University of Padova
Department of Information Engineering
Master Degree in

Control Systems Engineering

Master Thesis

Skeleton-based human action and gesture recognition for human-robot collaboration

Supervisor: Prof. Stefano Ghidoni
Co-supervisor: Matteo Terreran, PhD

Candidate: Margherita Lazzaretto
ID n. 1239115

Academic Year 2021/2022
Date 21/02/2022

Abstract

The continuous development of robotic and sensing technologies has led in recent years to an increased interest in human-robot collaborative systems, in which humans and robots perform tasks in shared spaces and interact with close and direct contacts. In these scenarios, it is fundamental for the robot to be aware of the behaviour that a person in its proximity has, to ensure their safety and anticipate their actions in performing a shared and collaborative task. To this end, human activity recognition (HAR) techniques have been often applied in human-robot collaboration (HRC) settings. The works in this field usually focus on case-specific applications. Instead, in this thesis we propose a general framework for human action and gesture recognition in a HRC scenario. In particular, a transfer learning enabled skeleton-based approach that employs as backbone the Shift-GCN architecture is used to classify general actions related to HRC scenarios. Pose-based body and hands features are exploited to recognise actions in a way that is independent from the environment in which these are performed and from the tools and objects involved in their execution. The fusion of small network modules, each dedicated to the recognition of either the body or hands movements, is then explored. This allows to better understand the importance of different body parts in the recognition of the actions as well as to improve the classification outcomes. For our experiments, we used the large-scale NTU RGB+D dataset to pre-train the networks. Moreover, a new HAR dataset, named *IAS-Lab Collaborative HAR* dataset, was collected, containing general actions and gestures related to HRC contexts. On this dataset, our approach reaches a 76.54% accuracy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human activity recognition (HAR) is a computer vision task that consists in understanding the human behaviour through the classification of a person's movements, according to a previously defined set of activities of interest. These can be performed in space with one or several body parts, during a certain, usually limited, time interval. The understanding of the activities aims at being universal, namely the same label should be associated to different persons performing the same activity in different fashions or under different circumstances.

Human activity recognition covers a vast area of applications, from safety monitoring in industry [1], public events [2] or private homes [3], to human-computer interaction [4], human-robot interaction [5–7], virtual reality [8] and healthcare applications [9, 10]. In this thesis, in particular, we focus on a human-robot collaboration scenario, in which recognising actions is particularly important in order for the robot to be aware of the movements performed by people in its proximity. Indeed, a collaborative task is carried out in an effective way if the robot understands *what* actions the person is doing, so that it can properly react to them while guaranteeing the human worker's safety.

Depending on the application, the recognition process can concern different types of movements, which can be classified hierarchically according to their complexity as follows [11]:

- *Actions*, consisting in simple atomic activities such as walking, writing or drinking. Usually, different subjects perform the same action in similar ways. Action recognition can be applied in a large variety of fields, from assisted living [3] to health-care [9] to human-robot collaboration [5].

- *Gestures*, which comprehend body movements that are aimed at an intentional or unintentional nonverbal communication, usually performed with the hands or with the upper part of the body. Their execution can be universal or can be influenced by the social and cultural environment as well as by the individual judgment. For example, a gesture with meaning *stop* can be done by putting

one or two hands ahead, by forming an X shape with the forearms or by waving both hands simultaneously. Gesture recognition is usually applied to achieve a communication between two entities, for example in virtual reality applications [8], human-computer interactions [4] or sign language recognition [12].

- *Behaviours*, namely physical actions that reflect an emotional or psychological state. Behaviour recognition can be applied for example in healthcare applications [10].

- *Interactions*, involving more than one entity, can in general refer to two people engaging in mutual actions or in the interaction of a human with an object. For example, human-object interaction recognition can be performed for safety monitoring [13].

- *Group actions*, consisting in actions performed together by a small group of people. Their recognition can be applied for example to some security applications [14].

- *Events*, which include activities that take place in a specific environment or high level actions regarding a large group of people. An application is the recognition of anomalous events in public events [2].

The first categories are considered more easy to recognise, while the complexity increases for the last entries of the list.

Nowadays, the recognition process is mostly based on supervised machine learning techniques, which can be summarised into three main stages:

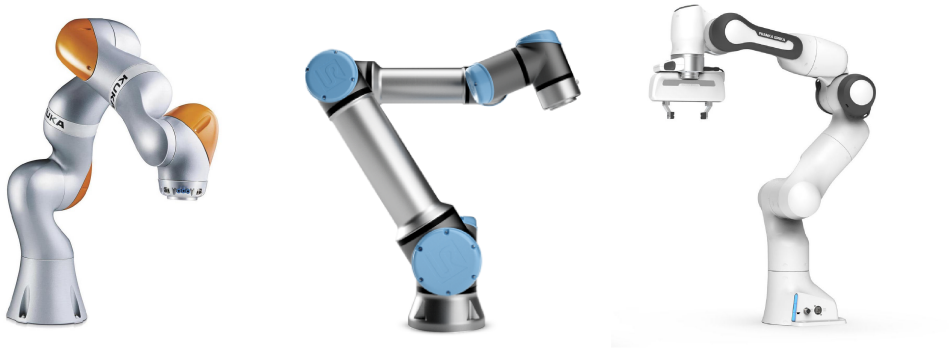
- *Data collection*: the first phase of recognising an activity consists in collecting data descriptive of the performed movements. The type of collected data depends on the sensors employed for their acquisition. Examples of data types are RGB-D videos captured with vision sensors, which include both RGB and depth frames, or signals collected with wearable sensors such as IMU and EMG sensors.

- *Features extraction*: Features are pieces of information that provide a shared representation for all the data belonging to the same category. For example, two actions classified with the same label should be described by a common set of features. In the HAR task, the extraction process aims at defining the set of features that are shared across all the data relative to the same activity. For each data type, different kinds of features can be extracted, and in some cases a data pre-processing phase can be carried out, after which the features are extracted from the processed data. Features extraction techniques divide into hand-crafted methods, which employ traditional computer vision algorithms to extract relevant information from the available data, usually based on some prior knowledge, and deep learning methods, in which a neural network is trained to automatically extract features from the collected data.

- *Classification*: the extracted features are then employed to classify the activities, by grouping together data with features that are considered close according to some measure of similarity. Classification methods can in general be divided into traditional machine learning methods and deep learning methods. In the second case, usually the same deep learning network employed to extract deep features is successively exploited to classify new, previously unseen data examples (*end-to-end learning*).

The choice of the learning method employed for recognition depends on the application and on the type of activity to be recognised. Generally, for simpler activities such as *actions*, *gestures* or *behaviours*, high level features can be more easily extracted from the data. An example is given by pose-based features, extracted from RGB-D data or from signals captured with wearable sensors (e.g., Xsens motion capture systems [15]). They consist in the representation of human movements through a sequence of keypoints of one or more parts of the body, such as hands in the case of *gestures*, hands and face in the case of *behaviours*, or the full body in the case of *actions*. The set of body keypoints is often referred to as *skeleton*, and methods that classify activities starting from pose features are called *skeleton-based*. The HAR complexity increases when more people are involved in the activity as in the cases of *interactions, group actions* and *events*. In these scenarios, the use of wearable sensors is less feasible and mostly vision-based approaches are adopted. For this reason, the learning methods employed in these cases usually rely on more low-level features, such as the ones extracted using deep learning methods from raw data like RGB-D videos. A more detailed overview on HAR methods will be outlined in Section 2.1 of the following chapter.

As mentioned above, in this work we are interested in the application of action and gesture recognition to human-robot collaboration scenarios. Indeed, in recent years the technological development of robotic systems and their ever-increasing popularity in various industrial sectors, as well as in everyday life, has led research to investigate ways of making maximum use of their potential. In particular, **human-robot collaboration (HRC)** in manufacturing has raised considerable interest, serving the purpose of optimally exploiting the capabilities of humans and robots in the execution of a shared task. While robots are better suited to perform repetitive and fatiguing tasks, other operations that involve cognitive skills and flexibility can be carried out more efficiently by a person. A specific class of robots, namely collaborative robots or *cobots*, are usually employed in these contexts, and they typically consist in medium-sized robotic arms such as KUKA LBR, UR and Franka Emika Panda robots, shown in Figure 1.1. While traditional industrial robots operate according to rigid pre-generated codes and are not developed to work in proximity and/or close contact with humans, collaborative robots are often required to dynamically adapt to the behaviour of a human operator in a shared workspace. Generally, the level of interaction that a person has with a collaborative robot can vary depending on the specific scenario considered, and can be classified according to different aspects [16]. For what concerns the methodology with which human and robot



(a) *KUKA LBR* (b) *UR16* (c) *Franka Emika Panda*

Figure 1.1: *Examples of collaborative robots.*

work together, this can be defined as:

- *coexistence*, when the human shares the workspace with the robot but they aim at achieving different goals
- *cooperation*, when the human and robot work toward the same objective, but do not perform shared tasks
- *collaboration*, when the human and the robot work on the same object in a shared workspace, and contact between them is allowed.

Moreover, according to the technical specification ISO/TS 15066:2016, different safety requirements for collaborative industrial robot systems can be defined depending on the level of contact allowed between human and robot, namely:

- *Safety-Rated monitored stop (SRMS)*, if the robot is programmed to safely stop when the human enters its workspace
- *Hand guiding (HG)*, when the only movement the robot can make with someone within its workspace is controlled and hand guided by a human operator
- *Speed and separation monitoring (SSM)*, when human and robot share the workspace and the robot adjusts its moving speed according to the distance from the person
- *Power and force limiting (PFL)*, when human and robot share the workspace and contact between human and robot is allowed, usually in a collaborative context.

In all the cases reported above, and especially in the ones in which a close contact between human and robot is envisaged, it is important for the robot to be aware of the surrounding context and of the actions and movements being performed by the human. This is true first of all for safety purposes, since human and robot operate in the same

workspace. Moreover, it can also be useful both for the purpose of dynamically controlling the behaviour of the robot depending on what the human is doing and to allow the human to actively communicate with the robot. Indeed, the robot awareness can be addressed as a HAR problem: by recognising the actions that a human operator is doing inside the workspace, the robot can react accordingly, while gesture recognition can be performed to receive directions and instructions.

1.1 Problem definition and goal of the thesis

This thesis focuses on vision-based human action and gesture recognition in a human-robot collaboration context using deep learning methods. In particular, a skeleton-based approach is adopted. The choice of using pose features is based on the state-of-the-art results obtained by skeleton-based approaches to HAR. The advantage of using skeleton features is given by different factors. First of all, compared to raw RGB-D data they are a very compact information representation that can be expressed with either 2D or 3D coordinates, encoding only the essential points needed to describe body movements. The spatial information is naturally defined by the relationships between joints in the human body, while temporal information is represented by the evolution of each joint in time. This high-level information is inherently free of disturbances that could be caused by the external environment, such as background, lighting and aesthetic differences of people such as clothes or skin colour.

By applying skeleton-based action recognition in a HRC context, we also aim at defining a generic framework applicable to different HRC scenarios, since pose features do not contain any information about the specific environment or the specific objects and tools used. For this purpose, we collected a dataset containing a selection of actions and gestures, which were chosen by reviewing several human-robot collaboration works [5, 6, 17–33]. All the actions and gestures reported in the surveyed papers that were deemed significant for a HAR task were included in the dataset. We highlight that collaboration scenarios are particularly interesting as they represent the latest and more advanced shared work setting between human and robot. While in other contexts it might be sufficient for the robot to stop or slow down when a person comes in its proximity, to make a collaboration setting safe and efficient it is necessary for the robot to understand *what* the human is doing, thus the task of action recognition becomes particularly relevant.

As both actions, such as *screw, hammer, walk*, and gestures, such as *stop, left, right*, are included in our dataset, the skeleton model considered in the recognition process contains both keypoints of the body and of hands. This makes our task particularly challenging, since usually action and gesture recognition are treated as separate tasks in literature. To perform action classification, a deep learning framework was adopted. More in detail, we first pre-trained a shift graph convolutional network [34] (see Section 4.2.1) on a large scale action recognition dataset (NTU RGB+D [35]). Then, we employed a deep learning technique called transfer learning, treated in more detail in

Section 5.2.1, to train this model to recognise actions from our newly collected dataset. Moreover, the effect of using only some specific subsets of the skeleton information was studied. In particular, besides training a network using the complete skeleton information (body + hands), we also trained separate network modules on the body skeleton sequences (hands excluded) and on the hands skeletons sequences. Then, different ways of ensembling these networks were tested and the results were compared with the ones obtained by training a single neural network using the full skeleton information. The fusion of different modules has was employed to better understand the role of the different body parts in the recognition of an activity, as well as to improve the classification results.

The remainder of this thesis is organised as follows. In Chapter 2, a brief overview of the related works in the human activity recognition field and of HAR applications to human-robot collaboration scenarios is reported. Chapter 3 discusses the process of skeleton features extraction, by introducing the main datasets and methods for 2D and 3D human body pose estimation. In Chapter 4, an overview of the main human action and gesture recognition datasets is presented, followed by a short theoretical introduction to the state-of-the-art deep learning models developed for skeleton-based action recognition. In Chapter 5 a detailed description of the proposed HAR for HRC framework is reported, putting an emphasis on the different steps followed in achieving our goal. In Chapter 6 the experiments performed to evaluate the proposed framework and the obtained results are described, from the collection of the dataset, to the pre-training of the neural networks employed, to the transfer-learning phase. Finally, Chapter 7 concludes with a short summary of our work and of the results achieved. Moreover, possible future directions and potential improvements of the conducted study are outlined.

Chapter 2

Related works

Being human action and gesture recognition in the context of human-robot collaboration the focus of this thesis, in this chapter an overview of the main works in literature related to these research areas is presented. In particular, first papers dealing with action recognition methods are surveyed and successively a more detailed review of works that apply action recognition in human-robot collaboration scenarios is presented.

2.1 Human action recognition

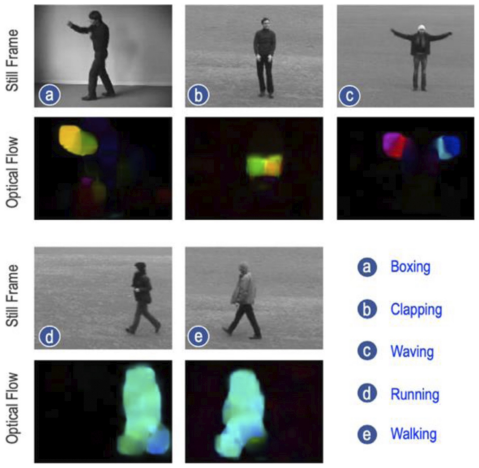
As mentioned in Chapter 1, the task of human action recognition has become relevant in the field of computer vision in recent years, being employed in the development of many important applications. Different techniques have been studied to recognise the movements performed by a person, which differentiate depending on the sensors used to acquire data, on the type of data acquired and on the features extracted from these data, and finally on the method used to classify the actions.

For what concerns the types of sensors, this thesis focuses on vision sensors, that are among the more widely employed, with the advantage of being non-intrusive low-cost devices that allow to acquire multiple streams of data simultaneously, such as RGB, depth, and infra-red video frames. However, usually a single vision sensor is not sufficient to acquire robust data, and a vision sensors network composed by multiple cameras is required in order to deal with common computer vision problems such as occlusions and restricted field of view. Moreover, other types of sensors could be used in recognising actions, alone or jointly with vision sensors, e.g., inertial measurement units (IMUs) or other wearable sensors [3, 6, 36]. The main drawbacks of these is that they usually require a higher technological knowledge from the user and they are more intrusive, since a physical contact of the person with the acquisition devices is needed.

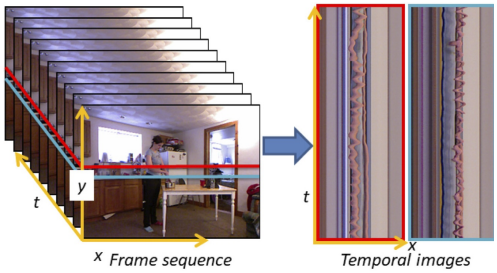
With respect to a traditional object recognition task, the main challenge in the task of action recognition consists in handling both a spatial and a temporal dimension. This

is usually done by extracting features from the available data that describe how the pose of a person changes over time. In vision-based approaches, the features can be either hand-crafted or obtained with learning methods. We can divide the different types of features as follows:

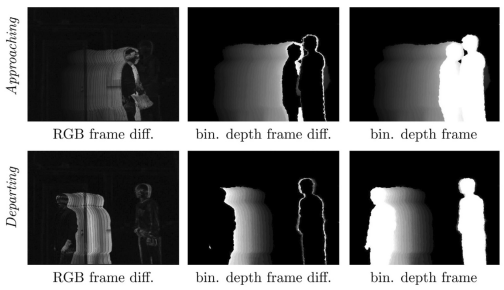
- *image representations*: to encode the body motion, a single 2D image can be generated in different ways. In [30] the differences in consecutive frames are used to obtain an optical flow image that describes the motion (Figure 2.1a). Another similar method consists in computing temporal templates starting from the optical flow, such as for example the motion energy image and the motion history image [37](Figure 2.1c). The main drawback of these encodings is that they collapse an intrinsically tridimensional information in a 2D image. In this way, the temporal order of the action is lost, which can cause difficulties in the distinguishing some kind of activities, in particular actions that are somehow temporally mirrored such as opening and closing a door or picking up and putting down an object. In [38], instead, a temporal image is computed starting from the 3D volume obtained by stacking consecutive frames, with x, y as spatial coordinates and t as temporal



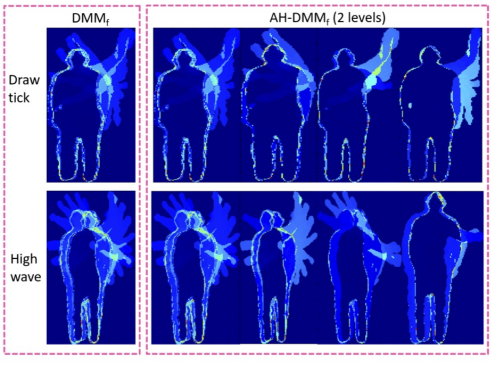
(a) *Optical flow*



(b) *Temporal images*



(c) *Temporal templates*

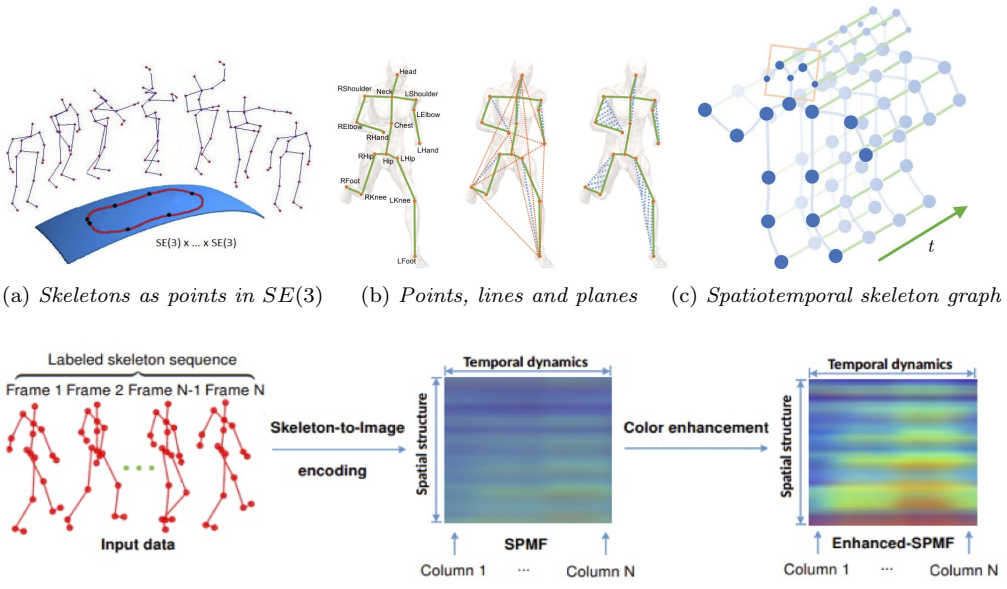


(d) *Depth motion maps*

Figure 2.1: *Examples of image representations.*

coordinate (Figure 2.1b). This volume is sliced at a predefined value of y to obtain a 2D image. The result of this operation is however strongly dependent on the choice of the slicing value of y and is not very robust to any changes of scenario. In [39], motion maps are generated from depth frames in an adaptive way to take into account the temporal order of the frames, leveraging the motion energy of each frame.

- *body pose representations*: another common type of feature exploited to recognise actions is the 2D or 3D body pose. This is usually described by a set of joints positioned in salient points of the human body (e.g., head, neck, shoulders, etc...), connected with each other according to the human anatomy. The body pose can be easily represented through a graph, where each node encodes a human body joint through its coordinates in space and each edge connects two adjacent joints in order to create a skeleton-like representation. Early models for body pose estimation employed classical computer vision approaches, such as extracting features of body parts with feature descriptors like the Histogram of Oriented Gradients, or adopting structural models to articulate constraints used to parse body parts [40]. However, deep learning approaches have recently outperformed classical body pose estimation methods, leading to higher estimation accuracies and better generalisation capabilities [40–42]. Skeleton joint information is typically processed to obtain more descriptive motion features. In [43], joints are represented as points in



(a) *Skeletons as points in $SE(3)$* (b) *Points, lines and planes* (c) *Spatiotemporal skeleton graph*

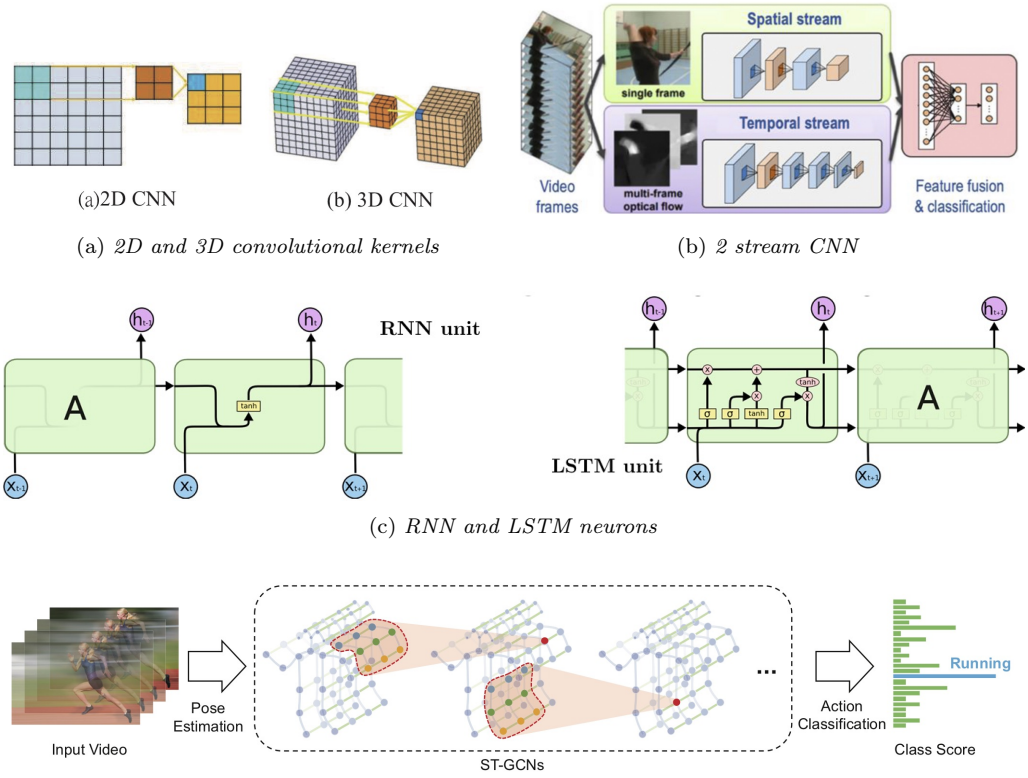
(d) *Skeleton Posture Motion Features*

Figure 2.2: *Examples of body pose representations.*

$SE(3)$ and the body motion is described through geometric relationships between the joints (Figure 2.2a). In [44], a motion map given by the evolution of the body poses in time is encoded in an image (Figure 2.2d). In this case, no temporal order information is lost in the creation of the image, since the spatial information of the skeleton can be easily condensed in a one dimensional vector containing the coordinates of all the body joints, while time is encoded in the second dimension. In [45], more geometric features are computed using the joints coordinates, such as lines (connections between joints) and planes (enclosed within three or more joints) (Figure 2.2b). Similarly, in [34, 46–50], four skeleton representations are derived, namely shape representation of joints, motion representation of joints, shape representation of bones and motion representation of bones. Moreover, in [34, 47–50] the body graph is extended to the temporal dimension, namely each joint at time t is connected with other joints at time t according to the body graph and also to its evolution at time $t+1$ (Figure 2.2c). Finally, [39] considers the pairwise differences of joint coordinates between two consecutive time instants and the difference in position of each joint with respect to a reference joint.

- *Deep features*: it is also possible to extract features from RGB or depth data using deep learning methods through the training of a neural network, which subsequently serves as action classifier. To capture both spatial and temporal features in a video, 3D convolutional neural networks have been introduced in several works [51–53]. These take as input sequences of images, i.e., image volumes, and extend the traditional kernel of convolutional networks to a three dimensional kernel (Figure 2.3a). In [30, 54], instead, the authors adopt a 2D convolutional network with multiple streams fused together at score level, where each stream captures either spatial or temporal features (Figure 2.3b). The introduction of recurrent neural networks and, in particular, long-short term memory networks (LSTMs) made it possible instead to learn both spatial and temporal information exploiting the network's integrated gated memory units [55, 56] (Figure 2.3c). More recently, transformer networks have been replacing LSTMs. Indeed, by introducing attention mechanisms, they are able to learn both the sequentiality and, more generally, the context encoded by the data being analysed [57]. Finally, we highlight that deep features can also be extracted from the previously listed representations such as optical flow [30] or pose based features [45], rather than directly from raw rata. A particular type of neural networks, called graph convolutional networks (GCNs), has been recently introduced to learn relationships between body keypoints during a person's movement, starting from a spatiotemporal body graph [34, 47–50] (Figure 2.3d).

Regarding the classification phase, traditional methods include support vector machines (SVM) [38, 58], random forest classifiers [38], hidden Markov models (HMMs) [29], or other classical statistics techniques such as chi-square based feature selection [59]. SVMs use hyperplanes to separate elements of different classes, with the possibility of



(a) *2D and 3D convolutional kernels*

(b) *2 stream CNN*

(c) *RNN and LSTM neurons*

(d) *Graph convolutional networks stages*

Figure 2.3: *Deep learning models for feature extraction and classification.*

choosing between different kernels depending on the needs, while HMMs model the time component of videos with transitions between states.

The increasing development of deep neural networks and the ongoing research on the topic has strongly impacted the performance of action recognition methods. As mentioned above, different types of trained neural networks can be used to classify actions. The main issue when dealing with deep learning methods concerns the amount of data needed to obtain good results and the high computational power required. More in detail, during the training phase the computational effort required by a network increases with the size of the data, which is determined both by the number of examples used and by the size of each example, and with the network complexity, which can be considered proportional to the number of operations computed for each input sample. Furthermore, if the data being fed to the network is heavy or the network is particularly complex, also the times necessary to classify new data increase. Generally, these two factors are connected, i.e. learning features from heavier and more complex data may require more complex networks and thus a higher computational cost. For these reasons, pose-based features are often preferred, since the sequences of skeleton joint coordinates represent a

lightweight input when compared to images or 3D volumes. Moreover, as already mentioned, the human skeleton can be easily represented with a graph, making it a good input for some deep learning models such as graph convolutional networks. In particular, according to the literature, the introduction of spatiotemporal graph convolutional networks [34, 47–50] for skeleton-based human action recognition has led to a great improvement in the performances obtained on many human action recognition datasets. State-of-the-art results in skeleton-based action recognition are also achieved with some transformer networks [60].

Finally, it is worth mentioning some recent attempts to learn and recognise human behaviour with deep reinforcement learning [61], which represents a powerful tool, modelled on natural animal behaviour, of learning more rewarding features. However, as the choice of hyperparameters for training is very delicate in order to obtain good results, these techniques have not yet managed to outperform other aforementioned methods.

2.2 Action recognition for human-robot collaboration

Human action recognition is largely applied in human-robot interaction contexts. The aim of the recognition can vary from social and educational purposes [62, 63] to safety and productivity improvement in manufacturing industry and assembly chains [5–7, 18, 29]. In social scenarios, mostly hand gestures or face expressions are recognised, allowing to communicate and interact with the robot. In industrial settings, instead, each specific scenario analysed generally entails a certain level of interaction between humans and robots, based on which the purpose of action recognition (safety, communication, etc.) is defined and the actions of interest required for the given scenario are designed.

When a low level of contact and collaboration between humans and robots are contemplated, the actions to be recognised include mainly circumstantial actions with respect to the robot's workspace. For example, in [5] the actions recognised are *passing, observing* and *dangerously observing, interacting*. The recognition is done with a multi-modal approach: a 3D-CNN is used to learn features from the sequences of RGB frames, while signals collected with haptic sensors are employed to detect collisions between human and robot and are fed to a 1D-CNN.

When more collaboration between human and robot is expected, generally the actions to be recognised concern instead different stages of assembly. For example, in [6], the recognised actions include *grabbing a tool from the toolbox, inserting the screws, tightening the screws* and *putting back a tool in the toolbox*. These are recognised using the fusion of signals collected with EMG and IMU sensors, as well as skeleton features, all classified using CNNs. In [30] the assembly actions include instead *cleaning, hammering, polishing, smearing, installing, screwing* and *marking*, recognised using a two-stream CNN as the one shown in Figure 2.3b. In [7] three types of interactions are taken into consideration, namely the human receiving a plate from the robot, the robot reacting to the human fetching a screwdriver by grasping screws to pass to the operator and finally the human

handing the robot a screwdriver. Here, a probabilistic approach that relies on the phase estimation of human movements is employed to classify human actions, where phase denotes the encoding of motion trajectories. In [29] it is described an industrial setting in which a two-armed robot cooperates with a person, and the actions to be recognised include receiving parts from one of the robot's two arms, joining two parts, screwing them together and putting the assembled product in a box. The recognition is done using both signals collected with inertial sensors placed on tools and body keypoints estimated from depth frames collected by a camera with a top view, while a hidden Markov model is employed in the classification phase. In [27] similar activities are considered, such as *taking a product or a component, move a product, grab a tool, put on screws, hold a product, tighten the screws, check* and *place product*. Here, image features and pose features are combined to capture the action patterns. In particular, the former are extracted from images cropped on hands, on which the attention of the recognition process is focused.

Finally, in some HRC scenarios, gesture recognition is performed to allow a better communication between human and robot. In [18], for example, six gestures the human uses to communicate with the robot (on, off, right, left, up, down) are recognised by fusing together three different modalities, namely speech command (with a CNN), hand motion (with a LSTM), and body motion (with transfer learning on the Inception-v3 network).

Clearly, recognising actions and gestures in a human-robot collaborative assembly process allows the human to communicate directly or indirectly with the robot, speeding up the time needed to complete an operation and improving the safety of the worker. However, in the literature there is a lack of a general framework for action recognition in HRC scenarios. Indeed, by reviewing works on action recognition for HRC in an industrial setting it emerges that most works focus on a case-specific scenario.

In particular, as emerges from what reported above, there are two main strands of work in this field, the former focusing on recognition of assembly actions and the latter on recognition of gestures to communicate with the robot. In the first case, usually an assembly operation is defined as a sequence of several actions that vary in type and order of execution according to the case examined. It follows that, for each action recognition methodology proposed, a new dataset, descriptive of the specific assembly operation considered, is acquired. These small datasets are in general not publicly available and only cover small sets of actions. There lacks instead a dataset and a general benchmark for the recognition of actions in a generic human-robot collaboration context. Moreover, often the considered assembly actions are paired with the tools employed to perform them [6, 28, 31, 33]. These tools can be recognised with object detection algorithms to improve the action recognition performance, but they make the recognition process even more specific, since in different contexts different tools are usually employed.

For what concerns communication gestures, they refer to movements, mostly linked to the hands or to the upper body portion, that aim at giving a direct instruction to the

robot, such as moving in a certain direction. Most of the times they are performed with a single hand and, again, they are specified according to the considered scenario, namely a vocabulary of gestures is predefined, assuming that these should always be performed in the same way [18, 20, 64].

Despite the specificity of the applications studied in this field, it can be noticed that many actions and gestures are recurrent among most of the human-robot collaboration scenarios presented in literature. It could then be interesting to define a general set of actions that do not depend on the specific environment in which they are performed or on the manipulated tools and objects, and that are considered useful to recognise in a generic human-robot collaboration context. Moreover, it could be useful to include in the recognition process both actions and gestures to improve the collaboration efficiency and to increase the variety of HRC scenarios enclosed in such general framework. These two issues are addressed in the action and gesture recognition framework for HRC proposed in this work.

Chapter 3

Human body pose estimation

Human body pose estimation is a computer vision task that aims at estimating the position of skeletal joints for each person detected in an image. Generally, the keypoints estimated from a RGB image are two-dimensional, but by exploiting depth sensors information or multi-view scenarios also 3D body keypoints can be estimated. The performance of pose estimation methods has drastically improved after the introduction of deep convolutional neural networks, with the consequence of encouraging research towards the study of skeleton-based HAR methods. Clearly, a robust estimation of the pose of the subject performing the action (i.e. the use of an accurate human pose estimation algorithm) is crucial to achieve a good performance in skeleton-based action recognition approaches.

In this chapter, a brief overview of pose estimation methods is provided. First, the most popular datasets employed for human pose estimation are described. Then, methods for estimating 2D body pose using RGB-D frames are presented. Finally, a brief overview on 3D body pose estimation methods is reported.

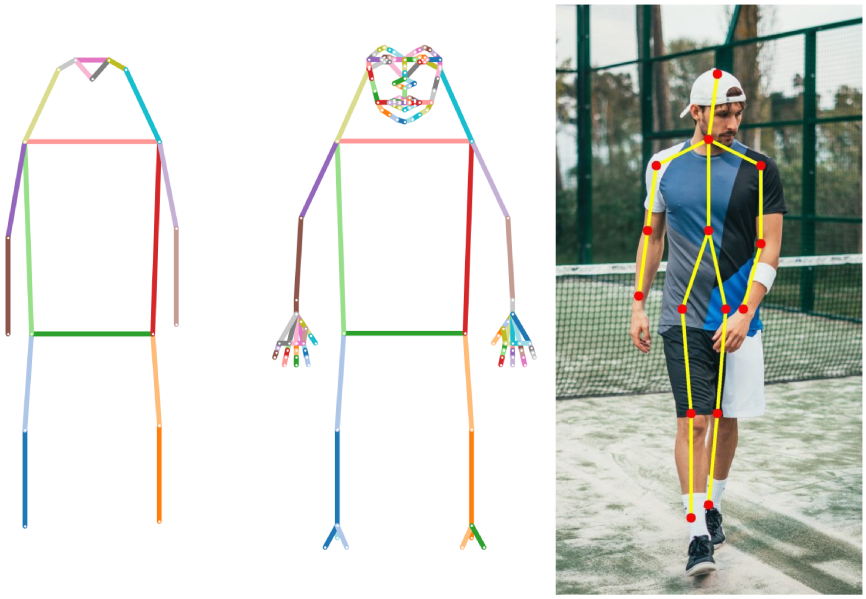
3.1 Datasets for body pose estimation

There exist different standard datasets commonly used to train deep learning based pose estimation models. Generally, each dataset differs for the convention used to annotate the human poses in the data, leading to different topologies of the skeletons. Two of the more commonly used datasets for training and testing body pose estimation networks are:

- MS COCO dataset [65]: orginally designed for object detection, it was later extended with body keypoints annotations of images picturing one or more person. It contains around $200k$ images and the body model includes 17 joints, as shown in Figure 3.1a. In [66] the dataset was extended with *whole-body* annotations. These include also the keypoints of hands, face and feet, with 42 (21 per hand), 68 and 6

(3 per foot) additional points respectively (Figure 3.1b).

- MPII dataset [67]: it contains body keypoints annotations for about $25k$ images extracted from YouTube videos. The body model contains 16 joints and is shown in Figure 3.1c



(a) *COCO body model* (b) *COCO whole-body model* (c) *MPII body model*

Figure 3.1: *Main skeleton conventions of body pose estimation datasets.*

Some 3D body pose estimation datasets are also available, like the HumanEva-I&II [68], the Human3.6M [69] and the MPI-INF-3DHP [70], but are not as varied in terms of background, subjects and environment settings as the previously described ones, making 2D pose estimation preferable in general. In particular, the mentioned 3D body pose datasets are all captured by multi-camera systems, whose complexity forces to collect all the samples in the same indoor environment, limiting background variability. Moreover, the number of subjects recorded is vey limited, namely 4 in the HumanEva dataset, 8 in the MPI-INF-3DHP and 11 in the Human3.6M.

Besides full body pose estimation datasets, several datasets designed specifically for hand pose estimation have been proposed, both for the 2D case, such as OneHand10k [71], and for the 3D case, like NYU [72], HANDS2017 [73] and FreiHAND [74]. Usually, they contain annotations of 21 keypoints for each hand, and can be used for tasks involving only hands such as gesture recognition. However, generally these datasets only contain cropped hands images, making it difficult to use any models trained on them, without further processing, in scenarios in which the camera is not close to the hands but it frames, for example, the entire body. Note that all the mentioned 3D hand pose datasets

are captured with a single camera and are developed for hand pose estimation from single depth images. Indeed, it can be observed that single-view 3D hand pose estimation is a more commonly addressed task than 3D body pose estimation, due to the greater accuracy of short-range depth frames. Hence, datasets for 3D hand pose estimation are more popular compared to the body estimation case. Another 2D hand pose estimation dataset is the CMU Panoptic Hands dataset [75], which was created by manually annotating the hands joints of the subjects in a subset of images taken from the MPII dataset and from the NZSL dataset [12], the latter proposed originally for sign language gestures recognition. Unlike the other datasets described above, the images in this dataset are not cropped on the hands and contain more varied, contextual information, making it more suitable for hands pose estimation in cases in which close-ups on the hands are not available.

3.2 Models for 2D body pose estimation

The datasets described above can be used to train deep learning models to predict the pose of a person. These models can be then exploited to estimate body poses in images of real-life scenarios, working as regressors for the body keypoints. In particular, in many applications such as skeleton-based human action recognition it can be useful to retrieve the body pose in real-time, especially when safety or communication purposes are involved. For example, we might be interested in recognising actions potentially dangerous for a worker in a controlled environment, in order to timely act in taking the necessary precautions ad avoid damages. Another example can be given by skeleton-based recognition of gestures employed to communicate: if the pose of the hands is estimated quickly, it is certainly easier to achieve a clearer and more fluid communication.

However, being body pose estimation a difficult task, the architectures that achieve state-of-the-art performances usually entail a high computational complexity, and time-effectiveness is often sacrificed for greater precision. This makes it difficult for most of them to be employed for real-time body pose estimation, even when relatively powerful hardwares are available. Achieving real-time pose estimation becomes even more difficult when more people are involved or when hands and face pose estimation are needed, since the estimation time scales with the number of subjects in the image and with the number of keypoints to detect.

One of the more widely used open source libraries for pose estimation is OpenPose [41]. This library provides different pre-trained models for multi-person pose estimation, allowing to estimate in real-time either 15, 18 or 25 body keypoints, 42 hand keypoints and 70 face keypoints. For body joints estimation, the available models are trained either on the MS COCO or on the MPII dataset, while the training for hand pose estimation was done on the CMU Panoptic Hands dataset [75]. When using OpenPose models with also hand or face estimation, the inference speed becomes slower than the one of the models for body pose estimation, but the networks provided are still able to guarantee real-time

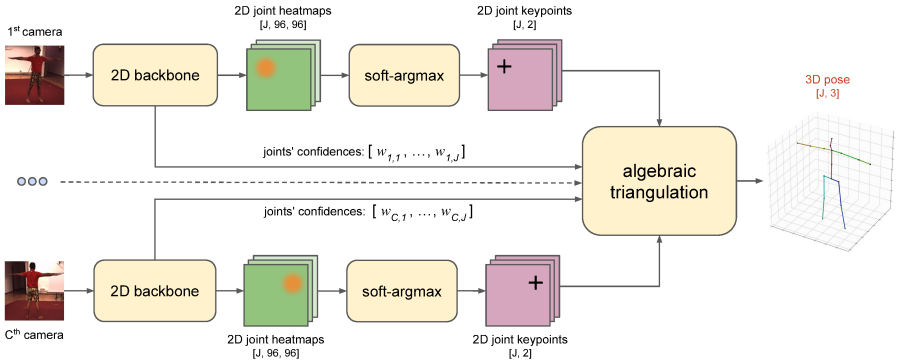
results when only one person is in the input image. The approach used by OpenPose is bottom-up, namely the trained network first estimates the position of all body parts and subsequently exploits part affinity fields to determine the connections between them, therefore separating different subjects.

A more recent open source library that is based on a similar bottom-up approach is OpenPifPaf [42], that uses composite fields to define associations between the detected keypoints. The body model used by OpenPifPaf is the one defined by the MS COCO dataset. In this library, also a network that estimates the COCO whole-body model, complete of hands and face joints, is available. However, while the OpenPifPaf network that provides the 23 body joints estimation can run in real time, the one for the whole-body pose estimation (i.e., body+hands+face) is much slower and is not really suitable for real-time applications. In particular, unlike OpenPose, this library does not provide dedicated networks for estimating only the hands (or only the face) joints, and since for the whole-body case a single network is used to estimate 133 joints, its complexity necessarily increases when compared to a network estimating only the body joints, leading to longer inference times.

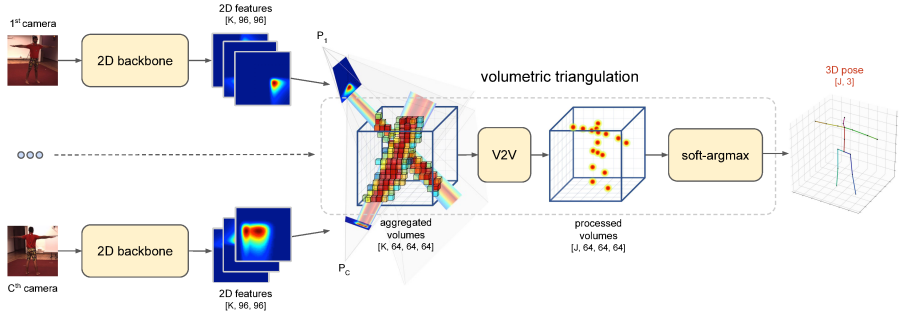
3.3 From 2D to 3D body pose estimation

The pose estimation architectures presented, namely OpenPose and OpenPifPaf, focus on predicting the body keypoints of people in an input image in terms of 2D coordinates. However, richer information about the body movement can be provided by a 3D estimation of the body keypoints. The task of 3D body pose estimation can be particularly difficult when approached directly, namely by regressing the 3D keypoints locations from images. More common techniques consist in extending the estimated 2D body keypoints into the three-dimensional space. One first method consists in using RGB-D sensors, such as Microsoft Kinect V2 or Intel RealSense cameras, which provide both color and depth information. By overlapping RGB and depth images through an appropriate affine transform it is possible to project the estimated 2D joints in 3D, where the third dimension is given by the depth value measured by the sensor. Another option is to use a multi-view setup, in which each camera independently estimates the 2D keypoints of the subjects in the scene and the different perspectives are then fused together by exploiting the extrinsic parameters of the cameras in the network. Using multiple cameras leads also to an advantage in terms of robustness of the estimated 3D poses, since it reduces the possibility of occlusion. However, to obtain good results, the camera network must be carefully calibrated in order to avoid large errors when fusing the different views. The works of [76–78] are all examples of state of the art methods for multi-view 3D pose estimation. The key idea in [76] is to use a matching algorithm to cluster the 2D poses detected in all views, where each cluster encodes the correspondences between keypoints of the poses of one subject across different views. From these encodings, the 3D body pose of each person is successively inferred. In [77], an algebraic

triangulation method and a volumetric triangulation method are proposed (Figure 3.2). The former is more immediate and consists in finding the 3D coordinates of a joint by solving an overdetermined system of equations on the homogeneous 3D joint coordinate vector, given the full camera projection matrices and the estimated 2D coordinates. The latter tries instead to address potential errors caused by inaccurate projection matrices by unprojecting the uninterpretable feature maps produced by the 2D backbone network into 3D volumes, which are then aggregated across different views to obtain 3D joint estimates. Finally, the approach proposed in [78] aims at minimising occlusion problems through an adaptive multi-view fusion method, which enhances the features in occluded views by leveraging those in visible views.



(a) *Algebraic triangulation*



(b) *Volumetric triangulation*

Figure 3.2: *Algebraic and volumetric triangulation methods proposed in [77] for multi-view 3D body pose estimation.*

Chapter 4

Skeleton-based human action recognition

As mentioned in the previous sections, there has been growing interest in the last years in skeleton-based action recognition methods. This is motivated by several factors:

- the intrinsic characteristics of pose-based features, which constitute a high-level information encoding body movements in a way that is independent of the surrounding context
- the development of new deep learning models particularly suited to learn from this type of graph-like information, such as graph convolutional networks
- the parallel ongoing improvement of body pose estimation techniques, on the goodness of which the outcome of skeleton-based recognition depends.

These points allow also to highlight the main challenges and critical aspects to be taken into consideration when dealing with skeleton-based action recognition methods. First of all, deep neural networks learning from skeleton features generally require a complete input, namely all the body joints should be correctly estimated. Thus, the increased importance in employing robust and accurate pose estimation algorithms. State-of-the-art body pose estimators performance is actually very high for what concerns the main body keypoints, but becomes less satisfying when hand pose estimation is required, especially when the hands are not closely framed by the camera. For this reasons, most skeleton-based action recognition models found in literature focus either on body keypoints or on hands keypoints, but rarely jointly on the whole-body skeleton model.

In this chapter, an overview on skeleton-based action recognition is given. First the main action recognition datasets that contain skeleton data are described. Subsequently, a theoretical introduction to the state-of-the-art deep learning methods used for skeleton-based action recognition is presented, with a focus on two main approaches: graph

convolutional neural networks and Transformer networks. In particular, in this thesis we focus on the former approach, investigating its performance and robustness in a human-robot collaboration scenario.

4.1 Action recognition datasets

In the literature there are several publicly available datasets used to benchmark action recognition models. For skeleton-based action recognition models, the interest is on datasets that also contain information about the 3D body joints of the people performing the actions. Among the main datasets for skeleton-based action recognition there are the MSR Action 3D dataset [79], the Northwestern UCLA dataset [80], the UTH-MHAD dataset [36] and the NTU RGB+D dataset [35]. These datasets mostly include everyday actions such as *walking, throwing, waving hands*, workplace activities such as *standing up, sitting down, picking up, carrying*, and sport-related actions such as *kicking* and *jogging*. They are all collected using Microsoft Kinect sensors, and the skeletons are extracted using the Kinect embedded skeleton tracker, that outputs the estimation of 25 body keypoints.

The most recent and also largest dataset among the ones mentioned is the NTU RGB+D dataset. This dataset contained originally 60 actions categories and was later extended with 60 additional actions to create the NTU RGB+D 120 dataset [81]. The dataset was acquired using three Kinect V2 cameras and contains RGB, depth and infrared information and skeleton annotations. The actions performed are grouped into three main categories: daily actions, medical conditions and mutual actions. There are available $56,880$ video samples for the first 60 actions and $57,600$ video samples for the additional 60 actions, since every action is performed multiple times by several different subjects, also covering changes in background and lighting conditions. Some examples of data contained in the NTU RGB+D dataset are shown in Figure 4.1. Many recent works use this dataset to train and test the proposed models for skeleton-based HAR [34, 44–50, 60], and in particular two standard cross-validation protocols are proposed for evaluation. In the first one, called cross-view, the actions collected by two cameras are used for the training phase while those collected by the third camera are used for the testing phase. In the second scenario, called cross-subject, the actions performed by 20 subjects are used for training while those performed by the remaining 20 are used for testing. Generally, a higher performance is achieved in the cross-view scenario, meaning that it is more complex to generalise on new subjects rather than on different viewpoints when performing skeleton-based action recognition.

As mentioned before, all the datasets considered in this section adopt a body model of 25 joints. In particular, no skeleton information related to the hands joints is contained in any of these, and there exist no public action recognition datasets for which the whole skeleton is annotated. A version of the NTU RGB+D dataset with skeletons annotations that include both hands and body joints, named NTU-X, was proposed in [82], but

Figure 4.1: *Sample frames of the NTU RGB+D dataset. First four rows show the variety in human subjects and camera views. Fifth row depicts the intra-class variation of the performances. The last row illustrates RGB, RGB+joints, depth, depth+joints, and IR modalities of a sample frame.*

unfortunately the extracted skeleton data were not made publicly available.

There exist instead specific datasets for hand gesture recognition, such as the DHG-14/28 dataset [83] and the First-Person Hand Action dataset [84], which include sequences of hand keypoints annotations. However, the gestures are performed with a single hand, which means that for each frame only one hand joints are annotated.

One thing that emerges from the reported description of the datasets is that, in general, the recognition of different types of movement, such as actions and gestures, is dealt with separately. However, there might exist scenarios in which the combined recognition of different categories of activities could be advantageous. For example, in the experimental part of this work, we focus on both action and gesture recognition. In particular,

we examine a human-robot collaboration context in which the considered actions are performed with the whole body, but for which the use of the hands is fundamental, either because they are used for communication or because the action involves the manipulation of specific objects. To investigate these aspects, we collected a novel human action recognition dataset containing body and hand pose annotations for several actions in a human-robot collaboration context. However, the size of the captured dataset is limited, and not sufficient to train a deep learning model from scratch; we then exploited a large-scale dataset to pre-train the network and learn more robust features. In the absence of datasets that contain annotations of both body and hand poses, we decided to re-annotate the NTU RGB+D dataset with the complete skeleton poses, similarly to what was done in the case of the NTU-X dataset. We choose the NTU RGB+D dataset since it is the largest available, containing a large variety of movements, ideal to learn general representations to be exploited in a transfer learning based approach.

4.2 Deep learning models for action recognition

In the literature, the best results in skeleton-based action recognition are achieved with two types of networks, namely graph convolutional networks (GCNs) and transformer networks. These models take as input sequences of skeleton-based features extracted for a person performing a specific activity. Sometimes, these features can be further processed before being fed to a neural network in order to train it to recognise actions. The type of processing depends also on the type of network used for the classification. For GCNs, for example, the joint features are fed to the network along with graph information, namely the links that connect the joints in the skeleton structure. In some cases, other features are computed starting from the joint coordinates, like the differential of spatial coordinates (*bones*), given by the difference between the coordinates of adjacent joints in the body graph, and the differential on the temporal dimension either of joints or of bones data, computed as the difference between the coordinates of a joint or of a bone respectively at two consecutive time instants .

4.2.1 Graph convolutional neural networks

A natural representation for skeleton data is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which each node $v_j \in \mathcal{V}$ corresponds to one skeleton joint, while an edge $e_{i,j} \in \mathcal{E}$ represents a skeleton bone, namely the spatial connection between two considered joints v_i, v_j. In the simplest case, a node feature consists in the 3D coordinates of the joint it represents, i.e. $v_j = (x_j, y_j, z_j)$. Graph neural networks (GNNs) [85] are a class of geometric deep learning models designed to perform inference on graphs, and can therefore be used to learn spatial and temporal relationships between joints of a subject performing an action. In particular, graph convolutional networks (GCNs) [86] extend the concept of standard convolutional neural networks.

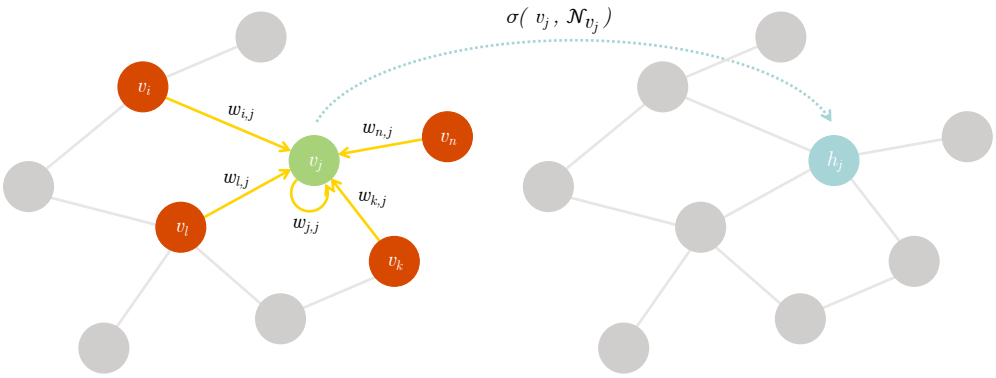


Figure 4.2: *Core operation of a graph convolutional neural network.*

The connectivity of a graph is described by the adjacency matrix $\mathbf{A}$, where the elements $a_{i,j} = 1$ if $e_{i,j} = (v_i, v_j) \in \mathcal{E}$, and $a_{i,j} = 0$ otherwise. Moreover, for each graph node v_j it is possible to define its neighbourhood $\mathcal{N}_{v_j} = \{v_i | e_{i,j} \in \mathcal{E}\}$. Then, the core operation in a graph convolutional neural network, computed for example for the first hidden layer of the network, can be written as:

$$h_j = \sigma\Big(\sum_{v_i \in \mathcal{N}_{v_j} \cup v_j} w_{i,j} \frac{v_i}{\sqrt{|\mathcal{N}_{v_j}|\,|\mathcal{N}_{v_i}|}} + b_j \Big) \tag{4.1}$$

where $w_{i,j}$ are learnable weights, b_j is a bias term and σ is a nonlinear activation function. The denominator term $\sqrt{|\mathcal{N}_{v_j}|\,|\mathcal{N}_{v_i}|}$ provides a normalisation factor that balances differences in node degrees. Figure 4.2 provides a graphical scheme of the above operation. For deeper network layers, it is sufficient to substitute v_i with the value of h_i computed at the previous hidden layer. It is also possible to enlarge the filter size, given by $|\mathcal{N}_{v_j}|$, by considering not just the 1-neighbours of each node but the k-nearest neighbours.

Equation 4.1 can be rewritten at graph-level as:

$$\mathbf{H}^{(k)} = \sigma\big(\tilde{\mathbf{A}}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}\big) \tag{4.2}$$

where $\tilde{\mathbf{A}} = (\mathbf{D} + \mathbf{I})^{-1/2}(\mathbf{A} + \mathbf{I})(\mathbf{D} + \mathbf{I})^{-1/2}$ is a normalised adjacency matrix (with self-loops). The matrix $\mathbf{D}$ is the degree matrix, namely a diagonal matrix with elements equal to the degree of each node. $\mathbf{H}^{(k)}$ is instead the matrix of stacked node representations h_j at layer k (each node corresponds to a row in the matrix).

The definition provided so far only covers the graph spatial relationships. In [47], spatio-temporal graph convolutional networks are introduced in the context of skeleton-based action recognition, and the temporal evolution is simply described by augmenting the graph with inter-frame connections, namely by connecting the same joints across consecutive frames (Figure 4.3). Then, by updating the neighbourhood definition to $\mathcal{N}_{v_j^t} = \{v_i^q | (v_j^t, v_i^t) \in \mathcal{E}, |q - t| \leq \lfloor \Gamma/2 \rfloor\}$, with Γ a parameter controlling the temporal

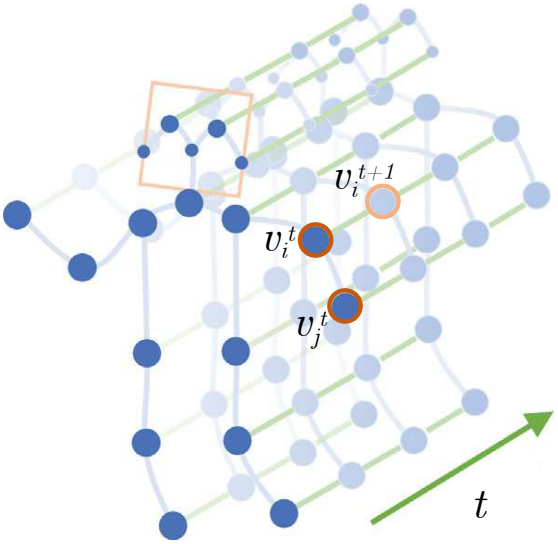


Figure 4.3: *Example of spatiotemporal skeleton graph given as input to a spatiotemporal GCN.*

range to be included in the neighbour graph (temporal kernel size), the implementation of a GCN defined above remains valid and also embeds the temporal structure of the graph. Moreover, in [47] also the definition of adjacency matrix is updated, namely $\mathbf{A}$ is expressed as the sum of different matrices $\mathbf{A}_\ell$, with the aim of partitioning the graph in significant components to improve the learning capability of the network. Finally, it is worth noticing that, since the degree of the graph nodes is not a fixed value and the neighbours do not have a fixed spatial order, the weights of the network cannot in general be shared as done with standard convolutional networks. The solution adopted in [47] consists in partitioning the neighbourhood of a node into a fixed number of subsets on which the weighting function is then defined.

Graph neural networks can be used for different classification tasks, node-wise, edge-wise or graph-wise. In the case of skeleton-based action classification, graph neural networks are employed for graph classification.

Despite the small input dimension of the spatiotemporal skeleton graph when compared to a sequence of full RGB frames, GCNs still entail a heavy computational cost. Moreover, the receptive field of both the spatial and the temporal graph are predefined by the adjacency matrix. The work of [34] addresses both of these problems by introducing Shift-GCNs. The proposed model consists of a graph convolutional network that employs a non-local spatial shift operation and an adaptive temporal shift operation. Taking inspiration from shift convolutions done in CNNs [87], a graph shift convolution consists in a graph shift operation done within the receptive field of each node, followed by a point-wise convolution operation.The main idea of the shift graph operation is shifting the features of the neighbour nodes to the current convolution node. The spatial shift operation is non-local when the receptive field of a node coincides with the whole graph. Non-local shift allows the network to better learn the relationships between body joints

during movements, implying that the most useful connections in terms of action recognition do not necessarily coincide with the physical ones derived from human anatomy. The adaptive temporal shift is instead obtained by adding a learnable temporal shift parameter that determines the temporal kernel size. The shift graph convolutions substitute graph convolutions in GCNs and allow to make the computational complexity of the network around 6.5 times lower when compared to other spatiotemporal GCN models.

The lightweight nature of the Shift-GCN has encouraged us to choose it as the model to conduct our experiments on skeleton-based action recognition in a human-robot collaboration setting. Indeed, we have already highlighted the importance of recognising actions in real-time in such scenario, and a fast model for classification can help improving the overall computational times.

4.2.2 Transformer networks

The Transformer architecture is another type of deep learning model that has allowed to achieve state-of-the-art results in different fields, starting from natural language processing and with rising interest also in computer vision. Transformer networks fully rely on attention mechanisms to model sequential information. Differently from recurrent neural networks (RNNs) and long-short term memory networks (LSTMs), that work well with short data sequences, transformers allow to learn also long-term dependencies. Generally, RNNs and LSTMs employ a context layer to store hidden states vectors and network outputs obtained at the previous time step in order to keep track of the sequentiality of the inputs. The attention mechanism used in Transformer networks aims instead at exploiting the information gathered at *all* previous time steps [88].

Introduced in [89], the original Transformer architecture has an encoder-decoder structure (Figure 4.4a). Attention is defined as a function over matrices containing queries $\mathbf{Q}$, keys $\mathbf{K}$ of dimension d_k and values $\mathbf{V}$ of dimension d_v (Figure 4.4b). More in detail, it gives as output a weighted sum of the values, where the weights are computed as a compatibility function between a query and the corresponding key, namely

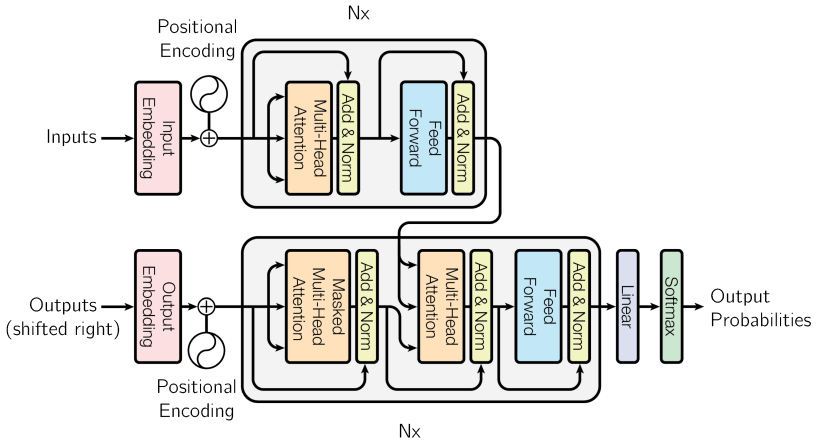
$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\mathbf{Q}\mathbf{K}^T/\sqrt{d_k})\mathbf{V} \tag{4.3}$$

where $softmax(z_i) = e^{z_i}/\sum_{j=1}^{|\mathbf{z}|} e^{z_j}$. Moreover, multi-head attention is defined as

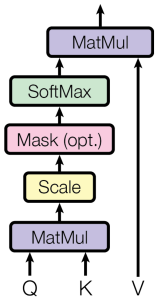
$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [head_1, ..., head_h]\mathbf{W}^O \tag{4.4}$$

where $head_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$ and $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ and $\mathbf{W}^O$ are learned parameters matrices. The first three project queries, keys and values to d_k, d_k and dv dimensions respectively, for h different times. Multi-head attention consists of several attention layers running in parallel, as shown in Figure 4.4c.

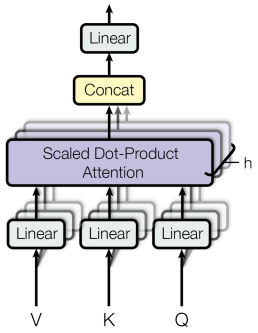
In the encoder-decoder attention layers, the queries come from the previous decoder layer, and the memory keys and values come from the output of the encoder. In addition, both encoder and decoder are provided with self-attention layers, in which queries,



(a) *Transformer architecture*



(b) *Scale dot-product attention*



(c) *Multi-head attention*

Figure 4.4: *Core architecture of a transformer network and the related attention and multi-head attention operations.*

keys and values all come from the output of the previous layer of the encoder or decoder respectively. This allows to properly weight previous inputs and outputs in new predictions, independently from the distance of the new input from the past elements of the sequence.

In [60], a spatial temporal transformer network for skeleton-based action recognition is introduced, consisting of two modules. A spatial self-attention module computes for a given frame at a time step t a query vector, a key vector and a value vector by applying trainable linear transformations to the node features, where the transformations are shared across all nodes. Then, for all pairs of body joints, attention weights $\alpha_{i,j}$ are computed as a query-key dot product and are used in the weighted sum of the values as in Equation 4.3. The process is repeated to obtain multi-head attention. In this way, the attention weights encode the learned correlations between pairs of body joints. The second module is the temporal self-attention module, which learns the dynamics of each

joint separately, along all the time frames considered.

Transformers and GNNs It is interesting to notice that Transformers can be seen as graph neural networks operating on a fully connected graph and with attention weights. In particular, in Equation 4.1 we can substitute $\sum_{v_i \in \mathcal{N}_{v_j} \cup v_j} w_{i,j} \frac{v_i}{\sqrt{|\mathcal{N}_{v_j}||\mathcal{N}_{v_i}|}} := \mathbf{m}_{\mathcal{N}_j}$ with an attention weighted sum of the neighbours to obtain a Graph Attention Network (GAT) [90], namely

$$\mathbf{m}_{\mathcal{N}_j} = \sum_{v_i \in \mathcal{N}_{v_j} \cup v_j} \alpha_{i,j} h_i, \qquad \alpha_{i,j} = \frac{\exp\left(a^T [\mathbf{W}h_i, \mathbf{W}h_j]\right)}{\sum_{v_k \in \mathcal{N}_{v_j}} \exp\left(a^T [\mathbf{W}h_k, \mathbf{W}h_j]\right)} \tag{4.5}$$

where a is a trainable attention vector and $\mathbf{W}$ is a trainable matrix. Moreover, a multi-head attention mechanism can be employed by redefining $\mathbf{m}_{\mathcal{N}_j}$ as:

$$\mathbf{m}_{\mathcal{N}_j} = [a_1, a_2, \ldots, a_K], \quad a_k = \mathbf{W}_k \sum_{v_i \in \mathcal{N}_{v_j} \cup v_j} \alpha_{i,j,k} h_i \tag{4.6}$$

Many recent action and gesture recognition works have proposed graph convolutional networks models with attention mechanisms to improve the performance, such as the 2s-AGCN model [50] and the STA-GCN model [91].

If considering a GAT receiving a fully connected graph as input, the neighbourhood of each node coincides with the whole graph. In this case, it is possible to see that Equation 4.6 and Equation 4.4 apply the same operation to their input, namely the basic transformer layer is exactly equivalent to a GNN layer using multi-headed attention.

This might give an intuition on why Transformers, as a particular case of graph neural networks, are well suited to deal with graph-like data such as the skeleton sequences used for human action recognition.

Chapter 5

A general framework for collaborative action recognition

Collaboration between humans and robots implies a shared workspace in which common tasks are carried out jointly. Human-robot collaboration has become of particular interest in manufacturing and industrial assembly, since the use of collaborative robots allows to increase productivity and improve worker comfort. For example, in a human-robot collaboration scenario in which an assembly procedure is carried out, tasks that are repetitive and those that require a greater physical effort can be assigned to the robot, while those requiring planning, reactivity and critical judgement are usually entrusted to human operators. Moreover, a collaborative setting allows to speed up the completion of an operation, as man and robot can work simultaneously without the need for the robot to stop whenever the operator approaches it.

In this context, the recognition of human actions and gestures is useful both for security and for communication purposes. Indeed, it allows the robot to understand the behaviour of the people it is collaborating with or to receive directions from them and to act accordingly.

In this chapter we focus on action recognition for human-robot collaborative assembly tasks. In particular, the key elements of the proposed framework for skeleton-based action and gesture recognition for HRC are outlined. First, a description of the collected dataset is reported, focusing on the actions and gestures selected for a general human-robot collaboration context. Then, the choice of the adopted skeleton model is explained. Finally, an overview on the transfer learning technique and on fusion methods for deep learning models is reported, highlighting their advantages and the motivations for their employment in the analysed scenario.

5.1 *IAS-Lab Collaborative HAR* dataset design

Starting from the issues outlined at the end of Section 2.2, a framework is proposed to recognise actions and gestures in a generic human-robot collaboration scenario. In particular, we focus on the design of a set of general HRC-related actions to compose our new dataset for collaborative HAR. Then, we define a skeleton model optimal to recognise the defined activities.

5.1.1 Human-robot collaboration actions and gestures

As mentioned in Section 2.2, there exist different recurring actions that are common to most human-robot collaborative assembly scenarios. In order to obtain a general model that could be used in different contexts, independently from the specific task carried out, we try to identify these actions to create a new dataset for HAR in human-robot collaborative settings (denoted in the following as *IAS-Lab Collaborative HAR* dataset). In this way, a model trained on these actions could be employed in a generic HRC scenario to improve the communication and collaboration between the robot and the human operator. The selected actions can be grouped in four main sets:

1. *Spatial movements*: WALK, REST
2. *Assembly actions*: PICK, PLACE, SCREW, INSERT/JOIN, HAMMER
3. *Collaborative gestures*: HAND TO, REQUIRE
4. *Communication gestures*: STOP, OK/CONFIRM, UP, DOWN, FORWARD, BACKWARD, LEFT, RIGHT, POINT

The first group refers to general movements that a person can do in the workspace. The generality of the context leads to define only two actions in this category, since WALK includes all movements that allow the worker to move around in the robot's workspace, while REST indicates that the human operator is not working. These actions are independent from the coordinates of the surrounding space in order to be applicable to different contexts. In specific settings, usually actions such as approaching or moving away from *something* were defined [17]. However, to recognise them an environment coordinate reference system should be previously set.

In the second group the most common assembly actions are defined. It was observed that assembly operations can be generally split into some base actions, each usually associated with a tool that depends on the applications. However, while the tools change from case to case, the base actions recur in many of the analysed scenarios and are the ones listed in group (2.).

The third group includes *collaborative gestures*, namely signals that the human can do to collaborate with the robot in exchanging tools or objects. The fourth set of actions groups instead gestures that the human operator can perform to communicate an

Action	Works	Action	Works
WALK	[5, 17]	STOP	[17–22]
REST	[17, 23, 24]	OK/CONFIRM	[18, 21, 22]
PICK	[6, 17, 19, 21, 23–28]	UP	[18, 21, 22]
PLACE	[19, 21, 27–29, 33]	DOWN	[18, 21, 22]
SCREW	[25, 27–31]	FORWARD	[22]
INSERT/JOIN	[6, 29, 33]	BACKWARD	[20, 22]
HAMMER	[23, 24, 30, 31]	LEFT	[18, 21, 22]
HAND TO	[6, 17, 26, 32]	RIGHT	[18, 21, 22]
REQUIRE	[29, 32]	POINT	[17]

Table 5.1: *Review of works about human-robot collaboration in which the actions of the* IAS-Lab Collaborative HAR *dataset appear.*

instruction to the robot, including directions of movement and signals of confirmation or halting.

Table 5.1 lists the works in which the selected actions appear. Some of the mentioned works include declinations of the reported action with respect to a specific tool or position in space.

The design of the set of actions to be recognised was made in order to include movements that are as generic as possible and that can in general be distinguished without pairing them to objects or tools and without setting a world coordinates system to locate the human operator in the workspace. It can be noticed that many actions require an active use of hands and that the hands movements are potentially discriminative in the recognition of the performed action, especially for what concerns gestures and assembly actions.

Some examples of actions belonging to the *IAS-Lab Collaborative HAR* dataset are reported in Figure 5.1. These show different subjects performing the designed actions and gestures. Moreover, they also highlight the fact that, in many cases, an action or gesture contained in the dataset could be performed by different people in several different ways, and that two different actions might, on the contrary, be executed in similar ways. For example, the gesture BACKWARD, shown in the last row of the mentioned figure, is performed by four subjects in four different ways: using either one or two hands and with different hands positions and orientations. In the case of the two gestures STOP and BACKWARD, shown in Figure 5.1g and Figure 5.1i respectively, the motion of the hand used is instead very similar. It can be observed that not defining in our dataset a univocal dictionary according to which the chosen actions and gestures should be performed makes the recognition task particularly challenging.

(a) *forward* (b) *screw* (c) *hammer* (d) *up*

(e) *hand to* (f) *insert/join* (g) *stop* (h) *right*

(i) *backward* (j) *backward* (k) *backward* (l) *backward*

Figure 5.1: *Examples of frames from the* IAS-Lab Collaborative HAR *dataset. The last row shows different variations of the same gesture. Note that, without loss of generality, directions are given from the point of view of the subject.*

5.1.2 Skeleton model

To recognise the actions and gestures listed in Section 5.1.1, skeleton features are employed. Since a large part of these actions requires the use of the hands, we assumed that body joints alone are not sufficient to recognise them accurately. For this reason, the skeleton model employed also includes hands joints. Indeed, they are deemed fundamental to recognise gestures, but are also considered important to discern actions that involve tool or object manipulation, especially since object recognition is not employed. It has already been highlighted that the estimation of hand joints is more challenging with respect to the one of the main body keypoints, especially when the visual sensor used for data acquisition frames a full body figure, which is in this case necessary since some of the actions involve full body movements. Because of this, in our skeleton model we consider only a subset of the most significant hands keypoints. In particular, the

choice of hand joints is limited to the wrist, three joints each for the thumb, index and middle finger and two joints for the ring finger, namely for each hand, 12 keypoints are considered. The rest of the joints are omitted, i.e., the ones of the little finger and of the tip of ring finger, the distal phalangeal joints of the index and middle finger and the carpometacarpal joint at the thumb's base. Indeed, these are considered less useful for the action recognition process as they are not primarily responsible for the hands movements included in our set of actions. Moreover, many of them are also among the most difficult to estimate, as will be shown in the experimental part of this work (see Section 6.3). We observe that removing them also helps reducing the complexity of the problem. For what concerns the body, some keypoints such as the ones corresponding to eyes, ears and feet are not considered either, leading to a 15 joints body model comprising head, neck, shoulders, elbows, wrists, pelvis, hips, knees and ankles. The models of the described body and hands skeletons are shown in Figure 5.2.

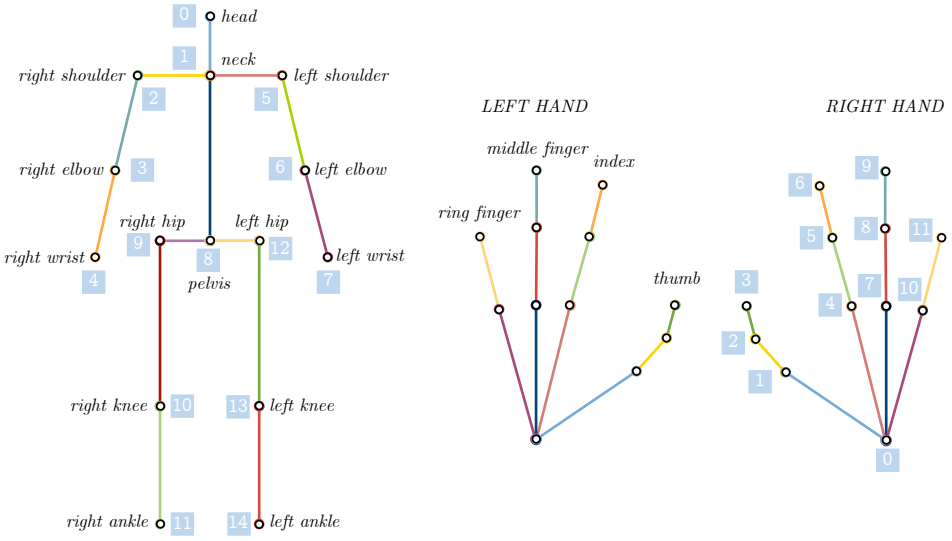


Figure 5.2: *Body and hands skeleton model used in the collaborative setup.*

5.2 Learning methods for the proposed framework

A deep learning model requires a lot of training data samples to properly learn to classify the actions. In many circumstances it can however be challenging to collect a sufficiently large dataset. Moreover, when the input data is complex, it can be difficult to find an optimal balance between the network's hyperparameters to achieve good results. In the following, we present two methodologies employed to deal with these problems in

the proposed framework: transfer learning and ensemble averaging.

5.2.1 Transfer learning

Transfer learning is a deep learning technique whose rationale is to exploit the hierarchy of features learned by a neural network to share representations across different tasks. More in detail, the features learned by a deep neural network are in general coarse features for what concerns the first layers of the network, and become more and more specialised with respect to the learned task in deeper network layers. This means that two networks trained to perform action classification on different sets of actions starting from the same type of input data are likely to learn similar features in the first layers of the network, while the features learned in deeper layers will be more related to the specific set of actions chosen.

Transfer learning consists in exploiting the features learned by a network trained on a certain dataset to perform a similar task on a new dataset. This can be done by *freezing* the first layers of the trained network and by *fine-tuning* only the last few layers on the new dataset, eventually changing the shape of the last layer according to the new task, as illustrated in Figure 5.3. Usually, the original network is trained on a large and varied dataset in order to learn robust representations. Then, the last layers of the network are re-trained, i.e., fine-tuned, on a smaller dataset. This allows to exploit the robust features learned by the network on new examples, so that a relatively small amount of data is sufficient to learn the new task. This also yields faster training times on the new dataset, since only few layers of the network need to be trained.

This learning technique applies well to action recognition for a specific task such as, for example, human-robot collaboration. Indeed, large action recognition datasets like the ones described in Section 4.1 contain everyday actions that mostly do not relate to a HRC scenario. Works on action recognition for human-robot collaboration often collect small datasets containing the execution of the specific actions they want to recognise,

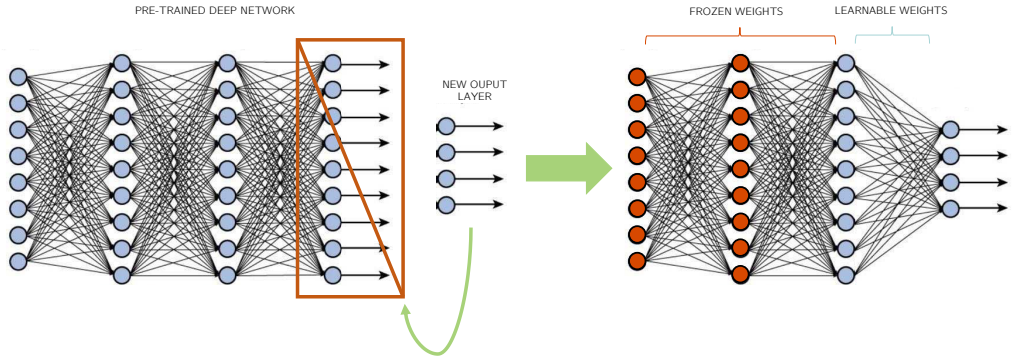


Figure 5.3: *Transfer learning pipeline.*

and use transfer learning to fine-tune the last layers of a network pre-trained on a large action recognition dataset [6, 18, 28, 30, 33]. As already anticipated in Section 4.1, here a similar approach is proposed, namely a neural network is trained on the NTU RGB+D dataset and the last layers are then fine-tuned to recognise the actions of our newly collected dataset described in Section 5.1.1.

5.2.2 Modularity and ensemble averaging: score-level fusion

By analysing the chosen actions it can be noticed that, in certain cases, either the body or the hands could be sufficient in recognising them, while in other cases it is more suitable to observe the whole body movement to understand which action is being performed. To understand better the role of hands and body in the recognition of the chosen actions, besides training a network on full skeleton data (*body+hands*), other networks are also trained to perform the same task but starting from different subsets of joints. In particular, we consider a network trained only on the 15 body joints, one trained only on the 24 hands joints, one trained on the 12 left hand joints and one on the 12 right hand joints.

Moreover, the results obtained with the different modules defined above can be ensembled together to improve the outcome of the classification. More in detail, the network trained to recognise actions starting from the 15 body joints can be fused either with the one trained using the joints of both hands or with the two networks each recognising actions starting from a single hand. One way of ensembling the information coming from different networks is at score-level, namely the $softmax$ scores given as output by the last layer of each network can be summed together, weighted by apposite values, to obtain a fused score on which the $argmax$ is then computed to get the predicted action label. The aim is to compare the results obtained with these ensembles with the ones obtained by training a single network on the full skeleton model.

The results obtained by fusing together different modules are expected to improve the accuracy when compared with the ones of a single model. Indeed, it has been proved that ensembling different neural networks trained on the same task can help reducing overfitting problems and improving the generalisation capability of the model [92]. This in general leads to higher classification accuracies, in particular when the networks are trained using different training data.

Chapter 6

Experiments

To test the effectiveness of the action and gesture recognition framework for human-robot collaboration presented in Chapter 5, all the elements described so far were put together in an experimental setting. The experiments were all carried out at the IAS-Lab of the Department of Information Engineering, University of Padova.

In this Chapter we outline the experimental steps followed to test the proposed action and gesture recognition framework. First, the 3D body pose estimation algorithm employed in the experiments is described. Then, we outline the process used to estimate full skeletons on the NTU RGB+D dataset used for pre-training, focusing the attention on the choice of the pose estimation algorithm employed. The technical aspects concerning the collection of our dataset, named *IAS-Lab Collaborative HAR* dataset, are successively reported, and a brief description of the Shift-GCN model used for action classification is presented, highlighting the modifications introduced to adapt this model to our data. Moreover, the data preparation process is outlined, concerning both the NTU RGB+D dataset used for pre-training and the *IAS-Lab Collaborative HAR* dataset collected. Then, the pre-training results on the NTU RGB+D dataset are discussed. Finally, the transfer learning and networks fusion results obtained on our dataset are presented.

6.1 Estimation of the 3D body pose

As mentioned in Chapter 3, the fundamental preliminary step to perform skeleton-based action recognition is to estimate the human pose in the 3D space. To do so, a system already implemented in the laboratory was exploited for our experiments. This consists in the cascade of two ROS[1] nodes. The first node receives a stream of RGB image data from a camera and, if a person is framed by the sensor, it estimates its 2D skeleton joints using the open source OpenPose library [41]. When running the node, it

[1]Robot Operating System, https://www.ros.org/

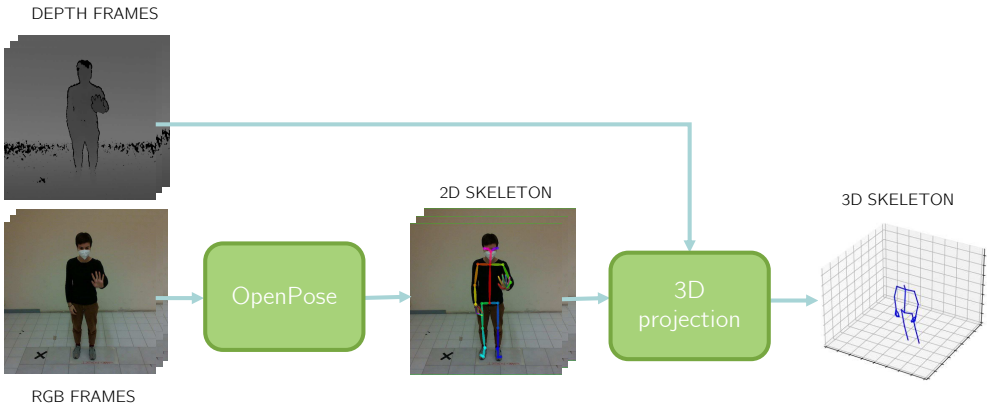


Figure 6.1: *Block scheme of the 3D body pose projection algorithm.*

is possible to choose wether to estimate body keypoints, hands keypoints, face keypoints or any combination of these three sets: we are interested in particular in body and hands joints. OpenPose provides real-time body pose estimation, and it was observed to work at around 6.5 FPS when estimating body and hands joints of a single person. The stream of estimated 2D skeleton joints is then published on a dedicated ROS topic. The second node reads the topic containing the 2D skeleton information along with the stream of depth frames coming from the vision sensor and the intrinsic parameters of the RGB camera c_x, c_y, f_x, f_y. For what concerns the depth frames, only the region that aligns with the RGB frames is considered. In this way, each estimated 2D keypoint (x_p, y_p) can be projected in the 3D space by taking as third coordinate z the value of the corresponding point in the depth frame. More in detail, the algorithm does not take the depth value at (x_p, y_p) coordinates directly, but considers a small window of pixels around that point and computes the median value of the pixels inside the window. Since the depth values measure the distance of a point from the camera reference system in millimetres, these are first converted to meters through a simple multiplication factor. Then, the intrinsic parameters of the RGB camera and the depth coordinate z in meters are exploited to convert the (x_p, y_p) coordinates from image pixels to coordinates in the 3D space measured in meters, using the geometric transformations:

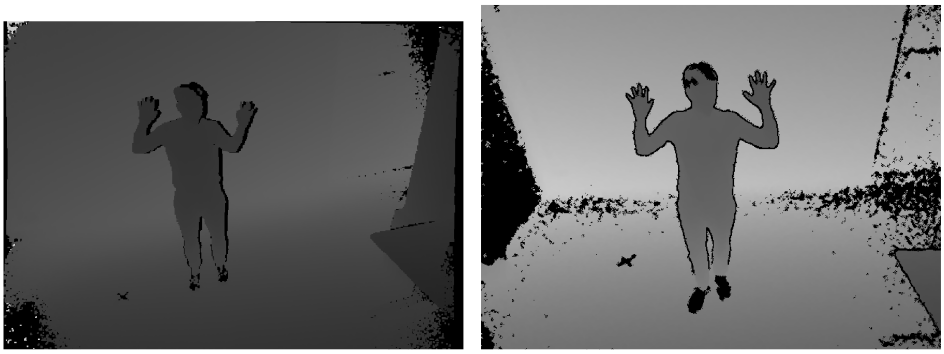
$$x = \frac{x_p - c_x}{z} f_x, \quad y = \frac{y_p - c_y}{z} f_y \tag{6.1}$$

The second node then proceeds to publishing the estimated 3D coordinates to another ROS topic. The block scheme summarising this procedure is shown in Figure 6.1. We notice that this projection process works well for body joints but becomes less stable for what concerns hands joints, especially for the ones located on the fingertips or on the hand contour, namely on the area perceived at the body shape boundary by the camera. Indeed, for these keypoints the values evaluated by the depth sensor are less robust. In addition to this, a window centered on a boundary body joint is likely to include

a relevant portion of background pixels. These can compromise the estimated depth value and make the points appear to be projected into the background. To avoid this behaviour, in the case of hands keypoints the projection algorithm was slightly modified so that, within the window, the lowest nonzero depth value detected is taken into account instead of the median. We point out that we can implement this adjustment because the scenario in which we acquire the data is free of obstacles that could cause occlusions and the subject performing the actions are standing in front of a uniform background, i.e., a white wall.

We highlight that the system just described has so far been implemented only for a single camera system, which entails not being robust against occlusion. Employing, for example, a multi-view camera system could help obtaining better estimated features and, in general, a more robust baseline system for 3D body pose estimation.

For what concerns the vision sensors used, in the laboratory there is availability of Microsoft Kinect V2 sensors and Intel RealSense L515 sensors. Both sensors measure depth using time-of-flight techniques, namely by timing a light pulse emitted by the sensor and reflected by the framed scene back to the sensor. However, RealSense cameras employ newer technologies and their depth resolution is higher compared to the one of Kinects, resulting to be more precise in detecting object contours. For this reason and due to the considerations on the robustness of the 3D projection reported above, a RealSense L515 sensor was employed in the experiments performed. Figure 6.2 shows a comparison of two depth frames captured with a Kinect sensor and RealSense L515 sensor respectively, from which it is possible to observe the significant difference between the precision of the two. The main drawback of the RealSense cameras used is that they can only be employed in artificially illuminated environments due to their high sensitivity to sunlight, which compromises the correct functioning of the depth sensor. Figure 6.3 shows two examples of depth frames captured with the RealSense L515 camera in presence of sunlight.



(a) Microsoft Kinect V2 depth (b) Intel RealSense L515 depth

Figure 6.2: *Comparison of the depth frames captured with different vision sensors.*

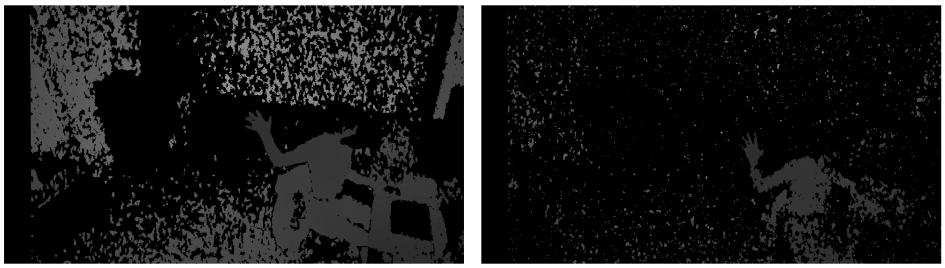


Figure 6.3: *Examples of noisy depth frames captured with a RealSense L515 in presence of soft (left) and direct (right) sunlight. Figure 6.2 shows instead an example of depth acquired in absence of sunlight.*

6.2 *IAS-Lab Collaborative HAR* dataset collection

To accomplish our goal we collect a new dataset, the *IAS-Lab Collaborative HAR* dataset, containing data of people performing the set of actions for human-robot collaboration scenarios described in Section 5.1.1. To obtain the 3D skeleton sequences of the actions performed, we exploit the system presented in Section 6.1. In particular, for the reasons explained previously, we use a single Intel RealSense L515 sensor, placed at a distance of about 2.5 m from the subjects, framing them in full figure. The 18 actions presented in Section 5.1.1 were performed by 6 different subjects, and each subject was asked to repeat the action for 5 times, for a total of $18 \times 6 \times 5 = 540$ samples of skeleton sequences. Each recording lasted about 5 s, during which sequences of about 32 skeletons were collected. Some example of the actions executed by one of the subjects, with the corresponding OpenPose skeleton estimations, are shown Figure 6.4. During the acquisition of the dataset, the subjects involved were only told the name of the actions and gestures reported in Section 5.1.1 and did not receive any further instruction on how to perform them. This allowed to increase the variability of the dataset, since many subjects performed the same actions in different manners. Moreover, this choice is also in line with the fact that we want to recognise actions and gestures executed as naturally as possible, without creating a rigid dictionary of movements and having to give many specific instructions to users. This can help achieving an easier, more immediate and natural communication system between man and robot. Figure 6.5 provides an example of variability, showing different possible executions of the same action. Finally, we highlight that subjects $1, 2, 4, 5, 6$ are right handed, while subject 3 is left handed. In the reported experiments, the data collected for the first five subjects constitute the training set, while data related to the sixth subject are used for validation.

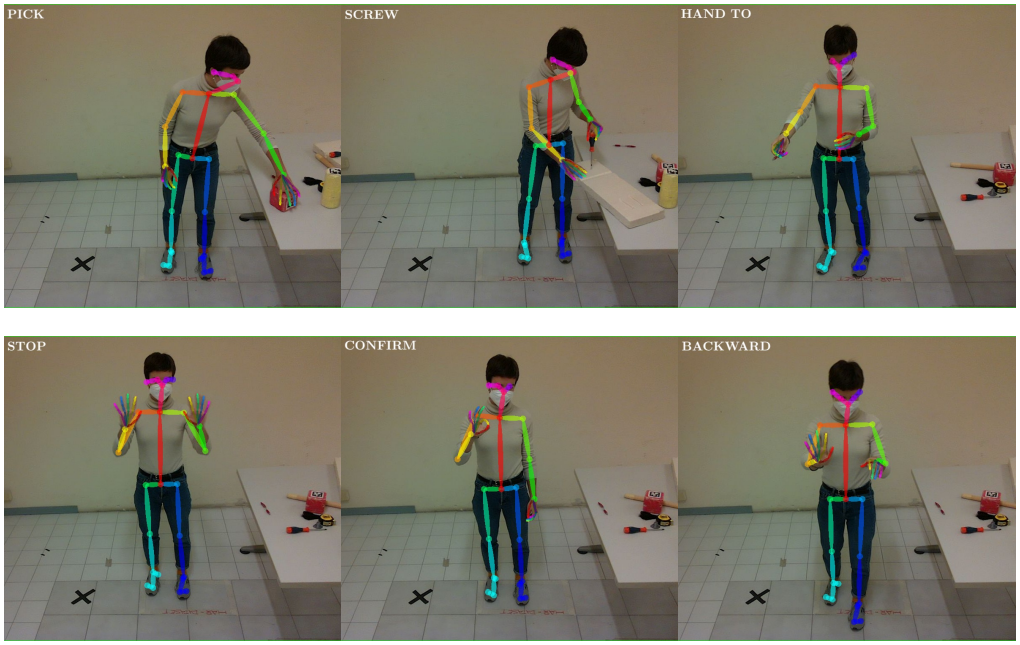


Figure 6.4: *Examples of frames from the* IAS-Lab Collaborative HAR *dataset, where the OpenPose skeleton estimation is highlighted.*

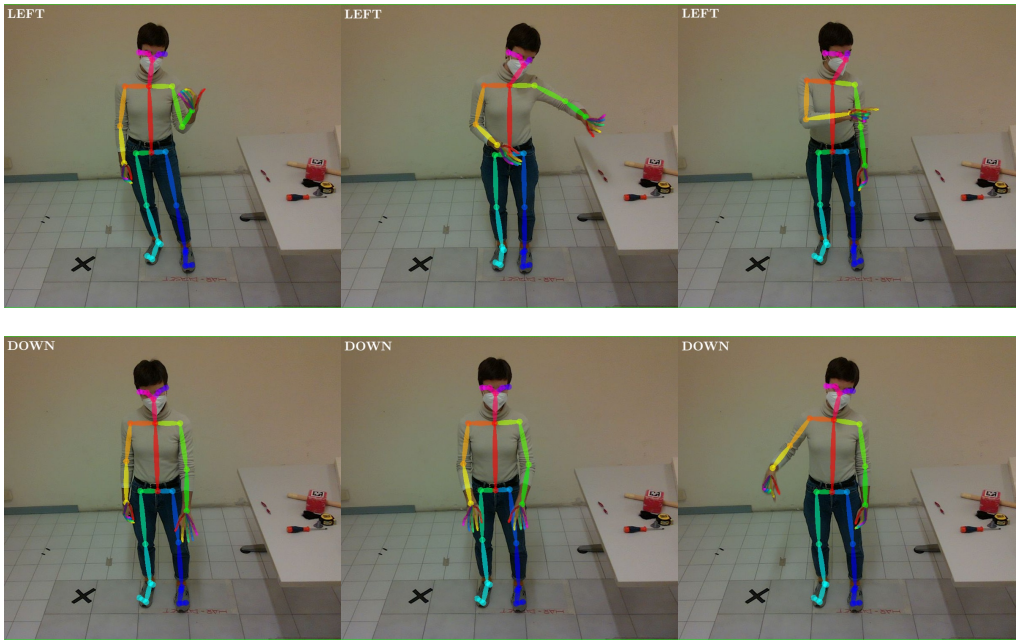


Figure 6.5: *Variations in the execution of gestures* LEFT *(top) and* DOWN *(bottom). In the first case it is assumed w.l.o.g. that direction is given from the person's point of view.*

6.3 Skeleton estimation in the NTU RGB+D dataset

The NTU RGB+D dataset described in Section 4.1 was employed to pre-train our networks. This dataset contains RGB frames, depth frames and also the skeleton annotations of the subjects performing the actions. However, the topology of the skeletons included in the dataset is different from the one estimated by OpenPose for what concerns some body joints (such as the ones along the spine) and it does not include hands joints. For this reason, the algorithm described in Section 6.1 was employed to re-estimate full skeleton data for the NTU RGB+D dataset, including both body and hands keypoints. Since this procedure was carried out offline, it was possible to compare the two pose estimation libraries presented in Section 3.2, namely OpenPose and OpenPifPaf. The only difference between these two estimators is that OpenPose returns two additional body joints with respect to OpenPifPaf, namely the neck and pelvis points, but these can be easily inferred also for OpenPifPaf outputs starting from the estimated keypoints of the shoulders and hips. Given this assumption, the outputs of these methods can be considered equivalent and can be compared. Since the full NTU RGB+D 120 dataset is very large and considering the time required for the skeleton estimation process (in particular in the case of OpenPifPaf, that runs at around $1\,FPS$ when the whole body estimation is required), our choice was to only use the first version of the dataset, containing 60 actions. From this group we further excluded 10 actions that involve the mutual interaction between two subjects, since the actions considered for our human-robot collaboration scenario are assumed to be performed by a single person. Starting from the extracted skeletons, we limit our analysis to the joints included in the body model described in Section 5.1.2. To compare the performance of OpenPose and OpenPifPaf, the following metrics were taken into account:

- the mean number of skeletons extracted for each video sample analysed (*MSPS, mean skeletons per sample*)
- the mean percentage of undetected joints for each considered frame (*MZJ, mean zero joints*)
- the mean number of frames per video for which a complete skeleton was extracted (*MFSF, mean full skeleton frames*)

The analysis was made for four different subsets of joints, namely for the full skeleton model, for body joints alone, for hands joints alone and for each hand considered singularly, and the results are reported in Table 6.1. The first observation that can be made on the reported results is that, in all cases, only a small percentage of the original data is usable for training. This follows from the fact that a valid input for a graph neural network should not have missing entries. Indeed, if for example all the undetected joints were set to $\mathbf{0}$, this would be equivalent to positioning the undetected body parts all at the origin of the reference system, deforming the body shape and inevitably compromising

Joints group	metric	OpenPose	OpenPifPaf
	MSPS	85.51	77.69
Whole body	MZJ	23.75%	22.15%
	MFSF	1.46%	5.025%
Body	MZJ	2.73%	4.2%
	MFSF	32.95%	22.79%
Hands	MZJ	21.07%	17.94%
	MFSF	2.26%	8.89%
Left hand	MZJ	11.44%	9.55%
	MFSF	8.87%	25.73%
Right hand	MZJ	11.27%	9.52%
	MFSF	11.09%	27.37%

Table 6.1: *Performance comparison of OpenPose and OpenPifPaf on skeleton estimation on the NTU RGB+D dataset.*

the network performance. Moreover, it can be observed that the overall performance of the OpenPifPaf model is better with respect to OpenPose, in spite of being slower, especially when hand pose estimation is involved. Indeed, the percentages of valid skeletons extracted with OpenPifPaf are higher in all cases except when only the body joints are considered. However, also in the case of body joints, the estimates obtained with OpenPifPaf seem to be more robust when compared to the ones of OpenPose, despite being a smaller number.

For the reasons just listed, our final choice of the data to be used for pre-training falls on the skeletons extracted with OpenPifPaf. Table 6.2 reports the number of valid skeleton sequences extracted by the OpenPifPaf model from the NTU RGB+D dataset, where we consider valid sequences with at least 3 full skeletons.

Joints group	Valid sequences
Whole body	9280
Body	17817
Hands	12320
Left hand	21689
Right hand	22433

Table 6.2: *Numbers of valid skeleton sequences extracted by OpenPifPaf from the NTU RGB+D dataset (out of* 47400*).*

Figure 6.6: *Examples of skeletons estimated with OpenPifPaf on the NTU RGB+D dataset.*

A further analysis was made on the full skeleton data (with 23 body joints and 21 joints for each hand) detected by OpenPifPaf, namely a list of the 20 most undetected joints was compiled. This turned out to contain all joints belonging to the two hands, including most of the joints of the little and ring fingers and the fingertips of the other fingers. This results supports our choice of the hand skeleton model defined in Section 5.1.2, in which many of the most undetected joints are excluded, helping to improve the likelihood of detecting all the necessary skeleton keypoints. An example of body poses estimated with OpenPifPaf on the NTU RGB+D dataset is reported in Figure 6.6

It is worth highlighting that a similar operation of extracting new, more complete skeletons, was made in [82] for the NTU-X dataset. In the mentioned article, more advanced techniques were employed to estimate the skeletons, leading to better results. It was in fact shown that the introduction of hands joints led to an overall improvement in the action recognition results. Indeed, the restricted number of valid skeleton sequences extracted with our method can represent a limit to our approach and we could expect that better results, with respect to the ones presented in the following sections, could derive from the use of better-extracted skeleton sequences.

6.4 Shift-GNC architecture

To perform action recognition we employ the Shift-GCN model [34] already presented in Section 4.2.1. This architecture achieves state-of-the-art results on the original NTU RGB+D dataset (e.g., 95.1% accuracy in the *cross-view* setup) and the code for the original network is made available by the authors [2]. Moreover, it has the advantage of being a lightweight model that requires a lower computational cost compared to other state-of-the-art graph convolutional networks models developed in literature - reportedly

[2] https://github.com/kchengiva/Shift-GCN

at least 6.5 times lower - thanks to the introduction of graph shift convolution operations. This aspect is particularly relevant in our human-robot collaboration setup, in which we would like the communication between the human operator and the robot to be as immediate as possible. For this reason, we would like to employ a network with fast inference speed, in order to develop a framework that could possibly work in real time. The original network takes as input a sequence of skeletons with 25 joints, and is made up of 10 blocks each containing a spatial graph shift convolution operation and an adaptive temporal shift operation. The original output layer contains 60 nodes to classify the NTU RGB+D dataset actions.

We modify this architecture to adapt it to our data as described in Section 5.2.1. For what concerns the pre-training with the *wholebody* skeletons sequences estimated on the NTU RGB+D dataset, the number of output neurons becomes 50, since 10 mutual actions are removed from the original dataset. Moreover, we define 4 different models each taking as input sequences with different numbers of joints, and in each case a different graph structure is specified. The four input formats for such models are the following:

- *wholebody*: consists in the full body graph with 39 joints as defined in Section 5.1.2 and illustrated in Figure 5.2 and replicated in Figure 6.7. The joint with identification number 0 of each hand is connected to the joint of the corresponding wrist, namely with the body joint number 7 for the left hand and with the body joint number 4 for the right hand

- *body*: in this case the input only contains the 15 body joints, with the body graph shown in Figure 5.2

- *hands*: the input consists in the 24 joints belonging to the two hands, and the graph considered for each hand is the one of Figure 5.2. To this graph, an additional edge is introduced, as shown in Figure 6.7, in order to connect the joints with ID 0 of the two hands

- *single hand*: in this last case, the input is made up of the 12 joints of a single hand, with the corresponding graph defined in Figure 5.2. We will consider two different networks with the same architecture for the *left hand* and the *right hand*.

Finally, for the classification of the 18 human-robot collaboration actions, the last layer of the four networks just described is modified to contain 18 neurons.

We highlight that a Shift-GCN performs in each block a number of shifting operations that is proportional to the dimension, i.e., the number of nodes, of the input graph. Hence, a network module taking as input a smaller graph is simpler and lighter with respect to a network with the same architecture having a larger input graph.

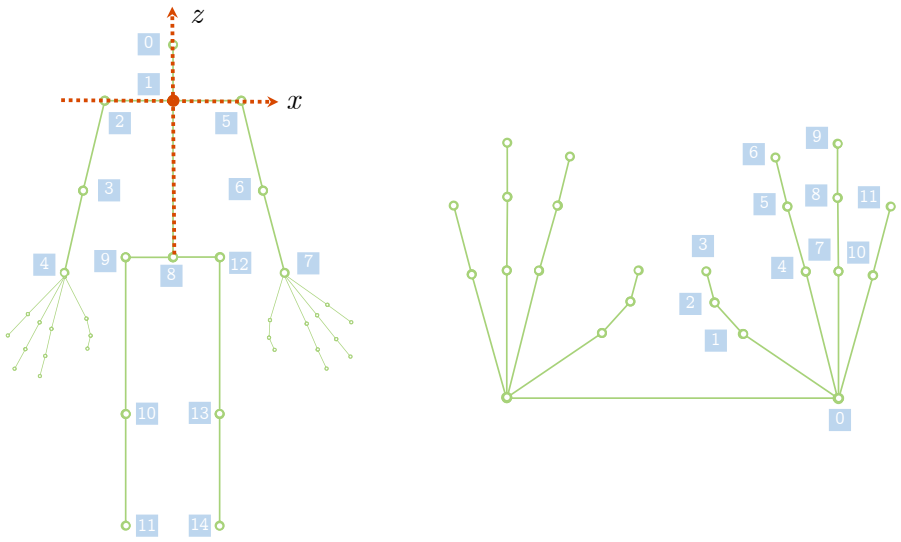


Figure 6.7: *Skeleton model: graph and reference system.*

6.4.1 Data preparation

The skeleton sequences of the dataset employed to train the model undergo some preprocessing operations before being fed to the network. In particular, the origin of reference system with respect to which the joints coordinates are expressed is shifted to a central joint of the skeleton. For the considered skeleton model, we choose as origin joint the neck, namely the one with ID 1, as illustrated in Figure 6.7. The z axis of the new reference system is taken parallel to the segment connecting the pelvis (ID=8) and the neck (ID=1) joints, while the x axis is considered parallel to the segment connecting the left shoulder joint (ID=5) with the right shoulder joint (ID=2). The newly defined reference system allows to consider all the performed movements as relative to the body and not to and external reference system. When only the hands are taken into account, the axes of the reference system are still defined with respect to the same body joints, while the origin is placed on the joint 0 of each hand (on the one of the right hand when both hands are considered). In this way, the hands movements do not depend on the absolute position of the hands in space, but the information regarding their orientation with respect to the rest of the body is maintained.

6.5 Pre-training on the NTU RGB+D dataset

As mentioned above, five separate networks were trained on the five joints subsets (*wholebody, body, hands, left hand, right hand*), using the skeleton data extracted by OpenPifPaf from the NTU RGB+D dataset. It was already observed that the number of valid samples obtained is considerably smaller with respect to the one of the original

dataset (see Table 6.2). Moreover, we highlight that the goal of pre-training the networks on the NTU RGB+D dataset is to learn features encoding general representations of the human body movements. For this reason, the predefined benchmarking instructions to split the dataset into training and test sets were not followed. In particular, these require that, for the *cross-view* setup, the actions recorded using the cameras with IDs 2 and 3 should be employed for training and the actions recorded with camera with camera 1 for testing, while for the *cross-subject* setup the actions performed by 20 subjects should be used for training and the remaining 20 for testing. For our experiments, we merge the two setups by taking as training set all the action samples recorded with cameras 2 and 3 joined with the actions recorded by camera 1 and performed by the 20 subjects considered for training in the *cross-subject* setup; as test set instead, we consider the remaining actions recorded with camera 1. In this way, we increase a little bit the size of the training set in order to learn features as general as possible during training, while only a smaller subset of actions is employed to validate the results and avoid overfitting. Moreover, it was observed that the placement of the three cameras recording the original NTU RGB+D dataset is such that the left hand is mostly visible by camera 1, the right hand is more visible by camera 3, while camera 2 is placed more or less in the middle. For this reason, only for what concerns the left hand, in the training set were included all the actions recorded using cameras 1 and 3 joined with the actions recorded by camera 2 and performed by the 20 training subjects of the *cross-subject* setup.

The results of the pre-training are reported in Table 6.3, that reports the accuracies obtained in training the five networks. The accuracy, or Top1 accuracy, refers to the percentage of correctly predicted actions in the test set, while the Top5 accuracy is the percentage of actions whose correct prediction falls in the five highest $softmax$ scores estimated by the network.

The first observation that can be made on the reported results is that the obtained accuracies are lower than the ones obtained using the Shift-GCN architecture on the original NTU RGB+D dataset [34]. However, we must first highlight that these results are not comparable with the ones obtained in [34] since a different different training and

Joints group	Top1 %	Top5 %
wholebody	59.44	81.34
body	90.40	97.68
hands	36.25	67.63
left hand	21.12	49.38
right hand	31.90	62.68

Table 6.3: *Accuracies of the networks trained on the skeleton sequences extracted from the NTU RGB+D dataset with OpenPifPaf.*

test sets are employed (this would have been true even if the original benchmarking setups were respected, given the reduced number of valid samples obtained with OpenPifPaf and the different number of actions considered). Moreover, the smaller dataset size leads to a worse generalisation capability for the network, negatively influencing our results.

Going more into detail for each specific case, it can be noticed that, in the case in which only the 15 body joints are analysed (i.e., *body* model), the network performs well in recognising the 50 actions, achieving an accuracy slightly higher than 90%. The performance of the network, however, rapidly deteriorates with the introduction of hands. In particular, the accuracy obtained in the *wholebody* case is more than 30 percentage points lower than the one of the *body* case. The reason for this can be traced back to the number of valid action samples, which for the *wholebody* analysis is almost half the one available for the *body* (see Table 6.2). Moreover, we can assume that, given the distance of the subjects from the cameras, the precision obtained for hand pose extraction is not very high, so that the introduction of hands poses estimated in this way makes the action recognition task more challenging. In any case, again with reference to the results obtained for the NTU-X dataset [82], we can assume that the low accuracies obtained in our results are strongly influenced by the limitations of the method used to estimate the full skeletons, complete of hands, from the original dataset, and not by the addition of the hand joints information per se.

For what concerns the training of the Shift-GCN using only hands information, it can be noticed that the accuracy obtained becomes even lower. However, this result is expected, since the actions contained in the NTU RGB+D dataset are everyday actions, most of which involve a limited use of hands. For this reason, we envisage that the network will find it more difficult to recognise actions from hands movements alone. Clearly, this behaviour is enhanced when only the joints of one hand are used to train the network. Moreover, it can be noticed that a better performance is obtained when considering the right hand joints rather than the left hand joints. This can be due to a higher percentage of right handed subjects in the population, making the movements performed with the right hand more significant than the ones of the left hand.

Although the above results are not optimal, we observe that the networks trained in this way are only used as a backbone for a subsequent transfer learning phase on the human-robot collaboration dataset acquired in our laboratory. The accuracy values obtained show that all networks learn some level of representations of the the actions from the training data. These might not be sufficient to recognise well the activities included in the NTU RGB+D dataset, but the coarse features learned in the first layers of the networks might still encode some useful information about the movements of the body parts considered. For this reason, even the networks that obtain low accuracies values are employed in the transfer learning experiments described in the following section.

6.6 Transfer learning on the *IAS-Lab Collaborative HAR* dataset

Starting from the pre-trained networks obtained in the previous section, we perform different transfer learning experiments with the skeleton data of the *IAS-Lab Collaborative HAR* dataset collected. We recall that the poses of our dataset were estimated using OpenPose, while the ones extracted from the NTU RGB+D dataset were obtained with OpenPifPaf. To the latter, two joints were added a posteriori, namely the neck and the pelvis, inferring their position as midpoints between the shoulders and between the hips respectively, in order to match the two skeleton models.

As already mentioned above, to fine-tune the models we detach from the five pre-trained networks the last linear layer with 50 nodes, responsible for classifying the NTU RGB+D actions, and substitute it with a linear layer composed of 18 neurons in order to classify the *IAS-Lab Collaborative HAR* actions. Then, we *freeze* the first layers of the pre-trained networks by blocking their weights update during the transfer learning training phase. As highlighted in Section 6.4, the Shift-GCN architecture consists in 10 main blocks, each containing spatial and temporal operations. In doing transfer learning, for each of the 10 blocks all the weights are considered either frozen or learnable, and different depths of learning, namely different numbers of learnable blocks, are analysed for each scenario.

In the following analysis, we first study the results obtained for each single module (*wholebody, body, hands, left hand, right hand*) separately. Then, we try ensembling the body and hands modules in different manners to improve the classification results: first the *body* and *hands* networks are fused together, then the *body* network is ensembled with the *left hand* network and the *right hand* network, as illustrated in the block scheme reported in Figure 6.8

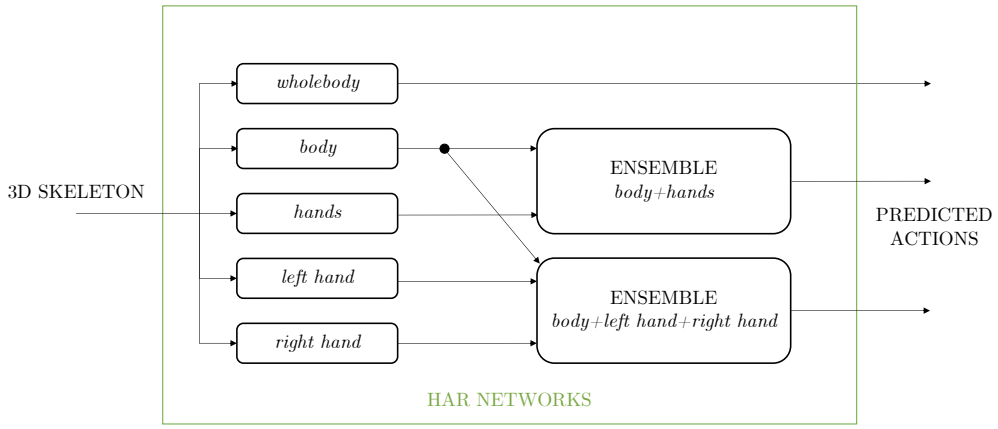


Figure 6.8: *Block scheme of the HAR networks ensembles.*

6.6.1 *IAS-Lab Collaborative HAR* data preparation

Before fine-tuning the pre-trained networks on the *IAS-Lab Collaborative HAR* dataset, the skeleton annotations obtained for this dataset are first reviewed. As done for the skeleton data extracted for the NTU RGB+D dataset, the number of valid skeleton sequences is counted, by eliminating all the sequences for which less than 3 complete skeleton acquisitions were made. The numbers of valid skeleton sequences obtained is reported in Table 6.4. It can be noticed that, for what concerns the 15 body joints, only one action sample is counted as not valid. The valid sequences for which the *wholebody* skeleton, comprehensive of body and hands, is captured are instead less than a half of the original 540 sequences collected. The number improves when the single hands joints are considered, namely when only the pose of one hand is fully estimated while neglecting the estimation of the other one. The valid skeleton sequences are then pre-processed as described in Section 6.4.1 before being fed to the network.

Joints group	Training	Validation	Total
wholebody	183	51	234
body	449	90	539
hands	184	51	235
left hand	328	69	397
right hand	301	69	370

Table 6.4: *Number of valid skeleton sequences for the* IAS-Lab Collaborative HAR *dataset collected.*

6.6.2 Transfer learning on the single networks modules

The first experiments conducted consist in training the last blocks of the five pre-trained networks on the newly collected dataset to recognise the new actions. In the following, we denote with ℓi, $i \in [1, 10]$ the 10 blocks of the Shift-GCN. The transfer learning experiments were conducted starting with learnable weights only for the blocks $\ell 9$ and $\ell 10$, and by unfreezing one more block at a time for subsequent experiments, to compare the performance of the networks for different depths of trainable layers in the network, where the depth here refers to the number of blocks with learnable weights. The optimiser used to update the network weights is a SGD optimiser with initial learning rate equal to $l_r = 0.1$ and momentum $m = 0.9$. Moreover, the learning rate was adjusted as in the original code at some predefined training epochs indicated in the list `step`, namely at each epoch contained in the mentioned list the learning rate was reduced of a factor 10. The training algorithm runs for 100 epochs, and the model achieving the highest validation accuracy, i.e., the lowest validation error, is saved. In Table 6.5 are

Model	l_r	`step`	m	bs	learnable Shift-GCN blocks	epochs
wholebody	0.1	[]	0.9	24	$\ell 7$, $\ell 8$, $\ell 9$, $\ell 10$	7
body	0.1	[30]	0.9	12	$\ell 9$, $\ell 10$	47
hands	0.1	[30, 50, 70]	0.9	12	$\ell 8$, $\ell 9$, $\ell 10$	76
left hand	0.1	[]	0.9	12	$\ell 6$, $\ell 7$, $\ell 8$, $\ell 9$, $\ell 10$	12
right hand	0.1	[]	0.9	24	$\ell 6$, $\ell 7$, $\ell 8$, $\ell 9$, $\ell 10$	14

Table 6.5: *Hyperparameters used in the transfer learning training phase.*

reported more in detail the hyperparameters for which the best results are obtained for the five different networks. These include the batch size bs used in the training phase, the initial learning rate l_r and momentum m of the optimiser, the list `step` of the epochs at which the learning rate is adjusted, the blocks of the Shift-GCN network that contain learnable weights and the number of the training epoch at which the lowest validation error is achieved. It can be observed that, in the cases for which the pre-training on the NTU RGB+D dataset led to lower accuracy values, a higher number of layers needs to be trained in the transfer learning phase. This happens because in the pre-training phase the networks are not able to learn sufficiently representative features, and more weights of the intermediate layers need to be updated to improve the results.

The accuracies obtained on the validation set using the above parameters are reported in Table 6.6. For each network, the Top1 accuracy and the Top3 accuracy are indicated. Note that in this case the Top3 accuracy is analysed, instead of the Top5 as in the case of the NTU RGB+D dataset, since it is considered more significant, being the number of classes contained in our dataset is smaller (18 instead of 50).

These results highlight that the best performance is obtained, also in this case, with the *body* network, for which a 62.22% accuracy is reached on the validation set. A relatively high percentage of actions is misclassified, but in this case the misclassifications

Model	Top1 %	Top3 %
wholebody	44.00	54.00
body	62.22	86.67
hands	50.00	64.00
left hand	59.42	73.91
right hand	59.42	71.01

Table 6.6: *TopN accuracies of the single modules after fine-tuning them on the* IAS-Lab Collaborative HAR *dataset.*

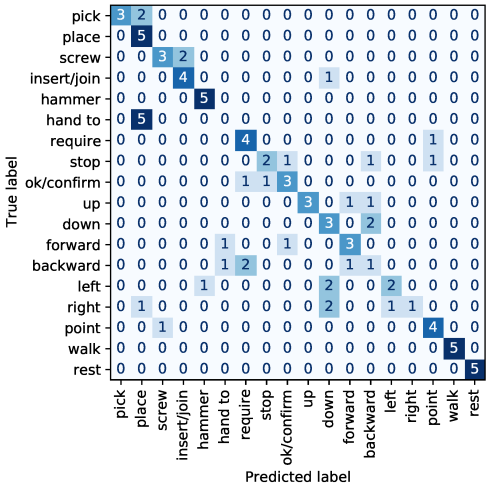


Figure 6.9: *Confusion matrix obtained on the validation set of the* IAS-Lab Collaborative HAR *dataset for the* body *network.*

reflect the importance of the use of hands in identifying the actions of the *IAS-Lab Collaborative HAR* dataset. Figure 6.9 shows the confusion matrix obtained on the validation set using the *body* network, and highlights how the majority of the actions that are wrongly classified belong to the *communication gestures* group, namely to the group of actions for which the hands are supposedly more informative. On the other hand, this results also highlight the importance of the body movements even in the recognition of gestures.

The unsatisfactory results obtained with the *wholebody* and the *hands* networks are mostly related to the low number of valid sequences available in these cases, which do not allow the networks to properly learn how to distinguish the different actions. Indeed, better results are obtained with the networks that recognise the actions starting from single hands, ensuring the informativeness of the hands joints for the analysed dataset. It is worth noting that, when a subject executes an action belonging to the dataset that actively involves the use of hands, two possibilities emerge:

- only one of the two hands performs the action while the other one is kept in a resting position,

- both hands are used to complete the action.

This is particularly relevant when recognising actions from a single hand, since even if the rest of the body is involved in an activity, the hand might be resting and completely irrelevant to the recognition process. For this reason, in the single hands analysis, the action samples were re-labelled in order to mark as belonging to the REST class all the sequences in which the considered hand is not actually performing the action.

From what discussed above, two fundamental problems emerge:

1. The difficulty in extracting valid skeletons sequences in the *wholebody* case, which strongly limits the action recognition outcomes
2. The importance of both hands and body joints in the recognition of the defined actions. Single networks receiving only *body* or only *hands* joints struggle to obtain good classification results.

We try to solve these issues by joining the networks that perform action recognition starting only from the body joints and the ones that only exploit the hands joints information, by means of a score-level fusion, as discussed in the following sections.

6.6.3 *Body+hands* ensemble

The first fusion method proposed consists in joining the results of the *body* and the *hands* networks. This is simply done by taking as new, final scores for the action classes a weighted sum of the two networks outputs. More in detail, if we denote with $\mathbf{o}_b \in [0,1]^{18}$ the output of the *body* network and with $\mathbf{o}_h \in [0,1]^{18}$ the output of the *hands* network, the output of the ensembled network is obtained as

$$\mathbf{o}_{b,h} = \alpha_b \, \mathbf{o}_b + \alpha_h \, \mathbf{o}_h, \quad \text{with } \alpha_o + \alpha_h = 1 \tag{6.2}$$

Then, the predicted action label is obtained by taking the argmax of the new output scores, namely $\ell_{pred} = argmax(\mathbf{o}_{b,h})$. In this case, the highest accuracy is achieved by setting $\alpha_b = \alpha_h = 0.5$, namely when body and hands joints information equally contribute to the recognition process. The accuracy result obtained is reported in Table 6.7, along with the outcomes achieved with the single networks on the same validation dataset, which contains an intersection of the valid body samples and the valid hands samples (coinciding in this case with all the hands samples). These show an improvement with respect to both the previously analysed models.

Model	Top1 %	Top3 %
body	56.00	84.00
hands	50.00	64.00
body+hands, $w_b = w_h = 0.5$	68.00	90.00

Table 6.7: *Body+hands ensemble accuracies on the* IAS-Lab Collaborative HAR *dataset.*

6.6.4 *Body+left hand+right hand* ensemble

Another interesting case consists in joining the movement information coming separately from the two hands with the one of the body. In this case, a slightly more

Algorithm 1 BODY+LEFT HAND+RIGHT HAND ENSEMBLE

Input: skeleton sequence s, output scores $\mathbf{o}_b$, $\mathbf{o}_{lh}$, $\mathbf{o}_{rh}$, predicted labels ℓ_b, ℓ_{lh}, ℓ_{rh}.

v_{flag}= CHECKVALIDITY(s)
if v_{flag} is *wholebody-valid* **then**
    if ℓ_{lh} =REST and ℓ_{rh} ≠REST **then**
        $\alpha_b = \alpha_{rh} = 0.5$
        $\mathbf{o}_{b,lh,rh} = \alpha_b\,\mathbf{o}_b + \alpha_{rh}\mathbf{o}_{rh}$
    else if ℓ_{lh} ≠REST and ℓ_{rh} =REST **then**
        $\alpha_b = 0.6$, $\alpha_{lh} = 0.4$
        $\mathbf{o}_{b,lh,rh} = \alpha_b\,\mathbf{o}_b + \alpha_{lh}\mathbf{o}_{lh}$
    else
        $\alpha_b = \alpha_{rh} = 0.358$, $\alpha_{lh} = 0.284$
        $\mathbf{o}_{b,lh,rh} = \alpha_b\,\mathbf{o}_b + \alpha_{lh}\mathbf{o}_{lh} + \alpha_{rh}\mathbf{o}_{rh}$
else if v_{flag} is *left-hand-valid* **then**
    if ℓ_{lh} =REST and ℓ_b ≠REST **then**
        $\mathbf{o}_{b,lh,rh} = \mathbf{o}_b$
    else
        $\alpha_b = 0.6$, $\alpha_{lh} = 0.4$
        $\mathbf{o}_{b,lh,rh} = \alpha_b\,\mathbf{o}_b + \alpha_{lh}\mathbf{o}_{lh}$
else if v_{flag} is *right-hand-valid* **then**
    if ℓ_{rh} =REST and ℓ_b ≠REST **then**
        $\mathbf{o}_{b,lh,rh} = \mathbf{o}_b$
    else
        $\alpha_b = \alpha_{rh} = 0.5$
        $\mathbf{o}_{b,lh,rh} = \alpha_b\,\mathbf{o}_b + \alpha_{rh}\mathbf{o}_{rh}$
return $\ell_{pred} = argmax(\mathbf{o}_{b,lh,rh})$

complex algorithm is used to derive the ensemble output scores. Starting from a skeleton sequence, we first separate the cases in which the full skeleton estimate is available from the ones in which only the body and one hand joints are fully estimated along the sequence. Then, in both cases, the predicted labels for each single network module are evaluated. If the predicted label of one hand is REST and is different from the labels predicted by the other networks, the scores of such network are discarded from the analysis. We assume in this case that the hand action labelled as REST falls in the cases of actions in which only one of the two hands is actively moving, while the other one stands still and is irrelevant in terms of action recognition. Once a "resting hand" is detected, the scores of the remaining networks are fused together as in the previous section. A more detailed pseudocode of this ensembling procedure is reported in Algorithm 1, in which are also reported the values of the weights employed to sum together the different scores, whose optimal values was found by trial and error. In the algorithm, the flag *wholebody-valid* means that the input skeleton sequence has complete annotations of the body and of both hands, the flags *left-hand-valid* and *right-hand-valid* indicate instead

Model	Top1 %	Top3 %
wholebody	44.00	54.00
body	62.22	86.67
hands	50.00	64.00
left hand	59.42	73.91
right hand	59.42	71.01
body+hands, $w_b = w_h = 0.5$	68.00	90.00
body+left hand+right hand, $w_b = w_{rh} = 0.358$, $w_{lh} = 0.284$	76.54	87.65

Table 6.8: *Body+left hand+right hand ensemble accuracies on the* IAS-Lab Collaborative HAR *dataset.*

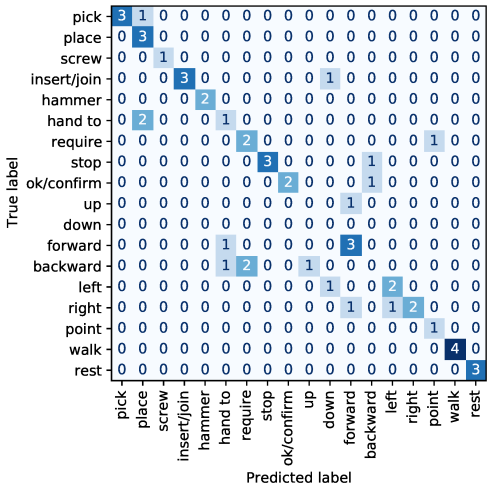
that the annotations of the skeleton sequence are complete only for the body and for the left or right hand respectively. We can observe that the weights employed give the same importance to the body and the right hand movements, while the left hand information is always weighted slightly less. Indeed, these are the weights for which a higher accuracy is reached, and can be explained by the fact that the majority of subjects in the collected dataset are right-handed and as a consequence tend to execute the actions with their right hand.

The accuracy obtained for this ensembling method and the comparison with the single networks and with the fusion discussed in the preceding section are shown in Table 6.8. It can be noticed that this way of fusing hands and body networks leads to an improved action recognition performance with respect to all the previously analysed cases. This method also allows to discard a lower number of skeleton acquisitions, since it is sufficient to have the full estimation of the pose of the body and of just one of the two hands.

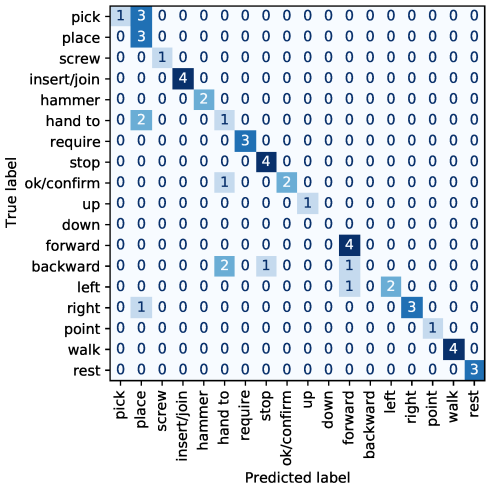
Figure 6.10 shows a comparison of the confusion matrices obtained for the two fusion methods just explained. In particular, we report the confusion matrix obtained by evaluating the *body+left hand+right hand* ensemble first on the same validation set of the *body+hands* ensemble, which contains only samples for which both hands are correctly estimated, and then on the extended validation set that includes all the action samples considered valid for the new ensemble. These matrices highlight both the difference in the number of samples to which the two methods are able to assign a label, as well as the improved performance of the *body+left hand+right hand* ensemble. It can also be noticed that for some actions such as DOWN, none of the collected recordings of the validation subject has a full skeleton estimation, with the consequence of this action never being classified with the *body+hands* fusion. However, when considering separately the two hands, only one out of the 5 samples is not validly estimated, and the action is correctly classified 3 times. Moreover, we remark that, for the collected dataset, some actions are particularly challenging to recognise as they can be performed in several different

ways. In particular, from Figure 6.10c it can be seen that most of the prediction errors concern the *gestures* categories. For example, one of the most misclassified actions with the *body+left hand+right hand* ensemble is BACKWARD, which is confused with STOP, FORWARD and HAND TO. Indeed, these actions can all be performed with similar hand and body movements, which explains the wrong predictions.

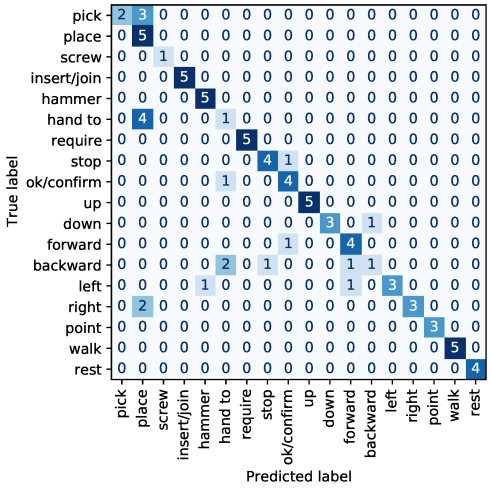
An observation that can be made on the two fusion methods studied is that, in general, having a higher amount of information, namely more estimated joints, should lead to better recognition performances. However, some actions can also be successfully recognised even when some information is missing. We would expect that the main advantage of jointly considering the hands movements is that, for certain actions, the relative movement of the two hands can be significant in describing the action. On the other hand, this movement is also encoded by the body joints, namely by how the arms and wrists move. Then, the single hands gestures can be studied separately without losing information about their relative position, and smaller network modules can be fused together. As mentioned in Section 5.2.2, splitting data in disjoint groups and fusing multiple classifiers together can help reducing the errors and improving the overall classification performance.



(a) *body+hands*



(b) *body+left hand+right hand (body+hands val. set)*



(c) *body+left hand+right hand*

Figure 6.10: *Confusion matrices obtained with the two networks ensembles (*body+hands *and* body+left hand+right hand*) on the validation subject of the* IAS-Lab Collaborative HAR *dataset.*

Chapter 7

Conclusions

In this work, we investigated a skeleton-based deep learning method for action and gesture recognition in a human-robot collaboration scenario. Unlike other HRC-related works in literature, we do not focus on a specific application, rather we try to define a generic framework within which several different HRC scenarios can be enclosed.

To this purpose, a new dataset (*IAS-Lab Collaborative HAR*) containing a careful selection of general actions related to the field of human-robot collaboration was first collected. In particular, we tried to include in it as many actions and gestures as possible that appeared significant to recognise in a potential HRC setting. The dataset contains the 3D annotations of the skeletons, comprehensive of body and hands joints, of the subjects performing the actions.

A transfer learning enabled method to recognising these actions was then presented. In particular, we employed as backbone the Shift-GCN model presented in [34], pre-trained on full-body skeleton annotations retrieved by us on the large scale action recognition dataset NTU RGB+D. The action recognition pipeline adopted only exploits the estimated skeleton joints, in order to be as independent as possible from the surrounding context and from any tools or objects that might be included in the execution of the movements of interest. Moreover, besides training a classifier to recognise actions using data relative to the *wholebody* skeleton model, we considered splitting the pose information into different, independent modules, i.e., we first separated the body and hands information, then we proposed an additional splitting of the two hands into separate modules. For each subset of joints, a network was pre-trained on the corresponding sequences of joints annotated for the NTU RGB+D dataset, and transfer learning was then applied to learn to classify the actions of our *IAS-Lab Collaborative HAR* dataset. Subsequently, two ways of ensembling these networks using a score-level fusion were proposed (*body+hands* and *body+left hand+right hand*). Dividing the information into disjoint modules has a dual purpose. First, it allows to understand more in detail the role of each analysed body part for the recognition of an action or gesture. Secondly, it helps improving the classification results, leveraging on the fact that learning the same task

separately from different data has proven over time to be an effective way of reducing classification errors and enhancing the overall performance.

The results of the experiments carried out on the *IAS-Lab Collaborative HAR* dataset show indeed the importance of incorporating both body and hands information in the understanding of human actions in a HRC context, in which different levels of activities are relevant, from contextual movements, to assembly operations, to collaborative gestures. They also confirm the effectiveness of ensembling smaller modules using a simple score-level fusion algorithm to obtain an improved classification performance. Indeed, the results obtained with the two fusion methods lead to an improvement compared to single networks: with the *body+hands* ensemble, a Top1 accuracy of 68.00% and a Top3 accuracy of 90.00% were reached, while using the *body+left hand+right hand* ensemble the best results were achieved, namely we obtained a Top1 accuracy of 76.54% and a Top3 accuracy of 87.65%.

7.1 Future developments

From the analysis on the experimental results, some critical aspects and limitations of the adopted approach have emerged. In the following, we review them and try to propose some possible improvements and future developments of this project.

7.1.1 Robust pose estimation and multi-camera networks

One of the main problems highlighted in the previous sections is that the overall performance of the system depends on the goodness of the skeleton features employed. This issue arises in the pose estimation processes both concerning the NTU RGB+D dataset with OpenPifPaf and the *IAS-Lab Collaborative HAR* dataset collection. In both cases, 3D joints are inferred from a single frame by first extracting the 2D information from the RGB data and by projecting the obtained keypoints on the depth frames to retrieve the third spatial coordinate. When estimating the body poses on the NTU RGB+D dataset, priority was given to the quality of the extracted data rather than the speed of estimation, in order to have as many valid sequences as possible to pre-train the networks, whereas in the acquisition of the new *IAS-Lab Collaborative HAR* dataset we tried to recreate a realistic setup in which the skeleton data were estimated and acquired in real time.

To improve the results of whole-body pose estimation on the NTU RGB+D dataset, a more sophisticated way to retrieve the skeletons from RGB-D data could be employed, such as the one proposed for the already mentioned NTU-X dataset [82]. In particular, in this work two 3D pose estimation libraries are exploited to find the full skeletons, namely SMPL-X [93] and Ex-Pose [94]. These might require longer estimation times, but are able to estimate the full skeleton sequences on most of the original NTU RGB+D action recordings, and the results reported in [82] prove the effectiveness of introducing hands

joints to improve action recognition performances.

For what concerns the real-time estimation and acquisition of 3D skeleton sequences, the main limit of the employed system lies instead in the use of a single camera. Indeed, when a subject is framed with a single sensor, it is very likely that some of its body parts are occluded, either by the subject's own body or by other objects in the work environment. Creating a work cell with a multi-camera network might reduce the probability of occlusions. Moreover, fusing together the body pose estimation results obtained with each sensor in the network would make the 3D skeleton estimates more robust and more likely to contain all the body joints required in the action recognition pipeline proposed. A further improvement in the estimation of the sequence of body joints could consist in tracking the estimated body joints, instead of just performing the estimation on the single frames, in order to obtain more complete skeleton sequences and to make the system more robust to fast body movements, which can easily lead to missing joints in the single-frame estimation. Finally, in using multiple sensors, it could be possible to dedicate one of them to frame hands from a closer point of view, in order to obtain a more precise estimate of their joints.

7.1.2 Dealing with partial inputs

The problem of the dependence of action recognition methods on the robustness of body pose features can also be seen from a different point of view. Indeed, even though a richer information certainly helps increasing the robustness of the learning and classification procedures, this issue also stems from the fact that the networks employed for skeleton-based human action recognition, such as graph neural networks, require complete inputs to correctly learn how to classify the actions and/or gestures. A learning algorithm able to deal with missing input features could help relax the requirements on the completeness of input skeleton sequences. In [95], an approach is proposed to adapt graph convolutional networks to deal with missing graph features, by representing the missing data through a Gaussian Mixture Model (GMM), exploited to compute the expected activation of neurons in the first hidden layer of the GCN, while keeping the other layers of the network unchanged.

7.1.3 Dataset and training improvements and real-case applications

Another limit of the presented framework lies in the restricted number of samples that we have been able to acquire for the *IAS-Lab Collaborative HAR* dataset. To improve the quality and reliability of our results, it could be useful to extend the collected dataset with new samples and/or to re-acquire the skeleton sequences using a more robust setup, such as a multi-camera network as explained above. Another interesting future development direction would be to test the proposed system for a real use case. We have mentioned that most of the works on action recognition for human-robot collaboration focus on

specific scenarios, in which usually tools or objects intended for the particular application are employed. On the contrary, our work tries to frame human-robot collaboration in a general way. The main idea is to either employ the proposed system alone in a given HRC scenario or, when needed, to couple it to a simple object recognition network in order to apply it to a more specific use-case.

We can also observe that, in light of the results obtained, which suggest that using the fusion of separate networks learning from different joints sets can lead to better classification results, a different approach in pre-training the networks could be adopted. In particular, we have highlighted that hands are not particularly informative for many of the actions included in the NTU RGB+D dataset. However, we have used the skeleton annotations obtained on this dataset in all the five different networks analysed in order to have better comparable results. Having established that learning body and hands movements separately is more effective that feeding sequences of *wholebody* skeletons to the network, we could consider employing a different dataset for the pre-training of the hands network, such as a dataset employed for sign-language recognition [12], with the aim of learning better representations for the hands movements in the early networks layers.

One final remark regards the training of the neural networks carried out in the experimental phase, during which only a few combinations of hyperparameters such as the batch size, the type of optimiser used and its learning rate, etc., were inspected. It is likely that a more exhaustive search for the optimal hyperparameters, as well as the implementation of cross-validation techniques, could also lead to improved and more accurate results.

Bibliography

[1] Wang Bo et al. "Skeleton-based violation action recognition method for safety supervision in the operation field of distribution network based on graph convolutional network". In: *CSEE Journal of Power and Energy Systems* (2021).

[2] Kun Liu et al. "Enhancing anomaly detection in surveillance videos with transfer learning from action recognition". In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 4664–4668.

[3] Caetano Mazzoni Ranieri et al. "Activity Recognition for Ambient Assisted Living with Videos, Inertial Units and Ambient Sensors". In: *Sensors* 21.3 (2021), p. 768.

[4] Svitlana Antoshchuk, Mykyta Kovalenko, and Jürgen Sieck. "Gesture recognition-based human–computer interaction interface for multimedia applications". In: *Digitisation of Culture: Namibian and International Perspectives*. Springer, 2018, pp. 269–286.

[5] Fatemeh Mohammadi Amin et al. "A mixed-perception approach for safe human–robot collaboration in industrial automation". In: *Sensors* 20.21 (2020), p. 6347.

[6] Md Al-Amin et al. "Action recognition in manufacturing assembly using multimodal sensor fusion". In: *Procedia Manufacturing* 39 (2019), pp. 158–167.

[7] Guilherme Maeda et al. "Phase estimation for fast action recognition and trajectory generation in human–robot collaboration". In: *The International Journal of Robotics Research* 36.13-14 (2017), pp. 1579–1594.

[8] K Martin Sagayam and D Jude Hemanth. "Hand posture and gesture recognition techniques for virtual reality applications: a survey". In: *Virtual Reality* 21.2 (2017), pp. 91–107.

[9] Md Atiqur Rahman Ahad, Anindya Das Antar, and Omar Shahid. "Vision-based Action Understanding for Assistive Healthcare: A Short Review." In: *CVPR Workshops*. 2019, pp. 1–11.

[10] Md Zia Uddin et al. "A body sensor data fusion and deep recurrent neural network-based behavior recognition approach for robust healthcare". In: *Information Fusion* 55 (2020), pp. 105–115.

[11] Djamila Romaissa Beddiar et al. "Vision-based human activity recognition: a survey". In: *Multimedia Tools and Applications* 79.41 (2020), pp. 30509–30555.

[12] Rachel McKee et al. *NZ Sign Language Exercises*. URL: https://www.wgtn.ac.nz/llc/llc_resources/nzsl/.

[13] Shuai Tang, Dominic Roberts, and Mani Golparvar-Fard. "Human-object interaction recognition for automatic construction site safety inspection". In: *Automation in Construction* 120 (2020), p. 103356.

[14] Tianmin Shu et al. "Joint inference of groups, events and human roles in aerial videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4576–4584.

[15] *Xsens Motion Capture Systems*. URL: https://www.xsens.com/motion-capture.

[16] Eloise Matheson et al. "Human–robot collaboration in manufacturing applications: a review". In: *Robotics* 8.4 (2019), p. 100.

[17] Alina Roitberg et al. "Human activity recognition in the context of industrial human-robot interaction". In: *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE. 2014, pp. 1–10.

[18] Hongyi Liu et al. "Towards robust human-robot collaborative manufacturing: Multimodal fusion". In: *IEEE Access* 6 (2018), pp. 74762–74771.

[19] Kai Zhang et al. "Human Motion Recognition for Industrial Human-Robot Collaboration based on a Novel Skeleton Descriptor". In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2020, pp. 404–410.

[20] Ziyang Song et al. "Attention-oriented action recognition for real-time human-robot interaction". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 7087–7094.

[21] Sara Sheikholeslami, AJung Moon, and Elizabeth A Croft. "Cooperative gestures for industry: Exploring the efficacy of robot hand configurations in expression of instructional gestures for human–robot interaction". In: *The International Journal of Robotics Research* 36.5-7 (2017), pp. 699–720.

[22] Panagiota Tsarouchi et al. "On a human-robot collaboration in an assembly cell". In: *International Journal of Computer Integrated Manufacturing* 30.6 (2017), pp. 580–589.

[23] Wenjin Tao et al. "A self-aware and active-guiding training & assistant system for worker-centered intelligent manufacturing". In: *Manufacturing letters* 21 (2019), pp. 45–49.

[24] Wenjin Tao, Ming C Leu, and Zhaozheng Yin. "Multi-modal recognition of worker activity for human-centered intelligent manufacturing". In: *Engineering Applications of Artificial Intelligence* 95 (2020), p. 103868.

[25] Zitong Liu et al. "Deep learning-based human motion prediction considering context awareness for human-robot collaboration in manufacturing". In: *Procedia CIRP* 83 (2019), pp. 272–278.

[26] Ali Ghadirzadeh et al. "Human-centered collaborative robots with deep reinforcement learning". In: *IEEE Robotics and Automation Letters* 6.2 (2020), pp. 566–571.

[27] Takuya Kobayashi et al. "Fine-grained action recognition in assembly work scenes by drawing attention to the hands". In: *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE. 2019, pp. 440–446.

[28] Peng Wang et al. "Deep learning-based human motion recognition for predictive context-aware human-robot collaboration". In: *CIRP annals* 67.1 (2018), pp. 17–20.

[29] Eva Coupeté, Fabien Moutarde, and Sotiris Manitsaris. "Multi-users online recognition of technical gestures for natural human–robot collaboration in manufacturing". In: *Autonomous Robots* 43.6 (2019), pp. 1309–1325.

[30] Qianqian Xiong et al. "Transferable two-stream convolutional neural network for human action recognition". In: *Journal of Manufacturing Systems* 56 (2020), pp. 605–614.

[31] Chengjun Chen et al. "Repetitive assembly action recognition based on object detection and pose estimation". In: *Journal of Manufacturing Systems* 55 (2020), pp. 325–333.

[32] Matteo Melchiorre et al. "Vision-based control architecture for human–robot hand-over applications". In: *Asian Journal of Control* 23.1 (2021), pp. 105–117.

[33] Wenjin Tao et al. "Real-time assembly operation recognition with fog computing and transfer learning for human-centered intelligent manufacturing". In: *Procedia Manufacturing* 48 (2020), pp. 926–931.

[34] Ke Cheng et al. "Skeleton-based action recognition with shift graph convolutional network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 183–192.

[35] Amir Shahroudy et al. "Ntu rgb+ d: A large scale dataset for 3d human activity analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1010–1019.

[36] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor". In: *2015 IEEE International conference on image processing (ICIP)*. IEEE. 2015, pp. 168–172.

[37] Earnest Paul Ijjina and Krishna Mohan Chalavadi. "Human action recognition in RGB-D videos using motion sequence information and deep learning". In: *Pattern Recognition* 72 (2017), pp. 504–516.

[38] Annalisa Franco, Antonio Magnani, and Dario Maio. "A multimodal approach for human activity recognition based on skeleton and RGB data". In: *Pattern Recognition Letters* 131 (2020), pp. 293–299.

[39] Runwei Ding et al. "Combining adaptive hierarchical depth motion maps with skeletal joints for human action recognition". In: *IEEE Access* 7 (2018), pp. 5597–5608.

[40] Liangchen Song et al. "Human pose estimation and its application to action recognition: A survey". In: *Journal of Visual Communication and Image Representation* 76 (2021), p. 103055.

[41] Zhe 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.

[42] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. "OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association". In: *arXiv preprint arXiv:2103.02440* (2021).

[43] Raviteja Vemulapalli, Felipe Arrate, and Rama 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.

[44] Huy Hieu Pham et al. "Spatio–temporal image representation of 3D skeletal movements for view-invariant action recognition with deep convolutional neural networks". In: *Sensors* 19.8 (2019), p. 1932.

[45] Songyang Zhang et al. "Fusing geometric features for skeleton-based action recognition using multilayer LSTM networks". In: *IEEE Transactions on Multimedia* 20.9 (2018), pp. 2330–2343.

[46] Yanshan Li, Rongjie Xia, and Xing Liu. "Learning shape and motion representations for view invariant skeleton-based action recognition". In: *Pattern Recognition* 103 (2020), p. 107293.

[47] Sijie Yan, Yuanjun Xiong, and Dahua Lin. "Spatial temporal graph convolutional networks for skeleton-based action recognition". In: *Thirty-second AAAI conference on artificial intelligence*. 2018.

[48] Ziyu Liu et al. “Disentangling and unifying graph convolutions for skeleton-based action recognition”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 143–152.

[49] Yuxin Chen et al. “Channel-wise topology refinement graph convolution for skeleton-based action recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13359–13368.

[50] Lei Shi et al. “Two-stream adaptive graph convolutional networks for skeleton-based action recognition”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 12026–12035.

[51] Christoph Feichtenhofer. “X3d: Expanding architectures for efficient video recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 203–213.

[52] Joao Carreira and Andrew 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.

[53] Xianhe Wen, Heping Chen, and Qi Hong. “Human assembly task recognition in human-robot collaboration based on 3D CNN”. In: *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE. 2019, pp. 1230–1234.

[54] Karen Simonyan and Andrew Zisserman. “Two-stream convolutional networks for action recognition in videos”. In: *arXiv preprint arXiv:1406.2199* (2014).

[55] Jiahui Yu et al. “A discriminative deep model with feature fusion and temporal attention for human action recognition”. In: *IEEE Access* 8 (2020), pp. 43243–43255.

[56] Amin Ullah et al. “Action recognition in video sequences using deep bi-directional LSTM with CNN features”. In: *IEEE access* 6 (2017), pp. 1155–1166.

[57] Kirill Gavrilyuk et al. “Actor-transformers for group activity recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 839–848.

[58] Honghai Liu et al. “Study of human action recognition based on improved spatio-temporal features”. In: *Human Motion Sensing and Recognition*. Springer, 2017, pp. 233–250.

[59] Muhammad Attique Khan et al. “Hand-crafted and deep convolutional neural network features fusion and selection strategy: an application to intelligent human action recognition”. In: *Applied Soft Computing* 87 (2020), p. 105986.

[60] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. “Spatial temporal transformer network for skeleton-based action recognition”. In: *International Conference on Pattern Recognition*. Springer. 2021, pp. 694–701.

[61] Yansong Tang et al. “Deep progressive reinforcement learning for skeleton-based action recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5323–5332.

[62] Gonçalo S Martins, Luis Santos, and Jorge Dias. “The GrowMeUp project and the applicability of action recognition techniques”. In: *Third workshop on recognition and action for scene understanding (REACTS). Ruiz de Aloza*. 2015.

[63] Chiara Filippini et al. “Facilitating the child–robot interaction by endowing the robot with the capability of understanding the child engagement: The case of mio amico robot”. In: *International Journal of Social Robotics* 13.4 (2021), pp. 677–689.

[64] Bo Li, Baoxing Bai, and Cheng Han. “Upper body motion recognition based on key frame and random forest regression”. In: *Multimedia Tools and Applications* 79.7 (2020), pp. 5197–5212.

[65] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[66] Sheng Jin et al. “Whole-body human pose estimation in the wild”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 196–214.

[67] Mykhaylo Andriluka et al. “2D Human Pose Estimation: New Benchmark and State of the Art Analysis”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014.

[68] Leonid Sigal, Alexandru O Balan, and Michael J Black. “Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion”. In: *International journal of computer vision* 87.1-2 (2010), p. 4.

[69] Catalin Ionescu et al. “Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments”. In: *IEEE transactions on pattern analysis and machine intelligence* 36.7 (2013), pp. 1325–1339.

[70] Dushyant Mehta et al. “Monocular 3d human pose estimation in the wild using improved cnn supervision”. In: *2017 international conference on 3D vision (3DV)*. IEEE. 2017, pp. 506–516.

[71] Yangang Wang, Cong Peng, and Yebin Liu. “Mask-Pose Cascaded CNN for 2D Hand Pose Estimation From Single Color Image”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.11 (2019), pp. 3258–3268. DOI: 10.1109/TCSVT.2018.2879980.

[72] Jonathan Tompson et al. “Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks”. In: *ACM Transactions on Graphics* 33 (Aug. 2014).

[73] Shanxin Yuan et al. “Depth-based 3d hand pose estimation: From current achievements to future goals”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2636–2645.

[74] Christian Zimmermann et al. “FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape From Single RGB Images”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[75] Tomas Simon et al. “Hand keypoint detection in single images using multiview bootstrapping”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 1145–1153.

[76] Junting Dong et al. “Fast and robust multi-person 3d pose estimation from multiple views”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7792–7801.

[77] Karim Iskakov et al. “Learnable triangulation of human pose”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7718–7727.

[78] Zhe Zhang et al. “AdaFuse: Adaptive Multiview Fusion for Accurate Human Pose Estimation in the Wild”. In: *International Journal of Computer Vision* 129.3 (2021), pp. 703–718.

[79] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. “Action recognition based on a bag of 3d points”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE. 2010, pp. 9–14.

[80] Jiang Wang et al. “Cross-view action modeling, learning and recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2649–2656.

[81] Jun Liu et al. “Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding”. In: *IEEE transactions on pattern analysis and machine intelligence* 42.10 (2019), pp. 2684–2701.

[82] Neel Trivedi, Anirudh Thatipelli, and Ravi Kiran Sarvadevabhatla. “NTU-X: An enhanced large-scale dataset for improving pose-based recognition of subtle human actions”. In: *Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing*. 2021, pp. 1–9.

[83] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. “Skeleton-based dynamic hand gesture recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 1–9.

[84] Guillermo Garcia-Hernando et al. “First-person hand action benchmark with rgb-d videos and 3d hand pose annotations”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 409–419.

[85] William L Hamilton. “Graph representation learning”. In: *Synthesis Lectures on Artifical Intelligence and Machine Learning* 14.3 (2020), pp. 1–159.

[86] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[87] Bichen Wu et al. "Shift: A zero flop, zero parameter alternative to spatial convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9127–9135.

[88] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[89] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[90] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[91] Wei Zhang et al. "STA-GCN: two-stream graph convolutional network with spatial–temporal attention for hand gesture recognition". In: *The Visual Computer* 36.10 (2020), pp. 2433–2444.

[92] Michael P Perrone and Leon N Cooper. *When networks disagree: Ensemble methods for hybrid neural networks*. Tech. rep. Brown Univ Providence Ri Inst for Brain and Neural Systems, 1992.

[93] Georgios Pavlakos et al. "Expressive body capture: 3d hands, face, and body from a single image". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 10975–10985.

[94] Vasileios Choutas et al. "Monocular expressive body regression through body-driven attention". In: *European Conference on Computer Vision*. Springer. 2020, pp. 20–40.

[95] Hibiki Taguchi, Xin Liu, and Tsuyoshi Murata. "Graph convolutional networks for graphs containing missing features". In: *Future Generation Computer Systems* 117 (2021), pp. 155–168.