classifiers are trained on the training and validation sets, and the performance is analysed with the remaining test set. The CNN is trained on the training set. After each epoch, the model is exposed to non-training data, i.e., the validation set, to calculate less-biased performance estimates and to determine if overfitting occurs. The final performance is calculated with the remaining test set. The same training, validation, and test sets are used for all three models. The best model is selected for further analysis.

The learned correlations of the models depend to a large extent on the training data, which are selected by random drawing from the glueing dataset. The absence or presence of data or noise, e.g. exceptional cases, disturbance by reflection, uneven lighting and others, can distort the results and deteriorate the analysis. Therefore cross-validation via a Monte-Carlo-Simulation (MCS) is conducted. The MCS measures the influence of the random distribution on the model's quality by re-training the model repeatedly with different splits of the dataset (Andrieu et al., 2003). The classification accuracy of the trained models is recorded and then plotted in a histogram. The lower the variance of this distribution, the lower the influence of the random dataset distribution on the model. Furthermore, the wrongly classified frames are analysed.

In the second experiment, the best model is selected to classify images from a live stream of a glueing process. The video streams contain new data generated on a different test stand. Here, the lighting situation is changed deliberately to create bright and dark lighting conditions. The ring light illuminates the scenery in red, and the red image plane is taken for further processing. The cropped ROI has a different resolution than the glue dataset and is grayscale but covers a similar physical area of the outlet hole. The resulting grayscale image is transformed into the required model input size and classified. If an image is classified as *full*, the glueing process is stopped. Several glueing cycles are conducted in bright and dark environmental situations.

All models are implemented with MATLAB R2020a Update 6 (6.5.0.1538580) 64-bit version installed in Windows 10 Pro operating system. The available hardware is a Microsoft Surface Book 3 with an Intel(R) Core(TM) i7-1065G7 processor with four cores. The installed GPU is an NVIDIA GeForce GTX 1650 with Max-Q Design and 4GB memory. The Surface Book has 32GB RAM.

6.3.5 Performance Metrics

The classification results of each model are evaluated using the confusion matrix. Based on the values for TP, TN, FP, and FN, the accuracy, precision, recall, and F1-score are calculated (cf. Section 2.3.4). If an image labelled as *full* is predicted as *full*, it is counted as TP, and if predicted as *empty*, as FN. For originally *empty* images, the prediction of the class *empty* is considered TN, and the prediction as *full* as FP. Furthermore, the training and testing time is recorded. The training time is the period from the start to the end of the training process of the CV model, including image loading, feature extraction, and model training. The testing time includes image loading, feature extraction, and classification of the entire test set. Since the models classify a live stream frame by frame, the period for the loading, preprocessing, and classifying of a single image is given as the inference time. It is calculated as the average time required for loading, preprocessing, and classifying an individual image based on 100 subsequent prediction cycles.

6.4 Results

This section presents and compares the results of all proposed CV models. The best-performing model is utilised in the cross-validation. The same model predicts labels for unknown images extracted from live glueing video streams.

6.4.1 Image Patch Brightness

The average brightness of the ROI is the input parameter for the classical approach. A fixed or adaptive threshold shall indicate that the glue is filled sufficiently into the panel. When monitoring the outlet hole, it is assumed that the average brightness increases with a strong gradient as soon as the glue is visible. The chronological sequence of filling the exit hole with an adhesive until it exits is shown in Table 6.2, row label. However, variations in the inserts, the filling process, and the lighting strongly affect the average brightness, as shown in Figure 6.6. The first sequence represents a regular sequence of the images obtained from an average glueing video. Once the glue is visible in the ROI, the brightness increases continuously. At some point, the escaping adhesive forms a sphere over the exit. The shadow of this sphere reduces brightness until the shadow exits the ROI. This is indicated by the average brightness dip at frame position 300 in Figure 6.6 a).

In some cases, bubbles appear on the adhesive's surface, significantly impacting brightness. In plot b) of Figure 6.6, the brightness decrease is visible. However, the glueing process shall be continued, as the target level has not been reached. Plot c) shows a glueing process in a brighter environment (sunny day) with a more reflective tab. The average brightness is higher, although a constant light source is applied. Here, only encapsulation can avoid such

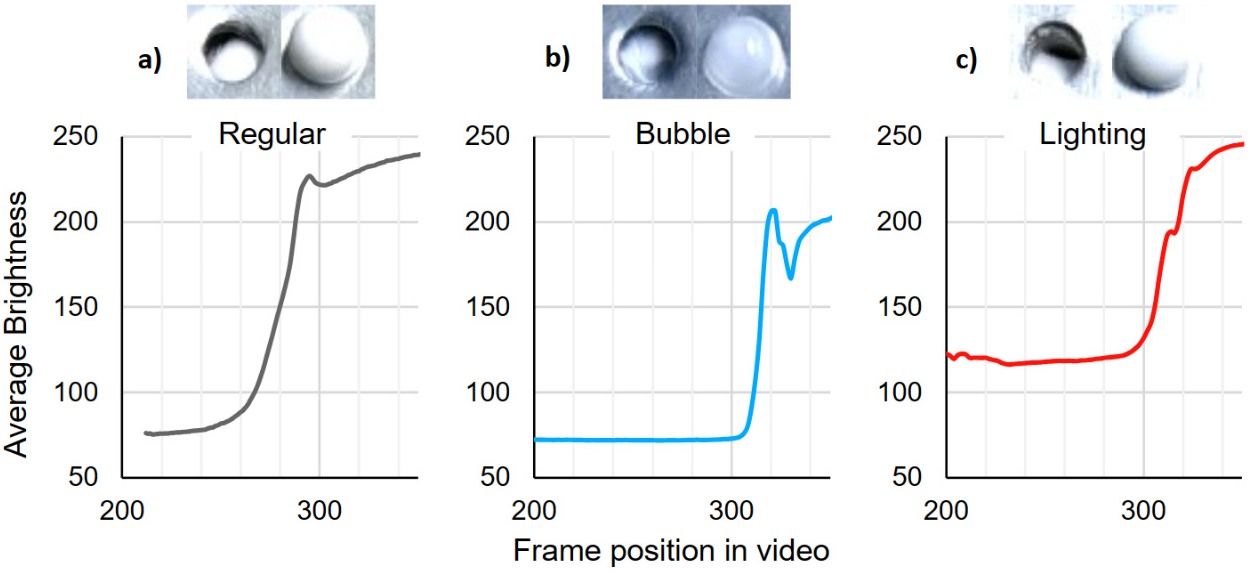


Figure 6.6: Average brightness of a) regular glueing behaviour, b) formation of bubbles at the outlet, c) highly reflective tab. Figure taken from (Simeth and Plapper, 2023).

light disturbances. Additionally, the surface of the outlet hole material has high reflection differences, so it appears in different shades, from dark grey to almost white. Overall, no thresholds applicable to all trials could be identified. The existence of glue can be detected, but it is impossible to distinguish different glue levels and thus effectively control the process. Thus, the model is discarded.

6.4.2 Tamura Features and Support Vector Machine

In the second approach, Tamura textural features are extracted and forwarded to an SVM image classifier. The implementation is shown in Figure 6.7. After loading an input image, it is transformed into a grayscale. Then, the Tamura features of the grayscale image are calculated in sequence. The output, a vector with six values, is forwarded to the SVM image classifier, which predicts the class label. The Tamura feature functions implementations are based on (Lee, 2018). The SVM is implemented with a linear kernel. Other parameters are determined automatically by the model based on provided training features (The MathWorks, 2023a).

The combined model is trained on the train and validation set (80% of the dataset) and tested on the remaining 20%. The prediction results on the test set are given in Table 6.3. The accuracy and the F1-score are both at a high level of 0.936. The recall is slightly larger

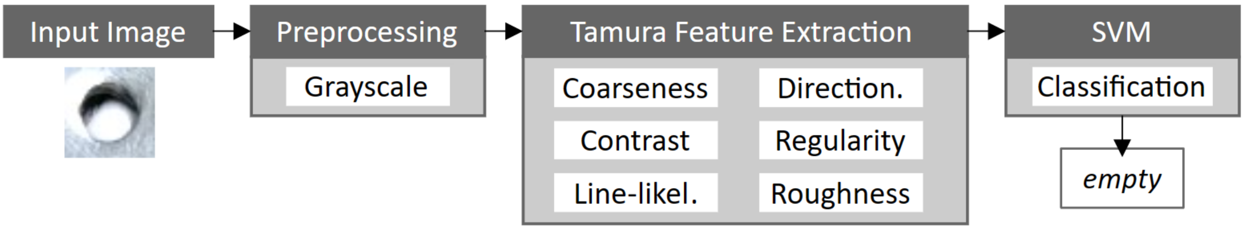


Figure 6.7: Implementation of the combined Tamura-SVM approach.

than the precision. Thus, the model tends to stop the glueing process too early since more *empty* images are classified as *full* (FP) than *full* as *empty* (FN). Overall, the model performs well on the classification task.

Table 6.3: Overview of the model performances.

Model	Accuracy	Precision	Recall	F1-score	Training [mm:ss]	Testing [s]	Inference [ms]
Tamura + SVM	0.936	0.929	0.944	0.936	1:18	13.5	26
ResNet50 + SVM	0.987	1	0.974	0.987	0:27	5.7	167
ResNet50	0.991	1	0.983	0.991	18:34	3.27	16

6.4.3 ResNet50 Features and Support Vector Machine

The next model combines CNN image feature extraction with an SVM image classifier. CNNs perform well in many image classification challenges. This hybrid approach tries to leverage the high potential of CNN feature extraction with a reduced training time of ML classifiers. Therefore, the pre-trained CNN ResNet50 is implemented. Feeding the glue images into the model requires the adaptation of the ROI to the model input layer. Thus, the images are resized at the beginning. Then, the ResNet50 extracts the image features. The images are read in batches of 32. The activations of the fully connected layer at the end of the CNN are taken. For each image, a feature vector of 1000x1 size is extracted and forwarded to the SVM for classification. The vector size relates to the 1000 object classes of the pre-trained CNN. The SVM is trained on the feature vectors in this approach, and the CNN itself is unchanged. Thus, the computationally expensive CNN training is omitted. The hybrid model is also trained on 80% of the data (training and validation set) and tested on the remaining test set. The implementation is visualised in Figure 6.8. The CNN ResNet50 is loaded with pre-trained weights into the model. As with the Tamura features, the SVM uses a linear kernel and other options are determined automatically.

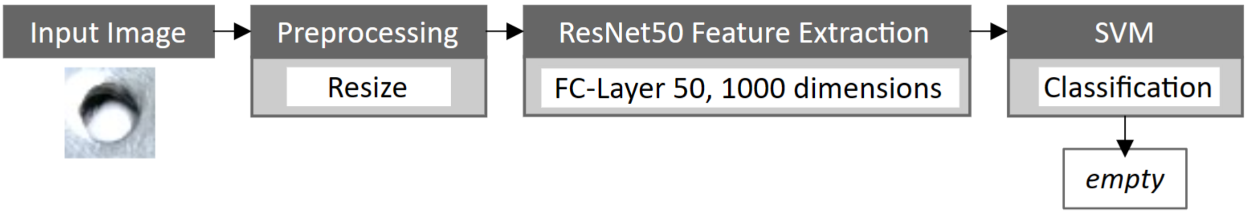


Figure 6.8: Implementation of the combined ResNet50-SVM approach.

The classification results of the hybrid CNN-SVM model are given in Table 6.3. The prediction accuracy and F1-score are close to 99%, and almost all frames are classified correctly. Furthermore, all predictions of the label *full* are correct, indicated by the precision of 1. However, some *full* frames are not identified and are classified as *empty*. Thus, this classifier stops too late since some *full* frames are not identified. However, with a recall of 0.974, there is less than one misclassification per input video. Thus, the delay is, on average, below one frame.

6.4.4 ResNet50 Convolutional Neural Network

The last model selected for the classification of time-dependent parameters is a CNN. Here, the CNN extracts the image features and conducts the classification. No additional classifier is necessary. Transfer learning is applied, and the pre-trained ResNet50 is selected. Transfer learning significantly reduces the required data to train the network compared to CNNs trained from scratch. The CNN, however, must be adapted to fit the classification problem, precisely the fully connected layer and the classification layer. The 50th layer of ResNet50 consists of the fully connected and the softmax classification layer. Both are exchanged to fit the two classes *full* and *empty*. An overview of the implementation is given in Figure 6.9.

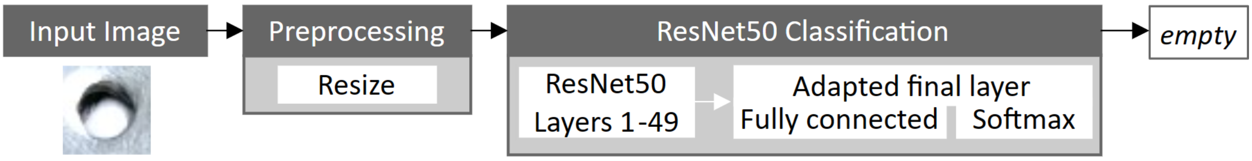


Figure 6.9: Implementation of the CNN ResNet 50 transfer learning approach.

The adapted CNN is trained for a maximum of 20 epochs on the training set. The training set is shuffled after each epoch. The training uses early stopping and ends if the validation loss does not improve over five consecutive epochs. Validation is conducted after each epoch. The training options are set to SGD optimiser with an initial learning rate of 0.01. The learning rate is adjusted after ten epochs to 0.001. Training takes a long time if the learning rate is too low. On the other hand, the model may not converge or reach a suboptimal result if it is too large. A scheduled learning rate decay after ten epochs is implemented to overcome this issue.

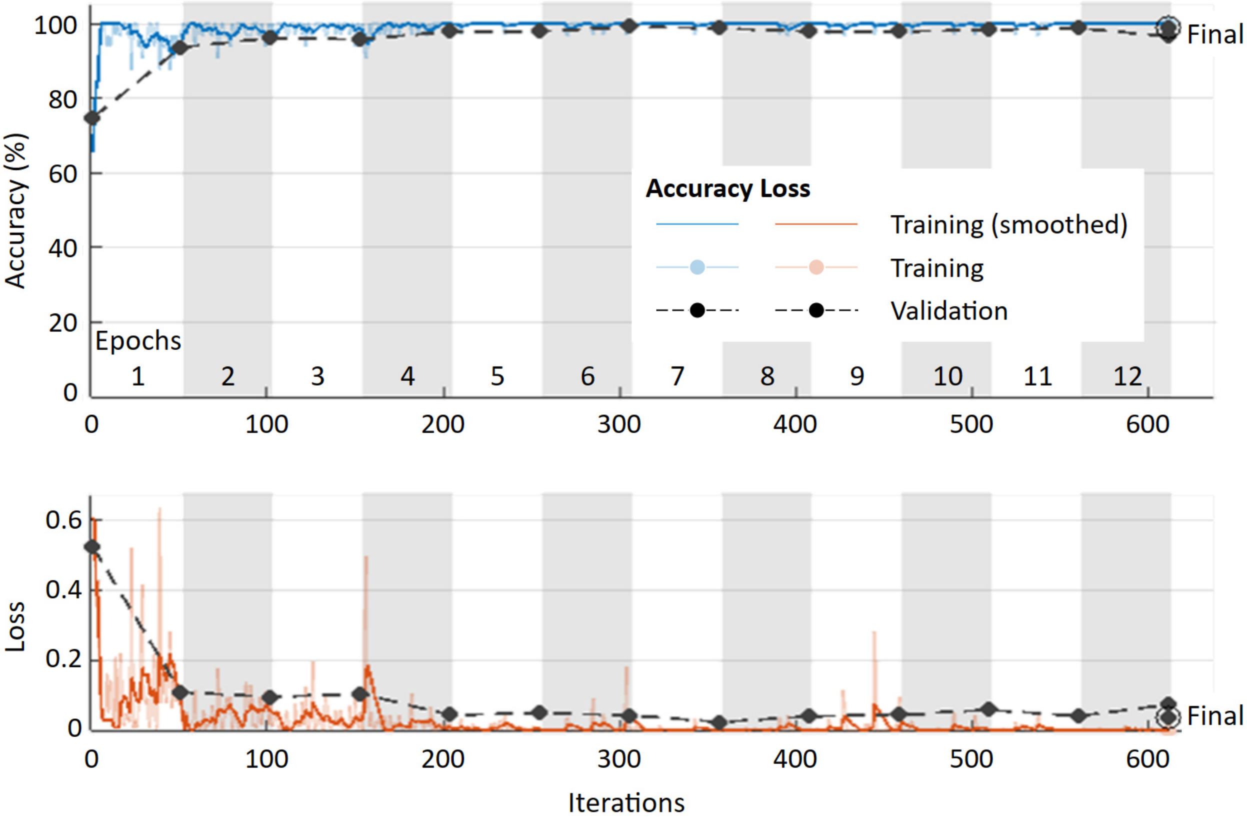


Figure 6.10: Training results of the CNN with validation.

The training results are given in Figure 6.10. On the horizontal axis, the iterations, i.e., the processing of a single batch (32 images), are plotted. The model converges to a high accuracy of >90% after the first epoch, and from epoch four, the variation in accuracy and loss is very small. Similar behaviour is seen in validation results on non-training data. After epoch seven, the change in both training and validation loss is marginal. Outliers in the loss exist only in a few iterations. The training is stopped after 12 epochs since no improvement in validation loss is achieved from the previous five epochs. The final validation accuracy is 98.7%. The training and validation plots relate well, and no overfitting is expected.

In Table 6.3 are the classification results on the test set. The model works almost perfectly and has four misclassifications in total, always FN. However, all *full* predictions are correct.

The accuracy and F1-score are above 99%. The training of the CNN for 12 epochs took 18:34 min. The testing and inference are rapid. It takes 3.3s to classify the whole test set. The inference time is 16 ms and, therefore, suitable for time-dependent parameter classifications regarding response time.

6.4.5 Comparison

The presented approaches considerably differ in their ability to detect the fluid level in bores robustly. Defining a threshold that works for the whole dataset was impossible with the first conventional, rule-based approach. The simple classification based on the average ROI brightness always indicated a substantial increase in brightness when glue started to fill the panel and the outlet hole. However, the average brightness significantly differed between the test set images at comparable glueing levels. Thus, it is impossible to determine whether a hole is half-full, full, or overfull (cf. Table 6.2) from the plot. In addition, the appearance of bubbles on the glue, the change in reflectivity, and the overall lighting situation (daylight, evening, night) further impact the outcome, which can not be compensated with the simple rule (cf. Figure 6.6).

The second and third approaches apply the same machine learning SVM image classifier but use different image features. Tamura features and the features extracted by the pre-trained CNN ResNet50 are the basis for training and classification. The Tamura features are six stochastic textural image characteristics. From ResNet50, a fully connected layer with 1000 features is extracted. The last approach applies transfer learning with a problem-adapted CNN ResNet50.

Regarding classification performance, all three models can predict well on the glueing test set. Although the Tamura-SVM model uses only six image descriptors, the accuracy is high. However, the hybrid CNN-SVM and the CNN models outperform the Tamura-SVM model. Both approaches have perfect precision and achieve an almost perfect accuracy of 99% on the test set. The difference in prediction quality between the two models is not significant.

Comparing the required time for training, CNN requires, by far, the most time. The training took 18:34 min. The SVM training is at 1:18 min for Tamura features and 0:27 min for the CNN features. This is 14 and 40 times faster than re-training the CNN. Surprisingly, the SVM training with a less dimensional feature vector took longer. This results from the implementation of feature extraction. CNNs have the advantage of GPU and parallelising calculations, e.g. loading multiple images simultaneously, whereas each Tamura feature is calculated sequentially for each image. On the contrary, the inference time for a single image is the fastest using CNN classification and the slowest for the hybrid CNN-SVM model. The CNN classifies, on average, in 16 ms, the Tamura-SVM model in 26 ms, and the CNN-SVM model in 167 ms.

Overall, the re-trained ResNet50 has the highest performance and predicts superfast. However, training is time-intensive, especially if multiple training cycles are conducted. The CNN-SVM has nearly the same prediction quality, and training results are available fast, but it takes much longer to classify a single image. This is critical if the model shall be applied in a time-dependent parameter classification task. However, the testing time per image is significantly less than the inference, indicating improvement potential with better model implementation.

6.4.6 Cross-Validation with Monte-Carlo Simulation

The influence of the random data drawings for balancing the two classes and splitting the data into training, validation, and test sets is analysed with an MCS. Therefore, the classification model is trained 500 times with different random dataset compositions. Since the hybrid CNN-SVM model is similar in prediction accuracy to the CNN model but more than 40 times faster in training, the hybrid model is chosen for cross-validation.

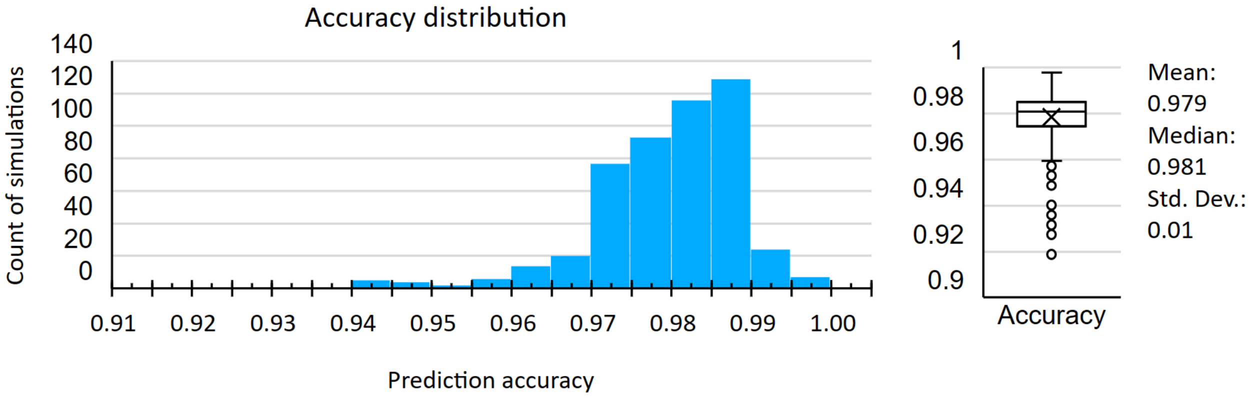
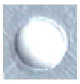
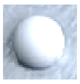
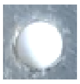
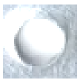
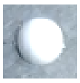
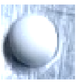
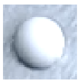
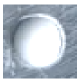
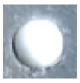


Figure 6.11: Histogram and box plot of the MCS with 500 simulations.

For each cross-validation cycle, the random drawing of each subset is repeated. The model is then trained on the training and validation set and tested on the test set, whereby the labels of those incorrectly classified frames are recorded in addition to the respective confusion matrices. Figure 6.11 presents the distribution of the prediction accuracy for all 500 simulations in a histogram. The expected value of the prediction accuracy is around 0.98, meaning that, on average, 98% of the frames were correctly classified across all found decision rules. The box plot shows that most observations are closely distributed around the median. The standard deviation is 0.01, and the predictions have a low variance.

The accuracy distributions of both classes are compared to detect a potential distortion of the results. On average, images of the class empty are predicted correctly to 98% and of the class *full* 97.7%. The difference is only 0.3%. Still, *full* images are classified not correctly more frequently; consequently, the process is rather ended too late (*full* classified incorrectly) than too early (*empty* classified incorrectly).

Table 6.4: Top 10 falsely classified frames with comments in the last column.

	Frame	Video	Label	Count	
1		18	*full*	582	First full
2		9	*empty*	367	Wrong label
3		14	*empty*	319	Last empty
4		1	*empty*	301	Last empty
5		15	*full*	269	First full
6		7	*full*	237	First full
7		9	*empty*	237	First full
8		2	*full*	216	First full
9		11	*empty*	205	First full
10		14	*full*	195	First full

Furthermore, the recordings of misclassified frames are evaluated. Those frames that led to misclassifications in a particularly high number of cases are of interest. Due to manual dataset processing and importance sampling in the initial dataset, each frame is present eight times in the dataset (original and augmented). In the following, original and augmented frames are combined and counted together. During the 500 MCS cycles with different dataset distributions, the hybrid model made 234,000 predictions on the 468 test set frames, where each frame, including its augmentations, could be misclassified 4,000 times.

Table 6.4 lists the ten most frequently misclassified frames. The evaluation reveals that all misclassifications always occurred at the boundary between the two categories. The 18 videos are divided into two classes, and the label changes from *empty* to *full* at a particular frame. The last frame labelled as *empty* or the first frame labelled as *full* is misclassified. The second most frequent misclassification reveals a dataset error, where a frame was wrongly assigned to the *empty* category but is *full*. The fact that the classifier identified this frame as *full* indicates its quality. The frame on the rank seven roots from the same video. Again, the glue forms a well-pronounced sphere above the outlet hole so that the frame appears to have more glue than the other falsely classified *full* frames. The boundary between *empty* and *full* is not well defined, and the transition is smooth. All misclassifications lie in the transition phase. The labelling quality of the frames in the transition phase must be increased, and the actual threshold level be aligned to improve the prediction quality. However, all top ten misclassifications are not critical for the reference process.

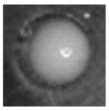
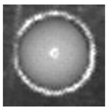
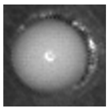
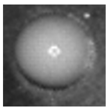
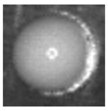
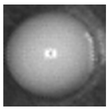
6.4.7 Testing on New Data

In the second experiment, the developed classification model is implemented on a new test setup (s. Figure 6.5, b). A larger view of the setup and the detection program is shown in Appendix A3.2. The target is testing the model on live glueing video streams with controllable glueing equipment in different lighting conditions. Therefore, an assembled panel is placed in the workspace of the industrial robot, and the glueing nozzle is positioned manually on an inlet hole. The other ventilation hole must still be in direct sight of the camera for monitoring. In the next step, the ROI of the outlet hole is defined. The camera provides a live stream with 24 FPS of a 5 MP image. A single frame is grabbed from the video stream, and the ROI is cropped and forwarded to the hybrid CNN-SVM model used in the cross-validation. The model classifies the ROI and returns a class label. However, not all frames are classified since the image processing and inference are slower than the camera framerate. Thus, the transition phase from *empty* to *full* may fall into the unprocessed period.

In total, 35 glueing processes are conducted in a bright and dark environment. Table 6.5 shows the last frames classified as *empty* and the first as *full*, exemplary for six videos. The difference in the images compared to Tables 6.2 and 6.4, and between the lighting environments, is significant. Still, the implemented model can differentiate the two glue-level classes correctly. Further examples are provided in Appendix A3.3.

All classified frames during the live glueing experiments are saved. The resulting time series of outlet hole images of each video are checked manually, and the frame position, where the

Table 6.5: Frames at the label change from the live glueing classifications.

	Bright environment			Dark environment		
Last empty						
First full						

label changes from *empty* to *full*, is defined. For all 35 glueing processes, the manual and automatic classification by the model is identical.

6.5 Intermediate Summary

This chapter covers the experimental analysis of the performance of learning-based CV models to determine time-dependent parameters. The glueing of inserts to panels is another joining process of the industry partner and the reference process for this chapter. With the procedure presented in Chapter 4, the glue level is identified as a time-dependent process-relevant parameter. The glue level is monitored at the outlet hole of the glued insert. The defined CV task is the accurate classification of the glue level into the binary classes *full* and *empty*, indicating whether the glueing process shall be continued or stopped.

In the first experiment, conventional and learning-based CV models are developed and compared regarding their classification accuracy. A hybrid CNN-SVM model and a CNN model show the best results of >99% correctly classified frames of glueing sequences and are superior to the conventional models.

A cross-validation of the first experiment's best-performing model analyses the training data's impact on the model performance. The learned correlations of the models depend to a large extent on the training data, which are selected by random drawing from the glueing dataset. Therefore, a Monte Carlo Simulation with 500 random dataset configurations is conducted. The prediction accuracy is, on average, 98% distributed over the interval $[0.91, 1)$. An analysis of the misclassified frames unveiled that most wrong predictions occur on frames directly at or close to the label change. I.e., the last video sequence images before the recorded glue level turn from *empty* to *full* or vice versa.

In second experiment, the developed classification model shall control real glueing processes based on live video streams. Therefore, a correct and timely response of the model is essential to avoid contamination of the panel with spilt glue. In all conducted 35 glueing trials in different bright and dark lighting environments, the learning-based CV model correctly classified the end of the process and was fast enough to stop the glueing process in time.

These experiments again demonstrate the learning-based CV model's strong performance for accurately and quickly classifying time-dependent parameters in changing environments. In the following chapter, the developed models are implemented on a technology demonstrator to showcase the whole automation of the reference processes. Furthermore, the procedure is validated in additional industrial case studies.

7 Validation

This chapter demonstrates the practical application and validation of the developed procedure and models. Section 7.1 details the development of a technology demonstrator specifically designed for the reference processes outlined in Sections 5.1 and 6.1. The examined models are implemented and applied to automate the reference processes using industrial hardware. Section 7.2 shows the application of the procedure in another industrial use case focusing on the processing of lithium-ion batteries, and 7.3 in a welding application of bent sheet metal parts. These real-world case studies show the effectiveness and reliability of the developed procedure and models across diverse industrial scenarios.

7.1 Case Study 1: Lightweight Panel Assembly Technology Demonstrator

Besides validating the developed procedure, the technology demonstrator showcases the technical feasibility of automating the reference processes of the industrial partner. The procedure is applied to the reference processes discussed in the preceding chapters to allow the experimental analysis of the CV models. Consequently, the outcome of the procedure is only briefly elucidated. This section focuses on the demonstrator setup and the testing results.

7.1.1 Demonstrator Setup

The technology demonstrator is designed to encompass both reference processes: the insertion process (s. Section 5.1) and the glueing process (s. Section 6.1). The overall configuration of the demonstrator is presented in Figure 7.1. The left image showcases the industrial robot (ABB IRB 120) and workspace, with the panels to be assembled located in the front right area. The orange component in the front left represents the insert tray, which holds the inserts required for the insertion process. A chessboard pattern is also visible in the back right area, serving as a calibration reference. It is used for camera and camera flange calibration.

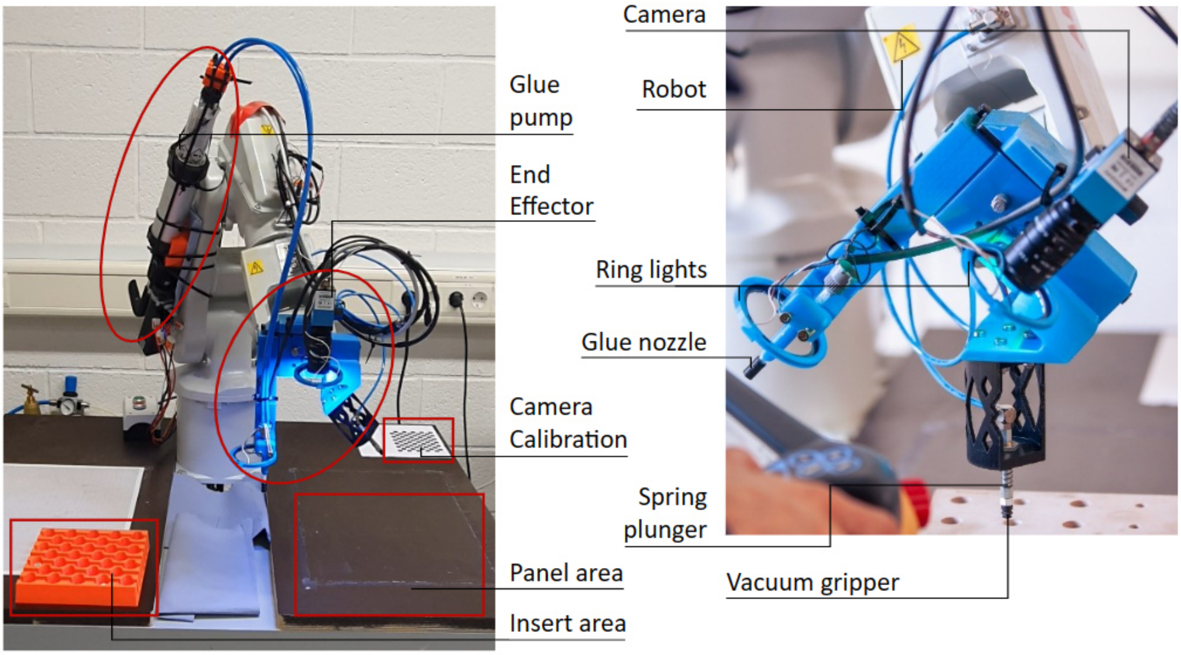


Figure 7.1: Left: Demonstrator setup and workspace. Right: Close up on the MFEE with a camera, vacuum gripper, glue pump, and lighting.

The robot features a Multi-Functional End Effector (MFEE) and a glue pump. The MFEE includes a suction gripper with a spring plunger, a nozzle for dispensing adhesive, an industrial camera (TheImagingSource, 5MP) with a fixed 8 mm lens, and two 24-LED ring lights. The nozzle is connected to the pump via hoses, and the camera and nozzle axes are parallel. A concentric ring light surrounds the camera named the camera light. The ring light mounted around the nozzle is called the glue light. A more detailed view of the robotic setup is provided in Appendix B1.1.

In addition to the depicted components, a workstation and an Arduino One µ-controller are part of the setup. All devices, including the workstation, robot, camera, and µ-controller, are interconnected through Ethernet. The computer executes the main production program and runs the CV models. It also sends commands to the robot and µ-controller and receives responses. The robot performs the manipulation and placement of inserts and the movement of the nozzle and camera. Furthermore, it controls the ejector of the vacuum gripper using 24V digital input/output (DI/DO) signals. The µ-controller is responsible for the control functions of the glue pump and the ring lights. The camera sends live streams or images on the GigE standard directly to the computer. A schematic illustration of the components, their tasks, and the communication is given in Appendix B1.2.

7.1.2 Procedure

Steps 1-4

In Sections 5.2 and 6.2, the initial four steps of the procedure are conducted. Four process-relevant parameters are identified for the automation of the two processes:

- Location-dependent:
  - Bore location
  - Tab location
  - Inlet/outlet location
- Time-dependent:
  - Glue level

The derived CV tasks are presented in Figure 7.2. Accurately placing an insert requires knowledge of the bore locations. This problem is thoroughly examined in Sections 5.3, 5.4, and 5.5. The tabs are stored in trays with a clearance larger than the fit of the insert and bore. Therefore, their position is required for precise picking with the suction gripper. In order to position the adhesive nozzle on the inlet hole on the tab, the locations of the ventilation holes are required. Lastly, determining the glueing level is necessary to control the glue dispensing process (s. Sections 6.3 and 6.4).

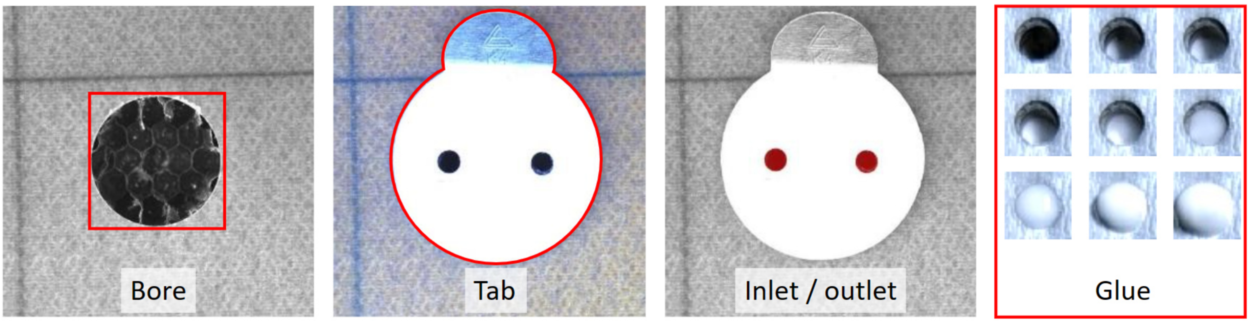


Figure 7.2: Derived CV tasks of both reference processes.

The previous chapters do not cover the detection of the tab and the ventilation holes. Thus, the CV tasks are specified. The picking point of the tab is defined by its ventilation holes, with the centre point between the two holes serving as the target position. Due to the handling flap, a bounding box centre does not represent the target point. Hence, the ventilation holes are selected as product characteristics to be identified for precise insert picking.

A two-step detection is proposed since the holes are always on the tab. First, the tab is identified in the tray or on the panel. Then, the ROI (the tab) is forwarded for ventilation hole detection. Based on the setup, one hole is selected as the inlet hole and the other as the outlet hole. The spatial positions of the tab and the ventilation holes on the image are required to calculate the real-world locations. Thus, the missing CV tasks are tab detection and ventilation hole detection.

Step 5: CV model development

The same model used for bore detection is proposed for tab detection, employing a YOLOv5 model trained on tab images to predict a bounding box encompassing the tab. The required accuracy of the bounding box is not high; it is sufficient if the bounding box encloses exactly one tab with both ventilation holes.

The predicted bounding box is cropped and then forwarded for hole detection. The holes themselves exhibit high contrast against the reflective surface of the tab. By intentionally overexposing the tab area through changes in ring light intensity, the holes become detectable by thresholding based on intensity values. Following further refinement, the positions of the holes on the patch are extracted with high accuracy. A conventional model is sufficient in this case, although a learning-based CV model could also be applied. This, however, requires the creation of a training dataset. The real-world coordinates are calculated with the spatial information of the ventilation holes in pixels. The centre between the hole coordinates is used for picking the insert.

The tab and ventilation hole detection process is illustrated in Figure 7.3. Image a) displays the entire image captured by the camera, showing an already placed insert on the panel. Tab detection predicts the bounding box containing the tab (red box). The patch inside the bounding box is cropped and used for hole detection (image b). It provides a close-up view of the insert. The patch is then binarised (image c), cleaned, and a binary mask is generated (image d). Finally, the mask is overlaid onto the cropped patch (image e). The pixel positions of the hole centres are converted to pixel locations on the original image and then transformed into real-world coordinates.

The same process is repeated with the insert in the bore for placing the glue dispensing nozzle. This time, one of the ventilation holes is directly selected to position the nozzle accurately. Also, for the glue classification, the ventilation hole positions are necessary. Once

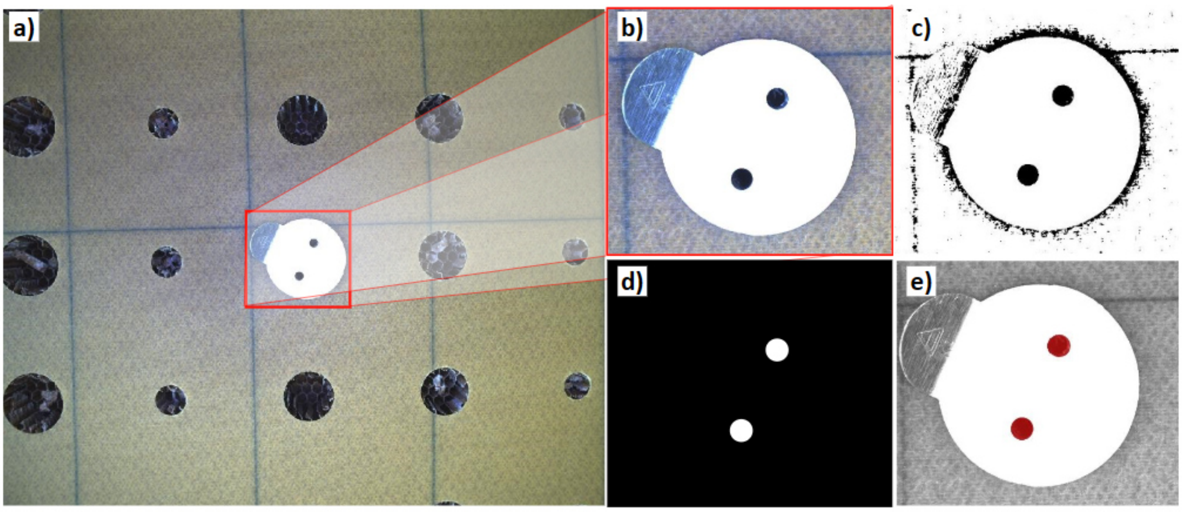


Figure 7.3: a) Tab detection with bounding box. b) Cropped bounding box. c) Binarised image. d) Cleaned binary mask. e) Overlayed mask with the cropped image.

the nozzle is placed on the inlet ventilation hole and the glueing process is started, the outlet hole must be monitored. To expedite outlet hole classification, the patch containing the outlet hole is cropped from the live video stream and classified using the glue classification model. Reducing the original 5 MP image to a 51x51-pixel image containing only the outlet hole enhances classification speed significantly (s. Section 6.4.7).

In summary, the following models are implemented on the technology demonstrator:

• Bores:	YOLOv5 object detection model
• Glue level:	CNN ResNet50 classification model
• Insert/tab:	YOLOv5 object detection model
• Ventilation holes:	Binarisation and thresholding based on the tab bounding box

7.1.3 Testing & Results

The four developed CV models are implemented on the workstation of the technology demonstrator. For testing the demonstrator, the following information is provided to the computer. The product data includes panel dimensions, the position of bores on the panel, and insert types for each bore. The insert data consists of the tray locations for each insert type. The panel and tray are then positioned in the workspace with rough alignment to a reference point, and the demonstration program is started.

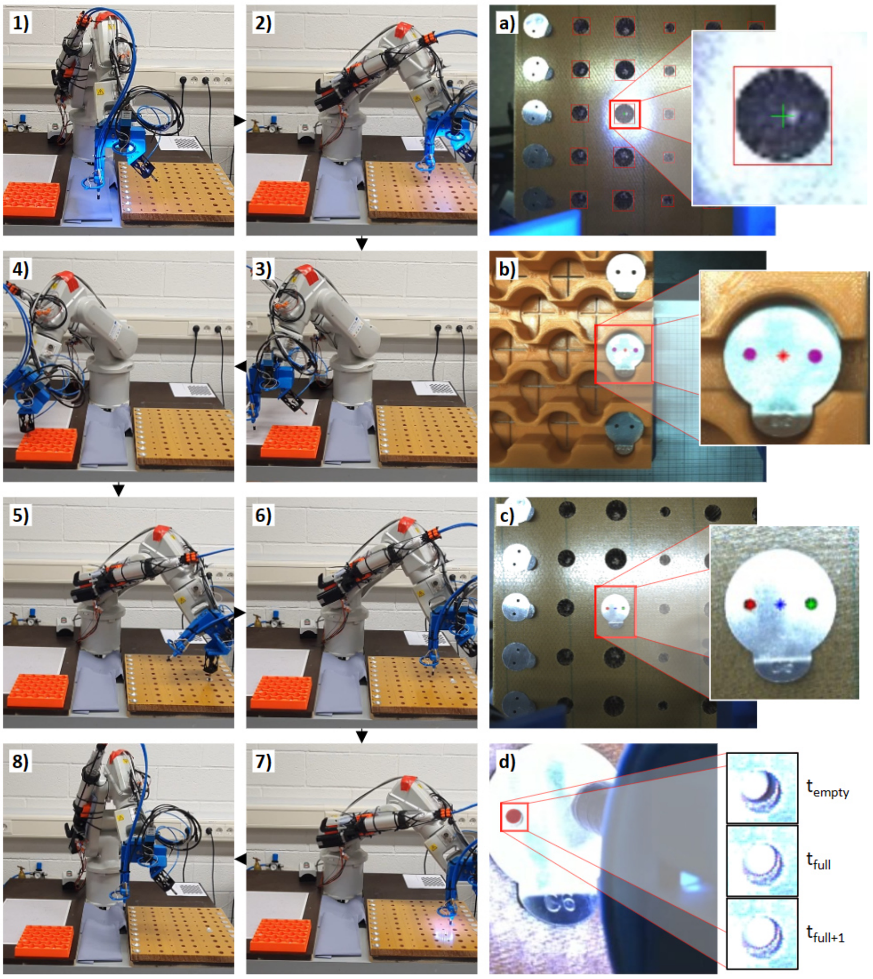


Figure 7.4: Implementation of the technology demonstrator. 1)-8) Robotic automation of the reference processes. a)-d) View of the wrist camera with CV model outputs.

The execution of the program is shown in Figure 7.4, images 1)-8), along with the CV model results in images a)-d). The Appendix B1.3 provides larger versions of each image of Figure 7.4. The first image depicts the robot in the initial position. The robot moves to the assumed bore location based on panel product information (image 2). An image is captured, and the bore detection model identifies the desired bore on the image. The detection result is shown in image a), with the target bore centre highlighted by a green cross. The pixel location is converted into real-world coordinates and stored for insertion.

Next, the robot moves to the defined insert pickup location for the corresponding insert type (image 3). Another image is taken, and the tab and ventilation hole detection programs determine the target position for the suction gripper. Tab detection provides the ROI, while ventilation hole detection identifies the centre point between the ventilation holes, indicated by a red star in image b). The tool is then switched to the suction gripper, and the insert is picked (image 4).

The robot carries the insert to the stored location and inserts it into the bore (image 5). The tool is changed back to the camera, and an image is captured to determine the positions of the inlet and outlet holes (image 6). The tab and ventilation hole detection programs again locate the holes on the tab. This time, the algorithm selects one hole as the inlet and the other as the outlet. Since the nozzle and camera do not move relative to each other, the hole closer to the nozzle is defined as the inlet. As a result, the nozzle itself cannot obstruct the outlet hole, and the camera can monitor the outlet. The detection result in image c) shows the inlet marked with a green star, the outlet with a red star, and the central position in blue. The pixel position of the inlet (green star) is converted into real-world coordinates.

Then, the tool is changed to the nozzle and moved to the calculated coordinates to dispense adhesive into the panel. The glue light is turned on, and the glueing process is initiated (image 7). While filling the adhesive, the camera monitors the outlet hole. The outlet location is cropped and fed to the glue-level classification model.

Image d) displays the tab with the outlet and nozzle, along with the last classified images. The image labelled *empty* represents the last outlet patch classified as empty, while the images labelled *full* and *full+1* are the first and second images classified as full. In contrast to Chapter 6.3, the decision rule is adjusted to an earlier stopping time due to the delay induced by the hoses connecting the pump and nozzle. After completing the glueing process, the robot is moved back to the starting position (image 8).

The robot movements are realised using parametrised device primitives or skilled motions. Device primitives refer to basic robot motions, such as linear or joint movement to a defined position in the workspace. A skilled motion implemented in the demonstrator is a parametrised sequence of device primitives for picking an insert, involving moving a specific distance down along the vertical axis, enabling the suction gripper, and moving a given distance up along the vertical axis.

Overall, the technology demonstrator successfully showcases the automation of the reference processes. The developed CV programs provide the necessary information for process automation with sufficient accuracy. In all trials, the programs accurately determined the identified process-relevant parameters. Multiple trials have demonstrated that industrial processes can be automated by leveraging the potential of learning-based CV models. The lab demonstration achieves Technology Readiness Level 4 (European Commission, 2014). Overall, the demonstration meets the requirements set by the industrial partner.

7.1.4 Limitations

Sensor Feedback: The system relies solely on a camera as a sensor. In the event of problems with picking and insertion, there is no feedback mechanism for the system to react and make adjustments accordingly.

Suction Gripper Design: The suction gripper is attached to a spring plunger. However, depending on the insert's footprint and the presence of residual core material, the spring may be too weak to push the insert into the bore against the resistance of the core material. Instead, the spring is compressed and compensates for the insertion motion.

Structural Rigidity of the MFEE: The MFEE, composed of 3D-printed parts, lacks overall rigidity. Under load, the parts may experience relative movement, leading to changes in the defined Tool Center Points (TCPs). This movement can introduce variability and inaccuracies in the system's performance.

Limitations of Insert Design: Some inserts used in the process do not have an assembly chamfer and are not designed for automation. Due to the tight tolerances of the insertion task and the sharp edges of the insert, the placement of these inserts may be challenging and result in unsuccessful insertions.

Glueing System: The glueing system employed is an adapted handheld glue pump. Due to its larger size, it is difficult to mount it directly on the robot without compromising its manoeuvrability. As a result, the pump is not directly mounted on the end effector. The long connection hoses between the pump and nozzle introduce significant delays in the pump system. Additionally, these hoses can cause adhesive leakage from the nozzle, even when the process is executed and stopped accurately. However, this drawback was accepted due to the substantial cost difference between the adapted handheld device and an industrial glue pump, which is 50-100 times more expensive.

7.2 Case Study 2: Processing of Lithium-Ion Batteries

The second case study results from a collaboration with a Luxemburgish SME primarily working with li-ion battery packs. Here, the target is again to automate a manual process for the technical treatment of battery packs and cells. However, due to confidentiality, only limited information on the process is explained, and parts of the procedure are kept short.[10]

7.2.1 Procedure

For identifying the process-relevant parameters and developing learning-based CV models, the procedure proposed in Section 4.4 is applied to the treatment process for battery packs. This section briefly presents the individual steps.

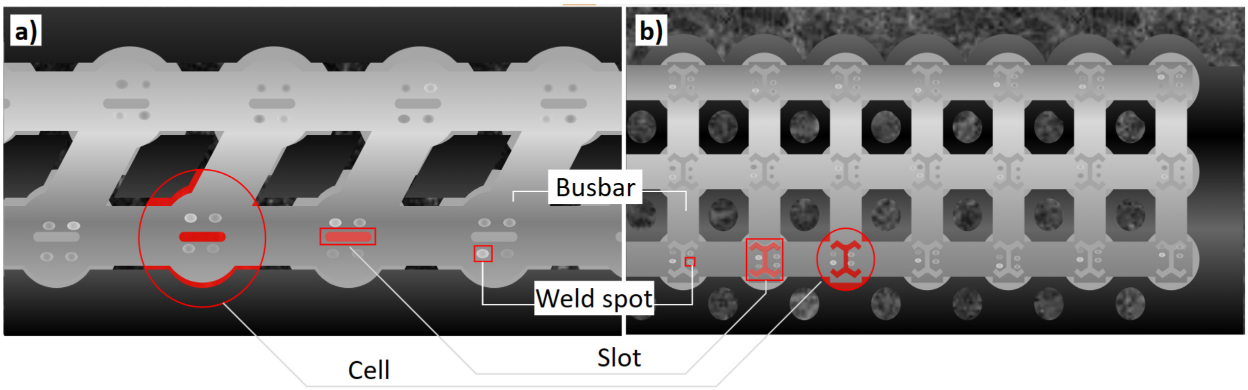


Figure 7.5: Schematic illustration of a) Battery A with a horizontal slot, b) Battery B with an x-shaped slot and more cells per pack.

Step 1: Product analysis

Various battery packs from different mobile applications with different dimensions and geometries are processed at the SME. Two packs are considered for this study, given in Figure 7.5. The packs have a different number and configuration of the single li-ion cells. Furthermore, each pack variant has unique busbars connecting the cells. The busbars have slots that are cutouts of the material. Below the cutouts, the electrode is visible. The packs use all the same li-ion cell types. The cells are arranged so that the electrodes lie in one plane. The busbars are attached to the cells via spot welds.

[10] The results of this section are partly based on a master thesis of Çağrı Ustundaĝ.

Step 2: Process analysis
The technical treatment process focuses on the busbars of the battery pack, although specific details remain confidential. Due to the sensitivity of the li-ion cells, careful handling is crucial to avoid short circuits or damage to the cells. Especially contact with the electrode must be avoided. The treatment process includes tasks executed directly at and near the weld spots.

Step 3: Process-relevant parameters
To ensure the accurate execution of the treatment, attention must be paid to the weld spots and slots. Each cell has one slot and is attached to the busbar with at least four weld spots. The slots vary depending on the battery pack variant. The position of the weld spots and slots relative to the cell and each other may differ. The distance between the height of the electrode plane is known for each pack, but the specific slot and weld spot positions are missing. However, the company receives the parts as they are and has no control over prior processes. In order to conduct the treatment process, the location-dependent parameters of slots and weld spots are required.

Step 4: CV task definition
Object detection or instance segmentation models are suitable for determining location-dependent parameters. After a discussion with the company, it is defined that a bounding box encompassing the slot is sufficient for the task. Additionally, an approximate location of the weld spots is required. A bounding box provides sufficient accuracy since special treatment is performed at the weld spots and their neighbouring area. Therefore, object detection models for slot and weld spot detection are developed.

Step 5: CV model development
During the project, newer YOLO versions were available compared to Section 5.5. Consequently, the updated versions are selected, implemented, and compared. The preselection is based on their performance on standard datasets, inference time, and documentation. This thesis presents only the results of the finally selected model YOLOv8.

After selecting the model, the dataset is generated and prepared. The received images show two different battery packs (s. Figure 7.5). The images are greyscale and captured with an industrial camera perpendicular to the surface. The images of the battery pack A show not more than eight cells with a horizontal slot (Figure 7.5 a). The resolution is 4.2MP. The other pack has up to forty cells per image and a resolution of 5MP (Figure 7.5 b). As a result, the object sizes differ between the images, with weld spots and slots having a larger pixel area for battery pack A compared to battery pack B. Moreover, both images exhibit a high noise level, including reflections, dirt, and material residues on the busbars.

When loading the images into the model, they are compressed to an input size of 640x640 pixels. Consequently, the weld spots become very small. Since many CV models struggle with small objects, a sliding window procedure is implemented, cropping the images into patches to reduce information loss. Each patch is then processed separately by the model and stitched together to reconstruct the whole image after processing all cropped patches.

The dataset consists of 275 images of battery pack A and three images of battery pack B available for annotation. Especially for battery B, the number is very small. Although cell annotation is not necessary for the process itself, it is performed at the request of the SME to highlight the detected slots and weld spots per cell. Various augmentation techniques increase the dataset size, including reflection, rotation, cropping, zooming, contrast and brightness adjustments, and mosaicking. The mosaicking technique, introduced with YOLOv4, involves cropping parts of different images and stitching them together to create new images. After augmentation and balancing, a total of 275 images for each pack are obtained, which are subsequently split into training, validation, and test sets in an 80%:10%:10% ratio. A pretrained YOLOv8x model was trained for 100 epochs on the training set using early stopping, with the best result achieved after 79 epochs. No issues were encountered during the training process.

7.2.2 Results & Discussion

The results obtained on the test set are presented in Table 7.1. The mAP is calculated for different IoU thresholds for each label. The mAP50, with an IoU threshold of 0.5, indicates a high overall performance of the model to detect all labelled object classes. The detection accuracy for cells and slots of both batteries is 100%, and for weld spots, 89%. As the IoU threshold increases, the performance for cells and slots remains consistently high. However, there is a significant decrease in performance for weld spots. The mAP50:95 is 42%-points lower compared to the mAP50. The training process on Google Colab using GPU took 56 minutes. The inference time with GPU is 92 ms per image. Predictions are made on each crop in the sliding window approach, increasing the overall prediction time. For instance, if there are 10 sliding window crops, the total inference time would accumulate to 920 ms.

Table 7.1: Results of YOLOv8x on li-ion test dataset.

Label	mAP50	mAP75	mAP50:95
Cell	1	1	0.961
Slot Battery A	1	0.990	0.840
Slot Battery B	1	1	0.983
Weld spot	0.890	0.453	0.470
Total	0.973	0.859	0.813

Model: YOLOv8.0.115. Weights: yolov8x.pt. Augmentation: offline. Python version: 3.10.12. Framework: torch-2.0.1+cu118. GPU: Tesla T4.

Figure 7.6 displays schematic example predictions for both battery packs. In the predictions, green circles indicate cells where a slot and four or more weld spots are correctly predicted. Cells with fewer than four weld spot predictions or no slot are highlighted in red. The example shows that the size of the cells, slots and weld spot on Battery B is much smaller in pixel size compared to Battery A. The change in the grey level on the busbars represents variations in illumination and high noise, strongly visible in the actual dataset.

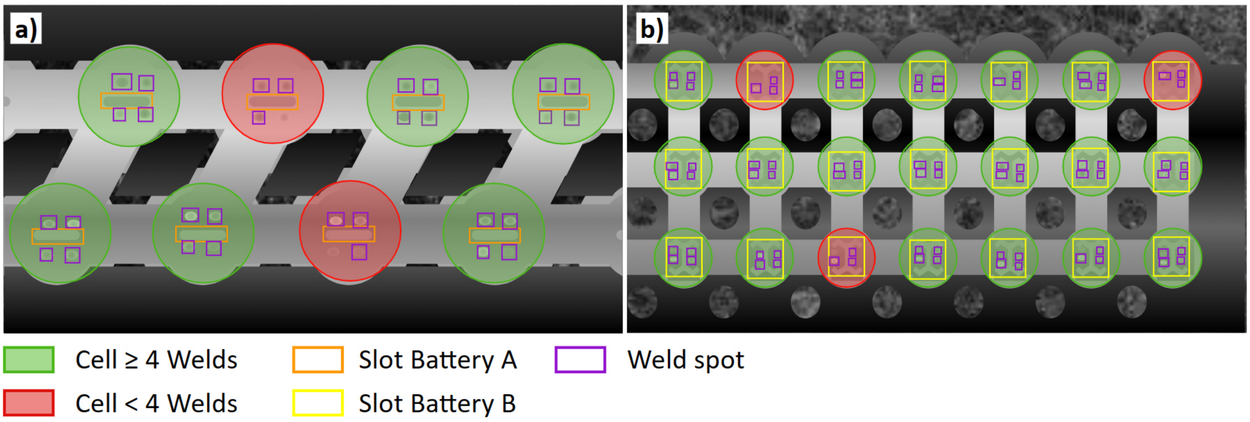


Figure 7.6: Schematic illustration of prediction results for a) Battery A and b) Battery B.

Figure 7.7 presents examples of correct and incorrect predictions. From all images, the high noise in the data is visible. For Battery B, the input images of the positive predictions differ tremendously. The left image in Figure 7.7 b) is dark, and the right image is overexposed. Furthermore, the weld spot is often not well pronounced and difficult to identify. Still, the model can identify all objects. Regarding the mispredictions, the presence of residue material obscures the weld spot and its surrounding region in Figure 7.7 c). For Battery B depicted in Figure 7.7 d), the slot on the left appears darker than the busbar, whereas it is brighter on the right image. Once again, the weld spots are difficult to discover. On the right side, scratches on the busbar make the weld spot nearly invisible.

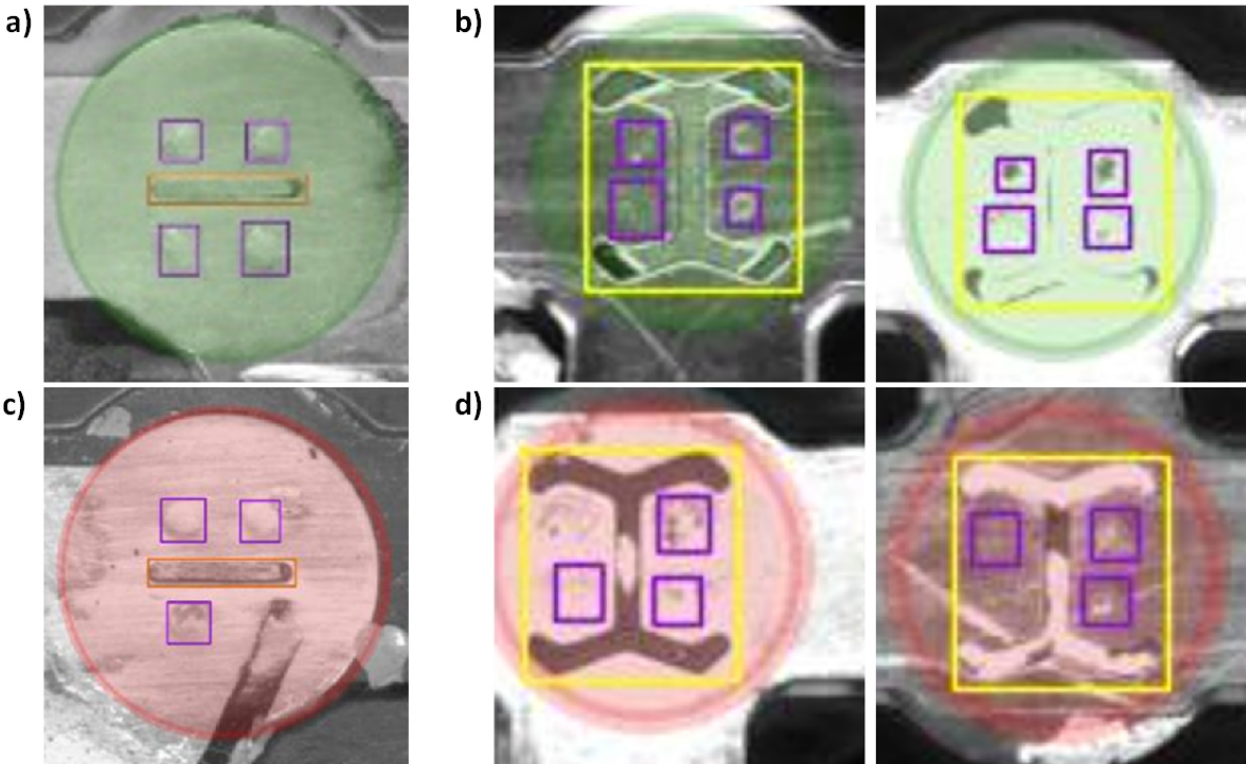


Figure 7.7: Correct (a,b) and incorrect (c,d) prediction results for Battery A (a,c) and Battery B (b,d).

The detection of cells, slots, and weld spots demonstrates a remarkable level of accuracy and confidence, considering the challenges posed by the dataset. Although there are instances where some weld spots are not detected, given the conditions under which these spots were not identified, the overall performance remains impressive. It is worth noting that the dataset used in this study is relatively small, particularly for battery B, which only has three fully annotated images. This limitation arises from the extensive effort required for image annotation. An average photo with 35-40 cells necessitates annotating a total of 210-240 objects per image.

Overall, the project partner is satisfied with the presented results. Moving forward, the aim is to utilise the developed model for semi-automatic labelling to increase the dataset. Thus far, the model has been applied solely to collected images. The next logical step involves implementing the model into the technical treatment process.

7.3 Case Study 3: Welding of Bent Sheet Metal Parts

Another case study is conducted with a manufacturer of bent sheet metal parts located in the Grand Region Luxembourg. This company represents an SME that produces only parts based on customer demand, typically in volumes ranging from 10 to 100 pieces per order. The manufacturing process involves laser cutting, bending, and welding sheet metal components. This study aims to develop an unwelded edge detection program enabling welding path generation and, finally, automatic robotic welding.[11]

7.3.1 Procedure

Step 1: Product analysis
The SME in this study specialises in producing welded bent sheet metal parts using steel or stainless steel materials of varying thicknesses. The geometries of these parts are customised and unique to each order, but they are consistently manufactured from cut metal sheets that are bent into shape and welded. The unwelded edges of the parts can range from simple straight lines to more complex geometries with different curvatures. The surface of the material ranges from highly reflective to matte. Examples of these parts are presented in Figure 7.8 a).

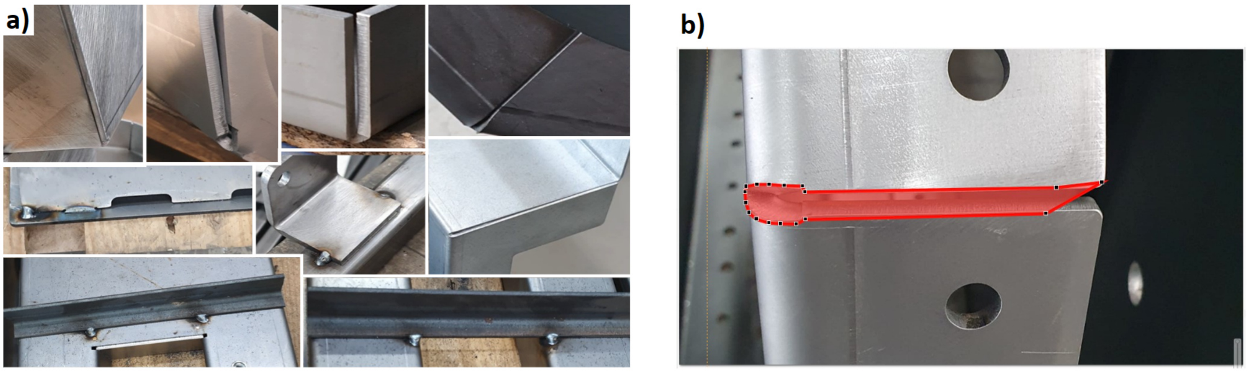


Figure 7.8: a) Example images of the dataset. b) Annotation process of unwelded edge instances with polygons.

[11] The results of this section are partly based on a student project of Jerôme Bortuzzo.

Step 2: Process analysis

This project's focus is on welding, which represents the joining task. Therefore, the bent parts are fixed on a jig and mounted onto a welding table. Assemblies of multiple parts are attached to each other using spot welds before welding. In all cases, fusion welding is performed using a torch. The welding parameters, such as torch speed and wire feed rate, are adjusted based on the type of material, thickness, and the gap between the unwelded edges. Workers weld the parts based on printed product drawings in the current manual process. Robot-based welding automation exists for some parts with larger volumes, where welding parameters are predefined based on material settings. However, the welding path is preprogrammed, and the welding process only functions correctly when the correct part is placed on the correct jig in the designated position on the welding table.

Step 3: Process-relevant parameters

The welding process requires specific parameters that depend on the properties of the bent sheet metal. The type of bent sheet metal is defined in the order, and this information is available. A precise welding path for the torch is necessary to complete the welding process. The rough location of the welding path can be estimated if the type of jig and its position on the welding table is known. However, due to manual positioning and fixation inaccuracies, the exact location of the welding path within the workspace is unknown. Therefore, determining the path in the workspace is a necessary task related to location-dependent parameters.

Step 4: CV task definition

For the proposition of a welding path, the unwelded edge must be identified in the image. The edge can be detected as a whole or by identifying individual support points. For instance, two support points are sufficient to reconstruct a straight line. However, the shape of the edge may be more complex than a straight line. Therefore, predicting support points with bounding boxes is critical. With instance segmentation, the whole unwelded edge is identified on pixel level in a single prediction. Hence, instance segmentation is selected for unwelded edge detection.

Step 5: CV model development

One of the most popular segmentation models is the Mask R-CNN proposed by He et al. with more than 10k citations (He et al., 2017). Despite its age, it remains a reference model in many benchmarks. An improved version of Mask R-CNN with better performance is available in the Detectron2 framework, which is selected for this study. Detectron2 is developed by Facebook AI Research (FAIR), offers excellent documentation, and receives regular updates.

For the training of the model, a dataset is required. Since no images of the parts are available, a new dataset is created. Images and videos are captured from bent sheet metal parts positioned on shelves and carriages in the buffer zone before welding. Images of parts on jigs and welding tables are not available. In total, 515 images of 24 different specimens are generated using a standard smartphone. Further examples are provided in Appendix B2.1. Each image in the dataset is annotated with polygons that enclose the unwelded edges to be welded. The number of support points required for the polygons varies between 7 and 20, depending on the complexity of the shape. Figure 7.8 b) illustrates an example of the data annotation process.

Online data augmentation is applied to increase the dataset size. Each image is augmented before being loaded into the model by randomly applying functions such as reflection and rotation. Furthermore, brightness, contrast, and saturation are adjusted. However, the augmented images are not permanently stored, which helps reduce the dataset size on the hard drive. The dataset is split into training, validation, and test sets in an 80%:10%:10% ratio. A pre-trained model of the modified Mask R-CNN available within the Detectron2 framework is trained for 100 epochs using early stopping. The best model is received after 50 epochs of training.

7.3.2 Results & Discussion

The model's performance on the test set is presented in Table 7.2. A single class is used, representing all unwelded edges, and the mAP50 is measured at 89%, indicating a good result. The majority of unwelded edges are successfully identified. However, as the IoU thresholds increase, the mAP decreases. Figure 7.9 provides examples of positive predictions where the highlighted polygons align well with the unwelded edges. Notably, the results are robust to varying backgrounds (e.g., metal table, pallet, desk, air cushion foil), and multiple unwelded edges within an image are accurately detected. The polygons exhibit high accuracy in these cases, with confidence scores consistently exceeding 90%.

Table 7.2: Results of Detectron2 Mask R-CNN model on unwelded edge test dataset.

Model	mAP50	mAP75	mAP50:95
Detectron2 Mask R-CNN	0.893	0.589	0.519

Model: R50-FPN. Weights: mask_rcnn_R_50_FPN_3x. Augmentation: Online. torch: 1.11; cuda: cu113; detectron2: 0.6.

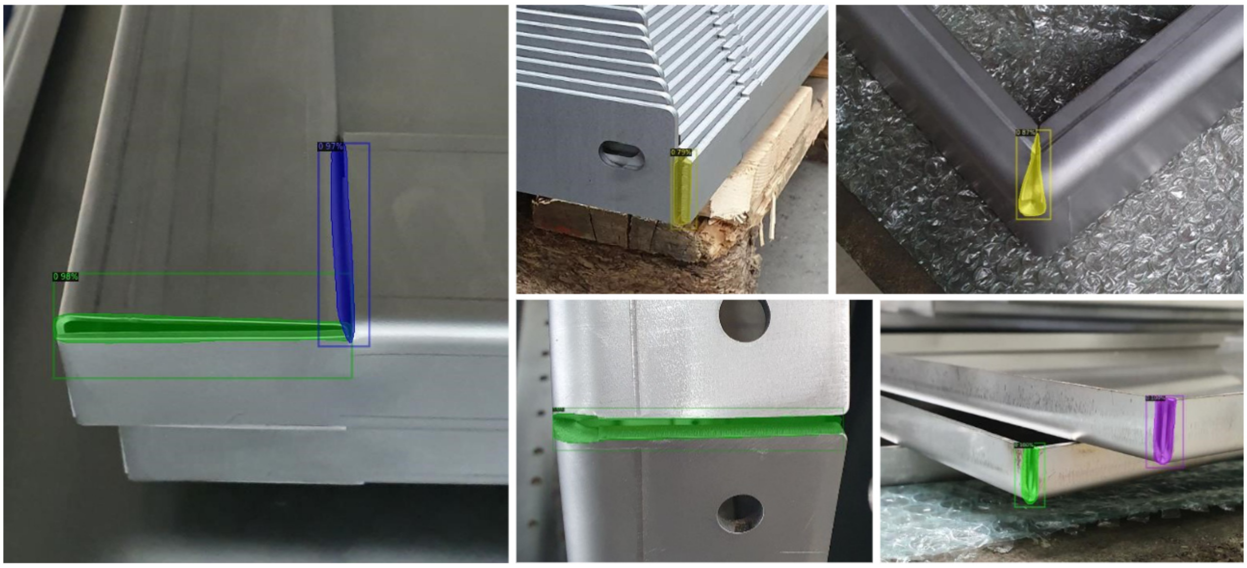


Figure 7.9: Instance segmentation results. Further examples are provided in Appendix B2.2.

During the evaluation of the results, three root causes for false predictions are identified (s. Figure 7.10). The most common issue is overlapping instances, where a second instance, either smaller (image a) or equal in size (image b) to the primary instance, is predicted with lower confidence. Proper post-processing techniques can address this by identifying instances with high overlap and suppressing the predictions with lower confidence scores.

Another root cause is corrupted instances, where only partial segments of the unwelded edges are detected, resulting in incomplete representations (s. Figure 7.10 c). Reconstructing these segmented results without additional information, such as CAD data, including the welding seam, is challenging. In particular, the corrupted instance in Figure 7.10 c) represents an unwelded edge with two spot welds. Instead of having one large instance, the edge should be divided into three separately annotated sections, enhancing training and segmentation. Additionally, the part exhibits two different joint types. The corrupted instance (white) corresponds to a lap joint with only one visible laser-cut edge, while the correctly detected

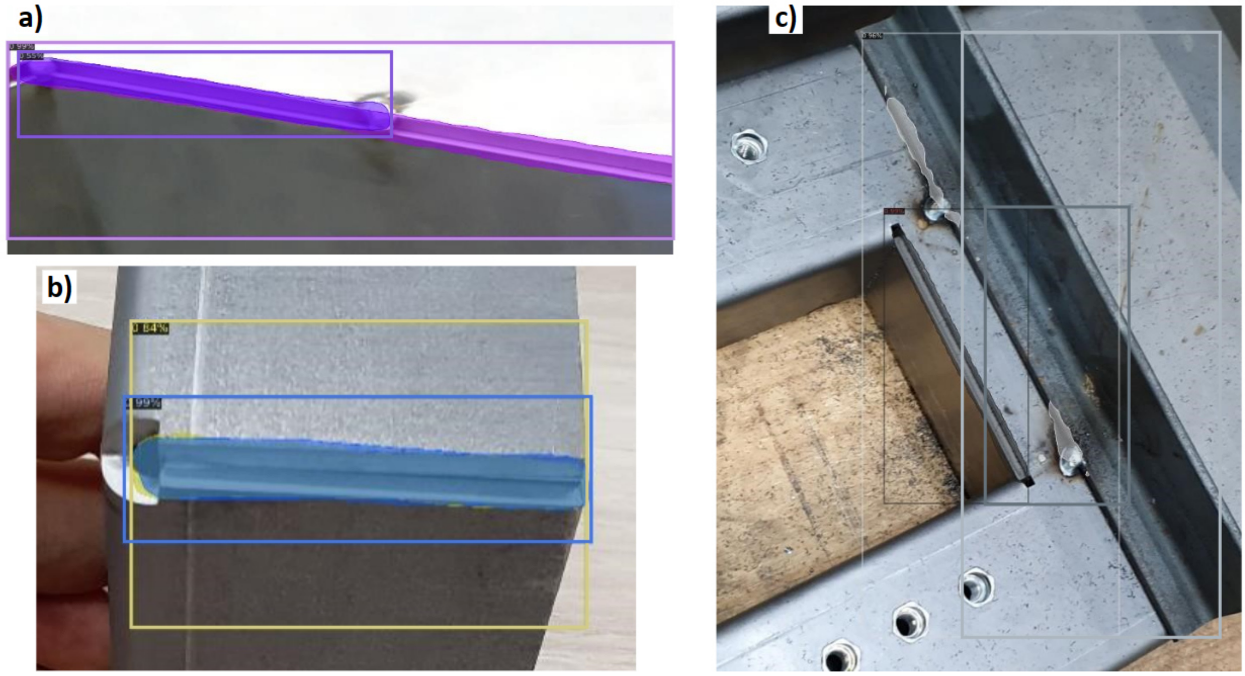


Figure 7.10: Problematic instance predictions. a) & b) Overlapping instances. c) Corrupted instances.

instance (anthracite) represents a corner joint with two orthogonally oriented cut edges. Assigning different labels to these joint types can improve the segmentation results.

False predictions also occur due to the similarity between unwelded edges in joints and cut edges of the bent sheet metal that are not subject to welding. In these cases, an edge not part of a welding joint is identified as an unwelded edge. Correcting these predictions automatically without additional input is again challenging. Generally, incorporating more part information improves the segmentation process as characteristics of identified edges, such as length, width, orientation, or type, can be checked for plausibility.

Overall, the unwelded edge segmentation results are promising. Considering the limitations of the dataset, such as the low number of specimens, background variations, and annotations, the predicted polygons exhibit high accuracy. They can serve as input for weld path generation. Appendix B2.3 provides a scene reconstruction with path approximation based on the instance bitmask. Additionally, the bitmask of the polygon segmentation is usable as an ROI for a depth sensor. Using the polygon on the colour image as a basis for extracting the depth map can enable the automation of the welding process for customised parts.

8 Summary and Outlook

Automating assembly processes in HMLV manufacturing is still challenging for many SMEs. Thus, these companies rely even today on a significant amount of manual operations with an overall low degree of automation. However, automation offers numerous benefits, such as increased productivity, higher quality, improved delivery reliability, and reduced rework and scrap. The emergence of AI-based algorithms has paved the way for assembly automation solutions that are compatible with multiple products while maintaining overall production flexibility. However, the adoption of such technologies in the HMLV industry remains low.

This thesis aims to integrate production engineering methodologies with CV, machine learning, and deep learning techniques to enable automation in this HMLV context. Chapter 2 analyses the relevant domains of HMLV, assembly automation, and learning-based CV. It is found that despite extensive research conducted, the presented approaches are largely individual and confined to laboratory settings or non-industrial use cases. Furthermore, there is a lack of guidelines on leveraging the potential of learning-based CV for assembly automation in HMLV.

In Chapter 3, the research objectives are defined as follows. The first objective is to develop a procedure that utilises learning-based CV methods to enable assembly automation, particularly the joining task, for companies and SMEs operating in HMLV environments. In the considered context, the main challenge lies in coping with high product variance rather than the complexity of the joining tasks. The second objective is the experimental analysis of such CV models regarding their performance, flexibility, and robustness, necessary for application in HMLV automation. Lastly, the procedure is validated in different industrial use cases.

Chapter 4 focuses on the first thesis objective and covers the procedure development. The procedure provides a systematic approach to identifying and determining parameters critical for enabling the automation of joining processes in HMLV assembly. Based on analysing typical joining motions and their combination with the geometric dimensionality of the joining task, two types of process-relevant parameters are defined. Location-dependent parameters relate to specific positions during the joining process, and time-dependent parameters vary with time during the joining execution. The developed procedure consists of five steps. Steps 1 and 2 derive the functional requirements of the existing products and their joining processes targeted for automation. Step 3 identifies the process-relevant parameters for the considered product-process combination. From there, a suitable type of CV task is proposed to determine the parameters in Step 4. The procedure concludes with a complete pipeline for developing learning-based CV models in Step 5.

Chapters 5 & 6 present the experimental analysis of the developed CV models for the reference joining processes. In Chapter 5, the reference process is the insertion of inserts into lightweight panels. Following the developed procedure, the positions of the panel bores are identified as a location-dependent process-relevant parameter. Object detection models are suitable for determining the parameters. In a pre-study, various conventional and learning-based models are implemented for bore detection on four products. Although conventional models achieve high prediction accuracy to some extent, they are vulnerable to product changes. The learning-based CV models offer robust detection across changing products. The experiments in the main study analyse the updated YOLOv5 detector regarding prediction accuracy, product flexibility, and robustness against changing lighting conditions. The model consistently achieves a high mean average precision of over 94% in all trials on existing product variants and new ones unknown to the model. The impact of the substantial environmental changes is minimal, emphasising the model's robustness.

Chapter 6 focuses on the analysis of time-dependent parameters. For the reference glueing process, the glue level is identified as a time-dependent process-relevant parameter. Different conventional and learning-based CV models are compared regarding their ability to accurately predict the glue level. Once again, learning-based models achieve the highest performance and classify the glue level with an accuracy of over 99%. One best-performing model is cross-validated using a Monte Carlo simulation with 500 iterations to identify the impact of the dataset on the classification results. The average accuracy is nearly unchanged, with over 98%. The learning-based CV model is then applied to classify live streams of the glueing process. Despite different hardware, image sizes, and fluctuating lighting conditions, it accurately predicts the filling level in all glueing trials. In summary, Chapters 5 & 6 demonstrate the suitability of learning-based CV models for application in HMLV processes with changing products and dynamic environments.

Chapter 7 presents the development of a technology demonstrator encompassing both reference processes discussed in Chapters 5 & 6. The demonstrator comprises an industrial robot with a suction gripper, a glue pump, and a wrist camera. Four process-relevant parameters, including bore location and glue level, are identified with the proposed procedure. A suitable CV model is developed and implemented for each process-relevant parameter. Overall, the technology demonstrator successfully showcases the automation of the reference processes and achieves TRL4, with the developed CV systems consistently providing the necessary information with sufficient accuracy. However, limitations remain solely related to the technical system and not the developed CV models themselves. The chapter furthermore presents two additional industrial case studies: a technical treatment process of lithium-ion batteries and a welding process of bent sheet metal parts. Due to challenging detection and segmentation tasks, determining the identified process-relevant parameters is very complex in these cases. However, the detection performance of the models remains high, with a mean average precision of 97% for detecting specific characteristics on battery pack bus bars and 89% for segmenting unwelded edges on bent sheet metal. The developed CV models are crucial enablers for the automation of these processes.

In conclusion, this research demonstrates the potential of learning-based CV systems for automating assembly processes in HMLV scenarios. The developed procedure provides a systematic approach to identify critical information required for automation, i.e., the process-relevant parameters. Furthermore, it covers all steps to determine the parameters, from selecting the appropriate learning-based CV models to their implementation. The experiments highlight the flexibility and robustness of the models in dealing with changing products and environments, meeting another important requirement of HMLV assembly scenarios.

This research serves as a starting point, and validation in further use cases, including technical implementation, will facilitate a more comprehensive evaluation of the procedure. Additionally, CV models alone are sometimes insufficient, and including additional depth or tactile sensor systems may be necessary to solve specific tasks. Therefore, further research should focus on integrating such sensor systems into the provided procedure. While this research primarily focuses on the joining task in assembly, other aspects, such as material handling, can be added to cover the complete range of assembly processes.

Bibliography

Acharya, Joydeep, Yasutaka Serizawa and Sudhanshu Gaur (2019). “Emerging Technologies: Connecting Millennials and Manufacturing”. In: *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, pp. 180–183. ISBN: 978-1-7281-6737-4. DOI: 10.1109/CogMI48466.2019.00034.

Adobe (2023). *Lizenzfreie Stockfotos und Bilder*. URL: https://stock.adobe.com/de/photos (visited on 19/05/2023).

Akerkar, Rajendra (2019). *Artificial Intelligence for Business*. SpringerBriefs in Business. Cham: Springer International Publishing. ISBN: 978-3-319-97435-4. DOI: 10.1007/978-3-319-97436-1.

Almabdy, Soad and Lamiaa Elrefaei (2019). “Deep Convolutional Neural Network-Based Approaches for Face Recognition”. In: *Applied Sciences* 9.20, p. 4397. ISSN: 2076-3417. DOI: 10.3390/app9204397.

Andrieu, Christophe et al. (2003). “An introduction to MCMC for machine learning”. In: *Machine learning* 50.1, pp. 5–43. DOI: 10.1023/A:1020281327116.

Apostolopoulos, Ioannis D. and Mpesiana A. Tzani (2022). “Industrial object and defect recognition utilizing multilevel feature extraction from industrial scenes with Deep Learning approach”. In: *Journal of Ambient Intelligence and Humanized Computing*. ISSN: 1868-5137. DOI: 10.1007/s12652-021-03688-7.

Awad, Ramez (2017). “Automatisierungs-Potenzialanalyse für die Montage - Produktdatenblatt”. Stuttgart.

Ballard, D H et al. (2016). “Object Detection using Circular Hough Transform”. In: *Pattern Recognition* 25.1, pp. 0–3. ISSN: 00313203. DOI: 10.1109/BMEiCON47515.2019.8990259.

Bochinski, Erik, Tobias Senst and Thomas Sikora (2017). “Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms”. In: *2017*

IEEE International Conference on Image Processing (ICIP). IEEE, pp. 3924–3928. ISBN: 978-1-5090-2175-8. DOI: 10.1109/ICIP.2017.8297018.

Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao (2020). “YOLOv4: Optimal Speed and Accuracy of Object Detection”. In.

Bohnen, Fabian, Matthias Buhl and Jochen Deuse (2013). “Systematic procedure for leveling of low volume and high mix production”. In: *CIRP Journal of Manufacturing Science and Technology* 6.1, pp. 53–58. ISSN: 17555817. DOI: 10.1016/j.cirpj.2012.10.003.

Brecher, Christian et al. (2011). “Integrative Produktionstechnik für Hochlohnländer”. In: *Integrative Produktionstechnik für Hochlohnländer*. Ed. by Christian Brecher. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. 2, pp. 17–82. ISBN: 978-3-642-20692-4. DOI: 10.1007/978-3-642-20693-1.

Bredies, Kristian and Dirk Lorenz (2011). *Mathematische Bildverarbeitung*. Wiesbaden: Vieweg und Teubner. ISBN: 978-3-8348-1037-3. DOI: 10.1007/978-3-8348-9814-2.

Brown, Micheal S. (2019). “Understanding color & the in-camera image processing pipeline for computer vision”. In: *Internationa Conference on Computer Vision*. Seoul.

Burger, Wilhelm and Mark James Burge (2015). *Digitale Bildverarbeitung*. 3. X.media.press. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-04603-2. DOI: 10.1007/978-3-642-04604-9.

Burggräf, Peter et al. (2019). “Automation decisions in flow-line assembly systems based on a cost-benefit analysis”. In: *Procedia CIRP* 81, pp. 529–534. ISSN: 22128271. DOI: 10.1016/j.procir.2019.03.150.

Caccamo, Chiara et al. (2022). “Using the Process Digital Twin as a tool for companies to evaluate the Return on Investment of manufacturing automation”. In: *Procedia CIRP* 107, pp. 724–728. ISSN: 22128271. DOI: 10.1016/j.procir.2022.05.052.

Canny, John (1986). “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6, pp. 679–698. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1986.4767851.

Chaple, Girish N., R. D. Daruwala and Manoj S. Gofane (2015). “Comparisions of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA”. In:

2015 International Conference on Technologies for Sustainable Development (ICTSD). 1. IEEE, pp. 1–4. ISBN: 978-1-4799-8187-8. DOI: 10.1109/ICTSD.2015.7095920.

Chen, Fei et al. (2011). “An assembly strategy scheduling method for human and robot coordinated cell manufacturing”. In: *International Journal of Intelligent Computing and Cybernetics* 4.4, pp. 487–510. ISSN: 1756-378X. DOI: 10.1108/17563781111186761.

Chen, Fei et al. (2014). “Optimal Subtask Allocation for Human and Robot Collaboration Within Hybrid Assembly System”. In: *IEEE Transactions on Automation Science and Engineering* 11.4, pp. 1065–1075. ISSN: 1545-5955. DOI: 10.1109/TASE.2013.2274099.

Chen, Fei et al. (2015). “Design of a novel dexterous robotic gripper for in-hand twisting and positioning within assembly automation”. In: *Assembly Automation* 35.3, pp. 259–268. ISSN: 0144-5154. DOI: 10.1108/AA-05-2015-046.

Chi, Jianning et al. (2019). “A Novel local human visual perceptual texture description with key feature selection for texture classification”. In: *Mathematical Problems in Engineering* 2019.1. ISSN: 15635147. DOI: 10.1155/2019/3756048.

Chu, Fu-Jen, Ruinian Xu and Patricio A Vela (2018). “Real-World Multiobject, Multigrasp Detection”. In: *IEEE Robotics and Automation Letters* 3.4, pp. 3355–3362. ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2852777.

Cramer, Martijn et al. (2018). “Towards robust intention estimation based on object affordance enabling natural human-robot collaboration in assembly tasks”. In: *Procedia CIRP* 78, pp. 255–260. ISSN: 22128271. DOI: 10.1016/j.procir.2018.09.069.

Dalianis, Hercules (2018). “Evaluation Metrics and Evaluation”. In: *Clinical Text Mining*. Cham: Springer International Publishing. Chap. 6, pp. 45–53. DOI: 10.1007/978-3-319-78503-5_6.

De Ginste, Lauren Van et al. (2019). “Defining Flexibility of Assembly Workstations Through the Underlying Dimensions and Impacting Drivers”. In: *Procedia Manufacturing* 39, pp. 974–982. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.01.391.

De Gregorio, Daniele et al. (2020). “Semiautomatic Labeling for Deep Learning in Robotics”. In: *IEEE Transactions on Automation Science and Engineering* 17.2, pp. 611–620. ISSN: 1545-5955. DOI: 10.1109/TASE.2019.2938316.

Deng, Di et al. (2015). “Sensor guided robot path generation for repair of buoyancy module”. In: *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, pp. 1613–1618. ISBN: 978-1-4673-9107-8. DOI: 10.1109/AIM.2015.7222774.

Deng, G. and L.W. Cahill (1994). “An adaptive Gaussian filter for noise reduction and edge detection”. In: *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*. pt 3. IEEE, pp. 1615–1619. ISBN: 0-7803-1487-5. DOI: 10.1109/NSSMIC.1993.373563.

Deng, Jia et al. (2009). “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. IEEE, pp. 248–255. ISBN: 978-1-4244-3992-8. DOI: 10.1109/CVPR.2009.5206848.

DFKI (2017). *Künstliche Intelligenz*. Tech. rep. Bitkom e.V, Deutsches Forschungszentrum für Künstliche Intelligenz.

DIN 8593 (2003). *Fertigungsverfahren Fügen - Teil 0: Allgemeines; Einordnung, Unterteilung, Begriffe*. Frankfurt a. M. DOI: https://dx.doi.org/10.31030/9500684.

DIN V 19233 (1998). *DIN V 19233: Leittechnik - Prozessautomatisierung: Automatisierung mit Prozessrechensystemen*. Ed. by Deutsches Institut für Normung e.V. Frankfurt a. M. URL: https://www.beuth.de/en/pre-standard/din-v-19233/3361842 (visited on 27/01/2023).

Doan, Ngoc Chi Nam and Wei Lin (2017). “Optimal robot placement with consideration of redundancy problem for wrist-partitioned 6R articulated robots”. In: *Robotics and Computer-Integrated Manufacturing* 48, pp. 233–242. ISSN: 07365845. DOI: 10.1016/j.rcim.2017.04.007.

Downs, Anthony et al. (2021). “Assessing Industrial Robot agility through international competitions”. In: *Robotics and Computer-Integrated Manufacturing* 70, p. 102113. ISSN: 07365845. DOI: 10.1016/j.rcim.2020.102113.

Drigalski, Felix von et al. (2022). “Team O2AC at the world robot summit 2020: towards jigless, high-precision assembly”. In: *Advanced Robotics* 36.22, pp. 1213–1227. ISSN: 0169-1864. DOI: 10.1080/01691864.2022.2138541.

Duda, Richard O. and Peter E. Hart (1972). “Use of the Hough Transformation to Detect Lines and Curves in Pictures”. In: *Communications of the ACM* 15.1, pp. 11–15. ISSN: 15577317. DOI: 10.1145/361237.361242.

Eissa, Aly, Mostafa Atia and Magdy Roman (2020). “AN EFFECTIVE PROGRAMMING BY DEMONSTRATION METHOD FOR SMES’ INDUSTRIAL ROBOTS”. In: *Journal of Machine Engineering* 20.4, pp. 86–98. ISSN: 1895-7595. DOI: 10.36897/jme/130944.

Elgendy, Mohamed (2020). *Deep Learning for Vision Systems*. Ed. by Manning. October. ISBN: 9781617296192.

Eng, How Lung and Kai Kuang Ma (2000). “Noise adaptive soft-switching median filter for image denoising”. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 4, pp. 2175–2178. ISSN: 15206149. DOI: 10.1109/ICASSP.2000.859268.

European Commission (2014). *Technology Readiness Levels (TRL)*. URL: https://ec.europa.eu/research/participants/portal4/doc/call/h2020/common/1617621-part_19_general_annexes_v.2.0_en.pdf (visited on 25/07/2023).

– (2022a). *2022 SME Country Fact Sheet Germany*. Brüssel. URL: https://single-market-economy.ec.europa.eu/smes/sme-strategy/sme-performance-review_en (visited on 27/01/2023).

– (2022b). *2022 SME Country Fact Sheet Luxembourg*. Brüssel. URL: https://single-market-economy.ec.europa.eu/smes/sme-strategy/sme-performance-review_en (visited on 26/01/2023).

Everingham, Mark et al. (2010). “The PASCAL Visual Object Classes (VOC) Challenge”. In: *International journal of computer vision* 88.2, pp. 303–338.

Eversheim, W (1989). *Organisation in der Produktionstechnik Band 4: Fertigung und Montage*. Düsseldorf: VDI Verlag. ISBN: 9783642613449.

Falco, Joe, Karl Van Wyk and Kenneth Kimble (2021). “Advances in Robot Technology Supporting Low-Volume/High-Mix Small Part Assembly Operations”. In: pp. 215–238. DOI: 10.1142/9789811222849_0008.

Fei-Fei, Li and Jia Deng (2017). *ImageNet Large Scale Visual Recognition Challenge*. Ed. by Stanford Vision Lab.

Feldmann, Klaus (2013). *Einführung*. Ed. by Klaus Feldmann, Volker Schöppner and Günter Spur. München: Carl Hanser Verlag GmbH & Co. KG. ISBN: 978-3-446-42827-0. DOI: 10.3139/9783446436565.

Feurer, Matthias and Frank Hutter (2019). "Hyperparameter Optimization". In: pp. 3–33. DOI: 10.1007/978-3-030-05318-5_1.

Fischer, Christian (2013). "Systematische Planung". In: *Handbuch Fügen, Handhaben und Montieren*. Ed. by Klaus Feldmann, Volker Schöppner and Günter Spur. München: Carl Hanser Verlag GmbH & Co. KG. Chap. 5.2.1.2, pp. 603–606. ISBN: 978-3-446-42827-0. DOI: 10.3139/9783446436565.

Flannigan, William C. et al. (2014). "Technologies guiding the future of robotics in manufacturing". In: *Proceedings of the 24th International Conference on Flexible Automation & Intelligent Manufacturing*. DEStech Publications, Inc., pp. 109–115. ISBN: 9781605951737. DOI: 10.14809/faim.2014.0109.

Franzkowiak, M, M Zäh and J Fleischer (2014). *Methodik zur Strukturierung von Vorrichtungssystemen in der Lohnfertigung*. Universitätsbibliothek der TU München.

Fu, Cheng-Yang et al. (2017). "DSSD : Deconvolutional Single Shot Detector". In.

Fukushima, Kunihiko (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4, pp. 193–202. ISSN: 0340-1200. DOI: 10.1007/BF00344251.

Gamal, Mohamed et al. (2021). "Anomalies Detection in Smart Manufacturing Using Machine Learning and Deep Learning Algorithms". In: *International Conference on Industrial Engineering and Operations Management*, pp. 1611–1622.

Gan, Zhi Lon, Siti Nurmaya Musa and Hwa Jen Yap (2023). "A Review of the High-Mix, Low-Volume Manufacturing Industry". In: *Applied Sciences* 13.3, p. 1687. ISSN: 2076-3417. DOI: 10.3390/app13031687.

Garrido-Jurado, S. et al. (2014). "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47.6, pp. 2280–2292. ISSN: 00313203. DOI: 10.1016/j.patcog.2014.01.005.

Gauthier, Nicolas et al. (2021). “Towards a Programming-Free Robotic System for Assembly Tasks Using Intuitive Interactions”. In: pp. 203–215. DOI: 10.1007/978-3-030-90525-5_18.

Ge, Zheng et al. (2021). *YOLOX: Exceeding YOLO Series in 2021.* URL: http://arxiv.org/abs/2107.08430 (visited on 26/01/2023).

Ghosh, Anirudha et al. (2020). “Fundamental Concepts of Convolutional Neural Network”. In: *Recent Trends and Advances in Artificial Intelligence and Internet of Things. Intelligent Systems Reference Library.* Ed. by V. Balas, R. Kumar and R. Srivastava. Cham: Springer, pp. 519–567. DOI: 10.1007/978-3-030-32644-9_36.

Girshick, Ross (2015). “Fast R-CNN”. In: *2015 IEEE International Conference on Computer Vision (ICCV).* IEEE, pp. 1440–1448. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.169.

Girshick, Ross et al. (2014). “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Glassner, Andrew (2021). *Deep learning : a visual approach.* San Francisco: No Starch Press. ISBN: 9781718500723.

Gonzalez, Rafael C. and Richard E Woods (2018). *Digital Image Processing.* 4. London: Pearson Education. ISBN: 987-0-13-335672-4.

Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). *Deep Learning.* MIT Press.

Grube Hansen, David, Ali Ahmad Malik and Arne Bilberg (2017). “Generic Challenges and Automation Solutions in Manufacturing SMEs”. In: pp. 1161–1169. DOI: 10.2507/28th.daaam.proceedings.161.

Gupta, Aman et al. (2021). “Adam vs. SGD: Closing the generalization gap on image classification”. In: *OPT2021: 13th Annual Workshop on Optimization for Machine Learning.* Virtual.

Harakannanavar, Sunil Swamilingappa, Prashanth Chikkanayakanahalli Renukamurthy and Kori Basava Raja (2019). “Comprehensive Study of Biometric Authentication Systems, Challenges and Future Trends”. In: *International Journal of Advanced Networking and Applications* 10.4, pp. 3958–3968. ISSN: 09750290. DOI: 10.35444/ijana.2019.10048.

He, Kaiming et al. (2016). “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2016-Decem. IEEE, pp. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90.

He, Kaiming et al. (2017). “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, pp. 2980–2988. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.322.

Hermawati, Fajar Astuti, Handayani Tjandrasa and Nanik Suciati (2018). “Combination of Aggregated Channel Features (ACF) detector and Faster R-CNN to improve object detection performance in fetal ultrasound images”. In: *International Journal of Intelligent Engineering and Systems* 11.6, pp. 65–74. ISSN: 21853118. DOI: 10.22266/ijies2018.1231.07.

Herps, Koen et al. (2022). “A simulation-based approach to design an automated high-mix low-volume manufacturing system”. In: *Journal of Manufacturing Systems* 64, pp. 1–18. ISSN: 02786125. DOI: 10.1016/j.jmsy.2022.05.013.

Hesse, Stefan (2000). *Fertigungsautomatisierung: Automatisierungsmittel, Gestaltung und Funktion*. Vieweg und Teubner Verlag. ISBN: 9783528039141.

– (2012). “Automatische Montagemaschinen”. In: *Montage in der industriellen Produktion*. Ed. by Bruno Lotter and Hans-Peter Wiendahl. 2. Auflage. Berlin, Heidelberg: Springer Verlag Berlin Heidelberg. Chap. 8, pp. 195–272. ISBN: 978-3-642-29060-2.

Hoshino, S., H. Seki and Y. Naka (2008). “Development of a flexible and agile multi-robot manufacturing system”. In: *IFAC Proceedings Volumes*. DOI: 10.3182/20080706-5-KR-1001.0157.

Hridoy, Monowar Wadud, Mohammad Mizanur Rahman and Saadman Sakib (2022). “A Framework for Industrial Inspection System using Deep Learning”. In: *Annals of Data Science*. ISSN: 2198-5804. DOI: 10.1007/s40745-022-00437-1.

Hu, Dunli et al. (2021). “Detection of material on a tray in automatic assembly line based on convolutional neural network”. In: *IET Image Processing* 15.13, pp. 3400–3409. ISSN: 1751-9659. DOI: 10.1049/ipr2.12302.

Hu, Jie et al. (2020). “Robotic deburring and chamfering of complex geometries in high-mix/low-volume production applications”. In: *2020 IEEE 16th International Conference* *on Automation Science and Engineering (CASE)*. IEEE, pp. 1155–1160. ISBN: 978-1-7281-6904-0. DOI: 10.1109/CASE48305.2020.9217042.

Hu, S. Jack (2013). “Evolving Paradigms of Manufacturing: From Mass Production to Mass Customization and Personalization”. In: *Procedia CIRP* 7, pp. 3–8. ISSN: 22128271. DOI: 10.1016/j.procir.2013.05.002.

Huang, M.Q., J. Ninić and Q.B. Zhang (2021). “BIM, machine learning and computer vision techniques in underground construction: Current status and future perspectives”. In: *Tunnelling and Underground Space Technology* 108.October, p. 103677. ISSN: 08867798. DOI: 10.1016/j.tust.2020.103677.

Illingworth, J. and J. Kittler (1987). “The Adaptive Hough Transform”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9.5, pp. 690–698. ISSN: 01628828. DOI: 10.1109/TPAMI.1987.4767964.

Irwansyah, Arif et al. (2015). “FPGA-based circular hough transform with graph clustering for vision-based multi-robot tracking”. In: *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. Vol. IEEE. IEEE, pp. 1–8. ISBN: 978-1-4673-9406-2. DOI: 10.1109/ReConFig.2015.7393313.

Ishiyama, Shin and Huimin Lu (2022). “3D object recognition for coordination-less bin-picking automation”. In: *International Symposium on Artificial Intelligence and Robotics 2022*. Ed. by Huimin Lu, Jintong Cai and Yuchao Zheng. SPIE, p. 36. ISBN: 9781510661288. DOI: 10.1117/12.2663323.

Jacques, Harald and Ralph Hansen (2010). “Komponenten”. In: *Taschenbuch der Automatisierung*. Ed. by Reinhard Langmann. 2nd ed. München: Fachbuchverlag Leipzig im Carl Hanser Verlag. Chap. 3, pp. 103–201. ISBN: 978-3-446-42112-7.

Jähne, Bernd (2002). *Digitale Bildverarbeitung*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-06732-1. DOI: 10.1007/978-3-662-06731-4.

Jelali, Mohieddine, ed. (2013). *Prozessautomatisierungstechnik*. URL: https://www.uni-due.de/imperia/md/content/srs/lehrangebot/akt-veranstaltungen/v-prozauto/skript/pat_teil_a_due.pdf (visited on 26/01/2023).

Jeschke, Sabina (2015). *Die Endmontage verlangt eine intelligente Automatisierung.* URL: https://www.maschinenmarkt.vogel.de/die-endmontage-verlangt-eine-intelligente-automatisierung-a-500458/ (visited on 17/09/2020).

Jiao, Licheng et al. (2019). "A Survey of Deep Learning-Based Object Detection". In: *IEEE Access* 7, pp. 128837–128868. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2939201.

Jocher, Glenn and Ayush Chaurasia (2023). *Ultralytics YOLOv8.* URL: https://github.com/ultralytics/ultralytics (visited on 27/01/2023).

Jocher, Glenn et al. (2022). "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation". In: DOI: 10.5281/ZENODO.7347926.

Johansen, Kerstin, Milad Ashourpour and Sagar Rao (2021a). "Positioning sustainable automation in production of customized products". In: *Procedia Manufacturing* 55, pp. 358–364. ISSN: 23519789. DOI: 10.1016/j.promfg.2021.10.050.

Johansen, Kerstin, Sagar Rao and Milad Ashourpour (2021b). "The Role of Automation in Complexities of High-Mix in Low-Volume Production – A Literature Review". In: *Procedia CIRP* 104, pp. 1452–1457. ISSN: 2212-8271. DOI: 10.1016/j.procir.2021.11.245.

Joshi, Shashidhar Ram and Roshan Koju (2012). "Study and comparison of edge detection algorithms". In: *2012 Third Asian Himalayas International Conference on Internet.* IEEE, pp. 1–5. ISBN: 978-1-4673-2590-5. DOI: 10.1109/AHICI.2012.6408439.

Karaulova, Tatyana et al. (2019). "Lean Automation for Low-Volume Manufacturing Environment". In: *Proceedings of the 30th DAAAM International Symposium.* Ed. by B. Katalinic. DAAAM International, pp. 0059–0068. DOI: 10.2507/30th.daaam.proceedings.008.

Karmakar P., Teng S. Zhang D. Liu Y. & Lu G (2017). "Improved Tamura Features for Image Classification Using Kernel Based Descriptors". In: *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–7.

Kinkel, Steffen (2009). "Potenziale der industriellen Automatisierung". In: *VDI/ISI-Pressekonferenz AUTOMATION*, pp. 0–9.

Kito, Kazumasa et al. (2017). "A robot controller for a working cell". In: *2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS).* IEEE, pp. 1–7. ISBN: 978-1-5386-3315-1. DOI: 10.1109/MHS.2017.8305164.

Kleindienst, Mario and Christian Ramsauer (2015). "Der Beitrag von Lernfabriken zu Industrie 4.0-Ein Baustein zur vierten industriellen Revolution bei kleinen und mittelständischen Unternehmen". In: *Industrie-Management* 3, pp. 41–44.

Kluge, Stefan Jens (2011). "Methodik zur fähigkeitsbasierten Planung modularer Montagesysteme". PhD thesis. Universität Stuttgart, p. 240. ISBN: 9783939890812.

Ko, Wilson K.H., Yan Wu and Keng Peng Tee (2016). "LAP: A Human-in-the-loop Adaptation Approach for Industrial Robots". In: *Proceedings of the Fourth International Conference on Human Agent Interaction*. New York, NY, USA: ACM, pp. 313–319. ISBN: 9781450345088. DOI: 10.1145/2974804.2974805.

Kocsi, Balázs et al. (2020). "Real-Time Decision-Support System for High-Mix Low-Volume Production Scheduling in Industry 4.0". In: *Processes* 8.8, p. 912. ISSN: 2227-9717. DOI: 10.3390/pr8080912.

Koga, Yotto, Heather Kerrick and Sachin Chitta (2022). "On CAD Informed Adaptive Robotic Assembly". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 10207–10214. ISBN: 978-1-6654-7927-1. DOI: 10.1109/IROS47612.2022.9982242.

Kolla, Sri Sudha Vijay Keshav (2022). "A Holistic Methodology to Deploy Industry 4.0 in Manufacturing Enterprises". PhD thesis. University of Luxembourg.

Krautheim, T.B. et al. (1997). *Herstellungsverfahren, Gebrauchsanforderungen und Materialkennwerte räumlicher elektronischer Baugruppen 3D-MID*. Handbuch für Anwender und Hersteller. Erlangen: Forschungsvereinigung Räumliche Elektronische Baugruppen 3-D MID e.V.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.

Kumar T.K., Arun et al. (2019). "Convolutional Neural Networks for Fingerprint Liveness Detection System". In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, pp. 243–246. ISBN: 978-1-5386-8113-8. DOI: 10.1109/ICCS45141.2019.9065713.

Kusuda, Yoshihiro (2010). “IDEC’s robot-based cellular production system: a challenge to automate high-mix low-volume production”. In: *Assembly Automation* 30.4, pp. 306–312. ISSN: 0144-5154. DOI: 10.1108/01445151011075753.

Lager, Anders et al. (2019). “Towards Reactive Robot Applications in Dynamic Environments”. In: *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, pp. 1603–1606. ISBN: 978-1-7281-0303-7. DOI: 10.1109/ETFA.2019.8868963.

Lang, Xianli et al. (2022). “MR-YOLO: An Improved YOLOv5 Network for Detecting Magnetic Ring Surface Defects”. In: *Sensors* 22.24, p. 9897. ISSN: 1424-8220. DOI: 10.3390/s22249897.

LeCun, Y. et al. (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. ISSN: 00189219. DOI: 10.1109/5.726791.

LeCun, Yann, Yoshua Bengio and Geoffrey Hinton (2015). “Deep learning”. In: *Nature* 521.7553, pp. 436–444. ISSN: 0028-0836. DOI: 10.1038/nature14539.

Lee, Marshal (2018). *Tamura in Python*. URL: https://github.com/MarshalLeeeeee/Tamura-In-Python/tree/master/referenced-matlab-code (visited on 19/04/2023).

Li, Chuyi et al. (2022). “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications”. In: DOI: 10.48550/arxiv.2209.02976.

Li, Jiangyun et al. (2018). “Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network”. In: *IFAC-PapersOnLine* 51.21, pp. 76–81. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.09.412.

Li, Zuoxin and Fuqiang Zhou (2017). “FSSD: feature fusion single shot multibox detector”. In: *arXiv preprint arXiv:1712.00960*.

Lin, Jia et al. (2020). “Optimization of Multi-objective Function of n-step Hybrid Flowshop Scheduling”. In: *2020 International Symposium on Semiconductor Manufacturing (ISSM)*. IEEE, pp. 1–4. ISBN: 978-1-6654-0364-1. DOI: 10.1109/ISSM51728.2020.9377505.

Lin, Tsung-Yi et al. (2014). *Microsoft COCO: Common Objects in Context*.

Liu, Wei et al. (2016). “SSD: Single Shot MultiBox Detector”. In: pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2.

Long, Xiang et al. (2020). "PP-YOLO: An Effective and Efficient Implementation of Object Detector". In.

Lotter, Bruno (2012a). "Flexible Montage mit Robotereinsatz". In: *Montage in der industriellen Produktion.* Ed. by Bruno Lotter and Hans-Peter Wiendahl. 2. Auflage. Berlin, Heidelberg: Springer Verlag Berlin Heidelberg. Chap. 9, pp. 273–284. ISBN: 978-3-642-29060-2.

– (2012b). *Montage in der industriellen Produktion.* Ed. by Bruno Lotter and Hans-Peter Wiendahl. 2. Auflage. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-29060-2. DOI: 10.1007/978-3-642-29061-9.

Lotter, Edwin (2012c). "Hybride Montagesysteme". In: *Montage in der industriellen Produktion.* Ed. by Bruno Lotter and Hans-Peter Wiendahl. 2. Auflage. Berlin, Heidelberg: Springer Verlag Berlin Heidelberg. Chap. 7, pp. 167–194. ISBN: 978-3-642-29060-2.

Lu, Yuqian, Xun Xu and Lihui Wang (2020). "Smart manufacturing process and system automation – A critical review of the standards and envisioned scenarios". In: *Journal of Manufacturing Systems* 56, pp. 312–325. ISSN: 02786125. DOI: 10.1016/j.jmsy.2020.06.010.

Ma, Haocheng and Lihui Peng (2019). "Vision Based Liquid Level Detection and Bubble Area Segmentation in Liquor Distillation". In: *2019 IEEE International Conference on Imaging Systems and Techniques (IST).* IEEE, pp. 1–6. ISBN: 978-1-7281-3868-8. DOI: 10.1109/IST48021.2019.9010097.

Malburg, Lukas et al. (2021). "Object Detection for Smart Factory Processes by Machine Learning". In: *Procedia Computer Science* 184, pp. 581–588. ISSN: 18770509. DOI: 10.1016/j.procs.2021.04.009.

Malik, Ali Ahmad and Arne Bilberg (2019). "Complexity-based task allocation in human-robot collaborative assembly". In: *Industrial Robot: the international journal of robotics research and application* 46.4, pp. 471–480. ISSN: 0143-991X. DOI: 10.1108/IR-11-2018-0231.

Mazzetto, Muriel et al. (2020). "Deep Learning Models for Visual Inspection on Automotive Assembling Line". In: *International Journal of Advanced Engineering Research and Science* 7.3, pp. 473–494. ISSN: 23496495. DOI: 10.22161/ijaers.74.56.

Meena, B., K. Venkata Rao and Suresh Chittineni (2020). “A survey on deep learning methods and tools in image processing”. In: *International Journal of Scientific and Technology Research* 9.2, pp. 1057–1062. ISSN: 22778616.

Mo, Zhimin, Liding Chen and Wenjing You (2019). “Identification and Detection of Automotive Door Panel Solder Joints based on YOLO”. In: *2019 Chinese Control And Decision Conference (CCDC)*. IEEE, pp. 5956–5960. ISBN: 978-1-7281-0106-4. DOI: 10.1109/CCDC.2019.8833257.

MTM (2011). *Basic MTM: MTM1 / UAS*. Ed. by Deutsche MTM-Vereinigung e.V. Zeuthen: MTM-Institut. ISBN: 3-978-3-9809466-4-3.

Mukhopadhyay, Priyanka and Bidyut B. Chaudhuri (2015). “A survey of Hough Transform”. In: *Pattern Recognition* 48.3, pp. 993–1010. ISSN: 00313203. DOI: 10.1016/j.patcog.2014.08.027.

Muller, Patrice et al. (2022). *Annual Report on European SMEs 2021/2022*. Luxembourg. DOI: 10.2826/50999.

Müller, R, M Vette-Steinkamp and A Kanso (2019). “Position and orientation calibration of a 2D laser line sensor using closed-form least-squares solution”. In: *IFAC-PapersOnLine* 52.13, pp. 689–694. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2019.11.136.

Müller, Rainer, Martin Florian Esser and Jan Eilers (2013). “Assembly Oriented Design Method for Reconfigurable Processes and Equipment”. In: *Future Trends in Production Engineering*. Ed. by Günther Schuh, Reimund Neugebauer and Eckart Uhlmann. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 251–257. DOI: 10.1007/978-3-642-24491-9_25.

Müller, Rainer et al. (2011). “Wandlungsfähigkeit in der Montage - Chance für eine schwer planbare Zukunft”. In: *Wettbewerbsfaktor Produktionstechnik*. Ed. by Christian Brecher et al. Aachen: Shaker Verlag. Chap. 4.3, pp. 423–448. ISBN: 978-3-8840-0087-0.

Nagashima, Hiroki and Seiichiro Katsura (2014). “Motion education system using impedance control based on spatial information”. In: *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*. IEEE, pp. 242–247. ISBN: 978-1-4799-2323-6. DOI: 10.1109/AMC.2014.6823289.

Nielsen, Michael A. (2015). *Neural Networks and Deep Learning.* Determination Press.

Nwankpa, Chigozie et al. (2021). “Achieving remanufacturing inspection using deep learning”. In: *Journal of Remanufacturing* 11.2, pp. 89–105. ISSN: 2210-464X. DOI: 10.1007/s13243-020-00093-9.

Nyhuis, Prof. Dr.-Ing. habil. Peter, Prof. Dr.-Ing. Gunther Reinhart and Prof. Dr.-Ing. Eberhard Abele (2008). *Wandlungsfähige Produktionssysteme - Heute die Industrie von morgen gestalten.* Garbsen: PZH Produktionstechnisches Zentrum, p. 166. ISBN: 9783939026969. DOI: https://doi.org/10.2314/GBV:633626406.

Ong, S.K. et al. (2020). “Augmented reality-assisted robot programming system for industrial applications”. In: *Robotics and Computer-Integrated Manufacturing* 61, p. 101820. ISSN: 07365845. DOI: 10.1016/j.rcim.2019.101820.

Onstein, Ingrid Fjordheim, Oleksandr Semeniuta and Magnus Bjerkeng (2020). “Deburring Using Robot Manipulators: A Review”. In: *2020 3rd International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS).* IEEE, pp. 1–7. ISBN: 978-1-7281-6419-9. DOI: 10.1109/SIMS49386.2020.9121490.

Otsu, Nobuyuki (1979). “A Threshold Selection Method from Gray-Level Histograms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1, pp. 62–66. ISSN: 0018-9472. DOI: 10.1109/TSMC.1979.4310076.

Pane, Yudha et al. (2020). “A System Architecture for CAD-Based Robotic Assembly With Sensor-Based Skills”. In: *IEEE Transactions on Automation Science and Engineering*, pp. 1–13. ISSN: 1545-5955. DOI: 10.1109/TASE.2020.2980628.

Pfeiffer, Rolf (2013). “Fügen durch Schrauben”. In: *Handbuch Fügen, Handhaben und Montieren.* Ed. by Klaus Feldmann, Volker Schöppner and Günter Spur. München: Carl Hanser Verlag GmbH & Co. KG, pp. 144–170. ISBN: 9783446428270. DOI: 10.3139/9783446436565.

Plapper, Peter et al. (2011). “Referenzsysteme für wandlungsfähige Produktion”. In: *Wettbewerbsfaktor Produktionstechnik: Aachener Perspektiven: Aachener Werkzeugmaschinen-kolloquium 2011.* Ed. by AWK Aachener Werkzeugmaschinen-Kolloquium et al. Aachen: Shaker Verlag, pp. 449–477. ISBN: 978-3-8440-0087-0.

Qinlan, Xie and Chen Hong (2009). “Image denoising based on adaptive and multi-frame averaging filtering”. In: *2009 International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009* 3, pp. 523–526. DOI: 10.1109/AICI.2009.490.

Razak, A. H.A. and R. H. Taharim (2009). “Implementing Gabor filter for fingerprint recognition using Verilog HDL”. In: *Proceedings of 2009 5th International Colloquium on Signal Processing and Its Applications, CSPA 2009*, pp. 423–427. DOI: 10.1109/CSPA.2009.5069264.

Redmon, Joseph and Ali Farhadi (2016). “YOLO9000: Better, Faster, Stronger”. In.

– (2018). “YOLOv3: An Incremental Improvement”. In.

Redmon, Joseph et al. (2016). “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2016-Decem. IEEE, pp. 779–788. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.91.

Ren, Shaoqing et al. (2017). “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6, pp. 1137–1149. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2577031.

Ren, Zhonghe et al. (2022). “State of the Art in Defect Detection Based on Machine Vision”. In: *International Journal of Precision Engineering and Manufacturing-Green Technology* 9.2, pp. 661–691. ISSN: 2288-6206. DOI: 10.1007/s40684-021-00343-6.

Rumelhart, David E., Geoffrey E. Hinton and Ronald J. Williams (1986). “Learning representations by back-propagating errors”. In: *Nature* 323.6088, pp. 533–536. ISSN: 0028-0836. DOI: 10.1038/323533a0.

Scherer, Dominik, Andreas Müller and Sven Behnke (2010). “Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition”. In: pp. 92–101. DOI: 10.1007/978-3-642-15825-4_10.

Schlette, C. et al. (2019). “Towards robot cell matrices for agile production – SDU Robotics’ assembly cell at the WRC 2018”. In: *Advanced Robotics*, pp. 1–17. ISSN: 0169-1864. DOI: 10.1080/01691864.2019.1686422.

Shao, Quanquan et al. (2020). “Location Instruction-Based Motion Generation for Sequential Robotic Manipulation”. In: *IEEE Access* 8, pp. 26094–26106. DOI: 10.1109/ACCESS.2020.2971570.

Sharifi, M., M. Fathy and M.T. Mahmoudi (2002). “A classified and comparative study of edge detection algorithms”. In: *Proceedings. International Conference on Information Technology: Coding and Computing.* IEEE Comput. Soc, pp. 117–120. ISBN: 0-7695-1506-1. DOI: 10.1109/ITCC.2002.1000371.

Sharma, Purvesh and Damian Valles (2020). “Deep Convolutional Neural Network Design Approach for 3D Object Detection for Robotic Grasping”. In: *2020 10th Annual Computing and Communication Workshop and Conference (CCWC).* IEEE, pp. 0311–0316. ISBN: 978-1-7281-3783-4. DOI: 10.1109/CCWC47524.2020.9031186.

Shen, Day Fann, Chui Wen Chiu and Pan Jay Huang (2006). “Modified Laplacian Filter and intensity correction technique for image resolution enhancement”. In: *2006 IEEE International Conference on Multimedia and Expo, ICME 2006 - Proceedings* 2006, pp. 457–460. DOI: 10.1109/ICME.2006.262571.

Simeth, Alexej, Atal Anil Kumar and Peter Plapper (2022). “Using Artificial Intelligence to Facilitate Assembly Automation in High-Mix Low-Volume Production Scenario”. In: *Procedia CIRP* 107, pp. 1029–1034. ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2022.05.103.

Simeth, Alexej and Peter Plapper (2023). “Artificial Intelligence Based Robotic Automation of Manual Assembly Tasks for Intelligent Manufacturing”. In: *Smart, Sustainable Manufacturing in an Ever-Changing World.* Ed. by Konrad von Leipzig and Vera Hummel. Stellenbosch: Springer Nature. Chap. 11, pp. 137–148. DOI: 10.1007/978-3-031-15602-1_11.

Simeth, Alexej, Jessica Plaßmann and Peter Plapper (2021). “Detection of Fluid Level in Bores for Batch Size One Assembly Automation Using Convolutional Neural Network”. In: *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems. IFIP International Federation for Information Processing. APMS 2021, IFIP AICT 632.* Ed. by A. Dolgui et al. Cham: Springer International Publishing. Chap. IFIP AICT, pp. 86–93. DOI: 10.1007/978-3-030-85906-0_10.

Singh, Swarit Anand and K. A. Desai (2023). “Automated surface defect detection framework using machine vision and convolutional neural networks”. In: *Journal of Intelligent Manufacturing* 34.4, pp. 1995–2011. ISSN: 0956-5515. DOI: 10.1007/s10845-021-01878-w.

Solomon, Chris and Toby Breckon (2011). *Fundamentals of digital image processing.* first publ. Wiley Blackwell. ISBN: 978 0 470 84472 4 (hardback).

Sorensen, Lars Caroe et al. (2018). “Automatic parameter learning for easy instruction of industrial collaborative robots”. In: *2018 IEEE International Conference on Industrial Technology (ICIT).* IEEE, pp. 87–92. ISBN: 978-1-5090-5949-2. DOI: 10.1109/ICIT.2018.8352157.

Stanevski, Nikolay and Dimiter Tsvetkov (2005). “Using Support Vector Machine as a Binary Classifier”. In: *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech’ 05)* 2, pp. 1–5.

Steinbauer, Carola Maria Theres (2012). “Modell zur Konfiguration der Kleinserienmontage”. PhD thesis. Technischen Universität München.

Süße, Herbert and Erik Rodner (2014). *Bildverarbeitung und Objekterkennung.* Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-8348-2605-3. DOI: 10.1007/978-3-8348-2606-0.

Szeliski, Richard (2022). *Computer Vision.* Texts in Computer Science. Cham: Springer International Publishing. ISBN: 978-3-030-34371-2. DOI: 10.1007/978-3-030-34372-9.

Tahmina, Tanjida et al. (2023). “A Survey of Smart Manufacturing for High-Mix Low-Volume Production in Defense and Aerospace Industries”. In: pp. 237–245. DOI: 10.1007/978-3-031-18326-3_24.

Tamura, Hideyuki, Shunji Mori and Takashi Yamawaki (1978). “Textural Features Corresponding to Visual Perception”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 8.6, pp. 460–473. ISSN: 0018-9472. DOI: 10.1109/TSMC.1978.4309999.

Tan, Chuanqi et al. (2018). “A Survey on Deep Transfer Learning”. In: *InInternational conference on artificial neural networks*, pp. 270–279.

Telgen, Daniël et al. (2014). “Hierarchical management of a heterarchical manufacturing grid”. In: *Proceedings of the 24th International Conference on Flexible Automation &*

Intelligent Manufacturing. DEStech Publications, Inc., pp. 825–832. ISBN: 9781605951737. DOI: 10.14809/faim.2014.0825.

Thanh, Le Thi et al. (2019). “Single image dehazing based on adaptive histogram equalization and linearization of gamma correction”. In: *Proceedings of 2019 25th Asia-Pacific Conference on Communications, APCC 2019*, pp. 36–40. DOI: 10.1109/APCC47188.2019.9026457.

The MathWorks (2021). *Pretrained Deep Neural Networks*. URL: https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html (visited on 08/04/2021).

– (2023a). *templateLinear Linear classification learner template*. URL: https://de.mathworks.com/help/stats/templatelinear.html (visited on 20/04/2023).

– (2023b). *What is Deep Learning?* URL: https://nl.mathworks.com/discovery/deep-learning.html (visited on 03/02/2023).

Ting, Kai Ming (2016). “Confusion Matrix”. In: *Encyclopedia of Machine Learning and Data Mining*. Boston, MA: Springer US, pp. 1–1. DOI: 10.1007/978-1-4899-7502-7_50-1.

Tkachenko, Maxim et al. (2022). *Label Studio: Data labeling software*. URL: https://github.com/heartexlabs/label-studio (visited on 26/01/2023).

Ulrich, Hans (1984). *Management*. Ed. by Thomas Dyllick and Glibert J. B. Probst. Bern: Paul Haupt Berne. ISBN: 9783258034461.

Vajna, Sandor et al. (2009). *CAx für Ingenieure - Eine praxisbezogene Einführung*. Springer Verlag Berlin Heidelberg. ISBN: 9783540360384. DOI: 10.1007/978-3-540-36039-1.

VDI 2860 (1990). *VDI-Richtlinie 2860: Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole*. Ed. by VDI Verein Deutscher Ingenieure e.V. Düsseldorf.

Vu, Thi-Thu-Huyen, Dinh-Lam Pham and Tai-Woo Chang (2023). “A YOLO-based Real-time Packaging Defect Detection System”. In: *Procedia Computer Science* 217, pp. 886–894. ISSN: 18770509. DOI: 10.1016/j.procs.2022.12.285.

Wang, Chien-Yao, Alexey Bochkovskiy and Hong-Yuan Mark Liao (2022). "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". In: DOI: 10.48550/arxiv.2207.02696.

Wang, Chien-Yao, I-Hau Yeh and Hong-Yuan Mark Liao (2021). "You Only Learn One Representation: Unified Network for Multiple Tasks". In.

Wang, Yan, Kensuke Harada and Weiwei Wan (2020). "Motion planning of skillful motions in assembly process through human demonstration". In: *Advanced Robotics* 34.16, pp. 1079–1093. ISSN: 0169-1864. DOI: 10.1080/01691864.2020.1782260.

Weng, Ching-Yen et al. (2020). "A Telemanipulation-Based Human–Robot Collaboration Method to Teach Aerospace Masking Skills". In: *IEEE Transactions on Industrial Informatics* 16.5, pp. 3076–3084. ISSN: 1551-3203. DOI: 10.1109/TII.2019.2906063.

Wiendahl, Hans-Peter, Peter Nyhuis and Jürgen Reichardt (2009). *Handbuch Fabrikplanung*. Carl Hanser Verlag GmbH Co KG. ISBN: 9783446224773.

Wu, Yuxin et al. (2019). *Detectron2*. URL: https://github.com/facebookresearch/detectron2 (visited on 01/05/2023).

Xie, Zhen, Josh Chen Ye Seng and Guowei Lim (2022). "AI-Enabled Soft Versatile Grasping for High-Mixed-Low-Volume Applications with Tactile Feedbacks". In: *2022 27th International Conference on Automation and Computing (ICAC)*. IEEE, pp. 1–6. ISBN: 978-1-6654-9807-4. DOI: 10.1109/ICAC55051.2022.9911143.

Yi, Jingru, Pengxiang Wu and Dimitris N Metaxas (2019). "ASSD: Attentive single shot multibox detector". In: *Computer Vision and Image Understanding* 189, p. 102827.

Ying, Xue (2019). "An Overview of Overfitting and its Solutions". In: *Journal of Physics: Conference Series* 1168, p. 022022. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1168/2/022022.

Yuan, Qilong et al. (2020). "Flexible telemanipulation based handy robot teaching on tape masking with complex geometry". In: *Robotics and Computer-Integrated Manufacturing* 66, p. 101990. ISSN: 07365845. DOI: 10.1016/j.rcim.2020.101990.

Zancul, Eduardo et al. (2020). "Machine Vision applications in a Learning Factory". In: *Procedia Manufacturing* 45, pp. 516–521. ISSN: 23519789. DOI: 10.1016/j.promfg.2020.04.069.

Zhang, Yunsheng, Jason Kung and Viraj Kadam (2020). *Defects location for metal surface.* URL: https://www.kaggle.com/datasets/zhangyunsheng/defects-class-and-location (visited on 01/05/2023).

Zhang, Zhengyou (2000). "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11, pp. 1330–1334. ISSN: 01628828. DOI: 10.1109/34.888718.

Zhao, Zhong-Qiu et al. (2019). "Object detection with deep learning: A review". In: *IEEE transactions on neural networks and learning systems* 30.11, pp. 3212–3232.

Zheng, Xiaoqing et al. (2021a). "A deep learning-based approach for the automated surface inspection of copper clad laminate images". In: *Applied Intelligence* 51.3, pp. 1262–1279. ISSN: 0924-669X. DOI: 10.1007/s10489-020-01877-z.

Zheng, Xiaoqing et al. (2021b). "Recent advances in surface defect inspection of industrial products using deep learning techniques". In: *The International Journal of Advanced Manufacturing Technology* 113.1-2, pp. 35–58. ISSN: 0268-3768. DOI: 10.1007/s00170-021-06592-8.

Zhou, Tong et al. (2021). "Multi-agent reinforcement learning for online scheduling in smart factories". In: *Robotics and Computer-Integrated Manufacturing* 72, p. 102202. ISSN: 07365845. DOI: 10.1016/j.rcim.2021.102202.

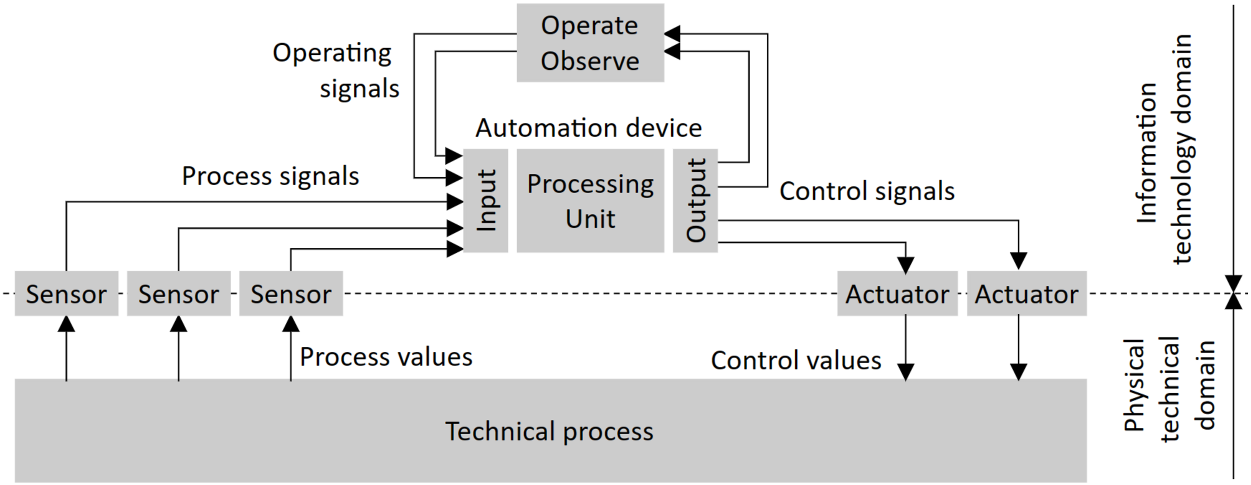
Zhu, Qing Hua et al. (2017). "Scheduling Transient Processes for Time-Constrained Single-Arm Robotic Multi-Cluster Tools". In: *IEEE Transactions on Semiconductor Manufacturing* 30.3, pp. 261–269. ISSN: 0894-6507. DOI: 10.1109/TSM.2017.2721970.

APPENDIX

Appendix A - Additional Material

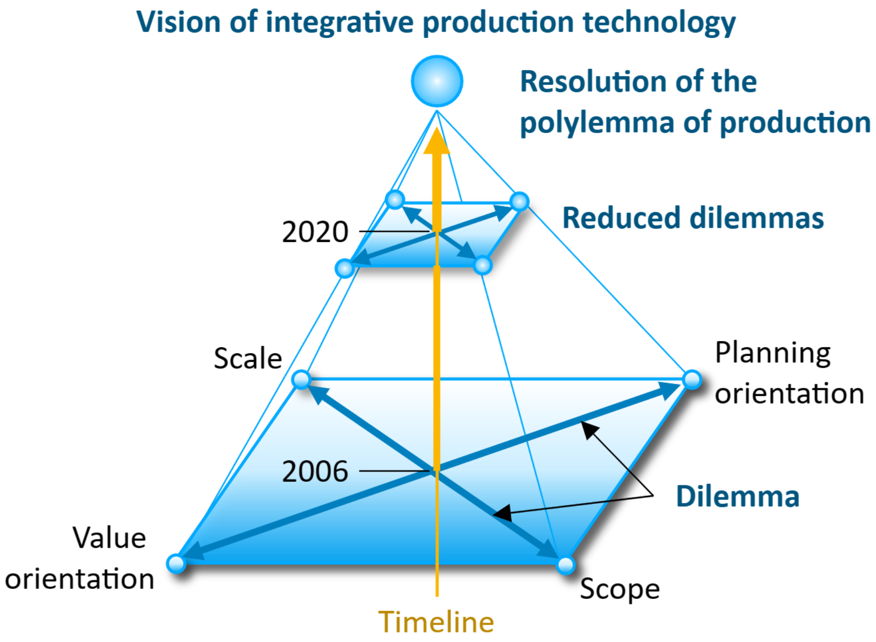
A1 - Additional Material on Chapter 2

A1.1 - Schematic illustration of an automation system



Schematic illustration of an automation system based on (Jacques and Hansen, 2010, p. 155).

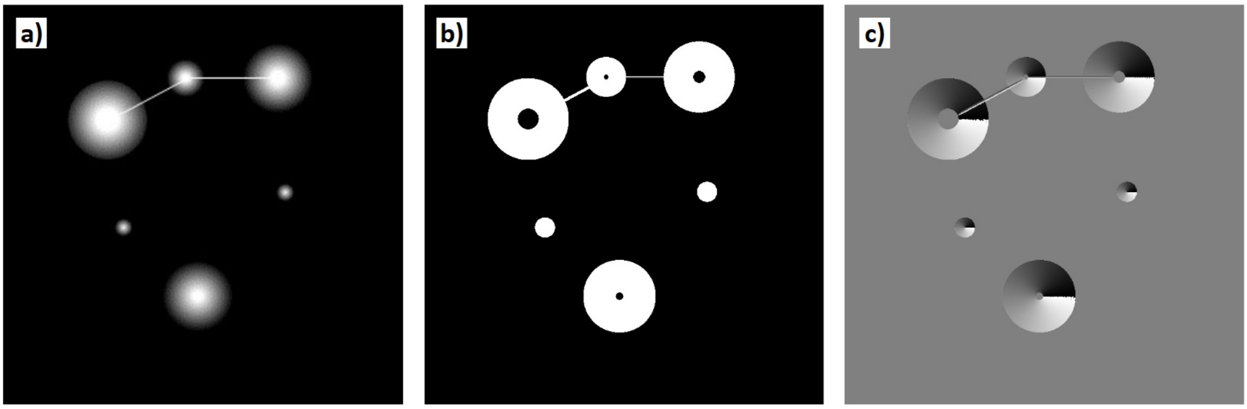
A1.2 - Polylemma of production



Polylemma of production (Brecher et al., 2011, p. 22).

The figure above illustrates the conflicting priorities of production and planning efficiency, namely the dichotomy between economies of scale vs economies of scope and the dichotomy between planning and value orientation. Standardisation of processes in production can increase efficiency and lower the cost per unit by creating scale effects, i.e., economies of scale. However, the advantageous scale effects usually come at the cost of limited adaptability of the production system to changing conditions. On the contrary, focusing on economies of scope leads to high adaptivity, i.e., keeping a high degree of freedom for produced goods in business processes and technical systems. This, however, implies higher unit costs compared to scale-optimised production systems. Similarly, planning and value orientation are usually contradictory in nature. The vision of integrative production technology is to resolve the polylemma of production by means of *high integrativity*. This entails the combination of research methodologies from diverse scientific disciplines to establish a holistic approach to address the complex challenges associated with the production polylemma. (Brecher et al., 2011, pp. 21–23)

A1.3 - Gradient edge detection



a) Original image. b) Gradient magnitude. c) Gradient direction.

The gradient of a point (x, y) regarding an image $f(x, y)$ is defined as (Chaple et al., 2015, Joshi and Koju, 2012):

$$\nabla f\left(x, y\right) = \text{grad}\left[f\left(x, y\right)\right] = \begin{bmatrix} g_x(x, y) \\ g_y(x, y) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

The Gradient Magnitude $M(x, y)$ at a point (x, y) is calculated with the Euclidean vector norm:

$$G(x, y) = \|\nabla f\left(x, y\right)\| = \sqrt{g_x^2(x, y) + g_y^2(x, y)}$$

The direction θ of the gradient at a point (x, y) is calculated with:

$$\theta(x, y) = \tan^{-1}\left[\frac{g_y(x, y)}{g_x(x, y)}\right]$$

The figure above provides an example of Gradient edge detection. Image a) shows the original grayscale input image. The Gradient Magnitude G is shown in image b). The edge direction is visualised in image c).

A1.4 - Circular Hough transform

The Circular Hough transform (CHT) is a modification of the Hough transform used specifically for circle detection. It involves three parameters (a, b, r) to define a circle, where (a, b) represent the circle's centre coordinates, and r is the radius. Determining a circle in the Euclidean space can be achieved using these parameters (Ballard et al., 2016):

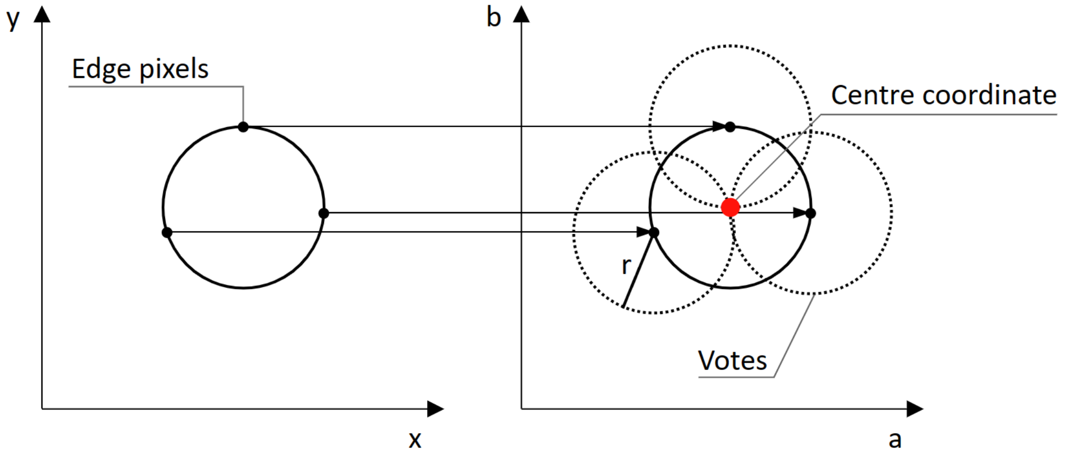
$$r^2 = (x - a)^2 + (y - b)^2$$

Here, (x, y) represents the edge pixels on the circle's circumference. Similar to the linear Hough transform, the common approach to observing the accumulator in Hough space involves using trigonometric equations for the circle:

$$x = a + r\cos(\theta)$$

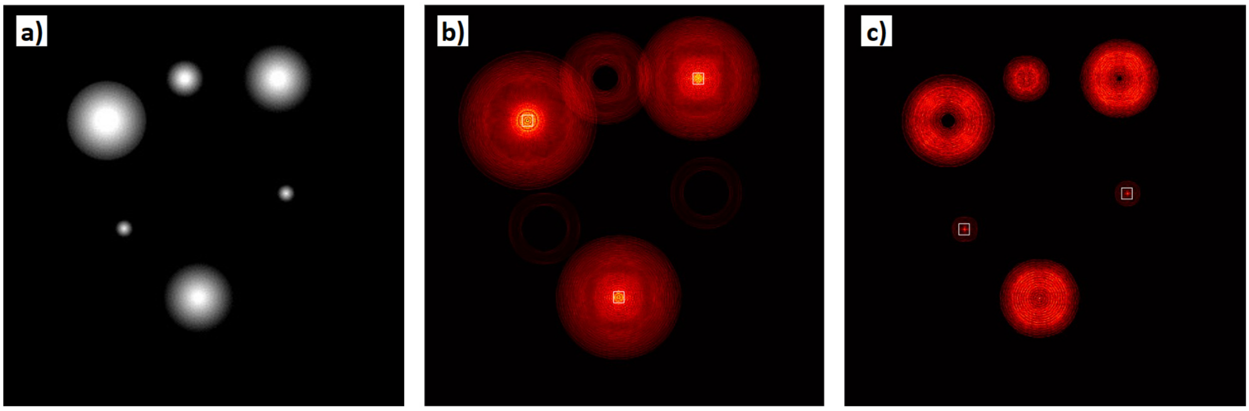
$$y = b + r\sin(\theta)$$

As the circle is defined by the three parameters (a, b, r), the parameter space belongs to R^3. The variable θ ranges from $[0, 360°]$, representing an angle. To simplify the CHT algorithms, the radius is often set to a constant value or a range denoted as (R_{min}, R_{max}).



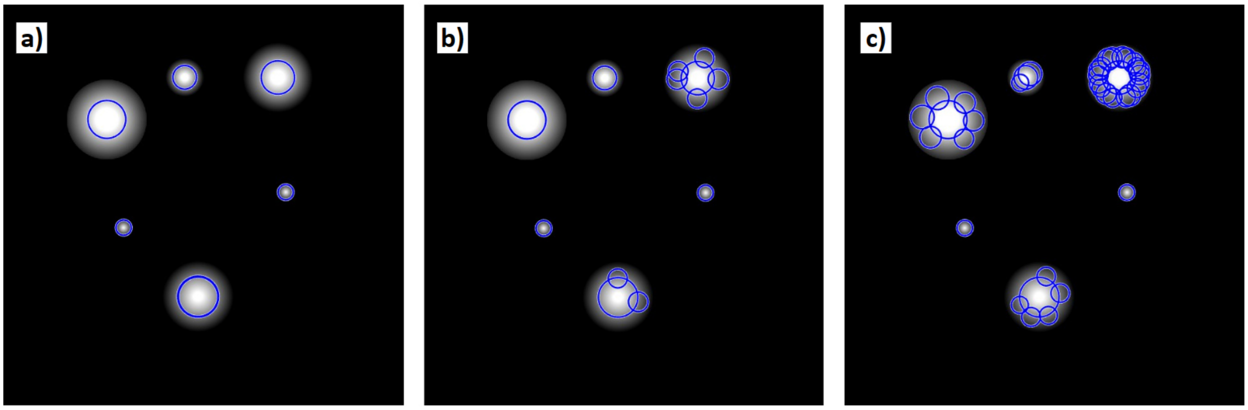
Conversion during CHT from x,y-space (left) to Hough space (right) with the parameters (a, b, r) for a constant radius based on (Irwansyah et al., 2015).

Since three parameters describe circles, each parameter constitutes a search dimension in the Hough space, leading to a 3D accumulator. The figure above illustrates the accumulator for a fixed value of r, where each point in the Euclidean space transforms into a circle in the Hough space. The circle with the maximum number of intersections in the Hough space corresponds to a detected circle in the parameter space (Ballard et al., 2016).



a) Original image. b-c) Accumulator votes for a fixed radius of 70 pixels (b) and 15 pixels (c) with peaks highlighted with a white box.

The figure above shows examples of the CHT accumulator applied on image a). Images b-c) depict the votes of the CHT for the fixed radii of 70 pixels and 15 pixels. The common intersection points form the peaks of the votes and are highlighted with white boxes. The search with a radius of 70 finds the three large circles, while the small circles are found with a radius of 15. The circle in the top middle is not detected.



CHT examples with different sensitivities: a) 0.90; b) 0.925; c) 0.95.

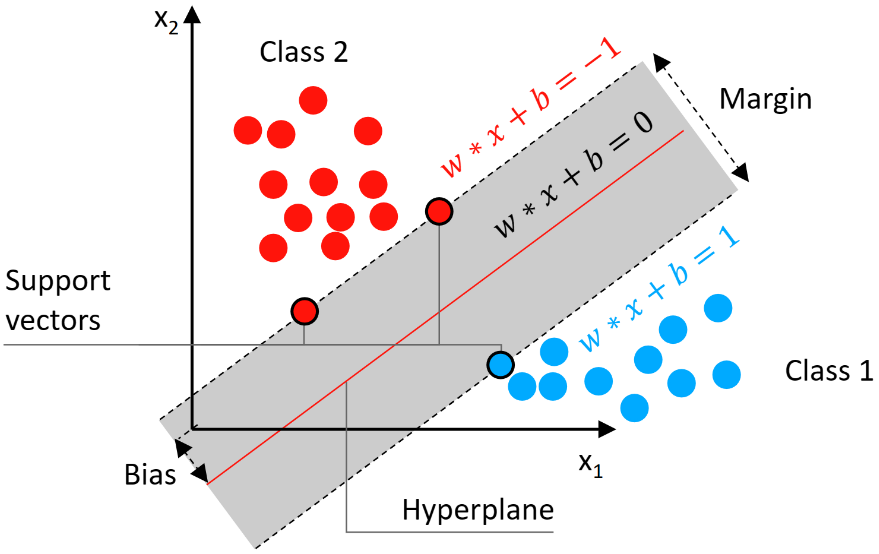
CHT implementations feature commonly tuning parameters such as the *Sensitivity* in MATLAB. This parameter defines the minimum votes required for acceptance of a circle. The figure above shows examples of CHT outputs with the variation of the sensitivity. In this case, a higher value reduces the required amount of votes.

A1.5 - Support Vector Machine

The SVM is a maximum margin classifier that tries to find the optimal hyperplane that best separates different classes in a dataset. Therefore, the hyperplane is chosen to maximise the distance between the hyperplane and the closest data points of each class, i.e., the support vectors. A hyperplane in an n-dimensional space is represented as a linear equation $w \cdot x + b = 0$, with x as an n-dimensional feature vector, w as weights and b as the bias term. The sign of the decision function is applied to classify any new data point:

$$y(x) = \begin{cases} +1, & \text{if } w \cdot x + b \geq 0 \\ -1, & \text{if } w \cdot x + b < 0 \end{cases}$$

Many hyperplanes are possible for a classification problem, and the target of the SVM training is finding the hyperplane separating the data best. An example of a two-dimensional SVM is given in the following figure.



Support Vector Machine Classification based on (Szeliski, 2022, p. 251)

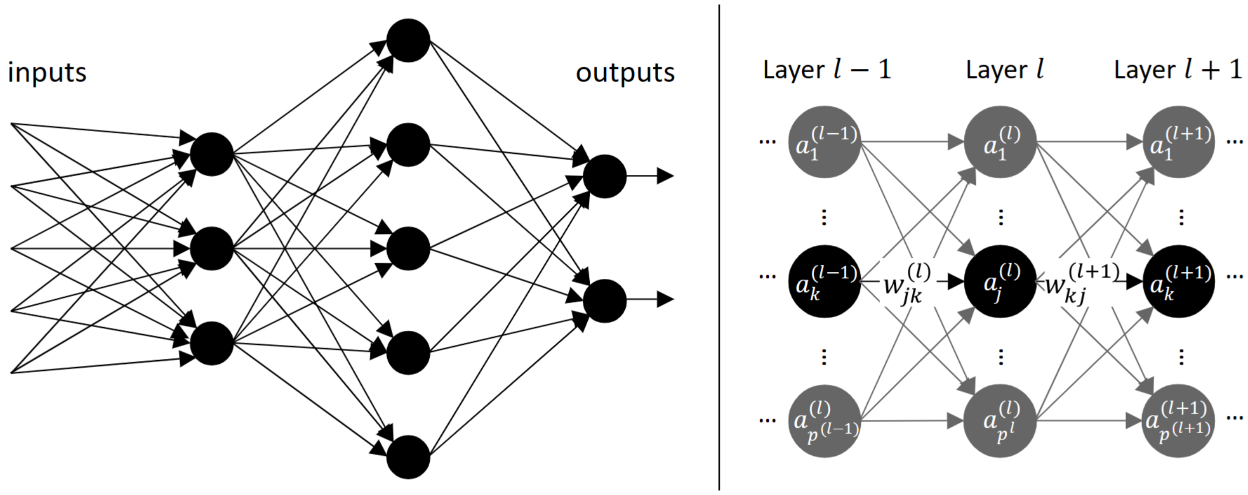
The optimisation problem is defined by maximising the margin. With the distance between the two hyperplanes defined by the support vectors $\frac{2}{||w||}$, the optimisation problem can be written as:

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$\text{such that} \qquad y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad \forall i \in \{1, \ldots, n\}$$

Source: Szeliski, 2022, pp. 250–254

A1.6 - Artificial and Convolutional Neural Networks



Left: Schematic of a Multi-layer Perceptron. Right: Weight w^l_{jk} of the k-th neuron in layer $l-1$ to the j-th neuron in the layer l and weight w^{l+1}_{kj} of the j-th neuron in layer l to the k-th neuron in the layer $l+1$.

A simple type of ANN and the fundamental architecture in DL is the multilayer perceptron (MLP). It consists of multiple layers of interconnected neurons or nodes and is designed to approximate complex non-linear functions. An example is given in the figure above. The basic math of an MLP involves two key components: the forward propagation and the back-propagation algorithm for training.

Forward propagation

The input of the MLP is fed through the layers of a neural network to generate the output. Each layer consists of nodes (neurons) that apply an activation function to the weighted sum of their inputs. The output is calculated as follows. For each neuron j in the layer l, calculate the weighted sum of the inputs z^l_j:

$$z^l_j = \sum_k w^l_{jk} \cdot a^{l-1}_k + b^l_j$$

where w^l_{jk} is the weight connecting the neuron k in layer $l-1$ to neuron j in layer l, a^{i-1}_k is the output of neuron k in layer $(l-1)$ and b^l_j is the bias of neuron j in layer l. Then, the output a^l_j of the neuron j is calculated by applying the activation function

$$a^l_j = f(z^l_j)$$

One of the common activation functions in CNN is the rectified linear unit, defined as $ReLU(x) = \max\{0, x\}$. To calculate the output of the final layer, a different activation function may be selected depending on the nature of the problem. Common activation functions are the sigmoid function for binary classification or the softmax function for multiclass classification.

Backward propagation

Training the model or learning means finding the right weights and biases. Therefore, the weights and biases of the neural network are updated during the training process. The target is to minimise the difference between the predicted output and the actual target value. During backpropagation, the gradients of the loss function with respect to the weights and biases are computed and used to update the parameters. This process is performed iteratively over multiple training examples until the model converges to a minimum error.

Let C be the loss function measuring the difference between the predicted output and target values. The output error δ_j^L for neuron j in the output layer (L) is given by:

$$\delta_j^L = \frac{\partial C}{\partial z_j^L}$$

where $\frac{\partial C}{\partial z_j^L}$ denotes the partial derivative of the loss function C regarding the weighted sum z_j^L. Next, the error is backpropagated to the previous layers. For each hidden layer l, the error δ_j^l for each neuron j in that layer is computed by:

$$\delta_j^l = \left(\sum_k \delta_k^{l+1} \cdot w_{kj}^{l+1} \right) \cdot f'(z_j^l)$$

where δ_k^{l+1} denotes the error of the neuron k in the next layer $(l+1)$, w_{kj}^{l+1} is the weight connecting neuron j in layer l to neuron k in layer $(l+1)$, and $f'(z_j^l)$ is the derivative of the activation function $f(z_j^l)$ regarding the weighted sum z_j^l.

The gradients of the loss function with respect to the weights and biases are used to update the parameters during the gradient descent optimisation. For each weight w_{jk}^l connecting neuron k in layer $(l-1)$ to neuron j in layer l, and bias b_j^l for neuron j in layer l, the gradients are computed as follows:

$$\text{For weights:} \quad \frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \cdot \delta^l_j$$

$$\text{For biases:} \quad \frac{\partial C}{\partial b^l_j} = \delta^l_j$$

After computing the gradients for all weights and biases, the model parameters are updated using a learning rate α to control the step size during optimisation. The steps are performed iteratively over the entire training dataset until a minimum error or a defined number of epochs is reached.

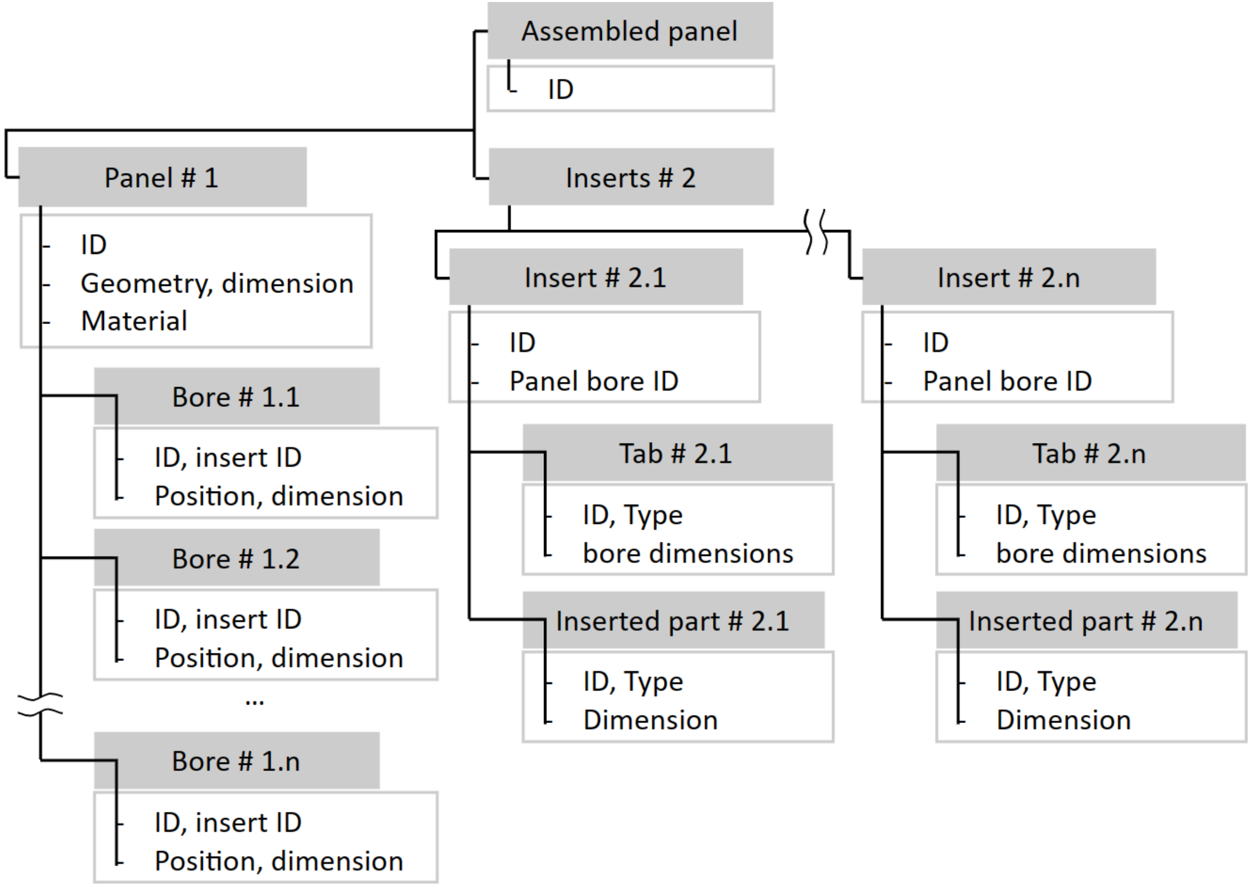
The main difference between an MLP to a CNN lies in the architecture and the usage of the model parameters. An MLP is a fully connected ANN where each neuron in a layer is connected to every neuron in the subsequent layer. As visualised, it consists of an input layer, several hidden layers, and an output layer. All neurons in the MLP are independent of each other, and there is no parameter sharing across different parts of the network. Each connection between a neuron in a layer and a neuron in a subsequent layer has an individual weight.

A CNN includes convolutional layers that use small filters (kernels) to convolute over the input data to identify patterns. The output of a convolutional layer usually passes through pooling layers to reduce the spatial dimensions. CNNs often include fully connected layers at the end to generate the final predictions. Due to the convolution of the filters over the entire input data, the kernel weights are shared across the neurons. In CNN, parameter sharing is a key feature. The same filter is used across different spatial locations of the input, allowing the network to learn and detect similar patterns at different positions. This reduces the number of parameters and makes CNNs more effective in handling large images or data with spatial patterns.

Souce: Nielsen, 2015

A2 - Additional Material on Chapter 5

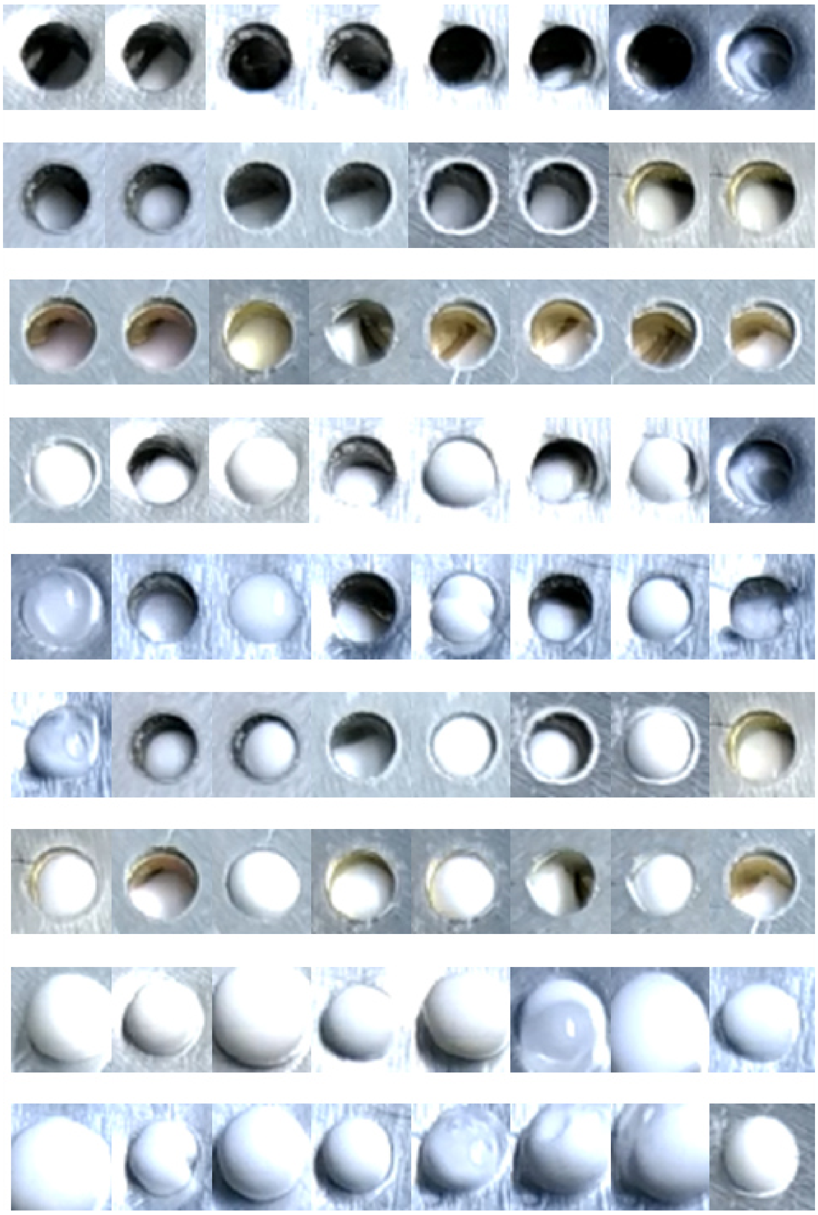
A2.1 - Product Structure of the Reference Products



Product structure and product characteristics of the assembled panel and its parts.

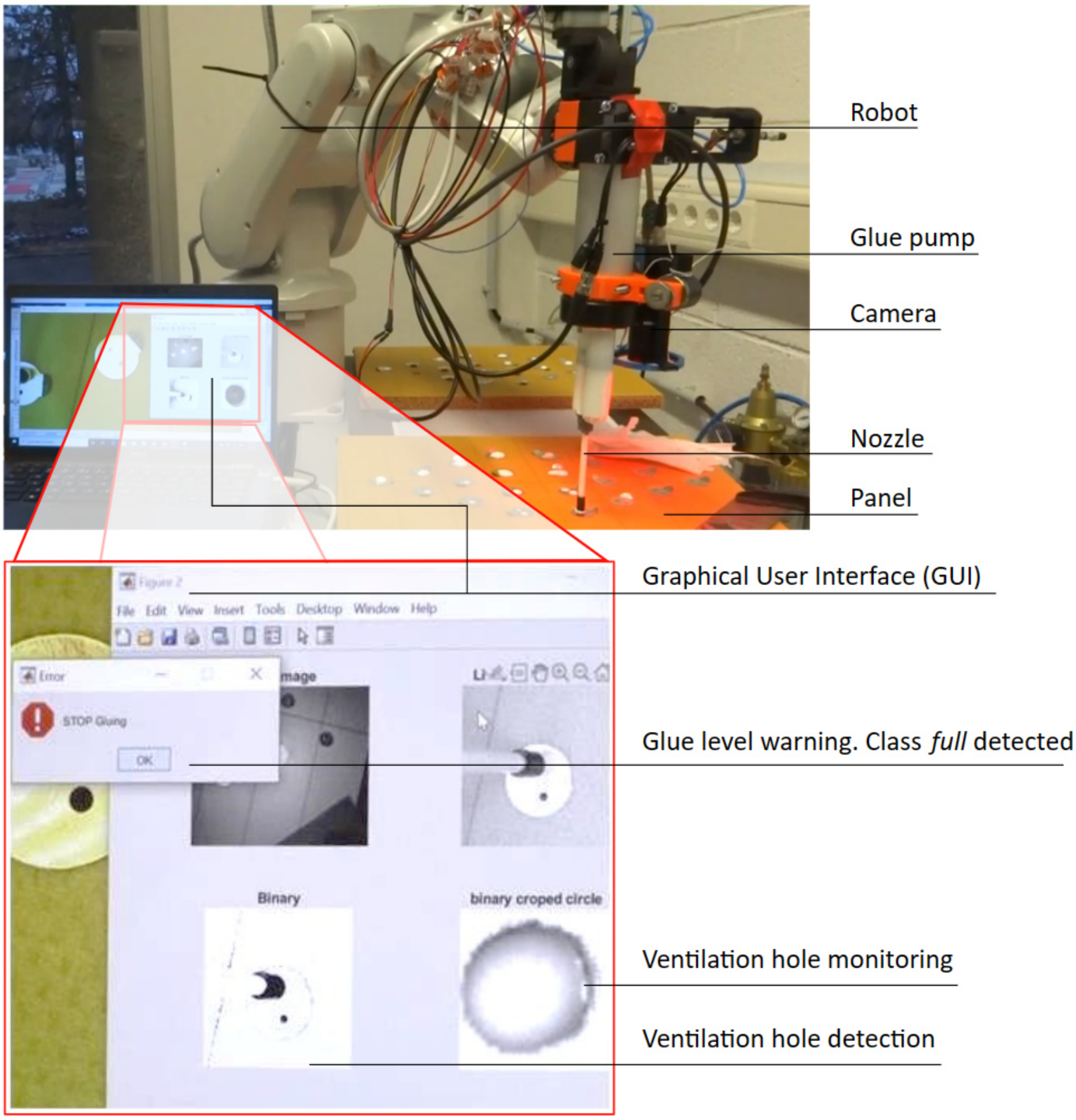
A3 - Additional Material on Chapter 6

A3.1 - Glueing Dataset Overview



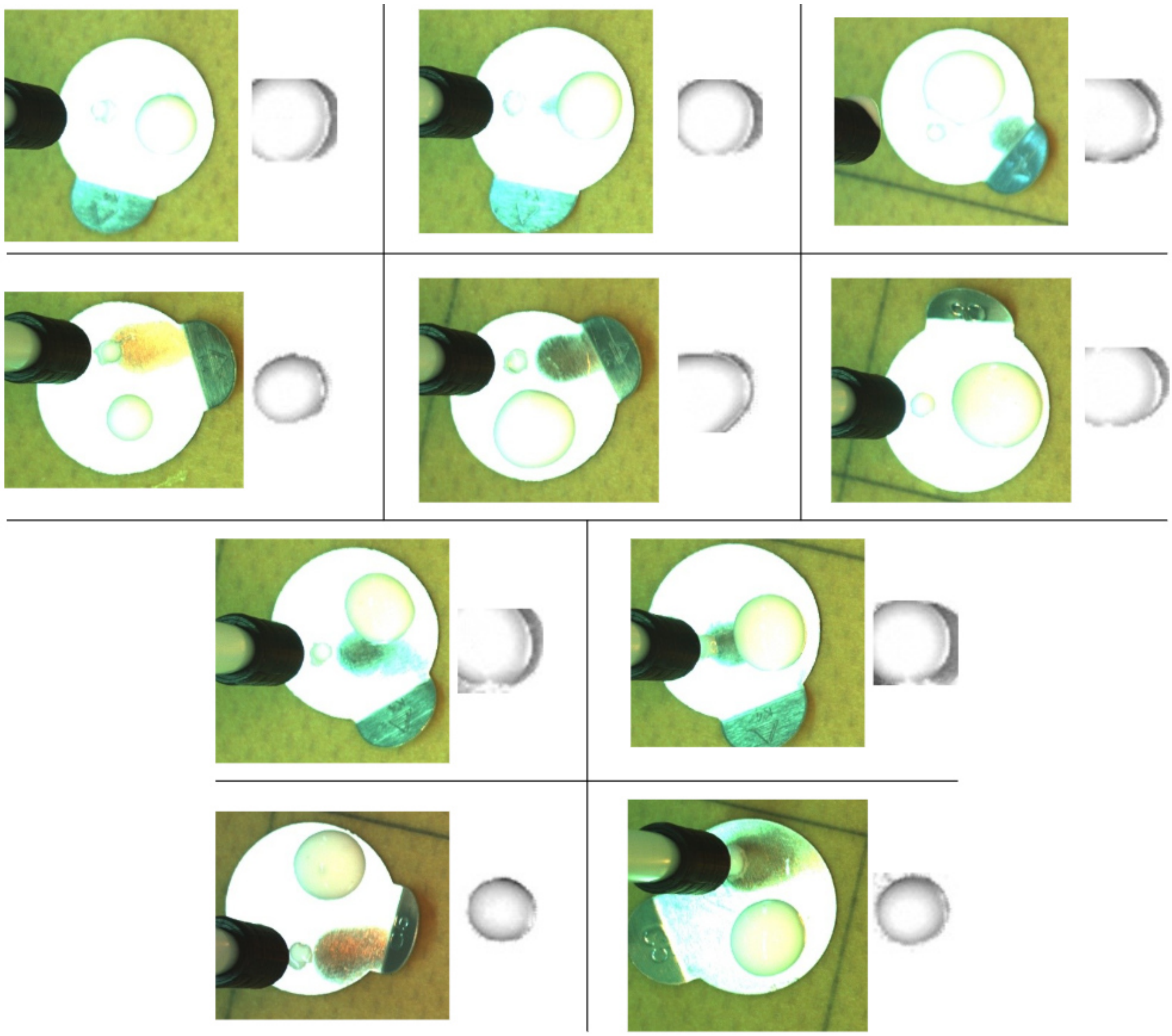
Example images of the glueing dataset used in Chapter 6.

A3.2 - Setup of the Live Glueing Trials


Robot
Glue pump
Camera
Nozzle
Panel
Graphical User Interface (GUI)
Glue level warning. Class *full* detected
Ventilation hole monitoring
Ventilation hole detection
Figure 2
File Edit View Insert Tools Desktop Window Help
Error
STOP Gluing
OK
mage
Binary
binary croped circle

Robotic test stand with glue pump and camera. The laptop displays the Glue Detection Software Graphical User Interface (GUI).

A3.3 - Results of the Live Glueing Trials



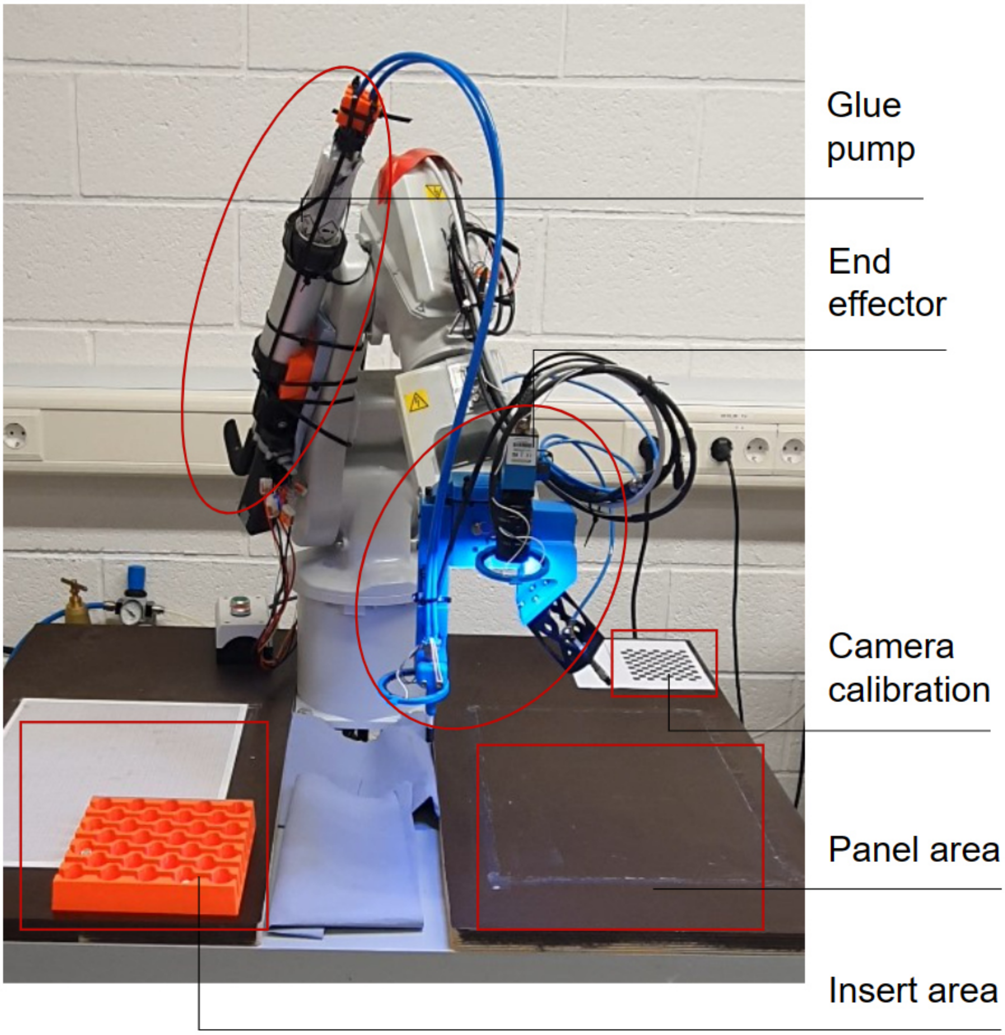
Examples of classified images and glued insert from above.

The figure above depicts the last image classified as *full* during a live glueing video and an image of the scenery after lifting the nozzle from the inlet hole. Due to the low viscosity of the wooden glue used for the experiments, the glue bubble dissolves into a disc-like shape.

Appendix B - Case Studies

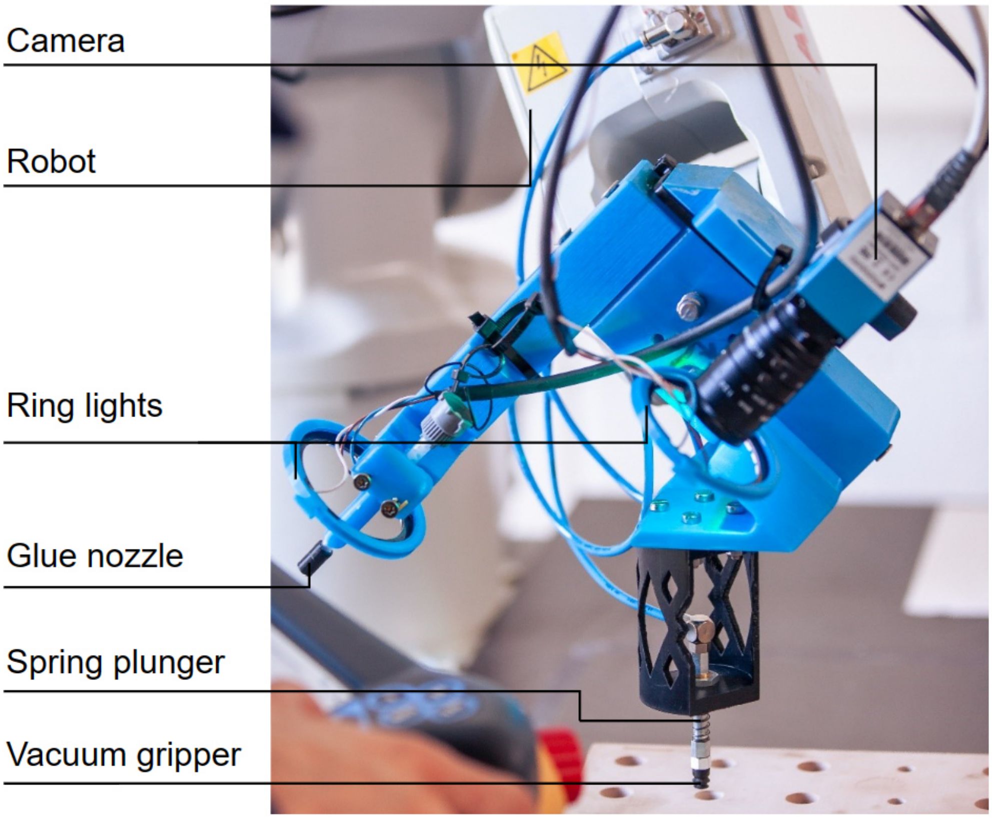
B1 - Case Study 1

B1.1 - CS 1 - Demonstrator Setup



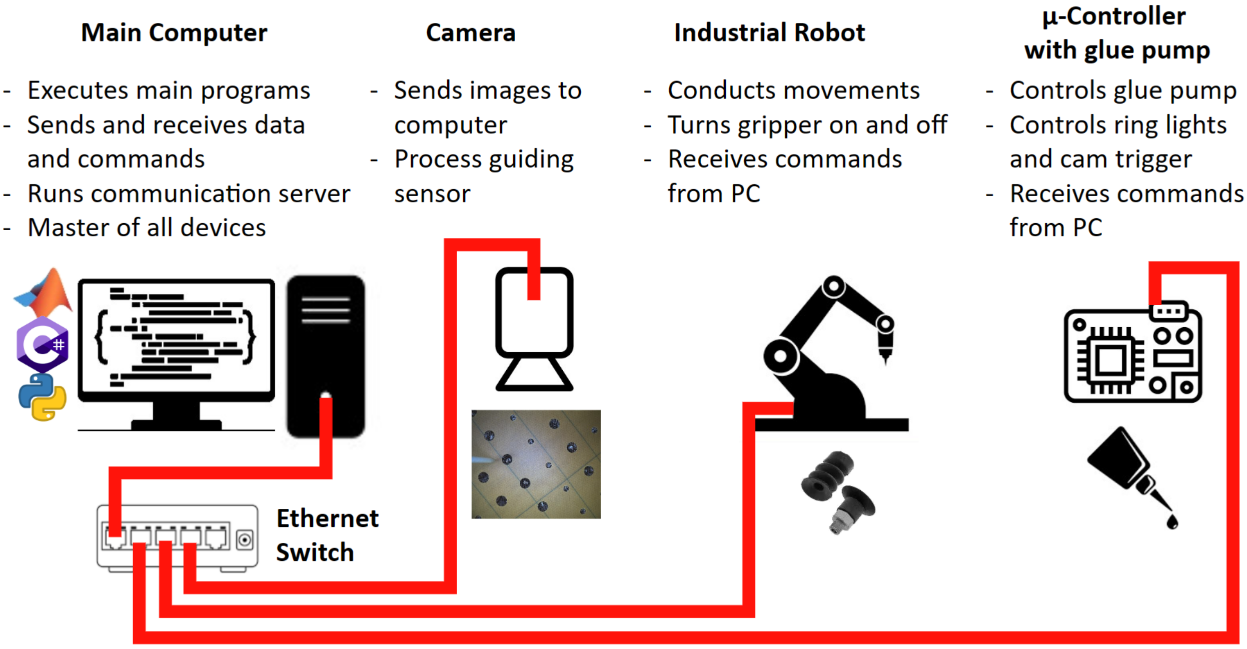
Overview of the technology demonstrator setup.

Case Study 1 is conducted on the test setup depicted in the previous figure. The setup features an industrial robot with a multifunctional end effector (MFEE). Areas for the panel placement, the insert provision, and the camera calibration using a chessboard are available. A closeup of the MFEE is provided in the next figure.



Multi-functional end effector with a camera, vacuum gripper, glue nozzle, and two ring lights.

The MFEE carries all the required tools for the demonstration. The tool centre points of each tool are programmed on the robot so that a point in space can be approached with each tool. The ring lights enlighten the bore for bore detection or the outlet hole during glueing.



Main Computer

- Executes main programs
- Sends and receives data and commands
- Runs communication server
- Master of all devices

Camera

- Sends images to computer
- Process guiding sensor

Industrial Robot

- Conducts movements
- Turns gripper on and off
- Receives commands from PC

µ-Controller with glue pump

- Controls glue pump
- Controls ring lights and cam trigger
- Receives commands from PC

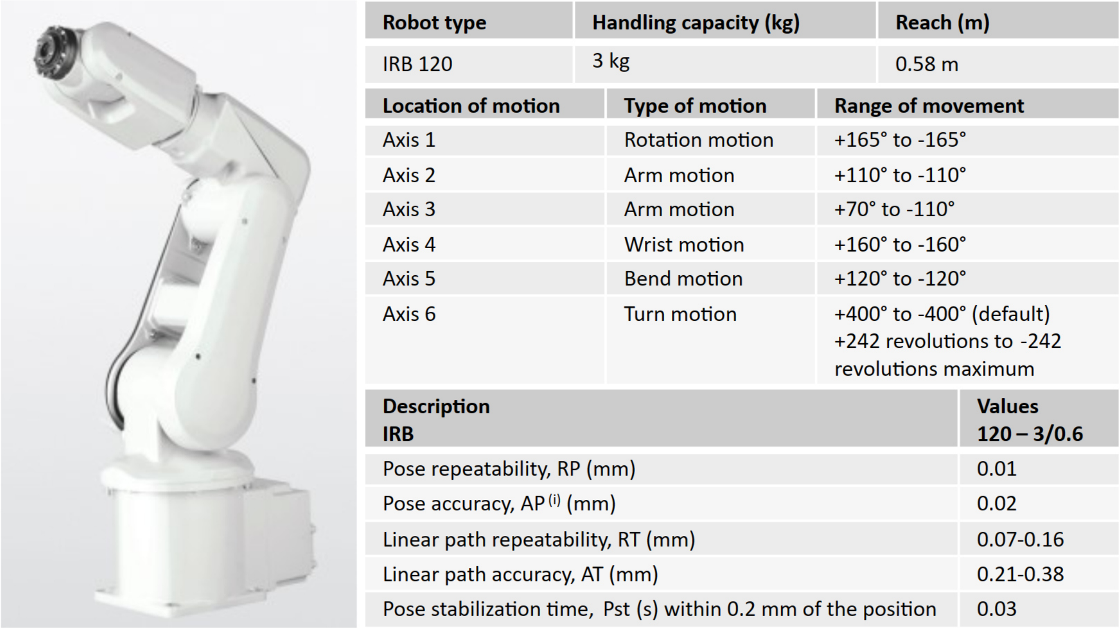
Wiring

All components are connected using Ethernet with TCP/IP communication. The main computer executes the production program and the CV algorithms. The camera sends images or video streams via the GigabitEthernet standard to the computer. The industrial robot receives commands and sends acknowledgements via socket communication. It controls the vacuum gripper with 24V industrial I/O channels. The Arduino One controls the ring lights and the glue pump. It receives and sends messages via MQTT communication.

Each component is specified in the following pages.

B1.2 - CS 1 - Demonstrator Hardware

Industrial robot



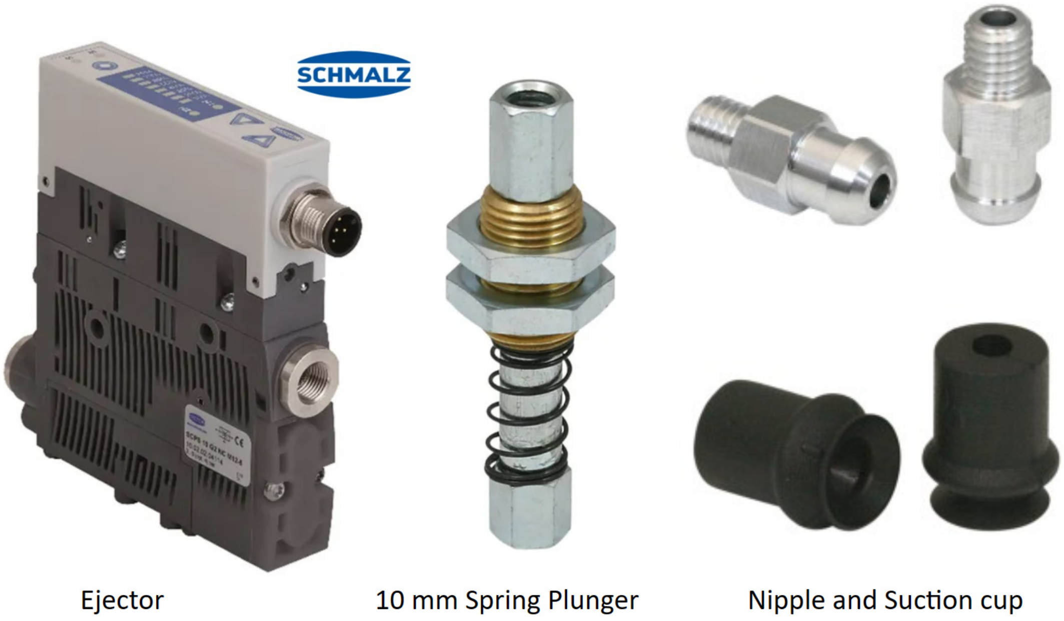
Robot type	Handling capacity (kg)	Reach (m)
IRB 120	3 kg	0.58 m

Location of motion	Type of motion	Range of movement
Axis 1	Rotation motion	+165° to -165°
Axis 2	Arm motion	+110° to -110°
Axis 3	Arm motion	+70° to -110°
Axis 4	Wrist motion	+160° to -160°
Axis 5	Bend motion	+120° to -120°
Axis 6	Turn motion	+400° to -400° (default) +242 revolutions to -242 revolutions maximum

Description IRB	Values 120 – 3/0.6
Pose repeatability, RP (mm)	0.01
Pose accuracy, AP (i) (mm)	0.02
Linear path repeatability, RT (mm)	0.07-0.16
Linear path accuracy, AT (mm)	0.21-0.38
Pose stabilization time, Pst (s) within 0.2 mm of the position	0.03

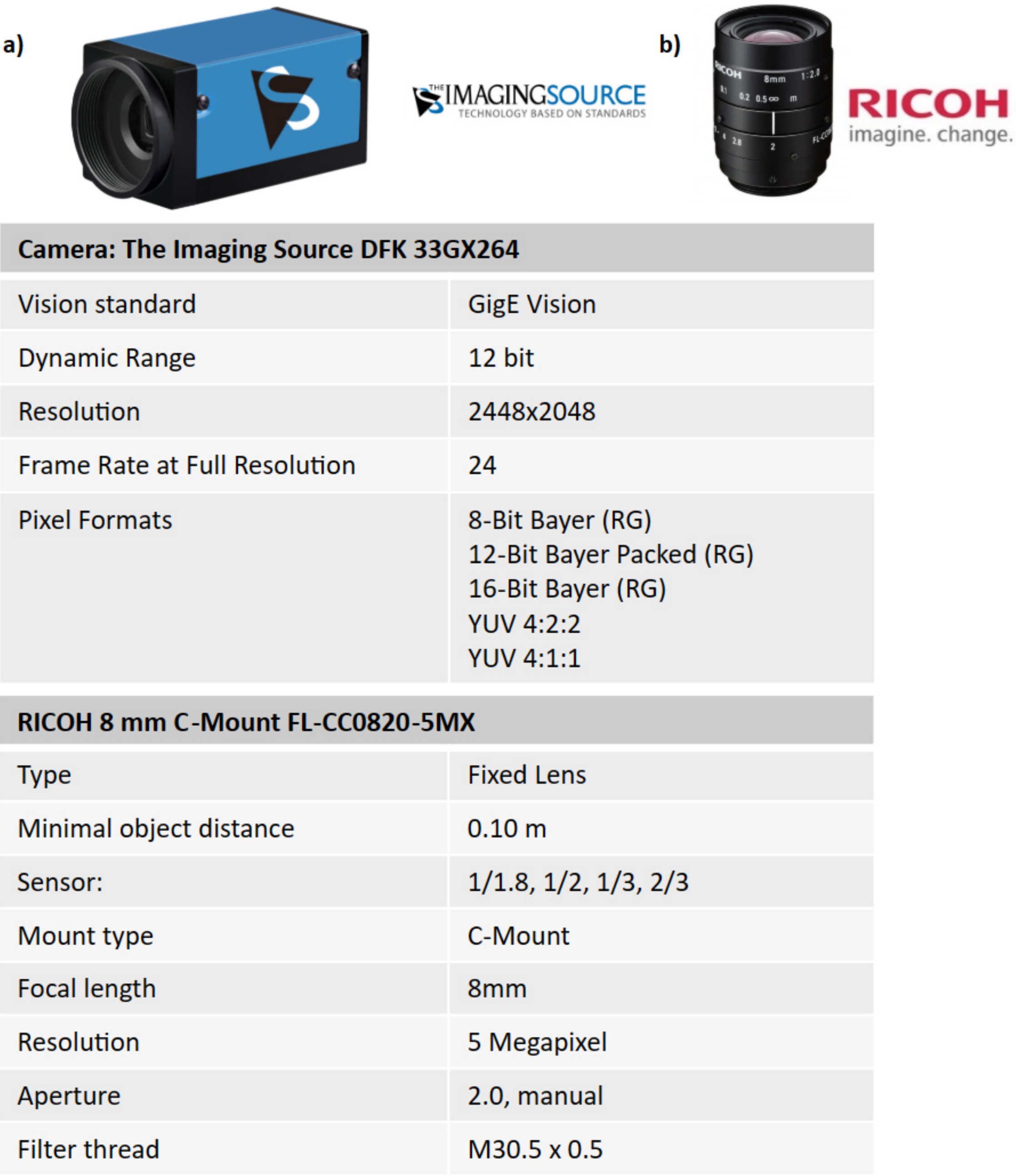
Industrial robot ABB IRB 120.

Vacuum gripper



Ejector, spring plunger, and suction gripper from Schmalz.

Camera system



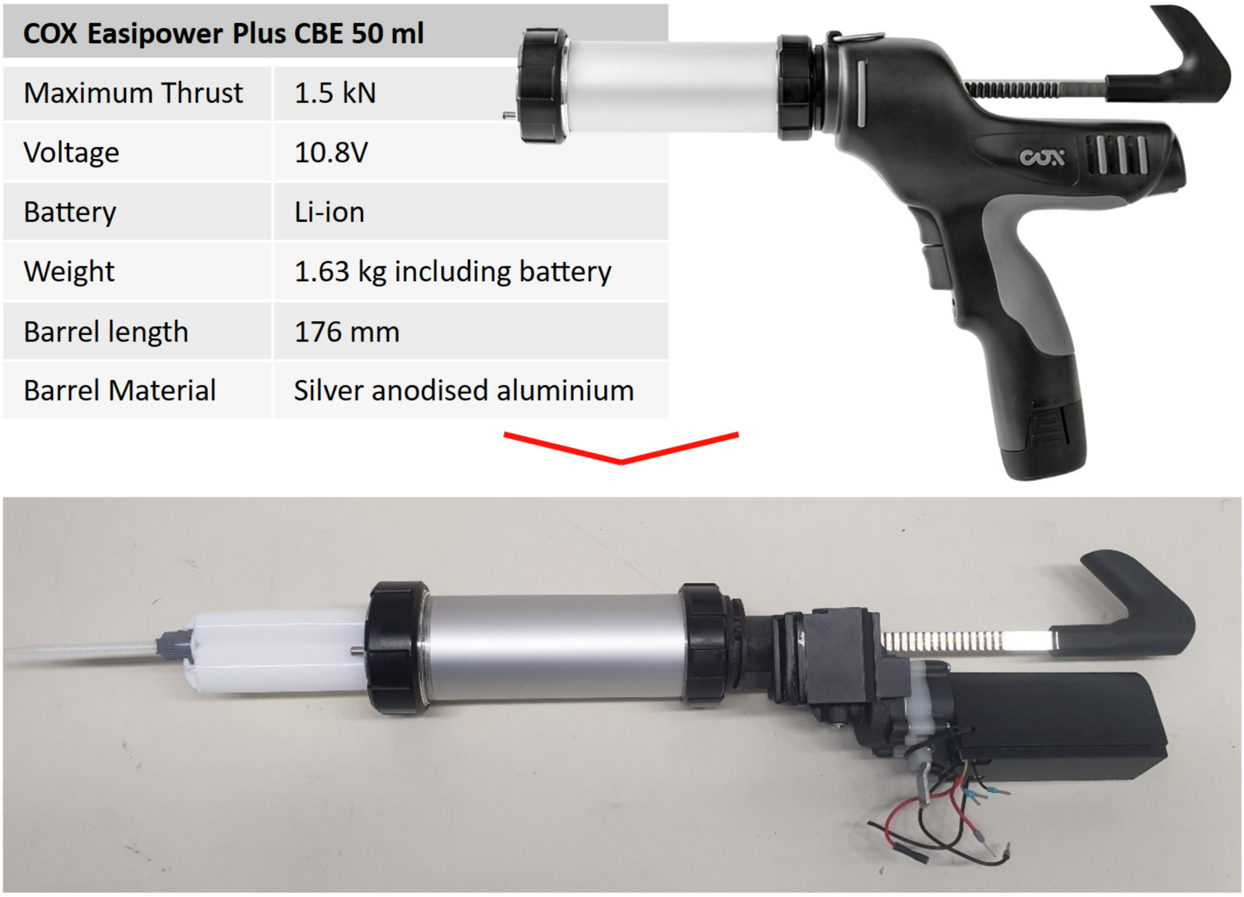
Camera: The Imaging Source DFK 33GX264	
Vision standard	GigE Vision
Dynamic Range	12 bit
Resolution	2448x2048
Frame Rate at Full Resolution	24
Pixel Formats	8-Bit Bayer (RG) 12-Bit Bayer Packed (RG) 16-Bit Bayer (RG) YUV 4:2:2 YUV 4:1:1

RICOH 8 mm C-Mount FL-CC0820-5MX	
Type	Fixed Lens
Minimal object distance	0.10 m
Sensor:	1/1.8, 1/2, 1/3, 2/3
Mount type	C-Mount
Focal length	8mm
Resolution	5 Megapixel
Aperture	2.0, manual
Filter thread	M30.5 x 0.5

Industrial camera and lens.

A 5MP industrial colour camera type DFK33GX264 from TheImagingSource is installed. The attached lens is a fixed 8mm low distortion lens of Ricoh.

Glue pump



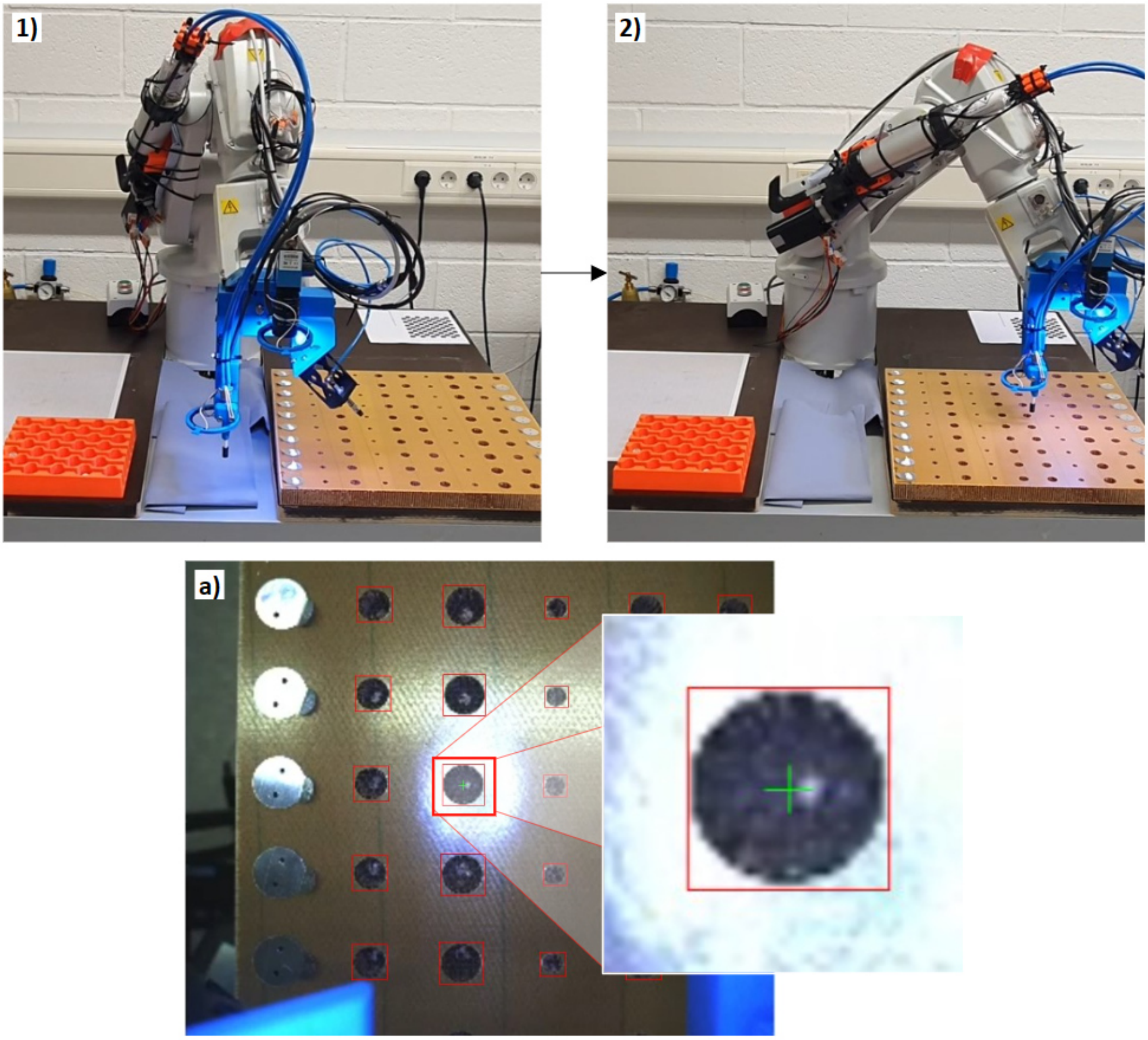
COX Easipower Plus CBE 50 ml	
Maximum Thrust	1.5 kN
Voltage	10.8V
Battery	Li-ion
Weight	1.63 kg including battery
Barrel length	176 mm
Barrel Material	Silver anodised aluminium

Modification of the handheld glue pump type COX Easipower Plus CBE 50ml.

The handheld electric glueing pump COX Easipower Plus CBE 50 ml is modified into a robot-mountable, controllable glue pump. Therefore, the housing is removed, and the PCB is connected to the Arduino One. The power supply via 10.8V DC is removed from the pump and located in the electric cabinet.

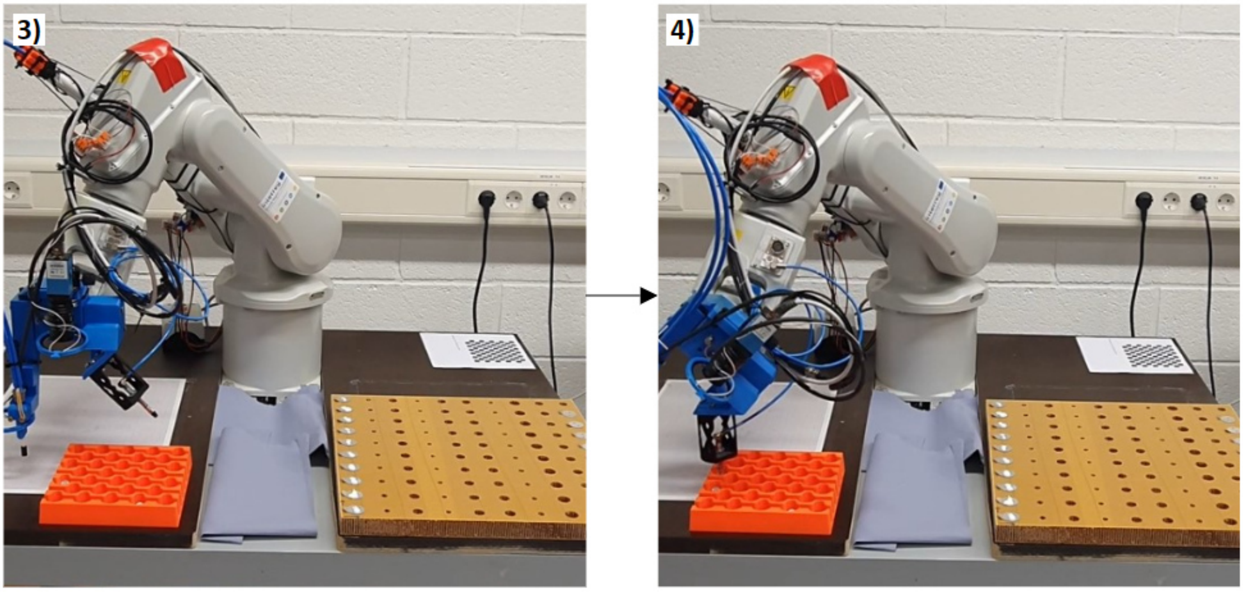
B1.3 - CS 1 - Validation

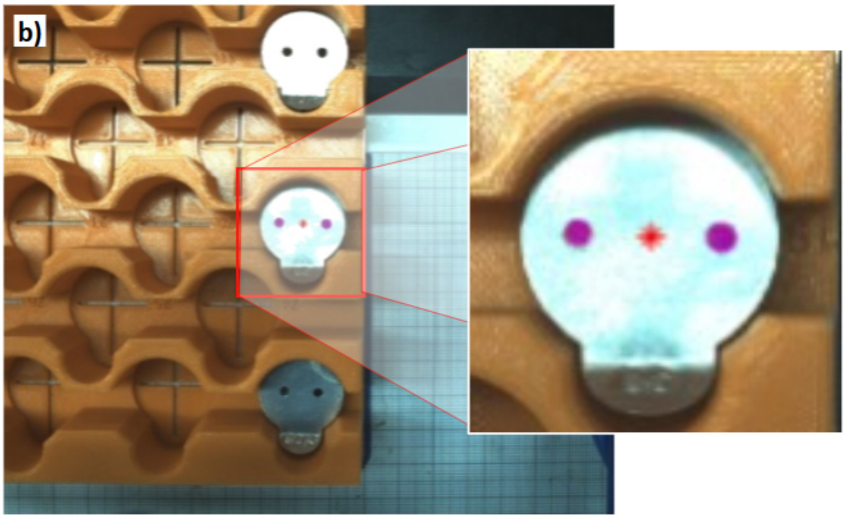
The following pages provide larger versions of the technology demonstration figures described in Section 7.1.3. The first two steps show the robot's motion to the assumed bore position and the bore detection. The detection result and the zoom to the central bore are shown in image a).



Bore Detection.

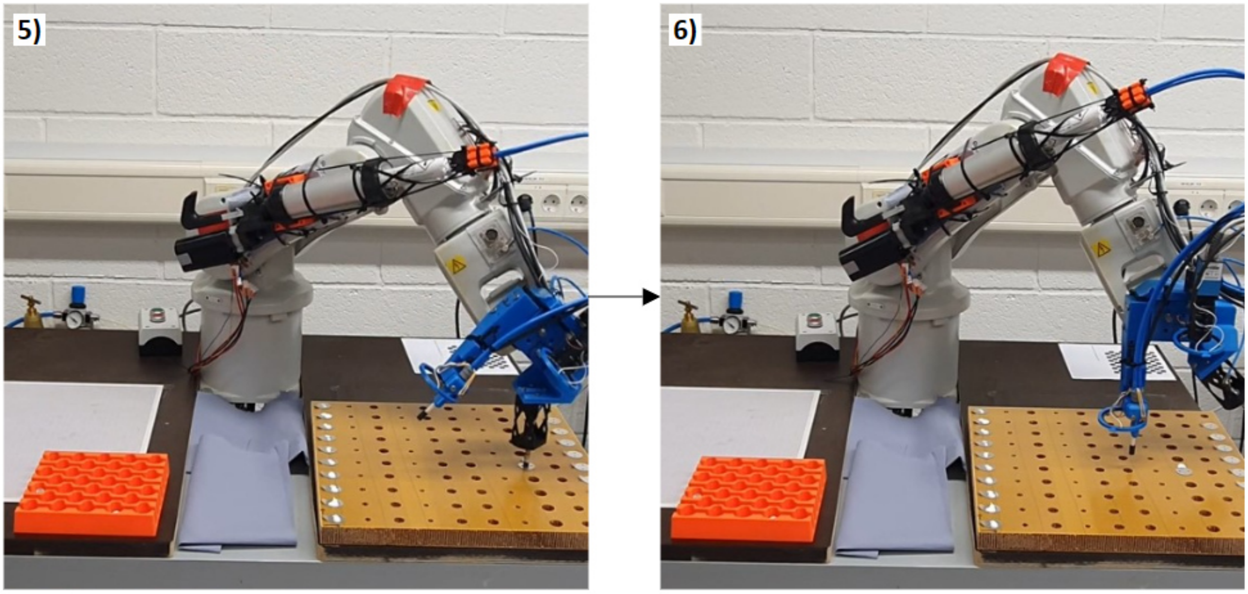
The next two steps visualise the tab detection, including the gripping point highlighted by a red star and the insert pick-up. After the detection, the MFEE rotates around the sixth roboter axis to bring the suction gripper in position. With a vertical motion, the insert is gripped.

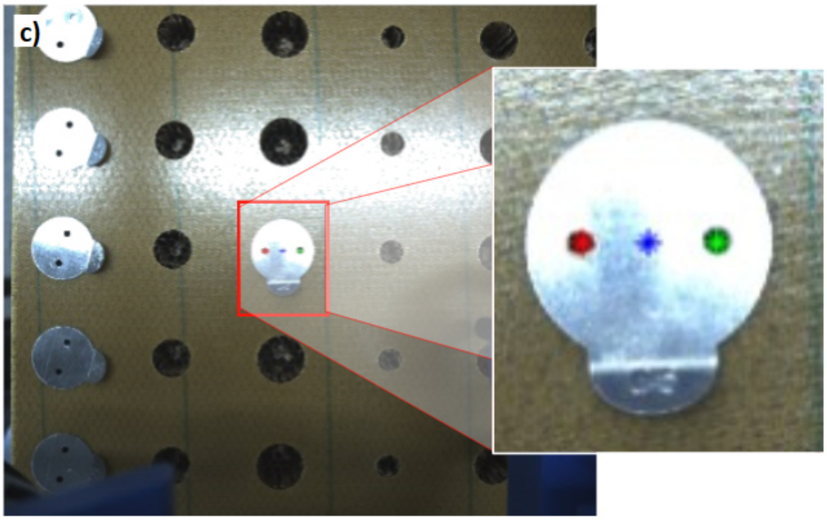




Tab and gripping point detection.

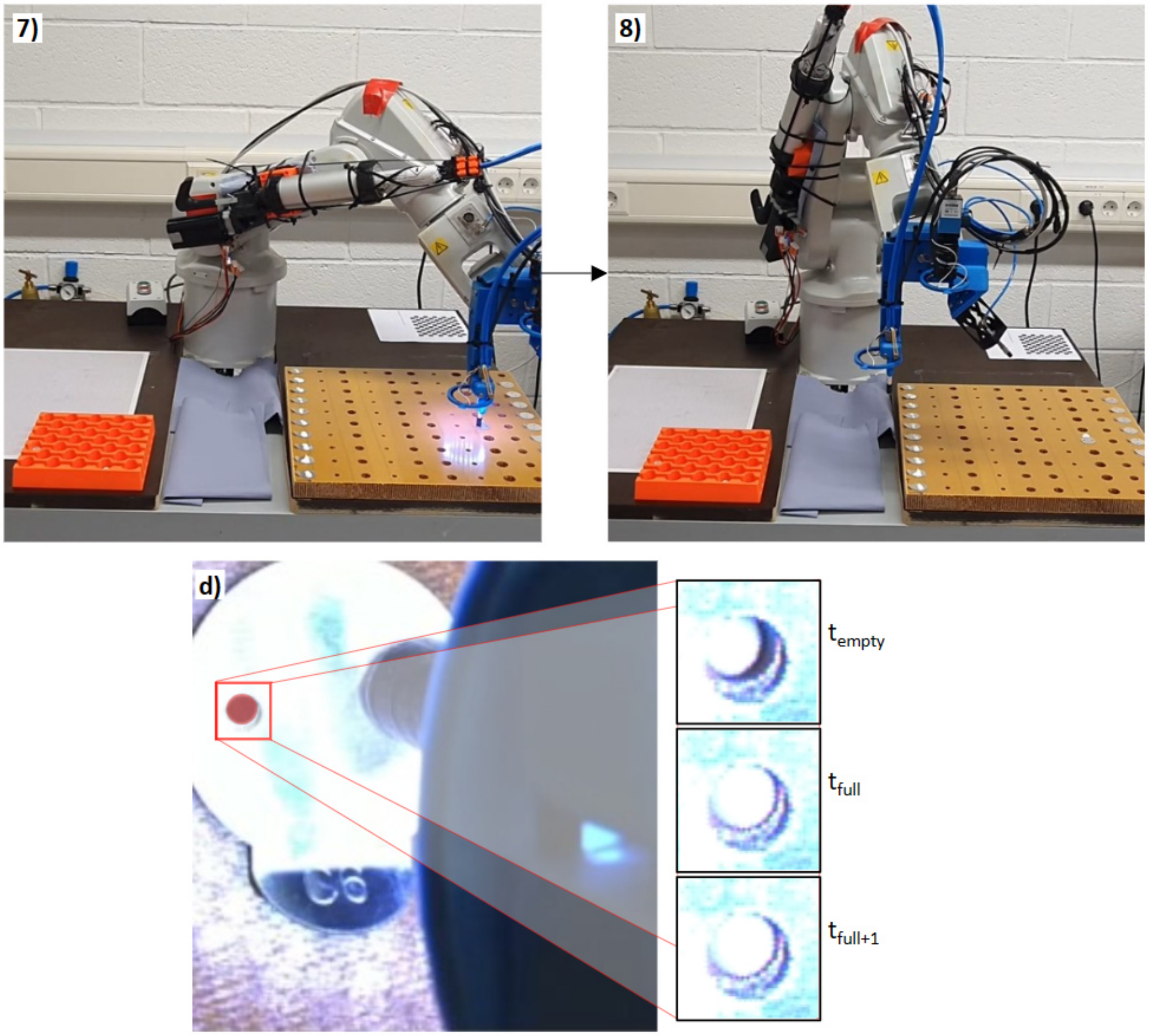
Steps 5 and 6 depict the insert placement and the detection of the ventilation holes. After placement, the MFEE rotates back to position the camera, and the CV algorithm detects the ventilation holes.





Ventilation holes detection with the selection of the inlet hole.

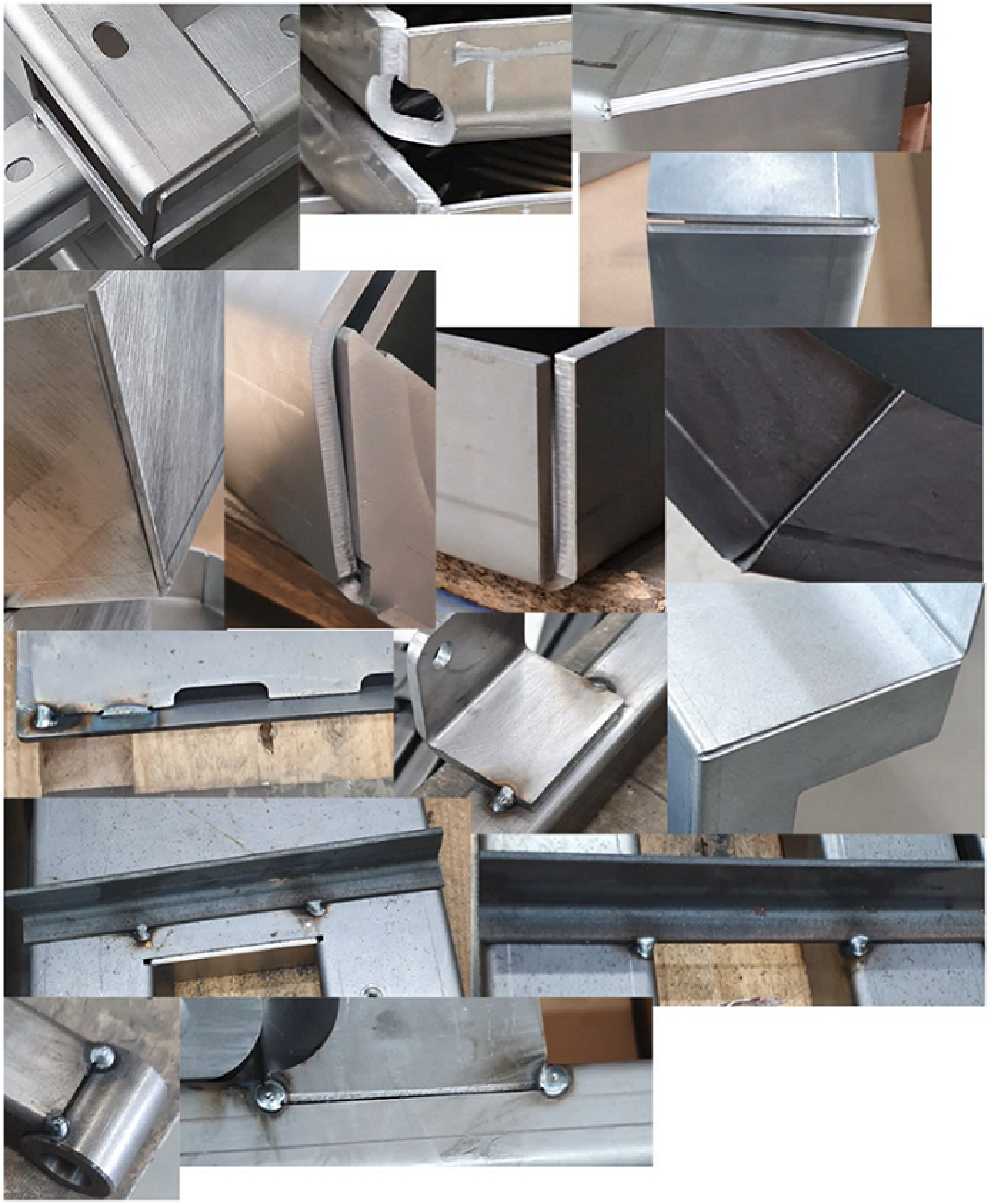
The last steps show the positioning of the nozzle on the inlet hole and the glueing. The ring light around the nozzle is on to enlighten the outlet area. The camera monitors the outlet hole and classifies the glue level. The classification results show the last image classified as empty and the first two images classified as full. Then, the glueing process is stopped, and the robot moves back to the starting position.



Nozzle positioning and live glue classification.

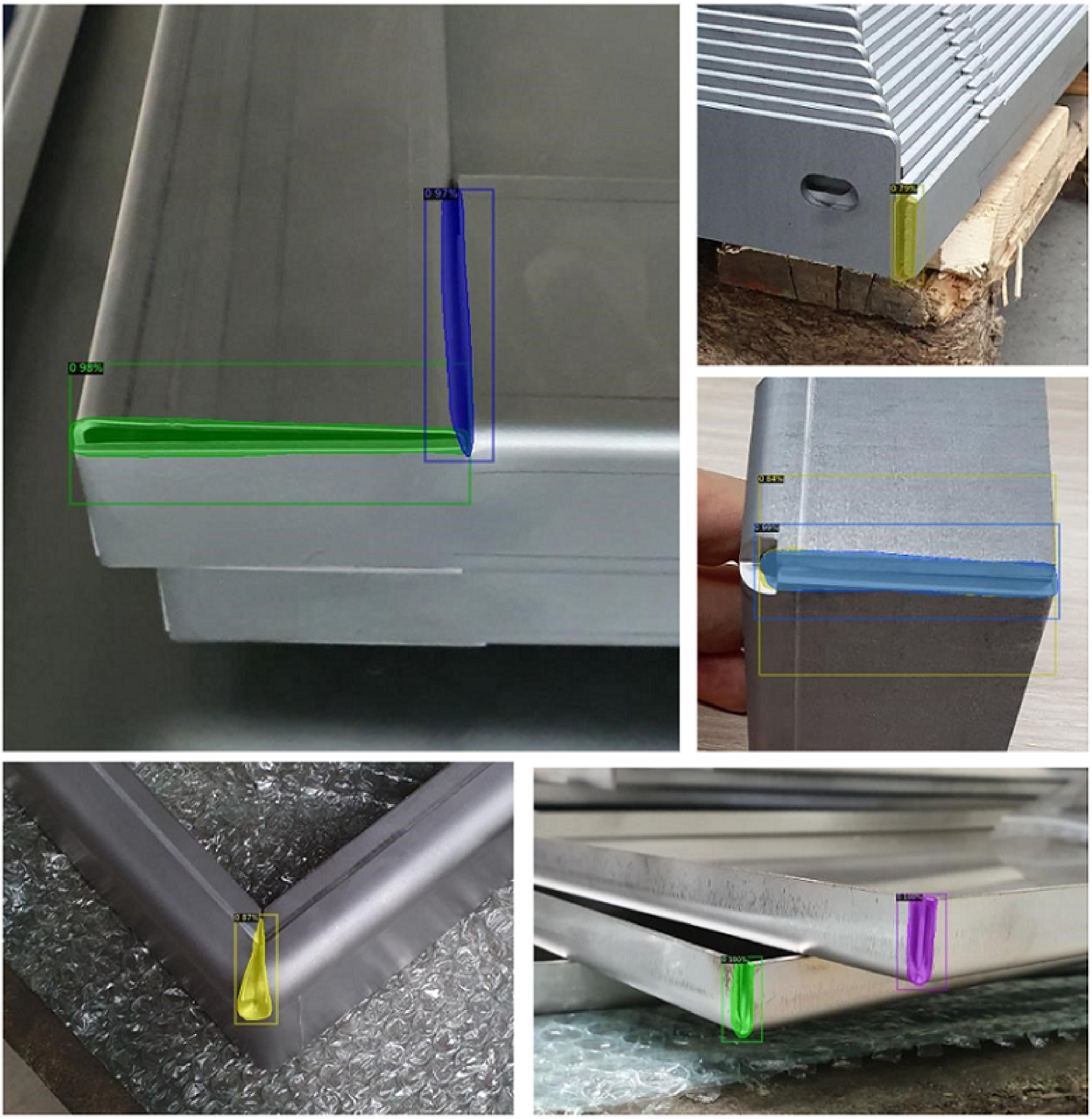
B2 - Case Study 3

B2.1 - CS3 - Bent Sheet Metal Dataset



Various examples of the bent sheet metal dataset with images of different specimens from different angles.

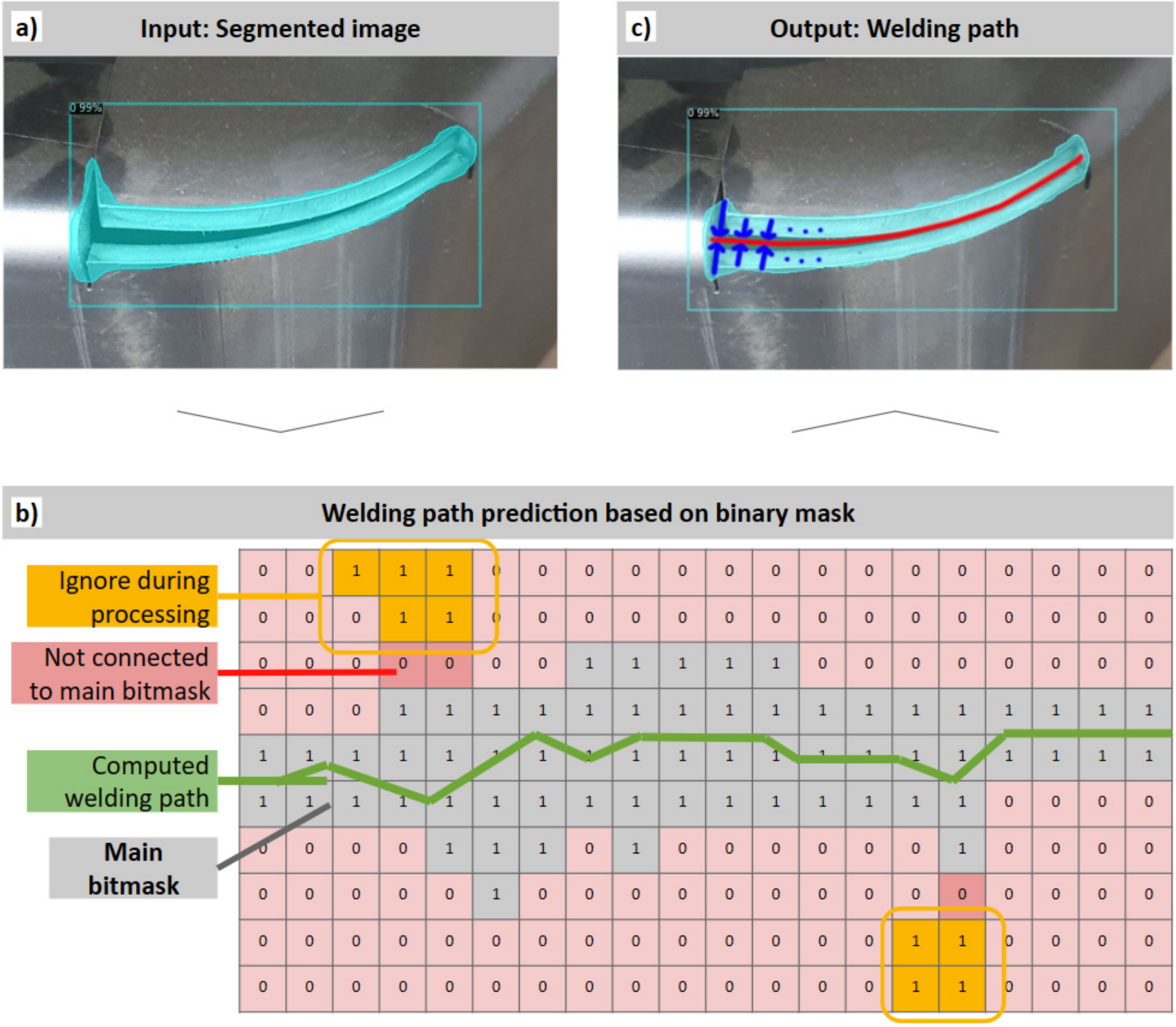
B2.2 - CS3 - Bent Sheet Metal Detection Results



Segmentation results of different specimens located on various surfaces. The unwelded edge is highlighted on the pixel level.

B2.3 - CS3 - Bent Sheet Metal Welding Path Prediction

Based on the predicted instances of the unwelded edges, a welding path is proposed. Therefore, the bitmask of the instance is analysed. Unconnected pixels are ignored during path calculation (see orange pixels). The main bitmask is averaged horizontally or vertically to estimate the welding path. Combined with an overlaid depth image, the welding path's start, support, and end point coordinates can be deducted.



0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Welding path estimation based on the instance bitmask.