NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

GONÇALO VELHO CABRAL PACHECO FURTADO

BSc in Electrical and Computer Engineering Sciences

HUMAN AWARE COOPERATION IN INDUSTRIAL MANIPULATORS

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
March, 2023

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

HUMAN AWARE COOPERATION IN INDUSTRIAL MANIPULATORS

GONÇALO VELHO CABRAL PACHECO FURTADO

BSc in Electrical and Computer Engineering Sciences

Adviser: José Antonio Barata de Oliveira
Full Professor, FCT-NOVA

Co-adviser: Francisco Antero Cardoso Marques
Associate Professor, FCT-NOVA

Examination Committee

Chair: Filipe de Carvalho Moutinho
Full Professor, FCT-NOVA

Rapporteur: Ricardo Alexandre Fernandes da Silva Peres
Full Professor, FCT-NOVA

Member: José Antonio Barata de Oliveira
Full Professor, FCT-NOVA

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
March, 2023

Human Aware Cooperation in Industrial Manipulators

Copyright © Gonçalo Velho Cabral Pacheco Furtado, NOVA School of Science and Technology, NOVA University Lisbon.
The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.10.10) [47].

Acknowledgements

First, I would like to thank my parents, Ana Rosa and Paulo José, for making this journey possible and believing in me to see it to its accomplishment, but most importantly for their patience during this long and lonely phase of mine, for which I am forever in debt to them.

Also, I want to thank my co-advisor, Francisco Marques, for his help in quickstarting the dissertation with insightful feedback and connecting me with valuable resources, such as the cooperative robots from Universal Robots, KUKA, and Doosan.

I would also like to thank my friends André, David, Manuel, and João Cabacinho for providing a refuge in stressful times, as well as my colleagues for their company throughout my academic journey. I want to give special thanks to Sr. Manuel for his company throughout this dissertation and his helping hand in assembling the setup used.

"You cannot teach a man anything; you can only help him discover it in himself." (Galileo)

Abstract

Industrial manipulators have grown to become the most adopted technology in the industry to help mitigate manual labor and shift the focus to the more core problems of the assembly line, but their adoption has created a physical and psychological barrier between them and the human workers. Therefore, this dissertation proposes a cognitive system that extends the capabilities of traditional autonomous systems by incorporating human-like abilities such as action recognition and cooperation.

To achieve this, a hierarchical architecture is adopted, consisting of three phases: Awareness, Cooperation, and Safety. The former extracts knowledge from visual data captured from the physical world, while the Cooperation phase utilizes this knowledge to make logical decisions regarding suitable cooperative goals. Finally, the Security phase plans collision-free trajectories for the manipulator to execute and achieve the desired goals. Each phase consists of multiple modules, each addressing specific functionality requirements.

This approach was implemented in the ROS environment where each module aggregates a set of nodes, some built to comply with the specific requirements others extracted from already advanced works. Its feasibility was tested in an experimental setup simulating an industrial environment which comprised of a UR3e cobot from Universal Robots mounted with a two-finger 140mm gripper from Robotiq and an Intel RealSense D435i camera.

The results demonstrated the effectiveness of the proposed architecture in coordinating cooperative responses, achieving collision-free trajectories, and promoting a cooperative environment between humans and industrial manipulators.

This research contributes to the field of human-robot cooperation and provides practical insights for engineers and industry professionals. By leveraging the potential of the hierarchical architecture and implementing the recommendations outlined in this research, industries can play a crucial role in achieving a more efficient and cooperative future between humans and machines.

Keywords: Hierarchical Architecture, Awareness, Cooperation, Safety, Manipulation

Resumo

Os manipuladores industriais tornaram-se amplamente adotados na indústria para reduzir a mão de obra manual e enfrentar desafios na linha de montagem. No entanto, a sua adoção tem levado a barreiras físicas e psicológicas entre os manipuladores e os trabalhadores humanos. Para superar essas barreiras, esta dissertação propõe um sistema que aprimora os sistemas autónomos ao incorporar conhecimentos semelhantes aos humanos, como o reconhecimento de ações e a cooperação.

Para tal, adota-se uma arquitetura hierárquica composta por três fases: Consciência, Cooperação e Segurança. A fase de Consciência extrai conhecimento de dados visuais capturados no mundo físico, enquanto a fase de Cooperação utiliza esse conhecimento para tomar decisões lógicas em relação a metas cooperativas. A fase de Segurança planeia trajetórias livres de colisão para o manipulador alcançar essas metas. Cada fase inclui vários módulos que abordam requisitos específicos de funcionalidade.

A abordagem proposta é implementada no ambiente ROS, sendo que cada módulo é composto por um conjunto de nós projetados para atender a requisitos específicos. Um ambiente experimental que simula uma situação industrial é utilizado para testes, envolvendo um cobot UR3e da Universal Robots equipado com uma garra de dois dedos de 140mm da Robotiq e uma câmera Intel RealSense D435i.

Os resultados demonstram a eficácia da arquitetura hierárquica na coordenação de respostas cooperativas, na obtenção de trajetórias livres de colisão e na promoção de um ambiente cooperativo entre trabalhadores humanos e manipuladores industriais.

Esta pesquisa contribui para o campo da cooperação humano-robô e fornece entendimentos práticos para engenheiros e profissionais da indústria. Ao aproveitar o potencial da arquitetura hierárquica e implementar as recomendações delineadas nesta pesquisa, as indústrias podem desempenhar um papel crucial na conquista de um futuro mais eficiente e cooperativo entre humanos e máquinas.

Palavras-chave: Arquitetura Hierárquica, Consciência, Cooperação, Segurança, Manipulação

Contents

List of Figures ix

List of Tables xii

1 Introduction 1
- 1.1 Problem . . . 1
- 1.2 Proposed Solution . . . 3
- 1.3 Outline of Dissertation . . . 4

2 State of the Art Review 5
- 2.1 Human Action Recognition and Prediction . . . 5
- 2.2 Action Recognition . . . 6
  - 2.2.1 Shallow Approaches . . . 7
  - 2.2.2 Deep Learning Methods . . . 18
- 2.3 Action Prediction . . . 23
  - 2.3.1 Partial Videos Prediction or Early Recognition . . . 23
  - 2.3.2 Motion Prediction . . . 24
  - 2.3.3 Path Prediction . . . 25

3 Proposed Approach 28
- 3.1 Architecture Overview - A Theoretical Approach . . . 28
- 3.2 Architecture Design - An In-Depth View . . . 29
  - 3.2.1 Safety . . . 30
  - 3.2.2 Awareness . . . 32
  - 3.2.3 Cooperation . . . 39
- 3.3 Experimental Setup - An Experimental Approach . . . 41
  - 3.3.1 Hardware . . . 41
  - 3.3.2 Software . . . 43
  - 3.3.3 Experimental Environment . . . 44
  - 3.3.4 Experimental Procedure . . . 46

3.4 Ethical Considerations 46
3.5 Limitations 47

4 Experimental Results **49**
4.1 Awareness 49
4.1.1 Action Recognition Module 49
4.1.2 Human Detection Module 53
4.1.3 Object Detection Module 55
4.2 Cooperation and Safety 57

5 Discussion **63**
5.1 Action Recognition Module 63
5.2 Human Detection Module 64
5.3 Object Detection Module 65
5.4 Cognitive Layer and Robot Manipulation Module 66

6 Conclusion **68**

Bibliography **70**

List of Figures

1.1 Illustration of the hierarchy levels in HRC taken from [81] 2

2.1 Comparison of MEI and MHI taken from [6]: (left) keyframe; (middle) MEI; (right) MHI 7
2.2 Visualization of MHV taken from [88]: (a) actor performing an action; (b) MHV 8
2.3 Human detection results from [91] 9
2.4 Output of the method proposed by [9] 10
2.5 Results from the spatio-temporal interest points obtained from the motion of the legs of a walking person from the proposed method in [43]. (Left) 3D plot of a leg pattern and the detected interest points. (Right) The detected interest points overlaid on single frames 11
2.6 Computation of dense trajectories taken from [86]. (a) Illustration of the proposed approach. (b) Visualization of dense trajectories 12
2.7 Scatter plot featuring a linear support vector's machine decision boundary from [90] 13
2.8 Example of k-NN classification taken from [89] 14
2.9 Framework of the model presented in [55] 15
2.10 Output of the proposed method from [94] 16
2.11 Depth images from a Kinect device taken from the MSR-Action3D dataset . 17
2.12 Example of a ResNet architecture taken from [21]. (a) Two-stream architecture (appearance + motion) with motion gating. (b) Illustration of a bidirectional gating accross the streams 19
2.13 3D CNN taken from [36] 20
2.14 The 3D CNN from [5] 21
2.15 Architecture overview from [5] 21
2.16 Comparison between activity classification and activity prediction taken from [62] 24
2.17 Picture of bench-press action taken from [84]. On the left is represented the projection of the trajectories onto the image plane and on the right the trajectories are illustrated in 3D space-time 25

2.18 Illustration of the proposed method by [4]. (a) Illustration of the computed trajectories (b) Overview of the proposed Social-LSTM which incorporates a pooling strategy 26

3.1 Hierarchical Architecture to be implemented 29
3.2 Hierarchical Architecture functionality 30
3.3 Visual representation of the functionality of the Robot Manipulation Module 31
3.4 Pick and Place function decision process 32
3.5 Visual representation of the functionality of the Action Recognition Module 33
3.6 Visual representation of the training stage 35
3.7 Seven frame examples, one from each video performing the cooperative action *asking for something* 36
3.8 Seven frame examples, one from each video performing the cooperative action *giving something* 36
3.9 Visual representation of the testing stage 37
3.10 Visual representation of the functionality of the Human Detection Module . 38
3.11 Visual representation of the functionality of the Object Detection Module . . 39
3.12 Overview of the decision process of the abstract cooperative action when inputted with the perception information 40
3.13 Visual representation of the functionality of the Cognitive Layer 41
3.14 Universal Robots cobots models' lineup from [73] 41
3.15 Robotiq two finger 140 mm gripper from [58] 42
3.16 Intel RealSense Depth Camera D435i [34] 42
3.17 Visualization of the setup used 45

4.1 Successful recognition of the ***asking for something*** action 50
4.2 Successful recognition of the ***giving something*** action 51
4.3 Failed recognition of the ***giving something*** action 52
4.4 Human 2D skeleton being detected by the OpenPose tool [42] in a RGB color scheme where the joints are seen as dots and the limbs as the connections between those dots 53
4.5 Visualization of the tool's [37] performance on the ROS RViz program on the right side as the human subject's 3D skeleton pose in green. On the left side is depicted a human subject being detected by the OpenPose tool [42] 54
4.6 Visualization of the module's performance on the ROS RViz program on the right side as the human subject's 3D skeleton pose represented by green spheres. On the left side is depicted a human subject being detected by the OpenPose tool [42]. 4.6a A general view of the module's functionality; 4.6b The human subject hand approaching the manipulator; 4.6c The human hand touching the manipulator and the collision being detected in the shoulder link with the color red 55

4.7 Visualization of the tool's [29] performance on the ROS RViz program. On the left top corner can be visualized the RGB segmentation of the objects detected in the environment. Below, on the left bottom corner is visualized their respective computed grasp approaches in green. On the right side is shown the computed objects 3D pose alongside the environment depth data 56
4.8 Visualization of the performance of the 4.8a Object Detection Module and the 4.8b Human Detection Module 57
4.9 Manipulator performing a typical pick and place operation while trying to avoid contact with the human subject. 4.9a The human subject is at the left of the manipulator while it performs a pick operation with a collision free trajectory. 4.9b The mentioned pick operation is being executed and visualized through the simulator. The human's 3D pose is shown as the green spheres 58
4.10 Manipulator performing a typical pick and place operation while trying to avoid contact with the human subject. 4.10a The human is positioned more to the left of the manipulator while it is seen halted for some time. 4.10b This action is displayed in the simulator where the human pose is incorrectly inputted 59
4.11 Manipulator performing a typical pick and place operation while trying to avoid contact with the human subject. 4.11a The manipulator is performing a trajectory to the place position that is in close collision with the human. 4.11b The simulator shows the planned trajectory as translucid purple and the humans pose 59
4.12 Manipulator performing a typical pick and place operation while trying to avoid contact with the human subject. 4.12a The human positions himself at the right of the manipulator while it finishes the place operation. 4.12b This action is shown in the simulator 60
4.13 Manipulator performing the ***Give to Human*** operation. 4.13a The manipulator is placing an object in the middle of the human subject hands (Mr. Manuel). 4.13b The simulator shows the human subjects pose as green spheres, the planned trajectory as translucid purple and the place pose between the human's hands 61
4.14 Manipulator performing the ***Take from Human*** operation. 4.14a The manipulator halts at the pick pose while the human places the object between the gripper while it grasps it. 4.14b The Object Detection Module shows the computation of the objects' pose. It also shows other undesired objects captured. 4.14c The simulator shows the humans pose as green spheres, the manipulator at the pick pose and the planned trajectory as translucid purple 62

List of Tables

2.1 A quick summary of the classification and the evolution of the state-of-the-art. The symbol '✗' applies to references that only focus on action representation performance. 18
2.2 Listing of all state-of-the-art for Action Recognition and comparison of their accuracy in the main datasets (Self-P means the datasets that are self-produced and not the main ones) 18
2.3 Listing of all state-of-the-art for Action Recognition and comparison in Deep Learning Methods 22
2.4 Listing of all state-of-the-art for Action Recognition and comparison of their accuracy in the main datasets in Deep Learning Methods 22
2.5 Listing of all state-of-the-art for Action Prediction and comparison 27

3.1 Description of the different datasets each model for action recognition available on the MMAction2 was pre-trained on 34
3.2 SlowFast model configuration 37
3.3 Summary of the system setup in terms of the software used, the hardware, the frameworks, the tools, and the drivers 44

4.1 Aggregation of the metrics obtained from the experiment. On the middle column it is shown the maximum guess probability of the top 1 label. On the last column it is presented the top 2 probability variation 50
4.2 Aggregation of the metrics obtained from the experiment. On the middle column it is shown the maximum guess probability of the top 1 label. On the last column it is presented the top 2 probability variation 51
4.3 Aggregation of the metrics obtained from the experiment. On the middle column it is shown the maximum guess probability of the top 1 label. On the last column it is presented the top 2 probability variation 52

1 Introduction

Human and machine, friends, or foes? This question arises as the consumer world continues to expand, presenting significant challenges for companies of all sizes in meeting high-volume and time-sensitive product demands. Manual labor alone is insufficient and can lead to high fatigue levels among human workers, making it unsustainable and inefficient. In response to this challenge, industrial manipulators were introduced as automatic handling systems, enabling human workers to focus on core problems within the assembly line.

1.1 Problem

However, industrial manipulators are typically pre-programmed to perform specific tasks, which limits their maneuverability in dynamic industrial environments. This constrained nature assumes an unchanging physical environment and has resulted in the creation of manipulator-specific workspaces, where human intervention occurs only when the manipulator is offline due to the risk of collision, which can have severe consequences, including fatalities. Additionally, this division between human workers and industrial manipulators is not only costly and time-consuming to address but can also shrink the human workspace and create a psychological barrier.

Therefore, the objective of this research is to develop a cognitive system that extends the capabilities of common autonomous systems, enabling human-like actions, such as action recognition and cooperation, for industrial manipulators. The aim is to promote a healthier and more efficient work ethic by fostering a symbiotic relationship between humans and machines.

The concept of Human-Robot Interaction (HRI) has emerged as a means to bridge the physical and psychological gaps between human workers and industrial manipulators. [32] classifies this concept into three categories: Huma-Robot Coexistence (HRCx), also known as coaction, refers to the capability of humans and robots sharing the same dynamic workspace without engaging in a common task or requiring coordination of actions and intentions; Human-Robot Cooperation (HRCp) operates at a higher level than HRCx. In

this category, humans and robots work together towards a shared goal, simultaneously meeting time and space requirements; Human-Robot Collaboration (HRC) represents the ability to perform complex tasks through direct human interaction in two modalities. Physical collaboration involves explicit and intentional contact with force exchange between humans and robots. Contactless collaboration, on the other hand, does not involve physical interaction. Instead, actions are coordinated through information exchange via direct communication (speech, gestures) or indirect communication (intention recognition, eye gaze direction, facial expressions).

In [81], a further division is proposed for HRC, as can be visualized in Figure 1.1. As illustrated, a distinction between safety, coexistence, and collaboration in human-robot interactions has been highlighted. This distinction serves as a nested framework for HRC encompassing a range of interactions, from sharing only the physical workspace without sharing the task to fully sharing tasks with cognitive engagement. In all cases, ensuring safety is of utmost importance and must be inherently guaranteed.

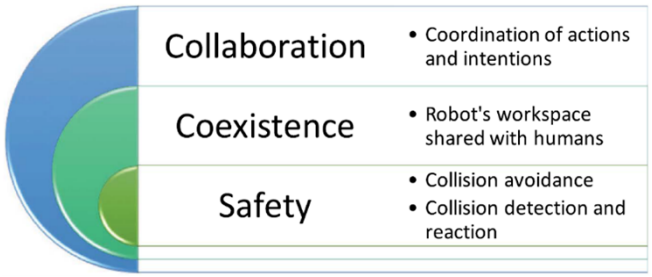


Figure 1.1: Illustration of the hierarchy levels in HRC taken from [81]

The higher levels of engagement require that the features of lower levels are ensured, as depicted in Figure 1.1. Safety features include internal and external mechanical, sensory, and control mechanisms to prevent collisions. In the event of accidental collisions, the robot should be capable of reacting by reducing forces upon impact, either through appropriate control laws or by employing lightweight robots with compliant joints.

Moving beyond safety, the framework introduces the concept of coexistence, where a robot and a human operator can safely share the workspace and potentially work on the same object without direct contact or coordination of actions and intentions. To attain this goal, the HRC framework needs some sort of awareness mechanism that outputs human information in 3D space. Depending on the goal, several methods exist for human detection. Some focus more on the motion of the human [6], [88], [86], [85], others on small features from the visual input [22], [43], [65], [17], [38], and others fully extract the human form [75], [91], [10].

Collaboration represents the next step, enabling the robot and the human operator to jointly perform complex tasks through direct interaction and coordination. This can be achieved through intentional physical contact with force exchanges or without contact, using gestures or voice commands. After human detection, the problem lies in how to use this information to classify the activities being performed. Direct classifiers, like the SVM

(Support Vector Machine) [44] or the k-NN (k-Nearest Neighbor) [75], infer directly from the extracted features due to their previously built form. Sequential approaches learn from the temporal evolution of features, which means that the action being performed will be classified as the most probable based on the learned features [20], [55]. Mid-Level Features approaches aim at learning an extra layer of representation to better abstract the low-level features [94]. On the other hand, deep learning methods have been shown to understand human actions quite accurately from the direct input data using several architectures of artificial neural networks such as CNNs (Convolutional Neural Networks) [36], [68], [69], [77], [79], [11], RNNs (Recurrent Neural Networks) [45], [70], ResNets (Residual Neural Networks) [54], [21], [76], and LRCN which, in its simple form, is the fusion of CNNs with RNNs [5], [19], [66], [49]. This is incredibly useful for coordination between actions. However, these approaches are only useful for the problem at hand if the robot can detect the activities before they are finished, i.e., at an early stage. The intelligent system can take stock of what has happened so far and predict the most likely future, which leads it to calculate not just a safe but also an efficient path trajectory, even if the human hasn't completed the task so far.

1.2 Proposed Solution

To achieve effective cooperation and collaboration, an awareness mechanism is essential. This research adopts a hierarchical architecture approach, comprising three phases: Awareness, Cooperation, and Safety. The Awareness phase extracts knowledge from visual data captured from the physical world, while the Cooperation phase utilizes this knowledge to make logical decisions regarding suitable cooperative goals. Finally, the Safety phase plans collision-free trajectories for the manipulator to execute and achieve the desired goals. Each phase consists of multiple modules, each addressing specific functionality requirements.

By developing this optimized architecture that adapts to varying cooperative human intentions, this research aims to mitigate the barrier between humans and machines in the industrial workspace, reduce construction costs, and enhance work efficiency between these agents. The modular design of the proposed framework offers a flexible and scalable solution, allowing for the replacement or addition of layers and modules while preserving the core characteristics.

The significance of this research lies in its potential to serve as a foundational framework for addressing practical Human-Robot Interaction challenges, encompassing the objectives of safety, coexistence, and collaboration. By addressing the challenges associated with cooperation between humans and industrial manipulators, this research aims to pave the way for a more cooperative and adaptive future, where industrial manipulators play a central role in meeting the growing custom demands of consumers.

1.3 Outline of Dissertation

The upcoming chapters will present a comprehensive review of state-of-the-art methodologies addressing action recognition and action prediction challenges using RGB and Depth images. A comparative analysis will be conducted, evaluating their performance on prominent datasets. Subsequently, the experimental chapters will be introduced, starting with an in-depth explanation of the proposed approach. The implementation of the methodology will be described, followed by the presentation of the obtained results. Additionally, a dedicated experimental chapter will discuss the outcomes related to the key components of the architecture. Finally, the last chapter will provide a conclusive summary of this dissertation, consolidating the findings and contributions made throughout the study.

2

State of the Art Review

In recent years, the field of robotics has witnessed significant growth, with various applications emerging in industrial environments. Particularly, robots, including manipulators, are becoming more intelligent and are transitioning toward social environments. In some workplaces, human-robot cooperation is being incorporated, necessitating the development of certain human-like features in manipulators to ensure seamless interaction with humans. These features, inherent to humans due to their highly sociable nature, pose significant challenges for robotic systems but are crucial for facilitating effective collaboration. They involve the recognition of activities occurring in shared environments and the ability to predict the trajectory of these activities. By integrating such capabilities, workspaces can be optimized for efficiency and safety. This chapter aims to review and discuss current approaches to computer vision-based action recognition and prediction methods in the context of human-robot cooperation.

2.1 Human Action Recognition and Prediction

In 1999, Aggarwal and Cai [2] conducted a comprehensive review of methods and techniques for human motion analysis, which had garnered increasing attention from computer vision experts at the time. Their work laid the foundation for understanding the different aspects of human motion analysis.

Building upon this foundation, Aggarwal and Ryoo [3] published a more extensive survey in 2011, categorizing analysis methods into distinct groups based on their underlying techniques. These categories included low-level feature-based approaches, mid-level representation-based approaches, and high-level semantic-based approaches. This categorization provided a holistic view of the field and contributed to the development of diverse methodologies.

In 2015, Vrigkas et al. [83] reviewed existing methods and techniques specifically focused on human activity recognition. Their review categorized these methods into rule-based, template-based, sensor fusion-based, and machine learning-based approaches. By examining the strengths and limitations of each category, they offered valuable insights

into the advancements in human activity recognition.

Recently, Kong and Fu [39] conducted a comprehensive survey in 2022, presenting an in-depth analysis of human action recognition and prediction methods. Their survey encompasses a broad spectrum of approaches, rendering redundant any further attempts to cover the same ground. As a result, the taxonomies presented in their survey serve as a foundation for structuring this chapter on literature review, ensuring a comprehensive and organized examination of the topic at hand.

2.2 Action Recognition

Action recognition refers to the process of inferring human actions based on complete action executions captured in a video. It is a fundamental task within the computer vision community, aiming to recognize and understand human actions from visual data.

In action recognition, a video serves as the input, containing a sequence of frames representing the complete execution of an action. From the perspective of a machine, these frames are merely arrays of pixels, lacking the inherent understanding of how to interpret and extract meaningful information from them. This leads to two main challenges: action representation and action classification.

Action representation involves converting the raw pixel data of an action video into a suitable representation that captures the salient features and characteristics of the action. This can be achieved by extracting visual features or encoding the video into a feature vector or a series of vectors that effectively represent the action's visual content.

Once an action has been appropriately represented, the action classification component comes into play. This component aims to infer the corresponding action label or category based on the extracted features. Machine learning algorithms, such as classifiers or deep neural networks, are commonly employed to make predictions about recognized actions.

Traditionally, these two components, action representation, and action classification, have been treated as separate stages in the action recognition pipeline. However, recent advancements in deep learning have allowed for the integration of these components into a unified, end-to-end trainable framework. Deep networks merge the tasks of action representation and action classification, enabling simultaneous feature extraction and action prediction. This integrated approach has shown promising results, leading to improved classification performance in action recognition tasks.

Next, a more detailed discussion about these two essential approaches to human action recognition will be provided while also supplying examples of the literature within each category. A summary can be found in Table 2.1.

2.2.1 Shallow Approaches

2.2.1.1 Action Representation

Action recognition poses a significant challenge in representing actions within videos due to variations in motion speed, camera view, appearance, and pose. Effective action representation methods need to be computationally efficient, capable of characterizing actions, and maximize the differences between actions to minimize classification errors. The recognition task is further complicated by appearance and poses variations within the same action category. Action representation approaches can be categorized into holistic and local features, with hand-crafted methods focusing on predefined parameters rather than automatically learned parameters in deep networks. Hand-crafted methods capture discriminative information through spatiotemporal interest points and descriptors, such as HOG, HOF, and MBH, while holistic representations summarize overall motion dynamics and appearance using BoW models and Fisher vectors. Although deep learning has shown superior performance, hand-crafted methods still provide interpretability and insights in action recognition tasks.

Holistic Representations Holistic representation methods in action recognition capture the motion information of the entire human subject, providing rich and expressive motion details. However, these representations are prone to sensitivity to noise and the inclusion of irrelevant information from the subject and cluttered backgrounds.

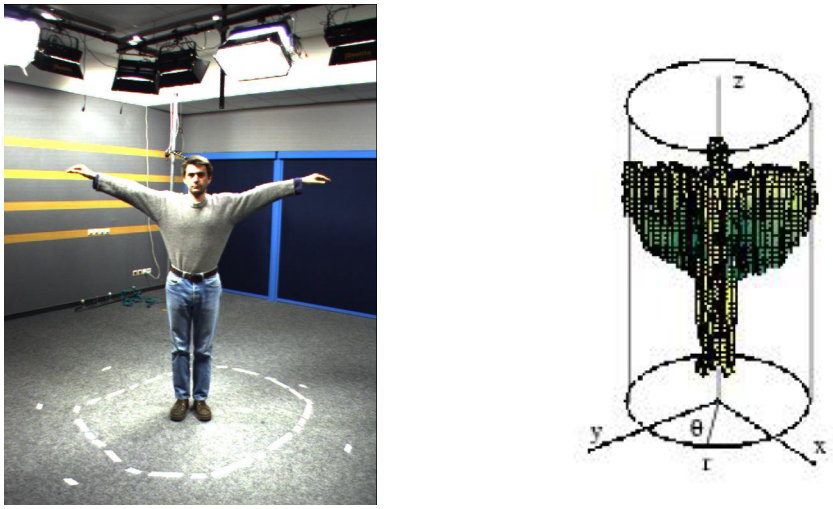
MEI and MHI: Pioneering work by [6] introduced Motion Energy Image (MEI) and Motion History Image (MHI) techniques, operating on silhouettes to encode dynamic human motion into a single image (see Figure 2.1). MEI highlights the spatial distribution of motion, indicating where the motion is occurring, while MHI depicts both the spatial distribution and the temporal characteristics of motion. Brighter values in the MHI correspond to more recent motion.



Figure 2.1: Comparison of MEI and MHI taken from [6]: (left) keyframe; (middle) MEI; (right) MHI

MHV: Although MEI and MHI demonstrated promising results, they are sensitive to changes in viewpoint. To address this limitation, [88] extended the concept to a 3D Motion History Volume (MHV) to remove viewpoint dependency in the final action

representation. MHV utilizes 3D voxels obtained from multiple camera views, showcasing the 3D occupancy in the resulting volume, represented in Figure 2.2. Fourier transform is then employed to create features that are invariant to location and rotation.



(a) (b)

Figure 2.2: Visualization of MHV taken from [88]: (a) actor performing an action; (b) MHV

Silhouette: Other researchers, like the work from [75] introduced metric learning techniques that aim at learning a distance metric or similarity measure between data samples. In the context of human activity recognition, metric learning can be used to define a similarity measure between different activity instances, enabling the system to compare and recognize activities based on their similarity. In this research, these activity instances are spatiotemporal interest points extracted from human silhouettes in video frames.

An interesting work by [91] represents human silhouettes in visual scenes using depth information captured by a Kinect sensor. The authors aim to leverage the depth information provided by the Kinect sensor to improve the accuracy and robustness of human detection compared to traditional RGB-based approaches. Depth information can provide additional cues about the spatial layout and geometry of the scene, which can be useful for distinguishing humans from the background and handling challenging scenarios such as occlusions. To extract human silhouettes the authors first start by preprocessing the depth information to remove noise and enhance relevant depth boundaries. Then, they introduce a depth-based segmentation algorithm to separate foreground objects, including humans, from the background. The depth-based segmentation is performed by exploiting the depth discontinuities and depth gradients present in the scene. Next, the research applies a set of features extracted from both the RGB and depth information to represent the detected regions. These features capture the appearance and geometric characteristics of the regions, represented in Figure 2.3. The authors employ a classifier to learn the discriminative patterns and distinguish humans from other objects based on these features.



(a) (b)

Figure 2.3: Human detection results from [91]

Shape: A different approach to action representation is done by [93] which focuses on developing a method for detecting human poses that can accurately handle the articulation and deformations of human body parts, a common struggle of traditional approaches due to the varying appearances and configurations of body parts. The authors present a detection framework called "Flexible Mixtures of Parts" that addresses the challenge of modeling human poses. This framework decomposes the human body into parts and independently models their articulation and deformations, instead of treating the whole body as a rigid structure. A graphical model representation is introduced to capture the relationships and spatial configurations between body parts, consisting of part detectors for individual body parts and a spatial model encoding their relationships and articulations. A structured learning algorithm is utilized to jointly infer the locations and configurations of body parts in an image. To handle appearance and pose variability, the research proposes a mixture model that combines multiple appearance templates per part, enabling the capture of various body part appearances and deformations. The mixture model is trained on a large dataset to learn appearance variations and associated probabilities.

The research articles by [10] and [9], respectively, focus on real-time multi-person 2D pose estimation using part affinity fields. The first research introduces a method that effectively estimates the poses of multiple individuals by modeling the pairwise relationships between body parts. It employs convolutional neural networks to extract features and predict part affinity fields, which capture the connectivity and spatial relationships between body parts. The method demonstrates its capability to handle occlusions and crowded scenarios, providing accurate and real-time pose estimation (see Figure 2.4). Building upon this work, the second research presents the OpenPose framework, which further enhances the initial method. OpenPose integrates part affinity fields into a unified framework for real-time multi-person 2D pose estimation. It introduces a deep neural network architecture that refines the estimated poses through post-processing steps, improving accuracy and ensuring correct connections between body parts. The framework is evaluated extensively, demonstrating its superior performance and efficiency compared to other state-of-the-art methods. Together, these research articles contribute significantly to the field of multi-person pose estimation. They address the challenges

of accurately estimating poses in real-time and crowded scenarios, providing effective solutions based on the concept of part affinity fields. The proposed methods offer practical applications in areas such as human-computer interaction, action recognition, and pose estimation. Overall, the research advances the field by presenting efficient and accurate approaches for real-time multi-person 2D pose estimation, facilitating various computer vision applications.

These silhouette-based and shape-based methods contribute to the diverse range of techniques available for action representation, capturing different aspects of human actions. While some prioritize accurate representation and pose estimation, others focus on the direct classification of atomic actions or offer multi-purpose applications such as object recognition.

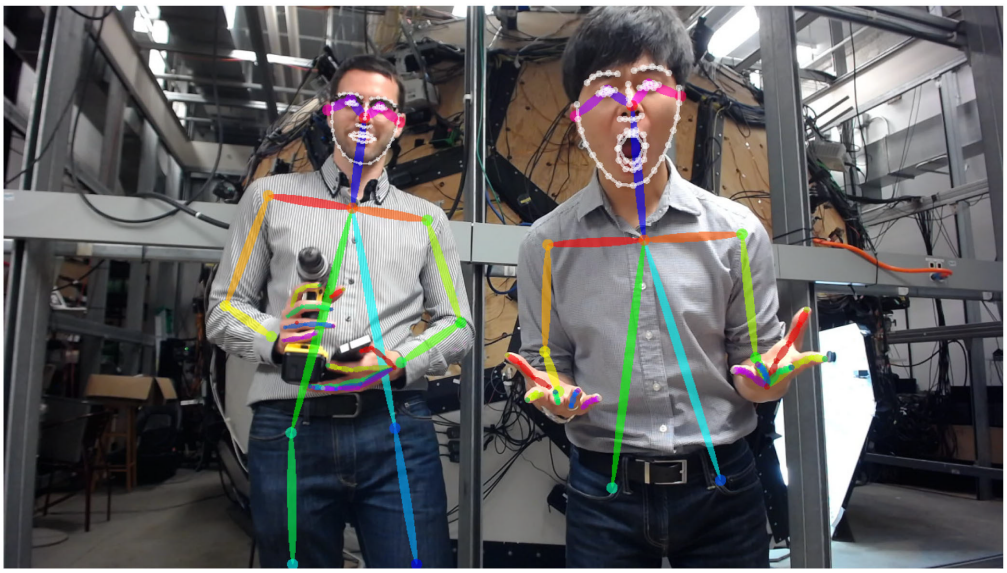


Figure 2.4: Output of the method proposed by [9]

Local Representations Local representations play a crucial role in action recognition by identifying specific local regions that contain salient motion information. Unlike holistic representations, which capture the motion of the entire human subject, local representations focus on localized areas of interest. This approach inherently addresses the challenges associated with holistic representations, such as the inclusion of irrelevant information and noise from the subject and background. By isolating the most relevant and informative regions, local representations provide a more focused and precise characterization of actions, enhancing the accuracy and robustness of action recognition systems.

Space-Time Interest Points: In 1987, Förstner et al. [22] introduced a fast interest point operator for automatic detection and precise localization of point-type features in digital images, such as distinct points, corners, and circular feature centers. This theoretical foundation laid the groundwork for subsequent work by Laptev [43], who utilized local structures to represent actions as spatiotemporal events (see Figure 2.5). The focus of

this work was on understanding human location and tracking relevant features. Building upon this, Schuldt et al. [65] extended the approach to video interpretation, using a walking model for human tracking. While this work primarily focused on action representation through local features, it served as a basis for the development of a visual pattern recognition tool, including an SVM classifier for the direct classification of atomic actions.

Another approach to action representation is the Histograms of Oriented Gradient (HOG) descriptors proposed by Dalal et al. [17]. This method focuses on detecting human silhouette contours using a trained SVM classifier for person/non-person classification, prioritizing the performance of the HOG detector rather than action classification.

Building upon the success of HOG-based descriptors, Kläser et al. [38] extended the key concepts of HOG to 3D representations, treating videos as spatiotemporal volumes. Their method, known as HOG3D, detected interest points based on a space-time extension of the Harris operator and constructed a spatiotemporal bag of words with a visual vocabulary. The parameters of this model were trained on the KTH dataset to achieve accurate action recognition through direct classification. The recognized actions in this work include human-to-object and human-to-human interactions.

These approaches demonstrate the use of local structures, HOG descriptors, and spatiotemporal representations for action representation and classification, each with a specific focus on feature detection, silhouette contours, or spatiotemporal features.

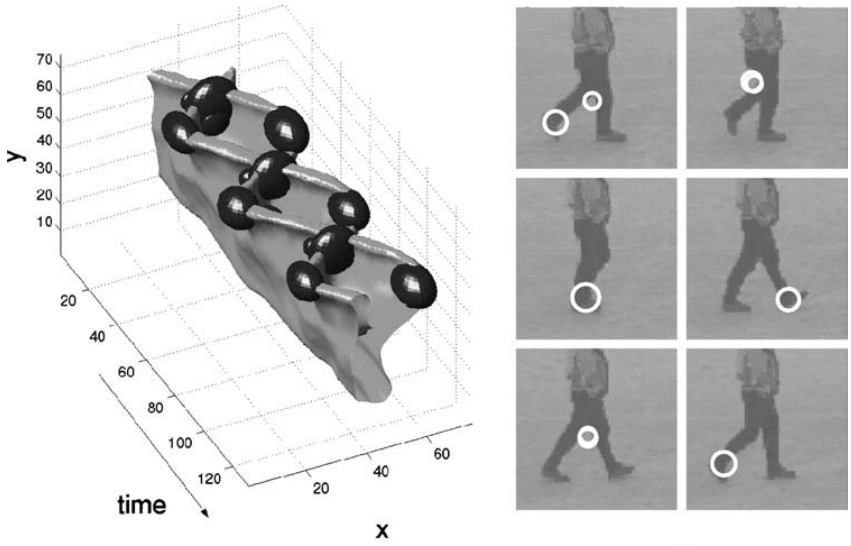
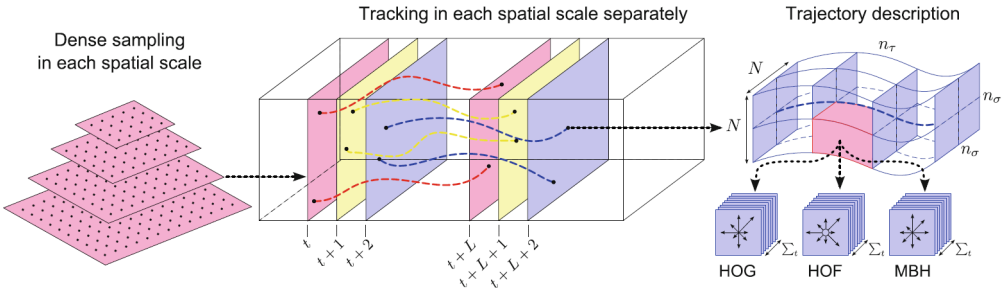


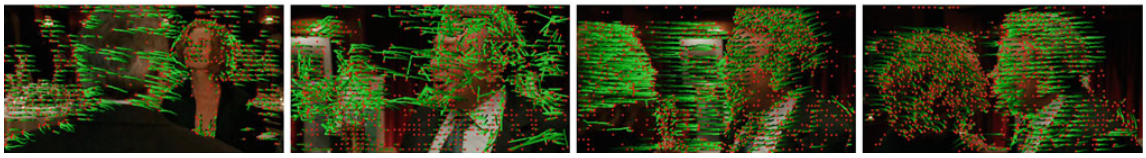
Figure 2.5: Results from the spatio-temporal interest points obtained from the motion of the legs of a walking person from the proposed method in [43]. (Left) 3D plot of a leg pattern and the detected interest points. (Right) The detected interest points overlaid on single frames

Motion Trajectory: In 2013, Wang et al. [86] proposed an approach to action recognition utilizing trajectory descriptors. Their method involved tracking the motion of objects

through trajectories from one frame to the next (see Figure 2.6). Initially, dense trajectories were computed, capturing the shape information of the trajectories over multiple frames. However, this only provided the trajectory shape information. To encode the motion information, descriptors such as HOG, HOF, and MBH were computed within the space-time volume. Subsequently, trajectory extraction was performed. For direct classification, a non-linear SVM with an RBF kernel was employed on the extracted trajectories. The recognized actions encompassed human-to-object and human-to-human interactions. Building upon this work, Wang et al. [85] further improved action recognition from dense trajectories by considering camera motion.



(a)



(b)

Figure 2.6: Computation of dense trajectories taken from [86]. (a) Illustration of the proposed approach. (b) Visualization of dense trajectories

2.2.1.2 Action Classifiers

After obtaining action representations, the next crucial step is to train action classifiers using labeled training samples. Action classifiers can be categorized into different types based on their characteristics and approaches. The classifiers are divided according to the following categories, and a comparison of the performance of the state-of-the-art that falls into each category is shown in Table 2.2, while a summary of all the mentioned state-of-the-art can be found in Table 2.1.

Direct Classification This category of approaches aims to summarize an action video by extracting a feature vector, which is then used for direct classification into specific action categories. The feature vector represents the essential information of the action and is designed to capture relevant spatiotemporal cues or other discriminative characteristics. The choice of features and the classifier depends on the specific requirements of the action recognition problem and the nature of the dataset. These methods leverage off-the-shelf classifiers, such as:

Support Vector Machine: SVMs are supervised learning models widely used for classification tasks. They aim to separate training examples into two categories by maximizing the margin between them, as depicted in Figure 2.7. The SVM algorithm constructs a hyperplane that best separates the data, and new examples are classified based on which side of the hyperplane they fall on. SVMs have been employed in various action recognition works, including [65], [38], [86], and [85], showcasing their effectiveness in this domain.

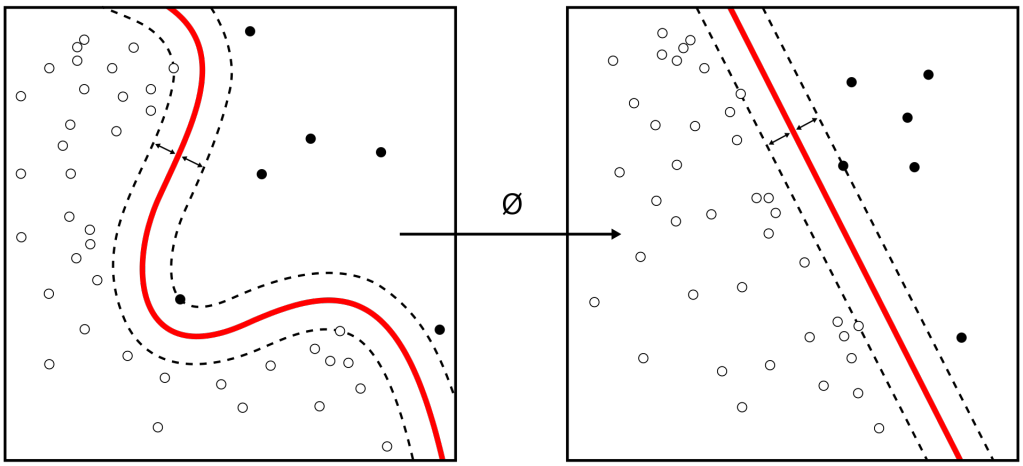


Figure 2.7: Scatter plot featuring a linear support vector's machine decision boundary from [90]

In the work presented in [44], action recognition is approached through two distinct methods. Firstly, automatic annotation of human actions is performed using movie scripts, enabling the association of actions with relevant textual information. Secondly, in the context of video classification, the method detects interest points based on the spatial-temporal extension of the Harris operator. This enables the construction of a spatiotemporal bag of features, which, in combination with the annotated vocabulary and an SVM classifier, facilitates human action recognition. Initially, the focus is on atomic actions with a local action representation. Subsequently, the scope is expanded to encompass human-to-object and human-to-human interactions, enhancing the recognition capabilities of the system.

K-nearest Neighbor: The KNN classifier is a popular method used for classification tasks. It operates by organizing all training data in a feature space, where each training example is represented by a feature vector along with its corresponding class label. When classifying a new example, the algorithm searches for the k closest neighbors based on a user-defined value of k, as depicted in Figure 2.8. The category assigned to the new example is determined by the majority class among its k nearest neighbors.

In the work conducted by [75], the KNN classifier is employed for action recognition. The method utilizes this classifier to classify action samples based on their feature vectors. Each training example represents a specific action, and the KNN algorithm determines the class label of new examples by considering the k nearest neighbors in the feature space.

By leveraging the KNN approach, the system can recognize and classify various human actions effectively.

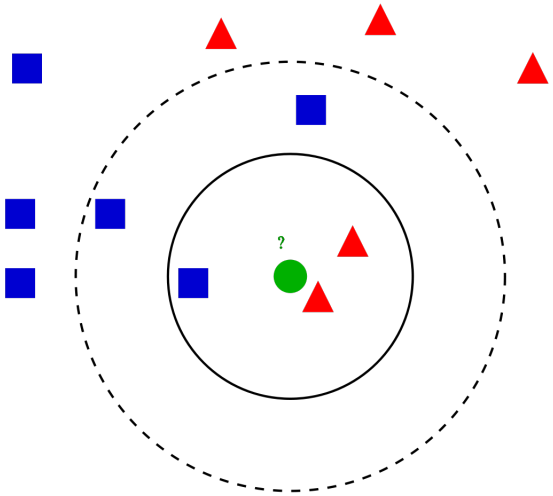


Figure 2.8: Example of k-NN classification taken from [89]

Sequential Approaches In the domain of action recognition, sequential approaches are employed to capture the temporal evolution of appearance or pose information. These methods utilize sequential state models to represent the dynamics of actions over time. By modeling the sequential nature of actions, these approaches can effectively capture the temporal dependencies and variations in action sequences.

Hidden Markov Models: One common type of sequential state model used in action recognition is the Hidden Markov Model (HMM). HMMs are probabilistic models that incorporate a hidden state sequence and an observed sequence of features. The hidden states represent the underlying action states, while the observed features correspond to the visual or pose information at each time step. HMMs enable the modeling of temporal transitions between action states, allowing for the recognition of complex action patterns.

Hidden semi-Markov Models: Hidden Semi-Markov Models (HSMMs) are an extension of Hidden Markov Models (HMMs) that capture temporal dependencies in sequential data by allowing the duration of each hidden state to follow a more general distribution. In the context of action recognition, HSMMs have been applied to model the temporal dynamics and hierarchical structure of human activities.

In a study by Duong et al. [20], they focused on learning and recognizing pre-labeled human activities during a morning routine using a Switching Hidden Semi-Markov Model (S-HSMM). This model consists of two layers: the atomic activity layer and the higher-level activity layer. The atomic activity layer represents individual activities and their durations, while the higher-level activity layer captures the temporal evolution and computes higher-level activities by aggregating lower-level ones into events. This approach enables the recognition of complex activities by capturing fine-grained actions and their temporal relationships. HSMMs offer a flexible framework for modeling temporal dependencies, allowing for the recognition of activities with varying durations and capturing

the hierarchical structure of human actions.

Max-Margin based Discriminative Feature Learning: In the study conducted by Raptis et al. [55], action recognition is addressed through the utilization of poselets keyframes. Poselets are specific parts of the human pose that play a crucial role in capturing discriminative information (see Figure 2.9). The authors propose a sparse representation of videos as a sequence of temporally discriminative keyframes, which essentially represents a collection of partial key poses of the actors involved. To encode these keyframes, HOG (Histograms of Oriented Gradients) and BOW (Bag-of-Words) components are employed, enabling spatial localization of poselet-like representations.

The classification of human activities is accomplished by learning from the keyframes within a max-margin discriminative framework. This approach allows for the learning of the most discriminative keyframes while also considering the local temporal context between them. The action recognition methodology presented in this work focuses on local representations and categorizes the recognized actions as atomic actions. By leveraging poselets keyframes and their corresponding features, this approach aims to capture the distinctive aspects of actions and enable accurate recognition based on localized pose information.

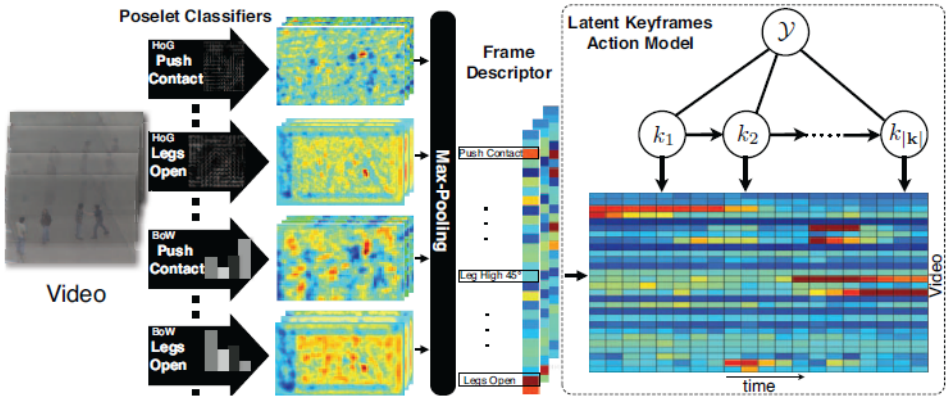


Figure 2.9: Framework of the model presented in [55]

Mid-Level Feature Approaches Mid-Level Feature Approaches aim to overcome the limitations of bag-of-words models in action recognition. While bag-of-words models are robust to background noise, they may lack expressiveness in capturing actions with significant appearance and pose variations. Moreover, there exists a semantic gap between low-level features and high-level actions, making it challenging to effectively represent actions using only low-level features. To address these challenges, hierarchical approaches have been proposed, introducing an additional layer of representations to better abstract low-level features for classification.

Semantic Descriptions: In hierarchical approaches, mid-level features are learned from the low-level features and subsequently utilized in the recognition task. These mid-level features can be viewed as knowledge extracted from the training database or expert-defined specifications. Recent advancements in action recognition have explored the

use of semantic descriptions or attributes, which provide a way to characterize complex human actions. By incorporating semantic information into activity classifiers, these approaches enhance the understanding of actions. For example, the work by Yao et al. [94] focuses on learning to understand human actions by matching specific attributes predicted from learned classifiers with static images of humans and objects (see Figure 2.10). This approach eliminates the need for explicit action representation and instead emphasizes action classification. The recognized actions are categorized as human-to-object and human-to-human interactions.

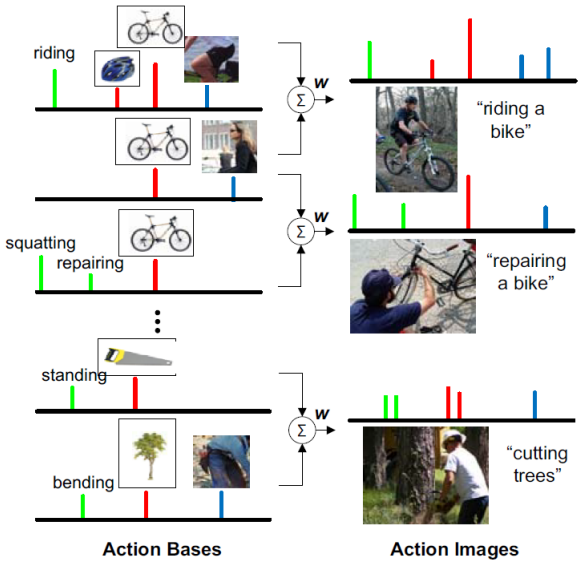


Figure 2.10: Output of the proposed method from [94]

While mid-level feature approaches that incorporate semantic descriptions have shown promising results in action recognition, it is important to note that they may require additional computational resources and annotations during the training phase. However, their ability to bridge the gap between low-level features and high-level actions contributes to more accurate and meaningful action recognition systems.

2.2.1.3 RGB-D Video Classifiers

Action recognition from RGB-D videos has gained significant attention with the introduction of cost-effective sensors like the Kinect. RGB-D videos offer an additional depth channel compared to traditional RGB videos, enabling the capture of 3D structural information. This depth information proves valuable in reducing background noise and simplifying intra-class motion variations.

The work by Shotton et al. [67] has had a profound impact on advancing state-of-the-art approaches in RGB-D video action recognition. Their contribution shifted the focus from action representation to action classification. As a result, action representation can now be classified as a holistic representation, while action classifiers have become the primary focus for RGB-D videos (see Figure 2.11). State-of-the-art approaches in this category

utilize the pre-built tracker introduced by Shotton et al. for action representation. Table 2.2 provides a comparison of the performance of various state-of-the-art approaches falling into this category.



Figure 2.11: Depth images from a Kinect device taken from the MSR-Action3D dataset

Joint-based approaches According to [80], skeleton-based action recognition approaches can be categorized into joint-based approaches and body-based approaches. Joint-based approaches represent the human skeleton using a set of points, namely the joints. For instance, in [87], 3D joint positions are estimated and used to construct a feature called local occupancy pattern (LOP), capturing the translational movement of each joint. The aggregation of consecutive translational movements of each joint over time is then structured by the Fourier Temporal Pyramid. The Actionlet Ensemble Model represents human actions by combining multiple captured features linearly, known as actionlets. Similarly, [92] extracts spatial-temporal joint differences to construct Eigen Joints, which are subjected to principal component analysis (PCA) and utilized by an NBNN classifier for activity recognition. Another example is [72], where the human skeleton is represented by fifteen joints, and various features, including body pose, hand position, motion information, image, and point-cloud features, are extracted and fed into a two-layered maximum-entropy Markov Model (MEMM) that captures the hierarchical nature of activities.

Body-based approaches On the other hand, body-based approaches represent the human skeleton in a stick-man figure style, with joints depicted as dots and limbs as the connections between consecutive dots. An example of a body-based approach is [80], which focuses on the relative geometry between different body parts. The geometry representing human actions is modeled as curves in the Lie group, enabling action classification using the Fourier temporal pyramid (FTP) and a linear SVM classifier.

Overall, these skeleton-based approaches, whether joint-based or body-based, contribute to the advancement of RGB-D action recognition by effectively leveraging the spatial-temporal information provided by the human skeleton.

Table 2.1: A quick summary of the classification and the evolution of the state-of-the-art. The symbol '✗' applies to references that only focus on action representation performance.

References	Year	Holistic	Local	Direct	Sequential	Mid-Level Feature	RGBD-Video
Forstner et al.[22]	1987		✓	✗			
Bobick et al.[6]	2001	✓		✗			
Schuldt et al.[65]	2004		✓	✓			
Weinland et al.[88]	2005	✓		✓			
Laptev et al.[43]	2005		✓	✗			
Dalal et al.[17]	2005		✓	✗			
Duong et al.[20]	2005	✓			✓		
Laptev et al.[44]	2008		✓	✓			
Klaser et al.[38]	2008		✓	✓			
Tran et al.[75]	2008	✓		✓			
Xia et al.[91]	2011	✓		✗			
Yao et al.[94]	2011		✓			✓	
Wang et al.[87]	2012	✓					✓
Yang et al.[92]	2012	✓					✓
Yang et al.[93]	2012	✓		✗			
Sung et al.[72]	2012	✓					✓
Raptis et al.[55]	2013		✓		✓	✓	
Wang et al.[86]	2013		✓	✓			
Wang et al.[85]	2013		✓	✓			
Vemulapalli et al.[80]	2014	✓					✓
Cao et al.[10]	2017	✓		✗			
Cao et al.[9]	2019	✓		✗			

Table 2.2: Listing of all state-of-the-art for Action Recognition and comparison of their accuracy in the main datasets (Self-P means the datasets that are self-produced and not the main ones)

Direct	**KTH**	**Weizman**	**UCF-50**	**HMDB-51**	**Hollywood2**	**IXMAS**
Schuldt et al.[65]	71.20%	–	–	–	–	–
Laptev et al.[44]	91.80%	–	–	–	–	–
Klaser et al.[38]	91.40%	84.30%	–	–	24.70%	–
Tran et al.[75]	–	**100.00%**	–	–	–	81.10%
Wang et al.[86]	**95.30%**	–	85.60%	48.30%	59.90%	**93.60%**
Wang et al.[85]	–	–	**91.20%**	**57.20%**	**64.30%**	–
Sequential	**UT-Int**	**Self-P**				
Duong et al.[20]	–	97.50%	–	–	–	–
Raptis et al.[55]	93.30%	–	–	–	–	–
Mid-Level	**PASCAL**	**Stanford 40**				
Yao et al.[94]	65.10%	45.70%	–	–	–	–
RGBD-Video	**CMU**	**Florence3D**	**MSR-A3D**	**UT-Kinect**	**MSRDaily3D**	**Self-P**
Wang et al.[87]	98.13%	–	88.20%	–	85.75%	–
Yang et al.[92]	–	–	91.97%	–	–	–
Sung et al.[72]	–	–	–	–	–	84.70%
Vemulapalli et al.[80]	–	90.88%	**92.46%**	97.80%	–	–

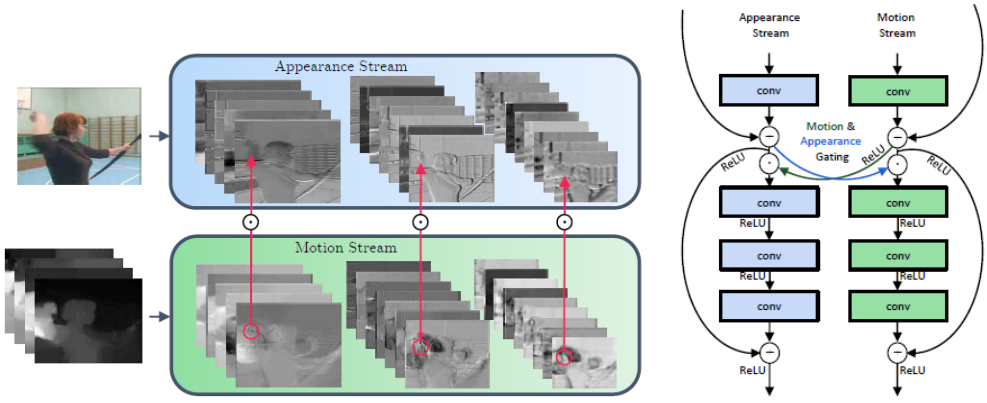
2.2.2 Deep Learning Methods

Deep learning methods have emerged as powerful techniques for action recognition, addressing the limitations of hand-crafted features. Traditional feature extraction methods

require extensive human labor and domain expert knowledge, often struggling to generalize well on large datasets. In contrast, deep learning techniques have gained increasing attention for their ability to learn highly effective and generalized features. The success of deep networks in action recognition can be attributed to their scalability, with networks containing tens of millions of parameters, and the availability of massive, labeled datasets. Consequently, recent deep networks have achieved remarkable performance on various action datasets.

Deep learning architectures encompass various types, including Deep Neural Networks, Deep Belief Networks, and Deep Reinforcement Learning. Among these architectures, Deep Neural Networks, particularly Convolutional Neural Networks (CNNs), have gained significant prominence independently [36], [68], [69], [77], [79], [11] or in combination [5], [19], [66], [49] with Recurrent Neural Networks (RNNs) [45], [70].

Residual Neural Networks (ResNets) have also been explored in action recognition [54], [21], [76]. ResNets, a variant of CNNs, incorporate skip connections or shortcuts to overcome short-term memory issues and alleviate accuracy saturation [21] (see Figure 2.12). Pseudo-3D Residual Nets [54] and (2+1)D convolutions in ResNets [76] are examples of these advancements.



(a) (b)

Figure 2.12: Example of a ResNet architecture taken from [21]. (a) Two-stream architecture (appearance + motion) with motion gating. (b) Illustration of a bidirectional gating accross the streams

CNNs are particularly popular in computer vision due to their ability to assign importance to various aspects within an image and distinguish between them [36] (see Figure 2.13). For instance, [68] employs a two-stream Convolutional Network for action recognition. The influence of depth in Convolutional Networks is studied by the same authors in [69], highlighting its impact on accuracy. [77] demonstrates that combining deep learning with local action representations like improved dense trajectories (iDT) and using a multi-class linear SVM as a classifier improves classification accuracy. Other works, such as [79] and [11], leverage CNNs for action recognition, with the latter reviewing state-of-the-art architectures for the Kinetics Human Action Video dataset and proposing

the two-stream inflated 3D ConvNet (I3D) approach, demonstrating the transferability of learned representations across datasets.

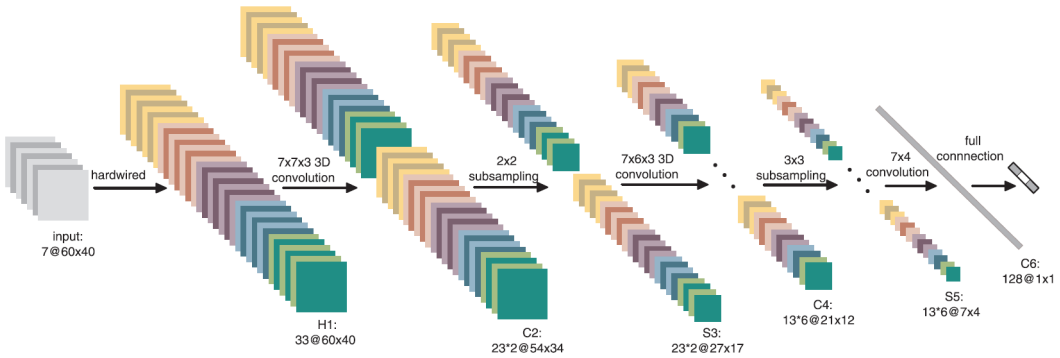


Figure 2.13: 3D CNN taken from [36]

RNNs, with their inherent memory capabilities suitable for processing sequential data, have been successfully applied to action recognition. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are two popular RNN architectures. [45] and [70] explore these architectures in the context of action recognition. The former introduces a Spatio-Temporal LSTM (ST-LSTM) for learning upon the 3D location of each joint in video frames, incorporating a skeleton-based tree traversal method to analyze kinematic properties and improve spatial evolution understanding. The latter presents a spatial-temporal attention-based model, built on an LSTM RNN, that selectively focuses on important joints from skeleton data, enhancing accuracy through joint learning.

Combining CNNs with RNNs has proven beneficial, as demonstrated in [5], which learns 3D features from CNNs (see Figure 2.14) and employs an LSTM RNN to classify sequences based on the temporal evolution of learned features (see Figure 2.15). Similar architectures, such as Long-term Recurrent Convolutional Networks (LRCN), are explored in [19] and [49], evaluating their performance not only for action recognition but also for image and video description. Additionally, soft attention-based models, like the one proposed in [66], selectively focus on essential parts of video frames.

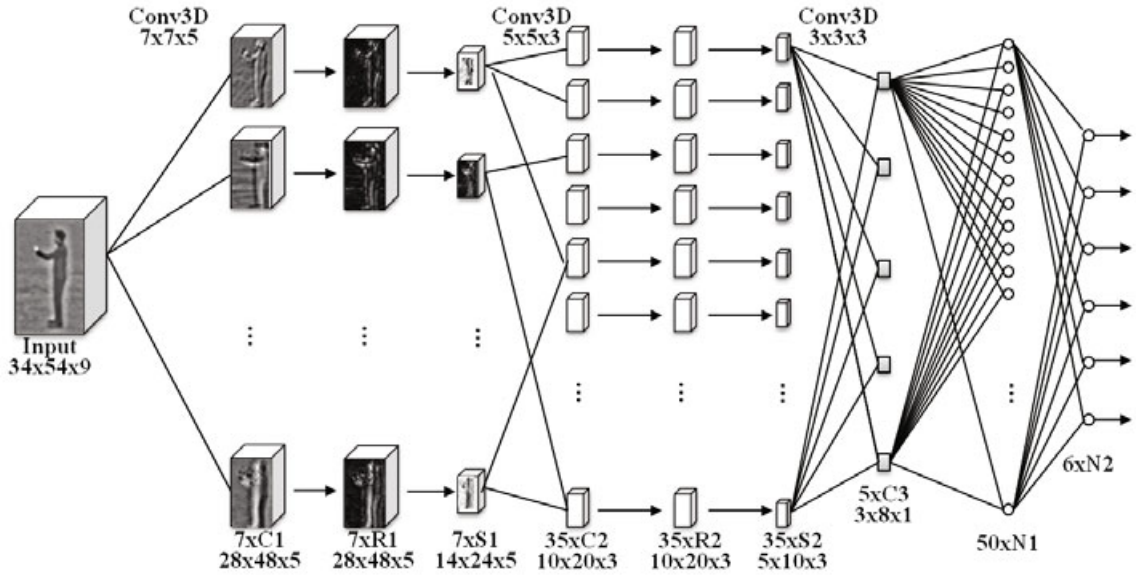


Figure 2.14: The 3D CNN from [5]

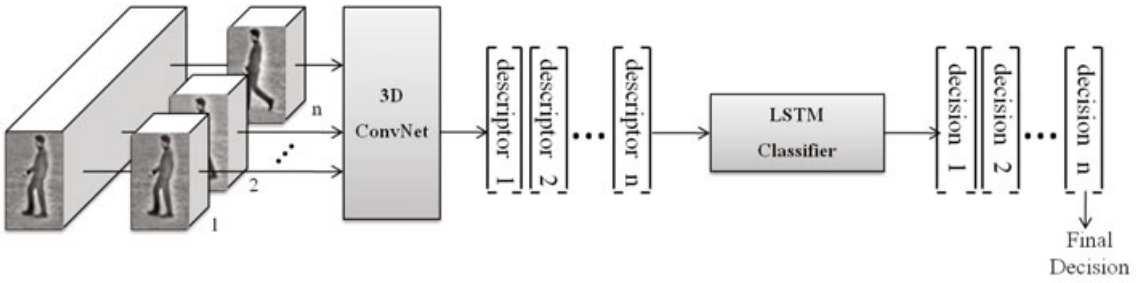


Figure 2.15: Architecture overview from [5]

Table 2.3 summarizes the detailed approaches discussed above, while Table 2.4 presents their respective performance on major datasets. These deep learning methods have significantly advanced the field of action recognition, leveraging the power of deep neural networks to learn discriminative and robust representations from raw data.

Table 2.3: Listing of all state-of-the-art for Action Recognition and comparison in Deep Learning Methods

References	Year	Method
Baccouche et al.[5]	2011	3D Convolutional Neural Network + LSTM Classifier
Ji et al.[36]	2012	3D Convolutional Neural Networks
Simonyan et al.[68]	2014	Two-Stream Convolution Networks (spatial + temporal)
Simonyan et al.[69]	2014	Two-Stream Convolution Networks (spatial + temporal)
Tran et al.[77]	2015	3D Convolutional Neural Networks
Donahue et al.[19]	2015	Long-Term Recurrent Convolutional Networks (CNN + LSTM + W)
Sharma et al.[66]	2015	Soft Attention Based Model (CNN + several deep ST-LSTM units)
Liu et al.[45]	2016	Spatio-Temporal LSTM (Tree Traversal) with Trust Gates
Ma et al.[49]	2016	LSTM-s (LSTM + CNN)
Song et al.[70]	2017	Spatio-Temporal Attention Model + LSTM
Qiu et al.[54]	2017	Pseudo-3D Residual Networks
Varol et al.[79]	2017	Long-term Temporal Convolutions for Convolutional Neural Networks
Carreira et al.[11]	2017	Two-Stream Inflated 3D Convolutional Neural Network (Flow + RGB) + IDT
Feichtenhofer et al.[21]	2017	Two-Stream Spatio-Temporal Residual Network + IDT
Tran et al.[76]	2018	Residual Networks with (2+1)D Convolutions

Table 2.4: Listing of all state-of-the-art for Action Recognition and comparison of their accuracy in the main datasets in Deep Learning Methods

References	KTH	UCF-101	UCF-11	HMDB-51	ActivityNet	NTU	SBU
Baccouche et al.[5]	**94.39%**	–	–	–	–	–	–
Ji et al.[36]	90.20%	–	–	–	–	–	–
Simonyan et al.[68]	–	88.00%	–	59.40%	–	–	–
Simonyan et al.[69]	–	–	–	–	–	–	–
Tran et al.[77]	–	90.40%	–	–	–	–	–
Donahue et al.[19]	–	82.66%	–	–	–	–	–
Sharma et al.[66]	–	–	**84.96%**	41.31%	–	–	–
Liu et al.[45]	–	–	–	–	–	77.70%	**93.30%**
Ma et al.[49]	–	–	–	–	55.10%	–	–
Song et al.[70]	–	–	–	–	–	**81.20%**	91.51%
Qiu et al.[54]	–	93.70%	–	–	**78.86%**	–	–
Varol et al.[79]	–	92.70%	–	67.20%	–	–	–
Carreira et al.[11]	–	**97.90%**	–	**80.20%**	–	–	–
Feichtenhofer et al.[21]	–	94.90%	–	72.20%	–	–	–
Tran et al.[76]	–	97.30%	–	78.70%	–	–	–

2.3 Action Prediction

Action prediction plays a crucial role in human-robot cooperation, enabling intelligent systems to anticipate and respond to human actions in real time. While traditional action recognition methods focus on labeling actions after they have occurred, action prediction approaches aim to infer action labels or future trajectories before the actions are completed. This predictive capability is essential for time-sensitive tasks and proactive decision-making. In this section, it is categorized the importance of action prediction into three categories: Partial Videos Prediction, Motion Prediction, and Path Prediction. A summary of the state-of-the-art discussed below can be found in Table 2.5.

2.3.1 Partial Videos Prediction or Early Recognition

Partial Videos Prediction involves recognizing activities in video sequences without observing the full length of the sequence. Early recognition is essential in scenarios where intelligent systems need to react promptly. Several methods have been proposed in this area.

Ryoo [62] presents a probabilistic action prediction approach using integral and dynamic bag-of-words representations. It focuses on recognizing ongoing activities by modeling the temporal changes of feature distributions (see Figure 2.16).

Ryoo et al. [63] introduce an early recognition method that considers activity onsets, which are signals observed before the start of an activity. Cascade histograms of weak classifier responses are constructed to capture the correlation between onsets and target activities.

Koppula and Saxena [41] propose a conditional random field (CRF) model for spatiotemporal activity prediction. They model the present, past, and future activities using CRFs and utilize an anticipatory temporal CRF for future activity anticipation.

Han et al. [31] use a two-stream Residual CNN and stacked LSTM to forecast generalized actions. The Residual CNN generates local action features, which are then fed into the LSTM for predicting the next action.

Villegas et al. [82] propose the Motion-Content Network (MCnet) to predict future video sequences by separately modeling motion and content. Motion captures local dynamics, while content encodes the spatial layout of salient parts, and both components are used to estimate future frames.

Kong et al. [40] introduce a mem-LSTM model that stores hard-to-predict training samples. By accumulating challenging samples, the model expands the classification boundaries and improves the prediction of incomplete videos.

Cai et al. [7] propose a two-stage learning framework for action prediction in partially observed videos. They transfer knowledge learned from complete videos to improve action recognition in partial videos, enhancing overall performance.

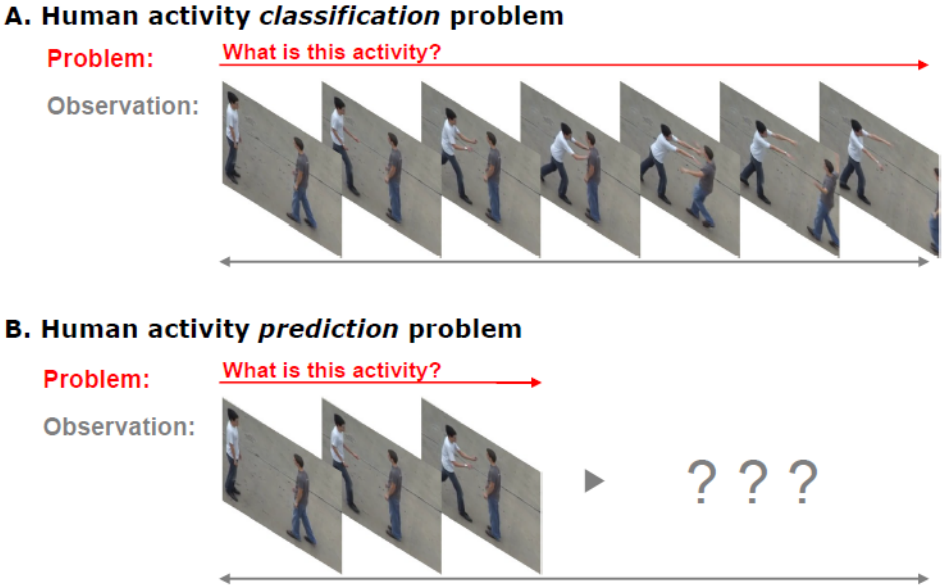


Figure 2.16: Comparison between activity classification and activity prediction taken from [62]

2.3.2 Motion Prediction

Motion Prediction focuses on estimating the next human pose or action trajectory. It is useful for short-term predictions and explicit action understanding. Several methods have been developed in this area.

Luo and Berenson [48] address human-reaching motion recognition and prediction using unsupervised online learning. They build a human motion library based on GMM and calculate the probability of different trajectories. The library is used to estimate palm and arm positions.

Jain et al. [35] propose Structural-RNN (S-RNN), which combines deep recurrent neural networks with spatiotemporal graphs. S-RNN captures high-level spatiotemporal knowledge from complex human activities, enabling accurate modeling and prediction of human poses and interactions.

Walker et al. [84] utilize variational autoencoders to forecast dense trajectories of pixels in static images. The network conditions the output space based on input images, leveraging object location and scene context to predict possible motions (see Figure 2.17).

Martinez et al. [50] demonstrate the effectiveness of deep recurrent neural networks, specifically Gated Recurrent Units (GRUs), in human motion prediction. Even basic GRU models, trained on ground truth and predictions, achieve state-of-the-art performance in short-term forecasting of future 3D poses.

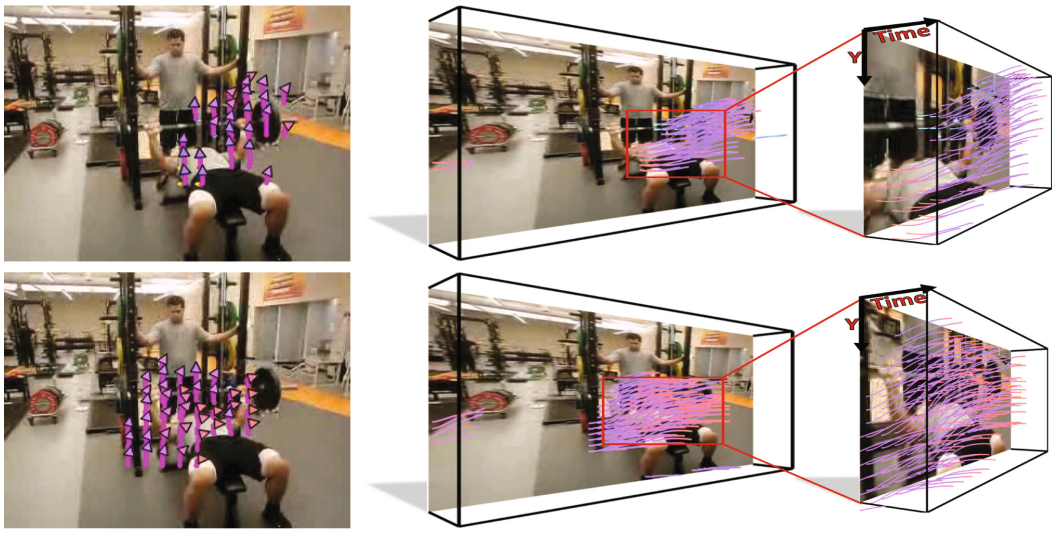


Figure 2.17: Picture of bench-press action taken from [84]. On the left is represented the projection of the trajectories onto the image plane and on the right the trajectories are illustrated in 3D space-time

2.3.3 Path Prediction

Path Prediction involves forecasting the trajectory of human motion, which is crucial in human-robot cooperative environments. It enables the system to anticipate human intentions and adapt its behavior accordingly. Several approaches have been proposed for path prediction.

Alahi et al. [4] propose Social LSTM, a recurrent neural network model that incorporates social interactions among pedestrians to predict their future trajectories. The model takes into account both local interactions and global scene context (see Figure 2.18).

Liu et al. [46] propose a recurrent model that incorporates spatial and temporal contexts to predict the next location in a given sequence. The model leverages the recurrent neural network (RNN) architecture to capture the dependencies between consecutive locations and incorporates spatial and temporal information to improve the accuracy of the prediction.

Su et al. [71] focuses on predicting plausible paths in crowded scenes. The authors propose a method that combines a recurrent neural network with social force modeling to generate realistic trajectories for individuals in a crowd. The approach takes into account both the individual's intentions and the social interactions among multiple individuals to forecast their plausible paths.

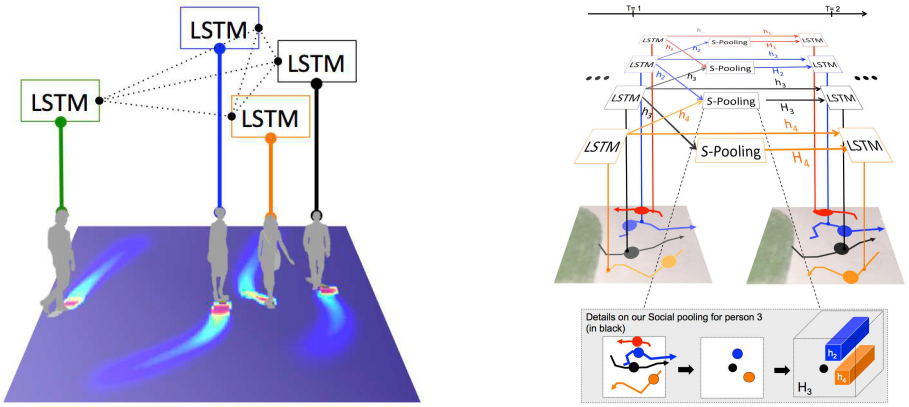
Gupta et al. [30] introduces a generative adversarial network (GAN)-based approach for generating socially acceptable trajectories. The proposed model leverages a combination of a generator network and a discriminator network to learn the distribution of socially acceptable trajectories. The discriminator helps guide the generator to produce trajectories that align with human behavior and social norms, making them suitable for action prediction in social environments.

Deo et al. [18] introduce a data-driven approach called Social GAN (Generative Adversarial Network) for pedestrian trajectory prediction. It uses a conditional GAN to generate plausible future trajectories based on observed motion patterns and social interactions.

Sadeghian et al. [64] propose Trajectron++, a generative model for multi-agent trajectory forecasting. It represents agents as a combination of point masses and rigid bodies, taking into account interaction forces, scene context, and past motion history to predict future trajectories.

Rhinehart et al. [56] present a method called Social-STGCNN (Social Spatio-Temporal Graph Convolutional Neural Network) for predicting pedestrian trajectories in crowded scenes. The model incorporates social context through graph convolutional networks and predicts future trajectories based on historical observations.

These are just a few examples of the methods and techniques used for action prediction in human-robot cooperation scenarios. They provide valuable insights in various contexts, such as predicting the next location, forecasting plausible paths in crowded scenes, and generating socially acceptable trajectories. The field is continuously evolving, with ongoing research focusing on improving prediction accuracy, handling complex interactions, and incorporating real-time feedback for adaptive decision-making in dynamic environments.



(a) (b)

Figure 2.18: Illustration of the proposed method by [4]. (a) Illustration of the computed trajectories (b) Overview of the proposed Social-LSTM which incorporates a pooling strategy

Table 2.5: Listing of all state-of-the-art for Action Prediction and comparison

References	Year	Method
Partial Videos Prediction		
Ryoo et al.[62]	2011	Integral BOW or Dynamic BOW
Ryoo et al.[63]	2015	Features + Onset signatures
Koppul et al.[41]	2015	CRF + ATCRF
Ma et al.[49]	2016	CNN + LSTM
Han et al.[31]	2017	CNN + LSTM
Villegas et al.[82]	2017	MCnet + RES
Kong et al.[40]	2018	CNN + LSTM
Cai et al.[7]	2019	Features + Knowledge transfer
Motion Prediction		
Luo et al.[48]	2015	two-layered GMM
Jain et al.[35]	2016	ST-Graphs + Structural-RNN
Walker et al.[84]	2016	Variational AutoEncoders
Martinez et al.[50]	2017	Seq2Seq + Residual
Path Prediction		
Alahi et al.[4]	2016	Social-LSTM
Liu et al.[46]	2016	ST-RNN
Su et al.[71]	2017	SRGP
Gupta et al.[30]	2018	Generator + Pooling Module + Discimina-tor (Socially-aware GAN)
Sadeghian et al.[64]	2019	Feature Extractor + Attention Module + LSTM based GAN (SoPhie)

3

Proposed Approach

As demonstrated in Chapter 2 (State-of-the-Art Review), there exists a broad spectrum of computer vision-based research that enables rapid classification of complex human activities. Building upon the problem outlined in Chapter 1 (Introduction), which pertains to the segregation of humans and manipulators in the industrial domain, this dissertation proposes a hierarchical architecture as an enhancement to current autonomous systems. This architecture incorporates two key elements: encapsulation of computer vision technologies to acquire knowledge about the physical environment (specifically, human intentions), and their subsequent integration preceding the decision-making component. This integration facilitates the selection of the most appropriate cooperative interaction. To validate the developed architecture, an experimental setup is proposed, comprising an industrial manipulator assigned a simple pick-and-place task, representative of typical industrial scenarios where human interaction is prevalent.

3.1 Architecture Overview - A Theoretical Approach

As shown in Figure 3.1, the hierarchical architecture breaks down into four layers, each represented with a different color to aid the reader's comprehension. Additionally, they are represented in this pyramid shape, to address the hierarchical nature of the main challenges regarding a human-robot interaction system which consists of first Awareness of something, or someone followed by the Cooperation decision and guaranteeing Safety while planning and executing in the latter stage. This pyramid design also serves as a visual guidance that the information gathered will become consequently more specific as it goes up the layers. The first layer, denoted as the Physical Layer and depicted in red, encompasses the hardware components of the architecture. It serves as the repository for vast quantities of information in its raw form. Represented by the color blue, the Interface Layer serves as the bridge connecting the physical world to the digital realm, enabling seamless communication between the two. The Digital Layer, illustrated in green, receives the information in a digital format and aggregates it. This layer plays a crucial role in extracting knowledge from the abundant data, utilizing various techniques

and algorithms, and disseminating it to the higher layers. Additionally, it directs specific knowledge to the preceding layers, as detailed later in this dissertation. At the pinnacle of the hierarchy, the Cognitive Layer, depicted in purple, possesses decision-making capabilities. It integrates the extracted knowledge and performs inferences to determine the most appropriate cooperative action based on human intentions. Both Awareness and Safety aspects are addressed within the Digital Layer, while Cooperation resides in the Cognitive Layer.

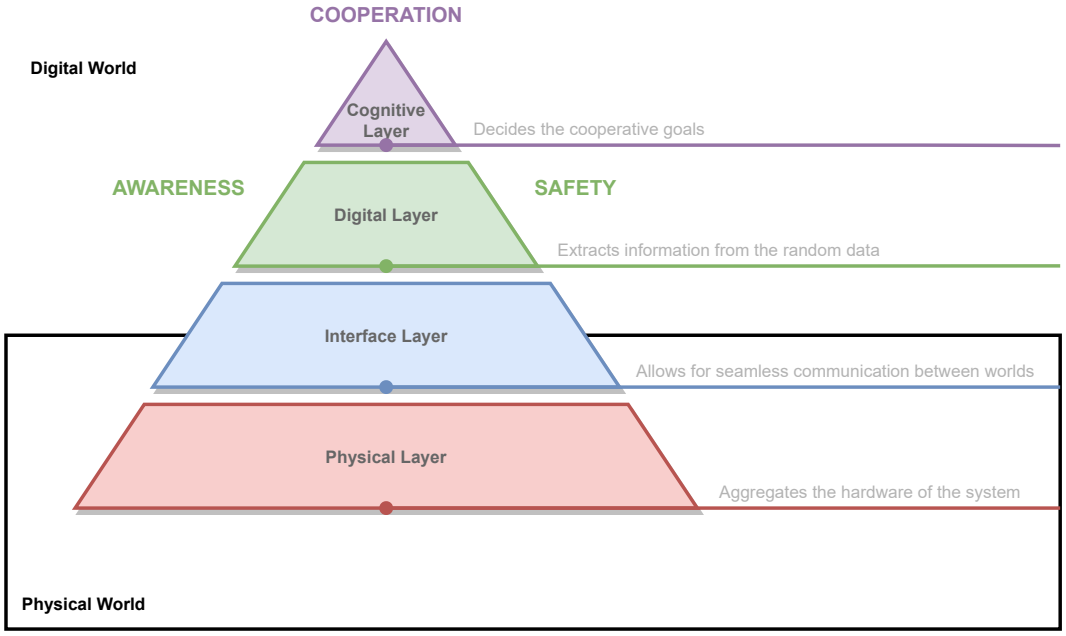


Figure 3.1: Hierarchical Architecture to be implemented

3.2 Architecture Design - An In-Depth View

Diving into the detailed implementation of the hierarchical architecture, the Physical Layer serves as the foundation for the architecture, encompassing the physical elements such as the UR3e robot, Robotiq Gripper, and Intel RealSense D435i camera.

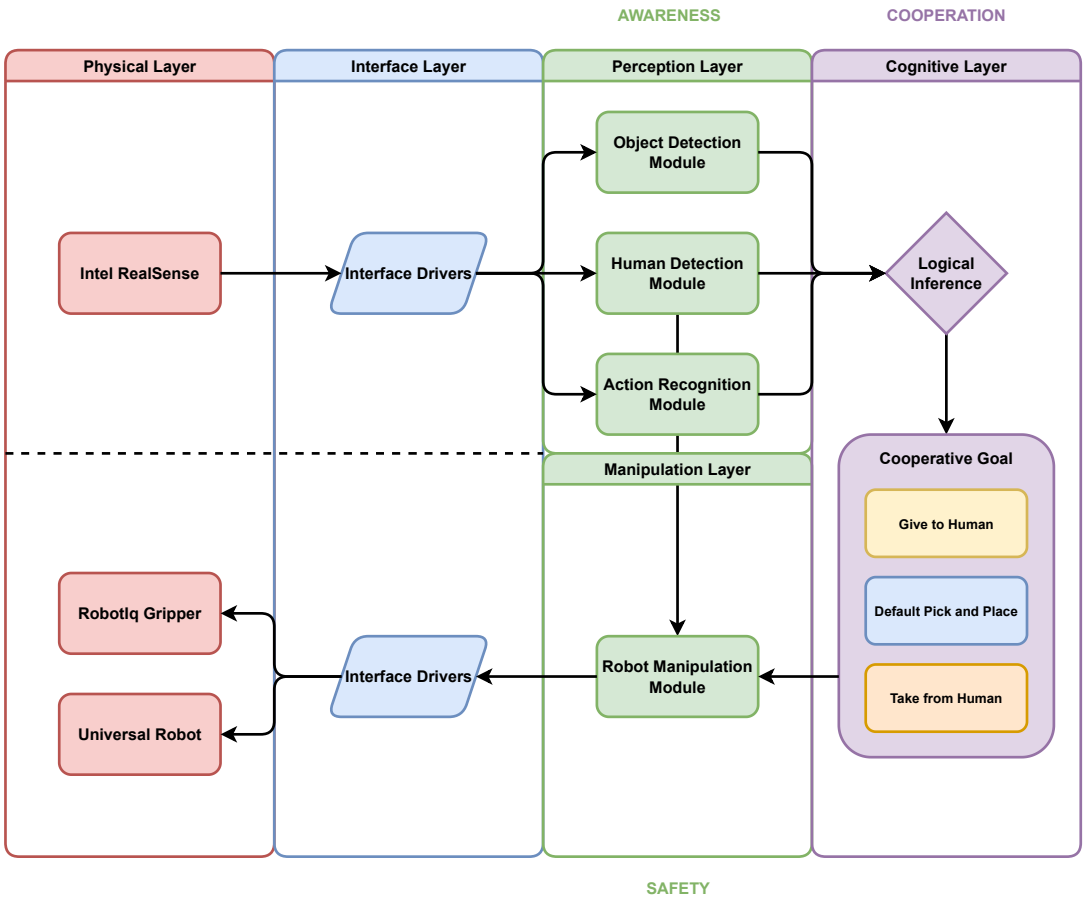


Figure 3.2: Hierarchical Architecture functionality

3.2.1 Safety

3.2.1.1 Robot Manipulation Module

Reasoning At the digital level, the industrial manipulator is maneuvered through a series of predefined commands executed sequentially upon receiving trigger signals. The Robot Manipulation Module, belonging to the Manipulation Layer, is responsible for delivering these commands. This module receives trigger signals from the Cognitive Layer and plays a crucial role in maneuvering the manipulator. The trigger signals received by this module consist of 3D location and rotation poses, which serve as target poses for the manipulator's end effector. This is essential for achieving precise positioning during pick-and-place operations. Unlike an autonomous system, the cooperative nature of the environment may require readjustment of pick or place poses, making this module pivotal in ensuring accurate manipulation. Figure 3.3 illustrates this concept.

Development To aid comprehension, Figure 3.3 presents the functionality of the Robot Manipulation Module in a specific ROS (Robot Operating System) context. A color scheme is implemented, with each module represented by a specific color corresponding

to its layer. The physical elements are depicted in red, while the ROS nodes that aggregate specific functionalities are represented in orange. Blue represents the tools that complement the nodes, and black arrows indicate the trigger signals received. Black lines ending in a dot shape represent control actions, which will be explained in detail later.

For seamless maneuvering of the UR3e manipulator and Robotiq Gripper, two tools were employed. The UR RTDE tool [57] facilitates control of the Robotiq Gripper. The **Gripper Control** node provided by this tool enables smooth operation of the gripper, allowing functions like opening and closing of the gripper. Additionally, the MoveIt 2 tool [51] simplifies the operation of the manipulator and other manipulators from different manufacturers. The **Move Group** node from this tool is utilized and provides three primary functions: defining a target pose for the end effector, planning a collision-free trajectory to the target pose, and executing the planned trajectory.

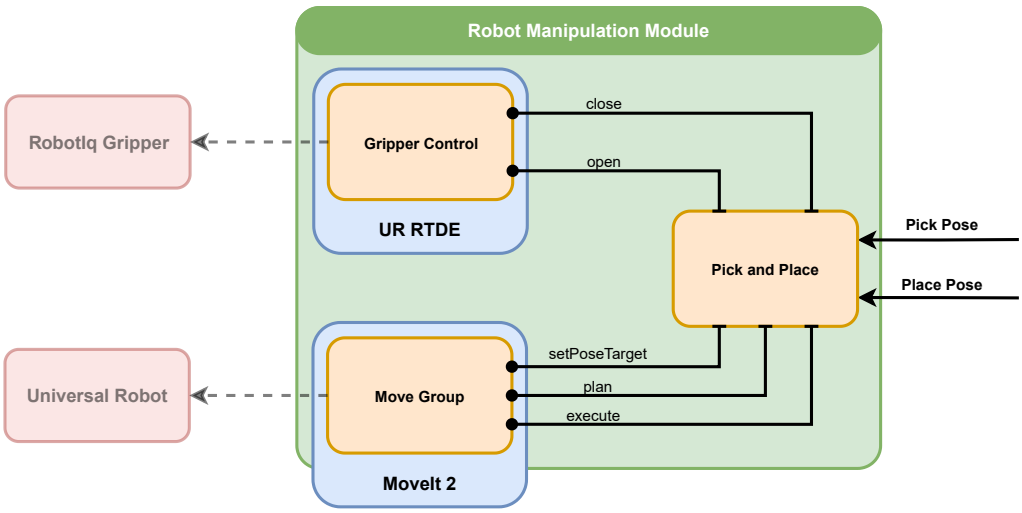


Figure 3.3: Visual representation of the functionality of the Robot Manipulation Module

To facilitate the typical pick-and-place operation, the **Pick and Place** node was developed within the Robot Manipulation Module. Figure 3.4 illustrates the framework of operations within this node. The gray boxes represent the sequence of triggers for the high-level functions. Upon receiving a pose with 3D location and rotation, the node sets that pose as the target for the manipulator's end effector. The Move Group is then triggered to plan a collision-free trajectory to the target pose, continually iterating until a successful trajectory is found. Once a trajectory is obtained, the execution command is triggered, leading the manipulator to follow the planned trajectory. Finally, the Gripper Control is triggered to open or close the gripper based on whether it is in the pick or place area.

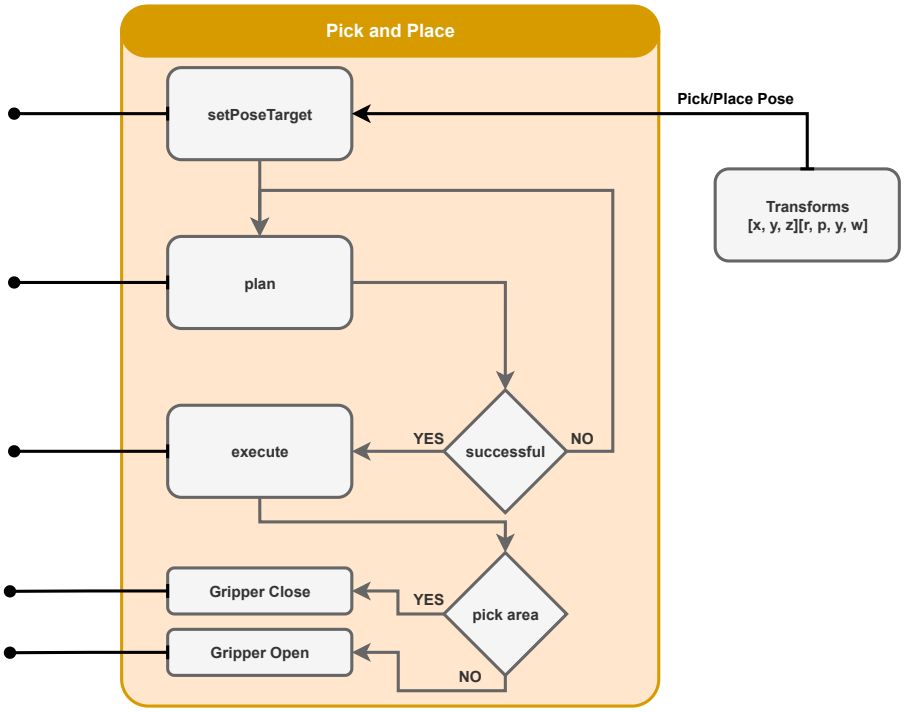


Figure 3.4: Pick and Place function decision process

This framework ensures seamless maneuvering of the manipulator, enabling precise and efficient pick-and-place operations in the cooperative environment.

3.2.2 Awareness

A mere autonomous system doesn't go beyond the setup described so far which requires an immutable environment to work properly. For this reason, the proposed architecture goes beyond it by introducing a perception element to the Physical Layer: the Intel RealSense D435i camera. This addition provides constant feedback on the physical environment, making the cognitive system dynamic and responsive to changes in its surroundings.

While this new element is a step further it is the implementation of what is called Perception Modules that contributes fully to the capture of knowledge to address the hypothesis at hand. These modules are distinguished in the Perception Layer, the other sublayer of the Digital Layer that plays a crucial role in capturing knowledge from the received physical environment data as is discussed in more detail below.

3.2.2.1 Action Recognition Module

Reasoning Cooperation is a central focus of this dissertation, and understanding human intention is essential for achieving efficient collaboration. To tackle this challenge, the Action Recognition Module was developed. This module is responsible for the real-time labeling of human actions as they are being performed.

Development Figure 3.5 provides a visual representation of the Action Recognition Module's functionality in a ROS-specific context, following the same visual scheme described earlier. The module utilizes the **Action Recognition** node provided by the MMAction2 tool [12] to label human actions using RGB data from the Intel RealSense camera.

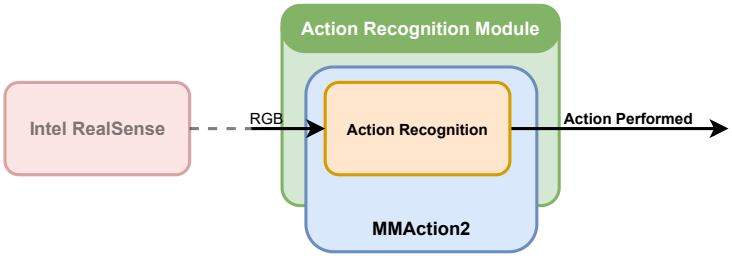


Figure 3.5: Visual representation of the functionality of the Action Recognition Module

One of the main advantages of using the MMAction2 tool is its provision of pre-made deep-learning models for video understanding. These models are pre-trained on extensive universal datasets, as detailed in Table 3.1. The MMAction2 tool offers configuration files that contain information about the model, dataset, training, validation, testing pipelines, optimizer configuration, and runtime settings. It also provides APIs for training and testing, allowing for easy adaptation with custom datasets and model settings. The training API generates a file that stores the model's weights, which are then used by the testing API for accurate predictions on real-time data, including MP4 videos or webcam feeds.

Table 3.1: Description of the different datasets each model for action recognition available on the MMAction2 was pre-trained on

Models	**#**	**Pre-Trained On**
C3D	1	UCF101
CSN	1	Kinetics400
I3D	1	Kinetics400
R2plus1D	1	Kinetics400
SlowFast	2	Kinetics400 + STHV1
SlowOnly	8	Kinetics400 + Kinetics600 + Kinetics700 + GYM99 + Jester + HMDB51 + UCF101 + STHV1
TANet	2	Kinetics400 + STHV1
TimeSformer	1	Kinetics400
TIN	3	Kinetics400 + STHV1 + STHV2
TPN	2	Kinetics400 + STHV1
TRN	2	STHV1 + STHV2
TSM	7	Kinetics400 + Diving48 + STHV1 + STHV2 + Jester + HMDB51 + UCF101
TSN	12	Kinetics400 + Kinetics600 + Kinetics700 + Diving48 + HMDB51 + UCF101 + STHV1 + STHV2 + MIT + MMIT + ActivityNet + HVU
X3D	1	Kinetics400
ResNet for Audio	1	Kinetics400

Before the model can effectively recognize actions, it must undergo training. Different models learn in various ways and perform differently depending on the dataset. However, the basic principle remains the same. During training, the model analyzes labeled data and detects patterns or clues in the information. These patterns are then stored as weights within the model's neurons and connections. The model assigns more significance to paths that accurately classify the input information, increasing their weights. In summary, this is how a deep learning model operates. During training, the model defines paths that best classify the provided training data. When presented with similar actions during testing, the model identifies similar patterns and concludes that the specific action is being performed. Validation is performed alongside training, using different videos to ensure the effectiveness of the training process. This involves adjusting the weights of the model if necessary. Figure 3.6 provides an abstract visualization of the training process.

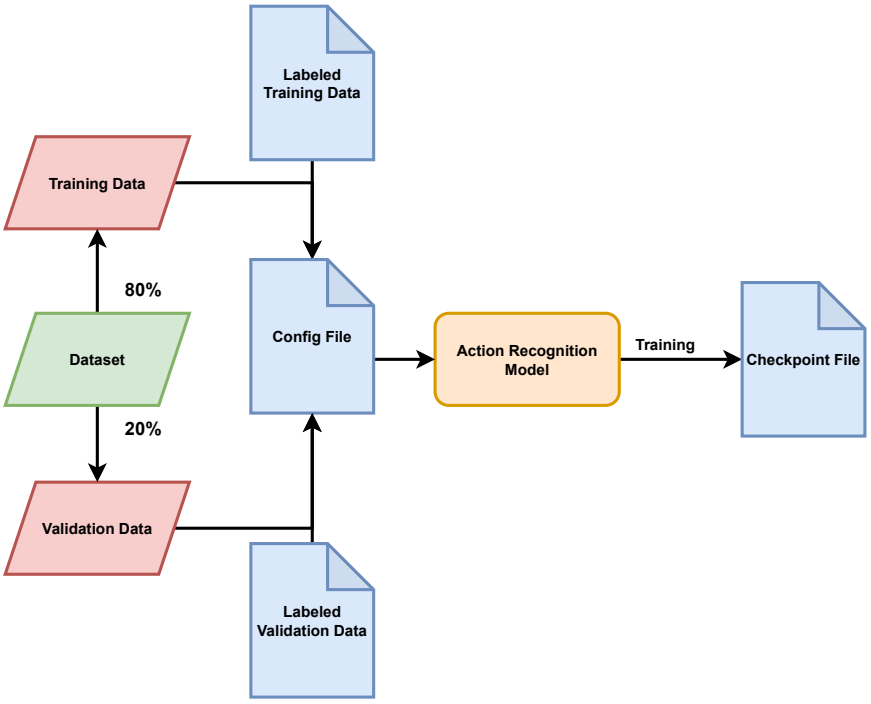


Figure 3.6: Visual representation of the training stage

The chosen model was trained to predict two classes efficiently: *giving something* and *asking for something*. The former class is assigned to any human in the image holding an object with their arms extended, while the latter class is assigned to any human in a beggar-like or waiting-to-be-given-something pose, with their arms slightly further apart. By reducing the number of classes, training becomes more focused.

To train the model, a dataset was created using raw frames. A total of 14 videos were recorded, seven for each action, in 1080p 60fps video settings. After trimming the videos to include only the intended actions, the dataset comprised 12,878 frames. Following the common practice of an 80% training and 20% testing split, the model was trained using 10 videos (five for each action) and tested using four videos (two for each action). This resulted in training with 8,704 frames and testing with 4,174 frames. Figures 3.7 and 3.8 display example frames from each cooperative action. Figure 3.7 showcases frames from videos depicting the *asking for something* action, while Figure 3.8 shows frames from videos depicting the *giving something* action. The human pose remains consistent across videos within the same action, ensuring better classification. The background was kept constant, with clothing and the object being held by the human as the only variables. This approach aims to make the cognitive system agnostic to these factors.



Figure 3.7: Seven frame examples, one from each video performing the cooperative action *asking for something*



Figure 3.8: Seven frame examples, one from each video performing the cooperative action *giving something*

To select the action recognition model, it was evaluated different configuration files for each pre-trained model available in the MMAction2 toolbox. Each model was tested using the custom-made dataset and their classification performance was analyzed. The SlowFast model pre-trained on the Something-Something dataset performed the best. While there was no directly similar label for the *asking for something* action in the pre-trained model, the giving something action was easily classified as *Holding something* or *Holding something in front of something* labels from the referenced dataset. Therefore, a pre-built base model based on the SlowFast network was chosen.

Table 3.2 provides detailed information about the four components of the chosen SlowFast model. The model utilizes the Recognizer3D as the recognizer and the ResNet3dSlowFast backbone for extracting feature maps. The classification head, named SlowFastHead, predicts three classes: *asking for something*, *giving something*, and *doing something*. No localizer component was used. The configuration file for each component is referenced in the table, with slight modifications made to the default settings for the SlowFast model trained on the Something-Something-V1 dataset. The changes primarily involved adjusting the number of classes in the classification head and dataset-related configurations. Training was conducted using the Google Colab platform and the respective API provided by the MMAction2 toolbox.

Table 3.2: SlowFast model configuration

Model [15]	
Recognizer	Recognizer3D [13]
Backbone	ResNet3dSlowFast [14]
Classification Head	SlowFastHead [16]
Localizer	N/A

After training, the model enters the testing phase, where its accuracy and other metrics are evaluated. During testing, the trained model classifies real-time data using the dataset labels and the stored weights obtained from the training phase. Figure 3.9 illustrates this process.

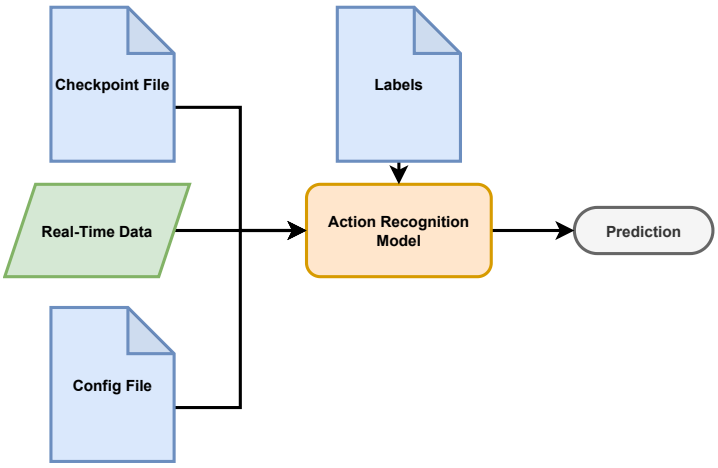


Figure 3.9: Visual representation of the testing stage

3.2.2.2 Human Detection Module

Reasoning To achieve full cooperation, certain minimum safety requirements must be met. One of these requirements is ensuring that the manipulator operates with collision-free trajectories. This can be achieved by being aware of the human presence, specifically their body pose, and location. Since the previous module focused on classifying human intention, which is independent of body location and poses, the Human Detection Module is introduced. This module is responsible for decomposing the human body pose and location in the 3D space using data received through the perception element. The information obtained from this module is crucial for collision-free planning.

Development Figure 3.10 provides a more specific illustration of the functionality of the Human Detection Module. This module utilizes two main tools: the OpenPose tool [9] (covered in Chapter 2 - State-of-the-Art Review) and the ROS OpenPose tool [37].

The OpenPose tool begins by segmenting humans based on RGB data, a process known as **Human Segmentation**. It then computes the 2D skeleton of each human, known as

Limb Computation. This tool is instrumental in providing information about human body pose from RGB data.

In addition to the OpenPose tool, the ROS OpenPose tool is used to complement the previous work. It utilizes depth data to convert the 2D human skeleton obtained from the OpenPose tool into the 3D space, a process referred to as **2D to 3D Conversion**. This conversion enables the representation of the human body's pose and location in a three-dimensional coordinate system.

By combining the outputs of the OpenPose tool and the ROS OpenPose tool, the Human Detection Module can accurately decompose the human body pose and location in the 3D space. This information is crucial for subsequent collision-free planning processes within the cognitive system.

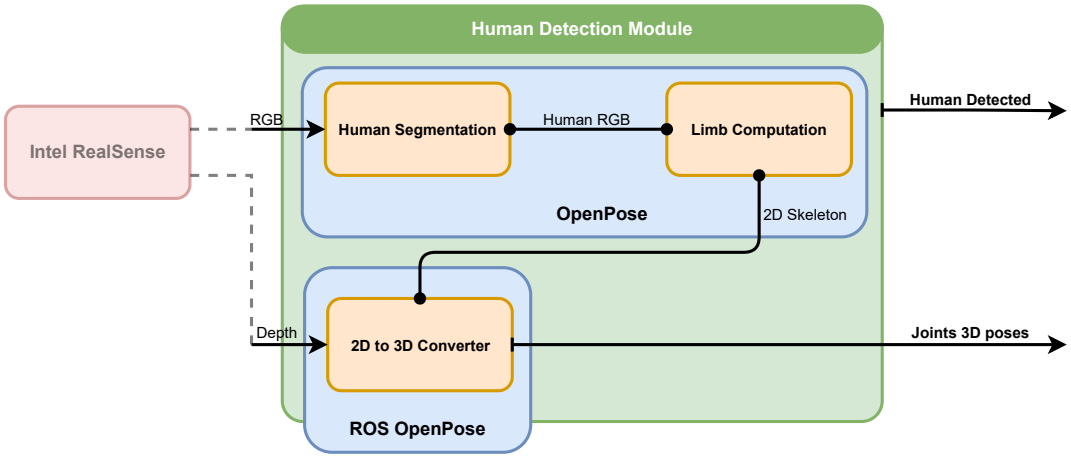


Figure 3.10: Visual representation of the functionality of the Human Detection Module

3.2.2.3 Object Detection Module

Reasoning To test the feasibility of the proposed architecture, the manipulator will be performing a typical pick-and-place operation. Furthermore, the Action Recognition Module, as discussed earlier, will distinguish between two cooperative human intentions: *asking for something* and *giving something* while holding an object. In response to the latter, as explained in the Cooperative Layer, the manipulator will be triggered to pick the object between the human's hands and place it in the default position. To achieve this, an Object Detection Module is proposed. This module will extract information about the location and orientation of existing objects in the physical environment from the received RGB-D data.

Development Figure 3.11 illustrates the process and tools used for the Object Detection Module. The colors used in the figure have the same significance as described earlier. The RGB-D data is first used to segment each object based on its individual RGB and depth information, a process known as **Object Segmentation**. The segmented objects are then used to estimate the best approach for grasping them, a step referred to as **Grasp**

Computation. These functionalities are provided by the Image Agnostic Segmentation tool [29], which performs the necessary computations for this module. The output of this module includes the 3D pose and point clouds of each detected object.

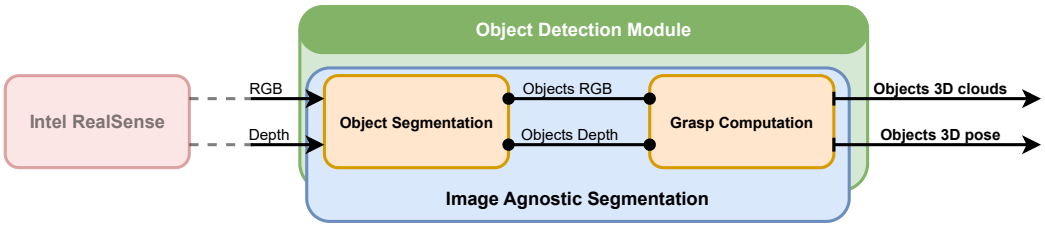


Figure 3.11: Visual representation of the functionality of the Object Detection Module

3.2.3 Cooperation

The Cognitive Layer bridges the Perception and the Manipulation Layers, integrating decision-making abilities based on the knowledge extracted from the physical world. This layer is responsible for determining the most suitable cooperative action to be taken based on the information received from the Perception Layer. In the context of this dissertation, a simple logical approach is used for decision-making, as depicted in Figure 3.12.

In the Cognitive Layer, three main operations can be executed by the manipulator. The first operation is the **Default Pick-and-Place**, which is similar to an autonomous system where there is a predefined set of pick and place positions to be reached. This operation is triggered when there is no human detected or when the human has not shown any cooperative intentions. It serves as a demonstration of the cognitive system's capability to perform a specific task autonomously.

If the human is labeled as *giving something* by the Action Recognition Module, the operation executed is **Take from Human**. In this case, the manipulator maneuvers to retrieve an object from the human's hand and place it on top of a table. On the other hand, if the human is labeled as *asking for something*, the manipulator performs the **Give to Human** operation. In this operation, the manipulator picks an object from the table and places it into the human's hands.

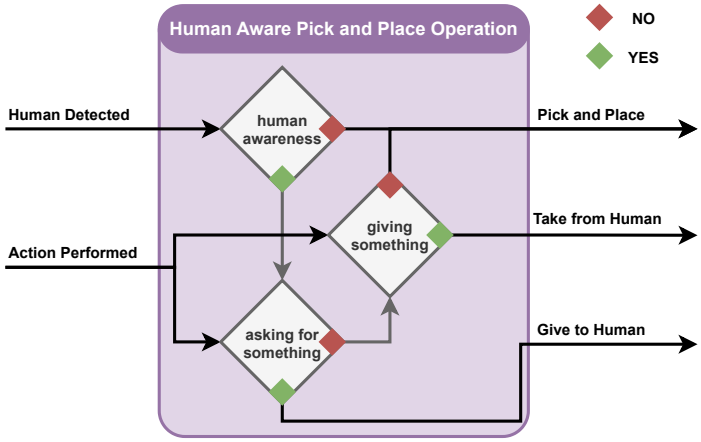


Figure 3.12: Overview of the decision process of the abstract cooperative action when inputted with the perception information

The preceding paragraph discussed the logical pattern of this layer and the reason behind the cooperative operations. Figure 3.13 provides a more detailed understanding of what happens when a specific operation is selected. The Robot Manipulation Module can maneuver the manipulator by defining a target 3D location and rotation pose for the end effector. It plans a collision-free trajectory for the manipulator to execute the desired action.

In the case of the **Default Pick-and-Place** operation, the target poses for picking and placing objects are predefined and do not require any adjustments. However, if the **Take from Human** operation is selected (indicated by the color orange in the figure), the pick area is modified. The pick area becomes a spherical region with its center located in the middle of the human's hands. The diameter of the sphere is set to be large enough to encompass any object that the human may be holding, but small enough to exclude any unsuitable objects. The Cognitive Layer iterates through the array of detected objects in the physical environment and selects the object whose pick pose falls within the defined pick area.

Similarly, for the **Give to Human** operation (represented by the color yellow), the change occurs in the place area. The place area is also positioned in the middle of the human's hands. The Cognitive Layer selects an object from the table and places it into the human's hands using the defined place pose.

Overall, the Cognitive Layer plays a crucial role in coordinating the manipulator's actions based on the extracted knowledge from the Perception Layer, enabling cooperation between the manipulator and humans.

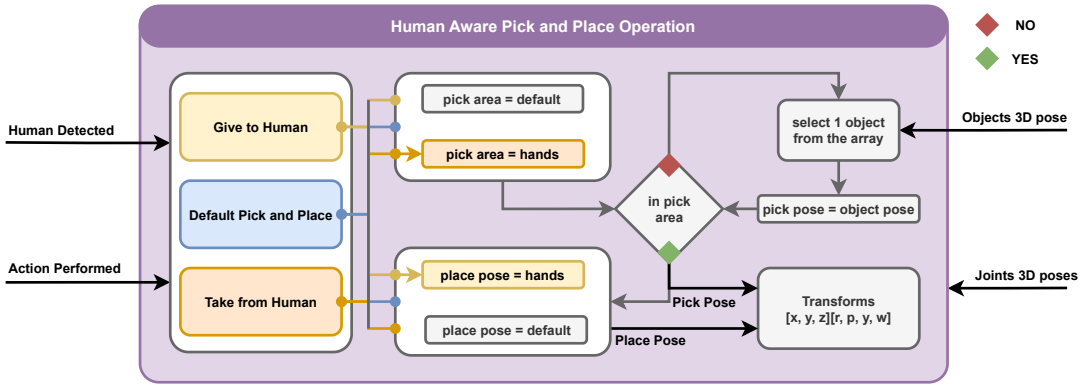


Figure 3.13: Visual representation of the functionality of the Cognitive Layer

3.3 Experimental Setup - An Experimental Approach

3.3.1 Hardware

For the experimental setup, several pieces of equipment were utilized to facilitate the functionality of the architecture and conduct the necessary tests. The chosen equipment includes:

Industrial Manipulator The UR3e manipulator [78] from Universal Robots was selected for the task of manipulation. This industrial manipulator is widely recognized as a market leader in the field of collaborative robots. Its capabilities and versatility make it an ideal choice for this experimental setup (see Figure 3.14).

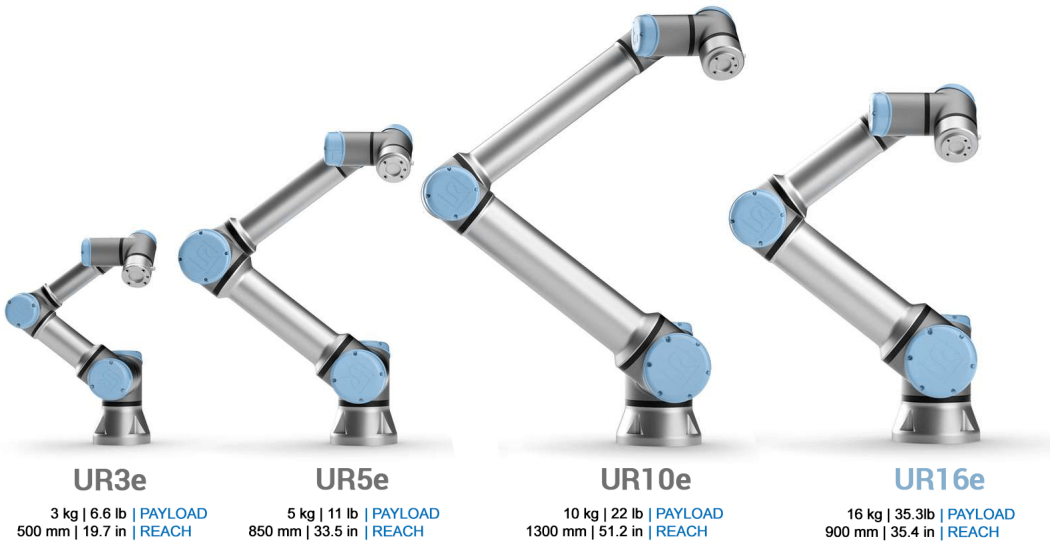


Figure 3.14: Universal Robots cobots models' lineup from [73]

Gripper To enable the manipulator to grasp and manipulate objects, a 140mm two-finger gripper from Robotiq [58] was employed. This gripper has gained popularity in industrial applications and benefits from strong community support. Its design and functionality align well with the requirements of the experiment (see Figure 3.15).



Figure 3.15: Robotiq two finger 140 mm gripper from [58]

Perception Element A crucial component of the architecture is the perception element responsible for capturing and translating the physical environment into the digital world. To fulfill this role, the Intel RealSense D435i camera [34] was chosen. This camera provides both RGB and depth information, offering a comprehensive view of the environment. Its widespread commercial use and strong community support make it an ideal choice for this experiment (see Figure 3.16).



Figure 3.16: Intel RealSense Depth Camera D435i [34]

3.3.2 Software

For the software side of the system, Ubuntu 20.04 LTS (Focal Fossa) [8] was used as the operating system, with a generic kernel version 5.8.0-0.50800 [74] and the Nvidia Driver Version 525.78.01 [52]. The hardware consisted of an Intel® Core™ i7-6700HQ CPU @ 2.60GHz, 8 GB of RAM, and a GeForce GTX 960M GPU.

Two ROS frameworks were employed for efficient development: ROS (Noetic) [61] and ROS 2 (Foxy) [60]. The ROS 1 Bridge package [1] facilitated communication between the two frameworks, enabling seamless data exchange via topics, actions, and services. The use of both frameworks was necessary as certain module tools, such as ROS OpenPose [37] and Image Agnostic Segmentation [29], were designed for the original ROS framework, while MoveIt2 [51] was developed for the ROS 2 framework. It should be noted that OpenCV [53], OpenPose [42], and MMAction2 [12] were not specific to any particular ROS framework.

The drivers used were RealSense ROS [33] and Universal Robots ROS2 Driver [59], both developed for the ROS 2 framework. Most of the code implementation was performed within the Universal Robots ROS2 Driver, simplifying the compilation process.

For the Universal Robot 3e, no additional configuration adjustments were required after installing the necessary dependencies. Regarding the Robotiq gripper it was used the UR-RTDE tool [57] to control seamlessly with the manipulator.

Regarding the Intel RealSense D435i camera, certain configurations were necessary. The point clouds were enabled to make the respective topic available, and the aligned depth feature was activated to ensure that the depth stream was aligned with the color stream. This alignment was crucial for the functionality of the Image Agnostic Segmentation tool. The default output resolution of the camera was set to 620x480 with a frame rate of 30 FPS.

Table 3.3: Summary of the system setup in terms of the software used, the hardware, the frameworks, the tools, and the drivers

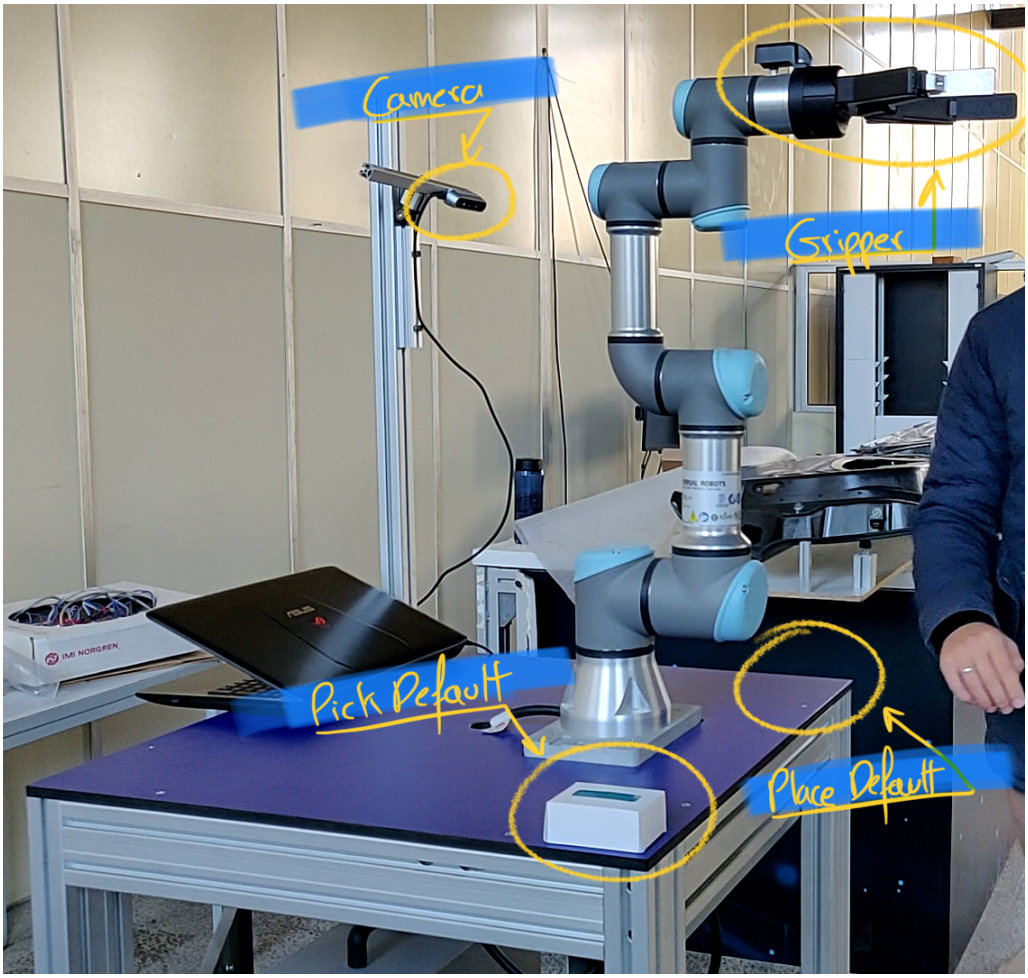
Component	**Specification**
Operating System	Ubuntu 20.04 LTS (Focal Fossa)
Kernel	5.8.0-0.50800-generic
Nvidia Driver	Version 525.78.01
CPU	Intel® Core™ i7-6700HQ CPU @ 2.60GHz
RAM	8 GB
GPU	GeForce GTX 960M
ROS Frameworks	ROS (Noetic) and ROS 2 (Foxy)
ROS 1 Bridge	Enables communication between ROS and ROS 2
OpenCV	Computer vision library
OpenPose	Human pose estimation library
ROS OpenPose	ROS wrapper for OpenPose
Image Agnostic Segmentation	Category-agnostic Segmentation for Robotic Grasping
MMAction2	Action recognition framework
MoveIt2	Manipulation platform for prototyping designs
RealSense ROS	Intel RealSense camera driver for ROS 2
Universal Robots ROS2 Driver	Universal Robots driver for ROS 2
UR Real-Time Data Exchange	A C++ interface for managing data from an UR robot

3.3.3 Experimental Environment

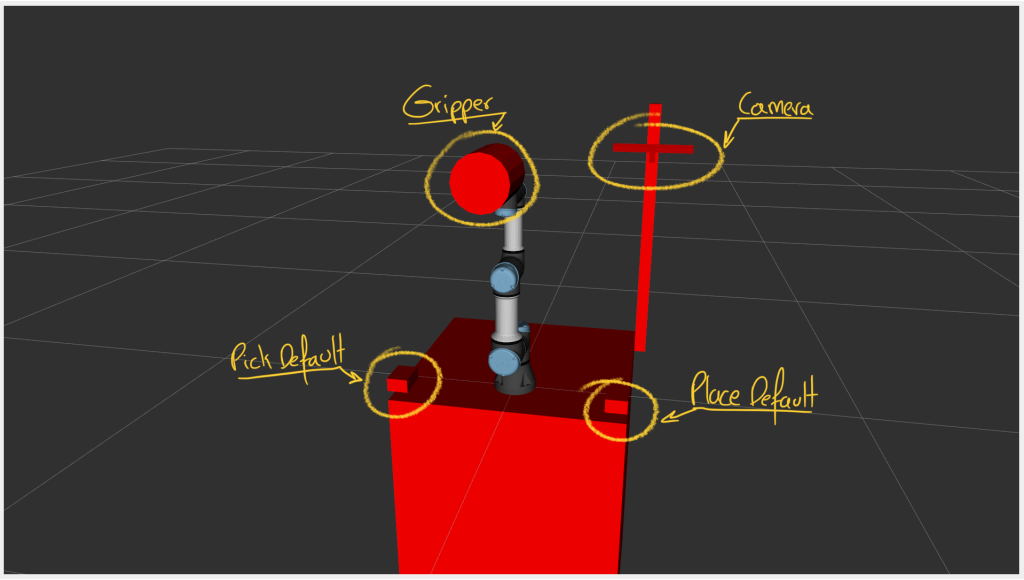
The experimental environment was carefully set up to facilitate the evaluation and testing of the architecture. The configuration of the hardware components within this environment is depicted in Figure 3.17, with specific areas of interest highlighted in yellow.

In the physical-world setup (see Figure 3.17a), the manipulator was positioned slightly away from the center of the table, creating a suitable workspace for the experiment. The Intel RealSense camera was mounted on the rear left corner of the table at an elevation of 67.5 centimeters ($z = +0.675$) and rotated 25 degrees ($roll = -\frac{25}{180}\pi$) to capture both the human subject and the table. The top right corner of the table was designated as the pick area, where objects could be placed for manipulation. Conversely, the opposite top corner served as the place area for object placement.

In the simulator setup (see Figure 3.17b), the manipulator was visualized similarly to its physical-world counterpart. The remaining key elements of the environment, such as the table, mounted camera, pick and place default areas, and the gripper, were manually added to the simulation and highlighted in red. In the absence of an immediate URDF file for the gripper, it was represented in the simulator as a cylinder with appropriate dimensions.



(a) Real-world setup environment. Highlighted in YELLOW are the key areas



(b) Respective simulator setup environment

Figure 3.17: Visualization of the setup used

3.3.4 Experimental Procedure

To comprehensively evaluate the functionality of the architecture and verify the effectiveness of each module, a systematic experimental procedure was designed. The procedure consisted of individual tests for each module followed by integrated tests involving all the modules together. The experimental steps were as follows:

Awareness In this phase, the Action Recognition Module was tested using the PC camera to assess the performance of the trained model. The Human Detection Module was evaluated by walking in front of the camera and occasionally interacting with the manipulator, allowing for the detection of human presence and interaction. The Object Detection Module was tested using various scenarios to capture and identify multiple objects in the environment.

Safety To ensure safe and collision-free operations, the Robot Manipulation Module was subjected to tests. Pre-defined pick and place poses were provided to the module's nodes, which were then used to plan and execute collision-free movements. The cognitive system constantly received human-relevant information to assess potential collisions and ensure the safety of the operation.

Cooperation This phase focused on evaluating the cooperative abilities of the architecture. A human subject performed a set of cooperative intentions in front of the Intel RealSense camera, representing actions such as asking for something or giving something. The cognitive system's response was observed to determine whether the manipulator executed appropriate actions based on the human's intentions.

By following this experimental procedure, the functionality of the architecture was thoroughly tested and evaluated. The individual module tests ensured the proper functioning of each component, while the integrated tests demonstrated the cognitive system's overall ability to perceive, manipulate, and cooperate in a collaborative setting.

3.4 Ethical Considerations

In any research involving human participants and human-robot interaction, ethical considerations are of paramount importance. The experimental setup described in this dissertation also took into account several ethical aspects. These considerations are outlined below:

Safety Precautions The safety of the human participants was given utmost priority. The UR3e cobot was programmed to operate at only 10% of its maximum joint speeds. This reduced speed ensured that any potential collisions would occur at a slower pace, minimizing the risk of harm to the human participants. Additionally, the manipulator

is equipped with effective proprioceptive sensors that can rapidly detect collisions and promptly halt all movements, providing an additional layer of safety.

Informed Consent All experiments involving human participants were conducted with their explicit permission and informed consent. The participants were fully aware of the potential risks associated with the experiment, including the possibility of collision accidents. They were provided with comprehensive information about the nature of the experiment, its objectives, and the potential risks involved. Only after obtaining their informed consent did the experiments proceed.

Participant Well-Being Throughout the experimental procedure, the well-being and comfort of the human participants were prioritized. Measures were taken to ensure that participants felt safe and secure during the experiments. Adequate breaks, as well as provisions for communication and clarification, were provided to address any concerns or questions raised by the participants.

Data Privacy The collection and handling of data related to human participants adhered to strict privacy guidelines. Any personal information or identifiable data obtained during the experiments was treated with confidentiality and stored securely. The use of anonymized data was encouraged whenever possible to protect the privacy of the participants.

Institutional Ethics Guidelines The experimental setup and procedures were conducted following the ethical guidelines and regulations set forth by the relevant institutional review board or ethics committee. The research team ensured compliance with all applicable ethical standards, seeking approval and oversight from the appropriate authorities.

By addressing these ethical considerations, the research aimed to conduct experiments responsibly and ethically, prioritizing the safety, well-being, and privacy of the human participants involved.

3.5 Limitations

The implementation of the hierarchical architecture described in this dissertation encountered several limitations and challenges. These limitations are discussed below:

Availability of Suitable Tools One of the major challenges faced during the implementation of the architecture was the availability of suitable tools designed to work in a ROS (Robot Operating System) environment. While there is a wide range of computer vision studies and algorithms available, finding tools specifically designed to run in a ROS environment proved to be more challenging, especially in the ROS2 environment. The

ROS2 environment was chosen for its advanced features and compatibility with modern systems, but it limited the availability of ready-to-use tools.

Limited Hardware Processing Power The experimental procedure revealed a limitation in terms of hardware processing power. The Action Recognition Module, which required significant computational resources, could not be effectively tested in the ROS environment alongside the other modules. The limited processing capacity of the PC hardware often led to the shutdown of the ROS core when multiple modules were active simultaneously. As a result, experiments dependent on the Action Recognition Module were conducted using "fake" messages that simulated the intended human actions, rather than real-time action recognition.

Simulated Environment The experimental setup included a simulator environment to complement the physical-world setup. While the simulator provided a convenient and controlled testing environment, it had limitations compared to physical-world scenarios. The simulated environment relied on manual additions of key elements, such as the table, camera, and gripper, which may not perfectly replicate physical-world conditions. These limitations could potentially affect the accuracy and realism of the experimental results obtained in the simulator.

Simplified Cooperative Actions The cooperative actions considered in the experimental procedure were simplified and limited to a specific set of actions, namely **Take from Human** and **Give to Human**. While these actions provided a reasonable basis for testing the cooperative behavior of the manipulator, they may not encompass the full range of potential cooperative interactions in physical-world scenarios. The architecture's performance and effectiveness in more complex and diverse cooperative tasks remain to be explored.

Hardware-Specific Limitations The experimental setup utilized specific hardware components, such as the UR3e cobot and the Intel RealSense D435i camera. The limitations and capabilities of these hardware components, such as joint speed and camera resolution, may have influenced the overall performance and outcomes of the architecture. Generalizing the findings and applying the architecture to different hardware setups may require further adaptation and optimization.

These limitations highlight the challenges faced during the implementation and experimental evaluation of the hierarchical architecture. While the architecture shows promise in enabling cooperative human-robot interactions, further research and improvements are necessary to address these limitations and expand its applicability in physical-world scenarios.

4

Experimental Results

This chapter provides a detailed account of the experimental results obtained from each module within the proposed hierarchical architecture. The chapter begins by covering the perception modules in the same sequence as presented in Chapter 3 (Proposed Approach). It then proceeds to showcase the results of the Robot Manipulation Module, along with the decision-making functionality of the Cognitive Layer, as Safety follows Cooperation according to the architecture.

4.1 Awareness

4.1.1 Action Recognition Module

The first results presented are respective to the functionality of the Action Recognition Module, responsible for labeling human intention from RGB data as it is being performed in real-time as it was discussed in the previous chapter. Before introducing the experiments conducted, it should be mentioned that training of the model in the detailed dataset took around 30 minutes, and the best top 1 accuracy was registered at epoch 3 as 100%.

The experiments conducted to test the Action Recognition Module involved capturing live RGB data from a PC camera while a human subject performed cooperative actions. The experiments focused on two cooperative intentions: *asking for something* and *giving something*. The results of the experiments are illustrated in Figures 4.1, 4.2, and 4.3.

In the first experiment (Figure 4.1), the human subject performed the *asking for something* action. Figures 4.1a and 4.1b show the subject with both arms extended and palms turned upwards, while Figures 4.1c and 4.1d show variations in the pose. The figures were analyzed, and the results are summarized in Table 4.1.

Table 4.1: Aggregation of the metrics obtained from the experiment. On the middle column it is shown the maximum guess probability of the top 1 label. On the last column it is presented the top 2 probability variation

Figure 4.1	Maximum Guess Probability	Top 2 Variation Probability
4.1a	66%	44%
4.1b	61%	36%
4.1c	48%	16%
4.1d	53%	23%

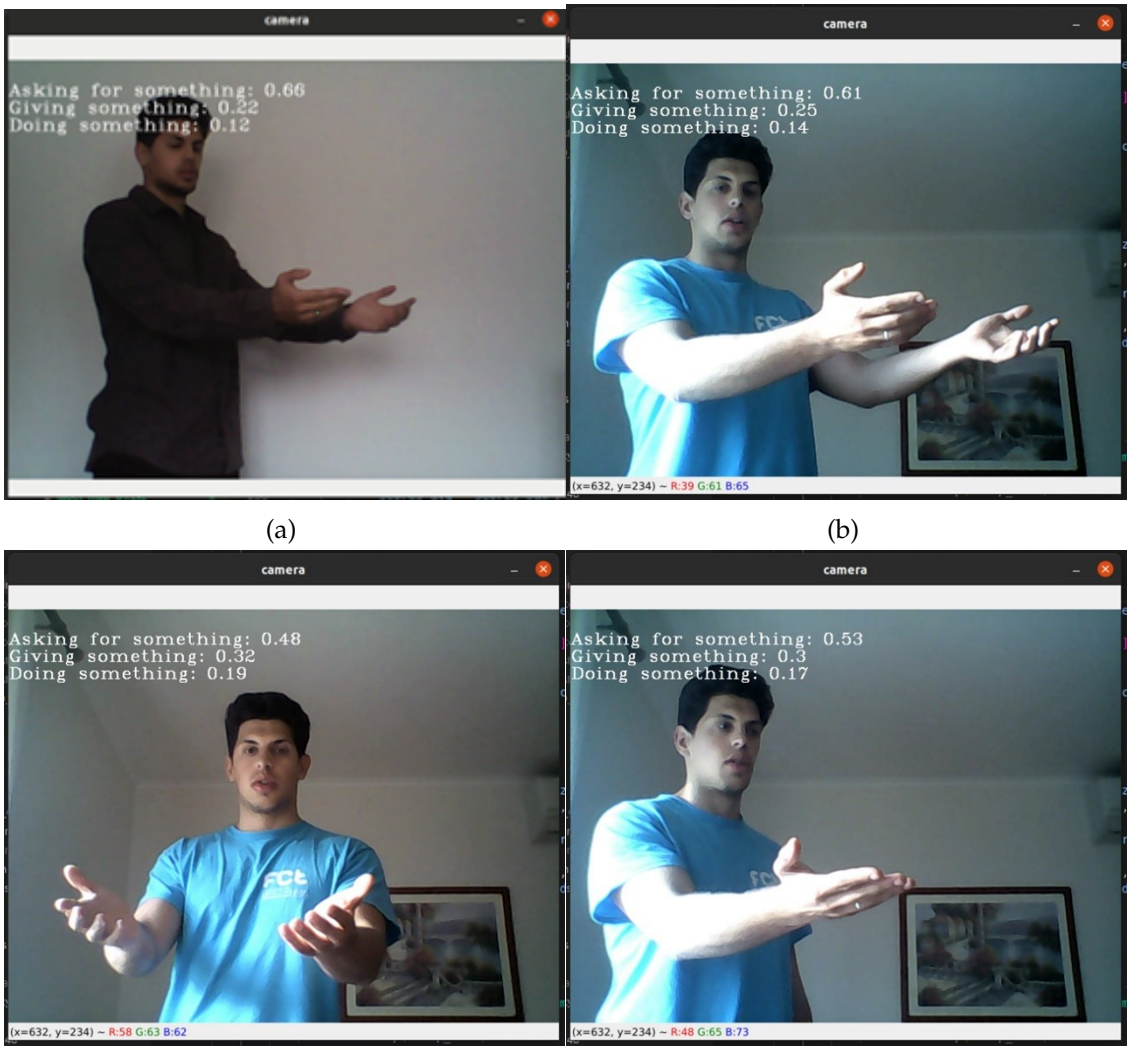


Figure 4.1: Successful recognition of the ***asking for something*** action

The second experiment (Figure 4.2) involved the human subject performing the *giving something* action. Figures 4.2a and 4.2b show the subject with a similar pose to the previous experiment, but this time there is an object held between the hands. Figures 4.2c and 4.2d show variations in how the subject extends the object with one arm. The metrics for this experiment are summarized in Table 4.2.

Table 4.2: Aggregation of the metrics obtained from the experiment. On the middle column it is shown the maximum guess probability of the top 1 label. On the last column it is presented the top 2 probability variation

Figure 4.2	Maximum Guess Probability	Top 2 Variation Probability
4.2a	54%	22%
4.2b	46%	5%
4.2c	47%	9%
4.2d	47%	10%

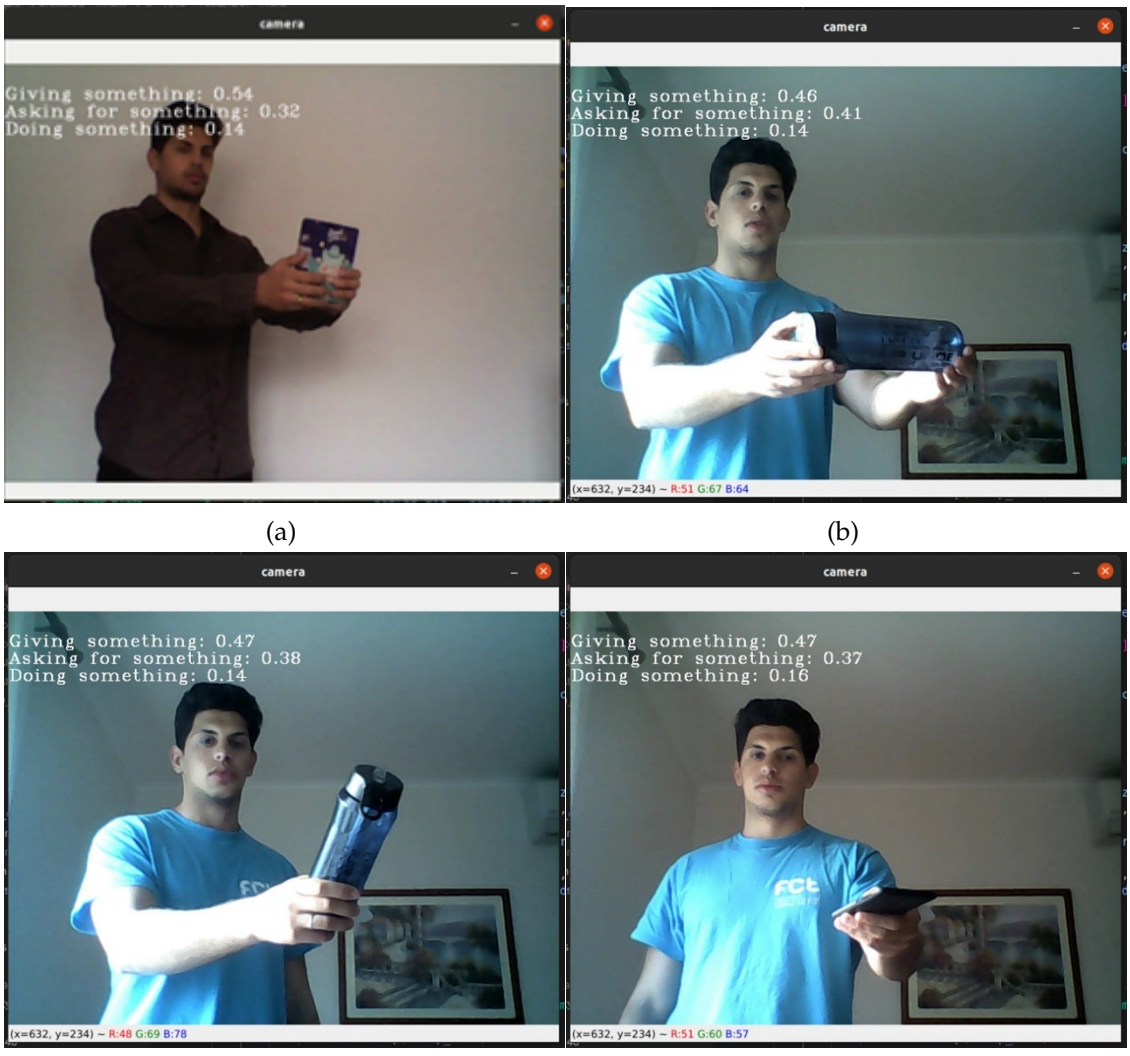
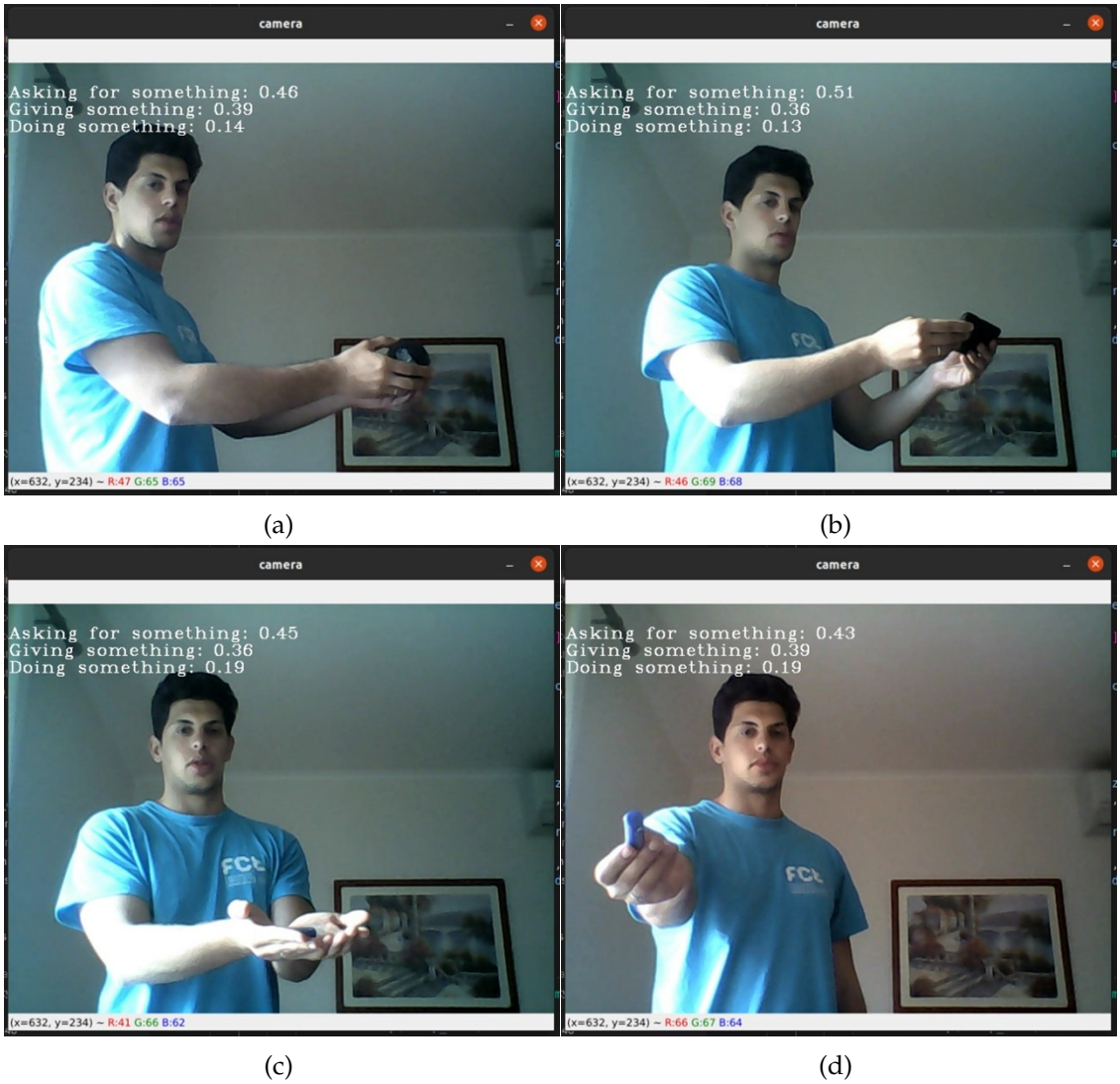


Figure 4.2: Successful recognition of the ***giving something*** action

In the third experiment (Figure 4.3), the human subject once again performed the *giving something* action. Figures 4.3a and 4.3b show the subject with a similar pose, but different objects are held. Figures 4.3c and 4.3d demonstrate variations in presenting a smaller object in different ways. However, in this case, the top-ranked action is incorrectly labeled as *asking for something*, as indicated in Table 4.3.

Table 4.3: Aggregation of the metrics obtained from the experiment. On the middle column it is shown the maximum guess probability of the top 1 label. On the last column it is presented the top 2 probability variation

Figure 4.3	Maximum Guess Probability	Top 2 Variation Probability
4.3a	46%	7%
4.3b	51%	15%
4.3c	45%	9%
4.3d	43%	4%



(a) (b)

(c) (d)

Figure 4.3: Failed recognition of the ***giving something*** action

Overall, these results provide insights into the performance of the Action Recognition Module in correctly identifying human intentions from RGB data. The misclassification in the third experiment highlights a potential limitation or area for improvement in the module's accuracy.

4.1.2 Human Detection Module

Similar to the preceding module, the Human Detection Module was fed with RBG-D data to extract the results.

OpenPose In the previous chapter, it was discussed that this module started by segmenting the human skeleton pose by utilizing the OpenPose tool [42]. Figure 4.4 shows this result in a captured scenario where there is a single human subject. The joints are visualized as dots and the limbs as the connections between the dots. The video from [27] best illustrates the speed, accuracy, and reliability of this tool. It immediately captures any human in the sight of the camera and computes their respective 2D skeleton pose with great accuracy at a frame rate usually lower than 3 FPS.

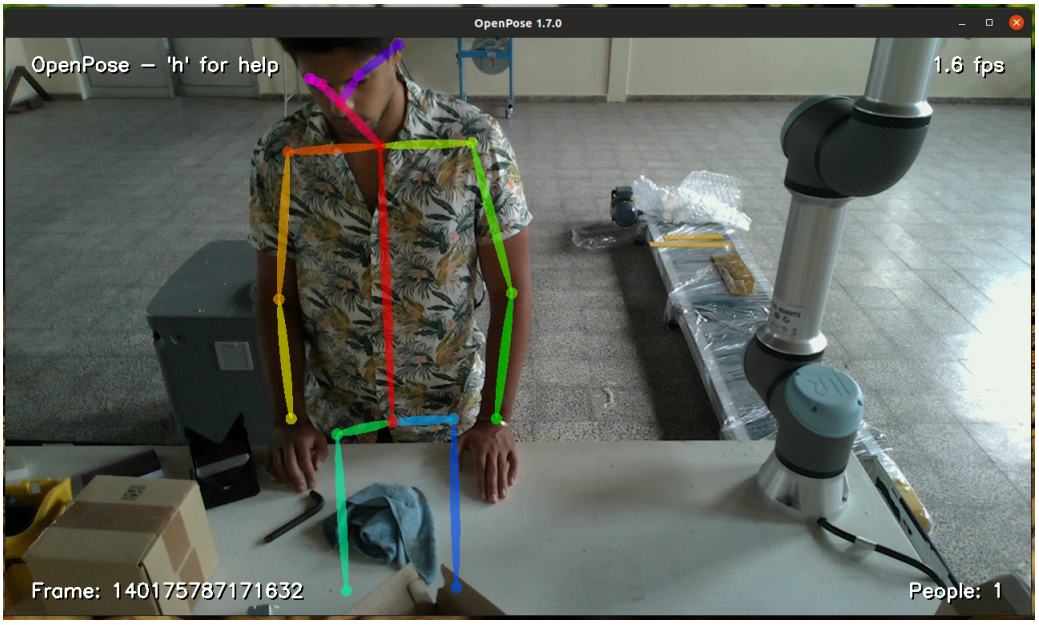


Figure 4.4: Human 2D skeleton being detected by the OpenPose tool [42] in a RGB color scheme where the joints are seen as dots and the limbs as the connections between those dots

ROS OpenPose Next, the human skeleton is translated into the 3D space with the help of the ROS OpenPose tool [37]. This functionality is captured in Figure 4.5. In this figure, on the left side, a human subject's 2D skeleton pose can be observed being captured by the previously mentioned tool, just like in Figure 4.4, and consequently, on the right side, the result of the implementation of the recent tool is represented, which corresponds to the output of the captured 2D skeleton pose in the 3D space.

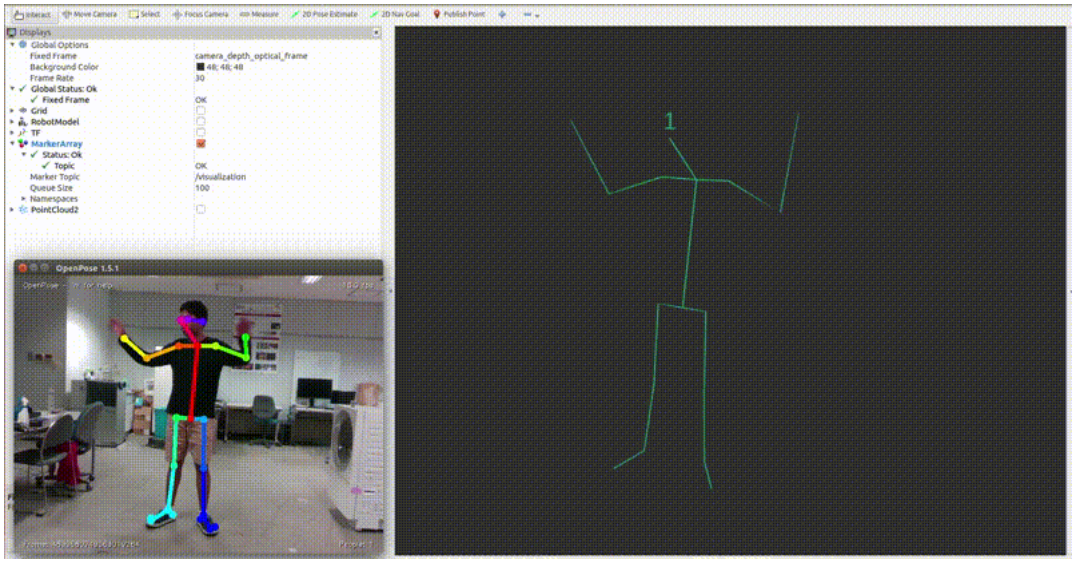
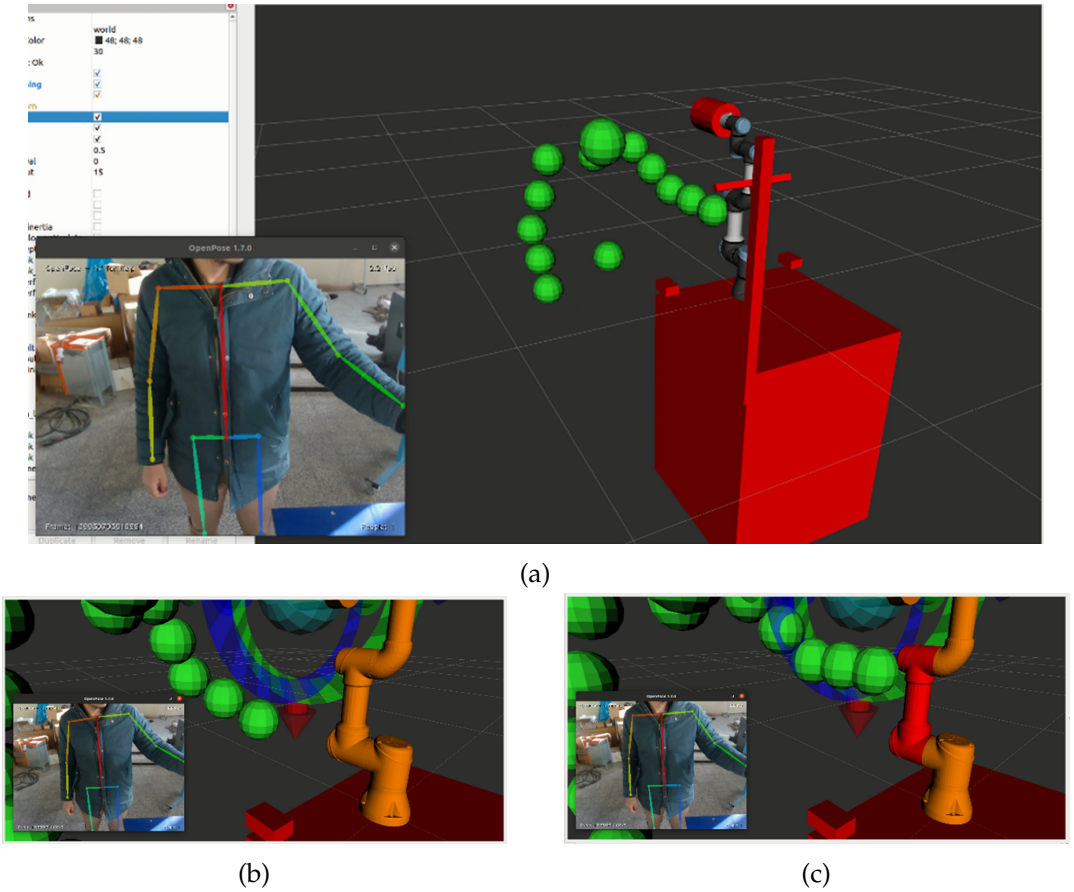


Figure 4.5: Visualization of the tool's [37] performance on the ROS RViz program on the right side as the human subject's 3D skeleton pose in green. On the left side is depicted a human subject being detected by the OpenPose tool [42]

My Pose Finally, the contribution of the previously mentioned tools to this thesis is shown in Figure 4.6. Figure 4.6a demonstrates a human subject touching the manipulator. As illustrated, the joints are represented as green spheres, and an additional sphere is also seen between consecutive joints. The joints visualized are the neck, the shoulder joints, the elbow joints, the wrist joints, the waist, and the head, which is represented as a bigger green sphere.

Consulting the video from [26], it can be seen that as soon as the human subject shows up in the camera's field of view, its 3D pose (the green spheres) is almost immediately outputted to the scenario with good accuracy, and the frame rate can be expected to be around the same lower than 3 FPS.

Figures 4.6b and 4.6c show the cognitive system recognizing when the human is touching the manipulator as a collision event.



(a)

(b) (c)

Figure 4.6: Visualization of the module's performance on the ROS RViz program on the right side as the human subject's 3D skeleton pose represented by green spheres. On the left side is depicted a human subject being detected by the OpenPose tool [42]. 4.6a A general view of the module's functionality; 4.6b The human subject hand approaching the manipulator; 4.6c The human hand touching the manipulator and the collision being detected in the shoulder link with the color red

4.1.3 Object Detection Module

Image Agnostic Segmentation The Object Detection Module is described as responsible for detecting objects from RGB-D data and computing their respective pose in 3D space. The Image Segmentation Agnostic tool [29] is responsible for the full operation of this module. To evaluate its functionality an initial experiment where there are different objects on top of a table was set up and the results from the mentioned tool are represented in Figure 4.7. On the top left corner of the image, the RGB segmentation of the estimated objects in the environment can be visualized. Below, in the bottom left corner, are their respective computed grasp approaches. On the right side, the computed 3D pose of the objects is shown alongside the environment depth data. Although not shown in the figure, this tool also takes in the RGB segmented objects and predicts their respective point clouds from the depth data.

By consulting the video from [28], the tool's accuracy, speed, and reliability can be

better analyzed. It can be estimated that the segmentation and 3D pose computation process takes roughly 10 seconds.

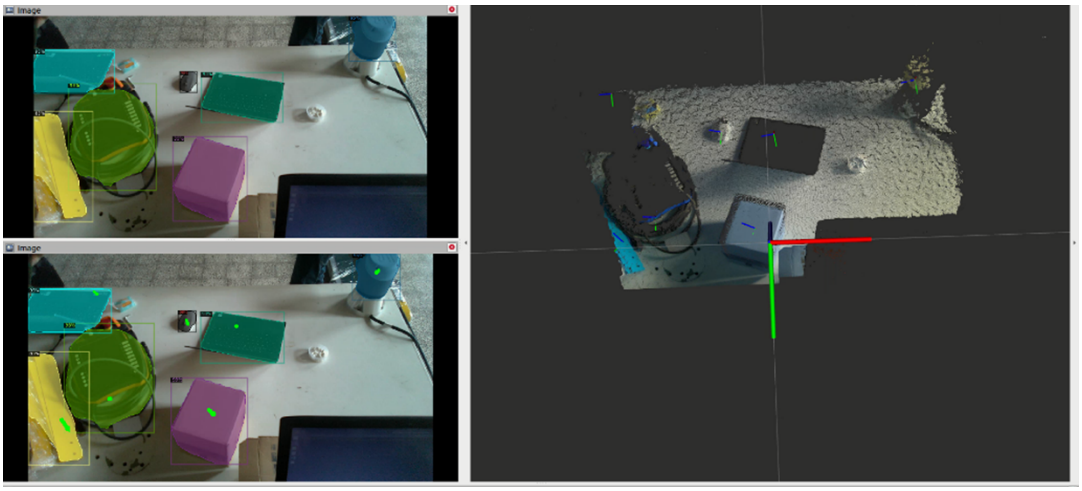
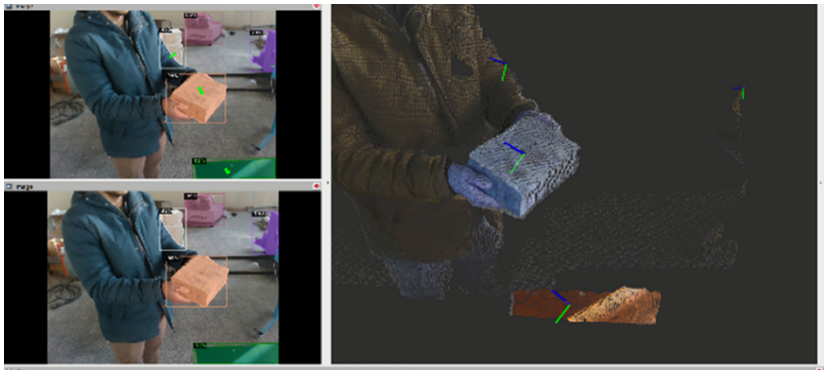
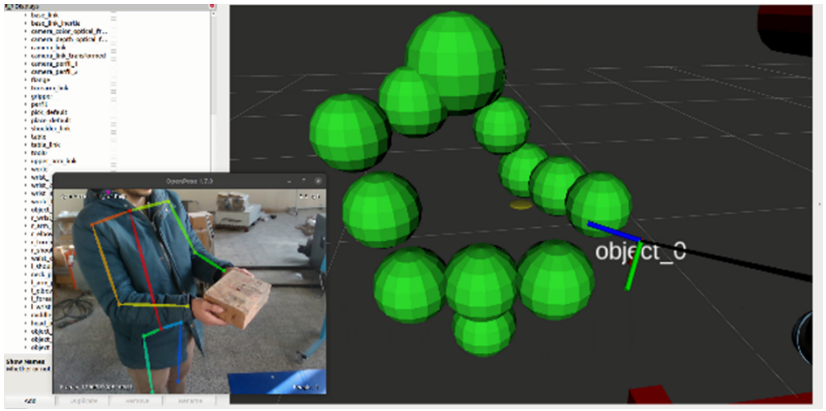


Figure 4.7: Visualization of the tool's [29] performance on the ROS RViz program. On the left top corner can be visualized the RGB segmentation of the objects detected in the environment. Below, on the left bottom corner is visualized their respective computed grasp approaches in green. On the right side is shown the computed objects 3D pose alongside the environment depth data

My Pose + Image Agnostic Segmentation These last two modules are responsible for the accuracy of the manipulator's positioning and collision avoidance in the physical world. For this reason, an experiment where a human subject is holding an object between his hands was conducted where only the results prominent from the two modules are presented and it is depicted in Figure 4.8. Figure 4.8a shows the currently mentioned module segmenting the object between the human's hands and its computed 3D pose. To complement, Figure 4.8b shows the computed 3D pose of the human, represented as green spheres, alongside the respective 3D pose of the object that the human is holding, classified as *object_0*.



(a)



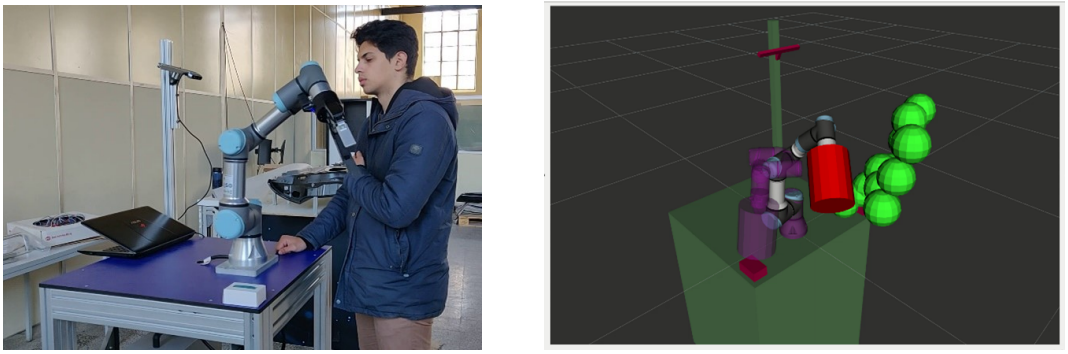
(b)

Figure 4.8: Visualization of the performance of the 4.8a Object Detection Module and the 4.8b Human Detection Module

4.2 Cooperation and Safety

As mentioned before, this layer is considered to be the brain of the architecture since it aggregates decision-making abilities to build cooperative manipulation goals dependent on the knowledge extracted from the perception modules, (i.e., human awareness). To evaluate such abilities, i.e., the decision-making visualized in Figure 3.12 of Chapter 3 (Proposed Approach) three experiments were conducted involving a human subject. First, he starts by operating in the work environment of the manipulator with no intention of cooperation. Next, he is seen *asking for something* and in contrast, in the final experiment, he is seen *giving something*. The base operation of this architecture is a simple pick-and-place operation, and each experiment will showcase this layer's decision accuracy and the feasibility of the triggered cooperative operation which is the entire responsibility of the Manipulation Layer. This feasibility will depend on the accurate placement of the pick and place poses and the maneuver of the manipulator safely following the logic from Figure 3.13 in Chapter 3 (Proposed Approach). For this reason, the results from the latter are presented alongside the decision that triggered them.

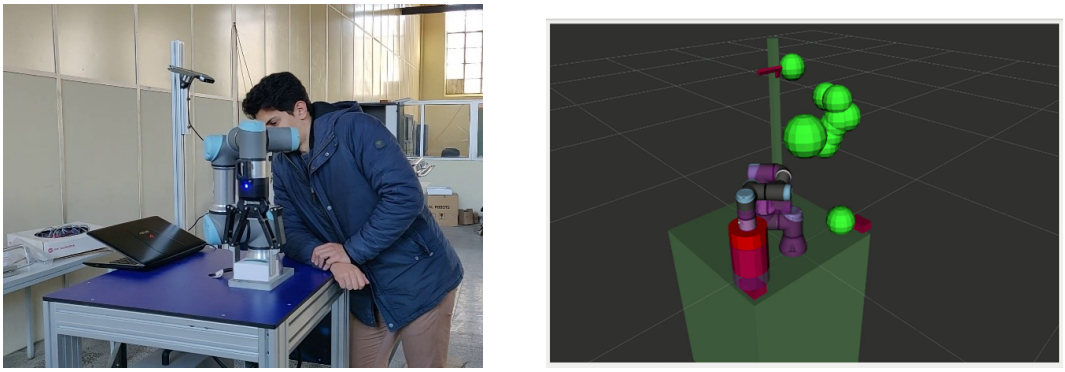
Default Pick and Place The first experiment is depicted in Figures 4.9 through 4.12 and also in the video from [24]. Here the cognitive layer was triggered to perform the Default Pick and Place operation with pre-defined pick and place poses. Bearing in mind that the manipulator is facing the camera backward, it is seen a human subject positioning himself to the left of the manipulator, as shown in Figure 4.9a. Here, the manipulator is seen performing the pick operation with a trajectory that doesn't collide with the human. Figure 4.9a shows this action in the simulator where the human is represented as green spheres, the collision elements shown as green boxes, and the planned trajectory the manipulator is executing as translucid purple. It also shows that the head of the human was not yet captured.



(a) (b)

Figure 4.9: Manipulator performing a typical pick and place operation while trying to avoid contact with the human subject. 4.9a The human subject is at the left of the manipulator while it performs a pick operation with a collision free trajectory. 4.9b The mentioned pick operation is being executed and visualized through the simulator. The human's 3D pose is shown as the green spheres

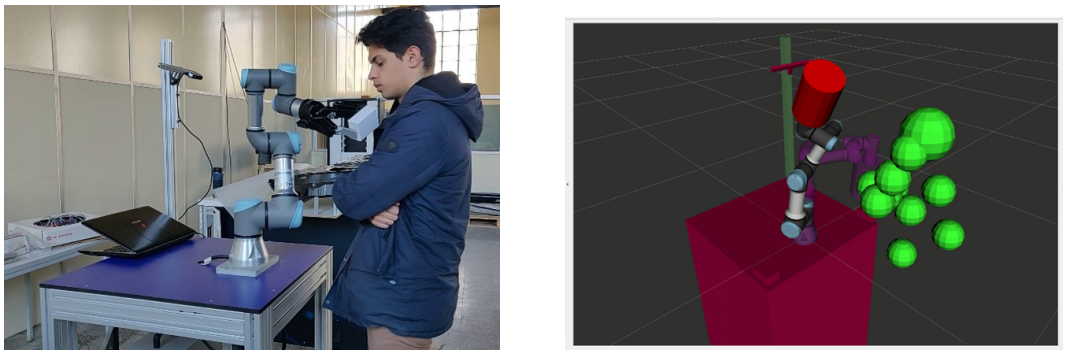
Next, in Figure 4.10a, the human is positioned more to the left while the manipulator is performing the pick operation. Here, the cognitive system finishes the current operation, but the manipulator halts and doesn't move anymore. The respective action is represented in the simulator through Figure 4.10b, where a safe trajectory for the manipulator to move isn't planned, and the human's 3D pose is not correctly captured.



(a) (b)

Figure 4.10: Manipulator performing a typical pick and place operation while trying to avoid contact with the human subject. 4.10a The human is positioned more to the left of the manipulator while it is seen halted for some time. 4.10b This action is displayed in the simulator where the human pose is incorrectly inputted

After that, in Figure 4.11a, the human is at some distance from the manipulator. After a few seconds, the manipulator starts performing the place operation with a trajectory very close to the human, and it finishes the operation by placing the object in the place position. Figure 4.11b shows that the cognitive system can plan a trajectory to the place position (shown as translucid purple), and the manipulator is seen performing it. Analyzing the planned trajectory, it can be seen that it collides with the human pose. Again, the human's 3D pose is not correctly captured.

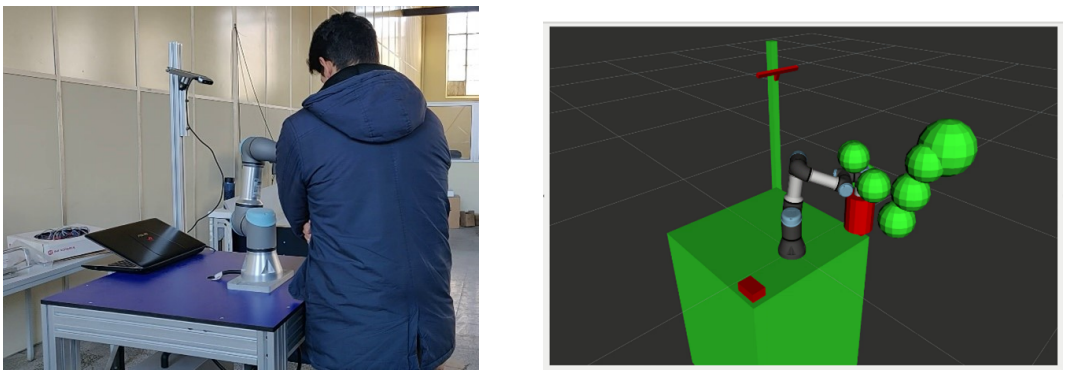


(a) (b)

Figure 4.11: Manipulator performing a typical pick and place operation while trying to avoid contact with the human subject. 4.11a The manipulator is performing a trajectory to the place position that is in close collision with the human. 4.11b The simulator shows the planned trajectory as translucid purple and the humans pose

Then, Figure 4.12a shows the human positioning himself at the right of the manipulator. Here, it can be seen that the latter is not performing any movement. On the simulator side, the cognitive system doesn't capture the human pose well (green spheres), and it isn't able to perform a safe trajectory to the final position, as shown in Figure 4.12b. To conclude the operation, the human steps down, and the cognitive system performs a safe

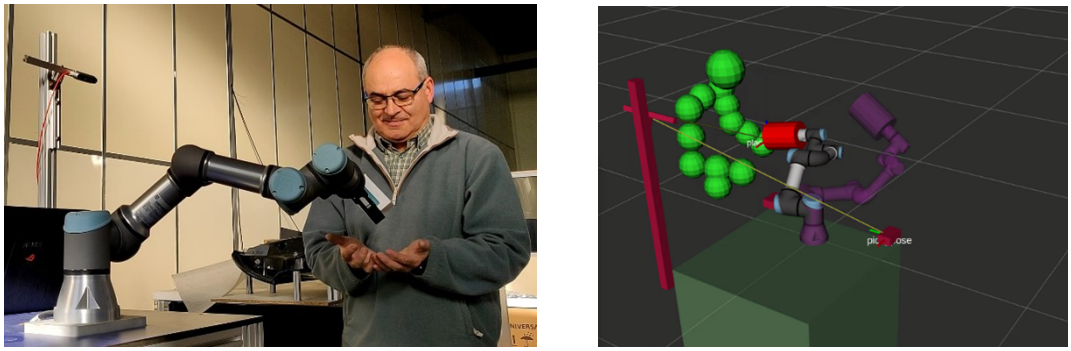
trajectory to the final position, but the manipulator doesn't move.



(a) (b)

Figure 4.12: Manipulator performing a typical pick and place operation while trying to avoid contact with the human subject. 4.12a The human positions himself at the right of the manipulator while it finishes the place operation. 4.12b This action is shown in the simulator

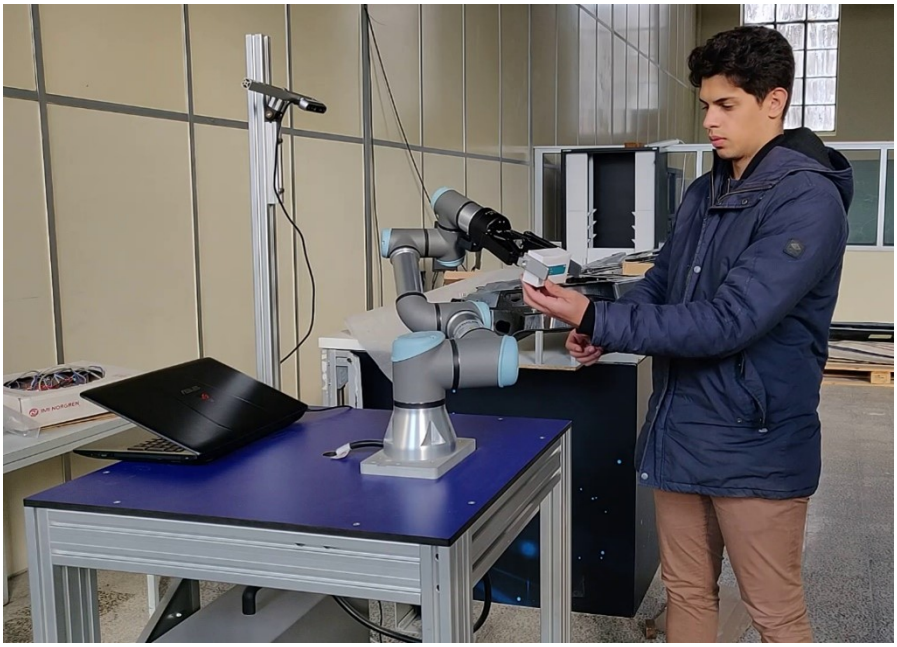
Give to Human The following experiment sees a human subject *asking for something* from the manipulator. This intention triggers the Cognitive Layer to perform the cooperative operation Give to Human as demonstrated in Figure 4.13 and the video from [23]. It illustrates a human (Mr. Manuel) in an *asking for something* pose. Based on this pose, the cognitive system triggers the respective cooperative action Give to Human. The manipulator is seen performing awkward trajectories through the action. It starts by picking up the object in the pick-up area. Then, it moves in a complex manner to reach the designated position. The human moves to what he thought was the drop position, but the manipulator then moves to the final place position where the human initially was, as illustrated in Figure 4.13a. The manipulator finishes the movement by opening the gripper and placing the object in the human's hand. Figure 4.13b illustrates the human's pose as green spheres, the manipulator reaching the target place position, and the planned trajectory as a translucent purple to reach the same position in the simulation environment.



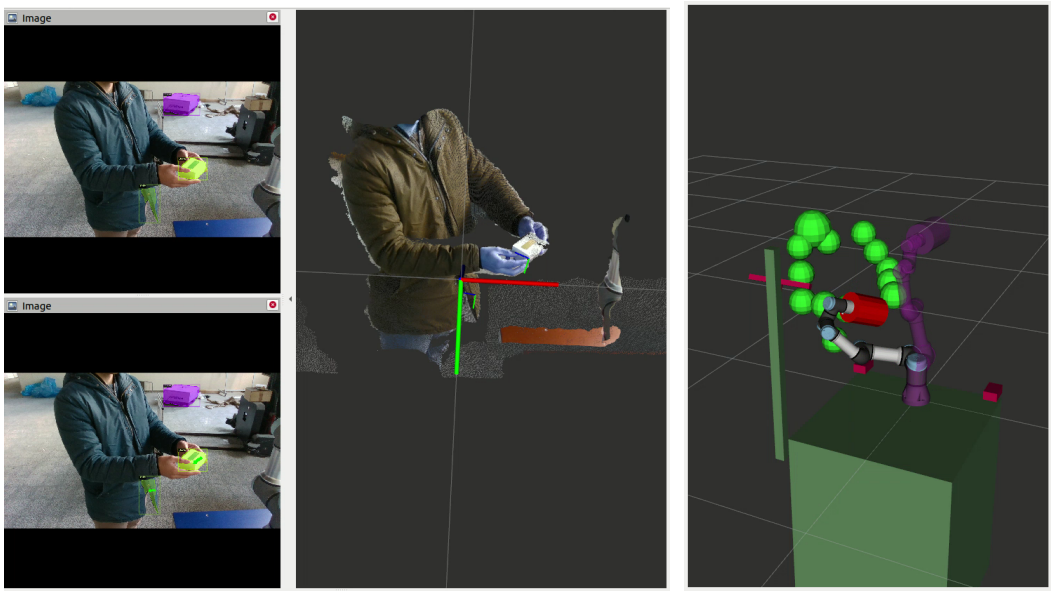
(a) (b)

Figure 4.13: Manipulator performing the ***Give to Human*** operation. 4.13a The manipulator is placing an object in the middle of the human subject hands (Mr. Manuel). 4.13b The simulator shows the human subjects pose as green spheres, the planned trajectory as translucid purple and the place pose between the human's hands

Take from Human Alternatively, a final experiment where a human subject is seen *giving something* to the manipulator is conducted, where it is triggered from the Cognitive Layer a response to perform the cooperative operation Take from Human as shown in Figure 4.14 and the video from [25]. The first observed instance in this action is the human subject in a *giving something* pose, which triggers the cognitive system to perform the respective cooperative action Take from Human. In Figure 4.14b, the object is segmented, and its respective 3D pose is computed. The cognitive system planned a safe trajectory to a pose slightly above the object that the human is holding, as shown as a translucent purple in Figure 4.14c. In this figure, the human's 3D pose is also observed as green spheres on the left side of the manipulator if we consider the latter is facing the camera backward. Once a planned trajectory is computed, the manipulator moves to the target pose where the object is. The human places the object between the manipulator's two-finger gripper while it closes the gripper and holds the object, as seen in Figure 4.14a. The manipulator did not finish the cooperative action by placing the object in the place area, but the simulator planned a trajectory to execute this task.



(a)



(b) (c)

Figure 4.14: Manipulator performing the ***Take from Human*** operation. 4.14a The manipulator halts at the pick pose while the human places the object between the gripper while it grasps it. 4.14b The Object Detection Module shows the computation of the objects' pose. It also shows other undesired objects captured. 4.14c The simulator shows the humans pose as green spheres, the manipulator at the pick pose and the planned trajectory as translucid purple

5 Discussion

In the previous chapter, the results obtained from several experiments regarding each stage of the architecture were presented. To highlight, it was observed that all the perception modules allow for a consistent extraction of accurate knowledge, some faster than others. The Cognitive Layer correctly infers the knowledge extracted, which outputs a suitable cooperative manipulation goal for the Robot Manipulation Module to attain. However, the Robot Manipulation Module is seen as lacking in dynamic responses, as changes in the environment are not reacted to once an operation has started. To better understand all the detailed results, a thorough discussion will be given below from each, following the order they were presented in the preceding chapter.

5.1 Action Recognition Module

Analysis The Action Recognition Module was initially mentioned to achieve a maximum classification accuracy of 100% at epoch 3 during training, which suggested overfitting. However, the predictions contradict this result, making it difficult to understand the underlying issue.

Figures 4.1, 4.2, and 4.3 display three experiments where a human subject performs the same action in four different ways to evaluate the model's performance. The first two variations closely resemble the training data, while the remaining two deviate slightly. Successful classifications are observed for the first two actions, while the last action presents incorrect predictions.

In the first experiment (see Figure 4.1), the *asking for something* action is accurately classified with high probability and top-2 variation. The model shows certainty in distinguishing this action from its opposite. Similar behavior is observed in the second experiment (see Figure 4.2) for the *giving something* action, where higher probabilities are assigned to actions resembling the training data. However, the top-2 variation indicates lower confidence in correctly classifying this action, with probability values approaching those of the opposite action. These observations lead to incorrect predictions for the *giving something* action in the third experiment (see Figure 4.3). These mispredictions occur

due to the presence of noise, such as objects in the background and different lighting conditions, which the model is not accustomed to. It is concluded that the *giving something* action is more accurately predicted when the object is clearly visible, while smaller or occluded objects lead to inconsistent predictions.

Notably, the *doing something* action does not show higher guess probabilities as no data with this action was included in the dataset. The MMAction2 tool proved valuable in achieving action recognition for this thesis, offering fast training and testing capabilities and access to video understanding APIs. However, it requires a significant learning curve, although ample documentation is available.

While the model demonstrates consistency in classifying cooperative actions when the human is present, it does have limitations.

Limitations One major limitation identified was the overfitting issue, where the model showed close prediction probabilities in some instances.

Future Proofing To address this issue, two potential actions can be taken. First, the dataset can be expanded with more diverse data. Alternatively, since the two similar actions (*asking for something* and *giving something*) may contribute to the overfitting, a more distinct *giving something* action could be introduced.

Summary To summarize, the Action Recognition Module exhibited consistent and accurate classification of the intended actions when the human subject was present. However, it showed limitations, particularly in cases of overfitting and challenges with object visibility. These limitations can be addressed by improving the dataset and training parameters, following established machine learning techniques.

Overall, the module's performance was promising, and with appropriate adjustments, it has the potential to enhance the architecture's decision-making capabilities.

5.2 Human Detection Module

Analysis Moving on to the discussion of the Human Detection Module, the analysis revealed important findings regarding the tools and techniques used. The OpenPose tool was found to efficiently compute a 2D skeleton pose, and the ROS OpenPose tool seamlessly translated this 2D information into the 3D space without sacrificing speed. The 3D information obtained from the ROS OpenPose tool consisted of joint locations such as the neck, shoulders, elbows, wrists, waist, and head joints, which were used as input to build collision objects.

The collision objects, represented as green spheres with a constant diameter, were utilized to ensure safe planning and avoid close proximity between the manipulator and humans. To reduce the number of collision elements, the lower body joints were excluded. Additionally, to simulate limbs, an extra sphere was added between consecutive joints.

The application of collision objects was demonstrated in Figures 4.6b and 4.6c, where the green sphere representing the left wrist touched the manipulator's shoulder link, resulting in a change in color from orange to red. This color change indicated a collision event between the manipulator and the human subject, triggering the cognitive system to restrict the manipulator's movement.

The development of both tools used in this module showed promise in achieving the thesis goal by providing reliable and accurate information for collision detection. The refresh rate of the human pose as spheres remained unaffected, maintaining the same speed as the previous tool.

Limitations However, there are limitations to consider. The tools are susceptible to occlusion, where body parts may be obstructed by objects between the camera and the limb or even by other limbs. Although the OpenPose tool provides estimates in such cases, it is not foolproof, and occlusion can lead to incomplete and misplaced measurements in the 3D space. This, in turn, may trigger false collisions with the manipulator or allow for the planning and execution of false safe trajectories.

Another limitation is the dependency on the hardware's quality. The hardware system used an outdated GPU, resulting in a lower computation rate of the skeleton pose by the OpenPose tool, averaging less than 3 FPS. This significantly impacted the cognitive system's overall speed, making it slow.

Summary In summary, despite the limitations mentioned, the tools utilized in the Human Detection Module allowed for reliable and accurate measurement of the skeleton. However, the speed of the cognitive system was compromised due to hardware limitations.

5.3 Object Detection Module

Analysis Now let's move on to discussing the Object Detection Module. The analysis reveals that the module successfully computes the grasp poses of objects, consistently providing accurate results. This can be observed in Figure B or the referenced video, where the module outputs grasp poses in the middle of the human skeleton hands, replicating physical-world scenarios.

Limitations However, the module does have limitations. One significant drawback is the computation time required for determining the object's 3D pose, which takes approximately 10 seconds. This delay makes the entire cognitive system slow to respond to changes in the environment, reducing its dynamic capabilities. Additionally, objects lacking salient edges or situated at a long distance from the camera are often only partially computed, leading to incomplete results. Moreover, when humans are present in the scene, the module may mistakenly detect their body parts as objects, further diminishing

its reliability in open space environments. This can potentially result in false grasp poses for the manipulator when performing the **Take from Human** cooperative operation.

Future Proofing To address these issues, it would be necessary to operate the module in a controlled environment where variables such as easily detectable objects and minimal movement can be regulated. However, such controlled conditions may not be suitable for detecting objects between human hands, which is the focus of this dissertation.

Summary In summary, the Object Detection Module provides somewhat reliable computation but suffers from slow speed. It performs best in less dynamic environments with close proximity. While it may have limitations, it is still sufficient to achieve the goal of this dissertation.

5.4 Cognitive Layer and Robot Manipulation Module

Analysis Moving on to the discussion of the Cognitive Layer, its performance will be evaluated by examining the experiments conducted. In the first experiment, where the manipulator performs the **Default Pick and Place** operation (see Figures 4.9 through 4.12), it is observed that the manipulator does not move after completing the pick operation due to the human obstructing the target pose (represented by the green spheres). This triggers the Robot Manipulation Module to replan a new trajectory indefinitely (as represented in Figure 3.4). Irregularities in the human pose can also be observed, which is a consequence of the occlusion problem discussed in the Human Detection Module. Once the human steps back, the manipulator performs a trajectory that nearly collides with the human. This issue arises from the planning algorithm, which reacts immediately without considering environmental dynamics, such as a human subject approaching the planned trajectory.

In the next experiment (see Figure 4.13), during the **Give to Human** operation, the manipulator exhibits awkward trajectories. This is due to the limitations of the geometric planners used by the OMPL algorithm [reference]. These planners only account for geometric and kinematic constraints, resulting in planned trajectories that disregard execution time. Adjusting parameters may partially alleviate this problem, but a more suitable algorithm is required.

In the last experiment (see Figure 4.14), the manipulator fails to place the object in the intended location during the **Take from Human** operation, despite a planned trajectory being present in the simulator. This failure is attributed to the requirement of the MoveIt 2 tool to run in a real-time kernel for efficient communication between the PC hardware and the manipulator's hardware. Since this requirement conflicted with other tools needed for the remaining modules, a generic kernel was used instead, compromising the reliability of the manipulator's maneuverability and leading to communication errors (timeout error) due to resource distribution.

Limitations It's important to note that low-performance hardware prevented the use of the Action Recognition Module in conjunction with the Cognitive Layer. As a result, the experimental procedures presented for the Cognitive Layer demonstrate commands sent to the architecture to simulate the automatic detection of human intention.

Future Proofing To improve the cognitive system, additional knowledge could be incorporated, such as using other models and actions to constantly update the cognitive system. For example, knowing whether the human is moving or still would be useful for the cognitive system to execute plans only when the human is still and pause execution when the human starts moving. Moreover, logical decision-making could be replaced with more suitable or optimized strategies, such as those derived from artificial intelligence. Additionally, the Robot Manipulation Module could benefit from a more adaptable algorithm that considers environmental dynamics and implements human-aware grasp approaches.

Summary In summary, the Cognitive Layer effectively coordinates information from the perception modules and plans cooperative or non-cooperative operations based on the intended intention. However, the Robot Manipulation Module has its limitations, as it plans based on individual instances without considering the continuity of the environment or time-optimized trajectories.

6

Conclusion

In conclusion, this dissertation aimed to investigate the impact of the proposed hierarchical architecture to address the problem of human-aware cooperation in industrial manipulators.

The analysis of existing literature revealed a significant gap in research about the application of a framework in the context of human-aware cooperation in industrial manipulators. This study fills that gap by providing empirical evidence and insights into the effectiveness of a hierarchical architecture in reducing the barrier between human workers and industrial manipulators and optimizing cooperation performance.

The application of the proposed hierarchical architecture revealed that the Cooperation Phase or Cognitive Layer efficiently coordinates a suitable cooperative response based on the knowledge extracted from the Awareness Phase. The latter is divided into three modules each designed to address the problem at hand. The main one, the Action Recognition Module provides accurate classifications of real-time human actions revealing a certain intention of cooperation. Important for human shape and position awareness is the Human Detection Module which computes collision spheres from the joint locations for collision-free trajectories to be performed. Also, the Object Detection Module provides reliable location poses of the objects in the physical world. On the other hand, the Robot Manipulation Module in the Safety Phase demonstrated it could maneuver the manipulator to attain accurately the target poses and perform the cooperative operations.

Furthermore, the study revealed that the applied hierarchical architecture can attain a cooperative environment between humans and industrial manipulators while also keeping safety in mind by performing collision-free trajectories. These insights can guide practitioners in optimizing the implementation of this architecture and maximizing its benefits.

The adoption of the presented hierarchical architecture in the industrial world can lead to effective decisions on cooperation goals and consequently the safe maneuverability of industrial manipulators. By implementing the findings of this study, industries can start moving away from area-specific manipulators and incorporating them in the same environment as human workers leading to fewer expenses and a more efficient work ethic.

However, it is important to acknowledge certain limitations of this study. Starting with the manipulation portion of the architecture, the Robot Manipulation Module, it was seen that it is very reactive while trajectory planning not accounting for the continuity of the environment nor performing the fastest trajectories. Also, the Move Group tool of this module requirements makes its respective operation unreliable alongside the quality of the hardware by occasionally occurring timeout errors. These limitations are accentuated by the processing power of the PC being distributed through the different tools implemented. The hardware quality is felt through the Awareness Phase and mainly in the Action Recognition Module since its use was compromised. The remaining Human and Object Detection Modules showed slower computation times which made the entire architecture less dynamic in generating promptly cooperative responses.

To improve upon the work presented the main contribution should be to introduce a more time-efficient algorithm with real-time collision checking to replan a new trajectory and avoid collisions even after the manipulator has started movement. In addition, better action recognition performance could be achieved by training the model more efficiently. In the Awareness Phase, introducing more knowledge could provide for a more diverse range of cooperative responses. Furthermore, the simple logic making-decision from the Cognitive Layer could be replaced completely by machine learning techniques.

In conclusion, this research provides empirical evidence supporting the effectiveness of the hierarchical architecture presented in allowing and improving human-aware cooperation in industrial manipulators. The findings underscore the importance of integrating innovative technologies to address the big three challenges of Awareness, Safety, and Cooperation in a shared environment between humans and manipulators and promote an efficient work ethic. The insights gained from this study contribute to the growing body of knowledge in the field of human-robot cooperation and provide a foundation for further research and practical implementation.

Overall, this study serves as a valuable resource for engineers and industry professionals seeking to reduce the barrier between industry workers and manipulators and enhance a more productive workflow. By harnessing the potential of the hierarchical architecture and implementing the recommendations outlined in this research, industries can play a vital role in achieving a more work-efficient and cooperative future between man and machine.

Bibliography

[1] R. 2. *ros1_bridgeonFoxy*. [Accessed: March 27, 2023]. 2021. URL: https://github.com/ros2/ros1_bridge/tree/foxy (cit. on p. 43).

[2] J. K. Aggarwal and Q. Cai. "Human motion analysis: A review". In: *Computer vision and image understanding* 73.3 (1999), pp. 428–440 (cit. on p. 5).

[3] J. K. Aggarwal and M. S. Ryoo. "Human activity analysis: A review". In: *Acm Computing Surveys (Csur)* 43.3 (2011), pp. 1–43 (cit. on p. 5).

[4] A. Alahi et al. "Social lstm: Human trajectory prediction in crowded spaces". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 961–971 (cit. on pp. 25–27).

[5] M. Baccouche et al. "Sequential deep learning for human action recognition". In: *International workshop on human behavior understanding*. Springer. 2011, pp. 29–39 (cit. on pp. 3, 19–22).

[6] A. F. Bobick and J. W. Davis. "The recognition of human movement using temporal templates". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.3 (2001), pp. 257–267 (cit. on pp. 2, 7, 18).

[7] Y. Cai et al. "Action knowledge transfer for action prediction with partial videos". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 8118–8125 (cit. on pp. 23, 27).

[8] Canonical. *Download Ubuntu Desktop*. [Accessed: March 27, 2023]. URL: https://ubuntu.com/download/desktop (cit. on p. 43).

[9] Z. Cao et al. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields". In: *IEEE transactions on pattern analysis and machine intelligence* 43.1 (2019), pp. 172–186 (cit. on pp. 9, 10, 18, 37).

[10] Z. Cao et al. "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 (cit. on pp. 2, 9, 18).

[11] J. Carreira and A. Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6299–6308 (cit. on pp. 3, 19, 22).

[12] M. Contributors. *OpenMMLab's Next Generation Video Understanding Toolbox and Benchmark*. https://github.com/open-mmlab/mmaction2. 2020 (cit. on pp. 33, 43).

[13] M. Contributors. *Recognizer3d*. [Accessed March 26, 2023]. URL: https://github.com/open-mmlab/mmaction2/blob/master/mmaction/models/recognizers/recognizer3d.py (cit. on p. 37).

[14] M. Contributors. *ResNet3dSlowFast*. [Accessed: March 26, 2023]. URL: https://github.com/open-mmlab/mmaction2/blob/master/mmaction/models/backbones/resnet3d_slowfast.py (cit. on p. 37).

[15] M. Contributors. *SlowFast R50*. [Accessed March 26, 2023]. URL: https://github.com/open-mmlab/mmaction2/blob/master/configs/_base_/models/slowfast_r50.py (cit. on p. 37).

[16] M. Contributors. *SlowFastHead*. [Accessed: March 26, 2023]. URL: https://github.com/open-mmlab/mmaction2/blob/master/mmaction/models/heads/slowfast_head.py (cit. on p. 37).

[17] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893 (cit. on pp. 2, 11, 18).

[18] N. Deo and M. M. Trivedi. "Convolutional social pooling for vehicle trajectory prediction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 1468–1476 (cit. on p. 26).

[19] J. Donahue et al. "Long-term recurrent convolutional networks for visual recognition and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634 (cit. on pp. 3, 19, 20, 22).

[20] T. V. Duong et al. "Activity recognition and abnormality detection with the switching hidden semi-markov model". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 838–845 (cit. on pp. 3, 14, 18).

[21] C. Feichtenhofer, A. Pinz, and R. P. Wildes. "Spatiotemporal multiplier networks for video action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4768–4777 (cit. on pp. 3, 19, 22).

[22] W. Förstner and E. Gülch. "A fast operator for detection and precise location of distinct points, corners and centres of circular features". In: *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*. Interlaken. 1987, pp. 281–305 (cit. on pp. 2, 10, 18).

[23] G. Furtado. *Cognitive Layer (Section 4.5) | Give to Human Operation*. Online video. 2023. URL: https://youtu.be/vwaeXQJun5k (cit. on p. 60).

[24] G. Furtado. *Cognitive Layer (Section 4.5) | Human Aware Pick and Place Operation*. Online video. 2023. URL: https://youtu.be/WojhtaEOuoE (cit. on p. 58).

[25] G. Furtado. *Cognitive Layer (Section 4.5) | Take from Human Operation*. Online video. 2023. URL: https://youtu.be/WTq3KL9onXE (cit. on p. 61).

[26] G. Furtado. *Human Detection Module (Section 4.1) | MyPose*. Online video. 2023. URL: https://youtu.be/Is0z_eqkEFc (cit. on p. 54).

[27] G. Furtado. *Human Detection Module (Section 4.1) | OpenPose*. Online video. 2023. URL: https://youtu.be/nMA7ih5_B-s (cit. on p. 53).

[28] G. Furtado. *Object Detection Module (Section 4.2) | Image Agnostic Segmentation*. Online video. 2023. URL: https://youtu.be/JeyLIJ-Sfzg (cit. on p. 55).

[29] A. Gouda, A. Ghanem, and C. Reining. *Category-agnostic Segmentation for Robotic Grasping*. 2022. arXiv: 2204.13613 [cs.RO]. URL: https://github.com/AnasIbrahim/image_agnostic_segmentation (cit. on pp. 39, 43, 55, 56).

[30] A. Gupta et al. "Social gan: Socially acceptable trajectories with generative adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2255–2264 (cit. on pp. 25, 27).

[31] T. Han et al. "Human action forecasting by learning task grammars". In: *arXiv preprint arXiv:1709.06391* (2017) (cit. on pp. 23, 27).

[32] A. Hentout et al. "Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017". In: *Advanced Robotics* 33.15-16 (2019), pp. 764–799 (cit. on p. 1).

[33] Intel. *Intel RealSense ROS 2 Driver*. [Accessed: March 27, 2023]. 2021. URL: https://github.com/IntelRealSense/realsense-ros/tree/ros2-development (cit. on p. 43).

[34] Intel RealSense. *Intel RealSense Depth Camera D435i*. [Accessed March 26, 2023]. URL: https://www.intelrealsense.com/depth-camera-d435i/ (cit. on p. 42).

[35] A. Jain et al. "Structural-rnn: Deep learning on spatio-temporal graphs". In: *Proceedings of the ieee conference on computer vision and pattern recognition*. 2016, pp. 5308–5317 (cit. on pp. 24, 27).

[36] S. Ji et al. "3D convolutional neural networks for human action recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2012), pp. 221–231 (cit. on pp. 3, 19, 20, 22).

[37] R. P. Joshi et al. *ROS OpenPose*. https://github.com/ravijo/ros_openpose. 2019 (cit. on pp. 37, 43, 53, 54).

[38] A. Klaser, M. Marszałek, and C. Schmid. "A spatio-temporal descriptor based on 3d-gradients". In: *BMVC 2008-19th British Machine Vision Conference*. British Machine Vision Association. 2008, pp. 275–1 (cit. on pp. 2, 11, 13, 18).

[39] Y. Kong and Y. Fu. "Human action recognition and prediction: A survey". In: *International Journal of Computer Vision* 130.5 (2022), pp. 1366–1401 (cit. on p. 6).

[40] Y. Kong et al. "Action prediction from videos via memorizing hard-to-predict samples". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018 (cit. on pp. 23, 27).

[41] H. S. Koppula and A. Saxena. "Anticipating human activities using object affordances for reactive robotic response". In: *IEEE transactions on pattern analysis and machine intelligence* 38.1 (2015), pp. 14–29 (cit. on pp. 23, 27).

[42] C. P. C. Lab. *OpenPose: Real-time multi-person keypoint detection library for body, face, and hands estimation*. [Accessed: March 27, 2023]. 2021. URL: https://github.com/CMU-Perceptual-Computing-Lab/openpose (cit. on pp. 43, 53–55).

[43] I. Laptev. "On space-time interest points". In: *International journal of computer vision* 64.2 (2005), pp. 107–123 (cit. on pp. 2, 10, 11, 18).

[44] I. Laptev et al. "Learning realistic human actions from movies". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8 (cit. on pp. 3, 13, 18).

[45] J. Liu et al. "Spatio-temporal lstm with trust gates for 3d human action recognition". In: *European conference on computer vision*. Springer. 2016, pp. 816–833 (cit. on pp. 3, 19, 20, 22).

[46] Q. Liu et al. "Predicting the next location: A recurrent model with spatial and temporal contexts". In: *Thirtieth AAAI conference on artificial intelligence*. 2016 (cit. on pp. 25, 27).

[47] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[48] R. Luo and D. Berenson. "A framework for unsupervised online human reaching motion recognition and early prediction". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 2426–2433 (cit. on pp. 24, 27).

[49] S. Ma, L. Sigal, and S. Sclaroff. "Learning activity progression in lstms for activity detection and early detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1942–1950 (cit. on pp. 3, 19, 20, 22, 27).

[50] J. Martinez, M. J. Black, and J. Romero. "On human motion prediction using recurrent neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2891–2900 (cit. on pp. 24, 27).

[51] *MoveIt 2 - Easy-to-use open source robotics manipulation platform for developing commercial applications, prototyping designs, and benchmarking algorithms*. [Accessed: March 26, 2023]. URL: https://github.com/ros-planning/moveit2 (cit. on pp. 31, 43).

[52] NVIDIA. *NVIDIA Driver Downloads*. [Accessed: March 27, 2023]. 2023. URL: https://www.nvidia.com/download/index.aspx (cit. on p. 43).

[53] OpenCV. *OpenCV: Open Source Computer Vision Library*. [Accessed: March 27, 2023]. 2021. URL: https://github.com/opencv/opencv/tree/master (cit. on p. 43).

[54] Z. Qiu, T. Yao, and T. Mei. "Learning spatio-temporal representation with pseudo-3d residual networks". In: *proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5533–5541 (cit. on pp. 3, 19, 22).

[55] M. Raptis and L. Sigal. "Poselet key-framing: A model for human activity recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2650–2657 (cit. on pp. 3, 15, 18).

[56] N. Rhinehart et al. "Precog: Prediction conditioned on goals in visual multi-agent settings". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2821–2830 (cit. on p. 26).

[57] S. Robotics. *A C++ interface for controlling and receiving data from an UR robot using the Real-Time Data Exchange (RTDE)*. [Accessed: March 26, 2023]. 2021. URL: https://sdurobotics.gitlab.io/ur_rtde/index.html (cit. on pp. 31, 43).

[58] ROBOTIQ. *2F-85 and 2F-140 Grippers*. [Accessed March 26, 2023]. URL: https://robotiq.com/products/2f85-140-adaptive-robot-gripper (cit. on p. 42).

[59] U. Robots. *Universal Robots ROS 2 Driver*. [Accessed: March 27, 2023]. 2021. URL: https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver/tree/foxy (cit. on p. 43).

[60] ROS. *ROS 2 Foxy Fitzroy*. [Accessed: March 27, 2023]. 2020. URL: https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html (cit. on p. 43).

[61] ROS. *ROS Noetic Ninjemys*. [Accessed: March 27, 2023]. 2020. URL: http://wiki.ros.org/noetic/Installation/Ubuntu (cit. on p. 43).

[62] M. S. Ryoo. "Human activity prediction: Early recognition of ongoing activities from streaming videos". In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 1036–1043 (cit. on pp. 23, 24, 27).

[63] M. Ryoo et al. "Robot-centric activity prediction from first-person videos: What will they do to me?" In: *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2015, pp. 295–302 (cit. on pp. 23, 27).

[64] A. Sadeghian et al. "Sophie: An attentive gan for predicting paths compliant to social and physical constraints". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1349–1358 (cit. on pp. 26, 27).

[65] C. Schuldt, I. Laptev, and B. Caputo. "Recognizing human actions: a local SVM approach". In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. IEEE. 2004, pp. 32–36 (cit. on pp. 2, 11, 13, 18).

[66] S. Sharma, R. Kiros, and R. Salakhutdinov. "Action recognition using visual attention". In: *arXiv preprint arXiv:1511.04119* (2015) (cit. on pp. 3, 19, 20, 22).

[67] J. Shotton et al. "Real-time human pose recognition in parts from single depth images". In: *CVPR 2011*. Ieee. 2011, pp. 1297–1304 (cit. on p. 16).

[68] K. Simonyan and A. Zisserman. "Two-stream convolutional networks for action recognition in videos". In: *arXiv preprint arXiv:1406.2199* (2014) (cit. on pp. 3, 19, 22).

[69] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014) (cit. on pp. 3, 19, 22).

[70] S. Song et al. "An end-to-end spatio-temporal attention model for human action recognition from skeleton data". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017 (cit. on pp. 3, 19, 20, 22).

[71] H. Su et al. "Forecast the Plausible Paths in Crowd Scenes." In: *IJCAI*. Vol. 1. 2017, p. 2 (cit. on pp. 25, 27).

[72] J. Sung et al. "Unstructured human activity detection from rgbd images". In: *2012 IEEE international conference on robotics and automation*. IEEE. 2012, pp. 842–849 (cit. on pp. 17, 18).

[73] M. C. Systems. *How Can Cobots Make Automation Easier?* [Accessed March 26, 2023]. 2020-03. URL: https://www.markonecs.com/how-can-cobots-make-automation-easier/ (cit. on p. 41).

[74] U. K. Team. *Ubuntu Mainline Kernel Builds*. [Accessed: March 27, 2023]. 2020. URL: https://kernel.ubuntu.com/~kernel-ppa/mainline/v5.8/ (cit. on p. 43).

[75] D. Tran and A. Sorokin. "Human activity recognition with metric learning". In: *European conference on computer vision*. Springer. 2008, pp. 548–561 (cit. on pp. 2, 3, 8, 13, 18).

[76] D. Tran et al. "A closer look at spatiotemporal convolutions for action recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 6450–6459 (cit. on pp. 3, 19, 22).

[77] D. Tran et al. "Learning spatiotemporal features with 3d convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497 (cit. on pp. 3, 19, 22).

[78] Universal Robots. *The UR3e - Compact table-top robot*. [Accessed March 26, 2023]. URL: https://www.universal-robots.com/products/ur3-robot/ (cit. on p. 41).

[79] G. Varol, I. Laptev, and C. Schmid. "Long-term temporal convolutions for action recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), pp. 1510–1517 (cit. on pp. 3, 19, 22).

[80] R. Vemulapalli, F. Arrate, and R. Chellappa. "Human action recognition by representing 3d skeletons as points in a lie group". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 588–595 (cit. on pp. 17, 18).

[81] V. Villani et al. "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications". In: *Mechatronics* 55 (2018), pp. 248–266 (cit. on p. 2).

[82] R. Villegas et al. "Decomposing motion and content for natural video sequence prediction". In: *arXiv preprint arXiv:1706.08033* (2017) (cit. on pp. 23, 27).

[83] M. Vrigkas, C. Nikou, and I. A. Kakadiaris. "A review of human activity recognition methods". In: *Frontiers in Robotics and AI* 2 (2015), p. 28 (cit. on p. 5).

[84] J. Walker et al. "An uncertain future: Forecasting from static images using variational autoencoders". In: *European Conference on Computer Vision*. Springer. 2016, pp. 835–851 (cit. on pp. 24, 25, 27).

[85] H. Wang and C. Schmid. "Action recognition with improved trajectories". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 3551–3558 (cit. on pp. 2, 12, 13, 18).

[86] H. Wang et al. "Dense trajectories and motion boundary descriptors for action recognition". In: *International journal of computer vision* 103.1 (2013), pp. 60–79 (cit. on pp. 2, 11–13, 18).

[87] J. Wang et al. "Mining actionlet ensemble for action recognition with depth cameras". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 1290–1297 (cit. on pp. 17, 18).

[88] D. Weinland, R. Ronfard, and E. Boyer. "Motion history volumes for free viewpoint action recognition". In: *Workshop on modeling people and human interaction (PHI'05)*. 2005 (cit. on pp. 2, 7, 8, 18).

[89] Wikipedia contributors. *K-nearest neighbors algorithm — Wikipedia, The Free Encyclopedia*. [Online; accessed 26-February-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=1073696056 (cit. on p. 14).

[90] Wikipedia contributors. *Support-vector machine — Wikipedia, The Free Encyclopedia*. [Online; accessed 26-February-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=Support-vector_machine&oldid=1064262971 (cit. on p. 13).

[91] L. Xia, C.-C. Chen, and J. K. Aggarwal. "Human detection using depth information by kinect". In: *CVPR 2011 workshops*. IEEE. 2011, pp. 15–22 (cit. on pp. 2, 8, 9, 18).

[92] X. Yang and Y. L. Tian. "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor". In: *2012 IEEE computer society conference on computer vision and pattern recognition workshops*. IEEE. 2012, pp. 14–19 (cit. on pp. 17, 18).

[93] Y. Yang and D. Ramanan. "Articulated human detection with flexible mixtures of parts". In: *IEEE transactions on pattern analysis and machine intelligence* 35.12 (2012), pp. 2878–2890 (cit. on pp. 9, 18).

[94] B. Yao et al. "Human action recognition by learning bases of action attributes and parts". In: *2011 International conference on computer vision*. IEEE. 2011, pp. 1331–1338 (cit. on pp. 3, 16, 18).

