

University of Sheffield

Digital-Twin Based Deep Learning for Human-Robot Collaboration



Shenglin Wang

Supervisor: Professor Lyudmila Mihaylova

Second Supervisor: Dr James Law

A thesis submitted in partial fulfilment of the requirements
for the degree of Ph.D in Automatic Control and System Engineering

in the

Department of Automatic Control and System Engineering

Declaration

All sentences or passages quoted in this document from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this dissertation have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure.

Name:

Signature:

Date:

Acknowledgements

The Ph.D study in Sheffield, which lasted four years, was an amazing and extraordinary journey. I would like to express my deepest appreciation to my supervisor, Prof. Lyudmila Mihaylova, for her supervision and guidance in my research and for her kind help in life. She is always willing to listen to my ideas, shares her knowledge generously, and encourages me to explore in my research. Her invaluable guidance, patience, and expertise throughout this journey have been vital to my success.

Also, I would like to thank Dr. James Law and Prof. Sanja Dogramadzi for the great help and valuable advice in the research and Ph.D study. I really appreciate Dr. Peng Wang for his great help and suggestion during my Ph.D journey, especially my first year of the Ph.D study. I would like to express my gratitude to Dr. Yasmeen Rafiq for the invaluable assistance and guidance provided throughout the project. I would also express my thanks to my colleagues and co-authors, Dr. Jingqiong Zhang and Tianhao Zhang.

Lastly I would like to dedicate this thesis to my parents, Zhaofeng Wang and Tingjuan Luo. Great appreciation for their unconditional support for me to pursue my dreams. I would also like to thank my girlfriend, Jingxuan Su for years of company, supporting me during both the peaks and the valleys of life.

Abstract

The main objective of this thesis is to create robust deep learning applications in the context of Human-Robot Collaboration (HRC) which is an important topic in manufacturing. It covers three distinct topics related to the application of deep learning and artificial intelligence in manufacturing, human-robot interaction, and neural network performance monitoring.

In the context of Industry 5.0, the integration of Digital Twins and artificial intelligence techniques, particularly deep learning, enhances flexibility and efficiency in smart manufacturing. This thesis introduces a deep learning-enhanced Digital Twin framework capable of detecting and classifying human operators and robots during the manufacturing process which is described in Chapter 3. The framework, developed using Unreal Engine 4 and compliant with the Robotics Operating System, demonstrates improved performance through a semi-supervised detector, ensuring safety and reliability. Evaluation results with a Universal Robot 10 in various scenarios highlight the framework's accuracy and reliability, with the data and a semi-automated annotation tool being made publicly available.

Furthermore, this thesis presents a framework for real-time 3D (three dimensional) human upper-body motion tracking and kinematic estimation in Chapter 4, essential for applications involving physical human-robot interaction, trajectory planning, and user safety. The proposed framework combines a Kalman filter with a deep Convolutional Neural Network (CNN-KF) to accurately infer joint positions from optical images, facilitating kinematic modeling and motion estimation. Evaluation experiments show

robust performance even in the presence of occlusions, with above 90% segment accuracy, with Root Mean Square Error and Mean Average Error reported below 0.05m in the presence of occlusions.

Lastly, this thesis explores the performance of deep neural networks in Chapter 5, specifically faster Region-Based Convolutional Neural Networks (R-CNNs), when tested with data significantly different from the training set. It introduces a framework to monitor neuron activation patterns within a faster R-CNN, using Kullback-Leibler divergence to calculate distances between activation pattern distributions. This enables real-time monitoring of the classifier's behavior when confronted with noisy and divergent data, as demonstrated on publicly available datasets, MNIST and PASCAL.

Overall, this research spans multiple areas of artificial intelligence and deep learning, showcasing their applicability and effectiveness in HRC and manufacturing.

Contents

List of Abbreviations	xii
List of Figures	xiii
List of Tables	xviii
1 Introduction	1
1.1 Aims and Objectives	3
1.2 Thesis Outline	4
1.3 Research Contributions	6
1.4 List of Peer Reviewed Publications	7
2 Literature Review	9
2.1 Background and the Development of Deep Learning and Computer Vi- sion in Manufacturing	10
2.2 Deep Learning and Computer Vision	11
2.2.1 Image Classification	11
2.2.2 Object Detection	13

2.2.3	Fully-supervised and Semi-supervised Deep Learning for Object Detection	14
2.2.4	Uncertainty and Activation Pattern Monitorings	16
2.3	Human Pose Estimation	17
2.3.1	2D (Two-Dimensional) Human Pose Estimation	18
2.3.2	3D (Three-Dimensional) Human Pose Estimation	19
2.3.3	Occlusions in human pose detection	20
2.4	Human-Robot Collaboration	21
2.4.1	From Simulatuion to Real (Sim2Real)	22
2.4.2	Digital Twin for HRC Safety and Resilience in Manufacturing	29
2.4.3	Robot-assisted dressing	32
2.5	Conclusions	33
3	A Deep Learning-enhanced Digital Twin Framework in HRC	37
3.1	Introduction	37
3.2	The Deep Learning-enhanced Digital Twin Framework	43
3.2.1	Communication Design of the Digital Twin	45
3.2.2	A Digital Twin for Synthetic and Real Data Acquisition	47
3.2.3	A Digital Twin for Intelligent Sensing and Machine Vision Tasks in Changeable Environments	52
3.2.4	A Semi-supervised Teacher-student Detector for Sim2Real	54
3.2.5	Relevance to the Standards and Regulations for HRC	55
3.3	Datasets	57

3.3.1	Semi-automated Annotation Tool	57
3.3.2	Real Data	58
3.3.3	Synthetic Data	60
3.4	Performance Evaluation and Validation	61
3.4.1	Evaluation Metrics	61
3.4.2	Experiment Setting	63
3.4.3	Performance Evaluation of Detection	64
3.4.4	Decision Making for Safe HRC	67
3.4.5	Discussion	71
3.5	Conclusions	72
4	Deep Learning-Enabled Resilience to Occlusion for Physical Human-Robot Interaction	73
4.1	Introduction	73
4.2	Proposed Framework	75
4.2.1	The CNN-KF model	76
4.2.2	Parametric multi-body model	79
4.2.3	Inverse kinematic solver	79
4.3	Experimental setup	81
4.3.1	Participants	81
4.3.2	Motion capture	82
4.3.3	Planned disruptions	83

4.3.4	Rigid-body model parameters	83
4.3.5	LM solver parameters	84
4.4	Results from User Trials	84
4.5	Analysis and Evaluation	89
4.5.1	Inverse kinematic solver	90
4.5.2	Handling occlusions with the CNN-KF	91
4.6	Conclusions	93
5	Real-time Activation Pattern Monitoring and Uncertainty Characterisation in Image Classification	95
5.1	Introduction	95
5.2	Methodology	98
5.2.1	Activation Pattern Representation	98
5.2.2	Central Activation Patterns	101
5.2.3	Activation Pattern Distance Distribution	102
5.2.4	Choice of Thresholds and Monitoring Zones	103
5.2.5	The Activation Pattern Monitoring Algorithm	103
5.3	Experiments and Analysis	106
5.3.1	Datasets and Implementation Details	106
5.3.2	Validation Results and Analysis	109
5.4	Conclusions	113

6	Conclusions and Future Works	114
6.1	Future Work	117
	Reference	120

List of Abbreviations

AI	Artificial Intelligence
AP	Average Precision
AR	Augmented Reality
BDD	Binary Decision Diagram
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CNN-KF	CNN with a Kalman filter
Cobot	Collaborative Robots
CV	Computer Vision
DH	Denavit-Hartenberg
DNN	Deep Neural Network
DoF	Degree-of-Freedom
DP	Dynamic Programming
DSED	Deeply Supervised Encoder Distillation
EMA	Exponential Mean Average
FK	Forward Kinematics

FN	False Negative
FP	False Positive
fps	frames-per-second
FTP	File Transfer Protocol
GCN	Graph Convolutional Network
GPU	Graphics Processing Unit
HRC	Human-robot Collaboration
IK	Inverse Kinematics
IMU	Inertial Measurement Unit
IoT	Internet of Things
IoU	Intersection over Union
KF	Kalman filter
KL	Kullback-Leibler
LLM	Large Language Model
LM	Levenberg-Marquardt
mAP	mean Average Precision
OT	Occupational therapist
PFL	Power and in-force Limiting
R-CNN	Region-based Convolutional Neural Network
RGB-D	Red, Green, Blue plus Depth camera
RoI	Region of Interest
ROS	Robot Operating System

RPN Region Proposal Network

SGD Stochastic Gradient Descent

Sim2Real Simulation to Real

SMPL Skinned Multi-Person Linear

TP True Positive

UE4 Unreal Engine 4

UR Universal Robot

URDF Unified Robot Description Format

List of Figures

2.1	A Digital Twin process facilitates the transfer of knowledge between Digital and Physical entities. The model is validated prior to the deployment of the physical asset in the actual environment. Moreover, feedback from the Physical asset contributes to the optimization of the entire process. The Digital Twin is also capable of tracking the performance of the physical asset.	29
3.1	Fig 3.1(a) and 3.1(b) shows the configuration of an industrial HRC process, where an operator exchanges components with a cobot at a shared handover location. The robot cell is open on one side, allowing staff to enter the cell under specific circumstances	38
3.2	A HRC cell is available in the Sheffield Robotics Lab at the University of Sheffield, UK. An HRC cell is shown in this picture, where there is an operator desk in front of the cobot and the operator exchanges components with the cobot on the desk. A Kinect sensor is mounted on the top of the cell to monitor the HRC operation.	41

3.3	Theoretical framework of using deep learning and Digital Twin techniques for monitoring Cobots towards safety and reliability. The framework is comprised of three layers: i) Digital Twin layer, ii) deep learning layer, and iii) real data generation layer. Digital Twin layer illustrates the Digital Twin in which a ROS-based communication system is designed for information transmission including robot pose, the orientation and position of the camera, etc. between the digital and the physical system. Deep learning layer represents how the synthetic dataset with accurate annotations is generated, then the detector is trained with the dataset. The detector is applied to monitor humans and the cobot in the physical system. In the meanwhile, it also illustrates how a semi-supervised detector is trained which will be explained in Section 3.2.4. In the real data generation layer, a deep learning-based annotation tool is developed to assist to collect and annotate real data.	44
3.4	A ROS based communication framework is designed for the Digital Twin. In this framework, cameras, cobots and users are regarded as nodes. In a ROS framework, nodes communicate with each other through topics, services, and actions provided by ROS.	46
3.5	The Synthetic data including different types of sensing information of the cobot generated from the Digital Twin	48
3.6	(a) represents a RGB image of the cobot, while the masks of the cobot and its components are illustrated from (b) to (c). The digital system can generate different component masks which is defined by users. The cobot mask can be separated into different components and components can be combined as the one. Consequently, users can obtain masks based on their requirements to meet different tasks.	49

3.7	Architecture of the Faster R-CNN. ResNet-50 extracts feature maps from the input image. In Region Proposal Network, regions of interest are generated. RoI Pooling processes the regions of interest and their corresponding feature maps to get new feature maps with fixed size. The FC (Fully connected layer) predicts the classes and the bounding boxes for these feature maps.	52
3.8	Framework of the semi-supervised method applied to train a detector. A teacher model is firstly trained with the synthetic data. The unlabeled real data is fed to the teacher model and the teacher model generates pseudo labels for the unlabeled real data during the testing mode. The pseudo labels are further filtered. Next a student model is trained with the real data with filtered pseudo labels.	54
3.9	Images with annotation information in the real dataset. From (a) to (d), the whole process of human-robot collaboration is captured from the Kinect V2 sensor mounted on the top of the UR10.	58
3.10	Images with annotation information in the synthetic dataset. (a) and (b) shows human images from COCO dataset, while (c) and (d) are robot images generated from the digital system of the Digital Tiwn. . .	60
3.11	Three safety criteria for safety decision making.	68
4.1	The Proposed framework for occlusion-robust and ergonomically safe physical human-robot interaction.	75
4.2	Human upper limb representation as a 10 DoF robotic arm adapted from Xsens MVN model [179]; two kinematic chains: 1) Left-arm and 2) Right-arm.	79
4.3	Participant p_1 performing dressing trial for spasticity pattern II.	85

4.4	World image [h] depicts CNN being able to learn p_1 's occluded left wrist joint location from partially visible forearm. In [i]-[j], the CNN is not able to learn p_1 's wrist joint location correctly due to occlusion by the garment.	86
4.5	Left-arm joint signals, in window W_1 the participant p_2 is waiting to be assisted with dressing, there are no occlusions, in windows W_2 and W_3 there are some occlusions, mainly caused by the occupational therapist during assisted dressing.	87
4.6	Right-arm joint signals, where in window W_1 there are no occlusions or disruptions, in windows W_2 and W_3 there are some occlusions and environmental disruptions.	89
4.7	Deviation of the hand pose estimation, between the IK solution and the VICON marker trajectory, during dressing scenarios I-IV with no/occlusion. The results represent the averages over three participants performing each dressing scenario.	90
4.8	Deviation of assisted dressing trajectory across all four spasticity patterns, between CNN and CNN-KF from the IK solution. The results are based on the averages of three participants performing all four dressing scenarios.	93
5.1	Overview of real-time activation pattern monitoring. The framework includes two phrases. In Phase 1, the activation patterns are first recorded. the central activation pattern of each class is found based on their similarities. In Phase 2, the network is monitored when a new image is fed to the network and the activation situation is monitored.	98

5.2	Visualisation of activation layers (first row) and the corresponding activation patterns (second row) of Number 1 in MNIST dataset. The activation pattern becomes more abstract from left to right as the layer in DNNs gets deeper.	100
5.3	From (a) to (i) represent the activation pattern distributions of digital number from 0 to 8 on MNIST dataset. 'Same' represents the distribution which the central activation pattern is compared to the activation patterns with the same class, while 'Different' represents the distribution which the central activation pattern is compared to the activation patterns with different classes.	109
5.4	From (a) to (i) represent the activation pattern distributions of different classes on PASCAL dataset. 'Same' represents the distribution which the central activation pattern is compared to the activation patterns with the same class, while 'Different' represents the distribution which the central activation pattern is compared to the activation patterns with different classes.	110

List of Tables

2.1	Summary of perception research under HRC	35
2.2	Summary of Sim2Real research under Digital Twin	36
3.1	Numbers of Images in different datasets. To guarantee the real data that contains different light factors, the data is collected under different lighting conditions: full light, semi-light, semi-dark and dark where the lighting condition is changing from light to dark. With respect to the Synthetic data, the data is generated without identifying lighting conditions.	57
3.2	Results under different lighting condition, mAP, AP50 and AP75 is utilized to evaluate three object detection models. Real represents the Faster R-CNN model trained with the real data. Synthetic represents the Faster R-CNN model trained with the synthetic data. Semi-supervised is the semi-supervised model.	65
3.3	mAP results at UR10 and Human under different lighting conditions. Real represents the Faster R-CNN model trained with the real data. Synthetic represents the Faster R-CNN model trained with the synthetic data. Semi-supervised is the semi-supervised model.	66

4.1	Frames of the the upper limb 10 DoF model. Each link can only connect with a 1-DoF joint and so the dummy links are used to create a higher DoF joint.	80
4.2	Upper limb spasticity patterns.	83
4.3	Average performance across the four dressing scenarios based on three participants performing each dressing trial.	91
4.4	Average performance index scores across the four dressing trajectory waypoints for the left-right upper limb, based on three participants performing the four dressing scenarios.	92
5.1	Prediction Results on MNIST	107
5.2	PASCAL object classes	107
5.3	Numbers of TP&FP on PASCAL.	108
5.4	Monitoring Classification Results on MNIST and PASCAL.	112

Chapter 1

Introduction

The advent of Artificial Intelligence (AI) in manufacturing signifies a transformative leap in the industrial sector, heralding the onset of what can be termed as the Fifth industrial revolution or Industry 5.0 [1]. The integration of AI into manufacturing brings about a period of unparalleled efficiency, adaptability, and enhancement in quality. This positions AI as a crucial force in transforming production landscapes on a global scale.

AI plays a crucial role in the manufacturing industry, encompassing a range of areas such as predictive maintenance, quality control, supply chain management, and design optimisation [2]. By leveraging its capacity to rapidly process and analyse large volumes of data, AI enables predictive maintenance, which minimises downtime and extends the lifespan of machinery. Additionally, AI enhances quality control by identifying defects and inconsistencies that human inspectors may overlook [3]. In the realm of supply chain management, AI offers valuable insights into demand forecasting and inventory optimization, resulting in more efficient production schedules and decreased waste.

An additional important use of AI in the manufacturing industry is in robotics. Instead of being simple machines that can be programmed, robots now have AI algorithms that make them adaptive and intelligent [4]. This allows them to learn and improve their

actions. These intelligent robots can work together with humans, which has led to the development of collaborative robots, also known as 'cobots' [5]. This collaboration enhances the interaction between humans and robots, making it safer and more efficient, and also creates new opportunities for manufacturing tasks.

Deep Learning, a branch of machine learning, has brought about a revolutionary change in data analysis and interpretation [6]. It has empowered machines to carry out intricate tasks with remarkable precision and effectiveness. When combined with CV, which facilitates the interpretation and processing of visual data, these technologies have paved the way for new possibilities in manufacturing.

The significance of Deep Learning in manufacturing lies in its ability to process large datasets, learning patterns and features that are often too intricate for traditional algorithms [7]. This capability has been instrumental in various applications, from predictive maintenance and quality control to complex assembly tasks [8]. CV, on the other hand, has enabled machines to interpret visual data, facilitating tasks such as defect detection, product sorting, and real-time monitoring of manufacturing processes. In the context of Industry 5.0, the integration of Deep Learning and CV signifies a move towards more intelligent, automated, and efficient manufacturing systems. These technologies are at the forefront of creating smart factories where machines can analyze, decide, and act with minimal human intervention.

Nonetheless, there are obstacles to overcome when incorporating these sophisticated technologies into the manufacturing sector. Ensuring the dependability of machine learning models, acquiring extensive and varied datasets for training purposes, and seamlessly integrating AI systems into existing manufacturing infrastructure pose substantial challenges. Moreover, as these technologies continue to advance, it becomes increasingly important to address ethical concerns and consider the impact on the workforce [9].

Deploying AI models in manufacturing faces several challenges. These include inte-

grating AI with existing infrastructure, requiring significant upgrades or redesigns [10]. Data quality and quantity are critical, as AI models need large, diverse datasets for training. Nonetheless, the process of data collection is often a resource-intensive endeavor, particularly in terms of financial cost, time, and human effort. This complexity arises from the need to gather large volumes of data, which is essential for ensuring accuracy and reliability, especially in fields that rely heavily on data-driven decisions [11]. One approach to addressing data problem in manufacturing is the generation of synthetic data in simulated environments, subsequently used to train AI models. However, the disparity between simulated conditions and actual manufacturing environments poses significant challenges, a phenomenon known as the 'Simulation to Reality' (Sim2Real) problem [12]. This discrepancy often hampers the AI model's performance in real-world scenarios, indicating the need for strategies that effectively bridge the gap between simulated training and practical application.

Safety in human-robot Collaboration (HRC) within manufacturing environments is a critical area of focus, aimed at ensuring the well-being of human workers alongside efficient and reliable robot operations [13, 14, 15]. It encompasses the development and implementation of various safety measures, protocols, and technologies to minimize risks associated with robot operation. Key aspects include the design of robots with safety features, the establishment of safety zones and barriers, regular safety training for employees, and the integration of advanced sensing and control systems to detect and prevent potential hazards [16]. The continuous advancement of AI and robotics further contributes to developing more intuitive and responsive systems, enhancing safety in dynamic and collaborative workspace.

1.1 Aims and Objectives

The purpose of this thesis is to investigate and strengthen the robustness of Deep Learning applications in HRC in manufacturing settings. The following are the primary

objectives of this research:

- Build and leverage Digital Twin technology for generating robust datasets to train Deep Learning models in HRC scenarios.
- Explore and bridge the gap between simulated environments and real-world manufacturing settings using the Digital Twin system.
- Examine the ability of physical HRC to remain effective in the presence of obstructions, which can be enabled by the use of Deep Learning methods.
- Investigate the utilisation of real-time activation pattern tracking and the characterisation of uncertainty in image categorisation, with a particular emphasis on the robustness and dependability of Deep Learning models in HRC scenarios.

1.2 Thesis Outline

The thesis is organised into six chapters. A brief overview of each chapter and the corresponding contributions is given below.

Chapter 1: This chapter presents the thesis topic and its objectives, highlighting the structure and principal contributions of each chapter. Author’s relevant publications are listed in the last section of this chapter.

Chapter 2: This chapter provides an overview of the utilisation of Deep Learning and CV in manufacturing. It covers common tasks such as classification and object detection. With respect to the the training of a Deep Learning model, it can be classified as fully-supervised and semi-supervised. The related works are reviewed.

Chapter 3: This chapter discusses the integration of Digital Twins and Deep Learning in Industry 5.0 to enhance smart manufacturing, particularly focusing on HRC. It introduces a Deep Learning-enhanced Digital Twin framework to improve safety and reliability in collaborative tasks. This framework, developed using Unreal Engine 4 and

complying with the Robotics Operating System, enables detection and classification of human and robot actions, facilitating autonomous robot decision-making. It includes a fully-supervised detector trained on synthetic data and a semi-supervised detector for bridging the gap between simulated and real environments. The effectiveness of this framework is validated in various scenarios, with its data and a semi-automated annotation tool made publicly available for research and operational use.

Chapter 4: This chapter describes a framework for tracking 3D human body motion and updating kinematic models in real-time, vital for physical HRC. It utilizes a Kalman filter fused with a deep Convolutional Neural Network (CNN-KF) for inferring joint locations from optical images. The framework employs an inverse kinematic solver with the Levenberg-Marquardt method for accurate motion estimation. Its effectiveness is demonstrated through dressing experiments in a motion capture lab, showing high accuracy in human posture estimation even in the presence of occlusions.

Chapter 5: This chapter examines the performance of Deep Neural Networks (DNNs), particularly focusing on faster region-based convolutional neural networks (R-CNNs), in scenarios where testing data significantly differs from training data. It introduces a framework to monitor neuron activation patterns within a faster R-CNN, using Kullback-Leibler divergence to calculate distances between these patterns. This approach helps observe the network’s behavior in challenging conditions, such as noisy or atypical data. The effectiveness of this framework is validated using the MNIST and PASCAL datasets, demonstrating its utility for real-time monitoring of supervised classifiers.

Chapter 6: In this chapter, a summary of all the methods proposed in the thesis is provided, along with an analysis of the corresponding results. Subsequently, directions and ideas for future work are presented based on these findings.

1.3 Research Contributions

Chapter 3: This chapter focuses on Deep Learning methods and Digital Twin to improve safety and reliability in HRC:

- A semi-supervised framework for the detection of humans and robots in manufacturing environments is proposed by adopting a faster region-based convolutional network [17]. It further minimises the gap between the simulation and the real world environment.
- A Digital Twin of an actual HRC system is developed based on Unreal Engine 4. This twin is capable of generating synthetic robot data, which can then be used to train Deep Learning models for the purpose of monitoring human-robot collaborative behaviours.
- The accuracy of the Digital Twin system that was developed is assessed using both simulated and actual data sets. The results show that the system can effectively identify and analyse human-robot behaviours to ensure safety. As part of the research, datasets created by the Digital Twin of a Universal Robot 10 (UR10) robot are made publicly accessible. Additionally, a semi-automated annotation tool is also developed.

Chapter 4: This chapter focuses on Deep Learning methods for human pose estimation in HRC:

- A framework, that is robust to occlusions and environmental disruptions, that uses a single camera to retrieve three dimensional (3D) joint location for a human arm.
- A robust online solution to the Inverse Kinematics (IK) problem that takes estimated hand positions from the CNN-KF and estimates user motion. This solves the IK problem for a hand position, finding an updated model configuration in joint space that satisfies kinematic constraints.

- Evaluation of the accuracy of the CNN, CNN-KF and the IK solution using the mean-squared error analysis, where the ground truth for the hand pose was provided by a VICON motion capturing system.

Chapter 5: This chapter focuses on the decision-making of Deep Learning methods:

- The neuron activation patterns are determined by computing the Hamming distance between the current activation pattern and the central activation pattern. Afterwards, the similarity of these distributions is described using Kullback-Leibler divergence.
- Monitoring zones are created through the process of decision making, where patterns with their respective probability values are considered and any changes in these patterns are visually represented.
- The monitoring framework’s effectiveness is showcased through the use of MNIST [18] and PASCAL [19] datasets.

1.4 List of Peer Reviewed Publications

The author’s publications with relevance to this thesis are listed as follows:

Journal Papers

- [J1] S. Wang, J. Zhang, P. Wang, J. Law, R. Calinescu and L. Mihaylova, “A Deep Learning-enhanced Digital Twin Framework for Improving Safety and Reliability in Human–robot Collaborative Manufacturing”, *Robotics and Computer-integrated Manufacturing*, 2024, 85: 102608. *Impact Factor 9.1*.
- [J2] Y. Rafiq, S. Wang, M. Al-Nuaimi, R. Hieron, L. Mihaylova and S. Dogramadzi, “Deep Learning-Enabled Resilience to Occlusion for Physical Human-Robot Interaction”, *To be submitted to a journal*.

Conference Papers

- [C1] S. Wang, P. Wang, L. Mihaylova, et al. “Real-time Activation Pattern Monitoring and Uncertainty Characterisation in Image Classification,” In *Proc. of 2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 2021, pp. 1-7, doi: 10.23919/FUSION49465.2021.9627071.
- [C2] T. Zhang, S. Wang, N. Bouaynaya, R. Calinescu and L. Mihaylova, “Out-of-distribution Object Detection through Bayesian Uncertainty Estimation,” In *Proc. of 2023 IEEE 26rd International Conference on Information Fusion (FUSION)*, 2023, pp. 1-8, doi: 10.23919/FUSION52260.2023.10224150.

Datasets

- [D1] S. Wang, J. Zhang, L. Mihaylova, and J. Law, “Human-Robot Video Data from a Manufacturing Factory”, [Online], Available on: <https://doi.org/10.15131/shef.data.19299539.v1>. [Accessed: 13-Jul-2023].
- [D2] J. Zhang, S. Wang, P. Wang, L. Mihaylova, and J. Law, “A vision data repository for human-ur10 robot interactions in manufacturing”, [Online], Available on: <https://doi.org/10.15131/shef.data.16669315.v1>. [Accessed: 13-Jul-2023]

Contribution to the Body of Knowledge in Robotics and Autonomous Systems

- [K1] P. Wang, S. Wang, J. Zhang, J. Law and L. Mihaylova, “2.6.1 – Monitoring RAS Operation”, CSI Cobot Demonstrator Project, [Online], Available on: <https://www.york.ac.uk/assuring-autonomy/guidance/body-of-knowledge/implementation/2-6/2-6-1/cobots/>

Chapter 2

Literature Review

This chapter of the thesis presents a comprehensive literature review on the intersection of Deep Learning and CV, crucial components of modern AI research. It begins with foundational knowledge in the field, followed by an exploration of how Deep Learning enhances CV tasks such as classification and object detection. The review delves into both fully-supervised and semi-supervised approaches, along with an analysis of uncertainty and activation pattern monitoring in Deep Learning. It further examines human pose estimation, covering 2D and 3D approaches and addressing the challenges of occlusions. The chapter concludes with insights into HRC, discussing the transition from simulation to reality, the role of Digital Twin in enhancing safety and resilience in manufacturing, and the specific application of robot-assisted dressing. This introduction sets the stage for a detailed exploration of these pivotal areas in AI research.

0

2.1 Background and the Development of Deep Learning and Computer Vision in Manufacturing

The field of CV, a vital subset of AI, has witnessed remarkable advancements in recent years. These developments have significantly enhanced the ability of machines to process and interpret visual data, leading to widespread applications in varied sectors such as healthcare, autonomous vehicles, robotics, and security [20]. The integration of CV in these domains not only represents a technological leap but also marks a paradigm shift in how industries operate and innovate.

For over four decades, the exploration of CV techniques in the manufacturing industry has been extensive. Its application spans diverse sectors including food, pharmaceuticals, automotive, aerospace, railway, semiconductor, electronic components, plastics, rubber, paper, and forestry [21]. Of particular interest in this exploration has been vision-based industrial inspection, which has become a critical component in the quality assurance and process optimisation of manufacturing operations [22].

The initial applications of CV in commercial manufacturing were somewhat limited, primarily due to the restricted computing capabilities that persisted until the 1990s [23]. This scenario, however, has dramatically transformed with advancements in semiconductor technology and computing power, which have consequently accelerated AI research and applications, especially in the last few years [24].

In image-based metrology, CV technologies have revolutionized measurement methodologies, transitioning from manual and error-prone approaches to automated, accurate, and reliable systems. This shift has notably enhanced quality assurance, reduced waste, and improved efficiency across various manufacturing sectors. Regarding manufacturing process interpretability, the integration of CV has facilitated real-time process monitoring and analysis. This advancement enables predictive maintenance, efficient fault detection, and optimized manufacturing processes, thereby enhancing overall op-

erational efficiency. In the realm of material structure analysis, CV has brought about a significant improvement in detecting and analyzing material defects that were previously undetectable by the human eye, thus ensuring higher product quality and reliability.

The continuous evolution of CV technologies is leading to their deeper integration into manufacturing processes, making them more pervasive and essential for modern manufacturing operations. This ongoing development is not just refining existing manufacturing capabilities but is also paving the way for innovative, efficient, and sustainable manufacturing practices.

In conclusion, this section underscores the transformative role of CV in the manufacturing sector and emphasizes the importance of ongoing research and development in this dynamic field. The future prospects of AI and CV in manufacturing promise increased automation, enhanced quality control, and the advent of intelligent manufacturing systems that are adaptive, efficient, and sustainable, which is critical for the continued evolution and competitiveness of the manufacturing industry.

2.2 Deep Learning and Computer Vision

The objective of a CV system is to create a symbolic representation of the objects present in a given scene. This representation encompasses an understanding of the scene and can subsequently be utilised to guide the subsequent operations of a robotic system. The CV field encompasses various tasks and algorithms, including detection, recognition, segmentation, and 3D reconstruction. In this section, an overview of the current advances in several significant CV techniques is presented.

2.2.1 Image Classification

Recognition task in CV refers to the process of identifying and categorising objects or patterns within images or videos. It involves training a machine learning model

to recognise specific objects or classes of objects based on their visual features. The goal of recognition tasks is to enable computers to understand and interpret visual information in a similar way to humans [20]. Accuracy, recall, precision, F1 Score, and ROC/AUC curves are commonly employed as evaluation metrics for recognition.

Class recognition does not focus on identifying a particular object. Instead, it aims to identify the presence of an instance belonging to a specific category of objects, such as cars or pedestrians. In class recognition problems, the input is an image, and the output is the classification of that image into one of the predefined categories. Class recognition is commonly approached as a classification problem, where machine learning algorithms, particularly convolutional neural networks (CNNs), are used to learn and classify images into different classes.

The advancement of CNNs has played a significant role in the progress of CV technologies. This breakthrough has led to the development of numerous CNN models, which have been widely used in classification problems. Over time, these models have become increasingly deeper. Some popular CNNs, listed in chronological order, include LeNet-5. [18], AlexNet [25], VGG-16 [26], R-CNN [27], Fast R-CNN [28], inception networks, ResNet-50 [29].

ResNet-50 [29] is a type of residual network that employs a unique structure that utilises residual connections or skip connections. This architectural innovation has led to significant improvements in Deep Learning models, particularly in complex tasks such as image classification. Prior to the introduction of ResNet-50, most CNN architectures focused on increasing the number of layers in the network, along with other necessary modifications to improve performance. However, this approach had a limitation in that the model's accuracy would plateau and then decline rapidly as the network became deeper. To address this issue, ResNet-50 introduced shortcut connections within its deep model. In the following years, ResNet and its variants have become one of the most important backbone for extracting features from images in the object detection networks. These connections allow inputs to bypass one or more layers and be added

back to the output of a layer further down the network. This design helps to address the vanishing gradient problem by facilitating the training of deeper networks. The core idea is that these skip connections enable the network to learn identity functions, ensuring that deeper models do not perform worse than their shallower counterparts.

2.2.2 Object Detection

Compared to image classification, object detection involves the identification and localisation of objects in an image or video. The goal is to not only classify the objects but also determine their precise locations by drawing bounding boxes around them. Object detection is widely used in various applications, including autonomous driving [30], surveillance systems [31], image and video analysis [32], and augmented reality [33]. It plays a crucial role in understanding and interpreting visual data by enabling machines to recognise and locate objects of interest.

One of the most famous series of object detection models is region-based CNN networks. RCNN stands for regions with CNN features. It is a landmark object detection model proposed in 2014 by R. Girshick [27]. RCNN introduced a two-stage approach to object detection. It starts by generating a set of object proposals using a selective search. Each proposal is then resized to a fixed size and fed into a pre-trained CNN model (such as AlexNet [25]) to extract features. Finally, linear SVM classifiers are used to predict the presence of an object within each region and recognise object categories.

SPPNet [34] introduced a Spatial Pyramid Pooling (SPP) layer, which allows a CNN to generate a fixed-length representation regardless of the size of the input image or region of interest. This eliminates the need to rescale the image or compute convolutional features repeatedly. By using SPPNet for object detection, the feature maps can be computed from the entire image only once, and fixed-length representations of arbitrary regions can be generated for training the detectors. SPPNet achieved a detection speed over 20 times faster than RCNN without sacrificing any detection accuracy, with a mean Average Precision (mAP) of 59.2% on the VOC07 dataset.

Fast RCNN [28] is a detector that was proposed as a further improvement of RCNN and SPPNet. It introduced several advancements in object detection. Fast RCNN allows simultaneous training of a detector and a bounding box regressor under the same network configurations. Instead of feeding multiple parts of the image into the CNN, Fast R-CNN processes the entire image with the CNN only once to create a feature map. From this map, it extracts fixed-size features from region proposals using a technique called ROI (Region of Interest) pooling. Fast RCNN achieved a detection speed over 200 times faster than RCNN.

Further, Faster RCNN [17] is developed as a two-stage object detection framework that combines a RPN with Fast RCNN. The main contribution of Faster RCNN is the introduction of the RPN, which generates region proposals in a nearly cost-free manner. The RPN shares convolutional layers with the Fast RCNN network, allowing for efficient computation. By integrating the proposal generation and object detection stages into a unified framework, Faster RCNN achieves a detection speed over 200 times faster than RCNN, with a mean Average Precision (mAP) of 70.0% on the VOC07 dataset. Faster RCNN has become a milestone in the development of object detection algorithms and has paved the way for further improvements in the field.

2.2.3 Fully-supervised and Semi-supervised Deep Learning for Object Detection

Numerous fully-supervised algorithms for object detection have been proposed, with one well-known series being the region-based convolutional neural networks, also known as two-stage detectors. This series includes R-CNN [27], Fast R-CNN [28], and Faster R-CNN [17]. In these two-stage approaches, the initial stage involves extracting image features using backbone networks such as ResNet [29]. The second stage generates region proposals for the subsequent localisation and classification of objects. Over time, the computation costs of region proposal generation in region-based convolutional neural networks have significantly decreased, transitioning from selective search [35] to

the RPN in Faster R-CNN [27]. The RPN offers real-time performance and has achieved notable improvements in detection accuracy.

The previously discussed object detection algorithms are classified as fully-supervised algorithms, which means that they require a large amount of labeled data for training. On the other hand, semi-supervised algorithms utilize a combination of labeled and unlabeled data, or pseudo-labeled data, to reduce the amount of required labeled data. Pseudo-label based approaches employ a teacher-student model, where a teacher model is first trained to generate pseudo-labels. These pseudo-labeled data, along with the unlabeled data, are then used to train the target student model. In the FixMatch algorithm proposed by Sohn [36], weakly-augmented data is used to generate pseudo-labels, and the same strongly-augmented images are used to predict whether they match the weakly-augmented ones. In another work by Sohn [37], pseudo-labels are generated using data augmentation, resulting in higher efficiency compared to the fully-supervised faster R-CNN algorithm [27]. Xu et al. [38] propose a soft teacher model that performs pseudo-labeling on weakly augmented data. The teacher model is updated using the student model, which employs an exponential mean average (EMA) strategy.

Prior research on semi-supervised methods [39] has shown that in order to train an effective detector, data pre-processing and data augmentation techniques are necessary. Furthermore, these methods typically focus on a single domain, where both labeled and unlabeled data are sourced from the same domain. However, this approach may lead to a decline in detection accuracy, particularly in unfamiliar environments. In contrast, the proposed semi-supervised object detection approach in Chapter 3 takes into account both the physical environment and simulation domains, enabling us to achieve satisfactory performance in novel real-world environments.

2.2.4 Uncertainty and Activation Pattern Monitorings

The importance of uncertainty in deep neural networks (DNNs) becomes evident when they are used in safety critical tasks. DNNs exhibit a surprising sensitivity to even minor variations in the input data, such as adversarial attacks, the presence of unseen objects, and occlusions. Such variations have the potential to cause failures in accurate and dependable decision making.

Several methods have been proposed to address decision making, such as Bayesian approaches [40, 41, 42], which aim to determine the most probable outcome based on probability. These approaches quantify uncertainty by generating an ensemble or using dropout during operation. While these methods overcome the challenges of directly implementing Bayesian inference, they are computationally expensive and do not perform well in real-time applications. Therefore, it is impractical to use Bayesian approaches in scenarios that require fast response times, such as autonomous driving and real-time tracking.

Verification problems of neural networks have been the focus of several works [43]. These works employ runtime verification algorithms to monitor the violation of correctness properties. Specifically, a series of methods have been proposed for monitoring activation patterns of neural networks [44, 45, 46, 47]. Cheng introduced a boolean abstraction method for monitoring deep neural networks (DNNs), where only the activation patterns of the final layers with respect to the ReLU activation function were considered. They were able to construct an efficient monitor using boolean logic operations and a binary decision diagram (BDD), which resulted in low computation costs for the MNIST dataset. The key idea behind monitoring activation patterns at runtime is the creation of a γ -comfort zone, which collects a sufficient number of activation patterns with correct predictions as the ground-truth. However, as the number of patterns, monitored neurons, and the abstraction parameter γ increase, storing patterns and monitoring computational costs become challenging, particularly in the context of object detection. In Chapter 5, this thesis focuses on addressing the issue of DNN

monitoring from the perspective of pattern distribution. Unlike Cheng’s approach of abstracting patterns using the Hamming distance, the Hamming distance is directly implemented as a distance metric to avoid storage issues and reduce the computational complexity associated with large ground-truth activation patterns.

2.3 Human Pose Estimation

Human pose estimation is a popular research topic in the field of CV. It involves predicting the joint locations of a human body from a single image or a sequence of images. This task has a wide range of potential applications, making it an active area of research. Recent advances in Deep Learning and the availability of large-scale datasets have enabled significant progress in 2D human pose estimation. However, 3D human pose estimation has not seen the same level of success, likely due to the lack of sufficient 3D in-the-wild datasets. Several methods [48, 49] have been proposed to address this issue, but there is still much room for improvement. The task has a wide range of applications, including tracking human movement and analyzing and detecting illegal or inappropriate behavior. In sports analysis, pose estimation can be used to automatically track and assess the accuracy of human movement. It is also a fundamental tool in fields such as human-computer interaction and augmented reality.

In the field of CV, there is a significant distinction between 2D and 3D pose estimation. The objective of 2D pose estimation is to predict the coordinates of body keypoints in a two-dimensional space. In simpler terms, the model determines the X and Y coordinates for each joint location. On the other hand, 3D pose estimation goes a step further by incorporating an additional Z-axis to infer the spatial position of the joints. Generally, 3D pose estimation is more challenging compared to its 2D counterpart. The development of an accurate and robust method for 3D pose estimation involves dealing with various limitations, including noisy background scenes, clothing, lighting conditions, small and barely visible joints, occlusions, and other factors that can sig-

nificantly alter the appearance of the body joints. This section aims to explore both the 2D and 3D human pose estimation fields.

2.3.1 2D (Two-Dimensional) Human Pose Estimation

2D human pose estimation refers to the task of detecting and localizing the key points or joints of a human body in a 2D image. It involves identifying the positions of body parts such as the head, shoulders, elbows, wrists, hips, knees, and ankles in the image. The goal is to accurately estimate the pose or body configuration of a person in the 2D space. This information can be used for various applications such as activity recognition [50], gesture recognition [51], HRC [52], and virtual reality [53]. Deep Learning techniques, particularly convolutional neural networks, have been widely used for 2D human pose estimation due to their ability to learn complex algorithms.

The challenge in 2D human pose estimation is to develop algorithms that are both highly accurate and efficient. High accuracy is important to ensure precise detection of human body information, which is crucial for downstream tasks such as 3D human pose estimation [54] and action recognition [55]. However, achieving high accuracy is challenging due to various factors. In real-world scenes, detection can be hindered by issues such as over- or under-exposure and the entanglement of people and objects [56]. Furthermore, the human body’s ability to move in a variety of ways and the occlusion of poses, including self-occlusion, make it difficult to accurately detect keypoints using visual features. Motion blur and video defocus in videos also reduce the accuracy of pose detection.

On the other hand, high efficiency is desired to enable real-time computing on different devices such as desktops and mobile phones [57]. However, there is often a trade-off between accuracy and efficiency. High accuracy models tend to be deeper, requiring increased computational and storage resources. This poses challenges in achieving real-time pose estimation, even with powerful GPUs.

There are two main frameworks in 2D human pose estimation: top-down and bottom-up. **Top-Down Framework** [58, 59, 60]: In the top-down framework, the approach starts by detecting human bounding boxes using an object detector. Then, a pose estimator is used to detect the keypoint locations within each bounding box. This framework relies on the accuracy of the object detector and the pose estimator. The object detector determines the performance of human proposal detection, while the pose estimator directly determines the accuracy of pose estimation. The top-down framework is scalable and can be improved with advancements in object detectors and pose estimators. **Bottom-Up Framework** [61, 62, 63]: In the bottom-up framework, the approach directly performs keypoint estimation in the original image without relying on human detection. This reduces computational overhead. However, a challenge in the bottom-up approach is determining the identities of the estimated keypoints.

In this chapter, Top-Down frameworks are mainly reviewed, focusing on their methodologies, key innovations, and applications in the field of 2D human pose estimation. This section begins by exploring the fundamental principles behind top-down approaches, including the initial step of human detection in images or video frames using advanced object detection algorithms. Various neural network architectures commonly employed in these frameworks are introduced, such as CNNs and R-CNNs, highlighting their roles in accurately identifying and localising human figures in diverse and complex environments.

2.3.2 3D (Three-Dimensional) Human Pose Estimation

The task of estimating the 3D locations of human body joints, known as 3D human pose estimation, can be approached as a regression problem. The goal is to predict the 3D coordinates of these joints from an image. Because it only requires an image as input, it faces challenges due to the absence of depth information.

The most straightforward approach for estimating the 3D pose is through direct methods, which involve training neural networks to estimate the 3D locations of human

body joints. In these methods, a likelihood heat map is predicted for each joint, and the joint location is determined by the maximum likelihood. Pavlakos et al. [64] proposed a method that predicts the voxel-wise likelihood for each joint in the 3D space and directly regresses the joint locations. Another approach by Luvizon et al. [55] uses a volumetric heat map to predict both 2D and 3D poses. The spatial designs of these heat maps, including their sizes and channels, play a crucial role in the accuracy of the predictions. However, they also significantly increase the computational cost and memory consumption.

Previous works [54, 65, 66] have taken 2D poses as inputs and predicted 3D poses. Zhao et al [67] introduced a graph convolutional network (GCN) that learnt local and global node relationships where a human pose skeleton can be represented by a graph. Typically, these approaches have a simple architecture and fast inference speed. However, their performance relies on the accuracy of the initial 2D pose estimation. SMPL-based methods apply a skinned multi-person linear (SMPL) model to predict 3D human joints. Such methods consider extra human body shape, providing more knowledge than traditional skeleton-based methods. Bogo et al [68] proposed a framework called SMPLify that applies a network to estimate 2D key points and maps the SMPL model to the predicted key points. Kanazawa et al. [69] proposed a network to map image pixels with SMPL models without auxiliary 2D key points. Likewise, Omran et al. [70] fitted 12 semantic parts of the human body from a semantic segmentation network to the SMPL models.

2.3.3 Occlusions in human pose detection

The presence of occlusion poses a significant challenge to the accuracy of Deep Learning methods, particularly when the object or person of interest is obstructed by another individual or object.

One method for dealing with occlusions is to make use of temporal data. In their study, Gu et al. [71] utilised a temporal regression network combined with a gated convolu-

tion module. This network is proficient in converting 2D joints to 3D and recovering occluded joints. Additionally, a localisation strategy is employed to transform the normalised pose into a global trajectory. In a similar vein, Cheng et al. [72] utilised the PoseFlow Tracker [73] to address the inconsistencies caused by occlusion, particularly in scenes with a moderate number of individuals. Ghafoor and colleagues [74] proposed a method for occlusion guidance that utilises binary values to indicate the absence of joints or joints with low confidence during 2D pose estimation. In the process of 3D pose estimation, the system takes into account both the 2D joints and occlusion guidance. In contrast, Liu et al. [75] divided pose estimation into two parts: detecting visible keypoints and understanding the reasoning behind occluded keypoints. They introduced the Deeply Supervised Encoder Distillation (DSED) network as a solution for occlusion reasoning. The DSED network boasts a dual-encoder structure: one encoder adopts a mentorship role, cherry-picking the most salient information requisite for the reconstruction of occluded joints, while its counterpart is trained to cull analogous information from observable cues. Evidently, the DSED network exhibits superior prowess in discerning occluded joints relative to the rudimentary hourglass model. Crucially, undertaking occlusion reasoning at the feature stage—prior to pose compilation—enhances the technique’s aptitude for multi-person contexts.

The techniques mentioned above partially neglect an essential aspect: the human body can be seen as a quasi-rigid structure, with strong mechanical connections between each joint.

2.4 Human-Robot Collaboration

Human-Robot Collaboration (HRC) is a multidisciplinary field at the intersection of computer science, engineering, social science, and psychology [76]. It focuses on understanding, designing, and evaluating robotic systems for use by or with humans. The emergence of HRC as a distinct field reflects the increasing sophistication and

variety of robots and their integration into human environments, from industrial automation to assistive companions. This introduction establishes the context for HRC, addressing the technical challenges, human-centered design considerations, and ethical implications associated with the coexistence and collaboration of humans and robots.

2.4.1 From Simulatuiou to Real (Sim2Real)

Collecting and annotating large amounts of data for training Deep Learning models can be a costly endeavor. This is particularly true when it comes to training models in new environments, such as manufacturing settings, where there is a need for a Deep Learning-based detector to identify robots and humans in HRC (human-robot collaboration). Unfortunately, there is currently no publicly available dataset specifically designed for training Deep Learning models in such scenarios. However, one potential solution to this problem is the use of Digital Twin technology. By simulating various scenarios in a digital system, Digital Twin can generate a significant amount of labeled data, which can then be used for training purposes [77, 78]. The data generated can be utilised for training Deep Learning models and implemented in real-world settings. Techniques referred to as Sim2Real [79, 80, 81] can be employed in these tasks. Moreover, they allow for the utilisation of solely the (simulated) virtual environment during the training, validation, and testing phases of the deep neural network (DNN) models. Nevertheless, there are instances where these models tend to exhibit inaccuracies when evaluated in real-world applications, primarily due to the disparities between the simulated virtual world and the actual world.

The primary aim of the study conducted by Tobin et al. [82] is to identify objects on a table in a real-world setting and determine their positions. In order to ensure the applicability of the trained model from the simulator to the physical environment, Tobin et al. [82] introduced randomisation in terms of distractors, objects, backgrounds, and lighting conditions. The model was directly trained on the simulator and achieved accurate estimation of the position of different objects with shape-based characteristics

on the table in the real world. In the domain of object detection, Tremblay et al. [83] utilized similar approaches as those presented in [82] for the purpose of identifying real objects within intricate backgrounds. In contrast to the technique proposed in [82], they introduced a novel element known as flying distractors, which enhances the precision of detection. In addition, the significance of each randomisation parameter was investigated by Tremblay et al. In their study, the simulation model uniformly randomises environment parameters during the training process. However, the complexity of the samples increases as the number of randomisation parameters increases [82, 84, 83]. Identifying the causes of failures during this randomization process proves to be challenging. To address these issues, Mehta et al. [85] determine the most informative variations in the environment within the given range of randomisation parameters.

In this thesis, the proposed Digital Twin utilises Domain Randomisation in its digital system. A straightforward and effective method is employed to generate the synthetic dataset. The digital system closely resembles the physical system, with the exception of the randomisation parameters. Additionally, while previous studies focused solely on synthetic data, the approach also incorporates unlabelled real data to reduce the disparity in Sim2Real [79, 80, 81].

Although DNNs have achieved state-of-the-art results in tasks like object detection and segmentation [86], they are often criticised for their reliance on data and computational resources. Popular public datasets such as COCO [87], PASCAL VOC [88], and ImageNet [89] have been designed specifically for tasks like object detection and semantic segmentation. DNNs trained on these datasets have even outperformed humans in tasks like object recognition. However, when DNNs are presented with objects that are not included in these datasets, their performance can significantly deteriorate. One possible solution to mitigate this problem is to expand the datasets, but this approach is inefficient as it requires manual data preparation. Alternatively, researchers can leverage powerful simulation platforms like Unity3D [90] to automatically generate the required data. This approach, known as "Sim2Real" techniques, has gained signif-

icant attention and holds promise for efficient and flexible data preparation for Deep Learning.

By incorporating simulation within the iteration, it becomes feasible to generate a significant volume of data that includes annotation information, as needed. Moreover, this approach allows for exclusive reliance on the simulated virtual environment to train, validate, and test DNN models. Promising outcomes have been demonstrated by some studies when transferring these models, trained in the virtual realm, to real-world applications [79]. Nonetheless, there are instances where these models exhibit subpar performance when evaluated in real-world scenarios, primarily due to disparities between the simulated virtual environment and the actual physical world.

Bridging the gap between physics simulators and the real world poses a significant challenge. The objective of the Sim2Real problem is to transfer virtual models into real-world scenarios. Currently, numerous research efforts are dedicated to addressing this reality gap. One approach involves the use of high-quality rendering simulators such as Unity3D [90], Unreal Engine 4 [91], and OpenGL [92]. These simulators can generate realistic simulated images that closely resemble data from the physical world. Additionally, two other strategies, namely Domain Randomisation (DR) and Domain Adaptation, have been proposed to tackle this challenge.

A domain is defined as \mathcal{D} , where a feature space $\mathcal{X} \subset R^d$ with d dimensions, along with a marginal probability distribution $P(\mathbf{X})$. In this domain, \mathcal{T} is defined as a task. Given a training set $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \mathcal{X}$ and its corresponding labels $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ from the label space \mathcal{Y} , the conditional probability distribution is denoted as $P(\mathbf{Y} | \mathbf{X})$.

It is assumed that there are two domains: the source domain, represented by a simulator $\mathcal{D}^s = \{\mathcal{X}^s, P(\mathbf{X}^s)\}$, with a task $\mathcal{T}^s = \{\mathcal{Y}^s, P(\mathbf{Y}^s | \mathbf{X}^s)\}$, and the target domain, represented by the physical world $\mathcal{D}^t = \{\mathcal{X}^t, P(\mathbf{X}^t)\}$, with its corresponding task $\mathcal{T}^t = \{\mathcal{Y}^t, P(\mathbf{Y}^t | \mathbf{X}^t)\}$.

Domain Randomisation

Tobin et al. [82] introduced a technique for training models on simulated images, specifically in the source domain \mathcal{D}^s . The key idea is that by randomising rendering in the simulator, the model can be adapted to real images in the target domain \mathcal{D}^t . The authors assume that a set of randomisation parameters can be controlled in the simulator. If the variability of the simulator is diverse enough, the physical world can be seen as another variation in the simulator, i.e. $\mathcal{D}^t \subset \mathcal{D}^s$ and $P(\mathbf{Y}^t | \mathbf{X}^t) \subset P(\mathbf{Y}^s | \mathbf{X}^s)$. As a result, the model trained in the simulator can generalise to the physical world without the need for additional adjustments during training.

The primary objective of Tobin et al. [82] is to identify objects on a physical table and estimate their positions. The authors demonstrate the use of simulated data randomisation during the training process. In order to ensure the transferability of the trained model from the simulator to the physical world, Tobin et al. [82] implemented randomisation in various aspects such as distractors, objects, backgrounds (including the table, floor, and robot), and lighting conditions. The model was directly trained on the simulator and successfully estimated the positions of objects with different shapes on the table in the physical world. In the context of object detection, Tremblay et al. [83] employed similar strategies as Tobin et al. [82] to detect real objects in more complex backgrounds. In addition to Tobin et al.’s method, they introduced a new component, flying distractors, which improved the accuracy of detection. Furthermore, they investigated the significance of each randomisation parameter. Sadeghi and Levine [84] combined deep reinforcement learning with their approach. They trained a vision-based control model for a quadrotor entirely in simulation, as a trial-and-error learning process is challenging in the physical world. The network successfully produced a collision-free flight.

In the training process, the environment parameters are randomly assigned in simulation. However, as the number of randomisation parameters increases, the sample complexity also increases [82, 84, 83]. Additionally, it is difficult to determine the exact

cause of failure when the transfer does not work. To address these issues, Mehta et al. [85] propose Active, which identifies the most informative environment variations within a given range of randomisation parameters.

In their work, James et al. [93] introduced an alternative approach called Randomised-to-Canonical Adaptation Networks (RCANs) to address the issue of increasing sample complexity. They utilized a cGAN, which is an image-conditional generative adversarial network, to convert random simulated images into a canonical form. Additionally, the generator component of the cGAN was able to transform real-world images into the canonical form after training. The researchers trained a vision-based closed-loop grasping reinforcement learning agent in a canonical simulator and then transferred it to the physical world.

Peng et al. [94] explored the concept of randomisation in dynamic systems of robots, in addition to randomising scene properties in simulators. Unlike high-fidelity rendering in simulation, they achieved success in developing a Sim2Real policy using low fidelity simulations. The parameters that were randomised include the mass of the robot's links, damping of joints, gains for controllers, and noise of observation, among others. This policy demonstrated the ability to adapt to various physical dynamics.

Domain Adaptation

According to Pang and Yang [95]'s classifications of transfer learning, Domain Adaptation can be regarded as a transductive transfer learning solution in which a set of labelled data is trained in source domain \mathcal{D}^s to learn a model to classify the unseen data in a target domain \mathcal{D}^t [96]. Base on the definition of transfer learning in [95], target of target domain \mathcal{D}^t is the same as that of source domain \mathcal{D}^s , i.e. $\mathcal{T}^t = \mathcal{T}^s$, while $\mathcal{D}^s \neq \mathcal{D}^t$. When it comes to label space, they shared the same label space, i.e. $\mathcal{Y}^s = \mathcal{Y}^t = \mathcal{Y}$ in a classification task.

Domain Adaptation has been proved as a successful method in bridging the reality gap.

One way to bridge the gap is to adopting a adversarial-based deep Domain Adaptation approach [97] which is based on generative adversarial networks (GANs) [98]. GAN is constructed by a generative model G and a discriminative model D . G extracts the data distribution, while D outputs a label whether a sample is from G or training datasets.

The GAN framework is trained with a mini-max function, and the generative model G is optimised to gain minimum loss while the discriminative model D can be trained to maximise the probability of the correct label:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (2.1)$$

where x is the input of D and z is the input of G .

Liu et al. [99] put forward a coupled framework CoGAN to generate target data which are coupled with synthetic source ones.

It illustrates the coupled framework of coGAN: GAN_1 generates source data while GAN_2 produces target data. The weights is shared in some layers of G and D which provide constraints for realising a a domain-invariant feature space with no supervision. Aiming to use the shared labels of synthetic target data to train the target model, the input noise should be adapted by a train coGAN to paired synthetic images from the two distributions and the labels also be shared.

Currently, generating synthetic data with annotations as the source that resemble the target data have been an interesting field of research. Yoo et al. [100] presented a synthetic data generation model based on pixel-pixel level adaptation in the GAN framework. They employed a real/fake discriminator to supervise the generation of realistic targets while the discriminator penalises an unrealistic target. Besides, a domain-discriminator is designed to ensure a generated target is associated to a source. A Simulated+Unsupervised (S+U) learning is proposed by Shrivastava et al. [101] to reduce the gap between synthetic and real image domains. In their model, unlabelled

real data are used to improve realism of simulator. Rather than directly using GANs framework, they introduced an adversarial network which inputs synthetic data rather random vectors. To improve realism, a refiner network is trained with optimising a combination of an adversarial loss and a self-regularisation loss. The advantages of [101] are avoiding artifacts and stabilising training. Different from some works where the generator is constrained by a noise vector or source images, Bousmalis et al. [102] presented a framework that the output of the generator G are conditioned on synthetic source data and a noise vector. Decoupled from classification task, the classifier T independently assigns a label to an image aside from the process of domain adaptation, while the discriminator D identifies real and fake images. Considering prior knowledge about low-level image adaptation process, they tried to maintain the generated images have similar foregrounds and different backgrounds from the source. With G , D and T , their optimisation becomes

$$\min_{G,T} \max_D V(D, G) = \alpha \mathcal{L}_d(D, G) + \beta \mathcal{L}_t(T, G) + \gamma \mathcal{L}_c(G) \quad (2.2)$$

where α , β , and γ are parameters that control the trade-off between the losses and $\mathcal{L}_c(G)$ decides the similarity described above, named the content-similarity loss.

2.4.2 Digital Twin for HRC Safety and Resilience in Manufacturing

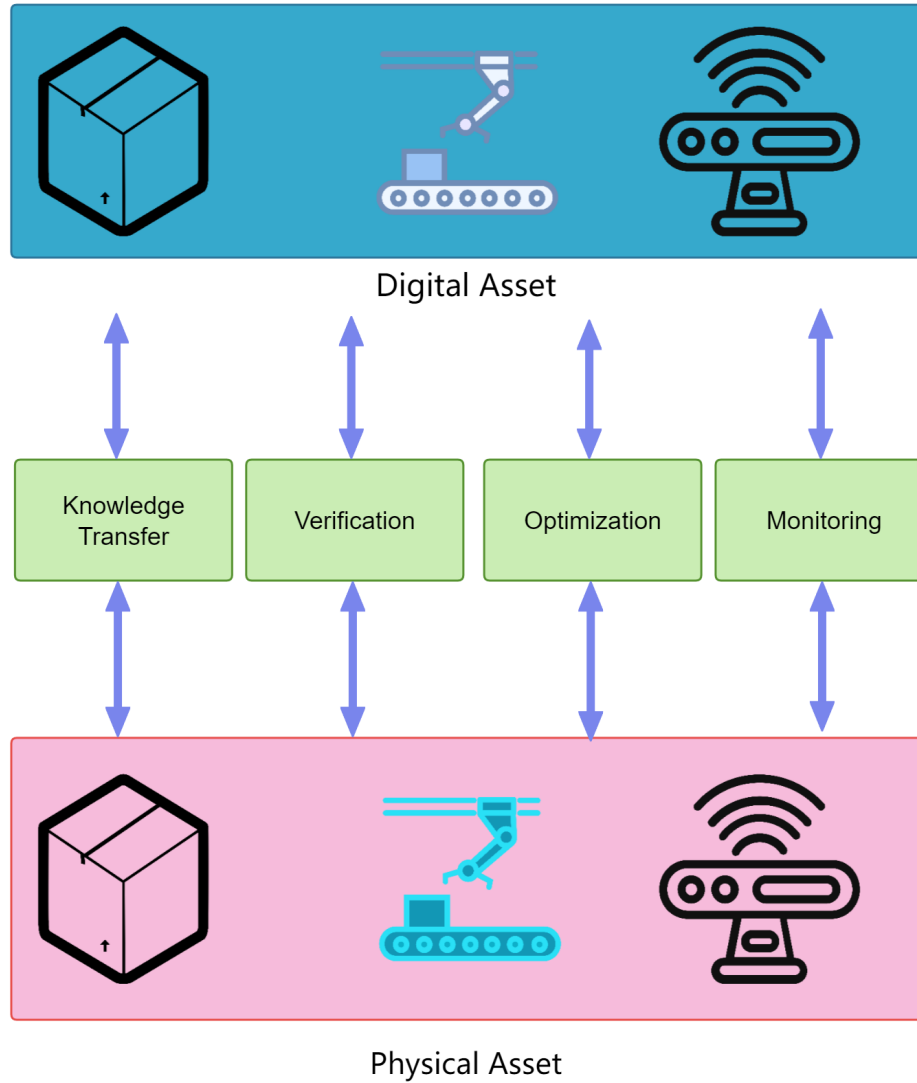


Figure 2.1: A Digital Twin process facilitates the transfer of knowledge between Digital and Physical entities. The model is validated prior to the deployment of the physical asset in the actual environment. Moreover, feedback from the Physical asset contributes to the optimization of the entire process. The Digital Twin is also capable of tracking the performance of the physical asset.

A Digital Twin is a virtual model of a physical entity that exists parallelly in the real world as shown in Fig 2.1. It is created by using real-world data to simulate and predict the behavior of the physical object. Digital Twin have applications in various industries such as manufacturing, healthcare, and smart cities [103]. In the field of medicine and public health, Digital Twin technology can transform traditional electronic health records and enable personalised treatments and interventions [104]. Digital Twin are both a digital shadow reflecting the status of the physical twin and a digital thread recording its evolution over time [105]. They can be used to understand complex systems, conduct in silico experiments, and support evidence-backed decision-making [106]. However, the development of Digital Twin faces challenges such as data communication, lack of standardised methodologies, and the need for interdisciplinary collaboration [107].

Simulation models have various functions in industry, including product design, testing, and delivery. However, the dynamic nature of demands, the requirement for real-time process monitoring, and the need for cost-effective production present new obstacles for simulation techniques [108]. Digital Twin have gained considerable interest from industry and academia, as they go beyond traditional simulation methods by incorporating real-time and historical data from their corresponding physical systems [109, 110]. This is particularly relevant in the context of production lines where humans and robots share workspace. Digital Twin have the ability to integrate both cyber and physical data throughout the entire lifespan of a product. They are widely acknowledged as a highly promising tool for the design, maintenance, and monitoring of smart manufacturing processes [109]. With the advancements in artificial intelligence, cyber-physical systems, big data, information fusion, and advanced sensing, the field of Digital Twin technology is rapidly evolving and transforming the manufacturing industry towards intelligent human-robot collaboration [111].

Sudhakar et al. [12] Investigate a method for utilising data generated by a Digital Twin to train a CV model. and they discuss the challenges of using synthetic data in training

CV models and aims to understand the critical aspects of the authoring process that impact model performance. The authors create a novel YCB-Real dataset by capturing images of YCB objects and a corresponding synthetic dataset, YCB-Synthetic, to study the effects of various artifacts on model performance. They analyse the trade-offs between artist time for fixing artifacts and model accuracy, providing insights on prioritising efforts in synthetic data generation. In contrast to the conventional approach of manual labeling, which is frequently laborious and time-consuming, this method greatly speeds up data collection by automating the labeling process. The Digital Twin enables faster gathering and processing of extensive datasets, making data management more efficient and scalable across a range of applications.

Malik and Brem [112] present a framework that utilises a Digital Twin to enhance industrial assembly systems. By incorporating the human presence, the Digital Twin effectively captures the system’s adaptability and dynamics, leading to enhanced safety in HRC. In their work, they tackle the growing complexity of contemporary manufacturing settings that urgently require adaptability, flexibility, and economic efficiency. The paper criticises existing automation technologies for their lack of human compatibility and their inability to co-exist harmoniously with humans, resulting in a continued heavy reliance on the human workforce for many operations. They further investigated the interaction between humans and robots via a Digital Twin, facilitating more instinctive communication methods, including hand gestures and smartwatches. This strategy improves safety by accurately predicting possible collisions and assists in high-variety, low-quantity production by allowing the robot to quickly adjust to changes in task performance without hindering human tasks. The authors of [113] suggest the use of a Digital Twin with machine learning capabilities as a testbed for evaluating a Deep Learning model for path planning. This approach is especially advantageous in situations where human validation is necessary, as any unresolved problems could pose a risk of harm to individuals. Moreover, the integration of Deep Learning techniques into an immersive AR setting allows for the mapping of virtual and physical objects

during interactive multi-functional tasks, resulting in enhanced visualisation of target objects [114]. Park and colleagues [115] have developed a hands-free interaction system in mixed reality environments, leveraging a Digital Twin for assistance.

Unlike previous research, the proposed Digital Twin has the capability to support the training of Deep Learning models by producing training datasets alongside testing and validating the model. This improves the efficiency of training the Deep Learning model in terms of both time and labor expenses.

2.4.3 Robot-assisted dressing

In the past few years, there has been significant interest in the area of robot-assisted dressing. This interest has been mainly motivated by the need to address challenges such as a shortage of nursing staff and the increasing number of elderly people. Various techniques have been developed that employ force sensors to enable robots to assist with dressing tasks. Erickson et al. [116] proposed a deep recurrent neural model that aims to predict the forces applied by a piece of clothing on the human body. This prediction is based on observations of haptic and kinematic data collected from the robot's end effector. In a similar vein, Clegg et al. [117] combined haptic feedback control and deep reinforcement learning to facilitate robot-assisted dressing. They utilised physics simulations to emulate different types of human impairments, which formed the basis for training control policies for both humans and robots.

On the other hand, visual sensors provide a cost-effective and easy-to-implement solution. Many research studies have utilised RGB or RGB-D cameras to determine points of contact and describe the interactions between humans and objects. [118, 119, 120]. Pignat et al. [121] introduced a significant advancement by proposing a hidden semi-Markov model. This model combines sensory data from the human user and motor commands from the robot to create a joint representation. In order to assist the robot in learning from human-led dressing demonstrations, the researchers utilised an AR tag to track the movement of the human hand.

2.5 Conclusions

This chapter is a review of the literature that focuses on the intersection of Deep Learning and CV, crucial components of modern AI research. It starts with foundational knowledge in the field and then explores how Deep Learning enhances CV tasks such as classification and object detection.

Table 2.1 summaries perception research associated with HRC, it reviews different kinds of method for better detection results both in accuracy and speed. However, though these methods had already achieved strong performance on public datasets, there still exists a significant gap between research and the real HRC environment. For example, challenges such as unseen objects, unfamiliar environments, new requirements for detection, and the ability to generalise across various real-world scenarios remain pressing issues. Additionally, the complexity of integrating these methods into existing HRC systems and ensuring they meet industry standards for reliability and safety adds further layers of difficulty.

The review delves into both fully-supervised and semi-supervised approaches, along with an analysis of uncertainty and activation pattern monitoring in Deep Learning. It also examines human pose estimation, covering 2D and 3D approaches and addressing the challenges of occlusions.

In table 2.1, it also summarises 2D and 3D human pose estimation methods. For 3D human pose estimation, though direct estimation solutions can achieve 3D prediction from images directly, the performance is not satisfying. Although SMPL-based methods can provide accurate predictions, they require huge computational resources and it is hard to achieve a real-time performance. In Chapter 4, the estimation of the 3D human pose is achieved by lifting the 2D human pose. Chapter 4 explores a post-processing scheme without involving temporal information during training stage but still can achieve strong prediction when occlusion occurs.

With respect to the gap between simulation and real environment in Digital Twin,

several methods have been summarised in [2.2](#), Domain Adaptation relies on training additional transfer Deep Learning model to achieve the knowledge transform between simulation and real environment which can be regarded as an indirect solution. However, Domain Randomisation learns directly from the simulation environment without extra transfer models. Moreover, powerful Digital Twin can provide photo-realistic simulation environment, it can naturely minimise the gap between simulation and real environment.

The chapter concludes with insights into HRC, discussing the transition from simulation to reality, the role of Digital Twin in improving safety and resilience in manufacturing, and the specific application of robot-assisted dressing.

Table 2.1: *Summary of perception research under HRC*

Perception in HRC	Application Types	Methods	Method Descriptions
Detection	Image Classification	LeNet [18], AlexNet [25], VGG [26]	Multiple layers
		ResNet [29]	Residual Network that utilises residual connections or skip connections
	Object Detection	RCNN [27]	Region-based CNN network; Linear SVM classifiers are used to predict the presence of an object
		SPPNet [34]	Spatial Pyramid Pooling layer
		Fast RCNN [28]	ROI (Region of Interest) pooling
		Faster RCNN [17]	Combines a RPN with Fast RCNN.
2D Human Pose	Top-down Framework	PoseWarper [58]	Learns temporal pose estimation from sparsely annotated videos
		RMPE [59]	Symmetric Spatial Transformer Network (SSTN) to extract high-quality dominant human proposals;
			Parametric Pose Non-Maximum Suppression (NMS) to eliminate redundant pose estimations;
	Bottom-Up Framework	Liu et al. [60]	Pose-Guided Proposals Generator (PGPG) to handle inaccurate bounding boxes and redundant detections
			Pose Temporal Merger for encoding keypoint spatiotemporal context;
			Pose Residual Fusion module for computing weighted pose residuals in dual directions;
			Pose Correction Network for refining pose estimations effectively
3D Human Pose	Direct Estimation	Cao et al. [61]	Part Affinity Fields (PAFs) to associate body parts with individuals in an image.
		higherHRNet [62]	High-resolution feature pyramids for scale-aware representation learning.
		Luo et al. [63]	Scale-adaptive heatmap regression (SAHR) method to adjust the standard deviation for each keypoint; Weight-adaptive heatmap regression (WAHR) to balance the fore-background samples
	2D to 3D Lifting	Pavlakos et al. [64]	A fine discretization of the 3D space; A coarse-to-fine prediction scheme
		Luvizon et al. [55]	A multitask framework for joint 2D and 3D pose estimation; Differentiable Soft-argmax for joint pose estimation
		SimpleBaseline3D [54]	A deep end-to-end framework for 3D pose estimation from 2D joint detections
		PoseFormer [122]	A spatial-temporal transformer-based approach for 3D human pose estimation
	SMPL-based	Graformer [66]	A novel transformer architecture combined with graph convolution
		SemGCN [67]	Semantic Graph Convolutional Networks (SemGCN) for regression tasks with graph-structured data
		SMPLify [68]	An interpenetration term that is differentiable concerning shape and pose
	Temporal Fusion	Kanazawa et al. [69]	An interpenetration term that is differentiable concerning shape and pose
		Omran et al. [70]	A statistical body model integrated within a CNN for 3D human pose estimation from 2D images
Occlusions in Human Pose Estimation	Temporal Fusion	Gu et al. [71]	A temporal regression network with a gated convolution module to transform 2D joints to 3
	Optical-flow	Cheng et al. [72]	An occlusion-aware deep-learning framework for 3D human pose estimation in videos
	Pose-flow	PoseFlow [73]	An efficient pose tracker based on pose flows.
	Occlusion-guided; Temporal	Ghafoor [74]	An occlusion-guided framework for 3D human pose estimation; Temporal dilated CNNs to handle severe occlusions effectively
	Skeleton-guided	Liu et al. [75]	A Skeleton-guided human Shape Fitting (SSF) method for generating accurate occlusion labels

Table 2.2: *Summary of Sim2Real research under Digital Twin*

Sim2Real in Digital Twin	Methods	Method Descriptions
Domain Randomisation	Tobin et al. [82]	Non-realistic random textures in a simulator for training a robust real-world object detector
	Tremblay et al. [83]	randomizing simulator parameters like lighting, pose, and object textures
	Sadeghi and Levine [84]	Domain Randomisation with 3D CAD models in Reinforce Learning environment
	Mehta et al. [85]	Active Domain Randomisation for selecting the most informative environment variations
	James et al. [93]	RCANs translate randomized rendered images into non-randomized canonical versions
	Peng et al. [94]	Randomisation scheme during training process
Domain Adaptation	Ganin et al. [97]	A GAN-based framework transferring knowledge from the source domain to the target domain
	Liu and Tuzel [99]	Coupled Generative Adversarial Network (CoGAN) for learning a joint distribution of multi-domain images
	Yoo et al. [100]	An image-conditional image generation model for knowledge transfer at a semantic level; Generates the target image at a pixel level.
	Bousmalis et al. [102]	Unsupervised pixel-level domain adaptation using Generative Adversarial Networks (GANs)

Chapter 3

A Deep Learning-enhanced Digital Twin Framework in HRC

3.1 Introduction

Collaborative robots (cobots) [123] are playing an increasingly important role in the smart manufacturing and Industry 5.0 era, as they have the potential to boost productivity, ensure safety, and liberate humans from labor-intensive activities [124, 112, 125]. Benefiting from the desirable productivity and precision through a series of repetitive tasks conducted by machines along with the flexibility of manual operations, cobots have shown their great potential to realize smart manufacturing, including flexibility and perform repetitive tasks. Examples include but are not limited to hazardous and extreme working environments such as quality inspections, machine tending, material handling, welding, and drilling.

The concept of HRC in Industry 5.0 is mostly conveyed by smart manufacturing where cobots work alongside humans in close proximity in a shared workspace and they are pre-programmed to interact with humans to carry out various tasks. However, human safety is a key prerequisite for the deployment of such robots. Traditional approaches to

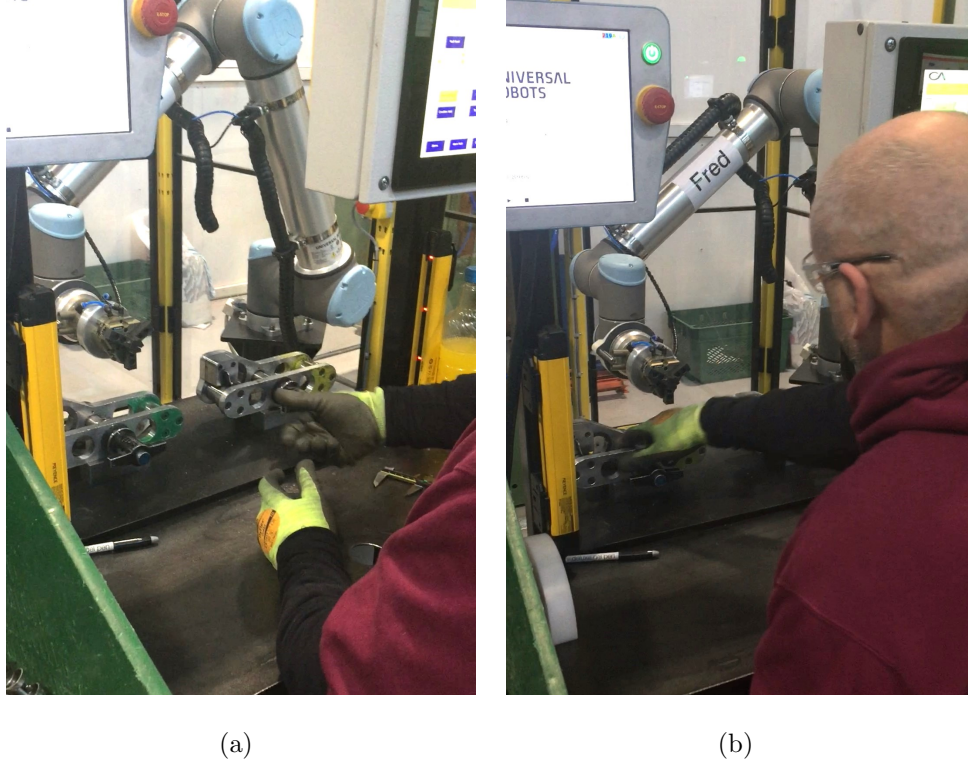


Figure 3.1: *Fig 3.1(a) and 3.1(b) shows the configuration of an industrial HRC process, where an operator exchanges components with a cobot at a shared handover location. The robot cell is open on one side, allowing staff to enter the cell under specific circumstances*

ensure robot safety in manufacturing require deployment of cages, as shown in Fig 3.1. Physical barriers, light gates, and laser rangefinders prevent direct contacts of cobots and humans [126]. These safety measures protect human workers, but they are bulky, inflexible (preventing true collaboration), and expensive.

In recent years significant research has been carried out to develop cage-free and more flexible safety solutions. Collision avoidance based solutions have been proposed in [127, 128, 129], where the pre-programmed trajectory of cobots are adapted to avoid collisions with dynamic obstacles, e.g., humans and other objects in the shared workspace. Unfortunately, these solutions lack the ability to distinguish ‘humans’ from

other objects, which could subsequently cause severe consequences. In addition, these solutions rely on the alignment of digital cobots designed by Computer-Aided Design (CAD) tools [130] to re-built digital cobots from Red, Green, Blue plus Depth (RGB-D) camera data. CAD models of cobots were combined with the data captured by RGB-D sensors. This leads to an easy separation of robots from surrounding objects and also from humans. The alignment between a CAD model and the caged cobot is typically done with the assistance of hand-eye calibration [130, 131]. However, the calibration quality is critical in determining the accuracy of alignment.

Besides CAD models, augmented and mixed reality techniques which integrate computer-generated virtual information into real-world scenes can help users to enhance their understanding and awareness to support safe interaction in HRC tasks [132, 133, 134]. Meanwhile, thanks to the rapid development of deep learning and CV techniques, a series of modern approaches have been proposed [135, 136], demonstrating success in scene understanding and visual perception, such as classification, object detection and segmentation.

Furthermore, Digital Twin of cyber-physical systems provide a real-time digital representation of physical collaborative manufacturing systems. This can greatly improve the systems' intelligence regarding design, production, operation, evaluation, health management and performance optimization [125, 137]. Digital Twin can contribute to a range of different aspects in challenging HRC systems [77], including to simulation, modelling, performance analysis, process monitoring, data collection, data mining, data fusion, interaction as well as cognitive service [111, 109]. This makes Digital Twin and intelligent solutions promising in avoiding the complex calibration process and in achieving identification of cobots and other objects in HRC without calibration at all.

Different definitions [111] of Digital Twin have been proposed and developed over time. According to [110], a Digital Twin is a set of coupled computational models and methods that evolve over time to persistently represent the structure, behaviour, and context of a unique physical asset such as a component, system or process. A

Digital Twin represents a real system, e.g. a city, cobot, aircraft, and acts as a coupled duplicate of the real world. It has several important characteristics: i) it is **universal** and can be applied to several domain areas, ii) it has a **modular structure**, which can be updated, expanded and developed further, 3) it is **connected with data** - both computer generated and from the real system. It can be used for a number of purposes – design, increasing safety and autonomy, and others, including for new functionalities. Fu et al. [111] point out four stages in the development of Digital Twin, with an increasing usage of data in the last two stages, including remotely, when data could be stored on a cloud and accessed via the IoT technologies. The surveys [138, 139] systematically review the recent developments of artificial intelligence-driven Digital Twin in the areas of cutting-edge robotics and smart manufacturing. Besides, multi-access edge computing was incorporated into Digital Twin, facilitating manufacturing processes towards smart and flexible [140, 141].

Having in mind these recent trends [142, 78], one can identify several gaps between the research in Digital Twin techniques and their applications in industry: i) Digital Twin need further developments in order to represent manufacturing systems in a wide range of complex environmental conditions and diverse production stages, ii) in the majority of cobot systems, safety is guaranteed via caged environments or additional safety sensors when the cobots are operated at higher speeds so as to meet production demands of end users, iii) the level of autonomy varies across different applications and is on the increase thanks to recent developments in intelligent sensing, CV and artificial intelligence techniques.

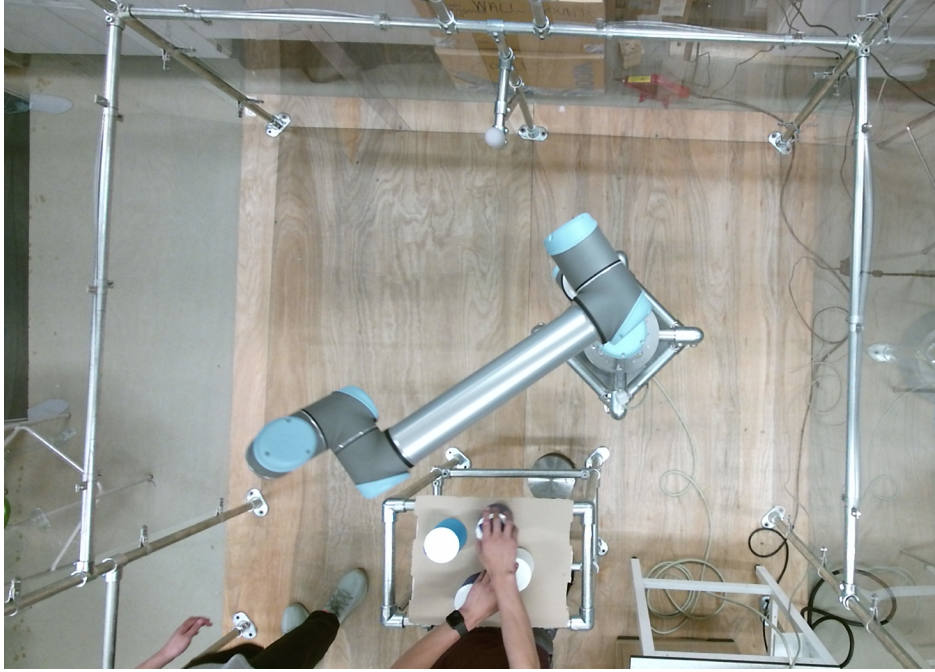


Figure 3.2: *A HRC cell is available in the Sheffield Robotics Lab at the University of Sheffield, UK. An HRC cell is shown in this picture, where there is an operator desk in front of the cobot and the operator exchanges components with the cobot on the desk. A Kinect sensor is mounted on the top of the cell to monitor the HRC operation.*

Aiming at contributing towards bridging these gaps, this chapter proposes an intelligent Digital-Twin-based safe human-robot collaboration framework. A Digital Twin is built to simulate the physical HRC system which is shown in Fig 3.2. A communication framework is further designed so that the Digital Twin can be synchronised with the physical HRC platform with the support of the Robot Operating System (ROS) [143]. Consequently, information including robot poses and kinematics can be shared between the digital and the physical systems flexibly and in a real-time manner. Owing to the Digital Twin's ability to create photo-realistic digital cobots and maintaining holistic cobot parameters, a diverse amount of synthetic cobot data with accurate labels are generated by the digital system. These data combined with human data from the COCO repository [87], are used to train deep learning models to monitor interactive

operations of robots and humans. The challenges stemming from the simulated Digital Twin environment and the real environment are addressed by further proposing a semi-supervised deep learning detector. The Digital Twin system is applied to analyse and validate how the environment, e.g. the lighting conditions, affect the performance of the deep-learning action-recognition system. With the proposed deep learning detector, humans and robots are monitored in the physical environment to ensure their safe separation. Therefore, by adopting a Deep Learning-enhanced Digital Twin Framework, this work contributes toward cost-effective and flexible systems for intelligent sensing and decision making.

The main contributions of this work are as follows: i) a semi-supervised framework for object detection is proposed by adopting a Faster region-based convolutional network [17]; ii) a Digital Twin of a physical HRC system is developed that generates synthetic robot data to train deep learning models for monitoring human-robot collaborative behaviors. iii) the performance of the developed Digital Twin system is validated and evaluated over both synthetic and real data sets, demonstrating that it can achieve accurate recognition of human-robot behaviors for safety assurance. Research outputs include publicly available datasets generated by the proposed Digital Twin of a Universal Robot 10 (UR10) robot, and a semi-automated annotation tool.

The remainder of this chapter is organised as follows. Section 3.2 describes the developed framework for safe and reliable HRC in detail. Section 3.3 describes the real and synthetic datasets along with the semi-automated annotation tool used in this work. Section 3.4 presents evaluation and validation of the detection and classification results under different lighting conditions, whilst explaining safety criteria for decision making and demonstrating how to implement or adopt the proposed framework into practical cases. Finally, Section 3.5 summarises the results and make a conclusion.

3.2 The Deep Learning-enhanced Digital Twin Framework

Traditional solutions to prevent hazardous human activities with cobots include physical safety barriers, proximity sensors, and light gates, which have major disadvantages of big size, difficult maintenance, inability to adapt under various operating conditions, and sometimes high cost [144, 145]. To meet the high requirements for cobots towards safety and reliability, this chapter proposes an intelligent and flexible deep learning-enhanced Digital Twin framework for monitoring the human-robot collaboration with a high level of autonomy in manufacturing.

The performance of the proposed framework is demonstrated and evaluated on a Universal Robots UR10 platform using a Microsoft Kinect V2 sensor as shown in Fig 3.2. The framework does not require any complicated and time-consuming sensor calibration.

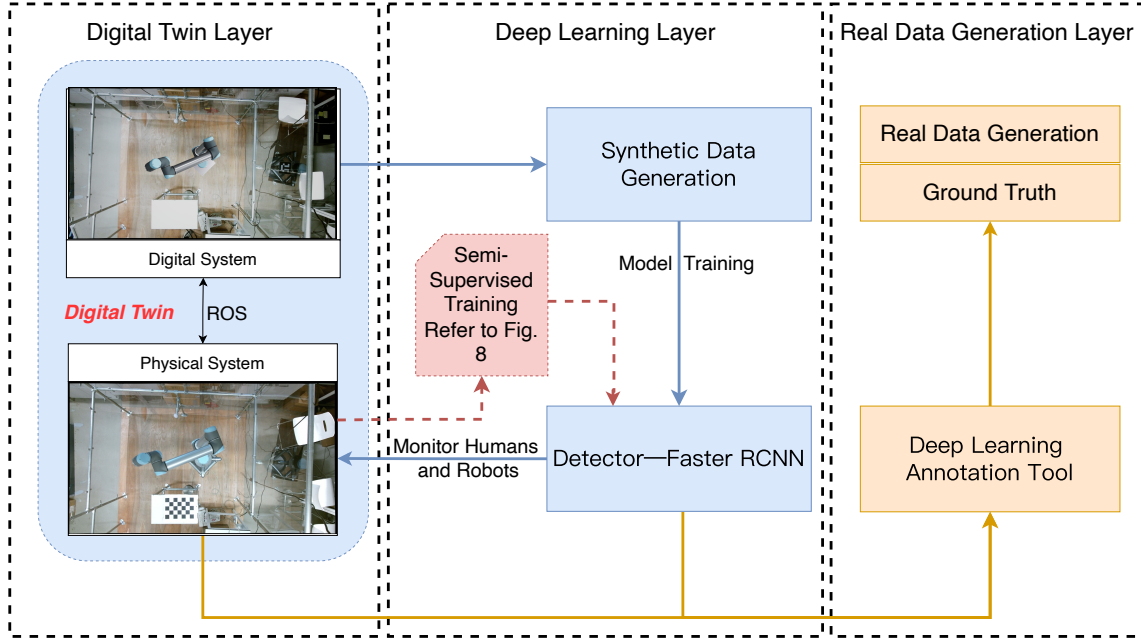


Figure 3.3: Theoretical framework of using deep learning and Digital Twin techniques for monitoring Cobots towards safety and reliability. The framework is comprised of three layers: i) Digital Twin layer, ii) deep learning layer, and iii) real data generation layer. Digital Twin layer illustrates the Digital Twin in which a ROS-based communication system is designed for information transmission including robot pose, the orientation and position of the camera, etc. between the digital and the physical system. Deep learning layer represents how the synthetic dataset with accurate annotations is generated, then the detector is trained with the dataset. The detector is applied to monitor humans and the cobot in the physical system. In the meanwhile, it also illustrates how a semi-supervised detector is trained which will be explained in Section 3.2.4. In the real data generation layer, a deep learning-based annotation tool is developed to assist to collect and annotate real data.

Fig. 3.3 shows the Digital Twin including the proposed deep learning model which consists of three layers: i) Digital Twin layer, ii) deep learning layer, and iii) real

data generation layer. In the Digital Twin layer, a virtual robot in the digital system captures the pose of the physical robot in the physical space during the working process via the ROS, so that the virtual robot performs in the same way as the physical robot. The virtual visual sensor in the digital system has a different function - to capture synthetic data of the robot with random position and orientation. The data annotation information is also generated automatically along with the collection of the synthetic data. During the synthetic data preparation, Domain Randomization as described in Section 3.2.2 is applied to the digital system with the aim of bridging the reality gap between the real world and the simulation.

In the Deep Learning layer, the synthetic data from the digital system is provided for training a Faster R-CNN detector. The detector combined with the deep learning annotation tool is applied to collect the annotated real data in the Real Data Generation layer. With the real data, a semi-supervised method described in Section 3.2.4 is implemented to train a new detector. This semi-supervised detector monitors the interactions between humans and robots in the physical system of the Digital Twin layer to achieve a safe HRC.

This framework provides a cost-efficient solution to generate data with accurate annotations and other types of sensor information such as mask, bounding boxes, RGB, and depth information. A semi-supervised deep learning model is presented to narrow the gap between the digital system and the physical system. Consequently, the detector proposed in this work can achieve more accurate detection, compared to those fully supervised detectors which are purely trained with real or synthetic data.

3.2.1 Communication Design of the Digital Twin

In traditional simulators, e.g., Gazebo [146] and CoppeliaSim [147], all designs, simulations and experiments are finished in such a closed environment without connecting to any other physical systems. However, a Digital Twin requires not only simulation but also a physical test. Consequently, to satisfy this requirement, a data transmission

framework is needed.

In the proposed Digital Twin, bidirectional data transmission is enabled between the physical system and the digital system, which should be capable of performing multiple processes in a real-time manner. UnrealCV [148] built a file transfer protocol (FTP)-based communication system that only listens to a single socket and the one-way transmission allows only one pack of control data during the whole transmission. Consequently, it cannot support a multi-user control at the same time, i.e., the camera and the cobot cannot be controlled in parallel. A higher level communication design is required to meet the synchronous data transmission between the digital system and the physical system in the Digital Twin.

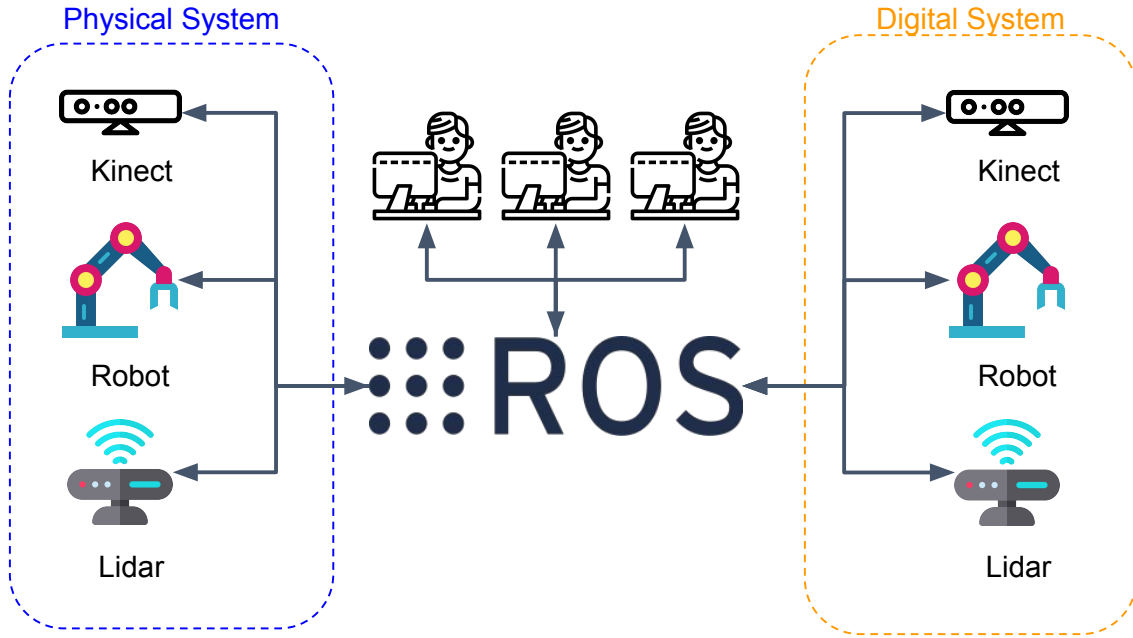


Figure 3.4: A ROS based communication framework is designed for the Digital Twin. In this framework, cameras, cobots and users are regarded as nodes. In a ROS framework, nodes communicate with each other through topics, services, and actions provided by ROS.

ROS has been used to facilitate the implementation of the overall system. ROS is a

distributed system where a synchronous data transmission can be achieved when the digital system and physical system do not need distant communication. A ROS based communication framework is built for the Digital Twin to achieve data transmission among multiple clients. Fig. 3.4 shows how the communication framework is implemented in the Digital Twin. Clients such as cameras, cobots and users are regarded as nodes. Different nodes communicate with each other through topics, services, and actions provided by ROS. For instance, a node can publish defined messages (data) onto a topic, and other nodes subscribing to the topic can receive the message. In this case, joint angles of the physical cobot in the physical system are published, and joint angle data are subscribed by the digital cobot in the digital system (see Fig. 3.3). As a result, both physical and digital cobots move synchronously and keep the same poses. In the meantime, the digital robot can also publish verified robot poses and trajectories to the physical system so that the physical robot can implement specific task without further tests and trials.

3.2.2 A Digital Twin for Synthetic and Real Data Acquisition

Data Acquisition and Data Types

Unreal Engine 4 (UE4) [91] is a powerful gaming engine that has the capability to simulate a physical world realistically. To some extent, the usage of UE4 can minimize the reality gap due to its photorealism. The developed Digital Twin framework uses UE4 as a digital system environment to generate the synthetic data with annotation information for training the developed Faster R-CNN [27] and validating its performance for detection of the areas of the human and of the cobot and making decisions on whether the safety standards are satisfied. With the assistance of the communication framework in the Digital Twin, users can control the camera mounted on the top of the physical robot cell and the physical robot in the physical system to collect the real data as well. In the physical system, to capture images of how the robot carry out its task, the robot arm is moving from one pose to another. At the same time, users can

control the camera to collect data from frame to frame. The size of collected images is 1920 x 1080. Although higher resolutions can bring more accurate detection results, it sacrifices inference speed. This resolution is a common standard in cameras, which can provide rich features for training the model and also ensure a reasonable inference speed. Furthermore, with the trained detector and an annotation tool described in Section 3.3.1, can collect and annotate the raw data efficiently and effectively reduce manual labeling time compared to traditional manual data acquisition and annotation.

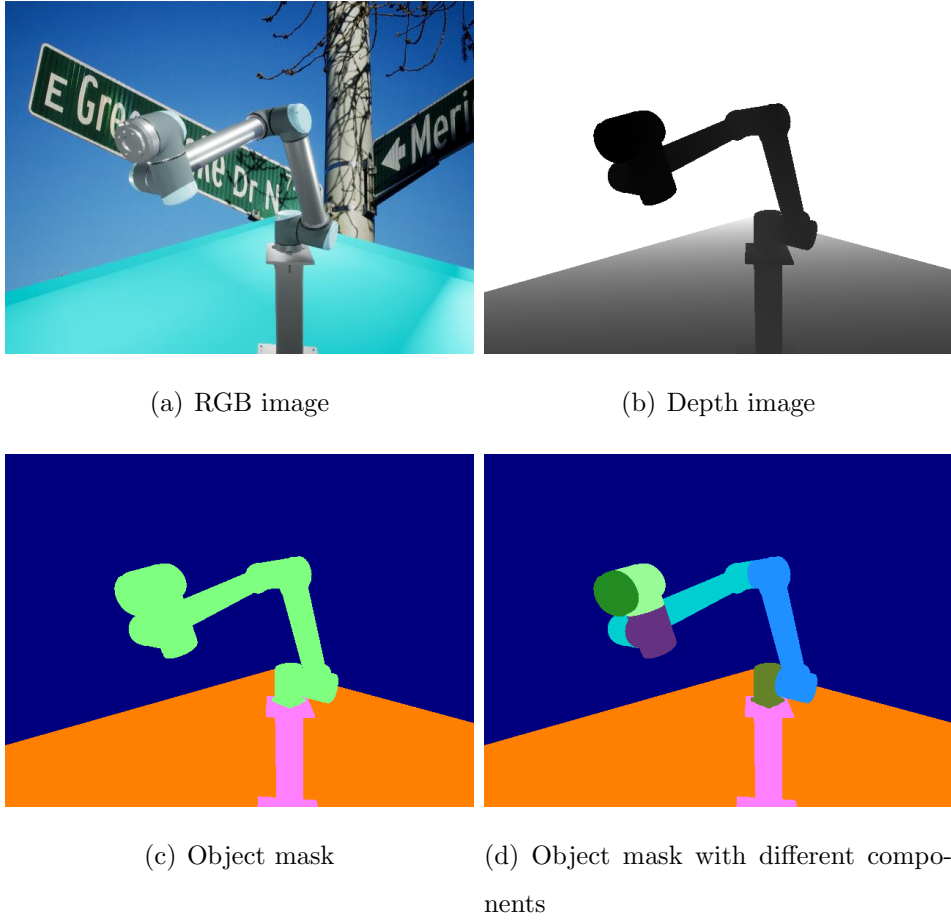


Figure 3.5: *The Synthetic data including different types of sensing information of the cobot generated from the Digital Twin*

Compared to in stock sensors such as RGB cameras that provide specific types of data, the Digital Twin system is more efficient and flexible in obtaining various sensing

information with the help of UE4. Fig 3.5 displays examples of different types of sensing information generated by UE4. UE4 renders objects with their original colors to generate RGB images as shown in Fig 3.5 (a) and it also provides depth information in Fig 3.5 (b). Depth information gives rich 3D information which is of benefit to get the location and orientation of objects. With additional user-defined color information, UE4 can also render an object with a defined single color. Consequently, the annotation of the object can also be obtained with the defined color. The accurate annotation, as demonstrated in Fig 3.5 (c) and (d), is useful for instance segmentation and object detection.

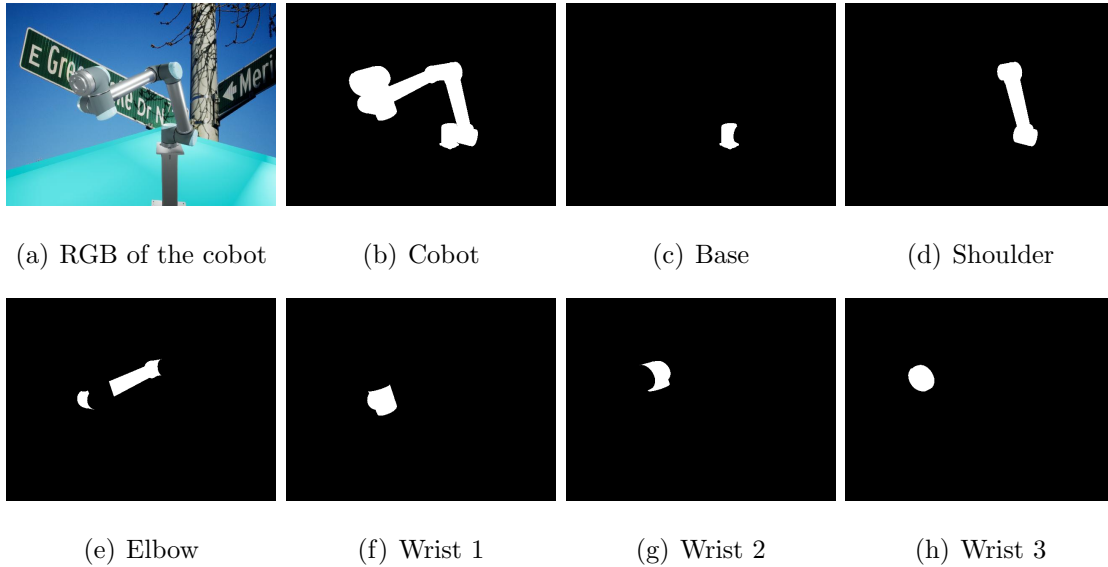


Figure 3.6: (a) represents a RGB image of the cobot, while the masks of the cobot and its components are illustrated from (b) to (c). The digital system can generate different component masks which is defined by users. The cobot mask can be separated into different components and components can be combined as the one. Consequently, users can obtain masks based on their requirements to meet different tasks.

Different from UnrealCV [148], the proposed Digital Twin framework provides more flexibility for users in how the annotation of an object is represented. In Unre-

alCV [148], masks of different objects can be obtained when the specific color is known. For instance, the blue color often represents the background, the green is the robot and the orange is the ground floor in Fig 3.5 (c). However, it is impossible for users to get component masks of an object, because their mask rendering solution only queries from object to object when rendering an object mask scene. This means that the components of the object are not queried during rendering and they cannot be rendered as different colors. Fig 3.5 (c) illustrates that the robot is rendered with single color. In the proposed digital system, the rendering logic is different from the UnrealCV [148] where the digital system both queries what objects exist in a scene but also checks the components of the objects during rendering of a mask scene. Consequently, it can render the components with defined colors which are specified by users when generating mask annotation in which the components of the cobot is rendered with different colors as shown in Fig 3.5 (d), compared to Fig 3.5 (c). Furthermore, different components can be identified in one object and the component masks can be obtained once the colors are known as shown in Fig. 3.6.

Through the proposed Digital Twin, it is easy and efficient to get these types of information which are expensive in traditional manual annotation. The flexibility of the data generation in the proposed Digital Twin is able to meet different tasks including robot detection, robot grasping, pose estimation, etc.

Domain Randomization

UE4 [91] is a powerful and widely-used game engine developed by Epic Games. It is known for its high-quality graphics, robust physics engine, and versatile development tools. UE4's advanced rendering capabilities allow for the creation of highly realistic simulations. This realism is essential for training machine learning models and testing algorithms in environments that closely mimic the real world. UE4 includes a robust physics engine that can simulate real-world dynamics with high accuracy. This is particularly important for robotics and autonomous systems, where understanding and

predicting physical interactions is crucial.

Bridging the reality gap between the physics simulators and the real world is challenging. The aim of the Sim2Real tool is to transfer the virtual models to the real world situations. One approach to generating high quality realistic virtual images is to deploy high-quality rendering simulators such as Unity3D [90], UE4 [91] and OpenGL [92]. In the next paragraphs, the main mathematical notations and concepts are introduced that are needed for the description of the DNN model.

In the proposed framework, Domain Randomization is applied in the digital system to generate abundant samples with the aim of bring the simulated images close to the real ones. It is demonstrated that the model trained over the synthetic data with Domain Randomization has accurate performance under different lighting conditions which will be illustrated in Section 3.4. The Domain Randomization helps improving the deep learning detector and its ability to work under a variety of conditions. Advantages of the digital system are its flexibility, ability to annotate images accurately and to diversify inputs in the feature space. Limitations exist in generating a real dataset with respect to sample diversity. These limitations are linked to a number of factors such as the fixed orientation and position of sensors, unchanged lighting conditions and unchanged backgrounds. These limitations may cause inadequate generalizations and lack of model adaptation in new environments. However, these limitations can be regarded as changeable variations with respect to randomization parameters in the simulators. The randomization parameters considered in the digital system are the following: strength and color of the direct light, position and orientation of the direct light, position and orientation of the camera, images of backgrounds which are from the COCO dataset [87] along with poses of the robot. With these randomization parameters, different kinds of samples can be easily generated, with different appearances. Consequently, the generated dataset can be diverse enough to help the source domain (simulation) to get closer to the target domain (real). It is difficult to collect such different kinds of samples in the real world system due to device limitations.

3.2.3 A Digital Twin for Intelligent Sensing and Machine Vision Tasks in Changeable Environments

The Digital Twin framework proposed in this chapter adopts the Faster R-CNN [17] as a detector to verify the performance of the model trained with different synthetic and real data, under different lighting conditions. The architecture of the considered Faster R-CNN [27] is presented in Fig 3.7. The Faster R-CNN [27] consists of two stages: 1) feature extraction from the input image, and 2) generation of potential region proposals where the location of the object of interest is, calculated with a RPN. As shown in [17], Faster R-CNN can achieve accurate detection in real-time performance. By using the Non-Maximum suppression operation [149], proposals with low confidence are filtered. The remaining proposals and feature maps are refined by the next layer for the Region of Interest (RoI) Pooling stage. The corresponding proposals are classified as different objects as well as their bounding boxes are predicted.

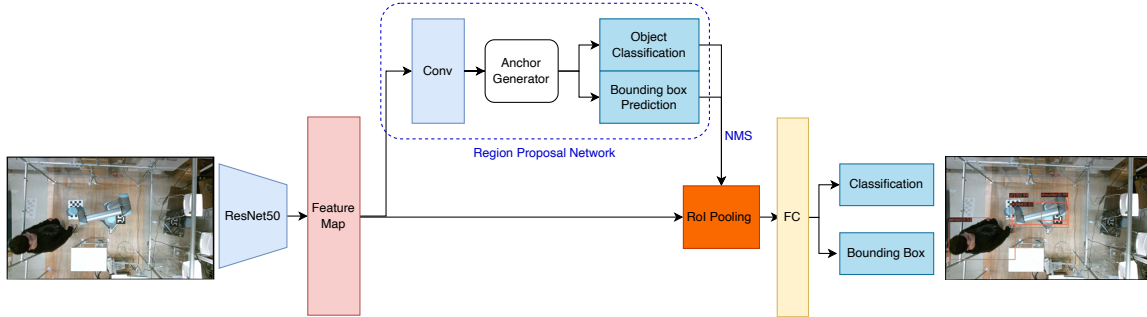


Figure 3.7: Architecture of the Faster R-CNN. ResNet-50 extracts feature maps from the input image. In Region Proposal Network, regions of interest are generated. RoI Pooling processes the regions of interest and their corresponding feature maps to get new feature maps with fixed size. The FC (Fully connected layer) predicts the classes and the bounding boxes for these feature maps.

The architecture of Faster R-CNN [17] includes ResNet-50 [29] for extracting features

from images. The residual block is defined as

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}, \quad (3.1)$$

where \mathbf{x} is the input image for the residual block, \mathbf{y} is the output image feature map which is coming out of the residual block. The function \mathcal{F} represents the residual mapping and $\{W_i\}$ denote the weights of layers in the residual block. The detector block includes two sub-tasks: object classification and bounding box regression for object detection. In the two-stage detector, both loss functions in the RPN and the final Region of Interests (RoI) results are considered.

In the RPN [17], the loss function L_{RPN} of the RPN is defined as:

$$\begin{aligned} L_{RPN}(\{p_i\}, \{t_i\}) &= \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\ &+ \lambda \frac{1}{N_{reg}} \sum_i p_i^* R(t_i - t_i^*), \end{aligned} \quad (3.2)$$

where p_i is the predicted probability of the i -th anchor, which is a binary result characterising whether the anchor is an object or not, and t_i is the corresponding bounding box prediction, N_{cls} is the normalized parameter for the classification. The classification loss in RPN is denoted as L_{cls} , and p_i^* is the corresponding ground-truth, whose value is 1 (positive) or 0 (negative). The balanced parameter is denoted as λ while the N_{reg} are normalized parameters of the regression. The bounding box is optimized with the smooth L1 regression loss function R , and t_i^* is the ground-truth of the bounding box of anchor i . The Smooth L1 function, also known as the Huber loss or Smooth L1 loss, is a loss function that combines the properties of L1 loss (mean absolute error) and L2 loss (mean squared error) [28]. The smooth L1 loss function R is defined in the form:

$$R(t_i - t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 & \text{if } |t_i - t_i^*| < 1 \\ |t_i - t_i^*| - 0.5 & \text{otherwise.} \end{cases} \quad (3.3)$$

For the classification loss function in the RPN, a binary cross entropy loss is adopted

$$L_{cls} = p_i^* \log(1 - p_i) + (1 - p_i^*) \log(p_i). \quad (3.4)$$

In the final RoI area, the cross entropy loss L_{cls}^{roi} for object classification and the smooth L1 loss L_{bbox}^{roi} for bounding box regression are introduced, so the total loss function L required to be minimized is

$$L = L_{RPN} + L_{cls}^{roi} + L_{bbox}^{roi}, \quad (3.5)$$

where $bbox$ denotes the bounding box regression.

3.2.4 A Semi-supervised Teacher-student Detector for Sim2Real

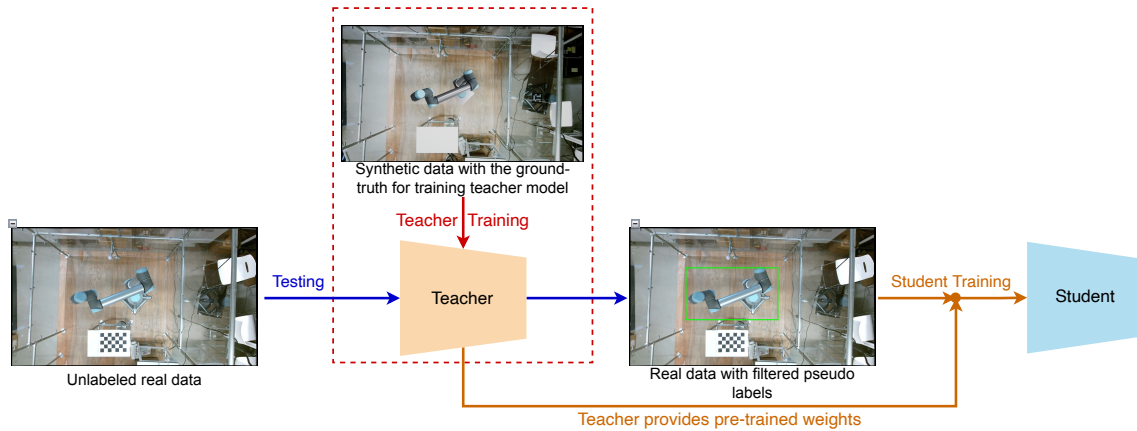


Figure 3.8: Framework of the semi-supervised method applied to train a detector. A teacher model is firstly trained with the synthetic data. The unlabeled real data is fed to the teacher model and the teacher model generates pseudo labels for the unlabeled real data during the testing mode. The pseudo labels are further filtered. Next a student model is trained with the real data with filtered pseudo labels.

A detector trained with the synthetic data can achieve an effective performance in the real world environment. It still needs to be validated whether the detector using both synthetic and real data would have accurate performance within the Digital Twin. A semi-supervised solution is proposed to train a detector of human actions and the whole framework is shown in Fig. 3.8. The proposed semi-supervised method is based on the

Faster R-CNN [17] framework. The proposed solution consists in a teacher-student model to train a student model through semi-supervised training. The teacher model is trained with synthetic data $\mathcal{D}^{syn} = \{\mathbf{X}^{syn}, \mathbf{Y}^{syn}\}$. Once the teacher model is trained, the real data is input without the ground-truth \mathbf{X}^{real} to the teacher model during the testing mode. Then the model will give predicted labels of \mathbf{X}^{real} , which is denoted as $\tilde{\mathbf{Y}}$. However, the real data with its predicted labels $\{\mathbf{X}^{real}, \tilde{\mathbf{Y}}\}$ cannot be used to train the student model directly, because some redundant and low-quality results exist in its prediction $\tilde{\mathbf{Y}}$. To filter these redundant and low-quality results, the Non-Maximum suppression operation [149] is implemented. The Faster R-CNN predicts objects in an image with their bounding boxes and classes with confidence which are regarded as the predicted label $\tilde{\mathbf{Y}}$ for an input \mathbf{X}^{real} . In the Non-Maximum suppression operation, the bounding boxes of each class are ranked by their confidence. The bounding boxes of each class with the highest confidence are remained which are $\hat{\mathbf{Y}}$ while the rest are filtered. After $\tilde{\mathbf{Y}}$ being filtered, pseudo labels $\hat{\mathbf{Y}}$ are obtained.

In the next step, the real data with its pseudo labels $\mathcal{D}^{pseudo} = \{\mathbf{X}^{real}, \hat{\mathbf{Y}}\}$ is applied to train the student model. The weight of the teacher model will be frozen as a pre-trained for training the student model. The student model is the final model that is applied to monitor the interactions between the robots and humans in the physical system. It achieves more accurate and more robust results under changing lighting conditions compared to the fully-supervised Faster R-CNN. The performance of the proposed framework is evaluated in Section 3.4.

3.2.5 Relevance to the Standards and Regulations for HRC

Digital Twin technology provides an enormous potential for incorporating health and safety regulations into cobot systems and vice versa, the Digital Twin can impact the standards and regulations towards higher safety and reliability of these systems. Some of the main safety regulation documents [144, 145, 150], especially applicable to manufacturing, do not consider various levels of autonomy for the needs in different

industrial applications. A part of the technical challenge is to identify and assess the underlying hazards and risks of these cobot systems when not being operated in power and in force limiting (PFL) mode. Particularly, this is especially important in highly automated manufacturing industry which employs intelligent sensing and artificial intelligence systems. In the considered UR10 cobot system, traditional sensors were used for which the current standards and regulations [144, 145] have well specified safety rules. These safety rules include proximity and light gates, to avoid hazardous humane-robot collisions.

This work proposes an autonomous decision-making framework utilizing vision cameras, with the advantage of being able to rapidly adapt to dynamic environments. Additionally, in consideration of the physical reconfiguration of safety sensors as robot movements are reprogrammed for conducting different tasks, the positioning and installation of vision sensors are relatively easy to achieve, compared with light gates and physical fences.

According to the relevant sensing standards [151] which illustrate the requirements for equipment using vision based sensors, several environmental factors should be considered when implementing such sensors into real industrial applications, including optical occlusion, various ambient temperatures and lighting conditions. Due to practical considerations of complex industrial conditions, the proposed detector for actions recognition of human-robot interactions is tested under different lighting conditions in terms of the accuracy of object detection as depicted in Section 3.4. In practice, the proposed detector can be embedded both in a Digital Twin platform and in the control algorithms of a cobot system. Accordingly, if dangerous scenarios such as unsafe interactions or abnormal operations are detected successfully, the operator would be alerted by relevant warnings whilst brake signals would be sent to the controller to delay or stop the robot movements for guaranteeing safety.

3.3 Datasets

Datasets	Full light	Semi- light	Semi- dark	Full dark	Total
Real Data	4,861	2,877	2,977	3,211	13,926
Synthetic Data	-	-	-	-	20,823

Table 3.1: *Numbers of Images in different datasets. To guarantee the real data that contains different light factors, the data is collected under different lighting conditions: full light, semi-light, semi-dark and dark where the lighting condition is changing from light to dark. With respect to the Synthetic data, the data is generated without identifying lighting conditions.*

To build a deep learning-based detector, two different datasets for training and testing are required. Table 3.1 gives details about the two different datasets used: the synthetic dataset for training and the real dataset for testing, described in Section 3.4.2. Benefiting from the efficient synthetic data generation, a synthetic dataset along with the annotation information is created within the digital system of the Digital Twin for model training purpose. With respect to the testing data, real datasets are collected from the physical system of the Digital Twin under the real working environment. With the assistance of the semi-automated annotation tool, the process of annotating the raw RGB data can be speed up for constructing the real dataset.

3.3.1 Semi-automated Annotation Tool

It is usually time-consuming and labour-intensive to annotate real data for each single image. Several commercial annotation tools are available, such as V7 [152] and Labelbox [153], supply AI functions to aid in data annotation. However, one major drawback is that these pre-defined AI models usually only work well in very limited scenarios, for example, detecting cars and humans for autonomous driving tasks. It

cannot meet various demands of annotating specific objects such as the robot UR10, and is not applicable to diverse industrial scenes.

The deep learning model in this framework is working with a semi-automated annotation tool which is developed based on Labelme [154]. The Digital Twin generates synthetic data and then the deep learning model is trained and tested using these data. Furthermore, the deep learning model is deployed with the annotation tool for acquisition and annotation of the real data from the physical system.

3.3.2 Real Data

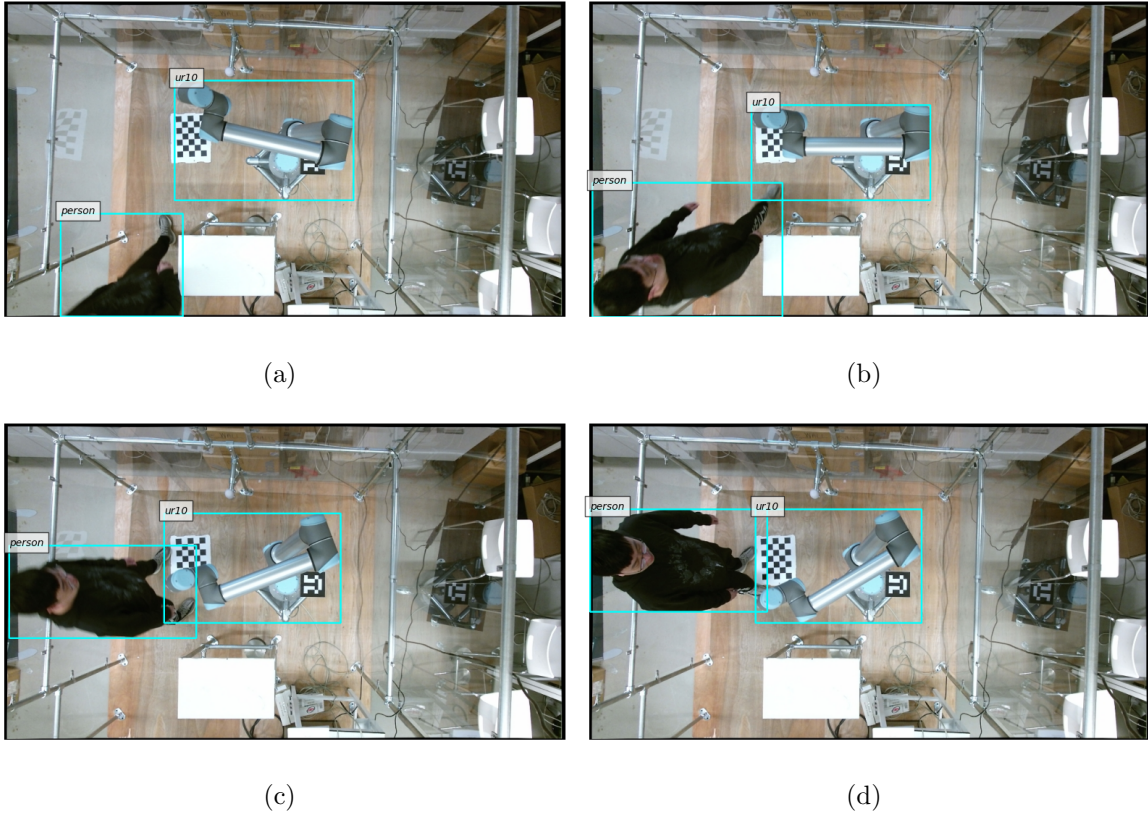


Figure 3.9: Images with annotation information in the real dataset. From (a) to (d), the whole process of human-robot collaboration is captured from the Kinect V2 sensor mounted on the top of the UR10.

In order to validate this framework, a real dataset is acquired by a Kinect V2 sensor based on a UR10 platform and the dataset is publicly. To simulate a real HRC scenario, three operators dressed in different clothes took part in the test whilst the Kinect V2 camera was mounted horizontally on the ceiling, looking down over the workspace. In this case, the field of view of the camera can capture one or two operators at the same time. Fig. 3.9 depicts that when a robot is working in a cell, an operator is moving into the cell and then interacting with the robot.

The real data was collected under various experimental conditions, by changing illumination levels and operators (humans). There are 4 different illumination levels and 2700 images were recorded respectively at each illumination level. Besides, 1653 images were saved with different operators. Totally this real dataset contains 12453 images.

3.3.3 Synthetic Data

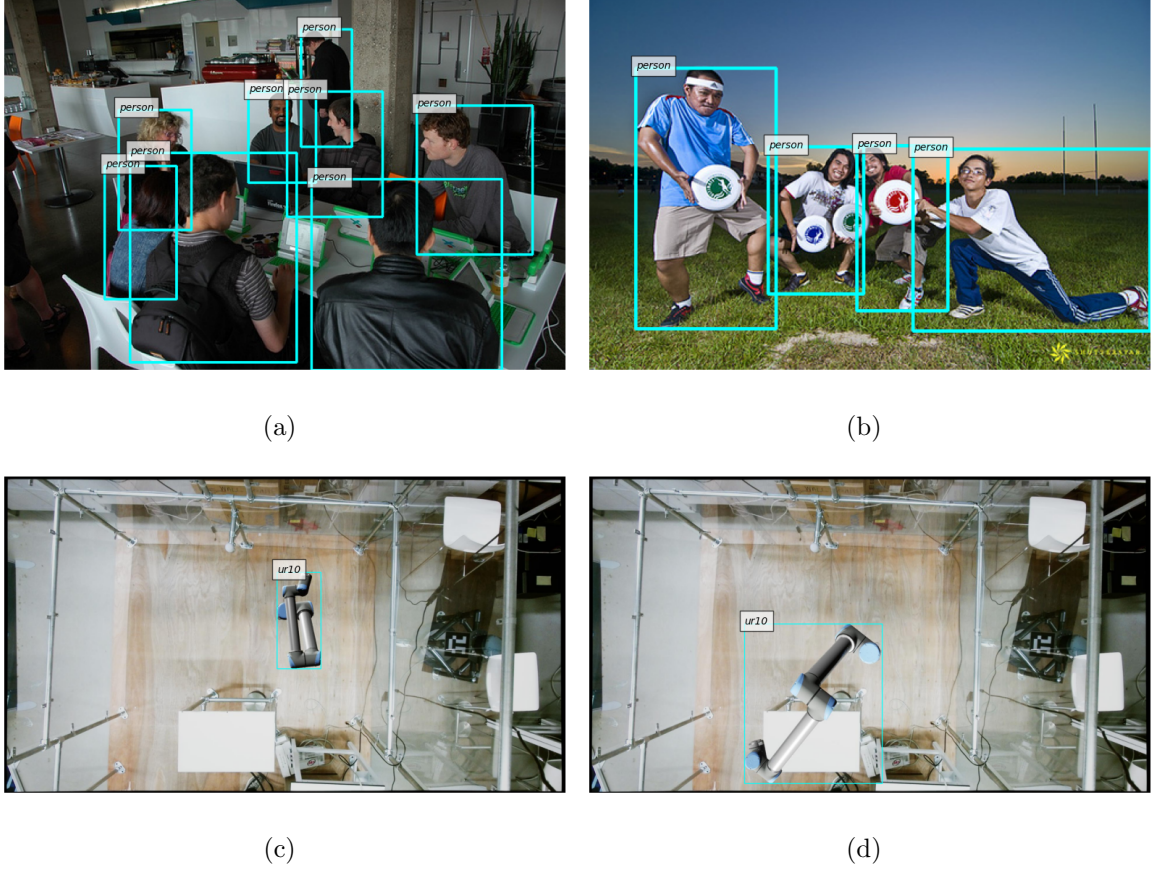


Figure 3.10: *Images with annotation information in the synthetic dataset. (a) and (b) shows human images from COCO dataset, while (c) and (d) are robot images generated from the digital system of the Digital Tiwn.*

The synthetic datasets include robot images that are generated using the proposed Digital Twin technique whilst operator data is gathered from the COCO database [87]. Fig. 3.10 shows people with different appearances and robot images that are fed into training a detector. With respect to the robot images generated from the digital system, Domain Randomization techniques such as different lighting conditions and different robot poses are applied during the data generation. To make the synthetic robot data looks similar to the physical system, the background of the synthetic data is captured

from the physical system.

The reason for merging human samples with annotation information from COCO data [87] with the robot data is that COCO [87] is a public dataset for object detection research and has collected abundant human images. It brings the advantage that the detector can learn diverse human actions from the training data set to improve its generalization. The detector is also capable of detecting different operators with different appearance. This is irrespective of how many operators get into the robot cell since it has learnt enough human data during the training process. Consequently, it can be considered as an effective way to construct a training dataset for HRC scenario without extra data collection and annotation. This synthetic dataset is randomly split into two parts, including 20823 images for training and 5206 images for validation. The image database used in this research is shared online, including the real dataset as well as the synthetic dataset.

3.4 Performance Evaluation and Validation

3.4.1 Evaluation Metrics

The performance of the proposed framework has been evaluated and validated over synthetic and real data under different lighting conditions.

The Average Precision (AP) [88] is adopted as the main evaluation metric, which is defined as

$$AP = \int_0^1 p(r)dr, \quad (3.6)$$

where p denotes the precision function and r is the recall function [155, 156]

$$\begin{aligned} \text{Precision } p &= \frac{TP}{TP + FP}, \\ \text{Recall } r &= \frac{TP}{TP + FN}, \end{aligned} \tag{3.7}$$

where TP represents the true positive values, FP is the false positive and FN is the false negative [88].

The average precision (3.7) represents the area under the precision-recall curve. The average precision has a high value when both precision and recall are high, and it has a small value when either of precision or recall is small. While the average precision AP is calculated for each class, the mean average precision (mAP) is calculated by taking the average of average precision across all the considered classes.

The IoU is defined as follows [155, 156]

$$\text{IoU} = \frac{A \cap B}{A \cup B}, \tag{3.8}$$

where A is the predicted bounding box of an object and B is the corresponding ground-truth bounding box.

The mean AP (mAP), AP at the Intersection over Union (IoU) over 50% (AP50) and the AP at the IoU over 75% (AP75) [87] are used to evaluate the performance of the CNN trained over different datasets and under different lighting conditions.

There are several reasons why mAP is the primary evaluation metric. In this framework, the camera is the only sensor that captures the robot and humans in HRC. The performance that the framework can classify and locate the objects is mainly evaluated in this chapter. mAP combines both precision and recall into a single metric, offering a balanced view of the model's performance. Precision measures how many of the detected objects are correct, while recall measures how many of the actual objects are detected. By considering both, mAP provides a more comprehensive evaluation. In the meanwhile, object detection not only requires correctly classifying objects but also

accurately localising them. The IoU threshold used in mAP calculations ensures that both aspects are considered. If the predicted bounding box is not sufficiently close to the ground truth, it is not counted as a true positive.

3.4.2 Experiment Setting

The Digital Twin is an excellent physical-virtual integrated system which can be used to study the impact of different environmental conditions, including the potential factors which may affect object detection, human action recognition and decision making. This Section 3.4.2 presents results over real and synthetic data with the Faster R-CNN described in Section 3.2.3. Two Faster R-CNN models are trained with different datasets: one is trained only with real data, the other is trained only with synthetic data. Then the performance of the proposed semi-supervised model is also evaluated which is described in Sub-section 3.2.4 which considers both the real data without the ground-truth and the synthetic data with the annotation. The teacher block within the semi-supervised model is firstly trained with the synthetic data and next the student model is trained with real data without the ground-truth. These models are trained on four Tesla V100 GPUs. The three models have been trained with the same strategy, with a stochastic gradient descent (SGD) algorithm.

For the distributed training, 16 samples per GPU are selected with a total of 64 batch size and the overall convergence of the stochastic gradient process takes up to 7 hours. The model trained by real data takes less than a half an hour. linear warmup, a learning rate schedule, is applied for training with an initial learning rate of 0.08 and the learning rate rises linearly after 500 iterations. Together with the stochastic gradient, a technique called momentum is used. Instead of using only the gradient of the current step in the search, the momentum uses the gradient of the past steps to determine the next direction to move. A weight decay of 0.0005 and momentum of 0.9 are applied during the training process.

3.4.3 Performance Evaluation of Detection

Four lighting conditions are considered in the experiment for evaluating the three models. Two Faster R-CNN models trained with two different datasets and the proposed semi-supervised model are evaluated under different lighting conditions.

The first evaluation is within a steady manufacturing environment, where a robot repeats the same routine with pre-defined program in the robot cell. The detection algorithm can achieve accurate and steady results by learning from similar scenes to the robot cell, i.e., the training dataset should be diversified to cover as many scenes as those in the robot working routine. Several environmental factors in real manufacturing scenes may affect the performance of a deep learning-based detector negatively, such as image noise, illumination, unseen objects [157, 158]. Among these factors in the robot cell, the room illumination has the greatest influence on the performance of the detection algorithm. The change of illumination may results from the sunlight or the lighting conditions of the factory which are unpredictable. Several solutions try to eliminate the negative effects of varying illumination including Data Augmentation [159], data collection [160], image preprocessing [161], etc. These kinds of solution aim to increase the diversity of the training data, making the model more robust to different scenarios it may encounter in real-world applications. In this section, the performance of semi-supervised solution trained with synthetic data in varying illumination will be discussed.

	Full light			Semi-light			Semi-dark			Dark		
	mAP	AP50	AP75	mAP	AP50	AP75	mAP	AP50	AP75	mAP	AP50	AP75
Real	0.692	0.98	0.781	0.661	0.974	0.841	0.645	0.965	0.742	0.605	0.968	0.674
Synthetic	0.789	0.978	0.913	0.773	0.965	0.93	0.585	0.844	0.677	0.608	0.904	0.712
Semi-supervised	0.781	0.993	0.924	0.768	0.989	0.928	0.679	0.966	0.804	0.701	0.972	0.817

Table 3.2: Results under different lighting condition, mAP, AP50 and AP75 is utilized to evaluate three object detection models. Real represents the Faster R-CNN model trained with the real data. Synthetic represents the Faster R-CNN model trained with the synthetic data. Semi-supervised is the semi-supervised model.

Table 3.2 shows that the semi-supervised solution achieves the best performance compared to those trained only with the real or synthetic data under four lighting conditions. From Tables 3.2 and 3.3, it is evident that when the lighting condition is becoming worse, the APs of the three models decline demonstrates that the lighting conditions is a critical factor that affects the performance of Faster R-CNN. Compared to the model trained with the real data, the model trained with the synthetic data and the semi-supervised model have better performance when the lighting is sufficient (full light) which are roughly 10% better than the model trained with real data. Especially in good lighting conditions (full light and semi-light), both the model trained with synthetic data and the semi-supervised model achieves over 76% mAP.

	Full light		Semi-light		Semi-dark		Dark	
	mAP_{UR10}	mAP_{human}	mAP_{UR10}	mAP_{human}	mAP_{UR10}	mAP_{human}	mAP_{UR10}	mAP_{human}
Real	0.689	0.695	0.648	0.673	0.596	0.694	0.625	0.586
Synthetic	0.864	0.714	0.835	0.711	0.693	0.477	0.708	0.509
Semi-supervised	0.79	0.773	0.768	0.768	0.7	0.659	0.792	0.611

Table 3.3: *mAP results at UR10 and Human under different lighting conditions. Real represents the Faster R-CNN model trained with the real data. Synthetic represents the Faster R-CNN model trained with the synthetic data. Semi-supervised is the semi-supervised model.*

With respect to AP50 and AP75, AP75 gives closer matching between the predicted bounding box and the ground-truth compared to the AP50 metric. From what AP50 and AP75 of these model in full light and semi-light is illustrated, the Faster R-CNN trained with the synthetic data and the semi-supervised model are above 91%, while the Faster R-CNN trained with the real data in the full light condition only has 78% AP75 in full light and 84% in semi-light. The performance of the model trained with the real data drops over 10% from AP50 to AP75, while the other two show smaller reduction in the average precision which means that the predicted bounding boxes of these models are more accurate and closer to the ground-truth bounding boxes.

However, when the lighting is insufficient (semi-dark and dark), the model trained on synthetic data shows a significant reduction in its performance. The APs of the semi-supervised model drop less compared to the model trained with synthetic data, when the lighting conditions change. The semi-supervised model also outperforms the Faster R-CNN model trained with the real data. Hence, the semi-supervised solution is robust to changes in the lighting conditions.

Table 3.3 gives the results for the mAP of UR10 robot and human under different lighting conditions. The Faster R-CNN trained with the synthetic and the semi-supervised model also achieves better performance than the network trained on real data with

good lighting conditions (full light and semi-light). However, mAPs with respect to both UR10 and humans declines when the lighting is reduced.

Even when the Faster R-CNN is purely trained with the synthetic data, it achieves a remarkable mAP_{UR10} under full and semi-lighting conditions. The semi-supervised model shows more robust behavior when the lighting conditions are changing. Furthermore, with respect to mAP_{human} , the semi-supervised algorithm has the best score compared to those models that are only trained with the real or the synthetic data. The mAP_{human} is above 77% under full lighting and also achieves 61% under the dark situation.

The proposed semi-supervised solution demonstrates robust performance across various lighting conditions. This robustness ensures reliable and consistent results, making it a viable option for real-world applications where lighting can vary significantly.

3.4.4 Decision Making for Safe HRC

The detection algorithm is implemented on a laptop with Nvidia RTX 2070 GPU. When monitoring the robot and the operator in the physical system, it can achieve the detection speed at about 20 frames-per-second (fps), which meets the real-time monitoring requirement in this case. However, the cumulative time delay due to data transmission and model inference time may lead to negative effects on the monitoring a safe HRC. In the meanwhile, some detection failures cannot be ignored, even though it rarely happens.

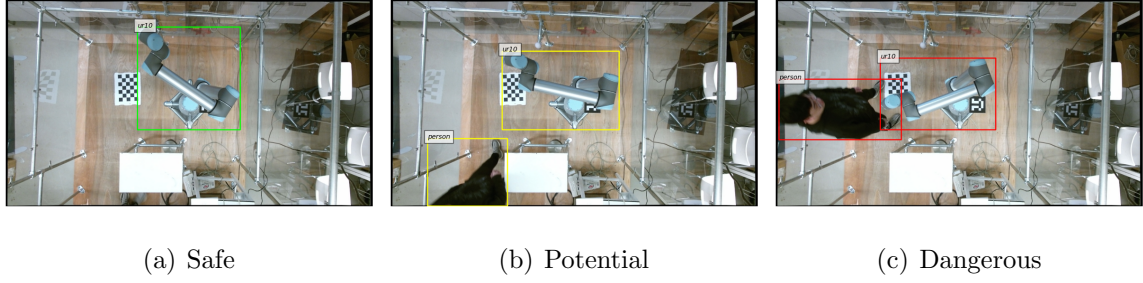


Figure 3.11: *Three safety criteria for safety decision making.*

To enhance the reliability and the safety of the HRC, three decision making criteria are defined to minimize the negative effects described above. A Faster R-CNN detector has the capability to detect objects of interest and their locations in an image. Because the camera used to monitor the interaction between operators and robot is mounted on the top of a HRC cell, it provides a horizontal two-dimensional vision space. With such a spacial relationship between camera frame and the world frame, the detection information (bounding boxes) can indicate how close between the operator and the robot and help to make a safe decision making. The safety decision making criteria can be defined as: i) **Safe**: Only the robot is detected and no operator enters the robot cell, the robot moves at normal moving speed as shown in Fig. 3.11 (a). ii) **Potential**: In Fig. 3.11 (b), the operator enters the robot cell and the bounding boxes of both the operator and the robot are detected and two bounding boxes do not overlap. And the robot reduces its speed to the half of the original speed. iii) **Dangerous**: If two bounding boxes overlap as shown in Fig. 3.11 (c), it means the operator is quite close to the robot. Therefore, the robot should stop immediately to avoid collision with the operator.

With different robot arm movement speed settings based on the safety decision making criteria, the detection algorithm can efficiently reduce the risk of the collision when the operator is getting close to the robot. In normal situations, the robot works at his preset full speed without any operator involvement. When the operator need to get

closer to the robot and interact with the robot, the robot firstly can be aware of the presence of the operator. Then the robot reduces its speed based on the safety decision making criteria. When an overlap between the bounding boxes surrounding the human and the robot end-effector occurs, the robot stops immediately. This allows the operator to have enough reaction time to potential danger due to the the movement of the robot arm, ensuring that any unforeseen movements or errors do not result in harm. By incorporating these safety measures, the system not only enhances the safety of the operator but also improves the overall efficiency and reliability of the robotic operations. The adaptive speed settings and immediate stop mechanism act as critical safeguards, providing a robust framework for human-robot interaction in various industrial environments.

By calibrating the camera parameters, the camera is positioned at 3 meters height from the ground. The horizontal distance between the operators and the robot is about 20 cm when their bounding boxes are overlapping at the beginning. In the proposed work, the human can keep a safe distance to the robot with the designed criteria based on the bounding box information. This is a different solution compared to the approach proposed by Liu and Wang [162] which is a collision-free HRC approach, requiring the position information for both the human and robot. The approach of Liu and Wang [162] requires extra sensor-robot coordinate calibration for the purpose of collision sensing which is not necessary in the proposed case.

Inspired from [163], the Kalman filter and Hungarian matching method are used here to improve the reliability of the inference process. The state of each detection box is defined as $\mathbf{x} = [\mu, \nu, s, \eta, \dot{\mu}, \dot{\nu}, \dot{s}]^T$, where (μ, ν) is the center of the bounding box in an image, s is the scale parameter and the η is the ratio of the height to width of the bounding box. The other variables $\dot{\mu}, \dot{\nu}, \dot{s}$ denote the respective speeds of the center coordinates and scale of the bounding box. When a bounding box is detected by the detector, it is applied to update its corresponding target state with the Kalman filter. The IoU distance between the detected and predicted box of an existing target

that is tracked, is calculated. The assignment between the current and predicted box is performed by the Hungarian matching algorithm. To reduce the delays from the inference time of the detector, the frequency of detection is reduced to detect an image every 4 frames. It significantly improve the speed from 20 fps to 100 fps by sampling detection results when updating the states of the tracking with this post-processing.

A detector may fail to detect objects in some frames which may reduce the reliability of the monitoring process and could raise risks of danger in HRC. Thanks to the Kalman filter, the negative effects of such detection failures can be eliminated to a great extent. For example, the detection result plays a role as “observation” in Kalman filter to update the estimated state. Even though some observation points are missed, the Kalman filter can skip the update step and rely solely on the prediction step to estimate the current state. This robustness is one of its key strengths. For the multi-object tracking problem [163], occlusions are also key factors that could reduce the quality of tracking performance. Thanks to the monitoring camera mounted on the top of the robot cell, some occlusions can be avoided.

Although in [134], a similar deep learning approach is proposed, it applies the Mask R-CNN [86] to extract mask information. The mask information helps to reconstruct 3D relationship between the human and the robot in order to calculate the direct distance for safe decision making. The proposed inference speed outperforms the speed reported in [86]. Mask R-CNN [29] which is an extension version of Faster R-CNN [17] requires extra computation cost to predict mask information. It would be difficult to achieve a real-time performance without extra post-processing, even though a real-time calculation is reported in [29]. In this case, the Kalman filter has improved the inference speed to 100 fps, which is regarded as real-time performance. Ensuring real-time performance is crucial for the framework when monitoring both the robot and the operator in HRC. Any delays could potentially lead to danger when the operator approaches the robot, as the system must account for quick and accurate decision-making to avoid collisions or hazardous interactions. In a HRC environment, ensuring

the operator's safety requires real-time monitoring and prompt responses to dynamic changes in their positions.

3.4.5 Discussion

From the analysis presented above, several advantages of the proposed framework are evident. First, it is easy to setup and deploy the proposed framework in manufacturing. Within the proposed Digital Twin, users can simply build a Digital Twin of the physical manufacturing workspace by introducing CAD models of real objects into the Digital Twin. The communication between the physical and digital systems can be established by the ROS.

Traditional deep learning application in manufacturing usually requires huge data collection and expensive manual annotation work. However, these can be avoided in the proposed framework by implementing efficient data generation and with semi-supervised method using the Sim2Real technique.

Besides, flexibility is another significant advantage of the proposed framework. This generative framework is not limited to detect humans and robot actions. It can also be extended to other objects by introducing new objects through adding their CAD models into the digital system. In the meanwhile, users also can specify annotation method to meet their requirements described in 3.2.2. Moreover, Faster R-CNN can be replaced with another detection model within the semi-supervised method. The adoption of the efficient data transmission scheme between the digital and the physical systems, together with the automatic annotation generation, can allow users to implement other tasks, such as reinforcement learning [164] and AR [165] in HRC.

This chapter also evaluates and discusses the effect of one key environment factor, lighting condition, on detection performance. Additionally, by introducing the Kalman filter and the Hungarian algorithm, the detector is enhanced to avoid detection failures whilst the inference speed is also improved. With these post-processing and decision

making rules, the safety distance between the human and robot is maintained which enhances the reliability of the HRC environment.

Digital Twin combined with artificial intelligence have a huge potential to make a difference in smart manufacturing. This was also demonstrated in [166, 167]. Moreover, the inclusion of cloud computing services in Digital Twin can lead to cyber-physical cloud manufacturing systems [168]. Digital Twin can be served as a platform for reinforcement learning training [169, 165], and meanwhile, reinforcement learning is promising to lead the next generation of Digital Twin.

3.5 Conclusions

This chapter explores the feasibility of a Digital Twin in smart manufacturing. It proposes a deep learning-enhanced Digital Twin for detecting and classifying human and robot actions for enhancing safety in manufacturing systems. A Digital Twin is designed for human-robot collaborations which generates synthetic data directly in the digital system. This helps with the generation of real data in the physical system with accurate annotation. The Digital Twin is an efficient tool for studying different levels of safety and to design decision making and control algorithms for manufacturing purposes.

The Robot Operating System is used to provide synchronous communication with, and real-time control of, the robot. The Digital Twin corresponding to the physical system is designed with the help of Domain Randomization and the powerful photorealistic Unreal Engine 4. Training of the developed deep learning algorithms is achieved successfully with synthetic data. A fully-supervised detection algorithm is shown to achieve successful detection results in the real environment. To ensure reliability of the system under different lighting conditions, a semi-supervised detector is proposed to take both synthetic and real data into the training and detection process, which helps in bridging the gap between the two systems in detecting humans and robots.

Chapter 4

Deep Learning-Enabled Resilience to Occlusion for Physical Human-Robot Interaction

4.1 Introduction

The integration of robots into human environments has led to an increasing need for safe and efficient human-robot collaboration. Dressing is a basic activity of daily living that does not benefit from assistive devices and can be challenging for those with mobility issues. Much work has been done in recent years to address this gap using robot assistance, with particular insights in safe human-robot interaction [170], motion control [171], visual and haptic feedback [172, 173] and garment manipulation and grasping [122].

In robot-assisted dressing, the robot continuously adapts its motion to the user’s movements using visual feedback. Occlusions introduce significant challenges since they lead to uncertainty regarding human pose estimation. This issue has previously been explored for recurrent neural networks and user arm occlusion caused by placing garments

on a Kinect sensor [174]. In this chapter, occlusions that naturally occur during assisted dressing by an occupational therapist is investigated. A Deep Learning approach is trained based on data collected during human-human dressing trials, facilitated by an experienced occupational therapist (OT).

The proposed framework consists of a CNN, IK solver, and a parametric multi-body model. The CNN was trained using a large dataset of human poses to infer joint locations from a stream of RGB images. The estimated joint positions are input to the IK solver, which provides a robust numerical solution based on the Levenberg-Marquardt (LM) method. This method updates the parameters of the multi-body model to estimate the human motion. The parametric multi-body model uses a 10 degree-of-freedom (DoF) representation of the human arm. To evaluate the accuracy of the framework, dressing experiments are conducted with healthy volunteers inside a motion capture laboratory featuring a VICON system, subsequently used as the ground truth for the volunteers' hand pose estimation. Mean-squared error analysis results are reported to demonstrate the convergence performance of the approach. The proposed framework has the potential to improve the safety and comfort of human-robot collaboration, with significant implications for the development of assistive robotics.

Three research questions are addressed. First, whether an IK solver can generate a ground truth within the user's ergonomic arm workspace that aligns with the VICON data set in the absence of occlusion. Secondly, whether fusing CNN data with the CNN-KF can improve robustness against occlusions is assessed. Lastly, whether the CNN-KF is robust against spontaneous arm motions. The main contributions in this chapter are:

- A framework, that is robust to occlusions and environmental disruptions, that uses a single camera to retrieve 3D joint location for a human arm.
- A robust online solution to the IK problem that takes estimated hand positions from the CNN-KF and estimates user motion. This solves the IK problem for a

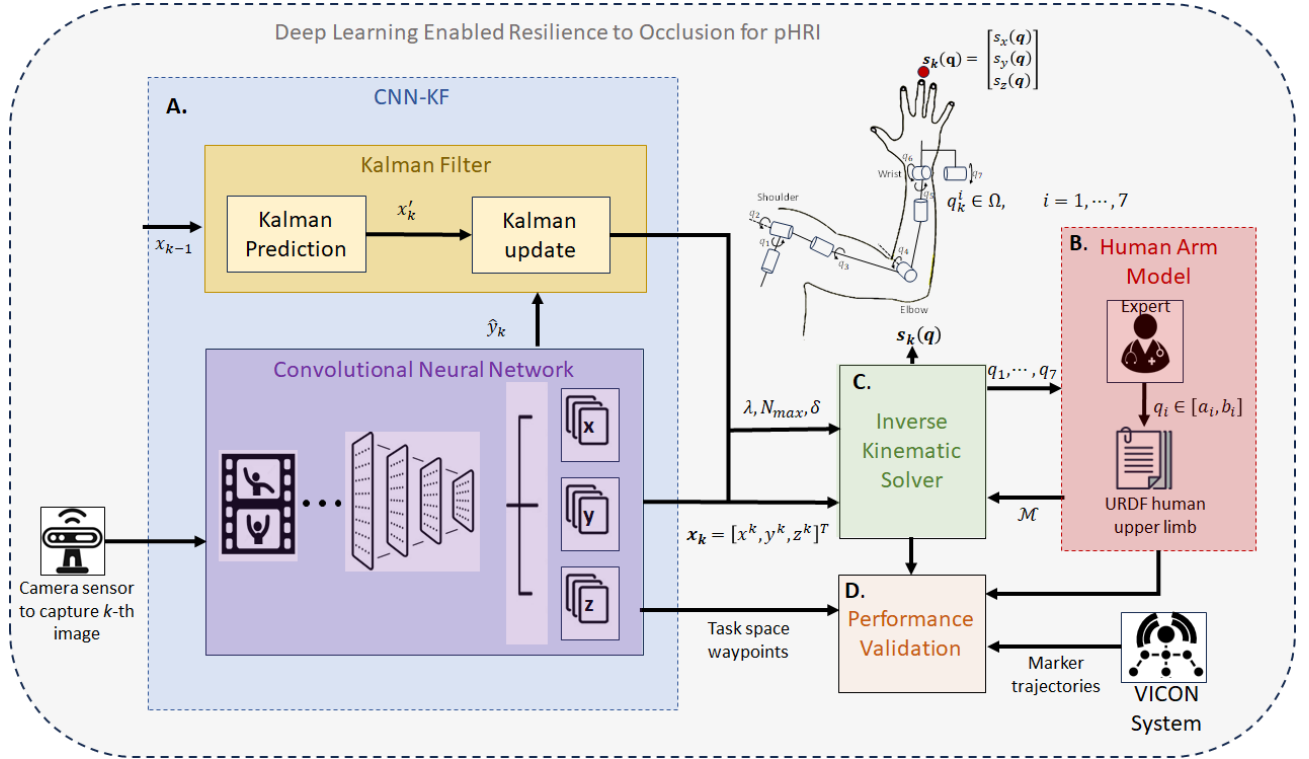


Figure 4.1: *The Proposed framework for occlusion-robust and ergonomically safe physical human-robot interaction.*

hand position, finding an updated model configuration in joint space that satisfies kinematic constraints.

- Evaluation of the accuracy of the CNN, CNN-KF and the IK solution using the mean-squared error analysis, where the ground truth for the hand pose was provided by a VICON motion capturing system.

4.2 Proposed Framework

Figure 4.1 presents the proposed framework to simultaneously predict the user's upper limb movements and update the kinematic model during assisted dressing in real-time. It adopts a CNN fused with a CNN-KF and robust numerical methods to solve the IK problem. There are four main components: A) machine learning component CNN-KF,

B) a parametric multi-body model of the human upper limb, C) a numerical inverse kinematic solver. Component D) is used to validate the performance of the proposed framework.

4.2.1 The CNN-KF model

The CNN was trained, using a large data set of human poses, to infer joint locations from a stream of RGB images taken as input from the dressing scenario. The estimated joint positions are then fused with the Kalman filter to regulate the learning model and to remove outliers. The CNN-KF joint estimations are provided as input to the inverse kinematic solver.

CNN

The CNN framework is Top-Down [175], with three stages being used to estimate 3D pose: human detection, 2D pose estimation and lifting the pose to 3D. For human detection, the detector used in this chapter follows the baseline of Faster R-CNN [27]. Given an image, the detector outputs bounding boxes which represent areas containing a human. Faster R-CNN [27] is a two stage detector which is comprised of feature extraction network and a Region Proposal Network (RPN). The feature extraction network takes an image as input and outputs feature maps, while RPN generates potential object areas in images, called region proposals. The region proposals together with their corresponding feature maps are processed by Region of Interest (RoI) Pooling and feed forward network to get the final detection results: the bounding boxes. HRNet [176] is then applied for 2D pose estimation. HRNet is a convolution-based neural network that encodes a high resolution image in different scale features. With respect to 2D human pose estimation, the network predicts K keypoints: the joints of human body. Consequently, K heatmaps are predicted. For a groundtruth keypoint, the groundtruth heatmap y_i is generated with 2D Gaussian where the centre is the location of the keypoint. To optimise the network, the mean square error (MSE) loss

function is introduced,

$$L_{2dKeypoint} = \frac{1}{K} \sum_{i=1}^K (y_i - \hat{y}_i)^2, \quad (4.1)$$

where \hat{y}_i is the prediction of keypoint i . Once the 2D keypoints are detected, a 2D human skeleton is formed and passed to the pose lift stage for lifting 2D to 3D pose. In order to lift the 2D human pose to 3D, a SimpleBaseline3D model [54] is trained to predict 3D human pose by taking 2D human pose as input.

Kalman filter fused CNN

The CNN signal is subject to noise mainly caused by occlusion and disturbances in the environment. For an occlusion-robust approach, the CNN fuses with a Kalman Filter (KF) to smooth a signal. The CNN-KF uses the traditional KF as defined in [177], to detect and remove outliers. The state-space model of the system is defined and it is assumed that the state vector is $\mathbf{x} = |\hat{\mathbf{y}}|$ where the i -th element of $\hat{\mathbf{y}}$ is a signal acquired from the CNN for the i -th joint location. The state equation describing the changes from one image frame to another is:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \quad (4.2)$$

where k denotes the discrete time (image frame), \mathbf{A} is the transition matrix that reflects the change in the image-frame and \mathbf{w}_{k-1} is vspace. vspace is white, Gaussian, with a covariance matrix \mathbf{Q} .

Furthermore, \mathbf{w}_{k-1} is independent from the state \mathbf{x}_{k-1} . The measurement model can be expressed as:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \quad (4.3)$$

where \mathbf{H} is an $m \times n$ matrix describing the relationship between CNN learned values and state. \mathbf{v}_k is the noise associated with the measurement and is white, Gaussian, with variance \mathbf{R} . The magnitude for covariance matrix \mathbf{R} is large, since it is expected to change in measurement noise due to occlusions. It is noted that \mathbf{Q} and \mathbf{R} are diagonal and are restricted to be positive definite. In the model the state vector

$\mathbf{x}_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k]^T$ represents the 3D joint location $[x, y, z]$ at image-frame k and also the respective speed of the change in position in \dot{x} , \dot{y} and \dot{z} .

In the state and measurement equations (4.2)-(4.3) the respective \mathbf{A} and \mathbf{H} matrices are in the form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{I} * t \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} * \mathbf{t} \end{bmatrix}, \quad (4.4)$$

where \mathbf{I} is a unit matrix, $\mathbf{0}$ is a zero matrix, t is the time between two consecutive frames, and adheres to a 30 fps rate, identical to that of the camera. Furthermore, the kinematic variable speed is constant because erratic upper limb motion patterns from subject is not expected. As an iterative feedback loop algorithm, the KF achieves optimality with the update step and the prediction step. In the prediction step, the predicted state \mathbf{x}'_k and predicted covariance \mathbf{P}'_k of the current state, independent of the current measurement are calculated:

$$\mathbf{x}'_k = \mathbf{A}\mathbf{x}_{k-1}, \quad (4.5)$$

$$\mathbf{P}'_k = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}. \quad (4.6)$$

In the update step, the optimal Kalman gain \mathbf{K}_k is computed and is used to estimate the mean and covariance of \mathbf{x}_k , which also takes as input the observed measurement \mathbf{z}_k :

$$\mathbf{K}_k = \mathbf{P}'_k \mathbf{H}^T (\mathbf{H} \mathbf{P}'_k \mathbf{H}^T + \mathbf{R})^{-1}, \quad (4.7)$$

$$\mathbf{x}_k = \mathbf{x}'_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \mathbf{x}'_k), \quad (4.8)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}'_k. \quad (4.9)$$

The KF is also susceptible to outliers caused by behaviours that are not considered in the model. The \mathbf{R} and \mathbf{Q} covariance matrices is manually adjusted to eliminate these outliers. The state variable \mathbf{x}_k is then provided as the initial guess for the IK solver, to search for a single solution in the arm workspace of the human subject.

4.2.2 Parametric multi-body model

The model uses a 10 degree-of-freedom (DoF) representation of the human arm, based on a set of rules from the Denavit-Hartenberg (DH) convention [178]. Figure 4.2 represents the human upper body model adapted from the Xsens MVN model [179], as an articulated multi-body system. Furthermore, each link can only connect with a 1-DoF joint and so dummy links are used to create a higher DoF joint.

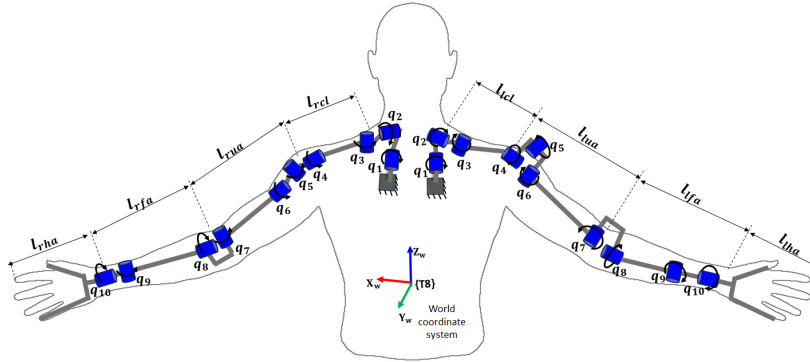


Figure 4.2: *Human upper limb representation as a 10 DoF robotic arm adapted from Xsens MVN model [179]; two kinematic chains: 1) Left-arm and 2) Right-arm.*

Finally, the human arm model is expressed within a Unified Robot Description (URDF) format, and specify the joint constraints. The IK solver takes as input the parametric model of the articulated multi-body, solves the IK problem and returns the updated model configuration in joint space and the Dof model of upper body is also explained in Table 4.1.

4.2.3 Inverse kinematic solver

A robust online numerical solver based on the Levenberg-Marquardt (LM) method is developed as described in [180]. The LM solver is used to dynamically update the joint angles of the parametric multi-body model based on run-time observations of the hand position (end-link) received from the CNN-KF. Furthermore, if the estimated end-link

State	Description
Base {0}	T8 Origin Frame
Clavicle {C1}	Anterior and Posterior Rotations
Clavicle {C2}	Elevation and Depression
Clavicle {C3}	Protraction and Retraction
Shoulder {S1}	Abduction and Adduction
Shoulder {S2}	Flexion and Extension
Shoulder {S3}	Pronation and Supination
Elbow {E1}	Flexion and Extention
Elbow {E2}	Pronation and Supination
Wrist {W2}	Flexion and Extention
Wrist {W3}	Radial/Ulnar Deviation

Table 4.1: *Frames of the the upper limb 10 DoF model. Each link can only connect with a 1-DoF joint and so the dummy links are used to create a higher DoF joint.*

lies beyond the workspace of the arm, the LM algorithm finds a solution, within the arm workspace, that minimises the least squares error.

So, let the complete joint configuration of the multi-body system in Figure 4.2 be specified by the vector of n joints $\mathbf{q} \in \Omega$, where $\mathbf{q} = [q_1, \dots, q_n]^T$ and each q_j is a joint angle. Ω is the set of possible joint configurations and, typically, bounds each angle to a range: $\mathbf{q} \in \Omega \iff q_j \in [a_j, b_j]$ for $a_j \leq b_j$. Links l_{lha} and l_{rha} in Figure 4.2 are *end-links*. The position of the i -th end-link is denoted s_i and is a function of the joint

angles $s_i(\mathbf{q})$. The target position for the i -th end-link is denoted t_i . The IK problem is to find values for \mathbf{q} such that $t_i = s_i(\mathbf{q})$, for all i . The LM solver, uses an iterative method to approximate a single best solution.

Finally, the LM solver calculates joint configurations (Ω) for a desired end-link pose (\mathbf{e}) based on a specified multi-body model (\mathcal{M}). The values for the error tolerance (δ), the damping constant (λ), and the maximum number of iteration (N_{max}) are set *a-priori*, before starting the computation. The LM solver will always return an ergonomically safe solution, which will help the robot to avoid putting the human user in uncomfortable positions, and enhance the human-robot interaction.

4.3 Experimental setup

4.3.1 Participants

The results of the trials involve a professional occupational therapist and three healthy volunteers (1 female and 2 male), age range 22-32 years, height range 160-179 cm, and weight range 62-96 kg. The volunteers were guided by the OT to mimic four typical upper body spasticity patterns defined in [181], often observed in stroke patients [182] with respect to the position of shoulder, elbow, forearm, and wrist joints. All participants gave written informed consents to take part and the trials were approved by the University of Sheffield Ethics Committee (043182). Table 4.2 presents Hefter's four spasticity patterns. 12 dressing trials are recorded, as a set of 4 trials pertaining to the spasticity patterns in Table 4.2, between the OT-participant_{*i*} dyad ($1 \leq i \leq 3$). Each trial lasted approximately 10-12 minutes, with a 5 minutes break between trials and 12000 data points are collected from each trial. In each trial the participant was seated on a chair while the OT stood next to them. Each trial lasted approximately 7-9 minutes, followed by a 5-minute break. With respect to image data, 20,000 RGB data points are collected from the video stream and 57,000 marker trajectory points from the motion capture system. The input stream of RGB images were recorded by a

12MP camera with auto-focus, f/1.8 aperture and optical image stabilisation, capable of capturing 4K video, set to 30 fps.

4.3.2 Motion capture

In the experiments, the occupational therapist assisted the volunteer in donning a robe. This process was initiated from the volunteer’s left ‘spastic’ arm, emulating a disability scenario, subsequently transitioning to the left shoulder, and finally to their right ‘healthy’ arm. An eight-camera VICON motion capture system is operated, operating at 120 Hz, to capture the 3D locations of markers. This allowed us to precisely monitor the participants’ positional dynamics throughout the procedure. Markers were affixed to discernible bony landmarks: at the base of both the left and right middle fingers, and on each shoulder. Complementing this, two reflective markers were positioned at the anterior aspect of the head, with an additional two markers at the posterior. Data inconsistencies arising from marker occlusions were rectified using linear interpolation. Data inconsistencies arising from marker occlusions during data collection were rectified using built-in interpolation functions in the Vicon Nexus system. Post-collection, the data was filtered using a 3rd order 20 Hz low-pass Butterworth filter. This VICON dataset provided the ground truth, corroborating the efficacy of the numerical solution addressing the IK challenge.

With respect to datasets for training CNN model for human pose estimation. The human detector and 2D pose estimation model was trained with the MS COCO dataset [87], which contains over 330,000 images, with more than 200,000 labeled images. There are 17 2D keypoints for each annotated human in the dataset. The 3D human pose model, on the other hand, are trained using the Human3.6M dataset [183], which includes about 3.6 million frames. The annotations for this dataset include 3D joint positions for each labeled human.





Segment	Pattern I	Pattern II	Pattern III	Pattern IV
				
Shoulder	Int. Rot./Abd.	Int. Rot./Abd.	Int. Rot./Abd.	Int. Rot./Abd.
Elbow	Flexion	Flexion	Flexion	Flexion
Forearm	Supination	Supination	Neutral	Pronation

Table 4.2: *Upper limb spasticity patterns.*

4.3.3 Planned disruptions

In each of the four dressing trials (see Table 4.2), a disruption is planned to elicit spontaneous movements from the participant. The planned disruptions were: a simulated fire alarm (d_1); a random call to the volunteer on their mobile phone (d_2); the volunteer randomly interacting with objects in the environment (d_3); and a random obstruction in the OT's pathway (d_4).

4.3.4 Rigid-body model parameters

The variables l_{cl} , l_{lua} , l_{lfa} and l_{lha} in the kinematic chain (Figure 4.2) represent the length from the left clavicle to the shoulder joint, the left upper arm, left forearm and left hand lengths, respectively. These can be estimated from the subject's height based on bio-mechanics literature [184]: $l_{cl} = 0.129 * H$, $l_{lua} = 0.186 * H$, $l_{lfa} = 0.146 * H$ and $l_{lha} = 0.108 * H$.

Furthermore, in the proposed framework, each joint constraint (bounds) will be specified by a domain expert (i.e., Physiotherapist or an Occupational Therapist) and will be adjusted according to the rehabilitation therapy plan for the user. However, participants were healthy adults and the range of motion for the upper body internal degrees of freedom were based on biomechanics literature for a healthy adult.

4.3.5 LM solver parameters

The IK solver was built using MATLAB-Simulink 2023a. The *inverseKinematics* system object was used to create the IK solver to calculate the joint configurations for the desired end-link pose based on the multi-body model of the human upper limb. In order to generate a suitable human-arm manipulator configuration, the *solverAlgorithmProperty* is set to *LevenbergMarquardt* algorithm. The *SolverParameters* used were: final error tolerance $\delta = 10^{-12}$; maximum iterations $N_{max} = 1500$; *MaxTime* = 30s and damping constant $\lambda = 0.1$. The parameters were set *a-priori* and all the other parameters were set as default. The simulation of the modules presented in the proposed framework (Figure 4.1) was run on a Dell 11th Gen Intel Core i7, Windows laptop.

4.4 Results from User Trials

The first phase of the scenario in Figure 4.3, from $k = 0$ to $k = 21$ seconds, involves the OT engaging participant p_1 in conversation to see how they are feeling. During this, p_1 's right-hand (RHA) rests on his right leg and the left arm is held in spasticity pattern II. In state 2, the OT pulls the garment over the left-hand (GoL_ha); very little motion is recorded in p_1 's right-hand, by both the CNN and the VICON system. Whereas in the left hand (LHA) there are oscillations, which are caused by occlusions. In state 3, disruption (d_3) is recorded, p_1 receives a mobile phone call. In order for p_1 to answer the call, the OT stops the dressing, p_1 pulls the mobile phone from his righthand trouser pocket. This can be visualised in the first dip in the curve (region [d], in Figure 4.3). p_1 then raises the mobile phone to his right-ear to speak to the caller, and keeps the right-hand in that position during the conversation. When the call is over, p_1 returns the mobile to the trouser's pocket, visualised by the second dip in the curve. In state 4, the OT pulls the garment over the left forearm (GoL_fa). Finally, in state 5, the OT holds the garment to the right shoulder (HG_rsho), for p_1 to put their healthy limb into the sleeve. The trial lasted 117 seconds.

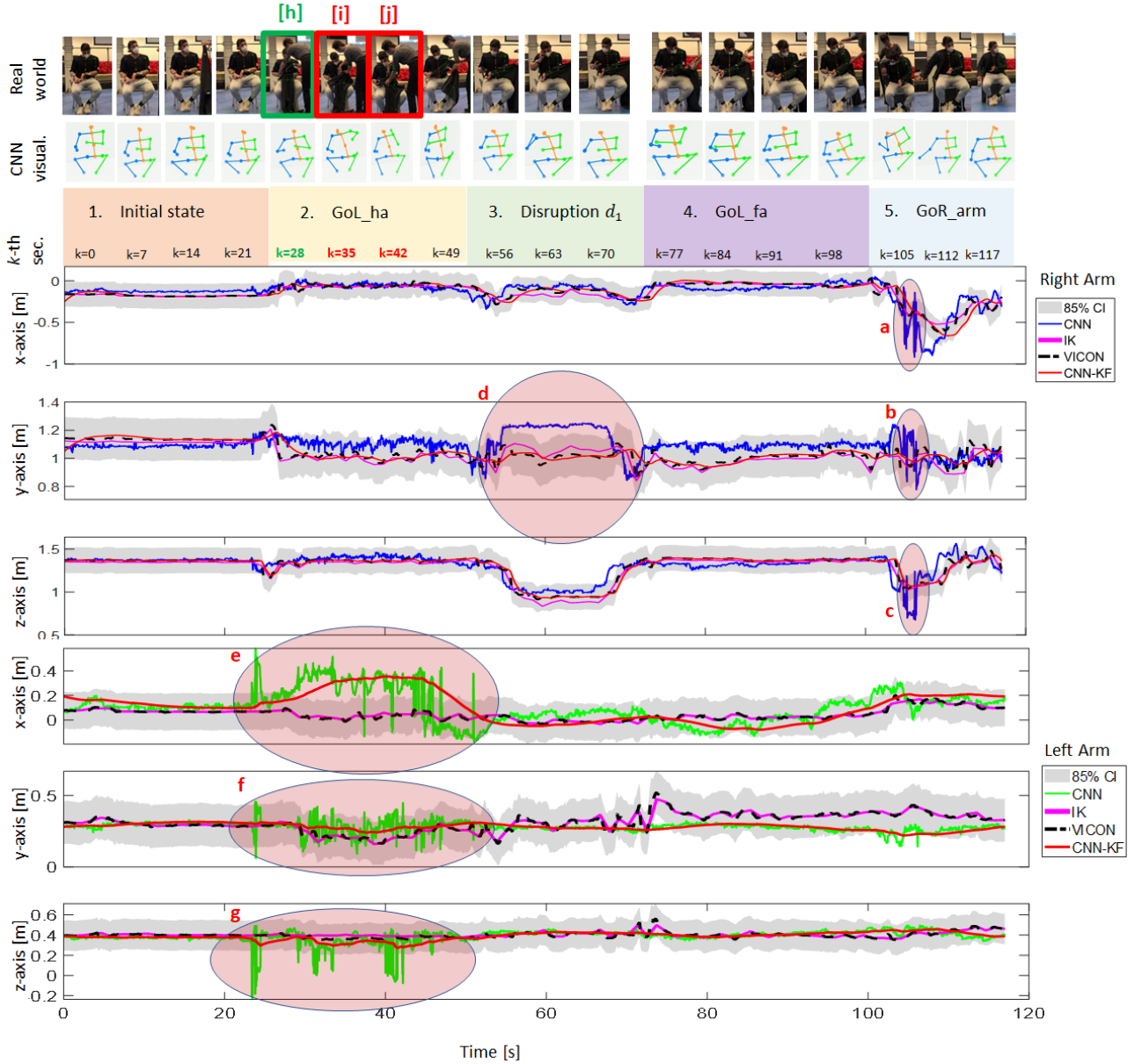


Figure 4.3: Participant p_1 performing dressing trial for spasticity pattern

II.

Although the CNN performed well, as shown in Figure 4.3, it significantly deviated from the VICON trajectory in the y-direction in region d . It seems likely that this is simply the result of one camera not always providing enough information to estimate 3D locations. In practice, one could instead use a multi-camera setup to maximize

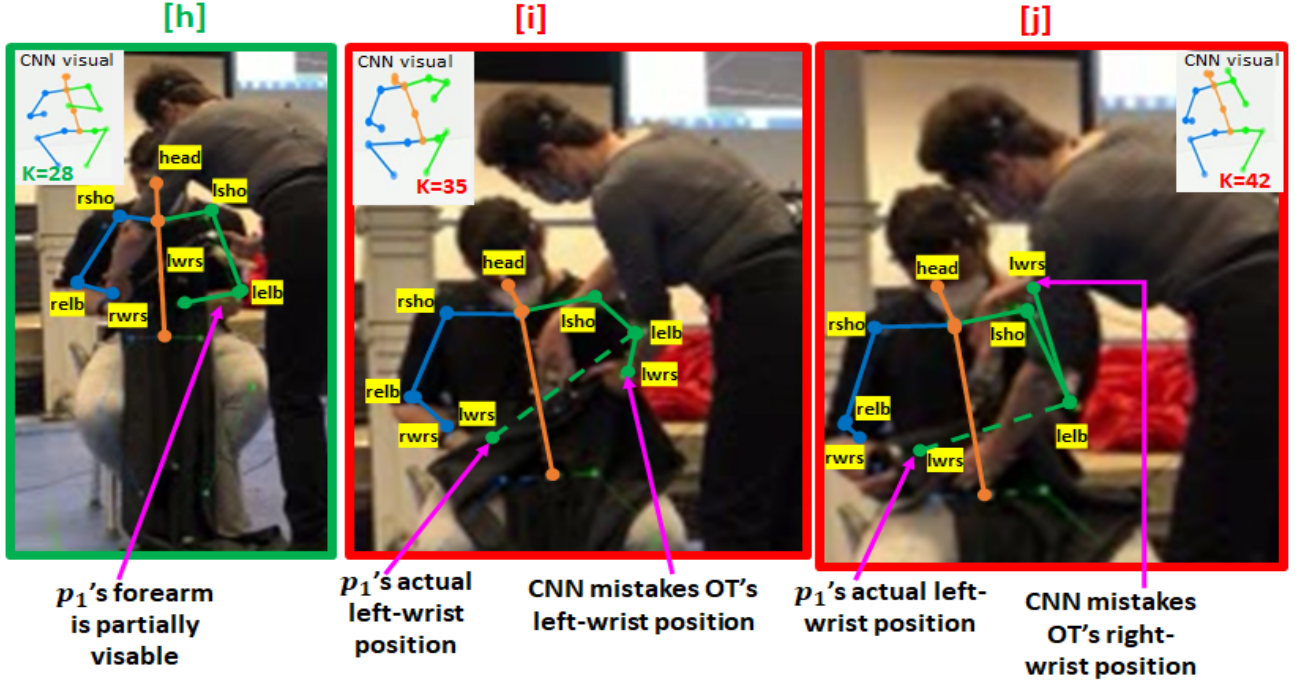


Figure 4.4: World image [h] depicts CNN being able to learn p_1 's occluded left wrist joint location from partially visible forearm. In [i]-[j], the CNN is not able to learn p_1 's wrist joint location correctly due to occlusion by the garment.

coverage and to minimize occlusion. Importantly, however, Figure 4.3 shows that when a Kalman Filter is applied to the CNN signal, adjusting the covariance matrices (Q and R) to reduce the Kalman Gain, the result was similar to the VICON signal, achieving results within the 85% CI. Regions [e-g] in Figure 4.3 show oscillations in the CNN curve for the left-wrist joint. However, p_1 had restricted movement in his left-arm (see the VICON curve and IK solution IK_{VICON}). The corresponding images [i, j], at $k = 35$ and $k = 42$ seconds, and magnified in Figure 4.4 show that the oscillations are caused by occlusion by the garment. The CNN mistakes the OT's wrist joint for p_1 's. In contrast, in image [h] in Figure 4.3 and 4.4, at $k = 28$ seconds, the CNN can learn the correct position of the left-wrist joint despite occlusion because p_1 's forearm was

partially visible. It is unsurprising that occlusion can lead to errors.

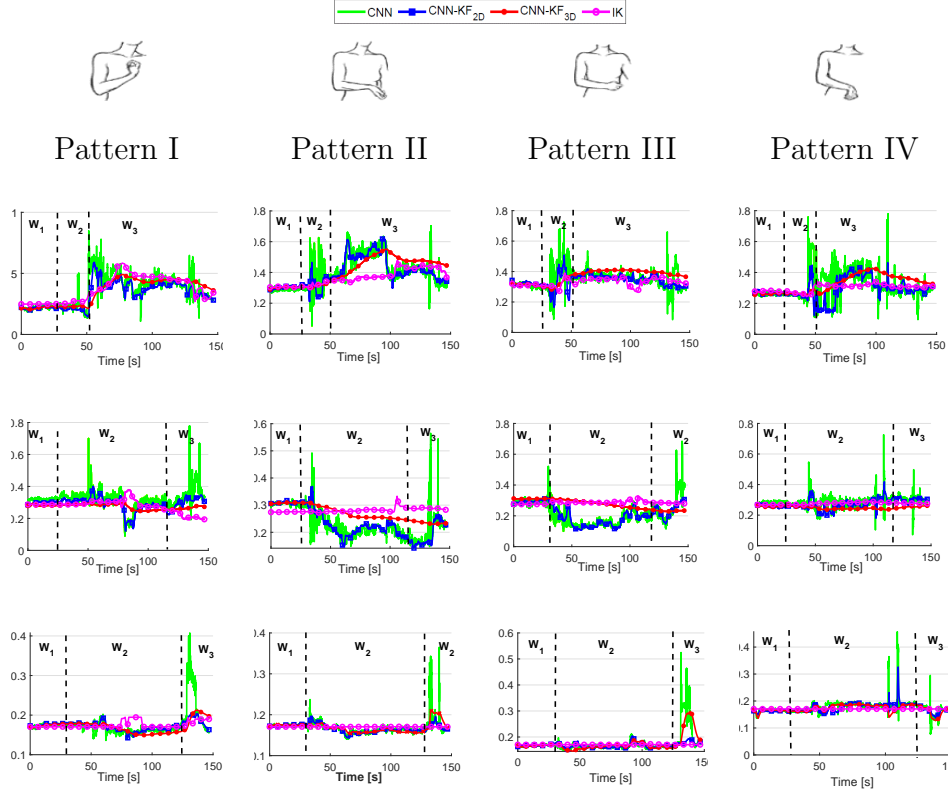


Figure 4.5: *Left-arm joint signals, in window W_1 the participant p_2 is waiting to be assisted with dressing, there are no occlusions, in windows W_2 and W_3 there are some occlusions, mainly caused by the occupational therapist during assisted dressing.*

Figure 4.5 gives the trajectories of the left-wrist (LWRS), left-elbow (LELB), and left-shoulder (LSHO) across the four spasticity patterns with p_2 . In window W_1 , where p_2 awaits dressing assistance, there are no occlusions and the signals remain stable. However, in LWRS window W_2 , occlusions arise as the OT pulls the garment over p_2 's hand segment and wrist joint. These occlusions cause the CNN to under-perform, leading to oscillations in the CNN signal.

Applying a Kalman filter (KF) to the CNN's 2D keypoints (CNN-KF_{2D}) results in a more dependable signal. Furthermore, by extending this to CNN's 3D keypoints (CNN-

KF_{3D}) and adjusting the Q and R covariance matrix, where Q is decreased and R is increased so the signal is significantly enhanced, effectively mitigating noise induced by occlusions. It's important to note that applying the KF to either the CNN's 2D or 3D keypoints doesn't yield substantial differences; rather, it's the fine-tuning of the Q and R covariance matrices that yields a smoother curve. Additionally, integrating a KF with a CNN that lifts 2D to 3D keypoints combines the depth-aware capabilities of pose estimation, while the KF aids in refining these estimates over time, taking into account the temporal dynamics and potential noise in the measurements. Future efforts will focus on dynamically adjusting Q and R using weighted Mahalanobis Distance to filter real-time noise outliers..

For the LSHO signal during W_2 , the OT is pulling the garment over the hand, forearm, and upper-arm segments. Interestingly, there are relatively few occlusions observed during this process. This is attributed to the OT's positioning: initially standing mostly in front of p_2 while assisting with pulling the garment over the hand and forearm segments, (consequently, more occlusions observed in W_2 for LWRS and LELB signals), and then moving to the left side of p_2 when pulling the garment over the left upper arm. As a result, the shoulder remains visible to the camera for the majority of the time, minimising occlusions and ensuring clearer signal.

Figure 4.6, depicts the corresponding outcomes for the right arm joint signals of p_2 . This arm represents the healthy limb and is less constrained by mobility issues. Compared to the left arm, it experiences fewer occlusions from the OT, but is affected more by environmental disruptions (Section "Planned disruptions"). Specifically, during disruption d_2 , deviations between the CNNs and the ground truth signal IK are notable, particularly in the RWRS and RELB signals. The deviation is likely due to single cameras sometimes lacking sufficient information to estimate 3D locations. Although, the CNN, was trained on multi-viewed data of the Human3.6M dataset, it may not have fully represented the diversity of poses and activities encountered in real-world scenarios. To mitigate this, employing a multi-camera setup could improve coverage and

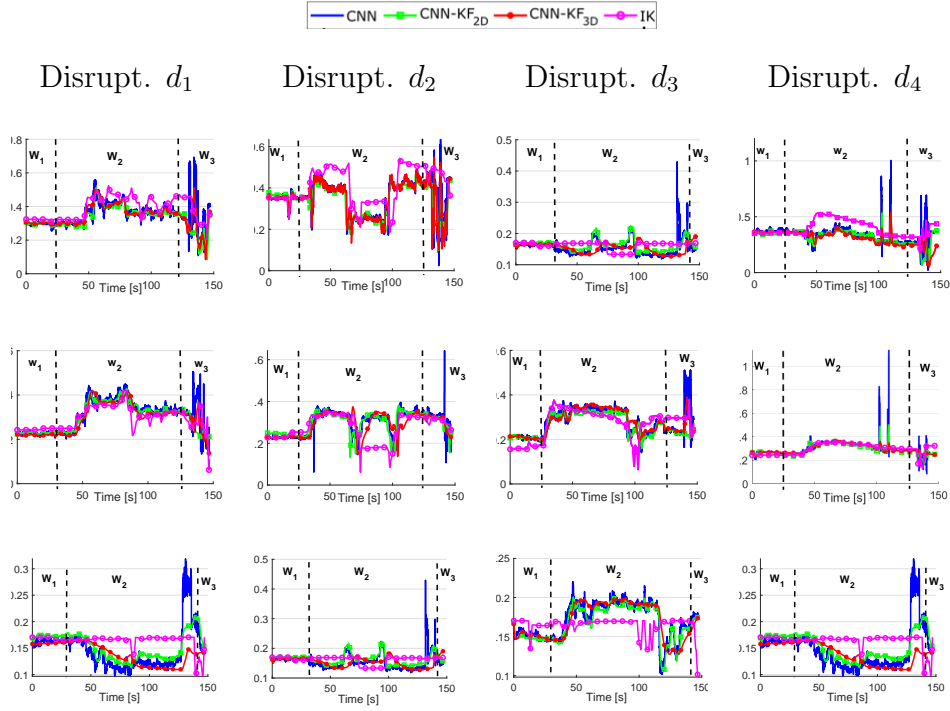


Figure 4.6: *Right-arm joint signals, where in window W_1 there are no occlusions or disruptions, in windows W_2 and W_3 there are some occlusions and environmental disruptions.*

reduce occlusions for more reliable signal estimation. Figure 1, in the Appendix, also illustrates how the CNN under-performs because it is not able to adapt to information from different viewpoints.

4.5 Analysis and Evaluation

Recall that the VICON system provided the ground truth for the hand location during the trials. In this section, the results of the regression analysis are reported which are conducted to assess the convergence performance and accuracy of the presented approach.

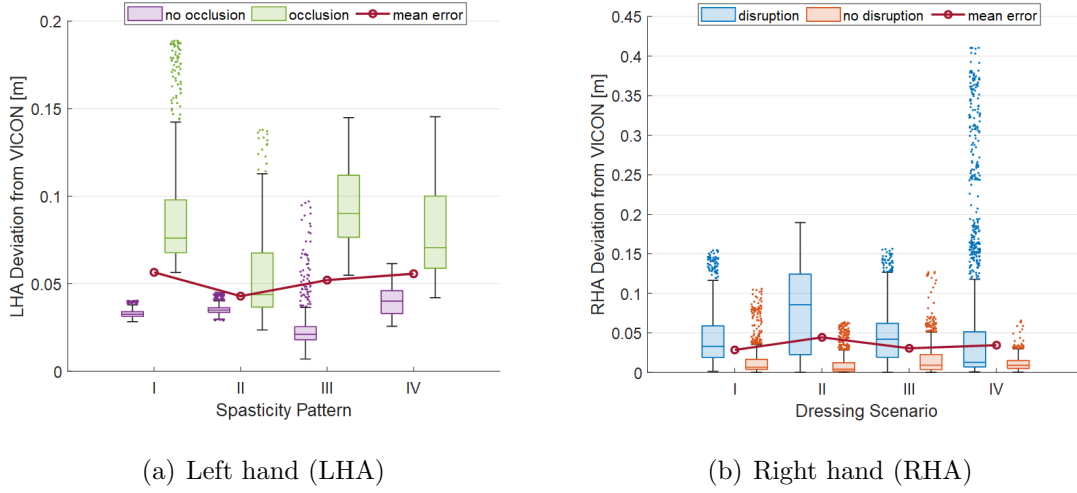


Figure 4.7: Deviation of the hand pose estimation, between the IK solution and the VICON marker trajectory, during dressing scenarios I-IV with no/occlusion. The results represent the averages over three participants performing each dressing scenario.

4.5.1 Inverse kinematic solver

The evaluation of the IK solver based on four dressing trials with three participant is presented. Figure 4.7.a shows the average deviation of IK solution from the VICON data set, for the left hand (LHA), for all four spasticity patterns. The IK solver successfully converges to the VICON trajectory marker with a median deviation below 0.047m when there were almost no occlusions. This concurs with the accuracy of above 98% across the scenarios, within 0.1m (Table 4.3) for the LHA with no occlusions; with both the RMSE and MAE below 0.05m across the scenarios. When occlusions were reported the median deviation increased with maximum upper whisker at 0.1454m and accuracy of below 89% across the scenarios. Furthermore, when the VICON system experienced occlusions, the deviation in the IK solution increased, resulting in outliers. Outliers indicate cases where the estimated configuration is infeasible and the LM solver aims to find a ‘best’ feasible solution.

The results for the right hand are similar. The IK solver achieved an accuracy of

Dressing scenario	Accuracy score %				RMSE [m]				MAE [m]			
	no occ.		occ.		no occ.		occ.		no occ.		occ.	
	LHA	RHA	LHA	RHA	LHA	RHA	LHA	RHA	LHA	RHA	LHA	RHA
I	100	98	74	86	0.025	0.026	0.049	0.034	0.033	0.013	0.092	0.044
II	99	99	89	62	0.009	0.013	0.032	0.053	0.035	0.010	0.054	0.079
III	98	97	57	84	0.046	0.025	0.066	0.059	0.024	0.016	0.094	0.045
IV	100	100	74	86	0.036	0.011	0.089	0.067	0.040	0.011	0.079	0.058

Table 4.3: *Average performance across the four dressing scenarios based on three participants performing each dressing trial.*

above 97% across the scenarios when there were no disturbances, with a mean RMSE below 0.026m. When disturbances were introduced, to illicit spontaneous arm motion, the median deviation increased, with an accuracy of below 86% across the dressing scenarios (Table 4.2).

The results indicate that the IK solver is consistent with the ground truth when there are no occlusions. When there are occlusions, and so the VICON data is likely to be incorrect, the IK solver returns a solution that is within the user’s workspace.

4.5.2 Handling occlusions with the CNN-KF

The dressing trajectory for the left upper limb consists of four waypoints: left-wrist (LWRS), left-elbow (LELB), left-shoulder (LSHO) and right-shoulder (RSHO), respectively.

In this section, the estimation of the upper limb joint locations of the CNN and the CNN-KF is compared with the IK solution based on the VICON data set, for each waypoint along the trajectory. Figure 4.8 visualises the average deviation of each waypoint for both the CNN and CNN-KF from the IK solution, across all four scenarios performed by three participants. The accuracy scores reported in Table 4.4 are based on an error margin from the ground truth within 0.1m. Both the CNN and the CNN-

Waypoint	Accuracy score %				RMSE [m]				MAE [m]			
	CNN		CNN-KF		CNN		CNN-KF		CNN		CNN-KF	
	no occ.	occ.	no occ.	occ.	no occ.	occ.	no occ.	occ.	no occ.	occ.	no occ.	occ.
LWRS	99	55	100	96	0.044	0.158	0.038	0.053	0.039	0.118	0.036	0.045
LELB	98	83	99	90	0.042	0.082	0.039	0.063	0.038	0.065	0.036	0.052
LSHO	100	98	100	100	0.046	0.058	0.058	0.042	0.058	0.043	0.058	0.041
RSHO	100	98	100	100	0.056	0.048	0.059	0.048	0.055	0.042	0.059	0.044
RELB	100	82	100	88	0.039	0.090	0.039	0.070	0.035	0.072	0.036	0.059
RWRS	98	87	98	92	0.055	0.077	0.055	0.059	0.051	0.056	0.052	0.047

Table 4.4: *Average performance index scores across the four dressing trajectory waypoints for the left-right upper limb, based on three participants performing the four dressing scenarios.*

KF successfully converge to the IK solution with a median deviation below 0.042m across all four waypoints when almost no occlusions are reported. The corresponding high accuracy scores of above 98% for both CNN and CNN-KF, with RMSE and MAE below 0.05m, concurs with these results. When occlusions are reported the median deviation increases, notably more for the CNN, particularly for spasticity pattern IV, with maximum median deviation of 0.128m for the $lelb_{IV}$ waypoint. This is due to the induced disturbance d_4 (random obstruction in OT’s pathway), which led to more occlusions caused by the OT. Whereas, the other disturbances had less impact on the left arm joint estimations because they elicited spontaneous motion patterns in the right arm. In comparison, the results reported for the approach, the CNN-KF, show robustness to occlusions, with median deviation reported below 0.05m and accuracy scores above 96% across all four waypoints. Moreover, Table 4.4 also reports the accuracy scores for the CNN and CNN-KF for the right arm joints, which were more exposed to spontaneous arm motions but less occlusions from the OT.

The higher accuracy scores for the CNN-KF show the robustness of the approach to both occlusions and to spontaneous motion. Likewise, the lower RMSE and MAE errors for the CNN-KF, in comparison to the CNN, also indicate robustness. Nonetheless,

the ground truth also includes outliers in the VICON data set, which may have some negative impact on the results. Furthermore, the CNN operates at a fixed speed of 18 fps, the CNN-KF, achieves a faster speed of 30.4 fps by sampling frames selectively. The speed tests were conducted using the Nvidia RTX 3090Ti GPU.

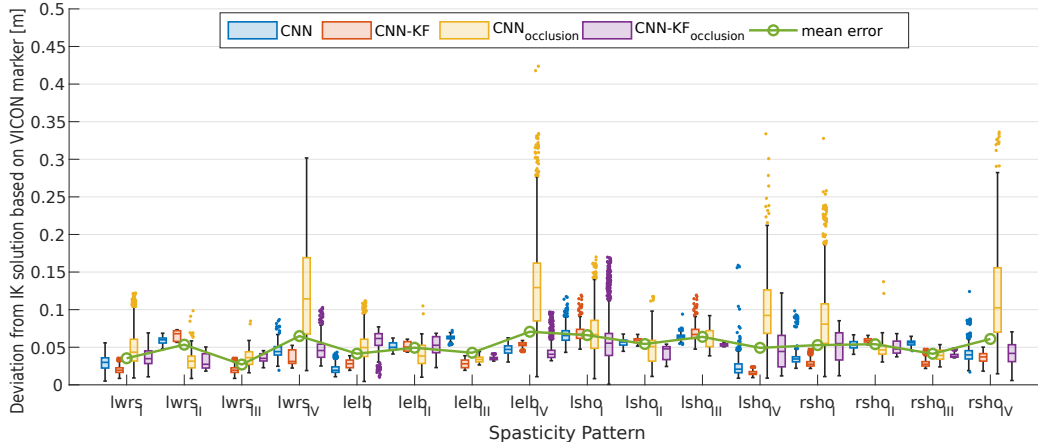


Figure 4.8: *Deviation of assisted dressing trajectory across all four spasticity patterns, between CNN and CNN-KF from the IK solution. The results are based on the averages of three participants performing all four dressing scenarios.*

4.6 Conclusions

A real-time tracking framework is introduced for human upper limb motion, specifically tailored to spasticity patterns observed in stroke patients. By modelling the problem as a partially observable dynamical system, 3D postures are inferred using a CNN with Kalman filtering (CNN-KF_{3D}). The evaluation compared these results with hand poses estimated by VICON reflective markers, revealing that CNN-KF_{3D} exhibited less deviation and greater robustness to occlusions compared to the CNN model alone. However, the approach under-performed during high-movement scenarios, attributed to a lack of training on multi-view data. Additionally, leveraging the Levenberg-Marquardt IK solver enhanced robustness by optimising solutions within

the ergonomic arm workspace, particularly when the desired hand position was unreachable.

Chapter 5

Real-time Activation Pattern Monitoring and Uncertainty Characterisation in Image Classification

5.1 Introduction

DNNs have attracted increasing attention [185, 186, 187] both in academia and industry during the past decades. They have been intensively investigated in the fields such as robotics [188, 189], autonomous driving [190] and manufacturing [191] which require high levels of safety due to involvements of human. Especially the performance of deep learning methods for image classification under uncertainties has been investigated. The [192, 193] summarise the recent state-of-the-art and how different uncertainties impact DNN methods. The quantification of uncertainties can be performed by propagating a tensor normal distribution as a prior of a CNN [194]. The mean and variance of the Gaussian distribution are propagated within a CNN frame-

work called PremiUm-CNN developed in [194]. The variance is especially informative and a small variance means accurate classification results. Another approach uses the Hamming distance [195] which characterises well the difference and similarity between binary strings.

Although there is a number of approaches that are able to quantify uncertainties in CNNs, such as [196, 197, 198, 199], there is a necessity of expanding these studies with different types of uncertainties - in the data: gradual and abrupt, due to environmental changes, including lighting and meteorological conditions, camera motion and other factors. Other effects can be intentionally introduced, such as adversarial attacks and are aimed to cause CNNs to make mistakes. It is important to identify when a trained CNN model performs inference correctly in order to provide a trustworthy result [200, 46, 45]. Ideally, CNN models have highly reliable performance with those inputs that have features similar to their training data sets. However, calculating similarity between inputs and training sets directly is of high computational complexity due to the reason that samples may have very high dimension.

This chapter develops a Faster R-CNN supervised classification framework able to quantify the impact of data on the performance of the classifier. The network has testing data that are not the same as the training data, hence the network is put outside its ‘comfort zone’, i.e. a wrong decision could be made by the network. This may results in potential hazards to human especially in those scenarios with human involvements. Hence, inspired by ideas from [44], this work presents an improved real-time activation pattern monitoring algorithm for monitoring the R-CNN features representations for different image inputs. The patterns of the ‘neurons’ inside the Faster R-CNN are monitored with different data. The approach uses the Hamming distance to characterise the difference and similarity between binary strings and the this is combined with the Kullback-Leibler divergence.

The main contributions of this work consist in the following: 1) distributions of the neuron activation patterns are calculated using the Hamming distance between the

current activation pattern and the central activation pattern. Next, the closeness of these distributions is characterised based on Kullback-Leibler divergence; 2) Monitoring zones are constructed based on decision making, by taking the patterns with the corresponding probability values and the changes in the patterns are visualised; 3) The efficiency of the monitoring framework is demonstrated over MNIST and PASCAL datasets.

The remainder of this chapter is organised as follows. The methodology proposed is elaborated in Section 5.2. Section 5.3 provides the experimental results and analysis, and the chapter is concluded in Section 5.4.

5.2 Methodology

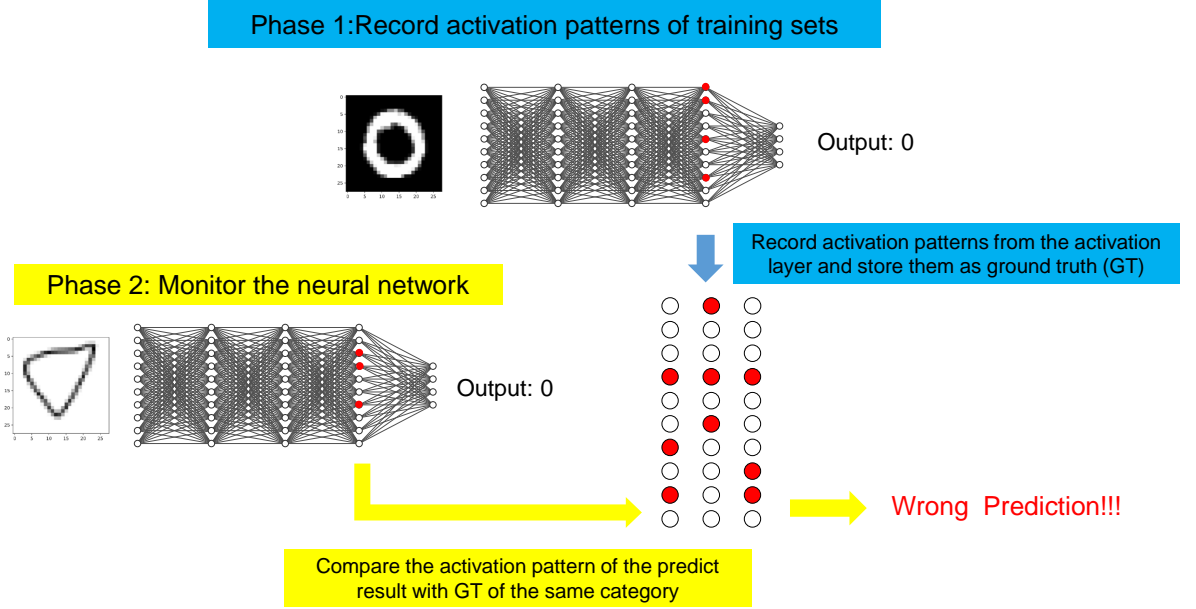


Figure 5.1: Overview of real-time activation pattern monitoring. The framework includes two phrases. In Phase 1, the activation patterns are first recorded. the central activation pattern of each class is found based on their similarities. In Phase 2, the network is monitored when a new image is fed to the network and the activation situation is monitored.

5.2.1 Activation Pattern Representation

A DNN model is defined as $\mathbf{y} = \mathbf{F}(\boldsymbol{\theta}, \mathbf{x})$ with the DNN hyperparameter $\boldsymbol{\theta}$ where \mathbf{y} is the output of the model and \mathbf{x} is the input of the model. The model can classify $\{c_1, \dots, c_l\} \in C$ classes and consequently, $\mathbf{y} \in C$.

The common activation function applied in the activation layers of DNN models in this chapter is the ReLU function that is in the form of

$$\sigma(a) = \max\{a, 0\}, \quad (5.1)$$

where a is the input value of ReLU function.

In this chapter, a neuron in the activation layer is considered as *activated* when its output is greater than zero. The output of the last activation layer in a DNN model is denoted as $\{v_1, \dots, v_d\}$, with d the dimension of the last activation layer. The architecture of the proposed activation pattern monitoring approach is shown on Fig 5.1. In Phase 1, activation patterns are systematically recorded. Subsequently, the central activation pattern for each class is determined by assessing the similarities among the recorded patterns. In Phase 2, the network undergoes real-time monitoring as new images are introduced. This allows for the observation and analysis of the activation responses within the network.

The binary activation pattern can be defined as follows:

Definition 1 (Binary Activation Pattern) Given the output of the last activation layer $\{v_1, \dots, v_d\}$, the activation pattern of a certain class c is defined as

$$P^c = \left(p(v_1), \dots, p(v_d) \right), \quad (5.2)$$

where $p(\cdot)$ defined in (5.3) is a function that maps a real number $v \in \mathbb{R}$ into binary numbers:

$$p(v) = \begin{cases} 1 & v > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (5.3)$$

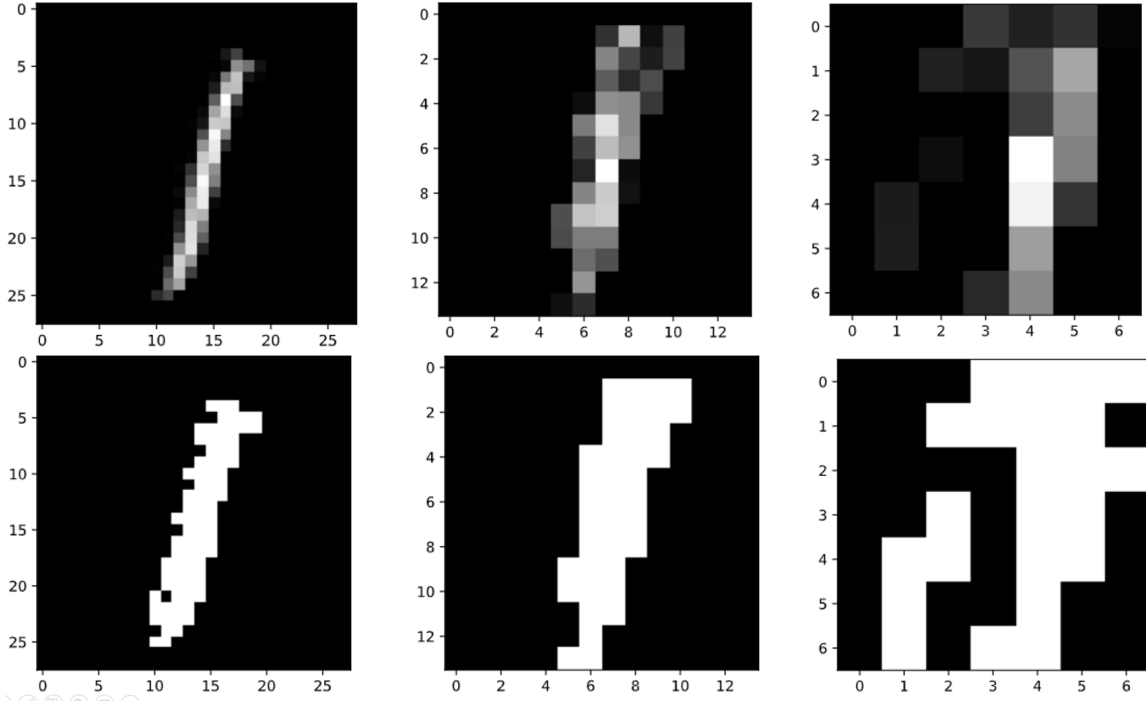


Figure 5.2: *Visualisation of activation layers (first row) and the corresponding activation patterns (second row) of Number 1 in MNIST dataset. The activation pattern becomes more abstract from left to right as the layer in DNNs gets deeper.*

For datasets such as MNIST and PASCAL, there are more than one class to be detected and classified. For clarity, let \mathcal{T} denote the training dataset, and $\mathcal{T}_c \subseteq \mathcal{T}$ denote images in the training dataset contain objects with a certain class c . The activation patterns of \mathcal{T}_c can then be organised as

$$\mathcal{P}_c = \left\{ P_0^c, \dots, P_i^c, \dots, P_n^c \right\}, \quad (5.4)$$

where n indicates the number of patterns of class c . Activation patterns of the whole training dataset can be defined similarly and denoted as \mathcal{P} , with $\mathcal{P}_c \subseteq \mathcal{P}$ stands. Figure 5.2 show examples of different activation patterns from different layers of Number 1 in MNIST dataset [18].

5.2.2 Central Activation Patterns

Given a class c , similar activation patterns for objects are expected to be contained in various input images. It is able to accumulate activation patterns of class c during the training process, and therefore find a central activation pattern that can represent class c for further applications.

The *central activation pattern* \tilde{P}^c of class c is defined as

$$\tilde{P}^c \triangleq \arg \max_P \sum_{i=0}^n \mathbf{H}(P, P_i^c), P_i^c \in \mathcal{P}_c, \quad (5.5)$$

where $\mathbf{H}(\cdot, \cdot)$ indicates the Hamming distance between two binary patterns P and P_i^c .

In this chapter, the Dynamic Programming (DP) algorithm [201] is exploited to solve (5.5). In the following, the application of DP in this case is summarised.

Here minimum sum of Hamming distances are denoted as followed:

$$\tau^c[j] = \min \sum \mathbf{H}(P[:j], P_i^c[:j]), \quad (5.6)$$

where $[:j]$ represents neurons from first to j -th in activation patterns and there are d neurons in total. Since the neurons in the activation patterns are independent on each other, the iterative update rule of $\tau^c[j]$ is,

$$\tau^c[j] = \tau^c[j-1] + \min \sum \mathbf{H}(q_j^c, p_i^c(v_j)), \quad (5.7)$$

where $q_j^c \in \{0, 1\}$ is the j -th neuron and $p_i^c(v_j)$ is j -th neuron of activation pattern P_i^c . By inferring q_j^c from 1 to d where d is also the dimension of activation patterns, that is

$$\tilde{P}^c[j] \triangleq \arg \min_{q_j^c} \sum \mathbf{H}(q_j^c, p_i^c(v_j)), \quad (5.8)$$

the central activation pattern of class c :

$\tilde{P}^c = (q_1^c, \dots, q_d^c)$ is finally obtained.

5.2.3 Activation Pattern Distance Distribution

So far the central activation pattern \tilde{P}^c of \mathcal{P}_c is obtained with the DP algorithm and a set of activation patterns \mathcal{P}_c is already recorded. They share the same class c . First the Hamming distance between the central activation pattern \tilde{P}^c and every activation pattern P_i^c from the considered set is calculated. Then a set of results for the Hamming distance is obtained. Then the Hamming distance is used to calculate sub-intervals.

While the central activation pattern \tilde{P}^c is representative for a certain class c , the extraction of activation patterns and comparison with \tilde{P}^c remains a challenge for real time activation pattern monitoring. The situation gets worse when the dimension of the activation pattern increases. To cope with the challenge, this chapter further propose the activation pattern distribution, which aims at distinguishing difference classes efficiently.

Given \mathcal{P}_c and \tilde{P}^c , the Hamming distances between $P_i^c \in \mathcal{P}_c$ and \tilde{P}^c , with $i = 0, \dots, n$ are first calculated, which are denoted as $\mathcal{D}_c = \{D_0^c, \dots, D_i^c, \dots, D_n^c\}$. Then the interval $[\min(\mathcal{D}_c), \max(\mathcal{D}_c)]$ is partitioned into m sub-intervals evenly and calculate the number of distances falling into each sub-interval. Let's denote the results as $N^c = \{N_0^c, \dots, N_j^c, \dots, N_m^c\}$, then the activation pattern distribution α is defined as follows

$$\alpha = N^c/n = \left\{ N_0^c/n, \dots, N_j^c/n, \dots, N_m^c/n \right\}. \quad (5.9)$$

With (5.9), the distribution of each class can be calculated. To distinguish different classes, the Kullback-Leibler (KL) divergence is employed as a metric, which is defined as

$$KL(\alpha||\beta) = \sum_j \alpha(j) \log \left(\frac{\alpha(j)}{\beta(j)} \right), \quad (5.10)$$

where β represents a distribution where the classes between the central activation pattern \tilde{P}^c and the activation pattern set \mathcal{P}_{c^*} are different between c and c^* . In α the \mathcal{P}_c and \tilde{P}^c share the same class c .

5.2.4 Choice of Thresholds and Monitoring Zones

In this chapter, two types of distance distributions of the neurones' patterns are investigated compared to the central activation patterns. The first type is denoted as 'Same', which indicates that the activation patterns and the central patterns are from the same object class. On the contrary, 'Different' is used to indicate that the activation patterns are from objects of different class compared to that of the central patterns. To distinguish the two pattern distance distributions, two thresholds S_0 and $S_{0.05}$ are defined to build three monitoring zones: $(0, S_0)$, $(S_0, S_{0.05})$ and $(S_{0.05}, +\infty)$.

The threshold S_0 characterises the shortest distance between the central activation pattern and activation patterns with different object class from the central activation patterns, i.e., the very first recorded distance from 0 in 'Different' seen in Figure 5.3 and Figure 5.4.

The $S_{0.05}$ threshold represents the distance where the accumulative probability from 0 of 'Different' distribution is 5%, i.e., $\int_0^{+\infty} \text{Dist. Different}'(x')dx' = 0.05$, and $S_{0.05} = x'$.

Therefore, the interval $(0, S_0)$ can be defined as a comfort zone which means the predicted result is trusted. When distances between prediction activation patterns and central activation patterns are in $(S_0, S_{0.05})$, it will be considered as a 'warning signal' which requires extra attention (manual) to aid decision-making of neural networks. As for distances in $(S_{0.05}, +\infty)$, the predictions are taken as 'not trust-able'.

5.2.5 The Activation Pattern Monitoring Algorithm

In the proposed monitoring algorithm, activation states of neurons from the close-to-output layer of the DNN model is monitored. To accomplish this, a two-phase algorithm is implemented as depicted in Fig. 5.1. The details are given in **Algorithm 1**. In Phase 1, the pre-trained model is fed with the training dataset again and the activation patterns of training samples will be recorded and stored as the ground-truth (GT). After Phase 1, when a new input comes to the model, the activation pattern and

prediction of the model will be the output. In Phase 2, the activation pattern is compared with the GT with the same label to find out their differences by calculating their Hamming distances. If the distance is larger than a threshold, the prediction is defined as a problematic decision that is unacceptable.

Algorithm 1 Real-time Activation Pattern Monitoring

Phase 1: Record activation patterns of training set \mathcal{T}

for $\mathcal{T}_c \subseteq \mathcal{T}$ **do**

for $\mathbf{x} \in \mathcal{T}_c$ **do**

$y \leftarrow \mathbf{F}(\mathbf{x})$ and P^c is the activation pattern of \mathbf{x}

if $y = c$ **then**

$\mathcal{P}_c \leftarrow \mathcal{P}_c \cup P^c$

end if

end for

end for

/★ Generate central patterns of different classes ★/

Phase 2: Monitor the Neural Network

$y' \leftarrow \mathbf{F}(\mathbf{x}')$ and P' is the activation pattern of \mathbf{x}'

for $c \in C$ **do**

if $y' = c$ **then**

 /★ Calculate the shortest Hamming distance between P' and \mathcal{P}_c ★/

$\text{Dist} \leftarrow \mathbf{H}(P', \mathcal{P}_c)$

if $\text{Dist} \in (0, S_0)$ **then**

 Print “ y' is trusted”

else if $\text{Dist} \in (S_0, S_{0.05})$ **then**

 Print “Require Human Judgement”

else

 Print “ y' is not trusted”

end if

end if

end for

5.3 Experiments and Analysis

To verify the proposed algorithm, it is applied into two tasks: 1) image classification on the MNIST dataset; 2) object detection on the PASCAL dataset.

Image classification is a fundamental task in CV that involves assigning a label to an entire image based on its content. The objective is to categorise the image into one of several predefined classes [25]. while object detection is a more advanced task in computer vision where the goal is to identify and locate multiple objects within an image. This involves not only classifying objects but also drawing bounding boxes around them to indicate their positions [27]. With respect to monitoring the activation pattern, implementing such an algorithm in object detection is more challenging. Multiple objects in the same image may affect each other's activation situation and confuse the DNN. Consequently, the location of objects should be considered when monitoring activation patterns.

5.3.1 Datasets and Implementation Details

Classification on MNIST Dataset

MNIST [18] dataset is a digital hand-written dataset which contains number 0-9, where the training dataset contains 60,000 images while the testing dataset consists of 10,000 images. In this chapter, activation patterns of the last activation layer with 40 neurons are monitored.

Table 5.1 presents the classification results on MNIST datasets. Activation patterns from those images with correct predictions on training set are treated as the ground-truth activation patterns to generate central activation pattern \tilde{P} of different classes.

Datasets	Correct	Wrong	Accuracy
Training	59,605	395	99.34%
Testing	9,881	119	98.81%

Table 5.1: *Prediction Results on MNIST*

Object Detection on the PASCAL Dataset

The PASCAL VOC 2007 dataset [19] contains 20 classes with around 10k images and 24k annotated objects. There are mainly four categories, i.e. **Vehicles**, **Households**, **Animals**, and **Person**. Each category contains several object classes. They are listed in Table 5.2 and numbered them to facilitate further descriptions [202].

Vehicles	Households	Animals	Person
Aeroplane: 0	Bottle: 4	Bird: 2	Person: 14
Bicycle: 1	Chair: 8	Cat: 7	
Boat: 3	Dining table: 10	Cow: 9	
Bus: 5	Potted plant: 15	Dog: 11	
Car: 6	Sofa: 17	Horse: 12	
Motorbike: 13	TV/Monitor: 19	Sheep: 16	
Train: 18			

Table 5.2: *PASCAL object classes*

In classification tasks, the ground-truth activation patterns are simply from those activation patterns with correct classifications on training set. Different from classification tasks, it is hard to extract the ground-truth activation patterns in object detection by using the same strategy.

Faster R-CNN[17] is implemented for detecting and recognising PASCAL objects in this chapter. Different from extracting the last activation layers only in the classification task, objects are needed to be detected, i.e. determine labels and bounding boxes of the

objects first, then the corresponding patterns in the close-to-output activation layers are extracted.

In object detection, both predicted labels and intersection over union (IoU) between the predicted bounding boxes and the ground-truth bounding boxes should be considered and the IoU is defined as

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \quad (5.11)$$

The predictions with $IoU > 0.5$ are defined as True Positive (TP), while those with IoU lower than 0.5 are defined as False Positive (FP) [19]. The number of TP/FP in training set and testing set is shown in Table 5.3.

Datasets	TP	FP	Accuracy
Training	12,411	26,429	32%
Testing	11,133	28,208	28%

Table 5.3: Numbers of TP&FP on PASCAL.

In this case, activation patterns of TP in the training set are treated as the ground-truth patterns and apply them to generate central activation patterns for 20 classes.

5.3.2 Validation Results and Analysis

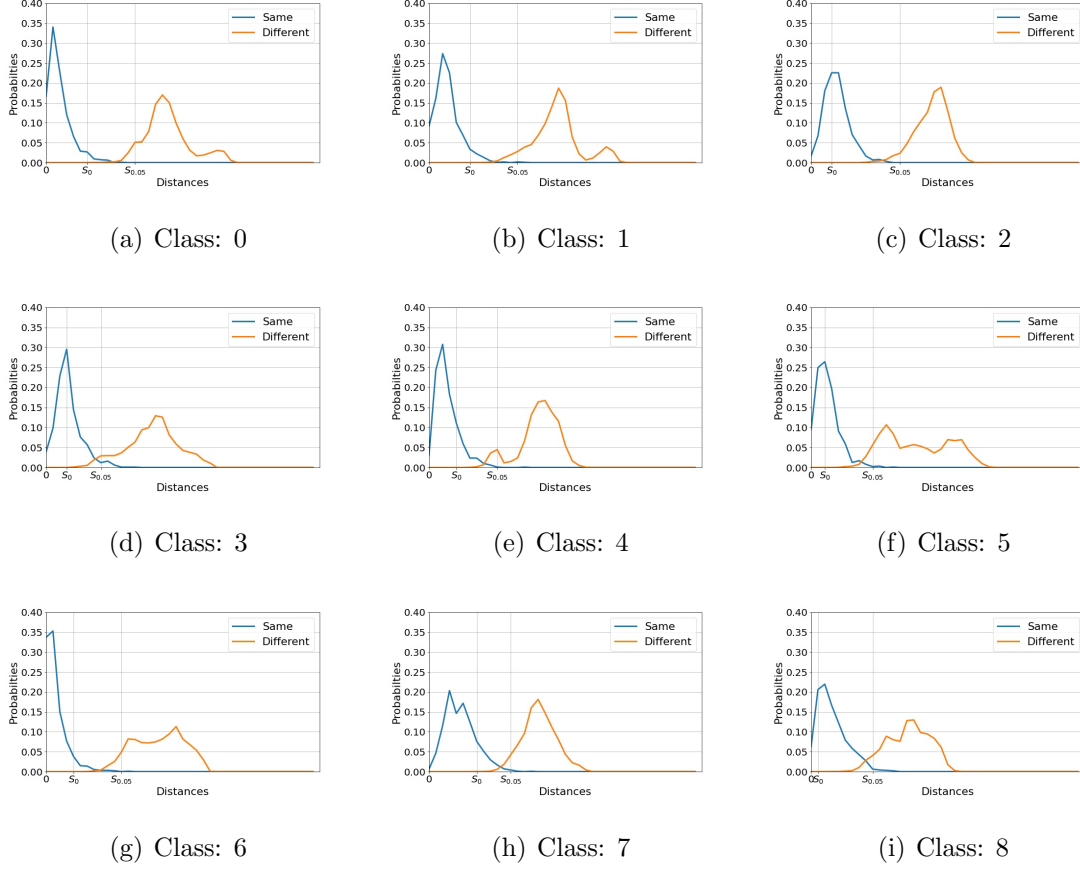


Figure 5.3: From (a) to (i) represent the activation pattern distributions of digital number from 0 to 8 on MNIST dataset. 'Same' represents the distribution which the central activation pattern is compared to the activation patterns with the same class, while 'Different' represents the distribution which the central activation pattern is compared to the activation patterns with different classes.

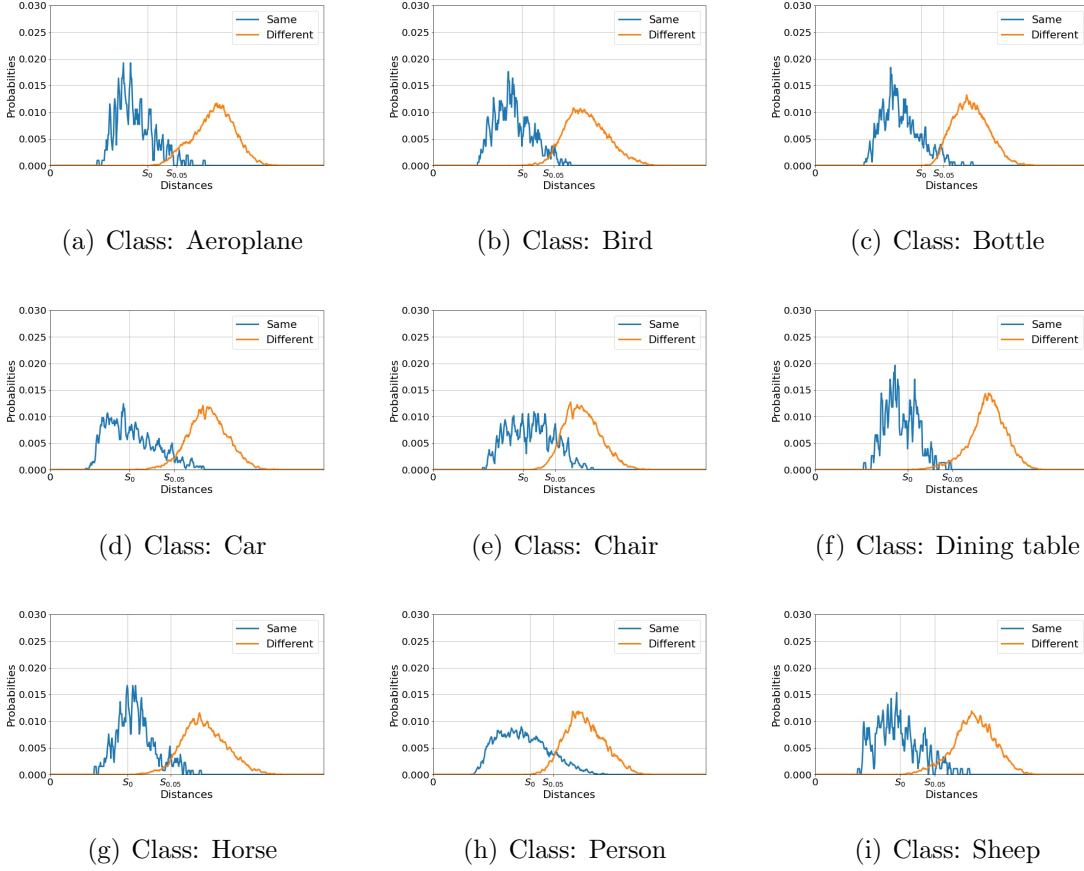


Figure 5.4: From (a) to (i) represent the activation pattern distributions of different classes on PASCAL dataset. 'Same' represents the distribution which the central activation pattern is compared to the activation patterns with the same class, while 'Different' represents the distribution which the central activation pattern is compared to the activation patterns with different classes.

Fig. 5.3 and Fig. 5.4 show distributions of distances between the central patterns and activation patterns. In both figures, The term 'Same' is used to represent activation patterns with the same classes as the central activation patterns while 'Different' are those patterns with different classes. From these two distributions under different classes, activation patterns with the same classes as the central activation have shorter

distances compared to those with different classes.

Each subplot from (a) to (i) in Fig. 5.3 corresponds to a digit class. The 'Same' curve (orange) shows the probability distribution of distances when comparing the central activation pattern to other activation patterns of the same class. The 'Different' curve (blue) represents the distribution of distances when the central activation pattern is compared to activation patterns of different classes. The 'Same' distributions tend to peak at lower distances, indicating a high similarity within the same class. In contrast, the 'Different' distributions are spread out, indicating greater variation and less similarity when compared to patterns from different classes.

Similar to the MNIST dataset, these plots in Fig. 5.4 show the activation pattern distributions for various object classes (e.g., Aeroplane, Bird, Car, etc.). Again, the 'Same' distributions generally indicate smaller distances, suggesting that the activation patterns are more similar within the same object class compared to different ones. The PASCAL dataset plots show more variability in the 'Different' distributions compared to the MNIST dataset, which may reflect the higher complexity and variability in real-world image data compared to the simpler handwritten digits of MNIST.

By analysing Hamming distance distributions of activation patterns as well as their Kullback-Leibler divergences, it is confirmed that activation patterns with the same class are clustered to their corresponding central activation pattern. As for a class, by using 'Same'/'Different' distributions, three monitoring zones can be built for monitoring the decision made by a neural network.

		Autonomous	Manual	Mis-
Dataset		Correct Clas-	Human	classified
		sification	Decision	
MNIST	Training	63.77%	35.56%	0.66%
	Testing	62.62%	36.73%	0.65%
PASCAL	Training	60.16%	36.24%	3.60%
	Testing	55.69%	38.99%	5.32%

Table 5.4: *Monitoring Classification Results on MNIST and PASCAL.*

Monitoring experiments are implemented on the MNIST and PASCAL datasets and the results are shown in Table 5.4. As for a test image, the neural network outputs a predicted result and its activation pattern. The distance between the pattern and its corresponding central activation pattern is calculated and to which zone the distance belongs to is also obtained. Table 5.4 presents the monitoring classification results in different datasets. The term ‘Autonomous Correct Classification’ is used to represent Faster R-CNN correct decisions - when the neural network works well. ‘Manual’ means additional human involvement is made in the decision making. ‘Misclassified’ represents that the proposed algorithm misclassifies the prediction made by the network. What can be identified is that the proposed algorithm achieved low misclassified monitoring results, i.e. over 99% accuracy of prediction in both training and testing sets of MNIST. As for complicated object detection tasks, the monitoring process within Faster R-CNN can also achieve a good performance with over 96% accuracy in the training phase and 94% accuracy during testing over PASCAL datasets.

These results are crucial for real-time monitoring of DNNs, particularly in determining when a network may be making decisions that do not align well with what it learned during training. By setting appropriate thresholds on these distributions, the system can categorise the predictions into trusted, review-needed, or not-trusted, enhancing

the safety and reliability of applications relying on DNNs. This mechanism is especially valuable in applications like autonomous driving or medical image analysis, where incorrect decisions can have serious consequences.

5.4 Conclusions

This chapter presents a real-time activation pattern monitoring algorithm of the Faster R-CNN in image classification and object detection. The real-time activation pattern monitoring algorithm is introduced to provide extra resilience in decision making for DNNs based systems. First the Kullback-Leibler divergence is calculated to find how different two distributions of the monitored patterns are. Next, the Hamming distance is calculated for decision making purposes. It gives the distance between the activation pattern of the current input and the corresponding central activation pattern. In this way a monitoring zone is represented and gives a level of trust in the obtained results. The proposed monitoring algorithm has been thoroughly verified over two different computer vision tasks: image classification and object detection - with MNIST and PASCAL data sets - and demonstrates its capacity and achieves very good monitoring performances.

Chapter 6

Conclusions and Future Works

This thesis has presented a comprehensive exploration of the integration of deep learning and computer vision in autonomous systems, particularly within the context of manufacturing and Industry 5.0. The research underscored the pivotal role of artificial intelligence in revolutionising modern manufacturing processes, from predictive maintenance and quality control to supply chain optimisation and human-robot collaboration.

In Chapter 1, it provides a thorough introduction to the integration of AI in manufacturing, which is a significant advancement towards Industry 5.0. This chapter discusses the vital role AI plays across various manufacturing processes. The chapter also highlights the evolving nature of robotics in manufacturing, with a focus on the development of cobots that work alongside humans to create safer and more productive environments. Deep Learning, particularly when combined with CV, is emphasised as crucial for enabling these advancements, allowing robots to perform complex tasks with high precision. The chapter addresses challenges such as the need for robust machine learning models, data acquisition for model training, and the integration of these systems into existing infrastructures. The objectives of the thesis are outlined, focusing on improving the robustness of Deep Learning applications in HRC within manufac-

turing settings. This involves leveraging Digital Twin technology to bridge the gap between simulated environments and real-world applications, enhancing the effectiveness of physical HRC, and ensuring the reliability of deep learning models through real-time activation pattern tracking and uncertainty characterisation in image classification.

In Chapter 2, it presents a thorough examination of the literature on deep learning and computer vision, which are crucial elements in contemporary AI research. It covers basic knowledge in the field and explores how deep learning enhances computer vision tasks such as classification and object detection. The chapter delves into both fully-supervised and semi-supervised approaches, addressing uncertainty and activation pattern monitoring in deep learning. It also examines human pose estimation, discussing 2D and 3D approaches and challenges with occlusions, and concludes with insights into human-robot interaction, focusing on the transition from simulation to reality and the role of digital twins in manufacturing.

In Chapter 3, it discusses the development of a Digital Twin framework for enhancing safety in manufacturing systems. The framework utilises deep learning techniques to detect and classify human and robot actions. The Digital Twin allows users to build a virtual representation of the physical manufacturing workspace by incorporating CAD models of real objects. The communication between the physical and digital systems is established using the ROS. One of the advantages of this framework is its flexibility. Users can easily add new objects to the digital system by introducing their CAD models, and they can specify annotation methods to meet their requirements. The framework employs the Sim2Real technique, which enables efficient data generation and semi-supervised learning, reducing the need for extensive data collection and manual annotation. This chapter also discusses the impact of lighting conditions on detection performance. It introduces the use of the Kalman filter and the Hungarian algorithm to enhance the detector's performance and maintain safety distances between humans and robots. In general, the Digital Twin framework combined with deep learning techniques

has the potential to improve safety and efficiency in smart manufacturing. Future work may involve exploring the integration of cloud computing services and reinforcement learning training in Digital Twins.

In Chapter 4, it explores the utilisation of deep learning techniques in the field of human-robot interaction, with a specific focus on the application of these techniques in robot-assisted dressing tasks. It presents a framework that combines a CNN-KF to estimate the 3D human pose from a single camera, and a numerical inverse kinematic (IK) solver to update the kinematic model of the human upper limb. The framework is robust to occlusions and spontaneous movements of the user, and ensures ergonomic and safe solutions for the robot motion planning. This chapter evaluates the precision and performance of the framework using data collected from dressing experiments with healthy volunteers and an occupational therapist. This chapter also discusses the challenges and future directions of research.

In Chapter 5, it investigates the performance of DNNs, specifically faster R-CNNs, in image classification when the testing data differ significantly from the training data. This chapter proposes a framework for monitoring the activation patterns within a faster R-CNN by representing distributions of neuron activation patterns and calculating corresponding distances between them using the Kullback-Leibler divergence. This allows for the observation of the activation states of neurons within the network when it is working with noisy and significantly different data. The proposed framework is validated on publicly available datasets, MNIST and PASCAL, and demonstrates real-time monitoring of supervised classifiers. This chapter also discusses the quantification of uncertainties in CNNs and the importance of identifying when a trained CNN model performs inference correctly. The monitoring algorithm is implemented in two phases: recording the activation patterns of the training dataset and comparing the activation patterns of new inputs with the ground-truth patterns. The chapter includes experimental results and analysis on the MNIST and PASCAL datasets, showing the effectiveness of the proposed algorithm in monitoring the decision-making of neural

networks.

6.1 Future Work

- **More challenging cases in manufacturing scenarios can be considered.**

Chapter 3 explores the monitoring of human-robot actions. However, it is important to consider the complex manufacturing environment in future research. For instance, future work will concentrate on more difficult scenarios involving multiple robots and operators. In addition to object detection, gesture recognition and pose estimation will also be taken into account to identify the actions of both human operators and robots. This will enable more sophisticated decision-making and control, providing greater flexibility and enhancing the system's resilience in complex tasks.

- **Integration of Multiple Sensors Fusion in Deep Learning Models for Enhanced HRC**

Future research should explore the integration of multiple sensors fusion in Deep Learning models to significantly enhance the robustness and accuracy of HRC. By leveraging data from diverse sensors, such as cameras, LiDAR, microphones, and inertial measurement units, robots can achieve a more comprehensive understanding of their environment and humans. This multi-modal approach can help overcome the limitations of individual sensors, leading to improved situational awareness and decision-making. Furthermore, advanced fusion techniques, such as attention mechanisms and graph neural networks, can be employed to effectively combine sensor data at various levels, ensuring real-time processing and response. This integration is expected to not only enhance task performance but also ensure safety and adaptability in dynamic and unstructured environments, paving the way for more intuitive and effective human-robot interactions.

- **Large language models [203] boost the development of HRC.** A large

language model is a type of AI system that is trained on a massive amount of text data to understand and generate human language. These models have gained significant attention and popularity for their ability to generate coherent and contextually relevant text, making them useful for various applications in fields like chatbots, virtual assistants, content generation, and even research in natural language understanding and generation. Several challenges of leveraging large language models (LLMs) for decision-making in robotic tasks has been discussed [204]. While LLMs have a wealth of semantic knowledge, it has the potential to transfer this kind of knowledge to train and teach the robot to finish more complicated tasks and has the ability to understand human's behaviour and implement intelligent decision-making.

- **Generative models bring the potential to bridge the gap between simulation to real in HRC.** Vast dataset is a crucial element for the development of AI. The availability of extensive and high-quality data significantly impacts the performance and capabilities of AI models. Large datasets enable the training of more complex and accurate models, allowing for better generalization and robustness in real-world applications [205]. The stable diffusion model, as exemplified by Zhang et al. [206], is capable of producing realistic and high-resolution images. These models have been successfully employed to generate visually appealing images with diverse variations. Furthermore, they can generate images based on textual inputs. In the field of robot learning, stable diffusion models demonstrate a remarkable capacity for zero-shot training without the need for additional data acquisition. In future research, these models could be utilized to generate photorealistic images with a high level of semantic knowledge, thereby enhancing the training of AI models in robot learning domains.
- **Identify the differences between simulation and reality via activation pattern monitoring.** Chapter 5 presents an algorithm for monitoring activation patterns. The algorithm enhances the decision-making process in systems

based on DNNs when the testing domain differs from the training domain. Additionally, the algorithm can be used to detect disparities between simulation and reality in HRC. This helps robots make robust decisions and enhances safety in manufacturing environments.

Reference

- [1] M. Ghahramani, Y. Qiao, M. C. Zhou, A. O'Hagan, and J. Sweeney, "Ai-based modeling and data-driven evaluation for smart manufacturing processes," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1026–1037, 2020.
- [2] J. Xu, M. Kovatsch, D. Mattern, F. Mazza, M. Harasic, A. Paschke, and S. Lucia, "A review on ai for smart manufacturing: Deep learning challenges and solutions," *Applied Sciences*, vol. 12, no. 16, p. 8239, 2022.
- [3] S. Sundaram and A. Zeid, "Artificial intelligence-based smart quality inspection for manufacturing," *Micromachines*, vol. 14, no. 3, p. 570, 2023.
- [4] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1395–1476, 2021.
- [5] A. M. Djuric, R. Urbanic, and J. Rickli, "A framework for collaborative robot (cobot) integration in advanced manufacturing systems," *SAE International Journal of Materials and Manufacturing*, vol. 9, no. 2, pp. 457–464, 2016.
- [6] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, pp. 1–21, 2015.
- [7] S. Sahoo, S. Kumar, M. Z. Abedin, W. M. Lim, and S. K. Jakhar, "Deep learning applications in manufacturing operations: a review of trends and ways forward,"

-
- Journal of Enterprise Information Management*, vol. 36, no. 1, pp. 221–251, 2023.
- [8] L. Zhou, L. Zhang, and N. Konz, “Computer vision techniques in manufacturing,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 105–117, 2022.
- [9] V. De Simone, V. Di Pasquale, and S. Miranda, “An overview on the use of ai/ml in manufacturing msms: solved issues, limits, and challenges,” *Procedia Computer Science*, vol. 217, pp. 1820–1829, 2023.
- [10] S. J. Plathottam, A. Rzonca, R. Lakhnori, and C. O. Iloeje, “A review of artificial intelligence applications in manufacturing operations,” *Journal of Advanced Manufacturing and Processing*, vol. 5, no. 3, p. e10159, 2023.
- [11] K. Xu, Y. Li, C. Liu, X. Liu, X. Hao, J. Gao, and P. G. Maropoulos, “Advanced data collection and analysis in data-driven manufacturing process,” *Chinese Journal of Mechanical Engineering*, vol. 33, no. 1, pp. 1–21, 2020.
- [12] S. Sudhakar, J. Hanzelka, J. Bobillot, T. Randhavane, N. Joshi, and V. Vineet, “Exploring the sim2real gap using digital twins,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 418–20 427.
- [13] A. Baratta, A. Cimino, M. G. Gnoni, and F. Longo, “Human robot collaboration in industry 4.0: a literature review,” *Procedia Computer Science*, vol. 217, pp. 1887–1895, 2023.
- [14] M. Faccio, I. Granata, and R. Minto, “Task allocation model for human-robot collaboration with variable cobot speed,” *Journal of Intelligent Manufacturing*, pp. 1–14, 2023.
- [15] J. Cai, A. Du, X. Liang, and S. Li, “Prediction-based path planning for safe and efficient human–robot collaboration in construction via deep reinforcement

- learning,” *Journal of Computing in Civil Engineering*, vol. 37, no. 1, p. 04022046, 2023.
- [16] G. Boschetti, M. Faccio, I. Granata, and R. Minto, “3d collision avoidance strategy and performance evaluation for human–robot collaborative systems,” *Computers & Industrial Engineering*, vol. 179, p. 109225, 2023.
- [17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
- [18] Y. LeCun, C. Cortes, and C. Burges, “MNIST handwritten digit database,” 2010.
- [19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [20] L. Zhou, L. Zhang, and N. Konz, “Computer vision techniques in manufacturing,” vol. 53, no. 1, pp. 105–117.
- [21] J. West, “Machine vision in the real world of manufacturing,” *Computer Design*, vol. 22, no. 5, pp. 89–96, 1983.
- [22] G. J. Agin, “Computer vision systems for industrial inspection and assembly,” *Computer*, vol. 13, no. 05, pp. 11–20, 1980.
- [23] A. Soini, “Machine vision technology take-up in industrial applications,” in *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat. IEEE*, 2001, pp. 332–338.
- [24] S. V. Mahadevkar, B. Khemani, S. Patil, K. Kotecha, D. Vora, A. Abraham, and L. A. Gabralla, “A review on machine learning styles in computer vision-techniques and future directions,” *IEEE Access*, 2022.

- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
- [28] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [30] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, “A survey of deep learning techniques for autonomous driving,” *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.
- [31] J. Xu, “A deep learning approach to building an intelligent video surveillance system,” *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 5495–5515, 2021.
- [32] G. Quellec, G. Cazuguel, B. Cochener, and M. Lamard, “Multiple-instance learning for medical image and video analysis,” *IEEE Reviews in Biomedical Engineering*, vol. 10, pp. 213–234, 2017.
- [33] S. Dargan, S. Bansal, M. Kumar, A. Mittal, and K. Kumar, “Augmented reality: A comprehensive review,” *Archives of Computational Methods in Engineering*, vol. 30, no. 2, pp. 1057–1080, 2023.

-
- [34] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [35] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.
- [36] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, “FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence,” *arXiv:2001.07685 [cs, stat]*, Nov. 2020, arXiv: 2001.07685.
- [37] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister, “A simple semi-supervised learning framework for object detection,” *arXiv preprint arXiv:2005.04757*, 2020.
- [38] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, “End-to-end semi-supervised object detection with soft teacher,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 3040–3049.
- [39] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, 2021.
- [40] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proceedings of International Conference on Machine Learning*, 2016, pp. 1050–1059.
- [41] A. Kendall and Y. Gal, “What uncertainties do we need in Bayesian deep learning for computer vision?” in *Proceedings of International Conference on Neural*

- Information Processing Systems (NIPS)*, September 2017, p. 5580–5590.
- [42] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” 2017.
- [43] M. Leucker and C. Schallhart, “A brief account of runtime verification,” *The Journal of Logic and Algebraic Programming*, vol. 78, no. 5, pp. 293–303, 2009.
- [44] C. Cheng, G. Nührenberg, and H. Yasuoka, “Runtime monitoring neuron activation patterns,” in *Proceedings of 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 300–303.
- [45] C.-H. Cheng, C.-H. Huang, and G. Nührenberg, “nn-dependability-kit: Engineering neural networks for safety-critical autonomous driving systems,” *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–6, 2019.
- [46] C.-H. Cheng, C.-H. Huang, T. Brunner, and V. Hashemi, “Towards safety verification of direct perception neural networks,” *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1640–1643, 2020.
- [47] C.-H. Cheng, “Provably-robust runtime monitoring of neuron activation patterns,” *ArXiv*, vol. abs/2011.11959, 2020.
- [48] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, “Total capture: 3d human pose estimation fusing video and inertial sensors,” in *Proceedings of 28th British Machine Vision Conference*, 2017, pp. 1–13.
- [49] H.-Y. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki, “Self-supervised learning of motion capture,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [50] N. Gupta, S. K. Gupta, R. K. Pathak, V. Jain, P. Rashidi, and J. S. Suri, “Human activity recognition in artificial intelligence framework: A narrative review,” *Artificial Intelligence Review*, vol. 55, no. 6, pp. 4755–4808, 2022.

-
- [51] J. Laplaza, J. J. Oliver, R. Romero, A. Sanfeliu, and A. Garrell, “Body gesture recognition to control a social robot,” *arXiv preprint arXiv:2206.07538*, 2022.
 - [52] F. Semeraro, A. Griffiths, and A. Cangelosi, “Human–robot collaboration and machine learning: A systematic review of recent research,” *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102432, 2023.
 - [53] A. G. da Silva, M. V. Mendes Gomes, and I. Winkler, “Virtual reality and digital human modeling for ergonomic assessment in industrial product development: A patent and literature review,” *Applied Sciences*, vol. 12, no. 3, p. 1084, 2022.
 - [54] J. Martinez, R. Hossain, J. Romero, and J. J. Little, “A simple yet effective baseline for 3d human pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2640–2649.
 - [55] D. C. Luvizon, D. Picard, and H. Tabia, “2d/3d pose estimation and action recognition using multitask deep learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5137–5146.
 - [56] H. Chen, R. Feng, S. Wu, H. Xu, F. Zhou, and Z. Liu, “2d human pose estimation: A survey,” *Multimedia Systems*, vol. 29, no. 5, pp. 3115–3138, 2023.
 - [57] D. Osokin, “Real-time 2d multi-person pose estimation on cpu: Lightweight openpose,” *arXiv preprint arXiv:1811.12004*, 2018.
 - [58] G. Bertasius, C. Feichtenhofer, D. Tran, J. Shi, and L. Torresani, “Learning temporal pose estimation from sparsely-labeled videos,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
 - [59] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, “Rmpe: Regional multi-person pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2334–2343.

-
- [60] Z. Liu, H. Chen, R. Feng, S. Wu, S. Ji, B. Yang, and X. Wang, “Deep dual consecutive network for human pose estimation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 525–534.
 - [61] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.
 - [62] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, “Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5386–5395.
 - [63] Z. Luo, Z. Wang, Y. Huang, L. Wang, T. Tan, and E. Zhou, “Rethinking the heatmap regression for bottom-up human pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 264–13 273.
 - [64] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, “Coarse-to-fine volumetric prediction for single-image 3D human pose,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7025–7034.
 - [65] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, “3D human pose estimation with spatial and temporal transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 656–11 665.
 - [66] W. Zhao, Y. Tian, Q. Ye, J. Jiao, and W. Wang, “Graformer: Graph convolution transformer for 3D pose estimation,” *arXiv preprint arXiv:2109.08364*, 2021.
 - [67] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, “Semantic graph convolutional networks for 3D human pose regression,” in *Proceedings of the*

-
- IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3425–3435.
- [68] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, “Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 561–578.
- [69] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, “End-to-end recovery of human shape and pose,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7122–7131.
- [70] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, “Neural body fitting: Unifying deep learning and model-based human pose and shape estimation,” in *Proceedings of IEEE International Conference on 3D Vision (3DV)*, 2018, pp. 484–494.
- [71] R. Gu, G. Wang, and J.-N. Hwang, “Exploring severe occlusion: Multi-person 3d pose estimation with gated convolution,” in *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 8243–8250.
- [72] Y. Cheng, B. Yang, B. Wang, W. Yan, and R. T. Tan, “Occlusion-aware networks for 3D human pose estimation in video,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 723–732.
- [73] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, “Pose flow: Efficient online pose tracking,” *arXiv:1802.00977*, 2018.
- [74] M. Ghafoor and A. Mahmood, “Quantification of occlusion handling capability of 3D human pose estimation framework,” *IEEE Transactions on Multimedia*, vol. 25, pp. 3311–3318, 2022.
- [75] Q. Liu, Y. Zhang, S. Bai, and A. Yuille, “Explicit occlusion reasoning for multi-person 3D human pose estimation,” in *Proceedings of European Conference on*

- Computer Vision (ECCV)*. Springer, 2022, pp. 497–517.
- [76] R. Jahanmahin, S. Masoud, J. Rickli, and A. Djuric, “Human-robot interactions in manufacturing: A survey of human behavior modeling,” *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102404, 2022.
- [77] Q. Qi, F. Tao, Y. Zuo, and D. Zhao, “Dgital twin service toowards smart manufacturing,” *Proceedings of the 51st CIRP Conference on Manufacturing Systems*, vol. 72, pp. 237–242, 2018.
- [78] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A. Nee, “Enabling technologies and tools for digital twin,” *Journal of Manufacturing Systems*, vol. 58, pp. 3–21, 2021, digital Twin towards Smart Manufacturing and Industry 4.0.
- [79] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-real transfer in deep reinforcement learning for robotics: a survey,” *arXiv preprint arXiv:2009.13303*, 2020.
- [80] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground truth from computer games,” in *Proceedings of European Conference on Computer Vision (ECCV)*, ser. LNCS, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906. Springer International Publishing, 2016, pp. 102–118.
- [81] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Proceedings of Conference on Robot Learning*. PMLR, 2017, pp. 1–16.
- [82] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.

-
- [83] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Bochoon, and S. Birchfield, “Training deep networks with synthetic data: Bridging the reality gap by domain randomization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 969–977.
- [84] F. Sadeghi and S. Levine, “CAD2RL: Real single-image flight without a single real image,” *arXiv preprint arXiv:1611.04201*, 2016.
- [85] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, “Active domain randomization,” in *Proceedings of Conference on Robot Learning*. PMLR, 2020, pp. 1162–1176.
- [86] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.
- [87] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 740–755.
- [88] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
- [89] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [90] A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, and D. Lange, “Unity: A general platform for intelligent agents,” *arXiv preprint arXiv:1809.02627*, 2018.

-
- [91] Epic Games, “Unreal engine.” [Online]. Available: <https://www.unrealengine.com>
- [92] M. Woo, J. Neider, T. Davis, and D. Shreiner, *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [93] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, “Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 627–12 637.
- [94] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” in *Proceedings of 2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.
- [95] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [96] G. Csurka, “Domain adaptation for visual applications: A comprehensive survey,” *arXiv preprint arXiv:1702.05374*, 2017.
- [97] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [98] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014.

-
- [99] M.-Y. Liu and O. Tuzel, “Coupled generative adversarial networks,” in *Advances in Neural Information Processing Systems*, 2016, pp. 469–477.
 - [100] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, “Pixel-level domain transfer,” in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 517–532.
 - [101] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2107–2116.
 - [102] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, “Unsupervised pixel-level domain adaptation with generative adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3722–3731.
 - [103] M. Vasumathi, A. Khan, M. Sadasivan, and U. Ramamoorthy, “Digital twins—a futuristic trend in data science, its scope, importance, and applications,” in *International Conference on Expert Clouds and Applications*. Springer, 2022, pp. 801–817.
 - [104] M. N. Kamel Boulos and P. Zhang, “Digital twins: from personalised medicine to precision public health,” *Journal of personalized medicine*, vol. 11, no. 8, p. 745, 2021.
 - [105] R. Saracco, “Digital twins: Bridging physical space and cyberspace,” *Computer*, vol. 52, no. 12, pp. 58–64, 2019.
 - [106] S. Barat, V. Kulkarni, T. Clark, and B. Barn, “Digital twin as risk-free experimentation aid for techno-socio-economic systems,” in *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems*, 2022, pp. 66–75.

-
- [107] S. Mihai, M. Yaqoob, D. V. Hung, W. Davis, P. Towakel, M. Raza, M. Karamanoglu, B. Barn, D. Shetve, R. V. Prasad *et al.*, “Digital twins: A survey on enabling technologies, challenges, trends and future prospects,” *IEEE Communications Surveys & Tutorials*, 2022.
- [108] J. Friederich, D. P. Francis, S. Lazarova-Molnar, and N. Mohamed, “A framework for data-driven digital twins for smart manufacturing,” *Computers in Industry*, vol. 136, p. 103586, 2022.
- [109] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.
- [110] M. G. Kapteyn, J. V. Pretorius, and K. E. Willcox, “A probabilistic graphical model foundation for enabling predictive digital twins at scale,” *Nature Computational Science*, vol. 1, no. 5, pp. 337–347, 2021.
- [111] Y. Fu, G. Zhu, M. Zhu, and F. Xuan, “Digital twin for integration of design-manufacturing-maintenance: An overview,” *Chinese Journal of Mechanical Engineering*, vol. 35, no. 1, pp. 1–20, 2022.
- [112] A. A. Malik and A. Brem, “Digital twins for collaborative robots: A case study in human-robot interaction,” *Robotics and Computer-Integrated Manufacturing*, vol. 68, p. 102092, 2021.
- [113] K. Dröder, P. Bobka, T. Germann, F. Gabriel, and F. Dietrich, “A machine learning-enhanced digital twin approach for human-robot-collaboration,” *Proceedia Cirp*, vol. 76, pp. 187–192, 2018.
- [114] Y. Ghasemi, H. Jeong, S. H. Choi, K.-B. Park, and J. Y. Lee, “Deep learning-based object detection in augmented reality: A systematic review,” *Computers in Industry*, vol. 139, p. 103661, 2022.

-
- [115] K.-B. Park, S. H. Choi, J. Y. Lee, Y. Ghasemi, M. Mohammed, and H. Jeong, “Hands-free human–robot interaction using multimodal gestures and deep learning in wearable mixed reality,” *IEEE Access*, vol. 9, pp. 55 448–55 464, 2021.
 - [116] Z. Erickson, H. M. Clever, G. Turk, C. K. Liu, and C. C. Kemp, “Deep haptic model predictive control for robot-assisted dressing,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4437–4444.
 - [117] A. Clegg, Z. Erickson, P. Grady, G. Turk, C. C. Kemp, and C. K. Liu, “Learning to collaborate from simulation for robot-assisted dressing,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2746–2753, 2020.
 - [118] E. Magrini, F. Flacco, and A. De Luca, “Estimation of contact forces using a virtual force sensor,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 2126–2133.
 - [119] K. Ehsani, S. Tulsiani, S. Gupta, A. Farhadi, and A. Gupta, “Use the force, Luke! Learning to predict physical forces by simulating effects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 221–230.
 - [120] N. Koganti, T. Tamei, K. Ikeda, and T. Shibata, “Bayesian nonparametric learning of cloth models for real-time state estimation,” *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 916–931, 2017.
 - [121] E. Pignat and S. Calinon, “Learning adaptive dressing assistance from human demonstration,” *Robotics and Autonomous Systems*, vol. 93, pp. 61–75, 2017.
 - [122] F. Zhang and Y. Demiris, “Learning garment manipulation policies toward robot-assisted dressing,” *Science Robotics*, vol. 7, no. 65, 2022.
 - [123] E. Magrini, F. Ferraguti, A. J. Ronga, F. Pini, A. De Luca, and F. Leali, “Human-robot coexistence and interaction in open industrial cells,” *Robotics and*

Computer-Integrated Manufacturing, vol. 61, p. 101846, 2020.

- [124] V. Villani, F. Pini, F. Leali, and C. Secchi, “Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications,” *Mechatronics*, vol. 55, pp. 248–266, 2018.
- [125] X. Ma, F. Tao, M. Zhang, T. Wang, and Y. Zuo, “Digital twin enhanced human-machine interaction in product lifecycle,” *Procedia Cirp*, vol. 83, pp. 789–793, 2019.
- [126] E. Matheson, R. Minto, E. G. G. Zampieri, M. Faccio, and G. Rosati, “Human–robot collaboration in manufacturing applications: A review,” *Robotics*, vol. 8, no. 4, 2019.
- [127] F. Flacco, T. Kroeger, A. De Luca, and O. Khatib, “A depth space approach for evaluating distance to objects,” *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 7–22, 2015.
- [128] B. Schmidt and L. Wang, “Depth camera based collision avoidance via active robot control,” *Journal of Manufacturing Systems*, vol. 33, no. 4, pp. 711–718, 2014.
- [129] J.-H. Chen and K.-T. Song, “Collision-free motion planning for human-robot collaborative safety under cartesian constraint,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4348–4354.
- [130] R. Y. Tsai, R. K. Lenz *et al.*, “A new technique for fully autonomous and efficient 3D robotics hand/eye calibration,” *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, 1989.
- [131] R. Horaud and F. Dornaika, “Hand-eye calibration,” *The International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, 1995.

-
- [132] C. Y. Siew, S.-K. Ong, and A. Y. Nee, “A practical augmented reality-assisted maintenance system framework for adaptive user support,” *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 115–129, 2019.
- [133] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, and J.-K. Kämäräinen, “AR-based interaction for human-robot collaborative manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 63, p. 101891, 2020.
- [134] S. H. Choi, K.-B. Park, D. H. Roh, J. Y. Lee, M. Mohammed, Y. Ghasemi, and H. Jeong, “An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation,” *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102258, 2022.
- [135] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Q. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J. Big Data*, vol. 8, no. 1, p. 53, 2021.
- [136] J. Fan, P. Zheng, and S. Li, “Vision-based holistic scene understanding towards proactive human–robot collaboration,” *Robotics and Computer-Integrated Manufacturing*, vol. 75, p. 102304, 2022.
- [137] C. Cimino, E. Negri, and L. Fumagalli, “Review of digital twin applications in manufacturing,” *Computers in Industry*, vol. 113, p. 103130, 2019.
- [138] Z. Huang, Y. Shen, J. Li, M. Fey, and C. Brecher, “A survey on ai-driven digital twins in industry 4.0: Smart manufacturing and advanced robotics,” *Sensors*, vol. 21, no. 19, p. 6340, 2021.
- [139] M. M. Rathore, S. A. Shah, D. Shukla, E. Bentafat, and S. Bakiras, “The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities,” *IEEE Access*, vol. 9, pp. 32 030–32 052, 2021.

-
- [140] C. Zhang, G. Zhou, J. Li, F. Chang, K. Ding, and D. Ma, “A multi-access edge computing enabled framework for the construction of a knowledge-sharing intelligent machine tool swarm in industry 4.0,” *Journal of Manufacturing Systems*, vol. 66, pp. 56–70, 2023.
- [141] C. Zhang, G. Zhou, Q. Xu, Z. Wei, C. Han, and Z. Wang, “A digital twin defined autonomous milling process towards the online optimal control of milling deformation for thin-walled parts,” *The International Journal of Advanced Manufacturing Technology*, vol. 124, no. 7-8, pp. 2847–2861, 2023.
- [142] S. Honig and T. Oron-Gilad, “Understanding and resolving failures in human-robot interaction: Literature review and model development,” *Frontiers in Psychology*, vol. 9, 2018.
- [143] Stanford Artificial Intelligence Laboratory et al., “Robotic operating system.” [Online]. Available: <https://www.ros.org>
- [144] “ISO/TS 15066:2016 Robots and robotic devices — Collaborative robots,” International Organization for Standardization, Geneva, CH, Standard, Feb. 2016.
- [145] “ISO 10218-1:2011 Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots,” International Organization for Standardization, Geneva, CH, Standard, July 2011.
- [146] C. Agüero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. Rivero, J. Manzo, E. Krotkov, and G. Pratt, “Inside the virtual robotics challenge: Simulating real-time robotic disaster response,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 494–506, April 2015.
- [147] E. Rohmer, S. P. N. Singh, and M. Freese, “Coppeliassim (formerly v-rep): a versatile and scalable robot simulation framework,” in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2013, www.coppeliarobotics.com.

-
- [148] W. Qiu and A. Yuille, “Unrealcv: Connecting computer vision to unreal engine,” in *Proceedings of European Conference on Computer Vision*. Springer, 2016, pp. 909–916.
- [149] A. Neubeck and L. Van Gool, “Efficient non-maximum suppression,” in *Proceedings of 18th International Conference on Pattern Recognition (ICPR’06)*, vol. 3. IEEE, 2006, pp. 850–855.
- [150] “ISO 20218-2:2017 Robotics — Safety design for industrial robot systems — Part 2: Manual load/unload stations,” International Organization for Standardization, Geneva, CH, Standard, dec 2017.
- [151] “IEC/TS 61496-4-3:2015 Safety of machinery - Electro-sensitive protective equipment - Part 4-3: Particular requirements for equipment using vision based protective devices (VBPD) - Additional requirements when using stereo vision techniques (VBPDST),” The British Standards Institution, London, UK, Standard, May 2015.
- [152] “AI data platform for automated annotation.” [Online]. Available: <https://www.v7labs.com/>
- [153] “The leading training data platform for data labeling.” [Online]. Available: <https://labelbox.com/>
- [154] K. Wada, “Labelme: Image Polygonal Annotation with Python.” [Online]. Available: <https://github.com/wkentaro/labelme>
- [155] R. Padilla, S. L. Netto, and E. A. B. da Silva, “A survey on performance metrics for object-detection algorithms,” in *Proceedings of the International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242.
- [156] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, “A comparative analysis of object detection metrics with a companion open-source toolkit,” *Electronics*, vol. 10, no. 3, 2021.

-
- [157] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, “A survey of uncertainty in deep neural networks,” *arXiv preprint arXiv:2107.03342*, 2021.
- [158] T. Han and Y.-F. Li, “Out-of-distribution detection-assisted trustworthy machinery fault diagnosis approach with uncertainty-aware deep ensembles,” *Reliability Engineering & System Safety*, vol. 226, p. 108648, 2022.
- [159] J. Xiao, W. Guo, J. Liu, and M. Li, “Generalization gap in data augmentation: Insights from illumination,” *arXiv preprint arXiv:2404.07514*, 2024.
- [160] L. Murmann, M. Gharbi, M. Aittala, and F. Durand, “A dataset of multi-illumination images in the wild,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4080–4089.
- [161] H. Liang, J. Gao, and N. Qiang, “A novel framework based on wavelet transform and principal component for face recognition under varying illumination,” *Applied Intelligence*, vol. 51, pp. 1762–1783, 2021.
- [162] H. Liu and L. Wang, “Collision-free human-robot collaboration based on context awareness,” *Robotics and Computer-Integrated Manufacturing*, vol. 67, p. 101997, 2021.
- [163] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and real-time tracking,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.
- [164] M. Kaspar, J. D. M. Osorio, and J. Bock, “Sim2real transfer for reinforcement learning without dynamics randomization,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4383–4388.
- [165] C. Li, P. Zheng, S. Li, Y. Pang, and C. K. Lee, “Ar-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop,” *Robotics and*

-
- Computer-Integrated Manufacturing*, vol. 76, p. 102321, 2022.
- [166] A. Sharma, E. Kosasih, J. Zhang, A. Brintrup, and A. Calinescu, “Digital twins: State of the art theory and practice, challenges, and open research questions,” *Journal of Industrial Information Integration*, vol. 30, p. 100383, 2022.
- [167] Y. Wang, X. Kang, and Z. Chen, “A survey of digital twin techniques in smart manufacturing and management of energy applications,” *Green Energy and Intelligent Transportation*, vol. 1, no. 2, p. 100014, 2022.
- [168] L. Hu, N.-T. Nguyen, W. Tao, M. C. Leu, X. F. Liu, M. R. Shahriar, and S. M. N. Al Sunny, “Modeling of cloud-based digital twins for smart manufacturing with mt connect,” *Procedia Manufacturing*, vol. 26, pp. 1193–1203, 2018, 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA.
- [169] Y. Liu, H. Xu, D. Liu, and L. Wang, “A digital twin-based sim-to-real transfer for deep reinforcement learning-enabled industrial robot grasping,” *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102365, 2022.
- [170] S. Li, N. Figueroa, A. Shah, and J. Shah, “Provably safe and efficient motion planning with uncertain human dynamics,” in *Proceedings of Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021.
- [171] Z. Erickson, H. M. Clever, V. Gangaram, G. Turk, C. K. Liu, and C. C. Kemp, “Multidimensional capacitive sensing for robot-assisted dressing and bathing,” in *Proceedings of 2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2019, pp. 224–231.
- [172] A. Clegg, Z. Erickson, P. Grady, G. Turk, C. C. Kemp, and C. K. Liu, “Learning to collaborate from simulation for robot-assisted dressing,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2746–2753, 2020.
- [173] F. Zhang, A. Cully, and Y. Demiris, “Probabilistic real-time user posture tracking for personalized robot-assisted dressing,” *IEEE Transactions on Robotics*, vol. 35,

- no. 4, pp. 873–888, 2019.
- [174] G. Chance, A. Jevtić, P. Caleb-Solly, G. Alenyà, C. Torras, and S. Dogramadzi, ““Elbows Out”—predictive tracking of partially occluded pose for robot-assisted dressing,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3598–3605, 2018.
- [175] T. D. Nguyen and M. Kresovic, “A survey of top-down approaches for human pose estimation,” *arXiv:2202.02656*, 2022.
- [176] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, “Deep high-resolution representation learning for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.
- [177] P. S. Maybeck, “The Kalman filter: An introduction to concepts,” in *Autonomous Robot Vehicles*. Springer, 1990, pp. 194–204.
- [178] H. Wang, H. Qi, M. Xu, Y. Tang, J. Yao, X. Yan, and M. Li, “Research on the relationship between classic Denavit-Hartenberg and modified denavit-hartenberg,” in *Proceedings of 7th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, 2014, pp. 26–29.
- [179] D. Roetenberg, H. Luinge, and P. Slycke, “Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors,” *Xsens Motion Technologies BV, Tech. Rep.*, vol. 1, pp. 1–7, 2009.
- [180] H. P. Gavin, “The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems,” *Department of Civil and Environmental Engineering, Duke University*, vol. 19, 2019.
- [181] H. Hefter, W. H. Jost, A. Reissig, B. Zakine, A. M. Bakheit, and J. Wissel, “Classification of posture in poststroke upper limb spasticity: A potential decision tool

- for botulinum toxin a treatment?” *International Journal of Rehabilitation Research*, vol. 35, no. 3, pp. 227–233, 2012.
- [182] C. B. Ivanhoe and T. A. Reistetter, “Spasticity: The misunderstood part of the upper motor neuron syndrome,” *American Journal of Physical Medicine & Rehabilitation*, vol. 83, no. 10, pp. S3–S9, 2004.
- [183] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.
- [184] D. A. Winter, *Biomechanics and motor control of human movement*. New Jersey, USA: John Wiley & Sons, 2009.
- [185] V. Sze, Y. Chen, T. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.
- [186] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artificial Intelligence Review*, pp. 1 – 62, 2020.
- [187] Z. Li, W. Yang, S. Peng, and F. Liu, “A survey of convolutional neural networks: Analysis, applications, and prospects,” *ArXiv*, vol. abs/2004.02806, 2020.
- [188] D. Kim, S.-H. Kim, T. Kim, B. B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kwon, H. Lee, J. Bae, Y.-L. Park, K.-J. Cho, and S. Jo, “Review of machine learning methods in soft robotics,” *PLOS ONE*, vol. 16, no. 2, pp. 1–24, 02 2021.
- [189] C. Duan, S. Junginger, J. Huang, K. Jin, and K. Thuro, “Deep Learning for Visual SLAM in Transportation Robotics: A review,” *Transportation Safety and Environment*, vol. 1, no. 3, pp. 177–184, 01 2020.

-
- [190] S. M. Grigorescu, B. Trasnea, T. T. Cocias, and G. Macesanu, “A survey of deep learning techniques for autonomous driving,” *J. Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.
- [191] Z. Kang, C. Catal, and B. Tekinerdogan, “Machine learning applications in production lines: A systematic literature review,” *Computers & Industrial Engineering*, vol. 149, p. 106773, 2020.
- [192] H. Wang and D.-Y. Yeung, “A survey on Bayesian deep learning,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–37, 2020.
- [193] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, “A review of uncertainty quantification in deep learning: Techniques, applications and challenges,” *Information Fusion*, vol. 76, pp. 243–297, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1566253521001081>
- [194] D. Dera, N. Bouaynaya, G. Rasool, R. Shterenberg, and H. Fathallah-Shaykh, “PremiUm-CNN: Propagating uncertainty towards robust convolutional neural networks,” *IEEE Transactions on Signal Processing*, pp. 1–1, 2021.
- [195] R. W. Hamming, “Error detecting and error correcting codes,” *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [196] G. Carannante, D. Dera, G. Rasool, N. C. Bouaynaya, and L. Mihaylova, “Robust learning via ensemble density propagation in deep neural networks,” in *Proceedings of the IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2020, pp. 1–6.
- [197] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill, and R. Ashmore, “Structural test coverage criteria for deep neural networks,” in *Proceedings from the*

-
- IEEE/ACM 41st International Conf. on Software Engineering: Companion Proceedings*, 2019, pp. 320–321.
- [198] D. Dera, G. Rasool, and N. Bouaynaya, “Extended variational inference for propagating uncertainty in convolutional neural networks,” in *Proceedings of the IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2019, pp. 1–6.
- [199] P. Wang, N. C. Bouaynaya, L. Mihaylova, J. Wang, Q. Zhang, and R. He, “Bayesian neural networks uncertainty quantification with cubature rules,” in *Proceedings from the 2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.
- [200] L. Swiniarski, J. Bruna, and K. Cho, “Study of activation patterns,” 2018. [Online]. Available: <https://github.com/lucas-swiniarski/Activation-Patterns>
- [201] R. Bellman, “Dynamic programming,” *Science*, vol. 153, no. 3731, pp. 34–37, 1966.
- [202] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.
- [203] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, “A survey of large language models,” *arXiv preprint arXiv:2303.18223*, 2023.
- [204] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, “Do as I can, not as I say: Grounding language in robotic affordances,” *arXiv preprint arXiv:2204.01691*, 2022.
- [205] M. Mazumder, C. Banbury, X. Yao, B. Karlaš, W. Gaviria Rojas, S. Diamos, G. Diamos, L. He, A. Parrish, H. R. Kirk *et al.*, “Dataperf: Benchmarks for data-

centric ai development,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

- [206] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.